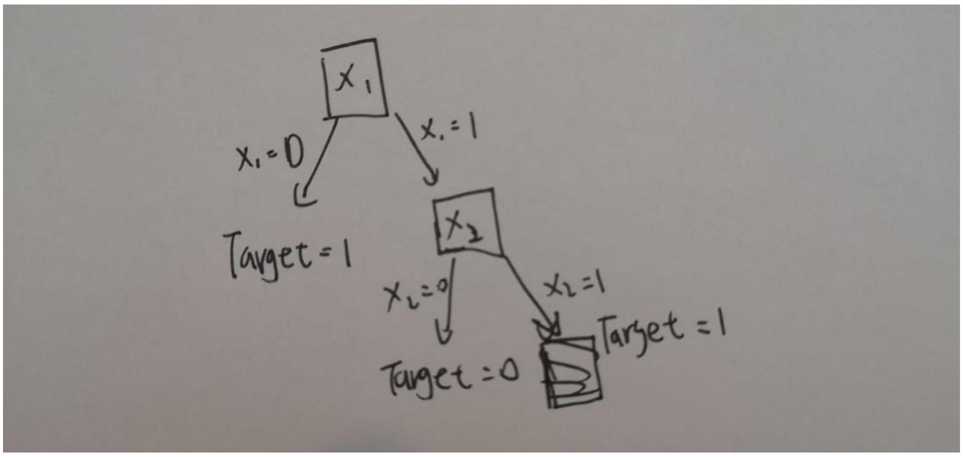
Z5305320
Dong AO
(a): Let’s say D represent this question data set. And X1, X2 and X3 represent the features.
(i):

The tree computed:



The training error is $\frac{1}{4}$ since I put dataset back to the tree, I find there is one piece of data is wrong and the total number of data is 4. Thus, it is $\frac{1}{4}$.

$$Entropy(D) = -\frac{1}{2} * \ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right) = 0.693147.$$

I use $Entropy(D_{X1})$ and etc. to denote the second term in the formular of $Gain(D, X_1)$ and etc.

$$Entropy(D_{X1}) = \frac{3}{4}\left(\frac{-2}{3} * \ln\left(\frac{2}{3}\right) - \frac{1}{3}\ln\left(\frac{1}{3}\right)\right) + \frac{(-1\ln(1))}{4} = 0.477386.$$

$$Entropy(D_{X2}) = \frac{1}{2}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) + \frac{1}{2}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) = 0.693147.$$

$$Entropy(D_{X3}) = \frac{1}{2}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) + \frac{1}{2}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) = 0.693147.$$

$$Gain(D, X_1) = 0.693147 - 0.477386 = 0.215762$$
$$Gain(D, X_2) = 0.693147 - 0.693147 = 0$$
$$Gain(D, X_3) = 0.693147 - 0.693147 = 0$$

Since X_1 has the highest gain, I choose X_1 as tree root. Then, once again:

$$Entropy(D_{X1}|X_1 = 1) = -\frac{1}{3}\ln\left(\frac{1}{3}\right) - \frac{2}{3}\ln\left(\frac{2}{3}\right) = 0.636514$$

$$Entropy(D_{X2}) = \frac{1}{2}Entropy(D_{X2}|X_1 = 1) + \frac{1}{2}Entropy(D_{X2}|X_1 = 2) =$$

$$= \frac{2}{3}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) + \frac{1}{3}(-1\ln(1)) = 0.462098$$

$$Entropy(D_{X3}) = \frac{1}{2}Entropy(D_{X3}|X_1 = 1) + \frac{1}{2}Entropy(D_{X3}|X_1 = 2) =$$

$$= \frac{1}{3}(-1\ln(1)) + \frac{2}{3}\left(-\frac{1}{2}\ln\left(\frac{1}{2}\right) - \frac{1}{2}\ln\left(\frac{1}{2}\right)\right) = 0.462098$$

$$Gain(D, X_2) = Entropy(D_{X1}|X_1 = 1) - \ Entropy(D_{X2}) = 0.636514 - 0.462098$$
$$= 0.174416$$

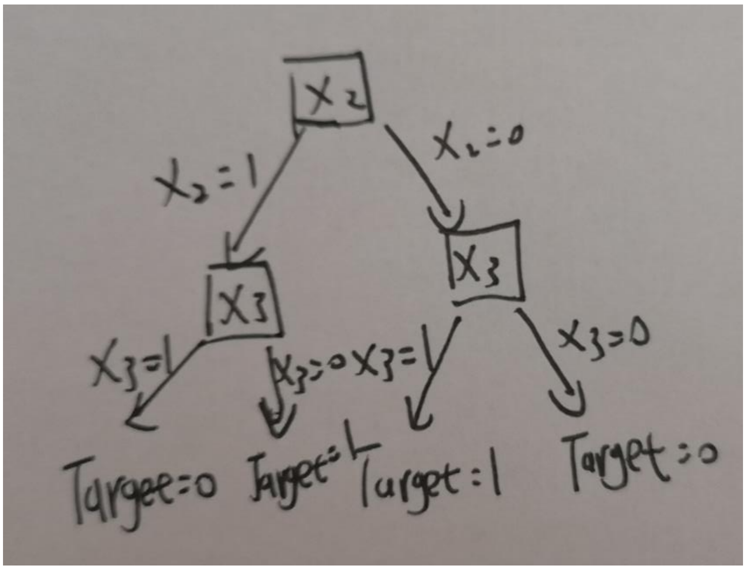
$$Gain(D, X_3) = Entropy(D_{X1}|X_1 = 1) - \ Entropy(D_{X3}) = 0.636514 - 0.462098$$
$$= 0.174416$$

Then, since they have the same gain, I can randomly choose one between X_2 and X_3.

(ii):

I try to put X_2 in the root and get the tree below, which has a 0 training error.

Decription: ID3 has the trend to avoid overfitting which appears in the tree below: it is doing better in training error but it is also overfitting over the dataset.



(b):

Dataset	Linearly Separable (Yes/No)
(i)	No
(ii)	No
(iii)	Yes
(iv)	Yes

Description:

1. I generate the correct data of X and the data of Y. Then I need to add a column of 1 to X because we need a bias term.

2. We use the standard perceptron: in each iteration, we calculate if there is a mistake, if yes, we try to update the weight with our produce of $X[i]$ and $y[i]$. If there is nothing to update, it means we have achieved. Otherwise, it means the dataset cannot converge.

```
import numpy as np
import pandas as pd # not really needed, only for preference
import matplotlib.pyplot as plt
def per(n):
    lists = []
    for i in range(1<<n):
        s=bin(i)[2:]
        s='0'*(n-len(s))+s
        lists.append(list(map(int,list(s))))
    return lists
def train_perceptron(X_data, y, size, max_iter=10000):

    eta=1
    np.random.seed(1)
    w = np.array([0 for i in range(size + 1)])
    nmb_iter = 0
    for _ in range(max_iter):
        X = X_data
        nmb_iter += 1
        yXw = (y * X) @ w.T
        mistake_idxs = np.where(yXw <= 0)[0]
        if mistake_idxs.size > 0:
            i = np.random.choice(mistake_idxs)
            w = w + y[i] * X[i]
#             print(f"Iteration {nmb_iter}: w = {w}")
        else: # no mistake made
            print(f"Converged after {nmb_iter} iterations")
            return w, nmb_iter
    print("Cannot converge")
    return w,nmb_iter
```

```
def get_y(target_x, size):
    strings = per(size)
    y = [-1 for i in range(len(strings))]
    for i in range(len(strings)):
        if strings[i] in target_x:
            y[i] = 1
    return y
```

```
def do_perceptron(size, target_x):
    target_y = get_y(target_x, size)
    target_y = np.array(np.mat(target_y).T)
    strings  = per(size)
    for string in strings:
        string.insert(0, 1)
    strings = np.array(strings)
    train_perceptron(strings, target_y, size, 10000)
```

```
target_x = [[0,1,1], [1,0,0], [1,1,0], [1,1,1]]
do_perceptron(3, target_x)
```

```
Cannot converge
```

```
target_x = [[0,1,0], [0,1,1], [1,0,0], [1,1,1]]
do_perceptron(3, target_x)
```

```
Cannot converge
```

```
target_x = [[0,1,0,0], [0,1,0,1], [0,1,1,0], [1,0,0,0], [1,1,0,0], [1,1,0,1], [1,1,1,0], [1,1,1,1]]
do_perceptron(4, target_x)
```

```
Converged after 18 iterations
```

```
target_x = [[1,0,0,0,0,0,0], [1,0,0,0,0,0,1], [1,0,0,0,1,0,1]]
do_perceptron(7, target_x)
```

```
Converged after 60 iterations
```

(c):

	ψ	$H_\psi(x,z)$	$x^{(t+1)}$
(i)	$\frac{1}{2}\lVert x\rVert_2^2$	$\frac{1}{2}\lVert x\rVert_2^2+\frac{1}{2}\lVert z\rVert_2^2-< z,x >$	$x^{(t)}-\alpha\nabla f(x^{(t)})$
(ii)	$\frac{1}{2}x^TQx$	$x^TQx+z^TQz-< Qz,x-z >$	$x^{(t)}-\alpha Q^{-1}\nabla f(x^{(t)})$
(iii)	$\sum_{i=1}^{p} x_i \ln \mathrm{x_i}$	$\sum_{i=1}^{p} x_i \ln \mathrm{x_i} - \sum_{i=1}^{p} z_i \ln \mathrm{z_i} - < \begin{pmatrix} lnz_1+1 \\ lnz_2+1 \\ \vdots \\ \vdots \\ lnz_p+1 \end{pmatrix}, x-z >$	$\begin{pmatrix} e^{-k_1}x_1^{(t)} \\ e^{-k_2}x_2^{(t)} \\ \vdots \\ \vdots \\ e^{-k_3}x_p^{(t)} \end{pmatrix}$ $(k_i=\left(\alpha\nabla f\left(x^{(t)}\right)\right)_i)$

(i):

$\frac{\partial \|x\|_2^2}{\partial x} = \frac{\partial \|x^T x\|_2}{\partial x} = 2x$, Thus, $\nabla\psi(z) = \frac{1}{2}\cdot 2z = z$

$$H\psi(x,z) = \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|z\|_2^2 - \langle z, x - z\rangle$$
$$= \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|z\|_2^2 + \|z\|_2^2 - \langle z, x\rangle$$
$$= \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|z\|_2^2 - \langle z, x\rangle$$

$$x^{(t+1)} = \arg\min_x \{\alpha\langle \nabla f(x^{(t)}), x\rangle + H\psi(x, x^{(t)})\}$$

Taking the derivative and let it be zero

Thus, $0 = \alpha\nabla f(x^{(t)}) + \frac{1}{2}\cdot 2x + (\frac{1}{2}\cdot 2x^{(t)})' - x^{(t)}$

$= \alpha\nabla f(x^{(t)}) + x + 0 - x^{(t)}$

$\cancel{x = \alpha\nabla f(x^{(t)}}$

which is $x = x^{(t)} - \alpha\nabla f(x^{(t)})$

So $x^{(t+1)} = x^{(t)} - \alpha\nabla f(x^{(t)})$

(ii):

No.

Date. / /

We have the formular $\frac{\partial x^TQx}{\partial x} = (Q+Q^T)x = 2Qx$

So, $\nabla\psi(x) = \frac{1}{2}\cdot 2Qx = Qx$

$H_\psi(x,z) = x^TQx - z^TQz - \langle Qz, x-z\rangle$

$\cancel{\nabla H_\psi(x,z) = Qx - Qz}$

$$\frac{\partial(H_\psi(x,z))}{\partial x} = Qx - 0 + 0 - Qz = Qx - Qz$$

Since $x^{(t+1)} = \arg\min_x \{\alpha\langle\nabla f(x^{(t)}), x\rangle + H_\psi(x, x^{(t)})\}$

We take the derivative of the RHS and let LHS ~~set t~~ set to zero.

Thus, $0 = \alpha\nabla f(x^{(t)}) + \frac{\partial(H_\psi(x,x^{(t)}))}{\partial x}$

$= \alpha\nabla f(x^{(t)}) + Qx - Qx^{(t)}$

~~that is~~ $\cancel{0 = \alpha\nabla f(x^{(t)}) + x - x^{(t)}}$

~~we have~~ $\cancel{x^{(t+1)} = x^{(t)} - \alpha\nabla f(x^{(t)})Q^{-1}}$

That is $0 = \alpha Q^{-1} f(x^{(t)}) + x - x^{(t)}$

we have $x^{(t+1)} = x^{(t)} - \alpha Q^{-1} f(x^{(t)})$

(iii):

No.

Date. / /

for a specific i, $\frac{d(x_i \log x_i)}{x_i} = \log x_i + x_i \cdot \frac{1}{x_i} = \log x_i + 1$

Thus $\nabla\psi(x) = \begin{pmatrix}\log x_1 + 1\\ \log x_2+1 \\ \vdots \\ \log x_p + 1\end{pmatrix}$ (since $\nabla\psi x = (\frac{\partial\psi}{\partial x_1}, \frac{\partial\psi}{\partial x_2}, \dots, \frac{\partial\psi}{\partial x_p})$)

We can have:

$$\nabla H_\psi(x,z) = \nabla\psi(x) - \frac{\partial\psi(z)}{\partial x} - \frac{\partial \langle\nabla\psi(z), x-z\rangle}{\partial x}$$

$$= \nabla\psi(x) - 0 + 0 - \nabla\psi(z) = \nabla\psi(x) - \nabla\psi(z)$$

Since $x^{(t+1)} = \underset{x}{\arg\min} \{\alpha\langle\nabla f(x^{(t)}), x\rangle + H_\psi(x, x^{(t)})\}$

We take the derivative of LHS and let RHS equal to zero

we have:

$$0 = \alpha\nabla f(x^{(t)}) + \nabla\psi(x) - \nabla\psi(x^{(t)})$$

$$\nabla\psi(\vec{x}^{(t+1)}) = \nabla\psi(\vec{x}^{(t)}) - \alpha\nabla f(\vec{x}^{(t)})$$

We use k_i to denote the kth term in Vector: $\alpha\nabla f(\vec{x}^{(t)})$

So $\log x_i^{(t+1)} + 1 = \log x_i^{(t)} + 1 - k_i \iff \log\frac{x_i^{(t+1)}}{x_i^{(t)}} = -k_i$

which is: $\frac{x_i^{(t+1)}}{x_i^{(t)}} = e^{-k_i}$, so $x_i^{(t+1)} = e^{-k_i} x_i^{(t)}$