COMS 4771 Machine Learning (Spring 2018)
Problem Set #3

Zhuangyu Ren, Yuchen Mo, Chengzhi Mao - - UNI: zr2209, ym2706, cm3797

April 16, 2019

Problem 1

Let $S = (x_1, y_1), \ldots, (x_n, y_n)$, $w = w_1, \ldots, w_d$

$$P[w|S] = \frac{P(S, w)}{P(S)}$$

$$w = \arg\max_w P[w|S] = \arg\max_w \frac{P(S, w)}{P(S)} = \arg\max_w \frac{P(S|w)P(w)}{P(S)} = \arg\max_w \log(P(S|w)P(w))$$

Because we can ignore the terms independent of w in $\log(P(w))$, so the maximum likelihood can be represented as:

$$\log(P(w)) = C_1 \frac{-w^2}{2\tau^2}$$

Similarly for $\log(P(S|w)$:

$$\log(P(S|w) = C_2 \frac{-(wx - y)^2}{2\sigma^2}$$

$$\log(P(w)P(S|w)) = C_2(\frac{-(wx - y)^2}{2\sigma^2} - \lambda w^2)$$

where $\lambda = C_1/(2C_2\tau^2)$

$$w = \arg\max_w \log(P(w|S)) = \arg\max_w C_2(\frac{-(wx - y)^2}{2\sigma^2} - \lambda w^2)$$

Finding the coefficients w for maximizing the above log likelihood is equivalent to minimizing the following ridge optimization criterion.

$$L = ||Xw - y||^2 + \lambda||w||^2$$

Problem 2

(a)

Using the ERM (empirical risk minimizer) algorithm.

$$f_m^{\text{ERM}} := \text{argmin}_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{f(x_i) \neq y_i\}$$

In this way, we will go through all the samples:

Input:A set of samples $S = \{x_i\}_{i=1}^m$ drawn iid from $\mathcal{D}$, the function class $\mathcal{F}$, and access to a query oracle $\mathcal{O}$ which on input $x_i \in S$ outputs y_i. It must be true that $\exists f \in \mathcal{F}$ such that err(f) = 0.
α=x_1
for $(x_i, y_i) \in S$ **do**
  if $x_i \leqslant \alpha$ and $y_i = 0$ **then**
    $\alpha = x_i$;
  end
end
return f_α

So the time complexity is O(m).
Because functions f_α are in the form of $f_\alpha = \mathbb{1}[x \geqslant \alpha]$. When we have only one data point x_1, this function can set α to some $\alpha \geqslant x_1$. But if we have two node points x_1 and x_2, set $x_1 = 1$ and $x_2 = 0$, the functions cannot find an α to shatter these two data points. So the VC dimension of $\mathcal{F}$ is 1.
From the VC Theorem in class, we can know that, if $m \geqslant C \cdot \frac{\text{VC}(\mathcal{F})\ln(1/\delta)}{\epsilon^2}$, then with probability at least $1 - \delta$,

$$\text{err}(f_m^{\text{ERM}}) - \text{err}(f^*) \leqslant \epsilon$$

We know the err(f^*)=0.
Also, $VC(\mathcal{F}) = 1$, plug into the theorem above, we can get that:
For any $\epsilon, \delta > 0$, if $m \geqslant O(\frac{\ln(1/\delta)}{\epsilon^2})$, then with probability at least $1 - \delta$, err(f_α) $\leqslant \epsilon$.

(b)

Input:A set of samples $S = \{x_i\}_{i=1}^m$ drawn iid from $\mathcal{D}$, the function class $\mathcal{F}$, and access to a query oracle $\mathcal{O}$ which on input $x_i \in S$ outputs y_i. It must be true that $\exists f \in \mathcal{F}$ such that err(f) = 0.

$l = 0$
$r = 1$
n = 1
while $\frac{1}{2^n} > \epsilon$ **do**
  $\alpha = l + (r - l)/2$
  Get y_α by making the query $\mathcal{O}(x_\alpha)$
  if $y_\alpha == 0$ **then**
    $l = \alpha$;
  else if $y_\alpha == 1$ **then**
    $r = \alpha$;
  $n = n + 1$
end
return f_α

Here, using binary search we can divide the [0,1] into $\frac{1}{\epsilon}$ intervals and find the α that is in $(\alpha^* - \epsilon, \alpha^* + \epsilon)$. So the time complexity and number of queries is $O(\log(1/\epsilon))$.
Because the distribution is uniform in [0,1], the pattern we learn using this algorithm is the same as true distribution. So we can saw that with probability 1, $\text{err}(f) < \epsilon$.

(c)

Input:A set of samples $S = \{x_i\}_{i=1}^m$ drawn iid from $\mathcal{D}$, the function class $\mathcal{F}$, and access to a query oracle $\mathcal{O}$ which on input $x_i \in S$ outputs y_i. It must be true that $\exists f \in \mathcal{F}$ such that err(f) = 0.
Sort all the m samples by the value of x_i and put them in this as cending order
Let $V = \mathcal{F}$
$l = 0$
$r = m - 1$
while $l < r$ **do**
  $mid = l + (r - l)/2$;
  if $\exists f_1, f_2 \in V$ such that $f_1(x_{mid}) \neq f_2(x_{mid})$ **then**
    Get y_{mid} by making the query $\mathcal{O}(x_{mid})$
    if $y_{mid} == 0$ **then**
      Set V to be $\{f \in V$: for all x_i that $i \leqslant mid, f(x_i) := 0\ \}$
      $l = mid + 1$;
    else if $y_{mid} == 1$ **then**
      Set V to be $\{f \in V$: for all x_i that $i \geqslant mid, f(x_i) := 1\ \}$
      $r = mid$;
  end
end
return *Any* $f \in V$

In this algorithm, sorting all m samples takes time O(mlog(m)), the binary search through m samples takes time O(log(m)), so the time complexity is O(mlog(m)).
In the worst case, each step in binary search we find f_1, f_2 that are not equal, then we need to do log(m) queries, so this algorithm makes O(log(m)) queries.
Set the set V to be a subset of the previous V can be done by setting the threshold number for the function, which can be done in $O(1)$.

Proof: The above algorithm will update the function set V every time it encounters a mismatch in the output of the function in the function set, the remaining functions f in set V have smaller empirical risk because the functions with error have been removed. Thus using the above algorithm, we can actually minimizing the empirical risk, and keep improving the prediction accuracy of the function set. At last, we will get a function set V which is the same as the result of empirical risk minimizer would get. Thus our resuling f is f_m^{ERM}, because our $f = \arg\min_{f \in F} \frac{1}{m} \sum_{i=1}^m 1(f(x_i) \neq y_i)$

The VC dimension of this problem is 1, by using Vapnik-Chervonenkis Theorem, we know that given the tolerance level ϵ, and confidence δ, if $m >= O(\frac{VC(F) \ln 1/\delta}{\epsilon^2}) = O(\frac{1}{\epsilon^2} \ln 1/\delta)$, then with probability $1 - \delta$, we can achieve the error smaller than ϵ.

Problem 3

To estimate

$$\mathrm{BER}(f) = \frac{1}{2}\left(\Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y | y = 1] + \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y | y = 0]\right)$$

We can use an estimator like

$$\overline{\mathrm{BER}}(f, S) = \frac{1}{2}\left(\frac{\mathrm{Count}[f(x) = 0, y = 1]}{\mathrm{Count}[y = 1]} + \frac{\mathrm{Count}[f(x) = 1, y = 0]}{\mathrm{Count}[y = 0]}\right)$$

.

According to the results from maximum likelihood estimation we know that $\mathbb{E}[\overline{\mathrm{BER}}] = \mathrm{BER}$. Now we shall prove that this estimator satisfies our concentration bound.

Denote $m_+ := \mathrm{Count}[y = 1]$, $m_- := \mathrm{Count}[y = 0]$, s.t. $m_+ + m_- = m$. In these 2 cases that our estimator may fail:

- m_+ and m_- are so small that we cannot bound our error in a reasonable range.
- m_+ and m_- are good, and the estimator still fails with a certain probability.

It's still possible that our estimator works in the first case, but we consider it fails with 100% to prove a stronger bound. Consider the probability that the first case occurs. Let $p = \min(\Pr[y = 1], \Pr[y = 0]) \leq 0.5$, $\mathrm{Count}_{\min} = \min(m_+, m_-)$. By Chebyshev we have

$$\Pr\left[|\mathrm{Count}_{\min} - pm| \geq c\right] \leq \frac{\mathrm{Var}[\mathrm{Count}_{\min}]}{c^2} = \frac{mp(1-p)}{c^2} \tag{1}$$

c is a constant that describes our lower bound on $\mathrm{Count}_{\min}$. We will derive such a c value in the following procedures.

And now we assume the 'm_+ and m_- are not too small' condition satisfied, we will then give the concentration bound of our estimator. We still consider a stronger case, i.e., we only consider the estimator succeeds when the following two inequalities both hold.

$$\frac{\mathrm{Count}[f(x) = 0, y = 1]}{m_+} \leq \epsilon$$
$$\frac{\mathrm{Count}[f(x) = 1, y = 0]}{m_-} \leq \epsilon$$

From Chernoff we know

$$\Pr\left[\left|\frac{\mathrm{Count}[f(x) = 0 | y = 1]}{m_+} - \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y | y = 1]\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 m_+}$$
$$\Pr\left[\left|\frac{\mathrm{Count}[f(x) = 1 | y = 0]}{m_-} - \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y | y = 0]\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 m_-}$$

Take the union bound, and apply absolute value inequality on the events we have

$$\Pr\left[\left|\overline{\mathrm{BER}} - \mathrm{BER}\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 m_+} + 2e^{-2\epsilon^2 m_-} \leq 4e^{-2\epsilon^2 \min(m_+, m_-)} \tag{2}$$

Say we hope this failure probability is no larger than $\frac{0.01}{16}$, from (2) we have

$$\min(m_+, m_-) \geq \frac{1}{\epsilon^2}\log(80) \tag{3}$$

Bring (3) back into (1), we can now decide the value of c.

$$c = pm - \frac{1}{\epsilon^2}\log(80) \tag{4}$$

Bring this lower bound and (4) into (1) we have our specific constraint on $\text{Count}_{\min}$:

$$\begin{aligned}
\Pr\left[|\text{Count}_{\min} - pm| \geq c\right] \leq \frac{\text{Var}[\text{Count}_{\min}]}{c^2} &= \frac{mp(1-p)}{(pm - \frac{1}{\epsilon^2}\log(80))^2} \\
&= \frac{m\epsilon^4 p(1-p)}{(pm\epsilon^2 - \log(80))^2} \\
&\leq \frac{m\epsilon^4 p(1-p)}{(pm\epsilon^2 - \frac{1}{200}pm\epsilon^2)^2} \\
&= \frac{1-p}{mp(1-\frac{1}{200})^2} \\
&\leq \frac{\epsilon^2}{1000(1-\frac{1}{200})^2} \\
&\leq \frac{1}{1000(1-\frac{1}{200})^2}
\end{aligned} \tag{5}$$

We used the inequality like $mp \geq \frac{1000}{\epsilon^2}$. Now we will summarize our proof.

- With $\frac{1}{1000(1-\frac{1}{200})^2}$ probability, we cannot use (2) to bound our estimation, so we consider this case as failed.
- With $(1 - \frac{1}{1000(1-\frac{1}{200})^2})$ probability, we can use (2) to bound our estimation with failure rate 1/1600.

Consider when we generate these samples, we can first randomly decide how many positive and negative samples we want, and then pick them using the same iid. sampler. This will assign the same probability to each data point as the original iid. sampler over the entire dataset, but here we can see that the two procedures in our case 2 are independent.

Therefore, the total failure rate is

$$\frac{1}{1000(1-\frac{1}{200})^2} + \frac{1}{1600}\left(1 - \frac{1}{1000(1-\frac{1}{200})^2}\right) \approx 0.00163 \leq 0.01$$

Problem 4

Achieving fairness through adversarial training and thresholding.

We first tried to use fully connected neural network to predict the income variable y as well as maintain the fairness given gender g and other information x. We use adversarial training to achieve fairness by removing the information related to the gender variable.

We construct a neural network F to predict a representation from given information, $h = F(x)$, then we use the representation h to predict the income y. Then we construct another discriminator network D which predicts the gender g using h. Then the function F learns to fool D by removing the gender related information. In doing this, we can achieve same prediction given gender information.

In addition, to achieve additional fairness, we set up a margin for the decision boundary to balance the accuracy of different gender.