EECS 16B Designing Information Devices and Systems II
Fall 2020 UC Berkeley **Note 15**

1 Module Overview

Now that we have grapsed a good understanding of dynamical systems and how to control them, we will be moving onto a new module. In the next set of notes, we will be focusing on Linear Algebra and how we can set up optimization problems to better **learn** and control our systems.

The subject of mathematical optimization is ubiquitous in many fields of study even outside Electrical Engineering or Control Systems. A common theme throughout this module will be to minimize some cost function inherent to our problem subject to a series of constraints.

The first problem that we will discuss is how to identify an unknown system through Least-Squares. This is formally referred to as **System Identification**.

2 Least-Squares

Let us recap the Least-Squares problem from 16A. Given a set of **observations** $y_i|_{i=1}^{m}$, we would like to explain our observations using a set of **features** $x_k|_{k=1}^{n}$.

The heart of Least-Squares assumes that this relation between y and x_i is *linear* meaning

$$y_i = \alpha_1 x_{i1} + \alpha_2 x_{i2} + \ldots + \alpha_{in} x_{in} + e_i \tag{1}$$

where e_i is a term accounting for the noise in our measurements of y_i.

We can set up a matrix vector equation by aggregating our features into a matrix A

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ddots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{mn} & x_{m2} & \ldots & x_{mn} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = A\vec{x} + \vec{e} \tag{2}$$

An alternate way to phrase the least-squares problem as an optimization problem is as follows

$$\min_{\vec{x} \in \mathbb{R}^n} \|\vec{e}\| \quad \text{subject to} \quad \vec{e} = A\vec{x} - \vec{y} \tag{3}$$

This gives the perspective that Least-Squares is searching for $\vec{x}$ that minimizes the error $\vec{e}$ between $\vec{y}$ and $A\vec{x}$. We won't derive the solution here, but remember from 16A/Math 54 that the solution to this problem is

$$\vec{x}^* = (A^T A)^{-1} A^T \vec{y} \tag{4}$$

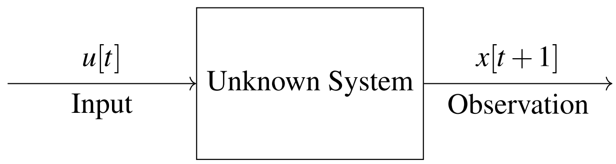
3 Scalar Systems

Now that we have gone over the Least-Squares problem, let's try to understand how we can use it to identify an unknown scalar system.

3.1 Simple Linear Model

Suppose we had an unknown linear system treated as a black-box model where we can give inputs and observe outputs but the parameters a and b are unknown.

$$x[t+1] = ax[t] + bu[t] + w[t]$$



In order to estimate the parameters a and b we can observe the intial state $x[0]$, give a sequence of inputs, and observe the following outputs

$$\begin{bmatrix} u[0] & u[1] & \dots & u[k] \end{bmatrix} \implies \begin{bmatrix} x[1] & x[2] & \dots & x[k+1] \end{bmatrix}$$

Using this information, we have a collection of k equations that we can aggregate into a matrix-vector equation

$$\vec{y} = \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[k+1] \end{bmatrix} = \begin{bmatrix} x[0] & u[0] \\ x[1] & u[1] \\ \vdots & \vdots \\ x[k] & u[k] \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[t] \end{bmatrix} = D\vec{p} + \vec{e} \tag{5}$$

The vector $\vec{y}$ holds the observations, the matrix D represents our data, $\vec{p}$ holds our parameters and $\vec{e}$ accounts for the error in our measurements. Since this is a Least-Squares problem, we can best estimate $\vec{p}$ as

$$\boxed{\vec{p}^* = (D^T D)^{-1} D^T \vec{y}} \tag{6}$$

Recall that in order for there to be a unique solution, the matrix D must be full rank or have linearly independent columns. We will revisit this at the end of the note.

3.2 "Nonlinear" Models

Now suppose we had more information about the system and knew that it can be represented as

$$x[t+1] = a_0 x[t] + a_1 e^{x[t]} + b_0 u[t] + b_1 u[t]^2 + w[t] \tag{7}$$

Although this system is "nonlinear," it is still linear with respect to its features so we can still use Least-Squares to estimate its parameters

$$\vec{y} = \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[k+1] \end{bmatrix} = \begin{bmatrix} x[0] & e^{x[0]} & u[0] & u[0]^2 \\ x[1] & e^{x[1]} & u[1] & u[1]^2 \\ \vdots & \vdots & \vdots & \vdots \\ x[k] & e^{x[k]} & u[k] & u[k]^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[t] \end{bmatrix} = D\vec{p} + \vec{e} \tag{8}$$

Again the least squares solution should be identical.

$$\vec{p}^* = (D^T D)^{-1} D^T \vec{y}$$

4 Vector Systems

The procedure of performing System Identification of a vector system that evolves over time is near identical. Suppose we had the following discrete-time system with $A \in \mathbb{R}^{2\times 2}$, $B \in \mathbb{R}^{2\times 2}$, and error $\vec{w} \in \mathbb{R}^2$.

$$\vec{x}[t+1] = A\vec{x}[t] + B\vec{u}[t] + \vec{w}[t] \tag{9}$$

Given a set of inputs $(\vec{u}[0], \dots, \vec{u}[k])$, let's try writing out the system of equations.

$$\begin{aligned} x_1[1] &= a_{11}x_1[0] + a_{12}x_2[0] + b_{11}u_1[0] + b_{12}u_2[0] + w_1[0] \\ x_2[1] &= a_{21}x_1[0] + a_{22}x_2[0] + b_{21}u_1[0] + b_{22}u_2[0] + w_2[0] \\ &\vdots \\ x_1[k+1] &= a_{11}x_1[k] + a_{12}x_2[k] + b_{11}u_1[k] + b_{12}u_2[k] + w_1[k] \\ x_2[k+1] &= a_{21}x_1[k] + a_{22}x_2[k] + b_{21}u_1[k] + b_{22}u_2[k] + w_2[k] \end{aligned}$$

This can be aggregated into the following matrix-vector equation

$$\underbrace{\begin{bmatrix} x_1[0] & x_2[0] & 0 & 0 & u_1[0] & u_2[0] & 0 & 0 \\ 0 & 0 & x_1[0] & x_2[0] & 0 & 0 & u_1[0] & u_2[0] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1[k] & x_2[k] & 0 & 0 & u_1[k] & u_2[k] & 0 & 0 \\ 0 & 0 & x_1[k] & x_2[k] & 0 & 0 & u_1[k] & u_2[k] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ b_{11} \\ b_{12} \\ b_{21} \\ b_{22} \end{bmatrix}}_{\vec{p}} = \underbrace{\begin{bmatrix} x_1[1] \\ x_2[1] \\ \vdots \\ x_1[k+1] \\ x_2[k+1] \end{bmatrix}}_{\vec{y}} + \underbrace{\begin{bmatrix} w_1[0] \\ w_2[0] \\ \vdots \\ w_1[k] \\ w_2[k] \end{bmatrix}}_{\vec{e}} \tag{10}$$

We can swap the rows and columns to rewrite this as two problems.

$$\underbrace{\begin{bmatrix} x_1[0] & x_2[1] & u_1[0] & u_2[0] \\ x_1[1] & x_2[1] & u_1[1] & u_2[1] \\ \vdots & \vdots & \vdots & \vdots \\ x_1[k] & x_2[k] & u_1[k] & u_2[k] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a_{11} \\ a_{12} \\ b_{11} \\ b_{12} \end{bmatrix}}_{\vec{p}_1} = \underbrace{\begin{bmatrix} x_1[1] \\ x_1[2] \\ \vdots \\ x_1[k+1] \end{bmatrix}}_{\vec{y}_1} + \underbrace{\begin{bmatrix} w_1[0] \\ w_1[1] \\ \vdots \\ w_1[k] \end{bmatrix}}_{\vec{w}_1}$$

$$\underbrace{\begin{bmatrix} x_1[0] & x_2[1] & u_1[0] & u_2[0] \\ x_1[1] & x_2[1] & u_1[1] & u_2[1] \\ \vdots & \vdots & \vdots & \vdots \\ x_1[k] & x_2[k] & u_1[k] & u_2[k] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a_{21} \\ a_{22} \\ b_{21} \\ b_{22} \end{bmatrix}}_{\vec{p}_2} = \underbrace{\begin{bmatrix} x_2[1] \\ x_2[2] \\ \vdots \\ x_2[k+1] \end{bmatrix}}_{\vec{y}_2} + \underbrace{\begin{bmatrix} w_2[0] \\ w_2[1] \\ \vdots \\ w_2[k] \end{bmatrix}}_{\vec{w}_2}$$

Finally, we can combine the two equations back into one matrix-**matrix** equation.

$$\underbrace{\begin{bmatrix} x_1[0] & x_2[1] & u_1[0] & u_2[0] \\ x_1[1] & x_2[1] & u_1[1] & u_2[1] \\ \vdots & \vdots & \vdots & \vdots \\ x_1[k] & x_2[k] & u_1[k] & u_2[k] \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix}}_{P} = \underbrace{\begin{bmatrix} x_1[1] & x_2[1] \\ x_1[2] & x_2[2] \\ \vdots & \\ x_1[k+1] & x_2[k+1] \end{bmatrix}}_{Y} + \underbrace{\begin{bmatrix} w_1[0] & w_2[0] \\ w_1[1] & w_2[1] \\ \vdots & \\ w_1[k] & w_2[k] \end{bmatrix}}_{W} \tag{11}$$

This matrix Least-Squares problem will have a similar solution of the form

$$P^* = (D^T D)^{-1} D^T Y \tag{12}$$

4.1 Dimensional Analysis

Suppose we had the following discrete-time system where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times d}$, and $\vec{w} \in \mathbb{R}^n$.

$$\vec{x}[t+1] = A\vec{x}[t] + B\vec{u}[t] + \vec{w}[t] \tag{13}$$

This would imply that the total number of unknowns in our system is $n^2 + nd$ meaning our P matrix will have $n^2 + nd$ entries. At each time-step k, we give d inputs and observe n outputs

$$\begin{bmatrix} u_1[k] & u_2[k] & \dots & u_d[k] \end{bmatrix} \implies \begin{bmatrix} x_1[k+1] & x_2[k+1] & \dots & x_n[k+1] \end{bmatrix} \tag{14}$$

This gives us a total of n observations at each time-step. Recall that if we have m unknowns in our system, we need at least m equations in order to be able to solve for a unique solution. Therefore, from this analysis, we see that this state-space system will require at least $n+d$ time-steps of observations in order to fully estimate the A and B matrices.

Note that the matrix D will be very large with size $(kn) \times (n^2 + nd)$ where $k > n+d$. We will see in a later note on how we can approximate this data matrix and solve least-squares more efficiently.

5 Data Matrix

Whenever we peform Least-Squares, the data matrix D must have linearly independent columns in order to have a unique solution. If D is not a matrix of full-rank, then the matrix $D^T D$ will be non-invertible.

5.1 Quick Lemma

To show this, we can show that the two subspaces of a matrix A, $\text{Nul}(A) = \text{Nul}(A^T A)$ are equivalent. This can be done by showing that the two sets are subsets of each other.

Suppose $\vec{x} \in \text{Nul}(A)$. Then $A\vec{x} = \vec{0}$. Left multiplying by A^T, it follows that

$$A^T A\vec{x} = A^T\vec{0} = \vec{0} \implies \vec{x} \in \text{Nul}(A^T A) \tag{15}$$

Now suppose $\vec{x} \in \text{Nul}(A^T A)$. Then $A^T A\vec{x} = \vec{0}$. Left multiplying by $\vec{x}^T$, it follows that

$$\vec{x}^T A^T A\vec{x} = \|A\vec{x}\|^2 = \vec{0} \implies A\vec{x} = \vec{0} \implies \vec{x} \in \text{Nul}(A) \tag{16}$$

Therefore, since the two sets are subsets of each other, we conclude that $\text{Nul}(A) = \text{Nul}(A^T A)$.

5.2 Rank-Nullity Theorem

So how does this fact help us understand that when $D^T D$ is invertible? It turns out that the Rank-Nullity Theorem holds this answer. If D is an $m \times n$ matrix, then $D^T D$ is an $n \times n$ matrix.

The Rank-Nullity Theorem states that for an $m \times n$ matrix A,

$$\text{Rank}(A) + \dim \text{Nul}(A) = n \tag{17}$$

Therefore, we can show that since $\text{Nul}(A) = \text{Nul}(A^T A)$, $\text{Rank}(A) = \text{Rank}(A^T A)$. For $D^T D$ to be invertible, it must be of rank n and it follows that D is equivalently of rank n.

Contributors:

- Taejin Hwang.

