CS 229, Problem Set #2

Summer 2023

Reyansh Gupta - reyansh - 06776137

Q1

a) The vocabulary size of the dictionary is: 1757

b) The accuracy obtained is 0.978 on the test set

c) The top 5 indicative words are: 'Claim'
'urgent!'
'tone'
'prize'
'won'

```
Size of dictionary:  1757
Naive Bayes had an accuracy of 0.978494623655914 on the testing set
The top 5 indicative words for Naive Bayes are:  ['claim', 'urgent!', 'tone', 'prize', 'won']
```

Q2

By definition → A Kernel exists when there is a feature map ϕ such that $K(x,y) = \langle \phi(x), \phi(y) \rangle$.

Also, for a Kernel function to be valid, it should be symmetric and Positive semi definite

a) $K(x,z) = K_1(x,z) + K_2(x,z)$

$$= \langle \phi_1(x), \phi_1(z) \rangle + \langle \phi_2(x), \phi_2(z) \rangle$$

$$= \left\langle \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}, \begin{bmatrix} \phi_1(z) \\ \phi_2(z) \end{bmatrix} \right\rangle$$

∴ It follows the rules and is a valid Kernel.

b) Let's assume $K_1(x,z)$ and $K_2(x,z)$ are deterministic but valid Kernels. If value of $K_2 > K_1$ → for example: $K_2(x,z) = 5$, $K_1(x,z) = 2$ → these are valid Kernels but $K(x,z) = 2-5 = -3$ which is less than 0. This violates the property of a Kernel being valid.

∴ Kernel is invalid.

c) $K(x,z) = a \langle \phi_1(x), \phi_1(z) \rangle$

$$\sqrt{a}\phi_1(x) \cdot \sqrt{a}\phi_1(x)$$

$$= \langle \sqrt{a}\phi_1(x), \sqrt{a}\phi_1(z) \rangle$$

Which is a valid Kernel

d) If we take a negative example where Kernel value is >0, a will be <0 which makes the Kernel invalid

e) $K(x,z) = K_1(x,z) K_2(x,z)$

$$K_1(x,z) = \sum_{i=1}^{n} \phi_1^{(i)T}(x)\, \phi_1^{(i)}(z) \qquad K_2(x,z) = \sum_{j=1}^{n} \phi_2^{(n)T}(x)\, \phi_2^{(n)}(z)$$

$$K(x,z) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\phi_1^{(i)T}(x)\, \phi_2^{(j)}(x)\right)\left(\phi_1^{(i)T}(z)\, \phi_2^{(j)}(z)\right)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \phi_{ij}(x)\, \phi_{ij}(z)$$

$$= \left\langle \begin{bmatrix} \phi_{11}(x) \\ \vdots \\ \phi_{nn}(x) \end{bmatrix}, \begin{bmatrix} \phi_{11}(z) \\ \vdots \\ \phi_{nn}(z) \end{bmatrix} \right\rangle$$

Hence, we can conclude that $K(x,z)$ is a valid Kernel

f) $K_{ij} = K(x_i, x_j) = f(x_i) \cdot f(x_j)$

Let 'a' be a vector → to prove K is PSD → $a^T K a$ should be ≥ 0

$$a^T K a = [a_1 \dots a_n] \begin{bmatrix} f(x_1)f(x_1) & \cdots & f(x_1)f(x_n) \\ f(x_2)f(x_1) & & \vdots \\ \vdots & & \vdots \\ f(x_n)f(x_1) & & f(x_n)f(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$$a^T K a = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot a_j \cdot f(x_i) \cdot f(x_j)$$

Assuming $f(x_i)$ and $f(x_j)$ are real valued functions for all points x_i and x_j.
$a_i \cdot a_j \cdot f(x_i) \cdot f(x_j)$ is always positive

Since $a^T K a \geq 0$ for all a, it is a valid Kernel

g) $K(x,z) = K_3(\phi(x), \phi(z))$
$= \langle \phi_3(\phi(x)), \phi_3(\phi(x)) \rangle$

Let 'a' be a vector → to prove K is PSD → $a^T K a$ should be ≥ 0

$$a^T K a = [a_1 \cdots a_n] \begin{bmatrix} K_3(\phi(x_1), \phi(x_1)) & \cdots & K_3(\phi(x_1), \phi(x_n)) \\ \vdots & \ddots & \vdots \\ K_3(\phi(x_n), \phi(x_1)) & & K_3(\phi(x_n), \phi(x_n)) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

We can write this as:
$$a^T K a = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \langle \phi_3(\phi(x_i)), \phi_3(\phi(x_j)) \rangle$$

$\langle \phi_3(\phi(x_i)), \phi_3(\phi(x_j)) \rangle$ is the inner dot product in 3^{rd} dimensional feature space and hence is always non negative

Since $a^T K a$ is ≥ 0 for all a, it's a valid Kernel

h) Let $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ where $a_0 \dots a_n \geq 0$. Then →

$K(x,z) = p(K_1(x,z))$
$= a_0 + a_1 K_1(x,z) + a_2 (K_1(x,z))^2 + \cdots + a_n (K_1(x,z))^n$

In questions →
a) We proved that $K_1(x,z) + K_2(x,z)$ is a valid Kernel
c) We proved that $a K_1(x,z)$ is a valid Kernel
e) We proved that $K_1(x,z) \cdot K_2(x,z)$ is a valid Kernel.

Using these results we can prove that $K(x,z) = p(K_1(x,z))$ is a valid Kernel

Q3 - A

Given update rule:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha\left(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)})\right)\phi(x^{(i+1)})$$

↳ θ value after $i+1$ iterations over training pts

Since $\theta^{(0)} = 0$

for $k = 1$ to i, θ is a linear combination of $\phi(x^{(k)})$ rescaled by some value (β_i)

$$\therefore \boxed{\theta^{(i)} = \sum_{k=1}^{i} \beta_k \phi(x^{(k)})}$$

i) As stated before → $\theta^{(0)} = 0$

which is the initialisation value

$\therefore \boxed{\theta^{(0)} = \beta_0 \phi(x^{(0)})}$ → here, $\beta = 0$ and $x^{(0)}$ does not exist → hence $\theta^{(0)} = 0$.

ii) for new prediction $x^{(i+1)}$ →

$$h_{\theta^{(i)}}(x^{(i+1)}) = g\left(\theta^{(i)T}\phi(x^{(i+1)})\right)$$

$$= \text{sign}\left(\sum_{k=1}^{i} \beta_k \phi(x^{(k)})^T \phi(x^{(i+1)})\right)$$

$$\boxed{h_{\theta^{(i)}}(\phi(x^{(i+1)})) = \text{sign}\left(\sum_{k=1}^{i} \beta_k < \phi(x^{(k)}), \phi(x^{(i+1)}) >\right)}$$

iii) Update rule:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha\left(y^{(i)} - h_{\theta^{(i)}}(\phi(x^{(i+1)}))\right)\phi(x^{(i+1)})$$

and $\theta^{(i+1)} = \sum_{k=1}^{i+1} \beta_k \phi(x^{(k)})$ (By the equation derived in previous results) —— ①

$$\therefore \theta^{(i+1)} = \underbrace{\sum_{k=1}^{i} \beta_k \phi(x^{(k)})}_{\text{till } i^{th} \text{ iteration}} + \alpha\left(y^{(i)} - g\left(\sum_{k=1}^{i} \beta_k < \phi(x^{(k)}), \phi(x^{(i+1)}) >\right)\right)\phi(x^{(i+1)}) \quad \text{—— ②}$$

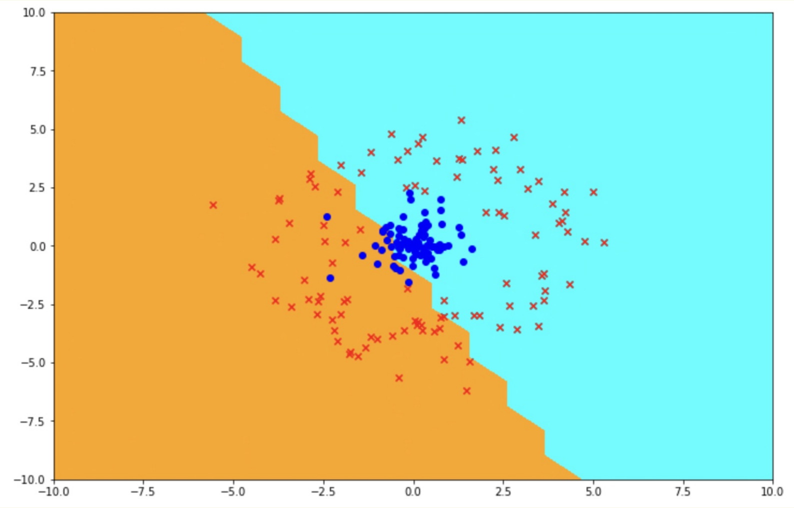
equating ① and ②, we get →

$$\underbrace{\beta_{i+1}\phi(x^{(i+1)})}_{(i+1)^{th} \text{ element}} + \sum_{k=1}^{i} \beta_k \phi(x^{(k)}) = \sum_{k=1}^{i} \beta_k \phi(x^{(k)}) + \alpha\left(y^{(i)} - g\left(\sum_{k=1}^{i} \beta_k < \phi(x^{(k)}), \phi(x^{(i+1)}) >\right)\right)\phi(x^{(i+1)})$$

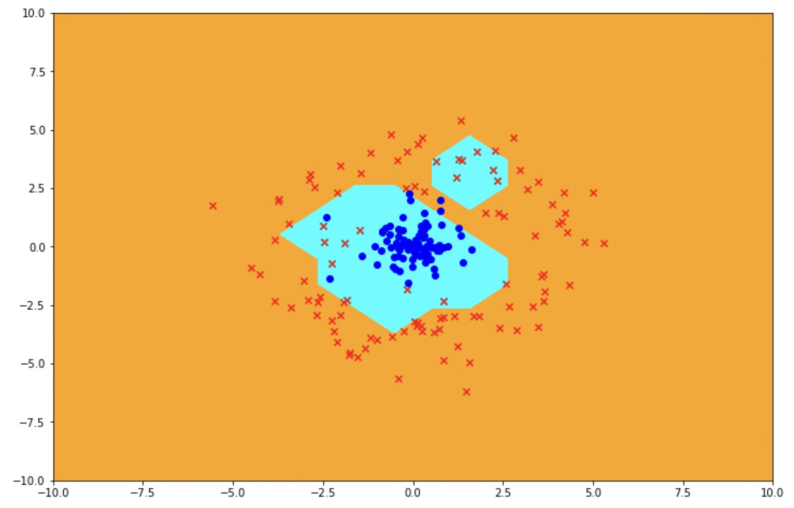
∴ Update rule becomes:

$$\boxed{\beta_{i+1} = \alpha\left(y^{(i)} - g\left(\sum_{k=1}^{i} \beta_k < \phi(x^{(k)}), \phi(x^{(i+1)}) >\right)\right)}$$

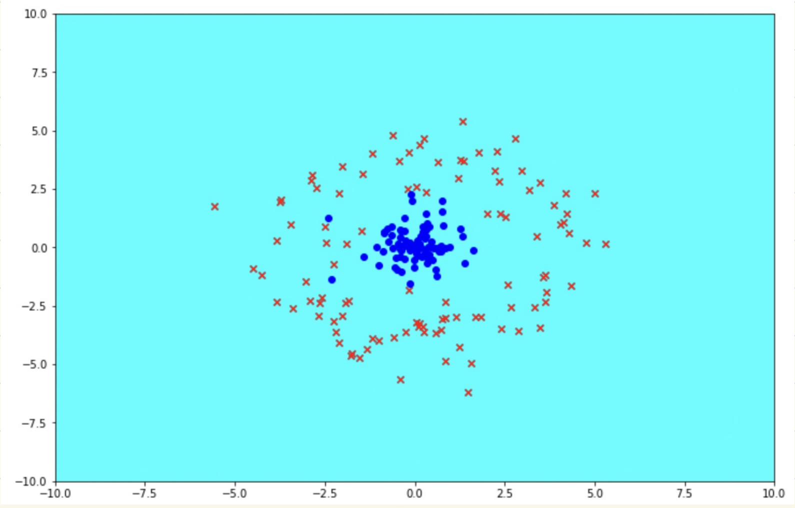
Q3 - B



a) Dot - product Kernel



b) RBF Kernel



c) Not a valid Kernel

Q3 - C

Dot product Kernel → attempted to establish a linear decision boundary but was not adequate for our non-linear dataset → leading to a non satisfactory result.

RBF Kernel → Since our data is somewhat radially distributed → hence, RBF could provide us with a better decision boundary and could learn the non linearity in the data easily.

Invalid Kernel → The invalid Kernel failed to learn anything about the data and gave no meaningful insights

Q4 - A

To show:

$$\nabla_{z^{(i)}} CE(y^{(i)}, \hat{y}^{(i)}) = \hat{y}^{(i)} - y^{(i)} \in \mathbb{R}^K$$

where $z^{(i)} \in \mathbb{R}^K$ is input to softmax func →

$$\hat{y}^{(i)} = \text{softmax}(z^{(i)})$$

$$CE = -\sum_{k=1}^{K} y_k^{(i)} \cdot \log(\hat{y}_k^{(i)}) \quad \text{—— (1)}$$

$$\rightarrow \hat{y}_k^{(i)} = \text{softmax}(z_k^{(i)}) = \frac{\exp(z_k^{(i)})}{\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})} \quad \text{—— (2)}$$

Substituting (2) in (1) →

$$CE = -\left(\sum_{k=1}^{K} y_k^{(i)} \cdot \log\left(\frac{\exp(z_k^{(i)})}{\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})}\right)\right)$$

$$= -\left(\sum_{k=1}^{K} y_k^{(i)} \cdot z_k^{(i)} - \sum_{k=1}^{K} y_k^{(i)} \cdot \log\left(\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})\right)\right)$$

$$= -\left(\sum_{k=1}^{K} y_k^{(i)} \cdot z_k^{(i)} - \log\left(\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})\right) \sum_{k=1}^{K} y_k^{(i)}\right)$$ (Term inside log not iterating over 'k')

$\sum_{k=1}^{K} y_k^{(i)} = 1$ [because it's a basis vector]

$$\nabla_{z^{(i)}} CE = \left[\frac{\partial CE}{\partial z_1^{(i)}}, \frac{\partial CE}{\partial z_2^{(i)}} \cdots \frac{\partial CE}{\partial z_K^{(i)}}\right]$$

$$\frac{\partial CE}{\partial z_j^{(i)}} = -\left(\frac{\partial}{\partial z_j^{(i)}}\left(\sum_{k=1}^{K} y_k^{(i)} \cdot z_k^{(i)}\right) - \frac{\partial}{\partial z_j^{(i)}} \log\left(\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})\right)\right)$$

$$= -\left(y_j^{(i)} - \frac{1}{\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})} \cdot \frac{\partial \sum_{\ell=1}^{K} \exp(z_\ell^{(i)})}{\partial z_j^{(i)}}\right)$$ $\left(\frac{\partial z_k^{(i)}}{\partial z_j^{(i)}} = 0, \text{ for all } j \neq k\right)$

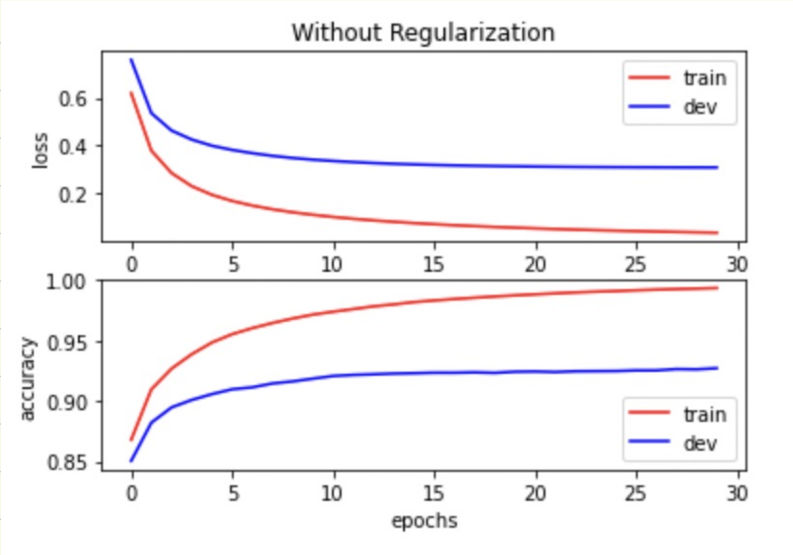
$$= -\left(y_j^{(i)} - \frac{1}{\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})} \cdot \exp(z_j^{(i)})\right)$$

$$\left(\frac{1}{\sum_{\ell=1}^{K} \exp(z_\ell^{(i)})} \cdot \exp(z_j^{(i)}) \rightarrow \text{softmax}(z_j^{(i)}) = \hat{y}_j^{(i)}\right)$$

$$\frac{\partial CE}{\partial z_j^{(i)}} = -\left(y_j^{(i)} - \hat{y}_j^{(i)}\right) = \hat{y}_j^{(i)} - y_j^{(i)}$$

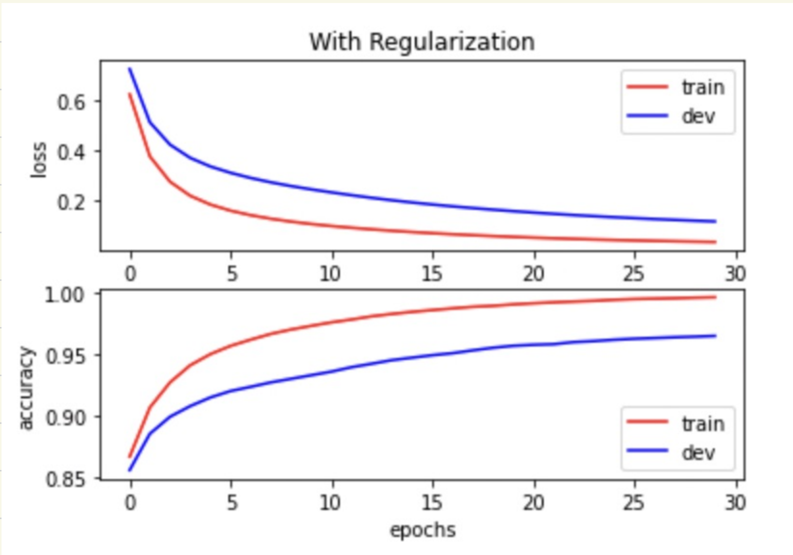
$$\boxed{\therefore \nabla_{z^{(i)}} CE = \hat{y}^{(i)} - y^{(i)}}$$ [Vector notation]

Q4 - B



Model without regularization

Q4 - C



Regularized Model

Both the models achieved similar high levels of accuracy on the training data.

However, we can see that the non regularized model had a larger gap between its training and development accuracies.

This suggests that the model had more variance problem compared to the regularized model.

∴ Regularizing helped in solving this issue.

Q4 - D

Test accuracy → 0.932 (Without regularization)

0.9653 (With regularization)

```
For model baseline, got accuracy: 0.932000
For model regularized, got accuracy: 0.965300
```

Q5 - A

Given:

$$\theta_{MAP} = \arg\max_{\theta} p(\theta | x, y)$$

$$= \arg\max_{\theta} \frac{p(\theta|x)\, p(y|x,\theta)}{p(y|x)} \quad [\text{From bayes' rule}]$$

We can remove the denominator since it's not dependent on θ.

$$\Rightarrow \arg\max_{\theta} p(\theta|x)\, p(y|x,\theta)$$

From the assumption $\rightarrow p(\theta) = p(\theta|x)$, we have:

$$\theta_{MAP} = \arg\max_{\theta} p(y|x,\theta) \cdot p(\theta)$$

Q5 - B

From a) $\rightarrow$ $\theta_{MAP} = \arg\max_{\theta} p(y|x,\theta) \cdot p(\theta)$

$$= \arg\max_{\theta} \log\left(p(\theta)\, p(y|x,\theta)\right)$$

$$\left[p(\theta) = \frac{1}{(2\pi)^{d/2} |\eta^2 I|^{1/2}} \exp\left(-\frac{1}{2} \|\theta\|_2^2 \cdot (\eta^2)^{-1}\right) \right] \qquad \left(\theta \sim N(0, \eta^2 I)\right)$$

$$\Rightarrow \arg\min_{\theta} -\log p(y|x,\theta) - \log\left(\frac{1}{(2\pi)^{d/2} |\eta^2 I|^{1/2}} \exp\left(\frac{-\|\theta\|_2^2}{2\eta^2}\right)\right)$$

$$= \arg\min_{\theta} -\log p(y|x,\theta) + \frac{\|\theta\|_2^2}{2\eta^2} \qquad [\arg\min \rightarrow \text{can remove the denominator}]$$

Comparing the equations, we get $\lambda = \frac{1}{2\eta^2}$

Q5 - C

The whole dataset can be written as:

$$\vec{y} = X\theta + \vec{\varepsilon}$$

where $\vec{\varepsilon} \sim \mathcal{N}(0, \sigma^2)$ and $\theta \sim \mathcal{N}(0, \lambda^2 I)$

To obtain a closed form solution → we need to calculate $p(\vec{y}|x, \theta)$ [from b]

$$\Rightarrow \vec{y}|x, \theta \sim \mathcal{N}(x\theta, \sigma^2)$$

$$p(\vec{y}|x, \theta) = \frac{1}{(2\pi)^{d/2} |\sigma^2|^{1/2}} \exp\left(-\frac{\|\vec{y} - X\theta\|_2^2}{2\sigma^2}\right)$$

from b) →

$$\theta_{MAP} = \arg\min_\theta - \log p(y|x, \theta) + \frac{\|\theta\|_2^2}{2\lambda^2}$$

$$= \arg\min_\theta - \log\left[\frac{1}{(2\pi)^{d/2} |\sigma^2 I|^{1/2}} \exp\left(-\frac{\|\vec{y} - X\theta\|_2^2}{2\sigma^2}\right)\right] + \frac{\|\theta\|_2^2}{2\lambda^2}$$

$$= \arg\min_\theta \frac{\|y - X\theta\|_2^2}{2\sigma^2} + \frac{\|\theta\|_2^2}{2\lambda^2}$$ [arg min → can remove the denominator]

$$= \arg\min_\theta \|y - x\theta\|_2^2 + \frac{\sigma^2}{\lambda^2}\|\theta\|_2^2$$

Let this be the loss function → $J(\theta)$. To find θ such that $J(\theta)$ is minimized → we find the gradient of $J(\theta)$ wrt θ and make it equal to 0.

$$J(\theta) = \|y - x\theta)\|_2^2 + \frac{\sigma^2}{\lambda^2}\|\theta\|_2^2$$

$$\nabla_\theta J(\theta) = -2x^T(\vec{y} - x\theta) + \frac{2\sigma^2}{\lambda^2}(\theta) = 0$$

$$= 2x^T(x\theta - \vec{y}) + \frac{2\sigma^2}{\lambda^2}(\theta) = 0$$

$$= 2x^Tx\theta - 2x^T\vec{y} + \frac{2\sigma^2}{\lambda^2}(\theta) = 0$$

$$= \theta\left(x^Tx + \frac{\sigma^2}{\lambda^2}\right) - x^T\vec{y} = 0$$

$$\theta = \left(x^Tx + \frac{\sigma^2}{\lambda^2}\right)^{-1} x^T\vec{y}$$

∴ Closed form expression for θ_{MAP} ⇒

$$\boxed{\theta_{MAP} = \left(X^Tx + \frac{\sigma^2}{\lambda^2}\right)^{-1} x^T\vec{y}}$$

Q5 - D

Given distribution:

$$b_L(z|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|z-\mu|}{b}\right) \quad \text{[probability density]}$$

$$\vec{y} = X\theta + \vec{\epsilon} \quad \text{where} \quad \epsilon \sim N(0, \sigma^2)$$

and $\theta_i \sim \text{Laplace}(0, b)$ [for every θ_i where $i = 1, \dots n$]

and as before $\rightarrow$ $\vec{y}|X \sim N(X\theta, \sigma^2)$

also,

$$\theta_{MAP} = \arg\min_{\theta} -\log(p(\theta)\, p(\vec{y}|x,\theta))$$

$$= \arg\min_{\theta} -\log\left(p(\vec{y}|x,\theta)\right) - \log p(\theta)$$

$$= \arg\min_{\theta} -\log\left(p(\vec{y}|x,\theta)\right) - \log \prod_{i=1}^{n} p(\theta_i)$$

[We have $\rightarrow$ $p(\vec{y}|x,\theta) = \frac{1}{(2\pi)^{d/2} |\sigma^2|^{1/2}} \exp\left(-\frac{\|\vec{y} - X\theta\|_2^2}{2\sigma^2}\right)$; $p(\theta_i) = \frac{1}{2b} \exp\left(-\frac{|\theta_i|}{b}\right)$]

$$\Rightarrow \arg\min_{\theta} -\log\left[\frac{1}{(2\pi)^{d/2} |\sigma^2|^{1/2}} \exp\left(-\frac{\|\vec{y} - X\theta\|_2^2}{2\sigma^2}\right)\right] - \sum_{i=1}^{n} \log\left[\frac{1}{2b} \exp\left(-\frac{|\theta_i|}{b}\right)\right]$$

$$= \arg\min_{\theta} \frac{\|\vec{y} - X\theta\|_2^2}{2\sigma^2} + \sum_{i=1}^{n} \frac{|\theta_i|}{b}$$ [arg min $\rightarrow$ can remove the denominator]

$$= \arg\min_{\theta} \|\vec{y} - X\theta\|_2^2 + \frac{2\sigma^2}{b} \|\theta\|_1$$

Comparing this to $J(\theta) = \|X\theta - \vec{y}\|_2^2 + \lambda \|\theta\|_1$

We can see that θ_{MAP} is equivalent to $J(\theta)$

$$\boxed{\lambda = \frac{2\sigma^2}{b}}$$