2.160 Identification, Estimation, and Learning

Lecture Notes No. 3
February 15, 2006

2.3 Physical Meaning of Matrix *P*

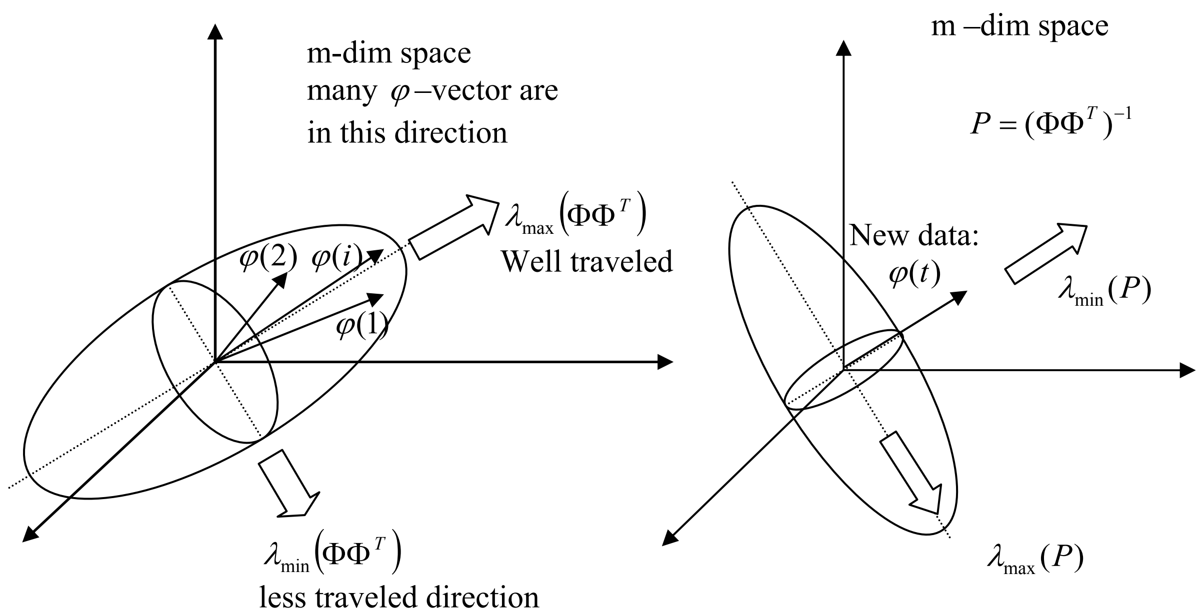
The Recursive Least Squares (RLS) algorithm updates the parameter vector $\hat{\theta}(t-1)$ based on new data $\varphi^T(t)$, $y(t)$ in such a way that the overall squared error may be minimal. This is done by multiplying the prediction error $\varphi^T(t)\hat{\theta}(t-1) - y(t)$ with the gain matrix which contains matrix $\boldsymbol{P_{t-1}}$. To better understand the RLS algorithm, let us examine the physical meaning of matrix $\boldsymbol{P_{t-1}}$.

Recall the definition of the matrix:

$$P_t^{-1} = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) = \Phi\Phi^T \tag{17}$$

$$\Phi = \left[\varphi(1)..\varphi(t)\right] \quad \in R^{m\times t}, \Phi^T \in R^{t\times m}, \Phi\Phi^T \in R^{m\times m}$$

Note that matrix $\Phi\Phi^T$ varies depending on how the set of vectors $\{\varphi(i)\}$ span the *m*-dimensional space. See the figure below.



Geometric Interpretation of matrix P^{-1}.

Since $\Phi\Phi^T \in R^{mxm}$ is a symmetric matrix of real numbers, it has all ***real*** eigenvalues. The eigen vectors associated with the individual eigenvalues are also real. Therefore, the matrix $\Phi\Phi^T$ can be reduced to a diagonal matrix using a coordinate transformation, i.e. using the eigen vectors as the bases.

$$\Phi\Phi^T \Rightarrow D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & \lambda_m \end{pmatrix} \in R^{mxm} \tag{19}$$

$$\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m = \lambda_{\min}$$

$$P = (\Phi\Phi^T)^{-1} \Rightarrow D^{-1} = \begin{pmatrix} 1/\lambda_1 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1/\lambda_m \end{pmatrix} \in R^{mxm} \tag{20}$$

The direction of $\lambda_{\max}(\Phi\Phi^T)$ = The direction of $\lambda_{\min}(P)$.

If $\lambda_{\min} = 0$, then $\det(\Phi\Phi^T) = 0$, and the ellipsoid collapses. This implies that there is no input data $\varphi(i)$ in the direction of $\lambda_{\min}$, i.e. the input data set does not contain any information in that direction. In consequence, the m-dimensional parameter vector θ cannot be fully determined by the data set.

In the direction of $\lambda_{\max}$, there are plenty of input data: $\varphi(i)\cdots$. This direction has been well explored, well excited. Although new data are obtained, the correction to the parameter vector $\hat{\theta}(t-1)$ is small, if the new input data $\varphi(t)$ is in the same direction as that of $\lambda_{\max}$. See the second figure above.

The above observations are summarized as follows:

1) Matrix ***P*** determins the gain of the prediction error feedback

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mathrm{K}_t e(t) \tag{17}$$

where K_t is a varying gain matrix: $\mathrm{K}_t = \dfrac{P_{t-1}\varphi(t)}{\left(1+\varphi^T(t)P_{t-1}\varphi(t)\right)}$

2) If a new-data point $\varphi(t)$ is aligned with the direction of $\lambda_{\max}\left(\Phi\Phi^T\right)$ or $\lambda_{\min}\left(P_{t-1}\right)$, $\varphi^T(t)P_{t-1}\varphi(t) << 1$,

then $\mathrm{K}_t \approx P_{t-1}\varphi(t)$ which is small. Therefore the correction is small.

3) Matrix $\boldsymbol{P}_t$ represents how much data we already have in each direction in the m-dimensional space. The more we already know, the less the error correction gain $\boldsymbol{K}_t$ becomes. Correction $\Delta\theta$ gets smaller and smaller as t tends infinity.

2.4 Initial Conditions and Properties of RLS

a) Initial conditions for $\boldsymbol{P_o}$

P_o does not have to be accurate (close to its correct value), since it is recursively modified. But P_o must be good enough to make the RLS algorithm executable. For this,

P_o must be a ***positive definite*** matrix, such as the identity matrix ***I.*** (21)

Depending on initial values of $\hat{\theta}(0)$ and P_o , the (best) estimation thereafter will be different.

Question: How do the initial conditions influence the estimate? The following theorem shows exactly how the RLS algorithm works, given initial conditions.

Theorem

The Recursive Least Squares (RLS) algorithm minimizes the following cost function:

$$J_t(\theta) = \frac{1}{2}\sum_{i=1}^{t}\left(y(i) - \varphi^T(i)\theta\right)^2 + \frac{1}{2}\left(\theta - \hat{\theta}(0)\right)^T P_0^{-1}\left(\theta - \hat{\theta}(0)\right) \tag{22}$$

where P_o is an arbitrary positive definite matrix (m by m) and $\hat{\theta}(0) \in R^m$ is arbitrary.

Proof Differentiating $J_t(\theta)$

$$\frac{dJ_t(\theta)}{d\theta} = 0 \quad \Longrightarrow \quad -\sum_{i=1}^{t}\left(y(i) - \varphi^T(i)\theta\right)\varphi(i) + P_0^{-1}\left(\theta - \hat{\theta}(0)\right) = 0 \tag{23}$$

Collecting terms

$$\underbrace{\left[\sum_{i=1}^{t}\varphi(i)\varphi^T(i) + P_0^{-1}\right]}_{P_t^{-1}}\theta = \sum_{i=1}^{t} y(i)\varphi(i)\varphi^T(i) + P_0^{-1}\hat{\theta}(0) \tag{24}$$

The parameter vector minimizing (22) is then given by

$$\begin{aligned}\hat{\theta}(t) &= P_t\left[\sum_{i=1}^{t} y(i)\varphi(i)\varphi^T(i) + P_0^{-1}\hat{\theta}(0)\right] \\ &= P_t\left[y(t)\varphi(t) + \underbrace{\sum_{i=1}^{t-1} y(i)\varphi(i)\varphi^T(i) + P_0^{-1}\hat{\theta}(0)}_{}\right]\end{aligned}$$

$$P_{t-1}^{-1}\hat{\theta}(t-1)$$

Recall $P_t^{-1} = \varphi(t)\varphi^T(t) + P_{t-1}^{-1}$

$$\begin{aligned}\hat{\theta}(t) &= P_t\left[P_t^{-1}\hat{\theta}(t-1) + y(t)\varphi(t) - \varphi(t)\varphi^T(t)\hat{\theta}(t-1)\right] \\ &= \hat{\theta}(t-1) + P_t\varphi(t)\left[y(t) - \varphi^T(t)\hat{\theta}(t-1)\right]\end{aligned} \tag{25}$$

Postmultiplying $\varphi(t)$ to both sides of (14)

$$\begin{aligned}P_t\varphi(t) &= P_{t-1}\varphi(t) - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}\varphi(t)}{\left(1+\varphi^T(t)P_{t-1}\varphi(t)\right)} \\ &= \frac{P_{t-1}\varphi(t)}{\left(1+\varphi^T(t)P_{t-1}\varphi(t)\right)}\end{aligned} \tag{26}$$

using (26) in (25) yields (18), the RLS algorithm,

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\left(1+\varphi^T(t)P_{t-1}\varphi(t)\right)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \tag{18}$$

Q.E.D.

Discussion on the Theorem of RLS

$$\hat{\theta}(t) = \arg\min_{\theta}\left[\underbrace{\frac{1}{2}\sum_{i=1}^{t}\left(y(i) - \varphi^T(i)\theta\right)^2}_{\substack{\text{Squared estimation error} \\ A}} + \underbrace{\frac{1}{2}\left(\theta - \hat{\theta}(0)\right)^T P_0^{-1}\left(\theta - \hat{\theta}(0)\right)}_{\substack{\text{Weighted squared distance from } \hat{\theta}(0) \\ \text{B}}}\right] \tag{27}$$

1) As t gets larger, more data are obtained and term A gets overwhelmingly larger than term B. As a result, the influence of initial conditions fades out.
2) In an early stage, i.e. small time index t, θ is pulled towards $\hat{\theta}(0)$, particularly when the eigenvalues of matrix P_0^{-1} are large.
3) In contrast, if the eigenvalues of P_0^{-1} are small, θ tends to change more quickly in response to the prediction error, $y(t) - \varphi^T(t)\theta$.
4) The initial matrix P_0 represents the level of ***confidence*** for the initial parameter value $\hat{\theta}(0)$.

Note: The ***P*** matrix involved in RLS with an initial condition $\boldsymbol{P_o}$ has been extended in the RLS theorem from the batch processing case of $P_t^{-1} = \sum_{i=1}^{t}\varphi(i)\varphi^T(i)$ to:

$$P_t^{-1} = \sum_{i=1}^{t}\varphi(i)\varphi^T(i) + P_o^{-1} \tag{28}$$

Other important properties of RLS include:

- Convergence of $\theta(t)$. It can be shown that

$$\lim_{t\to\infty}\left\|\hat{\theta}(t)-\hat{\theta}(t-1)\right\|=0 \tag{29}$$

  See Goodwin and Sin's book, Ch.3, for proof.

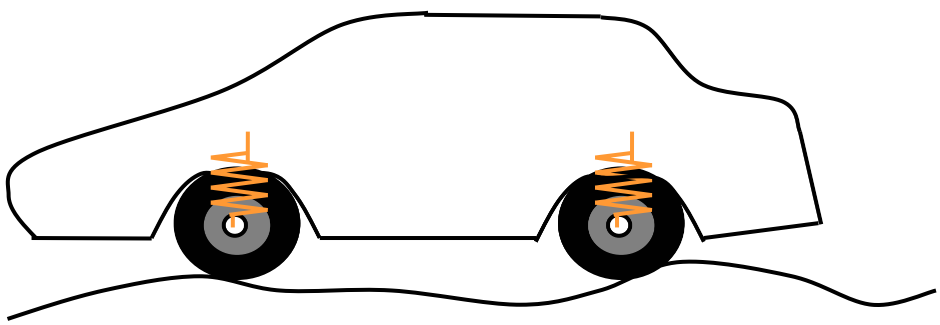
- The change to the ***P*** matrix: $\Delta P = P_t - P_{t-1}$ is negative semi-definite, i.e.

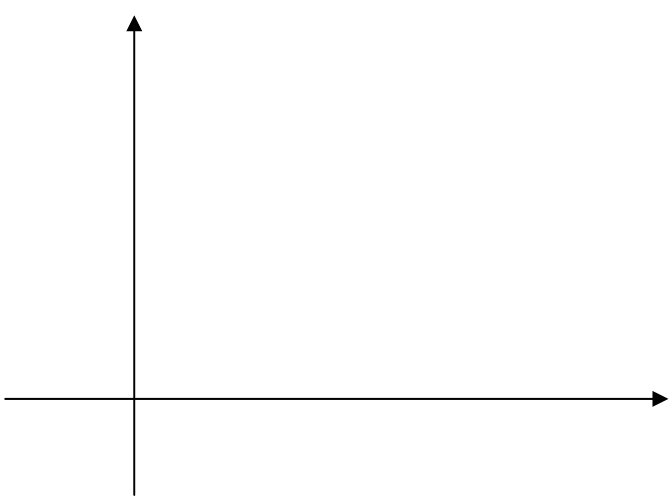
$$\Delta P = -\frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{\left(1+\varphi^T(t)P_{t-1}\varphi(t)\right)} \le 0 \tag{30}$$

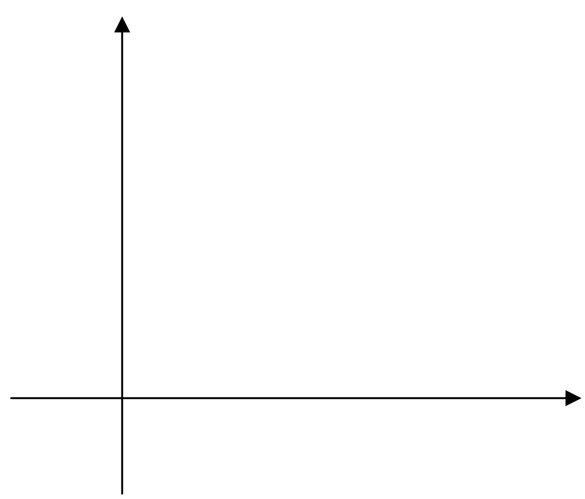
  for an arbitrary $\varphi(t) \in R^m$ and positive definite P_{t-1} .

  Exercise Prove this property.

2.5 Estimation of Time-varying Parameters



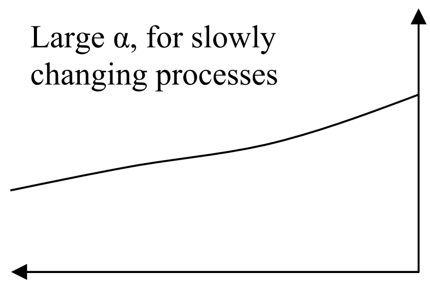


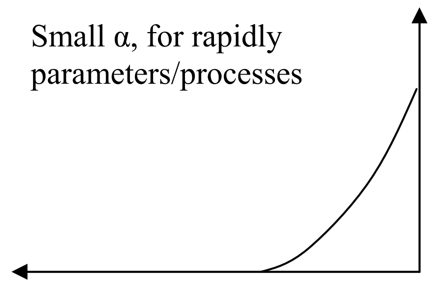


Least Squares with Exponential Data Weighting

Forgetting factor: $\boldsymbol{\alpha}$

$$0 < \alpha \le 1 \qquad (31)$$





Weighted Squared Error

$$J_t(\theta) = \sum_{i=1}^{t} \alpha^{t-i} e^2(i) \qquad (32)$$

$$\hat{\theta}(t) = \arg\min_{\theta} J_t(\theta) \qquad (33)$$

$\hat{\theta}(t)$ is given by the following recursive algorithm.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\left(\alpha + \varphi^T(t)P_{t-1}\varphi(t)\right)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \qquad (34)$$

$$P_t = \frac{1}{\alpha}\left[P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{\left(\alpha + \varphi^T(t)P_{t-1}\varphi(t)\right)}\right] \qquad (35)$$

Exercise: Obtain (34) and (35) from (32) and (33).

A drawback of the forgetting factor approach

When the system under consideration enters "steady state", the matrix $P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}$ tends to the null matrix. This implies

$$P_t \approx \frac{1}{\alpha}P_{t-1} \qquad (36)$$

As α<1, 1/ α makes P_t larger than P_{t-1}. Therefore { P_t } begins to increase exponentially.
The "Blow-Up" problem

<u>Remedy:</u>

Covariance Re-setting Approach

- The forgetting factor approach has the "Blow-Up" problem
- The ordinary RLS
  The P matrix gets small after some iterations (typically 10-20 iterations). Then the gain dramatically reduces, and $\hat{\theta}$ is no longer varying.

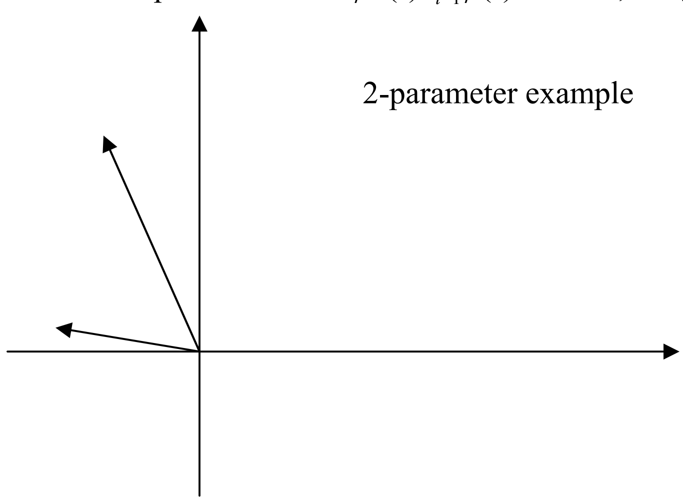
  The Covariance Re-Setting method is to solve these shortcomings by occasionally re-setting the P matrix to:
  $$P_{t^*} = kI \qquad 0 < k < \infty \tag{37}$$
  This re-vitalizes the algorithm.

2.6 Orthogonal Projection

The RLS algorithm provides an iterative procedure to converge to its final parameter value. This may take more than *m*(dimension of θ)steps.
The Orthogonal Projection algorithm provides the least squares solution exactly in *m* recursive steps

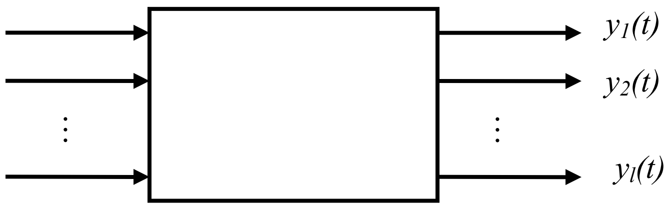
Assume $\Phi_m = [\varphi(1) \quad \varphi(2) \quad \dots \quad \varphi(m)]$ (38)

Spanning the whole *m*-dim space
Set $P_0=I$ (the *m*x*m* identity matrix) and $\hat{\theta}(0)$ arbitrary
Compute
$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\varphi^T(t)P_{t-1}\varphi(t)}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right) \tag{39}$$
where matrix P_{t-1} is updated with the same recursive formula as RLS
Note that +1 involved in the denominator of RLS is eliminated in (39)
This causes a numerical problem when $\varphi^T(t)P_{t-1}\varphi(t)$ is small, the gain is large.



This orthogonal projection algorithm is more efficient, but is very sensitive to noisy data.
Ill-conditioned when $\varphi^T(t)P_{t-1}\varphi(t) \approx 0$. RLS is more robust.

2.6 Multi-Output, Weighted Least Squares Estimation



l-output $\vec{y}(t) = \begin{pmatrix} y_1 \\ \vdots \\ y_l \end{pmatrix} \in R^l$

For each output $\hat{y}_i(t) = \varphi_i^T(t)\theta$

$$\hat{\vec{y}}(t) = \begin{pmatrix} \varphi_1^T \\ \vdots \\ \varphi^T \end{pmatrix} \theta = \Psi^T(t)\theta \qquad \Psi \in R^{l\times m} \tag{40}$$

Error $\vec{e}(t) = \begin{pmatrix} e_1 \\ \vdots \\ e \end{pmatrix} = \vec{y}(t) - \Psi^T(t)\theta$ (41)

Consider that each squared error is weighted differently, or
Weighted Multi-Output Squared Error:

$$J_t(\theta) = \sum_{i=1}^{t} \vec{e}^T(i) W \vec{e}(i) = \sum_{i=1}^{t} (\vec{y}(i) - \Psi^T\theta)^T W (\vec{y}(i) - \Psi^T\theta) \tag{42}$$

$$\hat{\theta}(t) = \arg\min_{\theta} J_t(\theta) \quad \Longrightarrow \quad \hat{\theta}(t) = \mathrm{P}_t \mathrm{B}_t$$

$$\mathrm{P}_t = \left[\sum_{i=1}^{t} \Psi^T(i) W \Psi(i) \right]^{-1}$$

$$\mathrm{B}_t = \sum_{i=1}^{t} \Psi^T(i) W \vec{y}(i) \tag{43}$$

The recursive algorithm

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mathrm{P}_t \Psi(t) W \left(\vec{y}(t) - \Psi^T(t)\hat{\theta}(t-1) \right) \tag{44}$$

$$\mathrm{P}_t = \mathrm{P}_{t-1} - \mathrm{P}_{t-1}\Psi(t)\left[W^{-1} + \Psi^T(t)\mathrm{P}_{t-1}\Psi(t) \right]^{-1} \Psi^T(t)\mathrm{P}_{t-1}$$