TTIC 31250 : Introduction to Theory of Machine Learning

Spring 2020

•

Homework 2

•

Solutions by

Jinhong Du

dujinhong@uchicago.edu

12243476

Exercises

1. **Kernels.** Recall that $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a legal kernel if there exists an implicit function ϕ such that $K(x, x') = \phi(x) \cdot \phi(x')$. (Here, $\mathcal{X}$ is our space of examples, such as $\{0,1\}^n$.)

   Often the easiest way to prove that some function is a legal kernel is to build it out of other legal kernels. In particular, suppose $K_1(x, x') = \phi_1(x) \cdot \phi_1(x')$ and $K_2(x, x') = \phi_2(x) \cdot \phi_2(x')$, where $\phi_1 : \mathcal{X} \mapsto \mathbb{R}^{N_1}$ and $\phi_2 : \mathcal{X} \mapsto \mathbb{R}^{N_2}$ for some N_1, N_2. (Let's not worry about infinite-dimensional implicit feature spaces.)

   (a) Show that for any constant $c \geq 0$, cK_1 is a legal kernel. That is, show how to define ϕ in terms of ϕ_1 such that $\phi(x) \cdot \phi(x') = cK_1(x, x')$.

   Proof. Since $c \geq 0$, $c = \sqrt{c} \cdot \sqrt{c}$. Define $\phi(x) = \sqrt{c}\phi_1(x)$, then

   $$cK_1(x, x') = c\phi_1(x) \cdot \phi_1(x') = \phi(x) \cdot \phi(x').$$

   Thus, cK_1 is a legal kernel. □

   (b) Show that the sum, $K_1 + K_2$, is a legal kernel. That is, show how to define ϕ in terms of ϕ_1 and ϕ_2 such that $\phi(x) \cdot \phi(x') = K_1(x, x') + K_2(x, x')$.

   Proof. Since

   $$K_1(x, x') + K_2(x, x') = \phi_1(x) \cdot \phi_1(x') + \phi_2(x) \cdot \phi_2(x') = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix} \cdot \begin{bmatrix} \phi_1(x') \\ \phi_2(x') \end{bmatrix}$$

   let $\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}$ from $\mathcal{X}$ to $\mathbb{R}^{N_1+N_2}$, we have that $K_1 + K_2$ is a legal kernel. □

   (c) Show that the product, $K = K_1K_2$, is a legal kernel. That is, show how to define ϕ in terms of ϕ_1 and ϕ_2 such that $\phi(x) \cdot \phi(x') = K_1(x, x')K_2(x, x')$. Hint: you will want to have $\phi : \mathcal{X} \mapsto \mathbb{R}^{N_1 N_2}$.

   Proof. From the property of Kronecker product and vectorization, we have

   $$\begin{aligned} K_1(x, x')K_2(x, x') &= [\phi_1(x) \cdot \phi_1(x')] \times [\phi_2(x) \cdot \phi_2(x')] \\ &= \mathrm{vec}(\phi_1(x)^\top \phi_1(x')\phi_2(x')^\top \phi_2(x)) \\ &= (\phi_2(x)^\top \otimes \phi_1(x)^\top)\, \mathrm{vec}(\phi_1(x')\phi_2(x')^\top) \\ &= (\phi_2(x) \otimes \phi_1(x))^\top\, \mathrm{vec}(\phi_1(x')\phi_2(x')^\top). \end{aligned}$$

   Also,

   $$\phi_2(x) \otimes \phi_1(x) = \begin{bmatrix} (\phi_2(x))_1\phi_1(x) \\ \cdots \\ (\phi_2(x))_{N_2}\phi_1(x) \end{bmatrix} = \mathrm{vec}\left(\begin{bmatrix} (\phi_1(x))_1(\phi_2(x))_1 & \cdots & (\phi_1(x))_1(\phi_2(x))_{N_2} \\ \vdots & \ddots & \vdots \\ (\phi_1(x))_{N_1}(\phi_2(x))_1 & \cdots & (\phi_1(x))_{N_1}(\phi_2(x))_{N_2} \end{bmatrix} \right)$$

   which is equal to $\mathrm{vec}(\phi_1(x)\phi_2(x)^\top)$. Define $\phi(x) = \mathrm{vec}(\phi_1(x)\phi_2(x)^\top)$ the vectorization of the outer produt of ϕ_1 and ϕ_2, then $K_1(x, x')K_2(x, x') = \phi(x) \cdot \phi(x')$, i.e. K_1K_2 is a legal kernel. □

   For instance, this implies that $(1 + x \cdot x')^d$ is a legal kernel. (Using the fact that "1" is itself a legal kernel, viewed as a function that always outputs 1).

2. **About δ.** Suppose we changed the definition of PAC-learning to only require an algorithm succeed in finding a low-error hypothesis with probability at least $\frac{1}{2}$ (rather than with probability $1-\delta$). It turns out that this does not change what is learnable. Specifically, suppose we had an algorithm $\mathcal{A}$ with at least a $\frac{1}{2}$ chance of producing a hypothesis of error at most $\frac{\epsilon}{2}$ when given m labeled examples drawn from distribution $\mathcal{D}$ and labeled by a target function $f \in \mathcal{C}$. We can convert such an algorithm into an algorithm $\mathcal{B}$ that has at least a $1-\delta$ probability of producing a hypothesis of error at most ϵ. The reduction is that we first run $\mathcal{A}$ on $N = \log_2 \frac{2}{\delta}$ different sets of m random examples (so with probability at least $1 - \frac{\delta}{2}$, at least one of the N hypotheses produced has error at most $\frac{\epsilon}{2}$). Then we test the N hypotheses produced on a new test set, choosing the one that performs best. Use Chernoff bounds to analyze this second step and finish the argument. That is, assuming that at least one of N hypotheses has error at most $\frac{\epsilon}{2}$, give an explicit bound (without O notation) on a size for the test set that is sufficient so that with probability at least $1 - \frac{\delta}{2}$, the hypothesis that performs best on the test set has error at most ϵ.

Proof. (1) At the first step, the probability of at least one of the N hypotheses produced has error at most $\frac{\epsilon}{2}$ on $\mathcal{D}$ is at least $1 - 2^{-\log_2 \frac{2}{\delta}} = 1 - \frac{\delta}{2}$.
(2) At the second step, assuming that at least one of N hypotheses has error at most $\frac{\epsilon}{2}$ on $\mathcal{D}$ and we denote one of them as h_0. Let $\mathcal{H}_\epsilon$ denote the set of hypotheses that has error at least ϵ, then $|\mathcal{H}_\epsilon| \leq N - 1$. Let n be the size of the new test set $\mathcal{S}'$. Let $x_{h,j}$ $(j = 1, \ldots, n)$ be the indicator random variable for the event that $h \in \mathcal{H}$ makes a mistake on the jth example in $\mathcal{S}'$. Then

$$err_{\mathcal{S}'}(h) = \frac{1}{n}\sum_{i=1}^{n} x_{h,j}, \quad \forall\, h \in \mathcal{H}, \qquad err_{\mathcal{D}}(h_0) < \frac{\epsilon}{2}, \qquad \text{and} \quad err_{\mathcal{D}}(h) > \epsilon, \quad \forall\, h \in \mathcal{H}'.$$

The probability of all hypothesis in $\mathcal{H}_\epsilon$ make error at most $(1-\alpha)\epsilon$ for $\alpha \in (0, \frac{1}{2})$ (so that $2(1-\alpha) > 1$) on the test set is

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} x_{h,i} < (1-\alpha)\epsilon,\ \forall\, h \in \mathcal{H}_\epsilon\right) &\leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} x_{h,i} < (1-\alpha) err_{\mathcal{D}}(h),\ \forall\, h \in \mathcal{H}_\epsilon\right) \\ &\leq |\mathcal{H}_\epsilon| e^{-\frac{\alpha^2}{2} n \cdot \min_{h \in \mathcal{H}_\epsilon} err_{\mathcal{D}}(h)} \leq (N-1) e^{-\frac{\alpha^2}{2} n \epsilon}.\end{aligned}$$

The probability of h_0 make error at least $(1-\alpha)\epsilon$ on the test set is

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} x_{h_0,i} > (1-\alpha)\epsilon\right) \leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} x_{h_0,i} > 2(1-\alpha)\frac{\epsilon}{2} \middle| err_{\mathcal{D}}(h_0) = \frac{\epsilon}{2}\right) \leq e^{-\frac{(1-2\alpha)^2}{6} n \epsilon},$$

where the first inequality comes from the following lemma:

Lemma 1. *Suppose that $X \sim$ Binomial(n, p), then the probability $\mathbb{P}(X \geq k)$ is increasing in p for $k \in \{0, 1, \ldots, n\}$ and n fixed.*

Proof. Let $f_k(p) = \sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i}$, then

$$\begin{aligned} f_k'(p) &= \sum_{i \geq k} \binom{n}{i} [i p^{i-1}(1-p)^{n-i} - (n-i) p^i (1-p)^{n-i-1}] \\ &= n \sum_{i \geq k} \binom{n-1}{i-1} p^{i-1}(1-p)^{n-i} - n \sum_{i \geq k} \binom{n-1}{i} p^i (1-p)^{n-i-1} \end{aligned}$$

Solution (cont.)

$$
\begin{aligned}
&= n\left[\sum_{i\geq k}\binom{n-1}{i-1}p^{i-1}(1-p)^{n-i-1}\right]\cdot(1-p) - n\sum_{i\geq k}\binom{n-1}{i}p^{i}(1-p)^{n-i-1} \\
&= n\sum_{i\geq k}\binom{n-1}{i-1}p^{i-1}(1-p)^{n-i-1} - n\sum_{i\geq k}\binom{n-1}{i-1}p^{i}(1-p)^{n-i-1} - n\sum_{i\geq k}\binom{n-1}{i}p^{i}(1-p)^{n-i-1} \\
&= n\sum_{i\geq k}\binom{n-1}{i-1}p^{i-1}(1-p)^{n-i-1} - n\sum_{i\geq k}\binom{n}{i}p^{i}(1-p)^{n-i-1} \\
&= n\sum_{i\geq k}\binom{n-1}{i-1}p^{i-1}(1-p)^{n-i-1}\left(1-\frac{np}{i}\right)
\end{aligned}
$$

If $k \geq \lceil np \rceil$, then $1 - \frac{np}{i} \geq 1 - \frac{np}{k} \geq 0$ for $i \geq k$ and thus $f_k'(p) \geq 0$.
If $k < \lceil np \rceil$, then the term $p^{i-1}(1-p)^{n-i-1}\left(1-\frac{np}{i}\right)$ is negative for $i \leq k$. Thus, $f_k'(p) \geq f_{k-1}'(p) \geq \cdots \geq f_0'(p) \equiv 0$ since $f_0(p) \equiv 1$.
So $f_k'(p)$ is always non-negative for fixed k and n. □

So the probability that either or both of the two above events happen is at most $(N-1)e^{-\frac{\alpha^2}{2}n\epsilon} + e^{-\frac{(1-2\alpha^2)^2}{6}n\epsilon}$, which we want to be less than $\frac{\delta}{2}$. Then we can find n such that $2\max\left\{(N-1)e^{-\frac{\alpha^2}{2}n\epsilon}, e^{-\frac{(1-2\alpha^2)^2}{6}n\epsilon}\right\} \leq \frac{\delta}{2}$. So a valid lower bound of n can be

$$
n \geq \max\left\{\frac{2}{\alpha^2}, \frac{6}{(1-2\alpha)^2}\right\}\cdot\frac{1}{\epsilon}\left[\ln(N-1) + \ln\left(\frac{4}{\delta}\right)\right],
$$

for $\alpha \in (0, \frac{1}{2})$. That is, when we have enough samples in $\mathcal{S}'$, then with high probability, all hypotheses that have error at least ϵ on $\mathcal{D}$ will make error at least $(1-\alpha)\epsilon$ on $\mathcal{S}'$, and at least one hypothesis h_0 will make error at most $(1-\alpha)\epsilon$ on $\mathcal{S}'$, so that we could at least choose h_0. If another hypothesis that performs better than h_0 on $\mathcal{S}'$ exists, then it must also have error at most ϵ on $\mathcal{D}$. Therefore, we are guaranteed to choose a good hypothesis with high probability.
Since

$$
\max\left\{\frac{2}{\alpha^2}, \frac{6}{(1-2\alpha)^2}\right\} = \begin{cases} \frac{2}{\alpha^2} & , 0 < \alpha \leq 2-\sqrt{3} \\ \frac{6}{(1-2\alpha)^2} & , 2-\sqrt{3} < \alpha < \frac{1}{2} \end{cases},
$$

$\frac{2}{\alpha^2}$ is decreasing on $(0, 2-\sqrt{3})$ and $\frac{6}{(1-2\alpha)^2}$ is increasing on $(2-\sqrt{3}, \frac{1}{2})$, we have that $\max\left\{\frac{2}{\alpha^2}, \frac{6}{(1-2\alpha)^2}\right\}$ is minimized when $\alpha = 2-\sqrt{3}$. So we just need

$$
n \geq \frac{2(7+4\sqrt{3})}{\epsilon}\left[\ln\log_2\left(\frac{1}{\delta}\right) + \ln\left(\frac{4}{\delta}\right)\right].
$$

Combining these two steps (i) and (ii), with probability at least $\left(1-\frac{\delta}{2}\right)^2 = 1-\delta+\frac{\delta^2}{4} \geq 1-\delta$, the hypothesis generated by $\mathcal{B}$ has error at most ϵ on $\mathcal{D}$ when $N = \log_2\frac{2}{\delta}$ and $n \geq \frac{2(7+4\sqrt{3})}{\epsilon}\left[\ln\log_2\left(\frac{1}{\delta}\right) + \ln\left(\frac{4}{\delta}\right)\right]$. □

Problems

3. **Tracking a moving target.** Here is a variation on the deterministic Weighted-Majority algorithm, designed to make it more adaptive.

   (a) Each expert begins with weight 1 (as before).

   (b) We predict the result of a weighted-majority vote of the experts (as before).

   (c) If an expert makes a mistake, we penalize it by dividing its weight by 2, but *only* if its weight was at least $\frac{1}{4}$ of the average weight of experts.

   Prove that in any contiguous block of trials (e.g., the 51st example through the 77th example), the number of mistakes made by the algorithm is at most $O(m + \log n)$, where m is the number of mistakes made by the best expert in that block, and n is the total number of experts.

Proof. Let M_t be the the number of mistakes we've made in first t trials, m_t be the number of mistakes the best expert has made in the first t trials, and W_t be the initial total weight of all experts.
Then each expert will have weight at least $\frac{1}{8}$ of the total weight of all experts at any time. We prove it by induction.

(1) At the first block, all weights are initialized as one so that they are all equal to the average weight W_0, which is larger than $\frac{1}{8}W_0$.

(2) Suppose that $w_{t,i} \geq \frac{1}{8}W_t$ for all i. Notice that $W_t \geq W_{t+1}$.If the weight of the ith expert is reduced after tth trial, i.e., $w_{t+1,i} = \frac{1}{2}w_{t,i}$, then $w_{t,i} \geq \frac{1}{4}W_t$ and thus $w_{t+1,i} \geq \frac{1}{8}W_t \geq \frac{1}{8}W_{t+1}$. Otherwise, the weight remains the same, $w_{t+1,i} = w_{t,i}$, then $w_{t+1,i} \geq \frac{1}{8}W_t \geq \frac{1}{8}W_{t+1}$.

For each block, let W be the initial total weight, and m be the number of mistakes made by the best expert in this block. Then the weight of the best expert at the end of this block is at least $\frac{1}{2^m} \cdot \frac{1}{8} \cdot \frac{W}{n}$.
At each trial, when making a mistake, at most $\frac{W}{4}$ of weight is fixed. While experts with at least half of total weight make a mistake, so at least $\frac{W}{2} - \frac{W}{4} = \frac{W}{4}$ of weight is divided by 2. So each time the experts make a mistake, at least $\frac{W}{8}$ is reduced. Thus, the total weight at the end of this block is at most $\left(\frac{7}{8}\right)^M W$.
Since the final weight of the best expert in this block should be at least $\frac{1}{8}$ of the final average weight, we have

$$\frac{1}{2^m} \cdot \frac{1}{8} \cdot \frac{W}{n} \leq \frac{W}{8}\left(\frac{7}{8}\right)^M,$$

which yields

$$\left(\frac{8}{7}\right)^M \leq 2^m n$$

$$M \leq \frac{1}{\log\left(\frac{8}{7}\right)}[m \log 2 + \log n].$$

So, in any contiguous block of trials, the number of mistakes made by the algorithm is at most $O(m + \log n)$, where m is the number of mistakes made by the best expert in that block, and n is the total number of experts.

□

4. **Decision lists revisited.** On Homework 1 you showed that any decision list over n Boolean variables can be written as a linear threshold function. This makes one wonder if the Perceptron algorithm would get a good mistake bound in learning them.

   Unfortunately, the Perceptron may make $2^{\Omega(n)}$ mistakes in learning a decision list over n Boolean variables. To prove this, give a set S of labeled examples in $\{0,1\}^n$ that

   (a) S is consistent with some decision list L and yet

   (b) S has the property that any LTF with integer weights that has zero error on S must have at least one weight that is exponentially large.

   Prove that your set S has properties (a) and (b).

Proof. Without loss of generality, we assume that n is even. Otherwise, we can discard the last variable. We first give the sample set as

$$S = \Bigg\{ (\mathbf{0}, 0),\ (e_2, b_2),\ (e_4, b_4),\ \ldots,\ (e_n, b_n),$$
$$(e_2 + e_4, b_2),\ (e_4 + e_5, b_4),\ \ldots,\ (e_{n-2} + e_{n-1}, b_{n-2}),$$
$$\left(e_1 + \sum_{i=1}^{\frac{n}{2}} e_{2i}, b_1\right),\ \left(e_3 + \sum_{i=1}^{\frac{n}{2}-1} e_{2+2i}, b_3\right), \ldots, \left(e_{n-3} + \sum_{i=1}^{2} e_{n-4+2i}, b_{n-3}\right) \Bigg\}$$

where $e_i = (0, \ldots, 0, \underset{i}{1}, 0, \ldots, 0)$ is the unit vector and $b_i = \frac{1+(-1)^i}{2}$ for $i = 1, \ldots, n$. Here $b_i = 0$ and 1 denote $-$ and $+$, respectively. The size of S is $1 + \frac{n}{2} + \frac{n}{2} + \left(\frac{n}{2} - 1\right) = \frac{3}{2}n$.

(a) S is consistent with the following decision list L,

```
if x_1 = 1 then −
else if x_2 = 1 then +
                ⋮
else if x_{n−1} = 1 then −
else if x_n = 1 then +
else −
```

The default output is $-$, which is consistent to $(\mathbf{0}, 0)$. For any other element in S, the label only depends on the first nonzero element of x. For example, if the ith entry of x is the first nonzero entry, then the label will be b_i, which is also consistent with L.

(b) For any linear threshold function $f(x) = +$ if and only if $\sum_{i=1}^{n+1} w_i x_i > 0$ with integer weights and $x_{n+1} \equiv 1$ that is consistent with $\mathcal{S}$, we have the following facts.

(1) For $i = 2, 4, \ldots, n$, $w_i > |w_{n+1}| \geq 0$. Since $f(\mathbf{0}) = 0$, we have $w_{n+1} \leq 0$. Also, $f(e_i) = 1$ implies $w_i + w_{n+1} > 0$, i.e. $w_i > -w_{n+1} = |w_{n+1}| \geq 0$.

(2) For $i = 1, 3, \ldots, n-1$, $w_i \leq -\sum_{j=1}^{\frac{n-i+1}{2}} w_{i-1+2j} - |w_{n+1}| \leq 0$. This is because

Solution (cont.)

$$f\left(e_i + \sum_{j=1}^{\frac{n-i+1}{2}} e_{i-1+2j}\right) = 0 \text{ if and only if } w_i + \sum_{j=1}^{\frac{n-i+1}{2}} w_{i-1+2j} + w_{n+1} \leq 0,$$

i.e., $w_i \leq -\sum_{j=1}^{\frac{n-i+1}{2}} w_{i-1+2j} - w_{n+1}$. While from (1) $w_{i-1+2j} > |w_{n+1}| > 0$ for $j = 1, \ldots, \frac{n-i+1}{2}$, so $w_i \leq 0$. Furthermore, $|w_i| = -w_i \geq \sum_{j=1}^{\frac{n-i+1}{2}} w_{i-1+2j} - |w_{n+1}| \geq 0$. So $w_i \leq 0$ for $i = 1, 3, \ldots, n+1$.

(3) For $i = 2, 4, \ldots, n$, $w_i > |w_{i+1}| + |w_{n+1}| \geq 0$. $f(e_i + e_{i+1}) = 1$ implies $w_i + w_{i+1} + w_{n+1} > 0$, i.e., $w_i > -w_{i+1} - w_{n+1} = |w_{i+1}| + |w_{n+1}|$.

Next, we can prove that the magnitude of w_1 is exponentially large. From (3) and (2) we have

$$w_i > |w_{i+1}| + |w_{n+1}| \geq \sum_{j=1}^{\frac{n-i}{2}} w_{i+2j} > 0$$

for $i = 2, 4, \ldots, n$. So

$$\begin{aligned}
|w_1| &\geq \sum_{j=1}^{\frac{n}{2}} w_{i-1+2j} - |w_{n+1}| \\
&\geq 2 \sum_{j=1}^{\frac{n}{2}-1} w_{i+1+2j} - |w_{n+1}| \\
&\geq 2^2 \sum_{j=1}^{\frac{n}{2}-2} w_{i+3+2j} - |w_{n+1}| \\
&\vdots \\
&\geq 2^{\frac{n}{2}} w_n - |w_{n+1}| \\
&\geq (2^{\frac{n}{2}} - 1) w_n.
\end{aligned}$$

From (1) we have $w_n > |w_{n+1}| \geq 0$ and w_n is integer, so $w_n \geq 1$. Therefore, $|w_1| = 2^{\Omega(n)}$.
Since the Perceptron Algorithm only changes the magnitude of each weight by at most 1, it will make $2^{\Omega(n)}$ mistakes in learning L over S. □