MACHINE LEARNING.
Assignment 4.
Due: December 4, 2017.

1. [25 points] **VC dimension**

   We studied in class the VC dimension for closed intervals. This question requires you to do some analysis in the same style.

   (a) [5 points] Consider the class of hypotheses of the form $[a, \infty)$ where a is a real number. What is the VC dimension for this class?

   (b) [5 points] Consider the class of hypotheses of one-sided intervals. Hypotheses in this class can have the form $(-\infty, a]$ or $[a, \infty)$ where a is a real number. What is the VC dimension for this class?

   (c) [5 points] Consider the class of hypotheses which allows finite unions of one-sided intervals. What is the VC dimension of this class?

   (d) [5 points] Consider the class of hypotheses of the form $[a, b] \cup [c, d]$ where $a, b, c, d \in \Re$. What is the VC dimension of this class?

   (e) [5 points] Consider the class of hypotheses consisting of unions of k intervals. What is the VC dimension of this class?

2. [35 points] **KL-divergence**

   We mentioned in class that the Kullback-Leibler (KL) divergence is a standard way to measure the difference between two probability distributions. In this question, we discuss some of its basic properties, as well as its relationship to maximum likelihood learning.

   The KL-divergence between two distributions of discrete random variables, P and Q is defined as:

$$KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

   Assume, for convenience, that $P(x) > 0, \forall x$

   (a) [5 points] Prove that $KL(P||Q) \geq 0, \forall P, Q$
   Hint: You may want to use Jensen's inequality: if X is a random variable and f is a convex function, then $E[f(X)] \geq f(E[X])$ (where E denotes, as usual, the expectation).

   (b) [5 points] When do we have $KL(P||Q) = 0$?

   (c) [5 points] What is the maximum value possible of $KL(P||Q)$, and when is it achieved?

   (d) [5 points] Is $KL(P||Q) = KL(Q||P)$? Justify your answer.

   (e) [5 points] The KL-divergence between two conditional probability distributions, $P(Y|X)$ and $Q(Y|X)$, is defined as follows:

$$KL(P(Y|X)||Q(Y|X)) = \sum_x P(x) \left(\sum_y P(y|x) \log \frac{P(y|x)}{Q(y|x)} \right)$$

   In other words, this is the expected value of $KL(P(Y|X = x)||Q(Y|X = x))$, with the expectation taken over random variable X. Prove the following rule, which is similar to the conditional probability chain rule:

$$KL(P(X, Y)||Q(X, Y)) = KL(P(X)||Q(X)) + KL(P(Y|X)||Q(Y|X))$$

(f) [10 points] Suppose that we are given a set of instances $x_i, i = 1 \ldots m$. Let $\hat{P}$ be the empirical distribution, based on counts for each value of X in this data. Suppose we are modeling P by a parameterized family of distributions, P_θ. Prove that finding the maximum likelihood value for θ is the same as finding P_θ that has the minimum KL-divergence from $\hat{P}$. That is, prove that:

$$\arg\min_\theta KL(\hat{P}||P_\theta) = \arg\max_\theta \sum_{i=1}^{m} \log P_\theta(x_i)$$

3. [30 points] **K-means**

Consider the data in the file "hw4.txt". This file contains a large number (210,012) of length 3 vectors, each on one line. Each vector represents the red, green, and blue intensity values of one of the pixels in the image shown at the right. The image has 516 rows and 407 columns. The pixels in the file are listed row by row from top to bottom, and within each row from left to right. For example, the first pixel in the file is the uppermost left pixel in the image. The second line of the file contains the pixel to the right of that one, and so on. In this assignment, we will explore clustering methods, applying them in particular to the problem of dividing the pixels of the image into a small number of similar clusters.

Consider the K-means clustering algorithm, as described in class. In particular, consider a version in which the inputs to the algorithm are:

(a) The set of data to be clustered. (I.e., the vectors $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots$)

(b) The desired number of clusters, K.

(c) Initial centroids for the K clusters.

Then the algorithm proceeds by alternating: (1) assigning each instance to the class with the nearest centroid, and (2) recomputing the centroids of each class—until the assignments and centroids stop changing.

There are many implementations of K-means publicly available. However, please implement K-means clustering in MATLAB by yourself. Then, use your implementation to cluster the data in the file mentioned above, using $K = 8$, and the initial centroids:

R	G	B
255	255	255
255	0	0
128	0	0
0	255	0
0	128	0
0	0	255
0	0	128
0	0	0

Turn in your code, as well as a report on all of the following:

- How many clusters there are in the end. (A cluster can "disappear" in one iteration of the algorithm if no vectors or closest to its centroid.)
- The final centroids of each cluster.
- The number of pixels associated to each cluster.
- The sum of squared distances from each pixel to the nearest centroid after every iteration of the algorithm.

Visualize your result by replacing each pixel with the centroid to which it is closest, and displaying the resulting image.

4. [10 points] **K-medoids**

   K-means clustering creates cluster centroids that do not correspond to any real data points. But in some applications, we want the cluster centers to be real data points. In this case, we can use an algorithm called K-medoids. For this algorithm, we choose the number of clusters K, and K initial points to serve as centers. Then, the following two steps are repeated:

   (a) Assign each point to the closest medoid

   (b) For each cluster, consider swapping the current medoid with any other point in the cluster. Pick as new medoid the point that generates the lower cost.

   What are the advantages and disadvantages of K-medoids, compared to K-means?