[Recursive Least Square]

$x_i = \{3, 4, 6, 10, \cdots, 22\}$

recursive mean $\quad x_i \in \mathbb{R}^1$

$i : 1 \sim N$

$$\bar{x}_N = \frac{x_1 + x_2 + \cdots + x_N}{N} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

x_{N+1}

$$\text{new mean} = \overline{x_{N+1}} = \frac{x_1 + x_2 + \cdots + x_N + x_{N+1}}{N+1} = \frac{1}{n+1}\sum_{i=1}^{N+1} x_i$$

We have to know all the data samples // memory issue.

How to save memory? $\Rightarrow$ Just memorize the previous mean value.

$$\overline{X_{n+1}} = \frac{x_1 + x_2 + \cdots + x_n + x_{n+1}}{n+1}$$

$$= \frac{\frac{(x_1 + x_2 + \cdots + x_n)}{n} \cdot n + x_{n+1}}{n+1}$$

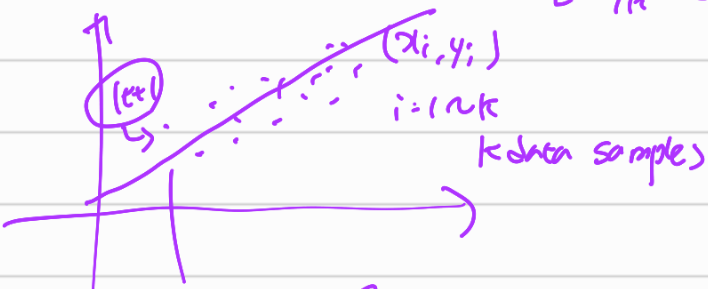
$$= \frac{\overline{x_n} \cdot n + x_{n+1}}{n+1}$$

(recursive estimation
재귀추정.)

: Save the information at k, and use it to infer the new information at $k+1$ from it.

For line fitting.

$$A_k = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_k & 1 \end{bmatrix} \quad b_k = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \quad x_k = \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix}$$



$y = \alpha x + \beta$

$$x_k^* = (A_k^T A_k)^{-1} A_k^T b_k$$

$$x_{k+1}^* = (A_{k+1}^T A_{k+1})^{-1} A_{k+1}^T b_{k+1}$$

$$A_k = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \\ \vdots \\ a_k^T \end{bmatrix} \quad A_{k+1} = \begin{bmatrix} a_1^T \\ \vdots \\ a_k^T \\ a_{k+1}^T \end{bmatrix} \quad b_k = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \quad b_{k+1} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ y_{k+1} \end{bmatrix}$$

$$A_{k+1} = \begin{bmatrix} A_k \\ a_{k+1}^T \end{bmatrix} \qquad = \begin{bmatrix} Y_k \\ y_{k+1} \end{bmatrix}$$

Can we obtain x_{k+1}^* from x_k^* and (a_{k+1}^T, y_{k+1})

$\hookrightarrow (x_{k+1}, y_{k+1})$ k data

x_{k+1}^*

$$= (A_{k+1}^T A_{k+1})^{-1} A_{k+1} b_{k+1}$$

$$(A_{k+1}^T A_{k+1})^{-1} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_k & x_{k+1} \\ 1 & 1 & 1 & & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_k & 1 \\ x_{k+1} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_1 & a_2 & \cdots & a_k & a_{k+1} \end{bmatrix} \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_{k+1}^T \end{bmatrix}$$

$$= \begin{bmatrix} A_k^T & a_{k+1} \end{bmatrix} \begin{bmatrix} A_k \\ a_{k+1}^T \end{bmatrix}$$

$$= (A_k^T A_k + a_{k+1} a_{k+1}^T)^{-1}$$

$$(A_{k+1}{}^T A_{k+1})^{-1} = (A_k{}^T A_k + a_{k+1} a_{k+1}{}^T)^{-1}$$

Matrix Inversion Lemma (= Woodbury matrix identity
= Sherman-Morrison Formula)

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$ A, D are invertible.

we define E, F

$$E = D - CA^{-1}B$$
$$F = A - BD^{-1}C$$

and assume that they are invertible.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} A^{-1} + A^{-1}BE^{-1}CA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}CA^{-1} & E^{-1} \end{bmatrix} \cdots ①$$

$$= \begin{bmatrix} I + BE^{-1}CA^{-1} - BE^{-1}CA^{-1} & -BE^{-1} + BE^{-1} \\ CA^{-1} + CA^{-1}BE^{-1}CA^{-1} - DE^{-1}CA^{-1} & -CA^{-1}BE^{-1} + DE^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} I & O \\ CA^{-1} + (CA^{-1}B - D)E^{-1}CA^{-1} & (-CA^{-1}B + D)E^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} I & O \\ CA^{-1} - EE^{-1}CA^{-1} & EE^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} I & O \\ O & I \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} F^{-1} & -A^{-1}BE^{-1} \\ -D^{-1}CF^{-1} & E^{-1} \end{bmatrix}$$ or ②

$$= \begin{bmatrix} AF^{-1} - BD^{-1}CF^{-1} & -BE^{-1} + BE^{-1} \\ CF^{-1} - CF^{-1} & -CA^{-1}BE^{-1} + DE^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} (A - BD^{-1}C)F^{-1} & 0 \\ 0 & (-CA^{-1}B + D)E^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} FF^{-1} & 0 \\ 0 & EE^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BE^{-1}CA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}CA^{-1} & E^{-1} \end{bmatrix}$$ or ①

$$= \begin{bmatrix} F^{-1} & -A^{-1}BE^{-1} \\ -D^{-1}CF^{-1} & E^{-1} \end{bmatrix}$$ or ②

① = ② $\quad A^{-1} + A^{-1}BE^{-1}CA^{-1} = F^{-1}$

$\underline{-E^{-1}CA^{-1} = -D^{-1}CF^{-1}}$

$E = D - CA^{-1}B$

$F = A - BD^{-1}C$

$$CA^{-1}F = ED^{-1}C$$

$$CA^{-1}(A - BD^{-1}C) = (D - CA^{-1}B)D^{-1}C$$

$$C - CA^{-1}BD^{-1}C = C - CA^{-1}BD^{-1}C$$

$$A^{-1} + A^{-1}BE^{-1}CA^{-1} = F^{-1}$$

$$A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} = (A - BD^{-1}C)^{-1}$$

$$\boxed{(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}}$$

matrix inversion lemma.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \Rightarrow \begin{bmatrix} A & u \\ v^T & 1 \end{bmatrix}$$

$$\begin{aligned}
(A - BD^{-1}C) = (A - uv^T)^{-1} \\
&= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \\
&= A^{-1} + A^{-1}u(1 - v^TA^{-1}u)^{-1}v^TA^{-1} \\
&= A^{-1} + \frac{A^{-1}uv^TA^{-1}}{1 - v^TA^{-1}u}
\end{aligned}$$

$$\underbrace{(A_{k+1}^T A_{k+1})^{-1}}_{P_{k+1}} = (A_k^T A_k + a_{k+1}a_{k+1}^T)^{-1}$$

$$= \underbrace{(A_k^T A_k)^{-1}}_{P_k} - \frac{(A_k^T A_k)^{-1} a_{k+1} a_{k+1}^T (A_k^T A_k)^{-1}}{1 - a_{k+1}^T (A_k^T A_k)^{-1} a_{k+1}}$$

$$P_{k+1} = P_k - \frac{P_k a_{k+1} a_{k+1}^T P_k}{1 - a_{k+1}^T P_k a_{k+1}} \checkmark$$

$$x_{k+1}^* = (A_{k+1}^T A_{k+1})^{-1} A_{k+1}^T b_{k+1}$$

$$= \left(P_k - \frac{P_k a_{k+1} a_{k+1}^T P_k}{1 - a_{k+1}^T P_k a_{k+1}}\right) \begin{bmatrix} A_k^T & a_{k+1} \end{bmatrix} \begin{bmatrix} Y_k \\ y_{k+1} \end{bmatrix}$$

$$\Downarrow$$

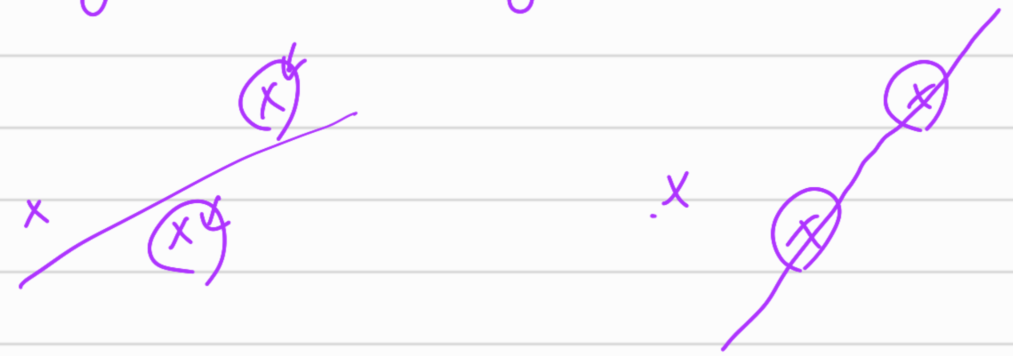
$$= \left(\qquad \text{"} \qquad \right) \left[A_k^T Y_k + a_{k+1} y_{k+1} \right]$$

$$= x_k^* - \frac{P_k a_{k+1} a_{k+1}^T}{1 - a_{k+1}^T P_k a_{k+1}} x_k^* + P_{k+1} a_{k+1} y_{k+1} \checkmark$$

$$P_k A_k^T Y_k = (A_k^T A_k)^{-1} A_k^T Y_k = x_k^*$$

Recursive LS.

Weighted Least Square.



$v_i v_j^T = 0$

$\Rightarrow$ multiple sensors.

$\underset{\sim}{b} = Ax + \underset{\sim}{v}$ noise $\qquad R = E(vv^T) = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_k^2 \end{bmatrix}$

$$\text{cost function } J = \frac{e_1^2}{\sigma_1^2} + \dots + \frac{e_k^2}{\sigma_k^2}$$

$$= e^T R^{-1} e$$

$$= (b - Ax)^T \underset{\sim}{R^{-1}} (b - Ax)$$

$$\frac{dJ}{dx} = 0$$

$$x^* = (A^T \underset{\sim}{R^{-1}} A)^{-1} A^T \underset{\sim}{R^{-1}} b$$

Recursive LS + Weighted LS. $\Rightarrow$ Kalman filter

KF, LQR

$\boxed{\text{past}} \rightarrow$ present estimation   KF

1) recursive weighted LS

2) constrainted optimization
$$\|Ax - b\|^2 \quad \text{s.t. } Cx = d$$

3) Dynamic programming.

present $\leftarrow \boxed{\text{future}}$ LQR

action $u(t)$