Parametric regression estimation. Part I: the least squares

Przemysław Śliwiński

***Abstract*—Here we start our least-squares journey...**

I. Introduction

A. LS recollection...[1]

We have:

- A random input sequence, $\{x_n\} \in [-\pi, \pi]$ for all $n = \ldots, -1, 0, 1, \ldots$.
- A system
$$m(x) = \sum_{k=1}^{K} \varphi_k(x) \cdot \alpha_k, \tag{1}$$
with some K, where $\{\alpha_k\}$ are unknown, and where, for instance, $\varphi_k(x) = \cos(kx)$.
- A random additive output noise, $\{z_n\} \sim N(0, \sigma_z^2)$.

The system's input-output equation is

$$y_n = m(x_n) + z_n = \sum_{k=1}^{K} \varphi_k(x_n) \cdot \alpha_k + z_n, \text{ for each } n. \tag{2}$$

Given a set of N input-output measurements pairs, $\{(x_n, y_n)\}$, $n = 1, \ldots, N$, the equation above leads to the system of equations

$$\begin{cases} y_1 = \alpha_1 \varphi_1(x_1) + \cdots + \alpha_K \varphi_K(x_1) + z_1 \\ \vdots \\ y_N = \alpha_1 \varphi_1(x_N) + \cdots + \alpha_K \varphi_K(x_N) + z_N \end{cases}$$

which can be represented in a compact form using a matrix notation

$$Y_N = \mathbf{\Phi}_K(X_N) \cdot A_K + Z_N$$

where

$$\begin{aligned} X_N &= \begin{bmatrix} x_1 & & x_N \end{bmatrix}^T, \\ Z_N &= \begin{bmatrix} z_1 & & z_N \end{bmatrix}^T, \\ Y_N &= \begin{bmatrix} y_1 & & y_N \end{bmatrix}^T, \end{aligned}$$

and where $\mathbf{\Phi}_K(X_N)$ is an $N \times K$ (N by K, i.e., with N rows and K columns) matrix of *regressors* (aka *independent variables*, *inputs*)

$$\Phi_K(X_N) = \begin{bmatrix} \varphi_1(x_1) & \cdots & \varphi_K(x_1) \\ \vdots & & \vdots \\ \varphi_1(x_N) & \cdots & \varphi_K(x_N) \end{bmatrix},$$

and where A_K is a vector of K unknown coefficients α_k

$$A_K = \begin{bmatrix} \alpha_1 & & \alpha_K \end{bmatrix}^T. \tag{3}$$

[1] Aka *Machine Learning* (ML) bread and butter/piece of cake/nuts and bolts...

B. Example

Example 1: Let $N = 3$ and $K = 3$, then the matrix of regressors takes a square shape and

$$\Phi_3(X_3) = \begin{bmatrix} \cos x_1 & \cos 2x_1 & \cos 3x_1 \\ \cos x_2 & \cos 2x_2 & \cos 3x_2 \\ \cos x_3 & \cos 2x_3 & \cos 3x_3 \end{bmatrix},$$

and for

$$A_3 = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T$$

we get

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} &= \begin{bmatrix} \cos x_1 & \cos 2x_1 & \cos 3x_1 \\ \cos x_2 & \cos 2x_2 & \cos 3x_2 \\ \cos x_3 & \cos 2x_3 & \cos 3x_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1 \cdot \cos x_1 + \alpha_2 \cdot \cos 2x_1 + \alpha_3 \cdot \cos 3x_1 + z_1 \\ \alpha_1 \cdot \cos x_2 + \alpha_2 \cdot \cos 2x_2 + \alpha_3 \cdot \cos 3x_2 + z_2 \\ \alpha_1 \cdot \cos x_3 + \alpha_2 \cdot \cos 2x_3 + \alpha_3 \cdot \cos 3x_3 + z_3 \end{bmatrix}. \end{aligned}$$

That is

$$Y_3 = \mathbf{\Phi}_3(X_3) \cdot A_3 + Z_3.$$

C. Exercises

Exercise 2: Taking

$$z_n \sim N(0, .01), x_n \sim U[-\pi, \pi],$$

and, *cf.* (1)

$$m(x) = -1 \cdot \cos(x) + 2 \cdot \cos(3x) - 1 \cdot \cos(5x),$$

generate $N = 128$ input-output pairs, *cf.* (2)

$$\{(x_n, y_n = m(x_n) + z_n)\}.$$

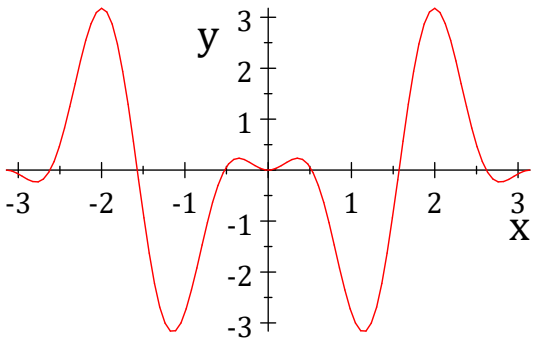

Fig. 1. The system $m(x)$

Exercise 3: Employ the **Matlab CVX**[2] library to find the vector, $\hat{A}_K = \left[\begin{array}{ccc} \hat{\alpha}_1 & \cdots & \hat{\alpha}_K \end{array} \right]$, of the *empirical coefficients* (aka *least-squares estimates*, *parameters*) of the model, *cf.* (1) and (3)

$$\hat{m}\left(x\right) = \sum_{k=1}^{K} \cos\left(kx\right) \cdot \hat{\alpha}_k, \text{ for } K = 5,$$

of the actual system $m\left(x\right)$ using the generated measurement set $\{(x_n, y_n)\}$.

Exercise 4: Take $K = 16, 32$ and 64 and repeat the experiment.

Exercise 5: Compare the results with the outcome of the Gasser-Müller algorithm for the same settings.

II. Code samples

A. CVX

```
% Measurements number N and model size K
N = ...;
K = .. .;
% Matrix of regressors FI and vector of parameters A
FI = ...;
Y = ...;
cvx_begin
    variable A(K)
    minimize(norm(FI * A - Y, 2))
cvx_end
```

B. LS

```
% Measurements number N and model size K
N = ...;
K = .. .;
% Matrix of regressors FI and vector of parameters A
FI = ...;
Y = ...;
A = inv(FI' * FI) * FI' * Y;
%or (Verify, why!)
A = FI\Y;
```

[2] http://cvxr.com/cvx/