Lecture 5

Wednesday, January 26, 2022 9:25 AM

Least Squares Parameter Estimation for Linearly Parameterized Systems

Admin

1. Today's OH: 2:30-3:30pm { Bluejeans / IBB 3303
2. Lecture notes posted
   -> clarification of math in lecture 4
3. HW: put refs.

Last time

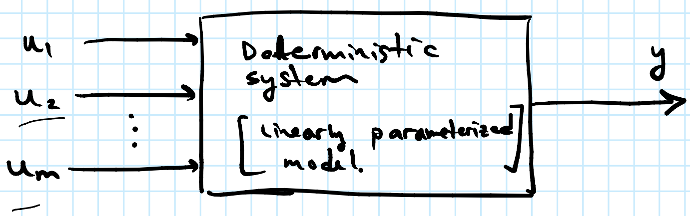
1. Used least squares to fit a line.
   -> estimated two parameters, $\hat{\beta}_1$, $\hat{\beta}_0$

Objective today

1. Formulation of least squares for linear parameter estimation.
2. Recursive least squares (RLS) algorithm.

Least Squares Parameter Estimation for Deterministic Systems

- Goal: Create a mathematical model to predict future behaviors.
  - Consider a multivariate deterministic system



linearly parameterized means that input-output relationship takes the form:

$$y = \underbrace{b_1 u_1 + b_2 u_2 + \cdots + b_m u_m}_{\text{parameters to estimate.}}$$

- parameters to estimate: $\theta = [b_1 \; b_2 \cdots b_m]^T \in \mathbb{R}^m$ ($c_1 \; c_2 \ldots c_n$) ($m \times 1$)
- observable input variable

$$\varphi = [u_1 \; u_2 \; \cdots \; u_m]^T \in \mathbb{R}^m \quad (m \times 1)$$

- output variable

$$y = \varphi^T \theta \qquad (1)$$

- Our problem: Find the system parameters θ from our observed data

$$\text{observable inputs}\left\{\begin{matrix} \varphi(1), & y(1) \\ \varphi(2), & y(2) \\ \vdots & \\ \varphi(N), & y(3) \end{matrix}\right\}\text{observable outputs} \quad \Bigg\} \text{ use these data to estimate } \hat{\theta}$$

- The system may be a linear <u>dynamic</u> system

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_m u(t-m)$$

$$\varphi(t) = [u(t-1) \quad u(t-2) \quad \dots \quad u(t-m)]^T$$

- The system may be a nonlinear dynamic system:

$$y(t) = b_1 u(t-1) + b_2 \underbrace{u(t-1)\,u(t-2)}_{\text{nonlinear}} + \dots \leftarrow \text{true system}$$

$$\varphi(t) = [u(t-1) \quad u(t-1)\,u(t-2)]$$

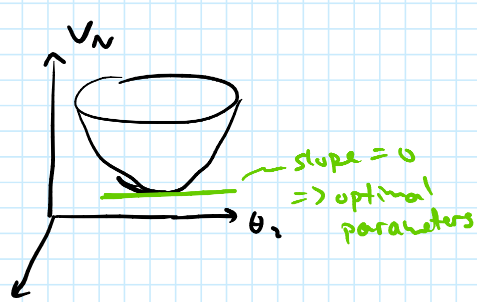
→ Key: parameters are linearly involved

- using $\hat{\theta}$, we can create an output predictor:

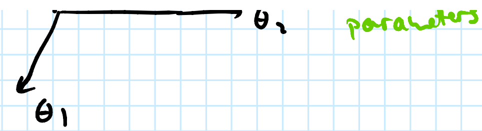
$$\hat{y}(t|\hat{\theta}) = \varphi^T(t)\,\hat{\theta} \leftarrow \text{best estimate using experimental data} \qquad (2)$$

- How do we determine $\hat{\theta}$?

$$\underset{\substack{\uparrow \\ \text{\# of data points}}}{V_N(\hat{\theta})} = \frac{1}{N}\sum_{t=1}^{N} \underbrace{\left(\hat{y}(t|\hat{\theta}) - y(t)\right)^2}_{\text{squared error.}} \qquad (3)$$



# of data points   squared error.



- We want to minimize $V_N(\theta)$ by finding the best $\hat{\theta}$

$$\hat{\theta} = \arg\min_{\theta} V_N(\theta) \qquad (4)$$

- We minize $V_N(\theta)$ by differentiating w.r.t θ and setting $= 0$.

$$\frac{\partial V_N(\theta)}{\partial \theta} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} \left\{ \left[\varphi^T(t)\theta - y(t) \right] \varphi(t) \right\} = 0. \qquad (5)$$

- Seperate into terms with θ or without

$$\underbrace{\left\{ \sum_{t=1}^{N} \overbrace{\left[\varphi(t)\, \varphi^T(t) \right]}^{m \times m} \right\}}_{\text{auto-covariance matrix}} \underbrace{\theta}_{m \times 1} = \underbrace{\sum_{t=1}^{N \,\leftarrow\, \#\text{ samples}} y(t)\, \varphi(t)}_{\text{Cross covariance vector}} \qquad (6)$$

$$\underbrace{\Phi}_{(m \times N)} \underbrace{\Phi^T}_{(N \times m)}, \text{ where } \Phi = [\varphi(1), \varphi(2) \cdots \varphi(N)]^T$$

<u>Note</u>: $m \leq N$

- If vectors $\varphi(1), \varphi(2), \cdots \varphi(N)$ span the entire m-dimensional space then,

  rank $\Phi = m \Rightarrow$ full rank matrix

  $\Rightarrow$ the matrix $\Phi \Phi^T$ is <u>invertible</u>

- Then, using Eqn 6), we find $\hat{\theta}$:

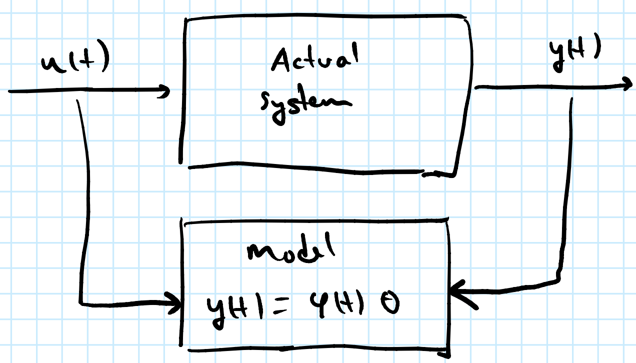
$$\hat{\theta} = \underbrace{\left[\sum_{t=1}^{N} \left(\varphi(t)\, \varphi^T(t) \right) \right]^{-1}}_{P = \left(\Phi \Phi^T \right)^{-1} \quad (8)} \underbrace{\sum_{t=1}^{N} y(t)\, \varphi(t)}_{= B \quad (9)}$$

$\Rightarrow \hat{\theta} = PB \leftarrow$ Best parameter estimate given N data points (10)

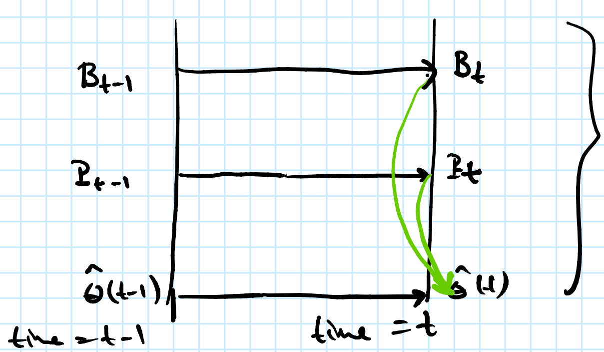
$\Rightarrow \hat{\theta} = PB \Leftarrow$ Best parameter estimate given N data points (10)

Recursive Least Squares Algorithm

- Least squares (Above) is useful for batch processing where all data are already collected
  - $\Rightarrow$ no more to consider for estimating $\hat{\theta}$
- when we are estimating parameters in real-time (i.e. online updating $\hat{\theta}$ is required), a recursive computing method is preferred.
  - Reduce computational overhead. $\Rightarrow$ faster computation
  - No need to store all of the data. $\Rightarrow$ higher frequency.



$\Downarrow$ process steps for updating.



Goal: update our variables and estimated parameters based only on $(t-1)$

$\Rightarrow$ **Next time**