Page:
Date:

Q1

$$\frac{1}{n}\sum L(y_i, w^T x_i + b) = \frac{1}{n}\sum \log\left(1+\exp\left(-y_i(w^T x_i+b)\right)\right)$$

Let $\tilde{x}_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix}$ & $\theta = \begin{bmatrix} b \\ w \end{bmatrix}$

Then $w^T x_i + b = \theta^T \tilde{x}_i$

$$\therefore \quad = \frac{1}{n}\sum \log\left(1+\exp\left(-y_i\theta^T \tilde{x}_i\right)\right)$$

Above form is equal to the -ve log likelihood for logistic regression.

Q2

$$J(w,b) = \sum_{i=1}^{n} J_i(w,b) = \frac{1}{n}\sum L(y_i, w^T x_i + b) + \frac{\lambda}{n}\sum_{i=1}^{n}\|w\|^2$$

$$\therefore J_i(w,b) = \frac{L(y_i, w^T x_i + b)}{n} + \frac{\lambda}{n}\|w\|^2$$

& $L(y_i, w^T x_i + b) = \max\{0, 1-yt\} = \max\{0, 1-y_i(w^T x_i+b)\}$

$$\therefore J_i(w,b) = \frac{\max\{0, 1-\overbrace{yt}^{y_i(w^T x_i + b)}\}}{n} + \frac{\lambda}{n}\|w\|^2$$

~~$\frac{dJ_i(w,b)}{\partial w}$~~

When $y_i(w^T x_i + b) < 0$

$$J_i(w,b) = \frac{\lambda}{n}\|w\|^2$$

$$\frac{dJ_i(w,b)}{db} = 0$$

$$\& \quad \frac{dJ_i(w,b)}{dw} = \frac{2\lambda w}{n}.$$

$$\therefore u_i = \begin{bmatrix} 0 \\ \frac{2\lambda}{n} w \end{bmatrix}$$

When $y_i(w^T x_i + b) > 0$

$$J_i(w,b) = \frac{1-y_i(w^T x_i + b)}{n} + \frac{\lambda}{n}\|w\|^2$$

$$\frac{dJ_i(w,b)}{db} = \frac{-y_i}{n}$$

$$\frac{dJ_i(w,b)}{dw} = \frac{-y_i x_i}{n} + \frac{2w\lambda}{n}$$

$$\therefore u_i = \begin{bmatrix} -y_i/n \\ \frac{-y_i x_i}{n} + \frac{2w\lambda}{n} \end{bmatrix}$$

Page:
Date:

$\therefore$ If $y_i(\theta^T x_i) < 0$ If $y_i(\theta^T x_i) > 0$

$$u_i = \begin{bmatrix} 0 \\ \frac{2\lambda}{n}\theta[2:] \end{bmatrix} \qquad u_i = \begin{bmatrix} -y_i/n \\ \frac{-y_i}{n} x_i + \frac{2\lambda}{n}\theta[2:] \end{bmatrix}$$

d) The empirical rate of convergance is faster for stochastic sub gradient method relative to the sub gradient method because sub gradient took ~35 cycles to coverges where stochastic sub Gradient took ~25 cycles to converge.

Q3

a) $K(u,v) = (\langle u,v\rangle + 1)^3$

$$= \langle u,v\rangle^3 + 1 + 3\langle u,v\rangle^2 + 3\langle u,v\rangle$$

$$\langle u,v\rangle = \overbrace{u_1 v_1 + \ldots + u_d v_d}^{d \text{ terms}}$$

$$\langle u,v\rangle^2 = \cancel{u_1^2 v_1^2 +} (\Sigma u_i v_i)(\Sigma u_j v_j)$$

$$= \sum_{j=1}^{d}\sum_{i=1}^{d} u_i v_i u_j v_j$$

$$= \underbrace{u_1^2 v_1^2 + \ldots + u_d^2 v_d^2}_{d \text{ terms}} + \underbrace{2 u_1 u_2 v_1 v_2 + \ldots + 2 u_{(d-1)} u_d v_{(d-1)} v_d}_{\frac{d(d-1)}{2}}$$

$$\langle u,v\rangle^3 = \underbrace{u_1^3 v_1^3 + \ldots + u_d^3 v_d^3}_{d \text{ terms}} + 3\underbrace{(u_1^2 u_2 v_1^2 v_2 + \ldots + u_d^2 u_{d-1} v_d^2 v_{d-1})}_{d(d-1) \text{ terms}}$$

$$+ 6\underbrace{(u_1 u_2 u_3 v_1 v_2 v_3 + \ldots + u_{d-2} u_{d-1} u_d v_{d-2} v_{d-1} v_d)}_{\frac{d(d-1)(d-2)}{3!}}$$

$\therefore \phi(u) = \{ \; 1 \quad 7$

$$\therefore \phi(u) = [\, 1 \; \sqrt{3}u_1 \ldots \sqrt{3}u_d \; \sqrt{3}u_1^2 \ldots \sqrt{3}u_d^2 \; \sqrt{6}u_1u_2 \ldots \sqrt{6}\, u_{(d-1)} u_d$$

$$u_1^3 \ldots u_d^3 \; \sqrt{3}\, u_1^2 u_2 \ldots \sqrt{3}\, u_d^2 u_{d-1} \; \sqrt{6}\, u_1 u_2 u_3 \ldots \sqrt{6}\, u_{d-2} u_{d-1} u_d \,]$$

b) k_1 is SP kernel & k_2 is SP kernel

Lets prove $\underbrace{a_1 k_1 + a_2 k_2}_{k_3}$ is symmetric & PSD kernel.

$k_3(u,v) = a_1 k_1(u,v) + a_2 k_2(u,v)$

We know k_1 & k_2 are SP kernels $\therefore k_1(u,v) = k_1(v,u)$
& $k_2(u,v) = k_2(v,u)$

$$\therefore k_3(u,v) = a_1 k_1(v,u) + a_2 k_2(v,u)$$
$$= k_3(v,u)$$

$\therefore k_3$ is symmetric.

Now, to prove PSD,

$$\underbrace{\begin{bmatrix} a_1 k_1(x_1,x_1)+a_2 k_2(x_1,x_1) & \cdots & a_1 k_1(x_1,x_n)+a_2 k_2(x_1,x_n)\\ \vdots & & \vdots \\ a_1 k_1(x_n,x_1)+a_2 k_2(x_n,x_1) & \cdots & a_1 k_1(x_n,x_n)+a_2 k_2(x_n,x_n)\end{bmatrix}}_{K_3}$$

Above can be rewritten as summition of two matrices

ie
$$\begin{bmatrix} a_1 k_1(x_1,x_1) & \cdots & a_1 k_1(x_1,x_n)\\ \vdots & & \vdots \\ a_1 k_1(x_n,x_1) & & a_1 k_1(x_n,x_n)\end{bmatrix} + \begin{bmatrix} a_2 k_2(x_1,x_1) & \cdots & a_2 k_2(x_1,x_n)\\ \vdots & & \vdots \\ a_2 k_2(x_n,x_1) & & a_2 k_2(x_n,x_n)\end{bmatrix}$$

$$= a_1 \underbrace{\begin{bmatrix} k_1(x_1,x_1) & \cdots & k_1(x_1,x_n)\\ \vdots & & \vdots \\ k_1(x_n,x_1) & & k_1(x_n,x_n)\end{bmatrix}}_{K_1} + a_2 \underbrace{\begin{bmatrix} k_2(x_1,x_1) & \cdots & k_2(x_1,x_n)\\ \vdots & & \vdots \\ k_2(x_n,x_1) & & k_2(x_n,x_n)\end{bmatrix}}_{K_2}$$

$$=$$

$$= a_1 K_1 + a_2 K_2$$

$$K_3 = a_1 K_1 + a_2 K_2.$$

To prove K_3 is PSD,

$$x^T K_3 x = \underbrace{a_1}_{\geq 0 \text{ given}} \underbrace{x^T K_1 x}_{\geq 0\ K_1 \text{ PSD}} + \underbrace{a_2}_{\geq 0 \text{ given}} \underbrace{x^T K_2 x}_{\geq 0 \text{ as } K_2 \text{ is PSD}}$$

Page:
Date:

$\therefore \forall\, x^T K_3 x \geq 0$
or K_3 is PSD

$\therefore k_3$ is symmetric & PSD $\Rightarrow k_3$ is an IP kernel

c) k is inner product kernel.

$\therefore \langle u, v\rangle = \langle v, u\rangle$ — ①
& $\langle u, u\rangle \geq 0$ — ②

To prove: Symmetric.

$$\begin{bmatrix} \langle x_1, x_1\rangle & \langle x_1, x_2\rangle & \cdots & \langle x_1, x_n\rangle \\ \langle x_2, x_1\rangle & & & \vdots \\ \vdots & & & \vdots \\ \langle x_n, x_1\rangle & & & \langle x_n, x_n\rangle \end{bmatrix}$$

$\because \langle x_2, x_1\rangle = \langle x_1, x_2\rangle$ (from property ①)

$\therefore k$ is symmetric.

To prove PSD

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}^T \begin{bmatrix} \langle x_1, x_1\rangle & \cdots & \langle x_1, x_n\rangle \\ \vdots & & \vdots \\ \langle x_n, x_1\rangle & \cdots & \langle x_n, x_n\rangle \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$= \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}^T \left[y_1 \begin{bmatrix} \langle x_1, x_1\rangle \\ \vdots \\ \langle x_n, x_1\rangle \end{bmatrix} + \cdots + y_n \begin{bmatrix} \langle x_1, x_n\rangle \\ \vdots \\ \langle x_n, x_n\rangle \end{bmatrix} \right]$$

$$= y_1 y_1 \langle x_1, x_1 \rangle + \cdots + y_n y_1 \langle x_n, x_1 \rangle$$
$$+ y_1 y_2 \langle x_1, x_2 \rangle \qquad + y_n y_2 \langle x_n, x_2 \rangle$$
$$\vdots \qquad \vdots$$
$$+ y_1 y_n \langle x_1, x_n \rangle \qquad + y_n y_n \langle x_n, x_n \rangle$$

$$= \overbrace{y_1^2 \langle x_1, x_1 \rangle + y_2^2 \langle x_2, x_2 \rangle + \cdots + y_n^2 \langle x_n, x_n \rangle}^{\geq 0}$$
$$+ 2 y_1 y_2 \langle x_1, x_2 \rangle + \cdots + 2 y_n y_{n-1} \langle x_n, x_{n-1} \rangle$$

Now, since K is symmetric, ~~& the diagonal values are ≥ 0.~~

K can be written as

$$K = U \Lambda U^T$$

& K is PSD iff $\lambda_i \geq 0$ for each i.

& $\because$ we know that the diagonal elements of K are $[\langle x_i, x_i \rangle]_i \geq 0$ {using property ①}

$\therefore$ the eigenvalues of $K \geq 0$

or K is PSD.

Question_2.m

```
clear all;
close all;
clc;

rng(0); % in Matlab
load nuclear.mat;
lambda = .001;
theta = [1 1 1];
obj = zeros(100, 1);
for iter = 1:100
    [p, n] = size(x);
    x_new = [ones(n, 1)'; x];
    a = theta * x_new;
    slack = 1 - (y .* (a));
    obj(iter) = sum(slack(slack > 0)) / n + lambda/2 *
sum(theta(2:end) .^2);
    subGrad = subGradient(x, y, theta, lambda);
    theta = theta - 100/iter * subGrad;
    % stopping criteria
    if obj(iter) < 1
        break
    end
end

plot(obj(1:iter))
```

subgradient_m

```
function [subGrad ] = subGradient( x, y, theta, lambda)
    %UNTITLED Summary of this function goes here
    %   Detailed explanation goes here
    [p, n] = size(x);
    x_new = [ones(n, 1)'; x];
    a = theta * x_new;
    slack = 1 - (y .* (a));
    b = (slack > 0);
    Ji_1 = [ 0 (lambda / n) * theta(2:end)];
    Ji_2 = [-y/n ; 1/n *(-[y;y]  .* x) + (lambda/n * (ones(n, 1) *
theta(2:end)))'];
    subGrad = (Ji_2 * b')' + sum(1-b) * Ji_1;
end
```

Question_3.m

```
clear all;
close all;
clc;

rng(0); % in Matlab
load nuclear.mat;
```

```
lambda = .001;
theta = [1 1 1];
[p, n] = size(x);
obj = zeros(n * 20000, 1);
flag = 0;
for iter = 1:100
    rng(0)
    x = x(:, randperm(n));
    x_new = [ones(n, 1)'; x];
    for i = 1:n
        a = theta * x_new;
        slack = 1 - (y .* (a));
        obj(i + (iter-1)*n) = sum(slack(slack > 0)) / n + lambda/2 *
sum(theta(2:end) .^2);
        subGrad = subGradient(x(:,i), y(:,i), theta, lambda);
        theta = theta - 100/iter * subGrad;
        % stopping criteria
        if obj(i + (iter-1)*n) < 1
            flag = 1;
            break
        end
    end
    if flag == 1
        break
    end
end

plot(obj(1:(i + (iter-1)*n)))
```