# Lecture 11: Parameter Estimation via Linear Regression

When modeling real dynamical systems it s often the case that a subset of the model parameters can only be esimated are are entrely unknown. The goal of parameter estimation is to determine these parameter values from experimental data. Consider the second-order dynamical system

$$\ddot{x} + a\dot{x} + bx = u(t) \tag{1}$$

which can be re-written in state-space form as

$$\dot{x}_1 = x_2 \tag{2}$$

$$\dot{x}_2 = -ax_2 - bx_1 + u(t) \tag{3}$$

where we've introduced the new variables $x_1 = \dot{x}$ and $x_2 = \ddot{x}$. Suppose that both parameters $a$ and $b$ are unknown. If we have access to this system and can perform experiments on it then we can sample the input/output states and collect a set of $N$ data points in the form of $\{u_k, x_{1,k}, x_{2,k}\}$ for $k = 1, \ldots, N$. The goal of parameter estimation is to estimate the $a$ and $b$ given this data. In this lecture, we will discuss an approach to parameter estimation that is based on linear regression.

**Linear Regression.** In linear regression we begin by splitting the data into *explanatory variables* $\{x_1, \ldots, x_n\}$ (also called independent variables) and typically one *response variables* $y$ (also called a dependent variable). (This same process can be repeated for to model additional response variables.) Linear regression then postulates a model of the form

$$y = \theta_0 + \sum_{j=1}^{m} \theta_j \xi_j \tag{4}$$

where $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_m]^T$ is a vector of $m$ parameters to be determined for the variable s $\{\xi_1, \ldots, \xi_m\}$ called *regressors*. The regressors can be linear or nonlinear functions of the explanatory variables and are chosen based on intuition or prior knowledge by the modeler. For example, $\xi_1 = x_1, \xi_2 = x_2, \xi_3 = x_1 x_2$ provides a set of three regressors from two explanatory variables. The parameter $\theta_0$ is called the bias and does not multiply any regressors.

Now suppose the response variable $y$ is corrupted by noise $v \sim \mathcal{N}(0, \sigma^2)$ and we collect $N$ sets of data points. The measured values are

$$z_k = \theta_0 + \sum_{j=1}^{m} \theta_j (\xi_j)_k + v_k \tag{5}$$

for $k = 1, \ldots, N$. Each of the regressors depends on the explanatory variables that we assume are measured without error, i.e., $(\xi_j)_k = \xi_j((x_1)_k, \ldots, (x_m)_k)$. Equation (5) can be rewritten in matrix form as:

$$\boldsymbol{z} = \boldsymbol{\xi}\boldsymbol{\theta} + \boldsymbol{v} \tag{6}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} 1 & (\xi_1)_1 & \cdots & (\xi_m)_1 \\ 1 & (\xi_1)_2 & \cdots & (\xi_m)_2 \\ \vdots & & & \\ 1 & (\xi_1)_N & \cdots & (\xi_m)_N \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{7}$$

where $\boldsymbol{z} = [z_1, \ldots, z_N]^T$ is the *response vector*, and

$$\boldsymbol{\xi} = \begin{bmatrix} 1 & (\xi_1)_1 & \cdots & (\xi_m)_1 \\ 1 & (\xi_1)_2 & \cdots & (\xi_m)_2 \\ \vdots & & & \\ 1 & (\xi_1)_N & \cdots & (\xi_m)_N \end{bmatrix} \tag{8}$$

is a $N \times (m+1)$ matrix of regressors (with a column of ones to multiply the constant bias $\theta_0$) and $\boldsymbol{v} = [v_1, \ldots, v_N]^T$ is a vector of measurement errors. The noise-less version of (6) corresponds to the ideal measurements $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ given by

$$\boldsymbol{y} = \boldsymbol{\xi}\boldsymbol{\theta} \, . \tag{9}$$

**Ordinary Least Square.** For a least-square model we assume that $\boldsymbol{\theta}$ is a deterministic but unknown constant (i.e., not a random vector) and that the random vector $\boldsymbol{v}$ is zero mean and uncorrelated with constant variance, $E[\boldsymbol{v}] = \boldsymbol{0}$ and $E[\boldsymbol{v}^T\boldsymbol{v}] = \sigma^2\boldsymbol{I}$. Let

$$\boldsymbol{\epsilon} = \boldsymbol{z} - \hat{\boldsymbol{y}} = \boldsymbol{z} - \boldsymbol{\xi}\boldsymbol{\theta}$$

represent the difference between the measured response and the model based on the regressors and estimated parameters. An estimate of $\boldsymbol{\theta}$ can be obtained by minimizing the sum of squared differences in $\boldsymbol{\epsilon}$. Define a scalar cost function as

$$J = ||\boldsymbol{\epsilon}||^2 = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}$$

The "best" $\boldsymbol{\theta}$ will be the one that minimizes $J$. We can expand $J$ as follows:

$$\begin{aligned} J &= \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon} \\ &= (\boldsymbol{z} - \boldsymbol{\xi}\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{\xi}\boldsymbol{\theta}) \\ &= (\boldsymbol{z}^{\mathrm{T}} - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^T)(\boldsymbol{z} - \boldsymbol{\xi}\boldsymbol{\theta}) \\ &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - \underbrace{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{\theta}}_{\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z}} - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{\theta} \\ &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} - 2\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{\theta} \end{aligned}$$

where we used the fact that $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{\xi}^T\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z}$ since this is a scalar. Since we are seeking to find the $\boldsymbol{\theta}$ that minimizes $J$ we proceed by taking the derivative of $J$ with respect to $\boldsymbol{\theta}$

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}}\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}\right) - \frac{\partial}{\partial \boldsymbol{\theta}}(2\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z}) + \frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{\theta}) \\ &= 0 - 2\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} + 2\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\boldsymbol{\theta} \end{aligned}$$

**Matrix Calculus.** If the above derivative seems unfamiliar it maybe worthwhile to review some common operations involving gradients in matrix calculus. Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors and $\boldsymbol{A}$ is a matrix where the sizes of the vectors and matrix are such that the following expression make sense. Some useful gradients are

- $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}) = \boldsymbol{y}$
- $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}) = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$
- $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}$
- $\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x}$ (if $\boldsymbol{A}$ is symmetric)

Setting the derivative equal to zero $\partial J/\partial \boldsymbol{\theta} = 0$ and denoting the $\boldsymbol{\theta}$ that satisfies this condition as $\hat{\boldsymbol{\theta}}$ gives:

$$\begin{aligned} \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} &= \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}\hat{\boldsymbol{\theta}} \\ (\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{z} &= \hat{\boldsymbol{\theta}} \end{aligned}$$

The quantity $\boldsymbol{\xi}^{+} \triangleq (\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}}$ is called the *pseudoinverse*.

**Pseudoinverse.** Every matrix $\boldsymbol{A}$ has a unique pseudoinverse $\boldsymbol{A}^{+} \triangleq (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$. If $\boldsymbol{A}$ itself is invertible then $\boldsymbol{A}^{+} = \boldsymbol{A}^{-1}$

Thus, the value

$$\implies \hat{\boldsymbol{\theta}} = \boldsymbol{\xi}^{+}\boldsymbol{z} \tag{10}$$

gives the parameter vector that matches the noisy data best (in a least-squares sense). Similarly, in the noise-free case the true parameter is

$$\boldsymbol{\theta}^{*} = \boldsymbol{\xi}^{+}\boldsymbol{y}$$

The covariance of the estimated parameter error can be computed as follows:

$$\begin{aligned} \mathrm{Cov}[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{*}] &= E[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{*})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{*})^{\mathrm{T}}] \\ &= E[(\boldsymbol{\xi}^{+}\boldsymbol{z} - \boldsymbol{\xi}^{+}\boldsymbol{y})(\boldsymbol{\xi}^{+}\boldsymbol{z} - \boldsymbol{\xi}^{+}\boldsymbol{y})^{\mathrm{T}}] \\ &= E[\boldsymbol{\xi}^{+}(\boldsymbol{z} - \boldsymbol{y})(\boldsymbol{z} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{\xi}^{+})^{\mathrm{T}}] \\ &= \boldsymbol{\xi}^{+}E[(\boldsymbol{z} - \boldsymbol{y})(\boldsymbol{z} - \boldsymbol{y})^{\mathrm{T}}](\boldsymbol{\xi}^{+})^{\mathrm{T}} \\ &= \boldsymbol{\xi}^{+}E[\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}](\boldsymbol{\xi}^{+})^{\mathrm{T}} \\ &= \boldsymbol{\xi}^{+}(\sigma^{2}\boldsymbol{I})(\boldsymbol{\xi}^{+})^{\mathrm{T}} \\ &= \sigma^{2}\boldsymbol{\xi}^{+}(\boldsymbol{\xi}^{+})^{\mathrm{T}} \\ &= \sigma^{2}(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}}((\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}})^{\mathrm{T}} \\ &= \sigma^{2}(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}}(\boldsymbol{\xi}(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}) \\ &= \sigma^{2}((\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1}\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1} \\ \implies \mathrm{Cov}[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{*}] &= \sigma^{2}(\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi})^{-1} \end{aligned}$$

This expression for the covariance can be used to determine the confidence of our estimate of the system parameters.

**Example (Linear 2nd Order System).** Let's return to the example of a second-order system with unknown parameters $a$ and $b$ as introduced in the beginning of this lecture. To write the model (2)–(3) in the form above, let $y = \dot{x}_2 - u(t) = -ax_1 - bx_2$ be the response variable. Here we assume we have access to the system input $u(t)$ and the full system state $x_1(t)$ and $x_2(t)$ (from which we can obtain the state rates $\dot{x}_1(t)$ and $\dot{x}_2(t)$). Now define the regressors $\xi_1 = x_1$ and $\xi_2 = x_2$. Then we have our linear regression model

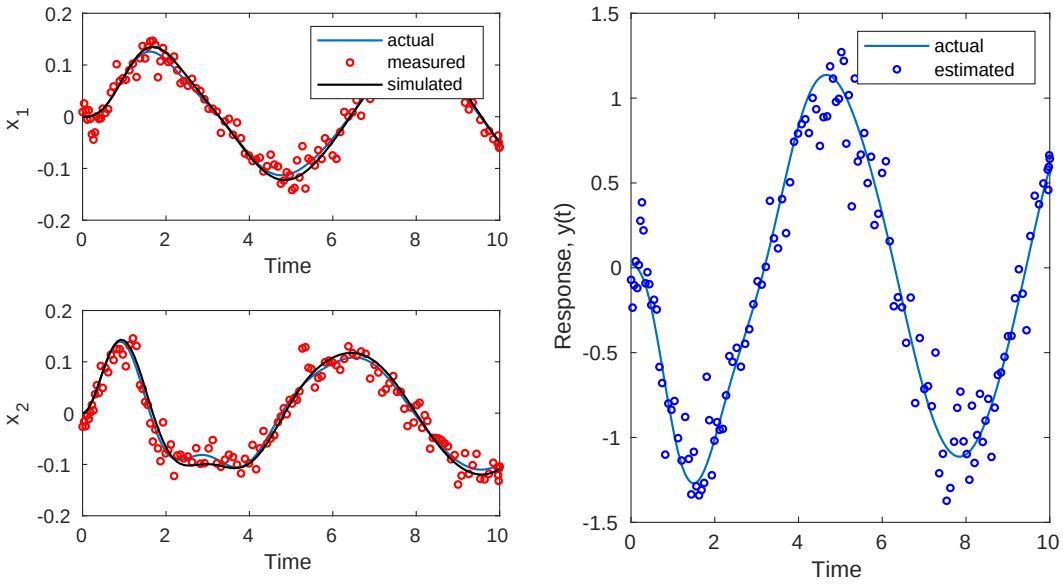
$$y(t) = \theta_0 + \theta_1 \xi_1(t) + \theta_2 \xi_2(t) \tag{11}$$

Suppose we perform an experiment where the system is driven by an input $u(t) = \sin t$ starting from the initial condition $x_1(t_0) = 0$ and $x_2(t_0) = 0$. To perform the regression we require the noise-corrupted response variable $z(t)$ sampled at $N$ time steps

$$\boldsymbol{z} = \begin{bmatrix} (\dot{x}_2)_1 - u_1 \\ (\dot{x}_2)_2 - u_2 \\ \vdots \\ (\dot{x}_2)_N - u_N \end{bmatrix}. \tag{12}$$

These data are shown as the blue markers in the figure below. We also require the explanatory variables (sampled as red dots) which are used to compute the regressor matrix

$$\boldsymbol{\xi} = \begin{bmatrix} 1 & (\xi_1)_1 & \xi_2)_1 \\ 1 & (\xi_1)_2 & (\xi_2)_2 \\ \vdots & & \\ 1 & (\xi_1)_N & (\xi_2)_N \end{bmatrix} \tag{13}$$




Then using (10) we solve for the estimated parameters

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \theta_0 \\ a \\ b \end{bmatrix} = \begin{bmatrix} 0.0172 \\ 1.1693 \\ 9.2404 \end{bmatrix} \tag{14}$$

The actual system has parameters $a = 1$, $b = 0$, and $\theta_0 = 0$ (no bias). If we re-simulate the system from the same initial condition using the estimated parameters we see that the simulated model matches the actual model quite well (left panels above).

**Metric and optimizers.** Other error metrics and optimizers can be considered beyond our least-square ($l_2$ norm), including maximum error ($l_\infty$ norm), mean absolute error ($l_1$ norm). While we derived an analytical expression above the minimization can also be achieved using other optimizers, e.g., `fminsearch` in Matlab.

**Weighted least square.** In our discussion above we treated all data points equally, however weighted least square estimation allows us to assign more or less confidence (i.e., weight) to different data points. This is achieved by replacing the uniform noise variance $E[\boldsymbol{v}^T\boldsymbol{v}] = \sigma^2\boldsymbol{I}$ assumed above with a diagonal matrix $\boldsymbol{R} = E[\boldsymbol{v}^T\boldsymbol{v}]$ that has different $\sigma^2$ values on the diagonal depending on the data points confidence (larger values representing lower confidence). In this case, the cost function to be minimized is also modified as $J = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{\epsilon}$.

**Recursive least square.** In our discussion above the estimation was done offline (i.e., once the data is collected it is batch processed). Online estimation can be achieved using a recursive formulation to estimate parameters in real-time (for example parameters that might change as the system is running). That is, we end up with an estimator of the form

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{\xi}_k^T\hat{\boldsymbol{\theta}}_{k-1})$$

where $\boldsymbol{K}_k$ is an estimator gain that multiplies the residual from the $k$th timestep to update the parameter estimate from the previous $k-1$ timestep. This form is useful for real-time implementation as it avoids having to store and work with increasingly large matrices as more data is added to the system.

**Nonlinear least square.** In some cases a more general relationship between the response variable and regressors is desired, for example, $y = f(\boldsymbol{\xi}, \boldsymbol{\theta})$. The optimality conditions lead to a set of nonlinear equations. There are no general methods for solving such systems, however there are iterative schemes, such as gradient descent, that may be suitable.

## References


- Simon: Chapter 3
- Morelli and Klein: Chapter 5
- Brunton and Kutz: Sec. 4.2