Machine Learning Techniques Homework #4

tags: `mlt` , `ntu` , `course` , `homework`

Cheng-Shih Wong, R04945028, mob5566@gmail.com *(mailto:mob5566@gmail.com)*

1.

For N is larget, the probability of one of the examples not being sampled is

$$\begin{aligned}\left(1-\frac{1}{N}\right)^{N'} &= \frac{1}{\left(\frac{N}{N-1}\right)^{N'}} \\ &= \frac{1}{\left(\frac{N}{N-1}\right)^{pN}} \\ &= \left(\frac{1}{\left(\frac{N}{N-1}\right)^{N}}\right)^{p} \\ &\approx \left(\frac{1}{e}\right)^{p} \\ &= e^{-p}\end{aligned}$$

Therefore, there are probably $e^{-p} \cdot N$ of the examples will not be sampled.

2.

Because $\sum_{k=1}^{3} E_{\text{out}}(g_k) = 0.75 \leq 1$, then we can obtain the minimum $E_{\text{out}}(G) = 0$ by covering an error prediction by other two correct predictions.

To obtain the maximum $E_{\text{out}}(G)$, we have to combine **exact two error predictions with one correct prediction** as much as possible.

Denote $\{d_k\}_{k=1}^{4}$ that $\bigcup\limits_{k=1}^{4} d_k = D$ and $d_i \cap d_j = \phi \text{ for } i \neq j \text{ and } 1 \leq i,j \leq 4$, where D is test data.

We can have

$|d_1| = 0.025|D|$ and $E_{d_1}(g_1) = E_{d_1}(g_2) = 1$ and $E_{d_1}(g_3) = 0$
$|d_2| = 0.125|D|$ and $E_{d_2}(g_1) = E_{d_2}(g_3) = 1$ and $E_{d_2}(g_2) = 0$
$|d_3| = 0.225|D|$ and $E_{d_3}(g_2) = E_{d_3}(g_3) = 1$ and $E_{d_3}(g_1) = 0$

That is $E_{d_1 \cup d_2 \cup d_3}(G) = 0$ and $E_{d_4}(G) = 1$, where $|d_4| = 0.625|D|$.

Therefore, the maximum possible $E_{\text{out}}(G)$ is 0.375.

The possible range of $E_{\text{out}}(G)$

$$0 \leq E_{\text{out}}(G) \leq 0.375$$

3

Let N be the number of examples, then the total error predictions are $N \cdot \sum_{k=1}^{K} e_k$.

To get the upper bound of $E_{\text{out}}(G)$, we can exploit a greedy though described below.

For an example x_n, if there are exact $\frac{K+1}{2}$ binary classification trees predict wrongly on x_n, then G will also predict wrongly on x_n.

Therefore, for the maximum possible error predictions of G, we have at most $\frac{N \cdot \sum_{k=1}^{K} e_k}{\frac{K+1}{2}}$ examples that will be predicted wrongly.

Finally, the upper bound of $E_{\text{out}}(G)$ is

$$\begin{aligned} E_{\text{out}}(G) &\leq \frac{\frac{2N \cdot \sum_{k=1}^{K} e_k}{K+1}}{N} \\ &\leq \frac{2}{K+1} \sum_{k=1}^{K} e_k \end{aligned}$$

4

From lecture 211 slide page 17, α_1 is optimal η that

$$
\begin{aligned}
\min_{\eta} \frac{1}{N} \sum_{n=1}^{N} ((y_n - s_n) - \eta g_1(\mathbf{x}_n))^2 &= \min_{\eta} \frac{1}{N} \sum_{n=1}^{N} ((y_n - 0) - 2\eta)^2 \\
&= \min_{\eta} \frac{1}{N} \sum_{n=1}^{N} (y_n - 2\eta)^2 \\
&\Rightarrow \frac{-2}{N} \sum_{n=1}^{N} (y_n - 2\alpha_1) = 0 \\
&\Rightarrow -2N\alpha_1 + \sum_{n=1}^{N} y_n = 0 \\
&\Rightarrow \alpha_1 = \frac{\sum_{n=1}^{N} y_n}{2N}
\end{aligned}
$$

$$
\begin{aligned}
\because s_n &= \alpha_1 g_1(\mathbf{x}_n) \\
&= \left(\frac{\sum_{n=1}^{N} y_n}{2N} \right) \cdot 2 \\
&= \frac{\sum_{n=1}^{N} y_n}{N}
\end{aligned}
$$

5

Let $s_n^0 = 0$ and $s_n^t = s_n^{t-1} + \alpha_t g_t(\mathbf{x}_n)$ after t iterations.

At t-th iteration, α_t is the steepest η that

$$
\min_{\eta} \frac{1}{N} \sum_{n=1}^{N} \left((y_n - s_n^{t-1}) - \eta g_t(\mathbf{x}_n) \right)^2 \tag{5.1}
$$

To obtain α_t, let the derivative of (5.1) equal to 0.

$$
\begin{aligned}
\frac{-2}{N}\sum_{n=1}^{N}\left((y_n - s_n^{t-1}) - \alpha_t g_t(\mathbf{x}_n)\right) g_t(\mathbf{x}_n) &= 0 \\
\Rightarrow \sum_{n=1}^{N}\left(y_n - (s_n^{t-1} + \alpha_t g_t(\mathbf{x}_n))\right) g_t(\mathbf{x}_n) &= 0 \\
\Rightarrow \sum_{n=1}^{N}\left(y_n - s_n^{t}\right) g_t(\mathbf{x}_n) &= 0 \\
\Rightarrow \sum_{n=1}^{N} y_n g_t(\mathbf{x}_n) - \sum_{n=1}^{N} s_n^t g_t(\mathbf{x}_n) &= 0 \\
\therefore \sum_{n=1}^{N} s_n^t g_t(\mathbf{x}_n) &= \sum_{n=1}^{N} y_n g_t(\mathbf{x}_n)
\end{aligned}
$$

6

First, we have

$$s_1 = s_2 = \cdots = s_N = 0$$

For convenience, we augment the input $\mathbf{x}$ to $\mathbf{x}'$ with a constant dimension 1.

We are to solve the linear regression problem

$$\min_{\mathbf{w}} \sum_{n=1}^{N}\left((y_n - s_n) - \mathbf{w}\mathbf{x}_n'\right)^2 = \min_{\mathbf{w}} \sum_{n=1}^{N}\left(y_n - \mathbf{w}\mathbf{x}_n'\right)^2 \tag{6.1}$$

$$\text{Let } \mathbf{w}^* \text{ be the optimal } \mathbf{w} \text{ minimize (6.1)} \tag{6.2}$$
$$\text{and } g_1(\mathbf{x}') = \mathbf{w}^*\mathbf{x}'$$

Then we have to find α_1 as the steepest η that minimize

$$\min_{\eta} \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \eta(\mathbf{w}^*\mathbf{x}_n')\right)^2$$

Assume that α_1 is the optimal η and $\alpha_1 \neq 1$.

Then we have $\bar{\mathbf{w}} = \alpha_1 \mathbf{w}^*$ that minimize

$$\sum_{n=1}^{N}\left(y_n - \alpha_1(\mathbf{w}^*\mathbf{x}_n')\right)^2 = \sum_{n=1}^{N}\left(y_n - \bar{\mathbf{w}}\mathbf{x}_n'\right)^2$$

and $\bar{\mathbf{w}} = \alpha_1 \mathbf{w}^* \neq \mathbf{w}^*$ contradict to (6.2).

Therefore, $\alpha_1 = 1$.

7

Continue from **Q.6**, we update all s_n by $s_n = \alpha_1 g_1(\mathbf{x}'_n) = \mathbf{w}^* \mathbf{x}'_n$.

For the second iteration, we have to solve the linear regression problem

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \left((y_n - \mathbf{w}^* \mathbf{x}'_n) - g_2(\mathbf{x}'_n) \right)^2 = \min_{\mathbf{w}} \sum_{n=1}^{N} \left((y_n - \mathbf{w}^* \mathbf{x}'_n) - \mathbf{w} \mathbf{x}'_n \right)^2 \tag{7.1}$$

Assume the optimal $\mathbf{w} = \hat{\mathbf{w}} \neq 0$ with minimum value of (7.1) as

$$\sum_{n=1}^{N} \left((y_n - \mathbf{w}^* \mathbf{x}'_n) - \hat{\mathbf{w}} \mathbf{x}'_n \right)^2 = \sum_{n=1}^{N} \left(y_n - (\mathbf{w}^* + \hat{\mathbf{w}}) \mathbf{x}'_n \right)^2$$

Therefore, $\mathbf{w}^* + \hat{\mathbf{w}}$ minimizes (6.1), but $\mathbf{w}^* + \hat{\mathbf{w}} \neq \mathbf{w}^*$ contradicts to (6.2).

Finally, $\hat{\mathbf{w}} = 0$, that is $g_2(\mathbf{x}') = 0$.

8

To implement $\mathtt{OR}(x_1, x_2, \ldots, x_d)$, we know that there is only one condition that $\sum_{i=0}^{d} w_i x_i < 0$, when $x_1 = x_2 = \cdots = x_d = -1$.

Let x_0 be the constant input 1, and $w_1 = w_2 = \cdots = w_d = 1$.

We will have $\sum_{i=1}^{d} w_i x_i = -d$ which is the minimum value under all x_i combinations.

After that, we let $w_0 = d - 1$, then

$$\sum_{i=0}^{d} w_i x_i = \begin{cases} -1, & \text{if } x_1 = x_2 = \cdots = x_d = -1 \\ 2 * (\text{number of postive } x_i) - 1, & \text{else} \end{cases}$$

Therefore, we have

$$g_A(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i \right) = \begin{cases} -1, & \text{if } x_1 = x_2 = \cdots = x_d = -1 \\ +1, & \text{else} \end{cases}$$

with

$$w_0 = d - 1, w_1 = w_2 = \cdots = w_d = 1$$

9

To implement $\mathtt{XOR}((x)_1, (x)_2, (x)_3, (x)_4, (x)_5)$, we first recognize that the “exclusive or operation” will judge whether the number of positive x_i is odd or not.

$$\sum_{i=1}^{5} x_i = \begin{cases} -5, & \text{if the number of positive } x_i = 0 \\ -3, & \text{if the number of positive } x_i = 1 \\ -1, & \text{if the number of positive } x_i = 2 \\ +1, & \text{if the number of positive } x_i = 3 \\ +3, & \text{if the number of positive } x_i = 4 \\ +5, & \text{if the number of positive } x_i = 5 \end{cases}$$

We can split these six conditions with five threshold θ_k that $\sum_{i=1}^{5} x_i \geq \theta_k$ where $\theta_k \in \{-4, -2, 0, 2, 4\}$ and $k = 1, \ldots, 5$ respectively.

$$\text{sign}\left(\sum_{i=1}^{5} x_i - \theta_k\right) = \begin{cases} (-1, -1, -1, -1, -1), & \text{if the number of positive } x_i = 0 \\ (+1, -1, -1, -1, -1), & \text{if the number of positive } x_i = 1 \\ (+1, +1, -1, -1, -1), & \text{if the number of positive } x_i = 2 \\ (+1, +1, +1, -1, -1), & \text{if the number of positive } x_i = 3 \\ (+1, +1, +1, +1, -1), & \text{if the number of positive } x_i = 4 \\ (+1, +1, +1, +1, +1), & \text{if the number of positive } x_i = 5 \end{cases}$$

Then we give the weights to the results from last step with $w_k \in \{1, -1, 1, -1, 1\}$.

That is

$$\sum_{k=1}^{5} w_k \cdot \text{sign}\left(\sum_{i=1}^{5} x_i - \theta_k\right) = \begin{cases} -1, & \text{if the number of positive } x_i = 0 \\ +1, & \text{if the number of positive } x_i = 1 \\ -1, & \text{if the number of positive } x_i = 2 \\ +1, & \text{if the number of positive } x_i = 3 \\ -1, & \text{if the number of positive } x_i = 4 \\ +1, & \text{if the number of positive } x_i = 5 \end{cases}$$

which is my “exclusive or network” architecture.

As a result, the smallest size of the middle layer of this network $D = 5$.

10

All the initial weights $w_{ij}^{(l)} = 0$, then the score of every neuron $s_j^{(l)} = \sum_{i=0}^{d^{(l-1)}} w_{ij}^{(l)} x_i^{(l-1)} = 0$.

The square error function is

$$e_n = \left(y_n - \tanh\left(s_1^{(L)}\right)\right)^2$$

The partial derivative of weights of last layer are

$$\begin{aligned}
\frac{\partial e_n}{\partial w_{i1}^{(L)}} &= -2\left(y_n - \tanh\left(s_1^{(L)}\right)\right) \tanh'\left(s_1^{(L)}\right)\left(x_i^{(L-1)}\right) \\
&= -2\left(y_n - \tanh\left(s_1^{(L)}\right)\right)\left(1 - \tanh\left(s_1^{(L)}\right)\right)\left(x_i^{(L-1)}\right) \\
&= -2y_n\left(x_i^{(L-1)}\right)
\end{aligned}$$

For the input of last layer L, $x_i^{(L-1)} = \tanh\left(s_i^{(L-1)}\right) = \tanh(0) = 0$ except the constant input $x_0^{(L-1)} = 1$.

For the partial derivative of rest layers,

$$\begin{aligned}
\frac{\partial e_n}{\partial w_{ij}^{(l)}} &= \frac{\partial e_n}{\partial s_j^{(l)}} \cdot \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} \\
&= \delta_j^{(l)} \cdot \left(x_i^{(l-1)}\right) \\
&= \sum_{k=1}^{d^{(l+1)}} \left(\delta_k^{(l+1)}\right)\left(w_{jk}^{(l+1)}\right)\left(\tanh'\left(s_j^{(l)}\right)\right)\left(x_i^{(l-1)}\right) \\
&= 0
\end{aligned}$$

Therefore, except the gradient component of bias weight at the last layer $\frac{\partial e_n}{\partial w_{01}^{(L)}} = -2y_n$ may be **not equal to** 0, **the rest of the gradient components** are all 0.

11

There is only one hidden layer, then the last layer of this network is layer 2.

Therefore, we have

$$\delta_1^{(2)} = -2\left(y_n - \tanh\left(s_1^{(2)}\right)\right)\left(1 - \tanh\left(s_1^{(2)}\right)\right)$$
$$\delta_j^{(1)} = \left(\delta_1^{(2)}\right)\left(w_{j1}^{(2)}\right)\left(1 - \tanh\left(s_j^{(1)}\right)\right)$$

Initialize all $w_{ij}^{(l)} = 1$, then

$$\because s_j^{(1)} = \sum_{i=0}^{d^{(0)}} w_{ij}^{(1)} x_i^{(0)} = \sum_{i=0}^{d^{(0)}} x_i^{(0)}$$
$$\therefore s_i^{(1)} = s_j^{(1)}, \forall 1 \le i, j \le d^{(1)}$$
$$\text{and } x_j^{(1)} = \tanh\left(s_j^{(1)}\right)$$
$$\therefore x_i^{(1)} = x_j^{(1)}, \forall 1 \le i, j \le d^{(1)}$$

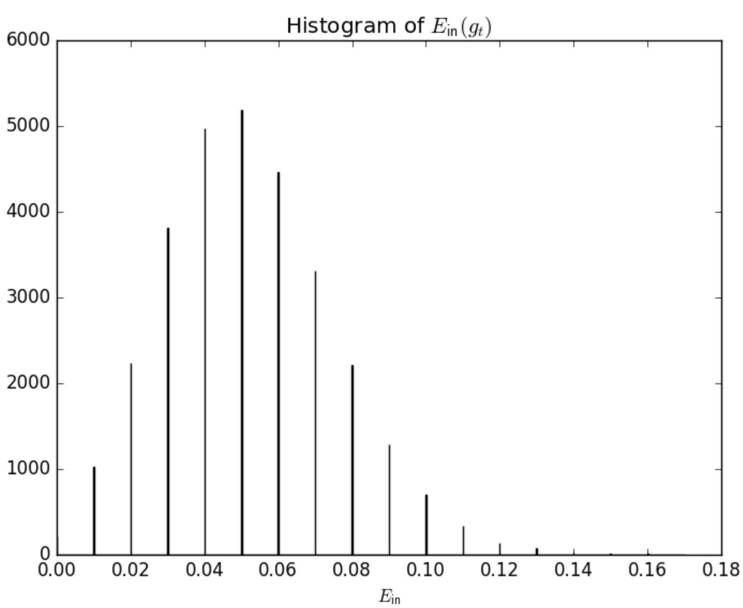
By the weight update rule from lecture 212 page 16

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta x_i^{(l-1)} \delta_j^{(l)}$$

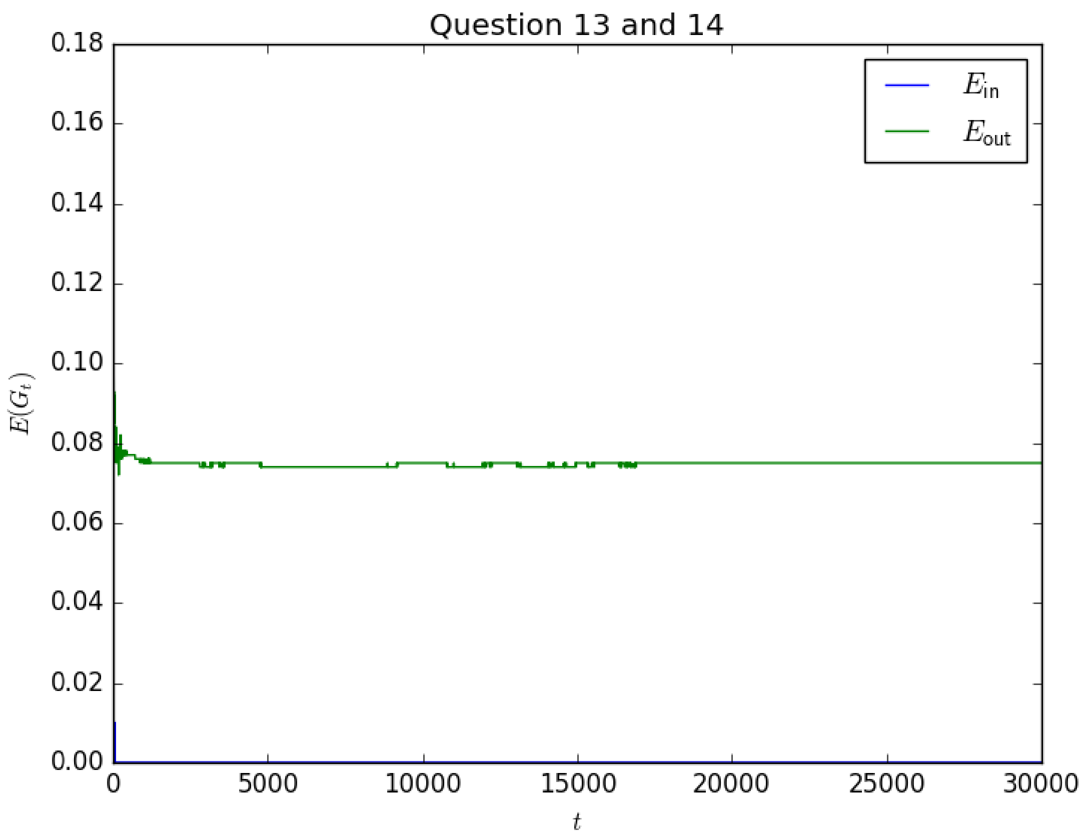
Then

$$w_{i1}^{(2)} \leftarrow w_{i1}^{(2)} - \eta x_i^{(1)} \delta_1^{(2)}$$
$$\therefore w_{i1}^{(2)} = w_{j1}^{(2)}, \forall 1 \le i, j \le d^{(1)}$$
$$\therefore \delta_i^{(1)} = \left(\delta_1^{(2)}\right)\left(w_{i1}^{(2)}\right)\left(1 - \tanh\left(s_i^{(1)}\right)\right) = \left(\delta_1^{(2)}\right)\left(w_{j1}^{(2)}\right)\left(1 - \tanh\left(s_j^{(1)}\right)\right) = \delta_j^{(1)}$$
$$\forall 1 \le i, j \le d^{(1)}$$
$$\therefore w_{ij}^{(1)} = 1 - \eta x_i^{(0)} \delta_j^{(1)} = 1 - \eta x_i^{(0)} \delta_{j+1}^{(1)} = w_{i(j+1)}^{(1)}$$
$$\forall i \text{ and } 1 \le j < d^{(1)}$$

12



13 & 14



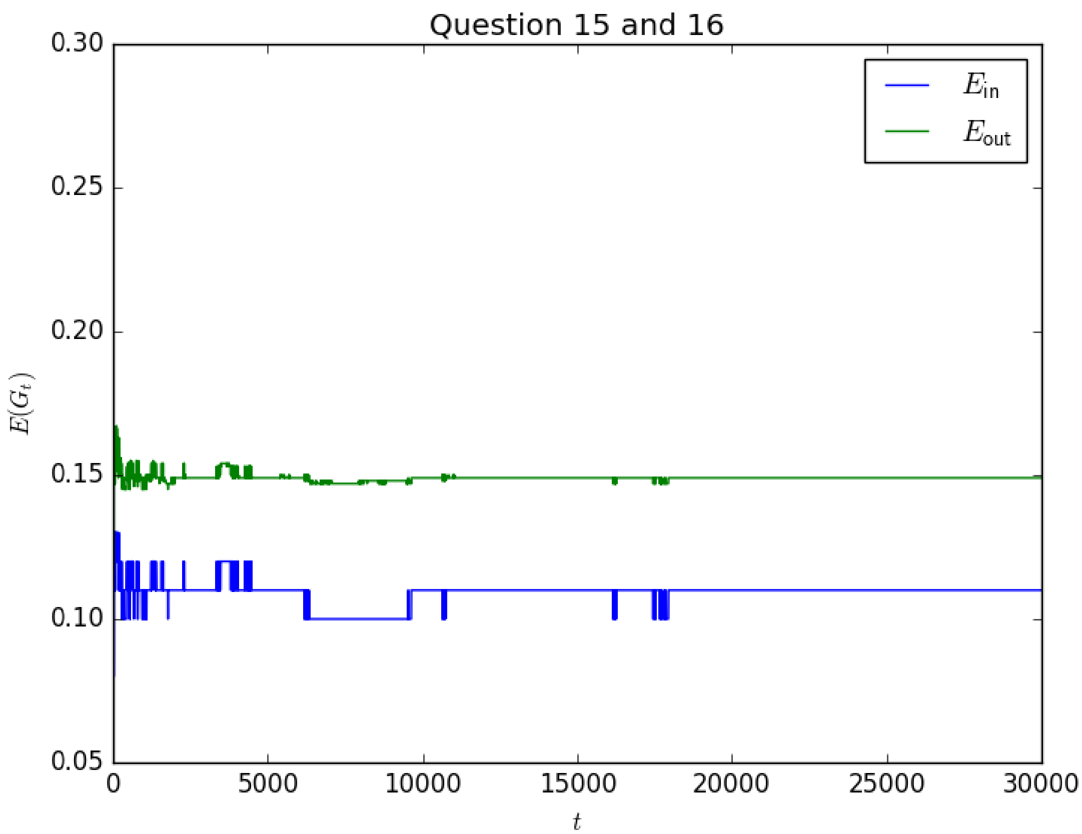
With bagging full grown trees, the in-sample error can down to 0.

However, the out-sample error converges at about 0.075.

When we aggregate more decision trees, the errors (in-sample and out-sample) goes lower and more stable.

😄
😄
😄
😄
😄
😄
😄
😄
😄

15 & 16



When we aggregate one-branch decision trees, the in-sample error converges at 0.11, and the out-sample error converges at about 0.15.

Obviously, the both errors rise a lot than the fully-grown trees, because the individual one-branch decision tree is restricted by the height of tree.

Therefore, the average error of one-branch decision tree is larger than fully-grown tree.

Another strange finding is that the curves of in-sample error and out-sample error are similar.

That is, the out-sample error rises up, when in-sample rises up; and vise versa.