Eduardo Costa de Oliveira
PPGMMC
eduardo.oliveira@sou.unijui.edu.br

Atividade Final
Computação Científica

30 de outubro de 2021

1 Proposição

Aplicar os conhecimentos adquiridos ao longo das aulas em um problema de modelagem matemática e computacional na sua área de pesquisa;
Utilize o MATLAB/Simulink com ao menos 3 das seguintes funcionalidades:

- Funções próprias
- Plots de gráficos
- Funções de Import/Export dados
- Import Data Tool
- Operações matriciais ou laços for Simulink

Contexto:

- Use uma de suas referências bibliográficas como base para este trabalho;
- Explique brevemente o objetivo/método deste trabalho/artigo referência;
- Reproduza os modelos apresentados;
- Compare os seus resultados com os apresentados pelo trabalho/artigo referência;
- Se o modelo for muito complexo ou extenso, é permitido reproduzi-lo parcialmente;

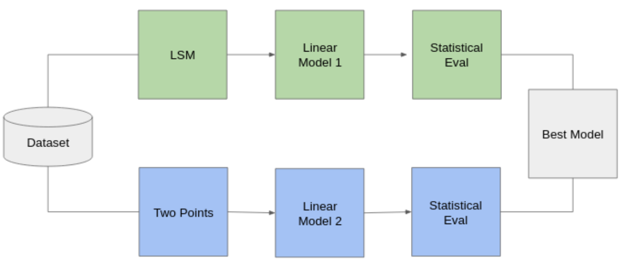
1.1 Resolução

1.2 Introduction

To achieve a better understanding of various phenomena it is important to establish the relationship between the variables. In many cases this relation is linear or can be converted to linear. This current script and related functions in Octave (Matlab compatible) implements:

- Reads a dataset from a csv file
- Evalauates the dataset with two evaluation methods
- Generates the respective linear functions (models)
- Statistically evaluates the performance of each model using RMSE, MAPE and R Squared
- Plots the data, models and residuals, also known as forecast error.

Figura 1: Model flowchart.



Source: The author.

The script and related functions are available for reproducibility at `https://github.com/OliveiraEdu/scientific_computing`.

1.3 Descritption

Modeling (Octave and Matlab compatible).

Given a dataset this application evaluates the data and generates two linear models:

- Model 1 - Evaluation applies the Least Square Method and generates the linear model [3].
- Model 2 - Evaluation takes two data points x,f(x) and generates the linear model [2].

1.4 Script

- *modeling.m*

1.5 Functions

- *linear_LSM.m* - Least Square Method evaluation and modeling
- *linear_two_points.m* - Two poins evaluation and modeling
- *plotting_data_models.m* - Scatter plot the data and plot the models on the same figure
- *plotting_residuals.m* - Plots the residuals (forecast errors)
- *read_prepare_data.m* - Reads the data and prepares for evaluations
- *statistical_eval.m* - Evaluates statistical metrics for both models (Mean Absolute Percentage Error, Root Mean Square Error and R-Squared)

1.6 Main Variables

- *data* - Stores the dataset, first column hold the values for the independent variable, second column the values for the dependent variable.
- *yHat_modeln* - Stores the predicted values evaluated from the model n.
- *betaHat_modeln* - Stores the values for the angular coefficient and intercept for the linear function for the model n.

1.7 Requirements

- Dataset must be a csv UTF-8 formatted file, no headers.
- Single dependent variable.
- Independent variable on column one.
- Dependent variable on column two.
- This a 2D evaluation.

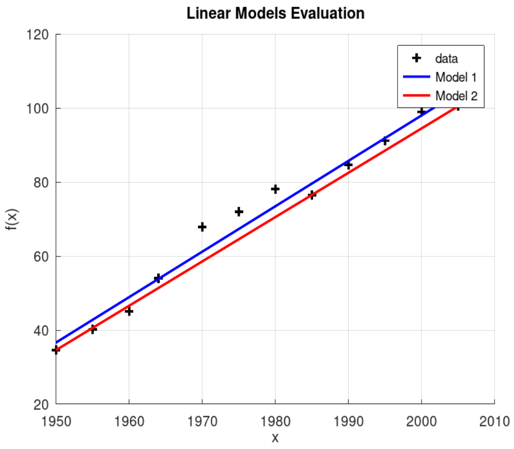
1.8 Example

Data set [1]

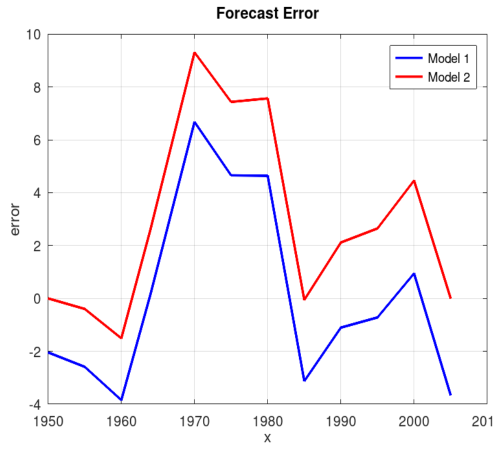
```


 data =

   1950.000     34.616
   1955.000     40.208
   1960.000     45.087
   1964.000     54.017
   1970.000     67.884
   1975.000     71.999
   1980.000     78.122
   1985.000     76.491
   1990.000     84.652
   1995.000     91.173
   2000.000     98.975
   2005.000    100.506

----------------- Least Squares Model -----------------
yHat_model1 =

    36.659
    42.796
    48.933
    53.843
    61.207
    67.345
    73.482
    79.619
    85.756
    91.893
    98.030
   104.167

betaHat_model1 =

  -2.3568e+03
   1.2274e+00

----------------- Two Points Model -----------------
yHat_model2 =

    34.616
    40.606
    46.596
    51.388
    58.576
    64.566
    70.556
    76.546
    82.536
    88.526
    94.516
   100.506

betaHat_model2 =

   1.1980e+00  -2.3015e+03

----------------- Statistical Evaluation of the Models -----------------
MAPE_model1 = 0.045177
MAPE_model2 = 0.044087
RMSE_model1 = 3.4069
RMSE_model2 = 4.4707
rsq_model1 = 0.9750
```

```
rsq_model2 = 0.9569
```

Listing 1: Output of the code.



(a) data and models



(b) forecast error

1.9 Code

```
close all;clear all;clc;

% Sample dataset, pick one and uncomment.

%data = csvread('ozone.csv')
data = csvread('energy_consumption.csv')
%data = csvread('vehicular_stopping.csv')

[t,x,y] = read_prepare_data(data);

[yHat_model1, betaHat_model1] = linear_LSM(x,y)

[yHat_model2, betaHat_model2] = linear_two_points(data)

[resid_model1,  resid_model2] = statistical_eval(y, yHat_model1, yHat_model2,
     betaHat_model1, betaHat_model2);

plotting_data_models(data,x,y, yHat_model1, yHat_model2)

plotting_residuals(x,resid_model1, resid_model2)
```

Listing 2: modeling.m.

```

function [t,x,y] = read_prepare_data( data);

  t = ones(1,length(data)).'; %generates a single column matrix with ones
  x = [t data(:,1)]; %generates a two columns matrix with ones and the
      independent variable
  y = data(:,2); %generates a single column matrix with the dependent variable

  endfunction
```

Listing 3: read_prepare_data.m.

```



%Least Squares Method approximation
%Based on Rawlings Chapter 3

function [yHat_model1, betaHat_model1] = linear_LSM(x,y);
```

```
  fprintf('----------------- Least Squares Model -----------------\n')
  step1 = x.'*x;

  %Sums of products between each independente variabile in turn and the
     dependent variable
  step2 = x.'*y;
  step3 = (x.'*x)^-1;
  step3 = step1^-1;

  %Normal Equation

  betaHat_model1 = step3*step2;

  %Residual

  yHat_model1 = x*betaHat_model1;

  p = (x*step3*x');

  yHat_model1 = p*y;

  %Symmetric matrix

  simmMatrix = p';

  %Idempotent matrix
  idempMatrix = p*p;
endfunction
```

Listing 4: linear_LSM.m.

```

%Two points linear aproximation
%based on Heinz, Chapter 1

function [yHat_model2, betaHat_model2] = linear_two_points(data);
  fprintf('----------------- Two Points Model -----------------\n')

  %Picks the second and the last points of x, f(x)
  l = length(data);
  o = l-1;
  p = l-o;

  %Evaluates the angular coefficient
  m = (data(l,2)-data(p,2))/(data(l,1)-data(p,1));

  %Evaluates the intercept
  n =data(l,1)*-m+data(l,2);

  betaHat_model2 = [m n];

  yHat_model2 = m*data(:,1)+n;

  %Evaluates residuals of model 2 (Heinz 1.4)
  %e1 = (data(:,2)- yHat_model2)/yHat_model2

  %e1 = data(:,2)- yHat_model2

endfunction
```

Listing 5: linear_two_points.m.

```


%Evaluates the models in relation to MAPE, MSE and RSQ

function [resid_model1,   resid_model2] = statistical_eval(y, yHat_model1,
    yHat_model2, betaHat_model1, betaHat_model2);
```

```
  fprintf('----------------- Statistical Evaluation of the Models
      ----------------\n')

  % To see how good the fit is, evaluate the polynomial at the data points and
      generate a table showing the data, fit, and error.
  % Also known as Forecast Error
  resid_model1 = y-yHat_model1;
  resid_model2 = y-yHat_model2;

  % Square the residuals and total them to obtain the residual sum of squares:
  SSresid_model1 = sum(resid_model1.^2);
  SSresid_model2 = sum(resid_model2.^2);

  %MAPE
  pre_MAPE_model1 = abs((yHat_model1-y)./y);
  MAPE_model1 = mean(pre_MAPE_model1(isfinite(pre_MAPE_model1)))

  pre_MAPE_model2 = abs((yHat_model2-y)./y);
  MAPE_model2 = mean(pre_MAPE_model2(isfinite(pre_MAPE_model2)))

  % Squared Error
  sqr_error_model1 = resid_model1.^2;
  sqr_error_model2 = resid_model2.^2;

  % Mean Squared Error
  MSE_model1 = mean(sqr_error_model1);
  MSE_model2 = mean(sqr_error_model2);

  % RMSE - Root Mean Squared Error
  RMSE_model1 = sqrt(MSE_model1)
  RMSE_model2 = sqrt(MSE_model2)

  % Compute the total sum of squares of y by multiplying the variance of y by
      the number of observations minus 1:
  SStotal_model1 = (length(y)-1) * var(y);
  SStotal_model2 = (length(y)-1) * var(y);

  % Compute R2 using the formula given in the introduction of this topic:
  % For linear regression only
  rsq_model1 = 1 - SSresid_model1/SStotal_model1
  rsq_model2 = 1 - SSresid_model2/SStotal_model2

  % Computing Adjusted R2 for Polynomial Regressions
  % Usually the adjusted R2 is smaller than simple R2. It provides a more
      reliable estimate of the power of your polynomial model to predict.
  %rsq_adj_model1 = 1 - rsq_model1 * (length(y)-1)/(length(y)-length(
      betaHat_model1))
  %rsq_adj_model2 = 1 - rsq_model2 * (length(y)-1)/(length(y)-length(
      betaHat_model2))
endfunction
```

Listing 6: statistical_eval.m.

```
%Plots the data, and predicted values of the models

function  plotting_data_models(data,x,y,yHat_model1,yHat_model2)

 scatter (data(:,1),data(:,2),'k','+','linewidth',2)
 hold
 plot(data(:,1), yHat_model1,'b','linewidth',2)
 plot(data(:,1), yHat_model2,'r','linewidth',2)
 legend('data', 'Model 1', 'Model 2')
 title ('Linear Models Evaluation')
 xlabel('x')
 ylabel('f(x)')
 grid

```

```
endfunction
```

Listing 7: plotting_data_models.m.

```

%Comparative plot of the two models residuals, aka forecast errors

function  plotting\_residuals(x,resid_model1, resid_model2)
 figure

 plot(x(:,2), resid_model1,'b','linewidth',2)

 hold

 plot(x(:,2), resid_model2,'r','linewidth',2)

 legend('Model 1','Model 2')
 title ('Forecast Error')
 xlabel('x')
 ylabel('error')

 grid

endfunction
```

Listing 8: plotting_residuals.m.

Referências

[1] Data set: Energy consumption, expenditures, and emissions indicators estimates, 2021.

[2] Stefan Heinz. *Matematical Modeling*. Springer, 1st edition, 2011.

[3] John O. Rawlings, Sastry G. Pantula, and David A. Dickey. *Applied Regression Analysis: A Research Tool,Second Edition*. Springer Science+Business Media, 2nd edition, 2001.