Statistics 572
Homework 1

Alejandro Robles

September 6th, 2018

1 Newton-Raphson Method for Gaussian Parameters

Generate sample of size 100 from N(2,1). Use Newton-Raphson method to estimate μ and σ^2 .
Use $\mu_0 = 3.5$ and ${\sigma_0}^2 = 0.1$, tol = 0.0001.

Solution:
Newton-Raphson method could be defined as:

$$\hat{\theta} = \theta_0 + I^{-1}(\theta_0) * U(\theta_0) \tag{1}$$

In class, we derived:

$$I^{-1}(\theta_0) = \begin{bmatrix} \frac{{\sigma_0}^2}{n} & 0 \\ 0 & \frac{2{\sigma_0}^4}{n} \end{bmatrix}$$

And

$$U(\theta_0) = \begin{bmatrix} \frac{n(\bar{Y}-\mu_0)}{{\sigma_0}^2} \\ \frac{1}{2{\sigma_0}^2}(\frac{\sum(Y_i-\mu_0)^2}{{\sigma_0}^2} - n) \end{bmatrix}$$

Therefore,

$$I^{-1}(\theta_0) * U(\theta_0) = \begin{bmatrix} \bar{Y} - \mu_0 \\ \frac{\sum(Y_i-\mu_0)^2}{n} - \sigma_0^2 \end{bmatrix}$$

Finally,

$$\begin{bmatrix} \hat{\mu} \\ \hat{\sigma^2} \end{bmatrix} = \begin{bmatrix} \mu_0 \\ {\sigma_0}^2 \end{bmatrix} + \begin{bmatrix} \bar{Y} - \mu_0 \\ \frac{\sum(Y_i-\mu_0)^2}{n} - \sigma_0^2 \end{bmatrix}$$

MATLAB Code:

```
% Sample Data
true_mu = 2;
true_sigma = 1;
n = 100;
r = normrnd(true_mu,true_sigma,1,n);
ybar = mean(r);

% Initial Guess
m0 = 3.5;
sigma_squared_0 = 0.1;

% Tolerance
tol = 0.00001;

% Max Iterations
loop = 0;
maxloop = 100;

% First Iteration
p0 = [m0 ;sigma_squared_0];
step = [ybar - p0(1) ; (sum(power((r - p0(1)), 2))/(n)) - p0(2)];

Nphat = p0  + step;

% Repeat until conditions met
while (abs(Nphat(1) - p0(1)) > tol  && abs(Nphat(2) - p0(2)) > tol) || loop < maxloop
    p0 = Nphat;
    step = [ybar - p0(1) ; (sum(power((r - p0(1)), 2))/(n)) - p0(2)];
    Nphat = p0 + step;
    loop = loop+1;
end

% Display result
disp(Nphat)
```

Result:

```
>> normal_nr_example
    2.0286
    0.9762
```

Discussion:

The Newton-Raphson method did not converge using tolerance condition. But after 100 iterations, we get numbers relatively close to $\mu = 2$ and $\sigma^2 = 1$. Maybe increasing max iteration will give algorithm enough iterations to converge using tolerance condition.

2 Exercise 2.1

Write a function using MATLABs functions for numerical integration such as quad or quadl that will find P(X $\leq$ x) when the random variable is exponentially distributed with parameter λ . See help for information on how to use these functions.

MATLAB Code:

```
function integral = exp_cdf(x, lambda)
exp_pdf = @(b)exppdf(b,1/lambda);
integral = quad(exp_pdf , 0, x);
```

Result:

```
>> exp_cdf(2,1)
ans =
    0.8647
```

Discussion:

This makes sense because the CDF of exponential distribution is F(x; λ) = 1 - $e^{-\lambda x}$

So therefore $F(x = 2; \lambda = 1) = 1 - e^{-2} = 0.8647$

3 Exercise 2.2

Verify that the exponential probability density function with parameter λ integrates to 1. Use the MATLAB functions quad or quadl. See help for information on how to use these functions.

MATLAB Code:

```
>> exp_cdf(100,1)
ans =
    1.0000
```

Discussion: Although the interval for exponential probability density function is between [0, ∞], we use 100 instead because this is a numerical approximation and quad does not accept inf as argument. Regardless, we see that the numerical approximation for this integral is 1.

4 Exercise 2.11

Using the functions fminbnd (available in the standard MATLAB package), find the value for x where the maximum of the N(3, 1) probability density occurs. Note that you have to find the minimum of f(x) to find the maximum of f(x) using these functions. Refer to the help files on these functions for more information on how to use them.

MATLAB Code:

```
function mode = find_norm_mode(mu, sigma)
norm_pdf = @(x) -normpdf(x,mu,sigma);
mode = fminbnd(norm_pdf, -100,100);
```

Result:

```
>> find_norm_mode(3,2)
ans =
    3.0000
```

Discussion: Because Normal distributions are unimodal such that the mode is always μ, then we know that the value for x where the maximum of the N(3, 1) probability density occurs is 3. Something to note is that we chose a range between -100 to 100 to look for the mode. This may or may not always be an optimal range to look for the mode.

5 Exercise 2.17

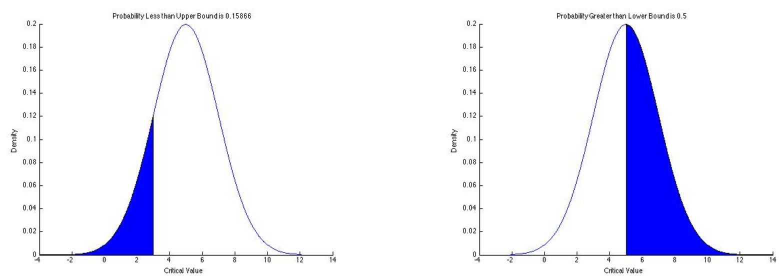
Using the function normspec, find the probability that the random variable defined in example 2.5 (Gaussian) assumes a value that is less than 3. What is the probability that the same random variable assumes a value that is greater than 5? Find these probabilities again using the function normcdf.

MATLAB Code:

```
% Set up the parameters for the normal distribution.
mu = 5;
sigma = 2;

% Set up the upper and lower limits. These are in the two element vector 'specs'.
specs = [-inf , 3];
% Use normspec to integrate pdf between limits presented
prob = normspec(specs, mu, sigma)

% Similarly,
specs2 = [5,inf];
prob2 = normspec (specs2, mu, sigma)

% Using normcdf to calculate the CDF for the Gaussian.
prob3 = normcdf(3,mu,sigma)
prob4 = 1- normcdf(5,mu,sigma)
```

Result:

```
>> exercise_2_17
prob =
    0.1587
prob2 =
    0.5000
prob3 =
    0.1587
prob4 =
    0.5000
```



(a) P(X ≤ 3) (b) P(X > 5)

Figure 1: CDF for N(5,2)

Discussion: First method is calculating $P(a < X < b) = \int_a^b f_X dx$, while second method is calculating $F_x(x) = P(X \leq x) = \int_{-\infty}^{x} f_X dt$. So if $a = -\infty$, which is the case for P(X ≤ 3), then both methods will be identical. Now, because Gaussian is symmetrical then we also have that $P(X > x) = 1 - F_x(x)$. This explains why P(X > 5) can be calculated using both methods as shown in above code.