Review:

Constrained Least Squares:

- $\tilde{y}_1 = H_1 x + v \longrightarrow$ measurements to fit
- $\tilde{y}_2 = H_2 x \longrightarrow$ constraint.

$$H_1 = \underbrace{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}_{\text{tall}} \in \mathbb{R}^{m_1 \times n} \qquad H_2 = \begin{bmatrix} \underline{\qquad\qquad} \end{bmatrix}_{\text{fat.}} \in \mathbb{R}^{m_2 \times n}$$

$m_1 \geq n$ $\qquad\qquad$ $m_2 < n$

$$\min_{\hat{x}} \; J = \frac{1}{2} e_1^T W_1 e_1 = \frac{1}{2}(\tilde{y}_1 - H_1 \hat{x})^T W_1 (\tilde{y}_1 - H_1 \hat{x})$$

s.t. $\tilde{y}_2 = H_2 \hat{x}$

optimal soln:

$$\bar{x} = (H_1^T W_1 H_1)^{-1} H_1^T W_1 \tilde{y}_1$$

unconstrained solution

$$\hat{x} = \bar{x} + K(\tilde{y}_2 - H_2 \bar{x})$$

unconst soln

optimal gain

actual meas

prediction of $\tilde{y}_2$ from $\bar{x}$

diff. between the meas. & prediction

where

$$K = (H_1^T W_1 H_1)^{-1} H_2^T \left(H_2 (H_1^T W_1 H_1)^{-1} H_2^T\right)^{-1}$$

SEQUENTIAL (BATCH) LS ESTIMATION:

$\tilde{y}_1 = H_1 x + v_1$ } → first batch $H_1 \in \mathbb{R}^{m_1 \times n}$ $m_1 \geq n$

$\tilde{y}_2 = H_2 x + v_2$ } → second batch $H_2 \in \mathbb{R}^{m_2 \times n}$

m_2 can be any size but in practice m_2 will be small

Don't have all data initially... Comes in two batches...

Note: m_2 is small → gain computational advantage

first: $\hat{x}_1 = (H_1^T W_1 H_1)^{-1} H_1^T W_1 \tilde{y}_1$

initial estimate → want to take advantage of initial estimate.

now we add batch 2...

want to solve for $\hat{x}_2$ ← best fit for all data.

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}}_{H} x + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}$$

$(A+B)^{-1} \neq A^{-1} + B^{-1}$

$\frac{1}{x+y} \neq \frac{1}{x} + \frac{1}{y}$

$$\hat{x}_2 = (H^T W H)^{-1} H^T W \tilde{y}$$

$$= \left([H_1^T H_2^T] \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \right)^{-1} [H_1^T H_2^T] \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{bmatrix}$$

$$= \left(H_1^T W_1 H_1 + H_2^T W_2 H_2 \right)^{-1} \left[H_1^T W_1 \tilde{y}_1 + H_2^T W_2 \tilde{y}_2 \right]$$

matrix inverses are computationally expensive and slower for large operations.

Define:

$P_1 = (H_1^T W_1 H_1)^{-1}$

$P_2 = (H_1^T W_1 H_1 + H_2^T W_2 H_2)^{-1}$

$$P_1^{-1} = P_2^{-1} - H_2^T W_2 H_2$$

$$P_2^{-1} = (\underbrace{H_1^T W_1 H_1}_{P_1^{-1}} + H_2^T W_2 H_2)$$

$$(*) \quad P_1^{-1} = P_2^{-1} - H_2^T W_2 H_2$$

$$\hat{x}_1 = P_1 H_1^T W_1 \tilde{y}_1$$

$$\Rightarrow P_1^{-1} \hat{x}_1 = H_1^T W_1 \tilde{y}_1$$

$$(*) \Rightarrow (P_2^{-1} - H_2^T W_2 H_2) \hat{x}_1 = H_1^T W_1 \tilde{y}_1$$

$$\hat{x}_2 = P_2 (H_1^T W_1 \tilde{y}_1 + H_2^T W_2 \tilde{y}_2)$$

$$= P_2 ((P_2^{-1} - H_2^T W_2 H_2) \hat{x}_1 + H_2^T W_2 \tilde{y}_2)$$

$$= \hat{x}_1 - P_2 H_2^T W_2 H_2 \hat{x}_1 + P_2 H_2^T W_2 \tilde{y}_2$$

$$\hat{x}_2 = \underset{\text{initial est.}}{\hat{x}_1} + \underbrace{P_2 H_2^T W_2}_{\substack{K_2 \\ \text{gain}}} (\underset{\substack{\text{actual}\\ \text{new meas}}}{\tilde{y}_2} - \underset{\substack{\text{prediction for new meas.,}\\ \text{based on init est.}}}{H_2 \hat{x}_1})$$

diff between pred. & meas

$$K_2 = P_2 H_2^T W_2$$

General Rule:

$$\boxed{\begin{aligned} \hat{x}_{k+1} &= \hat{x}_k + K_{k+1} (\tilde{y}_{k+1} - H_{k+1} \hat{x}_k) \\ \text{where } K_{k+1} &= P_{k+1} H_{k+1}^T W_{k+1} \\ P_{k+1}^{-1} &= P_k^{-1} + H_{k+1}^T W_{k+1} H_{k+1} \end{aligned}}$$

still need to compute.

$$\boxed{P_{k+1} = \left(P_k^{-1} + H_{k+1}^T W_{k+1} H_{k+1}\right)^{-1}}$$

$x \in \mathbb{R}^n$

P_{k+1}: $\mathbb{R}^{n\times n}$ — still inverting an $n\times n$ matrix

P_k^{-1}: $\mathbb{R}^{n\times n}$

$[H_{k+1}^T][W_{k+1}][H_{k+1}]$ — low rank if only a few new meas.

scalar — if only 1 new meas

$$\begin{bmatrix} \tilde{y}_k \\ \tilde{y}_{k+1} \end{bmatrix} = \begin{bmatrix} H_k \\ H_{k+1} \end{bmatrix} x$$

Woodbury Matrix Identity:

Matrix Inversion Lemma.

(Rank 1): Sherman Morrison Formula

worth memorizing.

Wikipedia:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

low rank

$$([A] + [U][C][V])^{-1} = [A]^{-1} - [A^{-1}U]\left([C]^{-1} + [V][A^{-1}U]\right)^{-1}[VA^{-1}]$$

small

trading inverting a big matrix

assuming we've previously computed A^{-1}

for inverting 2 small matrices

TO CHECK:

$$(A+UCV)(A+UCV)^{-1} = (A+UCV)\left(A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}\right)$$

(see the book)

$$= I$$

$\underbrace{(A}_{\text{full rank}}+\underbrace{B)}_{\text{low rank}}^{-1}$: natural Searle Identities $\quad A + ucv^T$

GOING BACK TO LS...

$$P_{k+1} = \left(P_k^{-1} + H_{k+1}^T W_{k+1} H_{k+1}\right)^{-1}$$

applying Woodbury MI

$$= P_k - P_k H_{k+1}^T \big(\underbrace{W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}^T}_{\text{small.}}\big)^{-1} H_{k+1} P_k$$

$$K_{k+1} = P_{k+1} H_{k+1}^T W_{k+1}$$

$$= \left(P_k - P_k H_{k+1}^T \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}^T\right)^{-1} H_{k+1} P_k\right) H_{k+1}^T W_{k+1}$$

$$= P_k H_{k+1}^T \left(I - \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}\right)^{-1} H_{k+1} P_k H_{k+1}^T\right) W_{k+1}$$

$$= P_k H_{k+1}^T \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}\right)^{-1} \underbrace{\left(W_{k+1}^{-1} + \overset{0}{\cancel{H_{k+1} P_k H_{k+1}}} - \overset{0}{\cancel{H_{k+1} P_k H_{k+1}^T}}\right) W_{k+1}}_{I}$$

$$= P_k H_{k+1}^T \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}\right)^{-1}$$

$$\boxed{\begin{aligned} K_{k+1} &= P_k H_{k+1}^T \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}\right)^{-1} \\ P_{k+1} &= P_k - K_{k+1} H_{k+1}^T P_k = \left(I - K_{k+1} H_{k+1}\right) P_k \end{aligned}}$$

Nonlinear Least Squares: (iteratively)

$\tilde{y} = f(x) + v$ want to estimate $x \to \hat{x}$

$\tilde{y} = \underbrace{f(\hat{x})}_{\hat{y}}$ $\quad e = \tilde{y} - \hat{y} = \Delta y$

$$\min_{\hat{x}} J = \frac{1}{2}\Delta y^T W \Delta y = \frac{1}{2}(\tilde{y} - f(\hat{x}))^T W (\tilde{y} - f(\hat{x}))$$

Linearization: $x_c \to$ current estimate

$\hat{x} = x_c + \Delta x$ (DO LS TO SOLVE FOR Δx

→ initialize x_c

→ iteratively adjust $x_c = x_c + \underline{\Delta x}$ solved for using LS

→ $\hat{x} = x_c$ at end.

$f(\hat{x}) \approx f(x_c) + H\Delta x$

where $H = \left.\frac{\partial f}{\partial x}\right|_{x_c}$

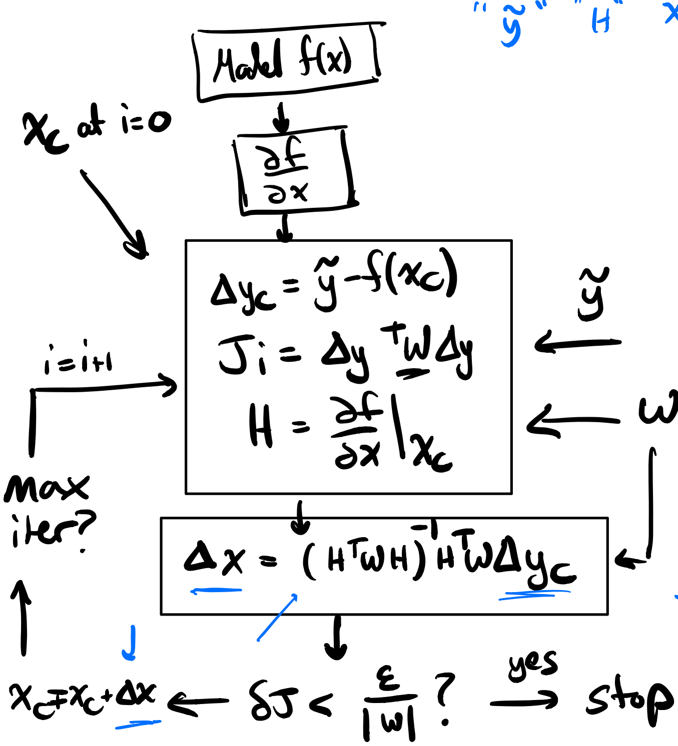
(Jacobian at x_c)

$$\Delta y = \tilde{y} - f(\hat{x}) \approx \underbrace{\tilde{y} - f(x_c)}_{\substack{\Delta y_c \\ \text{diff}}} - H\Delta x \qquad \Delta y = \Delta y_c - H\Delta x$$

meas; predicted meas

$$\min_{\Delta x} \frac{1}{2}\Delta y^T W \Delta y = \frac{1}{2}(\underbrace{\Delta y_c}_{\text{"}\tilde{y}\text{"}} - \underbrace{H}_{\text{"}H\text{"}}\underbrace{\Delta x}_{\text{"}\hat{x}\text{"}})^T W (\Delta y_c - H\Delta x)$$

$\left.\frac{\partial f}{\partial x}\right|_{x_c}$



Caveats:

- $f(x) \to$ differentiable
- x_c needs to start "close" to x.
  - what does this mean?
  - ⇒ depends
- local optima.

newton's method ??

Basis functions:

$$H = \begin{array}{c} \\ \uparrow \\ m \\ \downarrow \end{array} \overset{\longleftarrow n \longrightarrow}{\begin{bmatrix} h_1(t_1) & \cdots & h_n(t_1) \\ \vdots & & \vdots \\ h_1(t_m) & \cdots & h_n(t_m) \end{bmatrix}}$$

$m > n$

$h_i(t_j)$: basis functions

$\uparrow$ parameter index

basis functions

- functions of "time"
- one parameter per basis function

$$y(t) = \sum_i h_i(t)\, x_i \quad : \text{output.}$$

Polynomial functions in t:

$h_0(t) = 1$. $h_1(t) = t$, $h_2(t) = t^2$, etc... powers of t

$$H = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^n \\ \\ 1 & t_m & t_m^2 & \cdots & t_m^n \end{bmatrix} \rightarrow \text{Vandermonde Matrix}$$

$$y(t) = \sum_{i=0}^{n} t^i x_i$$

Sinusoidal functions: want to fit a periodic signal

$h_j^1(t) = \cos(j\omega t) \quad j = 0, 1, \ldots, n$

$h_j^2(t) = \sin(j\omega t)$ for $j = 0, 1, \ldots, n$

$$y(t) = \sum_{j=0}^{n} x_j^1 \cos(j\omega t) + \sum_{j=0}^{n} x_j^2 \sin(j\omega t)$$

$$H = \begin{bmatrix} \cos(0) & \cos(\omega t_1) & \cos(2\omega t_1) & \ldots & \sin(0) & \sin(\omega t_1) & \ldots \\ \cos(0) & \cos(\omega t_2) & \cos(2\omega t_2) & \ldots & '' & \sin(\omega t_2) & \ldots \\ & & & & & & \end{bmatrix} \begin{bmatrix} x_0^1 \\ x_1^1 \\ x_2^1 \\ \vdots \\ x_0^2 \\ x_1^2 \\ \vdots \end{bmatrix}$$

Nonlinear coord transform:

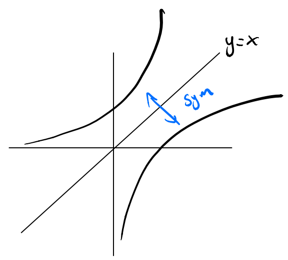
$$\underbrace{y(t) = x_1 e^{x_2 t}}_{\text{not a lin func of } x_i \ldots} \xrightarrow{\ln(\cdot)} y'(t) = \ln(y(t)) = \ln(x_1) + x_2 t$$

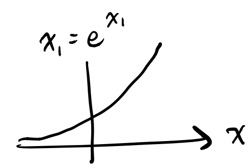
$$x_1' + x_2' t$$

$$\begin{bmatrix} \ln(y(t_1)) \\ \vdots \\ \ln(y(t_m)) \end{bmatrix} = \begin{matrix} y'(t_1) \\ \vdots \\ y'(t_m) \end{matrix} = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \boxed{x_1'} \\ x_2' \end{bmatrix}$$

$x_1' = \ln(x_1)$

$x_2' = x_2$

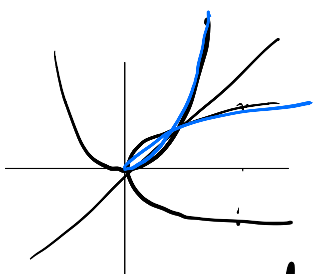


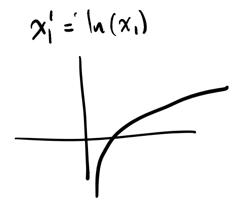


$$\begin{bmatrix} \hat{x}_1' \\ \hat{x}_2' \end{bmatrix} \longrightarrow \begin{matrix} x_1 = e^{x_1'} \\ x_2 = x_2' \end{matrix}$$

x_1' : anything

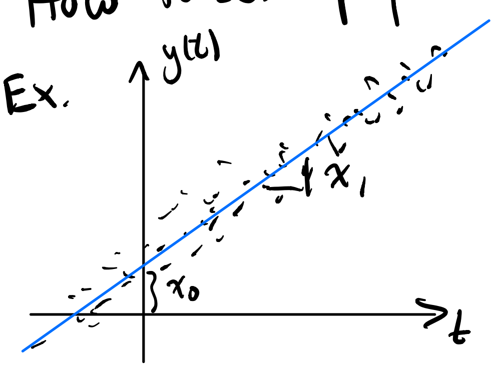
$\Rightarrow x_1 > 0$





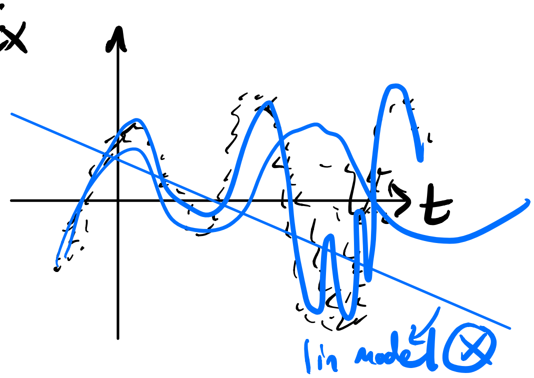
How to set up problems:

Ex.



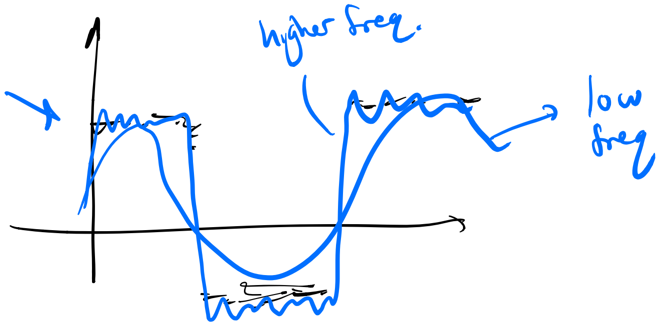
$$H = \begin{bmatrix} 1 & t_1 \\ & \\ & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Ex



$\omega \rightarrow$ small

$$\begin{bmatrix} 1 & \cos(\omega t_1) & \overrightarrow{\cos(2\omega t_1)} \cdots & -\sin(\omega t) \rightarrow \\ & & & \end{bmatrix}$$



more complicated curves $\longrightarrow$

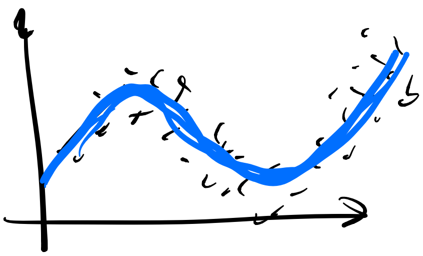
$|H|$ gets wider

$\times \downarrow$ longer

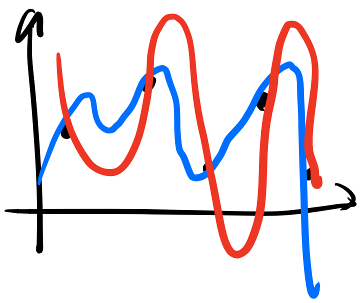
to fit more complicated curves

$\Longrightarrow$ need more data

want H to be tall



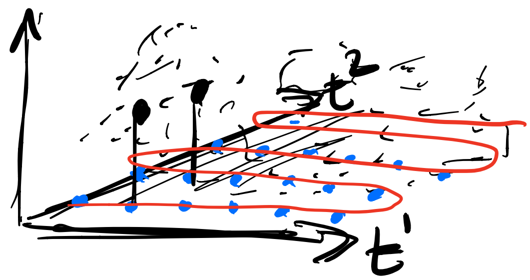
tall H



X square H ? ← overfitting

one parameter per data point.

2D SURFACE

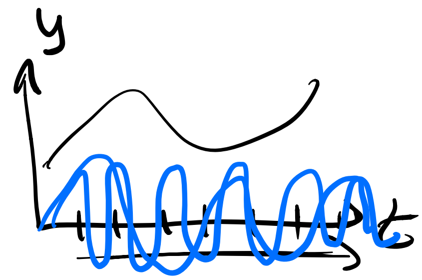


$y = Hx$

diff basis

$$
\begin{array}{c}
t^1, t^2 \\
y(t_1) \\
\vdots \\
y(t_m)
\end{array}
=
\begin{bmatrix}
 & h_j(t^1, t^2) &
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix}
$$

t^1, t^2

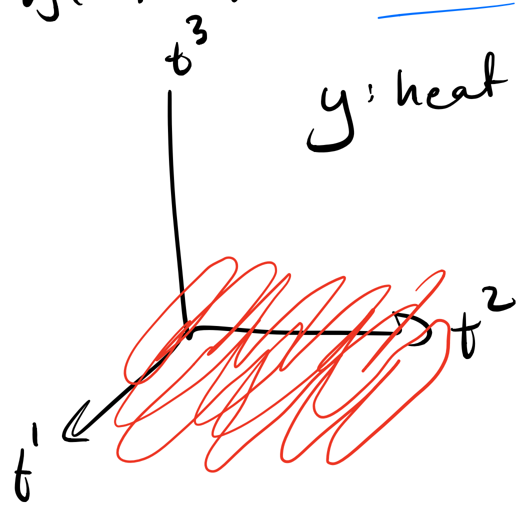


$h_j(t^1, t^2) = t^1 t^2$

... $h_j(t^1, t^2) = \cos(\omega t^1)\sin(\omega t^2)$

$h_j(t^1, t^2) = \cos(\omega t^1)$

y: heat map



MINIMUM VARIANCE (perspective on LS)

Estimator theory $W = R^{-1}$

covariance of the noise

$\rightarrow \tilde{y} = Hx + v$ $\quad v \sim N(0, R)$

Gaussian mean

compute an estimator function $\hat{x}(\tilde{y})$

mean: $E[\hat{x}] = x$

covariance

$E[(\hat{x}-x)(\hat{x}-x)^T]$

$\begin{bmatrix} \hat{x}_1 - x_1 \\ \vdots \\ \hat{x}_n - x_n \end{bmatrix} [\hat{x}_1 - x_1, \dots]$

- assume some class of functions for $\hat{x}(\cdot)$
- try to find the best estimator within that class

Properties of a good estimator

$0 = E((\hat{x}_i - x_i)(\hat{x}_j - x_j))$

- unbiased : $\boxed{E(\hat{x}(\tilde{y})) = x}$
- bias : $E(\hat{x}(\tilde{y}) - x)$

Expected Value $E(\cdot)$

$\rightarrow$ always over some prob. dist.

$E_{\tilde{y}}\, \hat{x}(\tilde{y})$

random variable

What is the minimum variance linear estimator?

Linear Estimator

$\hat{x} = M\tilde{y} + n$ $\quad$ pick M, n

unbiased

$E\hat{x} = x \Rightarrow E(M\tilde{y} + n) = E(MHx + n + Mv) = x$

$$E(M\tilde{y}+n) = E(MHx) + E(n) + E(\cancel{Mv})^{0} = x$$

2 conditions to impose

- $n = 0$
- $MH = I$

unbiased estimator

$$\hat{x} = M\tilde{y} \quad \text{s.t.} \quad \underline{MH = I}$$

$\boxed{?}$ $\underline{M = H^{-1}}$? $\quad MH = I$

$$(\underline{M + z})H = MH + \cancel{zH}^{0}$$

if $z \in$ left nullspace of H

Minimum Variance:

$$\min_{M} E\left((\hat{x}-x)^T(\hat{x}-x)\right) = J$$

$$\text{s.t.} \quad MH = I$$

need to solve this optimization problem.

$$\frac{\partial J}{\partial \underline{M}} = ?$$

$MH = I$

$$n\begin{bmatrix} C \end{bmatrix} \overset{n}{\begin{bmatrix} H \end{bmatrix}}$$

need to be invertible (many C's)

$$M = (CH)^{-1}C$$

for a lot of C's

$$MH = (CH)^{-1}CH = I$$

could choose $C = H^T$...

$$M = (H^TH)^{-1}H^T$$

- trace operator
- matrix inner products
- matrix derivatives

Formulas

$$\frac{\partial}{\partial X}\mathrm{Tr}(BXC) = B^TC^T$$

$$\frac{\partial}{\partial X}\mathrm{Tr}(XBX^T) = X(B+B^T)$$

what does this mean?

$$\mathrm{Tr}(BAC) = \mathrm{Tr}(CBA)$$

$$f(X) = \mathrm{Tr}(A^TX)$$

$$\frac{\partial f}{\partial X} = A$$

Trace Operator: $A \in \mathbb{R}^{n \times n}$

$$Tr(A) = \sum_i A_{ii} \quad \left(Tr(A) = \sum_i \lambda_i \ , \lambda_i \in \gamma(A) \right)$$

$$Tr(AB) = Tr(BA) \ \circledast \quad \text{in general } AB \neq BA$$

interesting case...

$$\underbrace{x^T y}_{\text{scalar}} = Tr(x^T y) = Tr(\underbrace{y x^T}_{\text{matrix}}) = Tr \left(\begin{bmatrix} y_1 x_1 & & y_1 x_n \\ & \ddots & \\ y_n x_1 & & y_n x_n \end{bmatrix} \right)$$

$$x_1 y_1 + \cdots + x_n y_n$$

Matrix Inner Products: $A = [A_1 \cdots A_n] \quad B = [B_1 \cdots B_n]$

$$x^T y = \sum_i x_i y_i \quad \text{matrix } \langle A, B \rangle = \sum_{i,j} A_{ij} B_{ij} = \sum_i A_i^T B_i$$

$$\boxed{\langle A, B \rangle = Tr(A^T B)} = Tr \left(\begin{bmatrix} - A_1^T - \\ - A_n^T - \end{bmatrix} \begin{bmatrix} | & & | \\ B_1 & \cdots & B_n \\ | & & | \end{bmatrix} \right)$$

$$= Tr \left(\begin{bmatrix} A_1^T B_1 & & \\ & \ddots & \\ & & A_n^T B_n \end{bmatrix} \right)$$

Matrix Derivatives

$X \in \mathbb{R}^{n \times n} \quad f : \mathbb{R}^{n \times n} \longrightarrow \mathbb{R}$

$\quad x \longmapsto \mathbb{R}$

$$\frac{\partial f}{\partial X} = ?$$

variable is a matrix

Recall:

- $\boxed{f(x) = c^T x} \ \boxed{\frac{\partial f}{\partial x} = c^T}$
- $f(x) = Ax \quad \frac{\partial f}{\partial x} = A$

variable is a vector...

$$f(X) = \langle A, X \rangle = \mathrm{Tr}(A^T X)$$

perturbation analysis...

$$\Delta f = \mathrm{Tr}(A^T \underline{\Delta X}) \qquad \cancel{\frac{\partial f}{\partial X} = A^T} \;\; ?$$

2 options:

- vectorize ΔX $\rightarrow$ put ΔX in vector form... $\dfrac{\partial f}{\partial \mathrm{vec}(X)} =$ matrix.

  stacking columns

$$X = [\underline{x_1 \cdots x_n}] \Rightarrow \mathrm{vec}(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Wikipedia: Vectorization

$$f(X) = \mathrm{Tr}(A^T X) = \mathrm{vec}(A)^T \mathrm{vec}(X) = [A_{11}\ A_{21} \cdots A_{n1}\ A_{12}\ A_{22} \cdots] \begin{bmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \\ \vdots \end{bmatrix}$$

$$\langle A, X \rangle = \sum_{ij} A_{ij} X_{ij}$$

$$\frac{\partial f}{\partial \mathrm{vec}(X)} = \mathrm{vec}(A)^T$$

similar to $\dfrac{\partial f}{\partial x} = c^T$ when $f(x) = c^T x$

- $\dfrac{\partial f}{\partial X}(\Delta X) = \Delta f$

  linear function

$$\frac{\partial f}{\partial X}(\cdot) = \langle F, \cdot \rangle = \mathrm{Tr}(F^T \cdot)$$

Question is what is F?

if $f(X) = Tr(A^TX) \Rightarrow F = A$

$\frac{\partial f}{\partial X}(\cdot) = Tr(A^T \cdot)$

$f(x) = \underline{C^Tx}$ ($C \cdot x$) $\quad \frac{\partial f}{\partial x} = C^T$

$f(X) = Tr(A^TX)$ ($A \cdot X$) $\quad \frac{\partial f}{\partial X} = A$

$f(X) = \underline{Tr(A^TX)} \Rightarrow \underline{\frac{\partial f}{\partial X} = A}$

d_i

$f(X) = Tr(A^TX)$
$\frac{\partial f}{\partial X} = A$

WRONG

$$\boxed{\underline{\Delta f} = \frac{\partial f}{\partial X} \underline{\Delta X}}$$

$= \underline{A}\,\underline{\Delta X}$

scalar = matrix

$f(x) = Tr(A^TX) \quad \underline{\frac{\partial f}{\partial X}} = A$

$f: \mathbb{R}^{n \times n} \longrightarrow \mathbb{R}$

$\Delta X \qquad \Delta f$

$$\boxed{\underline{\Delta f} = Tr\left(\frac{\partial f}{\partial X}^T \Delta X\right)}$$

TRUE

$\frac{\partial f}{\partial X} = A \qquad \Delta f = Tr(A^T \Delta X)$