Support Material for: Effective sample size for a mixture prior

Leonardo Egidi[*]

Department of Economics, Business, Mathematics and Statistics, University of Trieste, Via Valerio 4/1, 34127, Trieste, Italy.

A. Proof of Theorem 1

Proof 1. *According to the definitions in Equation (6) in the paper,* $ESS_\pi \leq ESS_{p_1} \Leftrightarrow D_{\pi,+}(\bar{\boldsymbol{\theta}}) \leq D_{p_1,+}(\bar{\boldsymbol{\theta}})$. *Thus, denoting* $H_{i,j}(\bar{\boldsymbol{\theta}}) = \frac{\partial^2 p_i(\bar{\boldsymbol{\theta}})}{\partial \theta_j^2}|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}}$, $H_{i,+}(\bar{\boldsymbol{\theta}}) = \sum_{j=1}^{d} H_{i,j}(\bar{\boldsymbol{\theta}})$, *and noting that* $\frac{\partial p_i(\bar{\boldsymbol{\theta}})}{\partial \theta_j}|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}} = 0 \ \forall i$ *due to assumption (iii), we have:*

[*]Leonardo Egidi
Email address: `legidi@units.it` (Leonardo Egidi)

$$
\begin{aligned}
&D_{\pi,+}(\bar{\boldsymbol{\theta}}) \leq D_{p_1,+}(\bar{\boldsymbol{\theta}}) \Leftrightarrow \\
&\qquad -\sum_{j=1}^{d} \frac{\partial^2 \log \pi(\bar{\boldsymbol{\theta}})}{\partial \theta_j^2} \leq -\sum_{j=1}^{d} \frac{\partial^2 \log p_1(\bar{\boldsymbol{\theta}})}{\partial \theta_j^2} \\
&\qquad \Leftrightarrow -\sum_{j=1}^{d} \frac{\partial}{\partial \theta_j}\left[\frac{1}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}\sum_{i=1}^{k}\psi_i \left(\frac{\partial p_i(\boldsymbol{\theta})}{\partial \theta_j}\right)\Big|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}}\right] \leq -\sum_{j=1}^{d}\frac{\partial}{\partial \theta_j}\left[\frac{\frac{\partial p_1(\boldsymbol{\theta})}{\partial \theta_j}|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}}}{p_1(\bar{\boldsymbol{\theta}})}\right] \\
&\qquad \Leftrightarrow -\sum_{j=1}^{d} \frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}) - (\sum_{i=1}^{k}\psi_i(\frac{\partial p_i(\boldsymbol{\theta})}{\partial \theta_j}|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}}))^2}{(\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}))^2} \\
&\qquad\qquad \leq -\sum_{j=1}^{d}\frac{H_{1,j}(\bar{\boldsymbol{\theta}})p_1(\bar{\boldsymbol{\theta}}) - (\frac{\partial p_1(\boldsymbol{\theta})}{\partial \theta_j}|_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}})^2}{(p_1(\bar{\boldsymbol{\theta}}))^2} \\
&\qquad \Leftrightarrow -\sum_{j=1}^{d}\frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}{(\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}))^2} \leq -\sum_{j=1}^{d}\frac{H_{1,j}(\bar{\boldsymbol{\theta}})p_1(\bar{\boldsymbol{\theta}})}{(p_1(\bar{\boldsymbol{\theta}}))^2} \\
&\qquad \Leftrightarrow \sum_{j=1}^{d}\left[\frac{H_{1,j}(\bar{\boldsymbol{\theta}})p_1(\bar{\boldsymbol{\theta}})}{(p_1(\bar{\boldsymbol{\theta}}))^2} - \frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}{(\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}))^2}\right] \leq 0 \\
&\qquad \Leftrightarrow \sum_{j=1}^{d} H_{1,j}(\bar{\boldsymbol{\theta}})\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}) - p_1(\bar{\boldsymbol{\theta}})\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}) \leq 0 \\
&\qquad \Leftrightarrow \sum_{j=1}^{d}[\psi_1 p_1(\bar{\boldsymbol{\theta}})H_{1,j}(\bar{\boldsymbol{\theta}}) + H_{1,j}(\bar{\boldsymbol{\theta}})\sum_{i=2}^{k}\psi p_i(\bar{\boldsymbol{\theta}}) - \\
&\qquad\qquad \psi_1 p_1(\bar{\boldsymbol{\theta}})H_{1,j}(\bar{\boldsymbol{\theta}}) - p_1(\bar{\boldsymbol{\theta}})\sum_{i=2}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})] \leq 0 \\
&\qquad \Leftrightarrow \sum_{j=1}^{d} H_{1,j}(\bar{\boldsymbol{\theta}})\sum_{i=2}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}) \leq \sum_{j=1}^{d} p_1(\bar{\boldsymbol{\theta}})\sum_{i=2}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}}) \\
&\qquad \Leftrightarrow \sum_{j=1}^{d} H_{1,j}(\bar{\boldsymbol{\theta}}) \leq \sum_{j=1}^{d} p_1(\bar{\boldsymbol{\theta}})\frac{\sum_{i=2}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})}{\sum_{i=2}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})} \\
&\qquad \Leftrightarrow H_{1,+}(\bar{\boldsymbol{\theta}}) \leq \sum_{j=1}^{d} p_1(\bar{\boldsymbol{\theta}})\frac{\sum_{i=2}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})}{\sum_{i=2}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}. \quad \square
\end{aligned}
$$

B. Proof of Theorem 2

Proof 2. *The proof stems directly from the proof of Theorem 1. Since $\boldsymbol{\psi}$ is assigned a hyperprior distribution, we have that the parameter space is of di-*

mension $d+k$ (when $\boldsymbol{\psi}$ is fixed, the parameter space has dimension d). Thus, the observed information for the mixture prior is:

$$
\begin{aligned}
-\sum_{j=1}^{d+k}\frac{\partial^2 \log\pi(\bar{\boldsymbol{\theta}})}{\partial\theta_j^2} &= -\left(\sum_{j=1}^{d}\frac{\partial^2}{\partial\theta_j^2}\log\pi(\bar{\boldsymbol{\theta}}) + \sum_{i=1}^{k}\frac{\partial^2}{\partial\theta_i^2}\log\pi(\bar{\boldsymbol{\theta}})\right)\\
&= -\left(\sum_{j=1}^{d}\frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})} + \sum_{i=1}^{k}\frac{\sum_{i=1}^{k}H_{i,i}(\bar{\boldsymbol{\theta}})p_i(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}\right)\\
&= -\left(\sum_{j=1}^{d}\frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})} + k\frac{\sum_{i=1}^{k}H_{i,i}(\bar{\boldsymbol{\theta}})p_i(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}\right).
\end{aligned}
$$

We can then proceed as for the proof of Theorem 1:

$$
\begin{aligned}
&D_{\pi,+}(\bar{\boldsymbol{\theta}}) \leq D_{p,+}(\bar{\boldsymbol{\theta}}) \Leftrightarrow\\
&\qquad\Leftrightarrow -\left(\sum_{j=1}^{d}\frac{\sum_{i=1}^{k}\psi_i H_{i,j}(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})} + k\frac{\sum_{i=1}^{k}H_{i,i}(\bar{\boldsymbol{\theta}})p_i(\bar{\boldsymbol{\theta}})}{\sum_{i=1}^{k}\psi_i p_i(\bar{\boldsymbol{\theta}})}\right) \leq -\sum_{j=1}^{d}\frac{H_{1,j}(\bar{\boldsymbol{\theta}})}{p_1(\bar{\boldsymbol{\theta}})}.
\end{aligned}
$$

We can then proceed analogously as for the previous proof, and the second addendum in this left term inequality is just summed to the right term inequality of Theorem 1, and the proof is derived. □

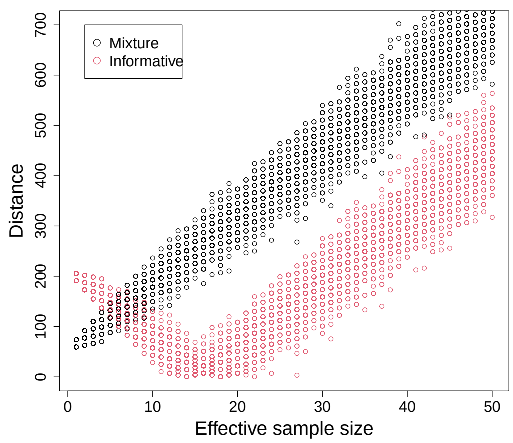
C. Further plots for the multinomial-Dirichlet simulation

Figures C.1 and C.2 report the distances $\delta(n, \bar{\boldsymbol{\theta}}, p_1, q_n)$ and $\delta(n, \bar{\boldsymbol{\theta}}, \pi, q_n)$ for the multinomial-Dirichlet simulation study in Section 3.

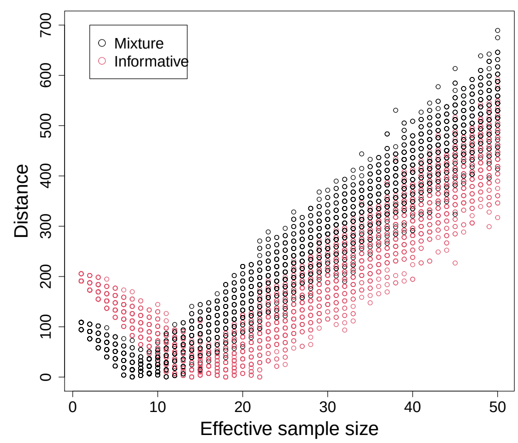
D. Jeffreys prior for an exponential model

Let $\boldsymbol{y} = (y_1, ..., y_n) \underset{iid}{\sim} \mathcal{E}xp(\boldsymbol{y}|\theta)$, with $p(\theta) = \mathcal{G}a(\theta|\alpha, \beta)$. The model likelihood is then given by:

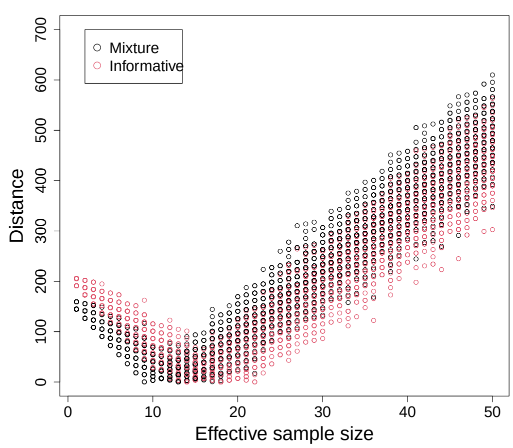
$$
L(\theta; \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i|\theta) = \theta^n \exp(-\theta \sum_i y_i). \tag{1}
$$



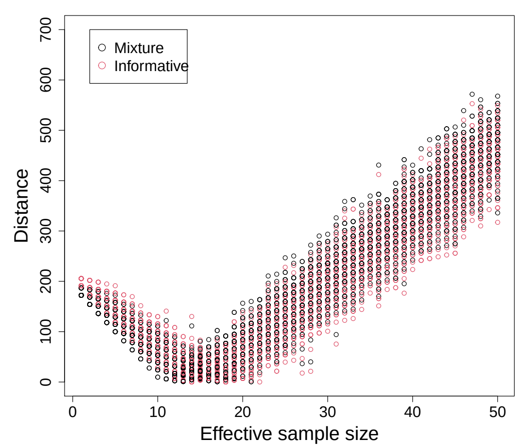
(a) $\boldsymbol{\psi} = (0.1, 0.4, 0.5)$



(b) $\boldsymbol{\psi} = (1/3, 1/3, 1/3)$

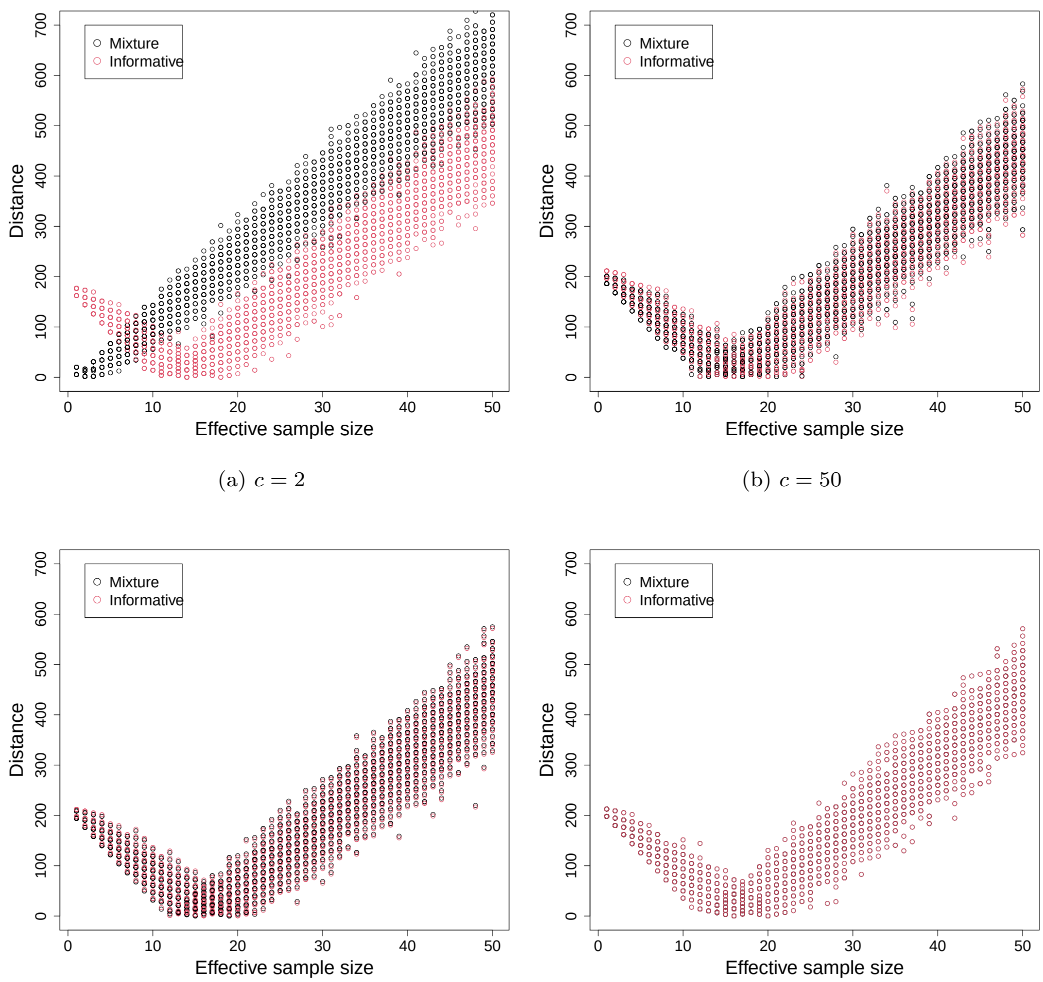


(c) $\boldsymbol{\psi} = (0.5, 0.2, 0.3)$



(d) $\boldsymbol{\psi} = (0.7, 0.1, 0.2)$

Figure C.1: Scenario A: distance $\delta(n, \bar{\boldsymbol{\theta}}, p_1, q_n$ for 100 simulated datasets under the informative $p_1(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha})$ and the mixture prior $\sum_{i=1}^{3} \psi_i p_i(\boldsymbol{\theta})$ under different choices for the mixture weights $\boldsymbol{\psi}$, where $p_2(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}/c)$, $p_3(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}/3c)$, $\boldsymbol{\alpha} = (5, 5, 10)$ and $c = 10$.



(a) $c = 2$

(b) $c = 50$

(c) $c = 100$

(d) $c = 1000$

Figure C.2: Scenario B: distance $\delta(n, \bar{\boldsymbol{\theta}}, p_1, q_n)$ for 100 simulated datasets under the informative $p_1(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha})$ and the mixture prior $\sum_{i=1}^{3} \psi_i p_i(\boldsymbol{\theta})$ under different choices for the factor c, where $p_2(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}/c)$, $p_3(\boldsymbol{\theta}) \sim \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}/3c)$, $\boldsymbol{\alpha} = (5, 5, 10)$ and $\boldsymbol{\psi} = (1/3, 1/3, 1/3)$.

We introduce the Fisher information for the exponential model computed for a single observation, $I_\theta = 1/\theta^2$. Let $q(\theta) = j(\theta)$, where $j(\theta) = \sqrt{I_\theta} = 1/\theta$ is the Jeffreys prior. Thus, the Jeffreys posterior $q_n(\theta|y_1, ..., y_n) = j_n(\theta|y_1, ..., y_n)$ is proportional to $j(\theta)L(\theta; \boldsymbol{y}) = \theta^{-1}\prod_{i=1}^n \theta \exp\{-\theta y_i\} = \theta^{n-1}\exp\left\{-\theta\sum_{i=1}^n y_i\right\}$, the kernel of a Gamma distribution, $\mathcal{G}a(\theta|n, \sum_i y_i)$. Thus:

$$j_n(\theta|\boldsymbol{y}) = \frac{(\sum_i y_i)^n}{\Gamma(n)}\theta^{n-1}\exp\left\{-\theta\sum_{i=1}^n y_i\right\}.$$

Given that $D_{j_n} = -\frac{d^2}{d\theta^2}\left[j_n(\theta|\boldsymbol{y})\right] = \frac{n-1}{\theta^2}$, by using the plug-in estimate $\bar{\theta} = \alpha/\beta$, we may compute the following quantities: 1) the distance between the informative prior p and the Jeffreys posterior j_n; 2) the distance between the Jeffreys prior j and the Jeffreys posterior j_n; 3) the distance between the mixture prior π and the Jeffreys posterior j_n.

E. `R` code for the multinomial-Dirichlet simulation study

For sake of brevity, we report the `R` code for the Scenario A of Section 3 (Scenario B code is pretty identical). The function `DD` allows to compute the second derivative of a probability density function, whereas the main function `Dist` allows to compute the distances between the distinct priors and the non-informative posterior.

```
library(DirichletReg)


# second derivative function
DD <- function(expr, name, order = 1) {
  if(order < 1) stop("'order' must be >= 1")
  if(order == 1) D(expr, name)
  else DD(D(expr, name), name, order - 1)
}


ess_mix = ess_p = c()
```

```
## scenario A: varying weights, fixed c

par(mar=c(5,5,2,1))

# possible mixture weights values
aa <- c(0.1, 1/3, 0.5, 0.7)
bb <- c(0.4, 1/3,0.2, 0.1)
cc <- c(0.5, 1/3, 0.3, 0.2)

# 4 simulation for each of the weights' scenarios
for (j in 1:4){
a <- aa[j]
b <- bb[j]
c <- cc[j]
a1 <- 5   # Dirichlet hyperpar. alpha1
a2 <- 5   # Dirichlet hyperpar. alpha2
a3 <- 10  # Dirichlet hyperpar. alpha3
k <- 3
weights <- c(a1, a2, a3)
w <- 10   # scaling factor
weights_q <- weights/w
theta_hat <- (weights-1)/(sum(weights)-k)  # mode

# 100 simulations, with n=1,2,...,50.
# For each simulation, I draw a multinomial sample Y
for (i in 1:100){

# main function to compute distances between each prior and the posterior
Dist <- function(x){
  Y <- rmultinom(x, 1, prob=c(1/3, 1/3, 1/3)) # multinomial likelihood
  S <- apply(Y,1,sum)
```

```
evaluate_d_prior <- function(theta, weights){ # mixture prior derivative
  theta1 <- theta[1]
  theta2 <- theta[2]
  theta3 <- theta[3]
  weights1 <- weights[1]
  weights2 <- weights[2]
  weights3 <- weights[3]
  D_prior_p1 <- -eval(DD(  expression(log( a*(gamma(weights1+weights2+weights3)/
                           (gamma(weights1)*gamma(weights2)*gamma(weights3)))*
                           (theta1^(weights1-1))*
                            (theta2^(weights2-1))*
                            (theta3^(weights3-1))+
                   b* (gamma(weights1/w+weights2/w+weights3/w)/
                           (gamma(weights1/w)*gamma(weights2/w)*gamma(weights3/w)))*
                           (theta1^(weights1/w-1))*
                            (theta2^(weights2/w-1))*
                            (theta3^(weights3/w-1))+
                   c* (gamma(weights1/(3*w)+weights2/(3*w)+weights3/(3*w)))/
                        (gamma(weights1/(3*w))*gamma(weights2/(3*w))*gamma(weights3/(3*w)))*
                           (theta1^(weights1/(3*w)-1))*
                            (theta2^(weights2/(3*w)-1))*
                            (theta3^(weights3/(3*w)-1)))), "theta1",2))
  D_prior_p2 <- -eval(DD(  expression(log( a*(gamma(weights1+weights2+weights3)/
   (gamma(weights1)*gamma(weights2)*gamma(weights3)))*
                           (theta1^(weights1-1))*
                           (theta2^(weights2-1))*
                            (theta3^(weights3-1))+
                   b* (gamma(weights1/w+weights2/w+weights3/w)/
                      (gamma(weights1/w)*gamma(weights2/w)*gamma(weights3/w)))*
                           (theta1^(weights1/w-1))*
                           (theta2^(weights2/w-1))*
                           (theta3^(weights3/w-1))+
                   c* (gamma(weights1/(3*w)+weights2/(3*w)+weights3/(3*w)))/
```

```
                      (gamma(weights1/(3*w))*gamma(weights2/(3*w))*gamma(weights3/(3*w)))*
                           (theta1^(weights1/(3*w)-1))*
                            (theta2^(weights2/(3*w)-1))*
                            (theta3^(weights3/(3*w)-1)))), "theta2",2))
  D_prior_p3 <- -eval(DD(  expression(log( a*(gamma(weights1+weights2+weights3)/
          (gamma(weights1)*gamma(weights2)*gamma(weights3)))*
          (theta1^(weights1-1))*
          (theta2^(weights2-1))*
          (theta3^(weights3-1))+
             b* (gamma(weights1/w+weights2/w+weights3/w)/
          (gamma(weights1/w)*gamma(weights2/w)*gamma(weights3/w)))*
          (theta1^(weights1/w-1))*
          (theta2^(weights2/w-1))*
          (theta3^(weights3/w-1))+
            c* (gamma(weights1/(3*w)+weights2/(3*w)+weights3/(3*w)))/
          (gamma(weights1/(3*w))*gamma(weights2/(3*w))*gamma(weights3/(3*w)))*
          (theta1^(weights1/(3*w)-1))*
          (theta2^(weights2/(3*w)-1))*
          (theta3^(weights3/(3*w)-1)) )), "theta3",2))

  D_prior_p <- sum(D_prior_p1, D_prior_p2, D_prior_p3)
  return(D_prior_p)
}

evaluate_d_prior2 <- function(theta, weights){ # informative prior derivative
  theta1 <- theta[1]
  theta2 <- theta[2]
  theta3 <- theta[3]
  weights1 <- weights[1]
  weights2 <- weights[2]
  weights3 <- weights[3]
  D_prior_p1 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
```

```
                                            (theta2^(weights2-1))*
                                            (theta3^(weights3-1)))), "theta1",2))
  D_prior_p2 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
                                            (theta2^(weights2-1))*
                                            (theta3^(weights3-1)))), "theta2",2))
  D_prior_p3 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
                                            (theta2^(weights2-1))*
                                            (theta3^(weights3-1)))), "theta3",2))

  D_prior_p <- sum(D_prior_p1, D_prior_p2, D_prior_p3)
  return(D_prior_p)
}

evaluate_d_posterior <- function(theta, weights, S){
                                # noninformative posterior derivative
  theta1 <- theta[1]
  theta2 <- theta[2]
  theta3 <- theta[3]
  weights1 <- weights[1]+S[1]
  weights2 <- weights[2]+S[2]
  weights3 <- weights[3]+S[3]
  D_post_p1 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
                                            (theta2^(weights2-1))*
                                            (theta3^(weights3-1)))), "theta1",2))
  D_post_p2 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
                                            (theta2^(weights2-1))*
                                            (theta3^(weights3-1)))), "theta2",2))
  D_post_p3 <- -eval(DD(  expression(log( (gamma(weights1+weights2+weights3)/
  (gamma(weights1)*gamma(weights2)*gamma(weights3)))*(theta1^(weights1-1))*
```

```
                                              (theta2^(weights2-1))*
                                              (theta3^(weights3-1)))), "theta3",2))

    D_post_p <- sum(D_post_p1, D_post_p2, D_post_p3)
    return(D_post_p)
  }

  # distances
  D1 <- abs(evaluate_d_prior(theta_hat, weights)-
            evaluate_d_posterior(theta_hat, weights_q/3, S))
  D2 <- abs(evaluate_d_prior2(theta_hat, weights)-
             evaluate_d_posterior(theta_hat, weights_q/3, S))
  return(c(D1,D2))
}

Dist <- Vectorize(Dist)
result<-Dist(seq(1:50))
par(mfrow=c(1,1))

if (i==1){
plot(1:50, result[1,], ylim=c(0,700),
     xlab = "Effective sample size", ylab = "Distance",
     cex.lab=2, cex.axis =1.6)
  points(1:50, result[2,], col=2)
  legend(2, 700, c("Mixture", "Informative"),col=c(1,2),
         pch=1, cex=1.6)
}else{
  points(1:50, result[1,])
  points(1:50, result[2,], col=2)
}

# ess computation
ess_mix[i] <- which.min(result[1,])
```

```
ess_p[i] <- which.min(result[2,])

}
dev.copy2pdf(file=paste("distance", j, ".pdf", sep=""), width=9,
             height=8)


pdf(file=paste("ESS", j, ".pdf", sep=""), width = 9, height = 8)
par(xaxt="n", mar=c(5,5,2,1))
boxplot(ess_p, ess_mix, ylab = "ESS", cex.lab =2, cex.axis =1.6)
par(xaxt="s")
axis(1,at =c("1", "2"), labels= c("Informative", "Mixture"), cex.lab=2, cex.axis=2)
dev.off()
}
```

F. R code for the phase I trial

We report the R code for the logistic regression for phase I trial, Section 4. For technical details about the implementation, see Morita (2008), Section 6, example 7.

```
library(distrEx)

# second derivative function
DD <- function(expr, name, order = 1) {
  if(order < 1) stop("'order' must be >= 1")
  if(order == 1) D(expr, name)
  else DD(D(expr, name), name, order - 1)
}

# initialization
m=m_mu=m_Beta=c()
  m_mix=m_mu_mix=m_Beta_mix=m_Jeffreys=m_mu_Jeffreys=m_Beta_Jeffreys=matrix(NA, 5,5)
```

```
  tab<-list()

# scaling factor values
KK=c(5, 100, 1000,10000)


# simulation
for (jj in 1:4){
K <- KK[jj]
for (k in 1:5){
  sigma2_Beta=c(0.5^2,1,2^2,3^2,5^2)
  sigma2_mu=c(0.5^2,1,2^2,3^2,5^2)
  mu_Beta=2.3980
  mu_mu=-0.1313
  X=c(100,200,300,400,500,600)

  X_stand<-c()

  for (h in 1:6){
    X_stand[h]<-log(X[h])-sum(log(X))/6
  }

Dist_p1=1/sigma2_mu[k]
Dist_p2=1/sigma2_Beta[k]

M=100
T=10000
X_rep<-matrix(NA, M, T)

for (j in 1:M){
  X_rep[j,]<-sample(X_stand, T, prob=rep(1/6,6), replace=T )
}
```

```
p<-matrix(NA, M, T)

for (j in 1:M){
  for (t in 1:T){
  p[j,t]=(exp(mu_mu+mu_Beta*X_rep[j,t]))/(1+exp(mu_mu+mu_Beta*X_rep[j,t])  )
}}

Dist_q1<-matrix(NA, M, T)
for (j in 1:M){
  for (t in 1:T){
Dist_q1[j,t]<-sum(p[1:j,t]*(1-p[1:j,t]))
}}

Dist_q2<-matrix(NA, M, T)

for (j in 1:M){
  for (t in 1:T){
  Dist_q2[j,t]<-sum((X_rep[1:j,t]^(2))*p[1:j,t]*(1-p[1:j,t]))
  }}



# Monte Carlo simulation
theta_rep<-matrix(NA, T,2)  #vector of parameters mu and Beta
y_rep<-matrix(NA, M, T)
p_rep<-matrix(NA, M, T)
Dist_q1_rep<-matrix(NA, M, T)
Dist_q2_rep<-matrix(NA, M, T)

for (t in 1:T){
  theta_rep[t,1]<-rnorm(1, mu_mu, sqrt(sigma2_mu[k]))
  theta_rep[t,2]<-rnorm(1, mu_Beta, sqrt(sigma2_Beta[k]))
    for (j in 1:M){
```

```
        p_rep[j,t]<-exp(theta_rep[t,1]+theta_rep[t,2]*X_rep[j,t])/
                    (1+exp(theta_rep[t,1]+theta_rep[t,2]*X_rep[j,t]))
        y_rep[j,t]<-rbinom(1,1,p_rep[j,t])
        Dist_q1_rep[j,t]<-sum(p_rep[1:j,t]*(1-p_rep[1:j,t]))
        Dist_q2_rep[j,t]<-sum((X_rep[1:j,t]^(2))*p_rep[1:j,t]*(1-p_rep[1:j,t]))
}
}

Dist_MC_mu<-c()
Dist_MC_Beta<-c()
Dist_q<-c()

for (j in 1:M){
  Dist_MC_mu[j]<-(1/T)*sum(Dist_q1_rep[j,])
  Dist_MC_Beta[j]<-(1/T)*sum(Dist_q2_rep[j,])
  Dist_q[j]<-Dist_MC_mu[j]+Dist_MC_Beta[j]
}

min1<-c()
min2<-c()

for (t in 1:T){
    # ess computation
min1[t]<-which.min(abs(Dist_p1-Dist_q1[,t]))
min2[t]<-which.min(abs(Dist_p2-Dist_q2[,t]))
}

m_mu[k]<-mean(min1)
m_Beta[k]<-mean(min2)
m_funct<-function(x){
  return(abs(Dist_p1+Dist_p2-Dist_q))
}
m[k]<-which.min(abs(Dist_p1+Dist_p2-Dist_q))     #valori interi
```

```
for (p in 1:3){
  sigma2_Beta=c(0.5^2,1,2^2,3^2,5^2)
  sigma2_mu=c(0.5^2,1,2^2,3^2,5^2)
H_vec<-c(0.2,0.5,0.8)
H<-H_vec[p]


theta_hat<-mu_mu
sigma2_Beta<-sigma2_Beta[k]
sigma2_mu<-sigma2_mu[k]

# ess for the mixture prior
Dist_p3<- -eval(DD(  expression(log( H*((1/((sqrt(2*pi))*sqrt(K*sigma2_mu)) )*
                     exp(-(0.5/(K*sigma2_mu))*(theta_hat-mu_mu)^2))+
                     (1-H)*((1/((sqrt(2*pi))*sigma2_mu) )*
                     exp(-(0.5/(sigma2_mu))*(theta_hat-mu_mu)^2 )))), "theta_hat",2))
Dist_p3_Jeffreys<- -eval(DD(  expression(log( H+
                          (1-H)*((1/((sqrt(2*pi))*sigma2_mu) )*
                    exp(-(0.5/(sigma2_mu))*(theta_hat-mu_mu)^2 )))), "theta_hat",2))

theta_hat<-mu_Beta

Dist_p4<--eval(DD(  expression(log( H*((1/((sqrt(2*pi))*sqrt(K*sigma2_Beta)) )*
                        exp(-(0.5/(K*sigma2_Beta))*(theta_hat-mu_Beta)^2))+
                        (1-H)*((1/((sqrt(2*pi))*sigma2_Beta) )*
                  exp(-(0.5/(sigma2_Beta))*(theta_hat-mu_Beta)^2 )))), "theta_hat",2))
Dist_p4_Jeffreys<--eval(DD(  expression(log( H+
                  (1-H)*((1/((sqrt(2*pi))*sigma2_Beta) )*
                  exp(-(0.5/(sigma2_Beta))*(theta_hat-mu_Beta)^2 )))), "theta_hat",2))

min3<-c()
```

```
min4<-c()
min3_Jeffreys<-c()
min4_Jeffreys<-c()

for (t in 1:T){

  min3[t]<-which.min(abs(Dist_p3-Dist_q1[,t]))
  min3_Jeffreys[t]<-which.min(abs(Dist_p3_Jeffreys-Dist_q1[,t]))
  min4[t]<-which.min(abs(Dist_p4-Dist_q2[,t]))
  min4_Jeffreys[t]<-which.min(abs(Dist_p4_Jeffreys-Dist_q2[,t]))
}

m_mu_mix[k,p]<-mean(min3)
m_Beta_mix[k,p]<-mean(min4)
m_mix[k,p]<-which.min(abs(Dist_p3+Dist_p4-Dist_q))

m_mu_Jeffreys[k,p]<-mean(min3_Jeffreys)
m_Beta_Jeffreys[k,p]<-mean(min4_Jeffreys)
m_Jeffreys[k,p]<-which.min(abs(Dist_p3_Jeffreys+Dist_p4_Jeffreys-Dist_q))

}

  vec<-list()
  for (k in 1:5){

    vec[[k]]<-rbind(c(m[k], m_mu[k], m_Beta[k],  m_mix[k,1], m_mu_mix[k,1], m_Beta_mix[k,1],
                      m_mix[k,2], m_mu_mix[k,2], m_Beta_mix[k,2],
                      m_mix[k,3], m_mu_mix[k,3], m_Beta_mix[k,3],
                      m_mix[k,4], m_mu_mix[k,4], m_Beta_mix[k,4],
                      m_mix[k,5], m_mu_mix[k,5], m_Beta_mix[k,5]
                      ))
 } }
```

```
# final tab
tab[[jj]]<-rbind( as.double(vec[[1]]), as.double(vec[[2]]), as.double(vec[[3]]),
            as.double(vec[[4]]), as.double(vec[[5]]))
}

vec_Jeffreys<-list()
for (k in 1:5){

  vec_Jeffreys[[k]]<-rbind(c( m_Jeffreys[k,1], m_mu_Jeffreys[k,1], m_Beta_Jeffreys[k,1],
                    m_Jeffreys[k,2], m_mu_Jeffreys[k,2], m_Beta_Jeffreys[k,2],
                    m_Jeffreys[k,3], m_mu_Jeffreys[k,3], m_Beta_Jeffreys[k,3]))
}

tab2<-rbind( as.double(vec_Jeffreys[[1]]), as.double(vec_Jeffreys[[2]]), as.double(vec_Jeffreys[[3]]),
            as.double(vec_Jeffreys[[4]]),  as.double(vec_Jeffreys[[5]]) )
```