Numerical problems must show work to receive credit. Programs must be well commented.

Problem Set 1

Problems 1-5 are 20 points each. Problem 5 is 80 points.

1. Take the SDSU plagiarism tutorial. Turn in your quiz results. You may retake it if you desire, the point is to learn the material.
2. A weighted coin comes up heads 55% of the time. Treating heads as the value 0 and tails as the value 1, compute the mean $\mu = E[X]$ and variance $\sigma^2 = E[(X-\mu)^2]$.
3. Why do we measure intensity in decibels as opposed to a linear scale?
4. A command and control system is designed to recognize the words: left, right, and forward. A frequentist analysis of 1000 commands shows the most common command is forward, used 50% of the time with left and right being used nearly equally, 26% for left and 24% for right. A user speaks a word and the system produces the following class-conditional probabilities: .45 for forward, .5 for left and .2 for right. According to Bayes decision rule, which class should be chosen to minimize the error?
5. Record a short whistle of constant frequency and save it as a wave file. Use Scientific Python's scipy.io.wavfile package to read the file. Window the data with a Hamming window and the magnitude squared frequency domain representation (see example in slides using numpy's DFT). Show your code and the plot. Be sure to label your axes and write a figure caption for the plot. You do not need to submit an electronic copy of this problem.
6. Write a Python program to process audio data. You will write driver function and two classes designed to stream audio. The first class, AudioFrames, takes a constructor containing a filename, and framing parameters giving the frame advance and length in milliseconds. Iterating over the class will produce a series of frames. The second class is RMSStream and takes an instance of AudioFrames as its constructor. Iterating over it produces RMS intensity values in dB. Write a function driver (driver.py) that reads the provided audio file "shaken.wav," and produces a time-intensity plot of RMS energy using 20 ms frames advanced every 10 ms.

   A skeleton of the code can be found on Blackboard. Preserve the interfaces as your code will be automatically tested for correctness.

   There are a number of functions that will be useful to you:

   - matplotlib module: It is traditional to import matplotlib.pyplot as plt. Several plt functions will be of interest to you: plot, xlabel, and ylabel.

- Use Scientific Python's scipy.io.wavfile package to read the audio data. Note that this function returns a numpy array which is an efficient representation of data with a large number of available operations.

  If you are unable to import scipy.io.wavfile, you have a problem with your installation. See the instructions for installing Anaconda and additional required packages on the course web page (materials tab).

- Use the numpy module's operations when computing RMS (typically imported as np). You may also find np.asarray and np.double useful.

What to turn in:

In addition to the written problems:

- turn in hardcopy of:
  - your code.
  - output of a run (you might not produce any output in this program other than the plot)
  - the requested figure with captions
- Submit to blackboard a zip or tar archive of your code. The top level of the archive should contain your driver program and the mydsp module. As a reminder, unit tests will be used to check functionality. If you do not meet the specified interface, your code will not function.

  Why do I have to turn in hardcopy and electronic copies of my code? Electronic copies let us test your code. Printed copies allow us to provide written feedback.