G Model
ECOCOM 610 No. of Pages 7
ARTICLE IN PRESS

Ecological Complexity xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Ecological Complexity

journal homepage: www.elsevier.com/locate/ecocom

Original Research Article

Stability, sensitivity and uncertainty rates in the flow equations of ecological models

Josué-Antonio Nescolarde-Selva*, José-Luis Usó-Doménech, Miguel Lloret-Climent

Department of Applied Mathematics, University of Alicante, Alicante, Spain

ARTICLE INFO

Article history:
Received 24 September 2016
Received in revised form 2 November 2016
Accepted 7 November 2016
Available online xxx

Keywords:
Ecological model
Flow equations
Impure systems
Sensitivity
Stability
State equations

ABSTRACT

In previous work by the authors about dynamic system modeling, basic ecosystems concepts and their application to ecological modeling theory were formalized. Measuring how a variable effects certain processes leads to improvements in dynamic systems modelling and facilitates the author's study of system diversity in which model sensitivity is a key theme. Initially, some variables and their numeric data are used for modeling. Predictions from the constructed models depend on these data. Generic study of sensitivity aims to show to what degree model behavior is altered by modification of some specific data. If small variations cause important modifications in the model's global behavior then the model is very sensitive in relation to the variables used. If uncertain systems are considered, then it is important to submit the system to extreme situations and analyze its behavior. In this article, some indexes of uncertainty will be defined in order to determine the variables' influence in the case of extreme changes. This will permit analysis of the system's sensitivity in relation to several simulations.

1. Introduction

In dynamic impure system modeling it is not possible to obtain a system as a model because it would suppose a complete knowledge of reality. Moreover, it is usual to consider that a model is more real when more variables are studied and when there are a lot of relationships between them. However, it seems that the increase of parameters does not necessarily contribute to obtaining a better simulation. For building models of impure systems, authors start from inductive-deductive methodology applied and published in other works concerning the study of the sensitivity and stability of ecological models (Cortés et al., 2000; Mateu et al., 1998; Nescolarde-Selva et al., 2015a; Nescolarde-Selva et al., 2015b; Sastre-Vazquez et al., 1999, 2000; Usó-Doménech et al., 1995; Usó-Doménech et al., 1997; Usó et al., 1997; Usó-Doménech et al., 2000; Usó-Doménech and Sastre-Vazquez, 2002; Usó-Doménech et al., 2002a; Usó-Doménech et al., 2002b; Usó-Doménech et al., 2006a; Usó-Doménech et al., 2006b; Usó-Doménech et al., 2016; Villacampa et al., 1999). Starting from this methodology, it is possible to determine one mathematical expression for a differential equation system set. These equations (state equations (Rodríguez et al., 2016, 2012) are elaborated hypothetical shapes (deductive methodology) and depend on variable set called flow variables. The mathematical expression of these variables is obtained from experimental data (inductive methodology).

2. Model theory

In this paper the authors assume that the impure system is defined starting from differential equations system called state equations $\left\{\frac{dy_i}{dt}\right\}_i$. We assume that the dynamics of the impure system can be modeled starting off with a set of ordinary differential equations as follows:

$$\frac{dy_i}{dt} = F(x), \overline{x} = x(t), t \geq 0; \overline{x}(0) = x_0, \forall j \tag{1}$$

$$x : [0, +\infty] \to R^n; y(t) = F(x(t)); F : R^n \to R$$

where R^n is the phase space, t the time and y is the state variable.

Hence we start off with a system of non-linear differential equations

$$\frac{dy_i}{dt} = \sum_{i=1}^{n} x_{ij}, \forall j = 1, 2, ..., n \tag{2}$$

where x_{ij} are the flow variables which produce the state variable y_j. The equations associated with flow variables are called *flow*

* Corresponding author.
E-mail address: josue.selva@ua.es (J.-A. Nescolarde-Selva).

http://dx.doi.org/10.1016/j.ecocom.2016.11.001
1476-945X/

Please cite this article in press as: J.-A. Nescolarde-Selva, et al., Stability, sensitivity and uncertainty rates in the flow equations of ecological models, Ecol. Complex. (2016), http://dx.doi.org/10.1016/j.ecocom.2016.11.001

equations. These equations represent the biological, chemical and physical processes in the ecosystems. They show the relations between external variables (forcing functions) and state variables (Jörgensen, 1988). Each one of the flow variables can depend either on the input variables or on the state variables. The flow variables determine the variations of the system states and characterize the actions that are taken which are accumulated in the corresponding levels or states. The flow variables determine how the available information is converted into an action. Equations that define the behavior of the system are associated with flow variable. These equations associated with a flow variable are called *flow equations* or *decision equations*. These equations represent the biological, chemical and physical processes in the ecosystems. They are the relationship between the external variables (forcing functions) and state variables.

Each one of the flow variables can depend either on the input variables or on state variables. We will call z the set formed by the state and input variables and we will identify it as an open subset of R^n. it is possible to write it as follows:

$$\forall x_{ij}, x_{ij} = f_{ij}(z_1(t), z_2(t), ..., z_n(t)) \tag{3}$$

$$z_i : [0, +\infty] \to z \in R^n, \left(f_{ij} : R^{n^2} \to R^n\right)$$

Our goal is to express every x_{ij} as a linear combination of transformed functions, so that they adjust to the model studied through linear regression.

For convenience, the transformed functions of order 0 are expressed by $T^1(z_r)$. The transformed function of order 1 by T^2 where $T^2(z_r, z_s) = z_r \circ z_s$. In general, a transformed function of order *k-1*, which is a composition of $z_{i1}, z_{i2}, ..., z_{ik}$, is expressed as

$$T^k(z_{i1}, z_{i2}, ..., z_{ik}) = z_{i1} \circ z_{i2} \circ ... \circ z_{ik} \tag{4}$$

And

$$x_{ij}(t) = \sum_{r=1}^{n} c_r^1 T^1(z_r) + \sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}^2 T^2(z_r, z_s) + \sum_{r=1}^{n}\sum_{s=1}^{n}\sum_{h=1}^{n} c_{rsh}^3 T^3(z_r, z_s, z_h) + ... + \sum_{r=1}^{n}\sum_{s=1}^{n}...\sum_{u=1}^{n} c_{rs...u}^p T^p(z_r, z_s, ..., z_u) + ... \tag{5}$$

Using the statistical regression methodology

$$x_{ij}(t) = \sum_{r=1}^{n} c_r^1 T^1(z_r) + \sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}^2 T^2(z_r, z_s) + \sum_{r=1}^{n}\sum_{s=1}^{n}\sum_{h=1}^{n} c_{rsh}^3 T^3(z_r, z_s, z_h) + ... + \sum_{r=1}^{n}\sum_{s=1}^{n}...\sum_{u=1}^{n} c_{rs...u}^p T^p(z_r, z_s, ..., z_u) + ... + b \tag{6}$$

Therefore, the equation of the state variable becomes

$$\frac{dy_j}{dt} = \sum_{i=1}^{n}\sum_{r=1}^{n} c_r^1 T^1(z_r) + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}^2 T^2(z_r, z_s) + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n}\sum_{h=1}^{n} c_{rsh}^3 T^3(z_r, z_s, z_h) + ... + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n}...\sum_{u=1}^{n} c_{rs...u}^p T^p(z_r, z_s, ..., z_u) + ... \tag{7}$$

Integrating using Euler's method

$$y_j(t+1) = y_j(t) + \delta_t \left[\begin{array}{l} \sum_{i=1}^{n}\sum_{r=1}^{n} c_r^1 T^1(z_r) + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}^2 T^2(z_r, z_s) + ... \\ + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n}...\sum_{u=1}^{n} c_{rs...u}^p T^p(z_r, z_s, ..., z_u) + ... \end{array} \right] \tag{8}$$

and in condensed form

$$y_j(t+1) = y_j(t) + \delta_t \left[\sum_{i=1}^{n} \Phi_{ij}(t) \right] \tag{9}$$

We consider a particular case based on transformed functions of order 0 and 1 of the SELEGO program (Cortés et al., 2000; Usó-Doménech et al., 2000) which generates well adjusted functions (flow variables) from field data. A generator and adjuster of non-linear functions within the numerical range of the variables were used. By means of an iterative process, the program looks for the best fit with data among a set of possible functions. The SELEGO computer program has the following characteristics:

1) Each variable considered represents an attribute of the object under study.

2) One variable has been selected as a dependent variable with a set of others being considered as independent.

3) The program carries out successive runs (trials) with a set of possible ways of building the function. These functions will always have the following generic form:

$$\Psi = \sum_k \sum_L T_k^i T_L^i D_{kL} \tag{10}$$

where D_{kL} are the coefficient that have to be determined and T_k^i, $T_k^i$are the transformed functions.

4) If, with a lineal combination of *n*-transformed functions, a correlation coefficient R_1 is obtained, one can discover another lineal combination of $n+1$ transformed functions, in which the correlation coefficient R_2, will probably be $R_2 > R_1$.

5) When one has idenified one function with n transformed functions from m variables with $n < m$, the variables will always be the most significant.

2.1. Generative grammar of transformed functions, GT. The language L (GT)

The basic elements in building this grammar are described in (Nescolarde-Selva and Usó-Doménech, 2014a, 2014b; Villacampa et al., 1999):

a) *Initial symbol*: *T*, a function that will indicate the transformation applied to a primitive variable or p-symbol.

b) *Terminal vocabulary*: *x* as primitive variable and its associative field.

c) *Auxiliar vocabulary*: The function *T*, starting as the p-symbol *x* and developing the corresponding transformed functions, *T(1), T (2), . . . ,T(n).*

d) *Grammar rules*: which build the transformed functions of different orders:

Rule I. T → T(i) i = 1,2,3, ...,n
Rule II. T(1) → Ti1(x), i = 1,2,3, ...,n
Rule III. T(2) → T2ij(x), i,j = 1,2, . . . ,n
. . . .
Rule N. T(n) → Tnij . . . w(x), i,j, . . . ,w = 1, . . . ,n

This can also be shown using a tree system in Fig. 1.

The grammar GT will generate the language L(GT) or the language of the transformed functions. The language L(GT) will be a *formal language*, where here it means that:

a) There exists primitive symbols (variables and transformed functions of order 1).

b) There exists complex expressions or formulas (transformed functions of order $n \geq 2$).

c) Each symbol is a consequence of one or more symbols (recursion process).

It is possible to generate symbols (chains) in this grammar with the property of deriveability.