Programming and Visualisation

This exercise sheet is to familiarize yourself with techniques that could be useful for solving the programming exercises and visualizing your results. Choose the programming language you are most comfortable with. Python[1], R, Julia or MATLAB). For more information, see the programming resources on ISIS.

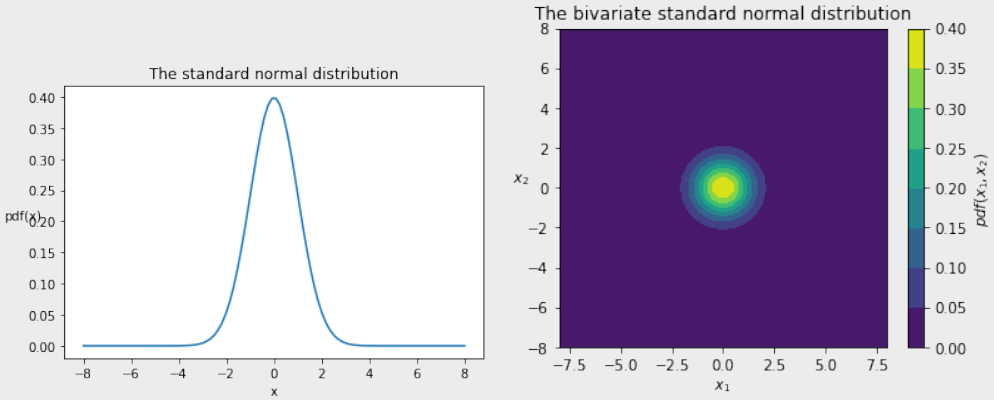
The full implementation of the solution will only be provided for this exercise. This will not be the case for other programming exercises throughout the course, only the results.

Exercise H0.1: The Gaussian distribution (homework, 3 points)

(a) Visualize the probability density function (pdf) of the standard normal distribution ($\mu = 0$, $\sigma^2 = 1$) by plotting $pdf(x)$ as a line plot with equally spaced argument $x \in [-8, 8]$.

(b) Visualize the pdf of a bivariate Gaussian distribution. Evaluate the pdf for the arguments $x_1 \in [-8, 8]$ and $x_2 \in [-8, 8]$. Visualize using a filled contour plot and include a colorbar.

Solution



[1] We recommend Python and the usage of Jupyter Notebooks. *Matplotlib* is the go-to package for plotting in Python. The Python package *seaborn* is also a useful visualization tool.

Exercise H0.2: Comparing apples and oranges (homework, 4 points)

The dataset[2] `applesOranges.csv` contains 200 samples (observations) with columns `x.1` and `x.2`) providing measurements of two types of objects indicated by the column `y`. Samples are represented by the 2D input vector $\underline{\mathbf{x}} \in \mathbb{R}^2$ and the label $y_T \in \{0, 1\}$. $y_T = 0$ indicates that the sample is that of an "apple", while $y_T = 1$ assigns an observation to "orange".

(a) Read the data contained in the file `applesOranges.csv`.

(b) Produce a 2D scatter plot for $\underline{\mathbf{x}}$. Use different markers to distinguish "apple" points from "orange" points. Add a legend to your plot.

(c) Visualize the conditional densities for "apples" and "oranges" separately (i.e. $P(\underline{\mathbf{x}} \,|\, y_T = 0)$ and $P(\underline{\mathbf{x}} \,|\, y_T = 1)$ by:

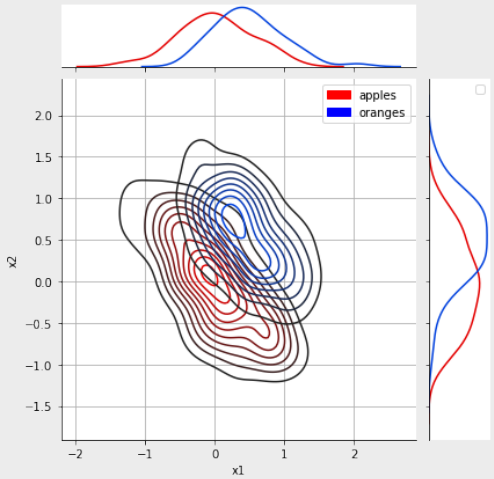
  a) computing a 2D histogram for all the points with $y_T = 0$ and turn it into a density,

  b) visualizing the bin counts as a heat map. You can produce a heatmap the same way you would visualize a 2D matrix as an image or you can produce a heat map using filled contour plots.

  c) Do the same for all points with $y_T = 1$.

Hint: It is possible to do this in Python using only *Matplotlib*. Visualizing both histograms in separate figures is a good start. Using filled contours with a transparancy (i.e. alpha channel) will allow you to overlay the histograms onto the same figure. Use a different colormap to distinguish between the two histograms. You can use interpolation to get smoother density plots. Another way to go about it in Python is to use *seaborn*. Seaborn makes it easy to compute and visualize a density in one go but you would still need to overlay both densities onto the same plot.

Solution



See corresponding jupyter notebook for more details and figures.

[2] This data file (and those required for future exercise sheets) is available on ISIS.

Exercise H0.3: Working with vectors and matrices (homework, 1 point)

We are going to use the data from `applesOranges.csv` to apply some common vector and matrix operations.

(a) Load the data contained `applesOranges.csv` such that your observations $\underline{\mathbf{x}}$ are stored in one matrix $\underline{\mathbf{X}} \in \mathbb{R}^{2\times 200}$ separate from the labels y_T.

(b) Generate a vector with 200 elements and set all its elements to 1.

(c) Concatenate the vector of ones to your input matrix $\underline{\mathbf{X}}$ such that these ones become the first row in the new matrix $\underline{\mathbf{X}}'$.

(d) Multiply each sample in $\underline{\mathbf{X}}'$ by the vector $\underline{\mathbf{w}} := (-1.05,\ 2.183,\ 2.171)$ and store the results of this projection. Use different techniques for calculate this projection:

   a) Use a dot product operation (e.g. `numpy.dot` for Python, `dot` in Matlab, ...)

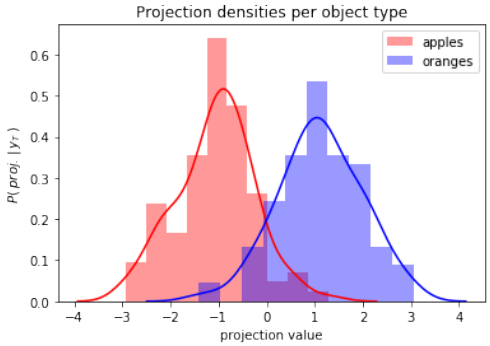
   b) Iterate over the elements of both vectors and calculate the sum of products (i.e. $\sum_i w_i x_i^{(\alpha)}$, where $x_i^{(\alpha)}$ is the i-th component of the observation (row) α in $\underline{\mathbf{X}}'$).

(e) Compute the exact same projection using *only* a matrix operation. Obtain the projection for each column without explicitly iterating over them, instead compute all projections using only a single instruction.

(f) Verify that the different techniques yield the same result.

(g) Create a scatter plot of the labels y_T vs. their corresponding projection.

(h) Can you reproduce the following density plots of the projections?



Exercise H0.4: Data Processing: Image data (homework, 2 points)

(a) Read in the data from the file `natIMG.jpg`.

(b) Plot the data using a heatmap with a 'color' scale ranging from black (minimal value of all pixels in that image) to white (maximal occuring pixel value).

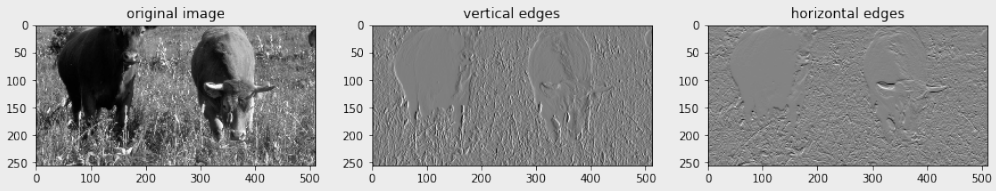
(c) Detect vertical edges in the image by convolving the image with a 3×3 Sobel filter. The kernel of the 3×3 Sobel filter is represented by the following matrix:

$$\underline{\mathbf{H}}_{\text{Sobel}} := \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

(d) Visualize the response of the edge detector applied to the image.

(e) Repeat the above for a horizontal edge detector.

Solution



Total 10 points.