Is readability compatible with accuracy?: from Neuro-Fuzzy to Lazy Learning*

Hugues Bersini, Gianluca Bontempi, and Mauro Birattari

Iridia - CP 194/6
Université Libre de Bruxelles
50, av. Franklin Roosevelt
1050 Bruxelles - Belgium
email: {bersini, gbonte, mbiro}@ulb.ac.be

Abstract. The composition of simple local models for approximating complex nonlinear mappings is a common practice in recent modeling and control literature. This paper presents a comparative analysis of two different local approaches: the neuro-fuzzy inference system and the lazy learning approach.
A *neuro-fuzzy* system is an hybrid representation which combines the linguistic description of fuzzy inference systems with learning procedures inspired by neural networks. *Lazy learning* is a memory-based technique that uses a query-based approach to select the best local model configuration by assessing and comparing different alternatives in cross-validation. The two approches are compared both as learning algorithms and as identification modules of an adaptive control system. The paper will show how the lazy learning is able to provide better modeling accuracy and control performance at the cost of a reduced readibility of the resulting approximator. Illustrative examples of identification and control of a nonlinear system starting from simulated data are given.

1 Introduction

The problem of modeling a process from observed data has been the object of several disciplines from nonlinear regression to machine learning and system identification. In the literature dealing with this problem, two main opposing paradigms have emerged: global versus local methods.

Global models have two main properties. First, they cover the whole set of operating conditions of the system underlying the available data. Second, global models solve the problem of learning an input-output mapping as a problem of function estimation, that

* The work of Gianluca Bontempi was supported by the European Union TMR Grant FMBICT960692. The work of Mauro Birattari was supported by the F.I.R.S.T. program of the Région Wallonne, Belgium.

is of choosing from a given set of parametric functions $f(\varphi, \alpha), \alpha \in \Lambda$, the one which best approximates the unknown data distribution. Examples are linear models, nonlinear statistical regressions, splines, neural networks.

The local paradigma originates from the idea of relaxing one or both of the global modeling features.

In the first case, the global description is replaced by a modular architecture where the different modules are simple models which focus on different part of the input space. It is the idea of *operating regimes* which assumes a partitioning of the operating range of the system in order to solve modelling and control problems (Johansen & Foss, 1993). Fuzzy inference systems (Takagy & Sugeno, 1985), RBF (Moody & Darken, 1989), CART (Breiman *et al.*, 1984), HME (Jordan & Jacobs, 1994), are well-known examples of this approach. It is important to remark how, although these architectures are characterized by an augmented readibility and an easier interpretation, they still are a particular type of functional approximators.

Memory-based methods (Atkeson, 1992) aim to solve the learning problem taking the opposite direction. Given that the problem of functional estimation is hard to be solved in a generic setting, they focus on approximating the function only in the neighborhood of the point to be predicted. To this aim, the whole data set is kept intact as opposed to functional methods which discard the data after use. Memory-based techniques are an old idea in time series prediction (Farmer & Sidorowich, 1987), classification (Cover & Hart, 1967) and regression (Cleveland, 1979). The idea of memory-based approximators as alternative to global models originated in non-parametric statistics (Epanechnikov, 1969) to be later rediscovered and developed in the machine learning field (Aha, 1989).

This paper will focus on *neuro-fuzzy inference systems* and *lazy learning* as prototypes of these two different ideas of local modeling. The aim is to provide the reader with a comparison between these two approaches in modeling and control.

Neuro-fuzzy systems (Brown & Harris, 1994), (Jang *et al.*, 1997), (Bersini & Bontempi, 1997) are an example of hybrid modeling. The basic idea underlying these models is to reconcile a dichotomy emerged in literature between different approaches to the implementation of intelligent systems: on one hand approaches, like neural networks, which renounce readability for performance and on the other knowledge based systems, like fuzzy systems, based on production rules with the aim of harmonizing the continuous nature of the reality with the symbolic nature of human reasoning. In fact, a third way is provided by hybrid approaches, that are methods which employ available knowledge as a way to improve not only the readability of the models but also the performance of data-

driven learning methods. In this contribution, we will propose our neuro-fuzzy technique which integrates a fuzzy clustering inizialization, a combination of a linear and non linear parameter estimation routines and a cross-validation procedure for model selection.

Lazy learning (Aha, 1997) designates the whole set of memory-based techniques that defer processing of the dataset until they receive request for information (e.g. prediction or local modeling). There has been recently a new impetus to the adoption of these techniques for modeling (Atkeson *et al.*, 1997a) and control problem (Schaal & Atkeson, 1994), (Atkeson *et al.*, 1997b). Here, we propose a lazy learning technique, having as main feature the adoption of enhanced statistical procedures to identify the local approximator. In particular, we use the PRESS statistic (Myers, 1990) which is a simple, well-founded and economical way to perform *leave-one-out* cross validation (Stone, 1974) and to assess the performance in generalization of local linear models.

The contribution of the paper in the control domain will be a comparison of the two approaches as alternative methods to extend linear control techniques to nonlinear discrete-time control problems (Murray-Smith & Johansen, 1997). In particular, we will see a self-tuning regulator (STR) architecture (Astrom, 1983) where discrete-time conventional control (e.g. generalized minimum variance, pole placement) is combined with local model identification. The control system can be thought of as composed of two loops. The inner one consists of the process and a feedback regulator. The parameters of the regulator are adjusted by the outer loop, represented by a neuro-fuzzy identifier or a lazy learning estimator, respectively.

The experimental results will show how the lazy learning approach outperforms the neuro-fuzzy method both in identification and control tasks. Moreover, we will show how lazy learning takes further advantage from its memory-based nature. In fact, this feature makes of lazy learning a promising method for extending adaptive techniques to local modeling methods.

The remainder of the paper is organized as follows. In section 2 we will describe our neuro-fuzzy architecture. In section 3 we will introduce the lazy modeling technique based on an iterative selection procedure. Details on the control system implementation are given in section 4.1. In section 5 simulation examples of identification and of control are given. Finally, in section 6 a comparison between the two approaches in terms of readibility versus accuracy is provided.

2 Neuro-fuzzy as a multimodel description

Let us consider a generic input-output mapping $f\colon \Re^m \to \Re$. Takagi and Sugeno (1985) introduced the fuzzy rule-based system for nonlinear modeling, usually referred in literature to as TS model. A TS fuzzy inference system is a set of r rules

$$\begin{cases} \text{If } \varphi_1 \text{ is } A_1^1 \text{ and } \varphi_2 \text{ is } A_2^1 \dots \text{ and } \varphi_m \text{ is } A_m^1 \text{ then } y^1 = f^1(\varphi_1, \varphi_2, ..., \varphi_m) \\ \dots \\ \text{If } \varphi_1 \text{ is } A_1^r \text{ and } \varphi_2 \text{ is } A_2^r \dots \text{ and } \varphi_m \text{ is } A_m^r \text{ then } y^r = f^r(\varphi_1, \varphi_2, ..., \varphi_m) \end{cases} \tag{1}$$

The first part (antecedent) of each rule is defined as a fuzzy AND proposition where A_j^i is a fuzzy set on the jth premise variable defined by the membership function $\mu_j^i\colon \Re^m \to [0,1]$. The second part (consequent) is a crisp function f^i $i = 1, \dots, r$ of the input vector $[\varphi_1, \varphi_2, \dots, \varphi_m]$.

By means of the fuzzy sets A_j^i the input domain of the function f is softly partitioned in smaller regions where the mapping is locally approximated by the models f^i. The TS inference system uses the weighted mean criterion to recombine all the local representations in a global approximator:

$$y = \frac{\sum_{i=1}^r \mu^i y^i}{\sum \mu^i} \tag{2}$$

where μ^i is the degree of fulfilment of the ith rule.

An interesting special case is provided by the linear TS fuzzy inference system where the consequents are linear models $f^i = \sum_{j=1}^m a_j^i \varphi_j + b^i$ (Sugeno & Kang, 1988). In this case the TS system can be used to return a local linear approximation about a generic point of the input domain. Consider for example an input $\hat{\boldsymbol{\varphi}} = [\hat{\varphi_1}, \hat{\varphi_2}, \dots, \hat{\varphi_m}]$. The TS rule combination (Eq. 2) returns a linear approximation $f_{lin}(\cdot)$ to the function $f(\cdot)$ about $\hat{\boldsymbol{\varphi}}$:

$$f_{lin}(\hat{\boldsymbol{\varphi}}) = \frac{\sum_{i=1}^r \mu^i (\sum_{j=1}^m a_j^i \hat{\varphi_j} + b^i)}{\sum \mu^i} \tag{3}$$

In a conventional fuzzy approach the membership functions and the consequent models are fixed by the model designer according to a priori knowledge. If this knowledge is not available but a set of input-output data is observed from the process f, the components of the fuzzy system (membership and consequent models) can be represented in a parametric form and the parameters tuned by a learning procedure. In this case the fuzzy system turns into a *neuro-fuzzy* approximator (Bersini & Bontempi, 1997). Neuro-fuzzy systems are a powerful trade off in terms of readability and efficiency between

a human-like representation of the model and a fast learning method. However, what mainly distinguishes neuro-fuzzy estimators from other kinds of non linear approximators is their potentiality for combining available a priori first principle models with data driven modeling techniques (Bontempi & Bersini, 1997). In fact, while learning methods provide the adaptation of the inference system to the observed data, the fuzzy architecture allows an easy integration into the system of available knowledge about the process to be modeled.

Let us see now in detail our neuro-fuzzy learning procedure.

2.1 Architecture and learning algorithms for neuro-fuzzy inference systems

In a neuro-fuzzy systems two types of tuning are required, designated as *structural* and *parametric tuning*.

Structural tuning concerns the structure of the architecture: which variables to account for in the rules, how to partition each variable domain, how many rules,.... Once available a satisfactory structure, the parametric tuning must search for the optimal membership functions together with the optimal parameters for the consequent models. There may be a lot of structure/parameter combinations which make the fuzzy model behaving in a satisfactory way. As a consequence, the search of the optimum in this two dimensional space is not that easy to conduct. As a rule, simple fuzzy models are generally preferred to complex ones so that, a first reduction of the optimization problem consists in restricting the search to region coding for simple fuzzy models. This is implicitly achieved by turning the cost to optimize into a combination of two objectives: good generalization performance and low complexity (Vapnik, 1995). We assume as index of complexity the number of rules of the architecture, so the goal of the whole tuning procedure is to find the optimal number of rules which gives the least error in generalization. Our learning procedure can be represented by the flow chart in Fig. 1. In this approach, the initialization of the architecture is provided by a hyperellipsoidal fuzzy clustering procedure inspired to (Babuska & Verbruggen, 1997). This procedure clusters the data in the input-output domain obtaining a set of hyperellipsoids which are a preliminary rough representation of the I/O mapping. The parameters of the ellipsoids (eigenvalues) are used to initialize the parameters of the consequent functions f^i, while their localization (projection of their mean on the input domain) determines the initial position of the membership functions in the input domain.

Once the initialization is done, the learning procedure begins. Two optimization loops are nested: the parametric and the structural one. The parametric loop (inner) finds the best

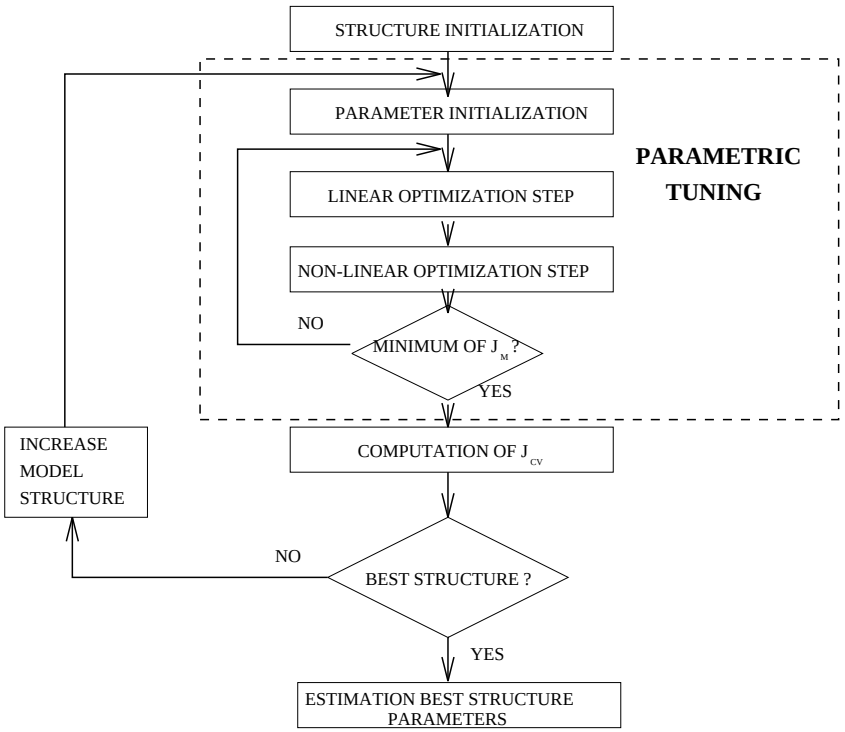

Figure 1: Flow-chart of the neuro-fuzzy learning procedure.

parametric configuration by minimizing a sum of squares cost function J_M depending exclusively on the training set. For linear TS models , the minimization routine can be furtherly decomposed in a least-squares problem to estimate the linear parameters of the consequent models f^i and a nonlinear minimization (Levemberg-Marquardt) to find the parameters of the membership functions A_j^i.

The structural identification loop searches for the best structure (in terms of optimal number of rules) by increasing gradually the number of rules, and consequently of local models. The different structures are assessed and compared according to their performance J_{CV} in cross-validation (Stone, 1974). The model with the best cross-validation performance is then selected as the candidate to represent the input-output mapping and consequently trained on the whole data set.

This procedure uses a high proportion of the available data to train the model, while also making use of all data points in evaluating the cross-validation error. Unfortunately, this approach has the disadvantage that the training process has to be repeated as many times as the number of partitions of the training set. Then, the whole learning process (i.e. the sequence of initialization, optimization and validation) results extremely time-consuming.

3 Lazy learning modeling

Lazy learning returns no functional approximation but the value of the unknown function is estimated focusing on the region surrounding the point where the estimation itself is required.

Let us consider an unknown mapping $f: \Re^m \rightarrow \Re$ of which we are given a set of N samples $\{(\boldsymbol{\varphi}_1, y_1), (\boldsymbol{\varphi}_2, y_2), \ldots, (\boldsymbol{\varphi}_N, y_N)\}$. These examples can be collected in a matrix $\boldsymbol{\Phi}$ of dimensionality $[N \times m]$, and in a vector $\mathbf{y}$ of dimensionality $[N \times 1]$.

Given a specific query point $\boldsymbol{\varphi}_q$, the prediction of the value $y_q = f(\boldsymbol{\varphi}_q)$ is computed as follows. First, for each sample $(\boldsymbol{\varphi}_i, y_i)$ a weight w_i is computed as a function of the distance $d(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_q)$ from the query point $\boldsymbol{\varphi}_q$ to the point $\boldsymbol{\varphi}_i$. Each row of $\boldsymbol{\Phi}$ and $\mathbf{y}$ is then multiplied by the corresponding weight creating the variables $\mathbf{Z} = \mathbf{W}\boldsymbol{\Phi}$ and $\mathbf{v} = \mathbf{W}\mathbf{y}$, with $\mathbf{W}$ diagonal matrix having diagonal elements $\mathbf{W}_{ii} = w_i$. Finally, a locally weighted regression model (LWR) is fitted solving the equation $(\mathbf{Z}^T\mathbf{Z})\boldsymbol{\beta} = \mathbf{Z}^T\mathbf{v}$ and the prediction of the value $f(\boldsymbol{\varphi}_q)$ is obtained evaluating such a model in the query point:

$$\hat{y}_q = \boldsymbol{\varphi}_q^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{v}. \tag{4}$$

Typically, the data analyst who adopts a local regression approach, has to take a set of decisions related to the model (e.g. the number of neighbors, the weight function, the parametric family, the fitting criterion to estimate the parameters). Our lazy learning method extends the classical approach with a method that automatically selects the adequate configuration.

To this aim, we import tools and techniques from the field of linear statistical analysis. The most important of these tools is the PRESS statistic (Myers, 1990), which is a simple, well-founded and economical way to perform *leave-one-out* cross validation and therefore to assess the performance in generalization of local linear models. This statistic returns the leave-one-out cross-validation error of a linear model at the same computational cost of the linear regression. Assessing the performance of each linear model, alternative configurations can be tested and compared in order to select the best one. This same selection strategy is indeed exploited to select the training subset among the neighbors, as well as various structural aspects like the features to treat and the degree of the polynomial used as a local approximator (Bersini *et al.*, 1998). The general ideas of the approach can be summarized as follows.

1. The task of learning an input-output mapping is decomposed in a series of linear estimation problems.

2. Each single estimation is treated as an optimization problem in the space of alternative model configurations.
3. The estimation ability of each alternative model is assessed by the cross-validation performance computed using the PRESS statistic.

In order to make these operations more effective, we proposed two innovative algorithms, based on the adoption of recursive techniques for the linear parameter estimation and on a paired permutation test for the comparison of the performance of different model candidates (Bontempi *et al.*, 1998).

4 Neuro-fuzzy and Lazy learning for control: a comparative analysis

Although nonlinearity characterizes most real control problems, methods for analysis and control design are considerably more powerful and theoretically founded for linear systems than for nonlinear ones. In the following, a comparison between the neuro-fuzzy and the lazy approach as two ways of extending linear techniques to nonlinear problems is provided.

Neuro-fuzzy architectures A neuro-fuzzy architecture is a particular example of local model network. It extends the concept of operating point by introducing the notion of *operating regime*. An operating regime is a set of operating points where the system behaviour can be described approximately with a simple model (Johansen & Foss, 1993), (Johansen & Foss, 1995). To each of them a validity region, and a local description of the system behavior are associated.

In the neuro-fuzzy formalism the validity region of a local model f^i is represented by the corresponding membership functions μ^i in Equation 1. . One major advantage of the approach is the possibility to integrate a priori knowledge with a parametric learning procedure. A disadvantage is related to the fact that, in order to cover the whole operating region, a generic neuro-fuzzy architecture still remains a nonlinear approximator. As a consequence, the estimation requires time-consuming learning and validation procedures.

Lazy learning This approach shares with neuro-fuzzy the idea of decomposing a difficult problem in simpler local problems. Also, both the approaches can return a local linear description of the process (see Eq. 4 and 3).

The main difference concerns the model identification procedure. Local model networks aim to estimate a global description to cover the whole system operating domain, whereas memory based techniques focus simply on the current operating point.

Neuro-fuzzy results more time consuming in identification but faster in prediction. However when new data are observed, model update may require to perform the whole neuro-fuzzy modeling process from scratch. On this matter lazy learning takes an advantage from the absence of a functional approximator: once a new input-output example is observed, it is enough to update the database which stores the set of input-output pairs. Therefore, lazy learning is intrinsically adaptive while neuro-fuzzy requires proper on-line procedures to deal with sequential problems.

In the following section we will introduce a local indirect controller where the two approaches are employed to implement the identification module. This will allow an experimental comparison both on identification and on control simulations.

4.1 The local self-tuning controller

Consider a class of discrete-time single input single output (SISO) dynamic systems whose equations of motion can be expressed in the form:

$$y(k) = f\big(y(k-1), \ldots, y(k-ny), u(k-d), \ldots, u(k-d-nu), \\ e(k-1), \ldots, e(k-ne)\big) + e(k), \quad (5)$$

where k denotes the time, $y(k)$ is the system output, $u(k)$ the input, $e(k)$ is a zero-mean disturbance term, $d > 0$ is the relative degree and $f(\cdot)$ is some nonlinear function. Defining the information vector as

$$\boldsymbol{\varphi}(k-1) = \big[y(k-1), \ldots, y(k-ny), u(k-d), \ldots, u(k-d-nu), e(k-1), \ldots, e(k-ne)\big], \quad (6)$$

the system (5) can be written in the form:

$$y(k) = f\left(\boldsymbol{\varphi}(k-1)\right) + e(k). \quad (7)$$

An indirect control scheme (Astrom & Wittenmark, 1990), combines a parameter estimator, which computes an estimate $\hat{\boldsymbol{\vartheta}}$ of the unknown parameters, $\boldsymbol{\vartheta}$ with a control law $u(k) = K(\boldsymbol{\varphi}(k), \boldsymbol{\vartheta})$ implemented as a function of the plant parameters. In conventional adaptive control theory, to make the problem analytically tractable, the plant is assumed to be a linear time-invariant system with unknown parameters.

$$A(z)y(k) = z^{-d}B(z)u(k) + C(z)e(k), \quad (8)$$

The identification of the unknown polynomials A, B and C is performed by a recursive parameter estimator which updates the same linear model when a new input-output sample is observed.

Our approach combines the local learning identification procedures described in sections 2 and 3 with conventional linear control techniques. There is no global linear model description but at each time-step the system dynamics (7) is linearized by the local model in the neighborhood of the current operating regime. In the neuro-fuzzy case we consider linear TS model, which can return at each operating point a linear approximation of the systems dynamics (see Eq. 3). In the lazy learning formalism the linear parametrization is returned by the local weighted regression (see Eq. 4).

In order to control the process, we adopt standard linear techniques as the minimum-variance (MV) and the pole-placement (PP) control technique (Astrom & Wittenmark, 1990). The MV control problem can be stated as finding the control law which minimizes the variance of the output. The MV controlled closed loop system is stable only if B has all of its roots inside the unit circle (minimum phase). However, more complex formulations are available in the case of a tracking problem or in the case of non minimum-phase systems (Generalized MV or GMV). In these cases it is possible to select properly the controller parameters in order to make the closed loop system asymptotically stable. Pole placement design is an alternative technique to deal with non minimum-phase configurations. The procedure requires first to choose the desired closed loop pole positions and then to calculate the appropriate controller.

The whole control algorithm is described in detail in Fig. 2. Note that with the term *Local model* we identify the identification module (neuro-fuzzy or lazy) which returns a local approximation to the system dynamics.

1. Acquisition of the the vector (6).
2. Linearization of the function $f(\cdot)$ about (6). The linearization is computed by the *Local model* algorithm.
3. Derivation of the polynomials A, B, C of (8) from the linearized model.
4. Design of a MVG/PP controller for (8) which satisfies the required properties (stability, accuracy, speed ...) of the closed loop behavior.
5. Computation of the control signal.

Repeat these steps at each sampling period.

Figure 2: The local self-tuning controller algorithm.

5 Simulation studies

In this experimental study we will consider a nonlinear SISO system described by the difference equation:

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)\big(y(k-2)-1\big)u(k-1)+u(k)}{1+y^2(k-1)+y^2(k-2)} \tag{9}$$

The system is represented in the input-output form $y(k+1) = f\big(y(k), y(k-1), y(k-2), u(k), u(k-1)\big)$. We assume to have an initial database DB of 5000 points collected by exciting the system with a random uniform input (zero mean and unit variance).

5.1 The local identification of a nonlinear discrete-time system

In this simulation we consider the task of predicting the output of system refeq:bench, once excited by a random test input $u(k)$ having the same statistical properties of the data in DB.

The prediction is done for 500 time steps assuming to have available at each instant k the regression vector $[y(k), y(k-1), y(k-2), u(k), u(k-1)]$. Let us see the performance of the two local approaches.

Neuro-fuzzy. We consider an architecture with triangular membership function and linear consequents. The training database DB is employed to select the neuro-fuzzy structure having the least generalization error in cross-validation.

We chose $r = 6$ number of rules as the optimal complexity. The model with 6 rules was then estimated on the whole data set. The plot in Fig. 3a shows the identification error. We obtained an RMSE=0.04 (root mean square error).

Lazy learning. We use the same training database DB. We adopted the recursive identification method described in (Bontempi *et al.*, 1998) to estimate the local model. In spite of the fact that a local model has to be estimated for each prediction, the whole learning process (training, validation and prediction) was largely shorter than in the neuro-fuzzy case (3 minutes vs. 2 days of computation time). Also, we were able to obtain a better performance (RMSE=0.03). The plot in Fig. 3b shows the identification error.

5.2 The local self-tuning control of a nonlinear discrete-time system

In this simulation we consider the control of the nonlinear system described by Eq. 9. The reference output $y_{ref}(k)$ is given by a periodic square wave. The controller is the pole placement STR controller described in section 4.1.

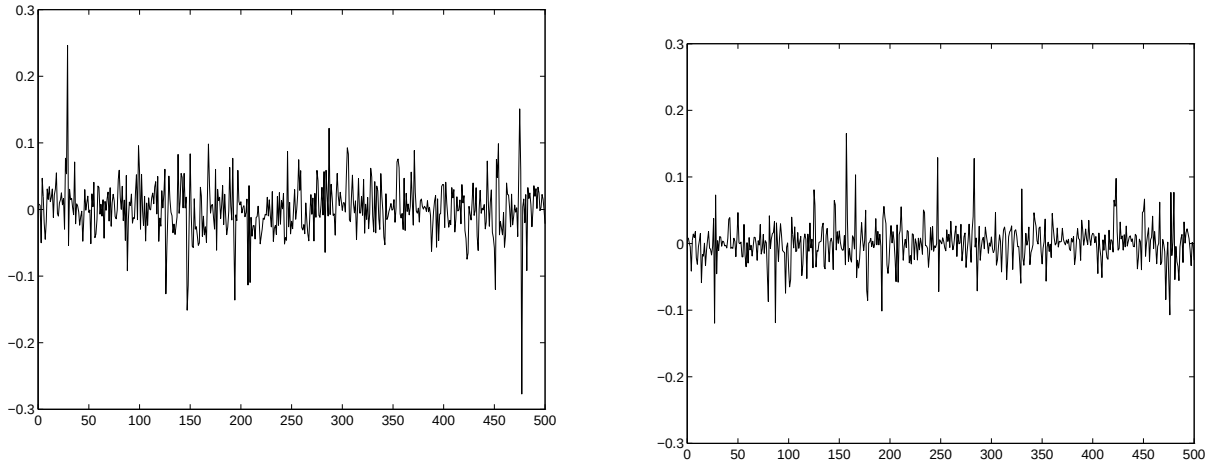

Figure 3: System identification: a) Neuro-fuzzy: identification error (RMSE=0.04) b) Lazy learning: identification error (RMSE=0.03).

Neuro-fuzzy controller. In order to control the system we adopt the same structure which was identified in the previous section (6 inference rules). The plot in Fig. 4a shows the reference and the system output.

The control system behaviour exhibits a steady-state error. A possible explanation could be related to the fact that to follow the reference value the control signal u has to reach values corresponding to regions of the input domain not enough represented in the training set DB. The control error is consequent to the extrapolation error of the neuro-fuzzy model.

Lazy learning controller We will present two simulations. In both of them, the training database is initialized with DB, but while in the first one, the training database is kept fixed all along the simulation (non adaptive case), in the second one the database is updated on-line each time a new input-output pairs is returned by the simulated system (adaptive case). The plot in Fig. 4b shows the reference and the system output in the non adaptive case while Fig. 4c presents the adaptive case. The lazy non adaptive controller has a better performance than the neuro-fuzzy one, but the steady state error persists. This is not the case for the adaptive formulation. It is interesting to see how in this case the lazy controller is able to cancel the steady state error after few simulation steps, compensating to the deficiency of the non adaptive version.

6 Is readability compatible with accuracy?

Neuro-fuzzy and lazy learning share the *divide-and-conquer* approach of increasing accuracy in modeling by decomposing complex global problems in simpler local sub-problems.

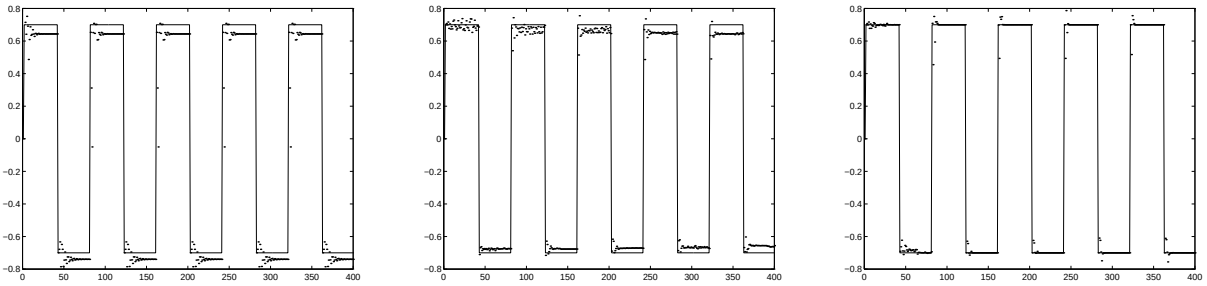

Figure 4: Self-tuning control: reference (solid) and system (dotted) outputs a) Neuro-fuzzy: RMSE=0.098 b) Lazy learning: RMSE=0.042 c) Lazy learning adaptive: RMSE=0.026 .

At the same time, they are on opposite sides for what concerns the readability of the resulting model.

Underlying neuro-fuzzy models is the idea that by hybridizing the adaptivity of neural nets together with the linguistic nature of fuzzy systems, it is possible to synthetize models which are not only accurate but also easy to interpret. On the other hand, while lazy learning appears to be among the most efficient techniques for modeling on the basis of data, it is also one of the least readable since it does not even conduct to any form of explicit representation neither in a linguistic form nor in a mathematical one. It rathers uses the raw data as the best model, never trying to explicitly capture the analytical structure underlying them.

Further, the experimental results of the paper seem to confirm that accuracy and readibility could direct towards opposite ways. The objective of extracting linguistic knowledge from an input-output mapping appears to be a constraint which limits the potentiality of the neuro-fuzzy model as a reliable estimator.

However and independently of our experiments, we can, by means of simple intuitive arguments, break the idea that you can both have readable and accurate models for the same price. Take the trivial example of fitting data sampled from a parabola shaped I/O distribution. If you use whatever statistical tools which include the possibility to fit data by polynomial, naturally the program will select for you a second degree polynomial as the best model to fit the data. As a matter of fact, a second degree polynomial is not a structure so easy to express with common words. Perhaps, the best you can do to describe the behaviour of a one-dimensional parabola is to use three rules akin to the following: “if x is nearly zero, y also”; “if x gets small, y will increase faster”; “if x increases, y will also increase faster”. Is it really necessary to use more than these three rules to capture the main information needed to describe the behaviour of the parabola? Suppose you try to accurately fit the data with a linguistic fuzzy model. You know this is something you can

do because it has been demonstrated that this type of model has sufficient approximation power once provided with the sufficient number of rules . Here resides a major problem since, even for a simple parabola, a very accurate fit will demand an impressive number of rules. Now, even if each of these rules is expressed in common words, having plenty of them becomes an information hard to be managed and with an informative power not superior to the original dataset.

In this context the main issue is no more the readibility/accuracy dilemma but becomes the generalization/approximation problem. If it is demonstrated that linguistic model can approximate any functional expression, there are no equivalent results that the generalization power of these formalisms is superior to others. The problem becomes important in the perspective of estimation from data where the real problem is not really to approximate a functional expression but how to select, starting from few samples, the model complexity that will generalize the best in front of fresh data. On this field, the lazy learning estimator appear to be superior to existing linguistic models. The availability of rigorous and validation algorithms is probably the secret of the success of lazy algorithms, which results more efficient in dealing with the key problem of estimation from limited data: find the best trade/off complexity vs. performance. On this matter, they can profit of the amount of theoretical results and experimental design collected along the years in linear statistical methods.

These reasons justify the separation to maintain, we believe, between the two objectives which are the readability of a model and its accuracy to fit the data. We believe also that the symbolism used for expressing the model should not be the same as the one use to construct the statistical approximator. If it is legitimate to aim at some qualitative knowledge of any observed process, it is far from obvious that this qualitative description could directly underlie the statistical structure that will accurately predict the data. In order to qualitatively reason about the process such linguistic type of knowledge could be useful, but as soon as precision is needed it is the right time to forget the common language and to use fine algorithms and clever mathematics.

References

[Aha, 1989] Aha D.W. 1989. Incremental, instance-based learning of independent and graded concept descriptions. *Pages 387–391 of: Sixth International Machine Learning Workshop*. San Mateo, CA: Morgan Kaufmann.

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

[Astrom, 1983] Astrom K.J. 1983. Theory and Applications of Adaptive Control - A Survey. *Automatica*, **19**(5), 471–486.

[Astrom & Wittenmark, 1990] Astrom K.J. & Wittenmark B. 1990. *Computer-controlled Systems: Theory and Design*. Prentice-Hall International Editions.

[Atkeson, 1992] Atkeson C.G. 1992. Memory-based approaches to approximating continuous functions. *Pages 503–521 of:* Casdagli M. & Eubank S. (eds), *Nonlinear Modeling and Forecasting*. Harlow, UK: Addison Wesley.

[Atkeson *et al.*, 1997a] Atkeson C.G. , Moore A.W. & Schaal S. 1997a. Locally weighted learning. *Artificial Intelligence Review*, **11**(1–5), 11–73.

[Atkeson *et al.*, 1997b] Atkeson C.G. , Moore A.W. & Schaal S. 1997b. Locally weighted learning for control. *Artificial Intelligence Review*, **11**(1–5), 75–113.

[Babuska & Verbruggen, 1997] Babuska R. & Verbruggen H. B. 1997. Fuzzy Set Methods for Local Modelling and Identification. *Pages 75–100 of:* Murray-Smith R. & Johansen T.A. (eds), *Multiple Model Approaches to Modeling and Control*. Taylor and Francis.

[Bersini & Bontempi, 1997] Bersini H. & Bontempi G. 1997. Now comes the time to defuzzify the neuro-fuzzy models. *Fuzzy Sets and Sytems*, **90**(2), 161–170.

[Bersini *et al.*, 1998] Bersini H. , Birattari M. & Bontempi G. . 1998. Adaptive memory-based regression methods. *In: Proceedings of the 1998 IEEE International Joint Conference on Neural Networks*. to appear.

[Bontempi & Bersini, 1997] Bontempi G. & Bersini H. . 1997. Identification of a sensor model with hybrid neuro-fuzzy methods. *Pages 325–328 of:* Bulsari A. B. & Kallio S. (eds), *Neural Networks in Engineering systems (Proceedings of the 1997 International Conference on Engineering Applications of Neural Networks (EANN '97), Stockolm, Sweden)*.

[Bontempi *et al.*, 1998] Bontempi G. , Birattari M. & Bersini H. . 1998. Recursive lazy learning for modeling and control. *In: Proceedings ofTenth European Conference On Machine Learning (ECML-98)*. to appear.

[Breiman *et al.*, 1984] Breiman L. , Friedman J.H. , Olshen R.A. & Stone C.J. 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

[Brown & Harris, 1994] Brown M. & Harris C.J. 1994. *Neurofuzzy adaptive modelling and control*. Hemel Hempstead: Prentice Hall.

[Cleveland, 1979] Cleveland W.S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, **74**, 829–836.

[Cover & Hart, 1967] Cover T. & Hart P. 1967. Nearest neighbor pattern classification. *Proc. IEEE Trans. Inform. Theory*, 21–27.

[Epanechnikov, 1969] Epanechnikov V.A. 1969. Non parametric estimation of a multivariate probability density. *Theory of Probability and Its Applications*, 153–158.

[Farmer & Sidorowich, 1987] Farmer J.D. & Sidorowich J.J. 1987. Predicting chaotic time series. *Physical Review Letters*, **8**(59), 845–848.

[Jang *et al.*, 1997] Jang J.-S. R. , Sun C.-T. & Mizutani E. 1997. *Neuro-Fuzzy and Soft Computing*. Matlab Curriculum Series. Prentice Hall.

[Johansen & Foss, 1993] Johansen T.A. & Foss B.A. 1993. Constructing NARMAX models using ARMAX models. *International Journal of Control*, **58**, 1125–1153.

[Johansen & Foss, 1995] Johansen T.A. & Foss B.A. 1995. Semi-Empirical Modeling of Nonlinear Dynamic Systems through Identification of Operating Regimes and Local Models. *Pages 105–126 of:* Hunt K.J. , Irwin G.R. & Warwick K. (eds), *Neural Network Engineering in dynamic control systems*. Springer.

[Jordan & Jacobs, 1994] Jordan M.I. & Jacobs R.A. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, **6**, 181–214.

[Moody & Darken, 1989] Moody J. & Darken C.J. 1989. Fast learning in networks of locally-tuned processing units. *Neural Computation*, **1**(2), 281–294.

[Murray-Smith & Johansen, 1997] Murray-Smith R. & Johansen T.A. (eds) 1997. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis.

[Myers, 1990] Myers R.H. 1990. *Classical and Modern Regression with Applications*. Boston, MA: PWS-KENT.

[Schaal & Atkeson, 1994] Schaal S. & Atkeson C. G. 1994. Robot Juggling: Implementation of Memory-Based Learning. *IEEE Control Systems*, February, 57–71.

[Stone, 1974] Stone M. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, **36**(1), 111–147.

[Sugeno & Kang, 1988] Sugeno M. & Kang G. T. 1988. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, **28**, 15–33.

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

[Vapnik, 1995] Vapnik V.N. 1995. *The Nature of Statistical Learning Theory*. Springer.