SGN-41006 Signal Interpretation

Exercise Set 1: January 13 - 14, 2016

Exercises consist of both pen&paper and computer assignments. Pen&paper questions are solved at home before exercises, while computer assignments are solved during exercise hours. The computer assignments are marked by text **python**

1. **python** *Load CSV file into Python.*

   Download the following file and extract the contents:

   `http://www.cs.tut.fi/courses/SGN-41006/exercises/locationData.zip`

   a) Read the file into numpy array using `numpy.loadtxt` function. Search for instructions using Google.

   b) When loaded, check the array shape using `numpy.shape`. It should be 600×3.

2. **python** *Plot the contents of the loaded matrix.*

   a) Create a 2D plot of the first two columns of the matrix. You need to import `matplotlib.pyplot` and state something like

   ```
   plt.plot(<column 1>, <column 2>)
   ```

   b) Create a 3D plot of all 3 columns. You will need to create a special subplot for this purpose with

   ```
   ax = plt.subplot(1, 1, 1, projection = "3d")
   ```

   After this, the plotting is done as

   ```
   plt.plot(<column 1>, <column 2>, <column 3>)
   ```

3. **python** *Basic image manipulation routines in Python.*

   a) Download the following file to your local folder:

   `http://www.cs.tut.fi/courses/SGN-41006/exercises/oulu.jpg`

   b) Load the image into numpy array. There are several ways of doing this, such as: `matplotlib.image.imread`, `scipy.ndimage.imread`, `PIL.Image.open` or `cv2.imread`.

   c) Show the image on screen (`matplotlib.pyplot.imshow`).

   d) Print the image shape.

   e) Compute the mean of the whole image.

   f) Compute the means of the three color channels (R, G, B).

g) Apply the white tophat operator to the image (`scipy.ndimage.morphology.white_tophat`) and show the result on screen.

4. **python** *Load Matlab data into Python.*

a) Download the following file to your local folder:

`http://www.cs.tut.fi/courses/SGN-41006/exercises/twoClassData.mat`

b) Load the file contents into Python. This can be done as follows.

```
>>> from scipy.io import loadmat
>>> mat = loadmat("twoClassData.mat")
```

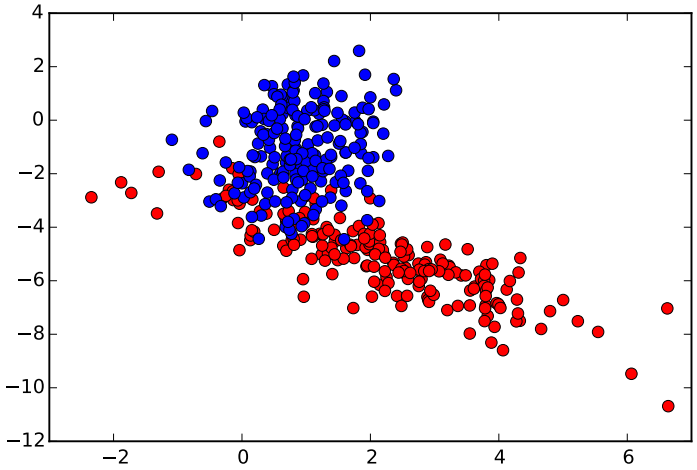
This generates a **dict** structure, whose elements can be accessed through their names.

```
>>> print(mat.keys()) # Which variables mat contains?
['y', 'X', '__version__', '__header__', '__globals__']
>>> X = mat["X"] # Collect the two variables.
>>> y = mat["y"].ravel()
```

The function `ravel()` transforms y from 400×1 matrix into a 400-length array. In Python these are different things unlike Matlab.

c) The matrix X contains two-dimensional samples from two classes, as defined by y. Plot the data as a scatter plot like the picture below. Hints:

- You can access all class 0 samples from X as: `X[y == 0, :]`.
- The samples can be plotted like: `plt.plot(X[:, 0], X[:, 1], 'ro')`



5. **python** *Define a function.*

   a) Implement a function that normalizes the data of question 4 by subtracting the mean and dividing by the standard deviation for each column separately. The function call should be:

   ```
   X_norm = normalize_data(X)
   ```

   b) Test your code with the following lines

   ```
   X_norm = normalize_data(X)
   np.mean(X_norm, axis = 0) # Should be 0
   np.std(X_norm, axis = 0)  # Should be 1
   ```