Chapter-5

Bayes Estimation on Mixture of some Life time Distribution

5.1 Introduction

A mixture distribution is a compounding of statistical distributions, which arises when sampling from inhomogeneous populations (or mixed populations) with a different probability density function in each component. The mixture model has been extensively studied and challenged in various scientific fields such as astronomy, genetics, bioinformatics, computer science, ecology, economics, engineering, robotics and biostatistics. For instance, in genetics, location of quantitative traits on a chromosome and interpretation of micro arrays both relate to mixtures, while, in computer science, spam filters and web context analysis (Jordan 2004) start from a mixture assumption to distinguish spam’s from regular emails and group pages by topic, respectively. A wide variety of mixture situations are described in Lindsay (1995). A finite mixture has a finite number of components. A general mixture density with k components can be written as

\[ f(x|\theta) = \sum_{j=1}^{k} p_j f_j(x|\theta_j) \]  

(5.1)

Where, \(0 \leq p_i \leq 1\), \(\sum_{i=1}^{k} p_i = 1\) and \(f_j\) is a density with parameter vector \(\theta_j\). In theory the \(f_j\)’s could be any parametric density, although in practice they are usually from the same parametric family. The parameters \(p_1,...,p_k\) will be called the mixing weights or mixing proportions. If \(\{x_1,...,x_n\}\) is a random sample from (5.1), then the likelihood function

\[ L(\theta, p|x) = \prod_{i=1}^{n} \sum_{j=1}^{k} p_j f(x_i|\theta_j) \]  

(5.2)

A part of this Chapter is communicated in the Journal, Statistics and Computing
leads to $k^n$ terms when the inner sums are expanded. As pointed out in Dempster et al. (1977), a mixture model can always be expressed in terms of missing (or incomplete) data. If, for $1 \leq i \leq n$, $z_i$ is a k-dimensional vector indicating to which component $x_i$ belongs, such that $z_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{k} z_{ij} = 1$, the density of the completed data $(x_i, z_i)$ is

$$\prod_{j=1}^{k} p_j^{z_{ij}} f_{j}^{z_{ij}}(x_j | \theta_j)$$

(5.3)

From this perspective, the model if hierarchical with, on top, the true parameters of the mixture, $\theta$, then the missing data whose distribution depends on $\theta$, $z \sim f(z | \theta)$, and, at the bottom, the observed data $x$, with distribution depending on $z$ and $\theta$, $x \sim f(x | 0, z)$. The variable $z_i$ is referred as auxiliary or latent variable, which identifies to which component the observation $x_i$ belongs. It can be seen that the distribution of $z = x_i | z_i$ follows multinomial distribution with parameters $p_1, \ldots, p_k$, i.e. $z \sim M_k(1; p_1, \ldots, p_k)$.

There are several motivations for considering mixtures of distributions as a useful extension to “standard” distributions. The most natural approach is to envisage a dataset as constituted of several strata or subpopulations. Each of the $x_i$’s is thus a priori distributed from either of $f_j$’s with probability $p_j$. Depending on the setting, the inferential goal may be either to reconstitute the groups, usually called clustering, to provide estimators for the parameters of the different groups or even to estimate the number of groups.

The Expected-Maximization method (Dempster, Laird and Rubin, 1977) is a basic tool for the estimation of parameters by maximum likelihood (m.l.) in mixture models. The Expected-Maximization (EM) method works iteratively, replacing each missing data $z_{ij}$ by its expectation.
\[ z_{ij}^{(m)} = \frac{p_j^{(m)} f_j^{(m)}(x_i)}{\sum_{t=1}^{k} p_t^{(m)} f_t^{(m)}(x_i)} \]  

(5.4)

where \( m \) indexes the current iteration step and \( p_j^{(m)} \) and \( f_j^{(m)} \) are the current evaluations of the parameters. New maximum likelihood estimates are then derived from the pseudo completed data \((x_i, z_{i}^{(m)})\). To avoid some defects of the EM method, Celeux and Deibolt (1985) introduced a stochastic version called SEM. Instead of estimating the missing data, they simulate \( z_{ij}' \) from their conditional distribution, i.e. multinomial distributions using weights \((5.3)\). For more detail about EM algorithm one may refer to section- 1.2.8 of chapter-1.

Bayesian approaches to mixture modeling have attracted great interest among researchers and practitioners alike. Bayesian framework also allows the complicated structure of a mixture model to be decomposed into a set of simpler structures through the use of hidden or latent variables. When the number of components is unknown, it can well be argued that the Bayesian paradigm is the only sensible approach to its estimation (Richardson and Green 1997). If \( \pi(\theta, p) \) denote the prior distribution on \((\theta, p)\). The posterior distribution is then

\[
\pi(\theta, p | x) \propto L(\theta, p | x) \pi(\theta, p) 
\]  

(5.5)

Where \( L(\theta, p | x) \) is the likelihood of mixture distribution as defined in (5.2). In terms of missing data structure the posterior distribution becomes

\[
\pi(\theta, p | x, z) \propto \prod_{i=1}^{n} \prod_{j=1}^{k} p_j^{z_{ij}} f_j^{z_{ij}}(x_i | \theta_j) \pi(\theta_j, p_j) 
\]  

(5.6)

The Gibbs sampler is the most commonly used approach in Bayesian mixture estimation. Here we present Gibbs algorithm given by Diebolt and Christian P. Robert (1994). Starting with an initial value \( \theta^{(0)} \), the algorithm simulates the following way:
Set $t=0.$

**Algorithm-5.1**

**Step - 1** Generate $z^{(t)} \sim f(z|x, \theta^{(t)})$,

(1) Generate $\theta_1^{(t+1)} \sim \Pi(\theta_1| x, z^{(m)}, \theta_2^{(m)}, ..., \theta_k^{(m)})$

. .

(k) Generate $\theta_k^{(t+1)} \sim \Pi(\theta_k| x, z^{(m)}, \theta_1^{(m)}, ..., \theta_{k-1}^{(m)})$

**Step - 2** Repeat step-1 for $t = 1,2,..,N$

Thus Gibbs sampling produces a Markov chain $(\theta^{(t)})$, usually ergodic with stationary distribution $\Pi(\theta|x)$. Therefore, after $t_0$ initial steps ("warming up"), the random variables $\theta^{(t_0+1)},...,\theta^{(t_0+k)}$ can be considered to be approximately distributed according to $\Pi(\theta|x)$. The resulting sample can then lead to an approximation of any well-defined posterior quantity by

$$E^{\Pi}[h(\theta)|x] \approx \frac{1}{K} \sum_{k=1}^{K} h(\theta^{(t_0+k)})$$

by the ergodic theorem. Gelfand and Smith (1990) propose an improvement on this approximation, namely to use instead the average of the conditional expectations $E^{\Pi}[h(\theta)|x, z^{(t_0+k)}]$, based on the Rao-Blackwell theorem. Since $\theta^{(t_0+k)}$ are correlated, several modifications of the algorithm have been proposed. See Besag and Green (1993), Smith and Robert (1993) and Tierney (1994) for detailed discussion on implementation.

Perfect sampler is another approach to generate a sample from posterior distribution in mixture context. Perfect sampling (perfect simulation, exact simulation, backward coupling, coupling from the past or CFTP) algorithms enable exact simulation of the invariant (or stationary) measure $\pi$ of a Markov chain, either exactly (that is, by drawing a random sample known to be from $\pi$) or approximately, but with computable order of accuracy. These were sparked by the seminal paper of Propp and Wilson (1996) in 1996, and several variations and extensions of this idea.
have appeared since. The essential idea of these approaches is to find a random epoch \(-T\) in the past such that, if we construct sample paths (according to a transition law of the chain that is invariant for \(\pi\)) from every point in the state space starting at \(-T\), then all paths will have coupled successfully by time zero. The common value of the paths at time zero is a draw from \(\pi\).

Let \(m^{th}\) state of finite state space \(S\) be denoted \(x_m, m=1,\ldots,M\), where \(M\) denotes number of states. Let \(X^{i, j}(t_1, x_m)\) denote the state at time \(t_2\) of a chain started in state \(x_m\) at time \(t_1 < t_2\). The algorithm can then be expressed as follows.

**Algorithm-5.2 (CFTP)**

*Step – 1* Set the starting value for the time to go back, \(T_0 \leftarrow -1\).

*Step – 2* Generate a random vector \(U_{T_0+1}^{T_0+1}\).

*Step – 3* Start a chain in each state \(x_m, m=1,\ldots,M\), of \(S\) at time \(T_0\), and run the chains \(X^{t+1}(T_0, x_m) = \phi(x^t(T_0, x_m), U^{t+1})\) to time \(0, t = T_0, T_0+1,\ldots,-1\).

*Step – 4* Check for coalescence at time \(0\), that is check if \(X^0(T_0, x_m)\) occupy the same state \(\forall m\). If so, this common value \(X^0\) is returned. Otherwise let \(T_0 \leftarrow (T_0 - 1)\) and continue from step -2.

For more detail about perfect sampling one can refer Dimakos (2001) for an excellent introduction, Schneider (2003), Mira (2001) and Wilson (1998), whose Website [http://dimacs.rutgers.edu/~dbwilson/exact.html](http://dimacs.rutgers.edu/~dbwilson/exact.html) is constantly updated. Casella et al. (2002) have used perfect sampling for posterior distributions associated with mixtures of exponential families and conjugate priors. In their paper, they demonstrate two approach to generate perfect sampler, (i) perfect slice sampler (ii) Automated coupling given by Breyer and Robert’s (2001). Here we present second approach, which is applied to small as well as large samples.

Given two Markov chains \((x_0^{(i)})\) and \((x_1^{(i)})\), the first chain \((x_0^{(i)})\) is run using standard Metropolis-Hasting scheme, while the second chain \((x_1^{(i)})\) is then based on the same proposed value \(y_t\) generate in first chain, with the difference that the Metropolis-Hasting scheme is now an independent scheme. The algorithm is described as:
Initialize $X_0$; set $t=0$.

**Algorithm –5.3 Automatic Coupling**

Initialize $x_0^{(t)}$, $x_1^{(t)}$; set $t=0$.

**Step 1:** Generate $u_t \sim U(0,1)$.

**Step 2:**

$$x_0^{(t+1)} = \begin{cases} 
    y_t & \text{if } u_t \leq \frac{\pi(y_t) q(x_0^{(t)} | y_t)}{\pi(x_0^{(t)}) q(y_t | x_0^{(t)})} \\
    x_0^{(t)} & \text{otherwise}
\end{cases}$$

**Step 3:**

$$x_1^{(t+1)} = \begin{cases} 
    y_t & \text{if } u_t \leq \frac{\pi(y_t) q(x_1^{(t)} | x_0^{(t)})}{\pi(x_1^{(t)}) q(y_t | x_0^{(t)})} \\
    x_1^{(t)} & \text{otherwise}
\end{cases}$$

**Step 4:** Repeat steps 2 to 3 until both accept a proposal $y_t$, i.e. the chains coalesce.

**Step 5:** Repeat steps 1 to 3 for $t=1,2,\ldots, N$.

Since mixture models are often used in situations where individual components are meaningless, it is important that methods are available for assessing model adequacy in terms of the number of mixture components. The number of mixture components is a key ingredient of the modeling. The inference problem can than be cast as one of choosing between the two, three or $n$ component models. Within the Bayesian approach, a definite advance has been the application by Richardson and Green (1997) of the reversible jump MCMC method of Green (1995), which allowed one to sample from the joint posterior distribution of all the parameters, including the number $k$ of components. Beside Richardson and Green (1997), other researchers have studied methods to estimate the posterior distribution of $k$. Some of them (Nobile 1994, Roeder and Wasserman 1997) have provided estimates of the marginal likelihoods of $k$ components, then used Bayes theorem to obtain the posterior of $k$. Stephen(2000a) have describe Birth and Death MCMC as an alternative of Reversible jump MCMC. Some other authors (Carlin and Chib 1995, Chib 1995, Raftery 1996) have avoided placing a prior distribution on $k$; instead, they have estimated the marginal likelihoods of $k$ components and used Bayes factors to test $k$ vs. $k+1$ components. Mengersen and Robert (1996) have employed a testing approach too, but relying on the Kullback - Leibler divergence as a measure of distance between mixtures with $k$ and $k+1$ components. Representations of the marginal likelihoods for $k$ components have been derived by
In this chapter we do Bayes estimation on different mixture distributions. Inferences on mixture of Weibull with ‘nearly instantaneous failure’ have been covered in Section-5.2. Mixture of von-Mises has been considered in section-5.3, while section-5.4 deals with mixture of Marshall-Olkin distribution. Various algorithms are proposed to find Bayes estimator of the parameters in mixture distribution.

5.2 Mixture of Weibull with ‘nearly instantaneous failure’

The mixed failure time distribution arises frequently in many different contexts in statistical literature. For instance, when we put units in a life testing experiment, then some of the units fail instantaneously and thereafter the life time of units follow a distribution such as exponential, weibull, gamma etc. Such situations may be represented as a mixture of singular distribution at zero (or $t_0$) and a two-parameter Weibull distribution. Lai et. al. (2007) has proposed a model as a mixture of generalized Dirac delta function and the 2-parameter Weibull. The density of a mixture can be given as:

$$ f(x) = p\delta_d(x-x_0) + q \lambda \alpha x^\alpha - 1 e^{-\lambda x^\alpha}, \quad p + q = 1, q = 1 - p, 0 < p < 1 \quad (5.7) $$

Where

$$ \delta_d(x-x_0) = \begin{cases} \frac{1}{d}, & x_0 \leq x \leq x_0 + d \\ 0, & \text{otherwise} \end{cases} $$

For sufficiently small $d$. Here $p>0$ is the mixing proportion. We note that

$$ \delta(x-x_0) = \lim_{d \to 0} \delta_d(x-x_0), $$

Where $\delta(\cdot)$ is the Dirac delta function that is well known in mathematical analysis. One can view the Dirac delta function as a normal distribution having a zero mean
and standard deviation that tends to 0. For a fixed value of d, the distribution becomes a mixture of a Weibull with Uniform distribution.

Here we consider the special case \( t_0 = 0 \) for the model presented in (5.7) so the given model becomes a mixture of Weibull with a uniform distribution. The pdf of mixture of Weibull with ‘nearly instantaneous failure’ occurring uniformly over \([0, d]\) can be given as

\[
f(x) = \frac{p}{d} + (1-p) \alpha \lambda x^{\alpha-1} e^{-(\lambda x)^\alpha}, 0 < p < 1 \tag{5.8}
\]

Taking \( f_1 \sim U (0, d) \) and \( f_2 \sim \text{Weibull} (\alpha, \lambda) \), then (5.8) can be written as

\[
f(x) = p f_1(x_i | d) + (1 - p) f_2(x_i | \alpha, \lambda) \tag{5.9}
\]

If \( X_1, X_2, ..., X_n \) are the random sample of size \( n \) from (5.8), then the likelihood function is given by

\[
L(p, \alpha, \lambda, d | x) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} p f_1(x_i | d) + (1 - p) f_2(x_i | \alpha, \lambda)
\]

Introducing Latent variable \( Z_i \) in the likelihood function, the likelihood function becomes

\[
L(p, \alpha, \lambda, d | x, z) = \prod_{i=1}^{n} \left\{ p f_1(x_i | d) \right\}^{z_i} \left\{ (1 - p) f_2(x_i | \alpha, \lambda) \right\}^{1-z_i} \tag{5.10}
\]

Such that

\[
P(z_i = 1) = p f_1(x_i | d), \quad P(z_i = 0) = (1 - p) f_2(x_i | d)
\]

Then Log-likelihood after simplification can be written as

\[
\ln(L(p, \alpha, \lambda, d | x, z)) = \sum_{i=1}^{n} z_i \ln \left\{ \frac{p}{1-p} \left[ \frac{f_1(x_i | d)}{f_2(x_i | \alpha, \lambda)} \right] \right\} + \ln(1 - p) f_2(x_i | \alpha, \lambda) \tag{5.11}
\]
5.2.1 Maximum Likelihood Estimators

To find MLE of the parameters of an underlying distribution we use EM algorithm given by Dempster, Laird and Rubin (1977).

The expectation step or E-step computes the expected likelihood for the complete data. The expectation is with respect to the computed conditional distribution of latent variables. Let $\theta$ be the complete collection of parameters occurring in the mixture, i.e. $\theta = (d, \alpha, \lambda)$.

**E-Step:** We take expectation and get Q function

$$Q(\theta|\theta^{(t)}) = E[\ln(L(p, \theta|Z,X)|x,\theta^{(t)})]$$

$$= \sum_{i=1}^{n} E(Z_i | x_i, \theta^{(t)}) \ln \left( \frac{p f_1(x_i | d)}{p f_1(x_i | d) + (1-p) f_2(x_i | \alpha, \lambda)} \right) + \ln((1-p) f_2(x_i | \alpha, \lambda)) \quad (5.12)$$

Where

$$E(Z_i | x_i, \theta^{(t)}) = P(z_i = 1 | x_i, \theta^{(t)})$$

$$= \frac{p f_1(x_i | d^{(t)})}{p f_1(x_i | d^{(t)}) + (1-p) f_2(x_i | \alpha^{(t)}, \lambda^{(t)})}$$

$$= \frac{p^{(t)}}{d^{(t)}} + \frac{(1-p^{(t)}) \alpha^{(t)} \lambda^{(t)} x_i^{\alpha^{(t)}-1} e^{-(\lambda^{(t)} x_i)}}{d^{(t)} + (1-p^{(t)}) \alpha^{(t)} \lambda^{(t)} x_i^{\alpha^{(t)}-1} e^{-(\lambda^{(t)} x_i)}} \quad (5.13)$$

**M-Step:** Here we maximize the expectation, i.e., the Q-function that we computed in the E-step. The two steps should be repeated as necessary

Maximizing $Q(\theta|\theta^{(t)})$ w.r.t. $\theta$ yields the update equations

$$\frac{\partial Q}{\partial p} = 0 \Rightarrow p^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} E(z_i) \quad (5.14)$$
\[
\frac{\partial Q}{\partial d} = 0 \Rightarrow -\frac{1}{d} \sum_{i=1}^{n} E(z_i) = 0
\] (5.15)

\[
\frac{\partial Q}{\partial \lambda} = 0 \Rightarrow \lambda^{(t+1)} = \left[ \frac{\sum_{i=1}^{n} (1 - E(z_i))}{\sum_{i=1}^{n} (1 - E(z_i)) x_i^{(t)}} \right]^{\frac{1}{\alpha^{(t)}}}
\] (5.16)

\[d_{n+1} = \text{Max}(t_i) \ 0 \leq t_i \leq d\]

\[
\frac{\partial Q}{\partial \alpha} = 0 \Rightarrow \sum_{i=1}^{n} (1 - E(z_i))\left[ -\frac{1}{\alpha^{(t)}} + \ln(\lambda^{(t)}) + \ln(x_i) - (\lambda^{(t)} x_i)^{\alpha^{(t)}} \ln(\lambda^{(t)} x_i) \right] = 0
\] (5.17)

Since equation (5.15) cannot solved analytically, we use \(d^{(t+1)} = \text{Max}(x_i)\) since \(0 \leq x_i \leq d\). To obtain \(\alpha^{(t+1)}\), we solve equation (5.17) numerically using Newton-Raphson method.

The EM algorithm starts by assigning initial values to all parameters to be estimated. It then iteratively alternates between two steps, the E-step and M-step. The E-step computes the expected likelihood and the M-step re-estimates all the parameters by maximizing the Q-function. If the convergence criteria is not met, then the parameters \(p, \alpha, \lambda, d\) are updated. We can repeat E-step followed by the M-step until the likelihood converges. Each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function.

### 5.2.2 Bayesian Estimate when all parameters are unknown

Let the prior

\[
\pi(p) = \frac{1}{B(\eta, \beta)} p^{\eta-1} (1 - p)^{\beta-1} \ ; \ \eta > 0, \beta > 0
\]

\[
\pi(d) = \frac{1}{\xi} \ ; \ 0 < d < \xi
\]
\[ \pi(\alpha) = \theta e^{-\alpha \theta}, \quad \theta > 0 \]
\[ \pi(\lambda) = \frac{b^a}{\lambda} \lambda^{-(a+1)} e^{-b/\lambda} \]

Then the joint pdf becomes
\[
f(x; p, d, \alpha, \lambda) = \prod_{i=1}^{n} f(x_i) \pi(p) \pi(d) \pi(\alpha) \pi(\lambda)
\]
\[ = \prod_{i=1}^{n} \left( \frac{p}{d} + (1-p) \alpha \lambda^a x_i \alpha^{-1} e^{-(\lambda x_i)^a} \right) \frac{1}{\xi} \theta e^{-\alpha \theta} \frac{1}{B(\eta, \beta)} p^{\eta-1} (1-p)^{\beta-1} \frac{b^a}{\lambda} \lambda^{-(a+1)} e^{-b/\lambda} \]

(5.18)

Thus, the posterior density is proportional to
\[
\Pi(p, d, \alpha, \lambda | x) \propto \prod_{i=1}^{n} \left( \frac{p}{d} + (1-p) \alpha \lambda^a x_i \alpha^{-1} e^{-(\lambda x_i)^a} \right) e^{-\alpha \theta} p^{\eta-1} (1-p)^{\beta-1} \lambda^{-(a+1)} e^{-b/\lambda} \]

(5.19)

To generate sample from (5.19), we use Population Monte Carlo algorithm and Mixture version of Metropolis-Hasting. The Population Monte Carlo algorithm (Cappé et al. 2004) is an iterated importance sampling scheme, it is an adaptive algorithm that calibrates the proposal distribution to the target distribution at each iteration by learning from the performance of the previous proposal distributions. A complete detail about PMC is given in section 1.2.7 of chapter-1. Letting \( \theta = (p, d, \alpha, \lambda) \), the PMC algorithm can be given as

**Algorithm-5.4**

*Initialize* \( 0_0^{(1)}, ..., 0_0^{(M)} \), set \( t=1 \).

*Step – 1 For* \( i=1,...,M \)

1.1.1 *Generate* \( 0_{(t)}^{(i)} \sim q_{it} (0_{(t-1)}^{(i)}) \)

1.1.2 *Compute*

\[ \rho_{(t)}^{(i)} = \frac{\Pi(0_{(t)}^{(i)})}{q_{it} (0_{(t-1)}^{(i)})} \]
1.2 Compute \( w^{(i)} = \frac{\rho^{(i)}}{\sum_{i=1}^{M} \rho^{(i)}} \)

1.3 Resample \( M \) values with replacement from the \( \theta^{(i)} \)'s using the weights \( w^{(i)} \).

Step 2: Repeat step 1 for \( t=2 \) to \( N \)

Below we present Metropolis- Hasting algorithm corresponding to mixture distribution.

**Algorithm-5.5**

Initialize \( \theta^{(0)} \); set \( t=0 \).

**Step 1:** Generate \( \tilde{\theta} \sim q(\theta | \theta^{(t)}) \) and \( u \sim \text{Uniform}(0,1) \)

**Step 2:** Compute

\[
 r = \frac{f(x | \tilde{\theta}) \pi(\tilde{\theta}) q(\theta^{(t)} | \tilde{\theta})}{f(x | \theta^{(t)}) \pi(\theta^{(t)}) q(\tilde{\theta} | \theta^{(t)})}
\]

**Step 3:** If \( u < r \) then

\[ \theta^{(t+1)} = \tilde{\theta} \]

else

\[ \theta^{(t+1)} = \theta^{(t)} \]

**Step 5:** Repeat steps 1 to 2 for \( t = 1, 2, ..., N \) and return the values \( \theta^{(1)}, ..., \theta^{(N)} \).

It is to be noted that in both the algorithms, the instrumental distributions \( q \), for \( (p, d, \alpha, \lambda) \) is taken as their prior distributions. Now, we define another approach to carry out bayes estimation in mixture context. Introducing latent in the structure, Posterior density can be written as

\[
 \prod_{i=1}^{n} \left( \frac{p}{d} \right)^{z_i} \left\{ (1-p) \alpha \lambda^\alpha x_i^{-\alpha-1} e^{-(\lambda x_i)^\alpha} \right\} \theta e^{-\alpha \theta} \eta^{-1} (1-p)^{\beta-1} \lambda^{-(\sigma+1)} e^{-b/\lambda}.
\]

(5.20)
Then the full conditional distribution of \( (p, d, \alpha, \lambda) \) is given by

\[
\Pi(d \mid z) \propto d^{-n_1}, \quad \Pi(p \mid z) \propto p^{n_1 + \eta - 1} (1 - p)^{n_2 + \beta - 1},
\]

\[
\Pi(\alpha \mid z, \lambda) = \alpha^{n_2} e^{-\alpha \theta} \prod_{i: z_i = 0} x_i^{\alpha-1} e^{-(\lambda x_i)^\theta}
\]

and

\[
\Pi(\lambda \mid z, \alpha) = \lambda^{n_1 \alpha - a - 1} e^{-b/\lambda} \prod_{i: z_i = 0} x_i^{\alpha-1} e^{-(\lambda x_i)^\theta}
\]

(5.21)

Where \( n_1 = \sum_{i=1}^{n} I_{z_i = 1}, n_2 = \sum_{i=1}^{n} I_{z_i = 0} \) and \( n = n_1 + n_2 \)

For generating a sample from this full conditional distribution, we use Algorithm-5.1 of this chapter.

**Algorithm-5.6**

*Initialize* \( p^{(0)}, d^{(0)}, \alpha^{(0)} \) and \( \lambda^{(0)} \); *Set* \( t = 1 \).

**Step – 1** Generate \( z_i^{(t)} \) \( (i = 1, \ldots, n) \) from

\[
P(z_i^{(t)} = 1) \propto \frac{p^{(t-1)}}{d^{(t-1)}}
\]

\[
P(z_i^{(t)} = 0) \propto (1 - p^{(t-1)})^\alpha \lambda^{(t-1)} \lambda^{(t-1)} \prod_{i: z_i = 0} x_i^{\alpha-1} e^{-(\lambda x_i)^\theta}
\]

**Step – 2** Compute \( n_1^{(t)} = \sum_{i=1}^{n} I_{z_i^{(t)} = 1} \) and \( n_2^{(t)} = \sum_{i=1}^{n} I_{z_i^{(t)} = 0} \)

**Step – 3** Generate \( p^{(t)} \sim \text{Beta}(n_1^{(t)} + \eta - 1, n_2^{(t)} + \beta - 1) \)

**Step – 4** Generate \( d^{(t)} \sim \text{Uniform}(d^{(t-1)}) \)

**Step – 5** Generate \( \alpha^{(t)} \sim \Pi(\alpha^{(t)} \mid n_2^{(t-1)}, \lambda^{(t-1)}) \) using M-H algorithm.

**Step – 6** Generate \( \lambda^{(t)} \sim \Pi(\lambda^{(t)} \mid n_2^{(t-1)}, \alpha^{(t-1)}) \) using M-H algorithm.

**Step – 7** Repeat step-1 to 6 for \( t = 2, \ldots, N \)

One may have always question in mind, which algorithm is used to generate a sample from posterior distribution. Here we have used the entire three algorithms to generate a posterior sample. A simulation study is carried out to compare the performance of different algorithms. A sample of size 50 was drawn from the population by taking different values of the parameters. On the basis of this sample, Bayes estimates are calculated using these algorithms. It has seen that, the number of iterations necessary to reach convergence for PMC, it is \(< 5000\), while for M-H, it is
< 15,000 and for Gibbs, it takes > 20,000. A deeper difficulty in implementing Algorithm 5.6 is the existence of computational trapping states. The results are stated in Table 5.1.

### 5.2.3 Application to Real data problem

Consider the wood dryness data of 40 boards analyzed in Muralidharan and Lathika (2006) with \( t_i = 0, i = 1,2,...,28 \) and the other positive observations are 0.0463741, 0.0894855, 0.4, 0.42517, 0.623441, 0.6491, 0.73346, 1.35851, 1.77112, 1.86047, 2.12125, 2.12. Here we have spread the zeros uniformly over an interval taking \( d = 0.042 \) so that \( t_1 = 0, t_2 = 0.0015, t_3 = 0.003, ..., t_{28} = 0.042 \). We obtain MLE estimates using EM Algorithm as \( \hat{\rho}_{MLE} = 0.7625, \hat{\alpha}_{MLE} = 1.5689 \) and \( \hat{\lambda}_{MLE} = 1.2458 \). PMC method is used to find Bayesian estimates. Below we present two bayes estimates related to same data set but with different prior values.

(i) \( \hat{\rho}_{BAYES} = 0.6899, \hat{d}_{BAYES} = 0.2404, \hat{\alpha}_{BAYES} = 0.9363 \) and \( \hat{\lambda}_{BAYES} = 1.9704 \) for \( \theta = 1.6, \eta = 1.1, \xi = 0.9, \beta = 2.3, a = 1.5, b = 1.9 \)

(ii) \( \hat{\rho}_{BAYES} = 0.7187, \hat{d}_{BAYES} = 0.2649, \hat{\alpha}_{BAYES} = 0.9015 \) and \( \hat{\lambda}_{BAYES} = 1.9704 \) for \( \theta = 2.5, \eta = 4.0, \xi = 0.05, \beta = 2.3, a = 2.0, b = 2.0 \)

<table>
<thead>
<tr>
<th>Table 5.1 Bayes estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>( p=0.2 )</td>
</tr>
<tr>
<td>( d=0.05 )</td>
</tr>
<tr>
<td>( \alpha=2.0 )</td>
</tr>
<tr>
<td>( \lambda=1.5 )</td>
</tr>
<tr>
<td>( p=0.6 )</td>
</tr>
<tr>
<td>( d=0.05 )</td>
</tr>
<tr>
<td>( \alpha=2.0 )</td>
</tr>
<tr>
<td>( \lambda=1.5 )</td>
</tr>
</tbody>
</table>
5.3 Mixture of two von Mises Distribution

Bayesian inference on von Mises distribution is already done in Chapter-2. Here we do Bayes estimation on the mixture of two von Mises distribution. The mixture density arising from the two von Mises distributions is given by

\[ f(x) = p f_1(x | \mu_1, \kappa_1) + (1 - p) f_2(x | \mu_2, \kappa_2) \]

where \( f_1 \sim \text{vM}(\mu_1, \kappa_1) \) and \( f_2 \sim \text{vM}(\mu_2, \kappa_2) \) are von Mises densities. Figure 5.1 shows the plot of von Mises distribution for \( \mu_1 = 1.5708, \kappa_1 = 2.0, \mu_2 = 4.7124, \kappa_2 = 5.0 \) and \( p=0.7 \).

In Chapter-2, it has been seen that, joint prior for \( \mu \) and \( \kappa \), independently as the product of exponential prior for \( \kappa \) and wrapped normal prior for \( \mu \) perform well compare to other prior’s, here also we select the same prior’s for \( \mu_1, \kappa_1 \) and \( \mu_2, \kappa_2 \), i.e. \( \mu_1 \sim \text{WN}(\rho_1,\sigma_1^2), \kappa_1 \sim \text{exp}(\eta_1) \), \( \mu_2 \sim \text{WN}(\rho_2,\sigma_2^2) \) and \( \kappa_2 \sim \text{exp}(\eta_2) \). The prior of \( p \) is taken as \( \text{Beta}(\alpha,\beta) \). Let \( \theta=(\mu_1,\mu_2,\kappa_1,\kappa_2) \), then the posterior distribution can be expressed as
\[ \Pi(\theta, p | x) \propto \prod_{i=1}^{n} (p \exp(\kappa_1 \cos(x_i - \mu_1)) + (1 - p)\exp(\kappa_2 \cos(x_i - \mu_2))) \rho^{-1} (1 - \rho)^{\beta - 1} \]

\[ \sum_{k = -\infty}^{\infty} \exp\left(-\left( \mu_1 + 2\pi k - \rho_1 \right)^2 / 2\sigma_1^2 \right) e^{-\kappa_1 / \eta_1} \sum_{k = -\infty}^{\infty} \exp\left(-\left( \mu_2 + 2\pi k - \rho_2 \right)^2 / 2\sigma_2^2 \right) e^{-\kappa_2 / \eta_2} \]

(5.22)

To generate sample from (5.22), we use mixture version of modified sampling-resampling method discussed in chapter 2 (Algorithm 2.1). The algorithm is as follows:

**Algorithm-5.7**

**Step 1:** obtain \((\theta_i, p_i), i=1, 2, ..., k\) from \(\pi(\theta, p)\)

**Step 2:** compute \(w_i = \Pi(\theta_i, p_i) / \pi(\theta_i, p_i)\)

**Step 3:** obtain \(q_i = w_i / \sum_{j=1}^{k} w_j\)

**Step 4:** select \(\theta^*, \rho^*\) corresponds to \(\max q_i\)

**Step 5:** repeat steps 1 to 4 for sufficiently large number of iterations say \(N\) times.

Let \(Z\) be Bernoulli random variable with \(P(Z = 1) = p\). Here \(z\) is the unobserved missing random variable, \(x\) is the observed data and \((z, x)\) is the complete data. The likelihood function for the complete data is

\[ L(p, \theta | x, z) = \prod_{i=1}^{n} \left( p f_1(x_i | \mu_1, \kappa_1) \right)^{z_i} \left( (1 - p) f_2(x_i | \mu_2, \kappa_2) \right)^{1-z_i} \]

where \(\theta\) is same as defined earlier. The posterior in terms of \(Z\) can be written as

\[ \Pi(\theta, p | x, z) \propto \prod_{i=1}^{n} \left( p \exp(\kappa_1 \cos(x_i - \mu_1)) \right)^{z_i} \left( (1 - p) \exp(\kappa_2 \cos(x_i - \mu_2)) \right)^{1-z_i} \rho^{-1} (1 - \rho)^{\beta - 1} \]

\[ \sum_{k = -\infty}^{\infty} \exp\left(-\left( \mu_1 + 2\pi k - \rho_1 \right)^2 / 2\sigma_1^2 \right) e^{-\kappa_1 / \eta_1} \sum_{k = -\infty}^{\infty} \exp\left(-\left( \mu_2 + 2\pi k - \rho_2 \right)^2 / 2\sigma_2^2 \right) e^{-\kappa_2 / \eta_2} \]

(5.23)
Then the full conditional distribution of \((p, \mu_1, \kappa_1, \mu_2, \kappa_2)\) is given by

\[
\Pi(p | z) \propto p^{n_1 + \sigma - 1} (1 - p)^{n_2 + \beta - 1} \tag{5.24}
\]
\[
\Pi(\mu_1 | z, \kappa_1) \propto \prod_{i : z_i = 1} \exp(\kappa_1 \cos(x_i - \mu_1)) \sum_{k = -\infty}^{\infty} \exp(-(\mu_1 + 2\pi k - \rho_1)^2) / 2\sigma_1^2 \tag{5.25}
\]
\[
\Pi(\kappa_1 | z, \mu_1) \propto \prod_{i : z_i = 1} \exp(\kappa_1 \cos(x_i - \mu_1)) e^{-\kappa_1 / \eta_1} \tag{5.26}
\]
\[
\Pi(\mu_2 | z, \kappa_2) \propto \prod_{i : z_i = 0} \exp(\kappa_2 \cos(x_i - \mu_2)) \sum_{k = -\infty}^{\infty} \exp(-(\mu_2 + 2\pi k - \rho_2)^2) / 2\sigma_2^2 \tag{5.27}
\]
\[
\Pi(\kappa_2 | z, \mu_2) \propto \prod_{i : z_i = 0} \exp(\kappa_2 \cos(x_i - \mu_2)) e^{-\kappa_2 / \eta_2} \tag{5.28}
\]

where, \(n_1\) and \(n_2\) are same as defined earlier.

For generating a sample from these full conditional distributions, we use gibbs sampling for mixture.

**Algorithm-5.8**

Initialize \(p(0), \mu_1(0), \kappa_1(0), \mu_2(0)\) and \(\kappa_2(0)\); Set \(t = 1\).

**Step 1** Generate \(z_i^{(t)} (i = 1, \ldots, n)\) from

\[
P(z_i^{(t)} = 1) \propto p^{(t-1)} \exp(\kappa_1^{(t-1)} \cos(x_i - \mu_1^{(t-1)}))
\]

\[
P(z_i^{(t)} = 0) \propto (1 - p^{(t-1)}) \exp(\kappa_2^{(t-1)} \cos(x_i - \mu_2^{(t-1)}))
\]

**Step 2** Compute \(n_1^{(t)} = \sum_{i=1}^{n} I_{z_i^{(t)} = 1}\) and \(n_2^{(t)} = \sum_{i=1}^{n} I_{z_i^{(t)} = 0}\)

**Step 3** Generate \(p^{(t)} \sim \text{Beta}(n_1^{(t)} + \alpha - 1, n_2^{(t)} + \beta - 1)\)

**Step 4** Generate \(\mu_1^{(t)} \sim \Pi(\mu_1 | z^{(t)}, \kappa_1^{(t-1)})\) using algorithm-2.1

**Step 5** Generate \(\kappa_1^{(t)} \sim \Pi(\kappa_1 | z^{(t)}, \mu_1^{(t)})\) using algorithm-2 of chapter 2.

**Step 6** Generate \(\mu_2^{(t)} \sim \Pi(\mu_2 | z^{(t)}, \kappa_2^{(t-1)})\) using algorithm-1 of chapter 2.

**Step 7** Generate \(\kappa_2^{(t)} \sim \Pi(\kappa_2 | z^{(t)}, \mu_2^{(t)})\) using algorithm-2 of chapter 2.

**Step 8** Repeat step 1 to 6 for \(t = 2, \ldots, N\)

A sample of size 50 is drawn from Von Mises by taking \(\mu_1 = 1.5708, \kappa_1 = 2.0, \mu_2 = 4.7124, \kappa_2 = 5.0\) and \(p = 0.7\). Using this sample a study is carried out in which, samples is generated from posteriors using Algorithm-5.7 and Algorithm-5.8. We have generated 30 multiple chains of size 500 and out of that one observation is selected.
using Algorithm -5.7, while 30 multiple chains of size 1000 are generated using Algorithm-5.8 for each conditional distributions and among 1000 observations, observation having maximum likelihood is selected. Bayes estimates are obtained on the basis of the generated posterior samples, which are $\hat{\mu}_1 = 1.8338$, $\hat{k}_1 = 2.4584$, $\hat{\mu}_2 = 4.6298$, $\hat{k}_2 = 5.0113$, $\hat{\rho} = 0.5670$ (Algorithm-5.7) and $\hat{\mu}_1 = 1.6474$, $\hat{k}_1 = 2.2956$, $\hat{\mu}_2 = 4.5315$, $\hat{k}_2 = 5.8709$, $\hat{\rho} = 0.5961$ (Algorithm-5.8). To compare the performance of the Bayes estimators, we plot predictive densities with original density. Figure 5.3, 5.4, 5.5, 5.6 and 5.7 represents prior and posterior samples of $\mu_1$, $k_1$, $\mu_2$, $k_2$ and $p$ based on Algorithm-5.7. The prior and posterior samples of $\mu_1$, $k_1$, $\mu_2$, and $k_2$ generated using Algorithm-5.8 are presented in Figure 5.8, 5.9, 5.10 and 5.11.

![Fig 5.2. Actual and Predictive pdf](image)

![Fig 5.3 Prior samples of $\mu_1$](image)

(a) 

(b) Posterior samples of $\mu_1$
Fig 5.4 Prior samples of $\kappa_1$

Fig 5.5 Prior samples of $\mu_2$

Fig 5.6 Prior samples of $\kappa_2$
Fig 5.7  Prior samples of \( p \)

Fig 5.8  Prior samples of \( \mu_1 \)

Fig 5.9  Prior samples of \( \kappa_1 \)
A study is carried out to check the performance of both the algorithms. In this study, we generate a sample of size 50 for different parametric ($\mu_1$, $\kappa_1$, $\mu_2$, $\kappa_2$, $p$) values. On the basis of this sample, Bayes estimates are calculated using Algorithm-5.7 and Algorithm-5.8. Results are reported in Table 5.2. It can be easily seen that the Bayes estimates calculated from the sample generated using both of the above algorithms perform quite well. Looking at a computational difficulty about both the algorithms, Algorithm-5.7 takes less move to reach stationary distribution compare to Algorithm-5.8. Another problem using Algorithm-5.8 is that for $p \leq 0.5$, it becomes difficult to simultaneous reallocation of a group of observations to different
components. For $p \geq 0.7$, it has been seen that in each chain, $z$’s allocate most of the observation to single group only.

**Table 5.2 Bayes Estimates**

<table>
<thead>
<tr>
<th>Parameter Values</th>
<th>Algorithm-5.7</th>
<th>Algorithm-5.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=0.25$</td>
<td>0.5019</td>
<td>0.2568</td>
</tr>
<tr>
<td>$\mu_1 = 1.5708$</td>
<td>1.5213</td>
<td>1.5994</td>
</tr>
<tr>
<td>$\kappa_1 = 2.0$</td>
<td>2.1102</td>
<td>1.5055</td>
</tr>
<tr>
<td>$\mu_2 = 4.7124$</td>
<td>4.6939</td>
<td>4.7020</td>
</tr>
<tr>
<td>$\kappa_2 = 5.0$</td>
<td>5.2755</td>
<td>4.1600</td>
</tr>
<tr>
<td>$p=0.5$</td>
<td>0.5943</td>
<td>0.5204</td>
</tr>
<tr>
<td>$\mu_1 = 3.14$</td>
<td>2.7685</td>
<td>3.2879</td>
</tr>
<tr>
<td>$\kappa_1 = 2.0$</td>
<td>1.5191</td>
<td>2.6596</td>
</tr>
<tr>
<td>$\mu_2 = 1.5708$</td>
<td>1.6829</td>
<td>1.7638</td>
</tr>
<tr>
<td>$\kappa_2 = 5.0$</td>
<td>4.6586</td>
<td>5.3398</td>
</tr>
<tr>
<td>$p=0.65$</td>
<td>0.7827</td>
<td>0.5630</td>
</tr>
<tr>
<td>$\mu_1 = 3.14$</td>
<td>3.1214</td>
<td>3.3345</td>
</tr>
<tr>
<td>$\kappa_1 = 0.5$</td>
<td>0.5019</td>
<td>1.2707</td>
</tr>
<tr>
<td>$\mu_2 = 1.5708$</td>
<td>1.6526</td>
<td>1.5762</td>
</tr>
<tr>
<td>$\kappa_2 = 5.0$</td>
<td>3.8479</td>
<td>4.9997</td>
</tr>
<tr>
<td>$p=0.75$</td>
<td>0.6004</td>
<td>0.9099</td>
</tr>
<tr>
<td>$\mu_1 = 3.14$</td>
<td>3.0379</td>
<td>3.2625</td>
</tr>
<tr>
<td>$\kappa_1 = 3.0$</td>
<td>2.8250</td>
<td>4.4153</td>
</tr>
<tr>
<td>$\mu_2 = 5.0$</td>
<td>4.4242</td>
<td>4.9538</td>
</tr>
<tr>
<td>$\kappa_2 = 0.5$</td>
<td>0.4843</td>
<td>0.5985</td>
</tr>
</tbody>
</table>
5.4 Mixture of two Marshall-Olkin Generalized Exponential distribution

In chapter 3, we have done inferences on parameters of MOGE. Here we do Bayesian inference on mixture of Marshall-Olkin Generalized Exponential distributions. Let us consider a two-component mixture of MOGE which has pdf

\[
f(x \mid \alpha_1, \lambda_1, \alpha_2, \lambda_2) = p \frac{\alpha_1 \lambda_1^x e^{\lambda_1 x}}{\left(e^{\lambda_1 x} - (1 - \alpha_1)\right)^2} + (1 - p) \frac{\alpha_2 \lambda_2^x e^{\lambda_2 x}}{\left(e^{\lambda_2 x} - (1 - \alpha_2)\right)^2},
\]

\(x > 0; \alpha_1 > 0, \lambda_1 > 0, \alpha_2 > 0, \lambda_2 > 0\)

Figure 5.12 represents the plot of MOGE distribution for \(\alpha_1 = 5.0, \lambda_1 = 1.5, \alpha_2 = 2.0, \lambda_2 = 0.5\) and \(p=0.6\).

We take prior of \(\alpha_1 \sim \text{exp}(\eta_1), \lambda_1 \sim \text{Gamma}(\kappa_1, \beta_1), \alpha_2 \sim \text{exp}(\eta_2)\) and \(\lambda_2 \sim \text{Gamma}(\kappa_2, \beta_2)\) and \(p \sim \text{Beta}(\tau, \gamma)\). Let \(\theta = (\alpha_1, \lambda_1, \alpha_2, \lambda_2)\), then the posterior distribution can be given as

\[
\Pi(\theta, p \mid x) \propto \prod_{i=1}^{n} \left( p \frac{\alpha_1 \lambda_1^x e^{\lambda_1 x_i}}{\left(e^{\lambda_1 x_i} - (1 - \alpha_1)\right)^2} + (1 - p) \frac{\alpha_2 \lambda_2^x e^{\lambda_2 x_i}}{\left(e^{\lambda_2 x_i} - (1 - \alpha_2)\right)^2} \right) p^{\tau - 1} (1 - p)^{\gamma - 1} e^{-\alpha_1/\eta_1} \\
\lambda_1^{\kappa_1 - 1} e^{-\beta_1 \lambda_1} e^{-\alpha_2/\eta_2} \lambda_2^{\kappa_2 - 1} e^{-\beta_2 \lambda_2}
\]

(5.30)

To generate sample from posterior distribution, we use Perfect sampling approach, based on Breyer and Robert’s (2001) automatic coupling.

Algorithm-5.9

Initialize \((\theta_0^{(0)}, p_0^{(0)})\) and \((\theta_1^{(0)}, p_1^{(0)})\); set \(t=0\).

Step 1: Generate \((\tilde{\theta}, \tilde{p}) \sim q_{ii} (\theta, p \mid \theta_0^{(t)}, p_0^{(t)})\) and \(u \sim \text{Uniform}(0,1)\)

Step 2: Compute

\[r_1 = \frac{f(x \mid \tilde{\theta}, \tilde{p}) \pi(\tilde{\theta}, \tilde{p}) q(\theta_0^{(t)}, p_0^{(t)} \mid \tilde{\theta}, \tilde{p})}{f(x \mid \theta_0^{(t)}, p_0^{(t)}) \pi(\theta_0^{(t)}, p_0^{(t)}) q(\tilde{\theta}, \tilde{p} \mid \theta_0^{(t)}, p_0^{(t)})}
\]

Step 3: If \(u < r_1\) then

\[\theta_0^{(t+1)} = \tilde{\theta}\]
else

\[ \theta_0^{(t+1)} = \theta_0^{(t)} \]

Step 4: Compute

\[ r_2 = \frac{f(x | \tilde{\theta}, \tilde{\rho}) \pi(\tilde{\theta}, \tilde{\rho}) q(\theta_1^{(t)}, \nu_1^{(t)} | \theta_0^{(t)}, \nu_0^{(t)})}{f(x | \theta_1^{(t)}, \nu_1^{(t)}) \pi(\theta_1^{(t)}, \nu_1^{(t)}) q(\tilde{\theta}, \tilde{\rho} | \theta_0^{(t)}, \nu_0^{(t)})} \]

Step 5: If \( u < r_2 \) then

\[ \theta_1^{(t+1)} = \tilde{\theta} \]

else

\[ \theta_1^{(t+1)} = \theta_1^{(t)} \]

Step 6: Repeat steps 1 to 5 until both accept a proposal \( \tilde{\theta} \), i.e. the chains coalescence.

Step 7: Repeat steps 1 to 6 for \( t=1,2,..., N \).

Here \( (\theta_0^{(0)}, \nu_0^{(0)}) \) represent the minimal element, while \( (\theta_1^{(0)}, \nu_1^{(0)}) \) represent maximal element. We run a perfect sampler based on the two chains starting from minimal and maximal element at time \(-T\). We generated the chain \( (\theta_0^{(t)}, \nu_0^{(t)}) \) forwards in time according to a regular Metropolis-Hastings scheme with a proposal \( q \). The chain \( (\theta_1^{(t)}, \nu_1^{(t)}) \) was then coupled to the first chain using step-4 and step-5. If coalescence occurred by time 0, this meant that CFTP was successful. Otherwise, we increase \( T \) by \( 2^k \), where \( k \) represents number of times CFTP was unsuccessful. The procedure was repeated until the chain coalescence.

To implement the above algorithm, a sample of size 50 is drawn from MOGE mixture by taking \( \alpha_1 = 5.0, \lambda_1=1.5, \alpha_2 = 2.0, \lambda_2=0.5 \) and \( p=0.6 \). To generate a sample from posterior, 50 multiple chains of Algorithm-5.9 has been run. In each chain, the value at which \( (\theta_0^{(t)}, \nu_0^{(t)}) \) and \( (\theta_1^{(t)}, \nu_1^{(t)}) \) coalescence is taken as final value.

Bayes estimates under squared error loss function can be given as \( \hat{\alpha}_1=4.8919, \hat{\lambda}_1=1.2229, \hat{\alpha}_2=1.9568, \hat{\lambda}_2=0.2446 \) and \( \hat{p}=0.1827 \). It is to be noted that instead of independent proposal suggested by Breyer and Robert’s (2001) and implemented by casella et. al (2002), here we take dependent proposal, i.e. \( (\tilde{\theta}, \tilde{\rho}) \sim q_{it}(\theta, \rho | \theta_0^{(t)}, \nu_0^{(t)}) \). Figure 5.13, 5.14, 5.15 and 5.16 illustrates the nature of
this coupling. Here we represent a single path of a chain. In most of the cases CFTP was successful at k=5.

Fig 5.12. pdf of mixture of two Marshall-Olkin Generalized Exponential distributions.

Fig 5.13 Sample path of chain $\alpha_{10}^{(t)}$ Sample path of chain $\alpha_{14}^{(t)}$
Figure 5.14 Sample path of chain $\alpha_{2_0}^{(t)}$

(a)

(b)

Sample path of chain $\alpha_{2_1}^{(t)}$

---

Fig 5.15 Sample path of chain $\lambda_{1_0}^{(t)}$

(a)

(b)

Sample path of chain $\lambda_{1_1}^{(t)}$

---

Fig 5.16 Sample path of chain $\lambda_{2_0}^{(t)}$

(a)

(b)

Sample path of chain $\lambda_{2_1}^{(t)}$
A drawback with this approach is obviously that the difference between the two extremes \( (\theta_0^{(t)}, p_0^{(t)}) \) and \( (\theta_1^{(t)}, p_1^{(t)}) \) may prevent coupling for a long run of iterations. Figure 5.18 presents an illustration of this problem for the same data.

5.5 Conclusion

Bayes estimation on mixture of some life time distribution was studied in this chapter. Here we consider three cases (i) Mixture of Weibull with ‘nearly
instantaneous failure’ (ii) Mixture of von Mises distributions (iii) Mixture of Marshall-Olkin Generalized exponential distribution.

In each case we proposed different algorithms to generate a sample from posterior distribution. In section 5.2, Bayes estimation on parameters of Mixture of Weibull with ‘nearly instantaneous failure’ has been done. Here we studied three different algorithms namely, Population Monte Carlo, Mixture version of Metropolis-Hasting and Gibbs sampler. In section 5.3, we proposed two new algorithms to generate a sample from mixture posterior distribution in Mixture of von Mises distributions. We have drawn various figures to explore the algorithms. Bayes estimation of mixture of MOGE has been done in section 5.4, in which automated coupling proposed by Breyer and Roberts’s (2001) was used.