HW 4

Class ID-19 Name: Sayeed Shafayet Chowdhury

Problem ~~4~~ 1(a)(i)

The problem asks us to show that in linearly separable case, the logistic regression solution makes w_0 and the weights $\bar{w} \to \infty$. The logistic function is defined as $\sigma(z) = \frac{1}{1+\exp\{-z\}}$

Now given the dataset, $\{(x_i, y_i)_{i=1}^{N}\}$, where $x_i \in \mathbb{R}^d$, $y_i \in \{0,1\}$, $\bar{w} \in \mathbb{R}^d$

the likelihood function becomes,

$$P(\{y_n\}_{n=1}^{N} \mid \{x_n\}_{n=1}^{N}, \bar{w}) = \prod_{n=1}^{N} \sigma(\bar{w}^T x_n)^{y_n} (1-\sigma(\bar{w}^T x_n))^{1-y_n}$$

This is the function we want to maximize with respect to $\bar{w}$. So,

$$\bar{w}^* = \underset{\bar{w}}{\arg\max} \left\{ \sum_{n=1}^{N} y_n \log \sigma(\bar{w}^T x_n) + (1-y_n) \log (1-\sigma(\bar{w}^T x_n)) \right\} \quad \text{(1)}$$

This is the same as minimizing the logistic regression loss function, $J = \sum_{n=1}^{N} -\{y_n \log \sigma(\bar{w}^T x_n) + (1-y_n) \log(1-\sigma(\bar{w}^T x_n))\}$

So, we will focus on solving (1), but it is not possible to solve (1) directly by setting the gradient to 0 and solving for $\bar{w}$. So, we'll solve for (1) using gradient ascent.

Now, let us examine the gradient arising from a single point of data. $\nabla_{\bar{w}} \{ y_n \log \sigma(\bar{w}^T x_n) + (1-y_n) \log(1-\sigma(\bar{w}^T x_n)) \}$

Here we note that, $1-\sigma(z) = 1 - \frac{1}{1+e^{-z}} = \frac{1+e^{-z}-1}{1+e^{-z}}$

$$= \frac{e^{-z}}{1+e^{-z}}$$

$$= \frac{1}{1+e^{z}}$$

$$= \sigma(-z) \text{ —— (A)}$$

and $\frac{d}{dz}\sigma(z) = (1-\sigma(z))\,\sigma(z)$ —— (B)

let us have a look at the gradient -

$$\nabla_{\bar{w}} \sum_{n=1}^{N} \{ y_n \log \sigma(\bar{w}^T x_n) + (1-y_n) \log(1-\sigma(\bar{w}^T x_n)) \}$$

$$= \sum_{n=1}^{N} \left[y_n \frac{1}{\sigma(\bar{w}^T x_n)} \nabla_{\bar{w}} \{\sigma(\bar{w}^T x_n)\} + (1-y_n) \frac{1}{\sigma(-\bar{w}^T x_n)} \nabla_{\bar{w}} \{\sigma(-\bar{w}^T x_n)\} \right]$$

$[\because 1-\sigma(\bar{w}^T x_n) = \sigma(-\bar{w}^T x_n)]$

$$= \sum_{n=1}^{N} \left[y_n x_n (1-\sigma(\bar{w}^T x_n)) - (1-y_n) x_n \sigma(\bar{w}^T x_n) \right]; \text{ [using (B)]}$$

$[\nabla_{\bar{w}} \{\sigma(-\bar{w}^T x_n) = -x_n \sigma(-\bar{w}^T x_n)(1-\sigma(-\bar{w}^T x_n)]$

Now, $1-\sigma(-\bar{w}^T x_n) = \sigma(\bar{w}^T x_n)$

$$= \sum_{n=1}^{N} \left[y_n x_n - y_n x_n \sigma(\bar{w}^T x_n) - x_n \sigma(\bar{w}^T x_n) + x_n y_n \sigma(\bar{w}^T x_n) \right]$$

$$= \sum_{n=1}^{N} \left[y_n x_n - x_n \sigma(\bar{w}^T x_n) \right] = \sum_{n=1}^{N} \left[x_n \left(y_n - \sigma(\bar{w}^T x_n) \right) \right]$$

Then using the gradient ascent update, we get -

$$\bar{w}^{(t+1)} \leftarrow \bar{w}^{(t)} + \alpha \left[\sum_{n=1}^{N} \left(x_n \left(y_n - \sigma\left((\bar{w}^{(t)})^T x_n \right) \right) \right) \right] \quad (2)$$

Now, let us consider the linear separability condition, it creates a problem for the unregularized logistic regression problem (the case considered here). We note that, decision boundary in a linear classifier is independent of the scale of the parameters. This can be seen as the decision boundary is the set $\{\bar{x} : \bar{w}^T \bar{x} = 0\}$ and this set doesn't change if $\bar{w}$ is multiplied by some constant. For a given decision boundary, however, the scale does affect the likelihood in logistic regression by causing the logistic function to become more steep. We can see this by thinking about the derivative of $\sigma(z)$ evaluated at $z=0$ versus the derivative of $\sigma(10z)$ at $z=0$. The derivatives are $\sigma(z)(1-\sigma(z))$ and $10\sigma(z)(1-\sigma(z))$, respectively, and so the linear regime in the middle is ten times steeper when input is scaled by a factor of ten.

If we currently have a decision boundary such that the data are all correctly classified, then increasing the scale of the weights pushes the predictions further towards their correct answers. Say, we have a set of weights $\hat{w}$ with unit norm, i.e. $\|\hat{w}\|=1$. Then we construct a logistic regression classifier with weights $\bar{w}=c\hat{w}$ and seek only to fit the constant $c>0$ to the data. Here, changing c does not move the decision boundary for the classifier. Let take the n^{th} example and examine the derivative of its log likelihood with respect to c:

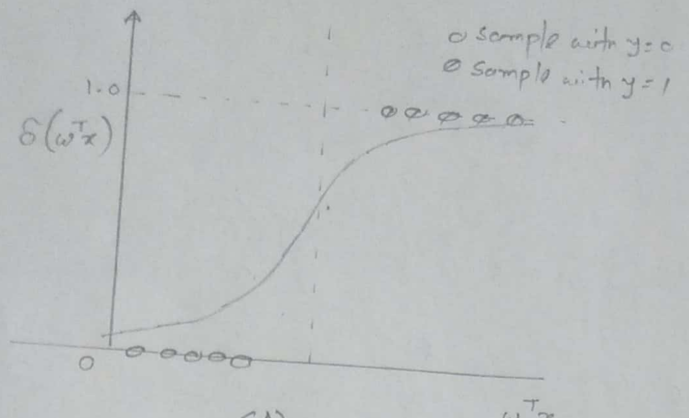
$$\frac{\partial}{\partial c}\left\{y_n \log \sigma(c\hat{w}^T x_n) + (1-y_n)\log(1-\sigma(c\hat{w}^T x_n))\right\}$$
$$= \hat{w}^T x_n \left(y_n - \sigma(c\hat{w}^T x_n)\right)$$

If $y_n=0$ then $(y_n - \sigma(c\hat{w}^T x_n)) < 0$ and if $y_n=1$ then $(y_n - \sigma(c\hat{w}^T x_n)) > 0$. We also note that due to the fixed decision boundary, if $y_n = 0$ is classified correctly then $\hat{w}^T x_n < 0$ and is positive otherwise. Similarly if $y_n = 1$ is classified correctly, then $\hat{w}^T x_n > 0$ and is negative otherwise. Thus the derivative of the log likelihood w.r.t. c is always positive for an example that $\hat{w}$ classifies correctly. Since in our case, the data are linearly separable, there exists a $\hat{w}$ such that all of the data have log likelihoods with positive derivaties with respect to c. In that situation, gradient ascent on c would cause it to grow without bound. This essentially drives the sigmoid function to be sharper and sharper until it becomes a Heaviside step function. So, to predict all the samples correctly with certainty, the gradient ascent on maximizing

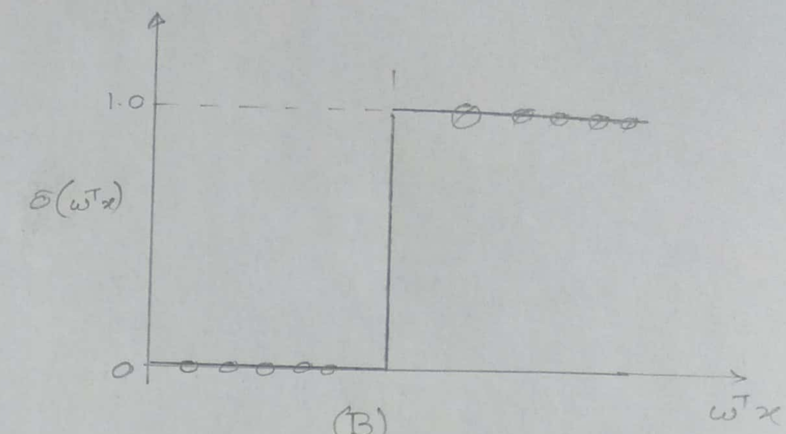
(LCL)

the log conditional likelihood sends the w to infinity.
Again, this can be seen as, for weight parameters w_j, $\frac{\partial}{\partial w_j} LCL = \sum_i (y_i - p_i) x_{ij}$, here we consider only positive samples ($y_i = 1$), x_{ij} is the j^{th} feature of i^{th} sample. Now this derivative will always be positive, as long as the predicted probability p_i is not perfectly one for all the positive samples. So, the logistic loss keeps on decreasing with iteration as w keeps increasing and eventually w goes towards infinity.

This fact can also be viewed as — in order to maximize the likelihood, we need to minimize $\log(1+e^{-y w^T x})$, so we want the exponent to be as negative as possible.
Now, to make the exponent $y \cdot \|w\|_2 \cdot \|x_2\|_2 \cos\theta$ as large as possible, the algorithm simply increases $\|w\|_2$ without bound.



(A)



(B)

We can also visualize this graphically. Although in (A), we can correctly classify the samples, still samples with $y=0$ have non-zero probability, while samples with $y=1$ have $p<1$. So the algorithm while trying to minimize the logistic loss drives w towards a position where all x_i's with $y_i=0$ have $p_i=0$ and all x_i's with $y_i=1$ have $p_i=1$ (as shown in (B)). This essentially drives the $w \to \infty$, as the slope of the sigmoid $\to \infty$ at the middle. Since with $w \to \infty$, $h(x) = \frac{1}{1+e^{-w^T x}}$, $\lim_{w\to\infty} h(x) = \text{sign}(w^T x)$

so $h(x)$ becomes a step function and to make it a step function, $w \to \infty$.

Now, from (2), $\bar{w}^{(k+1)} \leftarrow \bar{w}^{(k)} + \alpha\left[\sum_{n=1}^{N} \bar{x}_n \left(y_n - \sigma\left(\bar{w}^{T(k)} x_n\right)\right)\right]$

if we allowed the algo to run forever (only let it stop when $\|\bar{w}^{(k+1)} - \bar{w}^{(k)}\|_2 = 0$), the algo would not stop until the gradient $= 0$, meaning $y_n = \sigma\left(\bar{w}^{T(k)} x_n\right)$ for all n.

Now, say $y_n = 1$, then $\sigma\left(\bar{w}^{T(k)} x_n\right)$ must be 1 for algo to stop but $\sigma\left(\bar{w}^{T(k)} x_n\right) = \dfrac{1}{1+e^{-\bar{w}^{T(k)} x_n}}$, to make it exactly $=1$, $e^{-\bar{w}^{T(k)} x_n}$ must be $= 0 \Rightarrow \bar{w}^{(k)}$ must be $\rightarrow \infty$. So, the algo would keep chasing this criterion. However to prove that it would not converge in a finite no. of steps, meaning the $\nabla_{\bar{w}} J$ will not become 0, we need to investigate the convexity of J.

$$J(\theta) = -\sum_{n=1}^{N} \left\{ y_n \log h_\theta(x_n) + (1-y_n)\log(1-h_\theta(x_n)) \right\}$$

where $h_\theta(x)$ is the logistic function defined as,

$$h_\theta(x_n) = \frac{1}{1+e^{-\theta^T x_n}}$$

Now, $J(\theta)$ can be written as,

$$J(\theta) = \sum_{n=1}^{N} -\left\{ y_n \log\left(\frac{h_\theta(x_n)}{1-h_\theta(x_n)}\right) + \log(1-h_\theta(x_n)) \right\}$$

Now, $\log\left(\frac{h_\theta(x_n)}{1-h_\theta(x_n)}\right) = \log\left(\frac{\frac{1}{1+e^{-\theta^T x_n}}}{\frac{e^{-\theta^T x_n}}{1+e^{-\theta^T x_n}}}\right) = \log(e^{\theta^T x_n})$

$$= \theta^T x_n$$

So, $$J(\theta) = \sum_{n=1}^{N} -\left\{ y_n \theta^T x_n + \log(1-h_\theta(x_n)) \right\} \quad \text{—— (1)}$$

the 1st term is linear in θ, so it is convex

Scanned with CamScanner

let us take gradient of the 2^{nd} term.

$$\nabla_\theta\left[-\log(1-h_\theta(x))\right] = -\nabla_\theta\left[\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)\right]$$

$$= -\nabla_\theta\left[\log\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}\right]$$

$$= -\nabla_\theta\left[\log e^{-\theta^T x} - \log\left(1+e^{-\theta^T x}\right)\right]$$

$$= -\nabla_\theta\left[-\theta^T x - \log\left(1+e^{-\theta^T x}\right)\right]$$

$$= x + \nabla_\theta \left[\log\left(1+e^{-\theta^T x}\right)\right]$$

$$= x + \left(\frac{-e^{-\theta^T x}}{1+e^{-\theta^T x}}\right)x = x\left(1-\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}\right)$$

$$= h_\theta(x)\, x$$

the 1^{st} term of 1 was $-y_n\theta^T x_n$

gradient of 1^{st} term is $\nabla_\theta\left(-y_n\theta^T x_n\right) = -y_n x_n$

so, $$\nabla_\theta J(\theta) = \sum_{n=1}^{N}\left\{-y_n x_n + h_\theta(x_n)\, x_n\right\}$$

Proof that $J(\Theta)$ is convex

From ①, $J(\Theta) = \sum_{n=}^{N} - \left\{ y_n \theta^T x_n + \log \left(1 - h_\theta(x_n)\right)\right\}$

the 1st term is linear in θ, so it is convex.

We have shown above that $\nabla_\theta \left(-\log\left(1 - h_\theta(x)\right)\right) = h_\theta(x)\, x$

the hessian of this 2nd term is,

$$\nabla_\theta^2 \left[-\log\left(1 - h_\theta(x)\right)\right] = \nabla_\theta \left[h_\theta(x)\, x\right]$$

$$= \nabla_\theta \left[\left(\frac{1}{1+e^{-\theta^T x}}\right) x\right]$$

$$= \frac{1}{\left(1+e^{-\theta^T x}\right)^2} \left(-e^{-\theta^T x}\right) x x^T$$

$$= \left(\frac{1}{1+e^{-\theta^T x}}\right)\left(1 - \frac{1}{1+e^{-\theta^T x}}\right) x x^T$$

$$= h_\theta(x)\left[1 - h_\theta(x)\right] x x^T$$

Now, for any $v \in \mathbb{R}^d$, we have,

$$v^T \nabla_\theta^2 \left[-\log\left(1 - h_\theta(x)\right)\right] v = v^T \left[h_\theta(x)\left[1 - h_\theta(x)\right] x x^T\right] v$$

$$= h_\theta(x)\left[1 - h_\theta(x)\right] \|v^T x\|^2$$

we have $h_\theta(x) \in [0,1]$, so $[1 - h_\theta(x)] \in [0,1]$, so

$$v^T \nabla_\theta^2 [-\log(1-P_\theta(x))] v = P_\theta(x)[1-P_\theta(x)] \|v^T x\|^2 \geq 0$$

So the Hessian of $-\log(1-P_\theta(x))$ is positive semi-definite, so $-\log(1-P_\theta(x))$ is convex in θ.

So, both the 1st and 2nd terms in $J(\theta)$ are convex, so, $J(\theta)$ is convex in θ. Since $J(\theta)$ is convex, it would not get stuck at some other critical point (i.e. saddle point), rather there is a global minima with $\bar{w}$ of infinite magnitude and the algo would keep chasing this global minima. Again, with proper α, the algo would not converge. So, gradient descent would just push the loss down after every descent step. Since if $y_n = 1$, the gradient is +ve, so $\bar{w}^{(k+1)}$ is $> \bar{w}^{(k)}$ from ②. So, $\sigma(\bar{w}^{T(k+1)} x_n)$ gets closer to 1 compared to $\sigma(\bar{w}^{T(k)} x_n)$. So, the loss keeps going down with every descent step toward the unique global minima, without worrying about the descent being stuck at some finite critical point. So, the algo would not converge in finite no. of steps. (Proved)

Scanned with CamScanner

(ii) If we restrict $\|\bar{w}\|_2 \leq c_1$ and $|w_0| < c_2$ for some $c_1, c_2 > 0$, then the algo can not keep chasing the $\bar{w} \to \infty$, rather it is bounded by $\|\bar{w}\|_2 \leq c_1$. So it is like a forced convergence condition.

This non-convergence issue can be dealt with using L^2 regularization. The gradient of the resulting objective is-

$$\nabla_{\bar{w}} \left\{ \sum_{n=1}^{N} y_n \log \sigma(\bar{w}^T x_n) + (1-y_n) \log (1 - \sigma(\bar{w}^T x_n)) - \frac{\lambda}{2} \bar{w}^T \bar{w} \right\}$$

$$= \sum_{n=1}^{N} x_n (y_n - \sigma(\bar{w}^T x_n)) - \lambda w$$

We can adjust λ, scale λ down by a factor of N. So the rule is:

$$\bar{w}^{(k+1)} \leftarrow \bar{w}^{(k)} + \alpha \left(x_n (y_n - \sigma((\bar{w}^{T(k)} x_n) - \frac{\lambda}{N} \bar{w}^{(k)} \right)$$

which can be rewritten as:

$$\bar{w}^{(k+1)} \leftarrow \left(1 - \alpha \frac{\lambda}{N}\right) \bar{w}^{(k)} + \alpha x_n (y_n - \sigma((\bar{w}^{(k)})^T x_n))$$

This is why L^2 is often referred as "weight decay."

(iii) This linear separability does not cause nonconvergence for other linear classifiers that we have studied. Since for linear regression classifier, if data is linearly separable, the algorithm tries to find the line with

$\tilde{w}^T x + w_0$ such that all data are separated perfectly. but once that goal is achieved, it stops. there is no objective to chase after once the perfect classification is achieved. So nonconvergence is not a problem. Similarly, for perceptron, the goal is to just classify all the samples perfectly. once it is achieved, the algorithm stops. Again, for the SVM, since it is a separable case, with the maximum possible margin. SVM tries to find the decision boundary and once it is able to find it stops.

(b)(i) Since x_j is misclassified by $w^{(k)}$,

$$y_j w^{T(k)} x_j < 0$$

Now, $y_j (w^{(k+1)})^T x_j = y_j (w^{(k)} + y_j x_j)^T x_j$

$$= y_j (w^{(k)})^T x_j + \|y_j x_j\|_2^2$$

So, $y_j (w^{k+1})^T x_j > y_j (\omega^{(k)})^T x_j$ —— (A)

Since $y_j w^{T(k)} x_j < 0$ but we want to make $y_j (w^{(k)})^T x_j > 0$, So, we need to increase $y_j (w^i)^T x_j$, as i increases. So, according to (A), the move from $w^{(k)}$ to $w^{(k+1)}$ is in right direction.

(Showed)

(ii) 1. Here, $\rho = \min_j y_j (w^*)^T x_j$. Since every x_j is correctly classified by w^*, we have for all $n = 1$ to N, $y_n (w^*)^T x_n > 0$

So, $\rho > 0$, now, $(w^{(k)})^T w^* = [(w^{(k-1)})^T + y_j x_j^T] w^*$

$$= (w^{(k-1)})^T w^* + y_j (w^*)^T x_j$$
$$\geq (w^{(k-1)})^T w^* + \rho \quad \text{—— (c)}$$

Now, to prove that $(w^{(k)})^T w^* \geq k\rho$, let us use induction. If $k=0$, we get, $0 \cdot w^* \geq 0$. If the thesis is true for $(k-1)$, let us prove it for k. Using (c), we have,

$$(w^{(k)})^T w^* \geq (w^{(k-1)})^T w^* + \rho \geq (k-1)\rho + \rho = k\rho$$

2. $\|w^{(k)}\|^2 = \|w^{(k-1)} + y_j x_j\|^2 = \|w^{(k-1)}\|^2 + \|y_j x_j\|^2 + 2 y_j (w^{(k-1)})^T x_j$

$$\leq \|w^{(k-1)}\|^2 + \|y_j x_j\|^2$$
$$\leq \|w^{(k-1)}\|^2 + \|x_j\|^2 ; [\because \|y_j\|^2 = 1]$$

Since x_j is misclassified by $w^{(k-1)}$, so $2 y_j (w^{(k-1)})^T x_j < 0$

Now, let us prove by induction that $\|w^{(k)}\|^2 \leq kR^2$ where $R = \max_j \|x_j\|_2$. If $k=0$, we have $0 \leq 0 \cdot R^2$. If the claim is true for $(k-1)$, let us prove it for k. So,

$$\|w^{(k)}\|^2 \leq \|w^{(k-1)}\|^2 + \|x_j\|^2 \leq (k-1)R^2 + R^2 = kR^2$$

(Showed)

3. Using 1 & 2, we get, $\frac{(w^{(k)})^T w^*}{\|w^{(k)}\|} \geq \frac{k\rho}{\sqrt{k} R} = \sqrt{k} \frac{\rho}{R}$ —— (3)

Since $\|w^{(k)}\| \leq \sqrt{k} R$ (from (2)) $\Rightarrow \frac{1}{\|w^{(k)}\|} \geq \frac{1}{\sqrt{k} R}$

Scanned with CamScanner

So from ③, $k \leq \frac{R^2}{\rho^2} \frac{((w^{(k)})^T w^*)^2}{\|w^{(k)}\|^2} = \frac{R^2}{\rho^2} \frac{((w^{(k)})^T w^*)^2}{\|w^{(k)}\|^2 \|w^*\|^2} \|w^*\|^2$

Now using Cauchy-Schwarz inequality, $\frac{((w^{(k)})^T w^*)^2}{\|w^{(k)}\|^2 \|w^*\|^2} \leq 1$

So we got, $k \leq \frac{R^2 \|w^*\|^2}{\rho^2}$

(Showed)

(c) Here the primal problem of soft margin SVM is

$$\underset{\theta, \varepsilon}{\arg\min} \ \frac{1}{2}\left(\|\omega\|_2^2 + c\|\varepsilon\|_2^2\right)$$

subject to $y_j(w^T x_j + w_0) \geq 1 - \varepsilon_j$, $\varepsilon_j \geq 0$, $j = 1, \dots, N$

(i) the Lagrangian of the problem is

$$\mathcal{L}(w, w_0, \varepsilon, \lambda) = \frac{1}{2}\left(\|\omega\|^2 + c\|\varepsilon\|^2\right) + \sum_{j=1}^{N} \lambda_j \left[1 - \varepsilon_j - y_j(w^T x_j + w_0)\right]$$

let us minimize over $(\omega, \omega_0, \varepsilon)$:

$$\nabla_\omega \mathcal{L}(\omega, \omega_0, \varepsilon, \lambda) = w^* - \sum_{j=1}^{N} \lambda_j y_j x_j = 0$$

$$\Rightarrow w^* = \sum_{j=1}^{N} \lambda_j y_j x_j$$

$$\nabla_{\omega_0} \mathcal{L}(\omega, \omega_0, \varepsilon, \lambda) = 0 \Rightarrow \sum_{j=1}^{N} \lambda_j y_j = 0$$

$$\nabla_\varepsilon \mathcal{L}(\omega, \omega_0, \varepsilon, \lambda) = 0 \Rightarrow c\varepsilon_j^* = \lambda_j, \text{ for } \forall j$$

Now, since for all j, $\lambda_j \geq 0$, so $\epsilon_j = \frac{1}{c}\lambda_j$ (with $c>0$).
so, the optimization solution is automatically $\epsilon_j \geq 0, \forall j$, so the constraint $\epsilon_j > 0, \forall j$ can be removed without affecting the solution.

(ii) We proved these in (i) taking derivative of the Lagrangian w.r.t. w, w_0 & ϵ respectively & setting $=0$.

(iii) So, the Lagrangian function becomes,

$$\mathcal{L}(w^*, w_0^*, \epsilon^*, \lambda) = \frac{1}{2}\left(\left\|\sum_{j=1}^{N}\lambda_j y_j x_j\right\|^2 + \frac{\sum \lambda_j^2}{c}\right) + \sum_{j=1}^{N}\lambda_j\left[1 - \epsilon_j^* - y_j\left(w^{*T} x_j + w_0^*\right)\right] \quad \text{(A)}$$

the 2nd term becomes,

Now, $$\sum_{j=1}^{N}\lambda_j - \sum_{j=1}^{N}\lambda_j \epsilon_j^* - \cancel{\sum_{j=1}^{N}\lambda_j\sum_{k=1}^{N}\lambda_k y_k} \sum_{j=1}^{N}\lambda_j y_j\left[\sum_{k=1}^{N}\lambda_k y_k x_k\right]^T x_j - \underbrace{\left(\sum_{j=1}^{N}\lambda_j y_j\right)}_{0} w_0^*$$

$$= \sum_{j=1}^{N}\lambda_j - \sum_{j=1}^{N}\lambda_j \frac{\lambda_j}{c} - \sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k y_j y_k x_j^T x_k$$

the 1st term of (A) is, $\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k y_j y_k x_j^T x_k + \frac{1}{2}\sum\frac{\lambda_j^2}{c}$

so, from (A), we have,

the convex dual problem is,

$$\max_{\lambda \geq 0} \left\{ \sum_{j=1}^{N} \lambda_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} \lambda_j \lambda_k y_j y_k x_j^T x_k - \frac{1}{2} \sum_{j=1}^{N} \frac{\lambda_j^2}{C} \right\}$$

subject to $\sum_{j=1}^{N} \lambda_j y_j = 0$

(d) (i) Here the problem is,

$$\underset{w, w_0}{\text{minimize}} \ \frac{1}{2} \|w\|_2^2 \ \text{subject to} \ y_j (w^T x_j + w_0) \geq \gamma \quad j=1,\dots N$$

So the Lagrangian becomes,

$$\mathcal{L}(w, w_0, \lambda) = \frac{1}{2} \|w\|_2^2 + \sum_{j=1}^{N} \lambda_j \left[\gamma - y_j (w^T x_j + w_0)\right]$$

So, $\nabla_w \mathcal{L} = 0 \Rightarrow w^* = \gamma \sum_j \lambda_j y_j x_j$, $\nabla_{w_0} \mathcal{L} = 0 \Rightarrow \sum_j \lambda_j y_j = 0$

Now, if we proceed as shown before (and for the usual Hard margin - SVM), we can get the solution from the optimization as:

$$w_\gamma^* = \gamma \sum_{j \in V} \lambda_j^* y_j x_j, \quad V \text{ is set of support vectors: } \lambda_j > 0$$

Now, $w_{0\gamma}^* = -\frac{(x^+ + x^-)^T w_\gamma^*}{2} = \gamma w_0^*$ [where $w_0^* = -\frac{(x^+ + x^-)^T w^*}{2}$ is the optimal w_0 for $y_j (w^T x_j + w_0) \geq 1$]

Since $w_\gamma^* = \gamma \underbrace{\sum_{j \in V} \lambda_j^* y_j x_j}_{\substack{w^* \text{ if constraint} \\ \text{was } y_j (w^T x_j + w_0) \geq 1}} = \gamma w^*$

So, with $y_j(w^T x_j + w_0) \geq \gamma$, the decision boundary is

$$\gamma w^{*T} x + \gamma w_0^* = 0$$

$$\Rightarrow \gamma(w^{*T}x + w_0^*) = 0 \Rightarrow w^{*T}x + w_0^* = 0$$

So, a scalar multiple of $\bar{\theta}$ does not change the decision boundary.

(ii) Here, let's assume $w^T x_1 + w_0 = 1$, $w^T x_2 + w_0 = -1$

So, $\mathcal{L}(w, w_0, \nu_1, \nu_2) = \frac{1}{2}\|w\|_2^2 + \nu_1[1-(w^T x_1 + w_0)] + \nu_2[1 + w^T x_2 + w_0]$

$\nabla_w \mathcal{L} = 0 \Rightarrow w = \nu_1 x_1 - \nu_2 x_2$; $\nabla_{w_0}\mathcal{L} = 0 \Rightarrow \nu_1 = \nu_2$

$\Rightarrow w = \nu_1(x_1 - x_2)$, so let's take $\nu_1 = \nu_2 = \nu$

So, the lagrangian becomes,

$$\frac{1}{2}\|w\|^2 + 2\nu + (-\nu)w^T(x_1 - x_2) = \frac{1}{2}\|w\|^2 + 2\nu - \underbrace{w^T w}_{\|w\|^2}$$

$$= -\frac{1}{2}\|w\|^2 + 2\nu$$

$$= -\frac{1}{2}\nu^2\|x_1 - x_2\|^2 + 2\nu$$

So, the optimization becomes,

$$\max_{\nu \geq 0} -\frac{1}{2}\nu^2\|x_1 - x_2\|^2 + 2\nu$$

taking 1st derivative, we get, $-\nu\|x_1 - x_2\|^2 + 2 = 0$

$$\Rightarrow \nu = \frac{2}{\|x_1 - x_2\|^2}$$

$$\therefore w^* = \frac{2(x_1 - x_2)}{\|x_1 - x_2\|^2}, \quad w_0^* = \frac{-(x_1 + x_2)^T(x_1 - x_2)}{\|x_1 - x_2\|^2}$$

So, even with 2 data points (one from each class) the decision boundary is uniquely determined.

(Showed)