Variables and Softcomputing

Ossi Taipale
Taipale Engineering Ltd
Laserkatu 6
53850 LAPPEENRANTA
FINLAND
E-mail: ossi.taipale@kareltek.fi

Abstract

The national technology programme ”Adaptive and intelligent systems applications” has been going on for six years in Finland. Over one hundred companies, business units or research institutes have participated the programme. The programme is one of the largest information technology programmes ever done in Finland. The budget is over 30 million EUR. Tens of new working applications and products has been developed in the research- and product development projects (Oja, 1995). There have also been projects where applying adaptive and intelligent methods has been difficult. Often the reason has been searched from the modelling methods. However, when we look closer at these projects we find out that there have been difficulties in variable selection and in measurement data. These two steps, variable selection and creation of the data set from measurements, create the basis for modelling. When the basis is solid the modelling can be done by different methods.

In this article, object oriented methods are suggested for solving the variable selection problem. Both adaptive and intelligent methods and object oriented methods describe the human cognition. Object oriented methods help in defining the variables and the structure in modelling adaptive and intelligent systems.

1 Introduction

Adaptive and intelligent systems [soft computing] consist of methods modelling human cognition. Most important abstraction methods are neural networks, fuzzy logic and probabilistic methods. Genetic algorithms are optimization methods. Other methods are for example wavelet transformations, virtual reality, machine translation, artificial life etc. (The Cognizer Almanac, 1994).

The starting point in modelling is succesful selection of the variables. This is also one of the main targets of system design. Most common methods in system design are structured analysis and design (SA/SD) and object oriented analysis and design (OOA/OOD). The term object oriented (OO) means that the software consists of objects, that contain both data structure and behaviour. An interesting approach is to use OOA/OOD in specifying variables and data structure in adaptive and intelligent systems (figure 1).

	SA/SD	OOA/OOD
Methods of soft - computing		Article
Traditional methods		

Figure 1. Research area

2 Modelling

Modelling by adaptive and intelligent systems can be divided into two groups: classification (clustering, if the clusters are not known) of the variables and modelling the dependencies between variables. The target of the modelling is either to divide the data into classes (clusters) or explain causality. The terms class and causality are not well defined.

The following two rules are necessary in modelling dependence:

1) The model contains correct variables.
2) The behaviour of the object is described correctly.

These two rules soud trivial but most of the difficulties in the modelling projects come either from the variable selection or from the difficulties to describe the behaviour of the object (measurement data set, rules, distributions, time, etc.). To quote Dr. Jorma Rissanen, evaluator of our technology programme: ”too big share of the research in the adaptive and intelligent systems is concentrated on the estimation of the model parameters, when a more important problem is the selection of the model structure” (Rissanen, 1997). Correct modelling is the precondition for inference.

Dependency modelling methods are for example:

1. In mechanistic modelling variables and dependences are based on physical or chemical laws or on other theoretical dependencies.
2. In statistical models dependencies between variables are found by statistical methods.
3. In neural networks dependencies between variables are trained from examples.
4. In probabilistic methods probability distributions are established. Neural networks and probabilistic methods belong to statistical methods.
5. In expert systems rules are collected into the knowledge base and they define the dependencies between variables.
6. In fuzzy systems the dependencies between variables are described in the rule base and membership fuctions give the firing level of the rule.

Above methods are used also in solving classification and clustering problems. In classification the target is not to find dependencies between variables but share the objects into classes or clusters.

Modelling of the dependency is approximation of a function in multidimensional space. As geometrical interpretation, variables and dependencies define hyperplane in multidimensional space. In inference we use this hyper plane by moving along the surface. Variables and their dependencies (hyperplane) is common to all techniques modelling dependency.

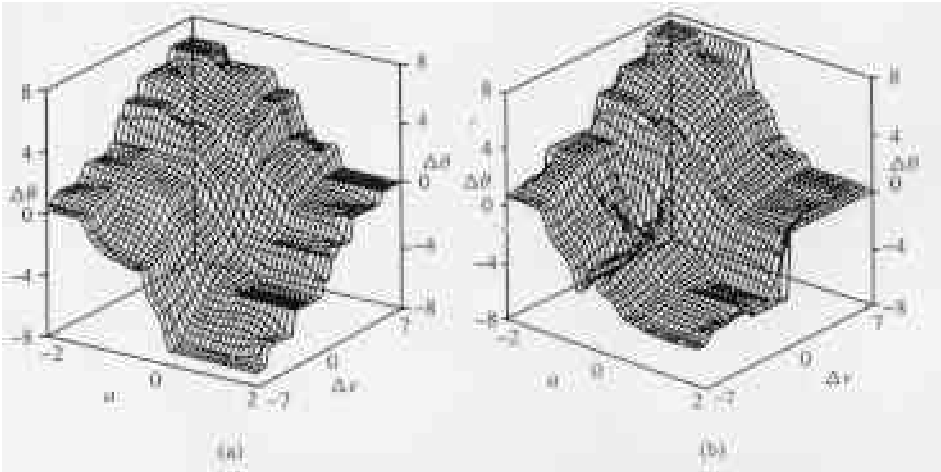

Figure 2. Control surface of the car speed: (a) self-organizing neural network, (b) supervised neural network, tuned by fuzzy logic (Kosko, 1997).

2.1 Errors in modelling

When using modelling methods in real world projects we select the structure of the model, the variables and collect the data set. A modelling project can turn into a data collection project, where the important steps, variable selection and definition of the data set, is tried to replace by amount of data. Often existing measuremets are accepted as model variables without paying enough attention to variable selection and model structure. This means that we have variables from different abstraction level and the model structure is not defined.

Based on a survey of 30 projects in our technology programme, it was found that the most typical errors in modelling were:

1. It has been suppopsed that extra variables vanish in the modelling process because there aren´t any correlations. This is not true. If we have data set with many variables, we notice that some of them correlate. In real world situations it is not possible to collect so many measurements that uncorrelated variables would vanish. Also the modelling methods don´t drop out variables that correlate via a third variable (For example CPU load and outside temperature correlate via third variable time). We should also remember that correlation is used in modelling but it doesn´t as such express causality.
2. It has been supposed that nowadays when we have lot of data we directly have a data set that expresses dependencies between variables. Unfortunately this kind of sample is very rare. Normally existing measurement data contains insufficient information because we get lot of measurements from some areas of the measurement space but the measurements don´t cover the whole measurement space.

3. The dynamics of the process must be known if we approximate it with a static model, because the dynamics can invert the dependencies. The static model doesn´t replace a dynamic model.

2.2 Errors in variable selection

A common misstake is to use too many variables. The variables are often from the wrong abstraction level, mostly too detailed variables. We can use also too few variables. Now some essential variables lack from the model. Also now the abstraction level is wrong. We can use in the model variables that don´t belog to the model at all. Now our modelling object will change.

By computer it is of course possible to analyse interdependencies between many variables, but it often means that we have many variables that should be combined to some correct abstraction level variables or to be rejected.

2.3 Measurement errors

There are many error sources in the measurements. It is impossible to make comprehensive listing of all error sources because the errors depend on the modelling object. Here are some typical error sources:

Inaccuracy causes noise in the measurements.
There is drift in the sensors because of fouling or too long calibration time.
Compensations can cover interesting measurements.
The amount of missing or wrong measurements has been 50 % in some quality database we have examined.
Latent variables have influence to the output but are not measured.
The data set can have too few measurements.
If the structure and the variable selection of the model is unclear there can be functional dependencies between variables.
The measurements don´t cover the measurement space (Hämäläinen, 1998).
Timing problems: delays, containers, production breakdowns, wrong time stamps etc.

3 Object oriented methods in modelling

Object oriented methods are valuable tools for describing the model structure and the variables. Object oriented approach consists of four concepts: identity, class, polymorfism and inheritance.

3.1 Elements of the object oriented system

Here is a short description of the elements of object oriented system. These elements can be used in describing adaptive and intelligent systems in defining the variables, in finding dependencies and in specifying the structure of the model. The description is based mainly on the references (Coad and Yourdon, 1991a and 1991b).

Class and objects

Object is an abstraction of the research area. It contains attributes and methods. Methods change the values of the attributes. Operations inside an object are encapsulated. Only the services implemented my methods are visible outside through protocol. Object is unique, it has identity.

Class describes the common attributes and methods of object set. Class creates objects. Objects are instances of the class. Methods and attributes can be inherited. This creates class hierarchy.

Inheritance

In the class hierarchy subclass inherits attributes and methods from the superclass. Class hierarchy builds up from generalization-specialization structures, where upper class defines general characteristics which lower classes specialize.

Attributes

Attribute is the state information of an object.

Methods

Execution of the method changes the values of the attributes and is causes the state transition. The life cycle of an object is the serie of state transitions. The object can use its own methods or call another object to produse needed services [responsibilities].

Structures

The structure consists of generalization-specialization –structures and of whole-part –structures. Generalization-specialization –structure is the class hierarchy with super- classes and subclasses. This structure defines the inheritance. Whole-part –structure means directed dependency between objects, where the lower object belongs to the upper.

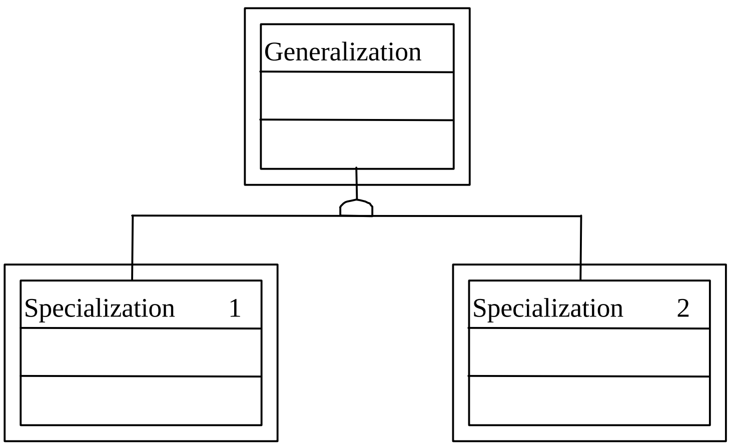

Figure 3. Generalization-specialization -structure (Coad and Yourdon, 1991a).

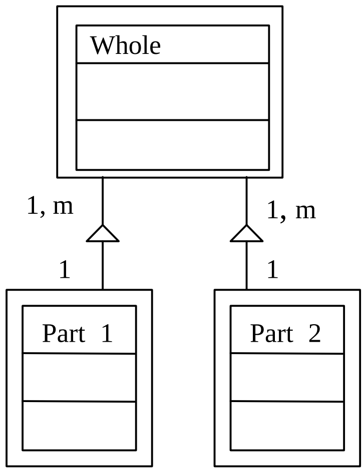

Figure 2. Whole-part -structure (Coad and Yourdon, 1991a).

4 Conclusions

By using object oriented methods we can say that adaptive and intelligent classification methods classify states of the objects (values of the attributes) into classes. Accordingly dependency model is an object, that encapsulates attributes (variables) and methods (dependencies betweeen variables). If dependence model consists of many objects then dependence model is a directed dependencew between objects. The structure of a classification model is generalization-specialization –structure and correspondingly the structure of a dependency model is whole-part –structure.

In the following table are elements of adaptive and intelligent systems and corresponding elements of object oriented systems:

Modelling element	OO-element	Interpretation
Variable	Attribute	Defines the object
Dependency	Method	Defines the behaviour of the object
Dependency model	Object	Geometrical interpretation is hyper plane
Class (attributes)	Class (attr. and methods)	Class of objets
Observation	State of an object (time)	Geometrical int. is point on the plane
Time	State in the life cycle of an obj.	Methods change the values of the attributes
Complete representation	All states of an object	Geom. int. known area of the plane
Inference	State transitions of an object	Geom. int. moving on the plane

Table 1. Modelling elements and object oriented elements (Taipale and Jurva, 1999)

References

Coad, P., Yourdon E. 1991a. Object-Oriented Analysis (2nd edition). Prentice-Hall, New Jersey, USA, 233 p.

Coad, P., Yourdon E. 1991b. Object-Oriented Design. Prentice-Hall, New Jersey, USA, 197 p.

The Cognizer Almanac. The First Step for Information on Emerging Technologies. Vol 1. Cognizer Co, Oregon, USA.

Hämäläinen J., Järvimäki I. 1998. Input projection method for safe use of neural networks based on process data. IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998.

Kosko, B. 1997. Fuzzy Engineering, Prentice Hall International Inc., New Jersey, 547 p.

Oja, E. and Taipale, O. 1995. Applications of Learning and Intelligent Systems –the Finnish Technology Programme, Industrial Conference – Applications, ICANN'95, Paris, France.

Rissanen, J., Louhenperä, R. 1997. Adaptive and intelligent systems applications 1994-1998. Helsinki, Tekes report 2/97.

Taipale, O., Jurva E. 1999. Selection of the variables and creation of the data set, Tekes 66/99, Helsinki, Finland (in finnish)