PS2

d1 (a) algorithm converges on set A but doesn't on set B.

(b) math,

$$\mathcal{L}(\theta) = \prod_{i=1}^{m} p(y^{(i)} | x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \left(\frac{1}{1 + e^{-y^{(i)} \times \theta^T x^{(i)}}} \right)$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{m} \log \left(1 + e^{-y^{(i)} \times \theta^T x^{(i)}}\right) = J(\theta)$$

$$\nabla_\theta (\log \mathcal{L}(\theta)) = \sum_{i=1}^{m} \left(\frac{e^{-y^{(i)} (\theta^T x^{(i)})}}{1 + e^{-y^{(i)} \times \theta^T x^{(i)}}} \times y^{(i)} x^{(i)} \right)$$

$$= -X^T (\text{probs}) * Y$$

$$\nabla_\theta J(\theta) = \sum_{i=1}^{m} \left(\frac{1}{1 + e^{-y^{(i)} \theta^T x^{(i)}}} \times e^{-y^{(i)} \theta^T x^{(i)}} \times -y^{(i)} x^{(i)} \right)$$

$$= -X^T (1 - \text{probs}) Y$$

The formula in the code somhow works which I do not understand, but my derived equation also works well.

note that B is linearly separable while A is not; also our loss function wants to maximize $y^{(i)} \theta^T x^{(i)}$, which is nothing but the functional margin which can be scaled to arbitrary values by scaling θ, we cannot do so for A. This is why it fails to converge for dataset B

note that if we make the learning rate very large then A fails to converge because it overshoots (the minimum) but B reaches the limit of diff. $< 10^{-15}$ very fast. if we decrease this limit then we might get overflows but theoretically we can always keep changing theta for the dataset B acc. to our cost fn.

Suppose we try to capture GM minimization,

$$J(\theta) = \sum_{i=1}^{m} \log\left(1 + e^{\frac{-y^{(i)}\theta^T m^{(i)}}{(\theta^T\theta)}}\right)$$

$$\nabla_\theta J(\theta) = \sum_{i=1}^{m} \left(\frac{1}{1+e^{-\frac{y^{(i)}\theta^T m^{(i)}}{(\theta^T\theta)}}} \times e^{-\frac{y^{(i)}\theta^T m^{(i)}}{\theta^T\theta}} \times y^{(i)}\right)\left(\frac{m^{(i)}(\theta^T\theta) - 2\theta(\theta^T m^{(i)})}{(\theta^T\theta)^2}\right)$$

$$= \left\{\frac{X^T(\theta^T\theta) - 2\theta(X\theta)^T}{(\theta^T\theta)^2}\right\}(1 - \text{probs})\,Y$$

with this code however, the value with which θ is initialized becomes crucial, I have observed this.

(c) is No, it will keep learning with arbitrarily large learning rates, although this will cause an overflow.
try lr = 4 5 e4 (converges in about ~200 iters)

(e) (i) Yes xD, it will converge with a large lr ($\sim 10^{10}$) ∵ the loss will become 0 a more quickly.

(ii) Yes & No, if our convergence cond^n is just the difference being $< 1e^{-15}$ then yes ∵ learning rate itself becomes $< 1e^{-15}$ but actually the fitted lines are not very accurate.

(iii) No, the features are already in a proper scale.

(iv) Yes, definitely, with regularization we add the required scaling constraint. (for me 0.001 reg. worked best).

(v) Yes, ∵ this will result in a linearly non-seperable data.

(d) Nope, SVMs are robust against such datasets ∵ they maximize GM with the constraint $\|\theta\| = 1$ ⇒ there is a scaling contraint on θ & we would not be able to take it to arbitrarily high values indefinitely.

Q2 (a) $\sum_{i \in I_{a,b}} P(y^{(i)}=1 \mid x^{(i)};\theta) = \sum_{i=1}^{m} P(y^{(i)}=1 \mid x^{(i)};\theta)$

∵ $(a,b) = (0,1)$ all eg.

$$= \sum_{i=1}^{m} \left(\frac{1}{1+\exp(-\theta^T x^{(i)})} \right)$$

$$= \sum_{i=1}^{m} \left(\frac{1}{1+\exp(-(\theta')^T x^{(i)'} - \theta_0)} \right)$$

$$= \sum_{i=1}^{m} \left(\frac{e^{\theta_0}}{e^{\theta_0} + e^{-\theta^T x^{(i)}}} \right)$$

Q2 (b) $l(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log (1-h(x^{(i)}))$

$$\frac{\partial l(\theta)}{\partial \theta_j} = -\sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x_j^{(i)} = 0$$

at $j=0$, $x_j^{(i)} = 1 \ \forall j \in I_{a,b}$ (≤ full set)

$$\Rightarrow \sum_{i=1}^{m} h(x^{(i)}) = \sum_{i=1}^{m} y^{(i)} \ \star$$

2+ 1-
3+ , 1+ 2- , 3-

clearly, $h(x^{(i)}) = P(y^{(i)}=1 | x^{(i)}; \theta)$

$y^{(i)} = I\{y^{(i)}=1\}$

& everything follows.

(b) No, suppose we have eg in set $(a,b) = (0.25, 0.75)$

s.t. $h(x^{(1)}) = 0.66$ & $y^{(1)} = 0$
$h(x^{(2)}) = 0.66$ & $y^{(2)} = 1$
$h(x^{(3)}) = 0.66$ & $y^{(3)} = 1$

$\Rightarrow \sum_{i \in I_{a,b}} h(x^{(i)}) = 0.66 \times 3$

at a threshold of 0.5 our model doesn't get 100% accuracy but (a) holds!

converse is also not true, eg.-

$h(x^{(1)}) = 0.51$ & $y^{(1)} = 1$
$h(x^{(2)}) = 0.51$ & $y^{(2)} = 1$
$h(x^{(3)}) = 0.51$ & $y^{(3)} = 1$

at 0.5 threshold, 100% accuracy but model isn't perfectly calibrated for range $(0.5, 0.52)$.

(c) with L2 regularization, model will not be well calibrated ∴, our earlier proof fails to hold.

i.e., it become $\sum_{i=1}^{m} h(x^{(i)}) + \lambda\theta_0 = \sum_{i=1}^{m} y^{(i)}$

reg. term

PAGE No. 86
DATE / /

P52 contd.

(d) $\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(\theta|n,y)$

$\theta_{MLE} = \underset{\theta}{\operatorname{argmax}}\ p(y|n;\theta)$

(a) $p(\theta|n,y) = p(y|n,\theta) = \dfrac{p(n,y|\theta)\,p(\theta)}{p(n,y)}$

$= p(y|n,\theta)\,p(n|\theta)\,p(\theta)$

(a) $p(\theta|n,y) = \dfrac{p(n,y|\theta)\,p(\theta)}{p(n,y)}$

also, $p(n,y|\theta) = p(y$

(a) $p(\theta|n,y) = \dfrac{p(\theta,n,y)}{p(n,y)}$

$= p(y|\theta,n)\,\dfrac{p(\theta,n)}{p(n,y)}\,p(\theta)$

$= \dfrac{p(y|\theta,n)\,p(\theta,n)}{p(n,y)}$

$= \dfrac{p(y|\theta,n)\,p(\theta|n)\,p(n)}{p(y|n)\,p(n)}$

$p(\theta|y,n) = \dfrac{p(y|\theta,n)\,p(\theta|n)}{p(y|n)}$

$p(y|n)$ → constant

$\Rightarrow \theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(y|\theta,n)\,p(\theta)$

(b) $\theta \sim \mathcal{N}(0, \eta^2 I)$

$$p(\theta) = \frac{1}{(2\pi)^{n/2} \sqrt{|\eta^2 I|}} \times \exp\left(-\frac{(\theta-0)^T (\eta^2 I)^{-1} (\theta-0)}{2}\right)$$

$$p(\theta) = \frac{1}{(2\pi)^{n/2} \times \eta^2} \times \exp\left(-\theta^T \theta / 2\eta^2\right)$$

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(y|m,\theta)\, p(\theta)$$

$$\text{now, } p(y|m,\theta)\, p(\theta) = p(y|m,\theta) \times \frac{1}{(2\pi)^{n/2} \eta^2} e^{-\theta^T\theta/2\eta^2}$$

$$l(\theta) = -\log\{p(y|m,\theta)\, p(\theta)\}$$

$$= -\log p(y|m,\theta) + \frac{n}{2}\log 2\pi + 2\log\eta + \frac{1}{2\eta^2}\theta^T\theta$$

$$\theta^T\theta \equiv \|\theta\|_2^2$$

$$\therefore \theta_{MAP} = \underset{\theta}{\operatorname{argmin}}\left(-\log p(y|m,\theta) + \frac{1}{2\eta^2}\|\theta\|_2^2\right)$$

$$\lambda = 1/2\eta^2$$

(c) $y = \theta^T x + \varepsilon$

$\varepsilon \sim \mathcal{N}(0, \sigma^2)$
$\theta \sim \mathcal{N}(0, \eta^2 I)$

$\because \varepsilon \sim \mathcal{N}(0, \sigma^2)$

$\Rightarrow (\vec{y} - \theta^T X) \sim \mathcal{N}(0, \sigma_I^2)$.

$$p(\vec{y} \mid \theta, X) = \frac{1}{(2\pi)^{n/2} \sigma^2} \exp\left(-\frac{(\vec{y} - \theta^T X - 0)^T (\sigma^2 I)^{-1} (\vec{y} - \theta^T X - 0)}{2}\right)$$

$$= \frac{1}{(2\pi)^{n/2} \sigma^2} \exp\left(-\frac{(\vec{y} - \theta^T X)^T (\vec{y} - \theta^T X)}{2\sigma^2}\right)$$

$$p(\theta) = \frac{1}{(2\pi)^{n/2} \eta^2} \exp\left(-\frac{\theta^T \theta}{2\eta^2}\right)$$

$$\theta_{MAP} = \underset{\theta}{\text{argmax}}\, p(\vec{y} \mid X, \theta)\, p(\theta)$$

as in prev. q,

$$\theta_{MAP} = \underset{\theta}{\text{argmin}} \left(-\log p(\vec{y} \mid X, \theta) + \frac{n}{2} \log 2\pi + 2 \log \eta + \frac{1}{2\eta^2} \theta^T \theta\right)$$

$$= \underset{\theta}{\text{argmin}} \left(\frac{n}{2} \log 2\pi + 2 \log \sigma + \frac{1}{2\sigma^2} (\vec{y} - \theta^T X)^T (\vec{y} - \theta^T X) + 2 \log \eta + \frac{1}{2\eta^2} \theta^T \theta\right)$$

$$\therefore \theta_{MAP} = \arg\min_{\theta} \left(\frac{(\vec{y} - \theta^T x)^T (\vec{y} - \theta^T x)}{2\sigma^2} + \frac{1}{2\eta^2} \theta^T \theta \right)$$

$$\therefore J(\theta) = \frac{1}{2\sigma^2} \|\vec{y} - \theta^T x\|_2^2 + \frac{1}{2\eta^2} \|\theta\|_2^2$$

$$J(\theta) = \frac{1}{2\sigma^2} (y - X\theta)^T (y - X\theta) + \frac{1}{2\eta^2} \theta^T \theta$$

$$\nabla_\theta J(\theta) = \frac{1}{2\sigma^2} \nabla_\theta \{(y^T - \theta^T X^T)(y - X\theta)\} + \frac{1}{2\eta^2} 2\theta$$

$$= \frac{1}{2\sigma^2} \nabla_\theta (y^T y - y^T X\theta - \theta^T X^T y + \theta^T X^T X\theta) + \frac{1}{\eta^2}\theta$$

$$= \frac{1}{2\sigma^2} (0 - 2X^T y + (\theta^T X^T X)^T + (\theta^T X^T X)) + \frac{1}{\eta^2}\theta$$

$$= 0$$

$$\Rightarrow (\theta^T X^T X)^T - X^T y + \frac{\sigma^2}{\eta^2}\theta = 0$$

$$X^T X\theta - X^T y + \frac{\sigma^2}{\eta^2}\theta = 0$$

$$\left(X^T X + \frac{\sigma^2}{\eta^2} I\right)\theta = X^T y$$

$$\boxed{\theta = \left(X^T X + \frac{\sigma^2}{\eta^2} I\right)^{-1} X^T y}$$

(d) If $f_{Z}(z|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|z-\mu|}{b}\right)$

$\theta \sim \mathcal{L}(0, bI)$

$\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$y = \theta^T x + \varepsilon$

similar to prev,

$$p(\theta) = \frac{1}{2b} \exp^{\frac{-|\theta|}{b}}$$

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{2\sigma^2} \|(X\theta - \vec{y})\|_2^2 + \log 2b + \frac{|\theta|}{b} \right)$$

$$\therefore J(\theta) = \|X\theta - \vec{y}\|_2^2 + \frac{2\sigma^2}{b} \|\theta\|_1$$

$\gamma = \frac{2\sigma^2}{b}$

** L_2 regression $\Rightarrow$ ridge regression

L_1 " $\Rightarrow$ Lasso "

also called weight decay OR shrinkage ∵ encourage parameters to stay close to their mean ($=0$) ∴ resulting in shrinkage.

** L_1 reg. results in sparse parameters, i.e. most of the values are zero.

Q4(a) $K(x,z) = K_1(x,z) + K_2(x,z)$

$z^T K z = z^T (K_1 + K_2) z$
$= z^T K_1 z + z^T K_2 z \geq 0.$

(b) $z^T K z = z^T (K_1 - K_2) z$
let $K_2 = \lambda K_1 \ (\lambda > 1)$

$= (1-\lambda) z^T K_1 z$
≤ 0
not PSD!

(c) is PSD $\because z^T K z = a(z^T K_1 z) \geq 0$

(d) is not PSD $\because z^T K z = -a(z^T K_1 z) \leq 0.$

(e) $K = K_1 K_2$

$K^T = K_2 K_1$

$z^T K z = \sum_i \sum_j z_i K(x^{(i)}, x^{(j)}) z_j$

$= \sum_i \sum_j z_i K_1(x^{(i)}, x^{(j)}) K_2(x^{(i)}, x^{(j)}) z_j$

$= \sum_i \sum_j z_i \phi_1(x^{(i)})^T \phi_1(x^{(j)}) \phi_2(x^{(i)})^T \phi_2(x^{(j)}) z_j$

$= \sum_i \sum_j z_i \left(\sum_k \phi_1(x^{(i)})_k \phi_1(x^{(j)})_k\right) \left(\sum_l \phi_2(x^{(i)})_l \phi_2(x^{(j)})_l\right)$

$= \sum_i \sum_j \sum_k \sum_l z_i \phi_1(x^{(i)})_k \phi_1(x^{(j)})_k \phi_2(x^{(i)})_l \phi_2(x^{(j)})_l z_j$

$$\Rightarrow z^T K z = \sum_j \sum_k \sum_\ell \left(z_j \phi_1(x^{(j)})_k \phi_2(x^{(j)})_\ell \right)^2$$

$$(z^T K z)^T = z^T K^T z$$

$$\Rightarrow z^T K z = z^T \left(\frac{1}{2} K + \frac{1}{2} K^T \right) z$$

$$= \frac{1}{2} z^T (K_1 K_2 + K_2 K_1) z$$

(f) $K(x, z) = f(x) f(z)$

$$z^T K z = \sum_i \sum_j z_i K_{ij} z_j$$

$$= \sum_i \sum_j z_i f(x^{(i)}) f(x^{(j)}) z_j$$

$$= \sum_i \left(z_i f(x^{(i)}) \right)^2 \quad \text{PSD}$$

(g) $K(x, z) = k_3(\phi(x), \phi(z)) = \phi'(\phi(x))^T \phi'(\phi(z))$

let $\phi'' = \phi'(\phi(\cdot))$ clearly K is PSD.

(h) $K(x, z) = p(K_1(x, z)) = p(\phi(x)^T \phi(z))$

$$= \sum_{i=0}^{n} k_i (\phi(x)^T \phi(z))^i$$

(b) $K(x, z) = p(K_1(x, z)) = \sum_i k_i (K_1(x, z))^i$

$k_i > 0$ ↳ PSD

$z^T K z \geq 0$ PSD

Q5 $\theta^{(i+1)} := \theta^{(i)} + \alpha (y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)})) x^{(i+1)}$

(a) (i) $\theta^{(i)} = \sum_{j=1}^{i} \beta_j \phi(x^{(j)})$ *

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

(ii) $h_{\theta^{(i)}}(x^{(i+1)}) = g\left(\sum_{j=1}^{i} \beta_j K(x^{(j)}, x^{(i+1)})\right)$

$= g\left(\sum_{j=1}^{i} \beta_j \langle x^{(j)}, x^{(i+1)} \rangle\right)$

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

$\therefore \beta_{i+1} = \alpha (y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)}))$

& $\theta^{(i+1)} = \sum_{j=1}^{i+1} \beta_j \phi(x^{(j)})$.

(b) Code

(c) dot product kernel behaves poorly ∵ it does not have the expressive power of higher order features & tries to classify points by making a linear decision boundary

Q6 Code