Problem 1

Given that $p(w) \sim \mathcal{N}(0, \tau^2 I)$

$$\Rightarrow p(w) = \frac{1}{\sqrt{(2\pi)^M \det(\tau^2 I)}} \exp\left(-\frac{1}{2} w^T (\tau^2 I)^{-1} w\right)$$

$$= \frac{1}{\tau^M \sqrt{(2\pi)^M}} \exp\left(-\frac{1}{2} w^T \frac{1}{\tau^2} I \; w\right)$$

$$= \frac{1}{\tau^M \sqrt{(2\pi)^M}} \exp\left(-\frac{\|w\|_2^2}{2\tau^2}\right)$$

We know that: w_{ML} maximizes $\prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w)$

w_{MAP} maximizes $p(w) \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w)$

$\Rightarrow$ w_{MAP} does not maximize $\prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w)$

$$\Rightarrow \frac{1}{\tau^M \sqrt{(2\pi)^M}} \exp\left(-\frac{\|w_{MAP}\|_2^2}{2\tau^2}\right) \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{MAP}) \geq \frac{1}{\tau^M (2\pi)^{M/2}} \exp\left(-\frac{\|w_{ML}\|_2^2}{2\tau^2}\right) \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{ML}) \quad \text{(1)}$$

$$\Rightarrow \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{ML}) \geq \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{MAP}) \quad \text{(2)}$$

sub (2) in (1)

$$\exp\left(-\frac{\|w_{MAP}\|_2^2}{2\tau^2}\right) \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{ML}) \geq \exp\left(-\frac{\|w_{ML}\|_2^2}{2\tau^2}\right) \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w_{ML})$$

$$\Rightarrow \quad -\|w_{MAP}\|_2^2 \geq -\|w_{ML}\|_2^2$$

$$\Rightarrow \quad \|w_{MAP}\|_2^2 \leq \|w_{ML}\|_2^2$$

$$\boxed{\Rightarrow \quad \|w_{MAP}\|_2 \leq \|w_{ML}\|_2}$$

Problem 2

a) $k(x,z) = k_1(x,z) + k_2(x,z)$

let $\vec{v} \in \mathbb{R}^D$; $v^T K v = v^T (K_1 + K_2) v$

$= \underbrace{v^T K_1 v}_{\geq 0} + \underbrace{v^T K_2 v}_{\geq 0}$

∵ they are given to be valid kernels they are PSD

$\Rightarrow v^T K v \geq 0 \Rightarrow K$ is PSD —①

∵ K_1 & K_2 are valid kernels, they are symmetric —②

① & ② show that $k(x,z)$ is a valid kernel.

where $K, K_1, K_2 \in \mathbb{R}^{D \times D}$

& $k_1(x,z) = \phi^{(1)}(x)^T \phi^{(1)}(z)$

& $k_2(x,z) = \phi^{(2)}(x)^T \phi^{(2)}(z)$

b) $k(x,z) = k_1(x,z) + k_2(x,z)$ Invalid Kernel

the difference between two PSD matrices is not necessarily PSD.

$v^T K v = v^T K_1 v - v^T K_2 v$

$v \in \mathbb{R}^D$ if $K_2 = 2K_1$

$v^T K v < 0 \Rightarrow$ not PSD, thus not a kernel

where $K, K_1, K_2 \in \mathbb{R}^{D \times D}$

& $k_1(x,z) = \phi^{(1)}(x)^T \phi^{(1)}(z)$

& $k_2(x,z) = \phi^{(2)}(x)^T \phi^{(2)}(z)$

c) $k(x,z) = a k_1(x,z)$

∵ K_1 is a symmetric matrix aK_1 is also symm.

∵ $a \in \mathbb{R}^+ \Rightarrow a v^T K_1 v \geq 0$

$\vec{v} \in \mathbb{R}^D \Rightarrow K$ is PSD

$\Rightarrow$ Valid Kernel

where $K, K_1, K_2 \in \mathbb{R}^{D \times D}$

& $k_1(x,z) = \phi^{(1)}(x)^T \phi^{(1)}(z)$

d) $k(x,z) = -a k_1(x,z)$

∵ K_1 is a symmetric matrix $-aK_1$ is also symm.

∵ $a \in \mathbb{R}^+ \Rightarrow v^T K v = v^T (-aK_1) v$

$= -a v^T K_1 v \leq 0$

$\Rightarrow$ not PSD.

i.e. not a valid kernel

where $K, K_1, K_2 \in \mathbb{R}^{D \times D}$

& $k_1(x,z) = \phi^{(1)}(x)^T \phi^{(1)}(z)$

e) $k(x,z) = k_1(x,z)\, k_2(x,z)$

This is elementwise multiplication. Elementwise multiplication is commutative $\Rightarrow$ K is also symmetric.

∵ K_1 & K_2 are symmetric they have valid unitary eigendecompositions.

let $K_1 = U \Lambda U^T = \sum_{i=1}^{D} \lambda_i u_i u_i^T$ $\quad K_2 = V \Sigma V^T = \sum_{j=1}^{D} \sigma_j v_j v_j^T$ where U & V are unitary

$$\Rightarrow K = K_1 \odot K_2 = \sum_{i,j} \lambda_i \sigma_j \underbrace{(u_i u_i^T)}_{\|u_i\|_2^2} \odot \underbrace{(v_j v_j^T)}_{\|v_j\|_2^2}$$

$$= \sum_{i,j} \lambda_i \sigma_j (u_i \odot v_j)(u_i \odot v_j)^T$$

this is also a valid unitary eigen decomp.
with $\lambda_i \sigma_j \geq 0 \Rightarrow K$ has eigenvalues ≥ 0
i.e. it's PSD
$\Rightarrow$ K is symm. & PSD $\Rightarrow$ valid kernel

f) $k(x,z) = f(x) f(z)$
$k(x,z)$ is symm. $\because$ scalar multiplication is commutative
let $f(x) = \phi^{(1)}(x)\, \phi^{(2)}(x)$

$$\Rightarrow K = \sum_i \sum_j \left[\phi_i^{(1)}(x)\, \phi_j^{(2)}(x)\right]\left[\phi_i^{(1)}(z)\, \phi_j^{(2)}(z)\right]$$

$$= \sum_i \phi_i^{(1)}(x)\, \phi_i^{(1)} z \sum_j \phi_j^{(2)}(x)\, \phi_j^{(2)}(z)$$

$$= K_1 \odot K_2$$

which is a valid kernel from part e.

g) $k(x,z) = k_3(\phi(x), \phi(z))$

$$= \phi(\phi(x))^T \phi(\phi(z))$$

$\phi : \mathbb{R}^D \to \mathbb{R}^M$ & $K_3 : \mathbb{R}^{M \times M}$
$\because$ ϕ is mapping to same domain as K_3, it's a valid kernel

h) $k(x,z) = p(k_1(x,z))$ $\quad p: \mathbb{R} \to \mathbb{R}$

$K = c_n k_1(x,z)^n + c_{n-1} k_1(x,z)^{n-1} + \cdots + c_0$, n= power
combined results from 2a, 2e, 2e $\quad$ c= coeff ≥ 0
$\Rightarrow$ K is a valid kernel

i) $k(x,z) = (x^T z + 1)^2 = (x^T z)^2 + 2x^T z + 1$

$$= (x^T)^2 z^2 + (\sqrt{2} x^T)(\sqrt{2} z) + 1$$

$\Rightarrow \phi(x) = [x^2, \sqrt{2}x, 1]$

j) $k(x,z) = \exp(-\|x-z\|^2/2\sigma^2)$

$$= \exp\left(\frac{-x^Tx - z^Tz + 2x^Tz}{2\sigma^2}\right)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-x^Tx}{2\sigma^2}\right)^n \sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-z^Tz}{2\sigma^2}\right)^n \sum_{k=0}^{\infty} \frac{1}{k!}\left(\frac{x^T}{\sigma}\right)^k \left(\frac{z^T}{\sigma}\right)^k$$

$$= \left(\sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-x^Tx}{2\sigma^2}\right)^n \sum_{k=0}^{\infty} \frac{1}{\sqrt{k!}}\left(\frac{x^T}{\sigma}\right)^k\right)\left(\sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-z^Tz}{2\sigma^2}\right)^n \sum_{k=0}^{\infty} \frac{1}{\sqrt{k!}}\left(\frac{z^T}{\sigma}\right)^k\right)$$

$$\Rightarrow \phi(x) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-x^Tx}{2\sigma^2}\right)^n \sum_{k=0}^{\infty} \frac{1}{\sqrt{k!}}\left(\frac{x^T}{\sigma}\right)^k$$

$$\phi(z) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-z^Tz}{2\sigma^2}\right)^n \sum_{k=0}^{\infty} \frac{1}{\sqrt{k!}}\left(\frac{z^T}{\sigma}\right)^k$$

from prev parts (2c, 2e, 2h)

$k(x,z) = \phi(x)^T\phi(z)$ is a valid kernel

Problem 3

(a) (i) Given: $\vec{w}_{t+1} \leftarrow w_t + y^{(n)}\vec{x}^{(n)}$ $\quad \vec{w} = \phi^T \vec{\alpha}_t$

case: $y^{(n)} = 1$

if $y^{(n)} h < 0$

$\vec{w}_{t+1} \leftarrow \Phi^T \vec{\alpha}_t + \phi(x^{(i)})$

$\phi(x^{(i)}) = \Phi^T e^i$ where e^i is the elementary vector with 1 at position (i).

$\Rightarrow \vec{w}_{t+1} \leftarrow \Phi^T (\vec{\alpha}_t + e^i)$ $\quad -①$

case: $y^{(n)} = -1$

if $y^{(n)} h < 0$

$\vec{w}_{t+1} \leftarrow \Phi^T \vec{\alpha}_t - \Phi^T e^i$

$= \Phi^T (\vec{\alpha}_t - e^i)$ $\quad -②$

using ① & ②

$$\vec{w}_{t+1} = \begin{cases} \Phi^T (\underbrace{\vec{\alpha}_t + y^{(i)} e^i}_{\vec{\alpha}_{t+1}}) & \text{when } y^{(n)} \vec{w}_t^T \phi(x^{(n)}) < 0 \\ \Phi^T \vec{\alpha}_t & \text{otherwise} \end{cases}$$

this shows $\vec{w}_{t+1}$ is of the form $\Phi^T \vec{\alpha}_{t+1}$

(ii) Proof by induction:

base case:

(1) given $\vec{w}_0 \leftarrow 0 \Rightarrow \vec{w}_0 = \Phi^T \vec{\alpha}_0$ where $\vec{\alpha}_0 = \vec{0}$

(2)

$$\vec{w}_1 = \begin{cases} \Phi^T (\vec{0} + y^{(i)} e^{(i)}) & y^{(n)} h < 0 \\ \Phi(\vec{0}) & \text{otherwise} \end{cases}$$

$$\Rightarrow \vec{w}_1 = \begin{cases} \Phi^T (y^{(i)} e^{(i)}) & y^{(i)} h < 0 \\ \vec{0} & \text{otherwise} \end{cases}$$

thus the base case holds when $\vec{w}_0$ & $\vec{w}_1$ were shown to be in the form of $\vec{w}_0 = \Phi^T \vec{\alpha}_0$ & $\vec{w}_1 = \Phi^T \vec{\alpha}_1$

Induction hypothesis: Assume that the prop holds true till $t-1$.

Induction step:

$$\vec{w}_t \leftarrow \begin{cases} \Phi^T(\vec{\alpha}_{t-1} + y^{(i)} e^{(i)}) & y^{(i)} h < 0 \\ \Phi^T \vec{\alpha}_{t-1} & \text{o/w} \end{cases}$$

$\vec{\alpha}_{t-1} + y^{(i)} e^{(i)}$ is an update to $\vec{\alpha}_{t-1}$

$$\Rightarrow \vec{\alpha}_t \leftarrow \vec{\alpha}_{t-1} + y^{(i)} e^{(i)}$$

$$\Rightarrow \vec{w}_t \leftarrow \begin{cases} \Phi^T \vec{\alpha}_t & y^{(i)} h < 0 \\ \Phi^T \vec{\alpha}_{t-1} & \text{o/w} \end{cases}$$

Hence proven!

(b) (i) As shown in part a) $\vec{\alpha}_{t+1} \leftarrow \begin{cases} \vec{\alpha}_t + y^{(i)} e^{(i)} & y^{(i)} h < 0 \\ \vec{\alpha}_t & \text{o/w} \end{cases}$

$\Rightarrow$ $\vec{\alpha}_{t+1}$ & $\vec{\alpha}_t$ differ by at max 1 element at position i.

(ii) $h \leftarrow \vec{\alpha}_t^T \Phi \phi(x^{(n)})$

$$h \leftarrow \vec{\alpha}_t^T \begin{bmatrix} \phi(x^{(1)})^T \phi(x^{(n)}) \\ \vdots \\ \phi(x^{(N)})^T \phi(x^{(n)}) \end{bmatrix} = \vec{\alpha}_t^T \begin{bmatrix} k(x^{(1)}, x^{(n)}) \\ \vdots \\ k(x^{(N)}, x^{(n)}) \end{bmatrix}$$

(c) **Algorithm: Kernel perceptron algorithm**

1. $\vec{\alpha}_0 \leftarrow \vec{0}$
2. for $t : 0$ to $T-1$ do:
3. Pick a random training example $(\vec{x}^{(n)}, y^{(n)})$ from $\mathcal{D}$ (with replacement)
4. $h \leftarrow \vec{\alpha}_t^T \begin{bmatrix} k(x^{(1)}, x^{(n)}) \\ \vdots \\ k(x^{(N)}, x^{(n)}) \end{bmatrix}$
5. if $y^{(n)} h < 0$ then
6. $\vec{\alpha}_{t+1} \leftarrow \vec{\alpha}_t + e^{(n)}$
7. end
8. end
9. return $\vec{\alpha}_T$

Given a new point x the classification now will be

$$\text{sign}\left(\vec{\alpha}_T^T \begin{bmatrix} k(x^{(1)}, x) \\ \vdots \\ k(x^{(N)}, x) \end{bmatrix}\right)$$

Problem 4

a) Given objective function:

$$\min_{\vec{w}, b, \xi} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

subject to: $y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) \geq 1 - \xi_i, \forall i = 1, \ldots, N$ (1)

$\xi_i \geq 0, \forall i = 1, \ldots, N$ (2)

(1) can be rewritten as: $\xi_i \geq 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b)$ (3)

(2) & (3) can be combined as $\xi_i \geq \max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b))$

Since the goal is minimize $E(\vec{w}, b)$, we can sub. this in

$$\Rightarrow \min_{\vec{w}, b} E(\vec{w}, b);\ E(\vec{w}, b) = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b))$$

b) i) $\nabla_{\vec{w}} E(\vec{w}, b) = \frac{1}{2}\times 2\vec{w} + C\sum_{i=1}^{N}\nabla_{\vec{w}}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b))$

$$\nabla_{\vec{w}}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b)) = \begin{cases} -y^{(i)}\vec{x}^{(i)} & \text{if } y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\Rightarrow \nabla_{\vec{w}}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b)) = -\mathbb{I}[y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1]\, y^{(i)}\vec{x}^{(i)}$$

$$\Rightarrow \nabla_{\vec{w}} E(\vec{w}, b) = \vec{w} - C\sum_{i=1}^{N}\mathbb{I}[y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1]\, y^{(i)}\vec{x}^{(i)}$$

ii) $\frac{\partial}{\partial b} E(\vec{w}, b) = 0 + C\sum_{i=1}^{N}\frac{\partial}{\partial b}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b))$

$$\frac{\partial}{\partial b}\max(0, 1 - y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b)) = \begin{cases} -y^{(i)} & \text{if } y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$= -\mathbb{I}[y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1]\, y^{(i)}$$

$$\Rightarrow \frac{\partial}{\partial b} E(\vec{w}, b) = -C\sum_{i=1}^{N}\mathbb{I}[y^{(i)}(\vec{w}^T\vec{x}^{(i)} + b) < 1]\, y^{(i)}$$

c) **Batch Gradient descent:**

Iter 5:	Iter 50:	Iter 100:	Iter 1000:	Iter 5000:	Iter 6000:
accuracy = 54.1667%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%
w: [[96.] [-36.64285714] [233.57142857] [88.28571429]]	w: [[-1.98076923] [-11.71153846] [25.35576923] [11.32692308]]	w: [[-1.99019608] [-4.81862745] [11.45098039] [5.74019608]]	w: [[-0.499501] [-0.3243513] [1.05538922] [1.28293413]]	w: [[-0.3517593] [-0.2779888] [0.88644542] [1.00329868]]	w: [[-0.33655448] [-0.28065645] [0.89411863] [0.98642119]]
b: -0.068928571	b: -0.28672539463	b: -0.2956852636	b: -0.31806328958	b: -0.3329031984	b: -0.33432381099

d) $\nabla_{\vec{w}} E^{(i)}(\vec{w}, b) = \frac{\vec{w}}{N} - C\,\mathbb{1}\left[y^{(i)}(\vec{w}^T \vec{x}^{(i)} + b) < 1\right] y^{(i)} \vec{x}^{(i)}$

$\frac{\partial}{\partial b} E^{(i)}(\vec{w}, b) = - C\,\mathbb{1}\left[y^{(i)}(\vec{w}^T \vec{x}^{(i)} + b) < 1\right] y^{(i)}$

e) **Stochastic Gradient descent:**

Iter 5:	Iter 50:	Iter 100:	Iter 1000:	Iter 5000:	Iter 6000:
accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%	accuracy = 95.8333%
w: [[-1.60513517] [-2.82975568] [7.75514067] [4.70009547]]	w: [[-1.68902612] [-0.17971377] [2.50267745] [2.78270712]]	w: [[-1.21320347] [0.08608695] [1.68120436] [2.20196636]]	w: [[-0.49457636] [-0.18894245] [0.95385434] [1.14885559]]	w: [[-0.42353581] [-0.2382758] [0.8887035] [1.06173562]]	w: [[-0.4428795] [-0.21702285] [0.90732014] [1.06339658]]
b: [-0.03916667]	b: [-0.07074783]	b: [-0.07740539]	b: [-0.10334756]	b: [-0.12178864]	b: [-0.12332915]

Problem 5

a) Error: 0.3750%

b) train50 error
Error: 5.0000%
Num of support vectors: 35

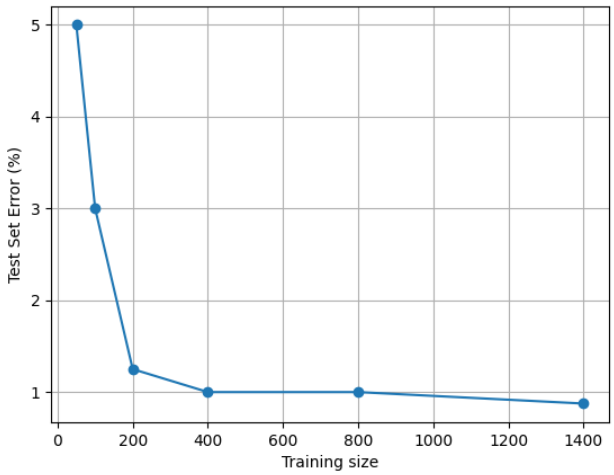
train100 error
Error: 3.0000%
Num of support vectors: 55

train200 error
Error: 1.2500%
Num of support vectors: 87

train400 error
Error: 1.0000%
Num of support vectors: 129

train800 error
Error: 1.0000%
Num of support vectors: 196

train1400 error
Error: 0.8750%
Num of support vectors: 234



c) The error for training size 50 & 100 seem to be larger for svm; however it is much smaller for the rest of the training sizes. We can say that the larger the test size got, the better the svm performed.