

Homework 2
Due: March 2, 2018 at 7:00 PM

Written Questions

Problem 1

(5 points)

Let \(x_1, x_2, ..., x_m\) be an i.i.d. (independent and identically distributed) sample drawn from distribution \(B(p)\) where \(B(p)\) denotes a Bernoulli distribution. Specifically,

\[
P(B = 1) = p, \quad P(B = 0) = 1 - p.
\]

Suppose \(p\) is an unknown parameter and we estimate it via:

\[
\hat{p} = \frac{1}{m} \sum_{i=1}^{m} (x_i).
\]

Show that \(\hat{p}\) is an unbiased estimator for \(p\). Recall that an estimator is unbiased if its expected value over all possible samples is equal to the parameter it is estimating. Note: In lecture, we showed this estimator is unbiased for a sample size of 3. For this question, we are asking you to generalize the argument to a sample size of \(m\).

Solution: We want to show that \(E[\hat{p}] = p\) for an arbitrary \(m\).

We know that \(E[x_i] = 1 \cdot p + 0 \cdot (1 - p) = p\) for any sample \(x_i\), so using linearity of expectation:

\[
E[\hat{p}] = E \left[ \frac{1}{m} \sum_{i=1}^{m} x_i \right]
\]

\[
= \frac{1}{m} \sum_{i=1}^{m} E[x_i]
\]

\[
= \frac{1}{m} \sum_{i=1}^{m} p
\]

\[
= p
\]

Thus \(E[\hat{p}] = p\), so our estimator is unbiased.
One could also use the binomial theorem, though it is more complicated:

\[
E[p] = E\left[ \frac{\text{# of 1's}}{\text{# of samples}} \right] \\
= \sum_{k=0}^{m} \binom{m}{k} p^k (1-p)^{m-k} \frac{k}{m} \\
= \sum_{k=1}^{m} \binom{m}{k} p^k (1-p)^{m-k} \quad \text{(since we get 0 when k=0)} \\
= \sum_{k=1}^{m} \binom{m-1}{k-1} p^k (1-p)^{m-k} \\
= p \sum_{k=1}^{m} \binom{m-1}{k-1} p^{k-1} (1-p)^{(m-1)-(k-1)} \\
= p \sum_{k=0}^{m-1} \binom{m-1}{k} p^k (1-p)^{(m-1)-k} \\
= p \cdot (p + (1-p))^{m-1} \quad \text{(using the binomial theorem)} \\
= p
\]

**Problem 2: Agnostic PAC Learning**

(25 points)

Previously, we looked at PAC learning under the assumption that the true hypothesis was a function within our set of hypotheses \(H\)—the realizable case. However, this assumption does not always hold. In some cases, the true hypothesis is a function \(f \notin H\)—the unrealizable case. Learning in the unrealizable case is also called agnostic learning. We will examine how PAC learning differs in this scenario.

Hoeffding’s inequality can give us a bound on sample size for the unrealizable case in which we have a finite set of hypotheses \(H\). The true function for labeling data is \(f\). Suppose we are labelling data \(x\) generated from distribution \(D\). Define the **expected error** of a hypothesis, \(\text{err}_D(h)\), as the expected proportion of data incorrectly labelled by the hypothesis \(h\), written as:

\[
\text{err}_D(h) = E_{x \sim D} [f(x) \neq h(x)].
\]

We have a sample \(S\). Define the **sampling error** of a hypothesis, \(\text{err}_S(h)\), as the proportion of data from the sample \(S\) incorrectly labelled by hypothesis \(h\). It may be written as:

\[
\text{err}_S(h) = \frac{1}{|S|} \sum_{(x,y) \in S} [y \neq h(x)].
\]

Define \(\text{ERM}(H) = \arg\min_{h \in H} \text{err}_S(h)\) and \(\text{RM}(H) = \arg\min_{h \in H} \text{err}_D(h)\) as the empirical risk minimizing hypothesis and the risk minimizing hypothesis, respectively. In the PAC setting, we content ourselves with an algorithm that, with probability at least \(1 - \delta\), returns a hypothesis \(\hat{h}\) whose error over the distribution \(D\) is within \(\epsilon\) of that of the risk minimizing solution, or:

\[
|\text{err}_D(\hat{h}) - \text{err}_D(\text{RM}(H))| \leq \epsilon.
\]

**Note:** We are considering PAC learning on a binary dataset.

a. First, consider the realizable case, \(f \in H\). What is the PAC learning bound for \(\hat{h} = \text{ERM}(H)\)? Express the sample size bound in terms of \(\epsilon, \delta, \) and \(|H|\).
b. Suppose we’re stuck with agnostic learning such that \( f \notin H \). Our best hypothesis is \( \hat{h} = \text{ERM}(H) \). Use Hoeffding’s inequality to show that, given \( \epsilon_1 \) and \( \delta_1 \), there’s an \( m \) such that sampling \( S = \{x_1, ..., x_m\} \sim D \) gives us \( |\text{err}_S(\hat{h}) - \text{err}_D(\hat{h})| \leq \epsilon_1 \), with probability at least \( 1 - \delta_1 \), for some \( h \).

c. What happens if we use a sample of size \( m \) to evaluate all the hypotheses in \( H \)? In particular, we want it to be simultaneously true that, for all \( h \in H \), \( |\text{err}_S(h) - \text{err}_D(h)| \leq \epsilon_1 \) with probability \( 1 - \delta \). Write an upper bound for the true failure probability in terms of \( \delta_1 \) using the union bound. Then, express \( \delta \) in terms of \( \delta_1 \) so that with probability \( 1 - \delta \) it is true that \( |\text{err}_S(h) - \text{err}_D(h)| \leq \epsilon_1 \) for all \( h \in H \).

d. Now we know that our error estimates for all \( h \in H \) are within \( \epsilon_1 \) of their true errors (with high probability). If we pick \( \hat{h} = \text{ERM}(H) \), how far might \( \hat{h} \) be from \( \text{RM}(H) \)? Call that \( \epsilon \) and write \( \epsilon \) in terms of \( \epsilon_1 \). That is, find \( \epsilon \) in terms of \( \epsilon_1 \) such that \( |\text{err}_D(\hat{h}) - \text{err}_D(\text{RM}(H))| \leq \epsilon \).

e. Putting it all together, define \( m \) in terms of \( \epsilon \) and \( \delta \) so that, with probability at least \( 1 - \delta \) (using the previously computed bound), \( |\text{err}_D(\text{ERM}(H)) - \text{err}_D(\text{RM}(H))| \leq \epsilon \).

Solution:

1. Using the theorem given in class, this is just want our sample size \( m \) to satisfy
   \[
   m \geq \frac{\log \left( \frac{|H|}{\delta} \right)}{\epsilon^2}
   \]

2. Hoeffding says that
   \[
   \mathbb{P}\left( \frac{1}{m} \sum_{i=1}^{m} Z_i - \mu > \epsilon \right) \leq 2 \exp\left( -\frac{2m\epsilon^2}{(a-b)^2} \right)
   \]
   where \( a \) and \( b \) are bounds on the range of values our samples can take. For our 0-1 loss function, these values are 1 and 0 respectively. Substituting in \( \text{err}_S(\hat{h}) \), \( \text{err}_D(\hat{h}) \), \( \epsilon_1 \), and \( \delta_1 \), and using algebra, we get
   \[
   \delta_1 \geq 2 \exp\left( -2m\epsilon_1^2 \right)
   \]
   \[
   \Rightarrow \log\left( \frac{\delta_1}{2} \right) \geq -2m\epsilon_1^2
   \]
   \[
   \Rightarrow -\log\left( \frac{\delta_1}{2} \right) \leq m
   \]
   \[
   \Rightarrow \log\left( \frac{\delta_1}{2} \right) \leq m
   \]

3. Consider how \( P(A \cup B) = P(A) + P(B) - P(A \cap B) \leq P(A) + P(B) \). Using the last expression, if the probability that \( \text{err}_S(h) \) differs significantly from \( \text{err}_D(h) \) is \( \delta_1 \) for all \( h \), then setting \( \delta \geq |H|\delta_1 \) is sufficient to guarantee that the probability that all that are correct is at least \( 1 - \delta \).

4. \[
|\text{err}_D(\hat{h}) - \text{err}_D(\text{RM}(H))| = \text{err}_D(\hat{h}) - \text{err}_D(\text{RM}(H)) \] (because \( \text{RM}(H) \) is the minimizer of this quantity)
   \[
   \leq \text{err}_S(\hat{h}) + \epsilon_1 - \text{err}_D(\text{RM}(H)) \] (with high probability)
   \[
   \leq \text{err}_S(\hat{h}) + \epsilon_1 - \text{err}_S(\text{RM}(H)) + \epsilon_1 \] (again applying bound from previous part)
   \[
   \leq \text{err}_S(\hat{h}) + \epsilon_1 - \text{err}_S(\hat{h}) + \epsilon_1 \] (because \( \hat{h} \) is minimizer of this \( \text{err}_S(\cdot) \))
   \[
   \leq 2\epsilon_1
   \]
5. Plugging into our answer from part 2:

\[
m \geq \frac{\log(\frac{2}{\delta_1})}{2\epsilon_1^2} \tag{1}
\]
\[
m \geq \frac{2\log(\frac{2|H|}{\epsilon})}{\epsilon^2} \tag{2}
\]

**Problem 3: Naive Bayes Maximum Likelihood**

(12 points)

Consider binary dataset \( S \overset{i.i.d.}\sim D \) with observations in the form \( \{(x_j^1, ..., x_j^n, y_j)\} \). Define \( c(y) \) as a function that counts the number of observations such that the label is \( y \).

\[
c(y) = \sum_{(x_j, y_j) \in S} [y_j = y] 
\]

Define \( c(i, y) \) as a function that counts the number of observations such that the label is \( y \) and \( x_i = 1 \).

\[
c(i, y) = \sum_{(x_j, y_j) \in S} [y_j = y, x_j^i = 1] 
\]

Define \( b \) as \( \mathbb{P}(Y = 1) \), and \( b^{i|y} \) as \( \mathbb{P}(X^i = 1 | Y = y) \). Prove that the following estimators are MLE for these parameters:

\[
\hat{b}_{MLE} = \frac{c(1)}{|S|} \quad \text{and} \quad \hat{b}^{i|y}_{MLE} = \frac{c(i, y)}{c(y)} 
\]

**Solution:** Let \( L(b, b^{i|y} | S) \) be the likelihood of the parameters of the model.

\[
L(b, b^{i|y} | S) = \mathbb{P}(S | b, b^{i|y})
\]
\[
= \prod_{j=1}^{n} \mathbb{P}(x_j, y_j | b, b^{i|y})
\]
\[
= \prod_{j=1}^{n} \mathbb{P}(y_j | b, b^{i|y}) \mathbb{P}(x_j | y_j, b, b^{i|y})
\]
\[
= \prod_{j=1}^{n} \mathbb{P}(y_j | b, b^{i|y}) \prod_{i=1}^{m} \mathbb{P}(x_j^i | y_j, b, b^{i|y})
\]
\[
= \prod_{j=1}^{n} b^{y_j} (1 - b)^{1 - y_j} \prod_{i=1}^{m} \left( b^{i|y_j} x_j^i \right)^{x_j^i} \left( 1 - b^{i|y_j} \right)^{1 - x_j^i}
\]

\[
\log L(b, b^{i|y} | S) = \sum_{j=1}^{n} y_j \log(b) + (1 - y_j) \log(1 - b) + \sum_{i=1}^{m} x_j^i \log(b^{i|y_j}) + (1 - x_j^i) \log(1 - b^{i|y_j})
\]
Now we differentiate with respect to the different parameters and set to 0:

\[
\frac{\partial}{\partial b} \log L(b, b^y | S) = \frac{c(1)}{b} - \frac{c(0)}{1-b} = 0
\]

\[
\frac{c(1)}{b} = \frac{c(0)}{1-b}
\]

\[
\frac{c(1)}{b} = \frac{|S| - c(1)}{1-b}
\]

\[
c(1) - bc(1) = b|S| - bc(1)
\]

\[
c(1) = b|S|
\]

\[
\hat{b}_{MLE} = \frac{c(1)}{|S|}
\]

\[
\frac{\partial}{\partial b^y} \log L(b, b^y | S) = \frac{c(i, y)}{b^y} - \frac{c(y) - c(i, y)}{1-b^y} = 0
\]

\[
\frac{c(i, y)}{b^y} = \frac{c(y) - c(i, y)}{1-b^y}
\]

\[
c(i, y) - b^y c(i, y) = b^y c(y) - b^y c(i, y)
\]

\[
c(i, y) = b^y c(y)
\]

\[
\hat{b}_{MLE}^y = \frac{c(i, y)}{c(y)}
\]

**Problem 4: Gradient Descent**

(18 points)

We have a convex function \( f \) over the closed interval \([-b, b]\) (for some positive number \( b \)). Let \( f' \) be the derivative of \( f \). Let \( \alpha \) be some positive number, which will represent a learning rate parameter.

Consider using gradient descent to find the minimum of \( f \): We start at \( x_0 = 0 \). Then, at each step, we set \( x_{t+1} = x_t - \alpha f'(x_t) \). If \( x_{t+1} \) falls below \(-b\), we set it to \(-b\), and if it goes above \( b \), we set it to \( b \).

We say that an optimization algorithm (such as gradient descent) \( \epsilon \)-converges if, at some point, \( x_t \) stays within \( \epsilon \) of the true minimum. Formally, we have \( \epsilon \)-convergence at time \( t \) if

\[|x_t' - x_{\text{min}}| \leq \epsilon, \quad \text{where } x_{\text{min}} = \arg\min_{x \in [-b, b]} f(x)\]

for all \( t' \geq t \).

a. For \( \alpha = 0.1, b = 1, \) and \( \epsilon = 0.001 \), find a convex function \( f \) so that running gradient descent does not \( \epsilon \)-converge. Specifically, make it so that \( x_0 = 0, x_1 = b, x_2 = -b, x_3 = b, x_4 = -b, \) etc.

b. For \( \alpha = 0.1, b = 1, \) and \( \epsilon = 0.001 \), find a convex function \( f \) so that gradient descent does \( \epsilon \)-converge, but only after at least 10,000 steps.

c. Construct a different optimization algorithm that has the property that it will always \( \epsilon \)-converge (for any convex \( f \)) within \( \log_2 (2b/\epsilon) \) steps.

d. Unfortunately, even if \( x_t \) is within \( \epsilon \) of \( x_{\text{min}} \), \( f(x_t) \) can be arbitrarily greater than \( f(x_{\text{min}}) \). However, consider the case where the derivative of \( f \) is always between \(-r\) and \( r \). (\( \forall x \in [-b, b], f'(x) \in [-r, r] \).) In this case, we can make a guarantee about the difference between \( f(x_t) \) and \( f(x_{\text{min}}) \).

Given that \( |x_t - x_{\text{min}}| \leq \epsilon \) and that \(-r \leq f'(x) \leq r\), find a bound on \( |f(x_t) - f(x_{\text{min}})| \) in terms of \( \epsilon \) and \( r \).
Solution:

a. We need a function with a steep negative gradient at $x = 0$ and $x = -1$ and a steep positive gradient at $x = 1$. For example, consider

$$f(x) = \left| 20x - \frac{1}{2} \right|$$

We start at $x_0 = 0$. Then we set $x_1 = x_0 - 0.1f'(x_0) = -0.1(-20) = 2$, so we set $x_1 = 1$. Then we set $x_2 = x_1 - 0.1f'(x_1) = 1 - 0.1(20) = -1$, and so on.

b. Consider $f(x) = 0.001x$. We start at $x_0 = 0$. Then we set $x_1 = x_0 - 0.1f'(x_0) = -0.1(0.001) = -0.0001$. So at each step we move 0.0001 to the left, until we reach $x_{\text{min}} = -1$ at time $t = 10000$.

c. Binary search: Start with $x_L = -b$ and $x_H = b$. At each step, set $x_t = \frac{1}{2}(x_L + x_H)$. Check $f'(x_t)$. If it’s positive, then the minimum must be to the left, so set $x_H = x_t$. If it’s negative, then the minimum must be to the right, so set $x_L = x_t$.

The search range $[x_L, x_H]$ has initial size $2b$ and is halved at each iteration, so after $t$ iterations it has size $2b/2^t$. Setting this equal to $\epsilon$ (the desired search range size):

$$\epsilon = \frac{2b}{2^t}$$

$$2^t = \frac{2b}{\epsilon}$$

$$t = \log_2 \frac{2b}{\epsilon}$$

So after $\log_2(2b/\epsilon)$ iterations, we will have $\epsilon$-convergence to the true minimum.

d. Let $a = \min(x_t, x_{\text{min}})$ and $b = \max(x_t, x_{\text{min}})$. Consider

$$f(b) - f(a) = \int_a^b f'(x) dx$$

If $f'(x) = r$, then this equals $(b - a)r$, which is at most $\epsilon r$. If $f'(x) = -r$, then this equals $-(b - a)r$, which is at least $-\epsilon r$. Therefore $-\epsilon r \leq f(b) - f(a) \leq \epsilon r$, so $|f(x_t) - f(x_{\text{min}})| \leq \epsilon r$. 
