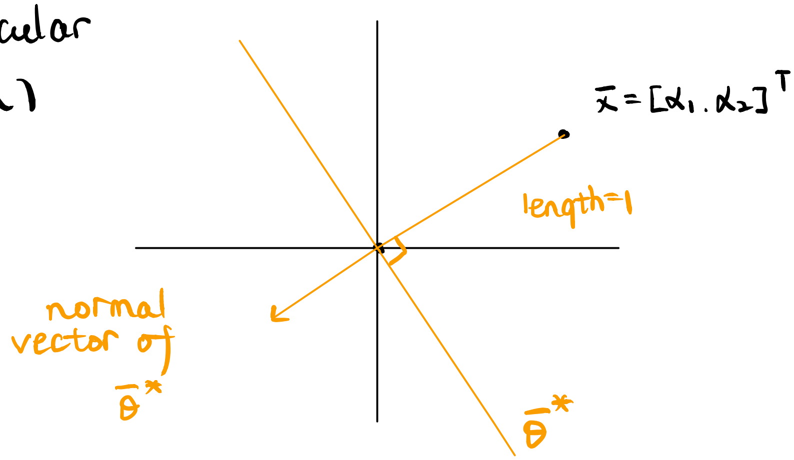
EECS 445 HW2

JunCheng An ; jamiean

1.1 a)

- Because we want to maximum hard margin and $b=0$. so I think $\bar{\theta}^*$ would lie on a line that pass through the origin and whose normal vector is exactly opposite to $\bar{x}$ because the label of $\bar{x}$ is $y=-1$, so $\bar{x}$ should be a support vector.
- Geometrically, as this

(use perpendicular to max margin)



and $\bar{\theta}^*$: $c\cdot[-\alpha_1, -\alpha_2]$ where $c>0$

$\because \bar{x}$ is a support vector with label $y=-1$

$\therefore y\cdot\bar{\theta}\cdot\bar{x} = (-1)\cdot c\cdot(-\alpha_1^2-\alpha_2^2) = 1$

$c\cdot(\alpha_1^2+\alpha_2^2) = 1$

$c = \dfrac{1}{\alpha_1^2+\alpha_2^2}$

$\therefore c = \dfrac{1}{\alpha_1^2+\alpha_2^2}$

$\therefore \bar{\theta}^* = \dfrac{-1}{\alpha_1^2+\alpha_2^2}\cdot[\alpha_1, \alpha_2]$

b) Say we have a $\vec{\theta}^* = [\theta_1 . \theta_2]$

- Because we want to max-margin on two points with different labels. so we must have the two points as support vector.

  - $\vec{x}^{(1)} = [-2, -1]^T, \; y^{(1)} = 1$

    $\Rightarrow y^{(1)} \cdot (\vec{\theta} \cdot \vec{x}^{(1)}) = 1 \cdot (-2\theta_1 - \theta_2) = -2\theta_1 - \theta_2 = 1$

  - $\vec{x}^{(2)} = [-1, -1]^T, \; y^{(2)} = -1$

    $\Rightarrow y^{(2)} \cdot (\vec{\theta} \cdot \vec{x}^{(2)}) = (-1) \cdot (-\theta_1 - \theta_2) = \theta_1 + \theta_2 = 1$

  $\Rightarrow$ because for support vectors $\vec{x}^{(i)}$: $y^{(i)} \cdot (\vec{\theta} \cdot \vec{x}^{(i)}) = 1$

  $\therefore \begin{cases} -2\theta_1 - \theta_2 = 1 \\ \theta_1 + \theta_2 = 1 \end{cases}$

  $\therefore \theta_1 = -2 \; ; \; \theta_2 = 3$

  $\therefore \vec{\theta}^* = [-2, 3]$

  $\gamma^{(1)}(\vec{\theta}) = \dfrac{-2 \times (-2) + 3 \times -1}{(2^2 + 3^2)^{0.5}} = \dfrac{\sqrt{13}}{13} = 0.2774$ for support vector

c) - say we have now $\vec{\theta}^* = [\theta_1 . \theta_2]$ and $b^* = b$

- same as b). see the two points as support vectors

  - $\vec{x}^{(1)} = [-2, -1]^T, \; y^{(1)} = 1$

    $\Rightarrow y^{(1)} \cdot (\vec{\theta} \cdot \vec{x}^{(1)} + b) = (-2\theta_1 - \theta_2 + b) = 1$

  - $\vec{x}^{(2)} = [-1, -1]^T, \; y^{(2)} = -1$

    $\Rightarrow y^{(2)} \cdot (\vec{\theta} \cdot \vec{x}^{(2)} + b) = -(-\theta_1 - \theta_2 + b)$

    $= \theta_1 + \theta_2 - b = 1$

- then we use b to solve θ_1 and θ_2

  $\Rightarrow -\theta_1 = 2 \Rightarrow \theta_1 = -2$

  $\Rightarrow 4 - \theta_2 + b = 1 \Rightarrow \theta_2 = b + 3$

$\therefore\ \vec{\theta}^* = [-2, b+3]$

$\therefore$ to minimize $\|\vec{\theta}\|^2 = 4 + (b+3)^2$

we take $b^* = b = -3$ s.t. $\|\vec{\theta}\|^2 = 4$

$\therefore\ \vec{\theta}^* = [-2, 0],\ b^* = -3$

$$\gamma(\vec{\theta}^*, b^*) = \frac{|-2\times -2 + 0 - 3|}{2} = \frac{1}{2}$$

1.2 a) $\cdot\ \sum_{i=1}^{2} \alpha_i y^{(i)} = \alpha_1 y^{(1)} + \alpha_2 y^{(2)} = 0$

$\therefore\ \alpha_1 \cdot 1 + \alpha_2 \cdot (-1) = 0 \Rightarrow \alpha_1 = \alpha_2$

$\therefore$ our maximizing problem becomes

$$\max\ \alpha_1 + \alpha_2 - \frac{1}{2}\left(\alpha_1\alpha_1 y^{(1)} y^{(1)} \cdot K(\vec{x}^{(1)}, \vec{x}^{(1)}) + \alpha_1\alpha_2 y^{(1)} y^{(2)} K(\vec{x}^{(1)}, \vec{x}^{(2)})\right.$$
$$\left.\alpha_2\alpha_1 y^{(2)} y^{(1)} \cdot K(\vec{x}^{(2)}, \vec{x}^{(1)}) + \alpha_2\alpha_2 y^{(2)} y^{(2)} K(\vec{x}^{(2)}, \vec{x}^{(2)})\right)$$

$\Rightarrow$ sub α_2 with α_1

$$2\alpha_1 - \frac{1}{2}\left(\alpha_1^2 \cdot K(\vec{x}^{(1)}, \vec{x}^{(1)}) - \alpha_1^2 \cdot K(\vec{x}^{(1)}, \vec{x}^{(2)})\right.$$
$$\left.-\alpha_1^2 \cdot K(\vec{x}^{(1)}, \vec{x}^{(2)}) + \alpha_1^2 \cdot K(\vec{x}^{(2)}, \vec{x}^{(2)})\right)$$

$\Rightarrow\ K(\vec{x}^{(1)}, \vec{x}^{(1)}) = \vec{x}^{(1)} \cdot \vec{x}^{(1)'} = \|\vec{x}^{(1)}\|^2 = 1 = \|\vec{x}^{(2)}\|^2 = \vec{x}^{(2)} \cdot \vec{x}^{(2)'} = K(\vec{x}^{(2)}, \vec{x}^{(2)})$

$$2\alpha_1 - \frac{1}{2}\left(2\alpha_1^2 - 2\alpha_1^2 \cdot K(\vec{x}^{(1)}, \vec{x}^{(2)})\right)$$

take derivative w.r.t α_1

$$2 - 2\alpha_1 + 2\alpha_1 \cdot K(\vec{x}^{(1)}, \vec{x}^{(2)}) = 0$$

$$\alpha_1 = \frac{1}{1 - K(\vec{x}^{(1)}, \vec{x}^{(2)})}$$

$$\therefore\ \alpha_1 = \alpha_2 = \frac{1}{1 - K(\vec{x}^{(1)}, \vec{x}^{(2)})} = \frac{1}{1 - \vec{x}^{(1)} \cdot \vec{x}^{(2)}}$$

b) $\bar{\theta}^* = \sum_{i=1}^{2} \alpha_i y^{(i)} \bar{x}^{(i)} = \frac{1}{1-\bar{x}^{(1)}\cdot\bar{x}^{(2)}} \cdot (\bar{x}^{(1)} - \bar{x}^{(2)})$

$$= \frac{\bar{x}^{(1)} - \bar{x}^{(2)}}{1-\bar{x}^{(1)}\cdot\bar{x}^{(2)}}$$

c) · two support vector should have same $y^{(i)}\cdot(\bar{\theta}\cdot\bar{x}+b) = 1$

$$\therefore \frac{\bar{x}^{(1)} - \bar{x}^{(2)}}{1-\bar{x}^{(1)}\cdot\bar{x}^{(2)}} \cdot \bar{x}^{(1)} + b = -\frac{\bar{x}^{(1)} - \bar{x}^{(2)}}{1-\bar{x}^{(1)}\bar{x}^{(2)}} \cdot \bar{x}^{(2)} - b$$

$$2b = -\frac{\bar{x}^{(1)2} - \bar{x}^{(1)}\bar{x}^{(2)}}{1-\bar{x}^{(1)}\bar{x}^{(2)}} - \frac{\bar{x}^{(1)}\bar{x}^{(2)} - \bar{x}^{(2)2}}{1-\bar{x}^{(1)}\bar{x}^{(2)}}$$

$$= -1 + 1 = 0$$

$$\therefore b^* = 0$$

d) $r(\bar{\theta}, b^*) = \frac{|\bar{\theta}\cdot\bar{x}+b|}{\|\bar{\theta}\|} = \frac{\frac{\bar{x}^{(1)2} - \bar{x}^{(1)}\bar{x}^{(2)}}{1-x^{(1)}\bar{x}^{(2)}} + 0}{\left\|\frac{\bar{x}^{(1)} - \bar{x}^{(2)}}{1-\bar{x}^{(1)}\bar{x}^{(2)}}\right\|} = \frac{1-\bar{x}^{(1)}\bar{x}^{(2)}}{\|\bar{x}^{(1)} - \bar{x}^{(2)}\|}$

$$= \frac{1-\bar{x}^{(1)}\bar{x}^{(2)}}{[(\bar{x}^{(1)} - \bar{x}^{(2)})(\bar{x}_1 - \bar{x}^{(2)})]^{0.5}} = \frac{1-\bar{x}^{(1)}\bar{x}^{(2)}}{(1+1-2\bar{x}^{(2)}\bar{x}^{(1)})^{0.5}}$$

$$= \frac{1-\bar{x}^{(1)}\bar{x}^{(2)}}{(2-2\bar{x}^{(2)}\bar{x}^{(1)})^{0.5}} = \frac{\sqrt{2}}{2}\cdot(1-\bar{x}^{(1)}\bar{x}^{(2)})^{0.5}$$

$$= \frac{(2-2\bar{x}^{(1)}\bar{x}^{(2)})^{0.5}}{2} = \frac{\|\bar{x}^{(1)} - \bar{x}^{(2)}\|}{2}$$

2\. a)

- for support vectors, we have $\alpha_i > 0$ because of "completeness slackness constraints"
- therefore, we find all points $\bar{x}^{(i)}$ that $\alpha_i > 0$

$\Rightarrow$ $\bar{x}^{(3)}, \bar{x}^{(6)}, \bar{x}^{(7)}, \bar{x}^{(8)}, \bar{x}^{(9)}, \bar{x}^{(11)}, \bar{x}^{(13)}, \bar{x}^{(14)}, \bar{x}^{(15)}, \bar{x}^{(17)}$ are support vectors.

b)

$$\bar{\theta}^* = \sum_{i=1}^{20} \alpha_i y^{(i)} \bar{x}^{(i)} = 0.05 \cdot (\bar{x}^{(11)} + \bar{x}^{(13)} + \bar{x}^{(14)} + \bar{x}^{(15)} - \bar{x}^{(3)} - \bar{x}^{(6)} - \bar{x}^{(8)} - \bar{x}^{(9)}) + 0.0159(\bar{x}^{(17)} - \bar{x}^{(7)})$$

$$= 0.05 \cdot [7.99, -0.63] + 0.0159 \cdot [3.59, 5.85]$$

$$= [0.456581, 0.061515]$$

- to find b^*, we need to find two closest support vectors with different labels

$\Rightarrow$ that is to find the positive point with highest $\bar{\theta} \cdot \bar{x}^{(i)}$ and negative point with lowest $\bar{\theta} \cdot \bar{x}^{(i)}$

- $\bar{x}^{(3)} \cdot \bar{\theta} = -0.966149$
- $\bar{x}^{(6)} \cdot \bar{\theta} = -0.445618$
- $\bar{x}^{(7)} \cdot \bar{\theta} = -1.112142$
- $\bar{x}^{(8)} \cdot \bar{\theta} = -0.72726$
- $\bar{x}^{(9)} \cdot \bar{\theta} = -0.99062$
- $\bar{x}^{(11)} \cdot \bar{\theta} = 0.41874$
- $\bar{x}^{(13)} \cdot \bar{\theta} = -0.578806$
- $\bar{x}^{(14)} \cdot \bar{\theta} = 0.80798$
- $\bar{x}^{(15)} \cdot \bar{\theta} = -0.16824$
- $\bar{x}^{(17)} \cdot \bar{\theta} = 0.88685$

$\therefore$ minneg $= -1.112142$ at $\bar{x}^{(7)}$

maxpos $= 0.88685$ at $\bar{x}^{(17)}$

$\therefore -(-1.112142 + b) = 0.88685 + b$

$\therefore b^* = 0.112646$

- equation to compute $b^* = \dfrac{MIN_{i:y^{(i)}=-1} \bar{x}^{(i)} \cdot \bar{\theta} + MAX_{i:y^{(i)}=1} \bar{x}^{(i)} \cdot \bar{\theta}}{2}$

3.1 (a) $k(\bar{x},\bar{z}) = \phi(\bar{x})\cdot\phi(\bar{z})$

$$= x_1^3 z_1^3 + 2x_1^2 x_2 z_1^2 z_2 + x_1 x_2^2 z_1 z_2^2 + x_1^2 x_2 z_1^2 z_2 + 2x_1 x_2^2 z_1 z_2^2$$
$$+ x_2^3 z_2^3 + 6x_1^2 z_1^2 + 12 x_1 x_2 z_1 z_2 + 6 x_2^2 z_2^2 + 9 x_1 z_1 + 9 x_2 z_2$$
$$= (x_1 z_1 + x_2 z_2)^3 + 6(x_1 z_1 + x_2 z_2)^2 + 9(x_1 z_1 + x_2 z_2)$$
$$= (x_1 z_1 + x_2 z_2)\cdot[(x_1 z_1 + x_2 z_2)^2 + 6(x_1 z_1 + x_2 z_2) + 9]$$
$$= (x_1 z_1 + x_2 z_2)\cdot(x_1 z_1 + x_2 z_2 + 3)^2$$
$$= k_1(\bar{x},\bar{z})\cdot k_2(\bar{x},\bar{z})$$

$\therefore k_2(\bar{x},\bar{z}) = (x_1 z_1 + x_2 z_2 + 3)^2$

3.2 (a) start from $\bar{\theta}^{(0)}$

$\rightarrow \bar{\theta}^{(0)} = \bar{0}$

$\rightarrow \bar{\theta}^{(1)} = \bar{\theta}^{(0)} + \eta\cdot\sigma(-y^{(1)}\bar{\theta}^{(0)}\cdot\bar{x}^{(1)})\cdot y^{(1)}\cdot\bar{x}^{(1)}$

$\rightarrow \bar{\theta}^{(2)} = \bar{\theta}^{(1)} + \eta\,\sigma(-y^{(2)}\bar{\theta}^{(1)}\cdot\bar{x}^{(2)})\cdot y^{(2)}\cdot\bar{x}^{(2)}$

$\rightarrow \bar{\theta}^{(3)} = \bar{\theta}^{(2)} + \eta\,\sigma(-y^{(3)}\cdot\bar{\theta}^{(2)}\cdot\bar{x}^{(3)})\cdot y^{(3)}\cdot\bar{x}^{(3)}$

$\rightarrow \bar{\theta}^{(4)} = \bar{\theta}^{(3)} + \eta\,\sigma(-y^{(1)}\bar{\theta}^{(3)}\bar{x}^{(1)})\,y^{(1)}\bar{x}^{(1)}$

$\rightarrow \bar{\theta}^{(5)} = \bar{\theta}^{(4)} + \eta\,\sigma(-y^{(2)}\bar{\theta}^{(4)}\bar{x}^{(2)})\,y^{(2)}\bar{x}^{(2)}$

$\rightarrow \bar{\theta}^{(6)} = \bar{\theta}^{(5)} + \eta\,\sigma(-y^{(3)}\cdot\bar{\theta}^{(5)}\cdot\bar{x}^{(3)})\,y^{(3)}\bar{x}^{(3)}$

$\rightarrow \bar{\theta}^{(7)} = \bar{\theta}^{(6)} + \eta\,\sigma(-y^{(1)}\bar{\theta}^{(6)}\bar{x}^{(1)})\,y^{(1)}\bar{x}^{(1)}$

$\rightarrow \bar{\theta}^{(8)} = \bar{\theta}^{(7)} + \eta\,\sigma(-y^{(2)}\bar{\theta}^{(7)}\bar{x}^{(2)})\,y^{(2)}\bar{x}^{(2)}$

$\rightarrow \bar{\theta}^{(9)} = \bar{\theta}^{(8)} + \eta\,\sigma(-y^{(3)}\bar{\theta}^{(8)}\bar{x}^{(3)})\,y^{(3)}\bar{x}^{(3)}$

$\therefore \bar{\theta} = \bar{\theta}^{(9)}$

$$= \eta\cdot\sigma(-y^{(1)}\bar{\theta}^{(0)}\cdot\bar{x}^{(1)})\,y^{(1)}\bar{x}^{(1)} \cdots + \eta\,\sigma(-y^{(3)}\bar{\theta}^{(8)}\bar{x}^{(3)})\,y^{(3)}\bar{x}^{(3)}$$
$$= \sum_{i=0}^{2} \eta\cdot\sigma(-y^{(1)}\bar{\theta}^{(3i)}\bar{x}^{(1)})\cdot y^{(1)}\cdot\bar{x}^{(1)}$$
$$+ \sum_{i=0}^{2} \eta\,\sigma(-y^{(2)}\bar{\theta}^{(3i+1)}\bar{x}^{(2)})\,y^{(2)}\bar{x}^{(2)}$$
$$+ \sum_{i=0}^{2} \eta\,\sigma(-y^{(3)}\bar{\theta}^{(3i+2)}\bar{x}^{(3)})\,y^{(3)}\cdot\bar{x}^{(3)}$$

$\therefore$ we can extract that

$$\alpha_1 = \Sigma_{i=0}^{2} \eta \sigma(-y^{(1)} \cdot \bar{\theta}^{(3i)} \cdot \bar{x}^{(1)}) \cdot y^{(1)}$$
$$\alpha_2 = \Sigma_{i=0}^{2} \eta \sigma(-y^{(2)} \bar{\theta}^{(3i+1)} \bar{x}^{(2)}) y^{(2)}$$
$$\alpha_3 = \Sigma_{i=0}^{2} \eta \sigma(-y^{(3)} \bar{\theta}^{(3i+2)} \bar{x}^{(3)}) y^{(3)}$$

(b)

(i) · Based on our result from (a),
we can find that each $\theta^{(n)}$ is related to only one α

· That is, we only need to update one element $\bar{\alpha}^{(i)}$ in $\bar{\alpha}$ in each update

Pseudocode:
$$\underbrace{\bar{\alpha}_i^{(k+1)}}_{i^{th}\text{ element}} = \bar{\alpha}_i^{(k)} + \eta \sigma[-y^{(i)} (\Sigma_{j=1}^{N} \bar{\alpha}_j^{(k)} \cdot \bar{x}^{(j)}) \bar{x}^{(i)}] \cdot y^{(i)}$$

because every $\bar{\theta}^{(k)} = \Sigma_{j=1}^{N} \bar{\alpha}^{(k)}(j) \cdot \bar{x}^{(j)}$

(ii) · $h(\bar{x}) = \sigma(\bar{\theta} \cdot \bar{x}) = \dfrac{1}{1 + \exp(-\bar{\theta} \cdot \bar{x})}$

$$= \frac{1}{1 + \exp(-\Sigma_{i=1}^{n} \alpha_i \cdot \bar{x}^{(i)} \cdot \bar{x})}$$

(c) ① $\bar{\alpha}_i^{(k+1)} = \bar{\alpha}_i^{k} + \eta \sigma(-y^{(i)} \cdot \Sigma_{j=1}^{N} \bar{\alpha}_j^{(k)} \cdot K(\bar{x}^{(j)}, \bar{x}^{(i)})] \cdot y^{(i)}$

② $h(\bar{x}) = \dfrac{1}{1 + \exp(-\Sigma_{i=1}^{n} \alpha_i \cdot K(\bar{x}^{(i)}, \bar{x}))}$

4. a) $\cdot\ H(Y|P) = P(P=\text{Delayed}) \cdot H(Y|P=\text{Delayed})$

$+ P(P=\text{On time}) \cdot H(Y|P=\text{On time})$

$- \ H(Y|P=\text{Delayed}) = -0.75 \cdot \log_2 0.75 - 0.25 \log_2 0.25$

$= 0.811278$

$- \ H(Y|P=\text{on time}) = -0.4 \log_2 0.4 - 0.6 \log_2 0.6$

$= 0.97095$

$\therefore H(Y|P) = \frac{4}{9} \times 0.811278 + \frac{5}{9} \times 0.97095 = 0.89999$

$\cdot\ H(Y|B) = P(B=\text{Decent}) \cdot H(Y|B=\text{Decent})$

$+ P(B=\text{slow}) \cdot H(Y|B=\text{slow})$

$+ P(B=\text{fast})\, H(Y|B=\text{fast})$

$- H(Y|B=\text{Decent}) = -0.75 \log_2 0.75 - 0.25 \log_2 0.25$

$= 0.81128$

$- H(Y|B=\text{slow}) = 0$

$- H(Y|B=\text{fast}) = 0$

$\therefore H(Y|B) = \frac{4}{9} \times 0.81128 = 0.36057$

$\cdot\ H(Y|Q) = P(Q=\text{Good}) \cdot H(Y|Q=\text{Good})$

$+ P(Q=\text{Great}) \cdot H(Y|Q=\text{Great})$

$+ P(Q=\text{Poor}) \cdot H(Y|Q=\text{Poor})$

$- H(Y|Q=\text{Good}) = -0.75 \log_2 0.75 - 0.25 \log_2 0.25$

$= 0.81128$

$- H(Y|Q=\text{Great}) = -0.5 \log_2 0.5 \times 2 = 1$

$- H(Y|Q=\text{Poor}) = -0.33 \log_2 0.33 - 0.67 \log_2 0.67$

$= 0.91829$

$\therefore H(Y|Q) = \frac{4}{9} \times 0.81128 + \frac{2}{9} \times 1 + \frac{3}{9} \times 0.91829$

$= 0.888887$

$\therefore H(Y|\text{Punctuality}) = 0.89999$

$H(Y|\text{Efficiency}) = 0.36057$

$H(Y|\text{Quality of service}) = 0.888887$

$\therefore$ Punctuality yield the highest entrophy

b) $H(Y) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.991076$

$\therefore IG(P, Y) = H(Y) - H(Y|P)$

$= 0.991076 - 0.89999 = 0.09109$

$IG(E, Y) = H(Y) - H(Y|E)$

$= 0.991076 - 0.36057 = 0.63051$

$IG(Q, Y) = H(Y) - H(Y|Q)$

$= 0.991076 - 0.888887 = 0.102188$

$\therefore IG(\text{Punctuality}|Y) = 0.09109$

$IG(\text{Efficiency}|Y) = 0.63051$

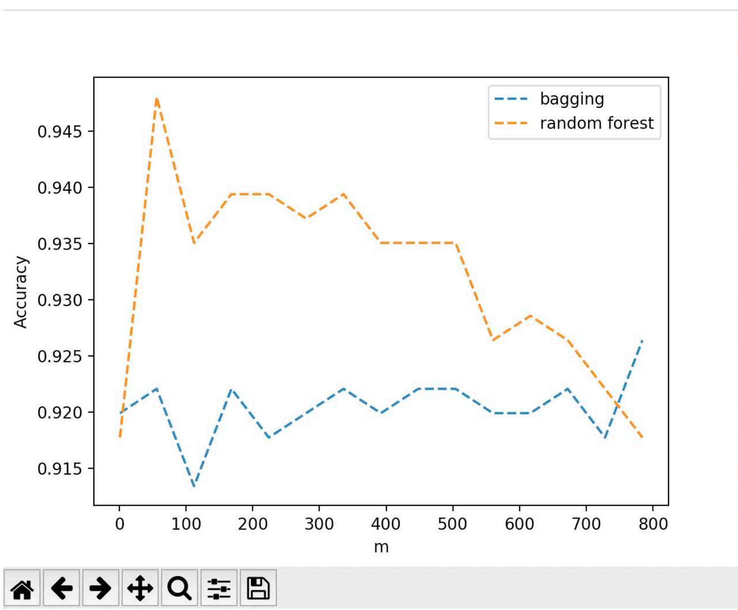
$IG(\text{Quality of service}|Y) = 0.102188$

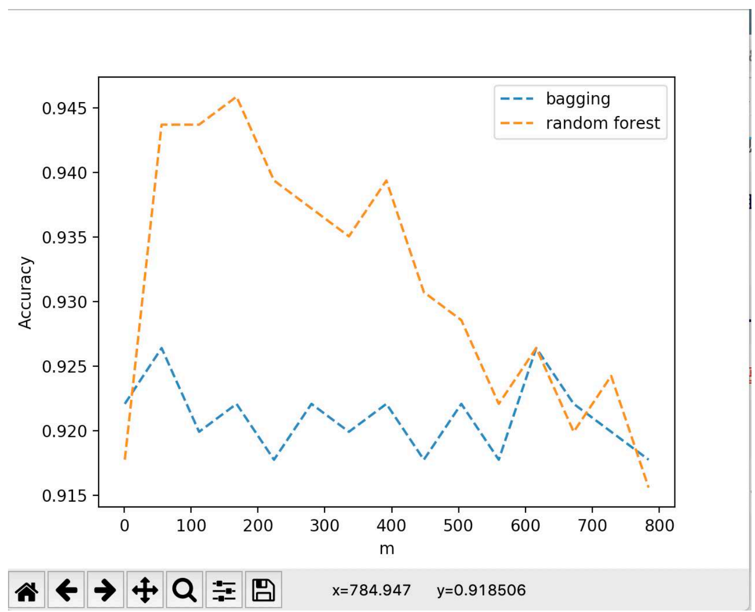
$\therefore$ Efficiency has the highest information gain

5\. · when m is relatively small, random_forest would perform much better than simply bagging. However, as m grows larger, the performance of random forest would goes down and bagging would almost remain the same. So for large m, the two performances are almost the same.

· It make sense to me because m does not change anything to simply bagging and it remains same; while m gets larger, random_forest gets more similar to bagging so their performance get closer.

· I test this twice because of the randomness and get a similar trend as following





6. a)

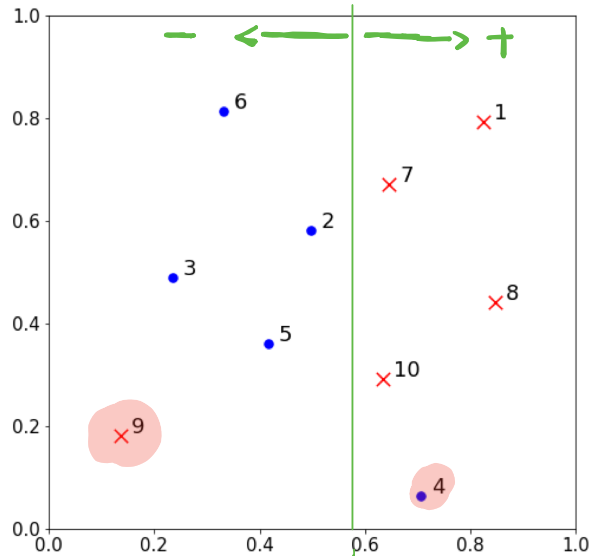


Figure 2: ●-negative points and ×-positive points.

- as shown, a vertical line between 2 and 7
- left of this line is "−" and right side is "+"

b)
- The point with red circle (9 & 4) would have largest weight!
- Because they are misclassified and given higher weight

c) $n = 10 \Rightarrow W_0(i) = 0.1$ for all i

- $\varepsilon_1 = 0.1 \times 2 = 0.2$
- $\alpha_0 = \frac{1}{2} \cdot \log\left(\frac{0.8}{0.2}\right) = \frac{1}{2}\ln 4$
- for correctly classified points

$$W_1(i) = C_1 \cdot 0.1 \cdot e^{-1 \times \frac{1}{2}\ln 4} = C_1 \cdot 0.1 \cdot 0.5 = 0.05 C_1$$

for misclassified points

$$W_1(i) = C_1 \cdot 0.1 \cdot e^{\frac{1}{2}\ln 4} = 0.2 C_1$$

$\therefore\ 0.05c_1 \times 8 + 0.2c_1 \times 2 = 0.4c_1 + 0.4c_1 = 1$

$\therefore\ c_1 = 1.25$

$\Rightarrow\ w_1(i) = 0.25 \quad i = 4, 9$

$w_1(i) = 0.0625 \quad o.w.$

$\therefore\ \varepsilon_1 = 0.25 \times 2 = 0.5$

$\therefore\ \varepsilon = 0.5$

d)

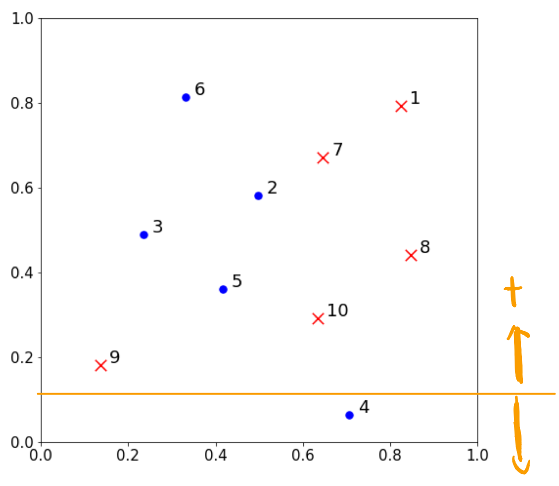


Figure 2: ●-negative points and ×-positive points.

- we have to have 9, 4 correctly classified.
- So we draw a horizontal line between 4 & 9
- upper half is "+" lower half is "−"

- we have $\varepsilon_1 = 0.0625 \times 4$ (only 4 neg mis) $= 0.25$

e)
- Yes
- $\alpha_1 = \frac{1}{2} \cdot \log\left(\frac{1-0.25}{0.25}\right) = \frac{1}{2}\ln(3) < \frac{1}{2}\ln 4 = \alpha_0$
- Therefore, the first classifier has higher weight than the second one. So, 4 and 9 may still be mis classified while we combine two weak classifier.

APPENDIX: CODE

```
"""
EECS 445 - Introduction to Machine Learning
Winter 2019
Homework 2, Ensemble Methods
Skeleton Code
"""

import random
import numpy as np
import matplotlib.pyplot as plt

from collections import Counter
from sklearn import metrics, utils
from sklearn.datasets import fetch_mldata
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split


def load_mnist(classes):
    """
    Load MNIST dataset for classes
    Every 25th sample is used to reduce computational resources
    Input:
        classes : list of ints
    Returns:
        X : np.array (num_samples, num_features)
        y : np.array (num_samples)
    """
    print('Fetching MNIST data...')
    mnist = fetch_mldata('MNIST original')
    X_all = np.array(mnist.data)[::25]
    y_all = np.array(mnist.target)[::25]
    desired_idx = np.isin(y_all, classes)
    return X_all[desired_idx], y_all[desired_idx]


def get_avg_performance(X, y, m_vals, n_splits=50):
    """
    Compare the average performance of bagging and random forest across 50
    random splits of X and y
    Input:
        X : np.array (num_samples, num_features)
        y : np.array (num_samples)
        m_vals: list - list of values for m
        n_splits: int - number of random splits
    Returns:
        bag_results : np.array (len(m_vals)) - estimate of bagging performance
        rf_results : np.array (len(m_vals)) - estimate of random forest performance
    """
    print('Getting bagging and random forest scores...')
    rf_results = []
    bag_results = []
    for m in m_vals:
        print('m = {}'.format(m))
```

```
        bagging_scores = []
        random_forest_scores = []
        for i in range(n_splits):
            X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
            random_forest_scores.append(random_forest(X_train, y_train, X_test, y_test, m))
            bagging_scores.append(bagging_ensemble(X_train, y_train, X_test, y_test))
        rf_results.append(np.median(np.array(random_forest_scores)))
        bag_results.append(np.median(np.array(bagging_scores)))
    return bag_results, rf_results


def plot_data(bagging_scores, random_forest_scores, m_vals):
    """
    Plot bagging and random forest accuracies
    Input:
        bagging_scores : np.array - array containing accuracies for bagging ensemble classifiers
        random_forest_scores : np.array - array containing accuracies for random forest classifiers
    """
    plt.figure()
    plt.plot(list(m_vals), bagging_scores, '--', label='bagging')
    plt.plot(list(m_vals), random_forest_scores, '--', label='random forest')
    plt.xlabel('m')
    plt.ylabel('Accuracy')
    plt.legend(loc='upper right')
    plt.savefig('ensemble.png', dpi=300)
    plt.show()


def random_forest(X_train, y_train, X_test, y_test, m, n_clf=10):
    """
    Returns accuracy on the test set X_test with corresponding labels y_test
    using a random forest classifier with n_clf decision trees trained with
    training examples X_train and training labels y_train.
    Input:
        X_train : np.array (n_train, d) - array of training feature vectors
        y_train : np.array (n_train) - array of labels corresponding to X_train samples
        X_test : np.array (n_test,d) - array of testing feature vectors
        y_test : np.array (n_test) - array of labels corresponding to X_test samples
        m : int - number of features to consider when splitting
        n_clf : int - number of decision tree classifiers in the random forest, default is 10
    Returns:
        accuracy : float - accuracy of random forest classifier on X_test samples
    """
    # TODO: Implement this function

    y_predict = np.zeros((10,X_test.shape[0]))
    boot_size = X_train.shape[0]
    for i in range(10):
        X_boot, y_boot = utils.resample(X_train, y_train, n_samples = boot_size)
        clf = DecisionTreeClassifier(criterion = 'entropy', max_features = m)
        clf.fit(X_boot, y_boot)
        y_pred = clf.predict(X_test)
        y_predict[i] = y_pred

    y_pred = []
    for i in range(X_test.shape[0]):
        y_pred = np.append(y_pred,Counter(y_predict[:,i]).most_common(1)[0][0])
```

```
    return metrics.accuracy_score(y_test, y_pred)


def bagging_ensemble(X_train, y_train, X_test, y_test, n_clf=10):
    """
    Returns accuracy on the test set X_test with corresponding labels y_test
    using a bagging ensemble classifier with n_clf decision trees trained with
    training examples X_train and training labels y_train.
    Input:
        X_train : np.array (n_train, d) - array of training feature vectors
        y_train : np.array (n_train) - array of labels corresponding to X_train samples
        X_test : np.array (n_test,d) - array of testing feature vectors
        y_test : np.array (n_test) - array of labels corresponding to X_test samples
        n_clf : int - number of decision tree classifiers in the random forest, default is 10
    Returns:
        accuracy : float - accuracy of random forest classifier on X_test samples
    """
    # TODO: Implement this function

    y_predict = np.zeros((10,X_test.shape[0]))
    boot_size = X_train.shape[0]
    for i in range(10):
        X_boot, y_boot = utils.resample(X_train, y_train, n_samples = boot_size)
        clf = DecisionTreeClassifier(criterion = 'entropy')
        clf.fit(X_boot, y_boot)
        y_pred = clf.predict(X_test)
        y_predict[i] = y_pred

    y_pred = []
    for i in range(X_test.shape[0]):
        y_pred = np.append(y_pred,Counter(y_predict[:,i]).most_common(1)[0][0])


    return metrics.accuracy_score(y_test, y_pred)


def main():
    """
    Analyze how the performance of bagging and random forest changes with m.
    """
    X, y = load_mnist([1,2,3,4])
    # Plot accuracies
    m_vals = [1] + list(range(56, 785, 56))
    bagging_scores, random_forest_scores = get_avg_performance(X, y, m_vals)
    plot_data(bagging_scores, random_forest_scores, m_vals)


if __name__ == '__main__':
    main()
```