EECS 545 Machine Learning: Homework #4

Due on March 17, 2022 (2 days free late.) at 11:59pm

Professor Honglak Lee Section A

Yuang Huang
yahuang@umich.edu

Problem 1

Neural Network Layer Implementation
Solution

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w}), \tag{1}$$

$$\mathbf{w}_{\mathrm{MAP}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w}) \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w}). \tag{2}$$

So we will have:

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\operatorname{argmin}}\ -(\sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w})), \tag{3}$$

$$\mathbf{w}_{\mathrm{MAP}} = \underset{\mathbf{w}}{\operatorname{argmin}}\ -(\log p(\mathbf{w}) + \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w})). \tag{4}$$

Because the prior $\mathbf{w} \sim \mathcal{N}(0, \tau^2 I)$, the value of $p(\mathbf{w})$ will decrease monotonically with $||\mathbf{w}||$, and then we will know $\log(\mathbf{w}) \propto ||\mathbf{w}||_2$ for MAP. We assume that $||\mathbf{w_{MAP}}||_2 > ||\mathbf{w_{ML}}||_2$, then we get:

$$\begin{aligned} &-(\log p(\mathbf{w_{MAP}}) + \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{MAP}})) - (-(\log p(\mathbf{w_{ML}}) + \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{ML}}))) \\ &= \log \frac{p(\mathbf{w}_{ML})}{p(\mathbf{w}_{MAP})} - \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{ML}}) + \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{MAP}}) \end{aligned} \tag{5}$$

From (1) we know that $\sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{MAP}}) < \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w_{ML}})$, so the above equation is smaller than zero which contradicts to (4) because $\mathbf{w}_{\mathrm{ML}}$ for the term $''\underset{\mathbf{w}}{\operatorname{argmin}} - (\log p(\mathbf{w}) + \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x^{(i)}};\mathbf{w}))''$ has smaller value than $\mathbf{w}_{\mathrm{MAP}}$. So our assumption is wrong, it can be proved that:

$$||\mathbf{w}_{\mathrm{MAP}}|| \leq ||\mathbf{w}_{\mathrm{ML}}|| \tag{6}$$

Problem 2

Direct construction of valid kernels

Solution
Part a:
• Symmetric
Because $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ are kernels, $k_1(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{z}, \mathbf{x})$ and $k_2(\mathbf{x}, \mathbf{z}) = k_2(\mathbf{z}, \mathbf{x})$. Then it is obviously that $k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$ so the matrix K is symmetric.
• positive semi-definite
Because $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ are kernels, $x^T K_1 x \geq 0$ and $y^T K_2 y \geq 0$. It is obviously that $x^T K x = x^T K_1 x + x^T K_2 x \geq 0$, so the matrix K is positive semi-definite.

Part b:
It is not a kernal.
Counterexample: We assume that :

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} K_2 = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \tag{7}$$

Because both K_1 and K_2 are symmetric and positive semi-definite, both k_1 and k_2 are kernels. So the matrix K:

$$K = K_1 - K_2 = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}, \tag{8}$$

we choose a vector $x = [1 \quad 1]^T$, then we will get

$$x^T K x = \big(-4\big) < 0. \tag{9}$$

Thus, it is not a kernel.

Part c:
• Symmetric
Because $k_1(\mathbf{x}, \mathbf{z})$ is a kernel, $k_1(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{z}, \mathbf{x})$. Thus, $ak_1(\mathbf{x}, \mathbf{z}) = ak_1(\mathbf{z}, \mathbf{x})$. So the matrix K is symmetric.
• positive semi-definite
Because $k_1(\mathbf{x}, \mathbf{z})$ is a kernel, $x^T K_1 x \geq 0$. It is obviously that $x^T K x = a x^T K_1 x \geq 0$ where a is a positive real number, so the matrix K is positive semi-definite.

Part d:
It is not a kernal.
Counterexample: We assume that : $a = 1$

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} K = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \tag{10}$$

Problem 2 continued on next page...

we choose a vector $x = [1 \quad 1]^T$, then we will get

$$x^T K x = \big(-2\big) < 0. \tag{11}$$

Thus, it is not a kernel.

Part e:

• Symmetric

Because $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ are kernels, $k_1(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{z}, \mathbf{x})$ and $k_2(\mathbf{x}, \mathbf{z}) = k_2(\mathbf{z}, \mathbf{x})$. Then it is obviously that $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{z}, \mathbf{x})k_2(\mathbf{z}, \mathbf{x}) = k(\mathbf{z}, \mathbf{x})$, so the matrix K is symmetric.

• positive semi-definite

Because $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$ are kernels, $x^T K_1 x \geq 0$ and $y^T K_2 y \geq 0$. $K_{(1)ij} = \sum_k^D u_{ik}\lambda_k u_{kj}$ where $K_1 = \mathbf{U}^T \Lambda \mathbf{U}$. Thus,

$$\begin{aligned} x^T K x &= \sum_i^D \sum_j^D x_i K_{(1)ij} K_{(2)ij} x_j \\ &= \sum_i^D \sum_j^D \sum_k^D x_i u_{ij} \lambda_k u_{kj} K_{(2)ij} x_j \\ &= \sum_i^D \sum_j^D \sum_k^D \lambda_k x_i u_{ik} K_{(2)ij} u_{kj} x_j \end{aligned} \tag{12}$$

We know $\lambda \geq 0$ because K_1 is positive semi-definite. Thus, $\sum_i^D \sum_j^D \sum_k^D \lambda_k x_i u_{ik} K_{(2)ij} u_{kj} x_j = \sum_i^D \sum_j^D \sum_k^D \lambda_k t_{ik} K_{(2)ij} t_{kj} \geq 0$ because K_2 is positive semi-definite. So, the matrix K is positive semi-definite.

Part f:

• Symmetric

We can get $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z}) = f(\mathbf{z})f(\mathbf{x}) = k(\mathbf{z}, \mathbf{x})$. Thus, the matrix K is symmetric.

• positive semi-definite

Because $K_{ij} = f(\mathbf{x}_i)f(\mathbf{x}_j)$, then we will get:

$$\begin{aligned} \mathbf{y}^T K \mathbf{y} &= \sum_i^D \sum_j^D y_i K_{ij} y_j \\ &= \sum_i^D \sum_j^D y_i f(\mathbf{x}_i) f(\mathbf{x}_j) y_j \\ &= \sum_i^D [y_i f(\mathbf{x_i})]^2 + \sum_{i \neq j}^D 2 y_i f(\mathbf{x}_i) f(\mathbf{x}_j) y_j \\ &= \sum_i^D [y_i f(\mathbf{x}_i) + y_j f(\mathbf{x}_j)]^2 \geq 0. \end{aligned} \tag{13}$$

Thus, the matrix K is positive semi-definite.

Part g:

• Symmetric

Because k_3 is a kernel, $k(\mathbf{x},\mathbf{z}) = k_3(\phi(\mathbf{x}),\phi(\mathbf{z})) = k_3(\phi(\mathbf{z}),\phi(\mathbf{x})) = k(\mathbf{z},\mathbf{x})$. So the matrix K is symmetric.

• positive semi-definite

Because k_3 is a kernel, $x^T K_3 x \geq 0$.

$$\begin{aligned}
\mathbf{y}^T K\mathbf{y} &= \sum_i^D \sum_j^D y_i K_{ij} y_j \\
&= \sum_i^D \sum_j^D y_i k(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) y_j \\
&= \sum_i^D \sum_j^D y_i k_3(\phi(\mathbf{x}^{(i)}),\phi(\mathbf{x}^{(j)})) y_j \geq 0.
\end{aligned} \tag{14}$$

Thus, the matrix K is positive semi-definite.

Part h:

• Symmetric

Because k_1 is a kernel, $k_1(\mathbf{x},\mathbf{z}) = k_1(\mathbf{z},\mathbf{x}) \Rightarrow [k_1(\mathbf{x},\mathbf{z})]^n = [k_1(\mathbf{z},\mathbf{x})]^n \Rightarrow a_n[k_1(\mathbf{x},\mathbf{z})]^n = a_n[k_1(\mathbf{z},\mathbf{x})]^n \Rightarrow k(\mathbf{x},\mathbf{z}) = \sum_1^N a_n[k_1(\mathbf{x},\mathbf{z})]^n = \sum_1^N a_n[k_1(\mathbf{z},\mathbf{x})]^n = k(\mathbf{z},\mathbf{x})$. So the matrix K is symmetric.

• positive semi-definite

From (e), we know $k(\mathbf{x},\mathbf{z}) = k_1(\mathbf{x},\mathbf{z})k_2(\mathbf{x},\mathbf{z})$ is a kernel. Let $k_2(\mathbf{x},\mathbf{z}) = k_1(\mathbf{x},\mathbf{z})$, then we will get $k_1(\mathbf{x},\mathbf{z})k_1(\mathbf{x},\mathbf{z}) = [k_1(\mathbf{x},\mathbf{z})]^2$ is a kernel. Also, we can get that $k_1(\mathbf{x},\mathbf{z})[k_1(\mathbf{x},\mathbf{z})]^2 = [k_1(\mathbf{x},\mathbf{z})]^3$ is a kernel. Thus, we will know that $[k_1(\mathbf{x},\mathbf{z})]^n$ is always a kernel where n is an positive integer number. In addition, we have $\mathbf{y}^T a_n[k_1(\mathbf{x},\mathbf{z})]^n\mathbf{y} \geq 0$ where a_n is a positive coefficeint. So, $\mathbf{y}^T K\mathbf{y} = \mathbf{y}^T \sum_1^N a_n[k_1(\mathbf{x},\mathbf{z})]^n\mathbf{y} = \sum_1^N \mathbf{y}^T a_n[k_1(\mathbf{x},\mathbf{z})]^n\mathbf{y} \geq 0$. Thus, the matrix K is positive semi-definite.

Part i:

$k(\mathbf{x},\mathbf{z}) = (\mathbf{x}^T\mathbf{z}+1)^2$. We assume $\mathbf{D} = 2$, then $\mathbf{x} = [x_1 \quad x_2]^T$, $\mathbf{z} = [z_1 \quad z_2]^T$. Thus, $k(\mathbf{x},\mathbf{z}) = (x_1z_1 + x_2z_2 + 1)^2 = x_1^2z_1^2 + x_2^2z_2^2 + 2x_1z_1 + 2x_2z_2 + 2x_1x_2z_1z_2 + 1 = \phi(\mathbf{x})^T\phi(\mathbf{z})$, so $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$.

Part j:

$$\begin{aligned} k(\mathbf{x},\mathbf{z}) &= \exp(\frac{-||\mathbf{x}-\mathbf{z}||^2}{2\sigma^2}) \\ &= \exp(\frac{-\mathbf{x}^T\mathbf{x}-\mathbf{z}^T\mathbf{z}+2\mathbf{x}^T\mathbf{z}}{2\sigma^2}) \\ &= \exp(\frac{-\mathbf{x}^T\mathbf{x}}{2\sigma^2})\times\exp(\frac{-\mathbf{z}^T\mathbf{z}}{2\sigma^2})\times\exp(\frac{\mathbf{x}^T\mathbf{z}}{\sigma^2}). \end{aligned} \tag{15}$$

Also, we know $\mathbf{x}^T\mathbf{z} = \sum_{k=1}^{\infty} x_k z_k$, so the third term $\exp(\frac{\mathbf{x}^T\mathbf{z}}{\sigma^2})$ can be expressed by:

$$\begin{aligned} &\exp(\frac{\mathbf{x}^T\mathbf{z}}{\sigma^2}) = \exp(\frac{\sum_{k=1}^{\infty} x_k z_k}{\sigma^2}) \\ &= \sum_{k=1}^{\infty}\sum_{n=0}^{\infty}\frac{(\frac{\sum_{k=1}^{\infty} x_k z_k}{\sigma^2})^n}{n!} \\ &= \sum_{n=0}^{\infty}\sum_{\sum t_D=n}\frac{C_n^{t_1}C_{n-t_1}^{t_2}C_{n-t_1-t_2}^{t_3}\cdots C_{t_D}^{t_D}(\frac{x_1z_1}{\sigma^2})^{t_1}(\frac{x_2z_2}{\sigma^2})^{t_2}\cdots(\frac{x_Dz_D}{\sigma^2})^{t_D}}{n!} \\ &= \sum_{n=0}^{\infty}\sum_{\sum t_D=n}\frac{\frac{n!}{t_1!(n-t_1)!}\frac{(n-t_1)!}{t_2!(n-t_1-t_2)!}\cdots\frac{(n-t_1-\cdots t_{D-2})!}{t_{D-1}!(n-t_1-\cdots t_{D-1})!}(\frac{x_1z_1}{\sigma^2})^{t_1}(\frac{x_2z_2}{\sigma^2})^{t_2}\cdots(\frac{x_Dz_D}{\sigma^2})^{t_D}}{n!} \\ &= \sum_{n=0}^{\infty}\sum_{\sum t_D=n}\frac{1}{t_1!\cancel{(n-t_1)!}}\frac{\cancel{(n-t_1)!}}{t_2!\cancel{(n-t_1-t_2)!}}\cdots\frac{\cancel{(n-t_1-\cdots t_{D-2})!}}{t_{D-1}!t_D!}(\frac{x_1z_1}{\sigma^2})^{t_1}(\frac{x_2z_2}{\sigma^2})^{t_2}\cdots(\frac{x_Dz_D}{\sigma^2})^{t_D} \\ &= \sum_{n=0}^{\infty}\sum_{\sum t_D=n}\frac{1}{t_1!}\frac{1}{t_2!}\cdots\frac{1}{t_D!}(\frac{x_1z_1}{\sigma^2})^{t_1}(\frac{x_2z_2}{\sigma^2})^{t_2}\cdots(\frac{x_Dz_D}{\sigma^2})^{t_D} \\ &= \sum_{n=0}^{\infty}\sum_{\sum t_D=n}\frac{(\frac{x_1}{\sigma})^{t_1}(\frac{x_2}{\sigma})^{t_2}\cdots(\frac{x_D}{\sigma})^{t_D}}{\sqrt{t_1!}\sqrt{t_2!}\cdots\sqrt{t_D!}}\frac{(\frac{z_1}{\sigma})^{t_1}(\frac{z_2}{\sigma})^{t_2}\cdots(\frac{z_D}{\sigma})^{t_D}}{\sqrt{t_1!}\sqrt{t_2!}\cdots\sqrt{t_D!}} \end{aligned} \tag{16}$$

Thus, the closed form of an infinite dimensional feature vector ϕ can be written by:

$$\phi(\mathbf{x}) = (\exp(\frac{-\mathbf{x}^T\mathbf{x}}{2\sigma^2}), \frac{{x_1}^{t_1}{x_2}^{t_2}\cdots{x_D}^{t_D}}{\sqrt{t_1!}\sqrt{t_2!}\cdots\sqrt{t_D!}\sigma^n} for\,all\,possible\,terms, \cdots) \tag{17}$$

Problem 3

Kernelizing the Perceptron

Solution

Part a(i):

• If $y^{(n)}h > 0$, then $\mathbf{w}_{t+1} = \mathbf{w}_t = \Phi^T \alpha_t$, and we have $\alpha_{t+1} = \alpha_t$

• If $y^{(n)}h < 0$, then $\mathbf{w}_{t+1} = \mathbf{w}_t + y^{(n)}\mathbf{x}^{(n)} = \Phi^T \alpha_t + y^{(n)}\phi(\mathbf{x}^{(n)}) = \Phi^T \alpha_t + \Phi^T \alpha_+$, where

$$\alpha_{+i} = \begin{cases} y^{(n)}, & i = n \\ 0, & i \neq n \end{cases} \tag{18}$$

Thus, $\mathbf{w}_{t+1} = \Phi^T(\alpha_+ + \alpha_t) = \Phi^T \alpha_{t+1}$, where $\alpha_{t+1} = \alpha_t + \alpha_+$

Part a(ii):

Assume $\mathbf{w}_0 = 0 = \Phi^T \times 0$,
then we will have $\mathbf{w}_1 = 0 + \Phi^T \alpha_+ = \Phi^T \times \alpha_1$,
then we will have $\mathbf{w}_2 = \mathbf{w_1} + \Phi^T \alpha_+ = \Phi^T \times \alpha_{\mathbf{1}} + \Phi^T \times \alpha_+ = \Phi^T \times \alpha_2$,
$\cdots$
$\mathbf{w}_t = \Phi^T \times \alpha_t$ and $\mathbf{w}_{t+1} = \Phi^T \times (\alpha_t + \alpha_+) = \Phi^T \times \alpha_{t+1}$ from a(i).
Thus, for $0 \leq t \leq T$, $\mathbf{w}_t$ can be expressed as $\Phi^T \alpha_t$.

Part b(i):

• If $y^{(n)}h > 0$, then $\mathbf{w}_{t+1} = \mathbf{w}_t = \Phi^T \alpha_t$, and we have $\alpha_{t+1} = \alpha_t$, there is no different element.

• If $y^{(n)}h < 0$, then $\mathbf{w}_{t+1} = \mathbf{w}_t + y^{(n)}\mathbf{x}^{(n)} = \Phi^T \alpha_t + y^{(n)}\phi(\mathbf{x}^{(n)}) = \Phi^T \alpha_t + \Phi^T \alpha_+ = \Phi^T \alpha_{t+1}$.

From (18), we know at most 1 element is different.

Part b(ii):

$$\begin{aligned} h(\phi(x^{(n)}, \mathbf{w}_t)) &= \mathbf{w}^T \phi(x^{(n)}) \\ &= (\Phi^T \alpha_t)^T \phi(x^{(n)}) \\ &= \alpha_t^T \Phi \phi(x^{(n)}) \\ &= \alpha_t^T \mathbf{k} \end{aligned} \tag{19}$$

where

$$\begin{aligned} \mathbf{k} &= \Phi\phi(x^{(n)}) \\ \text{and} \quad \mathrm{k_i} &= \phi(x^{(i)})^T \phi(x^{(n)}) = k(x^{(i)}, x^{(n)}) \end{aligned} \tag{20}$$

Problem 3 continued on next page...

Part c:

initial $a_0 \leftarrow 0$;
repeat(from $t = 0, t++$)
    Pick a random training example $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ from $\mathcal{D}$;
    $h \leftarrow \alpha_t^T \mathbf{k}$;
    if $y^{(n)} h < 0$ **then**
        $\alpha_{t+1} = \alpha_t + \alpha_+$;
    end
until $(t \geq T)$
return α_T

Algorithm 1: Kernelized version

First, calculate $h(\mathbf{x}; \mathbf{w}) = \alpha^T \mathbf{k}$, where α^T is trained, and $\mathbf{k}_i = \phi(x^{(i)})^T \phi(x)$ is calculated from (20). Then,

- If $h \geq 0$, $y = 1$
- Else $y = -1$

Problem 4

Implementing Soft Margin SVM by Optimizing Primal Objective

Solution
Part a:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ \textit{subject to} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \forall_i = 1, \dots, N \\ & \xi_i \geq 0, \forall_i = 1, \dots, N \end{aligned} \tag{21}$$

.
We can easily know:

$$\begin{aligned} & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \forall_i = 1, \dots, N \\ \Rightarrow & \xi_i \geq 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b), \forall_i = 1, \dots, N \end{aligned} \tag{22}$$

.
With the second requirement $\xi_i \geq 0, \forall_i = 1, \dots, N$, we can get the combined requirement:

$$\xi_i = \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) \tag{23}$$

Thus, the constrained minimization is equivalent to following minimization involving the hinge loss term:

$$\min_{\mathbf{w},b} E(\mathbf{w}, b), \quad E(\mathbf{w}, b) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) \tag{24}$$

Part b:
Because $f(x) = \max(0, x)$, then:

$$\begin{aligned} \nabla_{\mathbf{w}} \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) &= -\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)}\mathbf{x}^{(i)} \\ \Rightarrow \nabla_{\mathbf{w}} E(\mathbf{w}, b) &= \mathbf{w} - C\sum_{i=1}^{N}\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)}\mathbf{x}^{(i)} \\ \frac{\partial}{\partial b}\max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) &= -\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)} \\ \Rightarrow \frac{\partial}{\partial b}E(\mathbf{w}, b) &= -C\sum_{i=1}^{N}\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)} \end{aligned} \tag{25}$$

Problem 4 continued on next page...

Part c:

• NumIterations = 5:
$\mathbf{w} = [112. \quad -42.75 \quad 272.5 \quad 103.]^T$
$b = -0.12416667$ accuracy = 54.1667%

• NumIterations = 50:
$\mathbf{w} = [-2.01960784 \quad -11.94117647 \quad 25.85294118 \quad 11.54901961]^T$
$b = -0.37280358$ accuracy = 95.8333%

• NumIterations = 100:
$\mathbf{w} = [-2.55940594 \quad -5.28217822 \quad 11.37623762 \quad 5.75742574]^T$
$b = -0.38285$ accuracy = 95.8333%

• NumIterations = 1000:
$\mathbf{w} = [-0.46353646 \quad -0.32617383 \quad 1.05394605 \quad 1.27872128]^T$
$b = -0.40401205$ accuracy = 95.8333%

• NumIterations = 5000:
$\mathbf{w} = [-0.32083583 \quad -0.27904419 \quad 0.89262148 \quad 0.98660268]^T$
$b = -0.4184513$ accuracy = 95.8333%

• NumIterations = 6000:
$\mathbf{w} = [-0.32919513 \quad -0.28186969 \quad 0.886019 \quad 0.97483753]^T$
$b = -0.4199084$ accuracy = 95.8333%

Part d:

$$\begin{aligned}
\nabla_{\mathbf{w}} \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) &= -\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)}\mathbf{x}^{(i)} \\
\Rightarrow \nabla_{\mathbf{w}} E^{(i)}(\mathbf{w}, b) &= \frac{\mathbf{w}}{N} - C\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)}\mathbf{x}^{(i)} \\
\frac{\partial}{\partial b} \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)) &= -\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)} \\
\Rightarrow \frac{\partial}{\partial b} E^{(i)}(\mathbf{w}, b) &= -C\mathbf{I}[y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) < 1]y^{(i)}
\end{aligned} \tag{26}$$

Problem 4 continued on next page...

Part e:

- NumIterations = 5:

$\mathbf{w} = [-1.78136842 \quad -3.12818738 \quad 8.55400016 \quad 5.20287663]^T$
$b = -0.05416667$ accuracy = 95.8333%

- NumIterations = 50:

$\mathbf{w} = [-1.37946899e+00 \quad 9.07974830e-04 \quad 2.58689377e+00 \quad 2.85570760e+00]^T$
$b = -0.08671111$ accuracy = 95.8333%

- NumIterations = 100:

$\mathbf{w} = [-1.25745166 \quad 0.11439094 \quad 1.70851556 \quad 2.31719145]^T$
$b = -0.09433571$ accuracy = 95.8333%

- NumIterations = 1000:

$\mathbf{w} = [-0.48895966 \quad -0.18986655 \quad 0.95735748 \quad 1.14001054]^T$
$b = -0.12014856$ accuracy = 95.8333%

- NumIterations = 5000:

$\mathbf{w} = [-0.42761221 \quad -0.23477963 \quad 0.88908395 \quad 1.06544336]^T$
$b = -0.13850557$ accuracy = 95.8333%

- NumIterations = 6000:

$\mathbf{w} = [-0.44211714 \quad -0.21435765 \quad 0.90972215 \quad 1.06365376]^T$
$b = -0.14003648$ accuracy = 95.8333%

Problem 5

SciKit-Learn SVM for classifying SPAM

Solution
Part a:
Error: 0.375%

Part b:
- Size of training set = 50:
accuracy = 5.0000% Number of support vectors: 35
- Size of training set = 100:
accuracy = 3.0000% Number of support vectors: 55
- Size of training set = 200:
accuracy = 1.2500% Number of support vectors: 87
- Size of training set = 400:
accuracy = 1.0000% Number of support vectors: 128
- Size of training set = 800:
accuracy = 1.0000% Number of support vectors: 197
- Size of training set = 1400:
accuracy = 0.8750% Number of support vectors: 234

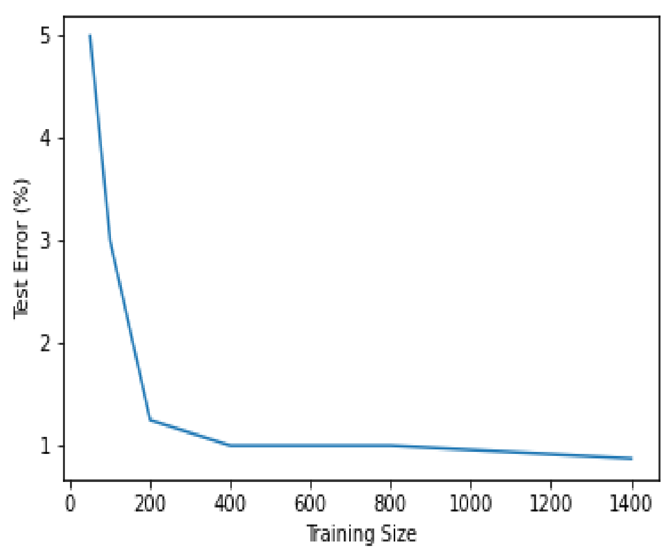

Figure 1: Test error with the size of training set

obviously, when the size of training set is equal to 1400, the error is the smallest, 0.87500%. Thus, size1400 gives the best set error.

Part c:

• Size of training set = 50:
accuracy of naive bayes = 3.8750% accuracy of SVM= 5.0000%
• Size of training set = 100:
accuracy of naive bayes = 2.6250% accuracy of SVM= 3.0000%
• Size of training set = 200:
accuracy of naive bayes = 2.6250% accuracy of SVM= 1.2500%
• Size of training set = 400:
accuracy of naive bayes= 1.8750% accuracy of SVM= 1.0000%
• Size of training set = 800:
accuracy of naive bayes= 1.7500% accuracy of SVM= 1.0000%
• Size of training set = 1400:
accuracy of naive bayes= 1.6500% accuracy of SVM= 0.8750%

Naive Bayes has better performance for smaller training sets and SVM has better performance for larger training sets.