EE 240: Pattern Recognition and Machine Learning
Homework 1

Due date: April 20, 2018

Description: These questions will explore different aspects of Bayes theorem, maximum likelihood estimation, and linear regression.

Reading assignment: AML: Ch. 1 & 5, CIML: Ch. 1–3, ESL Ch. 1–2

> **Homework and lab assignment submission policy:**
> All homework and lab assignments must be submitted online via https://iLearn.ucr.edu.
> Homework solutions should be written and submitted individually, but discussions among students are encouraged.
> All assignments should be submitted by the due date. There will be 25% penalty per day for late assignments. No grade will be given to homework submitted 3 days after the due date.

H1.1 Probability paradoxes.

(a) Suppose we are given N fair coins and we toss each of them T times. What is the probability that some coin out of N that has no head in T coin tosses? Remember the probability of no heads with $N = 1000, T = 10$ is ≈ 0.623. **(10 pts)**

(b) Let's assume that we have 35 students in EE240 class. What is the probability that two randomly chosen students have birthday on the same day. (Assume that everyone was born in a year with 365 days, i.e., a non-leap year.) **(10 pts)**

H1.2 **Exercise 1.10 in AMLbook**: Here is an experiment that illustrates the difference between a single bin and multiple bins. Run a computer simulation for flipping 1,000 fair coins. Flip each coin independently 10 times. Let's focus on 3 coins as follows. c_1 is the first coin flipped; c_{rand} is a coin you choose at random; c_{min} is the coin that had the minimum frequency of heads (pick the earlier one in case of a tie). Let ν_1, ν_{rand}, and ν_{min} be the fraction of heads you obtain for the respective three coins. For a coin, let μ be its probability of heads. **(20 pts)**

(a) What is μ for the three coins selected?

(b) Repeat this entire experiment a large number of times (e.g., 100,000 runs of the entire experiment) to get several instances of ν_1, ν_{rand}, and ν_{min} and plot the histograms of the distributions of ν_1, ν_{rand}, and ν_{min}. Notice that which coins end up being c_{rand} and c_{min} may differ from one run to another.

(c) Using part 2b plot estimates for $\mathbb{P}[|\nu - \mu| > \epsilon]$ as a function of ϵ, together with the Hoeffding bound $2^{-2\epsilon^2 N}$ on the same graph.

(d) Which coins obey the Hoeffding bound, and which do not? Explain why.

H1.3 Probability paradoxes continued (Monty Hall problem).

(a) Suppose you are on a game show, and you are given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who *knows what's behind* the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice? Justify your answer using Bayes theorem. **(10 pts)**

(b) Simulate this game and show that the probability of winning a car by switching doors matches your answer from the step above. **(10 pts)**

H1.4 Posterior probability estimation for bin selection problem (curse of dimensionality).

(a) Suppose we have ten bins (four labeled A, six labeled B). We randomly select a bin and draw one ball. Estimate the probability that we selected bin A given the ball is red for the following cases: **(10 pts)**

   i. Each bin has balls with two colors (red and blue). The distribution of red and blue balls in bin A is (0.3, 0.7). The distribution of red and blue balls in bin B is (0.7, 0.3).

   ii. Each bin has balls with four colors (red, blue, white, black). The distribution of balls in bin A is (0.1, 0.3, 0.2, 0.4). The distribution of balls in bin B is (0.4, 0.2, 0.3, 0.1).

(b) Suppose we have ten bins (four labeled A, six labeled B). Each bin has balls with two colors (red and blue). The distribution of red and blue balls in bin A is (0.3, 0.7). The distribution of red and blue balls in bin B is (0.7, 0.3). We randomly select a bin and draw two balls with *replacement*. That is, we select a bin, pick one ball, put it back, and pick another ball from the *same* bin. Estimate the probability that we selected bin A given the selected balls are red and blue. **(10 pts)**

(c) Suppose we have ten bins (four labeled A, six labeled B). Each bin has balls with four colors (red, blue, white, black). The distribution of balls in bin A is (0.1, 0.3, 0.2, 0.4). The distribution of balls in bin B is (0.4, 0.2, 0.3, 0.1). We randomly select a bin and draw two balls with replacement. Estimate the probability that we selected bin A given the selected balls are red and blue. **(10 pts)**

H1.5 Let us consider the problem of nearest-mean classifier. Suppose we are given N training samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ from two classes with $y_n \in \{+1, -1\}$. We saw in lecture 2 that we can decide a label for a test vector $\mathbf{x}$ as $g(\mathbf{x}) = \text{sign}\left(\mathbf{w}^T\mathbf{x} + b\right)$, where $\mathbf{w} = 2(\mu_+ - \mu_-)$ and $b = \|\mu_-\|_2^2 - \|\mu_+\|_2^2$. μ_+ is a mean vector for samples in the $+ve$ class and μ_- is a mean vector for samples the $-ve$ class. Show that $\mathbf{w}^T\mathbf{x} + b \equiv \sum_{n=1}^{N} \alpha_n \langle \mathbf{x}_n, \mathbf{x} \rangle + b$ and calculate the values of the α_n. **(10 pts)**

H1.6 K-nearest neighbor classification for MNIST data: In this problem set, we consider the problem of handwritten digit recognition. We will use a subset of the MNIST database, which has become a benchmark for testing a wide range of classification algorithms. See http://yann.lecun.com/exdb/mnist/ if you'd like to read more about it. You may want to import MNIST dataset using sklearn:

The sklearn.datasets package is able to directly download data sets from the repository using the function `sklearn.datasets.fetch_mldata`.

For example, to download the MNIST digit recognition database:

```
>>> from sklearn.datasets import fetch_mldata
>>> mnist = fetch_mldata('MNIST original', data_home=custom_data_home)
```

The MNIST database contains a total of 70000 examples of handwritten digits of size 28x28 pixels, labeled from 0 to 9:

```
>>> mnist.data.shape
(70000, 784)
>>> mnist.target.shape
(70000,)
>>> np.unique(mnist.target)
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9.])
```

In the MNIST database, each training or test example is a 28×28 grayscale image. To ease programming of learning algorithms, these images have been converted to vectors of length $282 = 784$ by sorting the pixels in raster scan (row-by-row) order.

In this question, we explore the performance of K nearest neighbor (K-NN) classifiers at distinguishing handwritten digits. Pick images corresponding to three digits in your student ID, say "1", "2", and

"7". To determine neighborhoods, let's use the Euclidean distance between pairs of vector-encoded digits $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{l=1}^{784} [\mathbf{x}_i(l) - \mathbf{x}_j(l)]^2}.$$

(25 pts)

(a) Implement a function that finds the K nearest neighbors of any given test digit, and classifies it according to a majority vote of their class labels. Construct a training set with 200 examples of each class ($N = 600$ total examples). What is the empirical accuracy (fraction of data classified correctly) of 1-NN and 3-NN classifiers on the test examples from these classes?

(b) Plot 5 test digits that are correctly classified by the 1-NN classifier, and 5 which are incorrectly classified. Do you see any patterns?

H1.7 Linear regression: Implement a solution for Exercise 1A described at the following link using Python: UFLDL linear regression. Data and starter codes in Matlab can be found at Starter Code GitHub Repo. **(25 pts)**

Maximum points: 150