Homework #5

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

Name: 徐靖, ID: 2200012917

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 $\eta > 0$, the number of experts N

$\boldsymbol{L}_0 \leftarrow 0^N$ ▷ Cumulative loss vector.

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

  $W(t) = \sum_i \exp(-\eta \boldsymbol{L}_{t-1}(i))$ ▷ Normalization coefficient.

  Select the i-th expert with probability $\boldsymbol{p}_t(i) = \frac{\exp(-\eta \boldsymbol{L}_{t-1}(i))}{W(t)}$

  Observe the loss vector $\boldsymbol{l}_t \in [0,1]^N$ for each expert ▷ The loss is guaranteed in $[0,1]$.

  Update the cumulative loss $\boldsymbol{L}_t \leftarrow \boldsymbol{L}_{t-1} + \boldsymbol{l}_t$

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:

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

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: $\mathrm{e}^{-x} \leq 1 - x + x^2, x > 0$.]

◀

Answer. (1) Actually,

$$\frac{W(t+1)}{W(t)} = \frac{\sum_i \exp(-\eta \boldsymbol{L}_t(i))}{W(t)} = \sum_i \frac{\exp(-\eta L_{t-1}(i))}{W(t)} \cdot \exp(-\eta l_t) = \boldsymbol{p}_t^\top \exp(-\eta \boldsymbol{l}_t)$$

(2) Given that,

$$\forall i, \exp(-\eta \boldsymbol{L}_t(i)) \leq W(T+1) = W(1) \prod_{t=1}^T \boldsymbol{p}_t^\top \exp(-\eta \boldsymbol{l}_t) \leq W(1) \prod_{t=1}^T \sum_{i=1}^n \boldsymbol{p}_t(i)(1 - \eta \boldsymbol{l}_t(i) + \eta^2 \boldsymbol{l}_t^2(i))$$

Take the logarithm of both sides, we have,

$$-\eta \boldsymbol{L}_t(i) \leq \ln N + \sum_{t=1}^{T} \ln\left(1 - \boldsymbol{p}_t^\top \boldsymbol{l}_t \eta + \sum_{i=1}^{n} \boldsymbol{p}_t(i) \boldsymbol{l}_t^2(i) \eta^2\right) \leq \ln N + \sum_{t=1}^{T} -\boldsymbol{p}_t^\top \boldsymbol{l}_t \eta + \eta^2$$

After sorting, we found,

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

◁

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 $\boldsymbol{w}$).

$\boldsymbol{w}_1 \leftarrow (1, 1, \cdots, 1)$ ▷ 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, $h_{t+1} \neq h_t$.

◀

Answer. (1) Assume that h_t is given to booster for K rounds, then after this round,

$$\beta_{t+K} > \frac{1}{2} - \gamma \geq \beta_{t+K-1}$$

And we find that for $k \in [K]$, we have:

$$\begin{aligned}
\beta_{t+k} &= \frac{\sum_{h_t(x_i)\neq y_i} w_{t+k}(i)}{\sum_{h_t(x_i)\neq y_i} w_{t+k}(i) + \sum_{h_t(x_i)=y_i} w_{t+k}(i)} \\
&= \frac{\sum_{h_t(x_i)\neq y_i} w_{t+k}(i)\alpha}{\sum_{h_t(x_i)\neq y_i} w_{t+k-1}(i)\alpha + \sum_{h_t(x_i)=y_i} w_{t+k-1}(i)} \\
&= \frac{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha^k}{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha^k + \sum_{h_t(x_i)=y_i} w_t(i)}
\end{aligned}$$

On the other hand,

$$\beta_t = \frac{\sum_{h_t(x_i)\neq y_i} w_t(i)}{\sum_{h_t(x_i)\neq y_i} w_t(i) + \sum_{h_t(x_i)=y_i} w_t(i)}$$

Thus,

$$\beta_{t+k} = \frac{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha^k}{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha^k + \frac{\sum_{h_t(x_i)\neq y_i} w_t(i)}{\beta_t} - \sum_{h_t(x_i)\neq y_i} w_t(i)} = \frac{\alpha^k}{\alpha^k + \frac{1}{\beta_t} - 1}$$

Therefore, K must satisfy:

$$\frac{1}{1 + (\frac{1}{\beta_t} - 1)\frac{1}{\alpha^K}} > \frac{1}{2} - \gamma \geq \frac{1}{1 + (\frac{1}{\beta_t} - 1)\frac{1}{\alpha^{K-1}}}$$

After sorting, we found,

$$K = \left\lceil \log_\alpha \left(\frac{1}{\beta_t} - 1 \right) \right\rceil + 1$$

(2) Assume h_t has error rate β' on the weighted sample (S, w_{t+1}). Then we have,

$$\beta' = \frac{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha_t}{\sum_{h_t(x_i)\neq y_i} w_t(i)\alpha_t + \sum_{h_t(x_i)=y_i} w_t(i)} = \frac{\alpha_t}{\alpha_t + \frac{1}{\beta_t} - 1} = \frac{1}{2} > \frac{1}{2} - \gamma$$

Thus $h_{t+1} \neq h_t$

◁

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.

◀

Answer. (1) Only one value is recorded. Each time a_i is passed in, the recorded value is changed to a_i with a probability of $\frac{1}{i}$. Finally, the recorded value is output. The distribution of this value is uniform on $\{a_i\}$. This algorithm uses $\log m$ of memory.

(2) Record two values, called sum and a. Each time a_i is passed in, first let sum increase by a_i^2, then change a to a_i with a probability of $\frac{a_i^2}{sum}$, and finally output the recorded value. The distribution of this value meets the requirements of the question. This algorithm uses $\log n + \log m$ of memory.

(3) Following are the steps.

- Create an array $\{r_i\}_t$ and copy first t items of stream to it.
- Now one by one consider all items from $(t+1)$th item to nth item.
  - Generate a random number from 1 to i where i is the index of the current item in stream. Let the generated random number is j.
  - If j is in range 1 to k, replace r_j with a_i

Given that for a_i, let $A_j(i)$ donates r_j had been replaced by a_i, $B_j(i)$ donates r_j has never been replaced after a_i passed. Thus the possibility for $a_i \in X$ is that:

$$\sum_{j=1}^{t} \mathbb{P}(A_j(i)B_j(i)) = \sum_{j=1}^{t} \mathbb{P}(A_j(i))\mathbb{P}(B_j(i)|A_j(i)) = \sum_{j=1}^{t} \frac{1}{i} \prod_{k=i+1}^{n} (1 - \frac{1}{k}) = \frac{t}{n}$$

This algoritm is correct.

◁

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 $\frac{n}{k+1}$.

◀

Answer. (1) If there is no majority element, since the algorithm always has an output, the output would be a false positive.

(2) If we have $k+1$ elements, and we are getting the data with round $1, 2, \ldots, k+1$, then at last there will be no element stored in our list. Thus the algorithm under counts all of the items 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.] ◀

Answer. (1) (a) Yes. When m and M are relatively prime, ax and b both traverse the remainder system modulo M, which means that I can adjust a, b so that $(h(x), h(y))$ is any (i, j), with equal probability.

(b) Yes. The same as (a).

(2) (a) No. If $M > 2$, let $x = 1, y = 2$, then $\mathbb{P}(h_{ab}(x) = 1, h_{ab}(y) = 1) \geq \frac{2}{M^2}$ (when $(a, b) = (0, 0), (M-1, 0)$). If $M = 2$, the answer is ordinary.

(b) No. Let $x = 1, y = M$, then $h_{ab}(1) = h_{ab}(M)$, thus $\mathbb{P}(h_{ab}(1) = 1, h_{ab}(M) = 0) = 0$

(c) No. As long as there exists $x \neq y \in \{0, 1, \ldots, n-1\}$ s.t. $x \equiv y \pmod M$, then there exists $\mathbb{P}(h_{ab}(x) = c, h_{ab}(y) = c) \geq \frac{2}{M^2}$. There are many such M, for example, 7,19.

◁

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. ◀

Answer. Let $\boldsymbol{x} = (1,2,3,4)^\top$, then construct $\boldsymbol{h}_i(\boldsymbol{x})$ as follows:

$$\boldsymbol{h}_1(\boldsymbol{x}) = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}, \quad \boldsymbol{h}_2(\boldsymbol{x}) = \begin{pmatrix}2\\2\\2\\2\end{pmatrix}, \quad \boldsymbol{h}_3(\boldsymbol{x}) = \begin{pmatrix}3\\3\\3\\3\end{pmatrix}, \quad \boldsymbol{h}_4(\boldsymbol{x}) = \begin{pmatrix}4\\4\\4\\4\end{pmatrix},$$

$$\boldsymbol{h}_5(\boldsymbol{x}) = \begin{pmatrix}1\\2\\3\\4\end{pmatrix}, \quad \boldsymbol{h}_6(\boldsymbol{x}) = \begin{pmatrix}2\\3\\1\\4\end{pmatrix}, \quad \boldsymbol{h}_7(\boldsymbol{x}) = \begin{pmatrix}3\\4\\1\\2\end{pmatrix}, \quad \boldsymbol{h}_8(\boldsymbol{x}) = \begin{pmatrix}4\\1\\3\\2\end{pmatrix},$$

$$\boldsymbol{h}_9(\boldsymbol{x}) = \begin{pmatrix}1\\3\\4\\2\end{pmatrix}, \quad \boldsymbol{h}_{10}(\boldsymbol{x}) = \begin{pmatrix}2\\1\\4\\3\end{pmatrix}, \quad \boldsymbol{h}_{11}(\boldsymbol{x}) = \begin{pmatrix}3\\1\\2\\4\end{pmatrix}, \quad \boldsymbol{h}_{12}(\boldsymbol{x}) = \begin{pmatrix}4\\3\\2\\1\end{pmatrix},$$

$$\boldsymbol{h}_{13}(\boldsymbol{x}) = \begin{pmatrix}1\\4\\2\\3\end{pmatrix}, \quad \boldsymbol{h}_{14}(\boldsymbol{x}) = \begin{pmatrix}2\\4\\3\\1\end{pmatrix}, \quad \boldsymbol{h}_{15}(\boldsymbol{x}) = \begin{pmatrix}3\\2\\4\\1\end{pmatrix}, \quad \boldsymbol{h}_{16}(\boldsymbol{x}) = \begin{pmatrix}4\\2\\1\\3\end{pmatrix}.$$

In fact, except for the first four functions with only one element in the range, the other 12 are permutations of $(1,2,3,4), (1,4,2,3), (1,3,4,2)$. We note that for any $i,j \in [4]$, these 12 sets of permutations just traverse all pairwise position relationships, which means that $P(h_{ab}(x) = i, h_{ab}(y) = j) = \frac{1}{16}$ holds for $i = j, i \neq j$ (the frequencies are given by $h_1, \ldots, h_4$ and $h_5, \ldots, h_{16}$ respectively), so the given H meets the requirements of the question. ◁

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.]

◀

Answer.

(1) By contradiction, if $|H| < M^2$, for $x \neq y, w, z$, if there exists $H_0 \subset H$, such that $h \in H$ is equivalent to $h(x) = w, h(y) = z$, and obviously H_0 is not empty, then

$$P(h(x) = w, h(y) = z) = \frac{|H_0|}{|H|} \geq \frac{1}{|H|} > \frac{1}{M^2}$$

So $|H| \geq M^2$

(2) For H and $\forall i \in [m]$, we construct the vector $\boldsymbol{v}_i = (2h_1(i) - 3, \ldots, 2h_{|H|}(i) - 3)$. Since H is 2-universal, we have

$$\boldsymbol{v}_i \cdot \boldsymbol{v}_j = \sum_{k \in [|H|]} h_k(i)h_k(j) = \sum_{h_k(i)=h_k(j), k \in [|H|]} 1 + \sum_{h_k(i) \neq h_k(j), k \in [|H|]} -1 = 0$$

So we get m pairwise orthogonal vectors on $\{-1, 1\}^{|H|}$ In addition, we note that $P(h(i) = 1) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, which leads to $\|v_i\|_1 = 0$. Let $\boldsymbol{v}_0 = (1, \ldots, 1)$, so:

$$\boldsymbol{v}_0 \cdot \boldsymbol{v}_i = \|v_i\|_1 = 0$$

So we have found the $m + 1$th orthogonal vector, and obviously $|H| \geq m + 1$. ◁