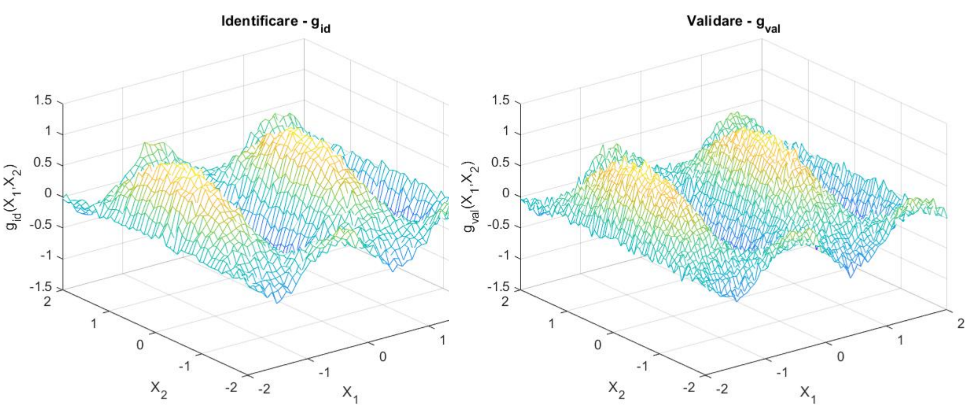
Report

Alexandru Zigler

November 2021

Introduction

Consider the function $g : \mathbb{R}^2 \longrightarrow \mathbb{R}$, given by an identification dataset that provides input-output samples of the function. They are illustrated in Fig. 1 (a). Using linear regression, we create a variable-degree polynomial approximator, in order to obtain a model of the function. This model will be further validated by use of another dataset shown in Fig. 1 (b). We want to find the degree that minimizes the mean square error, relative to the validation dataset, also taking into account the concept of overfitting.



(a) Identification (b) Validation

Figure 1

Structure of the approximator

Because we want the degree m to be variable, we determine the number of terms of the approximator, that we denote by $u(m)$.

$$
\begin{aligned}
u(m) &= u(m-1) + (m+1) \\
u(m-1) &= u(m-2) + m \\
u(m-2) &= u(m-3) + (m-1) \\
&\vdots \\
u(3) &= u(2) + 4 \\
u(2) &= u(1) + 3 \\
u(1) &= 3 \\
\hline
u(m) &= 3 + 3 + 4 + \cdots + m + (m+1) \\
u(m) &= \tfrac{(m+1)(m+2)}{2}
\end{aligned}
$$

We write $g(x_1, x_2) = \theta_1 \cdot 1 + \theta_2 \cdot x_1 + \theta_3 \cdot x_2 + \theta_4 \cdot x_1^2 + \theta_5 \cdot x_1 x_2 + \theta_6 \cdot x_2^2 + \ldots$. The equation has $u(m)$ terms and can be written in matrix form:

$$
\begin{bmatrix}
1 & x_{11} & x_{21} & x_{11}^2 & x_{11}x_{21} & x_{21}^2 & \ldots \\
1 & x_{11} & x_{22} & x_{11}^2 & x_{11}x_{22} & x_{22}^2 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\
1 & x_{1n} & x_{2n} & x_{1n}^2 & x_{1n}x_{2n} & x_{2n}^2 & \ldots
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\ \theta_2 \\ \vdots \\ \theta_{u(m)}
\end{bmatrix}
=
\begin{bmatrix}
y_{11} \\ y_{12} \\ \vdots \\ y_{21} \\ \vdots \\ y_{nn}
\end{bmatrix}
$$

We construct the regression matrix $\Phi \in \mathcal{M}_{n^2, u(m)}$, where n is the dimension of the two input vectors X_1 and X_2.

For m=1, the line is composed of the elements $[1, x_1, x_2]$.

For $m \geq 2$, we keep the elements generated at the previous degree and we add elements corresponding to the current degree. These can be obtained from power combinations of x_1 and x_2 (i.e. $x_1^{m-p} x_2^p$, $p = \overline{0, \mathrm{m}}$).

We solve the system $\Phi \cdot \Theta = Y$, obtaining the parameter set Θ.

$\hat{Y} = \Phi \cdot \Theta$.

MATLAB implementation

We mention a few implementation aspects.

We select x_{1i} and x_{2j} from the input vectors for each line in the matrix Φ, based on $k = \overline{0, \mathrm{n}^2}$.

The function "generateLine" takes the mentioned values and the current degree m, providing a vector that will be used to construct the matrix line by line.

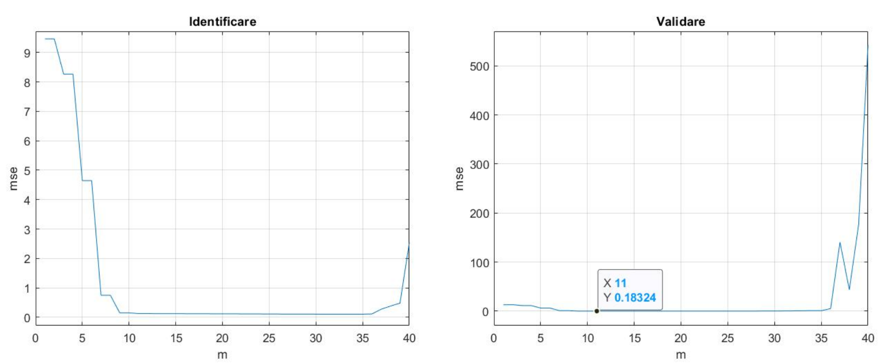
The equation $\Phi \cdot \Theta = Y$ is solved using the "\" operator. The result of the approximation (i.e. $\Phi \cdot \Theta$) has the form of a column vector of size n^2. Using the function "reshape", the vector is rearrange into a matrix $\hat{Y} \in \mathcal{M}_{n,n}$.

Similarly, we compute the matrix Φ corresponding to the validation dataset, then determine the approximated output using the vector of parameters Θ, obtained during the identification step.

We analyze the approximation performance by calculating the mean square error and we repeat the process for each m within the established range.

Tuning results

We calculate the mean square error (mse) for each degree both for identification and validation datasets. These data will be memorized in order to construct the graphs that illustrate the evolution of mean square errors - Fig. 2.



(a) (b)

Figure 2: Evolution of mean square errors

We note that initially ($m = \overline{1,7}$) the errors are high ($mse \in (0.7, 14)$) and they have a decreasing trend. Looking at the validation dataset, the minimum value of the error is 0.183 for m=11. For $m \geq 36$, the errors increase, indicating the onset of overfitting.

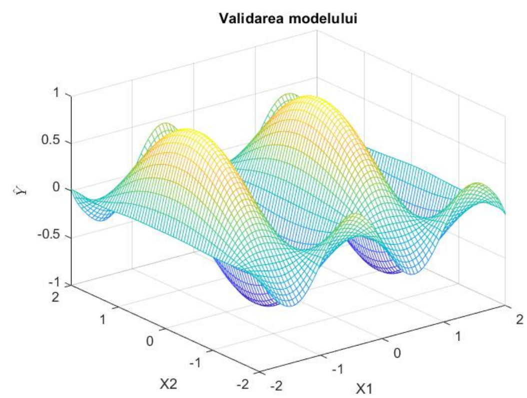


Figure 3: Approximation of the model for minimum mse

Once we determine the optimal degree $m = 11$ that provides the best approximation, we recalculate the matrix $\hat{Y}$ and we represent it graphically in Fig. 3.

Comparing the approximated model $\hat{g}_{val}(X_1, X_2)$ with the provided function $g_{val}(X_1, X_2)$ from Fig. 1(b), we see that $\hat{g}_{val}(X_1, X_2)$ is smoother than $g_{val}(X_1, X_2)$, which is affected by zero-mean noise.

To conclude, using linear regression, we constructed a variable-degree polynomial approximator. The concept of mean square error was essential in evaluating the performance of each degree. Thus, we found the model that best fits the initial function.