TITLE: Bayesian Semiparametric Methods with Quantile Functions: a Proposal with Applications to Prediction

AUTHOR: Venturini Sergio

TUTOR: Francesco Billari – Francesca Dominici

ABSTRACT: This thesis is about *density modeling* and *quantile data analysis* for skewed data. It aims at introducing two semiparametric Bayesian models for the estimation of tail probabilities for skewed distributions.

The first model is based on a mixture of gamma distributions. The main feature of the model is that only one parameter θ (in addition to the mixture weights) is used throughout. The parameters are estimated using a two-block Gibbs sampling. I will show how it is possible to implement a more efficient estimation algorithm by integrating out the parameter θ. In a simulation study on a real dataset the method is then compared to some other competing approaches. Results show the good predictive performance of the model in the estimation of tail probabilities. Moreover I present an analysis of the Medical Current Beneficiary Survey (MCBS) based on this model.

The second proposed model aims at exploiting the information provided by the data showing how in a two-sample problem the informative content of one sample can be used to better predict quantities related to the other sample. The context for this second model is still the estimation of skewed distributions. The model is based on SQUARE, a novel estimator of the mean difference of two non-negative random variables proposed by Dominici et al. [22]. The good features of the model are shown in an application to the estimation of a tail probability related to medical costs, using the National Medical Expenditures Survey (NMES).

UNIVERSITÁ COMMERCIALE "LUIGI BOCCONI" – MILANO
Facoltá di Economia
Dottorato di Ricerca in Statistica
XVII Ciclo

Bayesian Semiparametric Methods with Quantile Functions: a Proposal with Applications to Prediction

Tutor:
Francesco Billari
Francesca Dominici

Tesi di:
Sergio Venturini

BAYESIAN SEMIPARAMETRIC METHODS WITH QUANTILE FUNCTIONS: A PROPOSAL WITH APPLICATIONS TO PREDICTION

by

Sergio Venturini
Istituto di Metodi Quantitativi
Università Bocconi

Tutor: Francesco Billari – Francesca Dominici

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Milan, January 2006

A Deborah
aria dei miei polmoni
luce dei miei occhi
anima della mia vita

Abstract

Title: Bayesian Semiparametric Methods with Quantile Functions: a Proposal with Applications to Prediction
Author: Venturini Sergio
Tutor: Francesco Billari – Francesca Dominici

This thesis is about *density modeling* and *quantile data analysis* for skewed data. It aims at introducing two semiparametric Bayesian models for the estimation of tail probabilities for skewed distributions.

The first model is based on a mixture of gamma distributions. The main feature of the model is that only one parameter θ (in addition to the mixture weights) is used throughout. The parameters are estimated using a two-block Gibbs sampling. I will show how it is possible to implement a more efficient estimation algorithm by integrating out the parameter θ. In a simulation study on a real dataset the method is then compared to some other competing approaches. Results show the good predictive performance of the model in the estimation of tail probabilities. Moreover I present an analysis of the Medical Current Beneficiary Survey (MCBS) based on this model.

The second proposed model aims at exploiting the information provided by the data showing how in a two-sample problem the informative content of one sample can be used to better predict quantities related to the other sample. The context for this second model is still the estimation of skewed distributions. The model is based on SQUARE, a novel estimator of the mean difference of two non-negative random variables proposed by Dominici et al. [22]. The good features of the model are shown in an application to the estimation of a tail probability related to medical costs, using the National Medical Expenditures Survey (NMES).

Acknowledgements

Many people contributed to support me during the period in which this thesis has been conceived and written. It is really difficult to list all of them by strictly following an order of importance, if any has a sense. Without any doubt two of them have been invaluable and contributed by far more than the others with their ideas, time, patience and kindness: Francesca Dominici and Giovanni Parmigiani. Without their continuous support during the period I spent in Baltimore at the Johns Hopkins School of Public Health this work would not have been possible.

Secondly my gratitude goes to Luigi V. Tava, Valter Conca and the colleagues of the Quantitative Methods Department at the Bocconi School of Management. They supported me both morally and financially and gave me the opportunity to study and complete my education.

I would like to thank Pietro Muliere for the stimulating environment he has created at the PhD in Bocconi University. Thanks also to the faculty and staff at the Istituto di Metodi Quantitativi of the University, especially to Francesco Billari for the many suggestions and insights provided during the entire preparation of the thesis.

Among the others I would like to acknowledge Sonia Petrone and Pierpaolo De Blasi for several stimulating conversations. Thanks also to all the other friends and colleagues at the PhD, Bruno, Max, Raffaele and Valeria, for the nice time we spent together.

Above all I thank my beloved wife Deborah for her continuous and unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Some Examples of Problems to which the Methods can be Applied

It is very common in practice to find problems where the objective is the estimation of a density and the corresponding parameters of interest, such as tail probabilities. For example, consider the situation of an insurance company or a governmental health agency that needs to predict the number of customers or citizens that will ask for a reimbursement higher than a given threshold on the next hospitalization. It would be very useful in this situation to have a detailed model of the distribution of the medical costs because it would allow to estimate such a probability. On this output the company or the agency would base a lot of strategic decisions for the future. Similarly, financial institutions would like to estimate the potential loss that could occur in a future time. In all these situations what is of interest is the tail of a distribution. Thus it is important to develop methods that do not simply smooth the distribution of the data but that are able to perform well in a predictive sense (especially on the tails), taking into account the uncertainty of the model adopted.

Moreover these methods should exploit all the information that is available in the data. So if more than one sample is available, like for every treatment and control analysis, it would be very useful to *borrow strength across samples* for a more efficient estimation of the parameters of interest.

1.2 Structure and Aims

This thesis is about *density modeling* and *quantile data analysis*. The main goal is to develop flexible Bayesian models for the analysis of skewed data. In particular, the main ideas used in the next chapters are:

- set a theoretical framework which allows to estimate a density function very flexibly with the use of few parameters,
- exploit the information provided by the data as much as possible, showing how in a two-sample problem it can be possible to use the informative content of one sample to better predict quantities related to the other sample,
- do everything described above in a Bayesian setting, where additional information can be conveyed by using appropriate prior distributions.

In Chapter 2 I present a critical review of the relevant literature. The aim of this chapter is to briefly collect some material available in the literature about mixtures estimation and quantile data analysis and highlight the pros and cons of each method with respect to the ones presented in this thesis. In Chapter 3 I provide a flexible Bayesian model for the estimation of skewed densities. The model is based on a mixture of gamma distributions that are parametrized in a non-conventional way. In Chapter 4 I extend the model illustrated in Chapter 3 to a two-sample problem, such as a control and a treatment. The objective of the model developed in this chapter is still the estimation of skewed densities, but the more information provided by the two sample will allow a more efficient estimation of some quantities of interest. Chapter 5 concludes with some indications about future research directions.

1.3 Statement of Originality

The main stimulus for writing this thesis has come from the ascertainment that few papers and books exist at the moment on quantile-based techniques for data analysis. Even less has been produced for what regards Bayesian methods based on quantiles, the existing literature being almost exclusively on Bayesian quantile regression (for some examples see the works by Kottas et al. [52],[31],[51] and by

Yu et al. [109]). Working with quantile functions enabled me to understand their power and just catch a glimpse of their potential applications in many scientific situations (an evident example is *quantile regression*, see Koenker [48]).

While Chapter 2 is a review of very well known material, the originality for the method presented in Chapter 3 is claimed for the introduction of a mixture of gamma distribution, about which I found very few works in the literature. Even if the approach used in this chapter for the estimation of the parameters of a mixture distribution is not new, I will show that the proposed mixture of gamma distributions represents a parsimonious model to fit long-tailed distributions. An efficient sampling algorithms is also developed.

The second section of Chapter 4 contains a review of SQUARE, a novel estimator of the mean difference of two non-negative distributions proposed by Dominici et al. [22]. They showed that this estimator is more efficient than alternative methods, but it is not conceived for prediction purposes. Prediction is typically a task which can be tackled easily within the Bayesian framework. This is precisely what I do in the remaining sections of Chapter 4. Building on the SQUARE idea and on the results of the previous chapter, I present a Bayesian version of SQUARE, called B-SQUARE, which I will use almost exclusively for predicting certain tail probabilities. The motivation for this application of B-SQUARE lies on the fact that medical costs typically have a distribution which is characterized by the following features:

- very few hospitalizations with huge costs which cause a long tail in the distribution,
- in the two-sample situation the cases are often less than the controls,
- a significant fraction of zeros in the data.

In that chapter I introduce an explicit form for the controls response density function. This is an innovation which has not been addressed in the original SQUARE paper. This density could be used to propose a likelihood-based version of SQUARE, which originally is formulated as an empirical-based (i.e. nonparametric) model.

I want to spend few words about the title of this work. Strictly speaking the models considered here could not be referred to as *semiparametric* because they

do not contain any infinite-dimensional parameter. They are definitely *parametric* models since just finite-dimensional parameters are involved. I decided to use that terminology to stress the fact that the parametric hypotheses adopted are very weak. In fact, on one side a mixture distribution is used (in particular it is a *finite* mixture distribution, but it can be considered as a sub-case of a more general *continuous* mixture distribution), and on the other side the controls response distribution function is not given explicitly but in terms of the SQUARE hypothesis (i.e. the link between the two quantile functions).

Chapter 2

Mixture Distributions and Quantile Functions for Statistical Modeling: A Review

2.1 Introduction

In this chapter I briefly review the literature which is relevant for presenting the material contained in the next chapters. The aim here is to critically highlight the pros and cons of some of the tools available in the literature and at the same time anticipate the motivations at the ground of this thesis.

In the first section I summarize the approaches frequently used in practice for estimating mixture of distributions, both from the frequentist and Bayesian perspective. In the second section the objective is the review of the available tools for data analysis using quantile functions. This is a non-conventional but very powerful approach for data analysis (see Parzen [71],[72] and Gilchrist [33]) and in this section I emphasize the advantages to embrace such a view.

2.2 Finite Mixtures: a Flexible Tool for Distribution Fitting

Mixture methods have received an increasing attention and and attracted new research in the last few years. This growth of interest is due on one side to the

fact that mixtures represent a highly flexible but parametric tool for modeling a random variable (see Titterington et al. [101], Lindsay [58] and MacLachlan et al. [64] for the frequentist literature and Diebolt et al. [19], Robert [85] and Marin et al. [61] for the Bayesian literature). On the other side the development is the consequence of the availability of an ever increasing computing power. All these reasons are true especially for the Bayesian estimation of mixture models thanks to the increasing and widespread availability of Markov Chain Monte Carlo (MCMC) methods. However, although based on standard distributions, mixture models still pose highly complex computational and conceptual challenges that slow the systematic application of these models in practice.

After a brief review of some basic definitions, in this section I present the most used approaches for estimating the parameters of a mixture highlighting the weaknesses of each of them.

2.2.1 General Framework

Given a set of distributions f_j, a *finite mixture* is simply defined as the convex combination

$$\sum_{j=1}^{J} \pi_j f_j(y)\,, \quad \sum_{j=1}^{J} \pi_j = 1, \quad 0 \leq \pi_j \leq 1\,, \quad J > 1\,.$$

The f_j are called *mixture components* while the π_j are the *mixture weights*. Similarly it is possible to define a *continuous mixture* as

$$g(y) = \int_{\Theta} f(y|\theta)h(\theta)d\theta\,,$$

but they will not be considered in this work (for a short introduction see Carlin et al. [12] and O'Hagan et al. [70]). Most of the times the f_j belongs to a parametric family and each depend upon an unknown set of parameters $\boldsymbol{\theta}_j$, leading to the parametric (finite) mixture model

$$\sum_{j=1}^{J} \pi_j f_j(y|\boldsymbol{\theta}_j)\,. \tag{2.1}$$

The number of mixture components J is an unknown parameter that should be estimated using the available data as well. The disadvantage of this representation is that a lot information is required to estimate all the parameters

$(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J, \pi_1, \ldots, \pi_{J-1}, J)$ of the model. The proposal of the next chapter allows to define a much more parsimonious model (with only J parameters, that is the $(J-1)$ weights and one additional parameter shared by all the components) that still retain the flexibility of the mixture representation.

An important feature of mixtures is that all the moments are convex combinations of the f_j moments, that is

$$\mathbb{E}[Y^r] = \sum_{j=1}^{J} \pi_j \mathbb{E}^{f_j}[Y^r] .$$

Mixture models are mainly used for modeling heterogeneity in a cluster analysis context and in regression analysis, for example to model multilevel components. For this reason they have been widely applied in many different fields. Another frequent application is for density estimation, especially in a nonparametric setting. The most important example is the *kernel* density estimator (see Hastie et al. [35])

$$f(y) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{y - y_j}{h}\right),$$

where the number of components is equal to n, the number of observations, $\pi = 1/n$, the components f_j are set to $h^{-1}K((y-y_j)/h)$ and h is the so called *bandwidth*.

The most used in practice is the normal components mixture. The reason is mainly that normal distributions have parameters that are easily interpretable with respect to the phenomenon to which they are applied. The mixture proposed in the next chapter uses gamma distributions. This is motivated by the motivating application of the model, that is about medical costs, a strictly positive quantity.

2.2.2 Identifiability

In general a parametric family of densities $f(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J)$, is *identifiable* if distinct values of the parameter θ determine distinct members of the family of densities $\{f(y|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$, for a fixed y and where Θ is the parameter space. In other words a parametric family of densities is identifiable if

$$f(y|\boldsymbol{\theta}) = f(y|\boldsymbol{\theta}^*)$$

if and only if

$$\boldsymbol{\theta} = \boldsymbol{\theta}^* .$$

This definition of identifiability needs to be adapted to be suited for mixtures (see MacLachlan et al. [64]). Let

$$f(y|\boldsymbol{\theta}) = \sum_{j=1}^{J} \pi_j f_j(y|\theta_j)$$

and

$$f(y|\boldsymbol{\theta}^\star) = \sum_{j=1}^{J^\star} \pi_j^\star f_j(y|\theta_j^\star)$$

be two members of a parametric family of mixtures. This class of finite mixtures is said to be identifiable for $\boldsymbol{\theta} \in \Theta$ if

$$f(y|\boldsymbol{\theta}) = f(y|\boldsymbol{\theta}^\star) \quad \text{(a.s.)}$$

if and only if $J = J^\star$ and it is possible to permute the component labels so that

$$\pi_j = \pi_j^\star \quad \text{and} \quad f_j(y|\theta_j) = f_j(y|\theta_j^\star) \quad (j = 1, \ldots, J)\,.$$

The lack of identifiability of $\boldsymbol{\theta}$ due to the interchanging of the component labels is generally handled in practice by imposing an appropriate constraint on $\boldsymbol{\theta}$. A typical choice in the case of a mixture of normal distributions is to impose an ordering of the means, such as

$$\mu_1 < \mu_2 < \cdots < \mu_{J-1} < \mu_J\,.$$

Often it happen in practice that there is a natural ordering of the components according to the size of their means. Some other times the constraint imposed has no practical meaning. For the mixture proposed in the next chapter there will not be any identifiability problem and no constraint is needed because the mixture is defined such that the means and the variances are automatically ordered.

2.2.3 The Missing Data Formulation

For the estimation of parameters in a mixture distribution it is convenient to introduce the so called *missing data* formulation of a mixture model. This formulation is well suited not only when the missing data have a physical interpretation but also when they possess no practical interpretation. In general the missing data formulation can be thought of as a way to generate random values from a mixture model.

Consider an iid sample $\boldsymbol{y} = (y_1, \ldots, y_n)$ from a mixture with density $f_j(y|\boldsymbol{\theta}) = \sum_{j=1}^{J} \pi_j f(y|\theta_j)$ and introduce also other unknown quantities $\boldsymbol{x} = (x_1, \ldots, x_n)$ that represent the component labels of each y_i. The quantities $(\boldsymbol{y}, \boldsymbol{x})$ together are called the *complete data*. Note that $\Pr(x_i = j|\boldsymbol{\pi}, \boldsymbol{\theta}) = \pi_j$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$, $j = 1, \ldots, J$ and $i = 1, \ldots, n$. These missing data simplify a lot the structure of a mixture because usually, conditional on the x, the y are assumed to be independent observations from the densities

$$g(y_i|x_i = j, \boldsymbol{\pi}, \boldsymbol{\theta}) = f_j(y_i|\theta_j)\,.$$

Integrating out the missing data $x_1, \ldots, x_n$ allows to write the mixture as

$$\begin{aligned} g(y_i|\boldsymbol{\pi}, \boldsymbol{\theta}) &= \sum_{j=1}^{J} \Pr(x_i = j|\boldsymbol{\pi}, \boldsymbol{\theta}) g(y_i|x_i = j, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ &= \sum_{j=1}^{J} \pi_j f_j(y_i|\theta_j)\,. \end{aligned} \tag{2.2}$$

Sometime the components of the mixture have a physical interpretation. In these occasions inference for the x may be of interest in itself, and one may be interested in quantities such as the *classification probabilities*

$$\begin{aligned} \Pr(x_i = j|y_i, \boldsymbol{\pi}, \boldsymbol{\theta}) &\propto \Pr(x_i = j|\boldsymbol{\pi}, \boldsymbol{\theta}) g(y_i|x_i = j, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ &\propto \pi_j f_j(y_i|\theta_j) \end{aligned}$$

which, calculating the normalizing constant, gives

$$\Pr(x_i = j|y_i, \boldsymbol{\pi}, \boldsymbol{\theta}) = \frac{\pi_j f_j(y_i|\theta_j)}{\sum_{k=1}^{J} \pi_k f_k(y_i|\theta_k)}\,.$$

These will be used in the next chapter.

An obvious way to generate random values from a mixture is to think of the missing data x_i as a J-dimensional random vector $\boldsymbol{x}_i$ which elements are either 1 or 0. In particular only one of the J elements can be equal to 1, the j-th element, and the other are equal to 0. These random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are thus distributed according to a multinomial distribution consisting of one draw on J categories with probabilities $\pi_1, \ldots, \pi_J$, that is

$$\Pr(\boldsymbol{X}_i = \boldsymbol{x}_i) = \pi_1^{x_{1i}} \pi_2^{x_{2i}} \cdots \pi_J^{x_{Ji}}\,,$$

or

$$\boldsymbol{X}_i \sim \mathcal{M}_J(1;\pi)\,.$$

At this point it is possible to generate values first from the J populations

$$f_1(y|\theta_1),\ldots,f_J(y|\theta_J)$$

and then from the multinomial distribution above to get the J components mixture (2.1).

2.2.4 Maximum Likelihood and the EM algorithm

Before the advent of the EM algorithm and the MCMC technology, maximum likelihood (ML) estimation has been the most common approach to the fitting of mixture models. As it is well known, the ML approach is based on the estimation $\left(\widehat{\pi},\widehat{\theta}\right)$ of the mixture parameters (π,θ) obtained by maximizing the likelihood function

$$\mathbb{L}(\pi,\theta|\boldsymbol{y}) = \prod_{i=1}^{n}\left[\pi_1 f_1(y_i|\theta_1)+\cdots+\pi_J f_J(y_i|\theta_J)\right]. \tag{2.3}$$

Unfortunately the ML approach is beset with difficulties mostly caused by the fact for many choices of the f_j the likelihood is unbounded. Moreover ML estimation involves the expansion of (2.3) into J^n terms which is too expensive for more than a few observations. This situation precludes analytical solutions.

However for ML computations it is possible to use numerical optimization procedures. One of these is the *expectation-maximization algorithm*, or *EM algorithm* (see MacLachlan et al. [63]). This algorithm (as well as the MCMC procedures that are used for Bayesian estimation) is based on the missing data representation of a mixture. Using this formulation it is possible to write the (non-observed) log-likelihood of the complete data as

$$\log\mathbb{L}^c(\psi|\boldsymbol{y},\boldsymbol{x}) = \sum_{j=1}^{J}\sum_{i=1}^{n} x_{ji}\left[\log\pi_j+\log f_j(y_i|\theta_j)\right],$$

where $\psi=(\pi,\theta)$. The algorithm then proceeds in two steps:

1. ***E (expectation) step:*** calculate the conditional expectation of the complete-data log-likelihood, given the observed the observed data $\boldsymbol{y}$ and the current value for ψ, that is

$$Q\left(\psi|\psi^{(t-1)},\boldsymbol{y}\right) = \mathbb{E}_{\psi^{(t-1)}}\left[\log\mathbb{L}^c(\psi|\boldsymbol{y},\boldsymbol{x})\right], \tag{2.4}$$

2. ***M (maximization) step:*** this step requires the global maximization of (2.4) with respect to ψ to give the update

$$\psi^{(t)} = \arg\max_{\psi} Q\left(\psi|\psi^{(t-1)}, y\right) .$$

The E- and M-steps are then iterated till the difference

$$\mathbb{L}(\psi^{(t)}|y) - \mathbb{L}(\psi^{(t-1)}|y)$$

changes by an arbitrarily small amount. It is possible to show that the EM algorithm produces a sequence of likelihood values that is increasing (see MacLachlan et al. [63]). So to get convergence one just needs a likelihood that is bounded above.

2.2.5 Bayesian Estimation

Only after the advent of the MCMC technology, and still with the help of the missing data representation it has been possible to estimate mixture models in the Bayesian framework. The first and most used solution to Bayesian estimation of mixtures involves the Gibbs sampler with data augmentation, as detailed in Diebolt and Robert [19]. This algorithm for mixture estimation will be fully presented in the next chapter and there applied to a particular case of the general theory where the components f_j belong to the exponential family as well as the prior distributions (see Marin et al. [61]). For this reason here I discuss only some of the difficulties that the choice of the priors arises.

In Bayesian analysis one can represent "ignorance" or lack of information about a parameter of a model in different ways. The first and most famous approach has been proposed by Jeffreys [45], the *Jeffreys prior*. More recently in the literature it has been accepted that any prior distribution will contain some information about the parameters, and the emphasis has shifted towards the calculation of *reference priors* which result in posteriors which depend most heavily on the data. Reference priors provide a suitable starting point for a Bayesian analysis of the data, guaranteeing scientific objectivity without claiming to represent a definitively correct prior. More details and references may be found in Bernardo and Smith [7].

Unfortunately the reference prior for most mixture models gives an independent improper prior on the mixture model parameters, which cannot be used in a mixture context as it leads to improper posteriors for the component-specific parameters $\theta_1, \ldots, \theta_J$. It is possible to see this by considering the posterior distribution of θ_1 given complete data $(\boldsymbol{y}, \boldsymbol{x})$, where $\boldsymbol{x}$ assigns no observations to the first component. If $\theta_1, \ldots, \theta_J$ are considered a priori independent then $(\boldsymbol{y}, \boldsymbol{x})$ contains no information about the parameter θ_1, and so the posterior distribution $p(\theta_1|\boldsymbol{y}, \boldsymbol{x})$, will be the same as the prior distribution $p(\theta_1)$. If this prior distribution is improper then the posterior $p(\theta_1|\boldsymbol{y}, \boldsymbol{x})$ will also be improper.

In this thesis I will use proper priors. Sometimes they will be chosen to be only weakly informative about the parameters. However, I will emphasize that inference can be very heavily influenced by the priors used, even when the priors appear to be relatively flat. For sure much further work is required on the appropriate specification of priors.

A second known problem involved by Bayesian mixture estimation is the so called *label switching*, that I discuss briefly in the next section.

2.2.6 Label Switching

The problem of *label switching* is one of the main challenges of Bayesian estimation of mixtures and it is caused by the nonidentifiability of the components. That is, if exchangeable priors are used for the parameters of a mixture model, then the resulting posterior distribution will be invariant to permutations in the parameters labels. As a result, the marginal posterior distributions for the parameters will be identical for each mixture component. Therefore, during MCMC simulation, the sampler encounters the symmetries of the posterior distribution and it is then meaningless to draw inference directly from the MCMC output using ergodic averaging.

The simplest solution to this problem is just to impose an identifiability constraint like the ones reviewed in Section 2.2.2. Unfortunately in the Bayesian context these constraints do not always perform adequately and different solutions have been proposed among which tempering seems very promising (for a recent review of these methods see Jasra et al. [44]).

The important point here is that label switching is mainly a problem if one

wishes to use a mixture for clustering. Since this is not the case in this thesis and, as already mentioned, identifiability constraints are automatically imposed in the models proposed, label switching will not be a concern in what follows.

2.2.7 Estimation of the number of components

A further difficulty in mixture estimation arises when one assumes that the number of components is unknown. Again, different methods have been proposed for density estimation and for clustering. In a frequentist context information criteria, like the Akaike's information criterion (AIC) and some other indexes are the most preferred choices for the former types of problems, while testing for the number of mixture components (mainly with likelihood ratio tests) are used for the latter (see McLachlan and Peel [64] for a detailed review).

From a Bayesian point of view when testing is the objective either Bayes factors (see Kass and Raftery [47]) or entropy distance methods (see Sahu and Cheng [90]) can be used. If the perspective is instead on estimation, two important methods have been proposed: one is known as reversible jump MCMC (see Richardson an Green [81]) and the other as birth-and-death MCMC (see Stephens [99]). For a recent review of all these methods for mixtures estimation see Marin et al. [61].

In the model proposed in the next chapter the number of components J will be treated as a fixed nonrandom quantity and therefore it is not estimated using the data. For the mixture presented there J will be used in a way that is not common in practice, being the problem neither one of clustering nor of density smoothing. I would say that the mixture will be used for *density modeling*. During the presentation I will provide some advices on how to fix it.

2.2.8 Nonparametric alternatives

The nonparametric estimation of mixtures has been proposed both within the ML and the Bayesian setting. In the case of ML each of $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J$ in a finite mixture is an element of the same parameter space Θ. It is then possible to think of $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$ as defining a discrete probability distribution $G(\theta)$ over θ with $G(\theta_j) = \Pr(\theta = \theta_j) = \pi_j$, $j = 1, \ldots, J$. The function G is called the *mixing distribution* and it is a discrete probability measure on Θ. Lindsay [57], [58] considered the nonparametric ML estimation (NPMLE) of the mixing distribution

G (actually in a more general situation) and showed that it involves a standard problem of convex optimization. One of the consequences is that, even if one relaxes the hypothesis of finite support of G, under the condition that the likelihood is bounded, the NPMLE of G is concentrated on a support of cardinality at most equal to the number of distinct data points in the sample. This is a very useful result, especially from a computational point of view. This framework provide also the basis for empirical Bayes estimation (see Robbins [82], [83], Laird [54], Carlin and Louis [12]).

The most common approaches in Bayesian nonparametrics is to use a *Dirichlet process distribution* $\mathcal{D}(F_0, \alpha)$ on G (see Ferguson [29] and Antoniak [5]). The application to mixtures is quite straightforward (see O'Hagan [70]), even if the distribution presents some limitations. Another approach in Bayesian nonparametrics is due to Petrone [74] under the name of *Bernstein polynomials*, where bounded continuous densities with support on $[0, 1]$ are approximated by (infinite) beta mixtures with integer parameters.

In the next chapters two parametric Bayesian models are proposed. Further generalizations are possible within the framework of Bayesian nonparametrics, especially for B-SQUARE in Chapter 4. There have been rare attempts to build nonparametric Bayesian models starting from the quantile function. One of these is Hjort and Petrone [37], where methods for carrying out nonparametric Bayesian inference for the quantile function Q, when the prior for F is a Dirichlet process, are developed and applied to several interesting situations.

2.3 Quantile Functions: a Complete View of the Data at hand

Frequentist and Bayesian data analyses are usually performed by fitting models based on a given density function f. Apart from few cases, like quantile regression which is a deeply studied topic especially in econometrics (see Koenker et al. [49] and Koenker [48]), quantile methods for data analysis are not widely used in practice. And the reasons for this choice are not clear, given that they are able to exploit more efficiently the information contained in the sample.

Quantile methods were pioneered by Galton [30], who computed medians and

quartiles of conditional distributions of heights of sons given heights of parents, and discovered that they had constant scale and linear location. Moreover many facts about quantiles have a long history and were known before 1900 (see Hald [34]). Only recently, starting from the late 70s, these methods have been revitalized and developed mainly by Emanuel Parzen (see Parzen [71], [72]).

The objective of this section is to review the tools nowadays available for analyzing data using quantiles. As in the previous section on mixtures, I will insist on the characteristics of these methods that are most relevant for the next chapters. For some really basic definitions and properties see Appendix A.

2.3.1 Q-Q Plot: the Basic Tool

Used mainly for identifying the distribution of a set of data, the *Q-Q plot* is by far the most famous tool that exploits the quantiles. Suppose to have a statistical model stated in terms of the quantile function Q instead of the distribution function F (the quantile function Q of a given distribution function F is defined as the *generalized inverse function* of F and it is usually denoted as $F^{-1}(p)$; see A). Then collect a sample of n iid observations $(y_1, \ldots, y_n)$. This sample can be restated in terms of the *order statistics* $(y_{(1)}, \ldots, y_{(n)})$ (the order statistics are the sample version of the population quantiles $Q(p_1), \ldots, Q(p_n)$, where $p_i = i/(n+1)$, $i = 1, \ldots, n$). The Q-Q plot is a plot of the points $(y_{(i)}, Q(p_i))$, i.e. the n sample quantiles $y_{(i)}$ against the corresponding model quantiles $Q(p_i)$. If model Q is a good representation of the quantity under study, then the Q-Q plot should result approximately in a straight line. An inappropriate model will show some systematic curvature. As an example consider the Q-Q plots in Figures 2.1 and 2.2, where the order statistics for samples of log non-zero medical costs for people with (cases) and without (controls) a certain disease are plotted against the quantile of a fitted normal distribution. From the figures it is clear that the log-normal distribution is not a good model to use for these data. In general it may be that the overall shape of the plot corresponds to that of a simple function h of the model Q. In this case then a transformation of the model would be checked in the next Q-Q plot (for a review of the transformation rules for quantile functions see Appendix A). The theory of weak convergence of empirical processes forms the basis for the construction of confidence bands around the graphs, leading to hypothesis testing

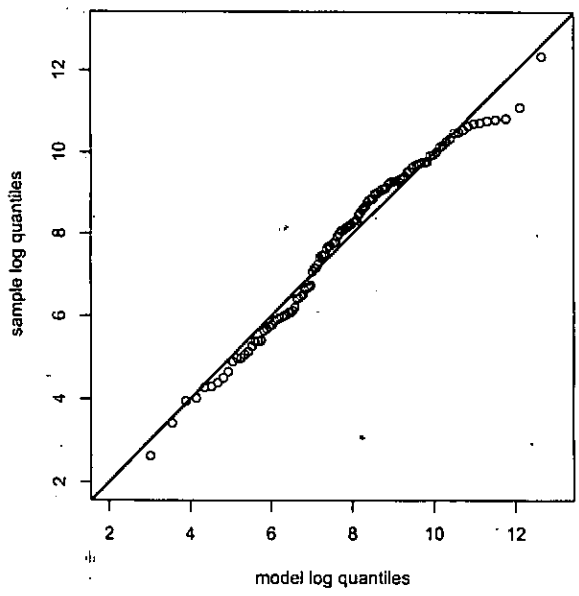

Figure 2.1: Q-Q plot of log non-zero medical costs for cases versus quantiles of a normal distribution.

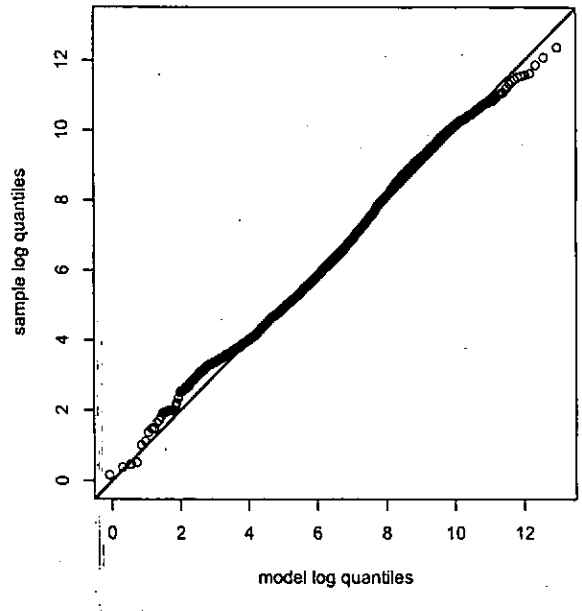

Figure 2.2: Q-Q plot of log non-zero medical costs for controls versus quantiles of a normal distribution.

(see Shorack et al. [96], van der Vaart et al. [103], Van der Vaart [102] and Shorack [95]).

The main merits of Q-Q plots stem from the following properties, taken from Embrechts et al. [26] and David [15].

(a) Comparison of distributions. If the data were generated from a random sample of the reference distribution, the plot should look roughly linear. This remains true if the data come from a linear transformation of the distribution.

(b) Outliers. If one or a few of the data are contaminated by gross error or for any reason are markedly different in value from the remaining values, the latter being more or less distributed like the reference distribution, the outlying points may be easily identified on the plot.

(c) Location and scale. Because a change of one of the distributions by a linear transformation simply transforms the plot by the same transformation, one may estimate graphically (through the intercept and slope) location and scale parameters for a sample of data, on the assumption that the data come from the reference distribution.

(d) Shape. Some differences in distributional shape may be deduced from the plot. For example if the reference distribution has heavier tails (tends to have more large values) the plot will curve down at the left and/or up at the right.

The Q-Q plot will be the main tool used in Chapter 4, actually the tool that motivated the entire model.

2.3.2 Other Useful Quantities

Once a reasonable model Q has been identified using the Q-Q plot, one can then estimate the parameters of the model by using some known methods, like maximum likelihood (see Appendix A). Using the chosen model it is then possible to calculate a series of interesting quantities, like moments, quartiles, skewness and kurtosis indexes, and many others (see Gilchrist [33]). The most useful quantities for the model in Chapter 4 are the *quantile density function* $q(p)$, the *density quantile function* $f(Q(p))$, and the *tailweight function* $TW(p)$. The quantile density function is the derivative of the quantile function $Q(p)$ with respect to p, that

is

$$q(p) = dQ(p)/dp.$$

It is an alternative way to state a model and it corresponds to the standard density function f when a model is stated using a distribution function F. The density quantile function is the density function f expressed as a function the percentile p, that is $f(Q(p)) = f(F^{-1}(p)) = f(x)$, given that $p = F(x)$. Finally the tailweight function is defined as

$$TW(p) = \frac{q(p)}{Q(p)}, \quad 0 < p < 1. \tag{2.5}$$

Note that $TW(p)$ is the derivative of the log quantile function $Q(p)$. The tailweight function is used to compare the tail heaviness of different distributions, i.e. a distribution G will have heavier tail than a distribution F if $TW_G(p) \geq TW_F(p)$ for $p \to \infty$. The tailweight function will be a useful in many places in Chapter 4.

2.3.3 Relating Two Distributions: The Shift Function and The Comparison Distribution

Lehmann [55] proposed the following model of two-sample treatment response:

> Suppose the treatment adds the amount $\Delta(x)$ when the response of the untreated subject would be x. Then the distribution G of the treatment responses is that of the random variable $X+\Delta(X)$ where X is distributed according to F.

Special cases include the location shift model, $\Delta(X) = \Delta_0$, and the scale shift model, $\Delta(X) = \Delta_0 X$. If the treatment is beneficial in the sense that $\Delta(x) \geq 0$ for all x, then the distribution of treatment responses, G, is stochastically larger than the distribution of control responses, F.

Doksum [20] introduced the so called *shift function* for both describing and testing the differences between two distributions F and G. He showed that if $\Delta(x)$ is defined as the "horizontal distance" between F and G at x so that

$$F(x) = G(x + \Delta(x)),$$

then $\Delta(x)$ is uniquely defined as

$$\Delta(x) = G^{-1}(F(x)) - x$$

so that $\Delta(X) + X$ has the same distribution as the treatment responses G. Thus, on changing variables so $\tau = F(x)$, the quantile treatment effect is given by

$$\delta(\tau) = \Delta(F^{-1}(\tau)) = G^{-1}(\tau) - F^{-1}(\tau)\,.$$

Note that it is possible to recover the mean treatment effect by integrating the quantile treatment effect over τ, that is

$$\bar{\delta} = \int_0^1 \delta(\tau)d\tau = \int_0^1 G^{-1}(\tau)d\tau - \int_0^1 F^{-1}(\tau)d\tau = \mu(G) - \mu(F)\,,$$

where $\mu(F)$ is the mean of the distribution F. The Doksum's shift function is also closely related to quantile regression (see Koenker [48]).

Given two iid samples from F and from G the function has been estimated by Doksum with the natural nonparametric estimator

$$\widehat{\Delta}(x) = \widehat{G}^{-1}\left(\widehat{F}(x)\right) - x\,,$$

where $\widehat{F}$ and $\widehat{G}$ are the empirical distribution functions of F and G.

The idea behind the shift function is that, if evidence exists in the Q-Q plot against the linearity, one possibility to model the cases and controls distributions is to assume that $Q_F(p)$, the quantile function of the cases, is an arbitrary function of $Q_G(p)$, the quantile function of the controls, that is $Q_F(p) = g(Q_G(p))$, or equivalently $F(x) = G(h(x))$. Doksum and Sievers [21] define $h(\cdot)$ as the "amount of shift" needed to bring the controls up to cases in distribution. For example, one might assume that $Q_F(p)$ is a smooth function of $Q_G(p)$ with λ degrees of freedom, $Q_F(p) = s(Q_G(p), \lambda)$, where s is a parametric or a non-parametric smoother.

Another quantity useful for distribution comparisons has been proposed by Parzen (see Parzen [71]). He called it the *comparison distribution* and is defined as $D(p) = G(F^{-1}(p))$, $0 < p < 1$. This function, or an estimate of it, may be plotted against the percentile p. If the two distributions are equal the graph of $D(p)$ would be a 45 straight line (see Parzen [72] for some examples).

As it is shown in Chapter 4, the approach adopted in SQUARE is to assume that the *log-quantile ratio* of the two distributions F and G is a smooth function of the percentile p. This choice presents many advantages over the shift function and the comparison distribution.

Chapter 3

Bayesian Density Estimation for Skewed Data

The aim of this chapter is to introduce a semiparametric Bayesian model for tail estimation of skewed distributions. This model will provide the distributional assumption that will be used in the next chapter. The model introduced in the next sections is based on a mixture of gamma distributions. The main feature of the model is that only one parameter θ (in addition to the mixture weights) is used throughout. The parameters are estimated using a two-block Gibbs sampling. I will show how it is possible to implement a more efficient estimation algorithm by integrating out the parameter θ. In a simulation study on a real dataset the method is then compared to some other competing approaches. Results show the good predictive performance of the model in the estimation of tail probabilities. An analysis of real data on the Medical Current Beneficiary Survey (MCBS) will also be shown.

3.1 Introduction

Skewed distributions are very common in data analysis. The typical situation that gives rise to a skewed distribution is the presence of few large values of the quantity under examination. It is a very well known fact that these observations heavily influence the results of a statistical analysis. To put a remedy to these situations many *robust* methods have been developed (see for example Huber [39],[40]). The

aim of these methods is to downsize the importance of the unusual large data. However there are occasions where these large observations are the focus of the analysis. Insurance companies and governmental health departments, for example, need to predict how many customers or citizens will ask for a reimbursement above a certain threshold. Similarly, financial institutions would like to know the potential loss that could occur in the next day, week or month with a given probability. In all these situations what is of interest is a tail of a distribution. Thus it is important to develop methods that do not simply smooth the distribution of the data but that are able to perform well from a predictive perspective, taking into account also the uncertainty of the model parameters.

One way to address these problems is to model the distribution of the data at hand using a *mixture of distributions*. Mixture models represent probably the best example of a *semiparametric* statistical model, that is a parametric model which is flexible enough to represent a large spectrum of different phenomena. This flexibility is particularly useful when one needs to model skewed distributions like those described in the examples above. But since the objective of the analysis is to set up a model with a good predictive performance, using a mixture model would not be sufficient. The best way to solve this second part of the problem is to use a Bayesian approach, because it allows to bring into consideration also the important issue of parameters uncertainty. Thus the keywords for the model presented below are mixture models and Bayesian predictive estimation.

The model introduced in this chapter is a mixture of gamma distributions that share a common parameter θ, the only one in the model (apart from the mixture weights). This allows to create a parsimonious model that is flexible enough to fit a wide range of skewed distributions. Although the model is based on the Bayesian literature on exponential family mixtures, I did not find prior reference to a similar model. The works published in the literature are mainly about mixtures of normal distributions. This is motivated by the fact that just a set of three or four heteroschedastic normal components can originate a wide variety of density shapes (see for example McLachlan and Peel [64], Section 1.5; for a comprehensive list of applications see the monograph by Titterington et al. [101] and the more recent article by Titterington [100]). Most of the works related to non-normal components involve instead mixtures of generalized linear models (GLM), which are capable to handle also the regression case and the presence of overdispersion (that is an

observed mean-variance relationship that does not match the hypothesized one) especially for discrete data (see Jansen [43], Wedel and deSarbo [105], Aitkin [1],[2] and Scallan [91]).

For what regards asymmetric distributions, the standard approach in the literature is to first transform the data using a Box-Cox transformation. I will follow the latter approach whenever it is convenient from a strictly practical point of view. It is important to stress right from now that this transformations do not have any influence on the results, since the objective of our approach is exclusively to estimate a tail probability. This is immediately clear if one considers the problem formally, that is given a monotone function g, then

$$\mathbb{P}(X > k) = \mathbb{P}(g(X) > g(k)) \ .$$

A further difference of my approach with respect to the literature is the importance often given in the latter to the determination of the number of components. This is a consequence of the type of applications to which mixtures are usually applied, that is clustering and classification. Some other times a mixture is used not for modeling but just for exploratory purposes, in which case n components are used (as in kernel density estimation). The purpose of this chapter is neither clustering nor smoothing. The aim of the model here is to provide a good representation of the data distribution and at the same time to produce an accurate estimate of its right-tail.

The remainder of this chapter is structured as follows. In Section 2 I introduce the gamma mixture model, some of its properties, the estimation approach and I provide some advices on how to choose the hyperparameters of the θ prior. In Section 3 I use the Medicare Beneficiaries Survey (MCBS) dataset both in the simulation study and the data analysis is presented. In Section 4 I test the gamma mixture model on the MCBS dataset by comparing its predictive performance with that of other competing models. In Section 5 I analyze the full MCBS dataset. Section 6 contains some concluding remarks.

3.2 The Gamma Mixture Model

3.2.1 Likelihood

Let Y be a positive random variable. In the applications presented in the next sections they are the non-zero medical expenditures paid by a national health program for a certain group of people. The gamma mixture model is defined as

$$f(y|\pi_1,\ldots,\pi_J,\theta) = \sum_{j=1}^{J} \pi_j f_j(y|\theta)\,, \tag{3.1}$$

where $f_j(y|\theta)$ is the density function of a gamma $\mathcal{G}a(j,\theta)$ random variable, that is

$$f_j(y|\theta) = \frac{\theta^j}{\Gamma(j)} y^{j-1} e^{-\theta y}. \tag{3.2}$$

The number of components J is a fixed nonrandom quantity. $\pi = (\pi_1,\ldots,\pi_J)$ is the vector of mixture weights, namely $0 \le \pi_j \le 1$ $(j = 1,\ldots,J)$ and $\sum_{j=1}^{J} \pi_j = 1$, while $\frac{1}{\theta}$ is the scale parameter of both the components and the whole model, in fact it satisfies

$$\begin{aligned} f(y|\pi_1,\ldots,\pi_J,\theta) &= \frac{1}{\frac{1}{\theta}} f\left(\frac{1}{\frac{1}{\theta}} y \middle| \pi_1,\ldots,\pi_J\right) \\ &= \theta\, f(\theta \cdot y|\pi_1,\ldots,\pi_J) \end{aligned}$$

(see Lehmann and Casella [56], page 167, and Carlin and Louis [12], page 31). In what follows, model (3.1) will be referred to as $\mathcal{M}ix\mathcal{G}a\,(\pi,\theta|J)$. Note that, since the means and variances of the components are equal respectively to $\frac{j}{\theta}$ and $\frac{j}{\theta^2}$ $(j = 1,\ldots,J)$, a density with an "as thick as you want" right tail can be obtained by setting J big enough. This represents the main motivation for proposing the gamma mixture model. In Figure 3.1 some of the gamma densities for the $\mathcal{M}ix\mathcal{G}a\,((\pi_1,\ldots,\pi_{10}),1|10)$ model are shown.

A nice property of a mixture is that moments are convex combinations of the moments of the f_j. For the mixture of gamma this implies that

$$\mathbb{E}\,[Y] = \sum_{j=1}^{J} \pi_j\, \mathbb{E}\,[Y_j] = \sum_{j=1}^{J} \pi_j \frac{j}{\theta}\,, \tag{3.3}$$

$$\mathbb{E}\,[Y^2] = \sum_{j=1}^{J} \pi_j\, \mathbb{E}\,[Y_j^2] = \sum_{j=1}^{J} \pi_j \frac{j\,(j+1)}{\theta^2}$$

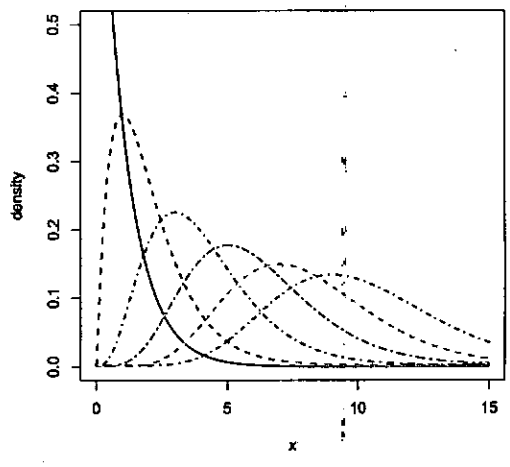

Figure 3.1: Some gamma densities for the $\mathcal{M}ix\mathcal{G}a\left((\pi_1,\ldots,\pi_{10}),1|10\right)$ model.

and thus

$$\text{var}\left[Y\right]=\sum_{j=1}^{J}\pi_j\frac{j\left(j+1\right)}{\theta^2}-\left(\sum_{j=1}^{J}\pi_j\frac{j}{\theta}\right)^2. \tag{3.4}$$

Given a sample $\boldsymbol{y}=(y_1,\ldots,y_n)$ of iid observations from (3.1), the likelihood is of the form

$$\mathbb{L}\left(\boldsymbol{\pi},\theta|\boldsymbol{y}\right)=\prod_{i=1}^{n}\sum_{j=1}^{J}\pi_j\,f_j\left(y_i|\theta\right). \tag{3.5}$$

Unfortunately this expression is unsuitable for the derivation of any Bayes estimators since it is made up by J^n different terms. This implies a similar combinatorial expansion also for the posterior expectation of the parameters, even though conjugate priors are used. Hence, the computational burden becomes too high to be used for more than a few observations. This is a consequence of the representation of the mixture given in (3.1). In a next section I will show a widely used alternative representation that is able to solve this problem.

As already mentioned in Chapter 2, another important issue to consider for a mixture is its invariance to permutations of the indexes of the components. This problem is often called *label switching* (see Jasra et al. [44]). A typical solution is to impose an *identifiability constraint*. The constraints usually adopted in the literature are either an ordering of the components means or variances or an ordering of the mixture weights (see Aitkin and Rubin [4]). The gamma mixture

model (3.1) automatically imposes a constraint on the means and variances of the components since

$$\frac{1}{\theta} < \frac{2}{\theta} < \cdots < \frac{J-1}{\theta} < \frac{J}{\theta}$$

for the means and

$$\frac{1}{\theta^2} < \frac{2}{\theta^2} < \cdots < \frac{J-1}{\theta^2} < \frac{J}{\theta^2},$$

for the variances, so the model is always identified and label switching is not a problem.

To have an idea of the possible density shapes that can be generated by a gamma mixture model, in Figure 3.2 some examples are reported. These densities have been obtained by generating the weights from a Dirichlet $\mathcal{D}_J(1,\ldots,1)$ distribution and the θ parameter from a gamma $\mathcal{G}a(1.5, 1.5)$ distribution.

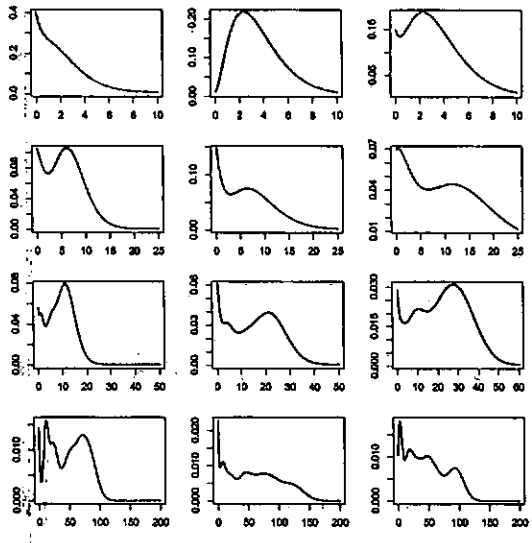

Figure 3.2: Some gamma mixture densities for $J = 3$ *(first row)*, $J = 10$ *(second row)*, $J = 25$ *(third row)* and $J = 100$ *(last row)*.

3.2.2 Priors

I propose a conjugate prior for θ, that is a gamma $\mathcal{G}a(\alpha, \beta)$ distribution. Indications about interpretation of the hyperparameters and choice of their values will be given in a next section. For the mixture weights a conjugate prior, that is a Dirichlet $\mathcal{D}_J(\gamma_1,\ldots,\gamma_J)$ distribution, is used as well. Rather than following the common

choice adopted in the literature (see Robert [85] and Marin et al. [61]), that is to set $\gamma_j = 1$ for all $j = 1, \ldots, J$, I suggest to use the following specification

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J) \sim \mathcal{D}_J\left(\frac{1}{J}, \ldots, \frac{1}{J}\right).$$

The Dirichlet distribution is a multivariate generalization of the beta distribution defined on the unitary simplex. For the beta distribution the effect of setting the parameters at values that are lower than one is to produce a U-shaped density. The same is true for the Dirichlet distribution, as is shown in Figure 3.3 for $J = 3$. The prior distribution for the weights is hence informative[1]. The advantage of this

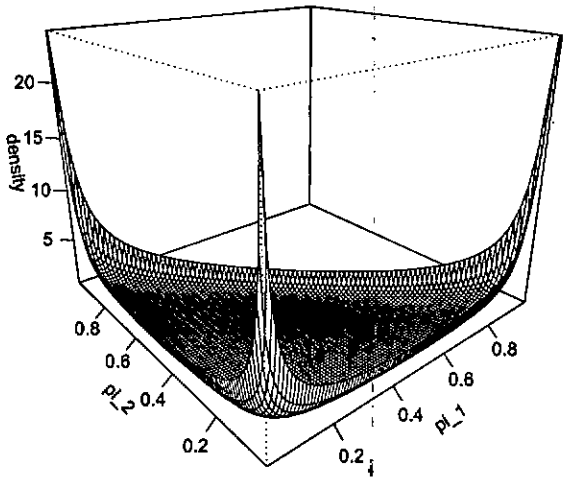

Figure 3.3: Dirichlet $\mathcal{D}_3\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ prior distribution.

choice is that it allows to select with high probability only a small subset of the mixture weights that are thus relatively larger than the others. This avoids to get a model that is too smooth (see the examples in the next sections). The complete specification of the model is summarized in Figure 3.4.

[1]Being $\mathcal{D}_J(1, \ldots, 1)$ the noninformative case.

Model : $f(y|\pi_1,\ldots,\pi_J,\theta) = \sum_{j=1}^{J} \pi_j f_j(y|\theta)\,, \qquad f_j(y|\theta) \equiv \mathcal{G}a(j,\theta)$

Prior structure : $\theta \sim \mathcal{G}a(\alpha,\beta)$

$\boldsymbol{\pi} = (\pi_1,\ldots,\pi_J) \sim \mathcal{D}_J\left(\frac{1}{J},\ldots,\frac{1}{J}\right)$

θ, $\boldsymbol{\pi}$ independent

Figure 3.4: The gamma mixture model.

3.2.3 Missing Data Structure

A very useful alternative representation for a mixture model is the one that uses the *missing data* approach (see Robert [85]). Consider a random sample $\boldsymbol{y} = (y_1,\ldots,y_n)$ from the mixture model (3.1), then it is possible to rewrite $y \sim f(y|\boldsymbol{\pi},\theta)$ as $y \sim f_x(y|\theta)$, where x is an integer between 1 and J identifying the component of the mixture generating the observation y. The variable x, which can be considered as a latent variable, takes value j with prior probability π_j, $1 \leq j \leq J$. The vector $\boldsymbol{x} = (x_1,\ldots,x_n)$ of components labels is the *missing data* part of the sample, since it is not observed. To illustrate this modification of the model consider the diagram in Figure 3.5, where the key idea is that y is conditionally independent from the mixture weights π, given the missing data x. This suggestion will be exploited in the next sections.

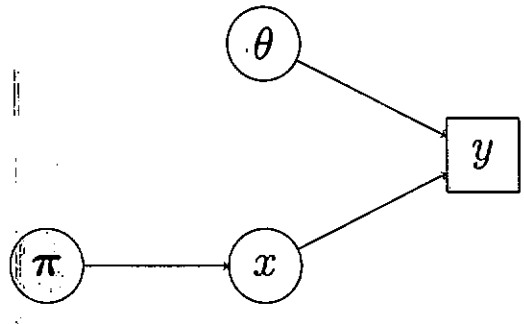

Figure 3.5: Directed acyclic graph (DAG) for the missing data representation of the gamma mixture.

Suppose the missing data $x_1,\ldots,x_n$ were available, then the model can be

written as

$$\begin{aligned} p(y_1,\ldots,y_n|x_1,\ldots,x_n,\theta) &= \prod_{i:\, x_i=1} f_{x_i}(y_i|\theta)\cdots\prod_{i:\, x_i=J} f_{x_i}(y_i|\theta) \\ &= \prod_{i=1}^{n} f_{x_i}(y_i|\theta) \\ &= \frac{\theta^{\sum_{i=1}^n x_i}}{\prod_{i=1}^n \Gamma(x_i)}\left(\prod_{i=1}^n y_i^{x_i-1}\right) e^{-\theta\sum_{i=1}^n y_i}. \end{aligned} \tag{3.6}$$

Thus, using (3.6) and the priors, the posterior distribution is obtained as

$$\begin{aligned} p(\pi_1,\ldots,\pi_J,\theta|y_1,\ldots,y_n,x_1,\ldots,x_n) &\propto p(y_1,\ldots,y_n|x_1,\ldots,x_n,\theta)\times \\ &\quad p(x_1,\ldots,x_n|\pi_1,\ldots,\pi_J)\times \\ &\quad p(\pi_1,\ldots,\pi_J)\,p(\theta) \\ &\propto \left(\prod_{j=1}^J \pi_j^{\frac{1}{J}+n_j-1}\right)\theta^{\alpha+(\sum_{i=1}^n x_i)-1}\times \\ &\quad e^{-(\beta+\sum_{i=1}^n y_i)\theta}, \end{aligned} \tag{3.7}$$

where $n_j = \sum_{i=1}^n \mathbb{I}(x_i = j)$, $j = 1,\ldots,J$, and $\mathbb{I}(\cdot)$ is the indicator function. The main consequence of this conditional decomposition is that for a given missing data structure $x_1,\ldots,x_n$ the conjugacy is preserved and therefore the simulation can be performed conditional on the missing data $x_1,\ldots,x_n$.

3.2.4 Posterior calculation

The posterior distribution of $(\pi_1,\ldots,\pi_J,\theta)$, given the sample $(y_1,\ldots,y_n)$, can be written as

$$p(\pi_1,\ldots,\pi_J,\theta|y_1,\ldots,y_n) \propto \left(\prod_{j=1}^J \pi_j^{\frac{1}{J}-1}\right)\theta^{\alpha-1}e^{-\beta\theta}\prod_{i=1}^n\left(\sum_{j=1}^J \pi_j\frac{\theta^j}{\Gamma(j)}y_i^{j-1}e^{-\theta y_i}\right).$$

As for the likelihood function, the computational burden that this equation requires is due to the J^n terms involved, each of the form

$$\left(\prod_{j=1}^J \pi_j^{\frac{1}{J}+n_j-1}\right)\theta^{\alpha+(\sum_{j=1}^J j\cdot n_j)-1}e^{-(\beta+\sum_{i=1}^n y_i)\theta},$$

where $n_1+\cdots+n_J = n$. As explained in the previous section, this combinatorial calculation can be carried out using Gibbs sampling, by introducing a set of missing data as part of the sample.

The implementation of the Gibbs sampling is straightforward and involves the iterative simulation from (3.7) for the parameters of the model and from $p(x_1, \ldots, x_n | \pi_1, \ldots, \pi_J, \theta, y_1, \ldots, y_n)$ for the missing data. The steps for the simulation, using the properly calculated full conditional distributions, are described in Figure 3.6.

Step 1. Simulate

$$\theta | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\pi} \sim \mathcal{G}a\left(\alpha + \sum_{i=1}^{n} x_i, \beta + \sum_{i=1}^{n} y_i\right) \tag{3.8}$$

$$\boldsymbol{\pi} | \boldsymbol{y}, \boldsymbol{x}, \theta \sim \mathcal{D}_J\left(\frac{1}{J} + n_1, \ldots, \frac{1}{J} + n_J\right), \tag{3.9}$$

where $n_j = \sum_{i=1}^{n} \mathbb{I}(x_i = j)$, $j = 1, \ldots, J$.

Step 2. Simulate, for every $i = 1, \ldots, n$,

$$p(x_i | y_i, \pi_1, \ldots, \pi_J, \theta) = \sum_{j=1}^{J} \pi_{ij} \mathbb{I}(x_i = j), \tag{3.10}$$

where

$$\pi_{ij} = \frac{\pi_j f_j(y_i|\theta)}{\sum_{k=1}^{J} \pi_k f_k(y_i|\theta)}, \qquad j = 1, \ldots, J. \tag{3.11}$$

Step 3. Update n_j, $j = 1, \ldots, J$.

Figure 3.6: *Algorithm 1*, Gibbs sampling.

As it is stated in the literature (see Robert [85], page 448), *Algorithm 1* is quite efficient, and 5,000 iterations are usually enough to get a reliable estimate of the stationary distribution of the chain. However it is possible to modify the algorithm in order to get rid of θ. This can be done by integrating it out from (3.6). The consequence of this modification is that the full conditionals of the missing data will no longer depend upon θ and so their chains will not be influenced by its sampling variation during the simulation. Even if I do not provide a formal proof for this, intuitively this change contributes to increase the efficiency of the algorithm. So in this last part of the section I show how the Gibbs sampler is modified.

First take equation (3.6) and integrate out θ

$$\begin{aligned} p(y_1,\ldots,y_n|x_1,\ldots,x_n) &= \int \frac{\theta^{\sum_{i=1}^n x_i}}{\prod_{i=1}^n \Gamma(x_i)} \left(\prod_{i=1}^n y_i^{x_i-1}\right) e^{-\theta\sum_{i=1}^n y_i} \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta}\, d\theta \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\prod_{i=1}^n y_i^{x_i-1}}{\prod_{i=1}^n \Gamma(x_i)} \int \theta^{\alpha+(\sum_{i=1}^n x_i)-1} e^{-(\beta+\sum_{i=1}^n y_i)\theta}\, d\theta \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\prod_{i=1}^n y_i^{x_i-1}}{\prod_{i=1}^n \Gamma(x_i)} \frac{\Gamma\left(\alpha+\sum_{i=1}^n x_i\right)}{\left(\beta+\sum_{i=1}^n y_i\right)^{\alpha+\left(\sum_{i=1}^n x_i\right)}}. \end{aligned} \tag{3.12}$$

This expression correctly depends only upon α and β, the hyperparameters of θ. Then (3.7) becomes

$$\begin{aligned} p(\pi_1,\ldots,\pi_J|y_1,\ldots,y_n,x_1,\ldots,x_n) &\propto p(y_1,\ldots,y_n|x_1,\ldots,x_n) \times \\ &\quad p(x_1,\ldots,x_n|\pi_1,\ldots,\pi_J) \times \\ &\quad p(\pi_1,\ldots,\pi_J) \\ &\propto \prod_{j=1}^J \pi_j^{\frac{1}{J}+n_j-1}, \end{aligned} \tag{3.13}$$

hence the full conditional of the mixture weights is still the Dirichlet distribution

$$\boldsymbol{\pi}|\boldsymbol{y},\boldsymbol{x} \sim \mathcal{D}_J\left(\frac{1}{J}+n_1,\ldots,\frac{1}{J}+n_J\right).$$

The next step is to find the new full conditional of the missing data, that will substitute (3.10). Note first that in *Algorithm 1* the missing data $\boldsymbol{x}=(x_1,\ldots,x_n)$ were independent, conditionally on the sample $\boldsymbol{y}$ and the weights $\boldsymbol{\pi}$, that is

$$p(x_1,\ldots,x_n|y_1,\ldots,y_n,\pi_1,\ldots,\pi_J,\theta) = \prod_{i=1}^n p(x_i|y_i,\pi_1,\ldots,\pi_J,\theta).$$

The modification introduced (i.e. the integration of θ) implies that the observations $\boldsymbol{y}$ (given the components labels $\boldsymbol{x}$) are no longer conditionally independent among themselves (compare in fact (3.6) with (3.12)). An intuitive explanation is that θ was a parameter shared by all the (y_i,x_i) pairs, $i=1,\ldots,n$. Removing θ has introduced dependence among the data.

A possible solution that can be suggested is to decompose the full conditional

$$p(x_1,\ldots,x_n|y_1,\ldots,y_n,\pi_1,\ldots,\pi_J)$$

into its building blocks, that is into the full conditionals

$$p(x_r|y_1,\ldots,y_n,x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n,\pi_1,\ldots,\pi_J),$$

with $r \in \{1, \ldots, n\}$. These expressions can be obtained by first noting that

$$\begin{aligned} p(x_r|\boldsymbol{y}, \boldsymbol{x}_{(-r)}, \pi) &= \frac{p(\boldsymbol{x}|\boldsymbol{y}, \pi)}{p(\boldsymbol{x}_{(-r)}|\boldsymbol{y}, \pi)} \\ &= \frac{p(\boldsymbol{x}|\boldsymbol{y}, \pi)}{\sum_{x_r=1}^{J} p(\boldsymbol{x}|\boldsymbol{y}, \pi)} \\ &= \Big(\text{since } p(\boldsymbol{x}|\boldsymbol{y}, \pi) \cdot p(\boldsymbol{y}|\pi) = p(\boldsymbol{y}|\boldsymbol{x}, \pi) \cdot p(\boldsymbol{x}|\pi)\Big) \\ &= \frac{p(\boldsymbol{y}|\boldsymbol{x}, \pi) \cdot p(\boldsymbol{x}|\pi)}{\sum_{x_r=1}^{J} p(\boldsymbol{y}|\boldsymbol{x}, \pi) \cdot p(\boldsymbol{x}|\pi)} \\ &= \frac{p(\boldsymbol{y}|\boldsymbol{x}) \cdot p(\boldsymbol{x}|\pi)}{\sum_{x_r=1}^{J} p(\boldsymbol{y}|\boldsymbol{x}) \cdot p(\boldsymbol{x}|\pi)}, \end{aligned}$$

for $x_r \in \{1, \ldots, J\}$, $r \in \{1, \ldots, n\}$ and where the notation $\boldsymbol{x}_{(-r)}$ means the vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ with the r-th element deleted. Thus substituting (3.12)

$$\begin{aligned} p(x_r|\boldsymbol{y}, \boldsymbol{x}_{(-r)}, \pi) &= \frac{\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\prod_{i=1}^n y_i^{x_i-1}}{\prod_{i=1}^n \Gamma(x_i)} \frac{\Gamma\left(\alpha+\sum_{i=1}^n x_i\right)}{\left(\beta+\sum_{i=1}^n y_i\right)^{\alpha+\left(\sum_{i=1}^n x_i\right)}} \times}{\sum_{x_r=k=1}^{J} \left\{ \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\prod_{i=1}^n y_i^{x_i-1}}{\prod_{i=1}^n \Gamma(x_i)} \frac{\Gamma\left(\alpha+\sum_{i=1}^n x_i\right)}{\left(\beta+\sum_{i=1}^n y_i\right)^{\alpha+\left(\sum_{i=1}^n x_i\right)}} \times \right.} \\ &\quad \left. \frac{\times \prod_{i=1}^n \prod_{j=1}^J \pi_j^{\mathbf{I}(x_i=j)}}{\times \prod_{i=1}^n \prod_{j=1}^J \pi_j^{\mathbf{I}(x_i=j)}} \right\} \\ &= \frac{\pi_j \frac{y_r^{x_r-1}}{\Gamma(x_r)} \frac{\Gamma\left(\alpha+\sum_{(-r)} x_i+x_r\right)}{\left(\beta+\sum_{i=1}^n y_i\right)^{x_r}}}{\sum_{k=1}^J \pi_k \frac{y_r^{k-1}}{\Gamma(k)} \frac{\Gamma\left(\alpha+\sum_{(-r)} x_i+k\right)}{\left(\beta+\sum_{i=1}^n y_i\right)^{k}}}, \end{aligned}$$

where the notation $\sum_{(-r)} x_i$ means to sum all the x_i apart from the r-th one. If one further assumes[2] $\alpha \in \mathbb{N}$, then (3.14) can be simplified to

$$\begin{aligned} p(x_r|\boldsymbol{y}, \boldsymbol{x}_{(-r)}, \pi) &= \frac{\pi_j \frac{y_r^{x_r-1}}{\Gamma(x_r)} \frac{\left(\alpha+\sum_{(-r)} x_i+x_r-1\right)_{x_r} \times \left(\alpha+\sum_{(-r)} x_i-1\right)!}{\left(\beta+\sum_{i=1}^n y_i\right)^{x_r}}}{\sum_{k=1}^J \pi_k \frac{y_r^{k-1}}{\Gamma(k)} \frac{\left(\alpha+\sum_{(-r)} x_i+k-1\right)_k \times \left(\alpha+\sum_{(-r)} x_i-1\right)!}{\left(\beta+\sum_{i=1}^n y_i\right)^{k}}} \\ &= \frac{\pi_j \frac{y_r^{x_r-1}}{\Gamma(x_r)} \frac{\left(\alpha+\sum_{(-r)} x_i+x_r-1\right)_{x_r}}{\left(\beta+\sum_{i=1}^n y_i\right)^{x_r}}}{\sum_{k=1}^J \pi_k \frac{y_r^{k-1}}{\Gamma(k)} \frac{\left(\alpha+\sum_{(-r)} x_i+k-1\right)_k}{\left(\beta+\sum_{i=1}^n y_i\right)^{k}}}, \end{aligned} \tag{3.14}$$

with $(n)_k = n \cdot (n-1) \cdot (n-1) \cdot \cdots \cdot (n-k+1)$ denoting the Pochhammer symbol.

[2] This is really an innocuous request.

The steps of this algorithm, to which I will refer to as *Algorithm 2* (i.e. the modified version of the Gibbs sampler), are summarized in Figure 3.7.

Step 1. Simulate

$$\boldsymbol{\pi}|\boldsymbol{y}, \boldsymbol{x} \sim \mathcal{D}_J\left(\frac{1}{J}+n_1, \ldots, \frac{1}{J}+n_J\right),$$

where $n_j = \sum_{i=1}^{n} \mathbb{I}(x_i = j)$, $j = 1, \ldots, J$.

Step 2. Simulate, for every $i = 1, \ldots, n$,

$$p(x_i|y_1, \ldots, y_n, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, \pi_1, \ldots, \pi_J) = \sum_{j=1}^{J} \pi_{ij}\mathbb{I}(x_i = j), \tag{3.15}$$

where π_{ij} is given by (3.14), $j = 1, \ldots, J$.

Step 3. Update n_j, $j = 1, \ldots, J$.

Figure 3.7: *Algorithm 2*, Gibbs sampling with θ integrated out.

3.2.5 Choice of the Hyperparameters

In this section I give some suggestions on how to choose the values of the hyperparameters α and β for the prior on θ. Only this prior will be considered because I already said in Section 3.2.2 that, for modeling purposes, the weights hyperparameters $(\gamma_1, \ldots, \gamma_J)$ are set exclusively to $\left(\frac{1}{J}, \ldots, \frac{1}{J}\right)$. I then avoid to consider any other value for them.

Remind first that the θ prior is a gamma $\mathcal{G}a(\alpha, \beta)$ distribution. To understand how to choose α and β it is useful to note that the mean and variance of model (3.1) are given by (3.3) and(3.4), that I report here for convenience

$$\mu = \mathbb{E}[Y] = \sum_{j=1}^{J} \pi_j \frac{j}{\theta} \tag{3.16}$$

$$\sigma^2 = \text{var}[Y] = \sum_{j=1}^{J} \pi_j \frac{j(j+1)}{\theta^2} - \mu^2 . \tag{3.17}$$

These formulas suggest, that, once a simulated chain is available from the Gibbs sampling, the posterior distributions of the model mean and variance can be obtained just by substituting the values of $\pi_j^{(m)}$ and $\theta^{(m)}$ at each iteration m of the Gibbs sampling. So the mean and variance values at the m-th iteration are given by

$$\mu^{(m)} = \sum_{j=1}^{J} \pi_j^{(m)} \frac{j}{\theta^{(m)}} \tag{3.18}$$

$$\left(\sigma^2\right)^{(m)} = \sum_{j=1}^{J} \pi_j^{(m)} \frac{j(j+1)}{\left(\theta^{(m)}\right)^2} - \left(\mu^{(m)}\right)^2. \tag{3.19}$$

These expressions can be used to monitor if the simulated values of θ are going in the right direction. Or it is better to say that they represent a preliminary check of the goodness of the proposed model. In fact if the posterior distribution of μ and σ^2 are centered respectively on the sample mean and the sample variance then this is an indication that the model is a reasonable representation of the data at hand. As is stated in Section 3.1, this is the main objective of the chapter.

To be able to choose "good" values for α and β one should first think about the interpretation of the parameter. In the case of θ, I already stated that $\frac{1}{\theta}$ is a scale parameter for the entire model, but it is possible to say something more. Equation (3.16) for the model mean suggests in fact that

$$\theta = \frac{\sum_{j=1}^{J} \pi_j j}{\mu}. \tag{3.20}$$

To better understand this expression, consider to standardize the data dividing them by the mean μ, that is to fit the model to $\tilde{y} = y/\mu$. This allows to rewrite the previous equation as

$$\theta = \sum_{j=1}^{J} \pi_j j, \tag{3.21}$$

from which it is possible to interpret θ as the average of the components labels weighted with the mixture weights.

A similar informal interpretation can be provided also for α and β. Usually this statements are drawn when the posterior distribution is available in closed form. This is not the case here for θ. However some remarks can be given by examining

the full conditional distribution (3.8). It follows in fact that

$$\begin{aligned}\mathbb{E}[\theta|\boldsymbol{y},\boldsymbol{x}] &= \frac{\alpha+\sum_{i=1}^{n}x_i}{\beta+\sum_{i=1}^{n}y_i} \\ &= \frac{\beta}{\beta+\sum_{i=1}^{n}y_i}\cdot\frac{\alpha}{\beta}+\frac{\sum_{i=1}^{n}y_i}{\beta+\sum_{i=1}^{n}y_i}\cdot\frac{\sum_{i=1}^{n}x_i}{\sum_{i=1}^{n}y_i} \\ &= \omega\cdot\frac{\alpha}{\beta}+(1-\omega)\cdot\frac{\bar{x}}{\bar{y}}, \end{aligned} \tag{3.22}$$

where $\omega = \frac{\beta}{\beta+\sum_{i=1}^{n}y_i}$. Note that the last expression is a linear convex combination of the θ prior mean and a sample estimate of it (see (3.20)). ω is then interpretable as the weight given to the prior information, while $(1-\omega)$ is the weight of the sample information borne by the sample. Moreover notice that when the data are transformed as $\widetilde{y} = y/\mu$, these expressions can be further simplified because in that case $\sum_i \widetilde{y}_i = n$, hence

$$\begin{aligned}\mathbb{E}[\theta|\widetilde{\boldsymbol{y}},\boldsymbol{x}] &= \frac{\alpha+\sum_{i=1}^{n}x_i}{\beta+n} \\ &= \frac{\beta}{\beta+n}\cdot\frac{\alpha}{\beta}+\frac{n}{\beta+n}\cdot\bar{x}. \end{aligned} \tag{3.23}$$

To sum up one can conclude that β represents the *prior sample size* and α determines the θ prior mean value. When both $\alpha \to 0$ and $\beta \to 0$ the prior becomes noninformative.

For a given value of J, a strategy for choosing α and β is reported below:

1. Calculate the quantity $\widetilde{\theta} = \frac{J}{\max(y_1,\ldots,y_n)}$ and check that $\frac{1}{\widetilde{\theta}} \leq \min(y_1,\ldots,y_n)$; the idea is that on average θ should take values that allow the set of gamma distributions in (3.1) to completely cover the observed range (the last gamma distribution should have a mean not smaller than the maximum observation and the first gamma distribution a mean not greater than the minimum observation). $\widetilde{\theta}$ is then a guess for the prior mean $\frac{\alpha}{\beta}$.

2. Choose a value k for the weight of the prior information ω in (3.22). Values between 0.2 and 0.5 are usually reasonable choices. Then choose β as $\frac{k\cdot\sum_{i=1}^{n}y_i}{1-k}$.

3. Then α can be set by rounding to the closest integer[3] the quantity $\widetilde{\theta}\cdot\beta$.

[3]The rounding is needed because of the assumption used to get (3.14).

Last but not least some words about J. The goodness of fit of the gamma mixture model is the result of the interplay among the grid of means

$$\frac{1}{\theta}, \quad \frac{2}{\theta}, \quad \cdots, \quad \frac{J-1}{\theta}, \quad \frac{J}{\theta},$$

the grid of variances

$$\frac{1}{\theta^2}, \quad \frac{2}{\theta^2}, \quad \cdots, \quad \frac{J-1}{\theta^2}, \quad \frac{J}{\theta^2},$$

and the sequence of ordered observations. The idea is that these grids should contain sufficient elements (the gamma distributions) to fit the data. So J and θ together affect the final result. Thus it is difficult to provide general advices, but the best solution is a careful calibration of J to every single case is (usually two o three attempts suffice). Sometimes a transformation of the data (like a log, a power or a root) can be useful.

3.3 Data

The dataset on which I based both the simulation study and the data analysis shown in the next two sections is the Medicare Current Beneficiary Survey (MCBS)[4]. It is a continuous, multipurpose survey of a U.S. nationally representative sample of Medicare beneficiaries (Medicare is a national health insurance program that provides coverage for people aged 65 or older, some people under age 65 with disabilities and for people with permanent kidney failure requiring dialysis or a kidney transplant). The central goals of MCBS are to determine expenditures and sources of payment for all services used by Medicare beneficiaries. The sample for MCBS is drawn from the Medicare enrollment file. Newly eligible beneficiaries are added to the sample once a year. In the dataset 26,834 hospitalizations distributed on four years (from 1999 to 2002) were available, for a total of 9,782 people and an average of 3,900 people per year.

[4]Sources: Johns Hopkins School of Public Health and Centers for Medicare & Medicaid Services

3.4 Simulation Study

The simulation study reported in this section aims at assessing the predictive performance of the gamma mixture model in the estimation of the right tail of the medical expenditures distribution for the MCBS dataset. Some words are needed before describing the study. First I am conscious that to conduct a complete comparison of the gamma mixture model for this kind of predictions it should be compared with some methods specifically structured for tail estimation, like a mixture of heavy tailed distributions (for example of Pareto distributions), maybe with tails that converge at different rates. This has not been done here because the reasons that motivated the application were not simply the estimation of the tails of the distribution but also the provision of an overall fit to the density of the data. Methods for estimation of upper order statistics, often called *extreme value theory (EVT)* (see Embrechts et al. [26] and Beirlant et al. [6]), have been conceived to model the distribution of the maximum or of other extreme values, without any intention to fit also the density of the data. The second comment is that I am aware also that a simulation study with 50 sub-samples is certainly not enough to draw firm conclusion. The enlargement of this simulation study is needed, but I expect that using a larger simulation will result in a clearer indication of the goodness of the gamma mixture.

3.4.1 Setup of the study

From the complete MCBS dataset described above 50 sub-samples of size equal to 10% of the original sample, the *training sets*, have been randomly drawn, while the remaining 90% constitute the *test sets*.

On each training set three estimators of the tail probability $\hat{p} = \mathbb{P}(y^* > k|\boldsymbol{y})$ have been calculated:

- the empirical distribution function (ECDF),
- a fitted log-normal distribution (LN),
- a gamma mixture model (MG).

The estimators of the tail probability for the ECDF and LN cases are straightfor-

ward, while for the MG

$$\mathbb{P}(y^* > k|\boldsymbol{y}) = \int \mathbb{P}(y^* > k|\boldsymbol{y}, \theta, \boldsymbol{\pi}) f(\theta, \boldsymbol{\pi}|\boldsymbol{y})\, d\theta\, d\boldsymbol{\pi}\,, \tag{3.24}$$

is the quantity of interest, where $F_j(y|\theta)$ is the distribution function of a $\mathcal{G}a(j, \theta)$ random variable. The predictive probability (3.24) can be estimated with

$$\begin{aligned}\widehat{\mathbb{P}}(y^* > k|\boldsymbol{y}) &= \frac{1}{M}\sum_{m=1}^{M} P\left(y^* > k \,\middle|\, \theta^{(m)}, \boldsymbol{\pi}^{(m)}\right) \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{J} \pi_j^{(m)}\left[1 - F_j\left(k \,|\theta^{(m)}\right)\right].\end{aligned}$$

On each test set the sample proportion $\#\{y > k\}/n_{\text{test}}$ is calculated and the analysis is repeated for different values of the threshold k, in particular for k equal to \$10,000, \$15,000, \$20,000, \$30,000, \$50,000, \$80,000 and \$100,000.

Before calculating the tail probability with the MG model the data have been transformed with a cubic root transformation. This dose not introduce any bias in the results since

$$\mathbb{P}(X > k) = \mathbb{P}\left(\sqrt[3]{X} > \sqrt[3]{k}\right).$$

The parameters chosen for the simulation are: J=200, α=96,000, β=32,000, 11,000 iterations (1,000 of which for burn-in), 100 bootstrap replications for the ECDF and LN estimators. These have been chosen by following the indications provided in Section 3.2.5.

3.4.2 Results

In Figures 3.8 and 3.9 the comparison of the estimators with respect to the test set is reported. Each plot contains the output of the simulation for a certain value of the threshold k. The vertical axes contain the absolute value of the difference between the tail probability estimated on the training set and on the test set. The lower the difference the best the prediction. Squares indicate the performance of the log-normal model, triangles of the empirical distribution function and circles of the gamma mixture. In each sub-sample of the simulation study the estimator with the best performance is indicated by the correspondent marker which is filled. Legends summarize for each threshold value the number of simulations in which each estimator has been the best.

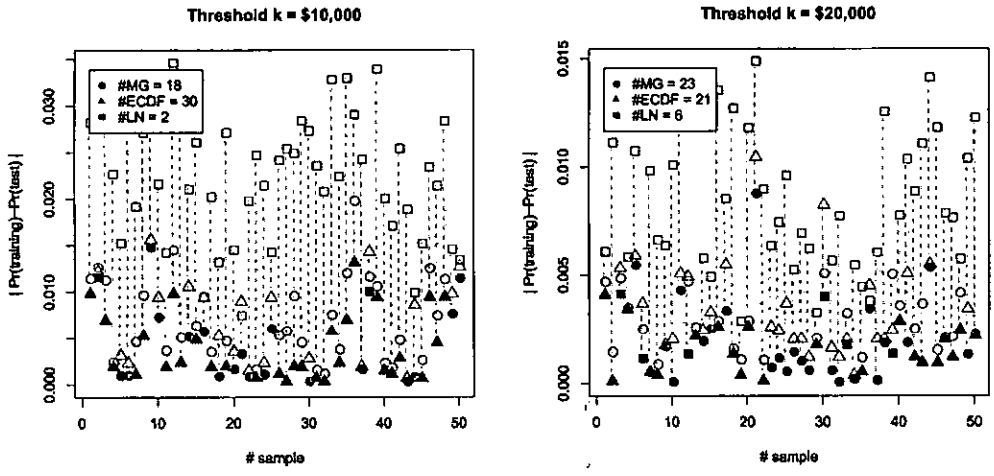

Figure 3.8: Predictive performance comparison of the estimators.

Figures 3.8 and 3.9 indicate that the mixture of gamma performs better than the empirical distribution as long as the threshold value increases. The log-normal model is always the worst. The main reason for this result is that the log-normal distribution is not sufficiently heavy-tailed to mimic the right tail of the these data.

In Figures 3.10 and 3.11 for each estimator the following quantities are reported:

- bias $= \frac{\sum_{\ell=1}^{50} \hat{p}^{(\ell)}}{50} - p_{\text{TRUE}}$, where p_{TRUE} is the sample proportion $\frac{\#\{y>k\}}{n}$ from the whole sample,
- mse $= \frac{\sum_{\ell=1}^{50} \left[\hat{p}^{(\ell)} - p_{\text{TRUE}}\right]^2}{50}$,
- relative bias (in %) $= \frac{\frac{\sum_{\ell=1}^{50} \hat{p}^{(\ell)}}{50} - p_{\text{TRUE}}}{p_{\text{TRUE}}} = \frac{\text{bias}}{p_{\text{TRUE}}}$,
- relative mse (in %) $= \frac{\text{mse}_{\text{ECDF}} - \text{mse}}{\text{mse}_{\text{ECDF}}}$.

These figures allow to assess the statistical performance of the estimators. In each plot the tail probability estimates for a threshold value are represented through a box plot for each estimator. The width of each box plot is proportional to the mean squared error. The horizontal dashed line is the estimate of the tail probability on the whole sample (this can be considered as the *true* value of the tail probability). The means are indicated with asterisks and joined with a solid

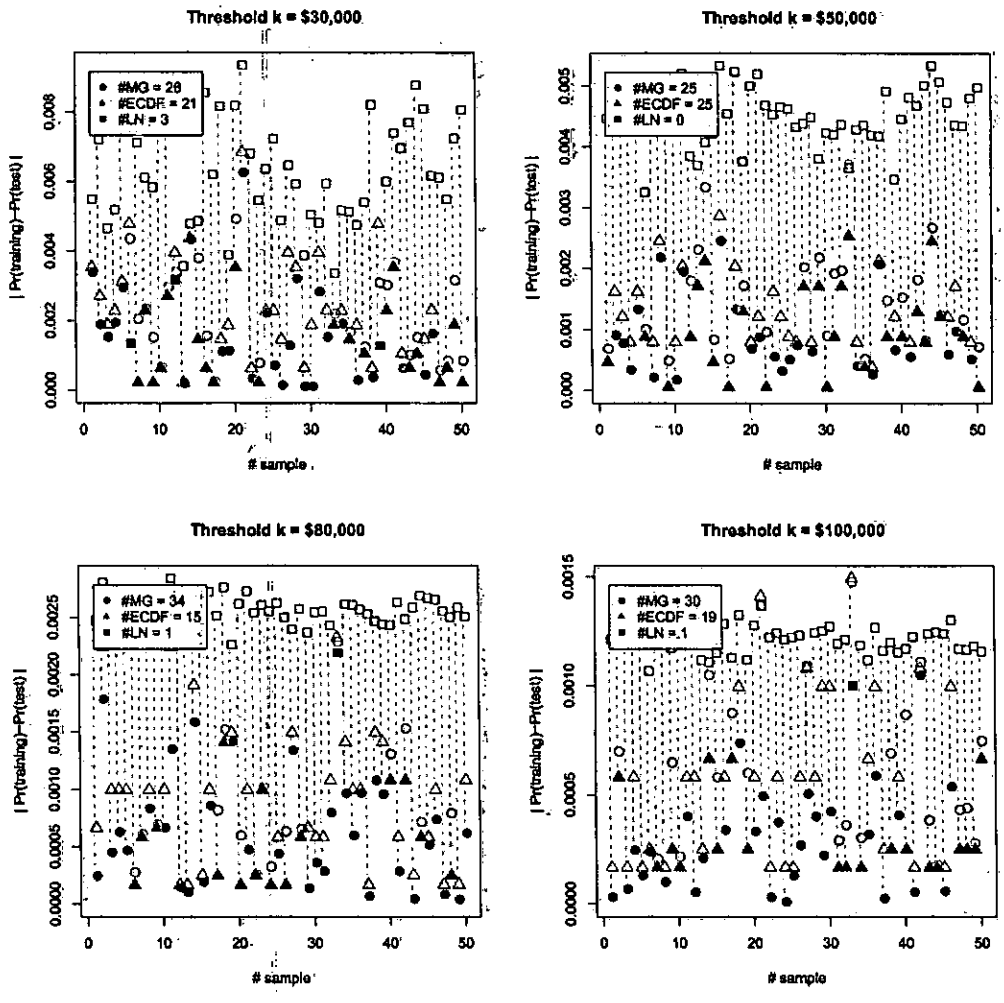

Figure 3.9: Predictive performance comparison of the estimators.

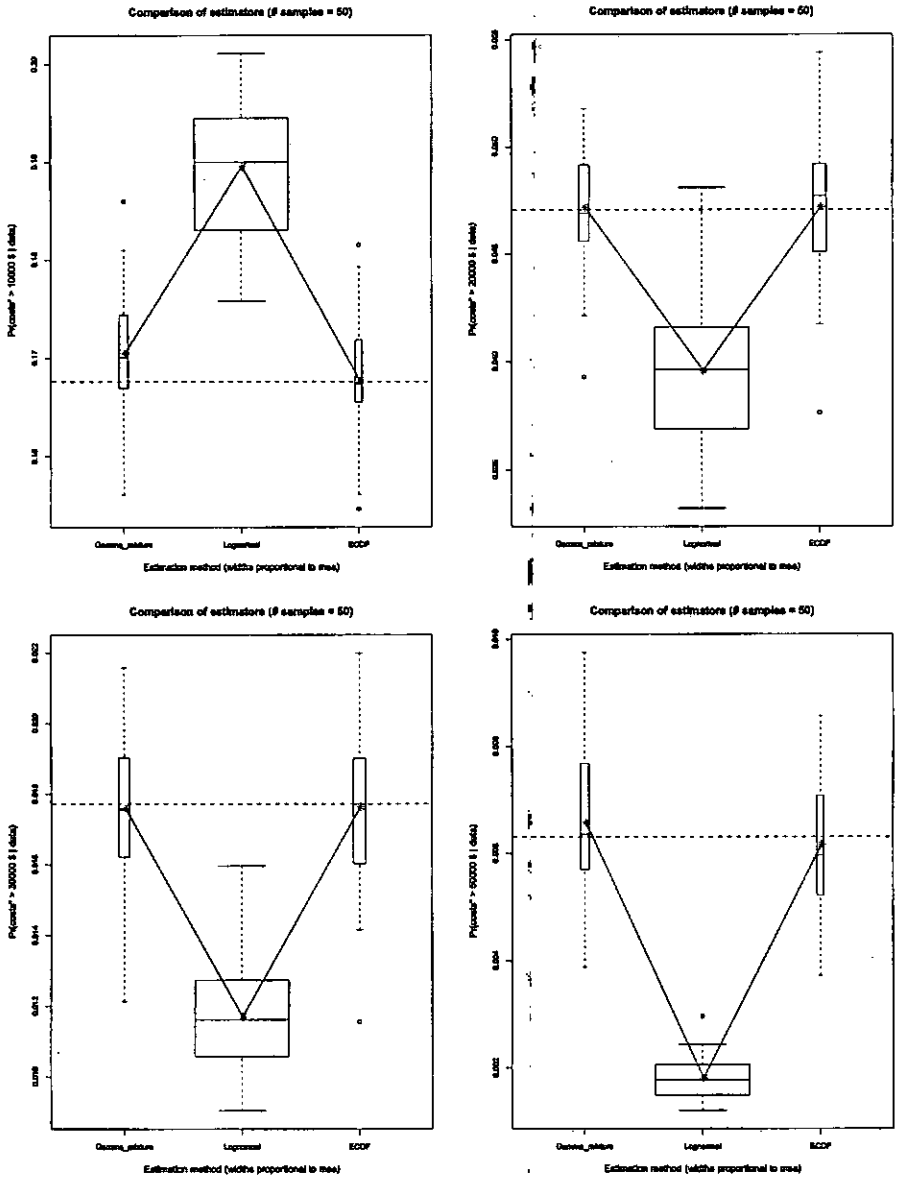

Figure 3.10: Statistical comparison of the estimators.

line. From these figures the conclusion is that the gamma mixture is alternatively more or less biased than the empirical distribution function for different values of the threshold, but more efficient for almost all the thresholds. This can be further checked by inspecting Figures 3.12 and 3.13.

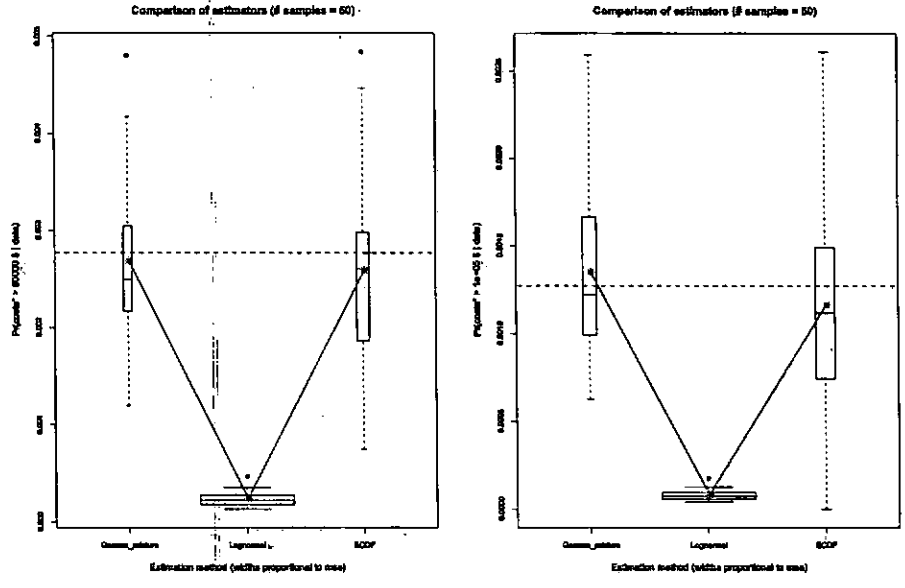

Figure 3.11: Statistical comparison of the estimators.

Note that in the relative mean squared error plot positive values correspond to greater efficiency of the estimator. From all these pictures results that the log-normal model is by far the most biased and less efficient.

3.5 Data Analysis

In this section I provide a complete data analysis of the MCBS dataset presented above. The aim is to provide an estimation of the risk to exceed a given threshold k for the medical costs in a single hospitalization. While in the previous simulation study all the available data have been used, in the next data analysis only the first hospitalization for each case (i.e. for each subject with a smoking attributable disease) is used. The reason for this choice is that the hospitalizations for each subject are evidently not independent and nothing has been done in the model for taking into account this dependence. The size of this reduced sample is $n = 2,833$.

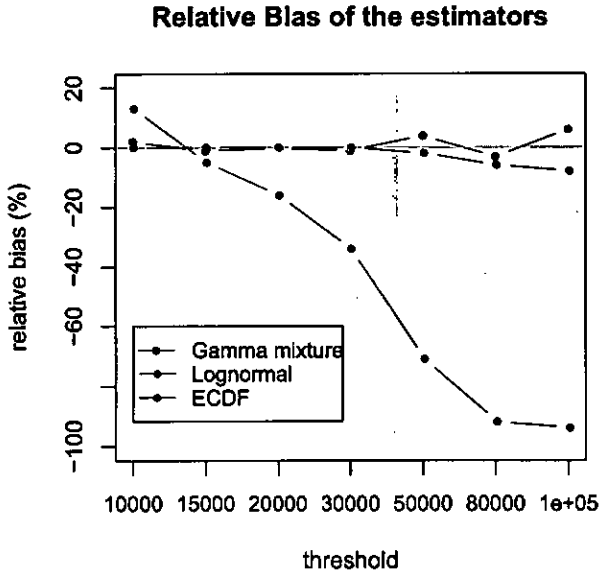

Figure 3.12: Relative bias of the estimators for different threshold values.

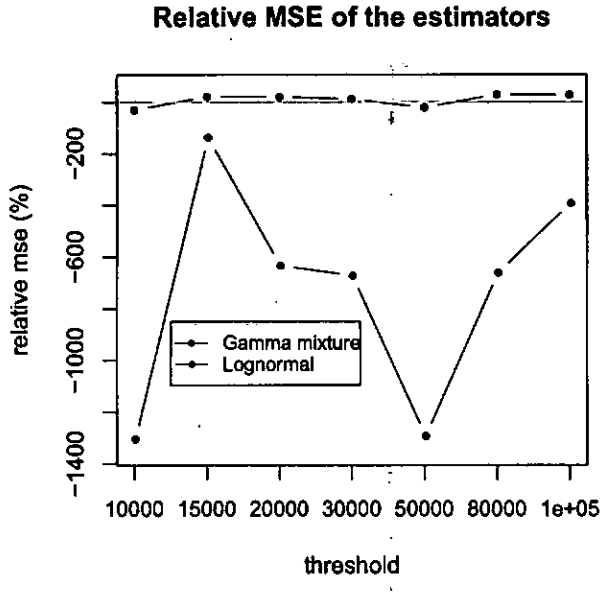

Figure 3.13: Relative mean squared error of the estimators for different threshold values.

The parameters for the Gibbs sampler have been set to J=200, α=44,520, β=12,720; 6,000 sampling iterations (1,000 of which for burn-in). The data have been transformed again with a cubic root transformation.

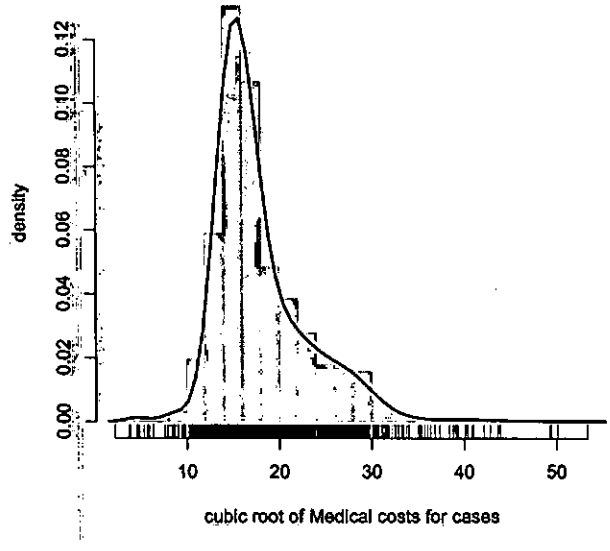

Figure 3.14: Fit of the gamma mixture model.

From the Figure (3.14) and (3.15) it is clear that the gamma mixture provides a very good representation of the data at hand.

In the next four pictures the output of the Gibbs sampling is reported. Figure 3.16 allows to understand how the model handles the number of components of the mixture. Note that even if J=200 components were available the estimation process select every time just a small subset of them. A posteriori the number of selected components is between 8 and 18. The solid line overimposed on the histograms are the correspondent kernel density estimators.

Figure 3.19 reports the plot of (3.18) and (3.19), where the vertical dashed lines indicate the sample mean and the sample variance.

Figure 3.20 is the final result of the data analysis. It reports the estimates of the tail probability for different values of the threshold, i.e. the "risk" of exceeding a given medical costs threshold in a single hospitalization. In the plot the 95% credible intervals for each estimate are also reported. As expected the risk decreases

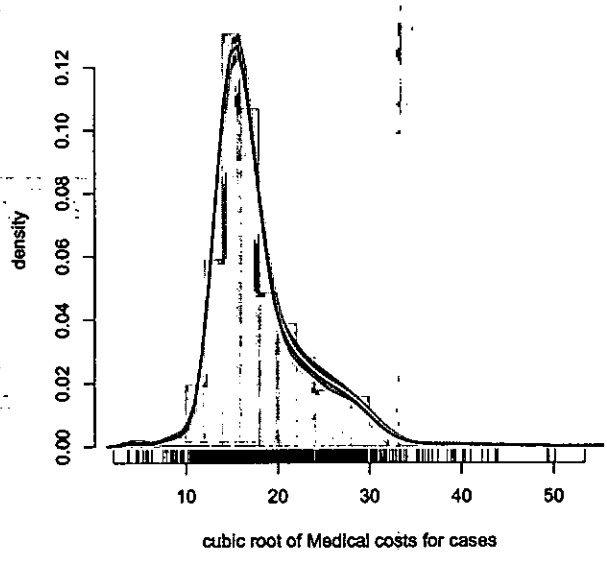

Figure 3.15: Fit of the gamma mixture model with credible interval.

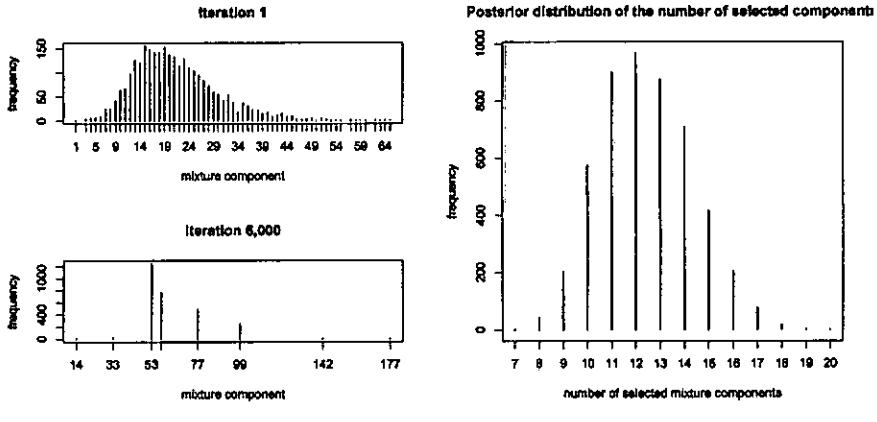

Figure 3.16: Missing data and number of selected mixture components.

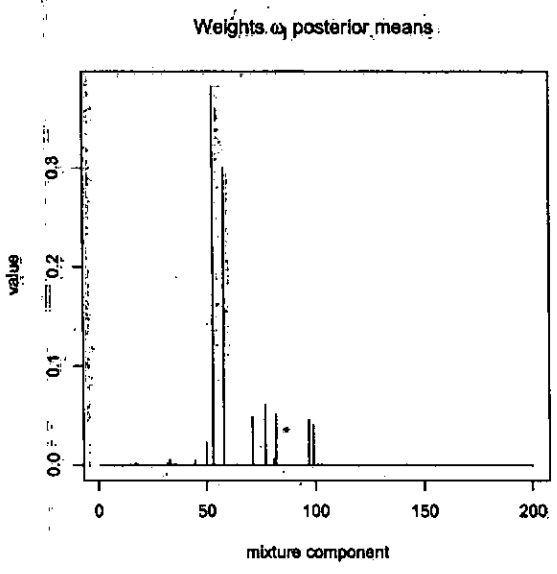

Figure 3.17: Posterior mean of the mixture weights.

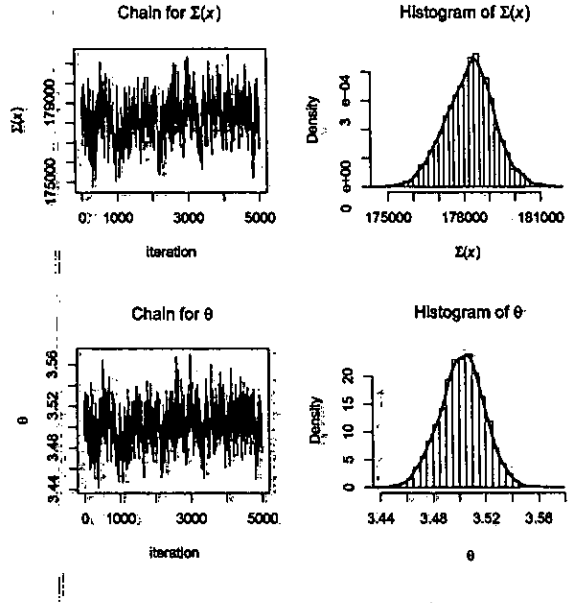

Figure 3.18: Posterior distribution for $\sum_{i=1}^{n} x_i$ theta.

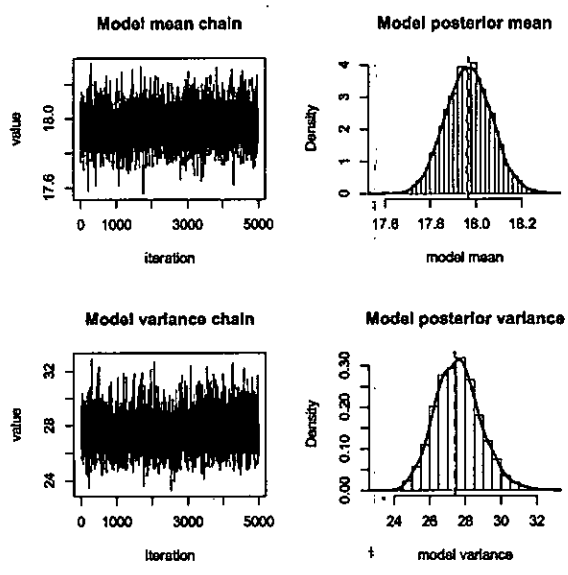

Figure 3.19: Posterior distribution for the model mean and variance.

as the threshold increases (in other words this is the estimated right tail of the medical costs distribution). Some caution should be used to conclude that the estimates for higher values of the threshold are more reliable.

3.6 Discussion

In this chapter I developed a parsimonious mixture model that involves just an additional parameter with respect to the set of mixture weights. This advantage is counterbalanced by fixing J at a high value (this allows to have a sufficiently large "basis" in the model). The results provided in the data analysis suggest that the model is able to select the appropriate number of components.

Even if it has not been formally proved, in this chapter I have shown how to increase the efficiency of the Gibbs sampler used for the estimation of the mixture parameters

The simulation study has shown that the gamma mixture model is able to outperform in a predictive sense other available estimators of the tail probability of a medical costs distribution. I have shown in particular that the gamma mixture model is more efficient than the competitors, which I included in the study.

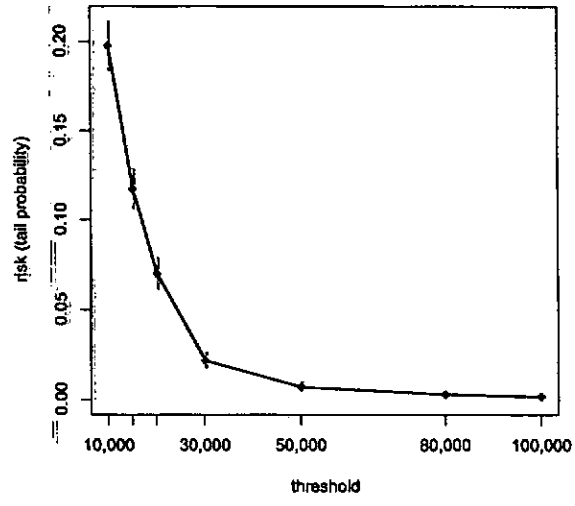

Figure 3.20: Risk to exceed a given medical costs threshold in a single hospitalization with 95% credible intervals.

Future work will regard a more extensive simulation study, both by including other methods more suited for tail estimation and by enlarging the study to more than 50 sub-samples. Part of future research will also be the assessment of the sensitivity of the results to the prior choice.

Chapter 4

Bayesian Smooth QUAntile Ratio Estimation (B-SQUARE)

4.1 Introduction

In this chapter I present a flexible Bayesian method based on the idea of borrowing strength across two samples to get more efficient estimates of certain quantities of interest. The approach is based on the mixture of gamma distributions described in the previous chapter and is build upon the previous work on SQUARE of Dominici et al. (2005) [22]. I show an application of the method to the estimation of a tail probability related to medical costs. The dataset used in the data analysis is the National Medical Expenditures Survey (NMES).

4.2 Background: Smooth QUAntile Ratio Estimation (SQUARE)

4.2.1 Basic idea

SQUARE (Dominici et al. [22]) is a semiparametric approach for estimating the mean difference between two skewed distributions. The work was motivated by an application involving the estimation of the medical costs attributable to smoking. The available data were medical costs paid by a national health insurance program for subjects with and without smoking attributable diseases (i.e. lung cancer

and coronary obstructive pulmonary disease). The challenge of that situation was represented by the following unique features of the data:

1. both medical costs distributions were highly skewed,
2. the set of cases (subjects with smoking attributable diseases) were much smaller than the set of controls,
3. in the samples there were many subjects with zero expenditures.

One possible approach (quite used in practice) to deal with these problems is to transform the data, typically with a logarithm, and assume that these have a normal distribution (in other words assume that the medical expenditures are log-normally distributed). This way to proceed presents the drawback of assuming that the distribution of the log-transformed costs is symmetric. One way to view the limitation of this approach is using a Q-Q plot (quantile-quantile plot). In this graph the sample quantiles of the two distributions are plotted against each other (Wilk and Gnanadesikan [107], Doksum and Sievers [21], Parzen [71], Wilcox [106], Gilchrist [33]). Under the assumption that the two sub-populations are log-normally distributed, that is $\log Y_1|\mu_1, \sigma_1^2 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $\log Y_2|\mu_2, \sigma_2^2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, it follows that

$$\log Q_g(p) = \mu_g + \sigma_g \Phi^{-1}(p)\,, \quad g = 1, 2\,,$$

and then the sample log-quantiles should satisfy the linear relation

$$\log Q_1(p) = \left(\mu_2 - \frac{\sigma_2}{\sigma_1}\mu_1\right) + \frac{\sigma_2}{\sigma_1} \log Q_2(p)\,.$$

This hypothesis is often belied by the data. In Figure 4.1 an example about medical costs for cases and controls is reported. From the figure it is evident that the assumption of log-normal expenditures, or of a linear relation between the quantiles, is not correct.

A proposal to solve the problem is to fit a non-linear smooth function to the Q-Q plot. However this idea has the undesirable property of conditioning on Q_2 (as in any regression approach) rather than treating the two quantile functions symmetrically. Moreover the smooth function would take values on the positive real line making the choice of its degrees of freedom critical. On addition, by

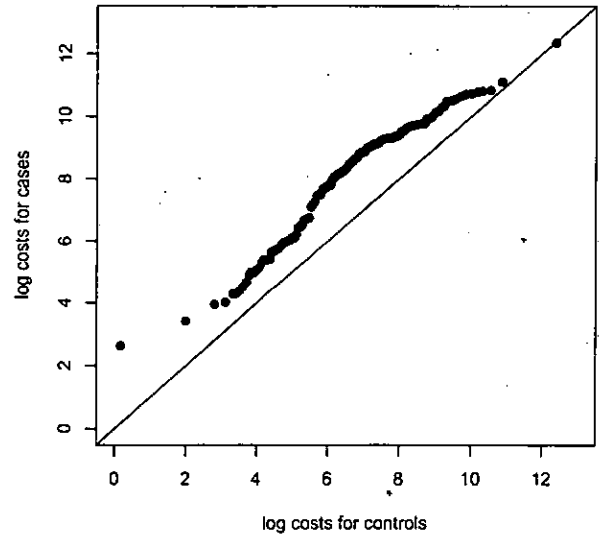

Figure 4.1: Q-Q plot of log non-zero medical expenditures.

smoothing the Q-Q plot one would estimate the mean difference by the sample mean difference which is unbiased but highly variable.

In SQUARE an alternative approach is proposed, where the assumption is to smooth the log quantile ratio across percentiles, that is assume

$$\log \frac{Q_1(p)}{Q_2(p)} = X(p,\lambda)\boldsymbol{\beta}\,, \quad 0 < p < 1\,, \tag{4.1}$$

where $X(p,\lambda)$ is the design matrix of a smooth function and λ are its degrees of freedom. For an example see at Figure 4.2, where the data are the same as in Figure 4.1.

This proposal has the advantage to treat the two quantile functions symmetrically and to spend the λ degrees of freedom over the space $(0,1)$, hence imposing stronger constraints in the tails where little information is available in the smaller sample.

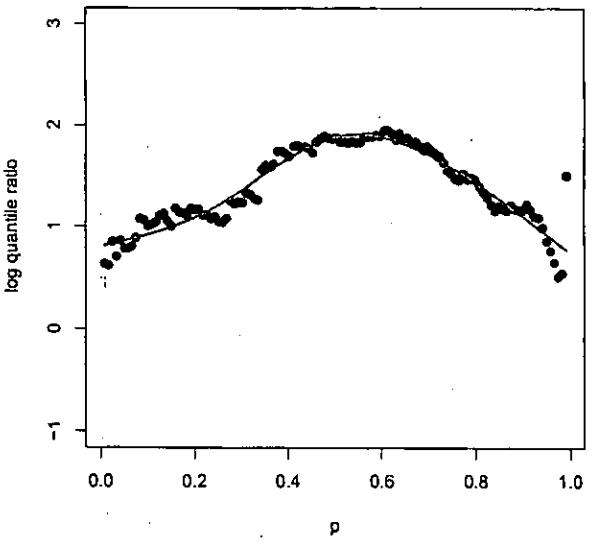

Figure 4.2: Log quantile ratio across percentiles with a smooth fitted function.

4.2.2 Definition

Consider two positive random variable Y_1 and Y_2 with cumulative distribution functions (cdf) F_1 and F_2 and quantile functions Q_1 and Q_2, where

$$Q_g(p) \equiv F_g^{-1}(p) \equiv \inf\{y : F_g(y) \geq p\}$$

for $0 < p < 1$ and $g = 1, 2$. The aim of SQUARE is to estimate the following quantity

$$\Delta = \mathbb{E}[Y_1] - \mathbb{E}[Y_2] = \int_0^1 \{Q_1(p) - Q_2(p)\}\, dp \tag{4.2}$$

assuming

$$\log \frac{Q_1(p)}{Q_2(p)} = X(p, \lambda)\boldsymbol{\beta}\,, \qquad 0 < p < 1\,, \tag{4.3}$$

that is the ratio of the quantiles is a smooth function of the percentiles with λ degrees of freedom.

The SQUARE assumption (4.3) allows to rewrite Δ as

$$\Delta = \int_0^1 Q_1(p)\left[1 - e^{-X(p,\lambda)\boldsymbol{\beta}}\right] dp = \int_0^1 Q_2(p)\left[e^{X(p,\lambda)\boldsymbol{\beta}} - 1\right] dp\,. \tag{4.4}$$

The most important point here for the development of the model in the next sections is that SQUARE is a nonparametric estimator, because no distributional assumptions are made neither for Y_1 nor for Y_2.

4.2.3 Estimation

Let us have two samples[1] $\boldsymbol{y}_1 = (y_{11}, \ldots, y_{1n_1})$ and $\boldsymbol{y}_2 = (y_{21}, \ldots, y_{1n_2})$ respectively from F_1 and F_2, define the order statistics for the two samples as $\boldsymbol{y}_{(g)} = (y_{g(1)}, \ldots, y_{g(n_g)})$, $g = 1, 2$ and suppose first that $n_1 = n_2$ (the case $n_1 << n_2$ will be discussed later).

The estimation procedure in SQUARE is composed of two steps. In the first step $\boldsymbol{\beta}$ is estimated by ordinary least squares (OLS) assuming the regression model

$$\log \frac{y_{1(i)}}{y_{2(i)}} = X(p_i, \lambda)\boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n, \tag{4.5}$$

where $n = \min(n_1, n_2)$ and $p_i = i/(n+1)$. In the original formulation of SQUARE it is assumed that $X(p_i, \lambda) = \sum_{k=0}^{\lambda} X_k(p_i)\beta_k$ where $X_k(p)$ are orthonormal basis functions with $X_0(p) \equiv 0$.

In the second step define $\boldsymbol{u}_1 = (\boldsymbol{y}_{(1)}, \boldsymbol{y}^*_{(1)})$ and $\boldsymbol{u}_2 = (\boldsymbol{y}_{(2)}, \boldsymbol{y}^*_{(2)})$, two samples of size $2n$, where

$$\begin{aligned} y^*_{1(i)} &= y_{2(i)} e^{X(p_i,\lambda)\widehat{\boldsymbol{\beta}}} \\ y^*_{2(i)} &= y_{1(i)} e^{-X(p_i,\lambda)\widehat{\boldsymbol{\beta}}}, \end{aligned}$$

with $\widehat{\boldsymbol{\beta}}$ estimated from the previous step and $i = 1, \ldots, n$. Then estimate Δ as

$$\begin{aligned} \widehat{\Delta}(\lambda) &= \bar{u}_1 - \bar{u}_2 \\ &= \frac{1}{2n}\sum_{i=1}^{n} y_{1(i)}\left[1 - e^{-X(p_i,\lambda)\widehat{\boldsymbol{\beta}}}\right] + \frac{1}{2n}\sum_{i=1}^{n} y_{2(i)}\left[e^{X(p_i,\lambda)\widehat{\boldsymbol{\beta}}} - 1\right]. \end{aligned} \tag{4.6}$$

Hence Δ is estimated as the sample mean difference of the two "extended samples" $\boldsymbol{u}_g$, $g = 1, 2$, that is the original vector or order statistics $\boldsymbol{y}_{(g)}$ augmented with the mapped values $\boldsymbol{y}^*_g$ from the other sample. Note that $\widehat{\Delta}(\lambda)$ is symmetric with respect to the two samples and that it is a linear combination of order statistics

[1]Note that the two distributions F_1 and F_2 are linked due to the SQUARE assumption on the two quantile functions Q_1 and Q_2. So the two sample are conditionally independent given the two distribution functions F_1 and F_2.

with weights estimated from the data, so it is related to L-estimators (see Serfling [93] and Huber [40]).

To extend SQUARE to the case in which $n_1 < n_2$ the authors propose to calculate $\widehat{\Delta}(\lambda)$ as in (4.6) but replacing $\boldsymbol{y_2}$ by $\boldsymbol{q_2}$, the linear interpolation of the order statistics $y_{2(i)}$ to the grid of points $p_{1i} = i/(n_1+1)$, $i = 1, \ldots, n_1$. A similar substitution can be done if $n_1 > n_2$.

The solution adopted to take into account also the problem of possible zero-cost observations is to define $\pi_g = \mathbb{P}(Y_g > 0)$ and $\mu_g = \mathbb{E}[Y_g | Y_g > 0]$, for $g = 1, 2$. Then redefine the mean difference as $\Delta = \pi_1\mu_1 - \pi_2\mu_2$. It follows that Δ can now be estimated by

$$\widehat{\Delta}(\lambda) = \widehat{\pi}_1\bar{u}_1 - \widehat{\pi}_2\bar{u}_2\,,$$

where $\widehat{\pi}_g$ is the proportion of non-zero costs for sample g.

The estimation of λ is done by B-fold cross validation (see Efron and Tibshirani [25]).

4.2.4 Special cases

Two special cases arise for specific choices of $s(p, \lambda; \boldsymbol{\beta}) = X(p, \lambda)\boldsymbol{\beta})$ and of the basis functions $X_k(p)$. In particular here it is of interest to consider two of these situations.

1. If $Y_g \sim \mathcal{U}[0, \theta_g]$, $g = 1, 2$, then $Q_1(p)/Q_2(p) = \theta_1/\theta_2$, that is the smoothing function of the log quantile ratio is a constant, and $\Delta = (\theta_1 - \theta_2)/2$. The SQUARE estimate of Δ is then

$$\widehat{\Delta}(\mathcal{U}nif, \lambda = 0) = \frac{1}{2}\left[\overline{y}_1\left(1 - e^{-\widehat{\beta}_0}\right) - \overline{y}_2\left(1 - e^{\widehat{\beta}_0}\right)\right],$$

   where $\widehat{\beta}_0 = \overline{\log y_1} - \overline{\log y_2}$.

2. If $Y_g \sim \mathcal{L}n(\mu_g, \sigma_g^2)$, $g = 1, 2$, then $\log Q_1(p)/Q_2(p) = \beta_0 + \beta_1\Phi^{-1}(p)$, that is the smoothing function of the log quantile ratio is linear in $\Phi^{-1}(p)$ (the quantile function of a standard normal random variable), where $\beta_0 = (\mu_1 - \mu_2)$ and $\beta_1 = (\sigma_1 - \sigma_2)$, and $\Delta = \exp\{\mu_1 + \sigma_1^2/2\} - \exp\{\mu_2 + \sigma_2^2/2\}$. The SQUARE estimate $\widehat{\Delta}(\mathcal{L}n, \lambda = 1)$ is obtained by fitting the regression model (4.5) with $X_0(p) = 1$ and $X_1(p) = \Phi^{-1}(p)$, and using the estimated $\boldsymbol{\beta}$ in (4.6). Note that $\widehat{\Delta}(\mathcal{L}n, \lambda = 1)$ is not the MLE of Δ, which is defined as

$\widehat{\Delta}_{\mathcal{L}n} = \exp\{\overline{\log y_1} + s_1^2/2\} - \exp\{\overline{\log y_2} + s_2^2/2\}$, where s is the standard deviation of the log-transformed data.

4.2.5 Statistical properties

In their work on SQUARE Dominici et al. [22] show that under certain conditions $\hat{\beta}$ is consistent and asymptotically normal and that $\widehat{\Delta}$ is asymptotically normal as well. In both the cases they also report the explicit form for the asymptotic variances. For a detailed proof of these statements see Cope [14].

From the simulation study performed in the original work by Dominici et al. [22] it follows that SQUARE is slightly biased but far more efficient then other competing estimator of the mean difference.

4.3 The Bayesian SQUARE (B-SQUARE) Model

The main idea implicit in the SQUARE approach is to borrow strength across the two samples in order to estimate more efficiently the mean difference Δ. This is particularly important because one sample (typically the set of cases) is often much smaller than the other. This approach can in principle be further improved by incorporating it in a Bayesian setting. This can be done by following either a parametric or a nonparametric formulation. In this thesis I will use a parametric, but flexible, Bayesian approach. On the one side, the introduction of parameters in SQUARE has the disadvantage to impose some constraints on the model. On the other side, a Bayesian approach takes into account also the uncertainty of these parameters, allowing the possibility to incorporate prior information that is not possible in a frequentist framework. As I will show, sometimes this trade-off is in favor of the Bayesian approach and other times in favor of the frequentist approach. To summarize, I am interested here in understanding how much one can gain in terms of efficiency by including prior information in the SQUARE setting.

4.3.1 Likelihood

Consider two conditionally independent positive random variables Y_1 and Y_2 given F_1 and F_2 with quantile functions Q_1 and Q_2. In the application, for example, Y_1

is the medical bill paid by a national health program (like the Medicare program in US) for the cases (typically subjects with a disease) in each hospitalization and Y_2 is the medical bill for the controls. Assume that the SQUARE assumption

$$\log \frac{Q_1(p)}{Q_2(p)} = X(p,\lambda)\,\beta, \qquad 0 < p < 1 \tag{4.7}$$

holds, where λ are the degrees of freedom of the smooth function whose design matrix is $X(p,\lambda)$. I will discuss different choices for this smooth function, the most frequently used being a natural cubic spline or a polynomial.

Assume also that $Y_1|\eta \sim F_1(\cdot;\eta)$, where F_1 is a given distribution function and η a set of parameters. In the next sections I will mainly use $F_1 \equiv \mathcal{M}ix\mathcal{G}a(\pi,\theta|J)$, but other choices will also be considered. The idea here is that the specification of F_1 together with the link between Q_1 and Q_2 implicitly determines also a distributional assumption for F_2. This is a consequence of the well known fact (see Gilchrist [33]) that to describe the probability structure of a (continuous) random variable X one can indifferently choose one of the following alternatives:

- *cumulative distribution function* $F(x) = \mathbb{P}(X \le x)$,
- *density function* $f(x) = \frac{dF(x)}{dx}$,
- *quantile function* $Q(p) = F^{-1}(p) \equiv \inf\{x : F(x) \ge p\}$,
- *quantile density function* $q(p) = \frac{dQ(p)}{dp}$,
- *density quantile function* $f_p(x) \equiv f(Q(p))$.

Unfortunately often it is not possible to convert a model stated in one form to another. For example, if $X \sim \mathcal{L}n(\mu,\sigma^2)$, then the cdf is known explicitly, but the quantile function is not, since

$$Q(p) = e^{\mu+\sigma\,\Phi^{-1}(p)},$$

where $\Phi^{-1}(p)$ is the quantile function of a standard normal random variable, that has no explicit form. For details about some properties of and relations among these objects see Appendix A.

I now show that given the assumptions above it is possible to obtain the explicit form of the density for Y_2. This density is actually a density quantile function since

it expressed as a function of the percentile p. Then it will be possible to calculate the likelihood for the parameters of the model. This is something different from SQUARE since it provides a completely nonparametric estimation approach.

Note first that (4.7) implies

$$Q_1(p) = Q_2(p)\, e^{X(p,\lambda)\beta} \quad 0 < p < 1\,, \tag{4.8}$$

and it is useful to remember that for a generic random variable X with density $f(x)$ and quantile function $Q(p)$ the quantile density and density quantile functions are linked through the following relation

$$q(p) = \frac{d}{dp}Q(p) = \frac{1}{f(Q(p))} \tag{4.9}$$

(see Appendix A for more details about this definition).

Differentiate now (4.8),

$$q_1(p) = q_2(p)\, e^{X(p,\lambda)\beta} + X'(p,\lambda)\,\beta\, Q_2(p)\, e^{X(p,\lambda)\beta}, \tag{4.10}$$

where $X'(p,\lambda)$ is the derivative of $X(p,\lambda)$ with respect to p, $0 < p < 1$.

Then apply (4.9) to $q_1(p)$ and $q_2(p)$ obtaining

$$\frac{1}{f_1(Q_1(p)|\eta)} = \frac{1}{f_2(Q_2(p)|\eta,\beta)}\, e^{X(p,\lambda)\beta} + X'(p,\lambda)\,\beta\, Q_2(p)\, e^{X(p,\lambda)\beta}, \tag{4.11}$$

or

$$f_2(Q_2(p)|\eta,\beta) = \frac{f_1\left(Q_1(p)|\eta\right)}{e^{-X(p,\lambda)\beta} - f_1\left(Q_1(p)|\eta\right)\, X'(p,\lambda)\,\beta\, Q_2(p)}\,. \tag{4.12}$$

Then substituting (4.8)

$$f_2(Q_2(p)|\eta,\beta) = \frac{f_1\left(Q_2(p)\, e^{X(p,\lambda)\beta}|\eta\right)}{e^{-X(p,\lambda)\beta} - f_1\left(Q_2(p)\, e^{X(p,\lambda)\beta}|\eta\right)\, X'(p,\lambda)\,\beta\, Q_2(p)}\,, \tag{4.13}$$

with $0 < p < 1$.

A first remark is that f_2 correctly depends both upon the SQUARE parameters β and the Y_1 parameters η through the f_1 density. Secondly f_2 is expressed in terms of the percentiles p. This will cause some problems in the computation of the likelihood (see next section). Finally, note that nothing in (4.13) assures f_2 to be positive. The following constraint on the feasible β has thus to be imposed

$$X'(p,\lambda)\,\beta \le \frac{1}{f_1\left(Q_1(p)\right) Q_1(p)}\,, \quad 0 < p < 1\,. \tag{4.14}$$

This constraint is a consequence of a known property about the product of two positive quantile functions (see Appendix A, Equation (A.5)).

Using the previous result and some known facts about quantile functions (see Appendix A) it is possible to write explicitly the likelihood for the model. This is given by

$$\mathbb{L}(\eta, \beta | y_{11}, \ldots, y_{1n_1}, y_{21}, \ldots, y_{2n_2}) = \prod_{i=1}^{n_1} f_1(y_{1i}|\eta) \times \prod_{i=1}^{n_2} f_2(Q_2(p_{2i})|\eta, \beta) . \quad (4.15)$$

The drawback of this expression is that it depends on the *true* p_{2i}, defined as $p_{2i} = F_2(y_{2(i)})$, $i = 1, \ldots, n_2$. In the next section I propose a way to solve this situation.

4.3.2 Likelihood approximation

Unfortunately the likelihood (4.15) cannot be calculated directly because it involves the true percentiles p_{2i}, that is the percentiles generated by F_2, which are unknown. In particular the problem arises for the quantities $X(p_{2i}, \lambda)$ and $X'(p_{2i}, \lambda)$ needed to compute $f_2(Q_2(p_{2i})|\eta, \beta)$.

In the following steps I describe a way to go around the impasse by opportunely approximating the likelihood.

1. Since $(y_{11}, \ldots, y_{1n_1})$ is assumed to be a sample from F_1, calculate $p_{1i} = F_1\left(y_{1(i)}\right)$, for $i = 1, \ldots, n_1$.
2. Calculate $\widetilde{y}_{2(i)} = y_{1(i)} e^{-X(p_{1i}, \lambda)\beta}$, such that $(p_{1i}, \widetilde{y}_{2(i)})$ define an approximation $\widetilde{Q}_2(p)$ of the true quantile function $Q_2(p)$ evaluated at $\{p_{1i}\}_{i=1}^{n_1}$.
3. Invert $\widetilde{Q}_2(p)$ to find $(\widetilde{p}_{21}, \ldots, \widetilde{p}_{2n_2})$ such that $y_{2(i)} = \widetilde{Q}_2\left(\widetilde{p}_{2i}\right)$, $i = 1, \ldots, n_2$.

Using these calculations the (4.15) can then be approximated as

$$\begin{aligned} \mathbb{L}(\eta, \beta | y_{11}, \ldots, y_{1n_1}, y_{21}, \ldots, y_{2n_2}) &= \prod_{i=1}^{n_1} f_1(y_{1i}|\eta) \times \prod_{i=1}^{n_2} f_2(Q_2(p_{2i})|\eta, \beta) \\ &\approx \prod_{i=1}^{n_1} f_1(y_{1i}|\eta) \times \prod_{i=1}^{n_2} f_2(\widetilde{Q}_2(\widetilde{p}_{2i})|\eta, \beta) , \end{aligned}$$

where $\widetilde{Q}_2(p)$ is the approximation to the true quantile function $Q_2(p)$ and $\widetilde{p}_{2i}$ are approximations to the true $p_{2i} = F_2\left(y_{2(i)}\right)$.

Note finally that, contrarily to SQUARE, here there is no need to distinguish between the cases $n_1 = n_2$ and $n_1 \neq n_2$. The model uses all the information available from the two samples without any need to interpolate the order statistics of the larger sample with the percentile of the smaller one.

4.3.3 Special cases

It is useful at this point to examine two special cases: the first where Y_1 is a uniform random variable and the smooth function is a constant, and the second where Y_1 is a log-normal and the smooth function is linear in $\Phi^{-1}(p)$. These exercises are helpful on one side to check the correctness of the expression for $f_2(Q_2(p)|\eta, \boldsymbol{\beta})$ and on the other side to present B-SQUARE with log-normal data, that will be used later in the simulation study. These examples clearly show that knowing two of the three elements $\{F_1, F_2, \log Q_1(p)/Q_2(p)\}$ fully specify the third one.

Special case 1: F_1 Uniform and $X(p, \lambda = 0) = 1$

Assume $Y_1|\theta_1 \sim \mathcal{U}[0, \theta_1]$ and $\log Q_1(p)/Q_2(p) = \beta_0$. Thus the density, distribution and quantile functions for Y_1 are given by

$$\begin{aligned} f_1(y_1|\theta_1) &= \frac{1}{\theta_1} \mathbb{I}_{[0,\theta_1]}\{y_1\} \\ F_1(y_1|\theta_1) &= \frac{y_1}{\theta_1}, \qquad y_1 \in [0, \theta_1] \\ Q_1(p|\theta_1) &= \theta_1 p, \qquad 0 < p < 1. \end{aligned}$$

It follows then from (4.13)

$$\begin{aligned} f_2(Q_2(p)|\theta_1, \beta_0) &= \frac{\frac{1}{\theta_1} \mathbb{I}_{[0,\theta_1]}\{Q_2(p)e^{\beta_0}\}}{e^{-\beta_0}} \\ &= \frac{1}{\theta_1 e^{-\beta_0}} \mathbb{I}_{[0,\theta_1 e^{-\beta_0}]}\{Q_2(p)\}, \end{aligned}$$

which is the density of a $\mathcal{U}[0, \theta_2]$ random variable evaluated in $Q_2(p)$ with $\theta_2 = \theta_1 e^{-\beta_0}$. Note that the density of Y_2 correctly depends both on the SQUARE parameter β_0 and the Y_1 parameter $\eta = \theta_1$.

Special case 2: F_1 Log-normal and $X(p, \lambda = 1) = [1, \Phi^{-1}(p)]$

Assume $Y_1|\mu_1, \sigma_1^2 \sim \mathcal{L}n(\mu_1, \sigma_1^2)$ and $\log Q_1(p)/Q_2(p) = \beta_0 + \beta_1 \Phi^{-1}(p)$. The density, distribution and quantile functions for Y_1 are then given by

$$\begin{aligned} f_1(y_1|\mu_1, \sigma_1^2) &= \frac{1}{\sqrt{2\pi}\sigma_1} \frac{e^{-\frac{(\log y_1 - \mu_1)^2}{2\sigma_1^2}}}{y_1} \\ F_1(y_1|\mu_1, \sigma_1^2) &= \Phi\left(\frac{\log y_1 - \mu_1}{\sigma_1}\right) \\ Q_1(p|\mu_1, \sigma_1^2) &= e^{\mu_1 + \sigma_1 \Phi^{-1}(p)}, \quad 0 < p < 1. \end{aligned}$$

where $\Phi^{-1}(p)$ is the quantile function of a standard normal random variable. Then from (4.12)

$$\begin{aligned} f_2(Q_2(p)|\mu_1, \sigma_1^2, \beta_0, \beta_1) &= \frac{\frac{1}{\sqrt{2\pi}\sigma_1} \frac{e^{-\frac{\left(\mu_1 + \sigma_1\Phi^{-1}(p) - \mu_1\right)^2}{2\sigma_1^2}}}{e^{\mu_1 + \sigma_1\Phi^{-1}(p)}}}{e^{-\beta_0 - \beta_1\Phi^{-1}(p)} \left[1 - \frac{1}{\sqrt{2\pi}\sigma_1} \frac{e^{-\frac{(\mu_1 + \sigma_1\Phi^{-1}(p) - \mu_1)^2}{2\sigma_1^2}}}{e^{\mu_1 + \sigma_1\Phi^{-1}(p)}} \beta_1 \right.} \\ &\quad \left. \frac{}{\frac{1}{\frac{1}{\sqrt{2\pi}} e^{-\frac{[\Phi^{-1}(p)]^2}{2}}} e^{\mu_1 + \sigma_1\Phi^{-1}(p)}} \right] \\ &= \frac{\frac{1}{\sqrt{2\pi}\sigma_1} \frac{e^{-\frac{[\Phi^{-1}(p)]^2}{2}}}{e^{\mu_1 + \sigma_1\Phi^{-1}(p)}}}{e^{-\beta_0 - \beta_1\Phi^{-1}(p)} \left[1 - \frac{\beta_1}{\sigma_1}\right]} \\ &= \frac{\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{[\Phi^{-1}(p)]^2}{2}}}{e^{(\mu_1 - \beta_0) + (\sigma_1 - \beta_1)\Phi^{-1}(p)} \frac{(\sigma_1 - \beta_1)}{\sigma_1}} \\ &= \frac{1}{\sqrt{2\pi}(\sigma_1 - \beta_1)} \frac{e^{-\frac{[\Phi^{-1}(p)]^2}{2}}}{e^{(\mu_1 - \beta_0) + (\sigma_1 - \beta_1)\Phi^{-1}(p)}} \\ &= \frac{1}{\sqrt{2\pi}(\sigma_1 - \beta_1)} \frac{e^{-\frac{\left((\mu_1 - \beta_0) + (\sigma_1 - \beta_1)\Phi^{-1}(p) - (\mu_1 - \beta_0)\right)^2}{2(\sigma_1 - \beta_1)^2}}}{e^{(\mu_1 - \beta_0) + (\sigma_1 - \beta_1)\Phi^{-1}(p)}} \\ &= \frac{1}{\sqrt{2\pi}(\sigma_1 - \beta_1)} \frac{e^{-\frac{(\log Q_2(p) - (\mu_1 - \beta_0))^2}{2(\sigma_1 - \beta_1)^2}}}{Q_2(p)}, \end{aligned}$$

which is the density function of a $\mathcal{L}n(\mu_2, \sigma_2^2)$ random variable evaluated in $Q_2(p)$ with $\mu_2 = (\mu_1 - \beta_0)$ and $\sigma_2 = (\sigma_1 - \beta_1)$. Note that the density of Y_2 correctly

depends both upon the SQUARE parameters $\boldsymbol{\beta} = (\beta_0, \beta_1)$ and the Y_1 parameters $\eta = (\mu_1, \sigma_1^2)$.

4.3.4 Prior Structure

Since in the simulation study I will consider two B-SQUARE models, one with $Y_1|\boldsymbol{\pi}, \theta \sim \mathcal{M}ix\mathcal{G}a\,(\boldsymbol{\pi}, \theta|J)$ and the other with $Y_1|\mu_1, \sigma_1^2 \sim \mathcal{L}n(\mu_1, \sigma_1^2)$, in this section I provide a prior structure for both the models.

For the model with $Y_1|\boldsymbol{\pi}, \theta \sim \mathcal{M}ix\mathcal{G}a\,(\boldsymbol{\pi}, \theta|J)$ the prior distributions for $\boldsymbol{\pi}$ and θ are the same as in Chapter 3, that is $\theta \sim \mathcal{G}a(\alpha, \beta)$ and $\boldsymbol{\pi} \sim \mathcal{D}_J\left(\frac{1}{J}, \ldots, \frac{1}{J}\right)$. For $\boldsymbol{\beta}$ a noninformative (improper) prior will be used.

For the model with $Y_1|\mu_1, \sigma_1^2 \sim \mathcal{L}n(\mu_1, \sigma_1^2)$ the prior for μ_1 is a normal $\mathcal{N}(\nu, \zeta^2)$ distribution and the prior for σ_1 is an inverse gamma $\mathcal{IG}(\phi, \rho)$ distribution. For $\boldsymbol{\beta}$ a noninformative (improper) prior is still used.

The two models are summarized in Figure 4.3 and 4.4.

Model : $Y_1|\theta, \boldsymbol{\pi} \sim \mathcal{M}ix\mathcal{G}a(\theta, \boldsymbol{\pi}|J)$

$$Q_1(p) = Q_2(p)\, e^{X(p,\lambda)\boldsymbol{\beta}}$$

Prior structure : $\theta \sim \mathcal{G}a(\alpha, \beta)$

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J) \sim \mathcal{D}_J\left(\tfrac{1}{J}, \ldots, \tfrac{1}{J}\right)$$

$$\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_\lambda) \sim \text{uniform (improper) prior}$$

θ, $\boldsymbol{\pi}$ and $\boldsymbol{\beta}$ independent

Figure 4.3: The B-SQUARE model with gamma mixture Y_1 data.

4.3.5 Posterior calculation

The posterior distribution is obtained by simulation using a Metropolis-Hastings algorithm with blocking (see Carlin et al. [12], O'Hagan [70] and Robert et al. [86]). I still keep distinct the presentation for the model with the gamma mixture and the model with the log-normal distribution.

Model : $Y_1|\mu_1, \sigma_1^2 \sim \mathcal{L}n(\mu_1, \sigma_1^2)$

$Q_1(p) = Q_2(p)\, e^{X(p,\lambda)\beta}$

Prior structure : $\mu_1 \sim \mathcal{N}(\nu, \zeta^2)$

$\sigma_1 \sim \mathcal{IG}(\phi, \rho)$

$\beta = (\beta_0, \beta_1) \sim$ uniform (improper) prior

μ_1, σ_1 and β independent

Figure 4.4: The B-SQUARE model with log-normal Y_1 data.

B-SQUARE with $Y_1|\theta, \pi \sim Mix\mathcal{G}a(\theta, \pi|J)$

The proposal distribution for θ is a gamma $\mathcal{G}a(A, B)$ distribution where A and B are chosen such that

$$\begin{cases} \mathbb{E}[\theta^*] = \frac{A}{B} = \bar{\theta} \\ \text{var}[\theta^*] = \frac{A}{B^2} = K_\theta \cdot \tilde{\theta} \end{cases},$$

that is

$$\begin{cases} A = \frac{\bar{\theta}^2}{K_\theta \cdot \tilde{\theta}} \\ B = \frac{\bar{\theta}}{K_\theta \cdot \tilde{\theta}} \end{cases},$$

where $\bar{\theta}$ and $\tilde{\theta}$ are respectively the mean and the variance of the chain estimated using only the first sample[2] $(y_{11}, \ldots, y_{1n_1})$ (see Chapter 3) and K_θ is a constant to be fixed such that the θ subchain has a reasonable acceptance rate.

The proposal distribution for π is a Dirichlet $\mathcal{D}_J(K_\pi \widehat{\pi}_1, \ldots, K_\pi \widehat{\pi}_J)$ distribution where $(\widehat{\pi}_1, \ldots, \widehat{\pi}_J)$ are the estimates of π obtained using only the first sample $(y_{11}, \ldots, y_{1n_1})$ (see Chapter 3) and K_π is a constant to be fixed such that the π subchain has a reasonable acceptance rate. This choice of the proposal parameters allows the expected proposal π^* to be equal[3] to $\widehat{\pi}$, while its variance to be inversely proportional to K_π, i.e. $\mathbb{E}[\pi^*] = \widehat{\pi}$ and $\text{var}[\pi^*] \propto K_\pi^{-1}$.

Finally the proposal for β is a λ-multivariate normal $\mathcal{N}_\lambda\left(\beta^{(m-1)}, K_\beta \Sigma_{OLS}\right)$

[2]The algorithm for θ is thus an independent Metropolis-Hastings.

[3]This means that the algorithm for π is an independent Metropolis-Hastings.

distribution with mean equal to the previous iteration value $\beta^{(m-1)}$[4] and variance-covariance matrix equal to Σ_{OLS}, the variance-covariance matrix of the ordinary least squares estimator β_{OLS}. K_β is a constant to be fixed such that the β subchain has a reasonable acceptance rate.

B-SQUARE with $Y_1|\mu_1, \sigma_1^2 \sim \mathcal{L}n(\mu_1, \sigma_1^2)$

The proposal distribution for μ_1 is a symmetric triangular

$$\mathcal{T}r\left(\mu_1^{(m-1)} - K_\mu \frac{s_1}{\sqrt{n_1}}, \mu_1^{(m-1)}, \mu_1^{(m-1)} + K_\mu \frac{s_1}{\sqrt{n_1}}\right)$$

centered on the previous accepted value $\mu_1^{(m-1)}$ and where $s_1/\sqrt{n_1}$ is the standard error for the mean of the log-transformed y_1 data (for a brief review of the triangular distribution see Appendix B). K_μ is a constant to be fixed such that the μ_1 subchain has a reasonable acceptance rate.

The proposal distribution for σ_1 is a log-normal $\mathcal{L}n\left(\sigma_1^{(m-1)}, K_\sigma^2\right)$ distribution with mean equal to the previous accepted value $\sigma_1^{(m-1)}$ and variance equal to K_σ^2, where K_σ is a parameter to fix such that the σ_1 subchain has a reasonable acceptance rate.

Finally the proposal for β is a bivariate normal $\mathcal{N}_2\left(\beta^{(m-1)}, K_\beta \Sigma_{OLS}\right)$ distribution with mean equal to the previous iteration value $\beta^{(m-1)}$ and variance-covariance matrix equal to Σ_{OLS}, the variance-covariance matrix of the ordinary least squares estimator β_{OLS}. K_β is a constant to be fixed such that the β subchain has a reasonable acceptance rate.

4.3.6 Choice of the hyperparameters

The advices for choice of the hyperparameters are the same given in Section 3.2.5 of the previous chapter. For what regards the choice of the proposal parameters, they are the result of a calibration aimed at reaching a reasonable acceptance rate for the Metropolis-Hastings algorithm.

[4] That is a random-walk Metropolis for β.

4.3.7 Output of the model

The first output of the model is the probability for Y_1 to exceed a given threshold k, that is

$$\mathbb{P}(y_1^* > k \,|\, y_1, y_2) = \int P(y_1^* > k \,|\, y_1, y_2, \theta, \pi)\, f(\theta, \pi \,|\, y_1)\, d\theta\, d\pi\,.$$

This quantity can be estimated with

$$\begin{aligned}\widehat{\mathbb{P}}(y_1^* > k \,|\, y_1) &= \frac{1}{M}\sum_{m=1}^{M} \mathbb{P}\left(y_1^* > k \,\middle|\, \theta^{(m)}, \pi^{(m)}\right) \\ &= 1 - \frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{J} \pi_j^{(m)} F_j\left(k \,|\, \theta^{(m)}\right), \qquad (4.16)\end{aligned}$$

where $F_j(\cdot \,|\, \theta)$ is the distribution function of a $\mathcal{G}a(j, \theta)$.

It is also possible to produce a second output which is an estimate of the mean difference $\Delta = \mu_1 - \mu_2$. Define $p_{1i}^{(m)} = F_1\left(y_{1i} \,|\, \theta^{(m)}, \pi^{(m)}\right)$ and $\widetilde{p}_{2i}^{(m)}$ such that $y_{2i} = \widetilde{Q_2}\left(\widetilde{p}_{2i}^{(m)}\right)$, then calculate

$$\begin{aligned}\widehat{\Delta}^{(m)} &= \frac{n_1}{n_1+n_2}\left\{\frac{1}{n_1}\sum_{i=1}^{n_1} y_{1(i)}\left[1 - e^{-X\left(p_{1i}^{(m)}\right)\beta^{(m)}}\right]\right\} \\ &+ \frac{n_2}{n_1+n_2}\left\{\frac{1}{n_2}\sum_{i=1}^{n_2} y_{2(i)}\left[e^{X\left(\widetilde{p}_{2i}^{(m)}\right)\beta^{(m)}} - 1\right]\right\}. \qquad (4.17)\end{aligned}$$

This quantity can then be compared with the result produced with SQUARE. The advantage of B-SQUARE is that it provides an estimate of the full posterior distribution of Delta, that is useful for assessing the uncertainty inherent in the estimation.

4.4 Data

The dataset used in the following analysis is the National Medical Expenditure Survey (NMES)[5]. It provides data on annual medical expenditures and disease status for a representative sample of the U.S. civilian, non-institutionalized population. NMES data derive from 1987. Recent updates (e.g. Medical Expenditure Panel

[5]Sources: Johns Hopkins School of Public Health and US Department of Health and Human Services, Public Health Service

Table 4.1: Composition of the NMES dataset.

	smokers	*non smokers*	
cases	165 (64%)	23 (70%)	188 (65%)
controls	4,682 (32%)	4,546 (28%)	9,228 (25%)
	4,847 (32%)	4,569 (18%)	9,416 (25%)

Survey, 1997) have insufficient sample size, although an update to the analysis may be possible with future data releases. In the dataset used in the analysis a total of 9,416 individuals were available (see table above, where numbers in parentheses represent the percentage of people in that cell with non-zero expenditures).

4.5 Data Analysis

In this section I present a data analysis of the NMES dataset using B-SQUARE. Like in the previous chapter, the aim is to assess the risk to exceed a given threshold for the medical costs of the cases (subjects with smoking attributable diseases, i.e. lung cancer and coronary obstructive pulmonary disease). Y_1 are the medical costs for the cases and Y_2 are the medical costs for the controls. B-SQUARE with $\mathcal{M}ix\mathcal{G}a(\theta, \pi|J)$ is used. The parameters of the model are then $J = 40$, $\alpha = 845$, $\beta = 13,000$, $X(p, \lambda)$ is a polynomial of degree $\lambda = 6$, data are transformed with a cubic root function (this is an innocuous step, as is discussed in details in Appendix A), 25,000 sampling iterations are used (5,000 of which for burn-in), $K_\theta = 1$, $K_\pi = 250$ and $K_\beta = 1.5$.

Figure 4.5 reports the fitted values for B-SQUARE. The gray points are data (under a cubic root transformation). On the vertical axis the log-quantile ratio is plotted. The solid line inside the dashed lines (the predictions bands) is the OLS estimate while the solid line inside the shaded band (the credible bands) is the B-SQUARE estimate. The main difference between the two models seems to be that B-SQUARE is less sensitive to extreme values.

In the next three pictures the output of the Metropolis-Hastings is reported. All the chains for the model parameters mixed well and the acceptance rate are good, being 28% for β, 25% for θ and less than 1% for π (this is the reason why

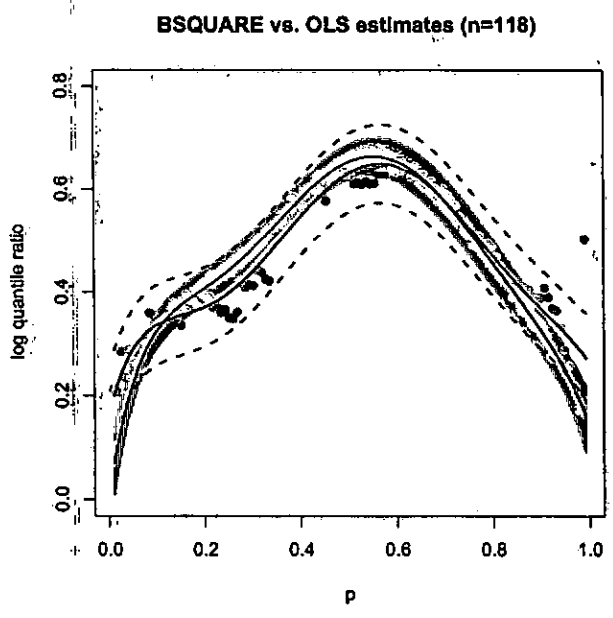

Figure 4.5: Fitted B-SQUARE model (on transformed data).

the output for the weights is not reported). The reason for the very low acceptance rate for π is that their starting values (an estimate using the model of the previous chpater, that is using just one sample for 1,500 iterations) were already close to convergence. Moreover the acceptance of $J = 40$ parameters in one shot (through the Dirichlet proposal distribution) is quite difficult. The solid line overimposed on the histogrmas are the correspondent kernel density estimators.

For what regards the output of the analysis, Figure 4.9 contains the risk to exceed in one year a given threshold k for the medical costs of a subject with smoking attributable diseases. The plot shows a reasonable behavior of this risk, that is a decrease for increasing values of k. The vertical line attached to each estimate is the credible interval. These interval are very narrow. Some caution should be used to conclude that the estimates for higher values of the threshold are more reliable.

Finally Figure 4.10 contains the posterior distribution of the mean difference Δ in (4.17), weighted for the percentage of non-zero costs among the cases and the controls. Note that the weighted simple mean difference is \$5,990.54, which is depicted in the plot using a vertical dashed line. The other two vertical solid lines

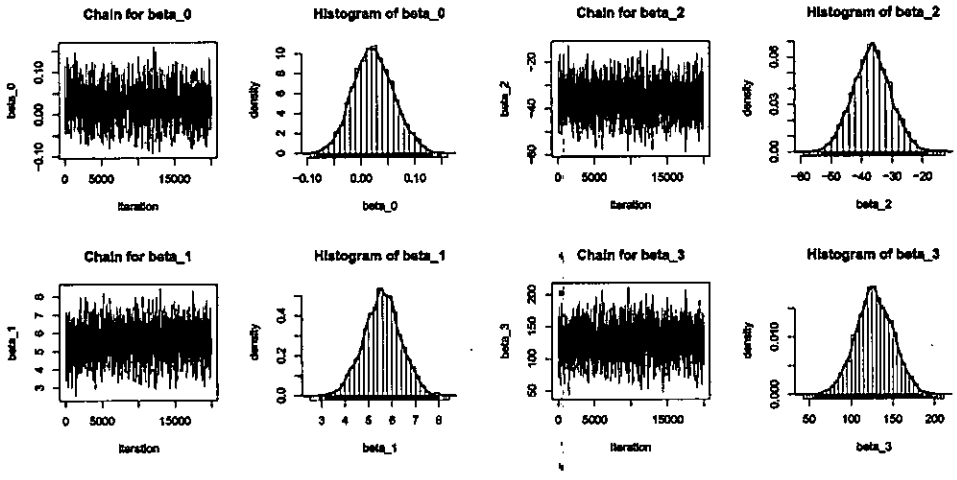

Figure 4.6: Posterior distribution for β (on transformed data).

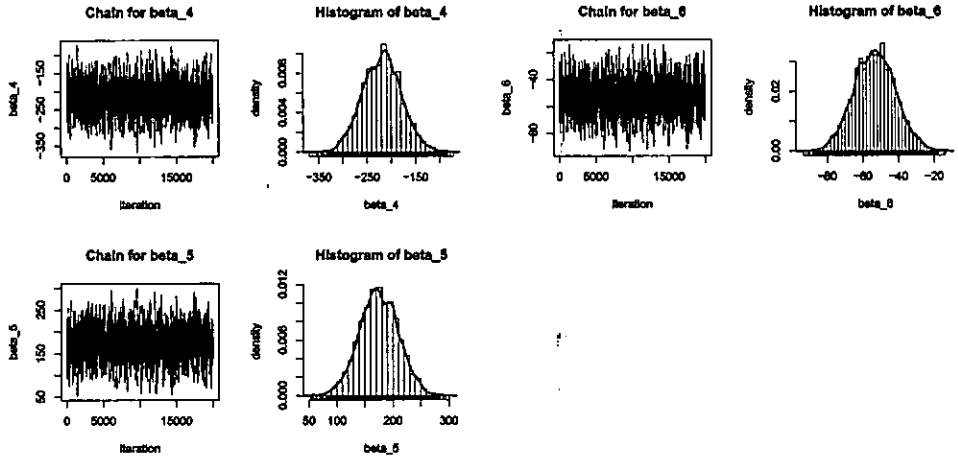

Figure 4.7: Posterior distribution for β (on transformed data).

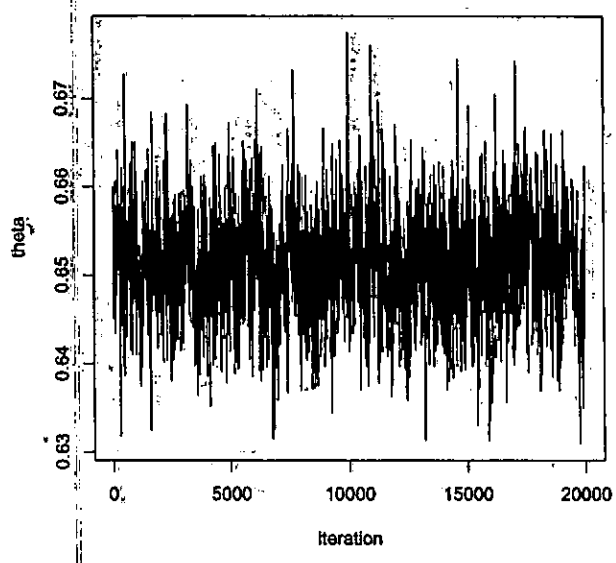

Figure 4.8: Posterior distribution for θ (on transformed data).

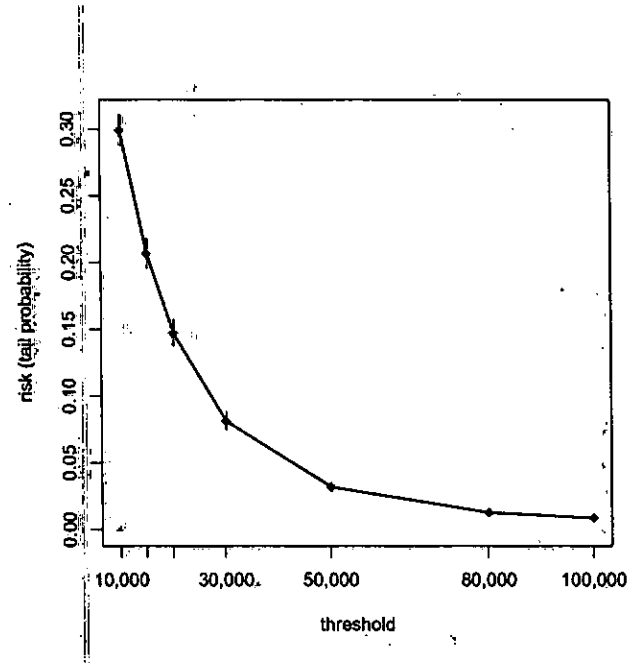

Figure 4.9: Risk to exceed a given medical cost threshold.

indicate the 95% credible interval for Δ, that contains the simple mean difference.

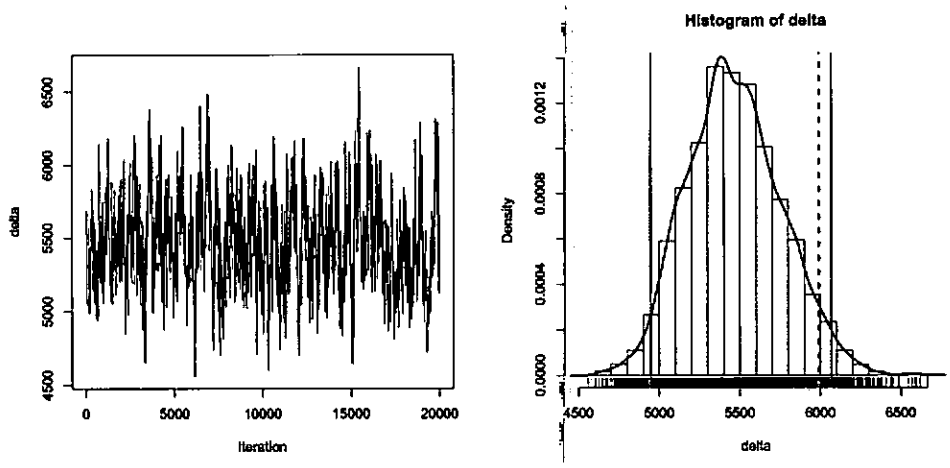

Figure 4.10: Posterior distribution for Δ.

4.6 Discussion

In this chapter I developed a Bayesian model that uses the information provided by two samples to better estimate the tail probability of a medical costs distribution and the mean difference of two populations, typically the population of the case and the population of the controls. The model is based on the SQUARE approach recently proposed by Dominici et al. [22], in which they assume a relation that links the two quantile functions.

The main finding of the chapter is that the likelihood function for the SQUARE approach has been given thanks to the nice properties of the quantile and quantile density functions. The exact likelihood cannot be calculated so an approximation has been proposed. The data analysis and numerous other analyses not reported in the chapter show that this approximation can be considered satisfactory at least from an applicative point of view.

Future work will regard an extensive simulation study that include other methods for tail estimation. Secondly future research will also study how to measure the gain in efficiency from borrowing strength across samples, as well as an assessment

of the sensitivity of the results to the prior choice.

Chapter 5

Discussion and Extensions

In the previous chapters two new Bayesian methods for estimating the density of skewed data have been proposed. These may have many potential applications in practice (public health, finance, etc.). They represent an attempt to develop semi-parametric Bayesian models that use quantile instead of cumulative distribution functions.

The first semiparametric Bayesian model for skewed data is based on a mixture of gamma distributions which share a common scale parameter θ. This is the only parameter in the model apart additional to the mixture weights. Thus an advantage of the gamma mixture model is the parsimony in the parametrization. In Chapter 3 I have also proposed an efficient computation algorithm obtained by integrating out the parameter θ. The model predictive performance have been compared to that of other competing approaches for tail estimation. The simulation study shows that the model outperforms the other methods in terms of efficiency of the estimation. Future work will be done about how to choose among different gamma mixture models. One possibility would be to use the *deviance information criterion* (DIC), but some preliminary results not reported in the thesis show that further research is needed on this side.

The second model, called B-SQUARE, builds on the previous gamma mixture and extends it in the direction of exploiting the information content of two samples (instead of only one) in estimating the skewed density. The idea of borrowing strength across sample is certainly not new in the literature. The building block of the model is in fact SQUARE, where this idea of gathering information from two samples for estimating a certain quantity is strongly implemented. The main

innovation introduced is the explicit form for the density of the second population. This allows to write the likelihood of the model providing in this way a likelihood-based version of SQUARE. The main drawback of B-SQUARE remains instead the computational burden required for the estimation. Further work should be done also with respect to this point.

5.1 Directions for Future Research

Several problems arise as themes of future research.

A first theme is the study of the statistical properties of the models introduced in the last two chapters, that is the gamma mixture model and the B-SQUARE model. In particular the estimation algorithms properties need to studied more deeply both to reduce their computational burden and to assess the efficiency that they allow to reach. Moreover a thorough study of goodness-of-fit tests for the models introduced would be very useful.

Secondly the B-SQUARE model can be extended in many directions, mainly to allow multiple group comparisons. Another very interesting theme of research is to extend B-SQUARE to a regression setting that involve the use of covariates. This has been already done for the SQUARE approach by Dominici et al. [23]. Implementing a similar approach for B-SQUARE is quite challenging, especially for the efficient computational algorithm that this would require.

A further research theme is the extension of B-SQUARE to a Bayesian nonparametric setting. Some indications on how to proceed already exist in the literature (see Ishwaran et al. [41], Gelfand et al. [31], Kottas et al. [51],[52], Hjort et al. [37] and Petrone et al. [75]) but a lot of work remains to be done.

In general I think that a systematic development of Bayesian data analysis techniques for quantile functions is really important and would be very useful both from an applied and a methodological point of view.

Appendix A

Some Useful Facts about Quantile Functions

A.1 Basic Definitions and Properties

In this section I report some definitions and properties that are used in the thesis (for proofs of these statements see Shorack [95]).

Definition A.1 (Quantile function) For any distribution function $F(\cdot)$ the *quantile function* of F is defined[1] as

$$Q(p) \equiv F^{-1}(p) \equiv \inf\{x \in \mathbb{R} : F(x) \geq p\}, \quad 0 < p < 1,$$

with the convention that $\inf \emptyset = \infty$.

Some of the properties of the quantile function are summarized in the next proposition (for proofs see Resnick [79] and Shorack [95]).

Proposition A.1 (Properties of a quantile function) If F is a distribution function[2], then for $p \in (0,1)$ and $x \in \mathbb{R}$ the following properties hold.

(a) $F(x) \geq p \quad \Longleftrightarrow \quad F^{-1}(p) \leq x.$

[1] Strictly speaking the quantile function of F is defined as the *generalized inverse* of the distribution function F. To avoid confusion with the standard notation for the ordinary inverse of a given function, some authors prefer to denote the quantile function with the notation $F^{-}(p)$.

[2] That is a right-continuous function..

(b) $F(x) < p \iff F^{-1}(p) > x$.

(c) $F(x_1) < p \leq F(x_2) \iff x_1 < F^{-1}(p) \leq x_2$.

(d) $F(F^{-1}(p)) \geq p$, with equality for F continuous.

(e) $F^{-1}(F(x)) \leq x$, with equality for F increasing.

(f) F is continuous $\iff$ F^{-1} is increasing.

(g) F is increasing $\iff$ F^{-1} is continuous.

(h) If $X \sim F$, then $\mathbb{P}[F^{-1}(F(X)) \neq X] = 0$.

Proposition A.2 (Moments or order k) If $X \geq 0$ and $X \sim F$, with quantile function $Q(p)$, then

$$\mathbb{E}[X^r] = \int_0^1 [Q(p)]^r dp. \tag{A.1}$$

More generally, if h is a non-decreasing function, then

$$\mathbb{E}[h(X)] = \int_0^1 h(Q(p))dp. \tag{A.2}$$

From the previous relations it follows that

$$\text{var}[X] = \int_0^1 (Q(p) - \mathbb{E}[X])^2 dp. \tag{A.3}$$

Proposition A.3 (Wasserstein distance) For $k = 1, 2$ define

$$\mathcal{F}_k \equiv \left\{ F : F \text{ is a distribution function, and } \int |x|^k dF(x) < \infty \right\}$$

$$d_k(F_1, F_2) = \int_0^1 |Q_1(p) - Q_2(p)|^k dp \qquad \text{for all } F_1, F_2 \in \mathcal{F}_k.$$

Then d_k are distance functions and $(\mathcal{F}_k, d_k)$ are complete metric spaces.

Proposition A.4 (Density quantile function, Parzen [71]) Let X be a random variable with $p = F(x)$ and $x = Q(p)$ for any pair (x, p), with $0 < p < 1$. Suppose also that $f(x) \equiv F'(x)$ is its density function and $q(p) \equiv Q'(p)$ the quantile density function. Then

$$f(x)q(p) \equiv f(Q(p))q(p) = 1 \quad \text{or } f(Q(p)) = \frac{1}{q(p)}, \quad 0 < p < 1. \tag{A.4}$$

Proof. Consider the identity $p = F(Q(p))$ and differentiate it with respect to p, then by the chain rule

$$\frac{dF(Q(p))}{dp} \cdot \frac{dQ(p)}{dp} = 1\,, \quad 0 < p < 1\,,$$

from which the statement follows immediately. □

Remark A.1 This result allows to plot the density function $(x, f(x))$ entirely from $(Q(p), 1/q(p))$, that is from the quantile function and its derivative, without being able to invert $Q(p)$ to get $F(x)$.

Example A.1 (Exponential Distribution) If $X \sim \mathcal{E}xp(\lambda)$, then the distribution function is

$$p = F(x) = 1 - e^{-\lambda x}\,, \qquad \lambda > 0, x > 0\,,$$

the density function is

$$f(x) = \lambda e^{-\lambda x}\,, \qquad x > 0\,,$$

the quantile function is

$$x = Q(p) = -\ln(1-p)/\lambda\,, \quad 0 < p < 1$$

and the quantile density function is

$$q(p) = \frac{dQ(p)}{dp} = \frac{1}{\lambda(1-p)}\,.$$

The previous proposition allows then to get the density quantile function

$$f(Q(p)) = \frac{1}{q(p)} = \lambda(1-p)\,.$$

Example A.2 (Log-normal Distribution) If $X \sim \mathcal{L}n(\mu, \sigma^2)$, then the distribution function is

$$p = F(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right), \qquad \sigma > 0\,,$$

where $\Phi(\cdot)$ is the distribution function of a standard normal random variable, the density function is

$$f(x) = \frac{1}{\sigma x}\phi\left(\frac{\ln x - \mu}{\sigma}\right),$$

where $\phi(\cdot)$ is the density function of a standard normal random variable, the quantile function is

$$x = Q(p) = e^{\mu+\sigma\Phi^{-1}(p)}, \quad 0 < p < 1,$$

where $\Phi^{-1}(\cdot)$ is the quantile function of a standard normal random variable, and the quantile density function is

$$q(p) = \frac{dQ(p)}{dp} = \frac{\sigma e^{\mu+\sigma\Phi^{-1}(p)}}{\phi(\Phi^{-1}(p))}.$$

The previous proposition allows then to get the density quantile function

$$f(Q(p)) = \frac{1}{q(p)} = \frac{\phi(\Phi^{-1}(p))}{\sigma e^{\mu+\sigma\Phi^{-1}(p)}}$$

(see Figure A.1 for an example with $\mu = 2$ and $\sigma = 1.5$).

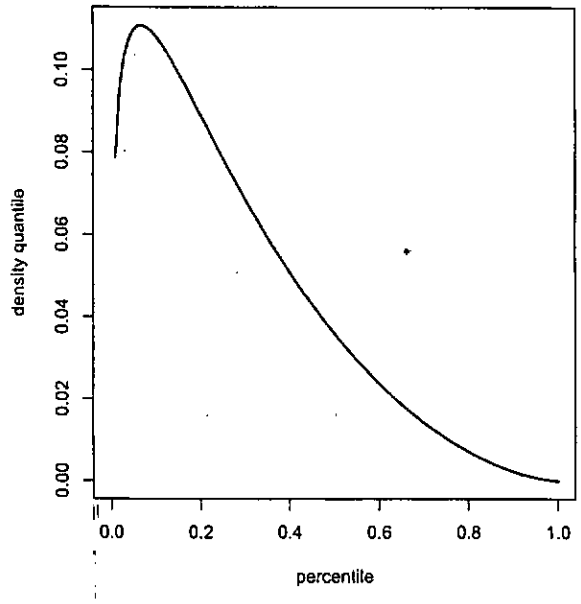

Figure A.1: Example of density quantile function for a log-normal random variable.

Example A.3 (Pareto Distribution) If $X \sim \mathcal{P}a(a, b)$, then the distribution function is

$$p = F(x) = 1 - b^a x^{-a}, \quad x > 0, a, b > 0,$$

the density function is

$$f(x) = ab^a x^{-a-1},$$

the quantile function is

$$x = Q(p) = \frac{b}{(1-p)^{\frac{1}{a}}}, \quad 0 < p < 1,$$

and the quantile density function is

$$q(p) = \frac{dQ(p)}{dp} = \frac{b}{a(1-p)^{\frac{a+1}{a}}}.$$

The previous proposition allows then to get the density quantile function

$$f(Q(p)) = \frac{1}{q(p)} = \frac{a}{b}(1-p)^{\frac{a+1}{a}}$$

(see Figure A.2 for an example with $a = 2$ and $b = 1$).

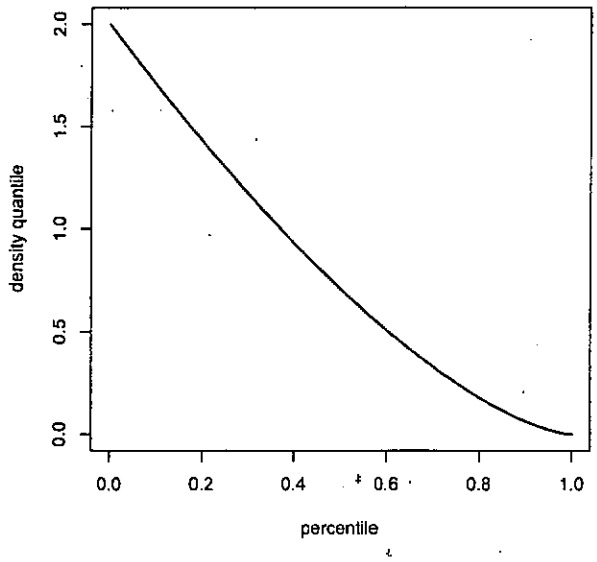

Figure A.2: Example of density quantile function for a Pareto random variable.

A.2 Quantile Function of a Transformed Random Variable

A very interesting property of quantile functions is that it is possible to apply to them a large set of transformations and still get a quantile function. These features

are particularly useful when one wants to model directly the quantile function of a certain random variable, like in SQUARE and B-SQUARE. In this section I report briefly the properties used in the chapters. The source for this section is mainly Gilchrist [33].

Addition rule

If $Q_1(p)$ and $Q_2(p)$ are two quantile functions, then $Q_1(p)+Q_2(p)$ is also a quantile function. This rule follows simply from the fact that a sum of two non-decreasing functions is still a non-decreasing one.

Multiplication rule

The product of two positive quantile functions $Q_1(p)$ and $Q_2(p)$, that is $Q(p) = Q_1(p) \times Q_2(p)$, is also a quantile function. This rule is important for SQUARE and B-SQUARE since they are defined as

$$Q_1(p) = Q_2(p)\, e^{X(p,\lambda)\boldsymbol{\beta}},$$

so to ensure that the product is a quantile function $e^{X(p,\lambda)\boldsymbol{\beta}}$ has to satisfy the condition $q_1(p) = dQ_1(p)/dp \geq 0$, that is

$$q_2(p)\, e^{X(p,\lambda)\boldsymbol{\beta}} + X'(p,\lambda)\boldsymbol{\beta}\, Q_2(p)\, e^{X(p,\lambda)\boldsymbol{\beta}} \geq 0,$$

or

$$X'(p,\lambda)\,\boldsymbol{\beta} \geq -\frac{1}{f_2\left(Q_2(p)\right) Q_2(p)}, \quad 0 < p < 1. \tag{A.5}$$

Then $e^{X(p,\lambda)\boldsymbol{\beta}}$ can be decreasing for some values of p but it has to satisfy this requirement.

Note that (A.5) is equivalent to condition (4.14) required to find the f_2 density in B-SQUARE.

Intermediate rule

If $Q_1(p)$ and $Q_2(p)$ are two quantile functions, then $Q(p) = \pi Q_1(p) + (1-\pi)Q_2(p)$, $0 \leq \pi \leq 1$, is also a quantile function. This means that a mixture of quantile functions is a quantile function. Note that it is not true that the quantile function

of a mixture is the mixture of the quantile functions of the components, that is

$$F(x) = \sum_{j=1}^{J} \pi_j F_j(x) \quad \nRightarrow \quad Q(p) \equiv F^{-1}(p) = \sum_{j=1}^{J} \pi_j Q_j(p)\,,$$

where F_j and Q_j are respectively the distribution and the quantile functions of the mixture components. This result reflects a situation that can happen in practice, i.e. the fact that a random variable may have an explicit distribution function but the quantile function is not writable or vice versa.

h-transformation rule

If $Q_X(p)$ is the quantile function of a random variable X and h is a non-decreasing function, then for $Y = h(X)$

$$Q_Y(p) = h(Q_X(p))\,. \tag{A.6}$$

This rule is very useful in B-SQUARE since it allows to work on a transformation of the original variables (typically a power transformation) and then re-transform back the result through

$$Q_Y(p) = h(Q_X(p)) \quad \Rightarrow \quad Q_X(p) = h^{-1}\left[Q_Y(p)\right]. \tag{A.7}$$

For example, suppose that B-SQUARE is estimated on the transformed variables $\widetilde{Y}_1 = h(Y_1) = Y_1^k$ and $\widetilde{Y}_2 = h(Y_2) = Y_2^k$, with $k > 0$, that is

$$Q_{\widetilde{Y}_1}(p) = Q_{\widetilde{Y}_2}(p)\, e^{X(p,\lambda)\tilde{\beta}}\,.$$

Then once $\widetilde{\beta}$ has been estimated it is possible to transform it back to the original scale by noting that (A.7) implies

$$Q_{Y_1}(p) = \left[Q_{\widetilde{Y}_1}(p)\right]^{\frac{1}{k}}\,,$$

and so the model can be rewritten as

$$Q_{Y_1}(p) = Q_{Y_2}(p)\, e^{X(p,\lambda)\beta}\,,$$

where $\beta = \widetilde{\beta}/k$.

Addition rule for quantile density functions

The sum of two quantile density functions $q_1(p)$ and $q_2(p)$ is itself a quantile density function, that is

$$q(p) = q_1(p) + q_2(p)\,.$$

Note also that using (A.4) this rule implies

$$f(Q(p)) = \frac{1}{\frac{1}{f(Q_1(p))} + \frac{1}{f(Q_2(p))}}\,,$$

where $Q(p) = Q_1(p) + Q_2(p)$.

A.3 Maximum Likelihood Estimation and Quantile Functions

Consider a sample $(y_1, \ldots, y_n)$ of iid observations from a distribution $F(y|\theta)$ with density $f(y|\theta)$, where θ is an unknown parameter. Then the likelihood of θ is defined by

$$\mathbb{L}(\theta|y_1, \ldots, y_n) = \prod_{i=1}^{n} f(y_i|\theta)\,.$$

It is possible to write the likelihood using the density quantile function $f(Q(p)|\theta)$ as

$$\mathbb{L}(\theta|y_1, \ldots, y_n) = \prod_{i=1}^{n} f(Q(p_i)|\theta)\,,$$

where $p_i = F(y_{(i)}|\theta)$, $i = 1, \ldots, n$, that is $(p_1, \ldots, p_n)$ are the actual p generated by the observed y for the given θ through model F.

In principle it is possible to restate all the maximum likelihood estimation theory in terms of the quantile function (see Appendix 2 in Gilchrist [33]).

Appendix B

The Triangular Distribution

A continuous random variable X that takes values in the interval $[a, c]$ is distributed according to a triangular distribution $\mathcal{T}r(a, b, c)$ if its density is

$$f(x|a,b,c) = \begin{cases} \frac{2(x-a)}{(c-a)(b-a)} & \text{for } a \leq x \leq b \\ \frac{2(c-x)}{(c-a)(c-b)} & \text{for } b < x \leq c, \end{cases}$$

where $b \in [a, c]$ is the mode. The distribution is symmetric when $(b-a) = (c-b)$. For some examples see Figure B.1.

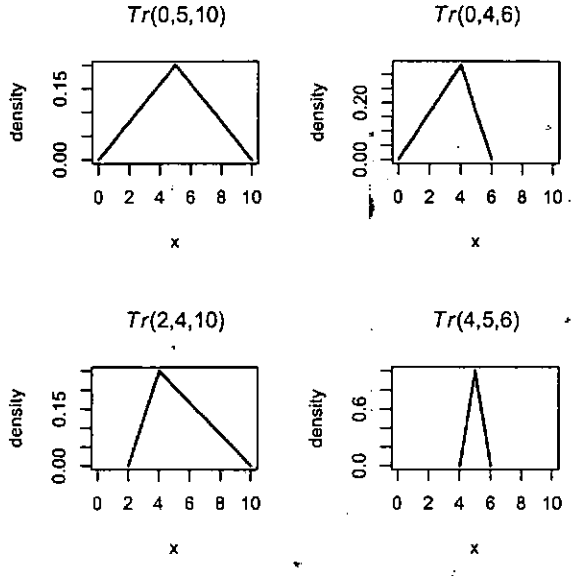

Figure B.1: Some examples of the triangular distribution $\mathcal{T}r(a, b, c)$.

The mean and variance are

$$\begin{aligned} \mathbb{E}[X] &= \frac{1}{3}(a+b+c) \\ \text{var}[X] &= \frac{1}{18}(a^2+b^2+c^2-ab-ac-bc)^2 . \end{aligned}$$

Since the cumulative distribution function and the quantile function are known explicitly, to draw random numbers from this distribution one can apply the inverse transformation method by generating deviates from a uniform $\mathcal{U}[0,1]$ distribution and then mapping them back through the quantile function (see the R code in the Appendix C).

Appendix C

R code

In this appendix I report all the R functions for the estimation of the models presented in the thesis.

C.1 Functions for the $\mathcal{M}ix\mathcal{G}a\left(\pi, \theta | J\right)$ Model

The next two functions allow the estimation of the parameters for the $\mathcal{M}ix\mathcal{G}a\left(\pi, \theta | J\right)$ mixture model. The first one implements algorithm (3.7), while the second implements algorithm (3.6).

```
mixgamma <- function(yy,JJ,GG,MM,a,b,alpha,std=F,cubic=F){
  NN <- length(yy)
  yy.temp <- yy
  if (cubic) yy <- yy^(1/3)
  mu <- mean(yy)
  if (std) yy <- yy/mu
  yy.grid <- seq(min(yy)*.66,max(yy)*1.5,length=GG)

  etich <- matrix(NA,nrow=NN,ncol=MM)
  pesi <- matrix(NA,nrow=MM,ncol=JJ)

  xx <- rep(1,NN)
  ww <- rep(1/JJ,JJ)
  theta <- rep(NA,MM)
  logff <- matrix(NA,JJ,GG)
  ff <- matrix(NA,MM,GG)

  for (m in 1:MM){
```

```
    for (nn in 1:NN) {
      temp <- log(seq(a+sum(xx)-xx[nn],a+sum(xx)-xx[nn]+JJ-1))
      pi <- (1:JJ-1)*log(yy[nn]) - lgamma(1:JJ) + cumsum(temp) - (1:JJ)*log(b+sum(yy))
      pi <- ( ww*exp(pi) ) / sum( ww*exp(pi) )
      xx[nn] <- sample(1:JJ,1,prob=pi)
      etich[nn,m]=xx[nn]
    }
    xx.counts <- table( factor(xx, levels=1:JJ ))
    ww <- rdirichlet(1, xx.counts + alpha )
    theta[m] <- rgamma( 1, sum(xx)+a, rate=sum(yy)+b )
    for (j in 1:JJ){
      logff[j,] <- log(ww[j]) + (j-1)*log(yy.grid) - theta[m]*yy.grid - lgamma(j) + j*log(theta[m])
    }
    ff[m,] <- apply( exp( logff ), 2, sum )
    pesi[m,] <- ww
  }
  return(list(ff=ff,yy.grid=yy.grid,theta=theta,label=etich,pesi=pesi))
}

mixgamma.theta <- function(yy,JJ,GG,MM,a,b,alpha,std=F,cubic=F){
  NN <- length(yy)
  yy.temp <- yy
  if (cubic) yy <- yy^(1/3)
  mu <- mean(yy)
  if (std) yy <- yy/mu
  yy.grid <- seq(min(yy)*.66,max(yy)*1.5,length=GG)

  etich <- matrix(NA,nrow=NN,ncol=MM)
  pesi <- matrix(NA,nrow=MM,ncol=JJ)

  xx <- rep(1,NN)
  ww <- rep(1/JJ,JJ)
  theta <- rep(JJ/max(yy),MM+1)
  logff <- matrix(NA,JJ,GG)
  ff <- matrix(NA,MM,GG)

  for (m in 1:MM){
    for (nn in 1:NN) {
      pi <- (1:JJ-1)*log(yy[nn]) - theta[m]*yy[nn] - lgamma(1:JJ) + (1:JJ)*log(theta[m])
      pi <- ( ww*exp(pi) ) / sum( ww*exp(pi) )
      xx[nn] <- sample(1:JJ,1,prob=pi)
      etich[nn,m]=xx[nn]
    }

    xx.counts <- table( factor(xx, levels=1:JJ ))
    ww <- rdirichlet(1, xx.counts + alpha )
    theta[m+1] <- rgamma( 1, sum(xx)+a, rate=sum(yy)+b )
    for (j in 1:JJ){
      logff[j,] <- log(ww[j]) + (j-1)*log(yy.grid) - theta[m+1]*yy.grid -
         lgamma(j) + j*log(theta[m+1])
    }
    ff[m,] <- apply( exp( logff ), 2, sum )
```

```
    pesi[m,] <- ww
  }
  return(list(ff=ff,yy.grid=yy.grid,theta=theta,label=etich,pesi=pesi))
}
```

The next are two utility functions for plotting the estimated mixture distribution on the data histogram together with its confidence bands.

```
plot.mixgamma <- function(vv,yy,ndens=5,xrange=c(min(yy),max(yy)),xxlab="x",yylab="density",
 nbin=10,histogram=F,std=F,cubic=F,bands=F){

  yy.temp <- yy
  if (cubic) yy.temp <- yy^(1/3)
  mu <- mean(yy.temp)
  if (std) yy.temp <- yy.temp/mu
  if (histogram) {
    hist(yy.temp,freq=F,breaks=nbin,col="lightgray",border="gray",xlim=xrange,
       xlab=xxlab,ylab=yylab,main="")
    if (bands) {
      yygrid <- c(vv$yy.grid,rev(vv$yy.grid))
      bb <- c(allcurves.q(vv$ff,0.05),rev(allcurves.q(vv$ff,0.95)))
      polygon(yygrid,bb,col="cornflowerblue",lty=1,lwd=2,border=NA) }
    lines(c(0,vv$yy.grid),c(0,apply(vv$ff,2,mean)),type="n") }
  else {
    plot(c(0,vv$yy.grid),c(0,apply(vv$ff,2,mean)),type="n",xlim=xrange,xlab=xxlab,ylab=yylab)
    if (bands) {
      yygrid <- c(vv$yy.grid,rev(vv$yy.grid))
      bb <- c(allcurves.q(vv$ff,0.025),rev(allcurves.q(vv$ff,0.975)))
      polygon(yygrid,bb,col="cornflowerblue",lty=1,lwd=2,border=NA) } }
  rug(yy.temp)
  for(i in sample(1:dim(vv$ff)[1],ndens)) lines(vv$yy.grid,vv$ff[i,],col="red")
  lines(vv$yy.grid,apply(vv$ff,2,mean),lwd=2)
}

allcurves.q <- function(data,perc){
  n <- dim(data)[2]
  temp <- rep(NA,n)
  for (i in 1:n) temp[i] <- quantile(data[,i],perc)
  return(temp)
}
```

C.2 Functions for the B-SQUARE Model

The following functions compute the likelihood and the parameter estimates for the B-SQUARE model respectively with

(i) a mixture of gamma distributions as F_1 and a polynomial as the link between Q_1 and Q_2,

(ii) a mixture of gamma distributions as F_1 and a natural cubic spline as the link between Q_1 and Q_2,

(iii) a log-normal distribution as F_1 and F_2.

```
loglik.poly <- function(prmt,yy1,yy2,degf,p2=FALSE){
  nn2 <- length(yy2)
  p2.approx <- vector(length=nn2)
  numcomp <- length(prmt)-degf-2
  yy1.o <- sort(yy1)
  yy2.o <- sort(yy2)

  ### calculating the approximated p2 points ###
  p1.tmp <- pmixgamma(yy1.o,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  matrix.p1.tmp <- poly.dmatrix(p1.tmp,degf,intercept=TRUE)
  q2.tmp <- yy1.o*exp(-matrix.p1.tmp%*%prmt[1:(degf+1)])
  pq2.tmp <- sortedXyData(p1.tmp,q2.tmp)
  for (i in 1:nn2) p2.approx[i] <- NLSstClosestX(pq2.tmp,yy2.o[i])

  ### calculating the derivative of the polynomial of order 'degf' ###
  derivat <- poly.dmatrix(p2.approx,degf,first.deriv=TRUE)
  dmatrix.p2.approx <- poly.dmatrix(p2.approx,degf,intercept=TRUE)

  ### calculating f2 in the approximated p2 points ###
  dens1 <- dmixgamma(yy1.o,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  yy1.o.tmp <- yy2.o*exp(dmatrix.p2.approx%*%prmt[1:(degf+1)])
  dens1.tmp <- dmixgamma(yy1.o.tmp,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  A <- exp(-dmatrix.p2.approx%*%prmt[1:(degf+1)])
  B <- derivat%*%prmt[2:(degf+1)]
  dens2 <- dens1.tmp/(A-dens1.tmp*yy2.o*B)

  loglik1 <- sum(log(dens1[dens1>0]),na.rm=TRUE)
  if (any(dens2<=0)) {loglik2 <- NA} else loglik2 <- sum(log(dens2[dens2>0]),na.rm=TRUE)
  if (p2) return(list(sum(loglik1,loglik2),p2.approx)) else return(sum(loglik1,loglik2))
}

loglik.ns <- function(prmt,yy1,yy2,degf,p2=FALSE){
  nn2 <- length(yy2)
```

```
  p2.approx <- vector(length=nn2)
  numcomp <- length(prmt)-degf-2
  yy1.o <- sort(yy1)
  yy2.o <- sort(yy2)

  ### calculating the approximated p2 points ###
  p1.tmp <- pmixgamma(yy1.o,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  q2.tmp <- yy1.o*exp(-cbind(1,ns(p1.tmp,degf))%*%prmt[1:(degf+1)])
  pq2.tmp <- sortedXyData(p1.tmp,q2.tmp)
  for (i in 1:nn2) p2.approx[i] <- NLSstClosestX(pq2.tmp,yy2.o[i])

  ### calculating the derivative of the natural cubic spline with 'degf' uniform knots ###
  knots <- sort(c(rep(range(p2.approx),4),attr(ns(p2.approx,degf),"knots")))
  derivat <- splineDesign(knots,p2.approx,4,rep(1,length(p2.approx)))
  derivat <- derivat[,-1,drop=FALSE]
  derivat.const <- splineDesign(knots,range(p2.approx),4,c(2, 2))
  derivat.const <- derivat.const[,-1,drop=FALSE]
  qr.derivat.const <- qr(t(derivat.const))
  derivat <- as.matrix((t(qr.qty(qr.derivat.const, t(derivat))))[, -(1:2)])

  ### calculating f2 in the approximated p2 points ###
  dens1 <- dmixgamma(yy1.o,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  yy1.o.tmp <- yy2.o*exp(cbind(1,ns(p2.approx,degf))%*%prmt[1:(degf+1)])
  dens1.tmp <- dmixgamma(yy1.o.tmp,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  A <- exp(-cbind(1,ns(p2.approx,degf))%*%prmt[1:(degf+1)])
  B <- derivat%*%prmt[2:(degf+1)]
  dens2 <- dens1.tmp/(A-dens1.tmp*yy2.o*B)

  loglik1 <- sum(log(dens1[dens1>0]),na.rm=TRUE)
  if (any(dens2<=0)) {loglik2 <- NA} else loglik2 <- sum(log(dens2[dens2>0]),na.rm=TRUE)
  if (p2) return(list(sum(loglik1,loglik2),p2.approx)) else return(sum(loglik1,loglik2))
}

loglik.ln <- function(prmt,yy1,yy2,p2=FALSE,exact=FALSE){
  nn2 <- length(yy2)
  p2.approx <- vector(length=nn2)
  yy1.o <- sort(yy1)
  yy2.o <- sort(yy2)

  if (!exact) {
    lyy1 <- log(yy1.o)
    lyy2 <- log(yy2.o)
    ### calculating the approximated p2 points ###
    p1.tmp <- pnorm(lyy1,mean=prmt[3],sd=prmt[4])
    q2.tmp <- lyy1-cbind(1,qnorm(p1.tmp))%*%prmt[1:2]
    pq2.tmp <- sortedXyData(p1.tmp,q2.tmp)
    for (i in 1:nn2) p2.approx[i] <- NLSstClosestX(pq2.tmp,lyy2[i])

    ### calculating f2 in the approximated p2 points ###
    dens1 <- dlnorm(yy1.o,meanlog=prmt[3],sdlog=prmt[4])
    lyy1.o.tmp <- lyy2+cbind(1,qnorm(p2.approx))%*%prmt[1:2]
    dens1.tmp <- dnorm(lyy1.o.tmp,mean=prmt[3],sd=prmt[4])
```

```
    B <- prmt[2]/dnorm(qnorm(p2.approx))
    dens2 <- dens1.tmp/(1-dens1.tmp*B)
    dens2 <- dens2/yy2.o }
  else {
    dens1 <- dlnorm(yy1.o,meanlog=prmt[3],sdlog=prmt[4])
    dens2 <- dlnorm(yy2.o,meanlog=(prmt[3]-prmt[1]),sdlog=(prmt[4]-prmt[2]))
    p2.approx <- plnorm(yy2.o,meanlog=(prmt[3]-prmt[1]),sdlog=(prmt[4]-prmt[2]))
  }

  loglik1 <- sum(log(dens1[dens1>0]),na.rm=TRUE)
  if (any(dens2<=0)) {loglik2 <- NA} else loglik2 <- sum(log(dens2[dens2>0]),na.rm=TRUE)
  if (p2) return(list(sum(loglik1,loglik2),p2.approx)) else return(sum(loglik1,loglik2))
}

bsquare.poly <- function(mcmc,burnin,step,theta.init,yyy1,yyy2,ddff,JJ,prior,
   weight.nonneg=c(1,1),cubic=FALSE,prnt=TRUE){

  n1 <- length(yyy1)
  n2 <- length(yyy2)
  yyy1.o <- sort(yyy1)
  yyy2.o <- sort(yyy2)

  prmt <- vector(length=(ddff+JJ+2))
  prmt.ts <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)

  prmt.prop <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)
  prmt.curr <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)
  all.lik <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=6)
  ll.na <- vector(length=mcmc[2]+burnin[2])

  dic.tmp <- vector(length=(mcmc[2]+burnin[2]))
  delta <- vector(length=(mcmc[2]+burnin[2]))

  accept_beta <- 0
  accept_weight <- 0
  accept_theta <- 0

  # weight and theta estimate using only y1
  weight.0 <- rep(1/JJ,JJ)
  ww_theta.tmp <- mixgamma(yyy1,JJ,(mcmc[1]+burnin[1]),weight.0,prior[[1]],prior[[2]],
     prior[[3]])
  weights.y1only <- mean(as.data.frame(ww_theta.tmp[[1]][(burnin[1]+1):(mcmc[1]+burnin[1]),]))
  names(weights.y1only) <- NULL
  theta.mean.y1only <- mean(ww_theta.tmp[[2]][(burnin[1]+1):(mcmc[1]+burnin[1])])
  theta.var.y1only <- var(ww_theta.tmp[[2]][(burnin[1]+1):(mcmc[1]+burnin[1])])

  prmt.c <- c(theta.init,weights.y1only,theta.mean.y1only)

  for (sim in 1:(mcmc[2]+burnin[2])){
    prmt.old <- prmt.c

    # updating the beta parameters
```

```
prmt[1:(ddff+1)] <- rmvnorm(1,prmt.c[1:(ddff+1)],step[[1]]*step[[2]])
prmt[(ddff+2):(ddff+JJ+2)] <- prmt.c[(ddff+2):(ddff+JJ+2)]
while (check.beta.poly(prmt,yyy1,ddff)){
  prmt[1:(ddff+1)] <- rmvnorm(1,prmt.c[1:(ddff+1)],step[[1]]*step[[2]]) }
prmt.prop[sim,1:(ddff+1)] <- prmt[1:(ddff+1)]
prmt.curr[sim,1:(ddff+1)] <- prmt.c[1:(ddff+1)]

ll.prop_beta <- try(loglik.poly(prmt,yyy1,yyy2.o,ddff,p2=FALSE))
if (is.na(ll.prop_beta)){
  all.lik[sim,(1:2)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_beta)) {prmt.c <- prmt.old; next}
ll.curr_beta <- try(loglik.poly(prmt.c,yyy1,yyy2.o,ddff,p2=FALSE))
if (is.na(ll.curr_beta)){
  all.lik[sim,(1:2)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.curr_beta)) {prmt.c <- prmt.old; next}
all.lik[sim,(1:2)] <- c(ll.prop_beta,ll.curr_beta)

AA_beta <- ll.prop_beta - ll.curr_beta
BB_beta <- 0
CC_beta <- 0

ratio_beta <- exp(AA_beta + BB_beta + CC_beta)
u_beta <- runif(1)
if (u_beta < ratio_beta) {
  prmt.c <- prmt
  accept_beta = accept_beta + 1 }
else {
  prmt <- prmt.c }

# updating the mixture weights parameters
prmt[(ddff+2):(ddff+JJ+1)] <- rdirich(1,step[[3]]*weights.y1only)
prmt.prop[sim,(ddff+2):(ddff+JJ+1)] <- prmt[(ddff+2):(ddff+JJ+1)]
prmt.curr[sim,(ddff+2):(ddff+JJ+1)] <- prmt.c[(ddff+2):(ddff+JJ+1)]
prmt[c(1:(ddff+1),ddff+JJ+2)] <- prmt.c[c(1:(ddff+1),ddff+JJ+2)]

ll.prop_weight <- try(loglik.poly(prmt,yyy1,yyy2.o,ddff,p2=FALSE))
if (is.na(ll.prop_weight)){
  all.lik[sim,(3:4)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_weight)) {prmt.c <- prmt.old; next}
```

```
ll.curr_weight <- try(loglik.poly(prmt.c,yyy1,yyy2.o,ddff,p2=FALSE))
if (is.na(ll.curr_weight)){
  all.lik[sim,(3:4)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.curr_weight)) {prmt.c <- prmt.old; next}
all.lik[sim,(3:4)] <- c(ll.prop_weight,ll.curr_weight)

AA_weight <- ll.prop_weight - ll.curr_weight
BB_weight <- ddirich(prmt[(ddff+2):(ddff+JJ+1)],rep(prior[[3]],JJ),logdens=TRUE) -
   ddirich(prmt.c[(ddff+2):(ddff+JJ+1)],rep(prior[[3]],JJ),logdens=TRUE)
CC_weight <- ddirich(prmt.c[(ddff+2):(ddff+JJ+1)],step[[3]]*weights.y1only,logdens=TRUE) -
   ddirich(prmt[(ddff+2):(ddff+JJ+1)],step[[3]]*weights.y1only,logdens=TRUE)
ratio_weight <- exp(AA_weight + BB_weight + CC_weight)
u_weight <- runif(1)
if (u_weight < ratio_weight) {
  prmt.c <- prmt
  accept_weight = accept_weight + 1 }
else {
  prmt <- prmt.c }

# updating the mixture theta parameter
prmt[ddff+JJ+2] <- rgamma(1,shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
   rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])))
prmt.prop[sim,(ddff+JJ+2)] <- prmt[ddff+JJ+2]
prmt.curr[sim,(ddff+JJ+2)] <- prmt.c[ddff+JJ+2]
prmt[1:(ddff+JJ+1)] <- prmt.c[1:(ddff+JJ+1)]

ll.prop_th <- try(loglik.poly(prmt,yyy1,yyy2.o,ddff,p2=TRUE))
if (is.na(ll.prop_th[[1]])){
  all.lik[sim,(5:6)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_th[[1]])) {prmt.c <- prmt.old; next}
ll.curr_th <- try(loglik.poly(prmt.c,yyy1,yyy2.o,ddff,p2=TRUE))
if (is.na(ll.curr_th[[1]])){
  all.lik[sim,(5:6)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.curr_th[[1]])) {prmt.c <- prmt.old; next}
all.lik[sim,(5:6)] <- c(ll.prop_th[[1]],ll.curr_th[[1]])

AA_th <- ll.prop_th[[1]] - ll.curr_th[[1]]
BB_th <- dgamma(prmt[ddff+JJ+2],shape=prior[[1]],rate=prior[[2]]) -
   dgamma(prmt.c[ddff+JJ+2],shape=prior[[1]],rate=prior[[2]])
```

```
    CC_th <- dgamma(prmt.c[ddff+JJ+2],shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
       rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])),log=TRUE) -
       dgamma(prmt[ddff+JJ+2],shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
       rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])),log=TRUE)

    ratio_th <- exp(AA_th + BB_th + CC_th)
    u_th <- runif(1)
    if (u_th < ratio_th) {
      prmt.c <- prmt
      dic.tmp[sim] <- -2*ll.prop_th[[1]]
      p2.approx <- ll.prop_th[[2]]
      accept_theta = accept_theta + 1 }
    else {
      prmt <- prmt.c
      dic.tmp[sim] <- -2*ll.curr_th[[1]]
      p2.approx <- ll.curr_th[[2]] }
    # storing information
    prmt.ts[sim,] <- prmt.c

    # computing delta
    p1 <- pmixgamma(yyy1.o,prmt.c[(ddff+2):(ddff+JJ+1)],prmt.c[ddff+JJ+2])
    dmatrix.p2.approx <- poly.dmatrix(p2.approx,ddff,intercept=TRUE)
    dmatrix.p1 <- poly.dmatrix(p1,ddff,intercept=TRUE)

    if (cubic) {
      delta[sim] <- ((sum(yyy1.o^3)+sum(yyy2.o^3*exp(dmatrix.p2.approx%*%
         (3*prmt.c[1:(ddff+1)])))*weight.nonneg[1]-(sum(yyy2.o^3)+sum(yyy1.o^3*exp(-
         dmatrix.p1%*%(3*prmt.c[1:(ddff+1)])))*weight.nonneg[2])/(n1+n2) }
    else {
      delta[sim] <- ((sum(yyy1.o)+sum(yyy2.o*exp(dmatrix.p2.approx%*%
         prmt.c[1:(ddff+1)]))*weight.nonneg[1]-(sum(yyy2.o)+sum(yyy1.o*exp(-dmatrix.p1%*%
         prmt.c[1:(ddff+1)])))*weight.nonneg[2])/(n1+n2) }
  }

  # computing deviance information criterion (DIC)
  dic.tmp.mean <- try(-2*loglik.poly(c(mean(as.data.frame(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),1:(ddff+1)])), mean(as.data.frame(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),(ddff+2):(ddff+JJ+1)])),mean(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),(ddff+JJ+2)])),yyy1,yyy2.o,ddff,p2=FALSE))
  if (is.numeric(dic.tmp.mean)) {
    pd <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) - dic.tmp.mean
    dic <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) + pd }
  else {
    pd <- NA
    dic <- NA
  }

  return(list(accept=c(accept_beta/sim,accept_weight/sim,accept_theta/sim),
     chain=prmt.ts,dic=dic,pd=pd,delta=delta,prop=prmt.prop,curr=prmt.curr,
     lik=all.lik,lik.na=ll.na,y1.only=ww_theta.tmp))
}
```

```
bsquare.ns <- function(mcmc,burnin,step,theta.init,yyy1,yyy2,ddff,JJ,prior,
   weight.nonneg=c(1,1),cubic=FALSE,prnt=TRUE){

  n1 <- length(yyy1)
  n2 <- length(yyy2)
  yyy1.o <- sort(yyy1)
  yyy2.o <- sort(yyy2)

  prmt <- vector(length=(ddff+JJ+2))
  prmt.ts <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)

  prmt.prop <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)
  prmt.curr <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=ddff+JJ+2)
  all.lik <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=6)
  ll.na <- vector(length=mcmc[2]+burnin[2])

  dic.tmp <- vector(length=(mcmc[2]+burnin[2]))
  delta <- vector(length=(mcmc[2]+burnin[2]))

  accept_beta <- 0
  accept_weight <- 0
  accept_theta <- 0

  # weight and theta estimate using only y1
  weight.0 <- rep(1/JJ,JJ)
  ww_theta.tmp <- mixgamma(yyy1,JJ,(mcmc[1]+burnin[1]),weight.0,prior[[1]],
     prior[[2]],prior[[3]])
  weights.y1only <- mean(as.data.frame(ww_theta.tmp[[1]][(burnin[1]+1):(mcmc[1]+burnin[1]),]))
  names(weights.y1only) <- NULL
  theta.mean.y1only <- mean(ww_theta.tmp[[2]][(burnin[1]+1):(mcmc[1]+burnin[1])])
  theta.var.y1only <- var(ww_theta.tmp[[2]][(burnin[1]+1):(mcmc[1]+burnin[1])])

  prmt.c <- c(theta.init,weights.y1only,theta.mean.y1only)

  for (sim in 1:(mcmc[2]+burnin[2])){
    prmt.old <- prmt.c

    # updating the beta parameters
    prmt[1:(ddff+1)] <- rmvnorm(1,theta.init,step[[1]]*step[[2]])
    prmt[(ddff+2):(ddff+JJ+2)] <- prmt.c[(ddff+2):(ddff+JJ+2)]
    while (check.beta.ns(prmt,yyy1,ddff)){
      prmt[1:(ddff+1)] <- rmvnorm(1,theta.init,step[[1]]*step[[2]]) }
    prmt.prop[sim,1:(ddff+1)] <- prmt[1:(ddff+1)]
    prmt.curr[sim,1:(ddff+1)] <- prmt.c[1:(ddff+1)]

    ll.prop_beta <- try(loglik.ns(prmt,yyy1,yyy2.o,ddff,p2=FALSE))
    if (is.na(ll.prop_beta)){
      all.lik[sim,(1:2)] <- c("ERROR","ERROR")
      prmt.c <- prmt.old
      prmt.ts[sim,] <- prmt.c
      ll.na[sim] <- TRUE
```

```
  next }
if (!is.numeric(ll.prop_beta)) {prmt.c <- prmt.old; next}
ll.curr_beta <- try(loglik.ns(prmt.c,yyy1,yyy2.o,ddff,p2=FALSE))
if (!is.numeric(ll.curr_beta)) {prmt.c <- prmt.old; next}
all.lik[sim,(1:2)] <- c(ll.prop_beta,ll.curr_beta)

AA_beta <- ll.prop_beta - ll.curr_beta
BB_beta <- 0
CC_beta <- 0

ratio_beta <- exp(AA_beta + BB_beta + CC_beta)
u_beta <- runif(1)
if (u_beta < ratio_beta) {
  prmt.c <- prmt
  accept_beta = accept_beta + 1 }
else {
  prmt <- prmt.c }

# updating the mixture weights parameters
prmt[(ddff+2):(ddff+JJ+1)] <- rdirich(1,step[[3]]*weights.y1only)
prmt.prop[sim,(ddff+2):(ddff+JJ+1)] <- prmt[(ddff+2):(ddff+JJ+1)]
prmt.curr[sim,(ddff+2):(ddff+JJ+1)] <- prmt.c[(ddff+2):(ddff+JJ+1)]
prmt[c(1:(ddff+1),ddff+JJ+2)] <- prmt.c[c(1:(ddff+1),ddff+JJ+2)]

ll.prop_weight <- try(loglik.ns(prmt,yyy1,yyy2.o,ddff,p2=FALSE))
if (is.na(ll.prop_weight)){
  all.lik[sim,(3:4)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_weight)) {prmt.c <- prmt.old; next}
ll.curr_weight <- try(loglik.ns(prmt.c,yyy1,yyy2.o,ddff,p2=FALSE))
if (!is.numeric(ll.curr_weight)) {prmt.c <- prmt.old; next}
all.lik[sim,(3:4)] <- c(ll.prop_weight,ll.curr_weight)

AA_weight <- ll.prop_weight - ll.curr_weight
BB_weight <- ddirich(prmt[(ddff+2):(ddff+JJ+1)],rep(prior[[3]],JJ),logdens=TRUE) -
   ddirich(prmt.c[(ddff+2):(ddff+JJ+1)],rep(prior[[3]],JJ),logdens=TRUE)
CC_weight <- ddirich(prmt.c[(ddff+2):(ddff+JJ+1)],step[[3]]*weights.y1only,logdens=TRUE) -
   ddirich(prmt[(ddff+2):(ddff+JJ+1)],step[[3]]*weights.y1only,logdens=TRUE)

ratio_weight <- exp(AA_weight + BB_weight + CC_weight)
u_weight <- runif(1)
if (u_weight < ratio_weight) {
  prmt.c <- prmt
  accept_weight = accept_weight + 1 }
else {
  prmt <- prmt.c }

# updating the mixture theta parameter
```

```
prmt[ddff+JJ+2] <- rgamma(1,shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
   rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])))
prmt.prop[sim,(ddff+JJ+2)] <- prmt[ddff+JJ+2]
prmt.curr[sim,(ddff+JJ+2)] <- prmt.c[ddff+JJ+2]
prmt[1:(ddff+JJ+1)] <- prmt.c[1:(ddff+JJ+1)]

ll.prop_th <- try(loglik.ns(prmt,yyy1,yyy2.o,ddff,p2=TRUE))
if (is.na(ll.prop_th[[1]])){
  all.lik[sim,(5:6)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_th[[1]])) {prmt.c <- prmt.old; next}
ll.curr_th <- try(loglik.ns(prmt.c,yyy1,yyy2.o,ddff,p2=TRUE))
if (!is.numeric(ll.curr_th[[1]])) {prmt.c <- prmt.old; next}
all.lik[sim,(5:6)] <- c(ll.prop_th[[1]],ll.curr_th[[1]])

AA_th <- ll.prop_th[[1]] - ll.curr_th[[1]]
BB_th <- dgamma(prmt[ddff+JJ+2],shape=prior[[1]],rate=prior[[2]]) -
   dgamma(prmt.c[ddff+JJ+2],shape=prior[[1]],rate=prior[[2]])
CC_th <- dgamma(prmt.c[ddff+JJ+2],shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
   rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])),log=TRUE) -
   dgamma(prmt[ddff+JJ+2],shape=(theta.mean.y1only^2/(theta.var.y1only*step[[4]])),
   rate=(theta.mean.y1only/(theta.var.y1only*step[[4]])),log=TRUE)

ratio_th <- exp(AA_th + BB_th + CC_th)
u_th <- runif(1)
if (u_th < ratio_th) {
  prmt.c <- prmt
  dic.tmp[sim] <- -2*ll.prop_th[[1]]
  p2.approx <- ll.prop_th[[2]]
  accept_theta = accept_theta + 1 }
else {
  prmt <- prmt.c
  dic.tmp[sim] <- -2*ll.curr_th[[1]]
  p2.approx <- ll.curr_th[[2]] }

# storing information
prmt.ts[sim,] <- prmt.c

# computing delta
p1 <- pmixgamma(yyy1.o,prmt.c[(ddff+2):(ddff+JJ+1)],prmt.c[ddff+JJ+2])
if (cubic) {
  delta[sim] <- ((sum(yyy1.o^3)+sum(yyy2.o^3*exp(cbind(1,ns(p2.approx,ddff))%*%
  (3*prmt.c[1:(ddff+1)])))))*weight.nonneg[1]-(sum(yyy2.o^3)+sum(yyy1.o^3*
  exp(-cbind(1,ns(p1,ddff))%*%(3*prmt.c[1:(ddff+1)])))))*weight.nonneg[2])/(n1+n2) }
else {
  delta[sim] <- ((sum(yyy1.o)+sum(yyy2.o*exp(cbind(1,ns(p2.approx,ddff))%*%
  prmt.c[1:(ddff+1)])))*weight.nonneg[1]-(sum(yyy2.o)+sum(yyy1.o*
  exp(-cbind(1,ns(p1,ddff))%*%prmt.c[1:(ddff+1)])))*weight.nonneg[2])/(n1+n2) }
```

```
  }

  # computing deviance information criterion (DIC)
  dic.tmp.mean <- try(-2*loglik.ns(c(mean(as.data.frame(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),1:(ddff+1)])),mean(as.data.frame(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),(ddff+2):(ddff+JJ+1)])),mean(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),(ddff+JJ+2)])),
     yyy1,yyy2.o,ddff,p2=FALSE))
  if (is.numeric(dic.tmp.mean)) {
    pd <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) - dic.tmp.mean
    dic <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) + pd }
  else {
    pd <- NA
    dic <- NA
  }

 return(list(accept=c(accept_beta/sim,accept_weight/sim,accept_theta/sim),
    chain=prmt.ts,dic=dic,pd=pd,delta=delta,prop=prmt.prop,curr=prmt.curr,
    lik=all.lik,lik.na=ll.na,y1.only=ww_theta.tmp))
}

bsquare.ln <- function(mcmc,burnin,step,theta.init,yyy1,yyy2,prior,weight.nonneg=c(1,1),
   cubic=FALSE,exct=FALSE,prnt=TRUE){

  n1 <- length(yyy1)
  n2 <- length(yyy2)
  yyy1.o <- sort(yyy1)
  yyy2.o <- sort(yyy2)

  prmt <- vector(length=4)
  prmt.ts <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=4)

  prmt.prop <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=4)
  prmt.curr <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=4)
  all.lik <- matrix(NA,nrow=mcmc[2]+burnin[2],ncol=6)
  ll.na <- vector(length=mcmc[2]+burnin[2])

  dic.tmp <- vector(length=(mcmc[2]+burnin[2]))
  delta <- vector(length=(mcmc[2]+burnin[2]))

  accept_beta <- 0
  accept_logmu <- 0
  accept_logsigma <- 0

  mlogy1 <- mean(log(yyy1))
  sd_mlogy1 <- sd(log(yyy1))
  se_mlogy1 <- sd_mlogy1/sqrt(length(log(yyy1)))

  prmt.c <- theta.init

  for (sim in 1:(mcmc[2]+burnin[2])){
    prmt.old <- prmt.c
```

```
# updating the beta parameters
prmt[1:2] <- rmvnorm(1,prmt.c[1:2],step[[1]]*step[[2]])
prmt[3:4] <- prmt.c[3:4]
while (check.beta.ln(prmt,yyy1)) prmt[1:2] <- rmvnorm(1,prmt.c[1:2],step[[1]]*step[[2]])
prmt.prop[sim,1:2] <- prmt[1:2]
prmt.curr[sim,1:2] <- prmt.c[1:2]

ll.prop_beta <- try(loglik.ln(prmt,yyy1,yyy2.o,p2=FALSE,exact=exct))
if (is.na(ll.prop_beta)){
  all.lik[sim,(1:2)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_beta)) {prmt.c <- prmt.old; next}
ll.curr_beta <- try(loglik.ln(prmt.c,yyy1,yyy2.o,p2=FALSE,exact=exct))
if (!is.numeric(ll.curr_beta)) {prmt.c <- prmt.old; next}
all.lik[sim,(1:2)] <- c(ll.prop_beta,ll.curr_beta)

AA_beta <- ll.prop_beta - ll.curr_beta
BB_beta <- 0
CC_beta <- 0

ratio_beta <- exp(AA_beta + BB_beta + CC_beta)
u_beta <- runif(1)
if (u_beta < ratio_beta) {
  prmt.c <- prmt
  accept_beta = accept_beta + 1 }
else {
  prmt <- prmt.c }

# updating the logmu parameter
prmt[3] <- rtriang(1,prmt.c[3]-step[[3]]*se_mlogy1,prmt.c[3],prmt.c[3]+step[[3]]*se_mlogy1)
prmt.prop[sim,3] <- prmt[3]
prmt.curr[sim,3] <- prmt.c[3]
prmt[c(1:2,4)] <- prmt.c[c(1:2,4)]

ll.prop_logmu <- try(loglik.ln(prmt,yyy1,yyy2.o,p2=FALSE,exact=exct))
if (is.na(ll.prop_logmu)){
  all.lik[sim,(3:4)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_logmu)) {prmt.c <- prmt.old; next}
ll.curr_logmu <- try(loglik.ln(prmt.c,yyy1,yyy2.o,p2=FALSE,exact=exct))
if (!is.numeric(ll.curr_logmu)) {prmt.c <- prmt.old; next}
all.lik[sim,(3:4)] <- c(ll.prop_logmu,ll.curr_logmu)

prior.prop_logmu <- dnorm(prmt[3],mean=prior[[1]],sd=prior[[2]],log=T)
```

```
prior.curr_logmu <- dnorm(prmt.c[3],mean=prior[[1]],sd=prior[[2]],log=T)

AA_logmu <- ll.prop_logmu - ll.curr_logmu
BB_logmu <- prior.prop_logmu - prior.curr_logmu
CC_logmu <- 0

ratio_logmu <- exp(AA_logmu + BB_logmu + CC_logmu)
u_logmu <- runif(1)
if (u_logmu < ratio_logmu) {
  prmt.c <- prmt
  accept_logmu = accept_logmu + 1 }
else {
  prmt <- prmt.c }

# updating the logsigma parameter
prmt[4] <- rlnorm(1,meanlog=log(prmt.c[4]),sdlog=step[[4]])
prmt.prop[sim,4] <- prmt[4]
prmt.curr[sim,4] <- prmt.c[4]
prmt[1:3] <- prmt.c[1:3]

ll.prop_logsigma <- try(loglik.ln(prmt,yyy1,yyy2.o,p2=TRUE,exact=exct))
if (is.na(ll.prop_logsigma[[1]])){
  all.lik[sim,(5:6)] <- c("ERROR","ERROR")
  prmt.c <- prmt.old
  prmt.ts[sim,] <- prmt.c
  ll.na[sim] <- TRUE
  next }
if (!is.numeric(ll.prop_logsigma[[1]])) {prmt.c <- prmt.old; next}
ll.curr_logsigma <- try(loglik.ln(prmt.c,yyy1,yyy2.o,p2=TRUE,exact=exct))
if (!is.numeric(ll.curr_logsigma[[1]])) {prmt.c <- prmt.old; next}
all.lik[sim,(5:6)] <- c(ll.prop_logsigma[[1]],ll.curr_logsigma[[1]])

prior.prop_logsigma <- log(dinvgamma(prmt[4],prior[[3]],prior[[4]]))
prior.curr_logsigma <- log(dinvgamma(prmt.c[4],prior[[3]],prior[[4]]))

prop_logsigma <- dlnorm(prmt[4],meanlog=log(prmt.c[4]),sdlog=step[[4]],log=T)
prop.curr_logsigma <- dlnorm(prmt.c[4],meanlog=log(prmt[4]),sdlog=step[[4]],log=T)

AA_logsigma <- ll.prop_logsigma[[1]] - ll.curr_logsigma[[1]]
BB_logsigma <- prior.prop_logsigma - prior.curr_logsigma
CC_logsigma <- prop.curr_logsigma - prop_logsigma

ratio_logsigma <- exp(AA_logsigma + BB_logsigma + CC_logsigma)
u_logsigma <- runif(1)
if (u_logsigma < ratio_logsigma) {
  prmt.c <- prmt
  dic.tmp[sim] <- -2*ll.prop_logsigma[[1]]
  p2.approx <- ll.prop_logsigma[[2]]
  accept_logsigma = accept_logsigma + 1 }
else {
  prmt <- prmt.c
```

```
      dic.tmp[sim] <- -2*ll.curr_logsigma[[1]]
      p2.approx <- ll.curr_logsigma[[2]] }

    # storing information
    prmt.ts[sim,] <- prmt.c

    # computing delta
    p1 <- plnorm(yyy1.o,meanlog=prmt.c[3],sdlog=prmt.c[4])
    dmatrix.p2.approx <- cbind(1,qnorm(p2.approx))
    dmatrix.p1 <- cbind(1,qnorm(p1))

    if (cubic) {
      delta[sim] <- ((sum(yyy1.o^3)+sum(yyy2.o^3*exp(dmatrix.p2.approx%*%
      (3*prmt.c[1:2])))*weight.nonneg[1]-(sum(yyy2.o^3)+sum(yyy1.o^3*exp(-dmatrix.p1%*%
      (3*prmt.c[1:2])))*weight.nonneg[2])/(n1+n2) }
    else {
      delta[sim] <- ((sum(yyy1.o)+sum(yyy2.o*exp(dmatrix.p2.approx%*%prmt.c[1:2])))*
      weight.nonneg[1]-(sum(yyy2.o)+sum(yyy1.o*exp(-dmatrix.p1%*%
      prmt.c[1:2])))*weight.nonneg[2])/(n1+n2) }
  }

  # computing deviance information criterion (DIC)
  dic.tmp.mean <- try(-2*loglik.ln(c(mean(as.data.frame(prmt.ts[(burnin[2]+1):
     (mcmc[2]+burnin[2]),1:2])),mean(prmt.ts[(burnin[2]+1):(mcmc[2]+burnin[2]),3]),
     mean(prmt.ts[(burnin[2]+1):(mcmc[2]+burnin[2]),4])),yyy1,yyy2.o,p2=FALSE,exact=exct))
  if (is.numeric(dic.tmp.mean)) {
    pd <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) - dic.tmp.mean
    dic <- mean(dic.tmp[(burnin[2]+1):(mcmc[2]+burnin[2])]) + pd }
  else {
    pd <- NA
    dic <- NA
  }

  return(list(accept=c(accept_beta/sim,accept_logmu/sim,accept_logsigma/sim),
     chain=prmt.ts,dic=dic,pd=pd,delta=delta,prop=prmt.prop,curr=prmt.curr,
     lik=all.lik,lik.na=ll.na))
}
```

C.3 Utility functions

These functions provide additional calculations needed in the previous ones. In particular, they implement some probability distributions and perform some important computation checks.

```
ddirich <- function(x,alpha,logdens=FALSE){
  logD <- sum(lgamma(alpha)) - lgamma(sum(alpha))
```

```
  s <- sum((alpha - 1) * log(x))
  if (logdens) pd <- (sum(s) - logD) else pd <- exp(sum(s) - logD)
  return(pd)
}

rdirich <- function(n,a){
  l <- length(a)
  x <- matrix(rgamma(l*n,a),ncol=l,byrow=TRUE)
  sm <- x%*%rep(1,l)
  return(x/as.vector(sm))
}

dmixgamma <- function(x,weight,rateparam){
numcomp <- length(weight)
  mixcomp <- matrix(NA,nrow=length(x),ncol=numcomp)
  for (i in 1:numcomp) mixcomp[,i] <- dgamma(x,shape=i,rate=rateparam)
  dens <- mixcomp%*%weight
  return(dens)
}

pmixgamma <- function(x,weight,rateparam){
  numcomp <- length(weight)
  mixcomp <- matrix(NA,nrow=length(x),ncol=numcomp)
  for (i in 1:numcomp) mixcomp[,i] <- pgamma(x,shape=i,rate=rateparam)
  cdf <- mixcomp%*%weight
  return(cdf)
}

rmixgamma <- function(n,ww,tt){
  JJ <- length(ww)
  rmixg <- vector(length=n)
  tmp.rmixg <- matrix(NA,nrow=JJ,ncol=n)
  tmp.lbl <- matrix(NA,nrow=JJ,ncol=n)
  for (i in 1:JJ) tmp.rmixg[i,] <- rgamma(n,shape=i,rate=tt)
  tmp.lbl <- rmultinom(n,1,ww)
  for (i in 1:JJ) rmixg <- rmixg + tmp.lbl[i,]*tmp.rmixg[i,]
  return(rmixg)
}

dtriang <- function(x,a=0,b=1,c=2) {
    if ((a<=b & b<c) | (a<b & b<=c)) {
      dt <- vector(length = length(x))
      for (i in 1:length(x)) {
        if  (x[i] >= a & x[i] <= c) {
          if (x[i] <= b) {
             dt[i] = 2*(x[i]-a)/((c-a)*(b-a)) }
          else {
             dt[i] = 2*(c-x[i])/((c-a)*(c-b)) } }
        else {
          dt[i] = NA } }
      return(dt) }
    else {stop("Inconsistent parameter values.\n")}
}
```

```
ptriang <- function(x,a=0,b=1,c=2) {
    if ((a<=b & b<c) | (a<b & b<=c)) {
      pt <- vector(length = length(x))
      for (i in 1:length(x)) {
        if  (x[i] >= a & x[i] <= c) {
          if (x[i] <= b) {
             pt[i] = (x[i]-a)^2/((c-a)*(b-a)) }
          else {
             pt[i] = 1-(c-x[i])^2/((c-a)*(c-b)) } }
        else {
          if (x[i] < a) pt[i] = 0
          if (x[i] > c) pt[i] = 1 } }
      return(pt) }
    else {stop("Inconsistent parameter values.\n")}
}

qtriang <- function(prob,a=0,b=1,c=2) {
    if ((a<=b & b<c) | (a<b & b<=c)) {
      qt <- vector(length = length(prob))
      temp <- ptriang(x=b,a=a,b=b,c=c)
      for (i in 1:length(prob)) {
        if  (prob[i] > 0 & prob[i] < 1) {
          if (prob[i] <= temp) {
             qt[i] = sqrt(prob[i]*(c-a)*(b-a))+a }
          else {
             qt[i] = c-sqrt((1-prob[i])*(c-a)*(c-b)) } }
        else {
          if (prob[i] == 0) qt[i] = a
          if (prob[i] == 1) qt[i] = c
          if (prob[i] < 0) qt[i] = NA
          if (prob[i] > 1) qt[i] = NA } }
      return(qt) }
    else {stop("Inconsistent parameter values.\n")}
}

rtriang <- function(n=1,a=0,b=1,c=2){
    if ((a<=b & b<c) | (a<b & b<=c)) {
      if  (n >= 1) {
        rt <- vector(length = n)
        for (i in 1:n) {
            u <- runif(1)
            rt[i] = qtriang(u,a=a,b=b,c=c) } }
      else {stop("n must be greater than or equal to 1.\n")}
      return(rt) }
    else {stop("Inconsistent parameter values.\n")}
}

dinvgamma <- function(x,shape,scale=1){
    if (shape <= 0 | scale <= 0) stop("Shape or scale parameter negative in dinvgamma().\n")
    alpha <- shape
    beta <- scale
    log.density <- alpha * log(beta) - lgamma(alpha) - (alpha+1) * log(x) - (beta/x)
```

```
    return(exp(log.density))
}

prob.predict <- function(mcmc.w,mcmc.theta,thresh,prnt=FALSE){
  numsim <- dim(mcmc.w)[1]
  JJ <- dim(mcmc.w)[2]
  pred.prob <- vector(length=numsim)
  for (i in 1:numsim){
    for (j in 1:JJ){
      pred.prob[i] <- pred.prob[i] + mcmc.w[i,j]*(1-pgamma(thresh,shape=j,rate=mcmc.theta[i]))
    }
  }
  return(pred.prob)
}

prob.predict.ln <- function(mcmc.mu,mcmc.sd,thresh){
  numsim <- length(mcmc.mu)
  pred.prob <- vector(length=numsim)
  for (i in 1:numsim) pred.prob[i] <- 1-plnorm(thresh,meanlog=mcmc.mu[i],sdlog=mcmc.sd[i])
  return(pred.prob)
}

poly.dmatrix <- function(p,df,intercept=FALSE,first.deriv=FALSE){
   if (!first.deriv) {
    dmatrix <- cbind(p)
    if (df > 1) for (i in (2:df)) dmatrix <- cbind(dmatrix,p^i)
    if (intercept) dmatrix <- cbind(1,dmatrix) }
  else {
    dmatrix <- cbind(rep(1,length(p)))
    if (df > 1) for (i in (2:df)) dmatrix <- cbind(dmatrix,i*p^(i-1)) }
  dimnames(dmatrix) <- NULL
  return(dmatrix)
}

check.beta.poly <- function(prmt,yy1,degf){
  numcomp <- length(prmt)-degf-2
  qq1 <- sort(yy1)
  dens1.q1 <- dmixgamma(qq1,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  p1 <- pmixgamma(qq1,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  derivat <- poly.dmatrix(p1,degf,first.deriv=TRUE)
  constr <- 1/(dens1.q1*qq1)
  B <- derivat%*%prmt[2:(degf+1)]
  return(any(B > constr))
}

check.beta.ns <- function(prmt,yy1,degf){
  numcomp <- length(prmt)-degf-2
  qq1 <- sort(yy1)
  dens1.q1 <- dmixgamma(qq1,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  p1 <- pmixgamma(qq1,prmt[(degf+2):(degf+numcomp+1)],prmt[degf+numcomp+2])
  knots <- sort(c(rep(range(p1),4),attr(ns(p1,degf),"knots")))
```

```
  derivat <- splineDesign(knots,p1,4,rep(1,length(p1)))
  derivat <- derivat[,-1,drop=FALSE]
  derivat.const <- splineDesign(knots,range(p1),4,c(2, 2))
  derivat.const <- derivat.const[,-1,drop=FALSE]
  qr.derivat.const <- qr(t(derivat.const))
  derivat <- as.matrix((t(qr.qty(qr.derivat.const, t(derivat))))[, -(1:2)])
  constr <- 1/(dens1.q1*qq1)
  B <- derivat%*%prmt[2:(degf+1)]
  return(any(B > constr))
}

check.beta.ln <- function(prmt,yy1){
  qq1 <- sort(yy1)
  dens1.q1 <- dlnorm(qq1,meanlog=prmt[3],sdlog=prmt[4])
  p1 <- plnorm(qq1,meanlog=prmt[3],sdlog=prmt[4])
  derivat <- 1/dnorm(qnorm(p1))
  constr <- 1/(dens1.q1*qq1)
  B <- prmt[2]*derivat
  return(any(B > constr))
}

DICmixgamma <- function(ww,theta,obs){
  nsim <- length(theta)
  dic.tmp <- rep(NA,nsim)
  for (i in 1:nsim) dic.tmp[i] <- -2*sum(log(dmixgamma(obs,ww[i,],theta[i])))
  dic.tmp.mean <- -2*sum(log(dmixgamma(obs,mean(as.data.frame(ww)),mean(theta))))
  pd <- mean(dic.tmp) - dic.tmp.mean
  dic <- mean(dic.tmp) + pd
  return(list(dic,pd))
}
```

Bibliography

[1] M. AITKIN. A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing*, **6**:251–262, 1996.

[2] M. AITKIN. A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics*, **55**:117–128, 1999.

[3] M. AITKIN. Likelihood and Bayesian analysis of mixtures. *Statistical Modelling*, **1**:287–304, 2001.

[4] M. AITKIN AND D. B. RUBIN. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society, B*, **47**:67–75, 1985.

[5] C. E. ANTONIAK. Mixture of Dirichlet processes with applications to some Bayesian nonparametric problems. *Annals of Statistics*, **2**:1152–1174, 1974.

[6] J. BEIRLANT, Y. GOEGEBEUR, J. SEGERS, AND J. TEUGELS. *Statistics of Extremes: Theory and Applications*. Wiley, Chichester, 2004.

[7] J. M. BERNARDO AND A. F. M. SMITH. *Bayesian Theory*. Wiley, Chichester, 1994.

[8] T. BJERKEDAL. Acquisition of Resistance of Guinea Pigs Infected with Different Doses of Virulent Tubercle Bacilli. *American Journal of Hygiene*, **72**:130–148, 1960.

[9] I. N. BRONSHTEIN, K. A. SEMENDYAYEV, G. MUSIOL, AND H. MUEHLIG. *Handbook of Mathematics*. Springer, Berlin, Fourth edition, 2004.

[10] M. B. BUNTIN AND A. M. ZASLAVSKY. Too much ado about two-part models and transformation? Comparing methods of modeling Medicare expenditures. *Journal of Health Economics*, **23**:525–542, 2004.

[11] B. P. CARLIN AND S. CHIB. Bayesian model choice via Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society, B*, **57**:473–484, 1995.

[12] B. P. CARLIN AND T. A. LUOIS. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC, Boca Raton, Second edition, 2000.

[13] G. CASELLA AND R. L. BERGER. *Statistical Inference*. Duxbury, CA, Second edition, 2002.

[14] L. COPE. *Some Asymptotic Properties of Smooth Quantile Ratio Estimation*. PhD thesis, Department of Applied Mathematics Johns Hopkins University, Baltimore, MD, 2003.

[15] H. A. DAVID AND H. N. NAGARAJA. *Order Statistics*. Wiley, Hoboken, NJ, Third edition, 2003.

[16] A. C. DAVISON. *Statistical Models*. Cambridge University Press, Cambridge, UK, 2003.

[17] A. C. DAVISON AND D. V. HINKLEY. *Bootstrap Methods and their Applications*. Cambridge University Press, Cambridge, UK, 1997.

[18] C. DE BOOR. *A Practical Guide to Splines*. Springer–Verlag, New York, Revised edition, 2001.

[19] J. DIEBOLT AND C. P. ROBERT. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, B*, **56**:363–375, 1994.

[20] K. DOKSUM. Empirical probability plots and statistical inference for nonlinear models in the two-sample case. *Annals of Statistics*, **2**:267–277, 1974.

[21] K. DOKSUM AND G. L. SIEVERS. Plotting with confidence: graphical comparisons of two populations. *Biometrika*, **63**:421–434, 1976.

[22] F. DOMINICI, L. COPE, D. Q. NAIMAN, AND S. L. ZEGER. Smooth quantile ratio estimation (SQUARE). *Biometrika*, **92**:543–557, 2005.

[23] F. DOMINICI AND S. L. ZEGER. Smooth quantile ratio estimation with regression: estimating medical expenditures for smoking attributable diseases. *Biostatistics*, 2006. (to appear).

[24] N. DUAN. Smearing estimate: a nonparametric retransformation method. *Journal of the American Statistical Association*, **78**:605–610, 1983.

[25] B. EFRON AND R. J. TIBSHIRANI. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, New York, 1993.

[26] P. EMBRECHTS, C. KLUPPELBERG, AND T. MIKOSCH. *Modelling Extremal Events*. Springer, Berlin, 1997.

[27] D. M. ESCOBAR AND M. WEST. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90**:577–588, 1995.

[28] B. S. EVERITT. *The Cambridge Dictionary of Statistics*. Cambridge University Press, Cambridge, Second edition, 2002.

[29] T. S. FERGUSON. Prior distributions on spaces of probability measures. *Annals of Statistics*, **2**:615–629, 1974.

[30] F. GALTON. *Natural Inheritance*. Macmillan, London, 1889.

[31] A. E. GELFAND AND A. KOTTAS. Bayesian Semiparametric Regression for Median Residual Life. *Scandinavian Journal of Statistics*, **30**:651–665, 2003.

[32] A. GELMAN, J. B. CARLIN, H. S. STERN, AND D. B. RUBIN. *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, Second edition, 2004.

[33] W. G. GILCHRIST. *Statistical Modelling with Quantile Functions*. Chapman & Hall/CRC, New York, 2000.

[34] A. HALD. *A History of Mathematical Statistics from 1750 to 1930*. Wiley, New York, 1998.

[35] T. HASTIE, R. J. TIBSHIRANI, AND J. FRIEDMAN. *The Elements of Statistical Learning*. Springer, New York, 2001.

[36] N. L. HJORT. Topics in Nonparametric Bayesian Statistics. In P. J. GREEN, N. L. HJORT, AND S. RICHARDSON, editors, *Highly Structured Stochastic Systems*, pages 455–487. Oxford University Press, Oxford, UK, 2003.

[37] N. L. HJORT AND S. PETRONE. Nonparametric Quantile Inference Using Dirichlet Processes. Unpublished manuscript, 2005.

[38] N. L. HJORT AND S. WALKER. Quantile Pyramids for Bayesian Nonparametrics. Unpublished manuscript, 2004.

[39] P. J. HUBER. *Robust Statistics*. Wiley, New York, 1981.

[40] P. J. HUBER. Robust Statistical Procedures. In *CBMS-NSF Regional Conference Series in Applied Mathematics, Number 68*. Soc. Industr. Appl. Math., Philadelphia, Pennsylvania, Second edition, 1996.

[41] H. ISHWARAN, L. F. JAMES, AND J. SUN. Bayesian Model Selection in Finite Mixtures by Marginal Density Decompositions. *Journal of the American Statistical Association*, **96**:1316–1332, 2001.

[42] H. ISHWARAN AND M. ZAREPOUR. Exact and approximate sum-representations for the Dirichlet process. *Canadian Journal of Statistics*, **30**:269–283, 2002.

[43] R. C. JANSEN. Maximum likelihood in a generalized linear finite mixture model by using the EM algorithm. *Biometrics*, **49**:227–231, 1993.

[44] A. JASRA, C. C. HOLMES, AND D. A. STEPHENS. Markov Chain Monte Carlo Methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, **20**:50–67, 2005.

[45] H. JEFFREYS. *Theory of Probability*. Oxford University Press, Oxford, Third edition, 1967.

[46] E. JOHNSON, F. DOMINICI, M. GRISWOLD, AND S. L. ZEGER. Disease cases and their medical costs attribuitable to smoking: an analysis of the national medical expenditure survey. *Journal of Econometrics*, **112**:135–151, 2003.

[47] R. KASS AND A. RAFTERY. Bayes factors. *Journal of the American Statistical Association*, **90**:773–795, 1995.

[48] R. KOENKER. *Quantile Regression.* Cambridge University Press, New York, 2005.

[49] R. KOENKER AND G. S. BASSETT. Regression quantiles. *Econometrica*, **46**:33–50, 1978.

[50] G. KOOP. *Bayesian Econometrics.* Wiley, New York, 2003.

[51] A. KOTTAS AND A. E. GELFAND. Bayesian Semiparametric Median Regression Modeling. *Journal of the American Statistical Association*, **96**:1458–1468, 2001.

[52] A. KOTTAS AND M. KRNJAJIC. Bayesian Nonparametric Modeling in Quantile Regression. Technical report, Department of Applied Mathematics and Statistics, Jack Baskin School of Engineering, University of California, Santa Cruz, 2005.

[53] N. M. LAIRD. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, **73**:805–811, 1978.

[54] N. M. LAIRD. Empirical Bayes estimates using the nonparametric estimate of the prior. *Journal of Statistical Computation and Simulation*, **73**:805–811, 1982.

[55] E. L. LEHMANN. *Nonparametrics: Statistical Methods Based on Ranks.* Holden Day, San Francisco, 1974.

[56] E. L. LEHMANN AND G. CASELLA. *Theory of Point Estimation.* Springer, New York, Second edition, 1998.

[57] B. G. LINDSAY. The geometry of likelihoods: a general theory. *Annals of Statistics*, **11**:86–94, 1983.

[58] B. G. LINDSAY. *Mixture Models: Theory, Geometry and Applications.* Institute of Mathematical Statistics, Hayward, California, 1995.

[59] W. G. MANNING. The logged dependent variable, heteroscedasticity, and the retransformation problem. *Journal of Health Economics*, **17**:283–295, 1998.

[60] W. G. MANNING AND J. MULLAHY. Estimating log models: to transform or not to transform? *Journal of Health Economics*, **20**:461–494, 2001.

[61] J. M. MARIN, K. MENGERSEN, AND C. P. ROBERT. Bayesian Modelling and Inference on Mixtures of Distributions. In D. DEY AND C. R. RAO, editors, *Handbook of Statistics 25*. Elsevier-Sciences, 2005. (to appear).

[62] S. MATTHEW. *Bayesian Methods for Mixtures of Normal Distributions*. PhD thesis, Magdalen College, Oxford, 1997.

[63] G. MCLACHLAN AND T. KRISHNAN. *The EM Algorithm and Extensions*. Wiley, New York, 1996.

[64] G. MCLACHLAN AND D. PEEL. *Finite Mixture Models*. Wiley, New York, 2000.

[65] G. J. MCLACHLAN AND K. E. BASFORD. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[66] A. J. MCNEIL, R. FREY, AND P. EMBRECHTS. *Quantitative Risk Management*. Princeton University Press, Princeton, NJ, 2005.

[67] J. MULLAHY. Much ado about two: reconsidering retransformation and the two-part model in health econometrics. *Journal of Health Economics*, **17**:247–281, 1998.

[68] V. N. NAIR. Q-Q Plots with Confidence Bands for Comparing Several Populations. *Scandinavian Journal of Statistics*, **9**:193–200, 1982.

[69] H. J. NEWTON. A Conversation with Emanuel Parzen. *Statistical Science*, **17**:357–378, 2002.

[70] G. O'HAGAN AND J. FORSTER. *Bayesian Inference*. Arnold, London, Second edition, 2004.

[71] E. PARZEN. Nonparametric statistical data modeling. *Journal of the American Statistical Association*, **74**:105–121, 1979.

[72] E. PARZEN. Quantile probability and statistical data modeling. *Statistical Science*, **19**:652–662, 2004.

[73] S. PETRONE. Bayesian density estimation using Bernstein polynomials. *Canadian Journal of Statistics*, **27**:105–126, 1999.

[74] S. PETRONE. Random Bernstein polynomials. *Scandinavian Journal of Statistics*, **26**:373–393, 1999.

[75] S. PETRONE AND P. VERONESE. A generalization of Bernstein polynomials with applications in Bayesian nonparametrics. Unpublished manuscript, 2003.

[76] M. POSTMAN, J. P. HUCHRA, AND M. J. GELLER. Probes of large-scale structure in the Corona Borealis region. *The Astronomical Journal*, **92**:1238–1247, 1986.

[77] R DEVELOPMENT CORE TEAM. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[78] NATIONAL CENTER FOR HEALTH SERVICES RESEARCH. *National Medical Expenditure Survey*. National Center for Health Services Research and Health Technology Assessment, 1987.

[79] S. I. RESNICK. *Extreme Values, Regular Variation, and Point Processes*. Springer, New York, 1987.

[80] S. I. RESNICK. *Probabilistic and Statistical Modeling of Heavy Tail Phenomena*. Springer, New York, 2006. (to appear).

[81] S. RICHARDSON AND P. J. GREEN. On Bayesian analysis of mixtures with an unknown number of components (with Discussion). *Journal of the Royal Statistical Society, B*, **59**:731–792, 1997.

[82] H. ROBBINS. An empirical Bayes approach to statistical decision problems. *Annals of Mathematical Statistics*, **35**:1–20, 1964.

[83] H. ROBBINS. Some thoughts on empirical Bayes estimation. *Annals of Statistics*, **11**:713–723, 1983.

[84] C. P. ROBERT. *The Bayesian Choice*. Springer, New York, 1994.

[85] C. P. ROBERT. Mixtures of distributions: inference and estimation. In W. R. GILKS, S. RICHARDSON, AND D. J. SPIEGELHALTER, editors, *Markov Chain Monte Carlo in Practice*, pages 441–464. Chapman & Hall/CRC, New York, 1996.

[86] C. P. ROBERT AND G. CASELLA. *Monte Carlo Statistical Methods*. Springer, New York, Second edition, 2004.

[87] K. ROEDER. Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, **85**:617–624, 1990.

[88] K. ROEDER AND L. WASSERMAN. Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, **92**:894–902, 1997.

[89] D. RUPPERT, M. P. WAND, AND R. J. CARROLL. *Semiparametric Regression*. Cambridge University Press, Cambridge, 2003.

[90] S. SAHU AND R. CHENG. A fast distance based approach for determining the number of components in mixtures. *Canadian Journal of Statistics*, **31**:3–22, 2003.

[91] A. J. SCALLAN. Fitting a mixture distribution to complex censored survival data using generalized linear models. *Journal of Applied Statistics*, **26**:747–753, 1999.

[92] M. J. SCHERVISH. *Theory of Statistics*. Springer, New York, 1995.

[93] R. J. SERFLING. *Approximation Theorems of Mathematical Statistics*. Wiley, New York, 1980.

[94] S. J. SHEATHER. Density estimation. *Statistical Science*, **19**:588–597, 2004.

[95] G. R. SHORACK. *Probability for Statisticians*. Springer, New York, 2000.

[96] G. R. SHORACK AND J. A. WELLNER. *Empirical Processes with Applications in Statistics.* Wiley, New York, 1986.

[97] B. W. SILVERMAN. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, London, 1986.

[98] M. STEPHENS. *Bayesian Methods for Mixtures of Normal Distributions.* PhD thesis, University of Oxford, Oxford, UK, 1997.

[99] M. STEPHENS. Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods. *Annals of Statistics*, **28**:40–74, 2000.

[100] D. M. TITTERINGTON. Mixture distributions. In *Encyclopedia of Statistical Sciences*, pages 399–407. Wiley, New York, 1997.

[101] D. M. TITTERINGTON, A. F. M. SMITH, AND U. E. MAKOV. *Statistical Analysis of Finite Mixture Distributions.* Wiley, Chichester, 1985.

[102] A. W. VAN DER VAART. *Asymptotic Statistics.* Cambridge University Press, Cambridge, UK, 1998.

[103] A. W. VAN DER VAART AND J. A. WELLNER. *Weak Convergence and Empirical Processes.* Springer, New York, 1996.

[104] L. WASSERMAN. *All of Nonparametric Statistics.* Springer, New York, 2006.

[105] M. WEDEL AND W. S. DESARBO. A mixture likelihood approach for generalized linear models. *Journal of Classification*, **12**:21–55, 1995.

[106] R. WILCOX. Comparing two independent groups via multiple quantiles. *The Statistician*, **44**:91–99, 1995.

[107] M. B. WILK AND R. GNANADESIKAN. Probability plotting methods for the analysis of data. *Biometrika*, **55**:1–17, 1968.

[108] K. YU, Z. LU, AND J. STANDER. Quantile regression: applications and current research areas. *The Statistician*, **52**:331–350, 2003.

[109] K. YU AND R. A. MOYEED. Bayesian quantile regression. *Statistics and Probability Letters*, **54**:437–447, 2001.

Index