HOMEWORK 7

>>NAME HERE<<
>>ID HERE<<

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

1 VC dimension (30 pts)

Let the input $x \in X = \mathbb{R}$. Consider $F = \{f(x) = \text{sgn}(ax^2 + bx + c) : a, b, c \in \mathbb{R}\}$, where $\text{sgn}(z) = 1$ if $z \geq 0$, and 0 otherwise. What is $VC(F)$? Prove it.

2 Verify PAC Bound (30 pts)

Let h be the VC dimension of function family F. For any $\delta > 0$, with probability at least $1 - \delta$ we have

$$R(\hat{f}_S) - \hat{R}_S(\hat{f}_S) \leq 2\sqrt{2\frac{h \log n + h \log \frac{2e}{h} + \log \frac{2}{\delta}}{n}},$$

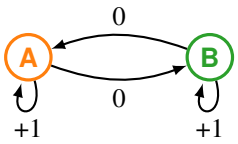
where $R(f) = \mathbb{E}1_{[f(x) \neq y]}$ is the risk of f, $S = (x_1, y_1), \ldots, (x_n, y_n)$ is a training set of size n, $\hat{R}_S(f) = \frac{1}{n}\sum_{i=1}^{n} 1_{[f(x_i) \neq y_i]}$ is the empirical risk of f on S, and $\hat{f}_S \in \arg\min_{f \in F} \hat{R}_S(f)$ is an empirical risk minimizer (ERM).
We now verify this bound on a simple classification task. Let $p(x) = \text{uniform}([-1, 1])$. Given x, the label is deterministic: $y = \text{sgn}(x)$. Recall $\text{sgn}(x) = 1$ if $x \geq 0$, and 0 otherwise. This means the true decision boundary is at $x = 0$. Let $f_\theta(x) := \text{sgn}(x - \theta)$ which has threshold at θ. Let $F = \{f_\theta : \theta \in [-1, 1]\}$.

1. Given S, find the smallest positive item: $a = \min_{(x_i, y_i) \in S : y_i = 1} x_i$ if one exists, otherwise let $a = 1$. Is $\hat{f}_S := f_a$ an ERM? Justify your answer.
2. Derive $R(f_a)$.
3. Derive $\hat{R}_S(f_a)$.
4. What is the VC dimension of F? Note F contains threshold classifiers of the type "left negative, right positive."
5. Fix $\delta = 0.05$ (95% confidence) and $n = 200$. Compute $2\sqrt{2\frac{h \log n + h \log \frac{2e}{h} + \log \frac{2}{\delta}}{n}}$. This is natural log.
6. Generate 10,000 random training sets from $p(x)$ and the associated labels. Each training set S has $n = 200$ points. On each S you will compute $R(f_a) - \hat{R}_S(f_a)$. Now you have 10,000 numbers. (1) Produce a histogram of them. (2) Find the 95% quantile of them. (3) Compare the 95% quantile to the bound in the previous question. Discuss your observations.

3 Q-learning (40 pts)

Consider the following Markov Decision Process. It has two states s. It has two actions a: move and stay. The state transition is deterministic: "move" moves to the other state, while "stay' stays at the current state. The reward

r is 0 for move, 1 for stay. There is a discounting factor $\gamma = 0.9$.



The reinforcement learning agent performs Q-learning. Recall the Q table has entries $Q(s, a)$. The Q table is initialized with all zeros. The agent starts in state $s_1 = A$. In any state s_t, the agent chooses the action a_t according to a behavior policy $a_t = \pi_B(s_t)$. Upon experiencing the next state and reward s_{t+1}, r_t the update is:

$$Q(s_t, a_t) \Leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') \right).$$

Let the step size parameter $\alpha = 0.5$.

1. Run Q-learning for 200 steps with a uniformly random behavior policy: $\pi_B(s_t) =$ move or stay with 1/2 probability for any s_t. Show the Q table at the end.
2. Reset and repeat the above, but with an ϵ-greedy behavior policy: at each state s_t, with probability $1 - \epsilon$ choose what the current Q table says is the best action: $\arg\max_a Q(s_t, a)$; Break ties arbitrarily. Otherwise (with probability ϵ) uniformly chooses between move and stay. Use $\epsilon = 0.5$.
3. Reset and repeat the above, but with a deterministic greedy behavior policy: at each state s_t use the best action $a_t \in \arg\max_a Q(s_t, a)$ indicated by the current Q table. If there is a tie, prefer move.
4. Without doing simulation, use Bellman equation to derive the true Q table induced by the MDP.