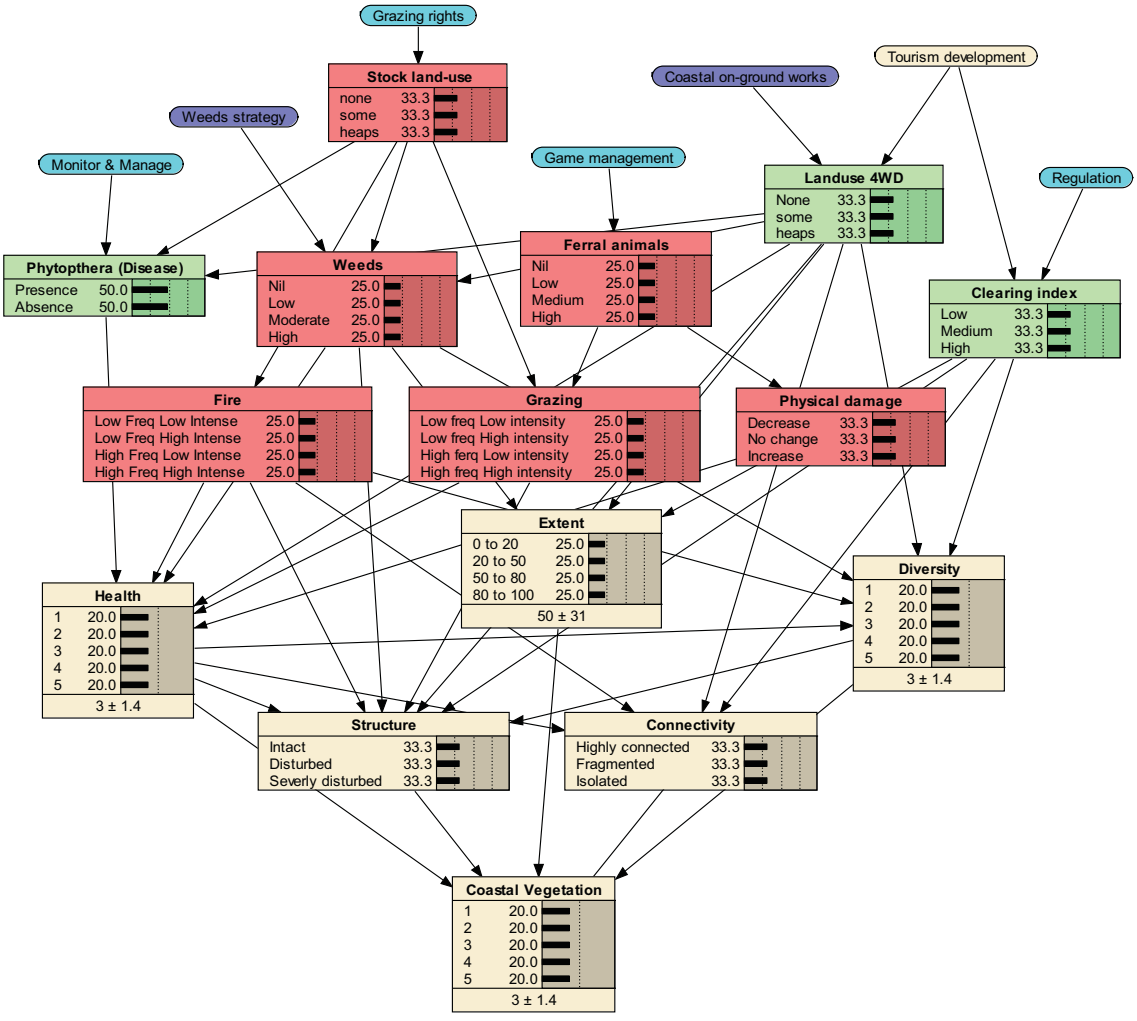
LANDSCAPE LOGIC

LINKING LAND AND WATER MANAGEMENT TO RESOURCE CONDITION TARGETS

Technical Report No. 7

Incorporating dynamics and feedback into Bayesian modelling of natural resource systems

February 2010



Australian Government

Department of the Environment, Water, Heritage and the Arts

Published February 2010
This publication is available for download as a PDF from www.landscapelogic.org.au

Cover: Bayesian Decision Network designed in discussion between Landscape Logic and staff of NRM Cradle Coast for the Arthur Pieman Reserve, Tasmania.

LANDSCAPE LOGIC is a research hub under the Commonwealth Environmental Research Facilities scheme, managed by the Department of Environment, Water Heritage and the Arts. It is a partnership between:

- **six regional organisations** – the North Central, North East & Goulburn–Broken Catchment Management Authorities in Victoria and the North, South and Cradle Coast Natural Resource Management organisations in Tasmania;
- **five research institutions** – University of Tasmania, Australian National University, RMIT University, Charles Sturt University and CSIRO; and
- **state land management agencies in Tasmania and Victoria** – the Tasmanian Department of Primary Industries & Water, Forestry Tasmania and the Victorian Department of Sustainability & Environment.

The purpose of Landscape Logic is to work in partnership with regional natural resource managers to develop decision-making approaches that improve the effectiveness of environmental management.

Landscape Logic aims to:

1. Develop better ways to organise existing knowledge and assumptions about links between land and water management and environmental outcomes.
2. Improve our understanding of the links between land management and environmental outcomes through historical studies of private and public investment into water quality and native vegetation condition.

Incorporating dynamics and feedback into Bayesian modelling of natural resource systems

By John Norton, Integrated Catchment Assessment & Management Centre, Fenner School of Environment & Society and Mathematical Sciences Institute, Australian National University

Summary

During the establishment phase of Landscape Logic (2006–07) there was much discussion internally, as well as between Landscape Logic staff and its partner organisations, on the merits of using Bayesian Decision Networks to model landscape processes such as the quality of rivers and estuaries, and changes in native vegetation.

In its research into water quality and native vegetation in Tasmania and Victoria, Landscape Logic aimed to assemble a wide range of data; from extensive analysis of existing and historical data, original biophysical and social research, econominic data, and expert opinion. In discussions, led by the Landscape Logic Knowledge Integration team, based at the Integrated Catchment Assessment and Management Centre at ANU, we came to the conclusion that Bayesian Decision Networks (BDNs) offered the best solution to include this wide range of information into useful predictive models for the purpose of assisting natural resource managers to make investment decisions.

This technical report was written by John Norton to express the views of the Landscape Logic Knowledge Integration team on the limitations of BDNs and how these could be reduced or eliminated. Section 1 of this report identifies the major benefits of BNs as:

- showing cause-effect relations directly and displaying them graphically
- easily constructed, extended and modified
- incorporating uncertainty in variables and relations yet can be understood without much mathematical background
- employ a fundamental and relatively simple method of combining imprecise information (Bayes' Rule), and
- produce outputs (probabilities of values or variables) well suited to decision support.

However, they also have significant limitations. First, time and space are not present as independent variables. Variables which depend on past events (dynamics) or on adjacent conditions (distributed behaviour) are not represented explicitly.

A second limitation of BNs is that they cannot contain feedback loops. This is essentially because the mechanism for using new information, such as a field measurement, to improve knowledge of related variables cannot deal with feedback. In natural systems, feedback loops commonly occur where A affects B and is also affected by B, perhaps indirectly. This report examines some ways to remove these limitations, while keeping as much as possible of the simplicity and flexibility which make BNs so attractive. Addition of supplementary variables to a BN to introduce time or spatial dependence are compared with other approaches which assume such dependence right from the start.

Some alternatives to BNs such as Hidden Markov Models (HMMs) are also considered.

Acronyms

ANU	Australian National University
BDNs	Bayesian Decision Networks
DBN	Dynamic BN
HMM	Hidden Markov Model
CPT	Conditional-Probability Table
ICAM	Integrated Catchment Assessment & Management Centre, Fenner School of Environment and Society and Mathematical Sciences Institute, Australian National University, Canberra.

Acknowledgements

Thanks are due to Jessica Hudspeth, whose report on a short research assignment in her PhD program at The Australian National University provided a first draft of parts of this review. Provision of the algal bloom model by Dr Carmel Pollino of iCAM, with the cooperation of Dr Angus Webb of The University of Melbourne, is gratefully acknowledged. Comments by Dr. Pollino on a draft of the report helped its balance greatly. The author also wishes to acknowledge the open-mindedness of the Knowledge Integration team and subsequently of the Landscape Logic Technical Advisory and Management Committees in accepting that knowledge-integration methods not limited to static BNs should be examined.

Contents

1. Introduction

The Knowledge Integration team in Landscape Logic is employing Bayesian Networks (BNs) as the main means of modelling ecological systems. BNs, explained in Section 2.1, have several appealing properties:

- they show cause-effect relations directly and are readily displayed graphically
- they are easily constructed, extended and modified
- they incorporate uncertainty in variables and relations yet can be understood without much mathematical background
- they employ a fundamental and relatively simple method of combining imprecise information (Bayes' Rule), and
- they produce outputs (probabilities of values of variables) well suited to decision support.

On the other hand, in their basic form BNs have significant limitations. First, time and space are not present as independent variables indexing the behaviour of the system, so dependence of variables on previous history (dynamics) or on adjacent conditions (distributed behaviour) is not represented explicitly. As discussed later, such dependence can be brought in by extending the list of variables but at the cost of making the BN more complex. The consequence of not modelling dynamics is that the model can only mimic the behaviour of the system as measured by single values of each variable, such as steady-state, average or extreme values. This is clearly unsatisfactory when the evolution of variables over a period is of interest, for instance when results of a given climate scenario, or of actions to counter a trend, are to be predicted.

A second limitation of BNs is that they cannot contain loops. This is essentially because the mechanism for using new information, such as a field measurement, to improve knowledge of related variables cannot deal with them. In natural systems, feedback loops commonly occur, where A affects B and is also affected by B, perhaps indirectly. Examples are predator-prey relationships in ecology and systems in which the consequences of an action cause the action to be modified, as in adaptive environmental management. Inability to model such loops is serious.

The purpose of this report is to examine some ways of removing these limitations, while keeping as much as possible of the simplicity and flexibility which make BNs so attractive. Addition of supplementary variables to a BN to introduce time or spatial dependence will be compared with other approaches which assume such dependence right from the start. Updating of knowledge in feedback loops is closely linked to handling dynamics. The cause-effect paths making up a loop have finite delays, so the implications of new information can be followed round the loop provided effects can be followed through time.

In examining alternatives to BNs, attention will be confined to mature approaches with clearly defined scope, well developed methods of establishing model structure, estimating model parameters and bringing in new information, and a history of successful application. State-variable models, fuzzy models and Hidden Markov Models (HMMs) will be considered. Markov Chain Monte Carlo modelling will not, even though it is receiving increasing attention, as it is less readily understood than the other approaches. "Universal" model structures such as artificial neural networks, radial basis functions and support vector machines are excluded because they yield black-box models with no guarantee of physical interpretability.

Section 2 concisely describes the modelling approaches and discusses their assumptions, scope and limitations in the context of Landscape Logic. In Section 3 BN modelling and the most promising of the other approaches are then applied to an application example, an algal bloom model where dynamics should not be ignored, feedback is present and degree of detail is a critical issue. Section 4 lists the steps in constructing BN models of systems with dynamics and/or feedback. This section is self-contained and could be read independently of the rest of the report. The final section draws lessons from the application example and makes recommendations for modelling dynamical and feedback systems in Landscape Logic. It can be taken as an executive summary.

Nomenclature

''Causative'' will be used instead of ''causal'' as the adjective referring to a cause, as the latter has a strict technical definition (a causal relation is one where the start of the response does not precede its stimulus: a causal system does not laugh before it is tickled). ''State'' will be used in a strictly defined sense in the sections on state-variable modelling and HMMs and will be avoided otherwise as far as possible. ''Controllable'' will be used in its informal sense, signifying a variable which can be altered, rather than applying to a system which can be driven through any specified state transition, the sense defined in control engineering.

Notation

There are many clashes between the standard notations of the various modelling approaches. For consistency, the notation commonest in state-variable modelling will be used as far as possible. Double subscripts, one to identify the variable and the other denoting time, are unavoidable; the second will always signify time, but will be omitted when inessential. Boldface denotes a vector. For single-subscripted vectors, the subscript signifies time.

2. Probabilistic modelling approaches

2.1 Bayesian Networks

2.1.1 Static BNs

A Bayesian network is a graphical model consisting of nodes connected by unidirectional lines (arcs) [1,2]. Each node represents a physical variable, treated as a random variable which can take any one of a number of discrete values. Variables may be Boolean (high/low, true/false, present/absent), integer or real. A continuous variable is handled by dividing its range into sub-ranges, each assigned a discrete value. Each arc connects two nodes and represents a cause-effect link between their variables, the arc running from cause to effect. The node causing the effect is called a parent node and the affected node a child. All nodes affecting the child's parents are ancestor nodes and all nodes affected by the child node are descendant nodes. Root nodes are those without parents and leaf nodes are nodes without children. [Loose and mixed metaphors are a feature of BNs]. Any other nodes are called intermediate nodes.

Target nodes are the output nodes about which the user wants information. Observation nodes are nodes where observations provide information about other variables. Controllable variables are variables whose values can be set, not merely observed, and context nodes help describe background causative conditions [1].

The relation between a child node and all it parents is described by a conditional probability table (CPT). For each possible combination of values of the parent nodes, any one entry in the CPT gives the probability that the child takes a particular one of its discrete values, given a particular combination of values of its parents' states. These probabilities depend only on the parent values (in the absence of related observations not yet taken into account). For each combination of values of the parent nodes, the sum of the probabilities of the possible values the child may take must be one. The size of the CPT is the product of the numbers of values at the child node and all its parent nodes.

When new information becomes available about the value of a node, the probabilities in the BN are updated by Bayes' Rule [3], which relates the conditional and marginal probabilities of two events A and B:

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)} \qquad (1)$$

Here $\Pr(A|B)$ is the posterior probability of A given B, $\Pr(A)$ is the prior marginal probability of A, $\Pr(B|A)$ is the conditional probability of B given A, and $\Pr(B)$ is the marginal probability of B. Event B is that a child node has a specific value and event A is that a parent node takes a particular one of its permitted discrete values. The prior $\Pr(A)$ does not employ any information about B. With B given, $\Pr(B)$ is a constant unaffected by the value of A and merely normalises $\Pr(A|B)$ so that its sum over all possible values of A is 1. This updating mechanism allows the probabilities of variables that are not directly observed to be updated, via the CPTs, from knowledge of related variables. The computing load of updating parents' probability distributions on the basis of new information about the child rises with the size of the CPT [1].

Propagation of information around BNs looks at first sight as if it involves updating a large part of the network each time an item of information arises. However, the scope of an update is limited by independence relationships. Nodes A and B are conditionally independent if there is no way to get from A to B via the directed arcs. Conditional independence (also known as d-separation) can also arise in causative chains.

For example, if A causes B which causes C, then C and A are conditionally independent, since if it is known *without uncertainty* that B has occurred then C is unaffected by any knowledge about A, and vice versa. [If an observation of B is subject to observational error, then knowledge of A contributes to knowledge of C, so A and C are no longer independent. This fact seems to be ignored by textbooks on BNs].

Common cause acts similarly; if B causes both A and C then A and C are again conditionally independent, since knowledge of A or C is unaffected by information about the other if B is known exactly. Common effects, on the other hand, imply conditional dependence. If A and C both cause B, then if an effect on B is known, then knowing for instance that Pr(B|C) is low increases the probability that A has a value for which Pr(B|A) is high; A and C compete to explain the data B. These relationships are summarised in Figure 1.

Constructing a BN consists of choosing the variables and their possible values, identifying the cause-effect relations and thence the graph structure, and filling in the probabilities in the CPTs.

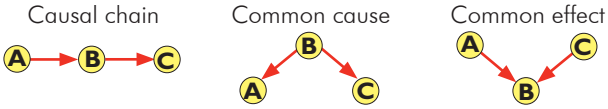


Figure 1. Common dependence relationships in Bayesian networks [1].

The model is calibrated by supplying the CPTs and initial node probability distributions. The effort required to do so and to update it when new information arrives depends, of course, on the numbers of variables, possible values and interconnections. Conversely, accuracy and fineness of resolution depend on how many nodes and arcs are used to model the processes and the number of discrete values allowed each variable. In a distributed system, coarseness of spatial quantisation will strongly influence model complexity. The resolution must reflect both the quality of information available and the degree of complexity permitted by computing load and comprehensibility. Compromise will usually be necessary between the user's desires and what can realistically be provided may in practice be the limiting factor. The complexity may in practice also be limited by difficulty in understanding and assessing the model. These points will be illustrated by the example in Section 3.

BNs have been used in a wide range of applications, including modelling ecological systems [4–9]. The model generally treats each cause or effect as a single item rather than a spatially distributed variable or time series. The next sub-section considers what happens when relations between an effect and the history of a cause, not just its value at one time or aggregated over one interval, have to be modelled.

2.1.2 Dynamical Bayesian Networks (DBNs)

DBNs [10] generalise BNs by modelling temporal relationships between the variables. The model is discrete-time as well as discrete-valued. That is, the variables are represented only as their values at regularly spaced sample instants $t = kT$, where T is the sampling interval (time step) and k any integer. To construct a DBN from a BN, one node is included for each variable at each sample instant. To keep the number of nodes as small as possible, the assumption is made that the probabilities of the next values of the variables can be computed from those of their current values and the forcing, without knowing the previous history. This assumption, discussed further in Section 2.2, is restrictive but widely applicable. Under it, it is enough to have nodes for variables at two successive sample instants k-1 and k, and arcs showing how, at sample instant k, each variable involved in the dynamics depends on that variable and others at time k–1 [10]. The DBN is run by stepping forward in time in steps of T.

DBNs can be updated using standard BN inference algorithms [1].

If also spatial dependence is to be modelled, the analogous assumption is that a variable at any given location depends only on variables at adjacent locations. The model can then be run by stepping through both time and space according to a specified ordering. Although the spatial relations are local, the current probabilities of the variables at all the locations have to be stored to allow the next probabilities to be computed. The number of nodes and arcs will thus be large unless the spatial dependence is especially simple. It *is*, for instance, in successive reaches of a stream network, where dependence is only in one spatial direction and there is only one spatial dimension for a given stream.

Because of the severe limitations imposed by complexity on modelling spatially distributed systems by extended BNs, the rest of this report will concentrate mainly on how to incorporate dynamics and feedback loops into models. To deal with spatially distributed systems where variables at many locations must be tracked through time, one can envisage modifying computational schemes for distributed systems, such as finite-element methods, by replacing deterministic relations between adjacent elements by conditional probability tables. Development of a scheme along these lines would be a large topic, well beyond the scope of this report.

2.2 State-variable models

State-variable models [11,12] are very widely employed for systems with dynamics. They exploit the assumption, acceptable for many systems, that the future behaviour of certain variables of interest, the *state variables*, can be predicted from their present values and their future forcing (external inputs), with no need to know past values. In other words, the influence of the history of the system up to time t on future behaviour is fully prescribed by the values of the state variables at t; those values constitute a full set of initial conditions for future evolution of the state. This property of the selected variables $x_1, x_2, \ldots x_n$, plus the condition that they contain no redundancies, defines them as the elements of the state vector $\mathbf{x}$ of the system. Usually the model is discrete-time, describing the behaviour of all the variables sampled at regular intervals T. With the state vector by convention written as a column if untransposed:

$$\mathbf{x}_k \equiv \begin{bmatrix} x_{1,k} & x_{2,k} & \ldots & x_{n,k} \end{bmatrix}^T \qquad (2)$$

is the state $\mathbf{x}$ at time kT with k any integer.

The dynamics part of the model consists simply of scalar difference equations giving the value of each state variable at the current sample instant kT from the state one sample interval ago and the forcing since. The equations are collected into the vector *state equation* of the form:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \tag{3}$$

where **f** is a vector of known functions and **u** is a vector of known forcing variables, with $\mathbf{u}_{k-1}$ covering the influence of forcing over the time interval from $(k\text{-}1)T$ to kT.

Some of the state variables may not be directly measurable, so a vector **y** of observed variables is defined, related to state by an instantaneous, non-dynamical *observation equation*:

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k) \tag{4}$$

where **h** is a vector of known functions, which depend on **u** only if there is instantaneous feed-through from forcing to observed output (a rare occurrence).

A discrete-time model of the form (3), (4) can be derived from a continuous-time, differential-equation model, e.g. a set of process rate equations, by analytical or numerical integration over T. In many cases **f** and **h** do not vary with time, simplifying integration and yielding a constant-parameter model.

Uncertainty is added to (3) and (4) by including the influence of unknown time-series vectors **w** and **v**, called respectively *process noise* and *observation noise*:

$$\left.\begin{aligned} \mathbf{x}_k &= \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \end{aligned}\right\} \tag{5}$$

If **f** and **h** are linear in all their arguments, this model can be analysed comprehensively and straightforwardly by linear (matrix) algebra. For instance, the responses of state and observed variables to any specified forcing and initial conditions can be predicted, either exactly from (3) and (4) or, if uncertainty is present, by use of (5) with **w** and **v** characterised as random variables. For **w** the mean $\overline{\mathbf{w}} \equiv E[\mathbf{w}]$ and covariance $\mathbf{Q} \equiv \mathrm{cov}(\mathbf{w}) \equiv E[(\mathbf{w} - \overline{\mathbf{w}})(\mathbf{w} - \overline{\mathbf{w}})^T]$ are specified, and similarly for **v**. The model (5) then gives as prediction results the means and covariances (indicating scatter) of the state and observed variables. In doing this it isn't necessary to assume anything about the probability distribution of **w** or **v**. Both are assumed to be white, though. In other words, successive values are uncorrelated with each other (but correlation between **w** and **v** at the same time does not raise difficulties). Consequently any known time structure of **w** or **v** must be modelled by an auxiliary noise model, driven by a white sequence. This is often feasible but adds extra state variables to **x**.

Many state-variable models are of systems whose dominant behaviour is well described by a modest number of variables. The model structure (3), (4) or (5) permits spatial-temporal dependence to be described. All spatial variables have to be discretised into a number of single-location variables then included in the state vector. The obvious drawback is that this may make the state dimension high. However, the number of model parameters to be supplied may be quite small if spatial influences over one time step are only local and they are similar over many successive steps. That is, the model may be relatively simple if only sparsely interconnected and uniform in behaviour.

A stream network is a sparsely connected system, but variations from reach to reach are likely to prevent the model from stepping through many spatial steps with the same parameter values. In addition, sometimes spatial lumping, like the time lumping in (3) and (4), is difficult or infeasible because the state at a given time and place depends on state behaviour, over a range of adjacent locations and previous times, too detailed to be well specified by an acceptable number of samples, and hence a sensible number of distinct variables. In that case a delay-differential or delay-difference equation is required, rather than a differential or difference equation, an infinite- rather than finite-dimensional state-variable model. *This difficulty in describing all the significant causative influence applies to modelling in general, not just state-variable modelling; the problem is just clearer in state-variable modelling.*

The ability to analyse state-variable models using standard mathematical tools is a great advantage and accounts for much of their popularity. It allows, for instance, the design of an effective and fairly simple algorithm (the Kalman filter) to estimate state from uncertainty-affected measurements in the presence of uncertain forcing. Efficient parameter estimators have also been developed to calibrate useful special cases of linear state-variable models.

Advantages of state-variable models may be summarised as:

- their origin as differential equations, conferring great flexibility in describing dynamics
- clear separation of the observation process and the dynamics
- clear separation of the known and uncertain items
- simple probabilistic specification of the uncertainties, not requiring distributional assumptions (although many authors make such assumptions for convenience or through ignorance)
- for linear systems with additive forcing and observation errors, easy analysis of their properties, conferring full understanding (e.g. through normal-mode analysis)
- mature tools for calibration and state estimation.

In practice, and in the great majority of

environmental models, **f** and **h** aren't entirely linear and we may well be interested in more than the means and covariances of the uncertain variables. Non-linearity makes analysis much harder and often impracticable. Moreover, if the distributions of **w** and **v** are anything but Gaussian (and perhaps even then), analytical prediction of the probability distributions of the state or observed variables is unlikely to be possible. However, in recent years development of a Monte Carlo approach, regrettably called *particle filtering* [13,14], has made it possible to investigate non-linear systems and systems with non-Gaussian forcing or observation noise. The idea is just to follow a large number of samples of state and forcing numerically through the state and observation equations, then examine the resulting sample distributions of the state, observations and anything else that depends on them. New observations can be exploited by calculating, for each sample of state, the likelihood of the error between the predicted and observed values. The probabilities of individual predicted state samples are then updated as the product of the prior probability and the likelihood, yielding the state's posterior probability density function in sample form. In this way state estimation becomes feasible, whatever the properties of **f**, **h**, **w** and **v**, so long as a large enough set of state samples can be processed.

There are snags, though. The most obvious is the computing load of running the model for a large number of sample state values. Less obviously, it is necessary to resample from the empirical distribution after each time step to avoid the samples diverging excessively. Conversely, the resampling scheme must be designed so that the resampled distribution does not collapse into a few values or even a single value. Particle filtering is best suited to models with few state variables and good knowledge of the relations between them, but analytically inconvenient relationships and "noise" properties. In particular, it has become popular for aerospace target tracking. Here the state variables are position, velocity and possibly acceleration components, i.e. only 6 or possibly 9 state variables per target, non-linearity arises from incompatibility of rectilinear motion coordinates and polar observations (range and direction), and noise distributions are mixtures, not amenable to analysis, due to several phenomena (clutter, glint, refraction, manoeuvres, turbulence). For models intended to aid environmental management, particle filtering will usually be excluded by lack of knowledge of the form of relations between variables and of the probability distributions of unobserved forcing, measurement and modelling errors. In rare cases where such knowledge is available, computational load is likely to be a problem.

Other features of state-variable models which may be seen as disadvantages (or as advantages, depending on the intended users) are:

- the abstract idea of state, allowing considerable freedom in the choice of state variables, including some which may not look intuitively all right. For instance, a second-order difference equation in a single response variable z, say, can be rewritten as two scalar state equations in the form required by (3), by selecting z_k and z_{k-1} as state variables, say x_1 and x_2, at time kT, say (making one of the scalar state equations $x_{2,k} = x_{1,k-1}$), an odd idea at first sight. Furthermore, any non-degenerate linear transformation of a valid set of state variables is also valid
- linear algebra, the natural language of linear, time-invariant (constant-parameter) state-variable models, nice if you speak the language, nasty if not
- putting the model into the standard form (5) may involve some work on the original, physically motivated equations (as in second-order example above), risking a loss in interpretability. The original graphical representation may be changed into a structurally simpler but less easily interpreted one
- uncertainty is prescribed by means and covariances alone. Covariances are not part of the intuitive equipment of most people (although not complicated), and they do not translate into probabilities or probability densities without some extra assumption about the form of the distribution. This is particularly a drawback when a distribution is heavily skewed or when the chance of some value being exceeded is of interest
- state variables are continuous-valued, i.e. real variables, not quantised. This is inappropriate when a variable is inherently integer-valued (e.g. binary) or when it is so vaguely known that only a few possible values (say low, medium, high) are justified
- the general form of each relation is assumed to be known. This can be the greatest strength of state-variable modelling, as it means that such prior knowledge is incorporated, or its greatest weakness, if the form is unknown.

2.3 Fuzzy modelling

The underlying idea of fuzzy modelling, fuzzy membership, is simple and has a long history [15]. A variable is classified as being in one or more sets, defined verbally and numerically, to a numerical extent between 0 (not a member) and 1 (completely a member). For example, high river flow rate might be defined as 100 units or more, with degree of

highness rising linearly from 0 at 50 to 1 at 100; medium flow as rising from degree 0 at 20 to 1 at 60 then back to 0 at 100; and degree of lowness of flow as being 1 up to 30 flow units, declining linearly to 0 at 50 units. A flow rate of 80 units then has membership 0.6 of high flow, 0.5 of medium and 0 of low.

This scheme conforms with subjectively viewing an item as having a number of attributes to varying extents (in this example, flow sort of medium but rather high, definitely not low). Vagueness is intrinsic to the scheme, although certainty can be expressed by membership 0 or 1. Once variables are classified by their memberships of fuzzy sets, the model comprises verbal rules which relate them (e.g. *if A is high and B is medium or low, then C is low*). A number of such rules yield results expressed as fuzzy memberships. A composition rule is needed to resolve them into a unique value ("defuzzification"). The rule is often all-or-nothing, e.g. highest wins.

It is not difficult to see that the entire process of fuzzification, application of rules and defuzzification amounts to a scheme for prescribing, piece by piece, numerical relations between variables. The relations are piecewise linear if the memberships vary piecewise linearly with their arguments, as in the little example above. In applications with a good deal of imprecise process knowledge, often expressed as verbal rules of thumb, control schemes based on fuzzy modelling have been effective where more formal design techniques (for instance optimising the expected value of a cost function) would be cumbersome or impracticable. Opinions of fuzzy modelling tend to polarise according to the degree to which the modeller is prepared to accept a largely informal process by which different practitioners would generally get different results, which is tested mainly by trial and error, and which is analysed with difficulty or not at all.

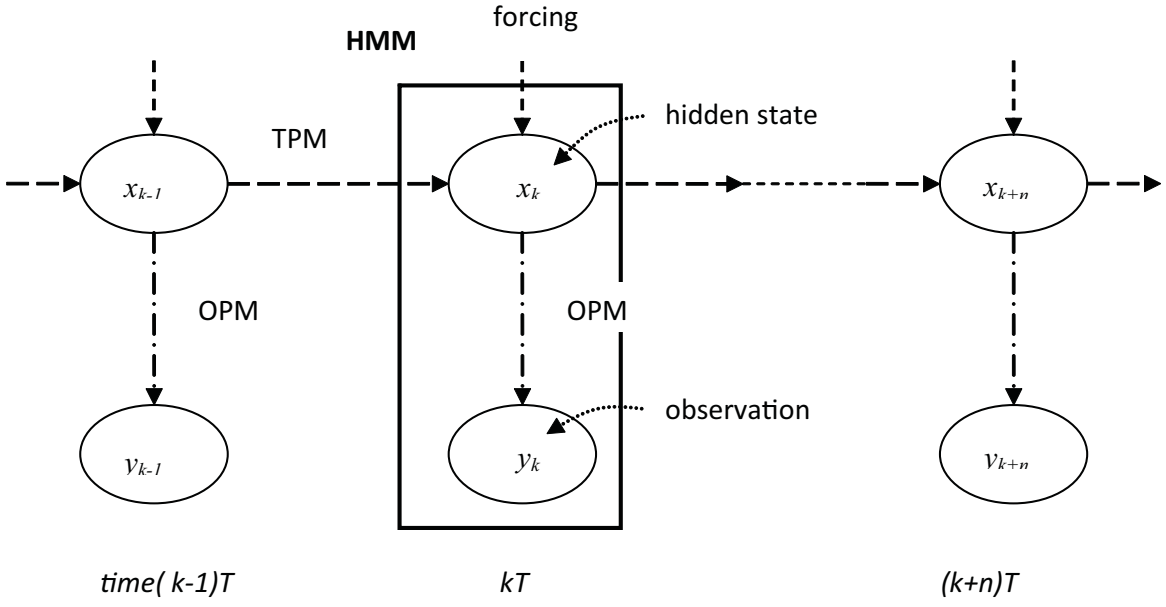
In spite of its popularity and effectiveness in some areas, fuzzy modelling is subject to substantial objections:

- it typically asks the modeller to supply a large number of numbers on subjective grounds
- the functional form of the relation between the value of a variable (such as flow) and its degree of membership of a fuzzy set (such as high flow) is largely arbitrary
- the continuous variation of degree of membership with argument may hide radical uncertainty and give a spurious impression of precision
- all-or-nothing defuzzification throws away information. Put another way, the result is insensitive to what emerges from part of the model, up to some threshold
- it isn't clear what uncertainties eventually dominate the results.

2.4 Hidden Markov models

HMMs [16–19] use tables of probabilities to describe relations between variables which can take only a finite number of possible values, in the same way as BNs. An HMM may indeed be viewed as a specialised sort of BN. However, HMMs employ the idea of state, linked to the Markov property which allows the next state to be found from current state and forcing alone, as in state-variable modelling. In each time step a new state is entered, which depends only on the state at the start of the step and the forcing during the step. However, there are

Figure 2. Successive steps of a Hidden Markov Model (boxed). TPM = transition probability matrix; OPM = observation probability matrix



differences:

- the state variables in a state-variable model are generally continuous-valued (even if discrete-time), whereas an HMM has a finite number of discrete state values
- a state-variable model retains the separate identities of the variables making up the state vector, but an HMM treats the state as a whole and computes the probabilities of each possible value of state
- an HMM is intrinsically probabilistic: each time step obeys a collection of probabilities of transitions between particular state values. State-variable models, on the other hand, mostly originate in deterministic differential-equation models, with uncertainty added if necessary through auxiliary "noise" variables.

In an HMM, the collection of probabilities of every possible state transition in one time step is written down as a matrix, with element (i,j) the probability that state value i goes to value j. As the new state value may coincide with the old one, element (i,i) is generally not zero. The state is not directly observed (it is hidden) but each state transition produces an observed output symbol, a value from a finite set of possible values. The probability distribution of this output is conditioned only on the current state, and the entire set of conditional probabilities may again be written as a matrix. As suggested by the use of matrices, a single index can be used to run through all possible values of the state at any one time. However, in environmental models it will often be helpful to think of the state as segmented into distinct variables, as in state-variable modelling. Similar comments go for the observations.

HMMs embody dynamical (state-transition) and non-dynamical (observation) processes, clearly separated just as in state-variable models. Like them they have time as an independent variable, yet they use conditional probabilities as BNs do. It therefore looks as if HMMs may provide the shape and power lacking in non-dynamical BNs, while retaining their way of handling uncertainty. Whether this is so will be tested by the application example in Section 3. Meanwhile, let us look at HMMs in a little more detail.

Figure 2 shows how the variables evolve in an HMM.

An HMM is specified by the:

- number N of state values
- number M of observation symbols
- set $Q = \{q_1, q_2, \ldots, q_N\}$ of possible state values
- set $V = \{v_1, v_2, \ldots, v_M\}$ of possible observation symbols
- state transition probability matrix $\mathbf{A}$ with $[\mathbf{A}]_{ij} \equiv a_{ij} = \mathbf{Pr}(x_{t+1} = q_j \mid x_t = q_i)$
- observation symbol probability matrix $\mathbf{B}$ with $[\mathbf{B}]_{ij} \equiv b_{ij} = \mathbf{Pr}(y_t = q_j \mid x_t = q_i)$
- initial state probability distribution $I = \{\pi_i\}$ where $\pi_i = \mathbf{Pr}(x_0 = q_i)$.

In addition to the Markov assumption, two other assumptions are made:

- the stationarity assumption: that state transition probabilities are independent of the time at which the transitions take place, i.e. that: $\mathbf{A}$ is independent of t
- the output independence assumption: that the current output is statistically independent of all earlier outputs.

While the Markov and stationarity assumptions apply in a wide range of situations, the output independence assumption is far more restrictive and can be a weakness in HMMs.

Use of an HMM poses three tasks:

1. Computing the probability of a particular output sequence given the parameters of the model
2. State estimation: finding the most probable sequence of hidden states that could generate a particular output sequence, given the parameters of the model
3. Identification: finding the model parameter values maximising the probability of a given output sequence or set of sequences.[12, 13]

Rabiner and Juang [12] describe solutions to all three. The first can be solved analytically by the *forward–backward algorithm*. The second can be addressed computationally in a number of ways, one the *Viterbi algorithm* [12,]. The identification problem is the most difficult of the three, often attacked by an iterative method, the *Baum-Welch method* [12]. The computational complexity of both the forward-backward and Viterbi algorithms, for a model with N possible states and an output sequence of length P, is $O(N^2P)$ [12]. The complexity of the Baum-Welch algorithm for W observation sequences of length P is $O(WN^2P)$, if in both cases a transition can occur between any two states, i.e. if the state transition matrix is full. In practice, many entries in the matrix will be zero, reducing the number of calculations to update the model. This factor will be important in the example later.

HMMs are widely used in signal-processing applications such as modelling digital communication channels, speech modelling, isolated word recognition and cryptology. Their main potential advantages in comparison to BNs for modelling managed environmental systems are:

- a basis in temporal relationships
- consequent ability to accommodate feedback loops
- clear distinction between dynamical and observation processes

- a very simple and uniform conceptual structure.

This last property is also their main potential disadvantage compared to BNs. The CPTs in a BN describe local relations, whereas in an HMM successive values of all the variables making up the state are related by the state transition probability matrix (TPM). It thus has the same number of rows and columns as the number, usually large, of possible discrete state values. However, many entries will be zero, because the corresponding state transitions cannot occur. Moreover, the state, as in state-variable models, need only involve those variables with dynamics. Variables whose cause-effect relations can be regarded as instantaneous follow from the state variables by parent-child links just like those in a static BN. A crucial question is how much excluding instantaneous relations from the TPM reduces the complexity of the model. Ultimately the test of an HMM is whether it can model the system, including the important dynamics and feedback, with as little complexity as possible. Because there is no redundancy among the state variables and the Markov assumption has been invoked to minimise the number of past values needed, the answer should generally be yes, provided all inessential state-to-state links are left out.

The application example below illustrates the process of developing a probabilistic model and brings out some factors affecting choices in that process. The model's main purpose is to predict the behaviour of a particular variable. Explanation of the behaviour of the rest of the system, while of interest, is secondary. As a result, simplifications of the model which merge variables may be permissible.

3. Application example

3.1 Selection of modelling approaches for trial example

For Landscape Logic, the modelling approach must be suited to probabilistic information which ranges greatly in degree of detail. It should also make good use of knowledge of the processes involved, without assuming that the specific forms of equations are known for all relations. This may exclude state-variable modelling, which does need such knowledge, as differential or difference equations with known structures, although some uncertainties can be lumped into the process noise and observation noise. However, state-variable modelling treats uncertainty in quite a restricted way, through means and covariances (or analogously through bounds [21]). Probability density functions are then obtainable only if uniquely determined by means and covariances (as are Gaussian or uniform densities, for instance).

Fuzzy modelling is superficially attractive when data are very limited or of doubtful quality, especially when some of the knowledge exists as verbal rules. However, fuzzy modelling is merely heuristic in how it quantifies and combines information, and it has little safeguard against errors in subjective judgement. Together with the drawbacks listed at the end of Section 2.3, this is sufficient to rule fuzzy modelling out for Landscape Logic.

Bayesian Networks and Hidden Markov Models remain. The next subsections compare a BN and an HMM which relate cyanobacterial (blue-green algae) blooming to causative factors such as nutrient and light availability and actions taken as a result of monitored water quality. The focus will be on how complex the models and their updating are and what sorts of information they can yield. Their strengths and weaknesses will be compared and conclusions drawn about their usefulness for modelling the types of situations expected in Landscape Logic. The example is not very complicated but is enough to bring out general points about the practical use of BNs and HMMs.

3.2 Cyanobacterial bloom modelling

Cyanobacterial blooms are a serious water-quality hazard because of the cyanotoxins they produce. Prediction of the likelihood of a bloom is desirable as an aid to management action. Management actions to avoid cyanobacterial blooms include reducing phosphorus level, by controlling runoff of fertiliser and waste, and aerating the water, effective since cyanobacteria bloom in warm, calm conditions and do not do well in agitated water [22].

The purpose of the model is to predict the occurrence of cyanobacterial blooms, rather than explain how and why blooms occur. Having a predictive rather than explanatory purpose allows the model structure to be simpler. Even so, predicting if and when a bloom will occur is made difficult by incomplete understanding of the many factors which influence cyanobacterial growth, including:

- light, critical since cyanobacteria are photoautotrophic [23]. Light intensity depends on day length, water turbidity and flow
- phosphorus, a crucial nutrient of cyanobacteria [24]
- temperature, which affects the rate of processes such as photosynthesis [23]; the growth rate of most cyanobacteria is highest above 25°C [25]

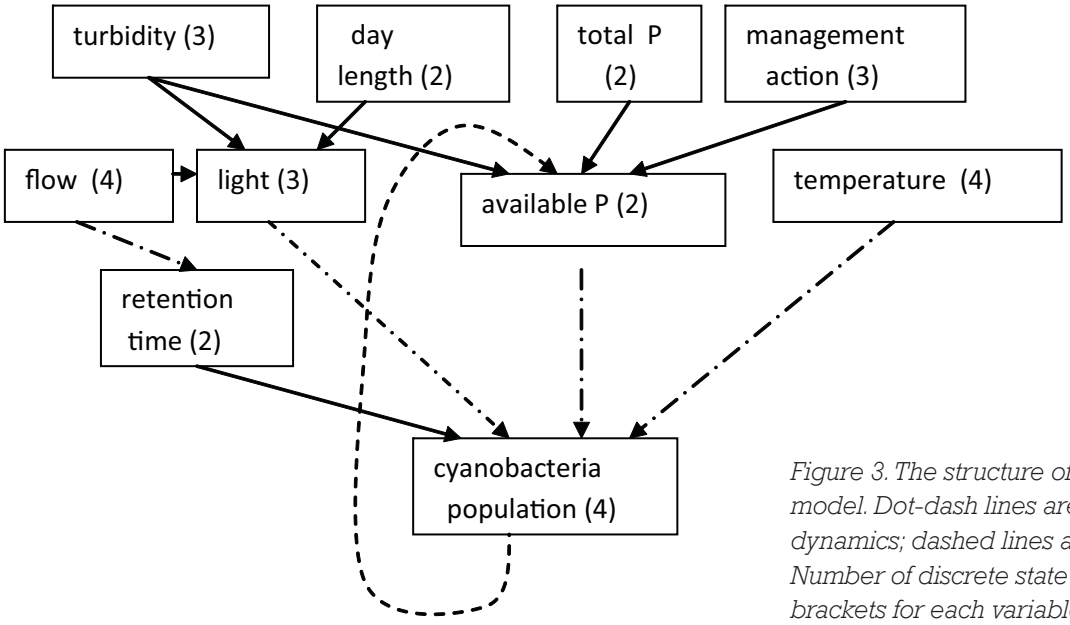


Figure 3. The structure of the BN model. Dot-dash lines are links with dynamics; dashed lines are feedbacks. Number of discrete state values is in brackets for each variable.

- retention time of water in the water body where the cyanobacteria grow. A long retention time is required for bloom formation, since growth rate tends to be low [25].

Two questions thus arise in comparing BNs and HMMs in this example: can the model tolerate considerable ignorance, and how much can it predict about a bloom?

3.3 BN model for cyanobacterial blooms

The BN model is based on that developed by Pollino and Webb [24] and Webb et al. [26] to model cyanobacterial populations in Bourke Weir, NSW, a four-kilometre, narrow, shallow impoundment on the Darling River with high phosphorus concentration and turbidity. The structure of the model is shown in Figure 3, with the numbers of possible values of the variables as in [24].

In its basic form [24] the BN model treats the dynamical links as static, relating variables measured at single, fixed times or averaged over given intervals (not necessarily the same for all). It also ignores the feedbacks. Figure 3 shows several additions and alterations to the model described in [24]. The "management action" node covers action in response to water-quality measurements, to reduce nutrient by controlling runoff from sources such as fertilised fields. Flow out of the impoundment might also be controlled but is not considered, to avoid too much complication. Making management action responsive to measurement imposes feedback, closing a unidirectional loop. Another feedback loop results if the influence of algal population growth on the amount of phosphorus remaining to nourish further growth is included. The effect of flow on turbidity has also been shown in Figure 3, turning turbidity from a root node into a child node. It does not establish a feedback loop, as turbidity has no influence on flow.

The numbers of possible values of the variables indicated in Figure 3 are about the smallest which make sense; a case could easily be made for increasing them. They give the following sizes of the conditional probability tables in the static version, ignoring for the moment dynamics, feedback and the flow-turbidity link:

- retention time, light, available P and temperature to population: 4 x 2 x 3 x 2 x 4 = 192
- flow to retention time: 2 x 4 = 8
- flow, turbidity and day length to light: 3 x 4 x 3 x 2 = 72
- turbidity, total P and management action to available P: 2 x 3 x 2 x 3 = 36.

Total number of CPT entries = 308.

Many of the CPT entries will be small, as the corresponding combinations of values of variables are improbable. However, blooms happen when conditions are unusual, so some care is needed if a parent probability entry is to be rounded to zero, removing all child probabilities conditioned on that value.

The number of CPT entries might also be reducible by exploiting knowledge of how the effects of more than one cause combine. For instance, the combined influence of light and available P on growth might be indicated via a node "P and light", with state quantised to, say, 3 values, as in [24]. The result would be to reduce the size of the CPT yielding population probabilities from 192 to 4 x 2 x 3 x 4 = 96 and add a CPT table of size 3 x 3 x 2 = 18 for the relation between the combined variable and light and available P. This reduces the total number of CPT entries by 78, about 25%. It illustrated how numerical data demands may be reduced, paradoxically, by introducing an extra node to reduce "fan in", i.e. the number of immediate causes of an effect. There is a price to be paid in explanatory power, as the model now only describes the combined effects of causes whose effects were formerly separated.

The message is unsurprising: explanatory detail must be traded against data demands. However, this example is also a reminder that in a BN the total size of the CPTs (and thus their data demands and the effort of updating variables' probabilities through them) rises not simply with total numbers of nodes and arcs, but rather with the number of parents of each child node. This may be important in deciding how to model dynamics, where similar overall behaviour can be reproduced by sub-models with differing structures, as shown later.

If the BN is to model the dynamics shown as dot-dash lines in Figure 3, neglecting the feedbacks for the moment, some time-labelled extra nodes will have to be added to enable the BN to be stepped through time, updating the probabilities of variables as measurements arrive. Time-stepping is also necessary if the time-spread consequences of specified input behaviour are to be predicted. The dynamical links in Figure 3 will be considered in turn, with the aim of seeing what state variables are necessary and which variables act as external forcing. The relations between them will initially be written as schematic state equations simply to show what relates to what. The corresponding implementation in a DBN is by a CPT giving probabilities for the time-labelled variable on the left-hand side for every combination of values of the time-labelled variables on the right-hand side. The other, non-dynamical relations are treated by CPTs just as in an ordinary BN.

The *light*, *available P* and temperature variables

influence *population* through growth rate, so those links are describable by a schematic, scalar, discrete-time state equation:

population xp,k at sample time k = xp,k-1 + fn(light, available P, temperature, all over interval from (k-1)T to kT, and retention time) (6)

Notice that the presence of x_p on both sides of (6) implies feedback from population to itself; this raises no difficulty in updating, where the time delay of T between the two population values separates them into two distinct entities. The same applies to every state variable. It is not obvious whether *retention time* affects *population* through some dynamics or instantaneously; population at exit is the result of the integrated growth rate over the retention time and might thus be related to retention time at the same instant (non-dynamically). For simplicity the relation will be taken as instantaneous. However, *retention time* is a state variable, say $x_{r,k}$ at sample instant k, since it is related to the external input *flow* by some dynamics, discussed later, which will determine how it is to be included in (6). Of the other parent variables of *population, temperature* is an external input, so its role in (6) is as forcing, say $u_{e,k-1}$, with probabilities or actual value supplied as data. Parent variables *light* and *available P* will be assumed to have instantaneous effects on the growth rate, so (6) need only contain their values $z_{l,k-1}$, $z_{a,k-1}$ for the interval $(k\text{-}1)T$ to kT. They are both influenced by *turbidity*. In this example they will be assumed to be affected instantaneously by *turbidity* (with the management action loop not considered as yet), although a more detailed model might attempt to account for the mixing properties and composition of sediment by adding dynamics between *turbidity* and *available P*.

The physics underlying the dynamical relation between flow and turbidity (omitted in [24]) leads one to expect that turbidity depends strongly on flow at the same time but also on the recent flow history. In other words, the forcing variable *flow*, denoted by $u_{f,k-1}$ for the interval between sample instants k-1 and k, affects *turbidity* through some dynamics, so *turbidity* is a state variable, say $x_{t,k}$ at time kT. Rainfall and river flow rate (and hence upstream stage) over some period have large effects, some cumulative, on sediment generation and transport. Rainfall measurements are not included in the model, and flow is mediated mostly by the management of a barrage. Turbidity is affected by things such as whether the flow is rising or receding and how rapidly it has been changing. In a discrete-time model these factors are conveyed by the flow values at recent sampling instants, say m daily values including the latest. We thus seem to need m root nodes for flow. However, if m is not small and the dependence of turbidity on those m values varies in a fairly simple way with time lag, it will normally be more economical to write current turbidity in terms of a small number of recent turbidity values and fewer than m flow values. That is, an autoregressive-moving average (ARMA) sub-model structure will require fewer terms than a purely moving average one. Typically, second-order ARMA models have enough flexibility to approximate fairly complicated dynamics well enough, so two successive turbidity values, at sample instants k–1 and k, would be selected as state variables $xt_{1,k}$, $x_{t2,k}$ at sample instant k, yielding two scalar state equations of the form:

$$\left.\begin{aligned} x_{t1,k} &= x_{t2,k-1} \\ x_{t2,k} &= fn(x_{t2,k-1}, x_{t1,k-1}, u_{f,k-1}) \end{aligned}\right\} \quad (7)$$

with the forcing being one (as in (7)) or perhaps two successive flow values.

Generalising this, an mth-order, one-input, one-output, ARMA-structure model of dynamics requires at most m successive values of the scalar dependent variable as nodes, m-1 arcs to interconnect them (as in the first equation of (7)) and a maximum of $2m$–1 arcs connecting the first m–1 of them and the forcing to the last of them (as in the second equation of (7)).

The dynamics giving retention time from flow are less easy to sort out. Before it is discretised, retention time is the time over which *time integral of flow = storage volume*. On the assumption that retention times $x_{r,k-1}$,$x_{r,k}$ at sample instants k-1 and k are both between $(p\text{–}1)T$ and pT, it turns out that:

$$x_{r,k} = fn(x_{r,k-1}, u_{f,k-1}, u_{f,k-p}, u_{f,k-p-1}, p) \quad (8)$$

where $u_{f,k-i}$ is flow in the interval between $(k\text{-}i)T$ and $(k\text{-}i+1)T$. If $x_{r,k-1}$,$x_{r,k}$ differ, the function on the right-hand side of (8) is modified slightly. In all, the state equation for x_r is a set of equations, (8) and its counterparts for all possible changes in retention time between times k-1 and k. It is tempting to avoid this complication by taking out the retention time node and taking its effect to be that of a fixed number of successive flow values.

Next the two feedback loops in Figure 3 have to be modelled. A mass balance for available P over the interval $(k\text{-}1)T$ to kT gives the change as due to net inflow and change in P incorporated in the population. This yields a first-order difference equation in state variable x_a (called z_a when we hadn't yet considered its status as a state variable). The equation is forced by an external variable, total P, denoted by u_p, say, by population x_p, and by management action. The response to management action may have dynamics, not considered here, reflecting the time-spread effects of the action on P inflow. The dynamics from population to available

P are handled by including $x_{p,k-1}$ on the right-hand side of the state equation for $x_{a,k}$. There is no need for an extra node, as $x_{p,k-1}$ is already a node as a result of appearing in the state equation (6) for $x_{p,k}$. Modelling the management action as affected by population assumes that the action has a consistent policy, expressible through probabilities in a CPT or even as a deterministic relation. The alternative is to treat the action as external forcing, independent of the other variables, "open-loop control" as in [23]. That loses the opportunity to investigate the efficacy of basing management action on monitoring its results, "closed-loop control". A natural assumption when the extent, but not the nature, of management action is varied according to its results, but with some delay, is that the value $x_{m,k}$ applied from time k is determined by adjusting $x_{m,k-1}$ by some amount depending on the observed population value $x_{p,k-1}$ or a more readily measured water-quality variable. In other words, such a policy is described by a state equation in x_m.

The loop is described by difference equations (the state equations) as a consequence of the time delays around the loop. [In practice dynamics, and hence delays, arise in three relations: in taking action in response to change in population, in that action affecting available P, and in available P influencing population. A judgement has to be made, as above, how much of the dynamics to model as such, and how much to treat as instantaneous].

With successive values of the state variables occupying separate nodes, there is no difficulty in updating on receipt of new information. This is in contrast to a static BN, which does not distinguish time-separated values of the same variable and is thus floored by a loop.

This discussion of what variables are needed to handle the dynamics has appealed to the ideas of state and state equations as defined in Section 2.2. However, the variables are described by their discrete probability distributions and the relations between them by CPTs. HMMs employ exactly the same ideas but go further towards state-variable modelling by standardisng the model structure in a way very reminiscent of state-variable modelling.

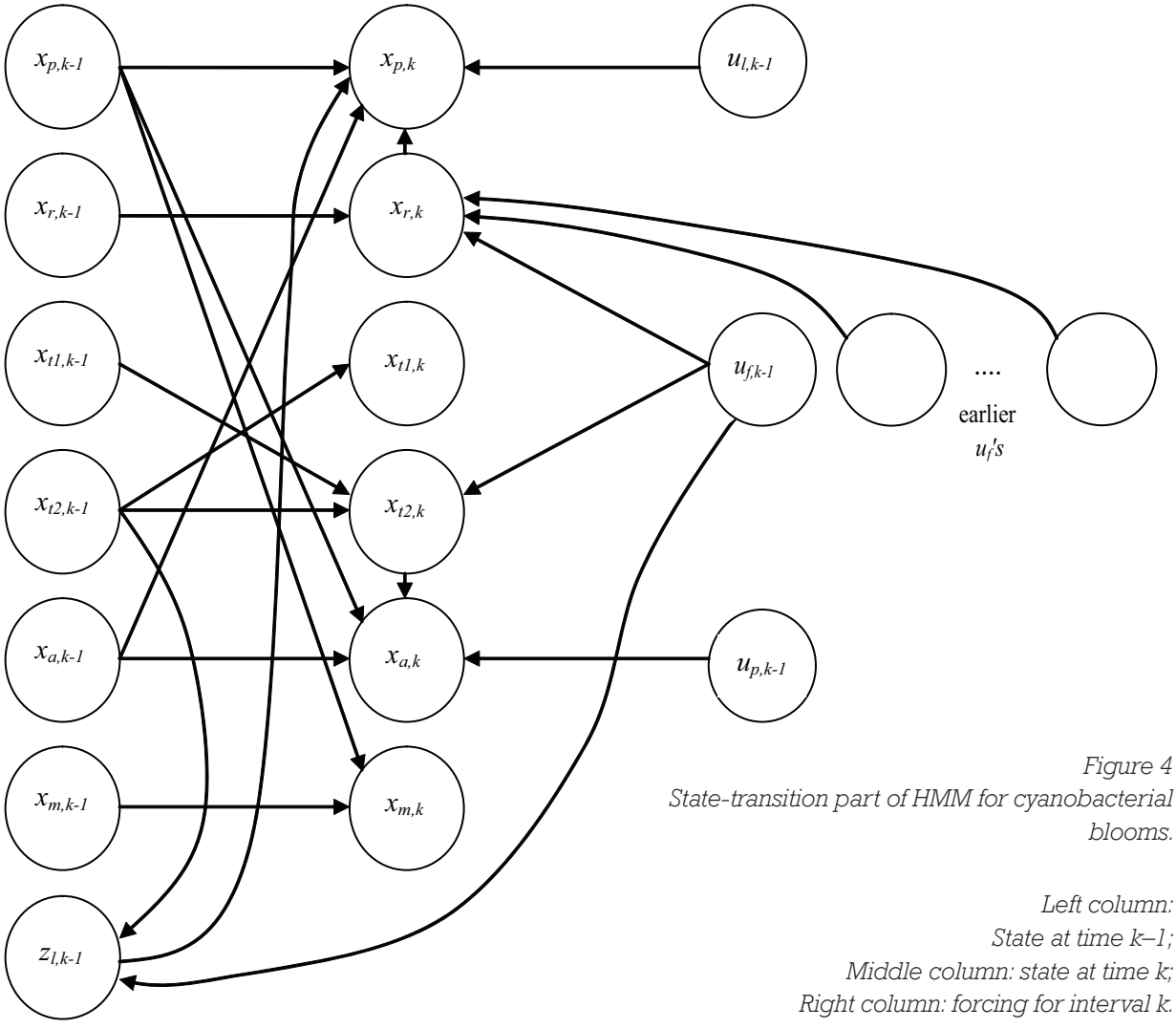


Figure 4
State-transition part of HMM for cyanobacterial blooms.

Left column:
State at time k–1;
Middle column: state at time k;
Right column: forcing for interval k.

3.4 HMM for cyanobacterial blooms

The structure of the state-transition npart of the HMM model for cyanobacterial blooms is shown in Figure 4, where the notation and assumptions are as in Section 3.3. The variables that define the state of the HMM have been chosen to be the same as those in the BN model, to maintain physical interpretability and allow comparison. The state vector at sample instant k is thus

$$\mathbf{x}_k \equiv \begin{bmatrix} x_{p,k} & x_{r,k} & x_{t1,k} & x_{t2,k} & x_{a,k} & x_{m,k} \end{bmatrix}^T \quad (9)$$

There are a few oddities in Figure 4. First, there is an instantaneous relation between two state variables, retention time x_r and population x_p. This does not conform with the Markov assumption that state at discrete time k depends only on state k-1 and forcing in interval k. State equation (6) can be rewritten to include the influence of $x_{r,k}$ on the right-hand side by using (8), the state equation for x_r, to eliminate $x_{r,k}$, leaving a more complicated state equation for $x_{p,k}$ together with (8). That legalises the model and confirms that the choice of state variables was all right.

However, in deciding in Section 3.1.2 whether to put in an intermediate combined ''P and light'' variable, it became clear that making the ''fan in'' to each child variable small was crucial, to minimise the number of CPT elements. This suggests that an instantaneous relation between $x_{r,k}$ and $x_{p,k}$ plus a relatively simple state equation for $x_{p,k}$, accounting for the influence of $x_{r,k-1}$ via $x_{r,k}$, is preferable to a more complicated state equation for $x_{p,k}$ with $x_{r,k-1}$ appearing explicitly. That is, it may be worth modifying the standard HMM state-transition structure by adjoining any instantaneous relations between state variables, wherever that reduces the total number of conditional probabilities to be supplied.

The same comments apply to the instantaneous link from x_{tl} to x_a in Figure 4.

A second feature of Figure 4 which demands thought is the presence of an intermediate variable $z_{l,k-1}$ (light) between $x_{t2,k-1}$ (turbidity), $u_{f,k-1}$ (flow) and $x_{p,k}$. As $z_{l,k-1}$ is instantaneously related to $x_{t2,k-1}$ and $u_{f,k-1}$, it is not a state variable and could be eliminated. It should, however, be retained as an argument of the state equation for $x_{p,k}$, both for physical interpretability and for the same reason as the ''P and light'' variable in Section 3.1.2 and $x_{r,k}$ in the previous paragraph: to minimise the total number of CPT elements.

The third oddity in Figure 4 is the presence of u_f at more than one sample instant in the forcing of x_r. This is not illegal, as no assumptions are made in HMM modelling about the time structure or independence of known forcing variables. The question does arise, though, whether multiple appearance of the same physical variable is economical. Assuming that the structure of the model of the dynamics between u_f and x_r cannot be simplified as was done for turbidity in Section 3.1.2, all the sample-instant values of u_f have to appear. Either (i) they are all regarded as elements of a forcing vector, the value of which is renewed at each sample instant, subject to the constraint that a value which appeared earlier as one element stays the same when it reappears as another, or (ii) the earliest sample is specified as scalar forcing and the later ones are generated as pseudo-state variables related to it by state equations which merely successively move the variable one sample interval forward in time and do not require CPTs. There is no difference in computing load but the latter is tidier.

The lessons from these anomalies with respect to the usual model structure are that:

- they can be removed, and the whole model put into standard HMM form, by eliminating intermediate variables and being flexible in defining state variables
- such tidying-up is a bad thing if it removes physically interpretable variables and/or produces more elaborate relations and thence larger ''fan in'' and more CPT elements.

The observation part of the HMM is simple. The observed symbols are the quantised and imprecise values of cyanobacteria population. Population is a state variable, so the observation process might be modelled without an observation probability matrix as: $y_k = x_{p,k}$ (10)

To do this, any uncertainty in the measurement process must be included as additional spread of the probability distributions conveyed by the CPTs for $x_{p,k}$. If, on the other hand, the consequences of better or worse monitoring are of interest, the quality of monitoring can be specified by a separate observation probability matrix.

4. Construction of BNs with dynamics and feedback

Introduction

A list of steps in constructing BN models of systems with dynamics and/or feedback will be given, with details of the thinking behind each step. The list is not claimed to be exhaustive or to prescribe the "best" way to go about the modelling. The advice it offers must be interpreted according to the purposes and limitations of the modelling. Its aim is to help in systematically constructing a model which does not oversimplify yet is no more complicated than necessary. Provision of data, elicitation of knowledge and calibration are considered only briefly. Model-structure features affecting complexity and data needs are discussed, however. This section concentrates on aspects which require more thought than might be obvious at first, and on factors which only arise when dynamics and feedback have to be modelled.

To help in seeing how to model dynamics economically, some of the ideas underlying two model types outlined in earlier section of this report, state-space models (SSMs) [11] and Hidden Markov Models (HMMs) [16,18], are used. Both sorts of model are simple in general structure and can represent a wide range of system behaviour. Previous acquaintance with such models isn't necessary to read this section.

Steps in model construction

Each major step in model construction is described below. Details appear in square brackets and examples in italics; they aren't essential to the sense.

It's worth stressing that constructing a model isn't really a linear procedure, going once through a sequence of steps. Often an earlier step will have to be revisited as the draft model proves too elaborate or too coarse, or as new information comes to hand from discussions, field tests or literature search. Sensitivity analysis [27] may help in assessing whether the model has become over-elaborate but is too large a topic to cover here. Reference [28] takes a less specialised look at the steps in developing a mathematical model of a dynamical system.

1. Choose the physical variables

This may require quite a bit of thought and judgement.

First, the limits of the model have to be set. Of the processes believed to occur in the system being modelled, we

- decide to leave out those which don't look important for the intended use of the model,
- exclude others by treating their outputs as model inputs, with definite values specified by a scenario or with a range of possible values, and
- regard others still as contributing to the uncertainty in model variables without modelling them explicitly.

Example

Imagine we are developing a BN model to predict stream-flow (an example we'll use many times). Stream-flow depends on rainfall in the catchment, evapo-transpiration, interception by vegetation and antecedent soil moisture. We may decide to

(i) regard potential evapo-transpiration as a known (perhaps uncertain) input, rather than modelling how it relates to temperature, wind speed and humidity, and

(ii) omit interception, because we know too little to estimate it, while admitting that leaving it out adds to uncertainty in the effective rainfall.

Second, we must decide on the degree of detail. We can model every stage in a cause-effect chain individually or lump them together, omitting the intermediate variables. Similarly, when a cause affects a variable by more than one route, we may choose to combine the parallel cause-effect links, again leaving out intermediate variables. Although such combining simplifies the model, it loses variables which might be useful for checking the credibility of the model or even calibrating it, so a compromise will often have to be made. Moreover, insight into the processes determining intermediate variables may help in selecting model structure, even when those variables do not appear explicitly.

Example

We can either model soil moisture explicitly or combine the sub-model predicting it with the sub-model relating it to stream-flow, yielding a single model in which soil moisture does not appear. As measurements of soil moisture are rarely available, omitting it seems sensible. On the other hand, if we don't consider it at all, we throw away any knowledge of what determines soil moisture, which might have suggested part of the model structure. The rate of change of soil moisture varies (up to saturation or complete dryness) through deficit or excess in a water balance, suggesting at least a first-order dynamical relation between the model inputs and soil moisture. [See items 2 and 4 for discussions of dynamics and model order]. Whether these dynamics need modelling depends on factors such as the time scale of model operation and whether the catchment is saturated for much of the time. In addition, the processes

generating stream-flow include quick flow such as overland flow, and much slower flow through infiltration. We would therefore expect two (or conceivably more) routes for flow, operating in parallel, which we can choose either to model separately or to combine. Often combining them will make sense, as we cannot measure either component of flow and have little prior knowledge of their relative sizes. That said, knowing that there are two flow components suggests that a second-order sub-model is appropriate for the dynamical response of stream-flow to effective rainfall.

2. Decide which parts of the model are static and which dynamical

What distinguishes a dynamical from a static model or sub-model is that the effects of a change in a variable are time-spread, not instantaneous changes to new values. Conversely, dynamics imply that the value of an affected variable at any instant depend on the previous history of the variables influencing it dynamically, not merely on their present values. Plainly a dynamical BN has to include values of some variables over some period of time. SSMs and HMMs suggest how to handle dynamics in BNs, through ideas covered in item 3 and later. Their structures are motivated largely by the wish to deal with time-spread effects as simply as possible, minimising the amount of history that has to appear in the model.

We aim to model no more than necessary as dynamical. Several practical questions concern what has to be modelled in that way:

(i) Can we afford to aggregate inputs and outputs over a period long enough for the dynamics not to show?
Example
If we are interested in water availability over a a period of some months, the dynamics describing how each day's rainfall affects the next few days' stream-flow can probably be ignored. All that matters is the model's mean stream-flow per unit rainfall over the period (influenced by initial soil moisture and mean evapo-transpiration as other inputs). On the other hand, if we want to predict daily stream-flow, we cannot ignore the dynamics.

(ii) Is the time scale of interest short enough for some of the variables subject to dynamics to be treated as constant in each period?
Example
In modelling the response of a small catchment to a short rain event, we can ignore the dynamics of seasonal variation in interception. We may also be able to treat evapo-transpiration rate as constant. If the flow peak is the main interest, we may be able to omit the dynamics of the slow flow component.

(iii) Are steady-state modelled outcomes, for instance of an NRM action, much more important than the transients on the way to them? If so, the dynamics can be ignored.
Example
Change in water yield resulting from revegetation. Depending on the nature of the vegetation and the time scale of interest, the long-term dynamics of growth, and hence transpiration, may have to be modelled even though the rainfall-runoff dynamics need not. This leads to a more refined question:

(iv) Are some of the cause-effect relations so much quicker than the rest that they can be treated as instantaneous? The answer is very often yes in environmental systems.

All these questions amount to asking which of the dynamics dominate (if any do) on the time scale of interest; only those dynamics need be modelled.

3. Identify the role of each variable

The roles are as **input**, **output**, **state or intermediate variables**. Each has a specific, defined meaning, so to avoid confusion the words can't be used casually in other senses. One of the most important features of HMMs and SSMs is that they clearly distinguish input, output and state variables. [They don't use intermediate variables].

Inputs are forcing variables, through which the outside world influences the rest of the model. They come in two sorts: those which can be assigned values and those which cannot. The latter have to be treated, as already mentioned, as just contributing to the uncertainty in the variables they affect, and do not appear explicitly in the model. Each quantifiable input appears either with a certain value, measured or specified, or with uncertainty described by the probabilities of taking each of a range of possible values.

Outputs are the variables which the model is intended to predict and which will be scrutinised. They need not be physically outputs of the system.

Example

Stream-flow at various points in a stream network may be of interest, not only the farthest downstream. Less obviously, the modelled soil moisture or groundwater exchanges may be model outputs although they are inputs to the remainder of the model determining stream-flow.

The other significant thing about an output is whether it can be measured (accurately or not), *i.e.* whether it provides information which can be processed by Bayes' rule to improve knowledge of the variables affecting it and/or affected by it.

The role of **state variables** is to model

dynamics. The idea of state arises in SSMs and HMMs. SSMs and HMMs employ the Markov assumption that there exists a collection of variables, the state variables jointly making up the state vector (or simply ''state''), with the nice property that their future behaviour is completely fixed by their present values and the future behaviour of the inputs. In other words, it isn't necessary to keep in the model any history of any variables, because the present state forms a full set of initial conditions for the future. It's quite surprising that this is possible for a system with a time-spread response to input values at any given instant, so here's a simple example.

Example

A system has dynamics and its variables are sampled at time intervals of 1, starting at time 0 and taking all earlier values as zero (i.e. ignoring them). Its immediate future output y_{k+1} *at time k+1 is related to the time series* $u_0, u_1, u_2, \ldots, u_k$ *of its input u by:*

$y_{k+1}=0.5u_k+0.4u_{k-1}+0.32u_{k-2}+\ldots 0.5(0.8)^r u_{k-r}+\ldots 0.5(0.8)^k u_0, k\geq 0.$

[Thus the influence of each previous input value dies away by 20% over one time interval]. It looks as if you have to use the entire history of the input to find the output at time k. On the contrary, exactly the same behaviour results if we rewrite the model as the state equation:

$y_{k+1}=0.8y_k+0.5u_k, k\geq 0$ *with* $y_0=0.$

Here y is a state variable (the only one needed in this instance), and y_k *contains all essential information about the influence of* $u_0, u_1, u_2, \ldots, u_{k-1}$ *on* y_{k+1} *and, through it, all later values of y.*

Note

The state variables are as defined above, *not* all the variables, an arbitrarily chosen collection of interesting variables or the variables thought to be important, as often found in common usage of the words.

The significance of the state-variable idea for BNs is that, as in this little example, dynamics can be represented by a state equation with state now and inputs now as the only variables on the right-hand side (causes) and state at the end of the next time interval on the left (effect). We use it to step forward in intervals, feeding in the input values as we go. At any time the BN need only include variables at two instants (the current state and input and the next state), not a whole history.

It's important to remember that the use of state variables relies on the Markov assumption, which is not universally valid. For instance, if an effect *y(t)* is related to cause *u(t)* and *y(t-T)*, in other words there is a delayed feedback, future *y* depends on the past behaviour of *y* over the whole interval of the delay. If there were no restrictions on *y*, that would entail knowing an infinite number of values of *y* in the interval. In practice, it is enough to know a finite, perhaps small, number of values. However, extra state variables have to be introduced to embody this information; this is discussed further in step 7.

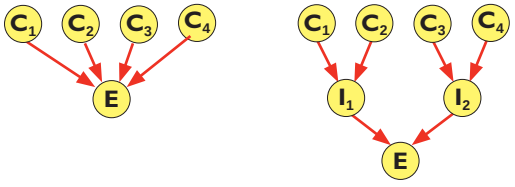
When the pure delay *T* occurs at the input or output, rather than within a feedback loop, we need only replace *u(t)* by *u(t-T)* or *y(t)* by *y(t+T)* in their relations with state at time *t*. The only complication arises when *T* depends on another variable, often as a transport delay varying with a flow rate. In these cases, *T* will itself be a variable.

Both SSMs and HMMs consist of **state equations** and **observation equations** linking state to outputs. The observation equations are static (non-dynamical) and raise no new issues in a BN.

Intermediate variables are all those which aren't inputs, outputs or state variables. They aren't essential parts of the model, in the sense that the model can relate the inputs via the state variables to the outputs without including them explicitly. They don't appear in SSMs or HMMs. Nevertheless, we may include them because they are physical variables which help to interpret the model or because their presence reduces fan-in (the number of variables directly affecting a given variable) and hence the complexity of the relations in the model.

Example

Fig. 5(a) shows relations between four causes, each with C possible values, and a single effect with E possible values. The probabilities of the values of the effect variable are given by those of the cause variables passed through a CPT with EC^4 *entries. Figure 5(b) shows the same relation in more detail, with two intermediate variables capable of taking any of I values. Now we need two CPTs each with* IC^2 *entries plkus one with* EI^2 *entries. Usually* $2IC^2 + EI^2 << EC^4$*, so introducing the intermediate variables reduces the overall complexity of the CPTs greatly. For example, with* $C = E = I = 5$, $2IC^2 + EI^2 = 375$ *and* $EC^4 = 3125$. *For* $C = 3, E = 2$ *and* $I = 3$, $2IC^2 + EI^2 = 72$ *and* $EC^4 = 162$. *However, note that for* $C = 2, E = 3$ *and* $I = 5$, *a very unfavourable case for including the intermediate variables (but perhaps not very likely in that* $I < E$*),* $2IC^2 + EI^2 = 115$ *and* $EC^4 = 48$.



(a) (b)

Figure 5. Introduction of intermediate variables

4. Add any extra state variables necessitated by dynamics

As illustrated by the state-equation example above, a single state variable may suffice for simple dynamics linking a single input to a single affected variable. However, in many cases the modelled dynamics are more complicated.

Example

If both quick and slow components of flow response to effective rainfall have to be considered in a rainfall-runoff model, the underlying relation is a second-order differential equation giving the second derivative of stream-flow in terms of its present value, its first derivative and the effective rainfall. Such an equation can be rewritten as a pair of first-order equations: $\ddot{z} = g(z, \dot{z}, u)$ *can be rewritten as:*

$$\left.\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g(x_1, x_2, u) \end{aligned}\right\} \text{ with } x_1 \equiv z \,.$$

[In practice, a discrete-time version with a sampling interval of an hour or a day is used].

Dynamics described by an *n*th-order differential equation can be written as *n* first-order differential equations and require *n* state variables. Overall, the number of state variables required in a model is the total number of initial conditions that must be given to allow the differential equations describing the dynamics to be solved (in theory; in practice in a BN, they are replaced by first-order, discrete-time, probabilistic relations as outlined in steps 8 to 12). Some will be physical variables, while others, as in this example, will be their derivatives or related to them. The uncertainties motivating use of a BN model may make it hard to estimate the orders of the dynamical relations. In deciding how many and which state variables to incorporate; trial and error may be needed, assessing the credibility of BN outputs and comparing them with measurements where possible.

It's sometimes handy to know that we have considerable freedom of choice in what variables to use as the state. Roughly speaking, if a particular collection of *n* variables is valid as state, so is any set of *n* functions of those variables from which the original variables can be found uniquely. This freedom often allows the relations between the state variables in the state equations to be simplified, but almost always at the cost of greater complexity in the input-to-state and state-to-output relations. For example, the state equations can generally be rearranged into decoupled form, where the future of each state variable depends on the current value of that variable alone among the state variables, but also on more complicated combinations of input variables. Although there may be instances where an opportunity for decoupling in a BN can be noticed *ad hoc*, it can only be carried out systematically if the coefficients in the state equations are all known exactly, which is not so in BNs.

5. Decide whether the model is continuous-time or discrete-time

In BN models, each set of causes and their effect are related through a conditional probability table with an entry for each possible combination of discrete values of cause and effect. This is easily extended to handle cause at one time and effect at another, and can thus deal with dynamics in principle. However, many if not all of the relations modelled are in reality a collection of rate relations for continuous-valued, continuous-time variables (corresponding to differential state equations), instantaneous state-to-output relations (corresponding to algebraic observation equations) and instantaneous relations for any intermediate variables. In practice, computation of the model output has to use values of the input, state and intermediate variables at particular instants, usually regularly spaced (by a fixed sampling interval), and any measurements will also be regularly spaced in time. We can turn a continuous-time, differential- and algebraic-equation model into a discrete-time model by integrating the differential equations over one sampling interval, giving the state vector at the end of the interval in terms of the forcing during the interval and the state at its start.

[If we denote the state at instant *k* by column vector:

$$\mathbf{x}_k \equiv \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}_k^T ,$$

the forcing by:

$$\mathbf{u}_k \equiv \begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix}_k^T$$

and the sampling interval by t, we integrate the continuous-time model:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u})$$

to get a relation of the form:

$$\mathbf{x}_{k+1} \equiv \mathbf{x}((k+1)\tau) = \mathbf{x}_k + \int_{k\tau}^{(k+1)\tau} \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))\, dt \cong \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad]$$

Even if the model were to be kept as continuous-time, integrating its differential equations would almost always (except when analytical solution is

possible) involve computation over a succession of finite time intervals, so in practice the model is discrete-time regardless. We shall therefore always take the model to be discrete-time, knowing that most of the variables are actually continuous-time.

6. Choose time and spatial scales and sampling intervals

The time scales of the model have already had to be considered in deciding what dynamics to include but that's not the end of the story. We also need to ask what information can be provided to the model and what information is to be obtained from it, and ensure that they are consistent with the time intervals chosen.

Too-coarse sampling in time can result in severe loss of information. [This is intuitively obvious but it's worth looking into just how this happens so as to see what it implies for choice of sampling interval (time step) in the model. Consider a time-varying quantity (''signal'') whose frequency content does not change with time (an idealisation to allow easy analysis) and extends only up to a frequency f cycles per unit time. Roughly speaking this means that the shortest significant feature comes and goes in not less than $\frac{1}{2}f$ time units. If the signal is sampled at time intervals of τ, the spectrum of the sampled signal consists of repeats of the 2-sided spectrum of the original, continuous-time signal, spaced in frequency by $1/\tau$ cycles per unit time [11]. The spectrum of the original, unsampled signal is recoverable from that of the sampled signal by low-pass filtering so long as the spectrum replicas do not overlap, i.e. provided $\tau < \frac{1}{2}f$. That is, sampling loses no information (surprisingly) if the sampling interval is less than half a cycle at the highest frequency present. This assumes distortionless, instantaneous sampling and perfect low-pass filtering of the sampled signal, both normally reasonable approximations. More importantly, it also assumes that (i) the information-bearing signal cuts off sharply at frequency f, and (ii) there is no noise present at frequencies above f. If (i) is not true, overlapping of the spectrum replicas at frequencies below f (called aliasing) superimposes information at higher frequencies than f on the information at frequencies below f, making it impossible to unscramble the two. If (ii) is not true, higher-frequency noise components add to those below frequency f, worsening the signal-to-noise ratio].

We conclude that at a given size τ for the model's time step, we can only get information about the components of the sampled quantity up to frequency $1/2\tau$, and then only with any information and noise at frequencies above $1/2\tau$ superimposed (aliased) onto it. To put it in less technical terms, when we sample, not only do we lose whatever happens between samples, but also we do not know how much of the sampled values results from rapid and how much from slower variations, and are thus prone to misinterpret any variation we see. This rather subtle point is often overlooked yet is critical to making sense of variables viewed only at intervals in time.

Moreover, aliasing prevents us from learning much about the characteristics of the noise present (measurement errors, effects of unmodelled inputs, ignored effects of modelled inputs, and error resulting from representing distributed and/or time-varying variables by spatial and/or temporal averages). All we can do is minimise aliasing by employing as short a sampling interval as possible. As the proportion of information and, to a lesser extent, noise tends to fall with increasing frequency, these ill effects are reduced at an increasing rate if the sampling interval is reduced.

Reducing the sampling interval introduces another potential problem, of misinterpreting short-term variation due largely to noise as part of a significant trend (as happens all the time in economics and weather forecasting). The problem is amenable to low-pass filtering to extract the slow components of interest. In a BN, the signal is whatever measure of central tendency or spread you choose to summarise the probability distribution of the variable.

7. Account for any pure delays, adding variables as needed

For pure delays at input or output, see the last paragraph of the discussion of state variables in step 3. When the delay T applies to a state variable in the state equation, we must account for the earlier behaviour of state in an interval T long, as discussed in the next-to-last paragraph about state variables in step 3. With the model discrete-time, this is easy so long as T can be approximated by an integer number d of sampling intervals. All that's necessary is to coin d-1 extra state variables, consisting of the original state variable successively delayed by 1, 2,..., d sampling intervals.

Example

The evolution of a state variable x affected by a feedback loop with a total delay of 3 units is described approximately (but well enough) by the discrete-time equation $x_{k+1}=0.8x_k-0.6x_{k-3} + 2u_k$, *where u is a forcing input. Clearly the delayed-state term* $-0.6x_{k-3}$ *can't be ignored, as it's comparable with* $0.8x_k$. *To include it in state equations relating quantities at time k and k+1 only, we define a new set of variables*

$$z_{1,k} \equiv x_k, z_{2,k} \equiv x_{k-1}, z_{3,k} \equiv x_{k-2}, z_{4,k} \equiv x_{k-3} \quad ,$$

giving:

$$\left.\begin{aligned} z_{1,k+1} &= 0.8z_{1,k} - 0.6z_{4,k} + 2u_k \\ z_{2,k+1} &= z_{1,k} \\ z_{3,k+1} &= z_{2,k} \\ z_{4,k+1} &= z_{3,k} \end{aligned}\right\}$$

which looks a bit peculiar but is indeed in the form (state at time k+1) = fn(state at time k and forcing at time k), as required, with the revised state vector:

$$\mathbf{x}_k \equiv \begin{bmatrix} z_{1,k} & z_{2,k} & z_{3,k} & z_{4,k} \end{bmatrix}^T .$$

8. Identify input-state, state-state and state-output actions, and any cause-effect links free of uncertainty

Steps 1 to 3 chose input, state, output and intermediate variables, and later stages may have added more state variables. That done, we can employ prior scientific knowledge, field tests, analysis of records (not covered in this note) and ''expert knowledge'' (subjective estimation with unknown reliability) to improve our initial ideas on what relations are significant enough to be included in the model.

At this stage it's worth checking whether any of the relations have small enough uncertainty to be treated as certain, i.e. unique numerical relations of known algebraic form. Where the relation is one-to-one, it needs no conditional probability table (CPT), of course. In fact, either of the variables can be eliminated, remembering to modify accordingly the argument of the probabilities in the upstream or downstream CPT of the other. If the certain relation is part of a many-causes-to-one-effect relation, it merely modifies the CPT for the rest of the causes and effect.

9. Look for decoupling and weak influences to simplify state equations

Some of the largest CPTs are likely to be those for the relations described by the state equations. There is thus a high premium on keeping those CPTs as small as possible. Their size is fixed by the number of arguments (cause and effect variables) and the number of discrete possible values of each.

Example

If state variable x has 3 possible values, with probabilities of x_{k+1} conditioned on the 3 possible values of x_k, 3 possible values of input u, the 2 of input v and the 2 of input w , the CPT for $p(x_{k+1} | x_k, v_k, w_k)$ has 3 x 36 = 108 entries.

An HMM is the extreme case of simple structure, achieved by treating the whole collection of state variables as one big state variable. The number of possible state values is the product of the numbers of its constituent variables, usually a large number. [This treatment of state is understandable, as HMMs developed mainly in signal processing, where the state consists of the values of a short string of symbols, successive samples of a single digital signal with a finite (and often small) alphabet. The problem is often to decide which were the transmitted values, having received them distorted and affected by noise].

Example

State variables x_1 , x_2 , x_3 have 3 possible values each. They are influenced to varying degrees by the inputs u, v and w as above. Consider three cases:

(i) There is no interaction among the state variables, so their state equations are decoupled and give rise to three CPTs each with 108 entries, a total of 324 entries.

(ii) There is full interaction, so each state variable is affected by all three state variable, and therefore has a CPT with 3 (3^3 x 12) = 27 x 108 = 2916 entries, so in all there are 8748 entries . We hope that many of them will be small enough to call zero.

(iii) The state variables are combined to form one variable with 3^3 = 27 possible values. A single CPT now describes the state equation, and has 27(27 x 12) = 8748 entries, the same total as in (ii). [A little thought shows that this is generally true, as exactly the same range of situations is allowed for].

The example shows that it does not matter whether we treat the state as a collection of separate variables or one composite variable. What is crucial in keeping the number of CPT entries to be supplied small, given the number of possible values of each variable, is minimising both the significant interaction between state variables and the number of inputs significantly influencing each state variable. This is a matter of choosing the state variables wisely, guided by prior knowledge, and of recognising when influences are weak enough to be ignored (given that coarsely discretising the variables has already incurred approximation error).

Example

If in the previous example x_1 is affected only by itself, x_2, u and v; x_2 is affected only by itself, x_3 and v; and x_3 is affected only by itself, u and w, the CPTs have respectively 162, 54 and 54 entries, a total of 270. Here x_3 is decoupled from x_1 and x_2. Four of the 9 possible input-to-state connections, and 4 of the possible 8 state-to-another-state connections, are taken to be

negligible. Deleting 8 of the 18 possible cause-effect links has reduced the number of CPT entries by a factor of over 30.

We have already noted that in a BN model, uncertainty will usually prevent us from simplifying those equations by decoupling, but partially decoupled dynamics are common in environmental (and other) systems.

10. Check complexity of observation equations

Sometimes a state variable is also an output, in which case its observation equation is trivial and has no CPT. More generally, two or more state variables affect an output variable and the size of its CPT is the product of the numbers of possible values of the state variables and output; once the state variables have been chosen, there is no room for manoeuvre to reduce the size of the CPT. However, in choosing the state variables the effects on the complexity of the observation equations, as well as the state equations, must be taken into account.

When the model is to aid management decisions, there may be scope for reducing the number of output values to 2 (corresponding to ''yes/no'' in the decision or 3 (corresponding to ''yes/don't know so get more information/no'') by inserting the decision thresholds.

11. Decide how to treat feedback, adding further state variables if necessary

Recall from step 3 that to handle dynamics a BN has to include state variables at a minimum of two times, the present and the end of the next sampling interval. Also, the presence of delay in the relation between past and future state makes extra state variables necessary, delayed versions of the original one (step 7). Separation in time of samples of the same variable, as between the state now and its next value in each state equation, is enough to make dealing with a feedback loop in a BN feasible. With successive values of the same variable represented as separate nodes, updating variables in a loop does not imply that ''*A* affects *B*, which affects *A*, which affects *B*... so their interaction prevents us from updating them''. With delay, say one sampling interval between B_k and A_{k+1}, A_k affects B_k, which affects A_{k+1}, and that's the end of it until we advance the clock one sampling interval and do it all over again. [The BN is an acyclic directed graph for computation over any one sampling interval]. If the delay is greater than one sampling interval, we coin extra state variables as discussed in step 7, with the mechanism of stepping forward in time unchanged.

At this point you may well ask what happens if the delay is much smaller than one sampling interval. The answer is that however small the delay, it is not zero and so the first instant at which the effect of something happening now can be registered by the model is the next sampling instant. The separation in time between values of the same variable at the start and end of a circuit of a feedback loop referred to above therefore always happens. This is so whether or not the loop has dynamics apart from the delay. If a delay of one sampling interval around the loop is an unacceptable approximation, the trick is to use a sub-multiple of that interval for the part of the computation involving that loop, retaining the original interval for any input of external information (output observations or input changes) and the rest of the model.

The only remaining question specific to modelling a feedback loop is where to put the necessary delay. Usually knowledge of the mechanisms being modelled, such as transport delay in a stream network, will decide this. If not, the location or distribution of the delay only matters if an intermediate variable in the loop interacts with another outside it. In such an instance it might be necessary to guess where the delay goes, see how the model results compare with known behaviour, and adjust the delay allocation by trial and error. The behaviour of the model will be sensitive to the location of the delay in such cases when the dynamics are significant on a time scale comparable with the delay, unsurprisingly. [Control theory shows that a small change in a pure delay in a feedback loop may have a large effect on closed-loop response, even making the difference between stable behaviour and runaway. Most environmental processes are inherently stable (or we shouldn't be here) but there might be applications where delays have to be treated carefully, for instance by augmenting the model with intermediate variables which can be monitored and compared with expectations or measurements].

12. Choose the discrete possible values of the variables

In principle, a BN model can deal with an uncertain relation between a continuous-valued cause *c* and a continuous-valued effect *e*. [Given the probability density function $p(c)$ and the conditional probability density function $p(e|c)$, the joint probability density function $p(c,e)$ is $p(e|c)p(c)$ and $p(e)$ is the integral of $p(c,e)$ over all possible values of *c*. If an inference about *c* is to be made from updated knowledge of *e*, $p(c|e)$ can be found from the prior $p(c)$ by Bayes' rule as $p(e|c)p(c)/p(e)$]. However, it is usually easier to employ discrete values for all variables, with associated probabilities. Discretisation inevitably loses detail but simplifies the application of Bayes' rule;

integration to get $p(e)$ from $p(e|c)p(c)$ becomes summation over the possible values of c, and in fact we need not find the values of $p(e)$ explicitly, as the values of $p(e|c)p(c)$ can be scaled to yield those of $p(c|e)$ merely by summing them and divided by the sum to make the total probability 1. In practice, the computations have to be done for intervals in each of the variables unless (very unusually) the probability density functions are known in algebraic form and are analytically tractable, so we might as well regard the variables as all discrete-valued.

The rule in selecting the discrete values for each variable is simply to use the smallest number that cover the range of interest and are not so widely spaced as to reduce the model's accuracy (affected by the quantisation of the variables) and resolution to the point where its results are useless. This is, of course, a matter of both judgement and trial and error, constrained also in many cases by computing load and limited ability to provide data. A sensitivity analysis is essential if the reliability of the results is to be quantified and the main sources of uncertainty (doubt as well as imprecision) pointed out.

13. Specify the conditional probability tables

Unless there is a large body of observational evidence, some or all of the conditional probabilities in the CPTs are subjective and represent beliefs, in line with Bayesian estimation but demanding scrupulous interpretation of the model results (as beliefs, not probabilities). The procedure for eliciting beliefs has, of course, to be viewed as provisional, subject to revision in the light of comparison of the model's behaviour with other knowledge. It's worth noting that subjective beliefs may be highly unreliable when they concern extremes, because of lack of experience of rare events. Conclusions from a BN (or any other Bayesian, subjective) model about risks associated with extreme behaviour are correspondingly unreliable.

When there is observational evidence, in the form of records, some of the CPT entries may be found by a calibration algorithm based on optimisation of the fit between model and observations. Details appear in the references in [2], which also contains a great deal of other illuminating material on HMMs, SSMs, BNs and connections between them.

5. Conclusions

Conventional state-variable modelling and fuzzy modelling have been rejected for dealing with dynamics and feedback in Landscape Logic cases where a static BN would otherwise be used. The reasons are listed at the ends of Sections 2.2 and 2.3. That said, the ideas underlying state-variable modelling have been found helpful in thinking about DBN and HMM modelling for the algal-bloom example. Specifically,

- exploiting the Markov assumption leads to economical choice of variables in the dynamical parts of the model
- classification of variables as state, forcing or observed throws up intermediate physical variables which could be eliminated but whose inclusion may make the model more economical, as well as more fully interpretable
- partitioning the state into physically distinct segments helps in seeing the sparseness structure of the state-transition matrix, and thence the sizes of the CPTs. Naively squaring the number of possible states overestimates the number of state-transition matrix elements enormously.

DBNS and HMMs do the same job by essentially the same means, and indeed Sections 3.3 and 3.4 show that the considerations in developing either sort of model are largely identical: inclusion, omission and merging of variables, coarseness of quantisation, fan in and CPT size, interpretability and explanatory power versus economy. An HMM can be viewed as a standardised representation of a DBN, which deals with time-stepping economically and makes the role of each variable clear. The example has shown, however, that an algebraically less tidy model may have smaller data demands.

The treatment by HMMs of the state as one composite variable makes sense when its components are all of the same physical type, as for instance in successive data symbols in digital communication or formants in speech modelling. In environmental modelling, by contrast, it is necessary and valuable to retain the separate identities of the variables comprising the state, to show how to keep the model structure economical as mentioned above and to avoid losing interpretability. An exception may be in modelling distributed systems, where it would be worth investigating the possibility of replacing a large number of spatially quantised variables representing a single distributed variable, e.g. a flow field, by a single ''pattern'' variable taking a fairly small number of possible values (perhaps accompanied by an amplitude variable). Introducing spatial dependence otherwise looks infeasible except in the simplest cases.

Introduction of time as an independent variable has been seen, in the example, to increase the number of conditional probabilities to be supplied greatly. Much of the increase is inevitable, as the simplest description of discrete-time dynamics, a first-order difference equation, includes the previous sample of the response (state, child) variable as an extra parent, multiplying the size of the corresponding CPT by the number of possible values of that variable.

Feedback loops pose no problem in an HMM, so long as the total delay around the loop is at least comparable with one sample interval, as then the variables involved can be updated in the normal time-stepping process. A static BN can be used sequentially to track changes via steady states or responses at fixed, pre-specified times. However, its inability to handle feedback makes it unusable for updating the CPTs giving effects of management actions, unless the actions are taken as so infrequent as to open the loop over the response time; this excludes timely management.

It is worth reiterating the limitation of static BN modelling, that the model can only mimic average, steady-state or extreme behaviour of the system, as measured by the data used to calibrate the model. If the model is intended to predict or explain conditions which depend on the dynamics of the responses, omission of the dynamics loses the possibility of gaining understanding of what happens, and makes it improbable that the model can discriminate critical situations well enough to be useful.

One-sentence conclusion: the HMM framework and the ideas of state-variable modelling provide a good basis, but not a rigid recipe, for developing dynamical Bayesian Network models.

References

1. K. B. Korb and A. E. Nicholson (2003) *Bayesian Artificial Intelligence,* Chapman & Hall/CRC Press.
2. K. Murphy (2006) *A Brief Introduction to Graphical Models and Bayesian Networks*, <www.cs.berkeley.edu/~murphyk/Bayes/bayes.html>, 2006.
3. D. R. Bellhouse (2004) The Reverend Thomas Bayes, FRS: a biography to celebrate the tercentenary of his birth, *Statistical Science* 19(1): 3-43.
4. M. E. Borsuk, M. E., Stow, C. A. and K. H. Reckhow (2003) An integrated approach to TMDL development for the Neuse River Estuary using a Bayesian probability network model (Neu-BERN). *Journal of Water Resources Planning and Management* 129: 271-282.
5. M. E. Borsuk, M. E., Stow, C. A. and K. H. Reckhow (2004). A Bayesian network of eutrophication models for synthesis, prediction, and uncertainty analysis. *Ecological Modelling*, 173: 219-239.
6. Pollino CA, White AK, Hart BT. 2007a. Examination of conflicts and improved strategies for the management of an endangered Eucalypt species using Bayesian networks. *Ecological Modelling* 201:37–59.
7. Pollino CA, Woodberry O, Nicholson AE, Korb KB, Hart BT. 2007b. Parameterisation and evaluation of a Bayesian network for use in an ecological risk assessment. *Environmental Modelling & Software* 22:1140-1152.
8. Ticehurst JL, Newham LHT, Rissik D, Letcher RA, Jakeman AJ. 2007. Bayesian Network Approach for Assessing the Sustainability of Coastal Lakes. *Environmental Modelling & Software* 22:1129-1139.
9. Ticehurst, J.L. R.A. Letcher, D. Rissik (in press), Integration modelling and decision support: a case study of the Coastal Lake Assessment and Management (CLAM) tool, *Mathematics and Computers in Simulation.*
10. J. Pearl and S. Russel (2000) Bayesian Networks, UCLA Cognitive Systems Laboratory, Technical Report (R-277), November, 2000. In M.A. Arbib (Ed.), *Handbook of Brain Theory and Neural Networks*, Cambridge, MA: MIT Press, 157-160.
11. R. A. Gabel and R.A. Roberts (1987) *Signals and Linear Systems*, 3rd Ed., Wiley.
12. T. Kailath (1980) *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ.
13. N. J. Gordon, D. J. Salmond, A. F. M. Smith (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proc.* F 140(2):107-113.
14. A. Doucet, J. F. G. de Freitas, N. J. Gordon (Eds.) (2001) *Sequential Monte Carlo Methods in Practice*, Springer, New York.
15. L. Zadeh (1965) Fuzzy sets, Inform. and Control 8: 338–353.
16. L. R. Rabiner and B. H. Juang (1986) An Introduction to Hidden Markov Models, *IEEE ASSP Magazine*, 3(1): 4-16.
17. L. R. Rabiner (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proc. IEEE*, 77(2): 257-286.
18. A. B. Poritz (1988) Hidden Markov Models: A Guided Tour, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1: 7-13.
19. M. Craven and D. Page (2006) *Hidden Markov Models*, http://www.biostat.wisc.edu/bmi576.html.
20. G. D. Forney (1973) The Viterbi algorithm, *Proc. IEEE* 61(3): 268-278.
21. J. P. Norton (1996) Roles for deterministic bounding in environmental modelling, *Ecological Modelling* 6:157-161.
22. N. Scott (2002) *Algae, cyanobacteria and water quality*, Agriculture and Agri-food Canada report, March 2002..
23. B. Guyen and A. Howard (2006) Modelling the growth and movement of cyanobacteria in river systems, *Science of the Total Environment*, 368: 898-908.
24. C. A. Pollino and J. A. Webb (2006) *Models for Ecological Risk Assessment of Cyanobacterial Blooms: How complex do they need to be?*, Personal communication (draft paper).
25. I. Chorus and J. Bartram (eds.) (1999) *Toxic Cyanobacteria in Water: A Guide to their Public Health Consequences, Monitoring and Management*, E & F N Spon, London/World Health Organisation.
26. J. A. Webb, N. A. Linacre and M. R. Grace (2006) *Management-oriented modelling of blue-green algal blooms: an example from Bourke Weir*, NSW, Australia, Personal communication (draft paper).
27. A. Saltelli, K. Chan, E. M. Scott (Eds.) *Sensitivity Analysis*. Wiley: Chichester, UK, 2000.
28. A. J. Jakeman, R. A. Letcher and J. P. Norton (2006), Ten interactive steps in model development and evaluation, *Environmental Modelling & Software*, 21(5): 602-614.