ML homework 2: Question 5

I don't agree with the answer provided by chatGPT, since as the question stated, what we knew about the sequence is that, "the first N − 1 terms of an integer sequence is generated from some polynomial of degree N", but this statement does not guarantee that the sequence is formed as:

$$(P(1), P(2), \dots, P(N-1))$$

Therefore, if we let $P(i) = y_i$, we cannot plug in the points $(P(i), y_i)$ to derive the coefficients. For example, if we have the sequence as:

$$(P(2), P(3), \dots, P(N))$$

Then we match the integer sequence $(y_{1,}\, y_2, \dots, y_{N-1})$ to the two cases above, we will get different coefficients, thus resulting in different polynomials. Since the polynomial is not uniquely determined, we cannot guarantee what the N-th term will be.

Therefore, to predict the N-the term, we need more information telling how the sequence is defined (i.e We need to know the specific arguments of the polynomial that correspond to the sequence.)

6. sample size = 4N with kind $\begin{cases} |A| = N \\ |B| = N \\ |C| = N \\ |D| = N \end{cases}$ $N \to \infty$ $\therefore$ equal probability

Green in A: {1, 3, 5, 7, 9, 11, 13, 15}

B: {2, 4, 6, 8, 10, 12, 14, 16}

C: {9, 10, 11, 12, 13, 14, 15, 16}

D: {1, 2, 3, 4, 5, 6, 7, 8}

Let the 5 tickets be t_1, t_2, t_3, t_4, t_5

The cases that 5 tickets with number i are all green:

i		i	
1.	$t_1, \dots, t_5 \in A \cup D$	9.	$t_1, \dots, t_5 \in A \cup C$
2.	$t_1, \dots, t_5 \in B \cup D$	10.	$t_1, \dots, t_5 \in B \cup C$
3.	$t_1, \dots, t_5 \in A \cup D$	11.	$t_1, \dots, t_5 \in A \cup C$
4.	$t_1, \dots, t_5 \in B \cup D$	12.	$t_1, \dots, t_5 \in B \cup C$
5.	$t_1, \dots, t_5 \in A \cup D$	13.	$t_1, \dots, t_5 \in A \cup C$
6.	$t_1, \dots, t_5 \in B \cup D$	14.	$t_1, \dots, t_5 \in B \cup C$
7.	$t_1, \dots, t_5 \in A \cup D$	15.	$t_1, \dots, t_5 \in A \cup C$
8.	$t_1, \dots, t_5 \in B \cup D$	16.	$t_1, \dots, t_5 \in B \cup C$

<u>Case 1</u>:

$\because$ half of the tickets in the bag $\in A \cup D$

$\therefore P(t_1, \dots, t_5 \in A \cup D) = \left(\frac{1}{2}\right)^5 = \frac{1}{32}$

Case 2:

$\because$ half of the tickets in the bag $\in B \cup D$

$\therefore P(t_1, \ldots, t_5 \in B \cup D) = \left(\frac{1}{2}\right)^5 = \frac{1}{32}$

Similarly, for Case 3, 4, $\quad P(t_1, \ldots, t_5 \in A \cup C) = \frac{1}{32}$

$P(t_1, \ldots, t_5 \in B \cup C) = \frac{1}{32}$

But all of the tickets $\in A \cup \in B \cup \in C \cup \in D$ are counted repeatedly

$\therefore$ we need to substract $\left(\frac{1}{4}\right)^5 \times 4 = \frac{1}{256}$

Consider the case that if we have 3 kinds in the 5 tickets.

1. $t_1, \ldots, t_5 \in \{B, C, D\}$ $\rightarrow$ $\{1, 3, 5, 7\}$ not all green $\because \exists t_i \in \{B \cup C\}$
2. $t_1, \ldots, t_5 \in \{A, C, D\}$ $\quad$ $\{2, 4, 6, 8\}$ not all green $\because \exists t_i \in \{C\}$
3. $t_1, \ldots, t_5 \in \{A, B, D\}$ $\quad$ $\{9, 11, 13, 15\}$ not all green $\because \exists t_i \in \{B \cup D\}$
4. $t_1, \ldots, t_5 \in \{A, B, C\}$ $\quad$ $\{10, 12, 14, 16\}$ not all green $\because \exists t_i \in \{D\}$

It is similar for the cases $2 \sim 4$, and also the case $t_1, \ldots, t_5 \in \{A, B, C, D\}$

Thus, the probability of winning the prize B:

$$\frac{1}{32} \times 4 - \frac{1}{256} = \frac{32}{256} - \frac{1}{256} = \frac{31}{256} \quad \square$$

9. Q: The probability of 5 green 5s = ?

$\because$ green 5 only happens when the ticket is of kind $A \cup D$

$\therefore$ $P(5 \text{ green } 5s)$

$= P(t_1, \ldots, t_5 \in \{A \cup D\})$

$= \left(\frac{1}{2}\right)^5$

$= \frac{1}{32}$ $\square$

$$\begin{cases} \mu_m & 1 \text{ coin} \\ 1-\mu_m & \text{no coin} \end{cases}$$

j. M machines, each with probability $\begin{cases} \mu_m & 1 \text{ coin} \\ 1-\mu_m & \text{no coin} \end{cases}$, where μ_m: unknown

at time t, pull machine $m = ((t-1) \bmod M) + 1$

i.e. at time $t=1$, pull machine $m = ((1-1) \bmod M) + 1 = 1$

$t=2$, pull machine $m = ((2-1) \bmod M) + 1 = 2$

after $t > M$, machine m being pulled N_m times, collect c_m coins

$(N_m \geq 1 \because t > M)$

<u>claim:</u> for $M \geq 2$, $\forall m = 1, \ldots, M$, $\forall t = M+1, M+2, \ldots$

$$P\left[\mu_m \leq \frac{c_m}{N_m} + \sqrt{\frac{\ln t + \ln M - \frac{1}{2}\ln \delta}{N_m}}\right] \geq 1 - \delta$$

<u>pf:</u> Using union bound, the total probability that μ_m exceeds the bound for any machine $(m = 1, \ldots, M)$ at any time step $(t = M+1, M+2, \ldots)$ is

$$\sum_{m=1}^{M} \sum_{t=M+1}^{\infty} P\left[\mu_m > \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln \delta}{N_m}}\right] \leq \sum_{m=1}^{M} \sum_{t=M+1}^{\infty} \delta t^{-2}$$

$$\leq \sum_{m=1}^{M} \sum_{t=1}^{\infty} \delta t^{-2}$$

$$= \sum_{m=1}^{M} \delta \cdot \frac{\pi^2}{6}$$

$$= M \cdot \delta \cdot \frac{\pi^2}{6}$$

Therefore, the probability for the bound $\frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln \delta}{N_m}}$ is exceeded by any machine $m \in \{1, \ldots, M\}$ at any time step $\geq M+1$ will $\leq M \cdot \delta \cdot \frac{\pi^2}{6}$

$\therefore\ \forall m \in \{1, \dots, M\},\ \forall t \in \{M+1, M+2, \dots\}$ where $M \geq 2$

$$P\left[\mu_m > \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta}{N_m}}\right] \leq M \cdot \delta \cdot \frac{\pi^2}{6}$$

$$\Rightarrow \quad P\left[\mu_m \leq \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta}{N_m}}\right] \geq 1 - M \cdot \delta \cdot \frac{\pi^2}{6}$$

By replacing δ by $\frac{6\delta}{\pi^2 M}$, we get:

$$\Rightarrow \quad P\left[\mu_m \leq \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln\left(\frac{6\delta}{\pi^2 M}\right)}{N_m}}\right] \geq 1 - \delta$$

$$-\frac{1}{2}\left(\ln 6\delta - \ln \pi^2 M\right)$$

$$= -\frac{1}{2}\left(\ln 6 + \ln\delta - \ln\pi^2 - \ln M\right)$$

$$= \underline{-\frac{1}{2}\ln 6} - \frac{1}{2}\ln\delta + \underline{\frac{1}{2}\ln\pi^2} + \frac{1}{2}\ln M$$

omit constant term ($\because$ constant does not affect much as $t \uparrow$)

$$\doteq -\frac{1}{2}\ln\delta + \frac{1}{2}\ln M.$$

$$\Rightarrow P\left[\mu_m \leq \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta + \frac{1}{2}\ln M}{N_m}}\right] \geq 1 - \delta$$

$$\Downarrow \quad \because \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta + \frac{1}{2}\ln M}{N_m}} < \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta + \ln M}{N_m}}$$

$$\Rightarrow P\left[\mu_m \leq \frac{c_m}{N_m} + \sqrt{\frac{\ln t - \frac{1}{2}\ln\delta + \ln M}{N_m}}\right] \geq 1 - \delta$$

$\square$

9. boolean function $h: \{-1,1\}^k \to \{-1,1\}$

↳ if symmetric ⇒ value only depends on $\#\{1\}$

Let $H = \{h \mid h:$ symmetric boolean functions$\}$

Suppose for any $h \in H$, $S_h = \{\vec{x} \in \{0,1\}^k \mid h(\vec{x}) = 1\}$

(i.e. S contains all input vectors that will be mapped to 1 by hypothesis h)

Let $\vec{V_i} = [\underbrace{1\,1 \cdots 1}_{i \text{ bits}}\underbrace{0 \cdots 0}_{k-i \text{ bits}}]^T$

$S = \{\vec{v_i} \mid 0 \le i \le k\}$ (i.e. the set that contains vectors of all possible number of 1s)

$S' = \{\vec{V}_{p1}, \vec{V}_{p2}, \ldots, \vec{V}_{pd}\} \subseteq S$ (i.e. the set that contains vector $h(\vec{V}_{pi}) = 1$ $i = 1, \ldots, d$, which is S_h eliminating equivalent ones under permutation)

$\Rightarrow S \cap S_h = S'$

$\because |S| = k+1 \quad \therefore |S'| \le k+1$

This means that $\nexists$ a set more than $k+1$ points that can be shattered

To show that the set of $k+1$ point can be shattered, we can construct symmetric Boolean function:

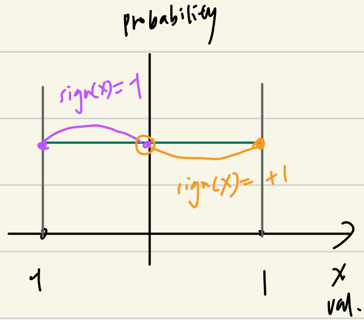
$$h(\vec{x}) = \begin{cases} y_0 & \text{if } \vec{x} \text{ equiv. to } \vec{v_0} \\ y_1 & \text{if } \vec{x} \text{ equiv. to } \vec{v_1} \\ \vdots & \vdots \\ y_k & \text{if } \vec{x} \text{ equiv. to } \vec{v_k} \end{cases}$$

where $y_i \in \{0,1\}$ for $i = 0, \ldots, k$

Therefore, $d_{vc}(H) = k+1$

∎.

10.



decision stump model: $h_{s,\theta}(x) = s \cdot \text{sign}(x-\theta)$

Let $\text{sign}(0) = -1$ for simplicity

Growth function $m_H(N) = 2N$, VC dimension $= 2$

For $x^t \in [-1,0]$ probability

$$y^t = \begin{cases} -1 & 1-p \\ +1 & p \quad \text{(flip)} \end{cases}$$

where $0 \le p < \frac{1}{2}$

For $x^t \in (0,1]$ probability

$$y^t = \begin{cases} +1 & 1-p \\ -1 & p \quad \text{(flip)} \end{cases}$$

where $0 \le p < \frac{1}{2}$

By definition,

$$E_{out}(h_{s,\theta}) = P(h_{s,\theta}(x) \ne y)$$

$$= P(h_{s,\theta}(x) = y \cap \text{flipped}) + P(h_{s,\theta}(x) \ne y \cap \text{not flipped})$$

$$= p \cdot \underline{P(h_{s,\theta}(x) = y)} + (1-p)\, P(h_{s,\theta}(x) \ne y)$$

$$= p \cdot \left[\underline{1 - P(h_{s,\theta}(x) \ne y)}\right] + (1-p)\, P(h_{s,\theta}(x) \ne y)$$

$$= p - p \cdot \boxed{P(h_{s,\theta}(x) \ne y)} + \boxed{P(h_{s,\theta}(x) \ne y)} - p \cdot \boxed{P(h_{s,\theta}(x) \ne y)}$$

$$= p + (1-2p) \cdot P(h_{s,\theta}(x) \ne y)$$

$P(h_{s,\theta}(x) \neq y)$

$= \boxed{P(h_{s,\theta}(x) \neq y \cap s=-1)} + \boxed{p(h_{s,\theta}(x) \neq y \cap s=+1)}$

$= ① \underline{p(h_{s,\theta}(x) \neq y \mid s=-1, x>\theta)} + ② \underline{P(h_{s,\theta}(x) \neq y \mid s=1, x \le \theta)}$

$③ \underline{p(h_{s,\theta}(x) \neq y \mid s=1, x>\theta)} + ④ \underline{p(h_{s,\theta}(x) \neq y \mid s=1, x \le \theta)}$

① $s=-1$, $x>\theta$

Prediction made by $h_{s,\theta}(x) = -1$

For $x>0$, $y=+1$ $\therefore h_{s,\theta}(x) \neq y$

For $x \le 0$, $y=-1$ $\therefore h_{s,\theta}(x) = y$

probability

$\because x>0,\ x>\theta$

$\therefore x \in (\theta, 1]$

$1-\frac{|\theta|}{2}$

$(s=-1)$

② $s=1$, $x \le \theta$

Prediction made by $h_{s,\theta}(x) = +1$

For $x>0$, $y=1$ $\therefore h_{s,\theta}(x) = y$

For $x \le 0$, $y=1$ $\therefore h_{s,\theta}(x) \neq y$

$\because x \le 0,\ x \le \theta$

$\therefore x \in [-1, \theta]$

③ $s=1$, $x>\theta$

Prediction made by $h_{s,\theta}(x) = +1$

For $x>0$, $y=1$ $\therefore h_{s,\theta}(x) = y$

For $x \le 0$, $y=-1$ $\therefore h_{s,\theta}(x) \neq y$

$\because x \le 0,\ x>\theta$

$\therefore x \in (\theta, 0]$

$\frac{|\theta|}{2}$

$(s=1)$

④ $s=1$, $x \le \theta$

Prediction made by $h_{s,\theta}(x) = -1$

For $x>0$, $y=1$ $\therefore h_{s,\theta}(x) \neq y$

For $x \le 0$, $y=-1$ $\therefore h_{s,\theta}(x) = y$

$\because x>0,\ x \le \theta$

$\therefore x \in (0, \theta]$

$\therefore$ Continue the process on the previous 2 page, we get:

$$E_{out}(h_{s,\theta}) = p + (1-2p)\cdot P(h_{s,\theta}(x) \neq y)$$

$$\Rightarrow \begin{cases} E_{out}(h_{-1,\theta}) = p + (1-2p)\cdot \underline{P(h_{-1,\theta}(x) \neq y)} = p + (1-2p)\cdot 1 - \frac{|\theta|}{2} \\ E_{out}(h_{1,\theta}) = p + (1-2p)\cdot \underline{P(h_{1,\theta}(x) \neq y)} = p + (1-2p)\cdot \frac{|\theta|}{2} \end{cases}$$

If $s=+1$, then $h_{s,\theta} = \text{sign}(x-\theta)$

$s=-1$, then $h_{s,\theta} = -\text{sign}(x-\theta)$

Thus we can modify $h_{s,\theta}$ to consider the 2 cons:

$$\boxed{h_{s,\theta} = \frac{1+s}{2}\,\text{sign}(x-\theta) - \frac{1-s}{2}\,\text{sign}(x-\theta)}$$

$\Rightarrow$ $h_{s,\theta} = \frac{1+1}{2}\,\text{sign}(x-\theta) - \frac{1-1}{2}\,\text{sign}(x-\theta) = \text{sign}(x-\theta)$ if $s=+1$

$h_{s,\theta} = \frac{1+(-1)}{2}\,\text{sign}(x-\theta) - \frac{1-(-1)}{2}\,\text{sign}(x-\theta) = -\text{sign}(x-\theta)$ if $s=-1$

Or we can write equivalently:

$$\boxed{h_{s,\theta} = \underbrace{\frac{1+s}{2}\,\text{sign}(x-\theta)}_{s=1} + \underbrace{\frac{1-s}{2}\left[-\text{sign}(x-\theta)\right]}_{s=-1}}$$

$$\therefore\ E_{out}(h_{s,\theta}) = \frac{1-s}{2}\left[p+(1-2p)\left(1-\frac{|\theta|}{2}\right)\right] + \frac{1+s}{2}\left[p+(1-2p)\cdot\frac{|\theta|}{2}\right]$$

$$= \frac{p}{2}\left[(1-s)+(1+s)\right] + \frac{1-2p}{2}\left[(1-s)\left(1-\frac{|\theta|}{2}\right)+(1+s)\frac{|\theta|}{2}\right]$$

$$= \frac{p}{2}\cdot 2 + \left(\frac{1}{2}-p\right)\left[1-s-\cancel{\frac{|\theta|}{2}}+\frac{s|\theta|}{2}+\cancel{\frac{|\theta|}{2}}+\frac{s|\theta|}{2}\right]$$

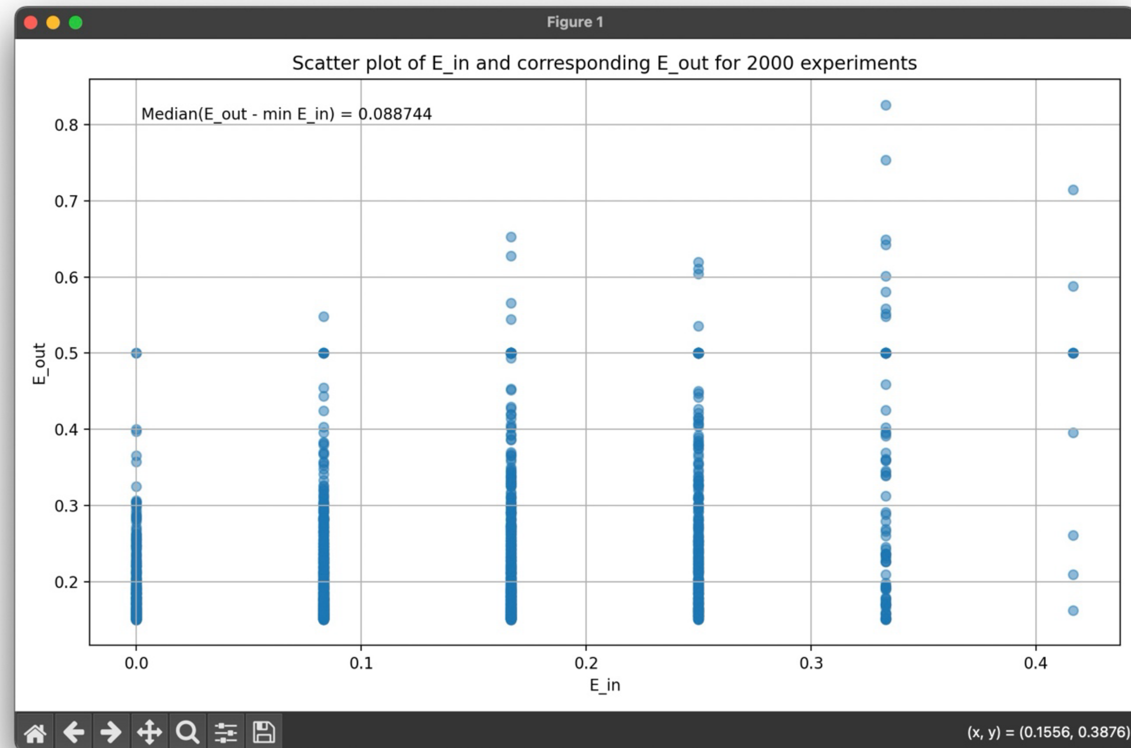
$$= p + \left(\frac{1}{2}-p\right)\left[1-s+s|\theta|\right]$$

$$= \cancel{p} + \frac{1}{2} - \frac{1}{2}s + \underline{\frac{1}{2}s|\theta|}\ \cancel{-p} + sp\ \underline{-sp|\theta|}$$

$$= \underbrace{\frac{1}{2}-s\left(\frac{1}{2}-p\right)}_{=u=\frac{1}{2}-v} + \underbrace{s\left(\frac{1}{2}-p\right)}_{=v}|\theta| \qquad \square$$

ML homework 2: question 11

The scatter plot of $(E_{out}(g),\ E_{in}(g))$ is as the figure below:



The median of the difference $E_{out}(g)$ - $E_{in}(g)$ is about 0.088744.

Code snapshot:

In the first part of my code, it's about the basic setups, like generating the x values and the y values with noise, combine them into tuples to present like data points, and sort the data points by the x value as required:

```python
for experiment_no in tqdm(range(2000)):
    x_arr = np.random.uniform(-1, 1, 12) # generate 12 x values, that are uniformly distributed in [-1, 1]
    y_arr = []
    for x_val in x_arr:
        if x_val > 0:
            y_arr.append(1)
        else:                          # assuming that sign(0) = -1
            y_arr.append(-1)

    # aim: add noise that flips the sign with 15% probability
    # explain: we generate noise that is -2y(15%) and 0(85%, which means without noise), so that when we add the noise to y,
    # explain: if y = 1, then y + noise = 1 + (-2) = -1
    # explain: if y = -1, then y + noise = (-1) + 2 = 1

    noise_arr = []
    np.random.seed(experiment_no)
    for y in y_arr:
        noise = np.random.choice([-2 * y, 0], p = [0.15, 0.85])
        noise_arr.append(int(noise))

    y_with_noise_arr = []
    for y, n in zip(y_arr, noise_arr):
        y_w_noise = y + n
        y_with_noise_arr.append(y_w_noise)

    data_points_list = list(zip(x_arr, y_with_noise_arr))
    sorted_data_points_list = sorted(data_points_list, key=lambda point: point[0])

    mean_x_list = []
    for i in range(0, len(x_arr) - 1):
        mean_x = (x_arr[i] + x_arr[i+1]) / 2
        mean_x_list.append(mean_x)

    # aim: generate a theta_list with the elements in it are (-1, mean_i), where mean_i is the mean of x_i and x_{i+1} (i starts from 1)
    theta_list = [(-1, mean_x) for mean_x in mean_x_list]
```

The next part is to calculate the in sample error of all possible combinations of s and theta, then find the minimum in sample error, and record its corresponding s and theta, if multiple pairs of s and theta can result in the minimum, then choose the optimal pair as the one with the smallest product:

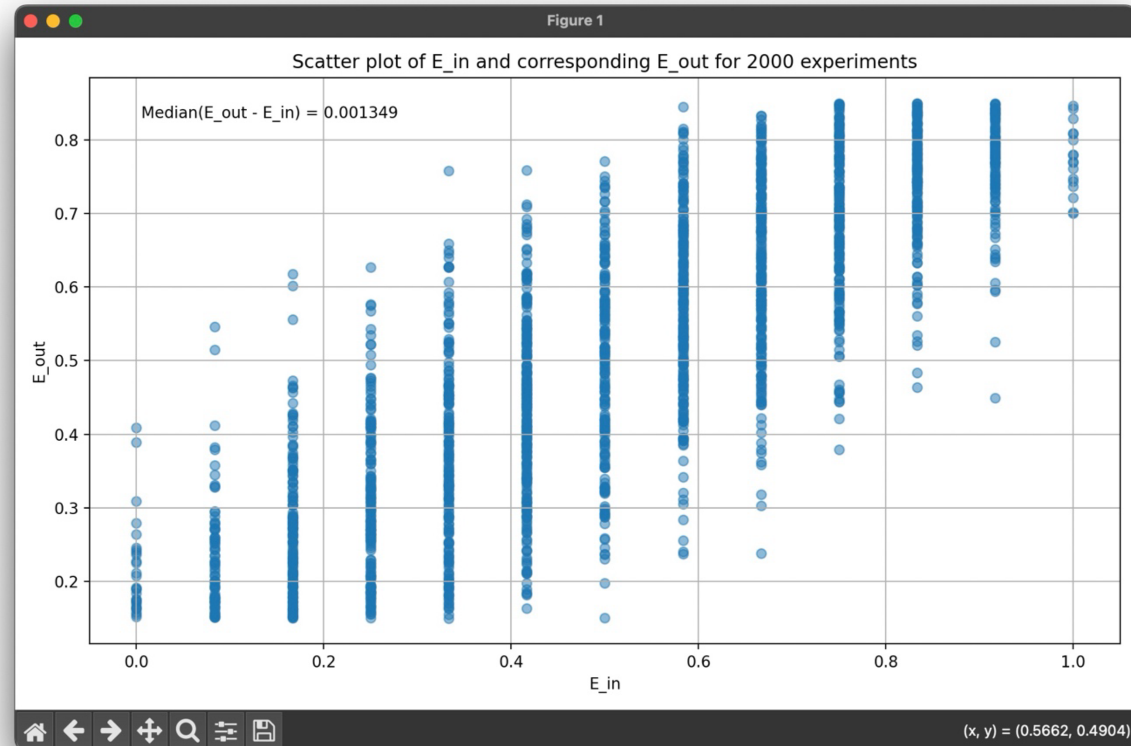
```python
    # aim: calculate E_in, record all the possible in sample error in E_in_list
    E_in_list = []
    s_theta_list = []

    for theta_tuple in theta_list:
        for theta in theta_tuple:
            for s in [-1, 1]:
                s_theta_list.append((s,theta))
                total_error = 0
                for x, y in sorted_data_points_list:
                    if x - theta > 0:
                        sign = 1
                    else:
                        sign = -1
                    prediction = s * sign
                    if prediction != y:
                        total_error += 1
                avg_total_error = total_error / 12
                E_in_list.append(avg_total_error)

    # aim: get g which corresponds to the minimum in sample error, and represent g as opt_s, opt_theta
    min_E_in = min(E_in_list)

    # subaim: save all pairs of (s, theta) in min_s_theta_list that will result in the minimum in sample error
    min_s_theta_list = []
    for index in range(len(E_in_list)):
        if E_in_list[index - 1] == min_E_in:
            min_s_theta_list.append(s_theta_list[index - 1])

    # subaim: save the s, theta we want(the pair that results in min(s * theta) if there's multiple pairs that generate minimum in sample error)
    if len(min_s_theta_list) != 1:
        opt_s, opt_theta = min(min_s_theta_list, key=lambda x: x[0] * x[1])
    else:
        opt_s, opt_theta = min_s_theta_list[0]
```

After this, calculating the corresponding out of sample error is quite simple, we just plug in the optimal s and theta values:

```
# aim: compute E_out(g)
v = opt_s * 0.35
u = 0.5 - v
E_out = u + v * abs(opt_theta)
```

The last part is recording the results of each experiment and plot the scatter plot. This part is quite simple so I won't put the code here, if other part of the code is needed, please let me know ☺

ML homework 2: question 12



Findings:

The median of $E_{out}(g_{RND}) - E_{in}(g_{RND})$ is about 0.001349, which is smaller than the resulting median of the previous question (0.088744).

I think the reason is because if we choose the optimal s and theta by finding the minimum in sample error, this will cause overfit to the data points, and while we only have 12 points in the dataset, this problem will become more severe, since it is hard to get a generalized result from merely 12 points, and if there's an extreme point (a point affected by noise), it will affect the hypotheses drastically. The bigger median of difference is telling that even though our model is performing well on the training data, it performs poorly on unseen data.

On the other hand, if we use randomly chosen s and theta, we introduce more randomness, probably having less probability to overfit, therefore making the result more generalized, so that the difference between in sample and out of sample error is smaller on average.

Snapshot of code:

Most of the code are the same as question 11, so I will provide the snapshot of the content that differs from the previous question. The first few 35 lines are the basic setup, which is the same as the previous question.

The modification is mostly presented in line 39, 40 in this snapshot, which shows that we randomly choose the value of s and theta, instead of calculating the minimum in sample error then decide the optimal s and theta:

```
36      data_points_list = list(zip(x_arr, y_with_noise_arr))
37      sorted_data_points_list = sorted(data_points_list, key=lambda point: point[0])
38
39      s = np.random.choice([-1, 1])
40      theta = np.random.uniform(-1, 1)
41
42      total_error = 0
43      for x, y in sorted_data_points_list:
44          if x - theta > 0:
45              sign = 1
46          else:
47              sign = -1
48          prediction = s * sign
49          if prediction != y:
50              total_error += 1
51      avg_total_error = total_error / 12
52
53      # aim: compute E_out(g)
54      v = s * 0.35
55      u = 0.5 - v
56      E_out = u + v * abs(theta)
57
58      # aim: store the result of in sample error and out of sample error, and their differences
59      E_in_all_experiments.append(avg_total_error)
60      E_out_all_experiments.append(E_out)
61      difference_list.append(E_out - avg_total_error)
62
63  median_difference = np.median(difference_list)
```