Measures of similarity, summary statistics and probabilities with PYTHON

Objective: The overall objective is to get a basic understanding for measures of similarity as well as summary statistics. Upon completing this exercise it is expected that you:

- Understand how to calculate summary statistics such as mean, variance, median, range, covariance and correlation.
- Understand the various measures of similarity such as Jaccard and Cosine similarity and apply similarity measures to query for similar observations.

PYTHON Help: You can get help in your Python interpreter by typing `help(obj)` or you can explore source code by typing `source(obj)`, where `obj` is replaced with the name of function, class or object.
Furthermore, you get context help in Spyder after typing function name or namespace of interest. In practice, the fastest and easiest way to get help in Python is often to simply Google your problem. For instance: `"How to add legends to a plot in Python"` or the content of an error message. In the later case, it is often helpful to find the *simplest* script or input to script which will raise the error.

Piazza discussion forum: You can get help by asking questions on Piazza:
piazza.com/dtu.dk/fall2019/october2019

Software installation: Extract the Python toolbox from the Dropbox folder . Start Spyder and add the toolbox directory (`<base-dir>/02450Toolbox_Python/Tools/`) to `PYTHONPATH` (Tools/PYTHONPATH manager in Spyder). Remember the purpose of the exercises is not to re-write the code from scratch but to work with the scripts provided in the directory `<base-dir>/02450Toolbox_Python/Scripts/` Representation of data in Python:

	Python var.	Type	Size	Description
	`X`	numpy.array	$N \times M$	Data matrix: The rows correspond to N data objects, each of which contains M attributes.
	`attributeNames`	list	$M \times 1$	Attribute names: Name (string) for each of the M attributes.
	`N`	integer	Scalar	Number of data objects.
	`M`	integer	Scalar	Number of attributes.
Classification	`y`	numpy.array	$N \times 1$	Class index: For each data object, `y` contains a class index, $\mathtt{y}_n \in \{0, 1, \ldots, C-1\}$, where C is the total number of classes.
	`classNames`	list	$C \times 1$	Class names: Name (string) for each of the C classes.
	`C`	integer	Scalar	Number of classes.

3.2 Summary Statistics

3.2.1 Consult the script `ex3_2_1.py`. Calculate the (empirical) mean, standard deviation, median, and range of the following set of numbers:

$$\{-0.68, -2.11, 2.39, 0.26, 1.46, 1.33, 1.03, -0.41, -0.33, 0.47\}$$

Script details:

- *Look at the help page of the functions* `mean()`, `std()`, `median()`, `min()` *and* `max()` *of NumPy* `array` *class.*

3.3 Measures of similarity

We will use a subset of the data on wild faces described in [1] transformed to a total of 1000 gray scale images of size 40×40 pixels, we will attempt to find faces in the data base that are the most similar to a given query face. To measure similarity we will consider the following measures: SMC, Jaccard, Cosine, ExtendedJaccard, and Correlation. These measures of similarity are given by:

$$\begin{aligned}
\mathrm{SMC}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\text{Number of matching attribute values}}{\text{Number of attributes}} \\
\mathrm{Jaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\text{Number of matching presences}}{\text{Number of attributes not involved in 00 matches}} \\
\mathrm{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|} \\
\mathrm{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^\top \boldsymbol{y}} \\
\mathrm{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\mathrm{cov}(\boldsymbol{x}, \boldsymbol{y})}{\mathrm{std}(\boldsymbol{x})\mathrm{std}(\boldsymbol{y})}
\end{aligned}$$

where $\text{cov}(\boldsymbol{x}, \boldsymbol{y})$ denotes the covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\text{std}(\boldsymbol{x})$ denotes the standard deviation of $\boldsymbol{x}$.

Notice that the SMC and Jaccard similarity measures only are defined for binary data, i.e., data that takes values of $\{0, 1\}$. As the data we analyze is non-binary, we will transform the data to be binary when calculating these two measures of similarity by setting

$$x_i = \begin{cases} 0 & \text{if } x_i < \text{median}(\boldsymbol{x}) \\ 1 & \text{otherwise.} \end{cases}$$

Note that, depending on the situation, it can be incorrect to encode information in a single binary attribute—and this is true for binary attributes in general. If the meaning behind the value 0 is not specifically non-presence of an attribute, it can be erroneous. For instance, if male/female is encoded in one binary attribute (male: 0, female: 1), some measures will not model the information carried in being male, and a one-of-out-K encoding would be a proper representation.

For the next step, we will look at the USPS handwritten digit database. The digits dataset contains 9298 16x16 handwritten (single) digits images in greyscale.

3.3.1 Inspect and run the script `ex3_3_1.py`. The script loads the digits dataset, computes the similarity between a selected query image and all others, and display the query image, the 5 most similar images, and the 5 least similar images. The value of the used similarity measure is shown below each image. Try changing the query image and the similarity measure and see what happens.

3.3.2 We will investigate how scaling and translation impact the following three similarity measures: Cosine, ExtendedJaccard, and Correlation. Let α and β be two constants. Which of the following statements are correct? Check your answers with the script `ex3_3_2.py`

$$\begin{aligned}
\text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\
\text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\
\text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\beta + \boldsymbol{x}, \boldsymbol{y})
\end{aligned}$$

Script details:

- *Type* `help(similarity)` *to learn about the Python function that is used to compute the similarity measures.*
- *Even though a similarity measure is theoretically invariant e.g. to scaling, it might not be exactly invariant numerically.*

3.4 Tasks for the report

Provide the basic summary statistics of your attributes preferable in a table and consider if attributes are correlated, see also the functions numpy.cov() and numpy.corrcoef(). Specifically address the questions:

- describe the basic summary statistics of the attributes.

References

[1] Tamara L Berg, Alexander C Berg, Jaety Edwards, and DA Forsyth. Who's in the picture. *Advances in neural information processing systems*, 17:137–144, 2005.