A Semiquantitative Approach to Study Semiqualitative Systems

Juan Antonio Ortega, Rafael M. Gasca, Miguel Toro, and Jesús Torres

Departamento de Lenguajes y Sistemas Informáticos
University of Seville
Avda. Reina Mercedes s/n – 41012 – Sevilla (Spain)
{ortega,gasca,mtoro,jtorres}@lsi.us.es

Abstract. In this paper is proposed a semiquantitative methodology to study models of dynamic systems with qualitative and quantitative knowledge. This qualitative information may be composed by: operators, envelope functions, qualitative labels and qualitative continuous functions. A formalism is also described to incorporate this qualitative knowledge into these models. The methodology allows us to study all the states (transient and stationary) of a semiquantitative dynamic system. It also helps to obtain its behaviours patterns. The methodology is applied to a logistic growth model with a delay.

1 Introduction

Models of dynamic systems studied in science and engineering are normally composed of quantitative, qualitative, and semiquantitative knowledge. Different approximations have been proposed when the qualitative knowledge is taken into account: transformation of non-linear to piecewise linear relationships, Monte Carlo method, constraint logic programming, probability distributions, causal relations, fuzzy sets, and combination of all levels of qualitative and quantitative abstraction [5], [9].

We are interested in the study of dynamic systems with quantitative and qualitative knowledge. All this knowledge should be taken into account when these models are studied. Different levels of numeric abstraction have been proposed in the literature: purely qualitative [6], semiquantitative [5] [8], numeric interval [14] and quantitative. The proposed methodology transforms a semiquantitative model into a family of quantitative models. A semiquantitative model may be composed of qualitative knowledge, arithmetic and relational operators, predefined functions (log,exp,sin,...), numbers and intervals.

A brief description of the proposed methodology is as follows: a semiquantitative model is transformed into a set of quantitative models. The simulation of every quantitative model generates a trajectory in the phase space. A database is obtained with these quantitative behaviours or trajectories. Techniques of Knowledge Discovery in Databases (KDD) are applied by means of a language to carry out queries about the qualitative properties of this time-series database. This language is also intended to classify the different qualitative behaviours of our model. This classification will help us to describe the semiquantitative behaviour of a system by means of a set of hierarchical rules obtained by means of machine learning algorithms.

The term KDD [1] is used to refer to the overall process of discovering useful knowledge from data. The problem of knowledge extraction from databases involves many steps, ranging from data manipulation and retrieval to fundamental mathematical and statistical inference, search and reasoning. Although the problem of extracting knowledge from data (or observations) is not new, automation in the context of databases opens up many new unsolved problems.

KDD has evolved, and continues to evolve, from the confluence of research in such fields as databases, machine learning, pattern recognition, artificial intelligence and reasoning with uncertainty, knowledge acquisition for expert systems, data visualization, software discovery, information retrieval, and high-performance computing. KDD software systems incorporate theories, algorithms, and methods from all of these fields.

The term data mining is used most by statisticians, database researchers and more recently by the business community. Data mining is a particular step in the KDD process. The additional steps in KDD process are data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge and proper interpretation of the results of mining ensure the useful knowledge is derived from the data [11]. A detailed descriptions of these steps may be found in [10].

The originality of our approach is that it combines in a proper way qualitative reasoning with machine learning techniques. This approach is appropriate to study all the states (transient and stationary) of a semiquantitative dynamic system. It also appropriated to obtain its behaviours patterns. However, some behaviours maybe not found with this approach, mainly, those behaviours obtained with narrowed domains of the parameters.

2 The Methodology

There has been a great deal of previous research studying the stationary state of a system, however, it is also necessary to study transient states. For example, it is very important in production industrial systems to improve their efficiency. Both states of a semiquantitative dynamic system may be studied with the proposed methodology. The methodology is shown in Figure 1.

Starting from a dynamic system with qualitative knowledge, a semiquantitative model S is obtained. A family of quantitative models F is obtained from S by means of the application of some transformation techniques which are bellow described.

Stochastic techniques are applied to choose a model $M \in F$. Every model M is quantitatively simulated obtaining a trajectory, which is composed by the values of all variables from its initial value until its final value, and the values of the parameters. Therefore, it contains the values of these variables in the transient and stationary states of the system.

A database of quantitative trajectories T is obtained with these quantitative behaviours. A language is proposed to carry out queries about the qualitative properties of the set of trajectories included in the database. A labelled database is obtained with the classification of these trajectories in according to a criterion.

Qualitative behaviour patterns of the system may be automatically obtained from this database by applying machine learning based on genetic algorithms. These algorithms are described in [2].

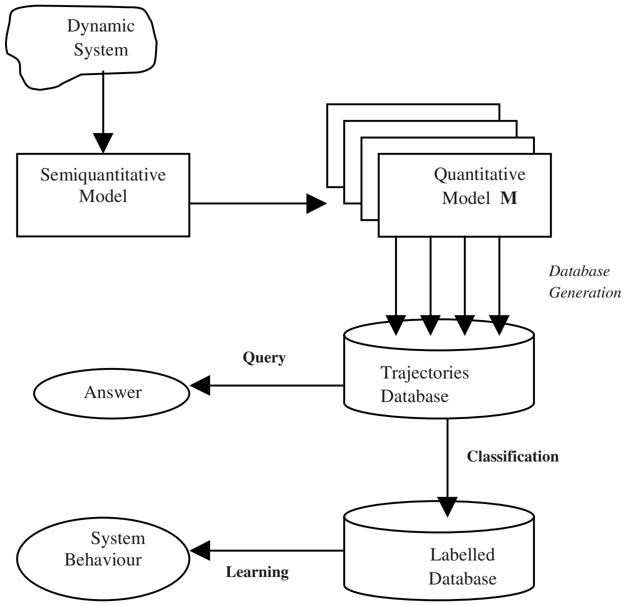

Fig. 1. Proposed methodology

3 Semiquantitative Models

A semiquantitative model S is represented by

$$\Phi(dx/dt,x,q,t),\ \ x(t_0) = x_0,\ \ \Phi_0(q,x_0) \tag{1}$$

being $x \in \Re^n$ the set of state variables of the system, q the parameters, t the time, dx/dt the variation of the state variables with the time, Φ constraints depending on $dx/dt,x,q,t$ and Φ_0 the set of constraints with initial conditions.

If the methodology is applied, the equations of the dynamic system (1) are transformed into a set of constraints among variables, parameters and intervals. In this paper, we are interested in those systems that may be expressed as (2) when the transformation rules are applied

$$dx/dt = f(x,p,t),\ \ x(t_0) = x_0,\ \ p \in I_p,\ \ x_0 \in I_0 \tag{2}$$

where p includes the parameters of the system and new parameters obtained by means of the transformation rules, f is a function obtained by applying the transformation rules, and I_p,I_0 are real intervals. The equation (2) is a family F of dynamic systems depending on p and x_0.

3.1 Qualitative Knowledge

Our attention is focused on those dynamic systems where there may be qualitative knowledge in their parameters, initial conditions and/or vector field. They constitute the semiquantitative differential equations of the system.

The representation of the qualitative knowledge is carried out by means of operators, which have associated real intervals. This representation facilitates the integration of qualitative and quantitative knowledge in a simple way, and the incorporation of knowledge from the experts [4].

Qualitative knowledge may be composed of qualitative operators, qualitative labels, envelope functions and qualitative continuous functions. This qualitative knowledge and its transformation techniques are now detailed.

Qualitative Operators

These operators are used to represent qualitative parameters and initial conditions. They may be unary *U* and binary *B* operators. Every qualitative operator *op* is defined by means of an interval I_{op}, which is supplied by the experts.

Each qualitative magnitude of the system has its own unary operators. Let U_x be the unary operators for a qualitative variable *x*, i.e. $U_x=\{VN_x, MN_x, LN_x, AP0_x, LP_x, MP_x, VP_x\}$. They denote for *x* the qualitative labels *very negative, moderately negative, slightly negative, approximately zero, slightly positive, moderately positive, very positive* respectively. Let *r* be a new generated variable and let I_u be an interval defined in accordance with [13], then the transformation rule for a unary operator is as follows

$$\mathrm{op_u(e)} \equiv \{\mathrm{r} \in \mathrm{I_u}\ ,\ \mathrm{e} - \mathrm{r} = 0\} \tag{3}$$

Let e_1,e_2 be two arithmetic expressions, and let op_b be a binary operator. The expression $op_b(e_1,e_2)$ denotes a qualitative relationship between e_1 and e_2. Binary qualitative operators are classified into:

- Operators related to the difference $\geq, =, \leq$, being their transformation rules:

$$\begin{aligned} \mathrm{e_1 = e_2} &\equiv \{\ \mathrm{e_1 - e_2 = 0}\ \} \\ \mathrm{e_1 \leq e_2} &\equiv \{\ \mathrm{e_1 - e_2 - r = 0},\ \mathrm{r} \in [-\infty, 0]\ \} \\ \mathrm{e_1 \geq e_2} &\equiv \{\ \mathrm{e_1 - e_2 - r = 0},\ \mathrm{r} \in [0, +\infty]\ \} \end{aligned} \tag{4}$$

- Operators related to the quotient {«,–<,~,≈,»,Vo, Ne,...}. The following transformation rule is applied:

$$\mathrm{op_b(e_1,e_2)} \equiv \{\mathrm{e_1 - e_2 * r = 0}\ ,\ \mathrm{r} \in \mathrm{I_b} \tag{5}$$

where I_b is an interval defined according to [7].

In order to maintain the consistency of the model, it is necessary to add constraints to guarantee the relation among the absolute and relative order of magnitude operators. in the general case [12].

Envelope Functions

An envelope function $y=g(x)$ represents the family of functions included between two defined real functions, a upper one $U{:}\ \Re \Rightarrow \Re$ and a lower one $L{:}\ \Re \Rightarrow \Re$.

$$\langle L(x), U(x), I \rangle, \qquad \forall x \in I{:}\ L(x) \leq U(x) \tag{6}$$

where I is the definition domain of g, and x is the independent. The transformation rule applied to (6) is

$$g(x) = \alpha L(x) + (1 - \alpha)\, U(x) \text{ with } \alpha \in [0,1] \tag{7}$$

where α is a new variable. If $\alpha=0 \Rightarrow g(x)=U(x)$ and if $\alpha=1 \Rightarrow g(x)=L(x)$ and any other value of α in (0,1) stands for any included value between $L(x)$ and $U(x)$.

Qualitative Continuous Functions

A qualitative continuous function $y=h(x)$ represents a set of constraints among the values of y and x according to the properties of h. It is denoted by

$$y=h(x), \quad h \equiv \{P_1, s_1, P_2, \ldots, s_{k-1}, P_k\} \tag{8}$$

being P_i the points of the function. Every P_i is defined by means of (d_i,e_i) where d_i is the qualitative landmark associated to the variable x and e_i to y. These points are separated by the sign s_i of the derivative in the interval between two consecutive points. A monotonous qualitative function is a particular case of these functions where the sign is always the same $s_1=\ldots=s_{k-1}$.

The transformation rules of a qualitative continuous function are applied in three steps:

1. *Normalization*:

   The definition of the function is completed and homogenised using these continuity properties:

   - □ a function that changes, its sign between two consecutive landmarks passes through a landmark whose value in the function is zero
   - □ a function whose derivative changes, its sign between two consecutive landmarks passes through a landmark whose derivative is zero

   The definition of any function (Equation 8) is always completed with: the extreme points ($-\infty$, $+\infty$), the points that denote the cut points with the axes, and where the sign of the derivative changes (a maximum or a minimum of h).

2. *Extension*:

   The definition of these functions is enriched by means of an automatic process, which incorporates new landmarks or qualitative labels. This extension is carried out to diminish the uncertainty in the definition of the function.

   The number of new landmarks included between each two consecutive original landmarks may be always the same. With this consideration, we don't loose the statistical representativity of the selected quantitative samples obtained for this function.

3. *Transformation*

A qualitative function h is transformed into a set of quantitative functions H.
The algorithm **`ChooseH`** is applied to obtain H.

```
ChooseH (h)
   for each monotonous region in h
         segment={Pm,...,Pn}
         choose a value for every Pi in the segment
                verifying the constraints of h
```

This algorithm divides *h* into its segments. A *segment* is a sequence of consecutive points $\{P_m,...,P_n\}$ separated by means of those points whose landmark $e_i=0$ or where $s_i \neq s_{i+1}$. The segments divide the function into their monotonous regions where their landmarks e_i have the same sign. The algorithm applies stochastic techniques to choose every quantitative function of *H*. These techniques are similar to the Monte Carlo method, however, the values obtained must satisfy the constraints of *h*. We use a heuristic that applies a random uniform distribution to obtain the values for every landmark of P_i.

4 Database Generation

A family *F* of quantitative models has been obtained when the transformation rules described in section 3.1 have been applied to the semiquantitative model *S*. This family depends on a set of interval parameters *p* and functions *H* defined by means of a set of quantitative points. Every particular model *M* of *F* is selected by means of stochastic techniques, and it is quantitatively simulated. This simulation generates a trajectory *r* that is stored into the database *T*.

The following algorithms are applied to obtain *T*.

```
ChooseModel (F)
    for each interval parameter or variable of F
           choose a value in its interval for it
    for each function h of F
           H:=ChooseH(h)

Database generation T
     T:={ }
     for i=1 to N
           M:=ChooseModel(F)
           r:= QuantitativeSimulation(M)
           T:=T ∪ r
```

being *N* the number of simulations to be carried out, and it is defined in accordance with the section *7*. Therefore, *N* is the number of trajectories of *T*.

5 Query/Classification Language

In this section, we propose a language to carry out queries to the trajectories database. It is also possible to assign qualitative labels to the trajectories with this language.

5.1 Abstract Syntax

Let T be the set of all trajectories r stored in the database. A query Q is: a quantifier operator $\forall, \exists, \aleph$ applied on T, or a basic query *[r,P]* that evaluates *true* when the trajectory r verifies the property P.

The property P may be formulated by means of the composition of other properties using the Boolean operators $\wedge, \vee, \neg$.

Table 1. Abstract Syntax of the Language

$Q : \forall\ r \in T \bullet [r,P]$	P: P_b	P_b: P_d
$\mid \exists\ r \in T \bullet [r,P]$	$\mid P \wedge P$	$\mid\ f(L(F))$
$\mid \aleph\ r \in T \bullet [r,P]$	$\mid P \vee P$	$\mid\ \forall t{:}F \bullet F$
$\mid\ [r,p]$	$\mid \neg P$	$\mid\ \exists t{:}F \bullet F$
P_d: EQ	F: F_b	F_b: e_b
$\mid\ CL$	$\mid F \,\&\, F$	$\mid e \in I$
	$\mid F \mid F$	$\mid u(e)$
	$\mid\ !\,F$	$\mid b(e,e)$

A basic property P_b may be: a predefined property P_d, a Boolean function f applied to a list L of points or intervals that verifies the formula F, or a quantifier $\forall, \exists$ applied to the values of a particular trajectory for a time t. This time may be: an instant of time, a unary time operator (i.e. a range of time), a predefined time landmark, or the list of times where the formula F is verified.

A defined property P_d is the one whose formulation is automatic. They are queries commonly used in dynamic systems. There are two predefined: *EQ* is verified when the trajectory ends up in a stable equilibrium; and *CL* when it ends up in a cycle limit.

A formula F may be composed of other formulas combined by means of Boolean operators &,|,!.

A basic formula F_b may be: a Boolean expression e_b, or if a numeric expression e belongs to an interval, or a unary u or binary b qualitative operator.

5.2 Semantics

The semantics of every instruction of this language is translated into a query on the database. The techniques applied to carry out this transformation come from the development of compilers of language programming. A query *[r,P]* is *true* when trajectory r verifies the property P. Semantics of a query with a quantifier depends on its related quantifier. If it is $\forall$, a Boolean value *true* is returned when all the trajectories $r \in T$ verify P. If it is $\exists$ then *true* is returned when there is at least one trajectory $r \in T$ that verifies the property P. If the quantifier is $\aleph$ then returns the number of trajectories of T that verifies P.

Let $\forall\ t$: $F_1 \bullet F_2$ be a basic property which is *true* if during the time that F_1 is satisfied, all the values of r verify F_2. For $\exists$ quantifier is *true* when at least a value of r that satisfies F_1, also satisfied F_2. In order to evaluate a formula F, it is necessary to substitute its variables for their values. These values are obtained from T.

5.3 Classification

A classification rule is formulated as a set of basic queries with labels, and possibly other expressions

$$[\mathrm{r,P_A}] \Rightarrow \mathrm{A,e_{n1},...} \qquad [\mathrm{r,P_b}] \Rightarrow \mathrm{B,e_{n2},...} \qquad ... \tag{9}$$

A trajectory r is classified with a label η if it verifies the property P_η.

Let $[r,P_A] \Rightarrow A,e_{A1}$ be a classification rule. A trajectory $r \in T$ is classified with the label A if it verifies property P_A. The result of evaluating e_{A1} for this trajectory is also stored into the database.

6 A Logistic Growth Model with a Delay

It is very common to find growth processes where an initial phase of exponential growth is followed by another phase of asymptotic approach to a saturation value. The following generic names are given: logistic, sigmoid, and s-shaped processes. This growth appears in those systems where the exponential expansion is truncated by the limitation of the resources required for this growth. They abound in the evolution of bacteria, in mineral extraction, in world population growth, in epidemics, in rumours, in economic development, the learning curves, etc.

In the bibliography, these models have been profusely studied. There is a bimodal behaviour pattern attractor: A stands for normal growth, and O for decay (Figure 5.b).

Differential equations of the model S are

$$\Phi \equiv \begin{cases} \mathrm{dx/dt=x(n\ r\text{-}m),\ y=delay_\tau(x),\ r>0,r=h(y),} \\ \mathrm{h} \equiv \{(-\infty,-\infty),+,(\mathrm{d_0},0),+,\ (0,1),+,\ (\mathrm{d_1,e_0}),\ -,(1,0),\ -\ (+\infty,-\infty)\} \end{cases} \tag{10}$$

being n the increasing factor, m the decreasing factor, and h a qualitative function with a maximum point at (x_1,y_0). The initial conditions are

$$\Phi_0 \equiv \{\ x_0 \in [LP_x,MP_x],\ LP_x(m),LP_x(n),\ \tau \in MP_\tau,VP_\tau\} \tag{11}$$

where LP,MP,VP are qualitative unary operators for x, τ variables.

We would like to know:

1. if an equilibrium is always reached
2. if there is an equilibrium whose value is not zero
3. if all the trajectories with value zero at the equilibrium are reached without oscillations.
4. To classify the database according to the behaviours of the system.

We apply our approach to this model. Firstly, the transformation rules are applied,

$$\Phi\begin{cases} dx/dt=x(n\,r-m),\ y=delay_{\tau}(x),\ x>0,\ r=H(y), \\ H,\ x_0\in[0,3],\ \ m,n\in[0,1],\ \tau\in[0.5,10] \end{cases} \qquad (12)$$

where *H* has been obtained by applying *Choose H* to *h*, and the intervals are defined in accordance with the experts' knowledge. The algorithm *Database generation T* returns the trajectories database.

The proposed queries are formulated as follows:

1. $r\in T \bullet [r,EQ]$
2. $r\in T \bullet [r, EQ \wedge \exists\ t\!: t\approx t_f \bullet\ !AP0_x(x)]$
3. $\forall\ r\in T \bullet [r, EQ \wedge \exists\ t\!: t\approx t_f \bullet\ AP0_x(x) \wedge\ \ length(dx/dt=0)=0\ ]$
4. being $AP0_x$ a unary operator of x. The list of points where dx/dt=0 is the list with the maximum and minimum points. If length is 0 then there are not oscillations.

   We classify the database by means of the labels:

   $[r,EQ \wedge length(dx/dt=0)>0 \wedge\ \exists t: t\approx t_f \bullet !AP0_x(x)] \Rightarrow recovered$,

   $[r,EQ \wedge length(dx/dt=0)>0 \wedge\ \ \exists t: t\approx t_f \bullet AP0_x(x)] \Rightarrow retarded$,

   $[r,EQ \wedge \exists t: t\approx t_f \bullet AP0_x(x)] \qquad\qquad \Rightarrow extinction$,

They correspond to the three possible behaviour patterns of the system (Fig. 6). They are in accordance with the obtained behaviours when a mathematical reasoning is carried out [3].

7 Conclusions and Further Work

In this paper, a methodology is presented in order to automate the analysis of dynamic systems with qualitative and quantitative knowledge. This methodology is based on a transformation process, application of stochastic techniques, quantitative simulation, generation of trajectories database and definition of a query/classification language. There is enough bibliography that studies stationary states of dynamic systems. However, the study of transient states is also necessary. These studies are possible with the proposed language.

The simulation is carried out by means of stochastic techniques. The results are stored in a quantitative database. It may be classified by means of the proposed language. Once the database is classified, genetic algorithms may be applied to obtain conclusions about the dynamic system.

In the future, we are going to enrich the query/classification language with: operators for comparing trajectories among them, temporal logic among several times of a trajectory, more type of equations, etc.

Acknowledgments. This work was partially supported by the Spanish Interministerial Committee of Science and Technology by means of the program DPI2001-4404-E.

References

1. Adriaans P. and Zantinge D. Data Mining. *Addison Wesley Longman.* (1996).
2. Aguilar J., Riquelme J.M. and Toro M. Decision queue classifier for supervised learning using rotated hiperboxes, *Lecture Notes in Artificial Intelligence* 1484: 326–336 (1998).
3. Aracil J., Ponce E., and Pizarro L. Behaviour patterns of logistic models with a delay, Mathematics and computer in simulation No 44, 123–141 (1997).
4. Gasca R.M. Razonamiento y Simulación en Sistemas que integran conocimiento cualitativo y cuantitativo, *Ph.D. diss., Seville University*.(1998).
5. Kay H. Refining imprecise models and their behaviours. *Ph.D. diss., Texas University,* (1996).
6. Kuipers B.J. Qualitative reasoning. Modelling and simulation with incomplete knowledge, *The MIT Press*, (1994)
7. Mavrovouniotis M.L. and Stephanopoulos G. Formal Order-of-Magnitude Reasoning. *Process Engineering Computer Chemical Engineering*, No. 12, 867-880, (1988).
8. Ortega J.A., Gasca R.M., and Toro M. Including qualitative knowledge in semiquantitative dynamic systems. *Lecture Notes in Artificial Intelligence* No.1415, 329–338, (1998).
9. Ortega J.A., Gasca R.M., and Toro M. Behaviour patterns of semiquantitative dynamic systems by means of quantitative simulations *The 16th International Joint Conference on Artificial Intelligence* Qualitative and Model based Reasoning for Complex Systems and their Control, Stockholm (Sweden), 42–48, (1999)
10. Ortega J.A. Patrones de comportamiento temporal en modelos semicualitativos con restricciones. *Ph.D. diss., Dept. of Computer Science, Seville Universit,* (2000).
11. Rastogi R. and Shim K. Data mining on large databases. *Bell laboratories*. (1999).
12. Travé-Massuyès, L., Prats F., Sánchez M., Agell N. Consistent Relative and Absolute Order-of-Magnitude Models, *In Proc. Of 16th International Workshop on Qualitative Reasoning, 185–192*, (2002).
13. Travé,-Massuyès L., Dague Ph., and Guerrin F. Le raisonement qualitativ pour les sciences de l'ingénieur, *Hermes Ed.* (1997)
14. Vescovi M., Farquhar A., and Iwasaki Y., Numerical interval simulation: combined qualitative and quantitative simulation to bound behaviours of non-monotonic systems. *Proceedings of 14th International Joint Conference on Artificial Intelligence*, 1806–1812, (1995).