Machine Learning HW3

Fall 2020

Study Group: Hsuan-Cheng Chen, Cheng-Hao Chou

PoKang Chen

1.

ML HW 3.

1. Assume that $\|w_{MAP}\|_2 > \|w_{ML}\|_2$.

$$\Rightarrow p(w_{MAP}) = \frac{1}{(2\pi)^{\frac{n+1}{2}} |\tau^2 I|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2\tau^2}(\|w_{MAP}\|_2)^2} < \frac{1}{(2\pi)^{\frac{n+1}{2}} |\tau^2 I|^{1/2}} e^{-\frac{1}{2\tau^2}(\|w_{ML}\|_2)^2} = p(w_{ML})$$

$$\Rightarrow p(w_{MAP}) \left[\prod_{i=1}^{m} p(y^{(i)} | x^{(i)}, w_{MAP}) \right] < p(w_{ML}) \left[\prod_{i=1}^{m} p(y^{(i)} | x^{(i)}, w_{MAP}) \right]$$

$$\leq p(w_{ML}) \left[\prod p(y^{(i)} | x^{(i)}, w_{ML}) \right]$$

w_{ML} is chosen to maximize $p(y^{(i)} | x^{(i)}; w)$

However, this contradicts b.c. w_{MAP} should maximize MAP, which is

$$p(w) \cdot \left[\prod_{i=1}^{m} p(y^{(i)} | x^{(i)}; w) \right]$$

$$\therefore \|w_{MAP}\|_2 \leq \|w_{ML}\|_2$$

2.

ML n : D $k_1, k_2 : R^D \times R^D$ $\phi : R^D \to R^M$ $p : R \to R$

HW3 d : M $f : R^D \to R$ $k_3 : R^M \times R^M$ $K : R^{N \times N}$

m : N

2.

$$u^T K u = \sum_{i,j}^{N} u_i u_j \cdot K(x^{(i)}, x^{(j)})_{N \times N}$$

(a)

$$u^T K u = \sum_{i,j}^{N} u_i u_j [k_1(x^{(i)}, x^{(j)}) + k_2(x^{(i)}, x^{(j)})]$$

$$= \sum_{i,j}^{N} u_i u_j k_1(x^{(i)}, x^{(j)}) + \sum_{i,j}^{N} u_i u_j k_2(x^{(i)}, x^{(j)}) \geq 0 \quad P.D \quad \therefore K \text{ is a kernel}$$

(b)

$$u^T K u = \sum_{i,j}^{N} u_i u_j [k_1(x^{(i)}, x^{(j)}) - k_2(x^{(i)}, x^{(j)})]$$

$$= \sum_{i,j}^{N} u_i u_j k_1(x^{(i)}, x^{(j)}) - \sum_{i,j}^{N} u_i u_j k_2(x^{(i)}, x^{(j)}) \not\geq 0 \quad \therefore K \text{ isn't a kernel}$$

(c)

$$u^T K u = a \sum_{i,j}^{N} u_i u_j \cdot k_1(x^{(i)}, x^{(j)}) \geq 0 \quad \therefore K \text{ is a kernel}$$

(d)

$$u^T K u = -a \sum_{i,j}^{N} u_i u_j \cdot k_1(x^{(i)}, x^{(j)}) < 0 \quad \therefore K \text{ isn't a kernel}$$

(e)

$$u^T K u = u^T K_1 K_2 u$$

$$= u^T K_1 (u u^T)(u u^T)^{-1} (u u^T)^{-1} (u u^T) K_2 u$$

$$= (u^T K_1 u) u^T (Q^T \Lambda Q)^{-1} (Q^T \Lambda Q)^{-1} u (u^T K_2 u) \qquad u u^T = Q^T \Lambda Q \text{ for that } Q Q^T = I, \ Q^T = Q^{-1}$$

$$= (u^T K_1 u)(u^T Q^T) \Lambda^{-1} \Lambda^{-1} (Q u)(u^T K_2 u)$$

$$= \underbrace{(u^T K_1 u)}_{>0} [(Q u)^T \Lambda^{-2} (Q u)] \underbrace{(u^T K_2 u)}_{>0} \geq 0 \quad \therefore K = K_1 K_2 \text{ is a kernel}$$

(f)

$$u^T K u = \sum_{i,j}^{N} u_i u_j f(x^{(i)}) f(x^{(j)}) = [u_1 f(x^{(1)}) + u_2 f(x^{(2)}) + \cdots + u_m f(x^{(m)})]^2 \geq 0 \quad \therefore K \text{ is a kernel}$$

(g) Since k_3 is a kernel $K = k_3(\Phi(x), \Phi(z))$ is a kernel too

(h)

$$u^T K u = \sum_{i,j}^{N} u_i u_j p[K_1(x^{(i)}, x^{(j)})] = \sum_{i,j}^{N} u_i u_j \left[\sum_k w_k (K_1(x^{(i)}, x^{(j)}))^k\right] \geq 0$$

K is a kernel

(i)

Gaussian kernel: $K = e^{\left(-\frac{\|x^{(i)} - z\|^2}{2\sigma^2}\right)} \rightarrow \|x^{(i)} - z\|^2 = x^{(i)T} z + x^{(j)T} z - 2 x^T z$

$$= e^{\left(\frac{x^T x}{-2\sigma^2}\right)} \times e^{\left(\frac{z^T z}{-2\sigma^2}\right)} e^{\frac{x^T z}{\sigma^2}} = f(x) \cdot f(z) \times \boxed{e^{\frac{x^T z}{\sigma^2}}} \rightarrow e^{\frac{x^T z}{\sigma^2}} = 1 + \frac{1}{1}\left(\frac{x^T z}{\sigma^2}\right) + \frac{1}{2!}\left(\frac{x^T z}{\sigma^2}\right)^2 + \frac{1}{3!}\left(\frac{x^T z}{\sigma^2}\right)^3 \cdots$$

$$= 1 \cdot 1 + \frac{x}{\sigma} \cdot \frac{z}{\sigma} + \frac{1}{2!}\left(\frac{x}{\sigma}\right)^2 \left(\frac{z}{\sigma}\right)^2 + \frac{1}{3!}\left(\frac{x}{\sigma}\right)^3 \left(\frac{z}{\sigma}\right)^3 + \cdots$$

$$= \begin{bmatrix} 1 \\ \frac{x}{\sigma} \\ \left(\frac{x}{\sigma}\right)^2 \\ \vdots \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{z}{\sigma} \\ \left(\frac{z}{\sigma}\right)^2 \\ \vdots \end{bmatrix} = \phi'(x) \cdot \phi'(z)$$

$\therefore \phi(x) = f(x)\, \phi'(x)$

$\phi(z) = f(z)\, \phi'(z)$

$\therefore k(x, z)$ is a kernel and $\phi(\cdot)$ is infinite dimensional

3.

3. (a) $w^{(1)} = w^{(0)} + \alpha[y^{(1)} - h(x^{(1)}; w)]x^{(1)}$

$w^{(0)} = \beta_0 \phi(x^{(0)})$, where $\beta_0 = 0$

$w^{(1)} = \beta_0 \phi(x^{(0)}) + \underbrace{\alpha[y^{(1)} - f(w^{(0)T}\phi(x^{(1)}))]}_{\beta_1}\phi(x^{(1)}) = \beta_0 \phi(x^{(0)}) + \beta_1 \phi(x^{(1)}) = \sum_{j=0}^{1} \beta_j \phi(x^{(j)})$

$\therefore w^{(0)}, w^{(1)}, w^{(2)}, \dots w^{(j)}$ are linear combination of $\phi(x^{(j)})$

$\therefore w^{(i)} = \sum_{j=0}^{i} \beta_j \phi(x^{(j)})$

(b)

kernel

$h(x^{(i+1)}; w^{(i)}) = f(w^{(i)T}\phi(x^{(i+1)})) = f(\sum_{l=1}^{i} \beta_l \cdot \phi(x^{(l)})^T \phi(x^{(i+1)})) = f(\beta_l \cdot k(x^{(l)T}, x^{(i+1)}))$

Could avoid explicitly write out what $\phi(x)$ is.

But since $\beta_i = \alpha[y^{(i)} - f(w^{(i-1)T}\phi(x^{(i)}))]$ is still represented by feature,

$= \alpha[y^{(i)} - f(\sum_{j=0}^{i-1} k(x^{(j)T}, x^{(i)}))]$ could successfully avoid $\phi(x)$

(c)

$w^{(i+1)} = w^{(i)} + \alpha[y^{(i+1)} - h(\phi(x^{(i+1)}); w^{(i)})]\phi(x^{(i+1)})$

$f(z) = \text{sign}(z) = \begin{cases} 1 & z \geq 0 \\ -1 & \text{otherwise} \end{cases}$ These are the cases of misclassification

$\therefore w^{(i+1)} = \begin{cases} w^{(i)} + \alpha[y^{(i+1)} - h(\phi(x^{(i+1)}); w^{(i)})]\phi(x^{(i+1)}) & \text{if } y^{(i+1)} = \hat{y}^{(i+1)} \\ w^{(i)} + 0 & \text{misclassification}, y^{(i+1)} \neq \hat{y}^{(i+1)} \end{cases}$

When updating $K(w^{(i)}, \phi(x^{(i+1)}))$, we could dot product both sides by $\phi(x^{(i+1)})$

$K(w^{(i+1)}, \phi(x^{(i+1)})) = \begin{cases} K(w^{(i)}, \phi(x^{(i+1)})) + \alpha y^{(i+1)} K(\phi(x^{(i+1)}), \phi(x^{(i+1)})) & \text{if } K(w^{(i)}, \phi(x^{(i+1)}))y^{(i+1)} < 0 \\ K(w^{(i)}, \phi(x^{(i+1)})) & \text{otherwise} \end{cases}$

$\Rightarrow$ We don't need to explicitly show the $\phi(x)$, instead using dot product in kernelized form.

□

4(a)

4.

(a)

Let $H(w,x,y) = \max\{0,\ 1-y^{(i)}(w^Tx+b)\}$

$$\frac{\partial H}{\partial w_j} H = \begin{cases} 0 & y^{(i)}(w^Tx+b)\ge 1 \\ -yx_j & y^{(i)}(w^Tx+b)<1 \end{cases} \qquad \frac{\partial H}{\partial b} H = \begin{cases} 0 & y^{(i)}(w^Tx+b)\ge 1 \\ -y & y^{(i)}(w^Tx+b)<1 \end{cases}$$

$$\therefore\ \nabla_w E(w,b) = w - C\sum_{i=1}^{N} I\{y^{(i)}(w^Tx+b)<1\}\, y^{(i)}x^{(i)}$$

$$\frac{\partial}{\partial b}E(w,b) = -C\sum_{i=1}^{N} I\{y^{(i)}(w^Tx+b)<1\}\, y^{(i)}$$

4(b)

```
w =  [[ 96.         -36.64285714 233.57142857  88.28571429]]
b =  [-0.06892857]
[Iter    5: accuracy = 54.1667%
w =  [[ -1.98076923 -11.71153846  25.35576923  11.32692308]]
b =  [-0.28672539]
[Iter   50: accuracy = 95.8333%
w =  [[-1.99019608 -4.81862745 11.45098039  5.74019608]]
b =  [-0.29568526]
[Iter  100: accuracy = 95.8333%
w =  [[-0.499501   -0.3243513   1.05538922  1.28293413]]
b =  [-0.31806329]
[Iter 1000: accuracy = 95.8333%
w =  [[-0.3517593  -0.2779888   0.88644542  1.00329868]]
b =  [-0.3329032]
[Iter 5000: accuracy = 95.8333%
w =  [[-0.33655448 -0.28065645  0.89411863  0.98642119]]
b =  [-0.33432381]
[Iter 6000: accuracy = 95.8333%
```

[Iter   5: accuracy = 54.1667%
[Iter   50: accuracy = 95.8333%
[Iter 100: accuracy = 95.8333%
[Iter 1000: accuracy = 95.8333%
[Iter 5000: accuracy = 95.8333%
[Iter 6000: accuracy = 95.8333%

4(c)

c.

$$\nabla_w E^{(i)}(w,b) = \frac{1}{N}\|w\| - C \cdot I\{y^{(i)}(w^T x^{(i)} + b) < 1\} y^{(i)} x^{(i)}$$

$$\frac{\partial}{\partial b} E^{(i)}(w,b) = -C \cdot I\{y^{(i)}(w^T x^{(i)} + b) < 1\} y^{(i)}$$

4(d)

```
w =  [[-1.60513517 -2.82975568  7.75514067  4.70009547]]
b =  [-0.03916667]
[Iter    5: accuracy = 95.8333%
w =  [[-1.57751374 -0.28825955  2.67365117  2.87843094]]
b =  [-0.07070155]
[Iter   50: accuracy = 95.8333%
w =  [[-1.3205845  -0.03813763  1.82861082  2.35350451]]
b =  [-0.07804253]
[Iter  100: accuracy = 95.8333%
w =  [[-0.57038277 -0.22985423  1.05386375  1.23410777]]
b =  [-0.10526065]
[Iter 1000: accuracy = 95.8333%
w =  [[-0.47823705 -0.29816286  0.98836263  1.17408872]]
b =  [-0.12475633]
[Iter 5000: accuracy = 95.8333%
w =  [[-0.49779203 -0.28000682  1.00578787  1.18060628]]
b =  [-0.12631455]
[Iter 6000: accuracy = 95.8333%
```

Iter   5: accuracy = 95.8333%
Iter   50: accuracy = 95.8333%
Iter 100: accuracy = 95.8333%
Iter 1000: accuracy = 95.8333%
Iter 5000: accuracy = 95.8333%
Iter 6000: accuracy = 95.8333%

4(e)

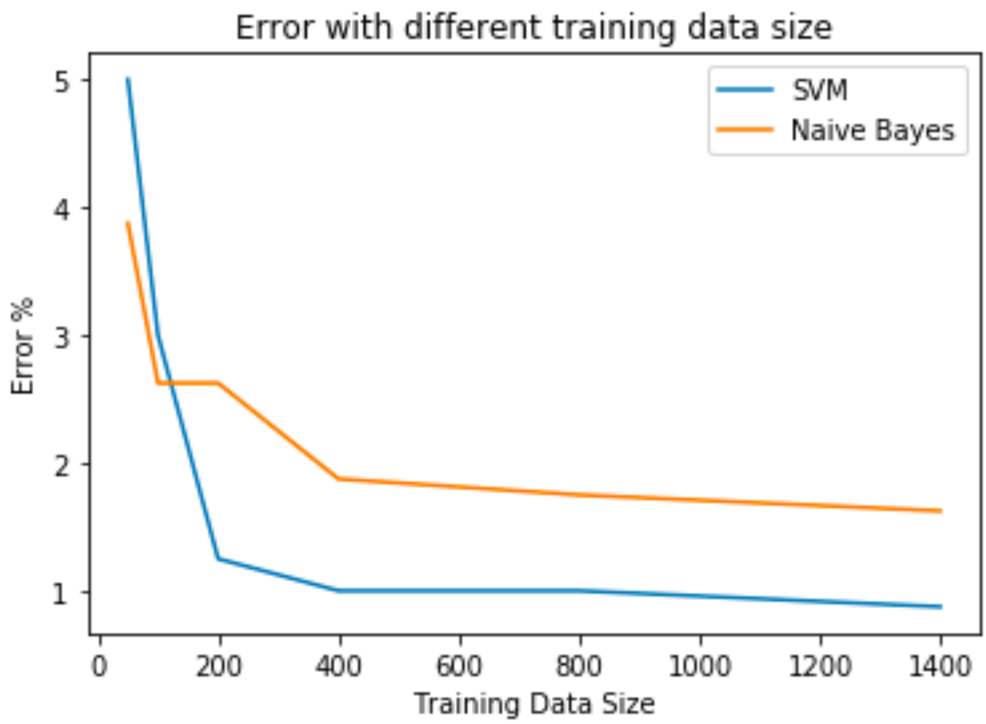
In general, the choice of the different method between sgd and bgd depends mainly on the distribution of the data. However, in this case, SGD works better maybe because the data type fits better for SGD.

5(a)
Error: 0.3750%

5(b)

data:50, Error: 5.0000%
data:100, Error: 3.0000%
data:200, Error: 1.2500%
data:400, Error: 1.0000%
data:800, Error: 1.0000%
data:1400, Error: 0.8750%

5(c)



As showed in the following plots, SVM works better as the training data size gets bigger. The error decreases when the training data size meet some value. The reason is that since the Naive Bayes is a probabilistic-based but not a real model as SVM.