Bayesian Statistics

Exercises 4

1. Consider the following setting, which is common in genomics. It is desired to identify which genes show different activity levels (often called "expression" levels) in two "conditions" - say, in heart vs liver. Many genes might show very similar activity levels in heart and liver, whereas others might be much more active in one or the other.

   To assess this we make measurements of the activity of each gene $j = 1, \ldots, m$ in many individuals $i = 1, \ldots, n$, in each condition $k = 0, 1$. Let Y_{ijk} denote the measurement on individual i, gene j, treatment k, and let D_{ij} denote the difference between the measurements in the two treatments: $D_{ij} := Y_{ij0} - Y_{ij1}$. We will assume that D is sufficient for all our inferences, and so you can forget about Y now and work only with D: I just wanted you to understand where D might come from in principle.

   We will assume a model for D:

   $$D_{ij}|\beta, \sigma \sim N(\beta_j, \sigma_j^2) \tag{1}$$

   where $\beta = (\beta_1, \ldots, \beta_m)$ is a vector of "effect"s, where β_j is the effect at gene j, and $\sigma = (\sigma_1, \ldots, \sigma_m)$ is a vector of standard deviation parameters. For each gene j we wish to test the null $H_j : \beta_j = 0$ (that is, that there is no treatment effect). For simplicity you can assume that $\sigma_j = 1$ is known for all j. You can also assume that $n = 10$ and $m = 1,000$.

   Assume that the true effects β_j are independent, and identically distributed, from a mixture distribution f_β

   $$f_\beta = \pi_0 \delta_0(\cdot) + (1 - \pi_0) N(\cdot; 0, \sigma_b^2). \tag{2}$$

   where δ_0 denotes a point mass on zero. That is, with probability π_0, $\beta_j = 0$, and otherwise (with probability $1 - \pi_0$) $\beta_j \sim N(0, \sigma_b^2)$.

   Consider implementing an Empirical Bayes approach to this problem. To do so, given data D we will need two steps:

A Estimate the hyper parameters π_0, σ_b in (2) by maximum likelihood. Call the estimates $\hat{\pi}_0, \hat{\sigma}_b$.

B Compute the posterior distribution $p(\beta_j|D, \hat{\pi}_0, \hat{\sigma}_b)$ for each j.

This question takes you through these two steps.

i) Define $\bar{D}_j = (1/n)\sum_i D_{ij}$. Show that the vector $\bar{D} := (\bar{D}_1, \ldots, \bar{D}_m)$ is sufficient for β. That is, $p(D|\beta) \propto p(\bar{D}|\beta)$ where the constant of proportionality does not depend on β. [This means that, as far as inference for β is concerned, the likelihood $p(D|\beta)$ is equivalent to the likelihood $p(\bar{D}|\beta)$, so from now on you can treat $\bar{D}$ as your data instead of D.]

ii) Derive an expression for the log-likelihood $l(\pi_0, \sigma_b) := \log(p(\bar{D}|\pi_0, \sigma_b))$. [Hint: note that the $\bar{D}_j$ are independent given π_0, σ_b]

iii) Write an R function to compute the log-likelihood $l(\pi_0, \sigma_b)$, or alternatively $l(\theta_1, \theta_2)$ where $\theta_1 = \log(\pi_0/(1-\pi_0)), \theta_2 = \log(\sigma_b)$. [The motivation for this reparameterization is that θ_1, θ_2 can take any value on the real line.] Try using the R function optimize (or another method if you prefer) to maximize the likelihood over π_0, σ_b (or θ_1, θ_2). [You may or may not find that this works... it is a somewhat tricky numerical problem. The reparameterization may help. Alternatively if you know about the EM algorithm you can try that.]

iv) Derive the posterior distribution $\beta_j|D, \pi_0, \sigma_b$. Hint: this posterior should be a mixture of a point mass at zero and a normal distribution. It may help to first derive $p(\beta_j = 0|D, \pi_0, \sigma_b)$, and then $p(\beta_j|D, \pi_0, \sigma_b, \beta_j \neq 0)$.

v) Implement a method that computes $p(\beta_j = 0|D, \hat{\pi}_0, \hat{\sigma}_b)$. Implement another method that takes these probabilities and rejects those tests j for which this probability is $< \alpha$.

vi) Perform a simulation study to illustrate your method and test how it performs. Here you are expected to decide for yourself how to measure performance; but the simulation study should involve simulating many data sets D and not just one! (If you are unable to get the optimization for π, σ_b to work then you can "cheat" in this step and use the true value of π, σ_b.)

2. Let $x_1, \ldots, x_n$ be independent and identically distributed $N(\mu, 1)$. Assume a prior for μ that is $N(0, 3^2)$. Explain what the following R code does - and, particularly, explain i) the vectorization being performed in the function `loglik`, and ii) the problems caused when computing the weights `w1`, and how the program solves this problem for computing `w2`. Compare the numerical results with analytical calculations for the same data x.

```
set.seed(111)
m=10000
n=1000
x = rnorm(n,0,1)

musamp = rnorm(m,0,3)

loglik = function(x,musamp){
  n=length(x)
  m = length(musamp)
  xx = rep(x,rep(m,n))
  ll.matrix=matrix(dnorm(xx,musamp,1,log=TRUE),nrow=m)
  rowSums(ll.matrix)
}

logl=loglik(x,musamp)
normalize=function(x){x/sum(x)}
w1 = normalize(exp(logl))
w2 = normalize(exp(logl-max(logl)))

sum(w1[musamp<0])
sum(w2[musamp<0])
sum(w2*musamp)
sum(w2*musamp^2) - sum(w2*musamp)^2
```

3. Let $x_1, \ldots, x_n$ be independent and identically distributed observations from the mixture density

$$f_X(x; \pi_0, \mu_0, \mu_1, \tau_0, \tau_1) = \pi_0 N(x; \mu_0, 1/\tau_0) + (1 - \pi_0) N(x; \mu_1, 1/\tau_1) \quad (3)$$

where $N(\cdot; \mu, 1/\tau)$ denotes the density of a normal distribution with mean μ and variance $1/\tau$.

Assume that the true values of the parameters are $\pi_0 = 0.2, \mu_0 = 0, \mu_1 = 4, \tau_0 = \tau_1 = 1$.

(a) The mixture model (3) can also be written using the following "latent variable representation": $z_i \sim \text{Bernoulli}(1 - \pi_0)$; $x_i | z_i = k \sim N(\mu_k, 1/\tau_k)$. This is called the latent variable representation because z_i is unobserved (latent) so the probability density of the x_i requires you to marginalize $p(x_i, z_i)$ over z_i. Show how marginalizing the joint distribution over z_i yields the density (3) for x_i.

(b) Simulate 1000 observations from this mixture distribution. (Make sure to set a seed so that your results are reproducible.) Plot a histogram of the results. [Hint: simulating from a mixture can be done by exploiting the latent variable representation.]

(c) Write a function to compute the log likelihood for a vector x at any given $\pi_0, \mu_0, \mu_1, \tau_0, \tau_1$. (You might also like to vectorize this function to allow the log-likelihood to be efficiently computed for any vectors of these parameters, analogous to the code in Q1)

(d) Assume now that you know the true values of the means and precisions $(\mu_0, \mu_1, \tau_0, \tau_1)$ and wish to estimate π_0. Assume that your prior is

$$\pi_0 \sim \text{Beta}(0.5, 0.5) \tag{4}$$

i) Write a function that uses important sampling, with the prior distribution as your importance sampling distribution, to obtain an approximation to the posterior distribution of π_0. Apply this to your simulated data set (created in part b) to obtain an approximate posterior mean and an approximate 90% posterior CI for π_0. (Unless you are unlucky the posterior CI should cover the true value of π_0! If not, try again with a new seed, to check whether there might be a problem with your implementation.)

ii) Use your log-likelihood function to compute the log-likelihood on a grid of values of π_0 from 0 to 1. Use this grid-based approach to obtain a discrete approximation to the posterior distribution for π_0. Again, obtain a posterior mean for π_0 and a 90% posterior CI.

iii) Use your answer to the previous part to develop an improved importance sampling function. (Note that, to satisfy the requirements that the support of q contains the support of p, the importance sampling function must be continuous, so you have to convert your discrete approximation to a continuous approximation.) Use this improved IS function to obtain another estimate for the posterior mean and an approximate 90% posterior CI for π_0. Comment on how the variance of the importance weights differs between this IS function and using the prior as an IS function.