10601 Machine Learning HW3

Hsueh Lin Huang
hsuehlih

1. Kernel feature mappings

1.

$$K(x,z)=\phi(x)\cdot\phi(z) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)\cdot(z_1^2, \sqrt{2}z_1z_2, z_2^2)$$
$$= (x_1z_1 + x_2z_2)^2$$

2.

(a)

$R^2 \to R^3$

$$\begin{array}{cc} 3\times 2 & 2\times 1 \\ \begin{bmatrix} & \\ & \\ & \end{bmatrix} & \begin{bmatrix} \\ \end{bmatrix} \end{array} \to (2 \text{ multiplication} + 1 \text{ addition}) \times 3$$

mul: $2\times3\times2 + 3 = 15$

add: $\underbrace{3\times2}_{\text{map to feature space}} + (3-1) = 8$

(b) $R^2 \to R^3$

$k(x,z) = (x_1^2, x_2^2, \sqrt{2}x_1x_2 \quad)$

mul: 3

add: 1

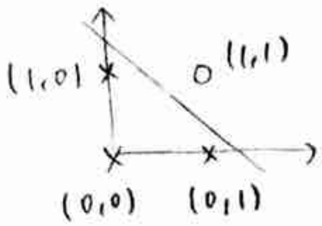
2. Perceptrons

1.

AND

x_1	x_2	AND
0	0	0
0	1	0
1	0	0
1	1	1

2.



$$\sum w_i x_i = -b$$

$$w_1 x_1 + w_2 x_2 = -b$$

$$x_2 = \frac{-b - w_1 x_1}{w_2}$$

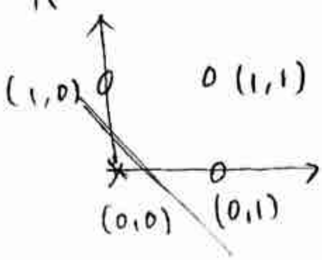
$w_1 = 1,\ w_2 = 1,\ b = -1.5$

3.

OR

x_1	x_2	OR
0	0	0
0	1	1
1	0	1
1	1	1

4.

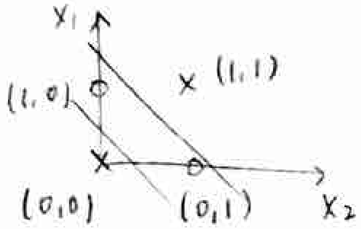


$w_1 = 1,\ w_2 = 1,\ b = -0.5$

5. XOR

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

6.



We need two straight lines to seperate the different outputs.

→ we can only solve this either change the transfer function so that it has more than one decision boundary, or use a more complex network that is able to generate more complex decision boundary.

3. Regression Theory

3.1 Linear Regression

1. (a)

$$f(x) = w^T x$$

Squared error loss $J(w) = \|y - f(x)\|^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2$

(b)

$$\frac{\partial J(w)}{\partial w^k} = -2 \sum_{i=1}^{n} (y_i - f(x_i))\, x_i^k$$

(c)

$$w^k_{new} = w^k + \underbrace{\alpha}_{\text{Step}} \cdot \frac{\partial J(w)}{\partial w^k} = w^k + \alpha \sum_{i} (y_i - f(x_i))\, x_i^k$$

2.

(a)

$$L(w \mid x, y) = p(y \mid x, w) = \prod_{i=1}^{n} p(y_i \mid x_i, w)$$

$$= \prod_{i=1}^{n} (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \frac{(y_i - f(x_i))^2}{\sigma^2}\right)$$

$$= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(x_i))^2\right)$$

(b)

$$\ell(w|X,y) = \ln(L(w|X,y)) = \ln\left((2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2\right)\right)$$

$$= \ln\left((2\pi\sigma^2)^{-\frac{n}{2}}\right) + \ln\left(\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2\right)\right)$$

$$= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

$$= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

(c)

$$\frac{\partial(\ell(w|X,y))}{\partial w_k} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))\, x_i^k$$

$$\frac{\partial J(w)}{\partial w_k} = -\sum_{i=1}^{n}(y_i - f(x_i))\, x_{ik}$$

minimize negative log likelihood function is the same as minimize the loss function.

3.

$\epsilon = N(0, \sigma^2)$

(a)

$$E\Big[\sum_{i=1}^{n}(\underbrace{y_i - \hat{f}(x_i)}_{\epsilon_i})^2\Big]$$

$\Rightarrow$ The expectation is taken over by ϵ

(b)

$$E[(y - \hat{f}(x))^2] = E[\hat{f}(x)^2 - 2\hat{f}(x)y + y^2] = E[\hat{f}(x)^2] - 2E[\hat{f}(x)]E[y] + E[y^2]$$

$$= E[(\hat{f}(x) - E[\hat{f}(x)])^2] + E[\hat{f}(x)]^2 - 2E[\hat{f}(x)]f(x) + E[(y - f(x))^2] + f(x)^2$$

$$= E[(\hat{f}(x) - E[\hat{f}(x)])^2] + \quad (\text{variance}^2)$$

$$(E[\hat{f}(x)] - f(x))^2 \quad + \quad (\text{bias}^2)$$

$$E[(y - f(x))^2] \quad (\sigma^2)$$

3.2 Regularization

1.(a)

$$J(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \frac{\lambda}{2}\|w\|^2$$

$$\frac{\partial J(w)}{\partial w^k} = -\sum_{i=1}^{n}(y_i - w^T x_i)x_i^k + \lambda w_i^k$$

(b)

(ii) is more likely to give sparse w, because it has additional constraint that $w^{k,(t+1)} < 0$, set $w^{k,(t+1)} = 0$ #

4. Programming logistic regression

1. $$f(x|w) = \frac{1}{1+e^{-x\cdot w}}$$

2. $$L(w|x,y) = \prod_{i=1}^{n} P(y_i | x_i, w) = \prod_{i=1}^{n} f(x|w)^{y_i}(1-f(x|w)^{1-y_i})$$

3. $$l(w|x,y) = \sum_{i=1}^{n} y_i \log f(x|w) + (1-y_i)\log(1-f(x|w))$$

$$= \sum_{i=1}^{n} \log(1-f(x|w)) + \sum_{i=1}^{n} y_i \log\frac{f(x|w)}{1-f(x|w)}$$

$$= \sum_{i=1}^{n} -\log(1+e^{w_0+x\cdot w}) + \sum_{i=1}^{n} y_i(w_0 + x\cdot w)$$

4. $$\frac{\partial l(w|x,y)}{\partial w^k} = -\sum_{i=1}^{n}\frac{1}{1+e^{w_0+x\cdot w}}e^{w_0+x\cdot w}x_i^k + \sum_{i=1}^{n} y_i x_i^k$$

$$= \sum_{i=1}^{n}(y_i - f(x|w))x_i^k$$

5. $$w^k_{new} = w^k + \alpha \frac{\partial l(w|x,y)}{\partial w^k} = w^k + \alpha \sum_{i=1}^{n} (y_i - f(w^T x_i)) x_i^k$$

6.
$$\frac{\partial\left(-l(w|x,y) + \frac{\lambda}{2}\|w\|^2\right)}{\partial w^k} = -\sum_{i=1}^{n}(y_i - f(x|w)) x_i^k + \lambda w_i^k$$

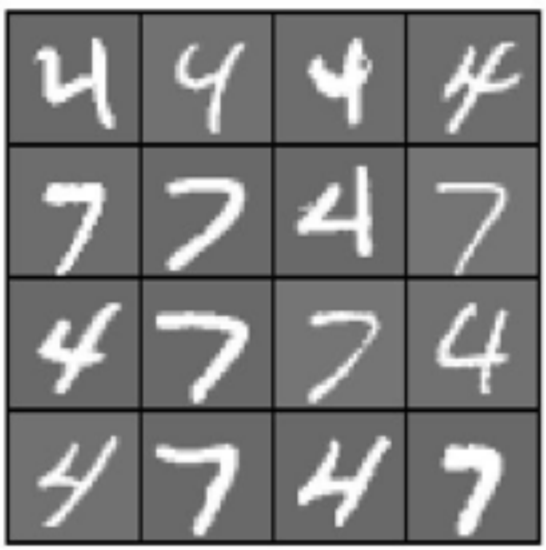
4 Programming Logistic Regression

4.3 Analysis of Results
w/o Regularization:
Your training accuracy is:99.2318%
Your testing accuracy is:98.7065%

Regularization:
Your training accuracy is:97.5799%
Your testing accuracy is:96.8159%

1. The accuracy drops after adding the regularization term. The purpose of adding regularization term is to avoid overfitting problem. However, if lambda is too large, then it might underfit the data because of high bias.

2.



5 Programming Kernel Perceptron

5.1 Perceptron
Your training accuracy is:98.4224%
Your test accuracy is:97.7114%

5.2 Kernel Perceptron
Your training accuracy is:97.7699%
Your test accuracy is:97.7114%

It seems like the accuracy between kernel perceptron and perceptron are the same.