NLMIMO, NON-LINEAR MULTI-INPUT MULTI-OUTPUT TOOLBOX

Edy Bertolissi, Antoine Duchâteau, Hugues Bersini
IRIDIA, Université Libre de Bruxelles
50, Av. Franklin Roosevelt, 1050 Bruxelles, Belgium
email: eberto@iridia.ulb.ac.be, aduchate@ulb.ac.be, bersini@ulb.ac.be

Abstract. NLMIMO, Non-Linear Multi Input Multi Output, is a MATLAB toolbox for modeling non-linear dynamic systems. The toolbox has been developed using an object oriented approach. Model representations for linear, Takagi-Sugeno, Mamdani, and lazy mappings, as well as various model identification techniques, have been integrated. They are available as a set of interchangeable classes which can be easily extended. The toolbox is completed by a series of tools for the visualization of the models, and a database management utility for listing the properties of the mappings associated to the system, comparing their performance and validating them.

1 Introduction

Learning models from observations and studying their properties is a major issue in several scientific and industrial fields. People store observed data since they believe that it is possible to extract some useful information from it. However once the data has been collected it is necessary to decide how to learn something from it. Building a model is a way for trying to capture, and mathematically define, the rules governing a process.

Since the idea of extracting useful knowledge is present in many disciplines, from econometrics to machine learning, from statistics to control, a large number of approaches have been developed to solve this problem. Least-squares linear approximators, neural networks, fuzzy systems, and lazy learning approximators, are different ways of transforming the available empirical data in a set of mathematical equations useful for the intended application. These approaches are not equivalent. They present discrepancies with respect to the class of functions they are adapted to, the approximation efficiency and the number of parameters needed to approximate a given nonlinearity. An approach can be global or local, which means that parameter adjustment can have a global or local impact on the model. Other important distinctions among the possible approaches are their computational efficiency, their transparency, that is how readable the parametrisation is, and how easy it is to introduce prior knowledge into the model.

Some approaches are better suited than others to solve certain classes of problems. NLMIMO has been designed as a toolbox which allows the user to choose, experiment and compare different approaches for modeling from data. In the case of multi-output systems it is even possible to choose a different modeling approach for each output, increasing the overall flexibility of the system. At present NLMIMO implements linear, Takagi-Sugeno [7], Mamdani [5], and lazy [1] modeling approaches. Takagi Sugeno and Mamdani models are two well known types of fuzzy models. Lazy models, also known as instance based models, defers processing of the training data until a query explicitly needs to be answered. This means that all the training data needs to be stored in memory and accessed when a query is made. Due to the structure of the toolbox, the inclusion of other approaches is a simple process.

2 Overview of NLMIMO

Raw data is rarely immediately beneficial, but its true value derives from the possibility of extracting information which could be used for understanding the process which generated the data, and predicting its future values. In many cases previous knowledge about the process is advantageous for better treating the observed data. This means that the process for identifying a model is not sequential but iterative: it implies feedback from the user and interactions where the model is refined until it provides a good description of the phenomenon underlying the data. During the refinement procedure, the user must always keep into mind the objective of the modeling procedure, the context in which the model will be used. For example a model suited for prediction is not always adapted to control.

The main scope behind NLMIMO is the integration of the software produced by the partners taking part in the FAMIMO project (ESPRIT LTR Project 21911) in a unique toolbox. Therefore it has been necessary to design a flexible system that can accommodate the needs of the different laboratories collaborating to the project.The use of an object oriented approach, which allows modularity, flexibility

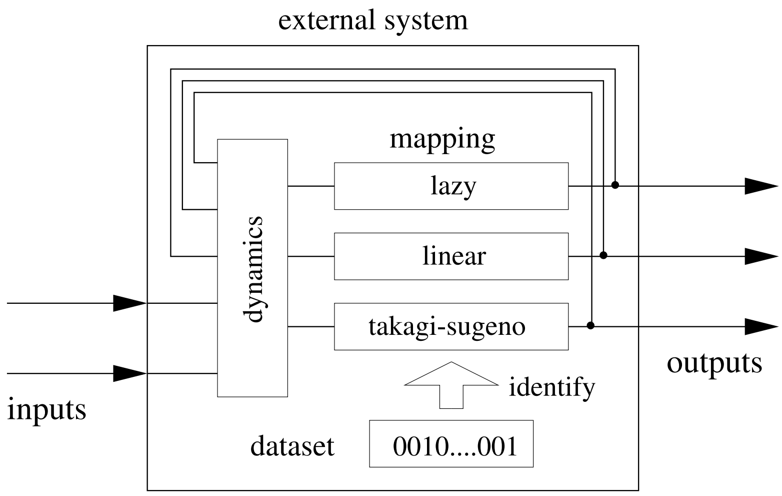

Figure 1: Relationship among the classes defined in the NLMIMO toolbox

and ease of extension had been a natural choice for the design of the toolbox. This approach has been possible thanks to the new object oriented features introduced starting from MATLAB 5.0.

The NLMIMO toolbox is built around three main classes. The `system` class, which defines all the attributes which characterize a system, the `dataset` class, which allows the storage and the processing of the data, and the `mapping` class, which defines the modeling of the process. The `system` class is the main class of the toolbox, and the other ones are used to describe some particular features of the system. In this way the data and the parameterization of the model are separate from the description of the system, and therefore they can be easily modified or updated without the need of altering the structure of the `system` class. Each class has methods for accessing, setting and displaying its attributes, and checking its integrity and consistency.

The `dataset` class is used to handle the data. Once an object belonging to this class is initialized with raw data, several methods for performing data manipulation (e.g. filtering) and data analysis (e.g. clustering) are implemented.

The `mapping` class is used to define an inputs/outputs relation. Since it is possible to choose among different possible alternatives for defining the mapping between a specified number of inputs and outputs, the `mapping` abstract class, defines all the features which are common to all the possible mappings. The definition of the details, which are dependent from the particular descriptor, are left to its subclasses. Virtually every type of non-linear function can be implemented as a subclass of the `mapping` class. At present linear systems, Mamdani fuzzy systems, Takagi-Sugeno fuzzy systems, and lazy learning systems (integrated from [4]) are available in the toolbox. New mappings can be easily added by defining other subclasses of the `mapping` class. With the exception of the linear models, all the mappings may be used to define nonlinearities. Since they have been shown to be universal approximators for certain classes of functions, it means that they are equivalent with respect to the nonlinearity which they approximate. Even if from the process point of view it is the nonlinearity that matters, not how it is parametrized, from the point of view of the designer the parameterization selected can be very important.

The `system` class centralizes the information needed to describe a general static or dynamic system. Since a system can be represented using an internal (state space based) or external (input-output based) representation, two subclasses, the `external` and the `internal` one, have been defined to implement these two alternative descriptions. Figure 1 shows a graphical representation of a 2 inputs 3 outputs system belonging to the `external` class. The object stores the data associated to the system which is used for the identification of the 3 mappings which define the input-output relations. It possible to see that a different type of mapping is used for modeling each output. The structure of this class allows a very general representation of dynamical systems, and lets the user experiment and mix different types and sizes of models.

3 NLMIMO functionalities

The NLMIMO toolbox offers to the user all the functionalities necessary to carry out a complete modeling of the system starting from data file. This can be done building the required objects, and applying the appropriate methods entering the appropriate commands at the MATLAB command line. An alternative approach consists in taking advantage of the NLMIMO graphical user interface (GUI), which has being designed to help the user to carry out the modeling process in a structured way.

The identification process of the model of the manifold pressure of a direct injection engine will be used to describe the functionalities of the toolbox. The engine has two inputs, the fresh air throttle control and the engine speed, and one output, the manifold pressure. A sequence of 2000 input-output pairs is available for identification purposes. First of all it is necessary to define the engine as an object belonging to the `external` class. Then it is necessary to specify the limits of the variables, the sampling time and dynamics of the system. Finally the training data is attached to the object, and the parameter identification procedure can start.

Parameter identification methods

Parameter identification is a process which consists in determining the parameters associated with the selected mapping. This is a procedure which is highly dependent of the type of chosen mapping, and the features to optimize.

Not all the mappings implemented in the toolbox require this step. Lazy learning modeling does not need parameter identification, since the local model is built on query-by-query basis. The whole parameter identification process consists in manually selecting the dimension of the regressor, defining the minimum and maximum number of neighbors considered in each query and choosing the type of approximator. Similarly for the Mamdani fuzzy systems there is no parameter identification step, since these models are usually defined starting from the knowledge of an expert about the process. The linguistic description of the process is used to build the model, which is then manually tuned to maximize its performance.

The other mappings implemented in the toolbox define methods which, once defined some mapping dependent constraints, automatically determine the parameters associated with the selected mapping. Linear models use the least mean square method to find the parameters associated to the model. Takagi Sugeno mappings offer a wide choice of identification methods: identification by product-space clustering (`fmclust` [2], `cluslms`, `cluslev`), incremental identification (`incrsie` [6], `tsgklmsxv` [3]), and global identification (`randlm`).

It is possible to use different mappings and choose alternative identification procedures in order to identify the model of the manifold pressure system. The left part of figure 2 shows the NLMIMO visualization tool after the system has been identified using five different identification procedures (listed in the top left corner of the image). Even if in this case only one identified model has been selected for visualization, it is possible to choose at the same time multiple mappings, and visually compare their shapes, errors and other interesting features.

Model validation methods

Model validation, the process which leads to the model selection, is application specific. It is often the result of a compromise between different factors, such as the quantitative measurements, the personal experience of the designer and the effort needed to implement a particular model in practice. The NLMIMO toolbox provides a series of functionalities for evaluating quantitative criteria commonly used in validation techniques. An error database allows to access to the simulation and prediction results of the different mappings, visualizing and comparing them. The results of other statistical criteria such as estimated prediction error and the results of the cross validation can be analyzed and compared.

The right part of figure 2 displays the error database of the manifold pressure which stores the mean square error of the different mappings. In this particular case it is possible to see that the identification methods based on the Takagi-Sugeno fuzzy system provide comparable results both on the training and validation set. The performance of the lazy mapping is inferior, mainly because the number of training points is quite low in comparison with the complexity of the process. Without surprises the linear

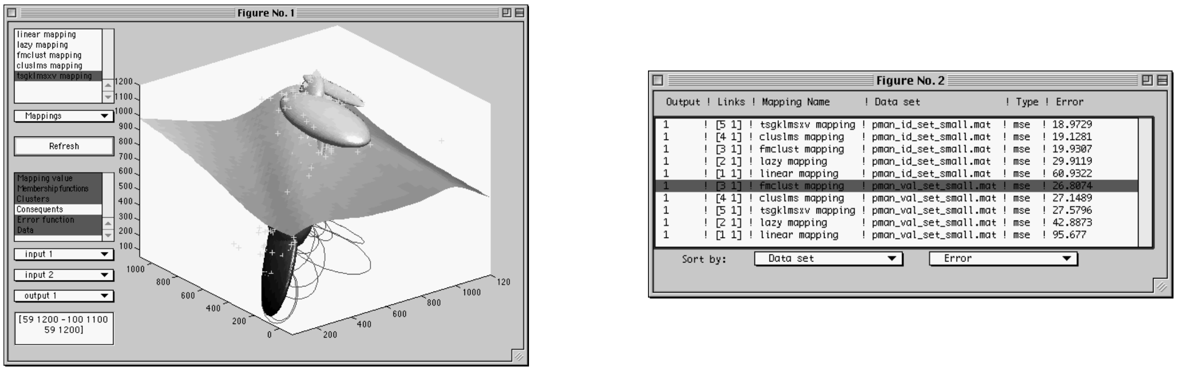

Figure 2: The model viewer (left) allows to visualize the models generated by the different mappings associated to the system, while the model error database (right) displays the list of all means squares error of the mappings of the system.

mapping gives very poor results. A similar database exists for storing the results of the means square simulation errors which are calculated using the input data and using the outputs calculated by the model to build the new regressor. This means that errors accumulate during the simulation, which let us check if the model really captures the dynamic of the system . The simulation of the models can be displayed to compare the performance of the mappings.

4 Conclusion

The NLMIMO toolbox implements several state of the art techniques used in multimodel based modeling and identification processes. Even if the modeling aspects of the toolbox are the predominant ones, support for the development of controllers and the study of their stability is also included in the toolbox. One of the main targets of this toolbox has been to design a flexible set of utilities which allow the user to experiment and compare several solutions. Even if computational efficiency was an important feature in the design of the toolbox, it was not the main issue. If a particularly performant system is required, the best approach is to reimplement the relevant functions once the best method has been identified. Future works include the integration of other identification techniques such as neural networks and mixture of experts in the toolbox, and the release this toolbox to the scientific community on the Internet.

References

[1] D. W. Aha. Editorial. *Artificial Intelligence Review*, 11(1–5):1–6, 1997.

[2] R. Babuska. *Fuzzy Modeling and Identification.* PhD thesis, Technische Universiteit Delft, 1996.

[3] H. Bersini, A. Duchateau, and N. Bradshaw. Using incremenmtal learning algorithms in the search for minimal and effective fuzzy models. In *Proceedings of the FUZZ-IEEE '97*, pages 1417–1422, Barcelona, Spain, 1997.

[4] M. Birattari and G. Bontempi. The lazy learning toolbox, for use with matlab. Technical Report TR/IRIDIA/99-7, IRIDIA-ULB, Brussels, Belgium, 1999.

[5] E. H. Mamdani. Application of fuzzy algorithm for control simple dynamic plant. *Proc. IEEE*, 121(12):1585–1588, 1974.

[6] D. Passaquay, P. Bortolet, S. Boverie, and A. Titli. Iterative fuzzy modelling of non-linear functions, application for the control of non-linear systems. In *Proceedings of the European Control Conference, ECC'99*, Karlsruhe, Germany, 1999.

[7] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.