Homework #5

Due: 2024-12-22 23:59 | 7 Problems, 100 Pts

Name: XXX, ID: XXX

Problem 1 (16'). Consider the following randomized online learning algorithm for expert advice problem.

Algorithm 1: Randomized online learning algorithm for expert advice

```
Set a constant η > 0, the number of experts N
L_0 ← 0^N                                              ▷ Cumulative loss vector.
for t = 1, 2, ···, T do
    W(t) = Σ_i exp(−η L_{t−1}(i))                      ▷ Normalization coefficient.
    Select the i-th expert with probability p_t(i) = exp(−η L_{t−1}(i)) / W(t)
    Observe the loss vector l_t ∈ [0, 1]^N for each expert   ▷ The loss is guaranteed in [0, 1].
    Update the cumulative loss L_t ← L_{t−1} + l_t
end
```

The expected loss is $\sum_{t=1}^{T} \boldsymbol{p}_t^\top \boldsymbol{l}_t$.

(1) (7') Prove that,

$$\frac{W(t+1)}{W(t)} = \boldsymbol{p}_t^\top \exp(-\eta \boldsymbol{l}_t).$$

(2) (9') Prove the following upper bound for expected loss:

$$\sum_{t=1}^{T} \boldsymbol{p}_t^\top \boldsymbol{l}_t - \boldsymbol{L}_T(i) \le \frac{\ln N}{\eta} + T\eta$$

for any $i \in [N]$.

[Hint: Consider the potential function $\Phi_t = \frac{1}{\eta}\ln(W(t))$. You may find the following inequality useful: $\mathrm{e}^{-x} \le 1 - x + x^2, x > 0$.]

◀

Problem 2 (15'). Consider the following boosting algorithm we learned in class.

Algorithm 2: Boosting algorithm

Input: Number of iterations M (where M is odd), a sample S of n labeled examples $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ with labels $y_1, \cdots, y_n$, a γ-weak ($\gamma > 0$) learner (i.e., an algorithm that given n labeled examples and a non-negative weight $\boldsymbol{w} \in \mathbb{R}^n$, gives an hypothesis with at least $\frac{1}{2} + \gamma$ accuracy on the weight $\boldsymbol{w}$).

$\boldsymbol{w}_1 \leftarrow (1, 1, \cdots, 1)$ $\quad\triangleright$ Initialize each example $\boldsymbol{x}_i$ to have a weight $\boldsymbol{w}_1(i) = 1$.

for $t = 1, 2, \cdots, M$ **do**

  Call the γ-weak learner on the sample S with weight $\boldsymbol{w}_t$ to get the hypothesis h_t.

  for $i = 1, 2, \cdots, n$ **do**

    if $\boldsymbol{h}_t(x_i) \neq y_i$ **then**

      $\boldsymbol{w}_{t+1}(i) \leftarrow \boldsymbol{w}_t(i) \cdot \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$

    else

      $\boldsymbol{w}_{t+1}(i) = \boldsymbol{w}_t(i)$

    end

  end

end

Output: The classifier $\mathrm{Maj}(h_1, \cdots, h_M)$.

Assume hypothesis h_t has error rate β_t on the weighted sample $(S, \boldsymbol{w}_t)$.

(1) (10') Suppose β_t is much less than $\frac{1}{2} - \gamma$. Then, after the booster multiples the weight of misclassified examples by $\alpha = \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$, hypothesis h_t will still have error less than $\frac{1}{2} - \gamma$ under the new weights. This means that h_t could be given again to the booster (perhaps for several times in a row). Calculate, as a function of α and β_t, how many times in a row h_t could be given to the booster before its error rate rises to above $\frac{1}{2} - \gamma$.

(2) (5') Modify the boosting algorithm in the following way: During the iteration, multiply the weight of each example that was misclassified by h_t by $\alpha_t = \frac{1-\beta_t}{\beta_t}$, instead of $\alpha = \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$. Prove that, $h_{t+1} \neq h_t$.

◀

Problem 3 (20'). Given a stream of integers $a_1, a_2, \cdots$, where $a_i \in \{1, 2, 3, \cdots, m\}$. The integers arrive one by one in the stream, and the total number of elements n is unknown in advance.

(1) (5') Give an algorithm that will select a symbol uniformly at random from the stream. How much memory does your algorithm require?

(2) (5') Give an algorithm that will select a symbol with probability proportional to a_i^2. How much memory does your algorithm require?

(3) (10') Give an algorithm to draw a uniform sample set X of size t ($t \leq n$). Prove the correctness of your algorithm.

◀

Problem 4 (10').

(1) (5') Construct an example in which the majority algorithm gives a false positive, i.e., stores a non-majority element at the end.

(2) (5') For any fixed $k \geq 2$, construct an example in which the frequent algorithm in fact does as badly as in the theorem, i.e., it under counts some item by $\frac{n}{k+1}$.

◀

Problem 5 (15'). Let

$$H = \{h \mid h_{ab} : \{1, 2, \cdots, m\} \to \{0, 1, \cdots, M-1\}, a, b \in \{0, 1, \cdots, M-1\}\}$$

be a set of hash functions. Is H always 2-universal under the following conditions? You don't need to prove your answer.

(1) (6') In this part, $h_{ab}(x) = ax + b \pmod M$.

  (a) (3') $M = p^k$, where p is a prime number greater than m and $k > 1$.

  (b) (3') $M = pq$, where p, q are prime numbers greater than m.

(2) (9') In this part, $m = M$ and M is a prime number.

  (a) (3') $h_{ab}(x) = x^a + b \pmod M$.

  (b) (3') $h_{ab}(x) = a^x + b \pmod M$.

  (c) (3') $h_{ab}(x) = ax^3 + b \pmod M$.

[Hint: In this problem, proving your answer may be a little difficult, which may use Bézout's identity and Fermat's little theorem in number theory. So, you do not need to prove it. However, finding out the answer is not so difficult. For example, you can write a program to draw your conclusion. You don't need to show your code, either.] ◀

Problem 6 (12'). Does there exist a set of hash functions $H = \{h \mid h : \{1, 2, 3, 4\} \to \{1, 2, 3, 4\}\}$, where $|H| \leq 16$ and H is 2-Universal? If your answer is yes, please give an example and show it is correct; if your answer is no, please prove it. ◀

Problem 7 (12'). Recall that a family of hash functions $H = \{h \mid h : [m] \to [M]\}$ is 2-universal, if and only if for all x and y in $\{1, 2, \cdots, m\}$, $x \neq y$, $\mathbb{P}_{h \sim H}[h(x) = w, h(y) = z] = \frac{1}{M^2}$. The randomness comes from the selection of h. Suppose $m \geq 2$.

(1) (2') Prove that, $|H| \geq M^2$.

(2) (10') Prove that, if $M = 2$, then $|H| \geq m + 1$.

[Hint: Construct some orthogonal vectors in $\{-1, 1\}^{|H|}$ *based on the hash functions in* H*.]*

◀