SGN-41007 Pattern Recognition and Machine Learning

Exercise Set 3: January 21–January 25, 2019

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`** and Pen&paper questions by text **`pen&paper`**

1. **`pen&paper`** *Design an optimal detector for step signal.*

   The lecture slides describe an optimal detector for a known waveform $s[n]$. Apply it to design the optimal detector for a step edge:

   $$s[n] = \begin{cases} -1, & \text{for } 0 \leq n < 10 \\ 1, & \text{for } 10 \leq n < 20 \end{cases}$$

   Simplify the expression as far as you can.

2. **`pen&paper`** *ROC and AUC.*

   A probabilistic classifier is used for classifying four test samples into two classes. As a result, we get the following prediction scores:

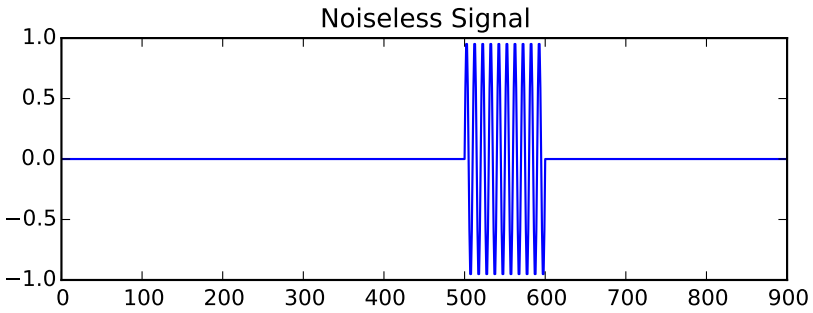
	Probability of class 1	**True label**
Sample 1	0.8	1
Sample 2	0.3	1
Sample 3	0.4	0
Sample 4	0.2	0

   Draw the receiver operating characteristic curve. What is the Area Under Curve (AUC) score?

3. **`python`** *Implement a sinusoid detector.*

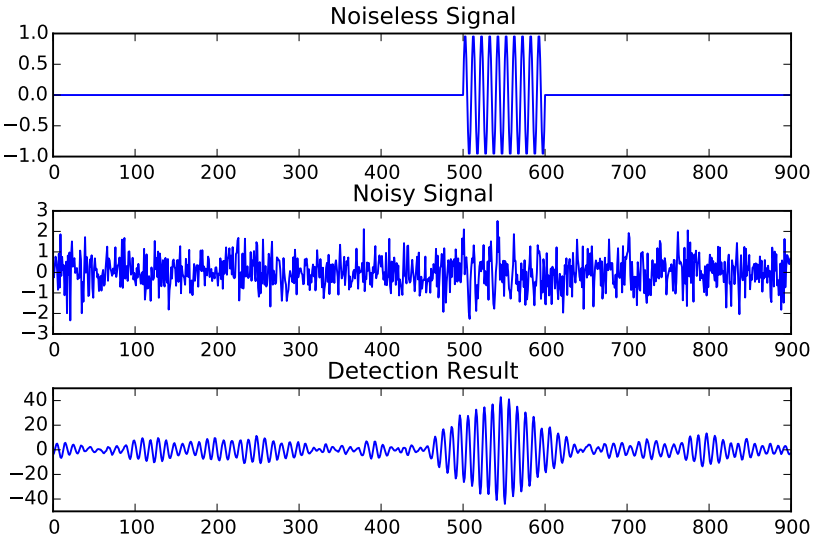
   In this exercise we generate a noisy sinusoid with known frequency and see how the sinusoid detector of the lecture slides performs.

   a) Create a vector of zero and sinusoidal components that looks like the plot below. Commands: `np.zeros`, `np.concatenate`. Sinusoid is generated by `np.cos(2 * np.pi * 0.1 * n)`.



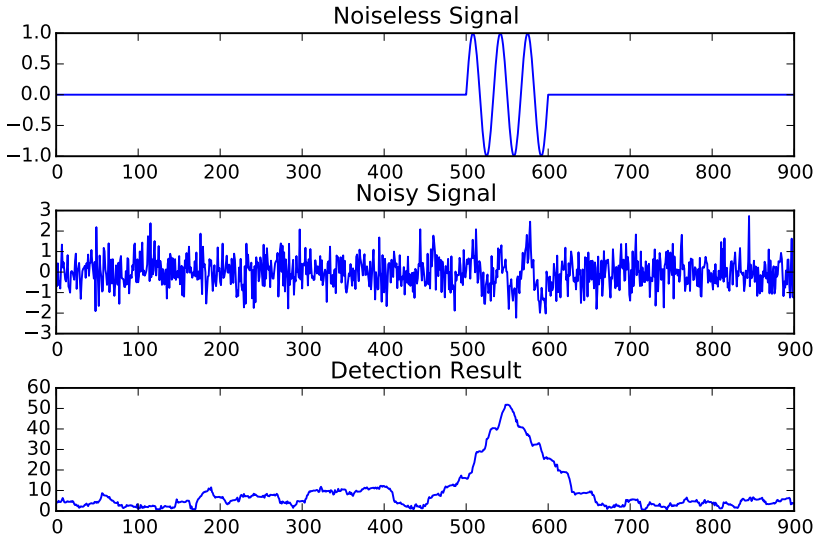
b) Create a noisy version of the signal by adding Gaussian noise with variance 0.5: `y_n = y + np.sqrt(0.5) * np.random.randn(y.size)`.

c) Implement the two detectors and reproduce the below plot.



4. **python** *Same as previous but different frequency and detector.*

Change the code of the previous exercise such that the frequency is 0.03 and the detector is the random signal version.



5. **python** *Train your first sklearn classifiers.*

In this exercise we will train a two classifiers and compare their performance. Before you start, load the following dataset (or find it from your disk—we used the same data in week 1):

`http://www.cs.tut.fi/courses/SGN-41007/Ex1_data.zip`

a) Load the file `twoClassData.mat` to your python workspace.

b) Split the data into training and testing sets: samples `X[:200]` are for training and `X[200:]` for testing.

c) Train a **KNN classifier**. Use default parameters and compute the accuracy using `sklearn.metrics.accuracy_score` on the test set.

d) Train an **LDA classifier**. Use default parameters and compute the accuracy using `sklearn.metrics.accuracy_score` on the test set.