SOME ASPECTS ON NONLINEAR BLACK-BOX MODELING IN SYSTEM IDENTIFICATION

Lennart Ljung

*Dept of Electrical Engineering, Linköping
University, Sweden, ljung@isy.liu.se*

1 INTRODUCTION

The key problem in system identification is to find a suitable model structure, within which a good model is to be found. Fitting a model within a given structure (parameter estimation) is in most cases a lesser problem. A basic rule in estimation is *not to estimate what you already know*. In other words, one should utilize prior knowledge and physical insight about the system when selecting the model structure. It is customary to distinguish between three levels of prior knowledge, which have been color-coded as follows.

- White Box models: This is the case when a model is perfectly known; it has been possible to construct it entirely from prior knowledge and physical insight.
- Grey Box models: This is the case when some physical insight is available, but several parameters remain to be determined from observed data. It is useful to consider two subcases:
 - Physical Modeling: A model structure can be built on physical grounds, which has a certain number of parameters to be estimated from data. This could, e.g., be a state space model of given order and structure.
 - Semi-physical modeling: Physical insight is used to suggest certain nonlinear combinations of measured data signal. These new signals are then subjected to model structures of black box character.
- Black Box models: No physical insight is available or used, but the chosen model structure belongs to families that are known to have good flexibility and have been “successful in the past”.

A nonlinear black box structure for a dynamical system is a model structure that is prepared to describe virtually any nonlinear dynamics. There has been considerable recent interest in this area with structures based on neural networks, radial basis networks, wavelet networks, hinging hyperplanes, as well as wavelet transform based methods and models based on fuzzy sets and fuzzy rules. This paper describes the common framework for these approaches.

It is pointed out that the nonlinear structures can be seen as a concatenation of a mapping from observed data to a regression vector and a nonlinear mapping from the regressor space to the output space. These mappings are discussed separately. The latter mapping is usually formed as a basis function expansion. The basis functions are typically formed from one simple scalar function which is modified in terms of scale and location. The expansion from the scalar argument to the regressor space is achieved by a radial or a ridge type approach.

Basic techniques for estimating the parameters in the structures are criterion minimization, as well as two step procedures, where first the relevant basis functions are determined, using data, and then a linear least squares step to determine the coordinates of the function approximation. A particular problem is to deal with the large number of potentially necessary parameters. This is handled by making the number of “used” parameters considerably less than the number of “offered” parameters, by regularization, shrinking, pruning or regressor selection.

A more comprehensive treatment is given in [8] and [4].

2 SYSTEM IDENTIFICATION

System Identification is the art and methodology of building mathematical models of dynamical systems based on input-output data. See among many references [5], [6], and [9].

We denote the output of the dynamical system at time t by $y(t)$ and the input by $u(t)$. The data are assumed to be collected in discrete time. At time t we thus have available the data set

$$Z^t = \{y(1), u(1), \dots, y(t), u(t)\} \quad (1.1)$$

A *model* of a dynamical system can be seen as a mapping from past data Z^{t-1} to the next output $y(t)$ (a *predictor model*):

$$\hat{y}(t) = \hat{g}(Z^{t-1}) \quad (1.2)$$

We put a “hat” on y to emphasize that the assigned value is a prediction rather than a measured, “correct” value for $y(t)$.

The problem is to use the information in a data record Z^N to find a mapping $\hat{g}_N$ that gives good predictions in (1.2).

3 NON-LINEAR BLACK BOX MODELS

In this section we shall describe the basic ideas behind model structures that have the capability to cover any non-linear mapping from past data to the predicted value of $y(t)$. A model structure is a parameterized mapping of the kind (1.2):

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) \quad (1.3)$$

The parameter θ is a vector of coefficients that are to be chosen with the help of the data. We shall consequently allow quite general non-linear mappings g . This section will deal with some general principles for how to construct such mappings.

Now, the model structure family (1.3) is really too general, and it turns out to be useful to write g as a concatenation of two mappings: one that takes the increasing number of past observations Z^{t-1} and maps them into a finite dimensional vector $\varphi(t)$ of fixed dimension and one that takes this vector to the space of the outputs:

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1}) = g(\varphi(t), \theta) \quad \text{where} \quad \varphi(t) = \varphi(Z^{t-1}) \quad (1.4)$$

Let the dimension of φ be d . We shall call this vector the *regression vector* and its components will be referred to as the *regressors*.

The choice of the non-linear mapping in (1.3) has thus been reduced to two partial problems for dynamical systems:

1. How to choose the non-linear mapping $g(\varphi)$ from the regressor space to the output space (*i.e.*, from R^d to R^p).

2. How to choose the regressors $\varphi(t)$ from past inputs and outputs.

The second problem is the same for all dynamical systems, and it turns out that the most useful choices of regression vectors are to let them contain past inputs and outputs, and possibly also past predicted/simulated outputs. The basic choice is thus

$$\varphi(t) = [y(t-1), \dots, y(t-n), u(t-1), \dots, u(t-m)] \quad (1.5)$$

More sophisticated variants are obtained by letting φ contain also $\hat{y}(t-k|\theta)$. Then (some of) the regressors will also depend on the parameter vector, which lead to so called *recurrent networks* and more complicated algorithms.

In case $n = 0$ in the above expression, so that the regression vector only contains $u(t-k)$ and possibly predicted/simulated outputs (from u) $\hat{y}(t|\theta)$, we talk about *output error models*.

Function Expansions and Basis Functions

The non-linear mapping $g(\varphi, \theta)$ is from R^d to R^p for any given θ . At this point it does not matter how the regression vector φ is constructed. It is just a vector that lives in R^d .

It is natural to think of the parameterized function family as function expansions:

$$g(\varphi, \theta) = \sum \theta(k) g_k(\varphi) \quad (1.6)$$

where g_k are the *basis functions* and the coefficients $\theta(k)$ are the “coordinates” of g in the chosen basis.

Now, the only remaining question is: How to choose the basis functions g_k ? Depending on the support of g_k (*i.e.* the area in R^d for which $g_k(\varphi)$ is (practically) non-zero) we shall distinguish between three types of basis functions: *Global, Ridge-type, and Local*. A typical and classical global basis function expansion would then be the Taylor series, or polynomial expansion, where g_k would contain multinomials in the components of φ of total degree k . Fourier series are also relevant examples. We shall however not discuss global basis functions here any further. Experience has indicated that they are inferior to the semi-local and local ones in typical practical applications.

Local Basis Functions Local basis functions have their support only in some neighborhood of a given point. Think (in the case of $p=1$) of the indicator

function for the unit cube:

$$\kappa(\varphi) = 1 \text{ if } |\varphi_k| \leq 1 \ \forall k, \text{ and } 0 \text{ otherwise} \quad (1.7)$$

By scaling the cube and placing it at different locations we obtain the functions

$$g_k(\varphi) = \kappa(\beta_k(\varphi - \gamma_k)) \quad (1.8)$$

By allowing β to be a matrix we may also reshape the cube to be any parallelepiped. The parameters β are thus *scaling* or *dilation* parameters while γ determine *location* or *translation*.

This choice of g_k in (1.6) gives functions that are piecewise constant over areas in R^d that can be chosen arbitrarily small by proper choice of the scaling parameters, and placed anywhere using the location parameters. Expansions like (1.6) will thus be able to approximate any function by a function that is piecewise constant over arbitrarily small regions in the φ -space. It should be fairly obvious that will allow us to approximate any reasonable function arbitrarily well. It is also reasonable that the same will be true for any other localized function, such as the Gaussian bell function: $\kappa(\varphi) = \exp(-|\varphi|^2)$

Expanding a scalar function the the regressor space In the discussion above, κ was a function from R^d to R . It is quite common to construct such a function from a scalar function $\sigma(x)$ from R to R , by expanding it. Two typical ways to do that are the *radial* and the *ridge* approaches.

Radial Basis Functions In the radial approach we have

$$\kappa(\varphi) = \sigma(\|\varphi\|_\beta^2) \quad (1.9)$$

Here the quadratic norm β will automatically also act as the scaling parameters. β could be a full (positive semi-definite, symmetric) matrix, or a scaled version of the identity matrix.

Ridge-type Basis Functions A useful alternative is to let the basis functions be local in one direction of the φ -space and global in the others. This is achieved quite analogously to (1.9) as follows.

$$g_k(\varphi) = \sigma(\beta_k^T(\varphi - \gamma_k)) \quad (1.10)$$

Here β_k is a d -dimensional vector. Note the difference with (1.8)! The scalar product $\beta_k^T \varphi$ is constant in the subspace of R^d that is perpendicular to the scaling vector β_k . Hence the function $g_k(\varphi)$ varies like σ in a direction parallel to β_k and is constant across this direction. This motivates the term *semi-global* or *ridge-type* for this choice of functions.

Connection to “Named Structures”

Here we briefly review some popular structures, other structures related to interpolation techniques are discussed in [8, 4].

Wavelets The local approach corresponding to (1.6,1.8) has direct connections to wavelet networks and wavelet transforms. The exact relationships are discussed in [8]. Loosely, we note that via the dilation parameters in β_k we can work with different scales simultaneously to pick up both local and not-so-local variations. With appropriate translations and dilations of a single suitably chosen function κ (the “mother wavelet”), we can make the expansion (1.6) orthonormal. The typical choices is to have $\beta_k = 2^k$ and $\gamma_j = j$, and work with doubly indexed expansions in (1.6). This is discussed extensively in [4].

Wavelet and Radial Basis Networks. The choice of the Gaussian bell function as the basic function without any orthogonalization is found in both wavelet networks, [10] and radial basis neural networks [7].

Neural Networks The ridge choice (1.10) with

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

gives a much-used neural network structure, viz. the *one hidden layer feedforward sigmoidal net*.

Hinging Hyperplanes If instead of using the sigmoid σ function we choose “V-shaped” functions (in the form of a higher-dimensional “open book”) Breiman’s *hinging hyperplane* structure is obtained, [2]. Hinging hyperplanes model structures [2] have the form

$$g(x) = \frac{1}{2}[(\beta^+ + \beta^-)x + \gamma^+ + \gamma^-] \pm \frac{1}{2}|(\beta^+ - \beta^-)x + \gamma^+ - \gamma^-|.$$

Thus a hinge is the superposition of a linear map and a semi-global function. Therefore, we consider *hinge* functions as semi-global or ridge-type, though it is not in strict accordance with our definition.

Nearest Neighbors or Interpolation By selecting κ as in (1.7) and the location and scale vector γ_k, β_k in the structure (1.8), such that exactly one observation falls into each “cube”, the nearest neighbor model is obtained: just load the input-output record into a table, and, for a given φ , pick the pair $(\hat{y}, \hat{\varphi})$ for $\hat{\varphi}$ closest to the given φ , $\hat{y}$ is the desired output estimate. If one replaces

(1.7) by a smoother function and allow some overlapping of the basis functions, we get interpolation type techniques such as kernel estimators.

Fuzzy Models Also so called *fuzzy models* based on fuzzy set membership belong to the model structures of the class (1.6). The basis functions g_k then are constructed from the fuzzy set membership functions and the inference rules. The exact relationship is described in [8].

4 ESTIMATING NON-LINEAR BLACK BOX MODELS

The predictor $\hat{y}(t|\theta) = g(\varphi(t), \theta)$ is a well defined function of past data and the parameters θ . The parameters are made up of coordinates in the expansion (1.6), and from location and scale parameters in the different basis functions.

A very general approach to estimate the parameter θ is by minimizing a criterion of fit. This will be described in some detail in Section 5. For Neural Network applications these are also the typical estimation algorithms used, often complemented with *regularization*, which means that a term is added to the criterion (1.11), that penalizes the norm of θ . This will reduce the variance of the model, in that "spurious" parameters are not allowed to take on large, and mostly random values. See e.g. [8].

For wavelet applications it is common to distinguish between those parameters that enter linearly in $\hat{y}(t|\theta)$ (i.e. the coordinates in the function expansion) and those that enter non-linearly (i.e. the location and scale parameters). Often the latter are seeded to fixed values and the coordinates are estimated by the linear least squares method. Basis functions that give a small contribution to the fit (corresponding to non-useful values of the scale and location parameters) can then be trimmed away ("pruning" or "shrinking").

5 GENERAL PARAMETER ESTIMATION TECHNIQUES

In this section we shall deal with issues that are independent of model structure. Principles and algorithms for fitting models to data, as well as the general

properties of the estimated models are all model-structure independent and equally well applicable to, say, linear ARMAX models and Neural Network models.

It suggests itself that the basic least-squares like approach is a natural approach, even when the predictor $\hat{y}(t|\theta)$ is a more general function of θ :

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta, Z^N) \quad (1.11)$$

where

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \|y(t) - \hat{y}(t|\theta)\|^2 \quad (1.12)$$

This procedure is natural and pragmatic – we can think of it as “curve-fitting” between $y(t)$ and $\hat{y}(t|\theta)$. It also has several statistical and information theoretic interpretations. Most importantly, if the noise source in the system is supposed to be a Gaussian sequence of independent random variables $\{e(t)\}$ then (1.11) becomes the Maximum Likelihood estimate (MLE),

It is generally agreed upon that the best way to minimize (1.12) is a *damped Gauss-Newton scheme*, [3]. By this is meant that the estimates iteratively updated as

$$\begin{aligned} \hat{\theta}_N^{(i+1)} &= \hat{\theta}_N^{(i)} - \mu_i \left[\frac{1}{N} \sum_{t=1}^N \psi(t, \hat{\theta}_N^{(i)}) \psi^T(t, \hat{\theta}_N^{(i)}) \right]^{-1} \times \\ &\quad \times \left[\frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t|\hat{\theta}_N^{(i)})) \psi(t, \hat{\theta}_N^{(i)}) \right] \end{aligned} \quad (1.13)$$

where

$$\psi(t, \theta) = \frac{\partial}{\partial \theta} \hat{y}(t|\theta) \quad (1.14)$$

The step size μ is chosen so that the criterion decreases at each iteration. Often a simple search in terms of μ is used for this. If the indicated inverse is ill conditioned, it is customary to add a multiple of the identity matrix to it. This is known as the *Levenberg-Marquard* technique.

It is also quite useful work with a modified criterion

$$W_N(\theta, Z^N) = V_N(Z^N) + \delta \|\theta\|^2 \quad (1.15)$$

with V_N defined by (1.12). This is known as *regularization*. It may be noted that stopping the iterations (i) in (1.13) before the minimum has been reached has the same effect as regularization. See, e.g., [8].

Measured of model fit Some quite general expressions for the expected model fit, that are independent of the model structure, can be developed.

Let us measure the (average) fit between any model (1.3) and the true system as

$$\bar{V}(\theta) = E|y(t) - \hat{y}(t|\theta)|^2 \quad (1.16)$$

Here expectation E is over the data properties (i.e. expectation over “ Z^∞ ” with the notation (1.1)).

Before we continue, let us note the very important aspect that the fit $\bar{V}$ will depend, not only on the model and the true system, *but also on data properties*, like input spectra, possible feedback, etc. We shall say that the fit depends on the *experimental conditions*.

The estimated model parameter $\hat{\theta}_N$ is a random variable, because it is constructed from observed data, that can be described as random variables. To evaluate the model fit, we then take the expectation of $\bar{V}(\hat{\theta}_N)$ with respect to the estimation data. That gives our measure

$$F_N = E\bar{V}(\hat{\theta}_N) \quad (1.17)$$

The rather remarkable fact is that if F_N is evaluated for data with the same properties as those of the estimation data, then, asymptotically in N , (see, e.g., [5], Chapter 16)

$$F_N \approx \bar{V}(\theta^*)\left(1 + \frac{\dim \theta}{N}\right) \quad (1.18)$$

Here θ^* is the value that minimizes the expected value of the criterion (1.12). The notation $\dim \theta$ means the number of estimated parameters. The result also assumes that the model structure is successful in the sense that $\varepsilon(t)$ is approximately white noise.

It is quite important to note that the number $\dim \theta$ in (1.18) will be changed to *the number of eigenvalues of $\bar{V}''(\theta)$ (the Hessian of $\bar{V}$) that are larger than δ* in case the regularized loss function (1.15) is minimized to determine the estimate. We can think of this number as the *efficient number of parameters*. In a sense,

we are “offering” more parameters in the structure, than are actually “used” by the data in the resulting model.

Despite the reservations about the formal validity of (1.18), it carries a most important conceptual message: If a model is evaluated on a data set with the same properties as the estimation data, then *the fit will not depend on the data properties*, and it will depend on the model structure *only in terms of the number of parameters used and of the best fit offered within the structure*.

The expression (1.18) clearly shows the trade off between *variance* and *bias*. The more parameters used by the structure (corresponding to a higher dimension of θ and/or a lower value of the regularization parameter δ) the higher the variance term, but at the same time the lower the fit $\bar{V}(\theta^*)$. The trade off is thus to increase the efficient number of parameters only to that point that the improvement of fit per parameter exceeds $\bar{V}(\theta^*)/N$. This can be achieved by estimating F_N in (1.17) by evaluating the loss function at $\hat{\theta}_N$ for a *validation data set*. It can also be achieved by Akaike (or Akaike-like) procedures, [1], balancing the variance term in (1.18) against the fit improvement.

The expression can be rewritten as follows. Let $\hat{y}_0(t|t-1)$ denote the “true” one step ahead prediction of $y(t)$, and let

$$W(\theta) = E|\hat{y}_0(t|t-1) - \hat{y}(t|\theta)|^2 \quad (1.19)$$

and let

$$\lambda = E|y(t) - \hat{y}_0(t|t-1)|^2 \quad (1.20)$$

Then λ is the *innovations* variance, i.e., that part of $y(t)$ that cannot be predicted from the past. Moreover $W(\theta^*)$ is the *bias error*, i.e. the discrepancy between the true predictor and the best one available in the model structure. Under the same assumptions as above, (1.18) can be rewritten as

$$F_N \approx \lambda + W(\theta^*) + \lambda \frac{\dim \theta}{N} \quad (1.21)$$

The three terms constituting the model error then have the following interpretations

- λ is the unavoidable error, stemming from the fact that the output cannot be exactly predicted, even with perfect system knowledge.
- $W(\theta^*)$ is the bias error. It depends on the model structure, and on the experimental conditions. It will typically decrease as $\dim \theta$ increases.

- The last term is the *variance error*. It is proportional to the (efficient) number of estimated parameters and inversely proportional to the number of data points. It does not depend on the particular model structure or the experimental conditions.

6 CONCLUSIONS

Non-linear black box models for regression in general and system identification in particular has been widely discussed over the past decade. Some approaches like Artificial Neural Networks and (Neuro-)Fuzzy Modeling, and also to some extent the wavelet-based approaches have typically been introduced and described out of their regression-modeling context. This has lead to some confusion about the nature of these ideas.

In this contribution we have stressed that these approaches indeed are “just” corresponding to special choices of model structures in an otherwise well known and classical statistical framework. The well known principle of *parsimony* – to keep the effective number of estimated parameters small – is an important factor in the algorithms. It takes, however, quite different shapes in the various suggested schemes: explicit regularization (a pull towards the origin), implicit regularization (stopping the iterations before the objective function has been minimized), pruning and shrinking (cutting away parameters – terms in the expansion (1.6) – that contribute little to the fit) etc. All of these are measures to eliminate parameters whose contributions to the fit is less than their adverse variance effect.

REFERENCES

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19:716–723, 1974.
- [2] L. Breiman. Hinging hyperplanes for regression, classification and function approximation. *IEEE Trans. Info. Theory*, 39:999–1013, 1993.
- [3] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, 1983.

- [4] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Deylon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box modeling in system identification: Mathematical foundations. *Automatica*, 31, 1995.
- [5] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J., 1987.
- [6] L. Ljung and T. Glad. *Modeling of Dynamic Systems*. Prentice Hall, Englewood Cliffs, 1994.
- [7] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, 78:1481–1497, 1990.
- [8] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31, 1995.
- [9] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall Int., London, 1989.
- [10] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Trans Neural Networks*, 3:889–898, 1992.