Stat 591 – Homework 05 *Solutions*

1. (a) First, let X be a random variable with distribution function F, and assume that F is strictly increasing and continuous. Define $U = F(X)$; the claim is that $U \sim \mathsf{Unif}(0,1)$. To prove this,

$$\mathsf{P}(U \le u) = \mathsf{P}(F(X) \le u) = \mathsf{P}(X \le F^{-1}(U)) = F(F^{-1}(u)) = u;$$

since this holds for all $u \in (0,1)$, the claim is true. Similarly, if $U \sim \mathsf{Unif}(0,1)$ and F is a distribution function as above, then we claim that $X = F^{-1}(U)$ has distribution F. To prove this,

$$\mathsf{P}(X \le x) = \mathsf{P}(F^{-1}(U) \le x) = \mathsf{P}(U \le F(x)) = F(x);$$

since this holds for all x, the claim is true. As an example, suppose the goal is to simulate an exponential random variable with scale parameter m (the mean). For this distribution, the distribution function is $F(x) = 1 - e^{-x/m}$ and the inverse is $F^{-1}(u) = -m\log(1-u)$. So, according to the result above, if $U \sim \mathsf{Unif}(0,1)$, then $X = -m\log U$ is exponential with scale m; here we used the fact that if $U \sim \mathsf{Unif}(0,1)$ then $1 - U \sim \mathsf{Unif}(0,1)$ too.

(b) Let $U_1, U_2 \overset{\text{iid}}{\sim} \mathsf{Unif}(0,1)$ and define

$$W = -2\log U_1 \quad \text{and} \quad V = 2\pi U_2,$$

Write $R^2 = W$ and set

$$X = R\cos V \quad \text{and} \quad Y = R\sin V.$$

The claim is that X and Y are independent standard normal. We have defined a transformation $(u_1, u_2) \mapsto (x, y)$; define the inverse transformation as

$$u_1 = e^{-(1/2)(x^2+y^2)} \quad \text{and} \quad u_2 = (1/2\pi)\arctan(y/x).$$

We need the (absolute) determinant of the partial derivative matrix

$$\begin{pmatrix} -xe^{-(1/2)(x^2+y^2)} & -ye^{-(1/2)(x^2+y^2)} \\ -\frac{1}{2\pi}\frac{y}{x^2+y^2} & \frac{1}{2\pi}\frac{x}{x^2+y^2} \end{pmatrix},$$

which is given by

$$\left| -\frac{1}{2\pi}\frac{x^2}{x^2+y^2}e^{-(1/2)(x^2+y^2)} - \frac{1}{2\pi}\frac{y^2}{x^2+y^2}e^{-(1/2)(x^2+y^2)} \right| = \frac{1}{2\pi}e^{-(1/2)(x^2+y^2)}.$$

Finally, since the densities of U_1, U_2 are constant, the joint density of (X, Y) is given by the Jacobian term above, which is exactly the joint density of two independent standard normals. That's it.

(c) One can use the inverse CDF method to get a standard normal sample, i.e., $X = \Phi^{-1}(U)$, where $U \sim \mathsf{Unif}(0,1)$ and Φ is the $\mathsf{N}(0,1)$ CDF. The Box–Muller method will also produce $X \sim \mathsf{N}(0,1)$. The point is that evaluating Φ^{-1} requires some special tricks (behind the scenes in R) which are not so simple, whereas Box–Muller only requires uniform samples and simple transformations. (Of course, one might argue that the trig functions are not simple.)

2. (a) Let X denote the accept–reject output. We want to show that, for any event A in the X-space, $\mathsf{P}(X \in A) = \int_A f(x)\,dx$. Following the hint:

$$\mathsf{P}(X \in A) = \mathsf{P}\Big(Y \in A \,\Big|\, U \le \frac{f(Y)}{Mg(Y)}\Big) = \frac{\mathsf{P}(Y \in A,\, U \le \frac{f(Y)}{Mg(Y)})}{\mathsf{P}(U \le \frac{f(Y)}{Mg(Y)})}.$$

The denominator is

$$\int \int_0^{f(y)/Mg(y)} du\, g(y)\, dy = \int \frac{f(y)}{Mg(y)} g(y)\, dy = \frac{1}{M}.$$

Similarly, the numerator is

$$\int_A \int_0^{f(y)/Mg(y)} du\, g(y)\, dy = \int_A \frac{f(y)}{Mg(y)} g(y)\, dy = \frac{1}{M} \int_A f(y)\, dy.$$

The ratio is $\int_A f(y)\, dy$, proving the claim.

(b) The acceptance probability is the denominator from the previous calculation, which equals M^{-1}. So, the only way that the acceptance probability equals 1—the most efficient algorithm possible—is if $M = 1$. The only way this can occur is if $f(x) \le g(x)$ for all x, but since both are PDFs, the integral constraint means they must be equal. So, the only way to guarantee that samples are accepted is to sample from the target distribution directly. This cannot be achieved since, at the start, we said sampling from the target f was not possible.

(c) To implement the accept–reject method to simulate from a gamma target distribution with non-integer shape parameter θ, we shall follow the hint and take the proposal to be a gamma distribution with shape $[\theta]$, the integer part, and scale $b = \theta/[\theta]$; the motivation behind introducing a scale parameter is to match the mean of the target distribution. That is, we have

$$f(y) = \text{dgamma}(y, \theta, 1) = \frac{1}{\Gamma(\theta)} y^{\theta-1} e^{-y}$$
$$g(y) = \text{dgamma}(y, [\theta], b) = \frac{1}{\Gamma([\theta])} \frac{1}{b^{[\theta]}} e^{-y/b}.$$

To implement the method, we need to find a constant M such that $f(y) \le Mg(y)$ for all y. That is,

$$M \ge \frac{f(y)}{g(y)} = \frac{\Gamma([\theta])}{\Gamma(\theta)} \Big(\frac{\theta}{[\theta]}\Big)^{[\theta]} \big(y e^{-y/\theta}\big)^{\theta-[\theta]} \quad \forall\, y.$$

The most efficient bound corresponds to maximizing the right-hand side with respect to y. Since there's only one simple term involving y, this is easy to do, and it gives

$$M = \frac{\Gamma([\theta])}{\Gamma(\theta)} \Big(\frac{\theta}{[\theta]}\Big)^{[\theta]} \big(\theta e^{-1}\big)^{\theta-[\theta]}$$

```
rgamma.ar <- function(n, shape, scale=1) {

  s <- shape
  s.int <- floor(s)
  b <- s / s.int
  M <- gamma(s.int) / gamma(s) * b**s.int * (s * exp(-1))**(s - s.int)
  f <- function(y) dgamma(y, shape=s)
  Mg <- function(y) M * dgamma(y, shape=s.int, rate=1 / b)
  acpt <- 0
  total <- 0
  X <- numeric(n)
  while(acpt < n) {

    total <- total + 1
    Y <- sum(-b * log(runif(s.int)))
    if(runif(1) <= f(Y) / Mg(Y)) {

      acpt <- acpt + 1
      X[acpt] <- Y

    }

  }
  return(list(X=scale * X, rate.true=1 / M, rate.obs=acpt / total))

}
o <- rgamma.ar(1000, shape=5.5)
print(o[-1])
ylim <- c(0, 1.05 * dgamma(4.5, shape=5.5))
hist(o$X, freq=FALSE, col="gray", border="white", xlab="x", ylim=ylim)
curve(dgamma(x, shape=5.5), add=TRUE)
```

Figure 1: R code for the gamma accept–reject method.

as the most efficient bound. R code to implement this method is given in Figure 1. In the case $\theta = 5.5$, we get $M = 1.0504$, so that the theoretical acceptance probability is 0.952. A sample of size $n = 1000$ is produced based on my algorithm, and my empirical acceptance probability is 0.946; yours will surely be a bit different. A histogram of the 1000 samples is shown in Figure 2 and it fits the target density (overlaid) very well, as we expect.

3. Let $X_1, \ldots, X_n$ be a sample from a location-shifted Student-t distribution, with (known) degrees of freedom ν and (unknown) location θ. With a flat (invariant) prior for θ, the goal is to use importance sampling to approximate the posterior mean (also, in this case, the Pitman estimator, which is the best equivariant estimator in a decision-theoretic sense).

   The likelihood function is

$$L(\theta) \propto \prod_{i=1}^{n} \Big(1 + \frac{(X_i - \theta)^2}{\nu}\Big)^{-(\nu+1)/2},$$

   and, for a flat prior, the posterior is proportional to this. Since n is relatively large,

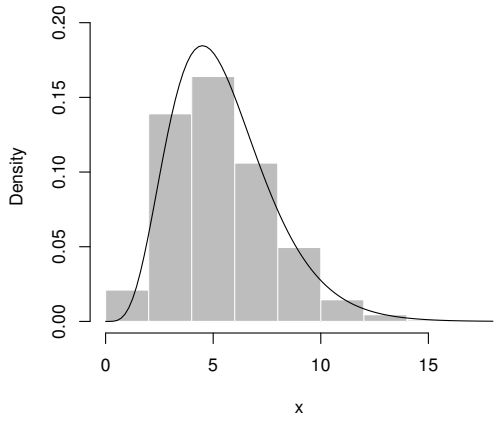

Figure 2: Histogram of 1000 $\mathsf{Gamma}(5.5, 1)$ samples using the accept–reject method.

we may approximate the posterior, i.e., $\theta \mid (X_1, \ldots, X_n)$ is approximately normal, with mean $\mu = \hat{\theta}_n$, the MLE, and variance $\sigma^2 = [nJ_n(\hat{\theta}_n)]^{-1}$, where

$$J_n(\hat{\theta}_n) = \frac{1}{n} \frac{\partial}{\partial \theta} \log L_n(\theta) \Big|_{\theta=\hat{\theta}_n}$$

is the observed information. Given a set of data, these things can be found numerically without much trouble, e.g., using the `nlm` function in R; see the code in Figure 3. Following the hint, we consider a Student-t proposal distribution, with $\mathrm{df} = 4$ and location and scale that match the normal approximation of the posterior, i.e., $\theta' \sim \mu + \sigma\, \mathsf{t}_4$. I simulated data from the Student-t distribution and applied the importance sampling strategy; the value of the Pitman estimator I get is 7.044 based on $M = 1000$ importance samples. (Here I also computed the so-called "effective sample size" which measures roughly how efficient the importance sampler is. Here I get ESS $\approx M$, which is the best possible.)

4. Following the description in [GDS], for data $X_1, \ldots, X_n$, the likelihood is

$$L(\alpha, \eta) \propto \alpha^n \eta^n \exp\Big\{\alpha \sum_{i=1}^n \log X_i - \eta \sum_{i=1}^n X_i^\alpha\Big\}.$$

The recommended prior is of the form $\pi(\alpha, \eta) \propto e^{-\alpha} \eta^{b-1} e^{-c\eta}$, where (b, c) are hyper parameters to be specified. In this case, the posterior density looks like

$$\pi(\alpha, \eta \mid X) \propto \alpha^n \eta^{n+b-1} \exp\Big\{\alpha\Big(\sum_{i=1}^n \log X_i - 1\Big) - \eta\Big(c + \sum_{i=1}^n X_i^\alpha\Big)\Big\}.$$

Though the posterior density is messy and not of a standard form, it is easy to evaluate, so the Metropolis–Hastings algorithm is a good candidate method for sampling from the posterior. Again, following [GDS], we shall consider a proposal distribution of the form:

$$\alpha' \mid \alpha \sim \mathsf{Exp}(\alpha) \quad \text{and} \quad \eta' \mid \eta \sim \mathsf{Exp}(\eta), \quad \text{independent.}$$

```
pit.is <- function(x, df, M) {

  prop.df <- 4
  loglik <- function(u) sum(dt(X - u, df=df, log=TRUE))
  gg <- function(u) -loglik(u)
  opt <- nlm(f=gg, p=mean(x), hessian=TRUE)
  mu <- opt$estimate
  sig <- 1 / sqrt(opt$hessian)
  U <- mu + sig * rt(M, df=prop.df)
  w <- exp(sapply(U, loglik)) / dt((U - mu) / sig, df=prop.df) * sig
  W <- w / sum(w)
  out <- sum(W * U)
  return(list(est=out, ess=M / (1 + var(W))))

}
n <- 50
df <- 5
theta <- 7
X <- theta + rt(n, df=df)
M <- 1000
print(pit.is(X, df, M))
```

Figure 3: R code for the importance sampling method.

In our notation from class, where $q(x, y)$ is the proposal density for y at a given state x, the proposal density looks like

$$q\big((\alpha, \eta), (\alpha', \eta')\big) = \frac{1}{\alpha\eta} \exp\Big\{-\frac{\alpha'}{\alpha} - \frac{\eta'}{\eta}\Big\}.$$

The code I provided on the website automatically computes this acceptance probability using the provided formulas for the transition and target densities. (Note that since the acceptance probability only depends on ratios, we don't need to know the normalizing constants on the posterior density.)

R code to implement the Metropolis–Hastings sampler is given in Figure 4. The data is as given in the problem, and here I shall take the hyper parameters as $b = 2$ and $c = 1$. This choice is based on trial-and-error, i.e., several choices of (b, c) were considered and I picked a pair for which the chain mixed well. In the code, `theta` represents the pair (α, η). A histogram of the marginal posterior distribution for α is shown in Figure 5. From this plot, it is clear that most of the posterior mass is away from $\alpha = 1$, e.g., $\Pi(\alpha > 1 \mid X) \approx 0.999$, so I would say that an exponential model (corresponding to $\alpha = 1$) would not fit the data well; a formal Bayes test of the hypothesis $H_0 : \alpha = 1$ would require Bayes factors, etc.

5. The classical ANOVA model is $Y_{ij} = \theta_i + \varepsilon_{ij}$ where $\varepsilon_{ij} \sim \mathsf{N}(0, \sigma^2)$, for $j = 1, \ldots, n_i$ and $i = 1, \ldots, I$, independent throughout. For the Bayes model, we take (hierarchical) priors as follows:

$$\begin{aligned} \theta = (\theta_1, \ldots, \theta_I) \mid (\mu_\pi, \sigma_\pi^2) &\overset{\text{iid}}{\sim} \mathsf{N}(\mu_\pi, \sigma_\pi^2) \\ (\sigma^2, \mu_\pi, \sigma_\pi^2) &\sim \mathsf{InvGam}(a_1, b_1) \times \mathsf{N}(\mu_0, \sigma_0^2) \times \mathsf{InvGam}(a_2, b_2), \end{aligned}$$

```
X <- c(0.56, 2.26, 1.90, 0.94, 1.40, 1.39, 1.00, 2.32, 2.08, 0.89, 1.68)
n <- length(X)
bb <- 2
cc <- 1
f <- function(theta) {

  alpha <- theta[1]
  eta <- theta[2]
  t1 <- sum(log(X)) - 1
  t2 <- sum(X**alpha) + cc
  o <- alpha**n * eta**(n + bb - 1) * exp(alpha * t1 - eta * t2)
  return(o)

}
dprop <- function(theta, theta0) exp(sum(dexp(theta, 1 / theta0, log=TRUE)))
rprop <- function(theta0) rexp(2, 1 / theta0)
# Load Metripolis-Hastings function "mh()" from the web
out <- mh(c(1,1), f, dprop, rprop, 10000, 1000)
plot(out$x[,1], type="l", col="gray", xlab="Iteration", ylab=expression(alpha))
plot(out$x[,2], type="l", col="gray", xlal="Iteration", ylab=expression(eta))
hist(out$x[,1], freq=FALSE, col="gray", border="white", xlab=expression(alpha))
```

Figure 4: R code for Metropolis–Hastings in the Weibull example in Problem 4.

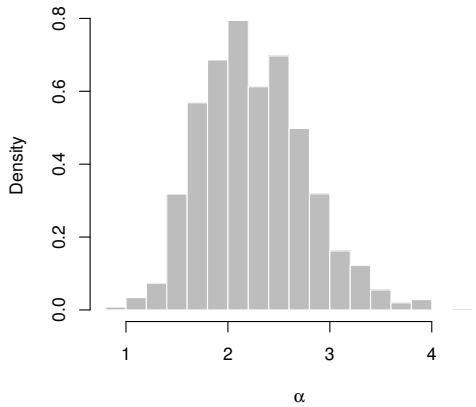

Figure 5: Histogram of 10000 samples from the marginal posterior distribution of α in the Weibull example from Problem 4.

where $(a_1, a_2, b_1, b_2, \mu_0, \sigma_0^2)$ are hyper parameters to be specified. The goal is to construct a Gibbs sampler to simulate from the posterior distribution of $(\theta, \sigma^2, \mu_\pi, \sigma_\pi^2)$ given data $Y = (Y_{ij})$. For this, we need the full conditionals, and following [GDS], we should have four such conditionals.

First we consider the conditional distribution of θ (the vector of treatment means) given everything else. When all other parameters are fixed, this is just a normal model with known variance and a normal prior for the mean—conjugate. Therefore,

$$\theta_i \mid (\sigma^2, \mu_\pi, \sigma_\pi^2, Y) \overset{\text{ind}}{\sim} \mathsf{N}\Big(\frac{n_i\sigma_\pi^2\bar{Y}_i + \sigma^2\mu_\pi}{n\sigma_\pi^2 + \sigma^2}, \frac{\sigma_\pi^2\sigma^2}{n\sigma_\pi^2 + \sigma^2}\Big), \quad i = 1, \ldots, I.$$

Next we consider the conditional distribution of σ^2 given everything else. Let $N = \sum_{i=1}^I n_i$. If θ were known, then the likelihood would look like

$$(\sigma^2)^{-N/2} \exp\{-D_1(\theta)/(2\sigma^2)\},$$

where $D_1(\theta) = \sum_{i=1}^I \sum_{j=1}^{n_i} (Y_{ij} - \theta_i)^2$. It is clear that this combines nicely, in a conjugate way, with the inverse gamma prior on σ^2 to give a modified inverse Gamma conditional posterior for σ^2:

$$\sigma^2 \mid (\theta, \mu_\pi, \sigma_\pi^2, Y) \sim \mathsf{InvGam}(a_1 + N/2, b_1 + D_1(\theta)/2).$$

In [GDS], the conditionals for μ_π and σ_π^2 do not depend on whether there is one variance (as in this case) or many variances (as in their example). So, we get the same distributions as they do. That is, given the vector θ, let $\bar{\theta} = I^{-1}\sum_{i=1}^I \theta_i$ be the average. Then

$$\mu_\pi \mid (\theta, \sigma^2, \sigma_\pi^2, Y) \sim \mathsf{N}\Big(\frac{I\sigma_0^2\bar{\theta} + \sigma_\pi^2\mu_0}{I\sigma_0^2 + \sigma_\pi^2}, \frac{\sigma_0^2\sigma_\pi^2}{I\sigma_0^2 + \sigma_\pi^2}\Big).$$

Finally, for given θ and μ_π, let $D_2(\theta, \mu_\pi) = \sum_{i=1}^I (\theta_i - \mu_\pi)^2$. Then the conditional posterior for σ_π^2 is

$$\sigma_\pi^2 \mid (\theta, \sigma^2, \mu_\pi, Y) \sim \mathsf{InvGam}(a_2 + I/2, b_2 + D_2(\theta, \mu_\pi)/2).$$

This completes the full conditionals. The Gibbs sampler can now be implemented by starting with some initial values, say $\theta_i^{(0)} = \bar{Y}_i$, $i = 1, \ldots, I$, $\mu_\pi^{(0)} = \bar{Y}$, and

$$\sigma^{2(0)} = \frac{1}{N - I}\sum_{i=1}^I \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_i)^2,$$

and then bouncing back and forth sampling various conditional distributions. See the R code in Figure 6. For simplicity in the code,[1] I will assume that the design is *balanced*, i.e., $n_i = n$ for all i, and $N = In$; your example is like this.

I selected some values for the hyper parameters (see the code) and ran the Gibbs sampler to get 5000 simulations from the posterior distribution. Figure 7 shows a

[1] This allows me to store the data Y as a matrix, rather than as a list.

```
anova.gibbs <- function(Y, M, B, a1, b1, a2, b2, mu.0, sig2.0) {

  I <- nrow(Y)
  n <- ncol(Y)
  N <- I * n
  Ybar <- apply(Y, 1, mean)
  D1 <- function(v) sum((Y - v)**2)
  theta <- matrix(0, nrow=M+B, ncol=I)
  sig2 <- mu.pi <- sig2.pi <- numeric(M+B)
  theta[1,] <- Ybar
  sig2[1] <- D1(Ybar) / (N - I)
  mu.pi[1] <- mean(Ybar)
  for(m in 2:(M+B)) {

    D2 <- sum((theta[m-1,] - mu.pi[m-1])**2)
    sig2.pi[m] <- 1 / rgamma(1, a2 + I / 2, b2 + D2 / 2)
    mm <- (I * sig2.0 * mean(theta[m-1,]) + sig2.pi[m] * mu.0) / (I * sig2.0 + sig2.pi[m])
    vv <- sig2.0 * sig2.pi[m] / (I * sig2.0 + sig2.pi)
    mu.pi[m] <- rnorm(1, mm, sqrt(vv))
    sig2[m] <- 1 / rgamma(1, a1 + N / 2, b1 + D1(theta[m-1,]) / 2)
    mm <- (n * sig2.pi[m] * Ybar + sig2[m] * mu.pi[m]) / (n * sig2.pi[m] + sig2[m])
    vv <- sig2.pi[m] * sig2[m] / (n * sig2.pi[m] + sig2[m])
    theta[m,] <- rnorm(I, mm, sqrt(vv))

  }
  out <- list(theta=theta[-B,], sig2=sig2[-B], mu.pi=mu.pi[-B], sig2.pi=sig2.pi[-B])
  return(out)

}
Y <- c(6.58, 6.54, 0.61, 7.69, 2.18, 3.84, 2.48, 3.89, 2.11, 2.46, 5.93, 5.65,
       1.32, 3.27, 6.90, 5.65, 1.81, 2.79, 3.53, 3.11, 5.58, 7.80, 6.33, 4.72,
       7.01, 3.96, 4.60, 5.47, 6.29, 1.97)
Y <- matrix(Y, nrow=5, byrow=TRUE)
o <- anova.gibbs(Y, 5000, 1000, 3, 1, 3, 1, 0, 2)
hist(o$sig2.pi, freq=FALSE, col="gray", border="white", xlab=expression(sigma[pi]^2))
```

Figure 6: R code for the Gibbs sampler in Problem 5.

histogram of the σ_π^2 samples. This parameter controls the variation in the treatment means, so small σ_π^2 corresponds to no significant treatment effects, i.e., all θs approximately equal. In this case, the marginal posterior for σ_π^2 is concentrated pretty close to zero, so there is evidence to suggest that there is no treatment effect.[2] In fact, the data I provided from this problem was obtained by iid normal sampling, so there really is no treatment effect, so the posterior is actually doing the right thing here.

[2]A proper Bayes test for no treatment effect requires further considerations.

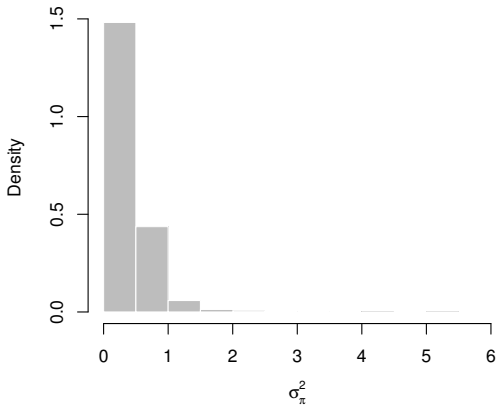

Figure 7: Histogram of 5000 samples from the marginal posterior for σ_π^2 in the ANOVA example in Problem 5.