System Identification Project Report November 12, 2022

Incer Maia
Kiss Tamás Norbert
Laza Oana Stefania

Group 30332
Project Index: 6

1 Introduction

This project is meant to treat the topic of model approximation of a linear trend given by the sold items of an engineering store, over the course of a few years.

That being the case, depending on the season, there will be different quantities of the same product sold, according to a certain demand.

In winter time, people need boilers/different heating systems, whereas in the warm months they look for air-cooling systems. The data used for approximation consists of the quantity of product units sold over a number of months.

2 Description

In general, periodic functions are written as the sum of simple waves mathematically represented by sines and cosines.

We were given a data set which contained a set of values that helped us generate a periodic input signal and in order to get the approximation, we needed to analyse its trend (how it goes up and down in a specific time stamp).

As a briefly description of the project, we will present the Fourier function that we used in order to solve the approximation problem and why we thought that this one is the best fit for

what we need.

The Fourier function is one of the most used and powerful functions in mathematical history and knowing that there's no better periodic function than the sinusoidal and cosine one, we chose to use the Fourier series approximation that contained both sin and cos, written as:

$$y(k) = t_0 + t_1 * k + \sum_{i=1}^{m} \left[a_i * cos\left(\frac{2 * \pi * i * k}{P}\right) + b_i * sin\left(\frac{2 * \pi * i * k}{P}\right) \right]$$

This formula consists in a first order element, $t_0 + t_1 * k$, and a Fourier basis depending on the number of samples, m. Choosing the right value for m is one of the most important thing in the approximation problem because we can face a lot of problems like over-fitting (m too big) - which occurs when the model fits exactly against its training data, id error small, val error large and under-fitting (m too small) - meaning that the model makes accurate but initially incorrect predictions, both id and val error are larger than they should be.

For example, it would be a big red flag if our model saw 89% accuracy on the identification data set, but only 60% accuracy on the validation data set.

So the number of samples/Fourier terms directly affects the approximation, making it more or less accurate.

In our case, we have monthly data analysis that helped us determine the trend of the sales in each month, meaning that in order to solve the approximation problem we used P = 12. Every term is connected to a sin or cos depending on the position of the element in the series, the odd positioned elements are connected with the cos function and the even positioned elements are connected with the sin function.

Having all this information, the next step is to compose structure of this function and to apply the linear regression method onto it in order to get the desired parameter vector theta.

The main goal of this project is to find the optimal parameter vector theta so that the approximation mean-squared error is minimum.

3 Key Features

In the structure of the regressor ϕ, we connected every odd positioned element with a cosine Fourier term and every even positioned element with a sine Fourier term, in order to obtain

the matriceal form for the linear regression method.

$$\phi(k) = \left[1, k, \sum_{i=1}^{m} \left[a_i * cos\left(\frac{2 * \pi * i * k}{P}\right) + b_i * sin\left(\frac{2 * \pi * i * k}{P}\right)\right]\right]$$

$$\theta = \phi / y_{id}$$

$$\theta = [t_0, t_1, a_1, b_1, ...a_m, b_m]$$

$$\hat{y}(k) = \phi * \theta$$

$$\hat{y}(k) = \left[1, k, \sum_{i=1}^{m} \left[a_i * cos\left(\frac{2 * \pi * i * k}{P}\right) + b_i * sin\left(\frac{2 * \pi * i * k}{P}\right)\right]\right] * {}^{t}[t_0, t_1, a_1, b_1, ...a_m, b_m]$$

$\hat{y}$ - the approximation of the function

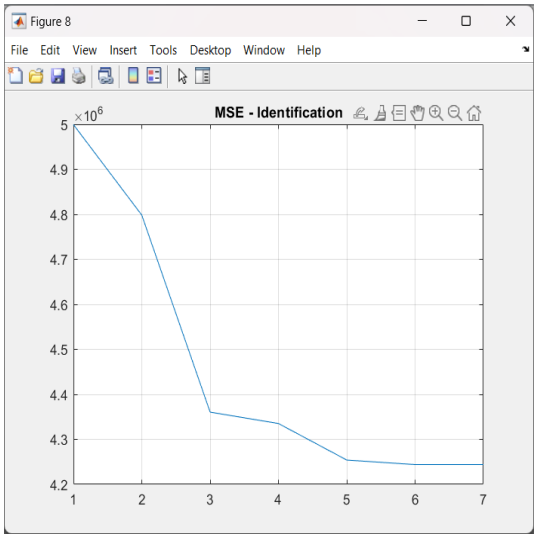
In order to test the accuracy of the approximation we should compute the mean-squared error. For getting the best approximation we have to use the same method both on the identification data set and validation data set after determining the parametric vector theta.

4 Results

To find the best approximation, we needed to compare the results we obtained using a number m, that is defined as the number of samples. If the chosen m is too big, it will lead to overfitting (low identification error, high validation error) or if m is too small, the approximation process will lead to under- fitting (high identification error, high validation error). So, for each m, between 1 and 7, we computed theta and the MSEs (identification and validation).

After computing the errors, we represented them in a graph to find the best m, which corresponds to the lowest value of the validation error vector. In our case, we found out that m = 5, is the best number of samples for our data approximation.

For example, if our model's accuracy regarding the identification data set is 90%, but the validation data set accuracy stands at about 60%, the value of m should be reconsidered.



(a) MSE Plot

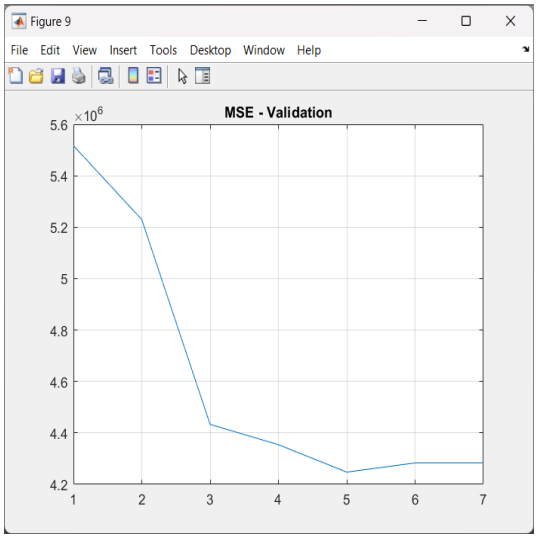
MSEs_id | MSEs_val

7x1 double

	1	2	3
1	4.9993e+06		
2	4.7982e+06		
3	4.3606e+06		
4	4.3352e+06		
5	4.2541e+06		
6	4.2439e+06		
7	4.2439e+06		

(b) MSE Values

Figure 1: Identification data set



(a) MSE Plot

MSEs_id | MSEs_val

7x1 double

	1	2
1	5.5169e+06	
2	5.2293e+06	
3	4.4332e+06	
4	4.3543e+06	
5	4.2476e+06	
6	4.2834e+06	
7	4.2834e+06	

(b) MSE Values

Figure 2: Validation data set

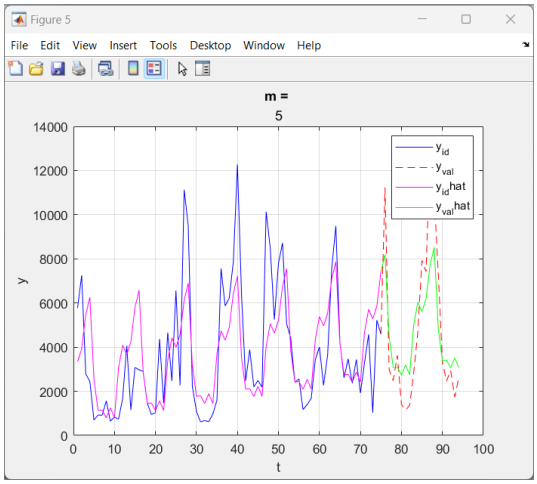


Figure 3: Final plot with the optimal value of m

5 Conclusion

This method has impressive accuracy, we obtained great values for the MSEs both identification and validation meaning that we chose the right value for m and it's proving the fact that the Fourier series approximation is one of the best fit for our problem.

6 Bibliography

- https://groups.csail.mit.edu/netmit/sFFT/soda_paper.pdf
- https://en.wikipedia.org/wiki/Fourier_transform
- https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c

```
clear variables;
syms k;
```

Extracting data & obtaining both identification and validation data set

```
load("product_6.mat");

time_id = time(1:round((80*length(time))/100));
y_id = y(1:round((80*length(y))/100));

time_val = time(round((80*length(time))/100):end);
y_val = y(round((80*length(y))/100):end);

MSEs_id = zeros(7, 1);
MSEs_val = zeros(7, 1);
```

Computing the structure of the function in order to use the linear regression method

```
for m = 1:7

    n = 2*m + 2;
    phi = zeros(1, n);
    for k = 1:length(time_id)
        i = 1;
        for j = 3:n
            if i <= m
                phi(k,1) = 1;
                phi(k,2) = k;
                if (mod(j,2)==0)
                    phi(k,j) = sin(2*pi*i*k/12);
                    i = i + 1;
                end
                if (mod(j,2)~=0)
                    phi(k,j) = cos(2*pi*i*k/12);
                end
            end
        end
    end

    tetha = phi\y_id;

    % Identification - obtaining the approximation of the function:
    for  k = 1:length(time_id)
        ff(k) = tetha(1)*1 + tetha(2).*k;
        for i = 1:m
            ff(k) = ff(k) + tetha(2*i+1)*cos(2*pi*i.*k/12) + tetha(2*i
+2)*sin(2*pi*i.*k/12);
```

```
        end
    end

    % Computing the error for the identification data set:
    mse_id = 1/length(time_id) * sum((y_id - ff').^2);
    MSEs_id(m, 1) = mse_id;

    % Validation - obtaining the approximation of the function:
    for k_v = 75:94
        ff_v(k_v) = tetha(1)*1 + tetha(2).*k_v;
        for i = 1:m
            ff_v(k_v) = ff_v(k_v) + tetha(2*i+1)*cos(2*pi*i.*k_v/12) +
 tetha(2*i+2)*sin(2*pi*i.*k_v/12);
        end
    end
    ff_vf = ff_v(75:94);

    % Computing the error for the validation data set:
    mse_val = 1/length(time_val) * sum((y_val - ff_vf').^2);
    MSEs_val(m, 1) = mse_val;

    % Plot:
    figure,
    plot(time_id, y_id, '-b'); hold; grid;
    plot(time_val, y_val, '--r')
    plot(time_id, ff, '-m')
    plot(time_val, ff_vf,'-g')
    title('m = ', m); xlabel('t'); ylabel('y');
    legend('y_{id}', 'y_{val}', 'y_{id}hat', 'y_{val}hat');
end

figure,
plot(1:7, MSEs_id); grid;
title('MSE - Identification');

figure,
plot(1:7, MSEs_val); grid;
title('MSE - Validation');
```