Optimization and Machine Learning, Spring 2021
Reference Solutions for Homework 4

1. [10 points] Hoeffding's inequality is a powerful technique—perhaps the most important inequality in learning theory for bounding the probability that sums of bounded random variables are too large or too small. Below are some related inequalities you are required to provide proof:

   (a) **(Markov's inequality).** Let $Z \geq 0$ be a non-negative random variable. Then for all $t \geq 0$, show that

   $$\mathbb{P}(Z \geq t) \leq \frac{\mathbb{E}(Z)}{t}, \tag{1}$$

   where $\mathbb{E}$ denotes the expectation operator. [5 points]

   (b) **(Chebyshev's inequality).** Let $Z \geq 0$ be a random variable with $\mathrm{Var}(Z) < \infty$. Show that

   $$\mathbb{P}(Z \geq \mathbb{E}[Z] + t \text{ or } Z \leq \mathbb{E}[Z] - t) \leq \frac{\mathrm{Var}(Z)}{t^2}, \qquad \text{for } t \geq 0, \tag{2}$$

   where $\mathrm{Var}(Z)$ denotes the variance of Z. [5 points]

   Solution:

   (a) *Proof.* We note that $\mathbb{P}(Z \geq t) = \mathbb{E}[\mathbf{1}\{Z \geq t\}]$, and that if $Z \geq t$, then it must be the case that $Z/t \geq 1 \geq \mathbf{1}\{Z \geq t\}$, while of $Z < t$, then we still have $Z/t \geq 0 = \mathbf{1}\{Z \geq t\}$. Thus

   $$\mathbb{P}(Z \geq t) = \mathbb{E}[\mathbf{1}\{Z \geq t\}] \leq \mathbb{E}\left[\frac{Z}{t} = \frac{\mathbb{E}[Z]}{t}\right],$$

   as desired. □

   (b) *Proof.* The result is an immediate consequence of Markov's inequality. We note that either $Z \geq \mathbb{E}(Z)+t$ or $Z \leq \mathbb{E}[Z] - t$, we have $(Z - \mathbb{E}(Z))^2 \geq t^2$. Thus,

   $$\begin{aligned}\mathbb{P}(Z \geq \mathbb{E}[Z] + t \text{ or } Z \leq \mathbb{E}[Z] - t) &= \mathbb{P}((Z - \mathbb{E}(Z))^2 \geq t^2) \\ &\leq \frac{\mathbb{E}[(Z - \mathbb{E}(Z))^2]}{t^2} = \frac{\mathrm{Var}(Z)}{t^2},\end{aligned}$$

   where the inequality holds due to the Markov's inequality. □

2. [10 points] Recall that to show VC dimension is d for hypotheses $\mathcal{H}$ can be done via showing that VC dim$(\mathcal{H}) \leq d$ and VC dim$(\mathcal{H}) \geq d$. More specifically, to prove that VC dim$(\mathcal{H}) \geq d$ it suffices to give d examples that can be shattered; to prove VC dim$(\mathcal{H}) \leq d$ one must show that no set $d+1$ examples can be shattered.

   For each one of the following function classes, find the VC dimension. State your reasoning based on the presented hint above. (Note that: solutions with the correct answer but without adequate explanation will not earn marks.)

   (a) **Halfspaces in** $\mathbb{R}^2$. Examples lying in or on the halfspace are labeled $+1$, and the remaining examples are labeled -1. [2 points]

   (b) **Axis-parallel rectangles in** $\mathbb{R}^2$. Points lying on or inside the target rectangle are labeled $+1$, and points lying outside the target rectangle are labeled -1. [2 points]

   (c) **Closed sets in** $\mathbb{R}^2$. All points lying in the set or on the boundary of the set are labeled $+1$, and all points lying outside the set are labeled -1. [3 points]

   (d) How many training examples suffice to assure with probability 0.9 that a consistent learner using the function classes presented in (b) will learn the target function with accuracy of at least 0.95? [3 points] (Hint: we use the following bounds on sample complexity: $m \geq \frac{1}{\epsilon}(4\log_2(2/\delta)+8\text{VC dim}(\mathcal{H})\log_2(13/\epsilon))$).

   Solution:

   (a) It is easily shown that any three non-collinear points (e.g., $(0,1)$, $(0,0)$, $(1,0)$) are shattered by $\mathcal{H}$. Thus, VC dim$(\mathcal{H}) \geq 3$. We now show that no set of size four can be shattered by $\mathcal{H}$. If at least three of the points are collinear then there is no halfspace that contains the two extreme points but does not contain the middle points. Thus the four points cannot be shattered if any three are collinear. Next, suppose that the points form a quadrilateral. There is no halfspace which labels one pair of diagonally opposite points positive and the other pair of diagonally opposite points negative. The final case is that one point p is in the triangle defined by the other three. In this case there is no halfspace which labels p differently from the other three. Thus clearly the four points cannot be shattered. Therefore we have demonstrated that VC dim$(\mathcal{H}) = 3$.

   (b) First, it is easily seen that there is a set of four points (e.g., $(0,1)$, $(0,-1)$, $(1,0)$, $(-1,0)$) that can be shattered. Thus VC dim$(\mathcal{H}) \geq 4$. We now argue that no set of five points can be shattered. The smallest bounding axis-parallel rectangles defined by the five points is in fact defined by at most four of the points. For p a non-defining point in the set, we see that the set cannot be shattered since it is not possible for p to be classified as negative while also classifying the others as positive. Thus VC dim$(\mathcal{H}) = 4$.

   (c) Any set can be shattered by $\mathcal{H}$, since a closed set can assume any shape in $\mathbb{R}^n$. Thus, the largest set that can be shattered by $\mathcal{H}$ is infinite, and hence VC dim$(\mathcal{H}) = \infty$.

   (d) The bound is $m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8\text{VC dim}(\mathcal{H})\log_2(13/\epsilon))$. Then just by plugging in the numbers (VC dim$(\mathcal{H}) = 4$, $\delta = 0.1$ and $\epsilon = 0.05$), we have $m \geq 5480$.

3. [20 points] 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 $y_1 = 0$ otherwise. We could generate such examples by running a greedy bot (or any other opponent bot) against a bot that we make 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:

$$\epsilon_t = \sum_{i=1}^{n} D_i^{(t)} \mathbb{1}(y_i \neq h_t(x_i)).$$

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

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

   Given ten training data points ($n = 10$) as shown in Fig. 1, please show that what is the minimum value

i	x_{i1}	x_{i2}	y_i
1	3.5	0.5	1
2	2.5	1.5	1
3	5.5	5.5	1
4	7.5	7.5	1
5	6.5	9.5	1
6	1.5	2.5	-1
7	4.5	4.5	-1
8	3.5	7.5	-1
9	1.5	11.5	-1
10	6.5	11.5	-1

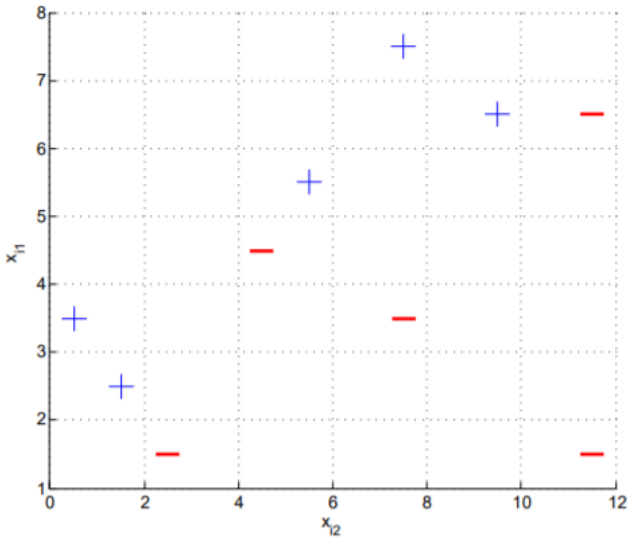

Figure 1: The training data in (a).

   of ϵ_1 and which of $h^{(1)}, ..., h^{(6)}$ 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. [4 points]
   Solution:
   The value of ϵ_1 for each of the classifiers is: $3/10, 3/10, 5/10, 3/10, 5/10$, and $3/10$. So, the minimum value is $3/10$ and classifiers 1, 2, 4, and 6 achieve this value.

(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)? Keep in mind that the log in the formula for α_t is a natural log (base e). [3 points]
Solution:
Plugging into the formula for α we get: $\alpha_1 = \frac{1}{2}\log\frac{1-\epsilon_1}{\epsilon_1} = \frac{1}{2}\log\frac{7}{3} = 0.4236$

(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 $D^{(t)}$, α_t, h_t, and $\{(x_i, y_i)\}_{i=1}^n$, that will ensure $\sum_{i=1}^n D_i^{(t+1)} = 1$. [3 points]
Solution:

$$Z_t = \sum_{k=1}^{n} D_T(k)\exp(-\alpha_t y_k h_t(x_k))$$

(3) Which points will increase in significance in the second round of boosting? That is, for which points will we have $D_i^{(1)} < D_i^{(2)}$? What are the values of $D^{(2)}$ for these points? [3 points]
Solution:
The points that $h^(1)$ misclassifies will increase in weight. These are the points $i = 7, 8, 10$ from the data table. Their new weight under D_2 will be:

$$\begin{aligned} D_2(i) &= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i)}{Z_t} \\ &= \frac{\exp\{0.4236\}}{3 * \exp\{0.4236\} + 7 * \exp\{-0.4236\}} \\ &= \frac{1}{6} \end{aligned}$$

(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 $h^{(1)}, ..., h^{(6)}$ will minimize ϵ_2? (Which classifier will be selected as the second weak classifier h_2?) What is its value of ϵ_2? [3 points]
Solution:
$h^{(4)}$ will be chosen.

Classifier	ϵ_2
$h^{(1)}$	$1/2$
$h^{(2)}$	$1/6 + 2/14 = 13/42$
$h^{(3)}$	$5/14 = 0.3571$
$h^{(4)}$	$3/14$
$h^{(5)}$	$1/6 + 4/14 = 19/42$
$h^{(6)}$	$2/6 + 1/14 = 17/42$

(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 $\epsilon = \frac{1}{n}\sum_{i=1}^n \mathbb{1}(y_i \neq h(x_i))$ 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? [4 points]
Solution:
The classifier after two rounds is:

$$h(x) = sign(0.5\log(7/3)h^{(1)}(x) + 0.5\log(11/3)h^{(4)}(x))$$

Since $log(11/3) > log(7/3)$ the classifier h will always go with the guess made by $h^(4)$. So, it will not do any better than the error we could get using a single weak classifier, $\epsilon = 3/10$. More rounds of boosting are necessary before the interplay of specific settings of the α becomes relevant and allows us to do better than a single weak classifier.

4. [10 points] Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, please verify the following new kernels will also be valid:

   (a) $k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')$, where $f(\cdot)$ is any function. [2 points]

   (b) $k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$, where $q(\cdot)$ is a polynomial with nonnegative coefficients. [2 points]

   (c) $k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}'))$. [3 points]

   (d) $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}'$, where $\mathbf{A}$ is a symmetric positive semi-definite matrix. [3 points]

   Solution:

   (a) Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, there must exist a feature vector $\phi(\mathbf{x})$ such that

   $$k_1(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}').$$

   Then we can rewrite the given kernel as

   $$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) \phi(\mathbf{x})^\top \phi(\mathbf{x}') f(\mathbf{x}') \\ &= \mathbf{v}(\mathbf{x})^\top \mathbf{v}(\mathbf{x}'), \end{aligned}$$

   where $\mathbf{v}(\mathbf{x}) \triangleq f(\mathbf{x}) \phi(\mathbf{x})$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.

   (b) Suppose $q(x) = \sum_{i=1}^n a_n x^n, \forall a_n \geq 0$, then the kernel can be expressed as

   $$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n.$$

   We focus on the i-th term of the kernel, which is $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$. Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, the product of kernels is also a valid kernel. Hence, $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$ is a valid kernel. With the fact that the sum of kernels is a valid kernel, the original kernel is valid.

   (c) Let $\mathbf{K}$ be the Gram matrix. The (i, j)-th entry of $\mathbf{K}$ is defined by $\mathbf{K}_{i,j} \triangleq k(\mathbf{x}_i, \mathbf{x}_j)$. Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, we have $k_1(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}', \mathbf{x})$. Hence, the Gram matrix $\mathbf{K}$ is symmetric. In addition, $k(\mathbf{x}, \mathbf{x}')$ is an exponential function, which leads to $k(\mathbf{x}, \mathbf{x}')$ is always greater than zero. Therefore, the Gram matrix $\mathbf{K}$ is positive definite. Applying, Mercer's condition, $k(\mathbf{x}, \mathbf{x}')$ is a valid kernel.

   (d) Since $\mathbf{A}$ is a symmetric positive semi-definite matrix, we can decompose $\mathbf{A}$ as $\mathbf{A} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix. When $\mathbf{A}$ is positive semi-definite, the entries of $\mathbf{\Lambda}$ are nonnegative. Hence, we can rewrite the kernel as

   $$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^\top \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top \mathbf{x}' \\ &= (\mathbf{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x})^\top (\mathbf{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x}') \\ &= \mathbf{\Phi}(\mathbf{x})^\top \mathbf{\Phi}(\mathbf{x}'), \end{aligned}$$

   where $\mathbf{\Phi}(\mathbf{x}) \triangleq \mathbf{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x}$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.