Herramientas Computacionales para Ciencias
Homework 11

Mauricio Sevilla*

29/04/2019

Rules

Note Read carefully the complete homework before starting, so you will know what is the results you have to get!.

This week we are going to concentrate on statistics and histograms from simulated data.

On this assignment you will have to construct some functions and use the `matplotlib` + `NumPy` +`SciPy` structure to plot the results. There is an additional point that must be developed on class before sending the assignment. This part must be saved on a jupyter Notebook named as your UniAndes username.

[2.0/5.0] Maxwell Boltzmann Distribution

This is one of the most important distributions on molecular dynamics, proposed by Maxwell and then by Boltzmann back in 1860 and 1872 (1877) respectively.

Based on the kinetic theory to describe ideal gases, the velocities on each direction distribute as a Gaussian, but when combining them,

$$v = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{1}$$

we get a different distribution.

One has to take a look to the derivation (Which with the statistical mechanics theory becomes very easy), because on the literature one can find that the velocity distribution is

$$f(v) = 4\pi v^2 \left(\frac{m}{2\pi k_B T}\right)^{3/2} e^{\frac{-mv^2}{2k_B T}} \tag{2}$$

but this result only applies to a 3 dimensional gas!.

Here, we will use results from a 2 Dimensional simulation!!!. If you want to see a small demonstration, see the following link

https://github.com/jmsevillam/Herramientas-Computacionales-UniAndes/blob/master/examples/Reverse.gif

The relationship we may use is

$$p(v) = 2\pi v \frac{m}{k_B T} \exp\left(-\frac{m}{2k_B T} v^2\right) \tag{3}$$

On the following link, you may find the data result from the simulation.

There are 3 columns, v_x, v_y and $v = \sqrt{v_x^2 + v_y^2}$

https://raw.githubusercontent.com/jmsevillam/Herramientas-Computacionales-UniAndes/master/Data/Velocities.dat

- Collect the data, (use `genfromtxt` as usual).
- Convince yourself that the distributions of v_x and x_y (first two columns of the file), by doing the histogram (you may use `plt.hist`).
- To *prove* that the velocities distribute as we said before, we are going to do a fit of the histogram to the model, so first define a function (3)

  Hint: You do not have to use all those values of $k_B T$ and m, use an parameter for the amplitude and the coefficient of the exponential.
- Do the histogram (with 20 bins) of the velocities (third column of the file) but!, save the values.

```
histogram=plt.hist(velocities,bins=20)
```

*email=j.sevillam@uniandes.edu.co

Then you will have save on `histogram` the amplitudes and limits of the boxes. To have one value for each box, we take the average

```
x_vals=(histogram[1][1:]+histogram[1][:-1])/2
frequencies=histogram[0]
```

- Use the function `curve_fit` with the variables `x_vals` and `frequencies`[1]. Hint: you can use `p0=[1000,1000**2.]` as your initial parameters.
- Plot again the histogram of velocities, but also the result from the fit.

[1.0/5.0] Wigner Distribution

There is a very important distribution on random matrix theory, which is called *Wigner Semicircle Distribution* named after the physicist Eugene Wigner (Nobel prize awarded), also known for his phase space representation of quantum mechanics.

This distribution can be found by calculating the distribution of the eigenvalues of symmetric random matrices with Gaussian distributed entries, and that is exactly how we will calculate it.

- We will generate 1000 matrices of size 100×100, so it is a good idea for you to define those values before starting.
- do a `for` loop running on the number of matrices (1000 in our case), and then inside that `for`, generate a Gaussian distributed matrix by using the function `np.random.uniform`. So far we have been using that function to generate a *list*, but, how shall we generate a matrix.

  To generate a matrix of size $N \times N$, with Gaussian distributed numbers with mean 0 and deviation 1, we use

```
Matrix=np.random.uniform(0,1,(N,M))
```

- We may use that structure to create a mean 0, standard deviation 1 Gaussian distributed matrix of size 100×100.
- Make the matrix symmetric. (Add it by the transpose.).

  Hint: The transpose of a matrix `M` on `python` can be calculated as `M.T`
- for each matrix, calculate its eigenvalues, it is easily done with the function

```
eigen=np.linalg.eigvals(matrix)
```

- Save on an array the values of all the eigenvalues of all the matrices, you can use the function `append`, such as

```
eigen_vals=np.append(eigen_vals,eigen)
```

- do a histogram of them. (Use 100 bins)

[1.0/5.0] Central Limit Theorem

The central Limit Theorem is one of the reasons, Gaussians are so important.

It stands that under some general conditions, the distribution of the sum S_n of n independent random variables, can be approximated by a Gaussian even if the original variables themselves are not normally distributed.

That means that, some results of Gaussian distribution, can be used (In some sense), to other distributions.

- We are going to construct a Gaussian distribution from the averages of uniform distributions, so we have to make a loop to generate the data. Use a `for` from 0 to 10000 (The number of *Experiments* we are having), and inside, create a set of random uniform variables (1000) and calculate the mean,

  Hint you can create a list or array outside the `for`, and use the `append` method, for example, if you choose a list you can use

```
data.append(np.random.random(1000).mean())
```

  if you choose an array you may use the structure described above for the eigenvalues.
- Plot the histogram of `data`.
- To be sure it is a Gaussian, define a function to make a fit (just like we did on the previous assignment.).
- Use the same strategy we saw before to do the fit Hint Generate the data to do the fit as,

```
histogram=plt.hist(data,bins=20)
x_vals=(histogram[1][1:]+histogram[1][:-1])/2
frequencies=histogram[0]
```

  and use the function `curve_fit` to find the optimal parameters.
- Plot again the histogram, but also the function fitted.

We are almost done with the course, I hope you have enjoyed the course as much as I did!.

[1] Remember that you can use different names, I only give a reference.