STA380 Practice Problems for Quiz 7

These problems are not to be handed in, but they are for extra practice for students to be prepared for the quiz.

1. Use `optimize()` to maximise the $Gamma(\alpha = 2, \beta = 3)$ pdf with respect to x.

2. Consider the Anna-11 distribution with probability density function:

$$f_A(a) = \frac{1}{(1-e^{-\rho})^{r-1}} \sum_{i=0}^{r-1} \binom{r-1}{i} (-1)^i e^{-\rho i} \frac{1}{\Gamma(r-1)} (a-\rho i)^{(r-1)-1} e^{-(a-\rho i)} \mathbf{1}_{\{a>\rho i\}}.$$

   Use `optimize()` to maximise the above pdf with respect to a between the interval $[0, 10]$.

3. Consider the Laplace distribution with probability density function:

$$f(x \mid \mu, b) = \frac{1}{2b} \exp\left\{-\frac{|x-\mu|}{b}\right\}, \quad x \in \mathbb{R},\ \mu \in \mathbb{R},\ b > 0.$$

   Here, μ is called the location parameter and b is called the diversity or scale parameter.

   (a) Derive the log likelihood function.

   (b) With $n = 10^4$ replicates, use `VGAM::rlaplace(n, location = 4, scale = 2)` to generate samples from the Pareto distribution. You can read the documentation yourself (`??VGAM::rlaplace`) if you want to understand what the arguments `location` and `scale` are.

   (c) Use `optim()` to maximize the log likelihood. Use all 5 methods. Do any of them provide a good result?

   (d) The closed-form MLEs are as follows:

$$\hat{b}_{MlE} = \frac{\sum_{i=1}^{n} |X_i - \mu|}{n}, \quad \hat{\mu}_{MLE} = \text{median}(X_1, \ldots, X_n)$$

   You don't need to derive these. Compare the answers of the closed-form MLE to the results from `optim()`. What do you conclude?

4. Derive the log likelihood function of the $Beta(\alpha, \beta)$ distribution. With $n = 10^4$ replicates, use `rbeta(n, shape1 = 3, shape2 = 4)` to generate samples from the Beta distribution. Then, use the Newton-Raphson Method to approximate the MLE of the generated sample.

5. Consider $X_i \overset{i.i.d}{\sim} Exponential(\theta)$ where $i = 1, 2, \ldots, n$ where $n = 10^4$. Suppose we had all of the realisations except for one of them, let's say X_1. Construct the EM algorithm and code it in `R`. Use `rexp(n-1, rate = 1/3)` to generate an imcomplete sample.

Below are the solutions.

1. ```
   op <- optimize(dgamma, interval = c(0, 100),
                  shape = 2, scale = 3, maximum = TRUE)
   op$maximum
   # graph; optional but nice to see
   curve(dgamma(x, shape =2, scale = 3), 0, 10)
   abline(h = dgamma(op$maximum, shape = 2, scale = 3),
          col = "red", lty = 2, lwd = 2)
   ```

   You could also brute force and directly code up the pdf instead of using `dgamma()`.

2. ```
   anna_pdf <- Vectorize(function(a, rho, r){
     if(a > ((r-1)*rho)){return(0)}
     total <- 0
     denom <- (1 - exp(-rho))^(r-1)
     for(i in 0:(r-1)){
       part1 <- choose(r-1, i) * exp(-i*rho) * (-1)^(i)
       q <- a - i*rho
       part2 <- if((q > 0) && (a < (r-1)*rho)){
         dgamma(q, shape = r - 1)
       } else {0}
       total <- total + part1*part2
     }
     return(total/denom)
   }, vectorize.args = "a")

   op <- optimize(anna_pdf, interval = c(0, 10),
                  rho = 2.7, r = 5, maximum = TRUE)
   op$maximum

   curve(anna_pdf(x, rho = 2.7, r = 5), 0, 10)
   abline(h = anna_pdf(op$maximum, rho = 2.7, r = 5),
          col = "red", lty = 2, lwd = 2)
   ```

3. (a) Let $\boldsymbol{\theta} = (\mu, b)$. The likelihood function is:

   $$L(\boldsymbol{\theta}|\mathbf{x}) = (2b)^{-n} \exp\left\{\sum_{i=1}^{n} \frac{-|x_i - \mu|}{b}\right\}$$

   The log-likelihood function is:

   $$l(\boldsymbol{\theta}|\mathbf{x}) = -n\log(2b) - \frac{1}{b}\sum_{i=1}^{n}|x_i - \mu|$$

   (b)
   ```
   n<-10^4
   set.seed(1)
   samp <- VGAM::rlaplace(n, location = 4, scale = 2)
   ```

   (c)
   ```
   laplace_likelihood = function(theta, sample){
     mu <- theta[1]
     b <- theta[2]
     # need to add constraint for the optimizer...
     if (b < 0) {return(Inf)}
     n <- length(sample)
     # By default, optim() gives the MINIMUM. need to negate for the max!
     return(-(n*log(2*b) - (1/b)*sum(abs(sample-mu))))
   }

   start_val <- c(1, 1)

   op_1 <- optim(start_val, sample = samp, laplace_likelihood, method = "Nelder-Mead")
   ```

```
op_2 <- optim(start_val, sample = samp, laplace_likelihood, method = "BFGS")
op_3 <- optim(start_val, sample = samp, laplace_likelihood, method = "CG")
op_4 <- optim(start_val, sample = samp, laplace_likelihood, method = "L-BFGS-B",
              lower = c(1e-6, 1e-6))
op_5 <- optim(start_val, sample = samp, laplace_likelihood, method = "SANN")
```

None of them actually provides a good answer. How disappointing. It seems like... you still need to learn how to take a derivative after all.

(d)

```
> mu <- median(samp)
> mu
[1] 3.982892
> sum(abs(samp - mu))/n
[1] 2.030977
```

Conclusion: there are some expressions where the numerical approximation methods will fail. This likely happens when the function is not smooth.

(e) Let $\boldsymbol{\theta} = (\alpha, \beta)$. The likelihood function is:

$$L(\boldsymbol{\theta}|\mathbf{x}) = \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^n \prod_{i=1}^{n} x_i^{(\alpha-1)}(1-x_i)^{(\beta-1)}$$

The log-likelihood function is:

$$\begin{aligned} l(\boldsymbol{\theta}|\mathbf{x}) &= n \log \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\} + (\alpha-1)\sum_{i=1}^{n} \ln(X_i) + (\beta-1)\sum_{i=1}^{n} \ln(1-X_i) \\ &= n\left\{\log(\Gamma(\alpha+\beta)) - \log(\Gamma(\alpha)) - \log(\Gamma(\beta))\right\} + (\alpha-1)\sum_{i=1}^{n} \ln(X_i) + (\beta-1)\sum_{i=1}^{n} \ln(1-X_i) \end{aligned}$$

The code is:

```
n<-10^4
samp <- rbeta(n, shape1 = 3, shape2 = 4)

beta_likelihood = function(theta, sample){
  alpha <- theta[1]
  beta <- theta[2]
  # need to add constraint for the optimizer...
  if (alpha <= 0 || beta <= 0) {
    # we can properly maximize now
    return(-Inf)
  }
  gams <- lgamma(alpha + beta) - lgamma(alpha) - lgamma(beta)
  n * gams + (alpha-1)*sum(log(sample)) + (beta-1) * sum(log(1-sample))
}

newton_raphson_beta <- function(theta0, sample, tol = 0.01){
  theta <- theta0
  no_root <- TRUE
  while(no_root){
    L <- numDeriv::grad(beta_likelihood, theta, sample = sample)
    H <- numDeriv::hessian(beta_likelihood, theta, sample = sample)

    new_theta <- theta - solve(H, L)
    if(sqrt(sum((new_theta - theta)^(2))) <= tol){
      no_root = FALSE
    }
    theta <- new_theta
  }
  return(theta)
}
newton_raphson_beta(theta0 = c(2, 1), sample = samp)
```

4. Below is the write-up for the EM algorithm. First, we want to find our initial starting point $\hat{\theta}^{(0)}$ by computing the incomplete log likelihood.

$$
\begin{aligned}
L(\theta|\mathbf{x}_{-1}) &= \frac{1}{\theta^{(n-1)}} e^{-\frac{1}{\theta}\sum_{i=2}^{n} x_i} \\
l(\theta|\mathbf{x}_{-1}) &= -(n-1)\ln(\theta) - \frac{1}{\theta}\sum_{i=2}^{n} x_i \\
\frac{d\, l(\theta|\mathbf{x}_{-1})}{d\,\theta} &= -\frac{n-1}{\theta} + \frac{1}{\theta^2}\sum_{i=2}^{n} x_i \\
0 &\stackrel{set}{=} -\frac{n-1}{\theta} + \frac{1}{\theta^2}\sum_{i=2}^{n} x_i \\
\hat{\theta}^{(0)} &= \frac{\sum_{i=2}^{n} x_i}{n-1}
\end{aligned}
$$

Below is for the complete log likelihood:

$$
\begin{aligned}
L(\theta|\mathbf{x}) &= \frac{1}{\theta^n} e^{-\frac{1}{\theta}\sum_{i=1}^{n} x_i} \\
l(\theta|\mathbf{x}) &= -n\ln(\theta) - \frac{1}{\theta}\sum_{i=1}^{n} x_i \\
&= -n\ln(\theta) - \frac{x_1}{\theta} - \frac{1}{\theta}\sum_{i=2}^{n} x_i
\end{aligned}
$$

Now let's compute the expected value (expectation step):

$$
\begin{aligned}
\mathbb{E}[l(\theta|\mathbf{x}) \,|\, \hat{\theta}^{(k)}, \mathbf{x}_{-1}] &= \mathbb{E}\left[-n\ln(\theta) - \frac{x_1}{\theta} - \frac{1}{\theta}\sum_{i=2}^{n} x_i \,\middle|\, \hat{\theta}^{(k)}, \mathbf{x}_{-1}\right] \\
&= -n\ln(\theta) - \frac{\mathbb{E}\left[x_1 \,\middle|\, \hat{\theta}^{(k)}, \mathbf{x}_{-1}\right]}{\theta} - \frac{1}{\theta}\sum_{i=2}^{n} x_i \\
&= -n\ln(\theta) - \frac{\hat{\theta}^{(k)}}{\theta} - \frac{1}{\theta}\sum_{i=2}^{n} x_i
\end{aligned}
$$

Thus we claim that:

$$
Q(\theta, \hat{\theta}^{(k)}) = -n\ln(\theta) - \frac{\hat{\theta}^{(k)}}{\theta} - \frac{1}{\theta}\sum_{i=2}^{n} x_i
$$

Maximization step:

$$
\begin{aligned}
\frac{d\, Q(\theta, \hat{\theta}^{(k)})}{d\,\theta} &= \frac{-n}{\theta} - \frac{\hat{\theta}^{(k)}}{\theta^2} + \sum_{i=2}^{n} \frac{x_i}{\theta^2} \\
0 &\stackrel{set}{=} \frac{-n}{\theta} - \frac{\hat{\theta}^{(k)}}{\theta^2} + \sum_{i=2}^{n} \frac{x_i}{\theta^2} \\
\hat{\theta}^{(k+1)} &= \frac{\sum_{i=2}^{n} X_i - \hat{\theta}^{(k)}}{n}
\end{aligned}
$$

```
n <- 10^4
incomp_samp <- rexp(n-1, rate = 1/3)

# Assume the sample we put in here is incomplete.
em_exp <- function(sample, tol = 0.0001){
  # Step 1: get the starting point
  theta <- sum(sample) / length(sample)
  no_sol <- TRUE
  while(no_sol){
    new_theta <- (sum(sample) - theta) / (length(sample)+1)
    if(sqrt(sum((new_theta - theta)^(2))) <= tol){
      no_sol = FALSE
```

```
    }
    theta <- new_theta
  }
  return(theta)
}

em_exp(incomp_samp)
```