Time series modeling using Fourier basis functions

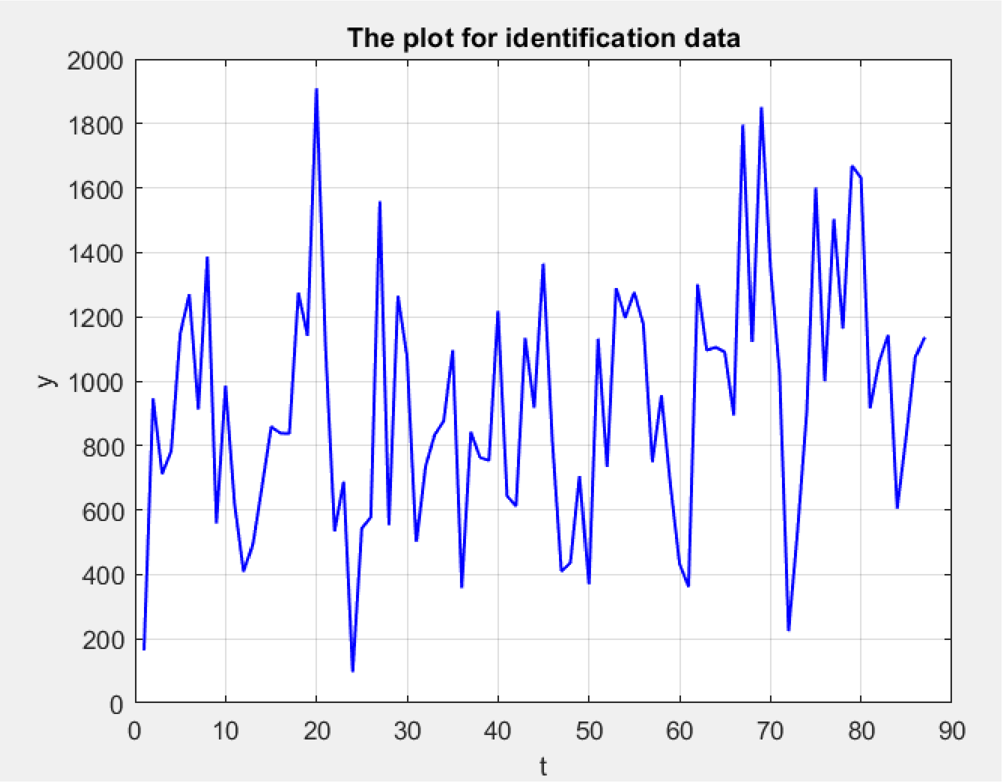
Osiac Maria Teodora
Vălcăuan Adina-Diana

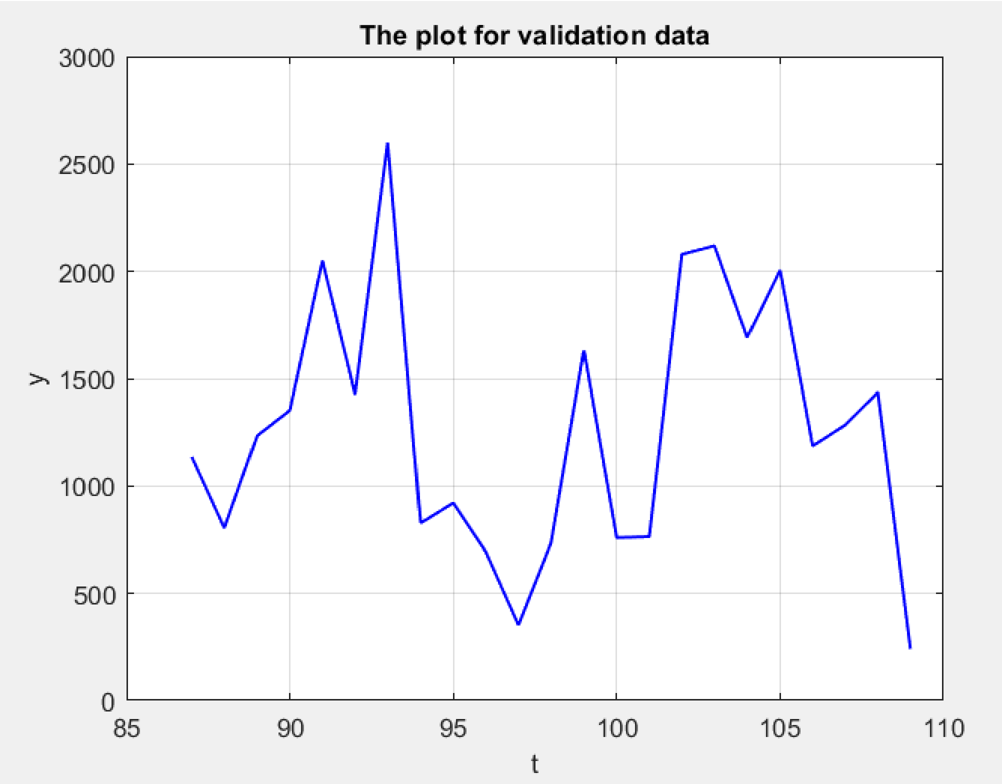
30331/2
product_3

Table of contents

Given dataset

The dataset given by the teacher is: ‘product_30.mat’. All the results presented in this report are based on this file.





Introduction

The project's topic is presented as a shop which sells specific goods. We are given a dataset holding details about the quantities that is being sold and the time. Using Linear Regression, we are supposed to design a mathematical model that shows us the trend of the product's sold quantities over the year and compare the true values with the approximated ones.

At the end, we will be able to estimate the sales amount. Even if it's not going to be precise, the increasing / decreasing trend will be noticeable.

Implementation - part 1

After a long set of tries on the paper, we discovered the algorithm that suited us the best. For the identification part, we created the first matrix with the regressors, that we will discuss about in the next slide. Then, we input theta which contains the parameters and y hat (yH), the estimated values.

The validation algorithm is very similar, the only difference is that we don't need to compute another theta.

Implementation of the approximator

The matrix that made a lot of thinking was the one with the regressors (‘a’ in the script). Having a couple of examples about how a row should look, we decided to try to compute one at a time.

For the first 2 columns it was easier, having 1 and the value from the given index in the time array, then we tried to discover a suitable combination between the column number and the elements of the matrix. We use the syntax: `a(i,2*j+1:2*j+2)` because we start the population from the column number 3 and for every pair of 2 columns, we will have a cos and a sin:

$$\left[\cos\left(\frac{2\pi j * timeId(i)}{P}\right) \quad \sin\left(\frac{2\pi j * timeId(i)}{P}\right)\right]$$

The MSE and best m

We computed one MSE for the validation and one for identification, in order to obtain the best m, using the formula: $mse(m) = \frac{1}{length} .* sum((y - yH)^2)$

mse_id

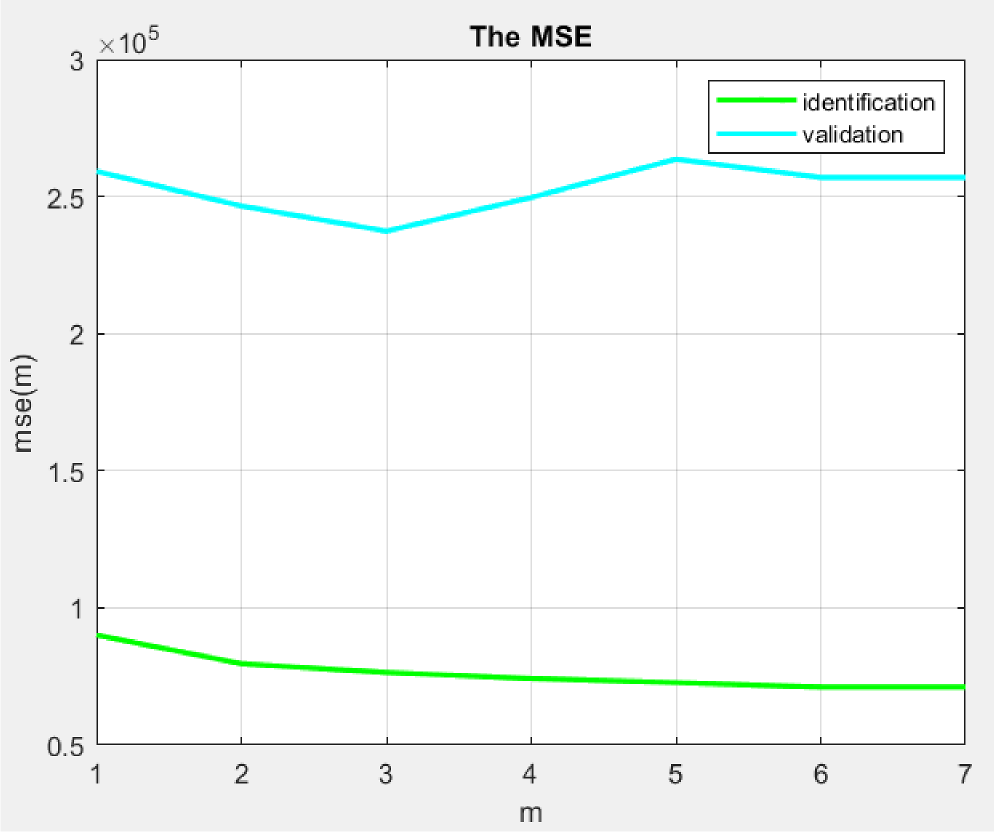
1x7 double

1	2	3	4	5	6	7
9.0080e+04	7.9610e+04	7.6415e+04	7.4240e+04	7.2724e+04	7.1111e+04	7.1111e+04

mse_val

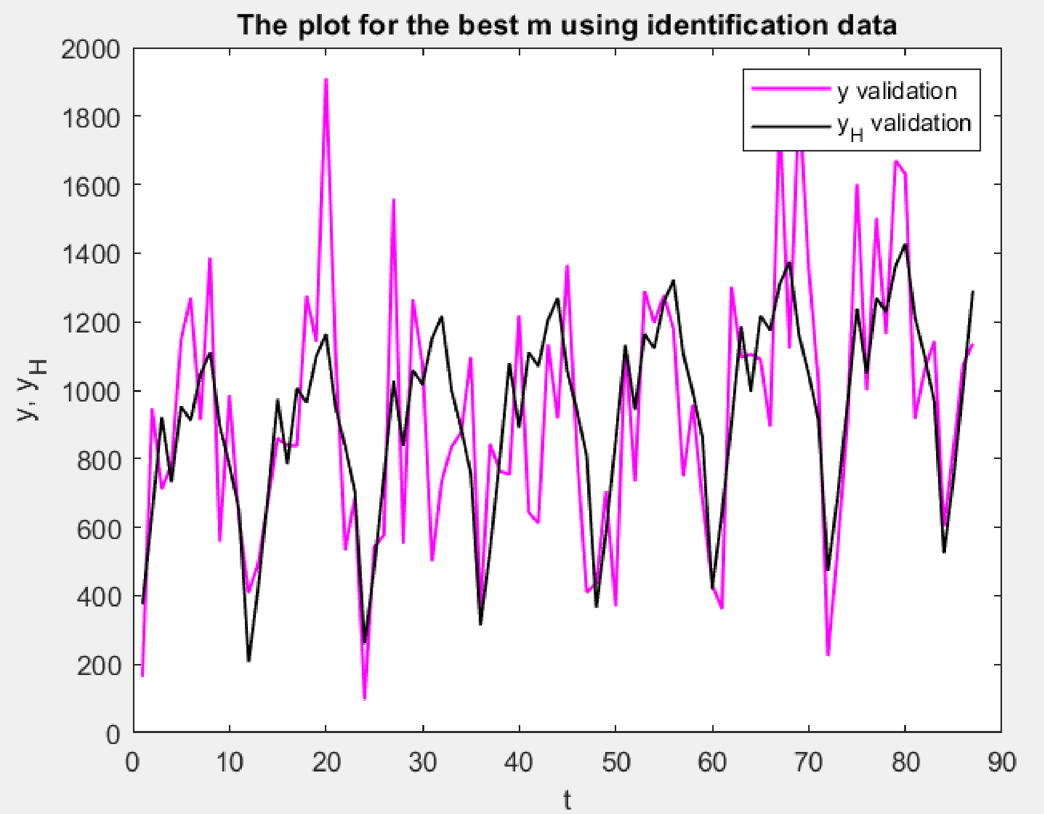
1x7 double

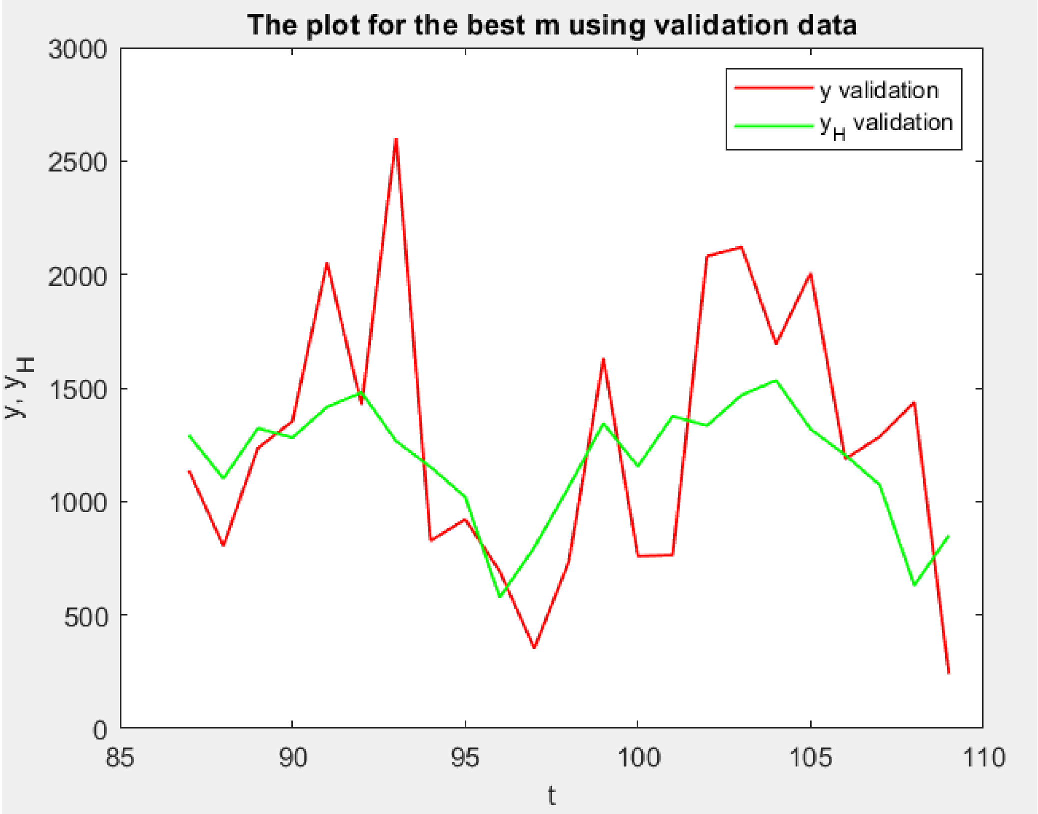
1	2	3	4	5	6	7
2.5913e+05	2.4650e+05	2.3735e+05	2.4962e+05	2.6356e+05	2.5690e+05	2.5690e+05



Looking at the graph and at the tables, the most convenient value is at x = 3, so our best m is 3.
Moreover, in the for-loop we used for computing the mse, we plotted the 2 graphs for every m, just to see ourselves how the graphs change.

Plots for the optimal value





Summary

As a conclusion, the plot with the identification is more precise. This may be because the algorithm was made with that specific data and because we used 80% of the dataset.

However, both plots show that we successfully managed to estimate the trends. This can help the inventory management of the store.

Appendix

```
%% PROJECT 1 -> Time series modeling using Fourier basis function
clc; clear
%% loading the data from the .mat file
load('product_30.mat');
figure
plot(time,y, 'blue', 'LineWidth', 1.2); grid
title("The plot for the data")
ylabel('y'), xlabel('t')
% the values we are going to work with
length1 = length(y);
floor_length = floor(length1 * 0.8); % we choose 80% of the data for identification
% the data for identification
yId = y(1:floor_length);
timeId = time(1:floor_length);
lengthId = length(yId);
figure
plot(timeId, yId, 'blue', 'LineWidth', 1.2);
title("The plot for identification data"); grid
ylabel('y'), xlabel('t')
```

```
% the data for validation
yVal = y(floor_length:length1);
timeVal = time(floor_length:length1);
lengthVal = length(yVal);
figure
plot(timeVal, yVal, 'blue', 'LineWidth', 1.2);
title("The plot for validation data"); grid
ylabel('y'), xlabel('t')
%% the algorithm for varying m
% we created a for loop, which helps us having a varying m, from 1 to 7
% this also helps us with the MSE
for m = 1:7
%% identification algorithm
P = 12; % the given period (because there are 12 months
for i = 1:length(yId) % we iterate through out identification data
a(i,1) = 1; a(i,2) = timeId(i); % we populate the first 2 columns with 1 and the desire variable from time array
for j = 1:m % we start to populate the rest of the columns
a(i,2*j+1:2*j+2) = [cos(2*pi*j*timeId(i)/P) sin(2*pi*j*timeId(i)/P)]; % we use pairs of columns in order to optain the desire form
end
end
```

```
theta = a \ yId; % the parameters matrix computation
yH_id = a*theta; % now we get y_hat -> estimated values
figure
plot(timeId,yId, 'LineWidth', 1.1); hold on % here we plot the 2 graphs together in
plot (timeId,yH_id, 'LineWidth', 1.1);
title("The plot for every m using identification data")
legend('y identification', 'y_H identification')
ylabel('y, y_H'), xlabel('t')
%% validation algorithm
% very similar with the identification one
P = 12;
for i = 1:length(yVal)
b(i,1) = 1; b(i,2) = timeVal(i);
for j = 1:m
b(i,2*j+1:2*j+2) = [cos(2*pi*j*timeVal(i)/P) sin(2*pi*j*timeVal(i)/P)];
end
end
yH_val = b*theta;
figure
plot(timeVal,yVal, 'black', 'LineWidth', 1.1); hold
plot (timeVal,yH_val, 'red', 'LineWidth', 1.1);
```

```
title("The plot for every m using validation data")
legend('y validation', 'y_H validation')
ylabel('y, y_H'), xlabel('t')
%% MSE
mse_id(m) = (1 / lengthId) .* sum((yId - yH_id).^2); % the MSE for the identification
mse_val(m) = (1 / lengthVal) .* sum((yVal - yH_val).^2); % the MSE for the validation
end
% plotting the MSE for identification and verification
n = 1:7;
figure
plot(n, mse_id(n), 'green', 'LineWidth', 2); hold on
plot(n, mse_val(n), 'cyan', 'LineWidth', 2); grid
title("The MSE")
legend('identification', 'validation')
ylabel('mse(m)'), xlabel('m')

%% Plotting the best m
% now that we discover the best value of m, we retrace the algorithm in order to see the bigger picture
m_best = 3;
P = 12;
% identification part
```

```
for i = 1:length(yId)
a(i,1) = 1; a(i,2) = timeId(i);
for j = 1:m_best
a(i,2*j+1:2*j+2) = [cos(2*pi*j*timeId(i)/P) sin(2*pi*j*timeId(i)/P)];
end
end
theta = a \ yId;
yH_id = a*theta;
figure
plot(timeId,yId, 'magenta', 'LineWidth', 1.2); hold on
plot (timeId,yH_id, 'black', 'LineWidth', 1.1);
title("The plot for the best m using identification data")
legend('y validation', 'y_H validation')
ylabel('y, y_H'), xlabel('t')
% validation part
for i = 1:length(yVal)
b(i,1) = 1; b(i,2) = timeVal(i);
for j = 1:m_best
b(i,2*j+1:2*j+2) = [cos(2*pi*j*timeVal(i)/P) sin(2*pi*j*timeVal(i)/P)];
end
end
```

```
yH_val = b*theta;


figure
plot(timeVal,yVal, 'r', 'LineWidth', 1.1); hold
plot (timeVal,yH_val, 'g', 'LineWidth', 1.1);
title("The plot for the best m using validation data")
legend('y validation', 'y_H validation')
ylabel('y, y_H'), xlabel('t')
```