MODELING DYNAMICAL SYSTEMS BY MEANS OF DYNAMIC BAYESIAN NETWORKS

Anna Łupińska–Dubicka

Faculty of Computer Science, Białystok University of Technology

Abstract: Bayesian networks (BNs) are powerful tools for modeling complex problems involving uncertain knowledge. They have been employed in practice in a variety of fields. Their extension to time-dependent domains, dynamic Bayesian networks (DBNs) allow to monitor and update the system as time procedes and predict further behavior of the system. Most practical uses of DBNs involve temporal influences of the first order, i.e., influences between neighboring time steps. This choice is a convenient approximation influenced by the existence of efficient algorithms for first order models and limitations of available tools. This paper presents how to create higher order dynamic Bayesian networks and shows that introducing higher order influences can improve the accuracy of the model. To introduce the formalism to the readers, it describes a hypothetical simplified model based on a DBN.

Keywords: Bayesian networks, temporal dependencies, dynamical models, dynamic Bayesian networks

1. Introduction

The world around us is dynamic. Most of the physiological processes occurring in the human body, like many natural phenomena or processes are of dynamical character. Consequently, the modeled phenomena very often are generate time series data. Every variable is observed in successive moments of time. In addition, the analyzed phenomenon is affected not only by the current observations, but also the findings in previous periods. In this case, using static models can be inadequate and can lead to obtaining in correct results. To present and analyze the phenomena that change over time we need to use a dynamic model that takes into account the relationship between the values of the model parts in different moments of time.

This paper concentrates on models that belong to the class of probabilistic graphical models, with their two prominent members, Bayesian networks (BNs) [20] and

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 9, pp. 77-92, 2012.

dynamic Bayesian networks (DBNs) [7]. BNs are widely used as practical tools for knowledge representation and reasoning under uncertainty in equilibrium systems. DBNs extend them to time-dependent domains by introducing an explicit notion of time and influences that span over time. Most practical uses of DBNs involve temporal influences of the first order, i.e., influences between neighboring time steps. This choice is a convenient approximation influenced by existence of efficient algorithms for first order models and limitations of available tools. After all, introducing higher order temporal influences may be costly in terms of the resulting computational complexity of inference, which is NP-hard even for static models. Limiting temporal influences to influences between neighboring states is equivalent to assuming that the only thing that matters in the future trajectory of the system is its current state. Many real world systems, however, have memory that spans beyond their current state.

The idea of increasing modeling accuracy by means of increasing the time order of the model was illustrated by Shannon [21]. In his seminal paper, he shows sentences in the English language, generated by a series of Markov chain models of increasing time order, trained by means of the same corpus of text. The following example sentences come from [3].

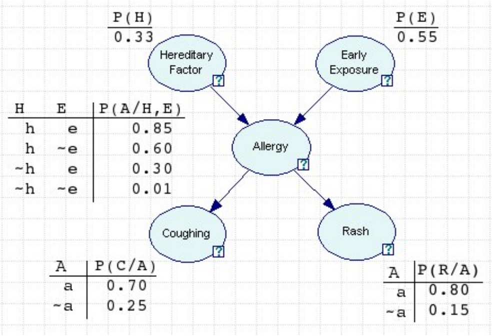
"saade ve mw hc n entt da k eethetocusosselalwo gx fgrsnoh,tvettaf aetnlbilo fc lhd okleutsndyeosthtbogo eet ib nheaoopefni ngent" In the above text, letters were generated by a zero order model. The only assumption was that the various letters appear with the different probablity. However, if we look at the various expressions of any language, we can see that the letters are very common in a certain context. In the case of English, the letter Q, in all probability will be the next U. Sample text generated using the chain the first order, in which the choice of each letter is a random function of its predecessor: *"t I amy, vin. id wht omanly heay atuss n macon aresethe hired boutwhe t, tl, ad torurest t plur I wit hengamind tarer-plarody thishand"* Going one step further, we can see that the digram TH in the English language often vowels A, E, I, O or U, slightly less consonant R or W and rarely other letters. In the following piece of text used the second-order Markov chain: *"Ther I the heingoind of-pleat, blur it dwere wing waske hat throos. Yout lar on wassing, an sit". "Yould," "I that vide was nots ther."* And so can look sample text, if we use the model of the fourth order. As we can see, most of the words are the words of English: *"His heard.» «Exactly he very glad trouble, and by Hopkins! That it on of the who difficentralia. He rushed likely?» «Blood night that."*

The resemblance of the latter sentence to ordinary English text, an informal measure of the model's accuracy, has increased dramatically between the first and the fourth orders. A first order model was essentially impotent in its ability to model the problem.

This paper presents how to create higher order dynamic Bayesian networks and shows that introducing higher order influences can improve the accuracy of the model. To introduce the formalism to the readers, it describes a hypotethical simplified model based on a DBN. The remainder of the paper is structured as follows. Section 2. introduces Bayesian networks. Section 3. presents a short description of methods dealing with dynamical processes. Section 4. presents in details dynamic Bayesian networks, one of the extension of BNs. Section 5. concludes the paper.

2. Bayesian Networks

Bayesian networks (BNs), also called belief networks or causal networks, belong to the family of probabilistic graphical models (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables Formally, a BN $\mathcal{B}$ is a pair $<\mathcal{G},\Theta>$, where $\mathcal{G}$ is an acyclic directed graph in which nodes represent random variables $X_1,\ldots,X_n$ and edges represent direct dependencies between pairs of variables. Θ represents the set of parameters that describes the probability distribution for each node X_i in $\mathcal{G}$, conditional on its parents in $\mathcal{G}$, i.e., $P(X_i|Pa(X_i))$. Often, the structure of the graph is given as a causal interpretation, convenient from the point of view of knowledge engineering and user interfaces. BNs allow for computing probability distributions over subsets of their variables conditional on other subsets of observed variables.



P(H)
0.33

P(E)
0.55

H	E	P(A/H,E)
h	e	0.85
h	~e	0.60
~h	e	0.30
~h	~e	0.01

A	P(C/A)
a	0.70
~a	0.25

A	P(R/A)
a	0.80
~a	0.15

Fig. 1. A simple BN illustrating selected causes and effects of allergy in children

Consider the simple BN shown in Figure 1, illustrating various causes and effects of allergy in children. All variables in this example are Boolean. The tendency to develop allergies has a hereditary component: Allergic parents are more likely to have allergic children, whose allergies are likely to be more severe than those from non-allergic parents. Exposure to allergens, especially in early life, is also an important risk factor for allergy. When an allergen enters the body of an allergic child, the child can cough or develop a rash. Figure 1 shows the dependency structure among the variables and the conditional probability distributions for each of the variables.

3. Modeling dynamical relationships

While Bayesian networks are powerful tool for representing uncertainty, they do not provide direct mechanism for representing temporal dependencies. Most of the events that we meet in our everyday life are not detected based on a particular point in time. They can be described through the multiple states of observations that all together lead up to final event. The ability to model effectively the temporal aspects of the domain plays cruicial role in modeling the World. For example, in medicine representing and reasoning about time is cruitial for prevention, diagnosis, terapeutic management, or prognosis. In industry, capturing the temporal aspects is essential for diagnosis and prediction of events and disturbances. The efforts to introduce temporality into Bayesian networks have resulted in a variety of networks for applications such as planning, diagnosis, forecasting, and scheduling.

Dean and Kanazawa [7] proposed a temporal belief network, a directed graphical model where nodes represent the truth of a state variable at the single point in time. The network is arranged into time slices representing the system's complete state at the single point in time, and time slices are duplicated over a predetermined and fixed–length time grid representing the time interval of interest. Links between state variables within the time slice are disallowed.

Network of dates proposed by Berzuini [4] represents a departure from the multiple instantiations approach because each temporal duration is represented by a node. Berzuini associates a probability density with each temporal random variable to represent a continous time.

An extention proposed by Santos and Young [14] is based on the definition of temporal aggretage (TA). Each aggregate represents a process changing over time. A temporal aggregate consists of the set of states, that the process can take on, and a set of temporal intervals each having an associated random variable. Their Probablistic Temporal Network (PTN) is a directed graph in which the nodes are TAs and the edges are the causal/temporal causal relationships between aggregates.

In Modifiable Temporal Bayesian Networks with Single–granularity (MTBN–SG) Aliferis and Cooper [1] distinguished three types of the variables: ordinary variables (corresponding to potentially observed phenomena), mechanism variables (corresponding to causal mechanism between variables), time–lag quantifier variables (corresponding to the time lag between the cause variable and the effect variable). All these variables create directed, possibly cyclic, graph over a range of time points.

Basing on the fact that in many cases there are only few state changes in the temporal range of interest Arroyo–Figueroa and Sucar [2] proposed an extension of Bayesian networks, called Temporal Nodes Bayesia Networks (TNBN). Their approach is based on the definition of *temporal node*, that is defined by a set of ordered pairs: the value of the variable and a time interval during the variable can change its state. Temporal interval of each temporal node are relative to the parent nodes. Each arc in TNBN corresponds to a causal–temporal relation.

In Networks of Probabilistic Events in Discrete Time (NPEDT) [12], like in TNBNs, each variable represent an event that can occur only once. However, they differ from TNBNs in that time is discretized using the same unit for all variables. The value taken on by a variable indicates the absolute, not relative, time at which the event occurs.

4. Dynamic Bayesian Networks

Dynamic Bayesian networks [7] are a temporal extension of Bayesian networks for modeling dynamic systems. While the Bayesian network shows the cumulative probability distribution over a set of random variables independent of time, the dynamic Bayesian networks can be seen as a multi–dimensional representation of a random process. DBNs allow the interpretation of the present, the reconstruction of the past and the forecasting of the future. Phenomena are located in time, and location at the time is ordered by the "earlier and later" relationship. Mostly due to the computational complexity of the inference algorithms, time is treated as a discrete variable. It should be noted that term *dynamic* means that we model the system's development over time and not that the model structure and its parameters change over time, even though the latter is theoretically possible.

4.1 Network structure

A DBN is a directed, acyclic graphical model of a stochastic process. It consists of time–steps and each of time–steps containts its own variables. The most common approach is usually assumed that the network meets the Markov property, i.e., the value of future states of the process are determined only by its current state, regardless of the past. In other words, the future states of the process are conditionally independent of the past states. Such a network is called a first order network. Usually, the DBN is often defined as a pair of BNs $(\mathcal{B}^1, \mathcal{B}^{\rightarrow})$, where $\mathcal{B}^1$ represents a priori probability distribution $P(\mathbf{Z}^1)$ of the model. Typically, $Z^t = (\mathbf{U}^t, \mathbf{X}^t, \mathbf{Y}^t)$, where $\mathbf{U}$ represents the input, $\mathbf{X}$ represents the hidden and $\mathbf{Y}$ represents the output variables of the model. $\mathcal{B}^{\rightarrow}$ is a two time slice BN (2TBN), that defines the transition distribution $P(\mathbf{Z}^t|\mathbf{Z}^{t-1})$ as follows [18]:

$$P(\mathbf{Z}^t|\mathbf{Z}^{t-1}) = \prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t)) \ ,$$

where Z_i^t is the i–th node at time t. $Pa(Z_i^t)$ are the parents of Z_i^t from the same or from the previous time–slice. The joint probablity distribution for a sequence of length T can be obtained by *unrolling* the $\mathcal{B}^{\rightarrow}$ network:

$$P(\mathbf{Z}^{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t)) \ .$$

Consider a two years old child whose parents suffer from allergy and who has been exposed to allergens. We know that this child has not developed any symptoms of allergy in the previous year. Suppose that we want to know the probability that allergy appears in the third year. If we use the BN pictured in Figure 1, we omit all historical information except that for the current year. Figure 2a) shows a DBN of first temporal order, which allows us to predict the probability of the child developing allergy in this and in the future years. Number of slices is the number of steps for which we perform the inference. The time step that is chosen for a dynamic Bayesian network varies on considered phenomenon. In this example, one step means one year. Temporal plate is the part of a DBN that contains nodes changing over time. *Hereditary Factor* is time invariant and, hence, is outside of the temporal plate. As we can see, now the value of the *Allegry* variable depends not only on *Hereditary factor* and *Early exposure* but also on its value from previous time step.

As mentioned before, the state at time t generally depends on the states at previous $k = 1$ time steps. There is nothing in the theory that prevents k from being any number between 1 and $t - k$. However, modeling dependencies of higher orders is very rare in the literature, probably because of the availability of both theoretical tools (algorithms) and practical (availability of software). In many cases, taking into consideration only the first–order dependences is probably sufficient. However, there is a possibility that some phenomena could be modeled with higher accuracy if they also take account of the influence of states earlier than immediately preceding the

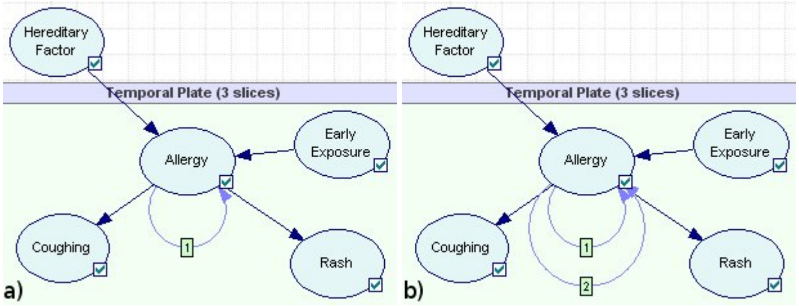


Fig. 2. A DBN modeling causes and effects an allergy in children: a) first order b) second order

current state of the model. It is likely that such simplification of dynamic models can lead to incomplete and even erroneous results. According to Murphy [16], it is possible to simulate k^{th}–ordered processes by means of first order Markov processes by adding new variables (called lag variables). This approach, unfortunately, obscures the model by introducing many unnecessary extra variables.

The SMILE$^{©}$ library, the part of the software developed by Decision Systems Laboratory of the University of Pittsburgh, provides an implementation of the extended DBNs formalism. This implementation is, to the best of my knowledge, the first implementation of temporal reasoning that provides support for k^{th}-order temporal arcs.

In case of k^{th}–ordered processes, $\mathcal{B}^{\rightarrow}$ can be defined not as 2TBN, but as a $(k+1)$TBN and describes the transition model $P(\mathbf{Z}^t|\mathbf{Z}^{t-1},\mathbf{Z}^{t-2},\ldots,\mathbf{Z}^{t-k})$ as follows:

$$P(\mathbf{Z}^t|\mathbf{Z}^{t-1},\mathbf{Z}^{t-2},\ldots,\mathbf{Z}^{t-k}) = \prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t))\ ,$$

In this case, the set of parents $Pa(Z_i^t)$ can contain nodes not only from the previous time–slice, but also from time–slices further in the past. The joint probablity distribution for a sequence of length T can be obtained by *unrolling* the (k+1)TBN network:

$$P(\mathbf{Z}^{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t))\ .$$

Typically, the older the child, the lower the probability of allergy appearing. And, generally, a child who has not developed allergy two years in a row has a lower chance of developing allergy in the third year. A reasonable expectation is that modeling higher order dependencies should increase the accuracy of the model. Figure 2b) shows a second time-order DBN, i.e., a model in which there are two temporal arcs from node *Allergy*, the first order takes the information from one step before, the second from two steps before.

4.2 Inference

The main goal of inference for DBNs is to calculate $P(X_{t1}|y_{t2:t3})$, where X_{t1} is the value at time $t1$ and $Y_{t2:t3}$ represents all observation between times $t2$ and $t3$. There exist several interesting ways of performing inference on DBNs. The three most common types are illustrated below.

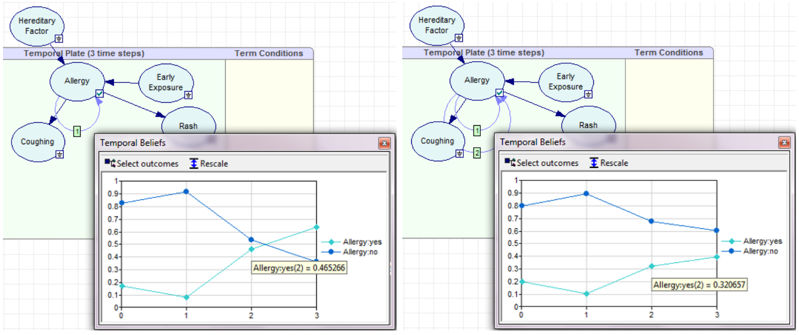


Fig. 3. First (a) and second (b) order DBN's temporal beliefs and probability of allergy in the third year (filtering)

Filtering. The current belief state is computed given all evidence from the past. It is used to keep track of the current state for making rational decisions. This task is traditionally called *filtering*, because we are filtering out the noise from the observations. However, in some circumstances the term *monitoring* might be more appropriate.

For example, we want to know the probability that allergy appears in the third year. Figure 3a) presents temporal beliefs for the first order network. As we can see, the probablity of allergy in the third year equals 46.5%. On the other hand, in Figure 3b) we can see that the probablity of allegry in the third year for the second order model is lower than from the first order model and equals 32.1%.

Smoothing. Sometimes we want to estimate the state of the past, given all the evidence up to the current time, i.e., $P(X_{t-l}|y_{1:t})$, where $l > 0$ is how far we want to look back. This type of the inference can be useful to get a better estimate of the past state, because more evidence is avaiable at time t than at time $t-l$.

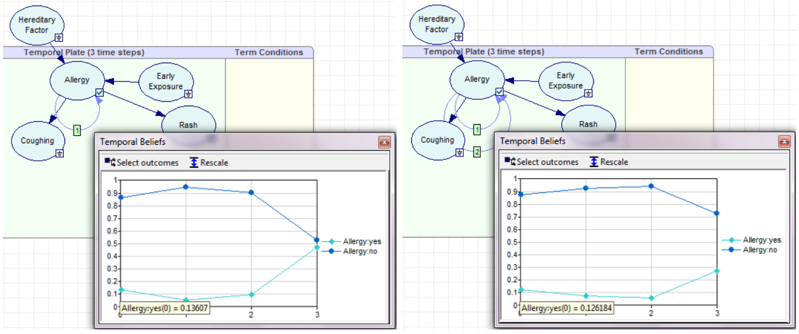


Fig. 4. First (a) and second (b) order DBN's temporal beliefs and probability of allergy in the first year (smoothing)

For example, we want to know how probable was that our three years old child suffered from allergy in its first year of life, given information of lack of any symptoms of allergy in the subsequent years. Figure 4 shows that for the first order dynamic network this probability equals 13.6%, and for the second order network it equals 12.6%.

Prediction. In addition to estimating the current or past state, we might want to predict the future, i.e., compute $P(X_{t+h}|y_{1:t})$, where $h>0$ is how far we want to look ahead. This kind of inference can be used to evaluate the effect of possible actions on the future state.

For example, we want to know the probability that our child will suffer from allegry in its the fourth year even though it has not developed any symptoms of allergy up to thrird year. In Figure 5, we can see again that the second order model is more precize and estimates this probablity at 39.7% while for the first order model this probability equals 63.6%.

While dynamic Bayesian networks are an extension of Bayesian networks there already exist algorithms for inference, divided into two major categories: exact inference and approximate infenerce.

Exact inference. The first approach for exact inference in DBNs is based on the notion that an unrolled DBN is in fact the same as a static BN. In this case we can

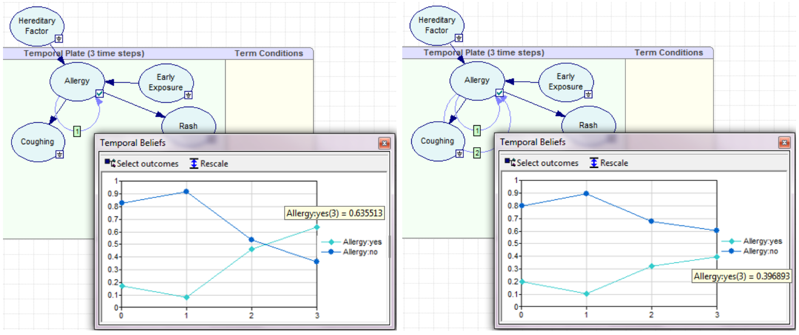


Fig. 5. First (a) and second (b) order DBN's temporal beliefs and probability of allergy in the fourth year (prediction)

use any of the inference algorithm for static BNs, such as the variable elimination or the junction tree algorithm. The basic idea of variable elimination is to take a probabilistic model over n variables and reduce it to a model over $n-1$ variables, while maintaining the ability of the model to answer queries of interest. This process is repeated until we have a trivial model from which we can look up answers immediately. The complexity of the algorithm depends on the amount of work it takes to eliminate a variable, which is sensitive to the order in which variables are eliminated.

The juntion tree algorithm converts the original Bayesian network into a tree structure where each node corresponds to a certain set of nodes (a clique) in the original network. This tree is obtained by following steps: (1) constructing an undirected graph called the moral graph from the belief network, (2) selectively adding arcs to the moral graph to form a triangulated graph, (3) identifying the maximal cliques from the triangulated graph, (4) building the junction tree, starting with cliques as the nodes, where each link between two cliques is labeled by a separator set. Unfortunately, when we create a junction tree from an unrolled DBN, the cliques tend to be very large, often making exact inference intractable.

The second approach for exact inference is to convert the DBN into a hidden Markov model (HMM) and apply the forward–backward algorithm. Converting a DBN to a HMM is only possible with discrete state DBNs. If there are N hidden variables per slice, and each such variable can have up to M values, the resulting HMM will have $S = M^N$ states. As long as S is not to large, this is good method, since the forwards–backwards algorithm is exact, and is very simple to implement.

The frontier algorithm [22] is based on the notion that in the forward–backward algorithm for HMM, variable X_t d–separates the past from the future. We can generalize this idea to DBNs noting that all nodes in a time slice d–separate the past from the future. This set of nodes is called the *frontier*. The modified $2TBN$ is here called a 1.5DBN H_t (H for half), because it is created by taking all nodes from slice 2 and only outgoing nodes from slice 1. For each H_t we construct the junction tree. Inference is performed on each separate tree and messages are passed between them via the interface nodes.

The interface algorithm [18] optimalizes the frontier algorithm by using to d–separate the past from the future only these hidden nodes in a time slice which have outgoing arcs. This subset of the frontier is called the *forward interface*.

Approximate inference Although the exact inference is always possible in discrete–state models, very often it can be computationally prohibitive. When faster results are needed, we can use approximate methods to perform inference. Generally, we can distinguish two types of approximations: *deterministic* and *stochastic*. Deterministic algorithms for the discrete–state DBNs include: the loopy belief propagation, the Boyen–Koller algorithm, and the factored frontier algorithm. Stochastic algorithms can be divided into two groups: offline and online. Offline methods are often based on importance sampling (likelihood weighting) or Monte Carlo Markov Chain (MCMC). Online methods usually rely on particle filtering (PF), also known as sequentional Monte Carlo, the bootstrap filter, the condensation algorithm, survival of the fittest, or interacting particle approximations.

In loopy belief propagation (LBP) we apply the Pearl's algorithm [20] to the original graph even if this graph contains loops (undirected cycles). In theory, this runs the risk of double counting information. However, it was shown that in practise this method works suprisingly well [19].

The basic idea behind the Boyen–Koller (BK) algorithm [6] is to approximate the joint distribution over the interface as a product of marginals. The marginals are exactly updated using the junction tree algorithm. BK constructs the junction tree for the 1.5 DBN H_t, but does not require that all the interface nodes are in the same clique. Since BK does exact inference in a two–slice DBN, sometimes it can be intractable.

The factored frontier (FF) algorithm [17] also represents the belief state as a product of marginals and thus is very similar to the Boyen–Koller algorithm. However, FF is simpler than BK, because instead of relying on the juction tree algorithm it computes the marginals directly.

Stochastic algorithms have the advantage over deterministic algorithms that they are easier to implement and that they are able to handle arbitrary models. Unfortunately, their main disadvantage is speed, they are often significantly lower than the deterministic methods, what makes them unsuitable for large model and/or large datasets. In [9] Doucet *et al.* present the Rao–Blackwellised Particle Filtering algorithm, which main idea is to integrate out some of the variables using exact inference, and apply stochastic to the remaining ones.

4.3 Learning

Creating dynamic Bayesian networks can be a complicated task. One way to construct Bayesian networks is from domain knowledge. However, in many domains, domain knowledge is not sufficient to construct a complete Bayesian network. In this case, Bayesian networks can be learned from data when data are available. The learning problem can be categorized into two groups depending on the knowledge about the structure: 1) parameter learning problem when the structure is known, and 2) structure learning problem when the structure is unknown, where the parameter learning is a part of the structure learning problem and is used as an inner loop of structure learning. Generally, the techniques for learning DBNs are the same as the techniques for learning static Bayesian networks.

Parameter learning. When the structure of the model is known, the learning task becomes one of parameter estimation. In case when we have a full sampling data, we determine the probability distibutions by computing statistics from the data samples. We want to find the values of the parameters of each CPD that maximize the likelihood of the training data, containing S independent sequences. Each of the sequences has the observed values of all n nodes per slice for each of T slices. When N is small compared to the number of parameters that we are estimating, we use the Maximum A Posterori (MAP) estimates rather than the Maximum Likelihood (ML) estimates.

In case when there are hidden variables and/or missing data, exact methods for computing the probablity distributions become intractable. In such case, we must use iterative methods, such expectation–maximization (EM) [15] or gradient ascent [5] algorithm to find a local maximum of the ML/MAP function. The key similarity between them is that the information they require is computed with the inference routines. Another similarity is that in general, both are guaranteed to find only a local optimum in the parameter space. The gradient ascent techniques have the advantage of greater generality, while the EM algorithm has the advantages of simplicity and robustness. Gradient ascent can be thought of as moving a point corresponding to the parameter values through parameter space so as to maximize the likelihood function. The EM algorithm consists of two major steps: an expectation step followed by a maximization step. The expectation is with respect to the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observation. The maximization step then provides a new estimate of the parameters. These two steps are iterated until convergence.

The parameter learning algorithm implemented in the SMILE$^{\smiley}$ library is based on fact that a DBN has a limited number of CPDs that need to be learned. By decomposing the DBN into several BNs we can use existing BN learning algorithms. The DBN is unrolled for $k+1$ time–slices, where k denotes the temporal order of the Markov process. Then, the unrolled DBN is decomposed into separate BNs accountable for the initial paramaters and the temporal parameters.

Structure learning. If we know a number and type of some states in the network, but we do not know their relations and mutual independence, according to [13] we need a structure learning algorithm searching over the space of possible, alternative structures to identify the one (or those) having the highest score. by the data. This requires a scoring function for candidate structures and an efficient search procedure, since the list of potential structures grows exponentially with the number of nodes. In addition to the computational cost, another important consideration is the amount of data that is required to reliably learn structure. But, in practise we can reduce the data requiremetns considerably, because we often have strong prior knowledge about the structure or at least parts of it. Dojer [8] proposed an algorithm for finding an optimal structure of the BN from data, relying on the relaxation of the acyclity constraint. While the unrolled DBN is always acyclic, we can use this algorithm also for learning the DBN structure.

Friedman [10,11] propose one method for learning both the network structure and the parameters from partially observed data the structural expectation–maximization algorithm (SEM). It can be described as an iteration of following steps: 1) adding a new node to the network, representing a hidden variable, and 2) finding as good as possible network connections for given set of nodes. These steps are repeated as long as the network keeps improving.

5. Summary

Because majority of events encountered in every day life are described by sets of observations taken in successive moments of time, we need models capable of dealing with temporal dependencies. This paper concentrats on dynamic Bayesian networks, an temporal extension of Bayesian networks that allow to model dynamical processes. It illustrates, by mean of an example, different cases of the inference in DBNs and shortly described learning concepts depending on given knowledge. Most practical uses of DBNs involve temporal influences of the first order, i.e., influences between neighboring time steps. It is likely that this can lead to incomplete or even erroneous results. This paper presents usage of higher orders dynamic Baseyian networks and shows that introducing influences of states earlier than the state immediately preceding the current state can improve accuracy of inference, not only in case of prediction but also in case of smoothing and filtering. In addition, this article shortly presents a description of different approaches dealing with dynamical processes.

Acknowledgments

The empirical part of the paper was performed using SMILE☺, an inference engine, and GeNIe, a development environment for reasoning in graphical probabilistic models, both developed at the Decision Systems Laboratory, University of Pittsburgh, and available at `http://genie.sis.pitt.edu/`.

References

[1] Constantin Aliferis and Gregory Cooper. A Structurally and Temporally Extended Bayesian Belief Network Model: Definitions, Properties, and Modeling Techniques. In *Proceedings of the Twelfth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 28–39, San Francisco, CA, 1996. Morgan Kaufmann.

[2] Gustavo Arroyo-Figueroa and Luis Enrique Sucar. Temporal Bayesian Network of Events for Diagnosis and Prediction in Dynamic Domains. *Applied Intelligence*, 23:77–86, October 2005.

[3] Jon .L. Bentley. *Programming pearls*. ACM Press Series. Addison-Wesley, 2000.

[4] Carlo Berzuini. Representing time in causal probabilistic networks. In *Uncertainty in Artificial Intelligence 5 Annual Conference on Uncertainty in Artificial Intelligence (UAI-89)*, pages 15–28, Amsterdam, NL, 1989. Elsevier Science.

[5] John Binder, Daphne Koller, Stuart Russell, Keiji Kanazawa, and Padhraic Smyth. Adaptive Probabilistic Networks with Hidden Variables. In *Machine Learning*, pages 213–244, 1997.

[6] Xavier Boyen and Daphne Koller. Approximate Learning of Dynamic Models. In *IN NIPS-11*, pages 396–402. MIT Press, 1998.

[7] Thomas Dean and Keiji Kanazawa. A Model for Reasoning About Persistence and Causation. *Computational Intelligence*, 5(2):142–150, 1989.

[8] Norbert Dojer. An Efficient Algorithm for Learning Bayesian Networks from Data. *Fundam. Inf.*, 103(1-4):53–67, January 2010.

[9] Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao–Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 176–183, San Francisco, CA, 2000. Morgan Kaufmann.

[10] Nir Friedman. Learning Belief Networks in the Presence of Missing Values and Hidden Variables. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 125–133. Morgan Kaufmann, 1997.

[11] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the Structure of Dynamic Probabilistic Networks. pages 139–147. Morgan Kaufmann, 1998.

[12] Severino F. Galán, Francisco Aguado, Francisco Javier Díez, and José Mira. Nasonet, Modeling the Spread of Nasopharyngeal Cancer with Networks of Probabilistic Events in Discrete Time. *Artificial Intelligence in Medicine*, 25(3):247–264, 2002.

[13] David Heckerman. A tutorial on learning with bayesian networks. Technical report, Learning in Graphical Models, 1996.

[14] Eugene Santos Jr. and Joel D. Young. Probabilistic temporal networks. Technical report, Air Force Institute of Technology, 1996.

[15] Steffen L. Lauritzen. The EM Algorithm for Graphical Association Models with Missing Data. *Computational Statistics and Data Analysis*, 19:191–201, February 1995.

[16] Kevin Murphy and Saira Mian. Modelling gene expression data using dynamic bayesian networks. Technical report, 1999.

[17] Kevin Murphy and Yair Weiss. The Factored Frontier Algorithm for Approximate Inference in DBNs. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 378–385, San Francisco, CA, 2001. Morgan Kaufmann.

[18] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.d. dissertation, University of California Berkeley, Computer Science Division, 2002.

[19] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *In Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[20] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann PUBLISHERs, Inc., San Mateo, CA, 1988.

[21] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[22] Geoffrey Zweig and Stuart Russell. Speech recognition with dynamic bayesian networks. Technical report, 1998.

MODELOWANIE SYSTEMÓW DYNAMICZNYCH PRZY UŻYCIU DYNAMICZNYCH SIECI BAYESOWSKICH

Streszczenie: Sieci Bayesowskie (Bayesian networks, BNs) są popularnym narzędziem do reprezentacji wiedzy w warunkach niepewności. Znalazły praktyczne zastosowanie w wielu dziedzinach. Ich rozszerzenie o domenę czasową, dynamiczne sieci bayesowskie (dynamic Bayesian networks, DBNs) umożliwiają monitorowanie oraz aktualizację systemów zmieniających się wraz z upływem czasu, a także predykcję przyszłego stanu takiego systemu. Większość praktycznych zastosowań dynamicznych sieci Bayesowskich bierze pod uwagę tylko zależności pierwszego rzędu, to znaczy, że bieżący stan systemu zależy tylko od jego stanu w bezpośrednio poprzedzającym go kroku czasowym. Takie założenie jest uproszczeniem, wynikającym najprawdopodobniej z braku efektywnych narzędzi zdolnych obsłużyć modele wyższych rzędów. Niniejszy artykuł przedstawia na przykładzie sposób w jakim tworzy się modele wyższych rzędów oraz pokazuje, wpływy wyższych rzędów mogą zwiększyć jakość modelu.

Słowa kluczowe: sieci bayesowskie, zależności temporalne, modele dynamiczne, dynamiczne sieci bayesowskie