Penzes Noemi;Popa Răzvan;Șucea Ionuț

Group 30333/2, dataset:15

A short report on the workflow of the assignment, namely describing the theoretical part plus some code snippets and plots.

System identification Project report

Semester 1, 2022-2023

Contents

[**Introduction and problem statement** 2](#_Toc119174916)

[**Methodology and algorithms** 2](#_Toc119174917)

[**MATLAB implementation and plots** 3](#_Toc119174918)

[**Further work and optimizations** 5](#_Toc119174919)

[**Conclusions** 6](#_Toc119174920)

[**Appendix** 7](#_Toc119174921)

# **Introduction and problem statement**

Predicting the future behaviour of time series using supervised learning is the main target of our System Identification project. To fully represent this concept, a set of data that reflects the activity of a store that sells engineering building items has been provided. More precisely, it is known the quantity of products that have been sold each month, throughout the span of 109 months. In our work, we will split the data into two parts: the first 80% will be used for identification and the other 20% for the validation of our model.

# **Methodology and algorithms**

The main idea behind every regression problem is thinking of a model that suits best the regressed variables, that means, thinking of a function that could best fit the full set of data. The situation induced by the given shop sales, however, makes the structure of the regressor a bit more difficult to find, but not impossible, as it is a mix of two base functions. Considering the needs of a person during specific months of the year and the technological trends that may appear, we can develop a suitable model that consists of a linear trend and a Fourier basis. The linear component stands for the ‘trendy’ items that may be purchased for a period, while the Fourier one helps us model the seasonality of products that have higher demand during wintertime, for example. Without further ado, this is how the structure of the chosen regressor looks like:

Where k is the current month, P = 12 because we assumed that the sales are annually, thus the cosine and sine period is 12. Now that this has been settled, another problem arises, which talks about the phenomenon of overfitting that negatively effects the performances on the validation data, even though the error on the training data is small. This occurs when the number of parameters is too high, so we will have to assess which m gives us the best results. We take different values of m since the Fourier basis is the only part that can be altered in this direction; m varies from 1 to 7 as it has been suggested in the assignment.

…

We have created a system of 109 equations (Eq.2) and now we will separate each

equation such that we will have a column vector with 109x1 elements, Φ matrix which will contain all the regressors with 109x(2m+2) elements, θ another column vector with all the scalars of the regressors with 109x1 elements. (Eq.3 & 4)

We will apply the decomposition from Eq.3 to the system in Eq.2 (note we used m = 1) for simplicity:

To compute the θ vector we will multiply on the left side both equations with Φ transposed:

After which we will multiply by the inverse of on the left again:

# **MATLAB implementation and plots**

The MATLAB code has been already commented for an easier understanding of it when reading it, but we will discuss the theory a little bit more in depth here. Firstly, we will look at the dataset and split the data into two parts (80%/20%). (Fig. 1.1 Fig. 1.2)

Graphical user interface, chart

Description automatically generatedGraphical user interface, chart

Description automatically generated

We will apply the algorithm detailed before on the identification dataset, to make an estimation on the validation dataset. For m = 1 we will calculate the regressors matrix and afterwards we compute which is the approximation of the real . Results can be seen below, not exactly accurate for the identification dataset (Fig. 2.2(a)) and on the validation dataset. (Fig. 2.2(b))

Chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

Calculating all the MSEs on the identification and the validation datasets for m=1…7 we get the following results, and we choose the one that has the lowest MSE on the validation dataset. (Fig. 3)

Chart, line chart

Description automatically generated

We denote that m= 5 is a “Best fit”. Calculating the approximation and plotting it for the identification and validation datasets. (Fig. 4.1 and Fig 4.2).

Graphical user interface

Description automatically generated with medium confidenceChart, line chart

Description automatically generated

# **Further work and optimizations**

We denote that the dataset has a positive and a negative trend and we do a piecewise decomposition of the set-in order to subtract the trends. (Fig. 5)

Graphical user interface, chart

Description automatically generated

Approximation of trends on the identification set is depicted by the trend lines and we assumed that the negative trend was consistent furthermore. (Fig. 6.1). Also, the new dataset after the subtraction of the trends can be observed. (Fig. 6.2)

Graphical user interface, chart, line chart

Description automatically generatedGraphical user interface, chart

Description automatically generated

After that we compute the MSE for identification and validation for m = 1…7. (Fig. 7)

Chart, line chart

Description automatically generated

Again, m = 5 is a “Best fit”, because m = 1 does not perform well on the identification dataset. (Fig. 8.1 & 8.2)

A picture containing chart

Description automatically generated Chart, line chart

Description automatically generated

# **Conclusions**

To sum things up, we can confirm that detrending the dataset in a piecewise manner improves the MSE by 2.5 half times because there was a different trend in the data. Further improvements can be brought to the regressor’s form or by increasing the datasets to work with.

# **Appendix**

%% Figure 1.1

load("product\_15.mat"); % Loading the dataset.

plot(y); % Plotting it for some first impressions.

%% Figure 1.2

id\_size = round(4/5\*length(y)); % Splitting the dataset in 80%/20%.

val\_size = length(y) - id\_size;

xline(id\_size,'k--'); % As stated above we divide the dataset

y\_id = y(1:id\_size); % into 2 parts y\_id, y\_val for identification

y\_val = y(id\_size+1:end); % and validation, respectively.

t\_id = time(1:id\_size); % Time vector is split as well.

t\_val = time(id\_size+1:end);

%% Figure 2.1

plot(y\_id); % Plotting y\_id for some first impressions.

%% Figure 2.2(a)(b)

P = 12; % The data is assumed to be periodic with P=12.

for m = 1:1 % The number of Fourier pairs is given by m parameter

phi\_id = zeros(id\_size,2\*m+2);

for i = 1:id\_size % The Phi\_id matrix will have the size of id\_size\*(2\*m+2)

phi\_id(i,1) = 1; % where m is the current number of Fourier pairs.

phi\_id(i,2) = t\_id(i);

coeff = 1;

for j = 3:2\*m+2

if mod(j,2) == 1

phi\_id(i,j) = cos(2\*coeff\*pi\*t\_id(i)/P);

else

phi\_id(i,j) = sin(2\*coeff\*pi\*t\_id(i)/P);

coeff = coeff + 1;

end

end

end

theta = phi\_id\y\_id; % Approximating the theta column vector for m = 1

y\_hat\_id = phi\_id\*theta; % We can see that it presents a positive

MSE\_id(m) = mean((y\_id-y\_hat\_id).^2); % trend along the entire identification

Figure % dataset, not the best MSE either.

plot(y\_id);hold; plot(y\_hat\_id);

phi\_val = zeros(val\_size,2\*m+2);

for i = 1:val\_size

phi\_val(i,1) = 1;

phi\_val(i,2) = t\_val(i);

for j = 3:2\*m+2

if mod(j,2) == 1

phi\_val(i,j) = cos(2\*coeff\*pi\*t\_val(i)/P);

else

phi\_val(i,j) = sin(2\*coeff\*pi\*t\_val(i)/P);

coeff = coeff + 1;

end

end

end

y\_hat\_val = phi\_val\*theta;

MSE\_val(m) = mean((y\_val-y\_hat\_val).^2);

figure

plot(y\_val);hold;plot(y\_hat\_val);

end

%% Figure 3

figure

plot(MSE\_id);hold;plot(MSE\_val);

%% Figures 4.1 & 4.2 % We look for m such that the MSE on the validation

m\_bestfit = find(MSE\_val == min(MSE\_val));

%% Figure 5

y\_id = y(1:id\_size);

figure

plot(y,'k');hold on % Further improvements brought to the regressors matrix

max\_ind = find(y == max(y))+1; % We find the maximum of the graph and do a pw decomp

y\_id1 = y(1:max\_ind); % Beginning->Maximum (red) has a positive trend

y\_id2 = y(max\_ind:id\_size); % Maximum->End of identification data (blue) has a

plot(time(1:max\_ind),y\_id1,'r',time(max\_ind:id\_size),y\_id2,'b'); % negative trend

xline(max\_ind-1,'--k');

%% Figure 6.1 6.2

figure

plot(y\_id,'k');

phi\_id1 = zeros(max\_ind,2); % We will approximate trends first.

for i = 1:max\_ind % We use 2 linear regressors for this.

phi\_id1(i,1) = 1;

phi\_id1(i,2) = t\_id(i);

end

theta\_1 = phi\_id1\y\_id1;

y\_hat\_id1 = phi\_id1\*theta\_1;

phi\_id2 = zeros(id\_size-max\_ind+1,2);

for i = 1:id\_size-max\_ind+1

phi\_id2(i,1) = 1;

phi\_id2(i,2) = t\_id(i+max\_ind-1);

end

theta\_2 = phi\_id2\y\_id2;

y\_hat\_id2 = phi\_id2\*theta\_2;

diff = y\_hat\_id1(end)-y\_hat\_id2(1);

for i = 1:length(y\_hat\_id2)

y\_hat\_id2(i) = y\_hat\_id2(i) + diff;

end

for i = 1:max\_ind % We can subtract the values of each regressor from the data, thus

y\_id(i) = y\_id(i) - y\_hat\_id1(i); % removing the positive and negative trends.

end

for i = max\_ind+1:id\_size

y\_id(i) = y\_id(i) - y\_hat\_id2(i-max\_ind);

end

figure

plot(y\_id); % The dataset looks more like some random noise now.

%% Figure 7

for m = 1:7 % After detrending the data set we can look for the periodicity

theta\_4 = zeros(2\*m+2,1); % of the values with the Fourier approximators.

phi\_id3 = zeros(id\_size,2\*m);

for i = 1:id\_size

coeff = 1;

for j = 1:2\*m

if mod(j,2) == 1

phi\_id3(i,j) = cos(2\*coeff\*pi\*t\_id(i)/P);

else

phi\_id3(i,j) = sin(2\*coeff\*pi\*t\_id(i)/P);

coeff = coeff + 1;

end

end

end

theta\_3 = phi\_id3\y\_id;

y\_hat\_id3 = phi\_id3\*theta\_3;

MSE\_id(m) = mean((y\_id-y\_hat\_id3).^2);

phi\_val = zeros(val\_size,2\*m+2);

for i = 1:val\_size

coeff = 1;

phi\_val(i,1) = 1;

phi\_val(i,2) = t\_val(i);

for j = 3:2\*m+2

if mod(j,2) == 1

phi\_val(i,j) = cos(2\*coeff\*pi\*t\_val(i)/P);

else

phi\_val(i,j) = sin(2\*coeff\*pi\*t\_val(i)/P);

coeff = coeff + 1;

end

end

end

theta\_4(1) = theta\_2(1);

theta\_4(2) = theta\_2(2);

for i = 1:2\*m

theta\_4(i+2) = theta\_3(i);

end

y\_hat\_val = phi\_val\*theta\_4;

MSE\_val(m) = mean((y\_val-y\_hat\_val).^2);

end

figure

%% Figure 8

m = 5; % We decided to choose m = 5 as the next candidate for the

theta\_4 = zeros(2\*m+2,1); % best approximation and we can see it fits very well on

phi\_id3 = zeros(id\_size,2\*m); % the dataset.

for i = 1:id\_size

coeff = 1;

for j = 1:2\*m

if mod(j,2) == 1

phi\_id3(i,j) = cos(2\*coeff\*pi\*t\_id(i)/P);

else

phi\_id3(i,j) = sin(2\*coeff\*pi\*t\_id(i)/P);

coeff = coeff + 1;

end

end

end

theta\_3 = phi\_id3\y\_id;

y\_hat\_id3 = phi\_id3\*theta\_3;

MSE\_id(m) = mean((y\_id-y\_hat\_id3).^2);

phi\_val = zeros(val\_size,2\*m+2);

for i = 1:val\_size

coeff = 1;

phi\_val(i,1) = 1;

phi\_val(i,2) = t\_val(i);

for j = 3:2\*m+2

if mod(j,2) == 1

phi\_val(i,j) = cos(2\*coeff\*pi\*t\_val(i)/P);

else

phi\_val(i,j) = sin(2\*coeff\*pi\*t\_val(i)/P);

coeff = coeff + 1;

end

end

end

theta\_4(1) = theta\_2(1);

theta\_4(2) = theta\_2(2);

for i = 1:2\*m

theta\_4(i+2) = theta\_3(i);

end

y\_hat\_val = phi\_val\*theta\_4;

MSE\_val(m) = mean((y\_val-y\_hat\_val).^2);

figure

plot(y\_id);hold;plot(y\_hat\_id3);

figure

plot(y\_val);hold;plot(y\_hat\_val);