COMS 4771 HW3

Due: Sat April 13, 2019 at 11:59pm

You are allowed to write up solutions in groups of (at max) three students. These group members don't necessarily have to be the same from previous homeworks. Only one submission per group is required by the due date on Gradescope. Name and UNI of all group members must be clearly specified on the homework. No late homeworks are allowed. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible solutions for homework questions is encouraged on piazza and with peers outside your group, but every group must write their own individual solutions. You should cite all resources (including online material, books, articles, help taken from specific individuals, etc.) you used to complete your work.

1 **[Bayesian interpretation of ridge regression]** Consider the following data generating process for linear regression problem in $\mathbb{R}^d$. Nature first selects d weight coefficients $w_1, \ldots, w_d$ as $w_i \sim N(0, \tau^2)$ i.i.d. Given n examples $x_1, \ldots, x_n \in \mathbb{R}^d$, nature generates the output variable y_i as

$$y_i = \sum_{j=1}^{d} w_j x_{i,j} + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$ i.i.d.

Show that finding the coefficients $w_1, \ldots, w_d$ that maximizes $P[w_1, \ldots, w_d | (x_1, y_1) \ldots, (x_n, y_n)]$ is equivalent to minimizing the ridge optimization criterion.

2 **[A question on active learning]** Suppose we are trying to learn the correct "threshold" for data in $[0, 1]$. Formally, let $\mathcal{D}$ be a distribution over $[0, 1] \times \{0, 1\}$, for $\alpha \in [0, 1]$ define $f_\alpha : [0, 1] \to \{0, 1\}$ by $f_\alpha(x) = \mathbb{1}[x \geq \alpha]$, and let $\mathcal{F} = \{f_\alpha \mid \alpha \in [0, 1]\}$ be the function class we are interested in learning. Furthermore, assume that there is a "true" target concept which always labels the data correctly (i.e. assume $\exists f_\alpha \in \mathcal{F} : \text{err}(f_\alpha) = 0$).

(a) Give a learning algorithm which given a sample $S = \{(x_i, y_i)\}_{i=1}^m$ drawn iid from $\mathcal{D}$ runs in time $O(m)$ and outputs a classifier $f_\alpha \in \mathcal{F}$ with the following property: for any $\epsilon, \delta > 0$ if $m \geq O\left(\frac{1}{\epsilon^2} \ln(1/\delta)\right)$ then with probability at least $1 - \delta$ over the draw of S, $\text{err}(f_\alpha) \leq \epsilon$. Make sure you prove that it has this property.

It is interesting to note that due to the assumption that there exists a true classifier in $\mathcal{F}$ which perfectly labels the points, it can actually be shown that only $O\left(\frac{1}{\epsilon} \ln(1/\delta)\right)$ points are required to PAC learn $\mathcal{F}$ (you do not need to prove this). However, this is essentially tight (it can be shown that there exist distributions on which that many points are needed). This is

disappointing since the problem is simple enough that one could hope for a logarithmic dependence on $1/\epsilon$ only (i.e. exponentially better than what we have right now). To achieve something like this we can look towards **active learning**.

Active learning is a slightly different learning framework in which the learner is allowed to have some amount of say into which examples it gets (as opposed to the PAC learning framework seen in class where the learner is simply given a set of examples). Active learning is interesting in part because it better reflects the process of learning that we are familiar with. Indeed, in real life we as learners can ask questions and demand explanations. We are never expected to learn by simply being handed a set of examples. Furthermore, in many settings active learning can yield exponential improvements over passive PAC learning. Indeed:

(b) Suppose that $\mathcal{D}$ is uniform over $[0, 1]$ and that the learner, rather than being given a set of points drawn iid from $\mathcal{D}$, is able to query for the true label of any point $x \in [0, 1]$ (and gets the answer in $O(1)$ time). Give a learning algorithm which given $\epsilon > 0$ makes at most $O(\ln(1/\epsilon))$ queries, runs in time $O(\ln(1/\epsilon))$, and returns some $f_\alpha \in \mathcal{F}$ such that $\text{err}(f) \leq \epsilon$ with probability 1. Make sure you prove that it has these properties.

Unfortunately there are two issues with an "active learning" solution of this type. First and foremost, $\mathcal{D}$ is usually unknown (above we assume it is uniform over $[0, 1]$) and it is not clear how to extend the above solution to work in such cases. Second, it is not always reasonable to assume that the algorithm can just ask for the label of ANY point in the input space. For example, in image classification most of the input space looks like nonsensical images to human beings. A nice way to deal with both issues is to assume that the algorithm gets points drawn iid from $\mathcal{D}$ as before but that they are **unlabelled**. The algorithm can then actively and sequentially query for the labels of these points, and do so only if the label is useful (i.e. if it brings more information). Formally, consider the following relatively general "active learning algorithm":

Algorithm 1 Active learning algorithm

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 $\text{err}(f) = 0$.
Let $V = \mathcal{F}$
for $x_i \in S$ **do**
  if $\exists f_1, f_2 \in V$ such that $f_1(x_i) \neq f_2(x_i)$ **then**
    Get y_i by making the query $\mathcal{O}(x_i)$
    Set V to be $\{f \in V : f(x_i) = y_i\}$
  end if
end for
return Any $f \in V$.

The issue with this algorithm is that it is often difficult to check the condition of the `if` statement and update V efficiently (i.e. without using too much running time). Also it is not clear in which order one should iterate through S. But for some simple problems there are solutions to these issues, in which case this active learning algorithm can give a much better label complexity (number of labeled points it needs) than the standard PAC alternatives.

(c) Going back to our problem from (a) and (b) rewrite Algorithm 1 so that:

* Your algorithm runs in time $O(m \log(m))$ (you can assume each call to $\mathcal{O}$ takes $O(1)$ time).
* Your algorithm makes $O(\log m)$ queries (calls to $\mathcal{O}$).
* If $m \geq O\left(\frac{1}{\epsilon^2} \ln(1/\delta)\right)$ then with probability at least $1 - \delta$ over the draw of S your algorithm returns f_α such that $\text{err}(f_\alpha) \leq \epsilon$.

As always make sure to prove that your algorithm has each of these properties.

Note that the label complexity of your new algorithm (i.e. how many examples it needs to have labelled) is exponentially smaller than that of the algorithm from part (a).

3 **[BER / application of large deviation theory]** Suppose we have a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$, a sample $S = \{(x_i, y_i)\}_{i=1}^m$ drawn iid from $\mathcal{D}$, and a classifier $f : \mathcal{X} \to \{0, 1\}$. We have usually looked at the 0-1 error of f which is given by $\mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x) \neq y]$ and which can be estimated via $\frac{1}{m} \sum_{i=1}^m \mathbb{1}[f(x_i) \neq y_i]$.

Sometimes, it is more reasonable to look at the **Balanced Error Rate** (BER) of f which is defined as:

$$\text{BER}(f) = \frac{1}{2}\left(\mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x) \neq y \mid y = 1] + \mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x) \neq y \mid y = 0]\right)$$

This is useful when one needs a classifier that has good accuracy on both classes (points where $y = 1$ and points where $y = 0$) despite having a distribution with very different priors (i.e. when one class is much more prevalent than the other).

Propose an estimator $\overline{\text{BER}}(f, S)$ of $\text{BER}(f)$ with the following property: for any $\varepsilon \in (0, 1)$, if $m \geq \frac{1000}{\varepsilon^2 \min(\mathbb{P}[y=1], \mathbb{P}[y=0])}$ then with probability at least 0.99 (over the draw of S),

$$\left|\overline{\text{BER}}(f, S) - \text{BER}(f)\right| \leq \varepsilon$$

Then show that your estimator has this property.

Hint: Use the Chernoff-Hoeffding bound seen in class.

4 **[Fairness competition question]** You'll compete with your classmates on designing a good quality income-level fair classifier.

- Sign up on `http://www.kaggle.com` with your columbia email address.
- Visit the COMS4771 competition at: `https://www.kaggle.com/c/fair-classification` and develop a fair income-level classifier.
- Your pdf write-up should describe your design for your classifier: What preprocessing techniques and classifier you used? Why you made these choices? What resources you used and were helpful? What worked and what did not work?

Evaluation criterion:

- You must use your (and your group members) UNI as your team name in order to get points. For example:
  * If you have two group members with uni: ab1234 and xy0987,
  * the team name should be: ab1234_xy0987
- Your grade on this problem depends on your ranking (metric is described on the Overview-Evaluation section) and your write-up.