Introduction to Machine Learning, Spring 2025
Homework 6

(Due May 25, 2025 at 11:59pm (CST))

May 6, 2025

1. Please write your solutions in English.
2. Submit your solutions to the course Gradescope.
3. If you want to submit a handwritten version, scan it clearly.
4. Late homeworks submitted within 3 days of the due date will be marked down 25% each day cumulatively Homeworks submitted more than 3 days after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
5. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

1. [25 points] [Boosting]

Suppose that we are interested in learning a classifier, such that at any turn of a game we can pose a question, like "should I attack this ant hill now?", and get an answer.That is, we want to build a classifier which we can feed some features on the current game state, and get the output "attack" or "don't attack". There are many possible ways to define what the action "attack" means, but for now let's define it as sending all friendly ants that can see the ant hill under consideration towards it.

Let's recall the AdaBoost algorithm described in class. Its input is a dataset $\{(x_i, y_i)\}_{i=1}^n$, with x_i being the i-th sample, and $y_i \in \{-1, 1\}$ denoting the i-th label, $i = 1, 2, ..., n$. The features might be composed of a count of the number of friendly ants that can see the ant hill under consideration, and a count of the number of enemy ants these friendly ants can see. For example, if there were 10 friendly ants that could see a particular ant hill, and 5 enemy ants that the friendly ants could see, we would have:

$$x_1 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

The label of the example x_1 is $y_1 = 1$, once the friendly ants were successful in razing the enemy ant hill, and $y_1 = -1$ otherwise. We could generate such examples by running a greedy bot (or any other opponent bot) against a bot that we periodically try to attack an enemy ant hill. Each time this bot tries the attack, we record (say, after 20 turns or some other significant amount of time) whether the attack was successful or not.

(a) Let ϵ_t denote the error of a weak classifier h_t:

$$\epsilon_t = \sum_{i=1}^{n} D_t(i)\mathbb{I}(y_i \neq h_t(x_i))$$

In the simple "attack" / "don't attack" scenario, suppose that we have implemented the following six weak classifiers:

$$\begin{aligned} h^{(1)}(x_i) &= 2 * \mathbb{I}(x_{i1} \geq 2) - 1, & h^{(4)}(x_i) &= 2 * \mathbb{I}(x_{i2} \leq 2) - 1, \\ h^{(2)}(x_i) &= 2 * \mathbb{I}(x_{i1} \geq 6) - 1, & h^{(5)}(x_i) &= 2 * \mathbb{I}(x_{i2} \leq 6) - 1, \\ h^{(3)}(x_i) &= 2 * \mathbb{I}(x_{i1} \geq 10) - 1, & h^{(6)}(x_i) &= 2 * \mathbb{I}(x_{i2} \leq 10) - 1. \end{aligned}$$

Given ten training data points ($n = 10$) as shown in Table 1:

i	x_{i1}	x_{i2}	y_i
1	1.5	0.5	1
2	2.5	1.5	1
3	3.5	3.5	1
4	6.5	5.5	1
5	7.5	10.5	1
6	1.5	2.5	-1
7	3.5	1.5	-1
8	5.5	5.5	-1
9	7.5	8.5	-1
10	1.5	10.5	-1

Table 1: The training data in (a).

please show that what is the minimum value of ϵ_1 and which of $h^{(1)}, \cdots, h^{(6)}$ achieve this value? Note that there may be multiple classifiers that all have the same ϵ_1. You should list all classifiers that achieve the minimum ϵ_1 value. [5 points]

(b) For all the questions in the remainder of this section, let h_1 denote $h^{(1)}$ chosen in the first round of boosting. (That is, $h^{(1)}$ was the classifier that achieved the minimum ϵ_1.)

   (1) What is the value of α_1 (the weight of this first classifier h_1)? [2 points]

   (2) What should Z_t be in order to make sure the distribution D_{t+1} is normalized correctly? That is, derive the formula of Z_t in terms of ϵ_t that will ensure $\sum_{i=1}^{n} D_{t+1}(i) = 1$. Please also derive the formula of α_t in terms of ϵ_t. [5 points]

(3) Which points will increase in significance in the second round of boosting? That is, for which points will we have $D_1(i) < D_2(i)$? What are the values of D_2 for these points? [5 points]

(4) In the second round of boosting, the weights on the points will be different, and thus the error ϵ_2 will also be different. Which of $h^{(1)}, \cdots, h^{(6)}$ will minimize ϵ_2? (Which classifier will be selected as the second weak classifier h_2?) What is its value of ϵ_2? [5 points]

(5) What will the average error of the final classifier H be, if we stop after these two rounds of boosting? That is, if $H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x))$, what will the training error $\epsilon = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq H(x_i))$ be? Is this more, less, or the same as the error we would get, if we just used one of the weak classifiers instead of this final classifier H [3 points]

Solution

2. [10 points] [Equivalence of PCA objectives]

   Consider a dataset of n observations $\mathbf{X} \in \mathbb{R}^{n \times d}$, and our goal is to project the data onto a subspace having dimensionality p, $p < d$. Prove that PCA based on projected variance maximization is equivalent to PCA based on projected error (Euclidean error) minimization.

   Solution

3. [15 points] [Performing PCA by Hand]

Let's do principal components analysis (PCA)! Consider this sample of six points $X_i \in \mathbb{R}^2$.

$$\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\}$$

(a) Compute the mean of the sample points and write the centered design matrix $\dot{X}$. [4 points] (Hint: The sample mean is by subtracting the mean from each sample.)

(b) Find all the principal components of this sample. Write them as unit vectors. [5 points] (Hint: The principal components of our dataset are the eigenvectors of the matrix $\dot{X}^\top \dot{X} =$. The characteristic polynomial of this symmetric matrix is $\det\left(\lambda I - \dot{X}^\top \dot{X}\right)$.)

(c) Which of those two principal components would be preferred if you use only one? [2 points]
What information does the PCA algorithm use to decide that one principal components is better than another? [2 points]
From an optimization point of view, why do we prefer that one? [2 points]

Solution