Dynamic Bayesian Networks: A factored model of probabilistic dynamics [extended abstract]

Sucheendra K. Palaniappan, P. S. Thiagarajan

School of computing, National University of Singapore,
{suchee,thiagu}@comp.nus.edu.sg

1 Introduction

The modeling and analysis of probabilistic dynamical systems is becoming a central topic in the formal methods community. Usually, Markov chains of various kinds serve as the core mathematical formalism in these studies. However, in many of these settings, the probabilistic graphical model called dynamic Bayesian networks (DBNs) [4] can be a more appropriate model to work with. This is so since a DBN is often a factored and succinct representation of an underlying Markov chain. Our goal here is to describe DBNs from this standpoint. After introducing the basic formalism, we discuss inferencing algorithms for DBNs. We then consider a simple probabilistic temporal logic and the associated model checking problem for DBNs with a finite time horizon. Finally, we describe how DBNs can be used to study the behavior of biochemical networks.

2 Dynamic Bayesian networks

There are many variants of DBNs with differing modeling capabilities. We shall begin with the simplest version in order to highlight the main features of a DBN.

A DBN $\mathcal{D}$ has an associated set of system variables $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ each taking values from a finite domain V. It also has a discrete time domain $\mathcal{T} = \{0, 1, \ldots\}$ associated with it. Often $\mathcal{T}$ will be a finite set. The structure of $\mathcal{D}$ consists of an acyclic directed graph $G_{\mathcal{D}} = (N, E)$ with $N = \mathcal{X} \times \mathcal{T}$. Thus there will be one node of the form X_i^t for each $t \in \mathcal{T}$ and each $i \in \{1, 2, \ldots, n\}$. The node X_i^t is to be viewed as a random variable that records the value assumed by the variable X_i at time t. The edge relation is derived by fixing the parenthood relation $PA : \mathcal{X} \to 2^{\mathcal{X}}$ over the system variables. Intuitively, $PA(X_i)$ is the set of system variables whose values at time t -probabilistically- influence the value assumed by X_i at time $t + 1$. This crucial structural information is to be obtained from the application at hand and will often be readily available.

The map PA will in turn induce the map $Pa : N \to 2^N$ given by: $Pa(X_i^t) = 0$ if $t = 0$. For $t > 0$, $X_j^{t'} \in Pa(X_i^t)$ iff $t' = t - 1$ and $X_j \in PA(X_i)$. The edge relation E is then given by: $(X_j^{t'}, X_i^t) \in E$ iff $X_j^{t'} \in Pa(X_i^t)$.

The dynamics of $\mathcal{D}$ is specified locally by assigning a *conditional probability table* (CPT) C_i^t to each node X_i^t. Suppose $Pa(X_i^t) = \{X_{j_1}^{t-1}, X_{j_2}^{t-1}, \ldots, X_{j_m}^{t-1}\}$. Then an entry in C_i^t will be of the form: $C_i^t(x_i \mid x_{j_1}, x_{j_2}, x_{j_m}) = p$. It says if the system

is in a state at $t-1$ in which $X_{j_l} = x_{j_l}$ for $1 \leq l \leq m$, then the probability of $X_i = x_i$ being the case at t is p. As might be expected, it is required that $\sum_{x_i \in V} C_i^t(x_i \mid x_{j_1}, x_{j_2}, x_{j_m}) = 1$ for each $(x_{j_1}, x_{j_2}, \ldots, x_{j_m}) \in V^m$.

For the simple version of a DBN that we are considering, the CPTs of a system variable are assumed to be time invariant. In other words, for each i and each $t, t' \in \mathcal{T}$ we will have $C_i^t(x_i \mid x_{j_1}, x_{j_2}, x_{j_m}) = C_i^{t'}(x_i \mid x_{j_1}, x_{j_2}, x_{j_m})$. Consequently one can specify both the structure and the local dynamics in terms of two adjacent slices. This so called "2-Time slice Bayesian network" ("2-TBN") representation of DBNs is quite standard [4]. An example of such a DBN but which have "unrolled" for illustration is shown in Fig.1.

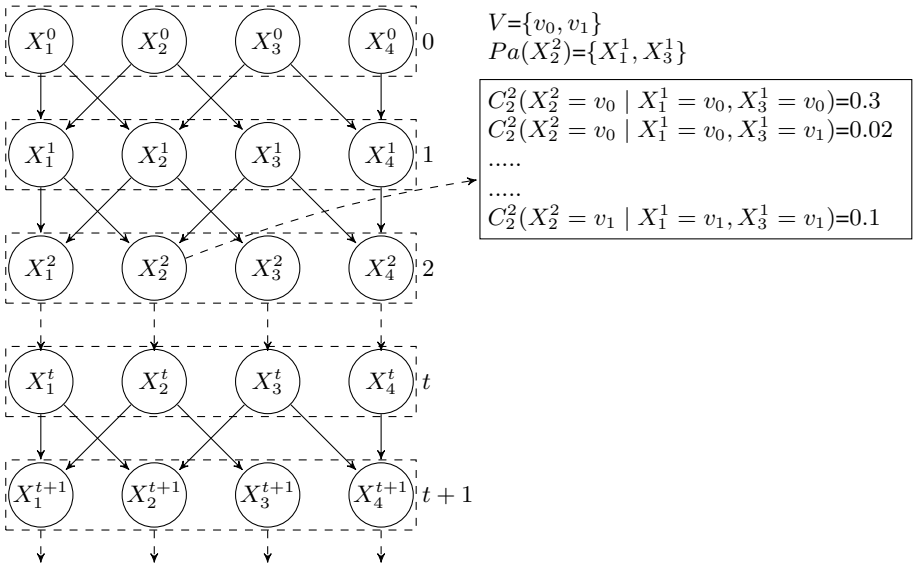

Fig. 1. Example of a DBN

Suppose $\mathcal{T} = \mathbb{N}$, the set of non-negative integers. Then the global dynamics of $\mathcal{D}$ can be described by a Markov chain. To bring this out, we first define $\hat{i} = \{j \mid X_j \in PA(X_i)\}$ to capture PA in terms of the corresponding indices. We let $\mathbf{s}_I$ will denote a vector of values over the index set $I \subseteq \{1, 2, \ldots, n\}$. It will be viewed as a map $\mathbf{s}_I : I \to V$. We will denote $\mathbf{s}_I(i)$ (with $i \in I$) as $\mathbf{s}_{I,i}$ or just $\mathbf{s}_i$ if I is clear from the context. If I is the full index set $\{1, 2, \ldots, n\}$, we will simply write $\mathbf{s}$.

The Markov chain $M_{\mathcal{D}}$ induced by $\mathcal{D}$ can now be defined via:
$M_{\mathcal{D}} : V^n \times V^n \to [0, 1]$ where for each $\mathbf{s}, \mathbf{s}' \in V^n$ we have $M_{\mathcal{D}}(\mathbf{s}, \mathbf{s}') = \prod p_i$ where $p_i = C_i(\mathbf{s}'_i \mid \mathbf{s}_{\hat{i}})$ for each i. It is easy to check that $M_{\mathcal{D}}$ is indeed a Markov chain.

We note that $M_{\mathcal{D}}$ has potentially K^n states and K^{2n} transitions where $K = |V|$. In contrast, the size of $\mathcal{D}$ is essentially determined by the CPTs and their description will be at most of size $n \cdot K^d$ where $d = max\{|PA(X_i)|\}_{1 \leq i \leq n}$. Often d will be much

smaller than n. In this sense the DBN is a succinct and factored representation of a large underlying Markov chain. Equally important, in many applications the transition probabilities of the chain may be obscure and inaccessible while the parenthood relationship between the variables as well as the CPT entries are easily identifiable.

In case $\mathcal{T} = \{0, 1, \ldots, T\}$ is a finite set, the global dynamics of $\mathcal{D}$ can still be be specified as a Markov chain $M_{\mathcal{D}}$. Its set of states will be $V^n \times \mathcal{T}$ and its transition relation given by:
(i) $M_{\mathcal{D}}((\mathbf{s}, t), (\mathbf{s}', t')) = 0$ if $t' \neq t+1$ or $t = t' = T$ and $\mathbf{s} \neq \mathbf{s}'$ (ii) $M_{\mathcal{D}}((\mathbf{s}, T), (\mathbf{s}, T)) = 1$ for every $\mathbf{s} \in V^n$. (iii) $M_{\mathcal{D}}((\mathbf{s}, t), (\mathbf{s}', t+1)) = \prod p_i$ if $t < T$ with $p_i = C_i(\mathbf{s}'_i \mid \mathbf{s}_{\hat{i}})$ for each i.

Thus every state of the form $(\mathbf{s}, t)$ with $t < T$ will be a transient state while every state of the form $(\mathbf{s}, T)$ will be an absorbing state.

There many variants of DBNs. Firstly, there could be dependencies between nodes belonging to the same time slice as shown in Fig 2(a). Secondly there could be dependencies across non-adjacent time-slices as shown in Fig 2(b). Some of the variables may be continuously valued. Yet another variation is where the CPTs for a system variable are time variant. In this case one must explicitly specify the CPT for each node instead of using a 2-TBN presentation. Last but not least, many of the state variable may be unobservable. The state of the system at any time point will be recorded as the values of the observable nodes. Apart from the CPTs specifying the local dynamics over the system variables, there will also be a probabilistic description of how the value of an observable variable is determined at a time point in terms of the values of its parent unobservable nodes. In fact it is this variant of a DBN that is typically used in AI applications [4]. We will however avoid this complication here. In what follows we will assume our DBNs to be as described above except that we will allow the CPTs corresponding each system variable to be time variant while the time domain is assumed to be finite.

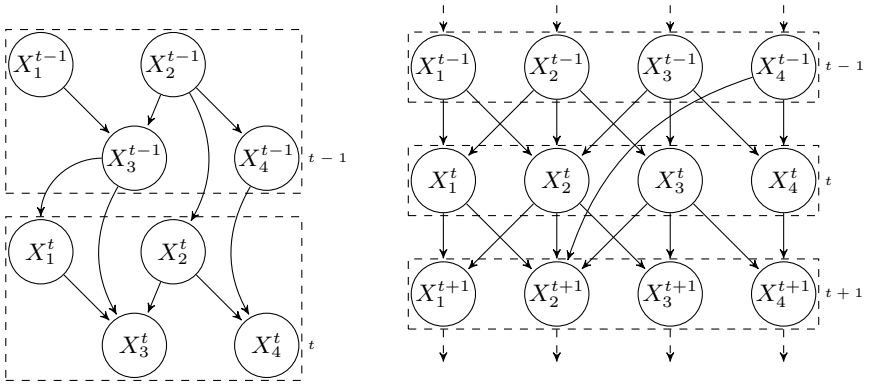

Fig. 2. Variants of DBNs

3 Probabilistic inferencing methods

The probability distribution $P(X_1^t, X_2^t, \ldots, X_n^t)$ describes the possible states of the system at time t. In other words, $P(\boldsymbol{X}^t = \boldsymbol{x})$ is the probability that the system will reach the state $\boldsymbol{x}$ at t. Starting from $P(\boldsymbol{X}^0)$ at time 0, given by $P(\boldsymbol{X}^0 = \boldsymbol{x}) = \prod_i C_i^0(\boldsymbol{x}_i)$, one would like to compute $P(X_1^t, \ldots, X_n^t)$ for a given t. As before, we will use $\hat{i} = \{j \mid X_j \in PA(X_i)\}$ to capture the set of indices of the parents of X_i and $V_{\hat{i}}$ will denote the tuple of values defined by $\hat{i}$.

We can use the CPTs to inductively compute this:

$$P(\boldsymbol{X}^t = \boldsymbol{x}) = \sum_{\boldsymbol{u}} P(\boldsymbol{X}^{t-1} = \boldsymbol{u}) \Big(\prod_i C_i^t(\boldsymbol{x}_i \mid \boldsymbol{u}_{\hat{i}}) \Big) \tag{1}$$

with $\boldsymbol{u}$ ranging over V^n.

Since $|V| = K$, the number of possible states at t is K^n. Hence explicitly computing and maintaining the probability distributions is feasible only if n is small or if the underlying graph of the DBN falls apart into many disjoint components. Neither restriction is realistic and hence one needs approximate ways to maintain $P(\boldsymbol{X}^t)$ compactly and compute it efficiently.

In fact, in most applications what one is interested in is the computation of the marginal distribution M_i^t of variable X_i. We shall view M_i^t to be a map $M_i^t : V \to [0, 1]$ satisfying $\sum_{v \in V} M_i^t(v) = 1$. Intuitively, $M_i^t(v)$ is the probability of X_i assuming the value v at time t. It is given by: $M_i^t(v) = \sum_{\mathbf{x}, \mathbf{x}(i) = v} \mathrm{P}(X_j^t = \mathbf{x}(j) \mid 1 \leq j \leq n)$.

Two interesting algorithms have been proposed for computing and maintaining these marginal probability distributions approximately [5, 6]. Such approximate distributions are usually called *belief states* and denoted by B, B^t etc. In Boyen-Koller algorithm (BK, for short) [5], a belief state is maintained compactly as a product of the probability distributions of independent clusters of variables. This belief state is then propagated *exactly* at each step through the CPTs. Then the new belief state is compacted again into a product of the probability distributions of the clusters. Consequently the exact propagation step of the algorithm can become infeasible for large clusters. To get around this, the factored frontier algorithm (FF, for short) maintains a belief state as a product of the marginal distributions of the individual variables.

FF computes inductively a sequence B^t of marginals as:

- $B_i^0(v) = C_i^0(v)$,
- $B_i^t(v) = \sum_{\boldsymbol{u} \in V_{\hat{i}}} [\prod_{j \in \hat{i}} B_j^{t-1}(\boldsymbol{u}_j)] C_i^t(v \mid \boldsymbol{u})$.

Both BK and FF algorithm can sometimes incur significant errors due to way they maintain and propagate the global probability distributions. To handle this the hybrid factored frontier algorithm (HFF, for short) was proposed in [7]. HFF attempts to reduce the error made by FF by maintaining the current belief state as a hybrid entity; for a small number of global states called *spikes*, their current probabilities are maintained. The probability distribution over the remaining states is represented, as in FF, as a product of the marginal probability distributions. HFF has been shown [7] to be scalable and efficient with reduced errors. It may be viewed as a parametrized extension to FF where σ, the number of spikes to be maintained is the parameter. When $\sigma = 0$ we get FF

whereas $\sigma =\mid V^n \mid$ corresponds to the exact inferencing algorithm. It turns out that that additional complexity of HFF (over that of FF) is only quadratic in σ and the accuracy increases monotonically as σ increases.

4 Probabilistic model checking on the DBNs

DBNs can be subjected to probabilistic model checking [8–10]. To illustrate this, we shall describe a simple probabilistic temporal logic and construct an approximate model checking procedure based on FF.

In our temporal logic, the atomic propositions will be of the form $(i, v)\#r$ with $\# \in \{\leq, \geq\}$ and r is a rational number in $[0, 1]$. The proposition $(i, v) \geq r$, if asserted at time point t, says that $M_i^t(v) \geq r$; similarly for $(i, v) \leq r$.

The formulas of our logic termed PBL (probabilistic bounded LTL) is then given by: (i) Every atomic proposition is a formula. (ii) If φ and φ' are formulas then so are $\sim \varphi$ and $\varphi \vee \varphi'$. (iii) If φ and φ' are formulas then so are $\boldsymbol{O}(\varphi)$ and $\varphi \boldsymbol{U} \varphi'$. Thus probability enters the logic solely at the level of atomic propositions.

The derived propositional connectives such as $\wedge, \supset, \equiv$ etc. and the temporal connectives $\boldsymbol{F}$ ("sometime from now") and $\boldsymbol{G}$ ("always from now") are defined in the usual way.

We assume we are dealing with a DBN $\mathcal{D}$ whose time domain is $\{0, 1, \ldots, T\}$ and whose CPTs for each system variable can vary over time. The formulas are interpreted over the sequence of marginal probability distribution vectors $\sigma = \mathbf{s}_0\mathbf{s}_1 \ldots \mathbf{s}_T$ generated by $\mathcal{D}$. In other words, for $0 \leq t \leq T$, $\mathbf{s}_t = (M_1^t, M_2^t, \ldots, M_n^t)$. Consequently $\mathbf{s}_t(i) = M_i^t$ for $1 \leq i \leq n$. We also let $\sigma(t) = \mathbf{s}_t$ for $0 \leq t \leq T$. We now define the notion of $\sigma(t) \models \varphi$ inductively:

- $\sigma(t) \models (i, v) \geq r$ iff $M_i^t(v) \geq r$. Similarly
  $\sigma(t) \models (i, v) \leq r$ iff $M_i^t(v) \leq r$.
- The propositional connectives $\sim$ and $\vee$ are interpreted in the usual way.
- $\sigma(t) \models \boldsymbol{O}(\varphi)$ iff $\sigma(t + 1) \models \varphi$.
- $\sigma(t) \models \varphi \boldsymbol{U} \varphi'$ iff there exists $t \leq t' \leq T$ such that $\sigma(t') \models \varphi'$ and for every t'' with $t \leq t'' < t'$, $\sigma(t'') \models \varphi$.

We say that the DBN $\mathcal{D}$ meets the specification φ and this is denoted as $\mathcal{D} \models \varphi$ iff $\sigma(0) \models \varphi$. The model checking problem is, given $\mathcal{D}$ and φ, to determine whether or not $\mathcal{D} \models \varphi$. To solve this problem, we begin by letting $SF(\varphi)$ denote the set of subformulas of φ. Since φ will remain fixed we will write below SF instead of $SF(\varphi)$.

The main step is to construct a labeling function st which assigns to each formula $\varphi' \in SF$ a subset of $\{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_T\}$ denoted $st(\varphi')$. After the labeling process is complete, we declare $\mathcal{D} \models \varphi$ just in case $\mathbf{s}_0 \in st(\varphi)$. Starting with the atomic propositions, the labeling algorithm goes through members of SF in ascending order in terms of their structural complexity. Thus φ' will be treated before $\sim \varphi'$ is treated and both φ' and φ'' will be treated before $\varphi' \boldsymbol{U} \varphi''$ is treated and so on.

Let $\varphi' \in SF(\varphi)$. Then:

- If $\varphi' = A$ -where A is an atomic proposition- then $\mathbf{s}_t \in st(A)$ iff $\sigma(t) \models A$. We run the DBN inference algorithm (say, FF) to determine this. In other words, $\mathbf{s}_t \in st(A)$ iff $B_i^t(v) \geq r$ where $A = (i, v) \geq r$ and B_i^t is the marginal distribution of X_i^t computed by the inference algorithm. Similarly $\mathbf{s}_t \in st(A)$ iff $B_i^t(v) \leq r$ in case $A = (i, v) \leq r$.
- If $\varphi' = \sim \varphi''$ then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \notin st(\varphi'')$.
- If $\varphi' = \varphi_1 \vee \varphi_2$ then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi_1)$ or $\mathbf{s}_t \in st(\varphi_2)$.
- Suppose $\varphi' = \boldsymbol{O}(\varphi'')$. Then $\mathbf{s}_T \notin st(\varphi')$. Further, for $0 \leq t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_{t+1} \in st(\varphi'')$.
- Suppose $\varphi' = \varphi_1 \boldsymbol{U}\ \varphi_2$. Then we decide whether or not $\mathbf{s}_t \in st(\varphi')$ by starting with $t = T$ and then treating decreasing values of t. Firstly $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi_2)$. Next suppose $t < T$ and we have already decided whether or not $\mathbf{s}_{t'} \in st(\varphi')$ for $t < t' \leq T$. Then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi_2)$ or $\mathbf{s}_t \in st(\varphi_1)$ and $\mathbf{s}_{t+1} \in st(\varphi')$.

$\varphi' = \boldsymbol{F}(\varphi'')$ and $\varphi' = \boldsymbol{G}(\varphi'')$ can be handled directly. As in the case of $\boldsymbol{U}$, we start with $t = T$ and consider decreasing values of t:

- Suppose $\varphi' = \boldsymbol{F}(\varphi'')$. Then $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi'')$. For $t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi'')$ or $\mathbf{s}_{t+1} \in st(\varphi')$.
- Suppose $\varphi' = \boldsymbol{G}(\varphi'')$. Then $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi'')$. For $t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi'')$ and $\mathbf{s}_{t+1} \in st(\varphi')$.

Due to the fact the model checking procedure just needs to treat one model which is a finite sequence, it is a very simple procedure. Its time complexity is linear in the size of the formula φ.

5 An application: DBN approximations of the ODEs based dynamics of biopathways

Biological pathways are usually described by a network of biochemical reactions. The dynamics of these reaction networks can be modeled and analyzed as a set of Ordinary Differential Equations (ODEs); one equation of the form $\frac{dy}{dt} = f(\mathbf{y}, \mathbf{r})$ for each molecular species y, with f describing the kinetics of the reactions that produce and consume y, while $\mathbf{y}$ is the set (vector) of molecular species taking part in these reactions and $\mathbf{r}$ are the rate constants associated with these reactions. These ODEs will be nonlinear due to the kinetic laws governing the reactions and high-dimensional due to the large number of molecular agents involved in the pathway. Hence closed form solutions will not be obtainable. Further many of the rate constants appearing in the ODEs will be unknown. As a result, one must resort to large scale numerical simulations to perform analysis tasks such as parameter estimation and sensitivity analysis. In addition, only a limited amount of noisy data of limited precision will be available for carrying out model calibration (i.e. parameter estimation) and model validation. Guided by these considerations a method for approximating the ODE dynamics of biological pathways as a DBN was developed in [11].

The first step in this approximation procedure is to discretize the time domain. For biopathways, experimental data will be available only for a few time points with the

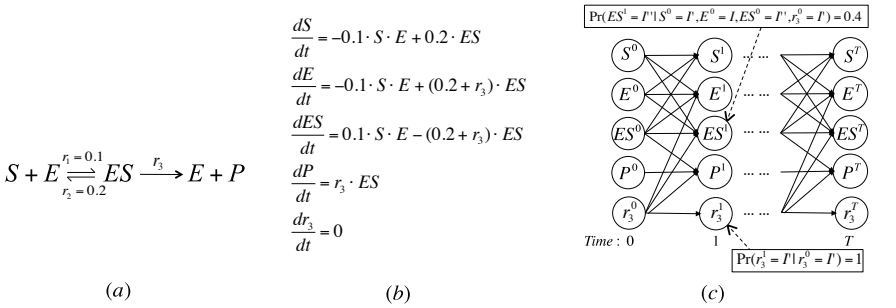

Fig. 3. (a) The enzyme catalytic reaction network. (b) The ODE model. (c) The DBN approximation.

value measured at the final time point typically signifying the steady state value. Hence we assume the dynamics is of interest only for discrete time points and, furthermore, only up to a maximal time point. We denote these time points as $\{0, 1, \ldots, T\}$. Next we assume that the values of the variables can be observed with only finite precision and accordingly partition the range of each variable y_i of the ODE into a set of intervals $\mathbf{I}_i$ ($\mathbf{I}_j$). The initial values as well as the parameters of the ODE system are assumed to be distributions (usually uniform) over certain intervals. We then sample the initial states of the system many times [11] and generate a trajectory by numerical integration for each sampled initial state. The resulting set of trajectories is then treated as an approximation of the dynamics of ODE system.

Unknown rate constants are handled as additional variables. We assume that the minimum and maximum values of these unknown rate constant variables are known. We then partition these ranges of values also into a finite numbers of intervals, and fix a uniform distribution over *all* the intervals. For each such variable r_j we add the equation $\frac{dr_j}{dt} = 0$ to the system of ODEs. This will reflect the fact that once the initial value of a rate constant has been sampled, this value will not change during the process of generating a trajectory. Naturally, different trajectories can have different initial values.

A key idea is to compactly store the generated set of trajectories as a DBN. This is achieved by exploiting the network structure and by simple counting. First we specify one random variable Y_i for each variable y_i of the ODE model

For each unknown rate constant r_j , we add one random variable R_j. The node Y_k^{t-1} will be in $Pa(Y_i^t)$ iff $k = i$ or y_k appears in the equation for y_i. Further, the node R_j^{t-1} will be in $Pa(Y_i^t)$ iff r_j appears in the equation for y_i. On the other hand R_j^{t-1} will be the only parent of the node R_j^t.

Suppose $Pa(Y_i^t) = \{Z_1^{t-1}, Z_2^{t-1}, \ldots, Z_k^{t-1}\}$. Then a CPT entry of the form $C_i^t(I \mid I_1, I_2, \ldots, I_k) = p$ says that p is the probability of the value of y_i falling in the interval I at time t, given that the value of Z_j was in I_j for $1 \leq j \leq k$. The probability p is calculated through simple counting. Suppose N is the number of generated trajectories. We record, for how many of these trajectories, the value of Z_j falls in the interval I_j simultaneously for each $j \in \{1, 2, \ldots, k\}$. Suppose this number is J. We then determine

for how many of these J trajectories, the value of Y_i falls in the interval I at time t. If this number is J' then p is set to be $\frac{J'}{J}$.

If r_j is an unknown rate constant, in the CPT of R_j^t we will have $P(R_j^t = I \mid R_j^{t-1} = I') = 1$ if $I = I'$ and $P(R_j^t = I \mid R_j^{t-1} = I') = 0$ otherwise. This is because the sampled initial value of r_j does not change during numerical integration. Suppose r_j appears on the right hand side of the equation for y_i and $Pa(Y_i^t) = \{Z_1^{t-1}, Z_2^{t-1}, \ldots, Z_\ell^{t-1}\}$ with $Z_\ell^{t-1} = R_j^{t-1}$. Then for each choice of interval values for nodes other than R_j^{t-1} in $Pa(Y_i^t)$ and for each choice of interval value $\widehat{I}$ for r_j there will be an entry in the CPT of Y_i^t of the form $P(y_i^t = I \mid Z_1^{t-1} = I_1, Z_2^{t-1} = I_2, \ldots, R_j^{t-1} = \widehat{I}) = p$. This is so since we will sample for all possible initial interval values for r_j. In this sense the CPTs record the approximated dynamics for all possible combinations of interval values for the unknown rate constants. The main ideas of this construction are illustrated in Fig.3. In this example we have assumed that r_3 is the only unknown rate constant.

After building this DBN, we use a Bayesian inference based technique to perform parameter estimation to complete the construction of the model (the details can be found in [11]). This will result in a calibrated DBN in which each unknown rate constant will have a specific interval value assigned to it. The one time cost of constructing the DBN can be easily recovered through the substantial gains obtained in doing parameter estimation and sensitivity analysis [11]. This method can cope with large biochemical networks with many unknown parameters. It has been used to uncover new biological facts about the complement system [12] where the starting point was a ODE based model with 71 unknown parameters. We have shown in [8] how a GPU based implementation of the approximation procedure can considerably extend the scalability of this approach. We have also shown how interesting properties of the dynamics of biological systems can be verified via probabilistic model checking along the lines sketched in the previous section (using FF).

6 Conclusion

Dynamic Bayesian networks are a formalism for representing complex stochastic dynamical systems in a compact and natural way. Their exact analysis is difficult. However efficient approximate inferencing algorithms are available using which one can also construct -approximate- probabilistic model checking procedures. We feel that they can play an important role in the modeling and analysis of biochemical networks. Our own work in which we have used DBNs to approximate the ODEs based dynamics of signaling paths supports this. A new avenue to explore in this connection will be to extract DBN models of biochemical networks from descriptions based on rule based languages such as kappa [13] and Bionetgen [14] as well as the chemical master equation [15]. In these formalism the underlying dynamic model is a Continuous Time Markov chain (CTMC). However the structure and locality of the interactions in the biochemical network under study should readily lend itself to a semantics based on DBNs. Finally, verification methods for DBNs based on Bayesian statistical model checking [16] are worth exploring further.

Acknowledgements The research reported here has been partially supported by the Singapore Ministry of Education grant MOE2011-T2-2-12.

References

1. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing **6** (1994) 512–535
2. Baier, C., Katoen, J.: Principles of model checking. The MIT Press (2008)
3. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In Gopalakrishnan, G., Qadeer, S., eds.: Proc. 23rd International Conference on Computer Aided Verification (CAV'11). Volume 6806 of LNCS., Springer (2011) 585–591
4. Koller, D., Friedman, N.: Probabilistic Graphical Models - Principles and Techniques. MIT Press (2009)
5. Boyen, X., Koller, D.: Tractable Inference for Complex Stochastic Processes. In: Proc. 14th Int. Conf. Uncertainty in Artificial Intelligence (UAI '98). (1998) 33–42
6. Murphy, K.P., Weiss, Y.: The Factored Frontier Algorithm for Approximate Inference in DBNs. In: Proc. 17th Int. Conf. Uncertainty in Artificial Intelligence (UAI '01). (2001) 378–385
7. Palaniappan, S.K., Akshay, S., Genest, B., Thiagarajan, P.S.: A Hybrid Factored Frontier Algorithm for Dynamic Bayesian Network Models of Biopathways. In: Proc. 9th Int. Conf. on Computational Methods in Systems Biology (CMSB '11). (2011) 35–44
8. Liu, B., Hagiescu, A., Palaniappan, S.K., Chattopadhyay, B., Cui, Z., Wong, W., Thiagarajan, P.S.: Approximate probabilistic analysis of biopathway dynamics. Bioinformatics **28**(11) (June 2012) 1508–1516
9. Langmead, C., Jha, S., Clarke, E.: Temporal Logics as Query Languages for Dynamic Bayesian Networks: Application to D. Melanogaster Embryo Development. Technical report, Carnegie Mellon University (2006)
10. Langmead, C.J.: Generalized queries and bayesian statistical model checking in dynamic bayesian networks: Application to personalized medicine. In: In: Proc. 8th Ann. Intnl Conf. on Comput. Sys. Bioinf. (CSB). (2009) 201–212
11. Liu, B., Hsu, D., Thiagarajan, P.S.: Probabilistic Approximations of ODEs based Bio-Pathway Dynamics. Theor. Comput. Sci. **412** (2011) 2188–2206
12. Liu, B., Zhang, J., Tan, P.Y., Hsu, D., Blom, A.M., Leong, B., Sethi, S., Ho, B., Ding, J.L., Thiagarajan, P.S.: A Computational and Experimental Study of the Regulatory Mechanisms of the Complement System. PLoS Comput. Biol. **7**(1) (01 2011) e1001059
13. Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J.: Rule-Based Modelling of Cellular Signalling. In: Proc. 18th Int. Conf. Concurrency Theory (CONCUR '07). (2007) 17–41
14. Faeder, J.R., Blinov, M.L., Goldstein, B., William, Hlavacek, S.: Rule-based modeling of biochemical networks. Complexity **10** (2005) 22–41
15. Henzinger, T.A., Mikeev, L., Mateescu, M., Wolf, V.: Hybrid numerical solution of the chemical master equation. In: Proceedings of the 8th International Conference on Computational Methods in Systems Biology. CMSB '10, New York, NY, USA, ACM (2010) 55–65
16. Jha, S.K., Clarke, E.M., Langmead, C.J., Legay, A., Platzer, A., Zuliani, P.: A Bayesian Approach to Model Checking Biological Systems. In: Proc. 7th Int. Conf. Computational Methods in Systems Biology (CMSB '09). (2009) 218–234