FUZZY INDUCTIVE REASONING LAB CLASS GUIDE

Software preparation

Objective: Initialization of Matlab path in order to be able to run the VISUAL-FIR software

Requirements: Windows (8, 7) / Unix and Matlab (2016b, 2016a, 2015b, 2015a, 2014b, 2014a, 2013a, 2012b). You need to have installed a 64 bits Matlab version. It does not work in a 32 bits version.

Steps to start using FIR:

1) Open Matlab
2) Include VisualFIR folders: *Saps* and *VisualFIR*, in the Matlab path with all its subfolders. Do it by selecting the *Set Path* button in the *Home* tab.
3) VisualFIR generates intermediate files in each process, so it is recommended to use a different folder for each data set that you work with.

Ways of running FIR:

1) Using the GUI-platform VisualFIR
2) Using a GUI that allows to perform all the FIR processes in an automatic way
3) Writing a script (not using any graphical interface)

Task 1 – Energy performance of residential buildings – GUI-platform VisualFIR

Objective: In this task, we will learn how to use FIR methodology by means of the GUI-platform Visual-FIR. We will use the data of the paper of Tsanas and Xifara, that is included in the folder.

In this paper the authors study the effect of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area and glazing area distribution) on two output variables, namely heating load and cooling load, of residential buildings. They investigate the association strength of each input variable with each of the output variables using a variety of classical and non-parametric statistical analysis tools, in order to identify the most strongly related input variables. Then, they compare a classical linear regression approach against a powerful state of the art nonlinear non-parametric method, random forests, to estimate heating load and cooling load.

It is important to take a first look to the problem and see the variables involved in this study, their meaning and the number of possible values that can take (see Table 1). You can take a look to the paper if you want to understand better the problem.

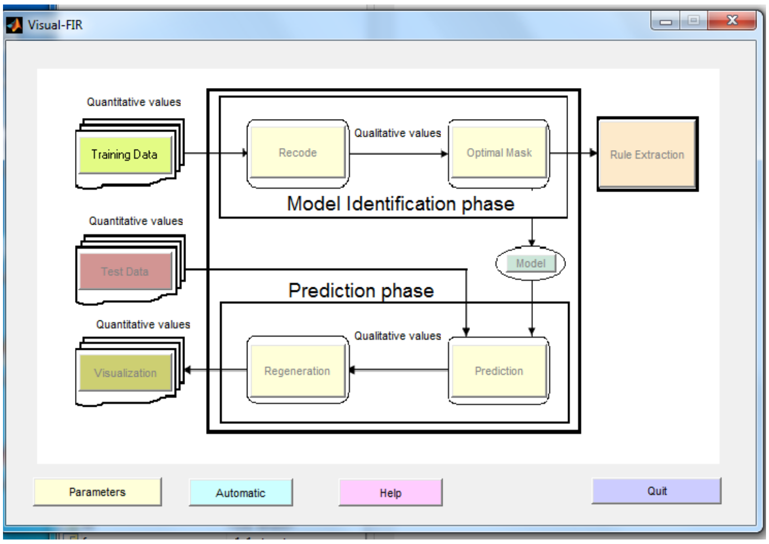
TABLE 1

Mathematical representation of the input and output variables to facilitate the presentation of the subsequent analysis and results.

Mathematical representation	Input or output variable	Number of possible values
X1	Relative Compactness	12
X2	Surface Area	12
X3	Wall Area	7
X4	Roof Area	4
X5	Overall Height	2
X6	Orientation	4
X7	Glazing area	4
X8	Glazing area distribution	6
y1	Heating Load	586
y2	Cooling Load	636

1) First, you should place in the folder where you have the data, in this case “EnergyBuildings/Data”. You can use the browser button that you have in the left hand side of the Matlab screen. Now you have the training and test data files in the left hand side (current folder).

2) Open the GUI-platform Visual-FIR by writing in the Matlab command window: *VisualFIR*. You get the following screen.



3) Load the data by pushing the *Training Data* button. Select the file: Y1Train_CV1.mat

As explained in detail in the paper, each one of the 768 simulated buildings can be characterized by eight building parameters which are: Relative Compactness (X1), Surface Area (X2), Wall Area (X3), Roof Area (X4), Overall Height (X5), Orientation (X6), Glazing Area (X7) and Glazing Area Distribution (X8). These parameters correspond to the input variables. In addition, they recorded the Heating Load (Y1) and the Cooling Load (Y2), which correspond to the output variables.

In this lab session, we will work only with the output variable Y1 and we will use only the first fold of the 10-fold cross validation mentioned in the article.

Select the input and the output variables. Notice that you can plot each variable if you like to analyse the kind of data that you are dealing with. Save the file and quit the screen.

4) Go further to the discretization step by pushing the *Recode* button. You should press the *Read File* button in order to capture the variables that have been saved in the previous step.

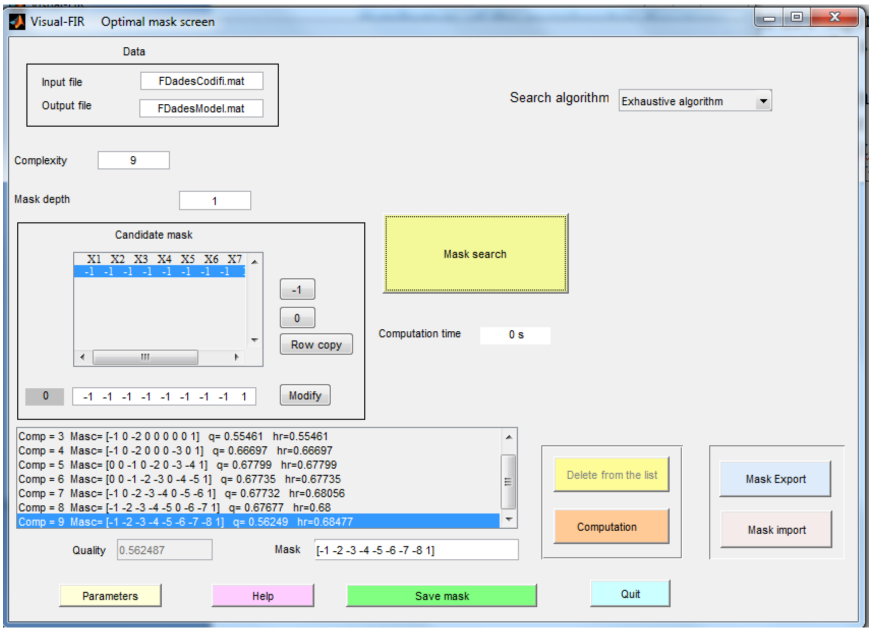
Discretize the input variables X1, X2, X3 and X8 into three classes and X4, X5, X6 and X7 into only two classes. The output variable Y1 is discretized into three classes. Use the equal with interval algorithm for the discretization of all the variables. Chose the algorithm and the number of classes in the right hand side of the screen and perform *Save variable*.

The equal with interval algorithm is the one that performs a uniform distribution of the membership functions. The equal frequency partition algorithm distributes the membership functions of a variable in such a way that all the classes contain the same number of data points.

Use *Output generation* to see the landmarks obtained and, then, use *recode* to perform the fuzzyfication and save the results in an intermediate file. Notice that you can choose different discretization algorithms. Moreover, it is also possible to define by hand the landmarks of a variable using the *Manual* button. Use *Datagram* and *Global Memb.* to see the result of the discretization chosen.

You also can save the discretization performed in order to use them with the rest of the folders (if you work with several folders), pushing the button *Landmarks save*. In this way, you can load it when you want, for example when working with the rest of the data folders.

5) Perform the modelling by means of the *Optimal mask* screen.



You should select the complexity (from 2 to 9), the mask depth (no limit), the search algorithm (exhaustive search or genetic algorithm), and push *Mask search*. Then, you should select the mask that you want to use from the list of optimal masks of each complexity obtained by FIR. You can use the optimal one (maximum quality *q*) or a suboptimal. You can also introduce a mask manually in the *Mask* window and then push the *Computation* button. Finally, you should save the mask by pushing *Save mask* button.

In this example, you are going to use a complexity of 9, in order that FIR searches the best masks for each complexity being the mask with higher complexity the one that contains all the input variables of the application at hand. The mask depth for the current data should be 1. In this way, the mask has only one row, studying only the causal relation between the different inputs and the output in the same temporal moment. The depth should be one because in this application there is no temporal relation between different samplings of the data. Each data point corresponds to a building, and therefore, it does not make sense to study causal relationships between variables of different buildings.

6) Load the test data by means of the *Test Data* screen. Choose the file: Y1Test_CV1.mat

7) The next step is to perform the prediction using the previous selected mask. To do so use the *Prediction* screen. You can open the set of *Parameters* involved in the modelling process. In this stage, the parameter that can be interesting to play with is *nVeins* that is the number of nearest neighbours. You should press the prediction button; now you need to regenerate to see the results of the prediction.

8) Regenerate using the *Regeneration* screen. Here you can see the predicted signal regenerated. The last step is to compare the predicted signal with the test real signal.

9) This is done in the *Visualization* screen. Pushing *Real output visualization* and *Prediction output visualization* buttons, both signals are shown. You can also see the errors by choosing an error measure and pushing *Visualization of the differences*. It is possible to save the results to get the predicted and real signals in order to perform your own plots or compute your own error measures.

Task 2 – Play with the parameters

Objective: Use different parameters to see the changes in the behaviour of the system's model and in the results.

1) Course of dimensionality problem: If we keep the same number of data instances and we increase the number of classes to represent each variable (those that make sense to be discretized in more than 2 classes), and choose a mask of higher complexity.

Recode all the variables that were previously discretized into 3 classes, into 5 classes. Choose the mask of highest complexity in the *Optimal mask* process. What happens when you perform the prediction? Why?

2) Study, in this application, if the selection of the discretization algorithms affects in a substantial way the prediction of the FIR models. What happens if all input variables are discretized using the equal frequency partition method instead of the equal with interval? Why? Take a look to the *Datagram* (Recode screen) for the different variables, for example X3 and X7.

3) How the suboptimal masks behave? Which variables do you think that represent better the input/output causality of the system?

4) Close the VisualFIR GUI and go to the folder Data-Autom.

Task 3 – Energy perf. of residential buildings – Automatic running of FIR

Objective: In this task, we will learn how to run FIR methodology by means of the Automatic GUI. This option is useful when you need to perform a k-fold cross validation. In that case, if you like, you can study more accurately which parameters get better results in one folder (using VisualFIR), and then apply them in the rest of the folders using the automatic running of FIR. Here you will use the same energy data than in the previous tasks.

1) To use the automatic GUI you need first to generate the AutomaticDiscretization.m and the AutomaticConfiguration.m files for each fold, where the discretization and the FIR parameters are set, respectively. You can find these two files already generated for the first fold, in the .zip given to you.

Open both files from inside Matlab. You will see that now the parameters are set to the same values than the ones used in task 1. The documents Automatic-FIR-ConfigurationFile.pdf and Automatic-FIR-DiscretizationFile.pdf describe each of the parameters involved in both, configuration and discretization, processes and explain how to use them.

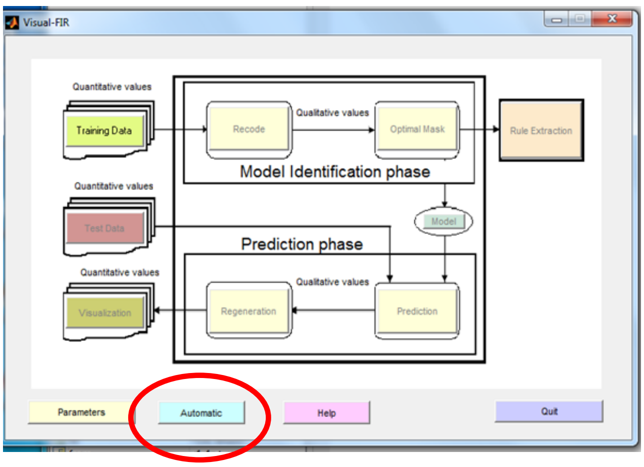
2) Once you have defined your discretization file AutomaticDiscretization.m, you need to load it into matlab and save it as a .mat file, because the automatic process of Visual-FIR reads it as .mat.

For now, we are going to keep the values already defined in the discretization file, so you just need to run the AutomaticDiscretization.m by selecting this file and then pushing run from the menu of the mouse right button. Afterwards, you need to save all the variables in the file AutomaticDiscretization.mat, by means of the matlab instruction:

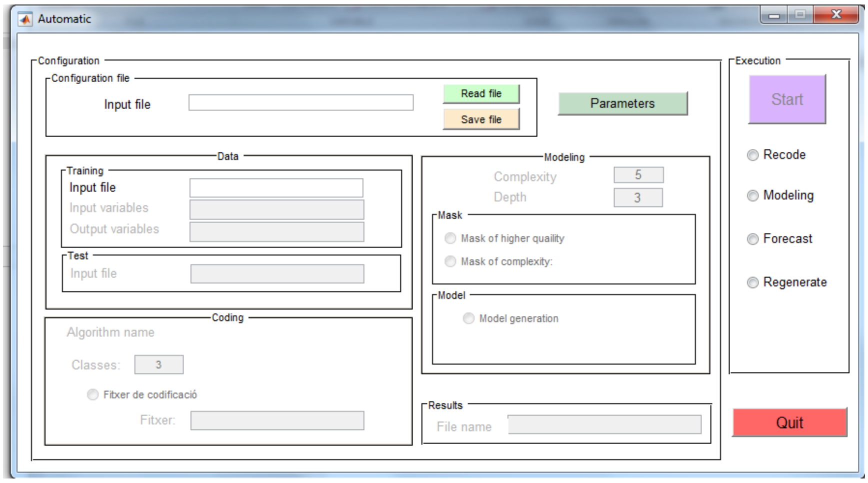
>> save AutomaticDiscretization

3) Clear the Workspace by typing clear in the matlab command window and perform the same process for the configuration parameters, i.e. run the AutomaticConfiguration.m and save all the variables in the file AutomaticConfiguration.mat.

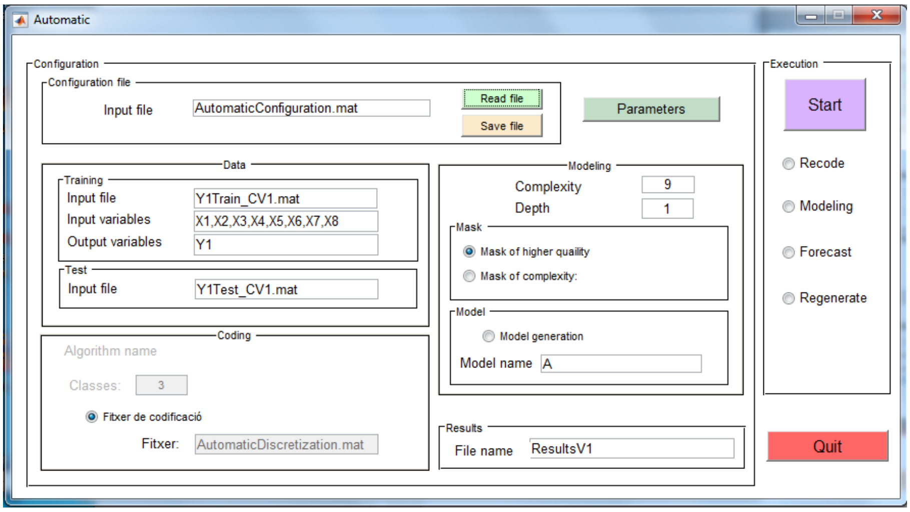
4) Open VisualFIR one more time. The automatic option for running the FIR methodology is included in the bottom of the VisualFIR window.



The following window appears:



5) You need to load the AutomaticConfiguration.mat file by selecting the *Read file* button. You will get the following screen:



where all the information needed to run FIR is loaded. You can change, if you like, some of the parameters directly in this screen. Notice that the results are going to be saved at the file the name of which has been defined in the AutomaticConfiguration file, in this example ResultsV1.mat.

6) You can repeat the same process for all the folds. Remember to clear the Workspace each time.

7) Close the VisualFIR GUI, clear the Workspace and go to the folder Data-NoVisual.

Task 4 – Energy perf. of residential buildings – Script in Matlab

Objective: In this task, we will learn how to run FIR methodology by using Matlab scripts. We will use the same energy data than in the previous tasks.

1) We have available three scripts: createFileDiscretization.m, createFileConfiguration.m and Auto.m. In the first one the Discretization parameters are defined. In the second one the

Configuration parameters are set. Finally, Auto.m is the script that loads the data generated with the previous scripts and executes sequentially all the processes of FIR methodology, i.e. recode, optimal mask, prediction and regeneration. The prediction results are saved in a file which name is set in the parameter *NomFitxerSor* that is located in the createFileConfiguration.m

2) In order to run it you should execute the three scripts in Matlab in the following order:

>> createFileDiscretization ('Discretization.mat')
>> createFileConfiguration ('Configuration.mat')
>> Auto('Configuration.mat')

The first script sets all the discretization parameters and saves them in the Discretization.mat file. The second script sets all the FIR parameters and save them in the Configuration.mat file. The third script uses both files to perform the FIR processes and save the results.

You can modify the scripts at your own convenience to run the FIR for all the folds in a given problem without using a graphical environment.