Assignment 03

Machine Learning, Summer term 2018

Norman Hendrich, Marc Bestmann, Philipp Ruppel

April 22, 2018

Solutions due by April 29

Assignment 03.1 (Random numbers, 1+1+1+1+2 points)

Generated datasets based on known distributions are often the best way to test and understand new algorithms. Numpy offers a wide range of functions to generate and work with random numbers.

a. Read the documentation for the *numpy.random* functions.

   Create arrays of $n \in [100, 1000, 10000, 100\,000]$ random numbers with uniform distribution. Plot the raw data, then generate and plot histograms with 10 bins. How do the mean, minimum and maximum values of the bins (occupation counts) behave?

b. Create random numbers from a Gaussian distribution with mean μ and variance σ^2. Plot the raw data, then generate and plot histograms.

c. As before, but using the Binomial distribution with parameters n and p.

d. Maybe combining multiple random numbers is even better than using single ones?

   Use numpy to generate new random numbers from a sum of individual numbers, $s_i = \sum_{j=1}^{M} r_j$, where the r_j are generated from a uniform distribution. Plot scatter plots and histograms of the resulting data sets for $M \in [2, 3, 5, 10, 20]$.

e. Generate random numbers with a uniform distribution in a circle of radius r. (Recent versions of numpy actually have a function for this, but the goal here is to understand the issue first and then to come up with your own solution.)

Assignment 03.2 (Linear Mapping, 1+1+1+1+1 points)

Load the data from *Adot.mat*. Each column of matrix X represents on data point.

a. Use the function *scipy.io.loadmat* to parse and load the Matlab/Octave *.mat* data file, then access the array(s) inside the data structures.

b. Create a numpy matrix for the linear mapping V:

```
theta = pi/3
V = [[cos(theta), -sin(theta)], [sin(theta), cos(theta)]]
```

   Apply the linear mapping on X to get $Y = VX$. Plot both X and Y in the same figure. What does the linear mapping V do?

b. Now apply the transpose of the linear mapping on Y to get $Z = V^t Y$. Plot Z and describe what the linear mapping $V^t V$ does.

c. What do the linear mappings $D1 = [20; 02]$ and $D2 = [20; 01]$ (Matlab matrix notation) do? Apply them on X and plot the results.

d. What does the linear mapping $A = V^t * D2 * V$ do? Apply it on X and plot the result.

Assignment 03.3 (Digit classification, 2+1+2+2 points)

In this exercise, we use a kNN classifier to classify handwritten digits from the USPS data-set. You can reuse your kNN classifier from Assignment 2 or use libraries from Scikit. The USPS data-set contains grayscale handwritten digit images scanned from envelopes by the U.S. Postal Service. The images are of size 16×16 (256 pixel) with pixel values in the range 0 to 255. We have 10 classes $\{1, 2, ..., 9, 0\}$. The training data has 10000 images, stored in a 10000×256 Matlab matrix (usps train.mat). The training label is a 10000×1 vector revealing the labels for the training data. There are 1000 test images for evaluating your algorithm in the test data (usps test.mat).

a. First, we want to classify digit 2 versus digit 3. Prepare the training data: Load the train data (*scipy.io.loadmat*) and prepare the training set for classes 2 and 3. We need to convert the data type from uint8 (8-bit unsigned integer) to double. Do the same for the test data.

b. Plot a few example images using *matplotlib.pyplot.imshow* and the grayscale colormap (*cmap='grey'*). Use *reshape* to convert the image vectors into 16×16 images.

c. Evaluate the performance of your classifier: Test your classifier with different values $k = 1, 3, 5, 7, 10, 15$ and plot the training and the test errors.

d. Now you can classify other digits. Run your algorithm to classify digit 3 from 8 and compare its performance with results from digit 2 versus 3.