

STA 410/2102 — Answers to Practice Questions for First Test

Please note that these questions do **not** cover all the topics that may be on the test.

**Question 1:** Suppose we have  $n$  binary (0/1) observations,  $y_1, \dots, y_n$ , that we model as being i.i.d. samples from a Bernoulli distribution with the probability of 1 being  $p$ . Suppose that  $1/10$  of the observations are 1s and  $9/10$  are 0s, so the maximum likelihood estimate of  $p$  is  $\hat{p} = 1/10$ . Suppose we use floating-point numbers of the form  $0.d_1d_2d_3 \times 10^E$ , where  $d_1, d_2, d_3$  are decimal digits and  $E$  is an integer exponent in the range  $-100$  to  $+100$ . For what values of  $n$  can we compute the probability of  $y_1, \dots, y_n$  using the maximum likelihood estimate  $\hat{p}$  without the result underflowing to zero?

An approximate answer is good enough. Recall that  $(1 - \epsilon)^{1/\epsilon} \approx e^{-1} \approx 0.37$  for  $\epsilon$  close to zero.

*The probability of  $y_1, \dots, y_n$  will be  $(1/10)^x(1 - 1/10)^{n-x}$ , where  $x$  is the number of the  $n$  observations that are 1s. Since we are assuming that  $x = n/10$ , we can write this probability as follows (noting that  $0.37^2 \approx 1/10$ ):*

$$10^{-n/10} (1 - 1/10)^{(10/10)n(9/10)} \approx 10^{-n/10} 0.37^{n(9/100)} \approx 10^{-n/10} 10^{-n(9/100)/2} \approx 10^{-n(29/200)}$$

*The smallest non-zero floating-point number that can be represented in the form described is  $0.100 \times 10^{-100} = 10^{-101}$  (or  $0.001 \times 10^{-100}$  if “unnormalize” numbers are allowed).*

*So underflow will occur approximately when  $-n(29/200)$  is less than  $-101$ . So we will be able to compute the probability without underflow for values of  $n$  less than about  $20200/29 \approx 697$ .*

**Question 2:** Suppose we try to solve the equation  $x^4 - 81 = 0$  using Newton-Raphson iteration.

- a) How will we find the next guess at the solution,  $x^{(t+1)}$ , from the current guess,  $x^{(t)}$ ?

*The derivative of  $x^4 - 81$  with respect to  $x$  is  $4x^3$ , so the update will be*

$$x \leftarrow x - (x^4 - 81) / (4x^3)$$

- b) Suppose we start from an initial value of  $x^{(0)} = 5$ . Here are the values found in the first four iterations:

$$\begin{aligned} x^{(1)} &= 3.912 \\ x^{(2)} &= 3.2722427442656 \\ x^{(3)} &= 3.03212968848923 \\ x^{(4)} &= 3.00050709092522 \end{aligned}$$

The exact answer is of course 3 (for the solution in this neighborhood).

Estimate what value of  $x^{(5)}$  will be the result of doing one more iteration, without actually doing this iteration. Try to get as accurate an answer as you can by considering in detail the rate at which the error ought to be going down.

*Newton iteration converges quadratically, so once the current estimate is near the correct answer, the error goes down as  $|e^{t+1}| = c|e^t|^2$ . The errors for the last two iterations shown are 0.0321, and 0.000507, so we can conclude that  $c$  is about  $0.000507/0.0321^2 = 0.49$ . We therefore expect the error after one more iteration to be about  $0.49 \times 0.000507^2$ , which is about 0.00000013.*

*(The actual value for the next estimate is 3.000000128534392.)*

**Question 3:** Suppose we numerically evaluate the integral

$$\int_0^1 x^4 dx$$

using the midpoint rule. Using 100 points, the approximation we get is 0.199983333625. Using 1000 points, the approximation we get is 0.1999983333363. The exact answer is of course  $1/5$ . Estimate what approximation we will get if we use the midpoint rule with 2000 points.

*For large  $n$ , the error using the midpoint rule should have the form  $cn^{-2}$ , for some  $c$ . Supposing that  $n = 1000$  is large enough for this to be true, doubling the number of points to  $n = 2000$  should result in the error going down by a factor of 4, from about  $-0.0000002$  to  $-0.00000005$ . So we would expect the estimate with 2000 points to be about 0.1999995.*

**Question 4:** You've probably heard the rule that about 68% of values from a normal distribution will be within one standard deviation of the mean. Write an R program to verify this using the Trapezoid Rule for integration, making use only of basic R facilities and the `dnorm` function (not `pnorm`).

```
# Integration of f from a to b using the Trapezoid Rule with n intervals.  
# It's not as efficient as it could be (for example, it evaluates f twice  
# at interior points).
```

```
trapezoid <- function (f,a,b,n)  
{  
  I <- 0  
  h <- (b-a)/n  
  for (i in 1:n)  
    I <- I + h * (f(a+(i-1)*h) + f(a+i*h)) / 2  
  I  
}
```

```
# Trapezoid Rule used to find the area under the standard normal density  
# curve between -1 and +1.
```

```
print (trapezoid(dnorm,-1,1,10000))
```

**Question 5:** Suppose we have  $n$  i.i.d. (independent, identically-distributed) data points  $x_1, \dots, x_n$ , that are real values in the interval  $(-1, +1)$ . We model these observations as having the distribution on  $(-1, +1)$  with the following density function:

$$f(x) = (1 + \theta x) / 2$$

where  $\theta$  is an unknown model parameter in the interval  $(-1, +1)$ . We wish to find the maximum likelihood estimate for  $\theta$ .

- a) Write down the likelihood function,  $L(\theta)$ , and the log likelihood function,  $\ell(\theta)$ , based on the observations  $x_1, \dots, x_n$ .

$$L(\theta) = \prod_{i=1}^n (1 + \theta x_i) / 2$$

$$\log L(\theta) = \sum_{i=1}^n \log((1 + \theta x_i) / 2) = -n * \log(2) + \sum_{i=1}^n \log(1 + \theta x_i)$$

- b) Fill in the body of the following R function so that it will compute the first derivative of the log of the likelihood for `theta` based on the observations in the vector `x`:

```
log_lik_deriv1 <- function (theta, x) {
  sum (x/(1+theta*x))
}
```

- c) Fill in the body of the following R function so that it will compute the second derivative of the log of the likelihood for `theta` based on the observations in the vector `x`:

```
log_lik_deriv2 <- function (theta, x) {
  sum (-x^2/(1+theta*x)^2)
}
```

You should use only basic R facilities, not `deriv` or `D`.

- d) Fill in the body of the following R function so that it returns the maximum likelihood estimate for  $\theta$  given the data vector `x`. You should find the MLE using Newton iteration for `niters` iterations starting from `initial_theta`:

```
mle <- function (x, initial_theta, niters) {
  theta <- initial_theta
  for (i in 1:niters) {
    theta <- theta - log_lik_deriv1(theta,x) / log_lik_deriv2(theta,x)
  }
  theta
}
```

You should use only basic R facilities for these questions, not the `nlm` function.

You do not need to do anything in this function to guard against the possibility of moving to a point outside the interval  $(-1, +1)$ . (We'll assume that the initial value is good enough to avoid this happening.)

**Question 6:** Suppose we have  $n$  i.i.d. data points,  $x_1, \dots, x_n$ , that are positive real numbers, with each having the distribution with density function

$$f(x) = \frac{1}{\theta(1+x/\theta)^2}$$

where  $\theta$  is an unknown positive model parameter.

Derive the formulas needed to use Newton-Raphson iteration to find the maximum likelihood estimate for  $\theta$ , and write an R program that takes as arguments a data vector  $x$ , an initial guess for  $\theta$ , and the number of Newton-Raphson iterations to do, and which returns a list consisting of the maximum likelihood estimate for  $\theta$  along with its standard error, found using the observed information.

```
ll <- function (x, theta) sum (-log(theta)-2*log(1+x/theta))

lld <- function (x, theta)
  -length(x)/theta + (2/theta)*sum(x/(x+theta))

lld2 <- function (x, theta)
  length(x)/theta^2 - (2/theta^2)*sum(x/(x+theta)) - (2/theta)*sum(x/(x+theta)^2)

mle <- function (x, k)
{
  theta <- 1

  for (i in 1:k)
  { theta <- theta - lld(x,theta)/lld2(x,theta)
    cat(i,": theta =",theta," ll =",ll(x,theta),"\n")
  }

  list (mle=theta, std.err=sqrt(-1/lld2(x,theta)))
}
```