Life of a Particle : Assignment 2

Sam Meehan

Due Date : 13 January 2017

How to Submit

This assignment should be submitted by replying to the email sent out rby the tutors - **Use the online form!** - with the link to your GitHub repository. Contained within this repository should be a markdown README.md file which provides a guide to the contents of the repository, along with instructions of how to run the code, where necessary. For some of the questions, you will need to write an extended answers (not just code), so your repository should also include your latex file and the compiled PDF document.

Coding Like a Hipster - (Just Directions)

You don't need to do any work here, these are just directions.

For many common tasks, there are built in functions already existing in python, or which can be imported from the `math` or `numpy` modules. In particular, the ones you may have used before are `min(LIST)`, `max(LIST)`, `sum(LIST)`, `len(LIST)`. However, these are not necessary for "good" programmers. Nearly all programs can be written using the following small set of operations

- + (add)
- ∗ (multiply
- / (divide)
- = (assignment)
- == (equals comparison)
- < (less than comparison)
- > (greater than comparison)

In addition to these operators, it is taken for granted in programming that you can use the following features as well

- variables and lists - for storing the initial dataset
- for loops
- if statements
- print statements - for viewing your code

In this quiz, these are the only things that may appear. If you determine that you absolutely need some other function or operator, then include a comment clearly describing why this is the case.

Finally, you are not allowed to "hard code" in your program, meaning that there cannot be code that you must manually change each time you run it. An example of this is the length of a list. If I have a list *[1,4,2,5,3,6]*, and you want to use the number of items in the list in your code, then the number "6" may not appear in your code.

Standard Deviation Like a Hipster
For this question, follow the directions to code like a “hipster”. Note that you can use the ROOT package to make histograms or graphs if you like.

Given a set of numbers Q (example Q =[2,3,4,2]), write a program which calculates the standard deviation of this set. For the purposes here, the definition of the standard deviation of a set of numbers is

$$\sigma(Q) = \sqrt{\frac{\sum_{i=0}^{N-1} (x_i - x_{avg})^2}{N}}$$

$$x_{avg} = \sum_{i=0}^{N-1} (x_i)$$

$$N = N(Q)$$

Numerical Integration Like a Hipster
For this question, follow the directions to code like a "hipster". Note that you can use the ROOT package to make histograms or graphs if you like.

One of the most powerful tools in math is the integral. In particular, this is powerful because it can be used to compute the area under a curve, a concept which can be generalized to refer to volumes or more complicated quantities, like the amount of work performed on a particle by an electric field as the particle moves through a trajectory. One typically learns to compute (definite) integrals as

$$I_{y(x)}(a,b) = \int_a^b x^n dx = \frac{x^{n-1}}{n-1}|_{x=a}^{x=b} = \frac{1}{n-1}(b^{n-1} - a^{n-1})$$

and this can be used to compute more complicated integrals like

$$I_{y(x)}(a,b) = \int_a^b y(x)dx$$

However, there is another method to compute the definite integral I computationally called a *Riemann Sum*. Graphically, it proceeds as shown in Figure **??**. The integral is **approximated** by a sum of the are of the rectangles of some width (Δx) with a height equal to $y(x_{eval})$ where $y(x_{eval})$ can be evaluated at the (i) left, (ii) right, or (iii) middle of the rectangle. In this case, the value of the integral is

$$I_{y(x)}(a,b) \sim \sum_{i=0}^{N-1} y(x_i) \times (x_{i+1} - x_I) = \sum_{i=0}^{N-1} y(x_i) \times \Delta x$$

Now, typically (but not necessarily) the value of Δx is fixed and taken to be $\Delta x = \frac{b-a}{N}$. Now, if you remember integral calculus, then you will remember that

$$\lim_{N\to\infty} \sum_{i=0}^{N-1} y(x_i) \times \Delta x = \int_a^b y(x)dx$$

Therefore, we can calculate the integral by writing a piece of code that performs this definite sum! So, the task is to write a piece of python code which performs the integral of $y(x) = 3x^3 - 4x^2 + 3.2198x$ from $a = 1$ to $b = 9$. Perform this calculation analytically and then perform it numerically using your code for N =[10,50,100,200]. How close does this approximation approach to the analytic result.

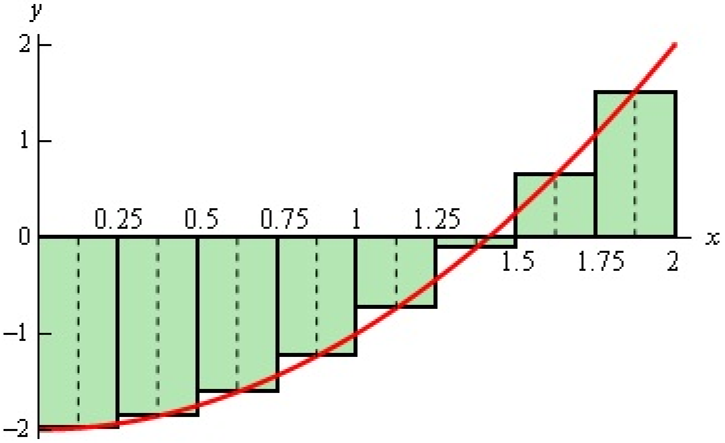

Figure 1: An illustration of the Riemann sum used to approximate the integral of the function $y(x)$ (the red curve) from x =0 to x =2. In this case, the step size Δx is taken to be 0.25 and the function $y(x)$ is evaluated at the center point of each interval.

Fitting a Line

For this question, follow the directions to code like a “hipster”. Note that you can use the ROOT package to make histograms or graphs if you like.

One procedure that is used regularly in computational modelling is that of “fitting”. This itself is a very broad topic and here we will implement one of the more basic procedures called the “minimization of least squares”. To make this concrete, let’s pretend that you have measurements of a particles position at equal time intervals and obtain the following measurements :

x [m]	y [m]
1.0	0.5
2.2	1.1
3.0	1.3
3.9	2.1
4.8	2.6

Now, you have a good reason to believe that this particle is moving on a straight line path, but your eyesight is not good and if you plot these points on a scatter plot, they don’t seem to be perfectly along a straight line. Nonetheless, you would like to determine the functional form of the path followed by the particle, which you know to be $y = f(x; m, b) = m \times x + b$. To do this, you can use the “minimization of least squares” as follows :

- Choose values for the parameters (m, b)
- Calculate the value of the fit error which equals $S = \sum_{points} (f(x_i) - y_i)^2$ as represented graphically in Figure **??**.
- Make a list which matches a single choice of (m, b) to a value of S and fill it with many (m, b, S) choices
- Find the entry in this list of (m, b, S) which has the smallest value of S. The corresponding (m, b) values are what we call the “best fit” values of the parameters of our models.

Use this procedure to determine the best fit values of the line which describes the data you observed and create a plot of the data points and your best fit line.

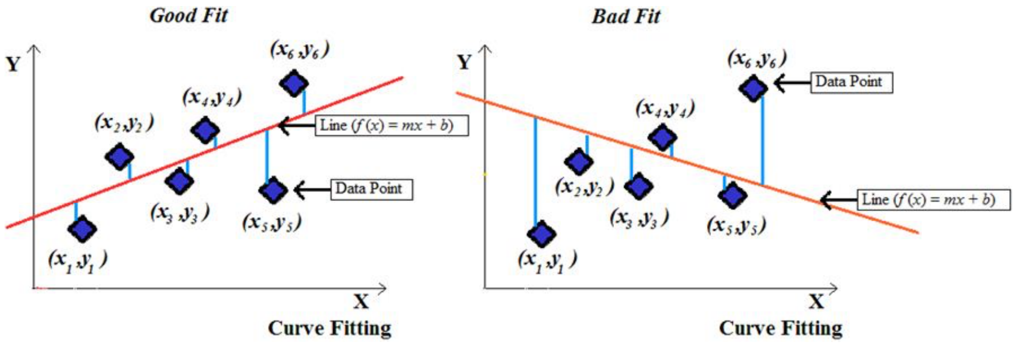

Figure 2: An illustration of the procedure to minimize the fit error and thereby determine the model parameters m and b that best describe our data.

Gaussian Random Numbers

For this question, you may import the python random library. However, you may only use the random.random() function call which generates a uniformly distributed random number between [0,1.0).

Starting from a set of uniformly distributed random numbers from [0,1.0) (Don't be a hipster, use python for this!) generate a set of 10000 random numbers according to a Gaussian distribution of mean $\mu = 2.0$ and width $\sigma = 1.4$. As a reminder, the Gaussian distribution is also called the "normal distribution" and has the following functional form :

$$Gauss(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} Exponential[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2]$$

where $Exponential[a] = e^a$

Modelling the Particle In a Box
We discussed at great length the solution of the Schrodinger equation for the case of a particle in a one-dimensional box with sides at x =0 and x =a (Go review Griffith's QM - Chapter 2.2 if you are not sure of how the calculation proceeds.). This is the model by which we want to describe a particle, namely the answer to the question "Where is the particle?". This will be done using the wave function $\Psi(x, t)$ of the particle, which itself may be complex valued as is composed as a linear superposition of the eigenfunctions

$$\Psi(x,t) = \sum_{n=0}^{\infty} c_n \phi_n(t) \psi_n(x) = \sum_{n=0}^{\infty} c_n e^{-\frac{i}{\hbar}E_n t} sin(\frac{n\pi}{a}x)$$

However, the probability rule, which will be a function describing the PDF $P(x)$ of the location of the particle, must be a real valued function from which a single event (a single collapse of the wave function) can be viewed as the generation of a single number from this distribution $P(x)$. Therefore, we must translate this complex function into a real-valued function. Two reasonable ways to do this are as follows :

- Square Then Sum : Take the square modulus of each of the eigenfunctions and then add these squared terms.
  - $P_A(x,t) = \sum_{n=0}^{\infty} c_n \phi_n(t) \psi_n(x) c_n^* \phi_n^*(t) \psi_n^*(x)$
- Sum Then Square : Add the individual eigenfunctions and then take the square modulus of this sum.
  - $P_B(x,t) = \sum_{n=0}^{\infty} c_n \phi_n(t) \psi_n(x) \times \sum_{m=0}^{\infty} c_m^* \phi_m^*(t) \psi_m^*(x)$

If you are feeling overwhelmed looking at these equations,its alright, we are going to deal with a simplified system. Let's imagine that we know that the particle is initially placed in the box in a state $\Psi(x, t)$ which is only composed of the E_1 and E_2 eigenstates. So the wave function is simply

$$\Psi(x,t) = c_1 \phi_1(t) \psi_1(x) + c_2 \phi_2(t) \psi_2(x)$$

First, describe how these two different possibly probability rules differ (if they do) qualitatively? Is there time dependence to one of the probability rules? What if you set t =0, meaning that the observation is made immediately after you put the particle in the box? Does the time dependence go away?

For this simplified system, you have been provided with a set of 5000 data measurements (included on the assignment page). Your goal is to determine which one of these two probabily rule transformations P_A or P_B is the one that really occurs in nature. To this end, you should probably start by examining the data, either by using descriptive statistics, or maybe making a histogram. Can you observe anything about the data just from this? Now try to generate a predictive set of data according to the two different models that you have for the probability, assuming that the measurements being made are performed immediately at t =0. To fully describe the model, there are therefore two separate questions to answer

- What is the probability rule that comes from nature?
- What are the coefficients c_1 and c_2? (To make things less involved, pretend that we know that c_1 and c_2 are positive and between $[0, 1]$.)

To go about this, we will need a way to compare your ensemble of predictions to that of the observations. This can be done by first casting the obervations or predictions in the form of two histograms (p and o) and then performing a calculation of the χ^2 of these two histograms where

$$\chi^2 = \sum_{i \in bins(p,o)} \frac{(p_i - o_i)^2}{\sigma_{p_i}^2 + \sigma_{o_i}^2}$$

and (p_i, o_i) is the bin content of p and o at bin i and(σ_{p_i}, σ_{o_i}) are the corresponding errors on these bins.