MACHINE LEARNING

ADITYA GAUTAM

agautam1@andrew

ASSIGNMENT #3

18th Feb 2016

Ques 1) Kernel feature Mapping

1) $x = (x_1, x_2)^T$

$\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$

$K(x,z) = \phi(x) \cdot \phi(z)$

$= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \cdot (z_1^2, \sqrt{2}z_1z_2, z_2^2)$

$= (x_1z_1 + x_2z_2)^2 = (x \cdot z)^2$

corresponding kernel functions = $\boxed{(x_1z_1 + x_2z_2)^2}$

2) $K(x,z)$ $x, z \in R^2$

a) Mapping into feature space.

$(x_1, x_2) \rightarrow \phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

Total no. of multiplication needed for above feature mapping = $\underline{3}$

Similarly, for z mapping, we would need $\underline{3 \text{ multiplication}}$

To multiply in feature space we would need 3 multiplication (dot product)

Thus, total multiplication = 3+3+3 = $\boxed{9 \text{ Multiplication}}$

$\boxed{\text{Total Addition} = 2}$

b) Computing through kernel function.

$(x_1z_1 + x_2z_2)^2 \rightarrow 2+1 \rightarrow \boxed{3 \text{ Multiplication}}$

$\boxed{\text{Addition} = 1}$

2) **Perceptrons**

$$y = \phi(\Sigma x_i w_i + b) \qquad \phi(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

(2) AND.

Case 1. $x_1 = 0, x_2 = 0 \Rightarrow y = 0 \Rightarrow \phi(\Sigma x_i w_i + b) \leq 0$

$\Rightarrow x_1 w_1 + x_2 w_2 + b \leq 0$

$\Rightarrow \boxed{b \leq 0}$ as $x_1 = x_2 = 0$ — (1)

Case 2) $x_1 = 0, x_2 = 1$ AND $\rightarrow 0$

$\Rightarrow 0 \cdot w_1 + 1 \cdot w_2 + b \leq 0$

$w_2 + b \leq 0$ — (2)

Case 3) $x_1 = 1, x_2 = 0$ AND $\rightarrow 0$

$\Rightarrow 1 \cdot w_1 + 0 \cdot w_2 + b \leq 0$

$w_1 + b \leq 0$ — (3)

Case 4) $x_1 = 1, x_2 = 1$ AND $\rightarrow 1$

$\Rightarrow 1 \cdot w_1 + 1 \cdot w_2 + b \geq 0$

$\Rightarrow w_1 + w_2 + b \geq 0$ — (4)

Solving above 4 inequalities we get,

$b \leq 0, w_1 \leq -b, w_2 \leq -b, w_1 + w_2 \geq -b$

$\boxed{b = -1, w_1 = 2/3, w_2 = 2/3}$

(7) AND table

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

3) OR contingency table

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

4) Case 1) $x_1 = 0, x_2 = 0 \quad y = 0$

$$y = 0 \Rightarrow \phi(\Sigma w_i x_i + b) \leq 0$$

$$\Rightarrow \boxed{b \leq 0} \quad — ①$$

Case 2) $x_1 = 0, x_2 = 1 \quad y = 1$

$$\Rightarrow w_1 \cdot 0 + w_2 \cdot x_2 + b \geq 0$$

$$\Rightarrow b + w_2 \geq 0 \quad — ②$$

Case 3) $x_1 = 1, x_2 = 0, \quad y = 1$

$$\Rightarrow w_1 + 0 + b \geq 0$$

$$w_1 + b \geq 0 \quad — ③$$

Case 4) $x_1 = 1, x_2 = 1, y = 1$

$$\Rightarrow w_1 + w_2 + b \geq 0 \quad — ④$$

Solving above 4 inequalities, we get a range of values.

One set of values

$$\boxed{w_1 = 2, w_2 = 2, b = -1}$$

5)

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

XOR Table

6) Case 1) $x_1=0$, $x_2=0$ XOR$=0$

$y=0 \Rightarrow \phi(\Sigma x_i w_i + b) \leq 0$

$\Rightarrow x_1 w_1 + x_2 w_2 + b \leq 0$

$\boxed{b \leq 0}$ —— (1)

Case 2) $x_1=0$, $x_2=1$, XOR$=1$

$y=1 \Rightarrow \Sigma x_i w_i + b > 0$

$\Rightarrow 1 \cdot w_2 + b > 0$ —— (2)

Case 3) $x_1=1$, $x_2=0$, XOR$=1$

$\Rightarrow 1 \cdot w_1 + w_2 \cdot 0 + b > 0$

$w_1 + b > 0$ —— (3)

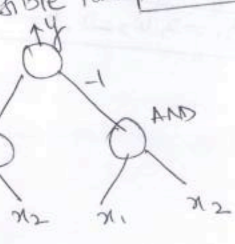
Case 4) $x_1=1$, $x_2=1$, XOR$=0$

$\Rightarrow 1 \cdot w_1 + 1 \cdot w_2 + b \leq 0$ —— (4)

$w_1 + w_2 + b \leq 0$

Above 4 equations are not possible to solve thus single layer perceptron is not possible for logical XOR. (2),(3),(4) contradicts (1).

However, two layer is possible, as $\boxed{\text{XOR} = \text{OR} - \text{AND}}$



3) Regression Theory

3.1) Linear Regression

q) $\{x_1, y_1\}, \ldots \{x_n, y_n\}$. $x \in R^M$.

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

a) Square loss error: error = misclassification $\Rightarrow f_i(x) \neq y_i$ for x_i

$J(w) = (y_i - f(x))$ for i^{th} sample

loss func — for all samples,

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f_i(x))^2$$

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

b) Partial derivation of $J(w)$ w.r.t w^k

$$\frac{\partial J(w)}{\partial w_k} = \frac{1}{n} \frac{\partial (\sum y_i - w^T x_i)^2}{\partial w_k} = \frac{2}{n}(y_i - w^T x_i) X - \cancel{x_k}$$

$$\boxed{= -\frac{2}{n} \sum (y_i - w^T x_i) x_i^k}$$

$$\Rightarrow \sum_{i=1}^{N} (y_i - w^T x_i) x_i^k$$

c) Gradient descent

$$w^k_{new} = w^k + \alpha \frac{\partial J(w)}{\partial w_k} \qquad \frac{\partial J(w)}{\partial w_k} = \sum_{i=1}^{n} x_i^k (y_i - w^T x_i) \quad \text{— (1)}$$

new param ← w^k_{new}; w^k → old params; α → step size

$$\boxed{w_{new}^k = w^k + \alpha \sum_{i=1}^{N} x_i^k (y_i - w^T x_i)} \quad \text{— from (1)}$$

do this till convergence.

2) $f(x) = \omega^T x$

a) Conditional likelihood L

$\cancel{L(\omega; x, y) = \prod P(y_i / x, \omega)}$

$$L(\omega; x, y) = \operatorname{argmax} \prod_{i=1}^{N} P\left(\frac{y_i}{x, \omega}\right)$$

$$P(y_i) = N(\omega^T x_i, \sigma^2)$$

$$P(y/x; \omega) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y - f(x;\omega)}{\sigma}\right)^2}$$

$$\Rightarrow \boxed{L(\omega; x, y) = \operatorname{argmax} \prod \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y - f(x;\omega)}{\sigma}\right)^2}}$$

b) log conditional likelihood.

$$\log(L(\omega; x, y)) = \operatorname{argmax}\left(\log \prod \frac{e^{-\frac{1}{2}\left(\frac{y - f(x,\omega)}{\sigma}\right)^2}}{\sqrt{2\pi\sigma^2}}\right)$$

as $\log(x \cdot y) = \log x + \log y$

$$\log(L(\omega; x, y)) = \sum \log \left(\frac{e^{-\frac{1}{2}\left(\frac{y - f(x,\omega)}{\sigma}\right)^2}}{\sqrt{2\pi\sigma^2}}\right)$$

$$\log(L(\omega; x, y)) = \sum \underbrace{\log \frac{1}{\sqrt{2\pi\sigma^2}}}_{C} + \sum \log e^{-\frac{1}{2}\left(\frac{y - f(x,\omega)}{\sigma}\right)^2}$$

$$= C - \frac{1}{2} \sum \left(\frac{y - f(x,\omega)}{\sigma}\right)^2$$

$$\boxed{\log(L(\omega; x, y)) = C - \frac{1}{2} \sum \left(\frac{y - f(x,\omega)}{\sigma}\right)^2}$$

$$C = \sum \log \frac{1}{\sqrt{2\pi\sigma^2}}$$

c) Maximizing the log likelihood

$\Rightarrow$ ~~taking~~ a derivative and putting it to zero

$$\frac{\partial \log L(\omega; x, y))}{\partial \omega} = \frac{\partial\left(c - \frac{1}{2}\sum\left(\frac{y - f(x,\omega)}{\sigma}\right)^2\right)}{\partial \omega}$$

$$= 0 - \frac{1}{2\sigma^2}\,\partial \frac{\sum(y - f(x,\omega))^2}{\partial \omega}$$

$$\frac{\partial \log L(\omega; x, y))}{\partial \omega} = \frac{-1}{2\sigma^2}\,\frac{\partial \sum_i (y - \omega^T x_i)^2}{\partial \omega}$$

$$= \frac{\cancel{-}1}{2\sigma^2}\,\sum_{i=1}^{N}\cancel{2}(y - \omega^T x_i)\,\cancel{-}x_i$$

$$\boxed{\frac{\partial \log L(\omega; x, y)}{\partial \omega} = \frac{\cancel{2}1}{\cancel{2}\sigma^2}\sum_{i=1}^{N}(y - \omega^T x_i)\,x_i}$$

Above function will give the derivative to zero at a point same as that of loss function proved earlier

so, Maximizing log likelihood is same as that of minimizing the least square error.

3) $y = f(x) + \epsilon$ $\quad$ ϵ : mean $= 0$, variance σ^2

$f(x) \downarrow$ defined by us $\quad$ $\epsilon \downarrow$ Noise

a)

$$E_D\left[\int_y \int_x (\cancel{h(x)} y - f(x))^2 p(y/x)\, p(x)\, dy\, dx\right]$$

point is x.

b) $E_D[(y - h(x))^2]$ $\qquad$ $\cancel{E(h(x))}$

$\cancel{l = E_D(h(x))}$ $\quad l = E_D(h(x))$

$$E[(\cancel{y} - y')^2] = E[(y - l + l - y')^2]$$

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

$$= E[(y-l)^2 + (h-\hat{y})^2 + 2[yl - l^2 + l\hat{y} - \hat{y}y]]$$

$$= E[(y-l)^2] + E[(h-\hat{y})^2] + 2(\cancel{E[yl]} - E[l^2] - E[l\hat{y}] - \cancel{E[\hat{y}y]})$$

$$\underbrace{E[(y-l)^2]}_{\text{Bais}} \quad \underbrace{E[(h-\hat{y})^2]}_{\text{Vaulane}} \quad 2(\dots) \to 0$$

Bais $\qquad$ Vaulane $+ \sigma^2$

$\sigma^2 \downarrow$ unavoidable error

3.2) Regularization.

$$L = \frac{1}{2}\sum_{i=1}^{N}(y_i - w^T x_i)^2$$ (original loss function)

a) $$L = \frac{1}{2}\sum_{i=1}^{N}(y - W^T x_i)^2 + \frac{\lambda}{2}||w||^2.$$ (loss function with regularization)

$$\frac{\partial L}{\partial w_k} = \frac{2}{2}\sum_{i=1}^{N}(y - W^T x_i)x_k + \frac{2 \times \lambda ||w_k||}{2}$$

$$\boxed{\frac{\partial L}{\partial w_k} = \sum_{i=1}^{N}(y - W^T x_i)x_k + \lambda ||w_k||}$$

b) ~~first~~ Algorithm is more likely to give a sparse matrix as it penalised the weight with high weightage of $w^{k,t}$.

$$w^{k,(t+1)} = w^{k,t} - \sum_{i=1}^{n}(y_i - w^T x_i)x_i^k - \lambda w^{k,(t)}$$

In second algo, it will ~~reduce~~ increase by λ. ~~t~~ however in other it will be by the magnitude of $w^{k,(t)}$ reduce

so, First Algorithm is likely to give sparse matrix with more w_i's set to zero

4.1) logistic Regression

$\{x^{(1)}, y^{(1)}\}, \ldots \{x^{(n)}, y^{(n)}\}$

1) logistic function.

$$f(x,w) = \frac{1}{1+e^{-a}}$$, where $a = \sum_{i=1}^{N} x^i w_i + \cancel{w_0}$, $N \to$ no. of feature.

2) conditional likelihood

$$P(Y=0/W) = \frac{1}{1+e^{(w_0 + \sum w_i x_i)}}$$

$$P(Y=1/W) = 1 - P(Y=0/W) = \frac{e^{(w_0 + \sum w_i x_i)}}{1+e^{w_0 + \sum w_i x_i}}$$

Conditional likelihood $= \prod P(Y^i | x^i, w)$ over all the samples/ training data

~~L(w; y, x) =~~

$$L(w;x,y) = \prod_i P(y^i | x^i, w)$$

$$L(w;x,y) = \arg\max_w \prod_i P(y^i | w, x^i)$$

3) ~~Cond~~ log conditional likelihood

$$\log[L(w;x,y)] = \log \prod_i P(y^i | x^i, w) \quad (\log ab = \log a + \log b)$$

$$= \sum_i \log P(y^i | x^i, w)$$

$$\log(L) = \sum_i y^i \log P(Y=1/x^i, w) + (1-y^i) \log P(Y=0/x^i, w)$$

$$= \sum_i y^i \log \frac{P(Y=Y x^i w)}{P(Y=0/x^i, w)} + \log P(Y^i=0/x^i, w)$$

$$= \sum_i y^i (w_0 + \sum_{j=1}^{n} w_j x_j^i) - \log(1 + e^{(w_0 + \sum_{j=1}^{n} w_j x_j^i)})$$

4) Derivative w.r.t w^k

$$\boxed{\frac{\partial L}{\partial w_k} = \sum_i x_k^i \left(y^i - \underbrace{P(y^i = 1 \mid x^i; w)}_{f(w^T x_i)}\right)}$$

5) Gradient descent

$$w_{new}^k = w^k + \alpha \frac{\partial L}{\partial w^k}$$

$$\boxed{w_{new}^k = w^k + \alpha \sum_{i=1}^{N} x_k^i (y^i - f(w^T x_i))}$$

6) Object = argmin $-l(w; x, y)$

Adding regularization:

Object = argmin $-l(w; x, y) + \frac{\lambda}{2} \|w\|^2$

~~Object = argmin~~

Gradient w.r.t w_i

$$\frac{\partial(\text{object})}{\partial w_i} = \frac{-\partial(l(w; x, y))}{\partial w_i} + \frac{\lambda \times \cancel{2}}{\cancel{2}} |w_i|$$

$$\boxed{\text{gradient w.r.t } w_i = \sum_{j=1}^{N} x_i^j (y^j - f(w^T x_j)) + \lambda |w_i|}$$

$$w_{new}^k = w^k + \alpha \frac{\partial(\text{object})}{\partial w_i}$$

$$= w^k + \alpha(\lambda |w_k|) + \sum_{j=1}^{N} x_k^j (y^j - f(w^T x^j))$$

4.3 Analysis of results (Programming logistic regression)

1) The Accuracy of the training set increased with the application of Regularization (lambda = 25).

   accuracy = 98.5075] without regularization
   # Mis classified Image = 15]

   accuracy = 98.7065] with L2 regularization
   # Mis classified Image = 13]

This is expected as penalizing the high weight features based on the training set would make the hypothesis much better i.e reduce overfitting. Likewise, we see that L2 regularization improves the overall accuracy of any training set. ~~However,~~ on test data, accuracy improved a little bit.

2) Mis classified Image count = 13 (with L2 regularization)
   figure attached