MIXED QUANTITATIVE AND QUALITATIVE MODELING: MEANS FOR DEALING WITH SYSTEM UNCERTAINTY

FRANÇOIS E. CELLIER

Department of Electrical and Computer Engineering, The University of Arizona, Tucson, Arizona 85721-0104, U.S.A.

e_mail: Cellier@ECE.Arizona.Edu

URL: http://www.ece.arizona.edu/~cellier

Abstract This paper describes an approach to mixed quantitative and qualitative modeling using fuzzy inductive reasoning. The approach has been verified by means of several large–scale engineering and biomedical applications to produce excellent results in predicting future system behavior.

Keywords: Qualitative modeling; Qualitative simulation; Fuzzy inductive reasoning; Estimation of modeling error.

INTRODUCTION

Models never reflect all facets of reality. Models are always somewhat reductionistic in nature, and consequently, simulation results are never totally accurate. The problem dealt with in this paper is how to cope with the various sources of uncertainty in a model, and how to interpret the simulation results in light of these uncertainties. Somehow, model uncertainties get translated into inaccuracies of the simulation results, but how?

The first aspect to be noticed is that *validity* of a model does not necessarily translate into *accuracy* of simulation results. Although this may seem counterintuitive, the two are often in competition with each other.

A *deductive model*, derived from physical principles, has usually a high degree of validity in the sense that somewhat valid results can be

obtained for a large range of values of its parameters. However, the results may not be very accurate due to the uncertainty of the actual parameter values and due to unmodeled dynamics.

An *inductive model*, derived from observations of the input/output behavior of the real system, may be very accurate when presented with the precise previously observed inputs, yet it usually has a low degree of validity in the sense that its predictions may be totally wrong when the model is driven by other, previously unobserved, input patterns.

Hence there is room for both deductive and inductive modeling in science and engineering. Either technology has its virtues and its deficiencies, and which of the two may be more appropriate in a given situation depends on the application area and the demands made on the model.

In this paper, a methodology is presented that preserves the best of both worlds by enabling the user to mix deductive and inductive models in a single modeling and simulation environment.

A second observation needs to be made. Assessing the inaccuracy of a simulation result is in itself a modeling task. Yet, the same methodology that is used to model the output to be predicted cannot be used to model its error. This would lead to a paradoxical situation. If it indeed were possible to compute, in a deterministic sense, the inaccuracy of a prediction made, then one could simply subtract the predicted prediction error from the prediction itself and obtain the precise value of the output. Evidently, this cannot be done. The modeling error can only be modeled in a statistical sense.

In this paper, a methodology shall be described that assesses the error of a prediction made simultaneously with making the prediction. The two are different facets of one and the same process and are inseparable one from the other. In a robust modeling methodology capable of dealing with model uncertainty, modeling the modeling error should not be an afterthought. Modeling the output and modeling its error should be done simultaneously. A modeling and simulation methodology that does not take the model uncertainty into consideration from the beginning is not robust when dealing with uncertain situations.

INDUCTIVE MODELING

Inductive modeling starts out with one or several sets of observations of input/output behavior of a system, and tries to deduce something about the relationship between these observational patterns. In the ideal situation, a relationship can be postulated that allows to reproduce the observed output patterns when presented with the observed input patterns, and that produces predictions of the output that are not totally incorrect when presented with different, hitherto unobserved, input patterns.

Inductive models come in two shades. *Parametric models* make an assumption about the structure of the relationship, and then optimize a set of unknown parameter values to obtain an optimal curve fit between the observed and predicted output trajectories. *Non–parametric models* do not make any assumption about the underlying model structure, and restrict themselves to intelligently characterize and catalog previously observed input/output patterns for future reference and interpolation.

A typical example of a parametric model is a neural network. Neural networks assume a very rich structure, in fact, a structure that is so rich that it can represent basically any behavioral pattern, and then optimize a large set of parameters (the weights) to fit the observed input/output patterns (supervised training).

There are three drawbacks to such an approach.

1. By assuming a structure in advance, it becomes very difficult to afterwards assess the error of that assumption itself. However, in a neural network, the assumed structure is so rich that the structural error is usually negligible.

2. Most parametric models are deterministic in nature. By embracing a totally deterministic modeling technique, it becomes difficult to assess the error of predictions made, because, as was explained before, this error cannot be estimated in a deterministic sense. In fact, this is hardly ever tried. For example, very few neural networks make any attempt at estimating the inaccuracy of their predictions. A neural network will predict *anything* for *any* input pattern it is presented with, irrespective of how unlikely the correctness of that prediction may be. This is a serious drawback of

all such modeling methodologies.

3. Inductive modeling is almost synonymous with optimization. The more parameters are to be optimized, the slower will the optimization be. Training a neural network by optimizing its weights in a supervised training mode is notoriously slow.

Non–parametric models may present an answer to these objections, as shall be shown. However, also they have their drawbacks. In particular, they offer no extrapolation capability whatsoever. A non–parametric model can only reproduce what it has been shown earlier in terms of input/output patterns. It cannot extrapolate beyond the range of its previous experiences. However, the extrapolation power of parametric models lies precisely in the structural assumption made, i.e., the more incorrect that structural assumption is, the more likely it will be that the extrapolated predictions are wrong. The enhanced validity of a parametric model hinges upon a correct guess of the underlying model structure. A neural network doesn't make any attempt at guessing the correct structure, and therefore, its extrapolation capability may, in fact, be a rather dubious virtue. Since the neural network never fails to make a prediction, the user is seduced into believing that the model is very robust, whereas, in reality, the only thing "robust" in this entire enterprise may be the gullibility of the user.

FUZZY INDUCTIVE REASONING

Fuzzy inductive reasoning is a qualitative modeling and simulation technique consisting of a non–parametric inductive modeling step followed by a deductive simulation step. The methodology is described in due course.

Quantitative real–valued variables are *fuzzified* into qualitative triples consisting of a class value, a fuzzy membership value and a side value. The technique is illustrated by means of the example shown in Fig.1.

The ambient temperature is classified into the five classes 'cold,' 'fresh,' 'moderate,' 'warm,' and 'hot,' using, in the shown example, popular knowledge to determine the so–called *landmarks*, i.e., the borders between neighboring classes. A quantitative value of $temperature =$

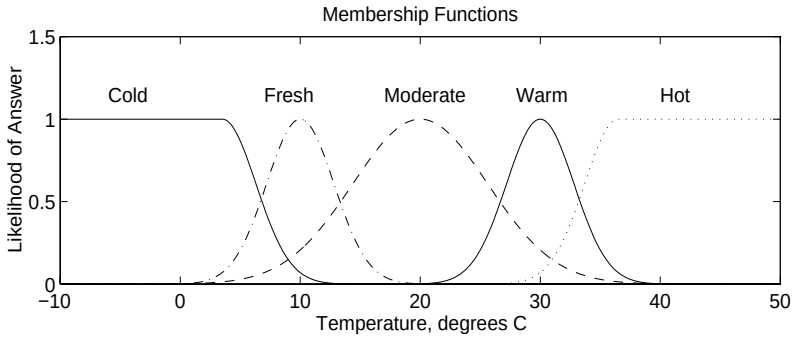

Figure 1: Fuzzy Recoding of Ambient Temperature

18.0 would in this case be recoded into a class value of 'moderate,' a membership value of 0.938, which is the highest value of all the bell–shaped membership functions associated with the quantitative value 18.0, and a side value of 'left,' since 18.0 lies to the left of the maximum of the corresponding membership function. Evidently, the qualitative triple contains exactly the same information as the original quantitative value. The original quantitative value can be regenerated whenever needed in the reverse operation of fuzzification, the so–called *defuzzification.*

Fuzzification of real–valued variables is done for three reasons:

1. It speeds up the optimization dramatically. Let a relationship between n inputs and one output be given. Rather than searching through a n–dimensional continuous search space to find the optimal input/output pattern, the search is limited to the very coarse n–dimensional discrete search space of class values. In this way, class values are used for determining the neighborhood of the optimal solution, whereas the fuzzy membership information is then used for interpolation in the vicinity of the optimal solution.

2. The optimization in the discrete space of the class values is deterministic. However, the subsequent interpolation in the continuous space of fuzzy membership values is stochastic. This approach is better capable of coping with model uncertainty than a purely deterministic approach.

3. An almost identical technique to the one that is used to predict, in a statistical sense, the fuzzy membership value of the output, can

also be used to assess, again in a statistical sense, the accuracy of the prediction made.

Once the real–valued variables have been fuzzified, qualitative modeling proceeds in two stages. In the first stage, it is determined, which input variables characterize best the input/output relationship. Given a set of input variables and a set of output variables of a system sampled at usually equidistant points in time. The value of a particular output variable, y_i, at time point t is to be determined in function of the inputs at the same time point, and both inputs and outputs at past sampling points, e.g.

$$y_1(t) = \tilde{\mathrm{f}}(y_3(t-2\delta t), u_2(t-\delta t), y_1(t-\delta t), u_1(t)) \tag{1}$$

where $\tilde{\mathrm{f}}$ denotes a qualitative relationship. Notice that $\tilde{\mathrm{f}}$ does not stand for any (known or unknown) explicit formula relating the input arguments to the output argument, but only represents a generic causality relationship that, in the case of the inductive reasoning methodology, will be encoded in the form of a tabulation of likely input/output patterns, i.e., a state transition table. In SAPS–II (the currently used implementation of the methodology), Eq.(1) is represented by the following matrix:

$$\begin{array}{c|ccccc} {}_t\backslash^x & u_1 & u_2 & y_1 & y_2 & y_3 \\ t-2\delta t & 0 & 0 & 0 & 0 & -1 \\ t-\delta t & 0 & -2 & -3 & 0 & 0 \\ t & -4 & 0 & +1 & 0 & 0 \end{array} \tag{2}$$

The negative elements in this matrix are referred to as m–inputs. m–inputs denote input arguments of the qualitative functional relationship. They can be either inputs or outputs of the subsystem to be modeled, and they can have different time stamps. The above example contains four m–inputs. The sequence in which they are enumerated is immaterial. They are usually enumerated from left to right and top to bottom. The single positive value denotes the m–output. The terms m–input and m–output are used in order to avoid a potential confusion with the inputs and outputs of the plant. In the above example, the

first m–input corresponds to the output variable y_3 two sampling intervals back, $y_3(t - 2\delta t)$, whereas the second m–input refers to the input variable u_2 one sampling interval into the past, $u_2(t - \delta t)$, etc. Notice that the Fuzzy Inductive Reasoning (FIR) methodology restricts itself to multi–input/single–output (MISO) systems, since multiple outputs (MIMO systems) can always be reduced to sets of MISO systems. Consequently, every FIR model has exactly one m–output.

In fuzzy inductive reasoning, such a representation is called a *mask*. A mask denotes a dynamic relationship among qualitative variables. A mask has as many columns as there are plant variables to look at, and it has a certain number of rows, the *depth* of the mask.

The optimal mask is the one abstraction that optimizes the predictive power of the model. The problem of finding the optimal mask relates to the struggle between generality and specificity. If more m–inputs are added to the mask, the observed patterns become more and more specific. Yet, chances are that a newly observed input pattern has never been seen before, making a prediction impossible. Removing m–inputs from the mask leads to bolder, less specific, patterns that are likely to be ambiguous. The so obtained model no longer represents the true dynamics of the system, leading to non–deterministic input/output behavior, i.e., to ambiguities in the predictions made. In SAPS–II, the Shannon entropy measure is used to determine the specificity of a given input/output relationship, whereas the observation ratio is used to determine its generality. The optimal mask is a compromise between the two measures. Details can be found in (Cellier *et al.*, 1996).

In the second stage, the optimal mask is applied to the observed input/output trajectories to obtain a finite state machine representation of corresponding input/output patterns. The resulting finite state machine represents the non–parametric inductive model of the system under study.

Once the qualitative model has been found, the qualitative simulation (forecasting of future behavior) again proceeds in two stages. In a first (deterministic) stage, future input patterns are compared with those stored in the experience data base (the finite state machine), and the most likely output is read out of the data base. In this stage, both the class and side values are used to determine the most similar pattern.

In a second (statistical) stage, a prediction needs to be made of the most likely fuzzy membership value of the output. The fuzzy membership value is a distance weighted average of the five nearest neighbors in the experience data base, whereby the distance function is computed in the input space, and the interpolation is done in the output space. Mathematically, this is done in the following way.

Since FIR deals with multi–input/single–output (MISO) systems exclusively, each state consists of a number of input variables and a single output variable. The first problem to be considered is one of normalization. Since the different input variables can represent arbitrary physical or other quantities, their absolute values can be vastly different one from another. In order to create a meaningful metric of proximity in the input space, it is necessary to normalize the input variables. This is accomplished using a pseudo–regeneration of the fuzzified input variables:

$$pos_i = class_i + side_i \cdot (1.0 - Memb_i) \tag{3}$$

where the *class* values are assumed to be integers starting from '1' representing the lowest class, and the *side* values are also integers assuming the values '-1' representing the logical value '*left*,' 0 representing the value '*center*,' and '$+1$' representing the value '*right*.' The index i represents the i^{th} input variable in the input state of the current observation. The position value, pos_i can be viewed as a normalized pseudo–regeneration of the i^{th} input variable. Irrespective of the original values of the input variable, pos_i assumes values in the range $[1.0, 1.5]$ for the lowest class, $[1.5, 2.5]$ for the next higher class, etc.

Similarly,

$$pos_{ij} = class_{ij} + side_{ij} \cdot (1.0 - Memb_{ij}) \tag{4}$$

represents the normalized pseudo–regeneration of the i^{th} input variable of the j^{th} nearest neighbor in the experience data base.

$$\mathbf{pos} = [\ pos_1,\ pos_2,\ \ldots,\ pos_n\ ] \tag{5}$$

is the position vector representing the current input state, assuming that the system to be modeled contains n m–inputs, and

$$\mathbf{pos_j} = [\ pos_{1j},\ pos_{2j},\ \ldots,\ pos_{nj}\ ] \tag{6}$$

represents the corresponding position vector of the j^{th} nearest neighbor.

The distance between the current input state and its j^{th} nearest neighbor is computed as:

$$dis_j = \|\mathbf{pos} - \mathbf{pos_j}\| \tag{7}$$

It is necessary to avoid distance values of 0.0:

$$d_j = \max(dis_j, \epsilon) \tag{8}$$

where ϵ is the smallest number that can be distinguished from 1.0 in addition.

$$s_d = \sum_{j=1}^{5} d_j \tag{9}$$

is the sum of the distances of the five nearest neighbors, and:

$$d_{\text{rel}_j} = \frac{d_j}{s_d} \tag{10}$$

are the relative distances. By applying this algorithm either to the entire experience data base or a suitable subset thereof, the five nearest neighbors can be determined while simultaneously computing their distance function.

The interpolation is done in the output space. Absolute weights are computed as:

$$w_{\text{abs}_j} = \frac{1.0}{d_{\text{rel}_j}} \tag{11}$$

and

$$s_w = \sum_{j=1}^{5} w_{\mathrm{abs}_j} \tag{12}$$

is the sum of the absolute weights. Hence the relative weights can be computed as:

$$w_{\mathrm{rel}_j} = \frac{w_{\mathrm{abs}_j}}{s_w} \tag{13}$$

Using this information, the membership value of the predicted output is determined as:

$$Memb_{\mathrm{out}} = \sum_{j=1}^{5} w_{\mathrm{rel}_j} \cdot Memb_{\mathrm{out}_j} \tag{14}$$

If one of the observations in the experience data base coincides, by chance, with the new observation, its relative distance value will be very close to 0.0, whereas that of the other four neighbors will be considerably larger. Consequently, only this data record will have an influence on determining the membership value of the output. On the other hand, if the five nearest neighbors are all approximately equally far away from the new observation, the relative distance values will all be approximately 0.2, and each of the corresponding records in the experience data base will have equal weight in determining the membership value of the new output.

This approach, although quite heuristic in nature, has been very successful at predicting, rather accurately, a significant set of variables from different application domains, including biomedicine (Nebot *et al.*, 1996), nuclear reactors (de Albornoz and Cellier, 1993), aircrafts (de Albornoz and Cellier, 1994), and switching circuits (de Albornoz *et al.*, 1994).

The idea behind assessing the accuracy of a prediction made is straightforward. It is directly related to the distance of the observations in the experience data base to the new observation. If the new observation is very close to one or several previous observations, then

the confidence in the prediction made is high. On the other hand, if the new observation is quite far away from its five nearest neighbors and a lot of interpolation has to be done to determine the new output value, then the prediction should be assessed as less accurate.

The average distance used to determine the confidence measure is computed as a weighted sum of the relative distances of the five nearest neighbors:

$$d_{\text{conf}} = \sum_{j=1}^{5} w_{\text{rel}_j} \cdot d_j \tag{15}$$

The largest possible value of the average distance can be calculated as:

$$d_{\text{conf}_{\max}} = \sqrt{\sum_{i=1}^{n} (n_{\text{cl}_i} - 1)^2} \tag{16}$$

where n_{cl_i} is the number of classes used in the fuzzification of the i^{th} input variable.

Finally, the confidence is evaluated as:

$$conf = 1.0 - \frac{d_{\text{conf}}}{d_{\text{conf}_{\max}}} \tag{17}$$

$conf$ is a *quality measure*, i.e., a real–valued number in the range $[0.0, 1.0]$. Values close to 1.0 denote a high confidence in the prediction made, whereas values close to 0.0 denote a very low confidence in the forecast. In the presentation, examples shall be presented of how this information is being used in practice.

MIXED QUANTITATIVE AND QUALITATIVE MODELING AND SIMULATION

Once a qualitative triple of an output variable has been predicted, it can be converted back to a quantitative signal using defuzzification.

Once this has been done, the so converted signal is conceptually no different from a quantitatively simulated variable, and can be used as an input to any quantitative model.

Hence fuzzification, in a mixed quantitative and qualitative model, assumes the role of an A/D converter in a sampled data model, whereas defuzzification assumes that of a D/A converter. However, this is where the similarity ends. Whereas sampled data models use deterministic reasoning about a large number of crisp levels, FIR models use fuzzy reasoning about a very small number of coarse, yet fuzzy, levels. The maybe most convincing application of FIR to mixed quantitative and qualitative modeling found so far is a model of the cardiovascular system, whereby the hemodynamics are described by a fairly large quantitative ODE model, whereas the central nervous control functions are described by five qualitative FIR models operating in parallel. This application was described in (Nebot, 1994, 1996).

REFERENCES

Cellier, F.E., A. Nebot, F. Mugica and A. de Albornoz (1996), "Combined Qualitative/Quantitative Simulation Models of Continuous–Time Processes Using Fuzzy Inductive Reasoning Techniques," *Intl. J. General Systems*, **24**(1–2), pp. 95–116.

de Albornoz, A., and F.E. Cellier (1993), "Variable Selection and Sensor Fusion in Automatic Hierarchical Fault Monitoring of Large Scale Systems," *Proc. QUARDET'93, Qualitative Reasoning and Decision Technologies*, Barcelona, Spain, pp. 722–734.

de Albornoz, A., and F.E. Cellier (1994), "Building Intelligence Into an Autopilot — Using Qualitative Simulation to Support Global Decision Making," *Simulation*, **62**(6), pp. 354–364.

de Albornoz, A., J. Sardá, and F.E. Cellier (1994), "Structure Identification in Variable Structure Systems by Means of Qualitative Simulation," *Proc. ESM'94, European Simulation MultiConference*, Barcelona, Spain, pp. 486–491.

Nebot, A. (1994), *Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning*, Ph.D. dissertation, Llen-

guatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.

Nebot, A. (1996), *Qualitative Modeling of Biomedical Systems*, Springer–Verlag, New York, Series in Biomathematics, accepted for publication.

Nebot, A., F.E. Cellier, and D.A. Linkens (1996), "Synthesis of an Anaesthetic Agent Administration System Using Fuzzy Inductive Reasoning," *AI in Medicine*, accepted for publication.