Notebook

January 27, 2019

Statement I $\frac{\sum_{i=1}^{n} a_i x_i}{\sum_{i=1}^{n} a_i} = \sum_{i=1}^{n} x_i$

False.
Example:
a = [1, 1]
x = [1, 1]
left = 1
right = 2
left != right

Statement II $\sum_{i=1}^{n} x_1 = nx_1$

True.

$\sum_{i=1}^{n} x_1 = x_1 + x_1 + ... + x_1 = nx_1$

Statement III $\sum_{i=1}^{n} a_3 x_i = n a_3 \bar{x}$

True.

$\sum_{i=1}^{n} a_3 x_i = a_3 \sum_{i=1}^{n} x_i = a_3(n\bar{x}) = n a_3 \bar{x}$

Statement IV $\sum_{i=1}^{n} a_i x_i = n\bar{a}\bar{x}$

False.

Example:
a = [0, 1]
b = [1, 0]
left = 0
right = 0.5
left != right

Question 4a Suppose we have the following scalar-valued function on x and y:

$$f(x,y) = x^2 + 4xy + 2y^3 + e^{-3y} + \ln(2y)$$

Compute the partial derivative of$f(x,y)$with respect to$x$.
$\frac{\partial f(x,y)}{\partial x} = 2x + 4y$

Now compute the partial derivative of $f(x, y)$ with respect to y:

$$\frac{\partial f(x,y)}{\partial y} = 4x + 6y^2 - 3e^{-3y} + 1/y$$

Finally, using your answers to the above two parts, compute $\nabla f(x,y)$ (the gradient of $f(x,y)$) and evaluate the gradient at the point $(x=2, y=-1)$.

$$\begin{aligned}\nabla f(x,y) &= (\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}) \\ &= (2x+4y, 4x+6y^2-3e^{-3y}+1/y)\end{aligned}$$

$$\nabla f(x,y)|_{(x=2,y=-1)} = (0, 13-3e^3)$$

Question 4b Find the value(s) of x which minimizes the expression below. Justify why it is the minimum.
$\sum_{i=1}^{10}(i-x)^2$

$$f(x)=\sum_{i=1}^{10}(i-x)^2$$

$$\begin{aligned}\frac{df(x)}{dx}&=\sum_{i=1}^{10}2(x-i)\\&=20x-110\end{aligned}$$

$$\frac{d^2f(x)}{dx^2}=20>0$$

f(x) is convex, minimum is obtained when $\frac{df(x)}{dx}=0$, namely x=5.5

Question 4c Let $\sigma(x) = \frac{1}{1+e^{-x}}$. Show that $\sigma(-x) = 1 - \sigma(x)$.

$$\sigma(-x) = \frac{1}{1+e^{x}}$$

$$\begin{aligned}1 - \sigma(x) &= 1 - \frac{1}{1+e^{-x}} \\ &= \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{e^{x}+1}\end{aligned}$$

Therefore, $\sigma(-x) = 1 - \sigma(x)$

Question 4d Show that the derivative can be written as:

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$\begin{aligned}\frac{d}{dx}\sigma(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}}\frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}}\frac{1}{e^{x}+1} \\ &= \sigma(x)(1-\sigma(x))\end{aligned}$$

Question 4e Write code to plot the function $f(x) = x^2$, the equation of the tangent line passing through $x = 8$, and the equation of the tangent line passing through $x = 0$.

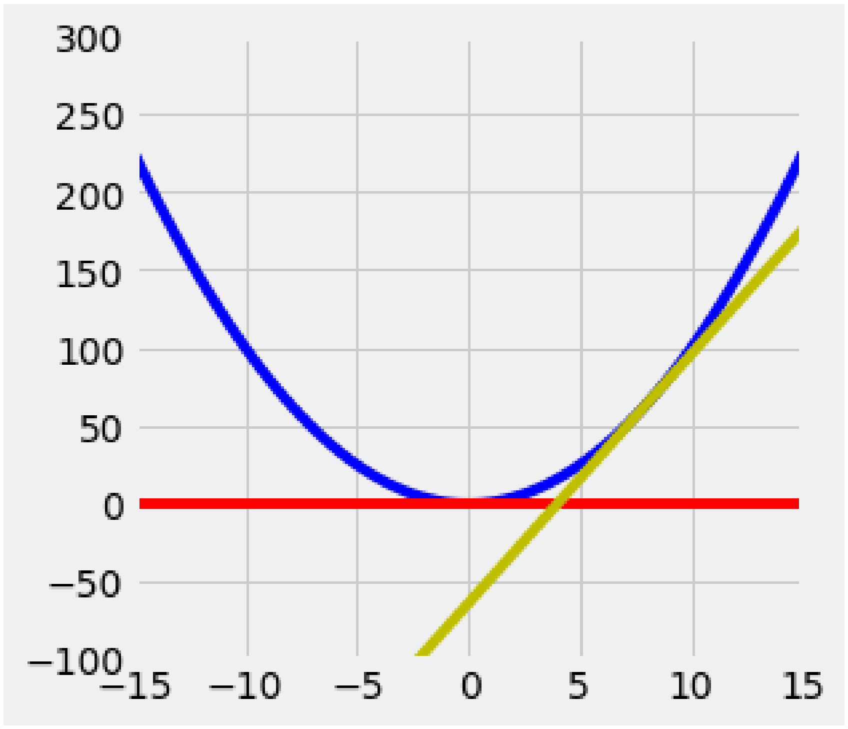
Set the range of the x-axis to (-15, 15) and the range of the y axis to (-100, 300) and the figure size to (4,4).

Your resulting plot should look like this:

You should use the `plt.plot` function to plot lines. You may find the following functions useful:

- `plt.plot(..)`
- `plt.figure(figsize=..)`
- `plt.ylim(..)`
- `plt.axhline(..)`

```
In [45]: def f(x):
             return x**2

         def df(x):
             return 2*x

         def tangent_line(x, x0, f, df):
             return df(x0)*(x-x0)+f(x0)

         def plot(f, df):
             x = np.linspace(-15, 15, 100)
             y = f(x)
             y1 = tangent_line(x, 0, f, df)
             y2 = tangent_line(x, 8, f, df)

             fig = plt.figure(figsize=(4,4))
             ax = fig.add_subplot(111)
             # plt.subplots_adjust(left=0.09, bottom=0.26,right=0.95,top=0.95,wspace=0.2,hspace=0.4)
             plt.plot(x, y, 'b')
             plt.plot(x, y1, 'r')
             plt.plot(x, y2, 'y')
         #     plt.title('$f(t)=3sin(2\pi t)$', size=18, y=1.03)
         #     plt.xlabel('t', size=16)
         #     plt.ylabel('f(t)', size=16)
             # ax.tick_params(axis='both', which='major', labelsize=26)
             # plt.legend(loc='best', fontsize=26)
             plt.xlim(-15, 15)
             plt.ylim(-100, 300)
             plt.xticks(np.arange(-15, 20, 5))

         plot(f, df)
```


300
250
200
150
100
50
0
−50
−100
−15
−10
−5
0
5
10
15

0.0.1 Question 5

Consider the following scenario:

Only 1% of 40-year-old women who participate in a routine mammography test have breast cancer. 80% of women who have breast cancer will test positive, but 9.6% of women who don't have breast cancer will also get positive tests.

Suppose we know that a woman of this age tested positive in a routine screening. What is the probability that she actually has breast cancer?

Hint: Use Bayes' rule.

$$\begin{aligned}
P(cancer = yes, test = positive) &= P(cancer = yes)P(test = positive|cancer = yes) \\
&= 0.01 * 0.80 \\
&= 0.008
\end{aligned}$$

$$\begin{aligned}
P(test = positive) = &\ P(cancer = yes)P(test = positive|cancer = yes)+ \\
&\ P(cancer = no)P(test = positive|cancer = no) \\
= &\ 0.01 * 0.80 + 0.99 * 0.096 \\
= &\ 0.10304
\end{aligned}$$

$$\begin{aligned}
P(cancer = Yes|test = positive) &= \frac{P(cancer = yes, test = positive)}{P(test = positive)} \\
&= 0.008/0.10304 \\
&= 0.078
\end{aligned}$$

Therefore, the probability that she actually has breast cancer is 7.8%.

0.0.2 Question 6

Generate a 2 by 2 plot that plots the function $g(t)$ as a line plot with values $f = 2, 8$ and $a = 2, 8$. Since there are 2 values of f and 2 values of a there are a total of 4 combinations, hence a 2 by 2 plot. The rows should vary in a and the columns should vary in f.

Set the x limit of all figures to $[0, \pi]$ and the y limit to $[-10, 10]$. The figure size should be 8 by 8. Make sure to label your x and y axes with the appropriate value of f or a. Additionally, make sure the x ticks are labeled $[0, \frac{\pi}{2}, \pi]$. Your overall plot should look something like the one above.

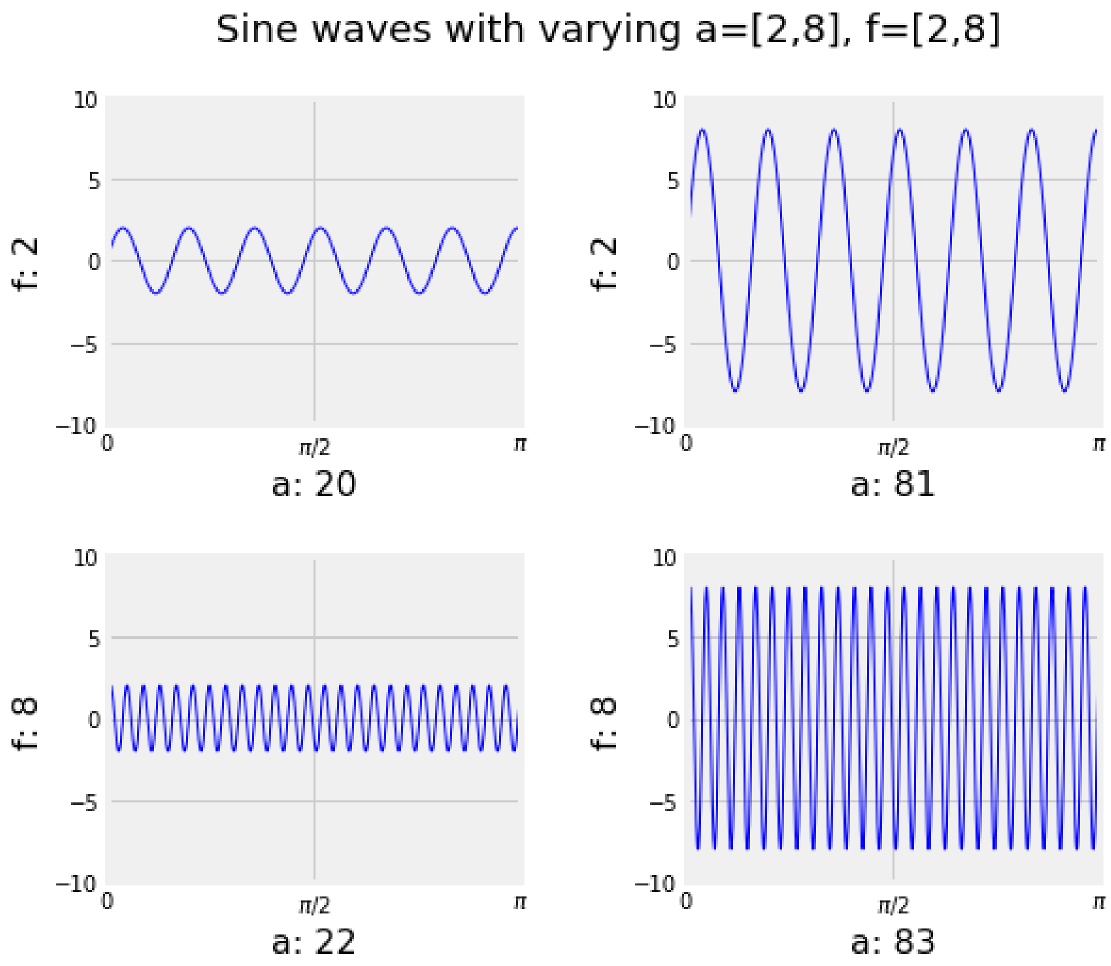
Hint 1: Modularize your code and use loops.

Hint 2: Are your plots too close together such that the labels are overlapping with other plots? Look at the `plt.subplots_adjust` function.

Hint 3: Having trouble setting the x-axis ticks and ticklabels? Look at the `plt.xticks` function.

Hint 4: You can add title to overall plot with `plt.suptitle`.

```
In [17]: import itertools

         def function_g(t, a=0, f=0):
             return a*np.sin(2*np.pi*f*t)

         def plot_xy(x, y, xlabel, ylabel):
             plt.plot(x, y, 'b', linewidth=1)
         #     plt.title('$f(t)=3sin(2\pi t)$', size=18, y=1.03)
             plt.xlabel(xlabel, size=16)
             plt.ylabel(ylabel, size=16)
             # ax.tick_params(axis='both', which='major', labelsize=26)
             # plt.legend(loc='best', fontsize=26)
             plt.xlim(0, np.pi)
             plt.ylim(-10, 10)
             plt.xticks([0, np.pi/2, np.pi], [0, '$\pi/2$', '$\pi$'])

         def plot_question(a_list=[], f_list=[], super_title=''):
         #     af_list = itertools.product(a_list, f_list)
             af_list = itertools.product(f_list, a_list)
             x = np.linspace(0, np.pi, 1000)

             fig = plt.figure(figsize=(8,8))
         #     plt.title('Sine waves with varying a=[2,8], f=[2,8]', size=18, y=1.03)
             plt.suptitle(super_title, size=18, y=0.97)
             for i, (f, a) in enumerate(af_list):
                 ax = fig.add_subplot(len(f_list), len(a_list), i+1)
                 plt.subplots_adjust(left=0.09, bottom=0.26,right=0.90,top=0.90,
                                     wspace=0.4,hspace=0.4)
                 plot_xy(x, function_g(x, a=a, f=f),
                         xlabel='a: ' + str(a) + str(i),
                         ylabel='f: ' + str(f))


         plot_question(a_list=[2,8], f_list=[2,8],
                       super_title='Sine waves with varying a=[2,8], f=[2,8]')
```



Sine waves with varying a=[2,8], f=[2,8]
f: 2
a: 20
f: 2
a: 81
f: 8
a: 22
f: 8
a: 83
10
5
0
−5
−10
0
π/2
π

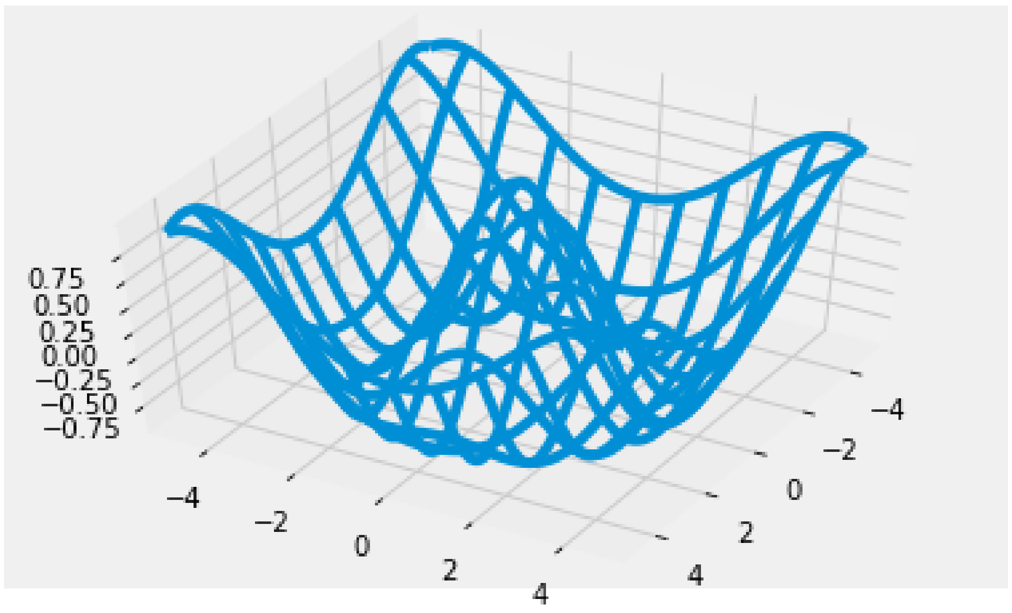
0.0.3 Question 7

We should also familiarize ourselves with looking up documentation and learning how to read it. Below is a section of code that plots a basic wireframe. Replace each `# Your answer here` with a description of what the line above does, what the arguments being passed in are, and how the arguments are used in the function. For example,

```
np.arange(2, 5, 0.2)
# This returns an array of numbers from 2 to 5 with an interval size of 0.2
```

Hint: The `Shift` + `Tab` tip from earlier in the notebook may help here. Remember that objects must be defined in order for the documentation shortcut to work; for example, all of the documentation will show for method calls from `np` since we've already executed `import numpy as np`. However, since `z` is not yet defined in the kernel, `z.reshape()` will not show documentation until you run the line `z = np.cos(squared)`.

```
In [18]: from mpl_toolkits.mplot3d import axes3d

         u = np.linspace(1.5*np.pi, -1.5*np.pi, 100)
         # Assign u as an array of 100 numbers evenly distributed
         # between 1.5*np.pi and -1.5*np.pi (end points included)
         [x,y] = np.meshgrid(u, u)
         # Crete a 2D meshgrid and return the coordinate matrices,
         # the ranges of x and y are same as u
         squared = np.sqrt(x.flatten()**2 + y.flatten()**2)
         z = np.cos(squared)
         # Assign z as the list of cosine values of the distance
         # from each pair of (x, y) to (0, 0)
         z = z.reshape(x.shape)
         # Reshape z from 1D list to 2D matrix, which is the same as x,
         # because x and y were flattened when calculating z above

         fig = plt.figure()
         ax = fig.add_subplot(111, projection='3d')
         # Add a 3D subplot and return the axes. The layout of all
         # subplots is 1*1 and this subplot is the first one (and the only one).
         ax.plot_wireframe(x, y, z, rstride=10, cstride=10)
         # Plot a 3D wireframe. Downsampling stride as 10 in each direction.
         ax.view_init(elev=50., azim=30)
         # Set the elevation as 50 and azimuth as 30 of the axes.
         plt.savefig("figure1.png")
         # Save the current figure as "figure1.png".
```


0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−4
−2
0
2
4
−4
−2
0
2
4

0.0.4 Question 8

For a data-driven question of your choice, describe your approach and thought process in addressing this question. Outline what a sensible workflow might look like, including framing the question and identifying relevant data. Also consider transversal issues such as ethics and governance with respect to your question.

This question is about data-driven reasoning; you should focus more on *what* to do, than on *how* do to it exactly. You may use any of the questions presented in the first lecture, excluding the real estate, crowd size, and COMPAS questions. A complete response should contain about 250 words.

Only given a title of an academic paper from a certain field, can we predict which journal in a list of candidates is the most possible one to accept it? By using data-driven method to answer this question, we might be able to get some clue about how to frame the title of a paper before submitting it.

The initial question I want to ask is what the difference of paper content is in high- and low-impact journals. However, I think it is a little abstract and more details are needed to narrow down the scope and make the question a concrete and approachable one. Considering the title is the most compressed data which can reflect what has been done in a paper. I rephrase the question as "Only given a title of an academic paper from a certain field, can we predict which journal in a list of candidates is the most possible one to accept it?"

A possible workflow is like this:

1. Identifying and framing data-enabled questions
   Given a dataset containing paper titles from a list of academic journals, can we learn and classify which title should go to which journal? For each journal, we can use the F1 score on test set to evaluate if the classification is successful. If is succeed, it indicates there are indeed some features learned, which represents the difference of the "taste" of different journals.

2. Imaging and identifying the nature of the data.
   There are many academic papers published on the internet and the corresponding paper titles and journal names are visible to everyone. Webscraping could be used to collect the data. However, the amount of data could be huge since there are so many papers from all different academic fields. It is hard to make a comprehensive study on all the papers in terms of the computational capability. A certain field and some journals could be selected as representatives in this study, namely, a cluster sampling is required in this study.

3. Designing the data collection procedure.
   According to 2, we should select some typical journals in the same fields and collecting the papers from these journals for a cluster sampling. Then, we would code webscraper to collect the titles of all the published papers in these journals.

4. Collecting the data.
   Run the webscraper in 3 to obtain data.

5. Exploratory data analysis.
   Check and clean the data to confirm the data obtained is what is required for this study.
   Are the titles in right format? They should not include extra symbols such as HTML tags from the webpage.
   Are the journal names are merged? The journal name might be the full name or abbreviation in the webpage, the different names for the same journal should be merged.

6. Data pre-processing
   The paper titles could be mixed with different writing styles. Some of them could be using words all starting with capital character, while some of them only has the first words starting with capitals. Pre-processing is required to clean the data.

7. Answer the question.
   Splitting the data into train/validation/test dataset.
   Draw simple statistics to explore the data to capture potential features.
   Establish feature vector and train classifier.

If the features can be used to classify the data in the test dataset, it indicates the features or the combination of features are found represents the difference of the "taste" of different journals.

8. Translating the results back to the domain.
   Translate the features or the combination of features found in 7 into words human can understand.

Possible issues:
Some journals might not allow a webscraper to crawl data in their websites.
Different journals may have different numbers of papers, which means it should be considered how to balance the data for different categories.