ECE M146 Homework 4
Introduction to Machine Learning
Instructor: Lara Dolecek
TA: Zehui (Alex) Chen, Ruiyi (John) Wu

Please upload your homework to Gradescope by April 30, 11:59 pm.
Please submit a single PDF directly on Gradescope
You may type your homework or scan your handwritten version. Make sure all the work is discernible.

1. In this section, you will use the k-NN classifier to predict whether a person survives or not on the Titanic. You will be using the same data set provided in HW3.

   Unlike what we did in the last homework, we are going to build the k-NN classifier from the first principle. You may **not** use **fitcknn** (**sklearn.neighbors.KNeighborsClassifier** for python) in this problem as you will get incorrect answer by using those built-in functions.

   The k-NN classifier classifies a data point with feature x_{test} based on a training set by performing the following procedures:

   - Compute the distance from x_{test} to the feature of all training points. We will use the Euclidean distance in this problem.
   - Find the k nearest neighbors of this point.
   - Classify this points as the majority class of its k nearest neighbors.

   We use the following two rules to handle ties:

   (a) Let d_k be the distance of the k-th nearest neighbor of x_{test}. If there are multiple training points that have distance d_k from x_{test}. Choose those points with the smallest indexes to be included in the k nearest neighbors. For example, let $k = 3$, if there is x_9 that is distance 1 away from x_{test}; x_1, x_3 and x_4 that is distance 2 away from x_{test}, then the 3 nearest neighbor of x_{test} are x_1, x_3 and x_9. Note that $d_k = 2$ in this example.

   (b) For even k, among all k nearest neighbors of a data point, if the number of points from class 0 is the same as the number of points from class 0, classify this data point as y_{tie} deterministically.

   Answer the following questions:

   (a) With $y_{tie} = 1$, implement the k-NN algorithm. Find and plot the training and testing error for $k = 1, 2, \cdots, 15$.

   (b) With $y_{tie} = 0$, implement the k-NN algorithm. Find and plot the training and testing error for $k = 1, 2, \cdots, 15$.

(c) Comment on the performance of the k-NN classifier in (a) and (b). How does larger k affect the training and testing error? How does even or odd k affect the performance of the k-NN classifier in (a) and (b), respectively? Are they contradictory to each other? Explain why. Hint: you may use your result from HW3 Q6 (a) to get some intuition.

2. In this section, you will consider a k-nearest neighbor classifier using Euclidean distance metric on a binary classification task. We assign the class of the test point to be the class of the majority of the k nearest neighbors.

   Consider the following two datasets:

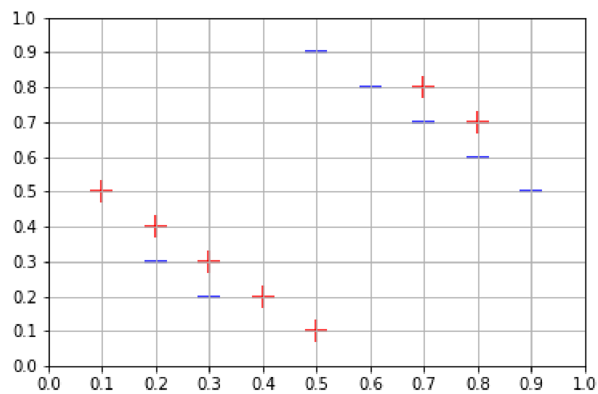

Table 1: KNN Example 1

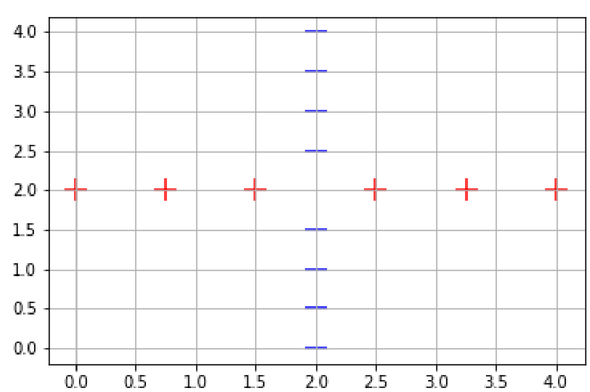

Table 2: KNN Example 2

   (a) For $k \in \{1, 3, 5, 7\}$, what values of k minimize leave-one-out cross-validation error for each dataset? What is the resulting validation error?

   (b) In general, what are the drawbacks of using too big a k? What about too small a k? To see this, calculate the leave-one-out cross-validation error for the dataset in Figure 1 using $k = 1$ and $k = 13$.

3. In this problem, we will derive the least square solution for multi-class classification. Consider a general classification problem with K classes, with a 1-of-K binary encoding scheme (defined latter) for the target vector $t, t \in \mathbb{R}^K$. Suppose we are given a training data set $\{x_n, t_n\}, n = 1, \cdots, n$ where $x_n \in \mathbb{R}^D$. For the 1-of-K binary encoding scheme, t_n has the k-th element being 1 and all other elements being 0 if the n-th data is in class k. We can use the following linear model to describe each class:

$$y_k(x) = w_k^T x + w_{k0},$$

where $k = 1, \cdots, K$. We can conveniently group these together using vector notation so that

$$y(x) = \tilde{W}^T \tilde{x},$$

where $\tilde{W}$ is a matrix whose k-th column comprises the $D+1$-dimensional vector $\tilde{w} = [w_{k0}, w_k^T]^T$ and $\tilde{x}$ is the corresponding augmented input vector $[1, x^T]^T$. For each new input with feature x, we assign it to the class for which the output $y_k = \tilde{w}_k^T \tilde{x}$ is largest. Define a matrix T whose n-th row is the vector t_n^T and together a matrix $\tilde{X}$ whose n-th row is $\tilde{x}_n^T$, the sum-of-squares error function can be written as

$$J(\tilde{W}) = \frac{1}{2} Tr \left\{ (\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T) \right\}.$$

Find the closed form solution of $\tilde{W}$ that minimizes the objective function $J(\tilde{W})$. Hint: You many use the following two matrix derivative about trace, $\frac{\partial}{\partial Z} Tr(AZ) = A^T$ and $\frac{\partial}{\partial Z} Tr(Z^T AZ) = (A^T + A)Z$.

4. You are given the following data set which is comprised of $\boldsymbol{x}^{(i)} \in \mathbb{R}^2$ and $y^{(i)} \in \{-1, 1\}$.

i	$x_1^{(i)}$	$x_2^{(i)}$	$y^{(i)}$
1	-3	9	1
2	-2.5	6.25	1
3	3	9	1
4	-1.5	2.25	-1
5	0	0	-1
6	1	1	-1

(a) Plot the data. Is the data linearly separable?

(b) Look at the data and circle the support vectors by inspection. Find and plot the maximum margin separating hyperplane.

(c) Find the α_i, w and b in

$$h(x) = \text{sign}\left(\sum_{n \in \mathcal{S}} \alpha_n y^{(n)} x^T x^{(n)} + b\right) = \text{sign}\left(w^T x + b\right),$$

where $\mathcal{S}$ is the index set of all support vectors. Do this by solving the dual form of the quadratic program. How is w and b related to your solution in part (b)?

5. In this exercise, we will use MATLAB to solve both the primal and the dual problem of SVM. In *Data.csv*, the first two columns contain feature vectors $x^{(i)} \in \mathbb{R}^2$ and the last column contains the label $y^{(i)} \in \{-1, 1\}$. We will use CVX as the optimization solver in this problem. For help with CVX, refer to the CVX Users' Guide. Attach your code for submission. For Python user, feel free to use the following libraries: math, csv, numpy, matplotlib and cvxpy.

   (a) **Visulization** Use different color to plot data with different labels in the 2-D feature space. Is the data linearly separable?

   (b) **The Primal Problem** Use CVX to solve the primal problem of this form:

   $$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ s.t. \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \cdots, m \end{aligned}$$

   Report w and b. Plot the hyperplane defined by w and b.

   (c) **The Dual Problem** Use CVX to solve the dual problem of this form:

   $$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} a_i a_j \langle x^{(i)}, x^{(j)} \rangle \\ s.t. \quad & 0 \leq \alpha_i, i = 1, \cdots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0. \end{aligned}$$

   Use the resulting a to identify the support vectors on the plot. Report you non-zero $a_i's$. How many support vectors do you have? Circle those support vectors. Note: The latter part of $W(a)$ is in quadratic form, i.e., $a^T P a$. To use CVX, first find P and then use *quad_form(a,P)*. For Python user, you will need to add a small number to the diagonal of P matrix to make cvxpy work. i.e. Run the following code before using cvxpy: "P += 1e-13 * numpy.eye(31)", where 31 is the total number of data. Also, assume it is 0 if a number is less than 1e-9.