

# Probability

i)  $W$  is gaussian with  $N(\mu, \sigma^2)$  &  $U$  is uniform random  $[a, b]$ .

Assume  $W$  &  $U$  are independent, what is  $E[Z]$  &  $\text{Var}[Z]$  if

$$Z = 3W + 2U?$$

$$\begin{aligned} E[Z] &= E[3W + 2U] \\ &= 3E[W] + 2E[U] \\ &= 3\mu + 2\frac{(a+b)}{2} \\ &= 3\mu + (a+b) \end{aligned}$$

$$\begin{aligned} \text{Var}[Z] &= \text{Var}[3W + 2U] \\ &= 3^2 \text{Var}[W] + 2^2 \text{Var}[U] \\ &= 9\sigma^2 + 4\frac{(b-a)^2}{12} \\ &= 9\sigma^2 + 2\frac{(b-a)^2}{3} \\ &= 9\sigma^2 + \frac{(b-a)^2}{3} \end{aligned}$$

		$Z$			
		$P(X, Y)$			
$X$	$T$				
		$a$	$b$	$c$	$d$
$F$	$0.1$	$0.2$	$0.1$	$0.1$	
	$0.1$	$0.1$	$0.2$	$0.1$	

a) Marginal distributions  $P_Y, P_Y(Y=a), P_Y(Y=b)$   
 $P_Y(Y=c), P_Y(Y=d)$

$$P_Y(Y=a) = 0.1 + 0.1 = 0.2$$

$$P_Y(Y=b) = 0.2 + 0.1 = 0.3$$

$$P_Y(Y=c) = 0.1 + 0.2 = 0.3$$

$$P_Y(Y=d) = 0.1 + 0.1 = 0.2$$

$Z$

<small>Note:</small> $E[W] = \mu$	$E[U] = \frac{(a+b)}{2}$
$\text{Var}(W) = \sigma^2$	$\text{Var}(U) = \frac{(b-a)^2}{12}$

$$\begin{aligned} b) P_T(X=1 \mid Y \in \{a, b, c, d\}) : \text{Prob} \\ \text{that } X \text{ is } T \text{ given that } Y \text{ takes } \{a, b, c, d\} \\ P_T(X=1 \mid Y \in \{a, b, c, d\}) = \frac{P_T(X=1, Y \in \{a, b, c, d\})}{P_T(Y \in \{a, b, c, d\})} \\ = \frac{P_T(X=1, Y=b) + P_T(X=1, Y=c) + P_T(X=1, Y=d)}{P_T(Y=b) + P_T(Y=c) + P_T(Y=d)} \end{aligned}$$

$$= \frac{0.2 + 0.1 + 0.1}{0.3 + 0.3 + 0.2}$$

$$= \frac{0.4}{0.8}$$

$$= 0.5$$

$Z$

## 2) Linear Algebra

i)  $A_k \in \mathbb{R}^{n \times n}$  for  $k = 1, \dots, k$  such that  $A_{ik} = A_{kj}^T \Leftrightarrow A_{ik}$  is symmetric. If all  $A_k$  have same set of eigenvectors  $u_1, u_2, \dots, u_n$  with corresponding eigenvalues  $\lambda_{ki}, \dots, \lambda_{kn}$  for each  $A_{ik}$ . Write Eigenvectors and their corresponding eigenvalues for.

$$a) C = \sum_{k=1}^K A_k$$

$$Av = \lambda v$$

$$\therefore A_k u_i = \lambda_{ki} u_i$$

$$C u_i = \left( \sum_{k=1}^K A_k \right) u_i = \sum_{k=1}^K (A_k u_i)$$

$$\text{Sub } A_k u_i = \lambda_{ki} u_i$$

$$(u_i = \sum_{k=1}^K (\lambda_{ki} u_i))$$

$$= \left( \sum_{k=1}^K \{\lambda_{ki}\} \right) u_i$$

$$(u_i = \lambda u_i)$$

$$\lambda = \sum_{k=1}^K \lambda_{ki} \Rightarrow \text{eigenvalues}$$

$$u_i \Rightarrow u_i \Rightarrow \text{eigenvectors}$$

$$\therefore \text{for } C = \sum_{k=1}^K A_k$$

$$\text{eigenvalues} \Rightarrow \sum_{k=1}^K \lambda_{ki} = \sum_{k=1}^K \lambda_{kn}$$

$$\text{eigenvectors} \Rightarrow u_1, \dots, u_n.$$

b)  $D = A_i^{-1} A_j A_i$  where  $i \neq j$  and  $i, j \in \{1, 2, \dots, n\}$ : Assume  $A_i$  invertible  
we assume  $A_i$  &  $A_j$  share same eigenvector  $u_i$

$$\because A_i u_m = \dim u_m ; A_j u_m = \dim u_m$$

for  $A_i^{-1} u_m \neq \frac{u_m}{\dim}$

$$\Rightarrow \dim A_i^{-1} \dim u_m$$

$$\Rightarrow \dim \frac{u_m}{\dim} \dim$$

$$\text{Matrix } D = A_i^{-1} A_j A_i$$

$$\text{Substitute } \Rightarrow A_i^{-1} A_j \dim u_m$$

$$\Rightarrow \dim A_i (A_j u_m)$$

$\therefore \lambda = \dim \Rightarrow \text{eigenvalue}$   
 $u_m \Rightarrow \text{eigenvector}$

$\therefore$  for  $D$

Eigen values

$\Rightarrow \dim: d_1, d_2, \dots, d_n$

Eigen vectors

$\Rightarrow u_m: u_1, \dots, u_n$ .

$$D u_m = \dim u_m$$

$$\dim: D u_m = \lambda u_m$$

2)  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $\text{Col}(A)$  - column space of  $A$ . For value of  $m$ , under what conditions on  $b$ ,  $\text{Col}(A)$  and  $\text{Span}(A)$  will equation  $Ax=b$  have.

a) no solution?

$Ax=b$  is solvable if & only if  $b \in \text{Col}(A)$

$\therefore$  for no solution

$b \notin \text{col}(A)$

b) Exactly one solution

$\Rightarrow Ax=b$  to have one solution

\*  $b$  must lie in  $\text{Col}(A)$

\*  $A$  must have full rank ( $\text{rank}(A) = n$ )  $\rightarrow$  columns of  $A$  be linearly independent.

$\therefore b \in \text{Col}(A)$  and  $\text{rank}(A) = n$

c) Infinitely many solutions

$\Rightarrow Ax=b$  to have infinite solutions

\*  $b$  must lie in  $\text{col}(A)$

\*  $A$  must have full rank ( $\text{rank}(A) < n$ )  $\rightarrow$  columns of  $A$  be linearly dependent.

$\therefore b \in \text{Col}(A)$  and  $\text{rank}(A) < n$

### 3) Matrix Calculus

Find derivative with respect to  $X$ . Follow convention that the derivative of scalar function  $f(x)$  with respect to  $X$  should have same dimension as  $X$ .

a)  $f(x) = \text{tr}(XX^T)$ , where  $X \in \mathbb{R}^{n \times n}$  &  $\text{tr}$  is trace of square matrix

$$\text{tr}(A) = \sum_{i=1}^n A_{ii}$$

$$\therefore \text{tr}(XX^T) = \sum_{i=1}^n \sum_{k=1}^n X_{ik}X_{ik} = \sum_{i=1}^n \sum_{k=1}^n X_{ik}X_{ik}$$

~~$$\frac{\partial}{\partial x} \text{tr}(XX^T) = 2X$$~~

$$\therefore f(x) = \sum_{i=1}^n \sum_{k=1}^n X_{ik}^2$$

$$\therefore \frac{\partial}{\partial x} f(x) = 2X$$

$\Sigma$

b)  $f(x) = a^T x b$  where  $X \in \mathbb{R}^{n \times n}$  and  $a \in \mathbb{R}^m$  &  $b \in \mathbb{R}^n$

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n a_i x_{ij} b_j$$

$$\frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x_{ij}} \left( \sum_{i=1}^m \sum_{j=1}^n a_i x_{ij} b_j \right) \\ = a_i b_j \Rightarrow a_i b_j^T \Rightarrow ab^T$$

$$\frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x} (a^T x b) = ab^T$$

$$= ab^T$$

$\Sigma$

$$c) f(x) = \|Xb\|^2, \text{ where } X \in \mathbb{R}^{m \times n} \text{ and } b \in \mathbb{R}^n$$

For norms:  $\|v\|_2 = \sqrt{\sum_{i=1}^n v_i^2}$  |  $\|v\|_2^2 = \sum_{i=1}^n v_i^2 = \bar{v}^T v$

But  $v = Xb$

Substitute:

$$\|Xb\|_2^2 = (Xb)^T (Xb) = b^T X^T X b$$

Since  $b^T X^T X b$  is scalar we can perform trace

$$f(x) = \text{tr}(b^T X^T X b) = \text{tr}(X b b^T X^T)$$

$$\frac{\partial}{\partial x} \text{tr}(X b b^T X^T) = \underline{\underline{2X b b^T}}$$

#### 4) MLE - MAP

$x \in \mathbb{R}$  is fixed and given. Assume  $\lambda \geq 0$  is scalar parameter, and:

$y \sim \text{Poisson}(\lambda)$

Where  $\text{Poisson}(\lambda)$  denotes the Poisson distribution with rate ( $\lambda$ ):

i) Maximum Likelihood Estimator.

a) PMF of  $y | \lambda$

$$p(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!}; y = 0, 1, 2, \dots, \lambda > 0$$

b) Assume  $N$  independent observations  $y_1, y_2, \dots, y_N$  are drawn, where  $y_n \sim \text{Poisson}(\lambda)$  for  $n = 1, 2, \dots, N$ . Derive joint PMF

$$p(y_1, y_2, \dots, y_N | \lambda).$$

$$\begin{aligned} p(y_1, y_2, \dots, y_N | \lambda) &= p(y_1 | \lambda) p(y_2 | \lambda) \dots p(y_N | \lambda) \\ &= \left( \frac{\lambda^{y_1} e^{-\lambda}}{y_1!} \right) \left( \frac{\lambda^{y_2} e^{-\lambda}}{y_2!} \right) \dots \left( \frac{\lambda^{y_N} e^{-\lambda}}{y_N!} \right) \\ &= \frac{\lambda^{(y_1+y_2+\dots+y_N)} e^{(-\lambda-\lambda-\dots-\lambda)}}{y_1! y_2! \dots y_N!} \\ &= \frac{\lambda^{\sum_{n=1}^N y_n} e^{-N\lambda}}{\prod_{n=1}^N y_n!} \end{aligned}$$

c) Write the negative log-likelihood function of joint PMF derived in the above and simplify it into the form that can be minimized (remove term that don't have  $\lambda$ )

$$\begin{aligned} L(\lambda) &= \log(p(y_1, y_2, \dots, y_N | \lambda)) \\ &= \log \left[ \frac{\lambda^{\sum_{n=1}^N y_n} e^{-N\lambda}}{\prod_{n=1}^N y_n!} \right] \end{aligned}$$

$$= \log \left[ \lambda^{\sum_{n=1}^N y_n} e^{-N\lambda} \right] - \log \left[ \prod_{n=1}^N y_n! \right]$$

$$\begin{aligned} \text{Using } \log(a-b) &= \log a + \log b \\ &= \log \left( \lambda^{\sum_{n=1}^N y_n} \right) + \log(e^{-N\lambda}) - \log \left[ \prod_{n=1}^N y_n! \right] \end{aligned}$$

$$\therefore \log(\lambda^{\sum_{n=1}^N y_n}) = \sum_{n=1}^N y_n \log(\lambda) \quad \Rightarrow \text{from } \log(a^b) = b \log(a)$$

$$\therefore \log(e^{-Nx}) = -Nx \quad \text{from } \log(e^x) = x$$

$$\therefore \log(\prod_{n=1}^N y_n!) = \sum_{n=1}^N \log(y_n!)$$

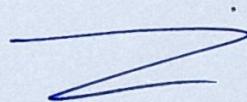
factoring in:

$$= \sum_{n=1}^N y_n \log(\lambda) - Nx - \sum_{n=1}^N \log(y_n!)$$

$$-L(\lambda) = -\sum_{n=1}^N y_n \log(\lambda) + Nx + \sum_{n=1}^N \log(y_n!)$$

ignore parts that don't have  $\lambda$  to minimize.

$$\therefore -L(\lambda) = -\sum_{n=1}^N y_n \log(\lambda) + Nx$$



2) Maximum-a-Posteriori (MAP) Estimation: Suppos  $\lambda \sim \text{Gamma}(\alpha, \beta)$  with PMF:

$$P(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \lambda > 0$$

a) Joint distribution  $P(y_1, \dots, y_N, \lambda)$  where each  $y_n \sim \text{Poisson}(\lambda)$  is drawn independently as above.

\* Incorporate prior distribution  $P(\lambda)$  and the likelihood  $P(y_1, \dots, y_N | \lambda)$

$$\begin{aligned} P(y_1, \dots, y_N, \lambda) &= P(y_1, \dots, y_N | \lambda) \cdot P(\lambda) \\ &= \left( \frac{\lambda^{\sum_{n=1}^N y_n} e^{-N\lambda}}{\prod_{n=1}^N y_n!} \right) \left( \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \right) \\ &= \left( \frac{\lambda^{\sum_{n=1}^N y_n + \alpha - 1} e^{-N\lambda - \beta\lambda}}{\prod_{n=1}^N y_n!} \right) \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \end{aligned}$$

b)  $\rightarrow$  Neg log of  $P(y_1, \dots, y_N, \lambda)$  and simplify into a form that can be minimize to find Map estimate  $\hat{\lambda}$ . (remove terms with  $\lambda$ )

We ignore  $\frac{\beta^\alpha}{\Gamma(\alpha)}$  and  $\prod_{n=1}^N y_n!$

$$\begin{aligned} \therefore -\log P(y_1, \dots, y_N) &= -\log \left[ \lambda^{\sum_{n=1}^N y_n + \alpha - 1} e^{-(N+\beta)\lambda} \right] \\ &= -\left( \sum_{n=1}^N y_n + \alpha - 1 \right) \log(\lambda) + (N+\beta)\lambda \end{aligned}$$

Formulate to find  $\lambda$ : (We derive -ve log with respect to  $\lambda = 0$ )

$$\frac{d}{d\lambda} \left( -\sum_{n=1}^N y_n + \alpha - 1 \right) \log(\lambda) = -\sum_{n=1}^N y_n + \alpha - 1$$

$$\frac{d}{d\lambda} (N+\beta)\lambda = N+\beta$$

$$\therefore -\sum_{n=1}^N y_n + \alpha - 1 + (N+\beta) = 0$$

$$\therefore \lambda = \frac{\sum_{n=1}^N y_n + \alpha - 1}{N+\beta}$$

$$\boxed{\lambda = \frac{\sum_{n=1}^N y_n + \alpha - 1}{N+\beta}}$$

Z

## 5. Linear Regression with Regularization

Data set N samples  $(x_i, y_i)$  where:

$$y_i = x_i^T w + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$$

and  $w \in \mathbb{R}^d$  is weight vector,  $\epsilon_i$  is Gaussian noise and  $x_i \in \mathbb{R}^d$

To enforce smoothness in the weights  $w$ , we introduce a regularizer based on differences between adjacent weights. This results in the following optimization problem

$$\min_w L(w) = \|y - Xw\|_2^2 + \lambda \|w\|_2^2 + M \|Dw\|_2^2$$

Where  $\lambda, M > 0$  are regularization parameters and  $\|Dw\|_2^2 = \sum_{i=2}^{d-1} (2w_i - w_{i-1} - w_{i+1})^2$ . Write D

① Find  $D \in \mathbb{R}^{(d-2) \times d}$  such that  $\|Dw\|_2^2 = \sum_{i=2}^{d-1} (2w_i - w_{i-1} - w_{i+1})^2$ . Write D explicitly. Note that  $\|\cdot\|_2$  denotes the usual  $l_2$  or Euclidean norm.

$$\|Dw\|_2^2 = \sum_{i=2}^{d-1} (2w_i - w_{i-1} - w_{i+1})^2$$

$$\sum_{i=1}^{d-2} (Dw)_i \Rightarrow 2w_i - w_{i-1} - w_{i+1}$$

$$\text{for } i=2 \Rightarrow (Dw)_1 = 2w_2 - w_1 - w_3 \Rightarrow \begin{bmatrix} -1 & 2 & -1 & \dots & 0 \\ 0 & 1 & 2 & -1 & 0 \end{bmatrix}$$

$$\text{for } i=3 \Rightarrow (Dw)_2 = 2w_3 - w_2 - w_4 \Rightarrow \begin{bmatrix} 0 & 1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \end{bmatrix}$$

$$\text{for } i=4 \Rightarrow (Dw)_3 = 2w_4 - w_3 - w_5 \Rightarrow \begin{bmatrix} 0 & 0 & -1 & 2 & -1 \end{bmatrix}$$

$$\therefore D = \begin{bmatrix} -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -1 & 2 & -1 \end{bmatrix}$$

② Derive the closed-form solution to  $\hat{w}$ , the minimize of  $L(w)$

$$L(w) = \|y - Xw\|_2^2 + \lambda \|w\|_2^2 + M \|Dw\|_2^2 \Rightarrow y^T y - 2y^T Xw + w^T (X^T X + M D^T) w$$

Expand

$$\|y - Xw\|_2^2 = (y - Xw)^T (y - Xw)$$

$$= y^T y - 2y^T Xw + w^T X^T Xw$$

$$\|w\|_2^2 = w^T w$$

$$M \|Dw\|_2^2 = M (Dw)^T D w$$

$$= w^T D^T D w$$

Find gradient

$$\text{Combine}$$

$$\Rightarrow y^T y - 2y^T Xw + (w^T X^T Xw + \lambda w^T w + M w^T D^T D w)$$

Find gradient.

$$\Rightarrow \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{x} \omega + \omega^T (\mathbf{x}^T \mathbf{x} + \lambda \mathbf{I} + \mu \mathbf{D}^T \mathbf{D}) \omega$$

$$\mathbf{y}^T \mathbf{y} \text{ is independent of } \omega \Rightarrow \text{constant} \therefore \nabla_{\omega}(\mathbf{y}^T \mathbf{y}) = 0$$

$$\begin{aligned}\nabla_{\omega}(-2\mathbf{y}^T \mathbf{x} \omega) &= \text{using } \nabla_{\omega}(\mathbf{b}^T \mathbf{A} \omega) = \mathbf{A}^T \mathbf{b} \\ &= -2\mathbf{x}^T \mathbf{y}\end{aligned}$$

$$\nabla_{\omega}(\omega^T \mathbf{x}^T \mathbf{x} \omega) \quad \text{using } \nabla_{\omega}(\omega^T \mathbf{A} \omega) = 2\mathbf{A} \omega$$

$$= 2\mathbf{x}^T \mathbf{x} \omega$$

$$\begin{aligned}\nabla_{\omega}(\lambda \omega^T \omega) &\quad \text{using } \nabla_{\omega}(\omega^T \omega) = 2\omega \\ &= \lambda 2\omega\end{aligned}$$

$$\begin{aligned}\nabla_{\omega}(M \omega^T \mathbf{D}^T \mathbf{D} \omega) &\quad \text{using } \nabla(\omega^T \mathbf{A} \omega) = 2\mathbf{A} \omega \\ &= M 2 \mathbf{D}^T \mathbf{D} \omega\end{aligned}$$

Combine

$$\begin{aligned}\nabla_{\omega} L(\omega) &= -2\mathbf{x}^T \mathbf{y} + 2\mathbf{x}^T \mathbf{x} \omega + \lambda 2\omega + M 2 \mathbf{D}^T \mathbf{D} \omega \\ &= 2 \left( \mathbf{x}^T \mathbf{x} \omega - \mathbf{x}^T \mathbf{y} + \lambda \omega + M \mathbf{D}^T \mathbf{D} \omega \right)\end{aligned}$$

Equal to 0:

$$\mathbf{x}^T \mathbf{x} \omega - \mathbf{x}^T \mathbf{y} + \lambda \omega + M \mathbf{D}^T \mathbf{D} \omega = 0$$

$$\mathbf{x}^T \mathbf{x} \omega + \lambda \omega + M \mathbf{D}^T \mathbf{D} \omega = \mathbf{x}^T \mathbf{y}$$

$$\therefore \omega^* = (\mathbf{x}^T \mathbf{x} + \lambda \mathbf{I} + M \mathbf{D}^T \mathbf{D})^{-1} \mathbf{x}^T \mathbf{y}$$

Z

③ Consider problem 1 without the Smart regularization

$$\min_{\omega} L_{\text{ridge}}(\omega) = \|y - X\omega\|_2^2 + \lambda \|\omega\|_2^2$$

The minimizer of  $L_{\text{ridge}}$  be  $\omega_{\text{ridge}}^*$ . Show that the "Smart" regularizer  $L$  factors by proving  $\|D\omega^*\|_2^2 \leq \|D\omega_{\text{ridge}}^*\|_2^2$ .

$$\|D\omega^*\|_2^2 \leq \|D\omega_{\text{ridge}}^*\|_2^2$$

$$\omega^* = L(\omega) = \|y - X\omega\|_2^2 + \lambda \|\omega\|_2^2 + M \|D\omega\|_2^2$$

$$\omega_{\text{ridge}}^* = L_{\text{ridge}}(\omega) = \|y - X\omega\|_2^2 + \lambda \|\omega\|_2^2$$

$$\text{Loss function: } L(\omega^*) = \|y - X\omega^*\|_2^2 + \lambda \|\omega^*\|_2^2 + M \|D\omega^*\|_2^2$$

$$L_{\text{ridge}}(\omega_{\text{ridge}}^*) = \|y - X\omega_{\text{ridge}}^*\|_2^2 + \lambda \|\omega_{\text{ridge}}^*\|_2^2$$

Compare

$$L(\omega^*) \leq L(\omega_{\text{ridge}}^*)$$

$$\therefore \|y - X\omega^*\|_2^2 + \lambda \|\omega^*\|_2^2 + M \|D\omega^*\|_2^2 \leq \|y - X\omega_{\text{ridge}}^*\|_2^2 + \lambda \|\omega_{\text{ridge}}^*\|_2^2 + M \|D\omega_{\text{ridge}}^*\|_2^2$$

at  $\omega_{\text{ridge}}^*$   $\Rightarrow$  the smoothness term  $M \|D\omega_{\text{ridge}}^*\|_2^2$  is not minimized since the ridge loss  $L_{\text{ridge}}(\omega)$  doesn't penalize smoothness.  $\top$

$$\therefore M \|D\omega^*\|_2^2 \leq M \|D\omega_{\text{ridge}}^*\|_2^2$$

since  $M > 0$ ;

$$\therefore \|D\omega^*\|_2^2 \leq \|D\omega_{\text{ridge}}^*\|_2^2$$

⑩ How smooth regularization affects the bias and variance of the learned model  $w$ .

Smooth regularization increases the bias because it forces the weights to vary less and leading to potential underfitting. Whereas it reduces the variance and this makes it more robust to noise.

## 6) Online Update

In linear regression we observe variable  $y$ ;

$$y = w^T x + \epsilon$$

where  $\epsilon \sim N(0, \sigma^2)$  and  $x, w \in \mathbb{R}^d$ . In MLE of  $w$  is

$$\hat{w} = (x^T x)^{-1} x^T y$$

If we get new observation  $(\tilde{x}, \tilde{y})$  and want to compute the updated MLE  $\hat{w}_{\text{new}}$  after adding observation.  $X_{\text{new}} = \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \in \mathbb{R}^{N+1 \times d}$  and  $y_{\text{new}} = \begin{pmatrix} y \\ \tilde{y} \end{pmatrix} \in \mathbb{R}^{N+1}$   
 If we simply apply  $\hat{w}_{\text{new}} = (x^T x_{\text{new}})^{-1} x^T y_{\text{new}}$  we will need to solve  $\hat{w}_{\text{new}}$ . This can be solved  $(x^T x_{\text{new}})$  by low rank correction of  $(x^T x)$ .

Note  $x_{\text{new}}^T x_{\text{new}} = x^T x + \tilde{x} \tilde{x}^T$  Using Sherman-Morrison-Woodbury formula we can show that:

$$(x_{\text{new}}^T x_{\text{new}})^{-1} = (x^T x)^{-1} - \frac{(x^T x)^{-1} \tilde{x} \tilde{x}^T (x^T x)^{-1}}{1 + \tilde{x}^T (x^T x)^{-1} \tilde{x}} \quad \dots \quad (3)$$

Since  $x_{\text{new}}^T y_{\text{new}} = x^T y + \tilde{x} \tilde{y}$ , we have

$$\begin{aligned} \hat{w}_{\text{new}} &= (x_{\text{new}}^T x_{\text{new}})^{-1} x_{\text{new}}^T y_{\text{new}} \\ &= \left[ (x^T x)^{-1} - \frac{(x^T x)^{-1} \tilde{x} \tilde{x}^T (x^T x)^{-1}}{1 + \tilde{x}^T (x^T x)^{-1} \tilde{x}} \right] (x^T y + \tilde{x} \tilde{y}) \quad \dots \quad (4) \end{aligned}$$

i) Derive  $\hat{w}_{\text{new}}$  using Sherman-Morrison formula. Your formula should express  $\hat{w}_{\text{new}}$  explicitly in terms of  $\hat{w}$

$$\hat{w}_{\text{new}} = (x^T x)^{-1} x^T y + (x^T x)^{-1} \tilde{x} \tilde{y} - \frac{(x^T x)^{-1} \tilde{x} \tilde{x}^T (x^T x)^{-1} (x^T y + \tilde{x} \tilde{y})}{1 + \tilde{x}^T (x^T x)^{-1} \tilde{x}}$$

$$\text{since: } \tilde{w} = (x^T x)^{-1} x^T y \\ = \hat{w} + (x^T x)^{-1} \tilde{x} \tilde{y} - \frac{(x^T x)^{-1} \tilde{x} \tilde{x}^T \hat{w} + (x^T x)^{-1} \tilde{x} \tilde{x}^T (x^T x)^{-1} \hat{w}}{1 + \tilde{x}^T (x^T x)^{-1} \tilde{x}}$$

Factoring out  $\tilde{x}^T \hat{w}$

$$\hat{w}_{\text{new}} = \hat{w} + \frac{(x^T x)^{-1} \tilde{x}^T \hat{w}}{1 + \tilde{x}^T (x^T x)^{-1} \tilde{x}} (\tilde{y} - \tilde{x}^T \hat{w})$$

② Initialize  $K = (X^T X)^{-1}$  and  $\omega = \hat{\omega}$  Use 3 & 4 to show that we can get  $K = (X_{\text{new}}^T X_{\text{new}})^{-1}$  and  $\omega = \hat{\omega}_{\text{new}}$  after running the following.

### Algorithm 1: Online update

1. Receive observation  $(\tilde{x}, \tilde{y})$
2. Set  $\epsilon \leftarrow K \tilde{x} / (1 + \tilde{x}^T K \tilde{x})$
3. Set  $\omega \leftarrow \omega + \epsilon (\tilde{y} - \tilde{x}^T \omega)$
4. Set  $K \leftarrow K - \epsilon \tilde{x}^T K$

$$\hat{\omega}_{\text{new}} = \hat{\omega} + \frac{(X^T X)^{-1} \tilde{x}}{1 + \tilde{x}^T (X^T X)^{-1} \tilde{x}} \frac{(\tilde{y} - \tilde{x}^T \omega)}{\epsilon} \quad \mid \epsilon = \frac{K \tilde{x}}{1 + \tilde{x}^T K \tilde{x}}$$

since  $\omega = \hat{\omega}$  and  $\omega = \hat{\omega}_{\text{new}}$

$$\omega = \omega + \epsilon (\tilde{y} - \tilde{x}^T \omega)$$

$$K_{\text{new}} = X_{\text{new}}^T X_{\text{new}} \quad \text{from (3)}$$

$$K_{\text{new}} = K - \frac{K \tilde{x} \tilde{x}^T K}{1 + \tilde{x}^T K \tilde{x}} = K - \left( \frac{K \tilde{x}}{1 + \tilde{x}^T K \tilde{x}} \right) \tilde{x}^T K$$

$$\therefore = K - \epsilon \tilde{x}^T K$$

Z

③ Assume that  $n$  new data points arrive in sequential order and we would like to update  $\hat{w}$  every time a new data point arrives. Consider the following two methods.

~~Method A~~

Compare the total complexity computation (in terms of no. of multiplications) of methods a and b by showing the computational complexity of initialization complexity of  $n$  updates. Answers should be in terms of  $N, d$ , and  $n$ . Which method is efficient?

a) First initialize  $\hat{w}$  using the original  $N$  data points. Then algorithm 1 once each of the new data point.

i) Initialize

$$\hat{w} = (X^T X)^{-1} X^T y$$

$$[X^T X] = d \left[ \begin{smallmatrix} N & d \\ d & d \end{smallmatrix} \right]$$

$$= O(Nd^2)$$

$$K = (X^T X)^{-1} = O(d^3)$$

$$X^T y = d \left[ \begin{smallmatrix} N \\ d \end{smallmatrix} \right] = [d]$$

$$= O(Nd)$$

$$(X^T X)^{-1} X^T y = d \left[ \begin{smallmatrix} d \\ d \end{smallmatrix} \right] = [d]$$

$$= O(d^2)$$

$$\Rightarrow Nd^2 + d^3 + d^2$$

$$= O(Nd^2 + d^3)$$

Update

$$ii) \hat{w} = \frac{K \hat{x}}{1 + \hat{x}^T K \hat{x}}$$

$$K \hat{x} = d \left[ \begin{smallmatrix} d \\ d \end{smallmatrix} \right] [d] = O(d^2)$$

$$= O(d^3)$$

$$\hat{x}^T K \hat{x} = [d] [d]$$

$$= O(d)$$

$$1 + \hat{x}^T K \hat{x} = O(1)$$

$$= O(d^2 + d + 1) = O(d^2)$$

Updating  $K$

$$\hat{x} + \epsilon (y - \hat{x}^T \hat{w})$$

$$\hat{x}^T \hat{w} = [d] [d]$$

$$= O(d^2)$$

Updating  $K$

$$\hat{x}^T K = [d] [d] [d]$$

$$= O(d^2)$$

Total single update

$$= O(d + d^2 + d^2)$$

$$= O(d^2)$$

$\Rightarrow$  Total cost for update

$$O(Nd^2 + d^3 + nd^2)$$

$\sum$

Method b:

$$\tilde{w}_{\text{new}} = (X_{\text{new}}^T X_{\text{new}})^{-1} X_{\text{new}}^T y_{\text{new}}$$

For  $i^{\text{th}}$  update, cost  $O((N+i)d^2)$

for  $n$  values:

$$\sum_{i=1}^n O((N+i)d^2) = O\left(nd^2 + \frac{n(n+1)}{2}d^2\right) = O(nd^2 + n^2d^2)$$

$$(X_{\text{new}}^T X_{\text{new}})^{-1}$$

↳ for inverse of a  $d \times d$   $\Rightarrow O(d^3)$

↳ for  $n$  updates  $\Rightarrow O(nd^3)$

$$(X_{\text{new}}^T y_{\text{new}})$$

↳ for each update  $O((N+i)d)$

↳ for  $n$  updates.

$$\sum_{i=1}^n O((N+i)d) = O\left(nd + \frac{n(n+1)}{2}d\right) = O(nd + n^2d)$$

↳  $\tilde{w}_{\text{new}} = O(d^2)$  per update

↳ for  $n$  updates  $= O(nd^2)$

Total cost:

$$O(nd^2 + n^2d^2 + nd^3 + nd + n^2d)$$

$$= O(n^2d^2 + nd^3)$$

Comparison

↳ Method a is efficient since its complexity scales linearly for larger values of  $n$ , for update  $O(nd^2)$  while for method b, it scales quadratically with  $n$ :  $O(n^2d^2)$ .