Data-Model Coupling

Emmanuel Frénod[1]
Hélène Flourent[1 2]

Simulations and predictions of complex phenomena (weather forecasts, floods, disease evolutions...) need the construction of a model which is able to manage and adapt to this complexity and extract complex information from noisy data. The objective of this lecture is to introduce a way to meet those needs. Indeed, the purpose of this lecture is to lead you to discover step by step through practical exercises the notion of model-data coupling and its use to perform Statistical Learning.

The assessment of this Teaching Unit will be only based on the evaluation of the handwritten answers to the exercises. The required figures must be presented, numbered and annotated in the appendix. All the figures must be cited in the answers. The handwritten answers have to be scanned, added to the appendix and e-mailed (emmanuel.frenod@univ-ubs.fr and helene.flourent@univ-ubs.fr) or deposited on the Moodle platform. The annotated code associated with the exercises (R or Python) must also be e-mailed or deposited. The title of the documents has to contain your name (Ex : *Answers_SecondName.pdf* or *Code_SecondName.R*).

1 Introduction

Exercise 1.1 Data-Model Coupling

1. Explain the global principle of Data-Model Coupling ;

2. Describe the interests and the limits of this approach ;

[1] Université Bretagne Sud, Laboratoire de Mathématique Atlantique, UMR CNRS 6205, Campus de Tohannic, Vannes, France

[2] Neovia, Route de Talhouët, Saint-Nolff, France

2 Construction and analyze of a simple mathematical model

Exercise 2.1 Model construction

1. Code for some values of the parameters N, a, b and c, the following model :

$$U_{n+1} = a \cdot U_n^c + b, \tag{2.1}$$

   where $n \in [\![0; N]\!]$ and $U(0) = U_0$.

2. Define the format you will use to store the time series that will come out your functions ;

3. Construct a function *Generate_Output* which takes as inputs U_0, N, a, b and c, and compute the time series (U_n) ;

4. Let $N = 40$, $a \in [2; 5]$, $b \in [1; 5]$ and , $c \in [0.1; 0.9]$. For different values of a, b and c, generate 6 curves. Present the results by specifying the values of a, b and c.

5. Describe the influence of each parameter on the curve profile. What kind of phenomena can be modeled thanks to this mathematical model ?

Exercise 2.2 Behavior of the parameters of the model

1. Describe the notion of "sensitivity of parameters" ;

2. By varying the values of the parameters a, b and c, describe the sensitivity of each of them. Briefly describe your approach ;

3. Construct a parameter sensitivity indicator and explain it ;

4. Compute the value of your indicator associated to each parameter ;

5. Conclude about the problems which may be encountered due to a high sensitivity of parameters.

Exercise 2.3 Distribution of the parameters

1. List and describe the inputs and the outputs of the model ;

2. What can be the law of probability of the parameters a, b and c ? Justify your answer ;

3. What can be the law of probability of the inputs ? Justify your answer ;

4. Set the parameters of those probability laws at values that look suitable to you. Specify the chosen values, justify them and present the obtained histograms ;

3 Simulation tests

Exercise 3.1 The notion of noisy data

1. Describe the notion of noise in the context of data collection ;

2. What problems may be encountered due to the presence of noise ?

3. What can be the law of probability of this random component ?

Exercise 3.2 Generation of a learning database

The goal of this exercise is to build a synthetic Learning Database containing M different time series. This database will serve as the support in the following exercises.

1. Verify that the parameters of the probability laws of a, b and c previously set, ensure the relative stability of the outputs. If it is not the case, modify those probability laws in order to ensure it. Fix N at 40 and M at 100 ;

2. Generate M values of the parameters a, b and c, and M values of U_0 ;

3. From those values, generate M curves and build a $M \times N$ database. Name it *Output_Curves* ;

4. Generate a matrix $M \times N$ of random noise components ;

5. Add those matrices to the generated database *Output_Curves*. Name it *Main_Learning_Base* ;

6. Build a SQL database, while ensuring to keep all the information used to generate each curve ;

7. Divide the database *Main_Learning_Base* into two datasets : a Training Dataset (*Main_Training_Base*), made of $0.7M$ curves and a Test Dataset (*Main_Test_Base*), made of $0.3M$ curves ;

Exercise 3.3 Learning of the parameters

We suppose now that we have a database, containing noisy data and a model, containing three unkonw parameters, a, b and c. This exercise consists in fitting the values of the parameters, a, b and c.

1. Describe the interest of this approach ;

2. Build an indicator measuring the relative difference between two curves ;

3. By using the previously built indicator, construct a function *RelDiff* which takes as input a triplet of values (a_t, b_t, c_t), T, and computes the relative difference between a curve of the Learning Database and a curve generated by the model from the same inputs and by using this triplet as the values of the parameters. Integrate Function *Generate_Output* in this function.

4. Let *All_U0* a vector containing a list of values of U_0. Describe what the following function does :

```
Predicted_Curves <- mapply(Generate_Output,
                             All_U0, N, a_t, b_t, c_t)
```

5. By using this function and on the basis of the function *RelDiff*, construct a function *f_obj*. This function *f_obj* has to take as inputs a triplet of values (a_t, b_t, c_t), T, and the list of values of U_0 associated to the curves of the Training Database. Then from those inputs, this function computes the average relative difference between the curves of the training database and the curves generated from the same inputs and by using the triplet (a_t, b_t, c_t) as the values of the model parameters ;

6. Explain what can be the purpose of this function and what is the origin of its name ;

7. Explain briefly what is the purpose of the algorithm DIRECT and its functioning ;

8. By using the function *f_obj*, use the algorithm DIRECT to adjust the parameters a, b and c from the training database. Explain the learning approach and justify the configuration of the algorithm ;

Exercise 3.4 Stability of the parameters

1. Sample randomly with replacement 50 curves among the 70 curves of the Training Dataset. Built a table containing these randomly selected curves and the associated information ;

2. By using the function *f_obj* and the algorithm DIRECT, adjust the parameters a, b and c from those new training database. What are the obtained values of the parameters a, b and c ?

3. Repeat this operation (Question 1 then 2) 10 times and build a table containing the values of the 10 obtained triplets (a_t, b_t, c_t) ;

4. Calculate the average, the relative variance and the relative standard deviation associated to each parameter ;

5. Describe the results and the interest of this approach ;

Exercise 3.5 Accuracy of the adjuted model

We fix the parameters a, b and c at their previously calculated average values.

1. Choose one or several indicators quantifying the goodness of fit of the adjusted model. Justify your choice ;

2. By using the chosen indicators, compute the accuracy of the model on the Training Dataset ;

3. As previously, compute the accuracy of the model on the Test Dataset ;

4. Describe the results ;

Exercise 3.6 Comparison of different algorithms

1. Find an optimization algorithm based on a gradient descent method. Briefly explain the functioning of this algorithm ;

2. Use this algorithm to adjust the parameters a, b and c from the training dataset by initializing the three parameters at 0. What are the obtained values of the parameters ?

3. Launch the adjustment of the parameters by testing different initializations of the parameters a, b and c (5 tests). Present the results and comment them ;

4. Calculate the average, the relative variance and the relative standard deviation associated to each parameter ;

5. Fix the parameters a, b and c at their average value previously calculated. By using the chosen indicators, compute the accuracy of the model on the training dataset and on the test dataset ;

6. Compare the algorithm DIRECT and the algorithm based on a gradient descent method. Which algorithm seems most suitable ? In which cases and why ?

Exercise 3.7 Conclusion about the approach

1. Sum up the global approach (From Exercise 2.1 to Exercise 3.5) ;

2. Compare this teaching approach with a classical approach ;

4 Machine Learning approach

4.1 Principle of the approach

Exercise 4.1 Machine Learning principle

1. Explain the global principle of Machine Learning ;

2. Describe the interests and the limits of this approach in comparison with Data-Model Coupling ;

3. Briefly explain the functioning of a Neural Network algorithm ;

4. Describe what can be the influence of the number of neurons and the number of layers on the results given by this algorithm ;

4.2 Use of a Neural Network algorithm

Exercise 4.2 Curve fitting using neural networks

1. Describe the content of the following table :

N° Curve	n	U_0	TS
1	1	$U_{0,1}$	$TS_{1,1}$
1	2	$U_{0,1}$	$TS_{2,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	N	$U_{0,1}$	$TS_{N,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	1	$U_{0,i}$	$TS_{1,i}$
i	2	$U_{0,i}$	$TS_{2,i}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	j	$U_{0,i}$	$TS_{j,i}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	N	$U_{0,i}$	$TS_{N,i}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
M	1	$U_{0,M}$	$TS_{1,M}$
M	2	$U_{0,M}$	$TS_{2,M}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
M	N	$U_{0,M}$	$TS_{N,M}$

where $U_{0,i}$ is the initial value used to generate the i^{th} curve. $TS_{j,i}$ corresponds to the value of the i^{th} curve at $n = j$.

Build this table and name it *Learn_Base.* ;

2. Build a table *Train_Base_1* containing the information relative to the $0.3M$ first curves , that is the $0.3M \times N$ first rows of the table *Learn_Base*. Build a table *Test_Base_1* containing the information relative to the $0.7.M$ last curves, that is the $0.7M \times N$ last rows of the table *Learn_Base* ;

3. Explain what the following code lines do :

   ```
   S <- function(TrainB, TestB){
     TrainB_S <- matrix(0,N*3,4)
     TestB_S <- matrix(0,N,4)
     for(i in 2:4){
       TrainB_S[,i] <-(TrainB[,i]-mean(TrainB[,i]))/
                                         diff(range(TrainB[,i]))
       TestB_S[,i] <-(TestB[,i]-mean(TrainB[,i]))/
                                         diff(range(TrainB[,i]))
     }
     return(list(TrainB_S, TestB_S))
   }

   Learn_Base_S <- S(Train_Base,Test_Base)
   Train_Base_S <- Learn_Base_S[[1]]
   Test_Base_S <- Learn_Base_S[[2]]

   colnames(Train_Base_S) <- c("n_curve","n", "U0", "TS")
   colnames(Test_Base_S) <- c("n_curve","n", "U0", "TS")
   ```

   Code those code lines ;

4. Find an existing function training Neural Networks. Use this function to fit the Neural Network linking the inputs (n and U_0) and the output (TS) on the dataset *Train_Base_1_S*. The used Neural Network has to be formed in 3 hidden layers made of 3, 6 and 3 nodes ;

5. Show the Neural Network diagram. How many parameters are fitted ? Compare this number with the number of parameters fitted in the Model-Data coupling approach (Exercise 3.3) ;

Exercise 4.3 Study of the fitted model

1. Generate for some curves of the dataset *Train_Base_1_S* a figure showing in the same graph the initial curve and the associated predicted curve. Do the same for some curves of the dataset *Test_Base_1_S* ;

2. Calculate the value of the accuracy indicators chosen in the exercise 3.5 on the datasets *Train_Base_1_S* and *Test_Base_1_S*. Describe the results ;

3. Calculate the value of the accuracy indicator chosen in the exercise 3.5 on the datasets *Train_Base_2_S* and *Test_Base_2_S*. Describe the results. Compare the results obtained with the Machine Learning approach and the those obtained with Model-Data Coupling approach. Explain and justify the differences ;

4.3 Conclusion about the two explored approaches

Exercise 4.4 Conclusion

1. Calculate the computation time of the adjustments performed with the two approaches ;

2. Conclude about the Model-Data Coupling approach and the Machine Learning approach : Differences, strength and limits ;