HTML HW3

B12901022 廖冠豪

In this problem we use $\{-1, +1\}$ to denote boolean values (outcome of binary classification)

5

We consider hypotheses $h_1, h_2 : \mathbb{R} \to \{-1, +1\}$, where

$$h_1(x) = \text{sign}(x)$$
$$h_2(x) = -\text{sign}(x)$$

and hypotheses sets $\mathcal{H}_1 = \{h_1\}$ and $\mathcal{H}_2 = \{h_2\}$.
As both $\mathcal{H}_1$ and $\mathcal{H}_2$ only contain one hypothesis, their VC dimension are both 0.

$$d_{VC}(\mathcal{H}_1) = d_{VC}(\mathcal{H}_2) = 0$$

Now consider the hypotheses set $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$, and $x = 1$. We have $h_1(x) = +1$ and $h_2(x) = -1$, so $\mathcal{H}$ can shatter a set of one data vector and its VC dimension is hence greater than or equal to 1.
We conclude that

$$d_{VC}(\mathcal{H}_1 \cup \mathcal{H}_2) \geq 1 > d_{VC}(\mathcal{H}_1) + d_{VC}(\mathcal{H}_2) = 0$$

Therefore we have disproved the statement.

In this problem we use $\{-1, +1\}$ to denote boolean values (outcome of binary classification)

6

In the super-market case, we hope to find $f(\mathbf{x})$ that minimizes

$$\mathop{\mathbb{E}}_{\mathbf{x},y\sim P(\mathbf{x},y)}[10P(+1|\mathbf{x})[\![f(\mathbf{x}=-1)]\!]+P(-1|\mathbf{x})[\![f(\mathbf{x}=+1)]\!]]$$

We can see that for any $\mathbf{x}$ exactly one out of $f(\mathbf{x})=+1$ and $f(\mathbf{x})=-1$ is true, so it would be optimal to choose an α such that the one that contributes to a smaller value in $\mathbb{E}$ is always chosen(i.e. $f(\mathbf{x})=-1$ if $10P(+1|\mathbf{x})$ is greater than $P(-1|\mathbf{x})$, and vice versa)

$$\begin{cases} P(+1|\mathbf{x})+P(-1|\mathbf{x})=1 \\ 10P(+1|\mathbf{x})\geq P(-1|\mathbf{x}) \end{cases}$$
$$\implies P(+1|\mathbf{x})\geq \frac{1}{11}$$

By the analysis above, we can see that the optimal choice of α is $\frac{1}{11}$, and we have the mini-target

$$f_{\text{MKT}}(\mathbf{x})=\text{sign}(P(y=+1|\mathbf{x})-\frac{1}{11})$$

In this problem we use $\{-1, +1\}$ to denote boolean values (outcome of binary classification)

7

$$
\begin{aligned}
E_{\text{out}}^{(1)}(h) &= \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) [\![h(\mathbf{x}) \neq f(\mathbf{x})]\!] \\
E_{\text{out}}^{(2)}(h) &= \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) \left(P(-1|\mathbf{x}) [\![h(\mathbf{x}) = +1]\!] + P(+1|\mathbf{x}) [\![h(\mathbf{x}) = -1]\!] \right) \\
E_{\text{out}}^{(2)}(f) &= \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) \left(P(-1|\mathbf{x}) [\![f(\mathbf{x}) = +1]\!] + P(+1|\mathbf{x}) [\![f(\mathbf{x}) = -1]\!] \right)
\end{aligned}
$$

We see that

$$
E_{\text{out}}^{(1)}(h) + E_{\text{out}}^{(2)}(f) = \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) \left(P(-1|\mathbf{x}) [\![f(\mathbf{x}) = +1]\!] + P(+1|\mathbf{x}) [\![f(\mathbf{x}) = -1]\!] + [\![h(\mathbf{x}) \neq f(\mathbf{x}))]\!] \right)
$$

Given $h(\mathbf{x}), f(\mathbf{x})$, define

$$
\begin{aligned}
a(\mathbf{x}) &= P(-1|\mathbf{x}) [\![f(\mathbf{x}) = +1]\!] + P(+1|\mathbf{x}) [\![f(\mathbf{x}) = -1]\!] + [\![h(\mathbf{x}) \neq f(\mathbf{x}))]\!] \\
b(\mathbf{x}) &= P(-1|\mathbf{x}) [\![h(\mathbf{x}) = +1]\!] + P(+1|\mathbf{x}) [\![h(\mathbf{x}) = -1]\!]
\end{aligned}
$$

We then consider the four different cases for $h(\mathbf{x}), f(\mathbf{x})$.

Case 1: $h(\mathbf{x}) = +1, f(\mathbf{x}) = +1$

$$
\begin{aligned}
a(\mathbf{x}) &= P(-1|\mathbf{x}) \\
b(\mathbf{x}) &= P(-1|\mathbf{x})
\end{aligned}
$$

Case 2: $h(\mathbf{x}) = +1, f(\mathbf{x}) = -1$

$$\begin{aligned} a(\mathbf{x}) &= P(+1|\mathbf{x}) + 1 \\ b(\mathbf{x}) &= P(-1|\mathbf{x}) \end{aligned}$$

Case 3: $h(\mathbf{x}) = -1, f(\mathbf{x}) = -1$

$$\begin{aligned} a(\mathbf{x}) &= P(+1|\mathbf{x}) \\ b(\mathbf{x}) &= P(+1|\mathbf{x}) \end{aligned}$$

Case 4: $h(\mathbf{x}) = -1, f(\mathbf{x}) = +1$

$$\begin{aligned} a(\mathbf{x}) &= P(-1|\mathbf{x}) + 1 \\ b(\mathbf{x}) &= P(+1|\mathbf{x}) \end{aligned}$$

We can see that for all possible cases we have $a(\mathbf{x}) \geq b(\mathbf{x})$.
Hence

$$E_{\text{out}}^{(1)}(h) + E_{\text{out}}^{(2)}(f) = \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) a(\mathbf{x}) \geq E_{\text{out}}^{(2)}(h) = \sum_{\mathbf{x} \sim P(\mathbf{x})} P(\mathbf{x}) b(\mathbf{x})$$

$\mathbb{Q}.\mathbb{E}.\mathbb{D}.$

Note that when dealing with a continuous distribution of $\mathbf{x}$*, we only need to change the summation to integral , and the following argument applies.*

8

Assuming that $X^T X$ is invertible, we can express $\mathbf{w}_{\text{LIN}}$ in the following form

$$\mathbf{w}_{\text{LIN}} = (X^T X)^{-1} X^T \mathbf{y}$$

Now if we replace x_0 with 1126, we get the new X matrix

$$X' = XD'$$

Where D' is the diagonal matrix with

$$D'_{ij} = \begin{cases} 1126 & i = j = 0 \\ 1 & i = j \neq 0 \\ 0 & \text{else} \end{cases}$$

Clearly ${D'}^{-1}$ exists, and D' is symmetric, so

$$({X'}^T X')({D'}^{-1} {(X^T X)}^{-1} {D'}^{-1}) = D'(X^T X) D' {D'}^{-1} {(X^T X)}^{-1} {D'}^{-1} = I$$

So $({X'}^T X')$ is invertible with

$${({X'}^T X')}^{-1} = {D'}^{-1} {(X^T X)}^{-1} {D'}^{-1}$$

Hence we can repeat the linear regression procedure and obtain

$$\begin{aligned} \mathbf{w}_{\text{LUCKY}} &= {({X'}^T X')}^{-1} {X'}^T \mathbf{y} \\ &= {D'}^{-1} {(X^T X)}^{-1} {D'}^{-1} {D'}^{-1} D' X^T \mathbf{y} \\ &= {D'}^{-1} {(X^T X)}^{-1} {D'}^{-1} X^T \mathbf{y} \\ &= {D'}^{-1} \mathbf{w}_{\text{LIN}} \end{aligned}$$

Hence we have proved the statement and found the diagonal matrix $D = {D'}^{-1}$

$$Dij = \begin{cases} \frac{1}{1126} & i = j = 0 \\ 1 & i = j \neq 0 \\ 0 & \text{else} \end{cases}$$

In this problem we use $\{-1,+1\}$ to denote boolean values (outcome of binary classification)

9

Let $f(\mathbf{x})$ be the target function we want to approximate with $\tilde{h}(\mathbf{x})$.
Consider $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_N, y_N), \}, \quad y_i \in \{-1, +1\}$.
The probability that f generates $\mathcal{D}$ is

$$P(\mathbf{x}_1)f(\mathbf{x}_1) \times P(\mathbf{x}_2)(1 - f(\mathbf{x}_2)) \times \dots \times P(\mathbf{x}_N)(1 - f(\mathbf{x}_N))$$

The likelihood that $\tilde{h}$ generates $\mathcal{D}$ is

$$P(\mathbf{x}_1)h(\mathbf{x}_1) \times P(\mathbf{x}_2)(1 - h(\mathbf{x}_2)) \times \dots \times P(\mathbf{x}_N)(1 - h(\mathbf{x}_N))$$

Also,

$$\begin{aligned} 1 - \tilde{h}(\mathbf{x}) &= 1 - \frac{1}{2}(\frac{\mathbf{w}^T\mathbf{x}}{\sqrt{1 + (\mathbf{w}^T\mathbf{x})^2}} + 1) \\ &= \frac{1}{2}(\frac{-\mathbf{w}^T\mathbf{x}}{\sqrt{1 + (\mathbf{w}^T\mathbf{x})^2}} + 1) \\ &= \tilde{h}(-\mathbf{x}) \end{aligned}$$

Therefore the likelihood for some hypothesis $\tilde{h}$ is proportional to

$$\prod_{n=1}^{N} \tilde{h}(y_n \mathbf{x}_n)$$

We hope to find

$$
\begin{aligned}
\arg\max_{\mathbf{w}} \prod_{n=1}^{N} \tilde{h}(y_n \mathbf{x}_n) &= \arg\max_{\mathbf{w}} \prod_{n=1}^{N} \left(\frac{1}{2} \left(\frac{y_n \left(\mathbf{w}^T \mathbf{x}_n\right)}{\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} + 1 \right) \right) \\
&= \arg\max_{\mathbf{w}} \ln \left(\prod_{n=1}^{N} \frac{1}{2} \left(\frac{y_n \left(\mathbf{w}^T \mathbf{x}_n\right)}{\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} + 1 \right) \right) \\
&= \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \ln \left(\frac{1}{2} \left(\frac{y_n \left(\mathbf{w}^T \mathbf{x}_n\right)}{\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} + 1 \right) \right) \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} - \ln \left(\frac{1}{2} \left(\frac{y_n \left(\mathbf{w}^T \mathbf{x}_n\right)}{\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} + 1 \right) \right) \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \mathrm{err}(\mathbf{w}, \mathbf{x}_n, y_n)
\end{aligned}
$$

Hence we've found the error function

$$
\begin{aligned}
\mathrm{err}(\mathbf{w}, \mathbf{x}_n, y_n) &= - \ln \left(\frac{1}{2} \left(\frac{y_n \mathbf{w}^T \mathbf{x}_n}{\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} + 1 \right) \right) \\
&= \ln \frac{2\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}}{y_n \mathbf{w}^T \mathbf{x}_n + \sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}}
\end{aligned}
$$

Therefore

$$
\tilde{E}_{\mathrm{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{2\sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}}{y_n \mathbf{w}^T \mathbf{x}_n + \sqrt{1 + (\mathbf{w}^T \mathbf{x}_n)^2}} \right)
$$

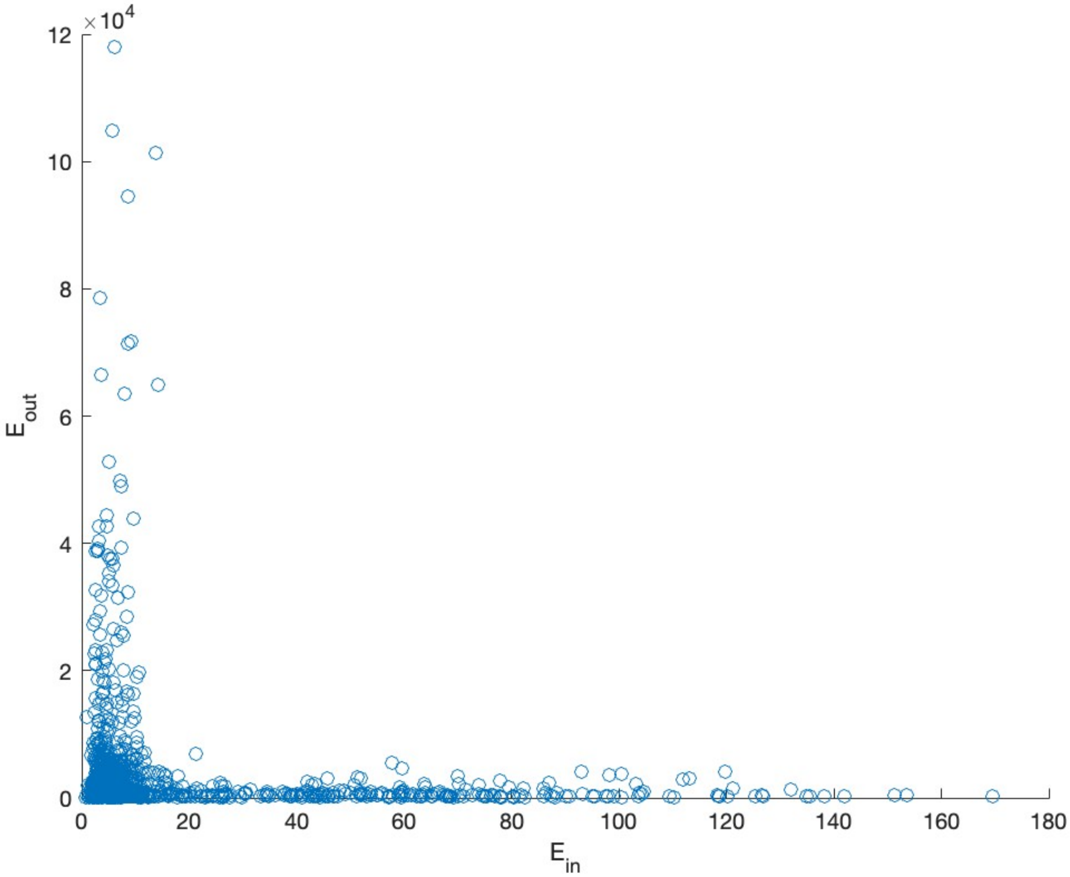
We can then compute $\nabla \tilde{E}_{\mathrm{in}}(\mathbf{w})$

$$
\begin{aligned}
(\nabla \tilde{E}_{\mathrm{in}}(\mathbf{w}))_i &= \frac{\partial \tilde{E}_{\mathrm{in}}(\mathbf{w})}{\partial \mathbf{w}_i} \\
&= \sum_{n=1}^{N} \frac{\partial \ln \bigcirc}{\partial \bigcirc} \frac{\partial \bigcirc}{\partial \square} \frac{\partial \square}{\partial \mathbf{w}_i} \qquad \left(\bigcirc = \left(\frac{2\sqrt{1 + \square^2}}{y_n \square + \sqrt{1 + \square^2}} \right), \square = \mathbf{w}^T \mathbf{x}_n \right) \\
&= \frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{\bigcirc} \right) \left(- \frac{2 y_n}{\sqrt{1 + \square^2} \left(y_n \square + \sqrt{1 + \square^2} \right)^2} \right) (\mathbf{x}_{n,i}) \\
&= \frac{1}{N} \sum_{n=1}^{N} \left(- \frac{y_n \mathbf{x}_{n,i}}{(1 + (\mathbf{w}^T x_n)^2)(y_n \mathbf{w}^T \mathbf{x}_n + \sqrt{1 + (\mathbf{w}^T \mathbf{w}_n)^2})} \right)
\end{aligned}
$$

Therefore we have

$$
\nabla \tilde{E}_{\mathrm{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left(- \frac{y_n \mathbf{x}_n}{(1 + (\mathbf{w}^T x_n)^2)(y_n \mathbf{w}^T \mathbf{x}_n + \sqrt{1 + (\mathbf{w}^T \mathbf{w}_n)^2})} \right)
$$

10



From the scatter plot we can see E_{out} is significantly larger than E_{out}, for the majority of data points E_{in} ranges between 0 to 2×10^4 and the maxima for E_{in} is about 1.2×10^5. On the other hand, for most data points E_{out} is less than 40, and the maxima of E_{out} is no less than 180.

This difference between E_{out} and E_{in} is consistent with theory, in lecture we're introduced with the equations

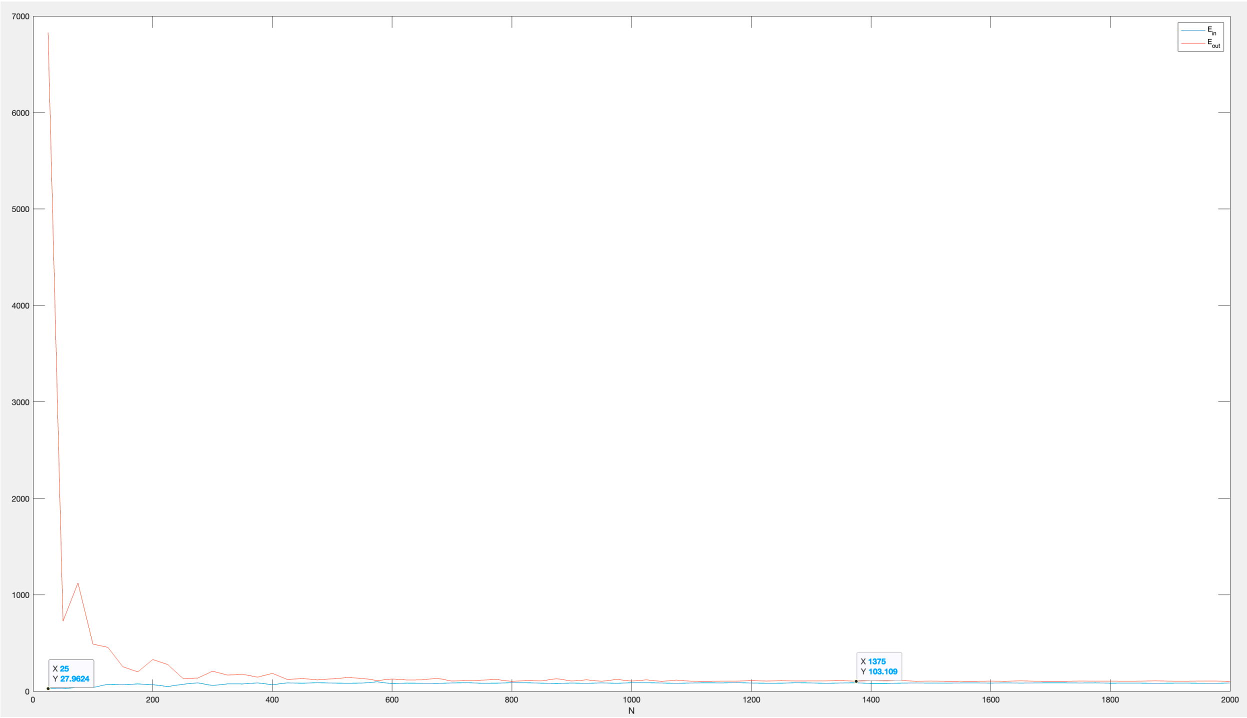
$$\overline{E_{out}} = \textbf{noise} \text{ level} \cdot (1 + \frac{d+1}{N})$$
$$\overline{E_{in}} = \textbf{noise} \text{ level} \cdot (1 - \frac{d+1}{N})$$

The expected generalization error is $\frac{2(d+1)}{N}$. In this case $N = 32$ is relatively small, so the difference between E_{out} an E_{in} is quite large.

```
data_vec = zeros(8192, 13);
y = zeros(1, 8192);
E_in = zeros(1126, 1);
E_out = zeros(1126, 1);
file = fopen("./data.txt");

line = fgetl(file);
line_count = 0;
while ischar(line)
    line_string = string(line);
    parts = strsplit(line_string, ' ');
    %disp(parts);
    y(line_count + 1) = str2double(parts(1));
    x = zeros(13, 1);
    x(1) = 1;
    %disp(length(parts));
    for j = 2:13
        foo = strsplit(parts(j), ':');
        x(j) = str2double(foo(2));
    end
    data_vec(line_count + 1, :) = x;
    line_count = line_count + 1;
    disp(line_count);
    line = fgetl(file);
end
%disp(data_vec(8192, :));


count = 1;
while count <= 1126
    in = 0;
    out = 0;
    indicies = zeros(1, 32);
    y_vec = zeros(32, 1);
    mat = zeros(32, 13);
    for i = 1:32
        r = randi([1, 8192]);
        indicies(1, i) = r;
        mat(i, :) = data_vec(r, :);
        y_vec(i, 1) = y(1, r);
    end
    w = pinv(mat) * y_vec;
    %disp(w);
    for i = 1:8192
        err = (dot(w, data_vec(i, :)) - y(1, i)) ^ 2;
        if ismember(i, indicies(1, :))
            in = in + err;
        else
            out = out + err;
        end
    end
    in = in / 32;
    out = out / 8160;
```

Code snapshot:

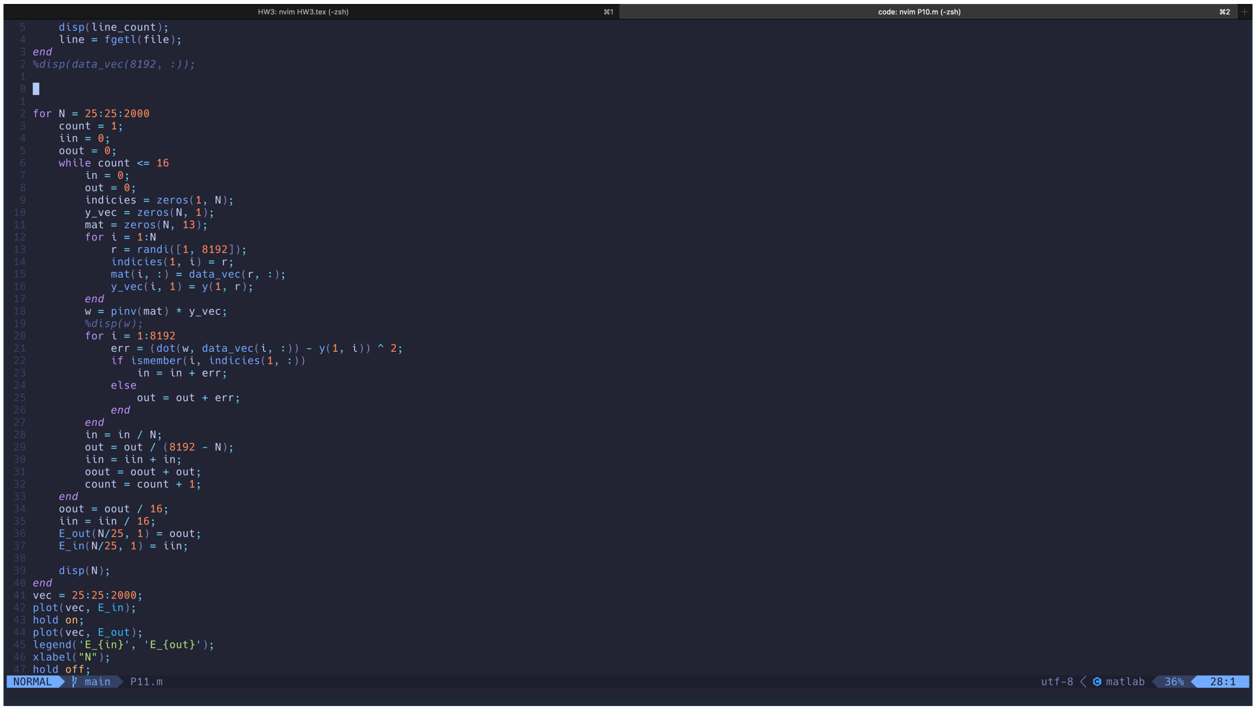
11



From the figure we see that for small N, as in problem 10, the difference between E_{out} and E_{in} is large, with E_{out} reaching almost 7000 and E_{in} approximately 27 for $N = 25$.
As N increases, E_{out} decreases rapidly until it reaches the same level as E_{in}(around 100). During the process, E_{in} also increases slightly. At approximately $N = 400$, E_{out} and E_{in} become almost identical For larger N, E_{out} and E_{in} are very stable, and their value don't change much as N increases to larger values.
This result is also consistent with theory, from the two equations

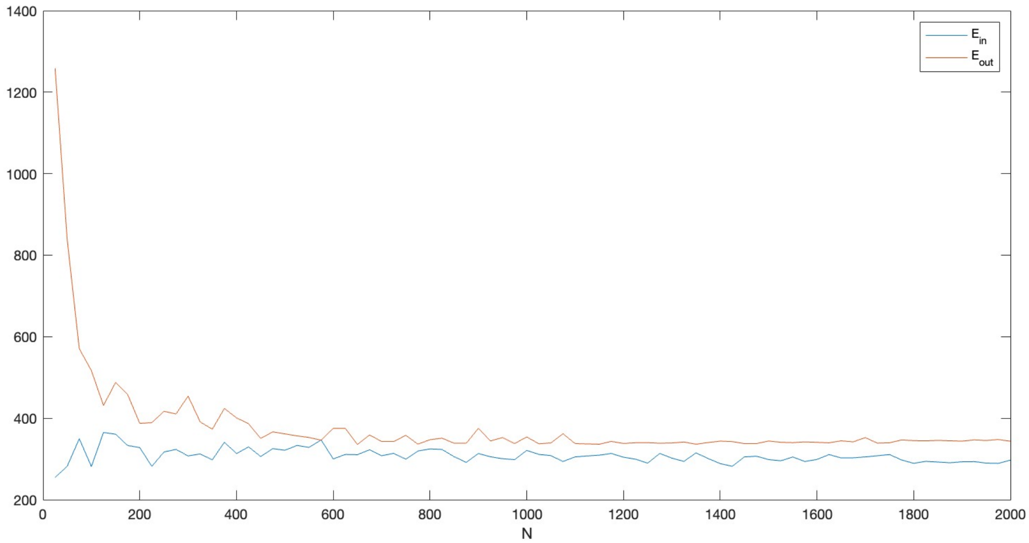
$$\overline{E_{out}} = \textbf{noise level} \cdot (1 + \frac{d+1}{N})$$
$$\overline{E_{in}} = \textbf{noise level} \cdot (1 - \frac{d+1}{N})$$

we see that as N becomes larger, both E_{out} and E_{in} converges to the noise level σ^2. This phenomenon is clearly reflected in the figure, as E_{out} and E_{in} approaches the same value for large N.


```matlab
    disp(line_count);
    line = fgetl(file);
end
%disp(data_vec(8192, :));



for N = 25:25:2000
    count = 1;
    iin = 0;
    oout = 0;
    while count <= 16
        in = 0;
        out = 0;
        indicies = zeros(1, N);
        y_vec = zeros(N, 1);
        mat = zeros(N, 13);
        for i = 1:N
            r = randi([1, 8192]);
            indicies(1, i) = r;
            mat(i, :) = data_vec(r, :);
            y_vec(i, 1) = y(1, r);
        end
        w = pinv(mat) * y_vec;
        %disp(w);
        for i = 1:8192
            err = (dot(w, data_vec(i, :)) - y(1, i)) ^ 2;
            if ismember(i, indicies(1, :))
                in = in + err;
            else
                out = out + err;
            end
        end
        in = in / N;
        out = out / (8192 - N);
        iin = iin + in;
        oout = oout + out;
        count = count + 1;
    end
    oout = oout / 16;
    iin = iin / 16;
    E_out(N/25, 1) = oout;
    E_in(N/25, 1) = iin;

    disp(N);
end
vec = 25:25:2000;
plot(vec, E_in);
hold on;
plot(vec, E_out);
legend('E_{in}', 'E_{out}');
xlabel("N");
hold off;
```

Code snapshot:

12



The figure above is mostly similar to that in problem 10 in their trend of growth(increase/decrease). But there are some differences.
Firstly we see that the value to which E_{out} and E_{in} converge is different. This may be due to the noise level term σ^2 in the equation. For this set the noise level for each dimension of $\mathbf{x}$ may be different, so training with different dimensions of $\mathbf{x}$ may lead to different σ^2 values. Secondly, the difference between E_{in} and E_{out} for small N and large N (stable value) is larger in problem 11. We can see that in problem 11 E_{out} decreased by more than 99%($\approx$ 7000 $\rightarrow\approx$ 100), and E_{in} increased by about 300%($\approx$ 27 $\rightarrow\approx$ 100), whereas in this problem E_{out} only decreased by approximately 70% ($\approx$ 1250 $\rightarrow\approx$ 400) and E_{in} only increased by about 100%($\approx$ 200 $\rightarrow\approx$ 400). This is due to the difference in d.
From the equations

$$\overline{E_{out}} = \textbf{noise} \text{ level} \cdot (1 + \frac{d+1}{N})$$
$$\overline{E_{in}} = \textbf{noise} \text{ level} \cdot (1 - \frac{d+1}{N})$$

we see that the difference between E_{in} and E_{out} at small N and their value at large N is $\frac{d+1}{N}$, hence for larger d and same N, E_{in} and E_{out} differs more with their asymptotic value. This is consistent with the cases for problem 11 and 12, as in problem 11 $d = 13$ and in problem 12 $d = 3$.

```
data_vec = zeros(8192, 3);
y = zeros(1, 8192);
E_in = zeros(80, 1);
E_out = zeros(80, 1);
file = fopen("./data.txt");

line = fgetl(file);
line_count = 0;
while ischar(line)
    line_string = string(line);
    parts = strsplit(line_string, ' ');
    %disp(parts);
    y(line_count + 1) = str2double(parts(1));
    x = zeros(3, 1);
    x(1) = 1;
    %disp(length(parts));
    for j = 2:3
        foo = strsplit(parts(j), ':');
        x(j) = str2double(foo(2));
    end
    data_vec(line_count + 1, :) = x;
    line_count = line_count + 1;
    disp(line_count);
    line = fgetl(file);
end
%disp(data_vec(8192, :));



for N = 25:25:2000
    count = 1;
    iin = 0;
    oout = 0;
    while count <= 16
```

Code snapshot:

In this problem we use $\{-1, +1\}$ to denote boolean values (outcome of binary classification)

13

This problem is done in collaboration with B12901035 鄭宇彥

Let $\mathcal{S}$ be the set that contains all 2^N dichotomies of some N data vectors $\mathbf{x} \in \mathcal{D}$, that is, $\mathcal{S}$ contains all 2^N distinct boolean vectors of length N.

Let $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2 \dots \mathcal{S}_N$ be subsets of S. The subset $\mathcal{S}_i$ contains all the dichotomies where exactly i data vectors are classified as $+1$.

From the definition above, we can see that $\mathcal{S}_i$ has $\binom{N}{i}$ elements, and any two of these subsets are mutually exclusive.

Now consider the set $\mathcal{S} \setminus (\mathcal{S}_k \cup \mathcal{S}_{k+1} \cup \dots \cup \mathcal{S}_N)$ for $k \in \{0, 1, 2, \dots, N\}$. For the dichotomies in this set, no k data vectors are scattered. This is true because for any k data vectors, this set does not contain any dichotomy where all these k data vectors are classified as $+1$, since any such dichotomy must be in one of $\mathcal{S}_k, \mathcal{S}_{k+1}, \dots, \mathcal{S}_N$.

The number of elements in $\mathcal{S} \setminus (\mathcal{S}_k \cup \mathcal{S}_{k+1} \cup \dots \cup \mathcal{S}_N)$ is

$$2^N - (\sum_{i=k}^{N} \binom{N}{i}) = \sum_{i=0}^{k-1} \binom{N}{i}$$

Hence we have

$$B(N, k) \geq \sum_{i=0}^{k-1} \binom{N}{i} \quad \forall k \in \{0, 1, 2, \dots, N\}$$

Lastly, for $k \in \{N+1, N+2, N+3 \dots\}$ obviously $B(N, k) = 2^N = \sum_{i=0}^{k-1} \binom{N}{i}$, as these N data vectors can be shattered.

Therefore, we have proven

$$B(N, k) \geq \sum_{i=0}^{k-1} \binom{N}{i}$$