上海科技大学 ShanghaiTech University

ADSP

Constrained MMSE

$$\text{Solution } \mathbf{C}^t \cdot \underline{\mathbf{w}} = \underline{\mathbf{f}} :$$

Case $N = M$:

$$\Rightarrow \; \underline{\mathbf{w}}^c = \left(\mathbf{C}^t\right)^{-1} \cdot \underline{\mathbf{f}}$$

$$\Rightarrow \; \textit{No degrees of freedom left for MMSE}$$

Case $N > M$:

$$\Rightarrow \; \text{Possible solution: } \underline{\mathbf{w}}^c = \left(\mathbf{C}^t\right)^{\dagger} \cdot \underline{\mathbf{f}}$$

$$\textit{Appendix:} \text{ Generalized inverse } \left(\mathbf{C}^t\right)^{\dagger} = \mathbf{C} \cdot \left(\mathbf{C}^t \cdot \mathbf{C}\right)^{-1}$$

$$\Rightarrow \; N - M \; \textit{degrees of freedom left over for MMSE}$$

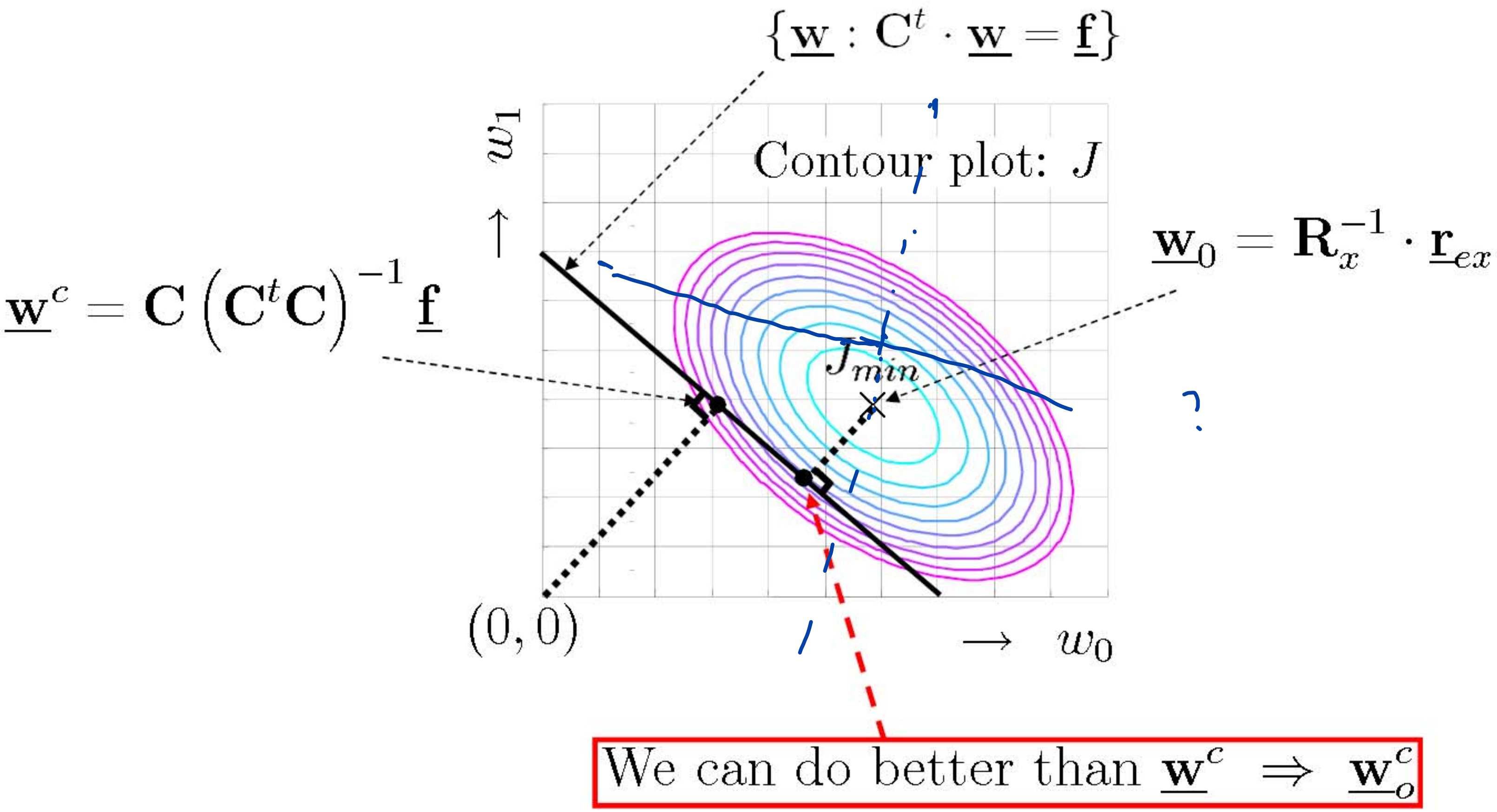
$N < M \;\Rightarrow\;$ Conflicting solution

(choose e.g. minimum norm soluton)

ADSP

Constrained MMSE

Use $N - M$ degrees of freedom to improve result: $\underline{\mathbf{w}}^c \Rightarrow \underline{\mathbf{w}}_o^c$



Constrained MMSE

Use Lagrange multipliers

Performance index:

$$
\begin{aligned}
J^c &= E\{r^2\} + \underset{1\cdot M}{\underline{\lambda}^t}\left(\mathbf{C}^t\underline{\mathbf{w}} - \underline{\mathbf{f}}\right)_{M\cdot 1} \\
&= E\{e^2\} - \underline{\mathbf{w}}^t\underline{\mathbf{r}}_{ex} - \underline{\mathbf{r}}_{ex}^t\underline{\mathbf{w}} + \underline{\mathbf{w}}^t\mathbf{R}_x\underline{\mathbf{w}} + \underline{\lambda}^t\left(\mathbf{C}^t\underline{\mathbf{w}} - \underline{\mathbf{f}}\right)
\end{aligned}
$$

Gradient vector: $\dfrac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathbf{w}}} = \underline{\nabla} = -2\underline{\mathbf{r}}_{ex} + 2\mathbf{R}_x\underline{\mathbf{w}} + \mathbf{C}\underline{\lambda}$

$$
\left.
\begin{aligned}
&\frac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathbf{w}}} = \underline{\mathbf{0}} \Rightarrow \underline{\mathbf{w}}_o^c = \mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} - \frac{1}{2}\mathbf{R}_x^{-1}\mathbf{C}\underline{\lambda} \\
&\frac{dJ^c}{d\underline{\lambda}^t} = 0 \Rightarrow \quad \text{左乘 } C^t \\
&\text{In optimum} \Rightarrow \mathbf{C}^t\underline{\mathbf{w}}_o^c = \underline{\mathbf{f}} \qquad C^t_{M\cdot N}
\end{aligned}
\right\} \Longrightarrow
$$

$$\Rightarrow \underline{\lambda} = 2\left(\mathbf{C}^t\mathbf{R}_x^{-1}\mathbf{C}\right)^{-1}\left(\mathbf{C}^t\mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} - \underline{\mathbf{f}}\right)$$

不能继续展开，因为C不一定是方阵

$\underline{f} = C^t R_x^{-1} \underline{r}_{ex} - \frac{1}{2} C^t R_x^{-1} C \underline{\lambda} \Rightarrow \underline{\lambda} = 2(C^t R_x^{-1} C)^{-1}(C^t R_x^{-1} \underline{r}_{ex} - f)$

Constrained MMSE

Finally

$$\left.\begin{aligned} \underline{\mathbf{w}}_o^c &= \mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} - \frac{1}{2}\mathbf{R}_x^{-1}\mathbf{C}\underline{\lambda} \\ \underline{\lambda} &= 2\left(\mathbf{C}^t\mathbf{R}_x^{-1}\mathbf{C}\right)^{-1}\left(\mathbf{C}^t\mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} - \underline{\mathbf{f}}\right) \end{aligned}\right\} \Longrightarrow$$

$$\boxed{\begin{gathered} \underline{\mathbf{w}}_o^c = \underline{\mathbf{w}}_o + \mathbf{R}_x^{-1}\mathbf{C}\left(\mathbf{C}^t\mathbf{R}_x^{-1}\mathbf{C}\right)^{-1}\left(\underline{\mathbf{f}} - \mathbf{C}^t\underline{\mathbf{w}}_o\right) \\ \text{with } \underline{\mathbf{w}}_o = \mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} \end{gathered}}$$

Check:

$$\mathbf{C}^t\underline{\mathbf{w}}_o^c = \mathbf{C}^t\underline{\mathbf{w}}_o + \left(\mathbf{C}^t\mathbf{R}_x^{-1}\mathbf{C}\right)\left(\mathbf{C}^t\mathbf{R}_x^{-1}\mathbf{C}\right)^{-1}\left(\underline{\mathbf{f}} - \mathbf{C}^t\underline{\mathbf{w}}_o\right) = \underline{\mathbf{f}}$$

Constrained MMSE

Notes:

- $J^c = J_{min} + \underline{\mathbf{d}}^t \mathrm{R}_x \underline{\mathbf{d}} + \underline{\lambda}^t \left(\mathrm{C}^t \underline{\mathbf{w}} - \underline{\mathbf{f}}\right)$ with $\underline{\mathbf{d}} = \underline{\mathbf{w}} - \underline{\mathbf{w}}_o$

- $$\left.\begin{aligned} \underline{\nabla} &= -2\underline{\mathbf{r}}_{ex} + 2\mathrm{R}_x \underline{\mathbf{w}} + \mathrm{C}\underline{\lambda} \\ \underline{\lambda} &= 2\left(\mathrm{C}^t \mathrm{R}_x^{-1} \mathrm{C}\right)^{-1} \left(\mathrm{C}^t \mathrm{R}_x^{-1} \underline{\mathbf{r}}_{ex} - \underline{\mathbf{f}}\right) \end{aligned}\right\} \Longrightarrow$$

  $$\boxed{\underline{\nabla} = -2\underline{\mathbf{r}}_{ex} + 2\mathrm{R}_x \underline{\mathbf{w}} + 2\mathrm{C}\left(\mathrm{C}^t \mathrm{R}_x^{-1} \mathrm{C}\right)^{-1} \left(\mathrm{C}^t \mathrm{R}_x^{-1} \underline{\mathbf{r}}_{ex} - \underline{\mathbf{f}}\right)}$$

- In optimum $\underline{\nabla} = 0 \;\Rightarrow\; \underline{\mathbf{w}}_o^c$

ADSP Part B: Adaptive signal processing

Focus on single channel adaptive algorithms using FIR structure

- Applications Adaptive Algorithms
- Minimum Mean Square Error (MMSE)
- Constrained MMSE
- Least Square (LS) 避免MMSE中统计量的出现. $E[\cdot]$, R_x, $\underline{r}_{ex}$
- Steepest Gradient Descent (SGD)
- Three LMS variants: NLMS, Complex LMS, Constrained LMS
- Newton
- Recursive Least Squares (RLS)
- Frequency Domain Adaptive Filter (FDAF)
- Summary

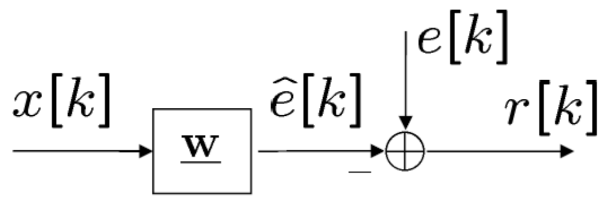
Part B: Adaptive signal processing

上海科技大学 ShanghaiTech University

ADSP

MMSE 需要统计量 $E[\cdot] \Rightarrow R_x,\ r_{ex}$

LS



Quadratic cost functions:

- Mean Square Error (MSE):

  $$J_{mse} = E\{r^2[k]\} = E\{(e[k] - \underline{\mathbf{w}}^t[k]\underline{\mathbf{x}}[k])^2\}$$ $w_o = R_x^{-1}\, \underline{r}_{ex}$

  Minimum MSE (MMSE) = Wiener

- **Least Squares** (LS): 选择 LS 的原因

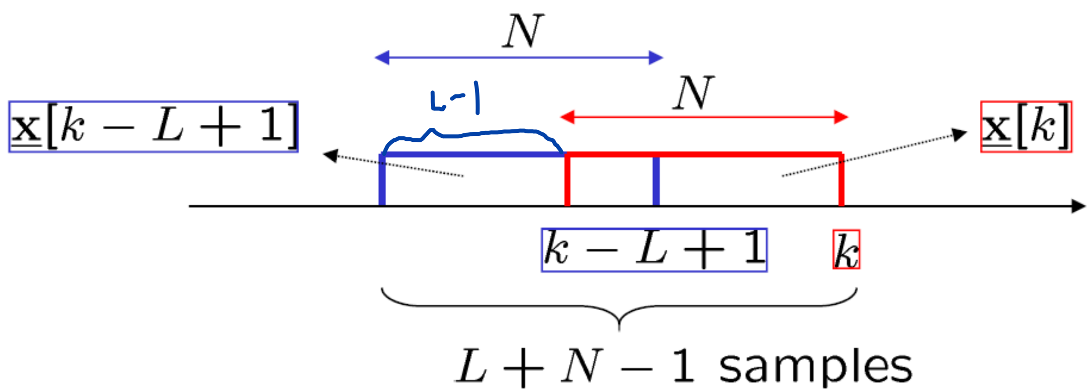
  If statistical info not available $\rightarrow$

  Use criterion based on data (thus without $E\{\}$)

ADSP

LS

Collect $L(\geq 1)$ data vectors $\underline{\mathbf{x}}[k-i]$ (of length N)



step = 1

可以 $\underline{x}[k-2i]$ 吗?

Available data (for $i=0,1,\cdots,L-1$) :

Input signals: $\underline{\mathbf{x}}[k-i]$

$$\underline{\mathbf{x}}^t[k-i]=(x[k-i],x[k-i-1],\cdots x[k-i-N+1])^t$$ $1 \times N$

Reference signals: $e[k-i]$

Residual signals: $r[k-i]=e[k-i]-\underline{\mathbf{x}}^t[k-i]\cdot\underline{\mathbf{w}}$

$$\bar{R}_x = X^t X \overset{?}{=} \begin{pmatrix} Lr_x[0] & Lr_x[1] & \cdots & Lr_x[N-1] \\ Lr_x[1] & Lr_x[0] & \cdots & Lr_x[N-1] \\ \vdots & & & \\ Lr_x[N-1] & Lr_x[N-2] & \cdots & Lr_x[0] \end{pmatrix} = \sum_{i=0}^{L-1} \underline{x}[k-i]\,\underline{x}^t[k-i] \overset{?}{=} LR_x$$

错的 错的

并不求期望

所以不会出现 r_x

Part B: Adaptive signal processing

上海科技大学 ShanghaiTech University

ADSP

LS

$$X^t[k] = \begin{pmatrix} x[k] & x[k-1] & \cdots & x[k-L+1] \\ x[k-1] & x[k-2] & \cdots & \vdots \\ \vdots & \vdots & & \\ x[k-N+1] & x[k-N] & & x[k-L-N+2] \end{pmatrix}$$

$$X[k] = \begin{pmatrix} x[k], & x[k-1] & \cdots & x[k-N+1] \\ x[k-1], & x[k-2] & \cdots & x[k-N] \\ \vdots & & & \\ x[k-L+1], & x[k-L] & \cdots & x[k-L-N+2] \end{pmatrix}$$

$$\mathbf{X}[k] = \begin{pmatrix} \boxed{\underline{\mathbf{x}}^t[k]}_{1\times N} \\ \underline{\mathbf{x}}^t[k-1] \\ \vdots \\ \boxed{\underline{\mathbf{x}}^t[k-L+1]} \end{pmatrix}_{L\times N} \quad \underline{\mathbf{w}} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}_{N\times 1}$$

$X^tX_{N\times N}$

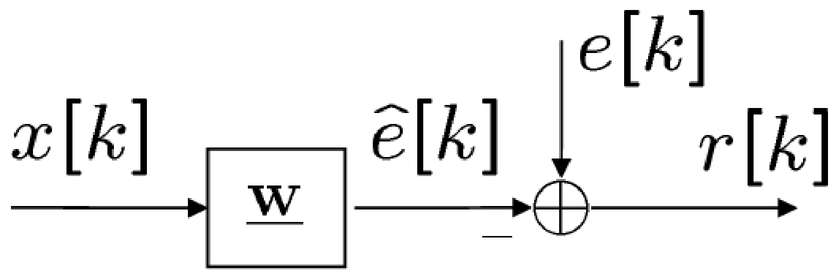
$XX^t_{L\times L}$

$$\underline{\mathbf{e}}[k] = \begin{pmatrix} e[k] \\ e[k-1] \\ \vdots \\ e[k-L+1] \end{pmatrix}_{L\times 1} \quad \underline{\mathbf{r}}[k] = \begin{pmatrix} r[k] \\ r[k-1] \\ \vdots \\ r[k-L+1] \end{pmatrix}_{L\times 1}$$

Simplified notation (skipping time indices):

$$\underline{\mathbf{r}} = \underline{\mathbf{e}} - \mathbf{X} \cdot \underline{\mathbf{w}}$$

LS



$\underline{\textbf{LS problem formulation:}}$

$$\underline{\mathbf{w}}_{LS,opt} = \arg\min_{\underline{\mathbf{w}}} |\underline{\mathbf{e}} - \mathbf{X} \cdot \underline{\mathbf{w}}|^2$$

$J_{MMSE} = E[r^2]$

LS

向量，不同于 MMSE

$$J_{LS} = \boxed{\sum_{i=0}^{L-1}} r^2[k-i] = \underline{\mathbf{r}}^t \cdot \underline{\mathbf{r}} = (\underline{\mathbf{e}}^t - \underline{\mathbf{w}}^t \mathbf{X}^t) \cdot (\underline{\mathbf{e}} - \mathbf{X}\underline{\mathbf{w}})$$

$N \times 1$ $L \times 1$ $L \times N$

$\underline{r}^t \cdot \underline{r} \Rightarrow$

$$= \underline{\mathbf{e}}^t \underline{\mathbf{e}} + \underline{\mathbf{w}}^t \mathbf{X}^t \mathbf{X} \underline{\mathbf{w}} - \underline{\mathbf{w}}^t \mathbf{X}^t \underline{\mathbf{e}} - \underline{\mathbf{e}}^t \mathbf{X} \underline{\mathbf{w}}$$

Minimum by setting gradient to 0:

$$\frac{\mathrm{d}J_{LS}}{\mathrm{d}\underline{\mathbf{w}}} = \underline{\nabla}_{LS} = -2\left(\mathbf{X}^t \underline{\mathbf{e}} - \mathbf{X}^t \mathbf{X} \cdot \underline{\mathbf{w}}\right) = \underline{\mathbf{0}}$$

$\Rightarrow$ **Normal equations:** $\mathbf{X}^t \mathbf{X} \cdot \underline{\mathbf{w}} = \mathbf{X}^t \underline{\mathbf{e}}$

无期望 $E[\cdot]$

With: $\overline{\mathbf{R}}_x = \underset{N\cdot L \,\cdot\, L\cdot N}{\mathbf{X}^t \mathbf{X}}$ and $\underline{\overline{\mathbf{r}}}_{ex} = \underset{N\cdot L \,\cdot\, L\cdot 1}{\mathbf{X}^t \underline{\mathbf{e}}} \Rightarrow$

类似于 自相关

$$\boxed{\underline{\mathbf{w}}_{LS} = \overline{\mathbf{R}}_x^{-1} \cdot \underline{\overline{\mathbf{r}}}_{ex}}$$

$w_0 = R_x^{-1} \cdot \underline{r}_{ex}$ 维纳滤波

$R_x = E[\underline{x}^t \underline{x}]$, $\underline{r}_{ex} = E[e \underline{x}]$

LS

Notes: $[\underline{x}[k], \underline{x}[k-1] \ldots]$ $\begin{bmatrix} \underline{x}[k]^T \\ \underline{x}[k-1]^T \\ \underline{x}[k-i]^T \end{bmatrix}$

对应 $X[k]$ 的行
$x^t[k]$ 的列

- Auto- and cross-correlation can be written as:

矩阵相乘写成列乘行的累加，即小矩阵的累加

$$\overline{\mathbf{R}}_x[k] = \underbrace{\mathbf{X}^t[k]}_{N\cdot L}\underbrace{\mathbf{X}[k]}_{L\cdot N} = \sum_{i=0}^{L-1} \underbrace{\underline{\mathbf{x}}[k-i]}_{\text{列向量}} \cdot \underbrace{\underline{\mathbf{x}}^t[k-i]}_{\text{行向量}}$$

$$\overline{\mathbf{r}}_{ex}[k] = \mathbf{X}^t[k]\underline{\mathbf{e}}[k] = \sum_{i=0}^{L-1} \underbrace{\underline{\mathbf{x}}[k-i]}_{\text{列向量}}e[k-i]$$

- Least square error can be written as:

$$J_{LS} = J_{LS,o} + (\underline{\mathbf{w}} - \underline{\mathbf{w}}_{LS})^t \cdot \overline{\mathbf{R}}_x \cdot (\underline{\mathbf{w}} - \underline{\mathbf{w}}_{LS})$$

$$\text{with } J_{LS,o} = J_{LS}|_{\underline{\mathbf{w}}=\underline{\mathbf{w}}_{LS}} = \underline{\mathbf{e}}^t\underline{\mathbf{e}} - \overline{\mathbf{r}}_{ex}^t\overline{\mathbf{R}}_x^{-1}\overline{\mathbf{r}}_{ex}$$

$$J_{MMSE}|_{\underline{w}=\underline{w}_o} = E\{e^2\} - E\{\hat{e}^2\} = E\{e^2\} - E\{(e-r)\underline{x}^t\underline{w}_o\} = E\{e^2\} - \underline{r}_{ex}^t\underline{w}_o$$
$$= E\{e^2\} - \underline{r}_{ex}^t R_x^{-1}\underline{r}_{ex}$$

LS

Correspondence with Wiener filter

$R_x = E\{\underline{x}[k]\,\underline{x}^t[k]\}$

Use time-averaging (ergodicity):

$X^t \cdot X = \sum_{i=0}^{L-1} \underline{x}[k-i] \cdot \underline{x}^t[k-i]$

$$\hat{\mathbf{R}}_x = \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i] \cdot \underline{\mathbf{x}}^t[k-i] = \frac{1}{L}\mathbf{X}^t \cdot \mathbf{X} = \frac{1}{L}\overline{\mathbf{R}}_x$$

$$\hat{\underline{\mathbf{r}}}_{ex} = \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i] \cdot e[k-i] = \frac{1}{L}\mathbf{X}^t \cdot \underline{\mathbf{e}} = \frac{1}{L}\overline{\underline{\mathbf{r}}}_{ex}$$

with $\hat{\mathbf{R}}_x$ estimate $\mathbf{R}_x$ and $\hat{\underline{\mathbf{r}}}_{ex}$ estimate $\underline{\mathbf{r}}_{ex}$

$$\Rightarrow \quad \hat{\underline{\mathbf{w}}}_{Wiener} = \left(\frac{1}{L}\overline{\mathbf{R}}_x\right)^{-1} \cdot \left(\frac{1}{L}\overline{\underline{\mathbf{r}}}_{ex}\right) = \overline{\mathbf{R}}_x^{-1} \cdot \overline{\underline{\mathbf{r}}}_{ex} = \underline{\mathbf{w}}_{LS}$$

$\underline{w}_{wiener} = R_x^{-1} \cdot \underline{r}_{ex}$

$\hat{\underline{w}}_{wiener} = \hat{R}_x^{-1}\,\hat{\underline{r}}_{ex}$

LS

Finally note that for ergodic processes:

$$\lim_{L\to\infty} \frac{1}{L}\overline{\mathbf{R}}_x = \lim_{L\to\infty} \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i]\cdot \underline{\mathbf{x}}^t[k-i] = \mathbf{R}_x$$

$$\lim_{L\to\infty} \frac{1}{L}\overline{\underline{\mathbf{r}}}_{ex} = \lim_{L\to\infty} \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i]\cdot e[k-i] = \underline{\mathbf{r}}_{ex}$$

$$\Rightarrow \lim_{L\to\infty}\{\underline{\mathbf{w}}_{LS}\} = \lim_{L\to\infty}\{\overline{\mathbf{R}}_x^{-1}\overline{\underline{\mathbf{r}}}_{ex}\} = \mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} = \underline{\mathbf{w}}_{Wiener}$$

$\underbrace{\frac{1}{L}\hat{R}_x^{-1}\cdot L\cdot \hat{\underline{r}}_{ex}} \to \hat{R}_x^{-1}\hat{\underline{r}}_{ex}$

Wiener 的 估计值

上海科技大学 ShanghaiTech University

ADSP

Part B: Adaptive signal processing

Focus on single channel adaptive algorithms using FIR structure

- Applications Adaptive Algorithms
- Minimum Mean Square Error (MMSE)
- Constrained MMSE
- Least Square (LS) → 用来估计 MMSE ($\hat{w}_o$ wiener)
- **Steepest Gradient Descent (SGD)** 避免 MMSE 中求逆之问题 R_x^{-1}
- Three LMS variants: NLMS, Complex LMS, Constrained LMS
- Newton
- Recursive Least Squares (RLS)
- Frequency Domain Adaptive Filter (FDAF)
- Summary

SGD 避免求逆困难的问题

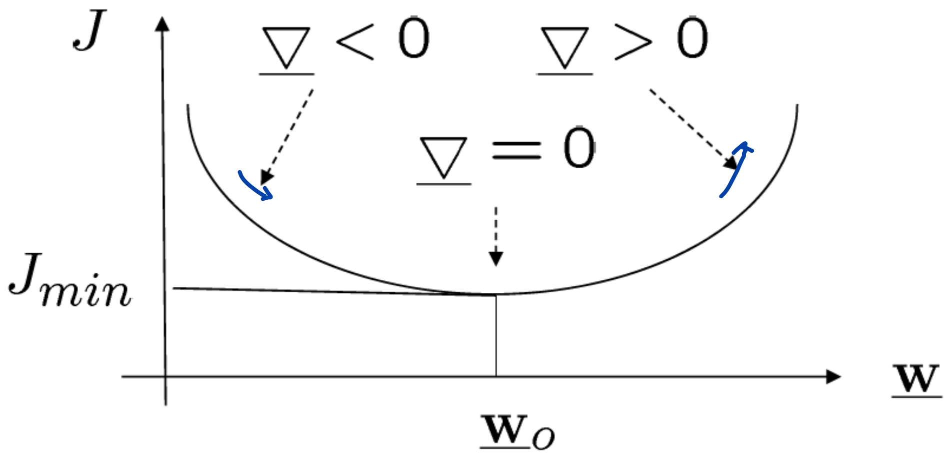
Problem: Optimal Wiener involves $\mathbf{R}_x^{-1}$

To avoid this inversion, estimate optimum iteratively

Goal: Decrease J each new iteration

SGD

SGD principle: Update in negative gradient direction



$\Rightarrow \quad \underline{\mathbf{w}} := \underline{\mathbf{w}} - \alpha \underline{\nabla}$ with adaptation constant $\alpha \geq 0$

步长

With $\underline{\nabla} = -2\left(\underline{\mathbf{r}}_{ex} - \mathbf{R}_x \underline{\mathbf{w}}[k]\right) \quad \Rightarrow$

SGD algorithm: $\boxed{\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha\left(\underline{\mathbf{r}}_{ex} - \mathbf{R}_x \underline{\mathbf{w}}[k]\right)}$

Note: Usually $\underline{\mathbf{w}}[0] = \underline{\mathbf{0}}$ No matrix inversion needed!

虽不需要求逆，但需要统计量信息

SGD

Simple 'proof' of fact that for SGD:

$$\lim_{k\to\infty} \{\underline{\mathbf{w}}[k]\} = \mathbf{R}_x^{-1} \cdot \underline{\mathbf{r}}_{ex} = \text{'Wiener solution'}$$

'Proof': $\underline{w}[k+1] = \underline{w}[k] + 2 \cdot 2 (\underline{r}_{ex} - R_x \underline{w}[k])$

For $k \to \infty$ we have: $\underline{\mathbf{w}}[k+1] \approx \underline{\mathbf{w}}[k] \approx \underline{\mathbf{w}}[\infty]$

Thus for $k \to \infty$ SGD reduces to:

$$\underline{\mathbf{w}}[\infty] = \underline{\mathbf{w}}[\infty] + 2\alpha \left(\underline{\mathbf{r}}_{ex} - \mathbf{R}_x \underline{\mathbf{w}}[\infty]\right)$$

$$\Rightarrow \; \underline{\mathbf{w}}[\infty] = \mathbf{R}_x^{-1} \cdot \underline{\mathbf{r}}_{ex} = \text{Wiener}$$

For exact proof we need **stability analysis**

w和J的平稳性分析

Stability SGD

Define weight error: $\underline{\mathbf{d}}[k] = \underline{\mathbf{w}}[k] - \underline{\mathbf{w}}_o$

$\underline{w}[k+1] = \underline{w}[k] - \alpha \cdot \underline{\Delta}$

SGD定义:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha\left(\underline{\mathbf{r}}_{ex} - \mathbf{R}_x \underline{\mathbf{w}}[k]\right) \quad - \underline{w}_o$$

$- \underline{w}_o$

$$\underline{\mathbf{w}}[k+1] - \underline{\mathbf{w}}_o = \left(\mathbf{I} - 2\alpha\mathbf{R}_x\right)\underline{\mathbf{w}}[k] - \underline{\mathbf{w}}_o + \underbrace{2\alpha\underline{\mathbf{r}}_{ex}}_{2\alpha \cdot R_x \underline{w}_o}$$

$$\Rightarrow \quad \underline{\mathbf{d}}[k+1] = \left(\mathbf{I} - 2\alpha\mathbf{R}_x\right)\underline{\mathbf{d}}[k]$$

$(1 - 2\alpha R_x)\underline{w}_o$

Recursion:

$$\underline{\mathbf{d}}[k] = \left(\mathbf{I} - 2\alpha\mathbf{R}_x\right)\underline{\mathbf{d}}[k-1] = \cdots = \left(\mathbf{I} - 2\alpha\mathbf{R}_x\right)^k \underline{\mathbf{d}}[0]$$

Stable iff: $\lim_{k\to\infty} \left(\mathbf{I} - 2\alpha\mathbf{R}_x\right)^k = \mathbf{0}$

求解 k 次幂，将其分解成对角阵.

Note: If stable $\Rightarrow$ $\underline{\mathbf{d}}[\infty] = 0$ $\Rightarrow$ $\underline{\mathbf{w}}[\infty]$ = Wiener

Stability SGD

How do weights converge:

Use eigenvalue decomposition (see Appendix):

With: $\mathbf{Q}^h \cdot \mathbf{Q} = \mathbf{Q} \cdot \mathbf{Q}^h = \mathbf{I}$ and $\mathbf{R}_x = \mathbf{Q}\Lambda\mathbf{Q}^h$

$$\Rightarrow \quad (\mathbf{I} - 2\alpha\mathbf{R}_x)^k = \left(\mathbf{Q}\mathbf{Q}^h - 2\alpha\mathbf{Q}\Lambda\mathbf{Q}^h\right)^k$$

$$= \mathbf{Q}\,(\mathbf{I} - 2\alpha\Lambda)^k\,\mathbf{Q}^h$$

Change of variables:

$\underline{d}[k] = \mathbf{Q}(\mathbf{I} - 2\alpha\Lambda)^k \mathbf{Q}^h \underline{d}[0]$

$\mathbf{Q}^h \underline{d}[k] = \mathbf{Q}^h \mathbf{Q}(\mathbf{I} - 2\alpha\Lambda)^k \mathbf{Q}^h \underline{d}[0]$

迭代出

$$\underline{\mathbf{D}}[k] = \mathbf{Q}^h \cdot \underline{\mathbf{d}}[k]$$

$\underline{d}[k]$与$\underline{d}[0]$之关系 →

$$\underline{\mathbf{d}}[k] = (\mathbf{I} - 2\alpha\mathbf{R}_x)^k\,\underline{\mathbf{d}}[0] \quad \Rightarrow \quad \underline{\mathbf{D}}[k] = (\mathbf{I} - 2\alpha\Lambda)^k\,\underline{\mathbf{D}}[0]$$

Recursion stable iff: $\lim_{k\to\infty} (\mathbf{I} - 2\alpha\Lambda)^k = 0$ 收敛条件

$-1 < 1 - 2\alpha\lambda_i < 1$

$0 < 2\alpha\lambda_i < 2 \Rightarrow 0 < \alpha < \frac{1}{\lambda_i}$

for all i: $|1 - 2\alpha\lambda_i| < 1 \Rightarrow \alpha < \frac{1}{\lambda_i} \Rightarrow \alpha < \frac{1}{\lambda_{max}}$

ADSP

Stability SGD

Recursion stable iff: $\lim_{k\to\infty} (\mathbf{I} - 2\alpha\mathbf{\Lambda})^k = \mathbf{0}$

Both matrices $\mathbf{I}$ and $\mathbf{\Lambda}$ diagonal $\Rightarrow$

$$|1-2\alpha\lambda_i| < 1 \quad \Leftrightarrow \quad 0 < \alpha < \frac{1}{\lambda_i} \text{ for } i = 0, 1, \cdots, N-1$$

SGD algorithm stable if: $0 < \alpha < \dfrac{1}{\lambda_{max}}$

For adaptation constant α in this region:

$$\lim_{k\to\infty}\{\underline{\mathbf{w}}[k]\} = \underline{\mathbf{w}}_o = \mathbf{R}_x^{-1} \cdot \underline{\mathbf{r}}_{ex}$$

$$J|_{\underline{\mathbf{w}}=\underline{\mathbf{w}}_o} = E\{r^2[k]\} = J_{min} = E\{e^2\} - \underline{\mathbf{r}}_{ex}^t \mathbf{R}_x^{-1} \underline{\mathbf{r}}_{ex}$$

SGD需要统计量信息

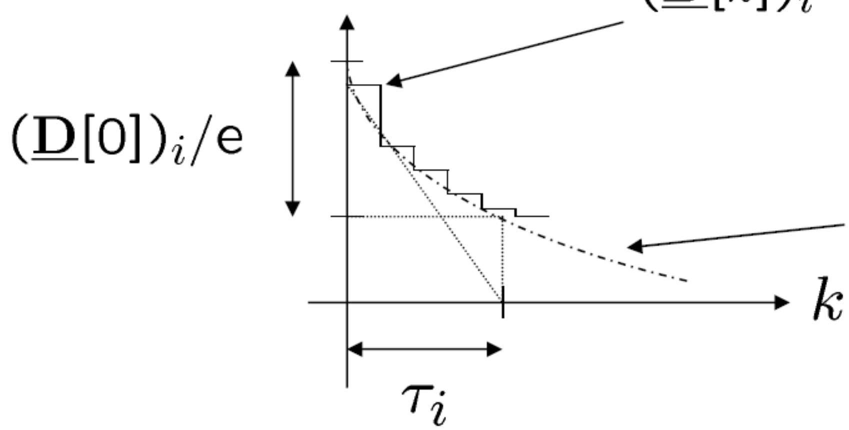
Convergence rate SGD

Behaviour coefficient i of $\underline{\mathbf{D}}[k] = (\mathbf{I} - 2\alpha\mathbf{\Lambda})^k\,\underline{\mathbf{D}}[0]$:

$(\underline{w})_i \to (\underline{w}_0)_i$ 的时间

$$(\underline{\mathbf{D}}[k])_i = (1 - 2\alpha\lambda_i)^k \cdot (\underline{\mathbf{D}}[0])_i$$

$\underline{D}[0]_i = [Q^h(\underline{w}[0] - \underline{w}_0)]_i$



$$\approx \mathrm{e}^{-k/\tau_i} \cdot (\underline{\mathbf{D}}[0])_i$$

定义用来估算时间 逼近曲线

$x \to 0$ $\ln(1+x) \sim x$

Time constant follows from:

$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{\mathbf{D}}[0])_i = (1 - 2\alpha\lambda_i)^k \cdot (\underline{\mathbf{D}}[0])_i$$

$\ln(e^{-\frac{k}{\tau_i}}) = \ln(1 - 2\alpha\lambda_i)^k$

$-\frac{k}{\tau_i} = k\ln(1 - 2\alpha\lambda_i)$

和k无关? $\Rightarrow$ $k \to \infty$ $\tau_i = -\dfrac{1}{\ln(1-2\alpha\lambda_i)}$

Convergence rate SGD

$\Rightarrow$ Time constant average weights behaviour:

$x \to 0$

$\ln(1+x) \to x$

$(\underline{w})_i \to (\underline{w}_o)_i$ 的收敛时间

$$\tau_{av,i} = \frac{-1}{\ln(1-2\alpha\lambda_i)} \qquad \text{for small } \alpha : \ \tau_{av,i} \approx \frac{1}{2\alpha\lambda_i}$$

可以从功率角度理解

Similar derivation for ~~N~~MSE: $\tau_{\cancel{n}mse,i} \approx \dfrac{1}{4\alpha\lambda_i}$ $\quad r^2 = 4\alpha\lambda$

$w_i \to (w_o)_i \Rightarrow \underline{w} \Rightarrow \underline{w}_o \to \frac{1}{2\alpha\lambda_{min}} \to \frac{1}{2\lambda_{min}}$

Notes on overall time constant τ_{av}:

$\Delta J[k] = (\underline{w}[k] - \underline{w}_o[k])^t R_x (\underline{w}[k] - \underline{w}_o[k])$

$= \underline{d}^t[k] R_x \underline{d}[k]$

$= (Q\underline{D}[k])^t Q\Lambda Q^t (Q\underline{D}[k])$

$= \underline{D}^t[k] \Lambda \underline{D}[k]$

- Depends on eigenvalue spread $\Gamma_x \doteq \lambda_{max}/\lambda_{min}$ 常数 ?

Thus, the larger Γ_x the longer it takes for adaptation

Q: What happens for white noise input process?

$\underline{D}[k] = (I - 2\alpha\Lambda)^k \underline{D}[0]$

取向量的一个元素

$\Delta J[k] = \underline{D}^t[0]((I-2\alpha\Lambda)^k)^t \Lambda (I-2\alpha\Lambda)^k \underline{D}[0]$

$\Delta J[k] = (1-2\alpha\lambda_i)^{2k} \lambda_i D_i^2[0]$

$\downarrow$

$J[0]$

Part B: Adaptive signal processing

上海科技大学 ShanghaiTech University

ADSP

Convergence rate SGD

① 保证收敛则步长 $\alpha < \frac{1}{\lambda_i}$

for all i, then $\alpha < \frac{1}{\lambda_{max}}$

② 收敛时间：

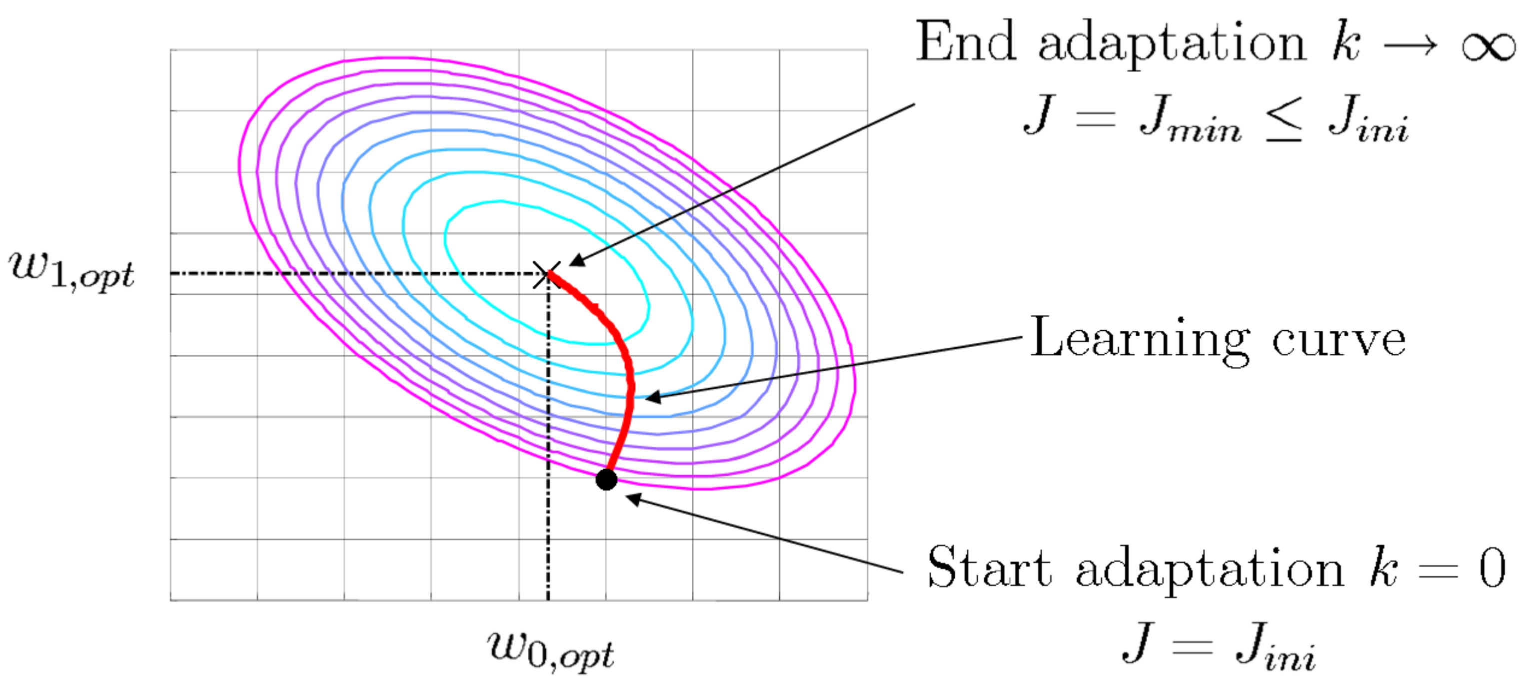
$(\underline{w})_i \to (\underline{w}_o)_i \Rightarrow \tau_i = \frac{1}{2\alpha\lambda_i}$

for all i, $\underline{w} \to \underline{w_o} \Rightarrow \tau = \frac{1}{2\alpha\lambda_{min}}$

1 极之取 $\tau = \frac{1}{\alpha\lambda_{min}} > \frac{1}{\frac{1}{\lambda_{max}} \cdot \lambda_{min}} = \frac{\lambda_{max}}{\lambda_{min}}$

Learning curve in contour plot J

$$\Gamma_x = \frac{\lambda_{max}}{\lambda_{min}} = 3$$



Convergence rate SGD

Learning curves for different α