CS 504 – Programming Languages for Data Analysis

Assignment 2: Normal PDF

## I. Problem: Normal PDF

The Gaussian distribution is a symmetric "bell curved" shaped distribution, which can be found throughout nature. In fact, it is defined by two parameters, $\mu$ and $\sigma^2$, which are the mean and variance, respectively. The mean $\mu$ can take on any value and $\sigma^2$ is restricted to be positive. The phrase standard normal signifies the case of a normal distribution with $\mu = 0$ and $\sigma^2 = 1$. In the general case, the PDF is given by,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The CDF of the normal distribution is not available as a simple expression. However, it is frequently needed, and hence statistical tables or software are often used. The CDF of the standard normal random variable is,

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right)$$

The second expression represents $\Phi(.)$ in terms of the error function *erf(.)*. It is a mathematical special function defined as,

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt$$

With $\Phi(.)$ (or alternately *erf(.)*) tabulated, one can move on to a general normal random variable with mean $\mu$ and variance $\sigma^2$. In this case, the CDF is available via,

$$\Phi\left(\frac{x-\mu}{\sigma}\right)$$

***Your main task in this assignment is to write functions in Julia, Python, R, and Octave to plot the standard normal PDF***, along with its first and second derivatives. The first derivative is clearly 0 at the PDF's unique maximum at $x = 0$. The second derivative is 0 at the points $x = -1$ and $x = +1$. These are exactly the inflection points of the normal PDF (points where the function switches between being locally convex to locally concave or vice-versa). In fact, you have to explore the packages that can help you to program and to illustrate with plots the normal PDF and its first and second derivatives.

**Hint:** To help you, I offer you the Julia code to have an idea how to write the programs that plot the normal PDF and its first and second derivatives. The following code example illustrates the use numerical derivatives from the *Calculus* package. The code also presents two alternative ways of implementing $\Phi(.)$ and shows they are equivalent. One way uses *cdf()* from the Distributions package and the other way uses *erf()* from the *SpecialFunctions* package.

```
1   using Distributions, Calculus, SpecialFunctions, Plots; pyplot()
2
3   xGrid = -5:0.01:5
4
5   PhiA(x) = 0.5*(1+erf(x/sqrt(2)))
6   PhiB(x) = cdf(Normal(),x)
7
8   println("Maximum difference between two CDF implementations: ",
9   maximum(PhiA.(xGrid) - PhiB.(xGrid)))
10
11  normalDensity(z) = pdf(Normal(),z)
12
13  d0 = normalDensity.(xGrid)
14  d1 = derivative.(normalDensity,xGrid)
15  d2 = second_derivative.(normalDensity, xGrid)
16
17  plot(xGrid, [d0 d1 d2], c=[:blue :red :green],label=[L"f(x)" L"f′(x)" L"f′′(x)"])

18  plot!([-5,5],[0,0], color=:black, lw=0.5, xlabel="x", xlims=(-5,5), label="")
```

Lines 5-9 are dedicated to showing the equivalence of the two ways of implementing $\Phi(.)$. In line 11 we define the function *normalDensity()*, which takes an input z, and returns the corresponding value of the PDF of a standard normal distribution. Then in lines 14-15, the functions *derivative()* and *second_derivative()* are used to evaluate the first and second derivatives of *normalDensity* respectively. The curves are plotted in lines 17-18.

Which programming language that provided relevant solution for this assignment? Which difficulties will you face if you want to solve this problem using Scala programming language?

## II. Useful links

1. [For loop with Julia]
2. [Python For Loops]

3. R for loop
4. For loop with Octave
5. Plotting with Julia
6. Line Plots - R Base Graphs
7. Python Matplotlib
8. 2D & 3D Plots with Octave
9. Online Scala Editor
10. Programming in Scala
11. Scala Tutorial

## III. Submission

You have to submit a pdf file that contains four outputs that are related to Julia, Python, R, and Octave. In addition, you have to submit the sources files in Julia, Python, R, and Octave. ***Please, do not use .zip nor .rar file in your submission***.

Good luck :)