Research on Modeling the Nonhomogenous Markov Decision Systems with Dynamic Bayesian Network

Guo Junwen ,Qin zheng, Heng Xingchen,

School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Summary
For the purpose of the further wide application of dynamic Bayesian networks（DBNs) to many real complex systems, a new approach was presented to improve the modeling of the Non time homogenous Markov Decision systems with DBNs, in which the extended hidden variables were introduced into the evolutional process to build Markov models required by the hypothesis conditions, a structure learning algorithm of DBNs was given from the incomplete data set and when the extended hidden variables are existed. The sufficient statistics of the subsequent time slices were estimated using Bayesian probability statistical method, and then the time-variant transition probabilities were learned using both of current sufficient statistics and estimated sufficient statistics. The theoretical analysis and simulation results show that the proposed approach is valid.
Key words:
Dynamic Bayesian network, Markov model, hidden variables, Bayesian probability statistical method

Introduction

Bayesian network is known as probabilistic network, Bayesian belief network or causal network. It combines graph theory with probability to express complex uncertainty among random variables. Bayesian Network has been developed well as a kind of uncertain reference method.

While Dynamic Bayesian Networks (DBN) [1,2,3,4] is a species of Bayesian networks (BN) designed to model stochastic temporal processes, which models the stochastic evolution of a set of random variables over time. Owing to DBN's significant advantages in describing nonlinear, temporal, evolving and uncertain relationships and strong ability of probabilistic inference, studies on modeling, learning and inference of DBN have been developed widely. And also, DBN have been used for many purposes in different fields. The widely used Hidden Markov Models (HMMs) [5] and Kalman filters could be looked upon as the special forms of DBN. Literature [7] showed that DBN could be better than HMMs on standard speech recognition tasks.

However DBN is more complex than BN, and a few problems of DBN still need to be researched further. In order to simplify the modeling, learning and inference of DBN, almost all the studies on DBN share two common assumptions. One is Markov assumption [6, 8, 9], which assumes that the future is conditionally independent of the past given the current state. The other is time-invariant assumption, i.e., the transition probabilities and dependence relationship among variables are independent of time. In many complex systems, such as in macroeconomic system, disease diagnosis system etc, evolution of the observed partial variables does not accord with Markov process, although the underlying model maybe accord with Markov process. Thus we could not establish Markov models with these partial variables. Similarly, the transition probabilities usually vary with time and do not meet the time-invariant assumption in the above complex systems. These two assumptions, to great degree, have constrained the application of DBN in these systems.

Literatures [11, 12] introduced hidden variables to primary DBN model as partial state information so as to build Markov models, and time-variant transfer probability model through polynomial curve-fitting method. In 1998 Friedman even extended SEM (Structural EM) [10] to the DBN structure learning in the presence of hidden variables. However, the introduction of hidden variables in literatures[11,12] didn't involve the change of state variable set and qualitative dependence relationship between state variables with time , didn't take advantage of Bayesian probability method as the base of DBN inference to build transfer probability model, and didn't give out experimental results of modeling the complex system. While SEM algorithm still didn't consider random variable set and qualitative dependence relationship between state variables with time, and didn't solve the problem of local optimum. At present little research work on how to relax these three assumptions has been developed

This paper makes efforts to relax the above two assumptions, and applies DBN to modeling complex systems in which the two assumptions are not satisfied. For this purpose, the paper extends the concept of the hidden variables described in literature [11], and studies on how to construct Markov models with extended hidden

Manuscript received June 25, 2006.
Manuscript revised June 30 , 2006.

variables, how to learn the structure of DBN with hidden nodes, and how to build time-variant transition probability models have been developed in the paper. The theoretical analysis and experiment demonstrate the validity of the methods.

2. Dynamic Bayesian Network Model

A BN describes a probability distribution over a fixed set of variables. DBN extends this representation to model temporal processes. We use capital letters, such as *X, Y,Z* for variable names and lowercase letters *x, y, z* to denote specific values taken by those variables. Sets of variables are denoted by boldface capital letters X, Y, Z etc, with sets of values denoted by boldface lowercase letters x, y, z etc. Assume that changes occur between discrete time slices that are indexed by the non-negative integers and that X = {X_1, X2, …X_n} is a set of attributes that evolves with the process changes. X_i[t] is a random variable that denotes the value of the attribute X_i at time t, and X[t] is the set of random variables X_i[t]. The probability distribution over all the random variables can be represented as follows:

$$P(X[1],X[2]...X[t]) = P(X[1])P(X[2]|X[1])...P(X[t]|X[1],...,X[t-1]) \quad (1)$$

Obviously, such a distribution can be extremely complex. For simplicity of DBN's modeling, learning and inference, two assumptions are introduced. One is Markov assumption *I*(X [*t* + 1], *{*X [1]… *X* [*t* -1]*}* |X[*t*]), which means that the state variables at each time slice are only dependent on the state of the last time slice. Given the Markov assumption, the Eq. (2) can be rewritten as:

$$P(X[1],X[2]...X[t]) = P(X[1])P(X[2]|X[1])...P(X[t]|X[t\text{-}1]) \quad (2)$$

Within a finite interval 0...*T*, *DBN* can be notionally "unrolled" into a BN over X[0]…X[*T*].The joint distribution over X[0], . . . ,X[*t*] is:

$$P(X[0],...,X[T]) = P(X[0]) = \prod_{0}^{T-1} P(X[t+1]|X[t]) \quad (3)$$

The other assumption introduced is time-invariant assumption, i.e., the transition probability *P*(**X** [*t* + 1]|**X**[*t*]) and the dependence relationship among variables are independent of *t* and does not vary with time. Given the assumption, a DBN can be simplified further into two parts: A prior network *B*0 that specifies a distribution over initial states **X**[0]; and A transition network $B_{\rightarrow}$ over the variables **X**[0] ∪**X**[1] that is taken to specify the transition probability for all *t*. Thus, a DBN can be defined by a pair (B_0, $B_{\rightarrow}$).The transition probability can be computed as follows:

$$P(X[1])|X[0]) = \prod_{i=1}^{n} P_{B\rightarrow}((X_i[1]|pa(X_i[1])) \quad (4)$$

Where pa(*Xi*[1]) denotes the value of the parent node set of X_i[1]. Hence the joint distribution of a DBN over **X**[0], . . . ,**X**[*T*] can be simply represented as follows:

$$P(X[0]...X[T]) = P_{B_0}(X[0])\, P^{T}_{B\rightarrow}(X[1]|X[0]) \quad (5)$$

The introduction of the above two assumptions makes the DBN' modeling and learning be very easy. Only two very simple networks-prior network and transition network have to be handled. There exist many real application scenes in which these assumptions are not satisfied, so it is necessary to study the modeling problem of DBN in these application scenes. The solutions are to make the Markov model by adding extended hidden variables to the model as for the process that is not Markov. Furthermore, as for the transition probabilities that vary with time, they are constructed directly from dataset time-variant transition probability models.

3. DBN Model with Hidden Variables

Because it is not sure that the values of all variables in DBN model can be observed in some real applications, such as disease diagnose, situation assessment, tracking mobile robot and so on. Though the DBN model can satisfy the process, partial variables can only be observed and these partial variables can't assure that the variables in t time slice are irrelevant with these variables in t+1 time slice. So the Markov assumption isn't satisfied and the model which is built with these partial variables will not be Markov model. We consider adding hidden variables to evolutional process. Here a formal definition is given as follows.

Definition 1 If some accessorial variables are added to the DBN model as partial state information the dependence relationship of the variable set on the variables in several foregoing time slices could be transferred through the accessorial variables. So the variable set in each time slice is directly dependent on the variables in the last time slice, and the Markov assumption is satisfied. The added accessorial variables are called hidden variables.[11]

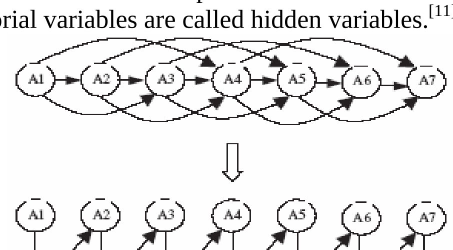


Fig. 1 An example of adding hidden variables to construct Markov model

Fig.1 is an example of adding hidden variables to construct Markov model. Where *A*1,*A*2, . . . , *A*7 represent the evolution sequence of variable *A* in 7 time slices, and the variable *Ai* (*i* = 1, 2, . . . , 7) in each time slice is directly dependent on the variables in the preceding *m* time slices (Here, *m* equals to 3.). Hence, the evolution sequence does not meet the Markov assumption. Our solution to this problem is to add hidden variables, such as *H*1,*H*2, . . . , *H*7 in figure1. Then a new evolution sequence (*A*1,*H*1), (*A*2,*H*2), . . . , (*A*7,*H*7) is formed, which satisfies Markov process and at the same time, can represent the dependence relationship among *A*1,*A*2, . . . , *A*7 through the hidden variables *H*1,*H*2, . . . , *H*7.

Theorem 1 After hidden variables are introduced, no matter what long original DBN model's evolution sequence is, as long as the evolution process doesn't satisfy the Markov assumption, we can construct Markov model through adding hidden variables without increasing the complexity of the original DBN model.

Proof: It is proved that the complexity of the network model is the exponential function of the degree of the model (the maximal summation of in-degree and out-degree of each variable in the model) [13].

We use the *m* to represent the degree of model, the exponential function *f(m)* to represent the complexity of the network model and the *l* to represent the evolution sequence length. Suppose that $l \geqslant 2n + 1$, the original DBN's degree of model *m* is equal to 2*n. After hidden variables are introduced, it is confirmed that the degree of model *m*' is equal to 4 no matter what value *l* takes as shown in figure1. Therefore, if *n* > 2 and *m* >= *m'*, *f(m')* <= *f(m)* can be gotten by the character of exponential function. So the original model's *f(m)* doesn't increase.(Please notice that *n*= 2 is the minimal value causing that the evolution sequence does not meet the Markov assumption).

Another implied assumption of DBN is that random variable set and qualitative dependence relationship between state variables don't vary with time. However, if this assumption is followed strictly in some complex systems, structure learning of DBN is hard to fit the real probability distribution. Hence, the paper extends original DBN model shown in Fig. 2.

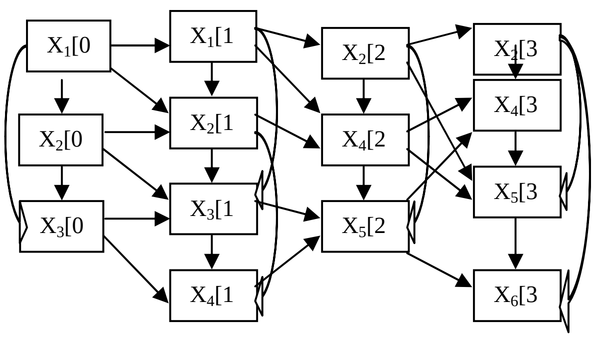


Fig. 2 The Extended DBN Model

The structure of prior network B_0 and transition network $B_{->}$ vary in different time slice. Firstly, the numbers and definitions of random variables in every time slice are different. For example, there are four variables $X_1[1]$, $X_2[1]$, $X_3[1]$, $X_4[1]$ in the time slice 1, but these variables are substituted by $X_2[2]$, $X_4[2]$, $X_5[2]$ in time slice 2. Secondly, the dependence relationship between $X_1[1]$ and $X_3[1]$ in time slice 1 doesn't exist in time slice 2. So it is necessary to learn the structure of initial network B_0 and transition network $B_{->}$ in every time slice again. To adapt to the character of change of DBN structure in evolution process, the paper extends the concept of the hidden variables described as follows:

Definition 2 Suppose that a random variable *v* begins to have effect on DBN's global probability distribution p_g or have no effect on p_g any more, only when DBN's a local probability distribution p_l reaches the specified state *s* or the specified evidence *e* occurs. And before or after that, the random variable *v* isn't considered and isn't involved in the evidence propagation among nodes. So the random variable *v* is also called hidden variable.

It is concluded from definition1 and definition2 that the hidden variable has two functions: 1) transferring dependence relationship between variables. 2) satisfying the change of the number of the variable set and the qualitative dependence relationship among variables with time. For the structure learning of DBN with these hidden variables, the paper introduces genetic algorithm to EM algorithm, the EM_GA algorithm so as to solve the problem of local optimum and speed up the convergence of EM algorithm. Furthermore, the changes of DBN structure with time are considered through the distribution of all random variables on whole time slices given by experts in advance and learning the structure iteratively over every time slice .

As soon as a time slice *t* comes, the structure of DBN begins to be updated. At first the variable set in the time slice *t* is updated according to the distribution of variable set over all time slices provided by experts. Then, if the current sample data of random variable set are complete, DBN's structure is learned using classical BDE algorithm, otherwise using SEM_GA algorithm. After that, if the time slice t doesn't end up, these sample data and prior information of next time slice will be supplemented and necessary sampling initialization will be processed ahead. When the next time slice comes, the structure learning continues again.

The learning of DBN composes of two parts. One is learning the structure of B_0; the other is learning the structure of $B_{->}$.The whole process of learning the B_0 using SEM-GA algorithm is as follows:

Step1 Build initial network structure group Q, complete the incomplete dataset *D* using the current network B_0^c and EM algorithm, and get the complete dataset *Dc*.

Step2 As for the current evolutional group Q, do the crossover or mutation operations according to the crossover probability P_c and mutation probability P_m, and get the evolved group Q'.
Step3 As for each network B_0 in Q', examine if network B_0 is a directed acyclic graph. If it is not, network B_0 is assigned a smaller fitness; otherwise calculate the fitness F_{B0} according to the following formula: F_{B0} =*MDL score*(B_0 : D_c), where MDL score (B_0: D_c) is the Minimal Description Length (MDL) score of network B_0 and dataset D_c.
Step4 Choose from Q' λ individuals have the highest fitness to form the next generation Q, where λ represents the size of the evolutionary group.
Step5 Select B_0', make $F_{B0}'_{\_} = \underset{B_0}{\mathrm{Max}} \arg(F_{B0})$. If $F_{B0}' > F_{B0}^c$, then $B_0^c = B_0'_{\_}$.
Step6 If the terminative condition of the algorithm is satisfied, then quit; otherwise, go to Step 1 and continue the above process.

The process of learning the $B_{->}$ resembles the process of learning the B_0.Because we can consider DBN's structure learning as combination optimization problem of random variables in DBN model with a score metric and this problem resembles TSP, the paper introduces operators used in TSP to SEM_GA algorithm and good effect is gotten in final simulation experiment.

4. Time-variant Transition Probability Models

As the structure of the transition network $B_{->}$ varies with time slices, transition probabilities vary with them. All of those do not meet the time-invariant assumption. Learning DBN will become much more difficult. As we have known, the probabilistic learning of BN or DBN is in essence the compromise of the prior belief and the sufficient statistics from dataset. Since the prior belief could be given by users or field experts, the key to learn time-variant transition probabilities is to obtain the time variant sufficient statistics from dataset [14].

The time-variant sufficient statistics can be decomposed into two sets. One includes the sufficient statistics of current time slices in the dataset, which could be directly computed from the dataset. The other contains the sufficient statistics of posterior time slices not existing in the dataset, which could not be directly computed from the dataset. Our solution is to estimate the sufficient statistics of posterior time slices from the sufficient statistics of the current time slices using Bayesian probability statistics method. At last, we can learn the time variant transition probabilities with both current sufficient statistics and estimated sufficient statistics. The process of learning time-variant transition probabilities is shown in Fig. 3.

The sufficient statistics of the discrete random variables can be stored in conditional frequency matrix S. All the conditional frequency matrixes comprise a matrix sequence S_0, S_1,…, S_T , where S_t(t = 0, 1, . . . , T) represents the conditional frequency matrix at the time slice t. The element S_{ijt}(i, j = 0, 1, . . .,N; t = 0, 1, . . . , T) in the matrixes denotes the times of that X[t + 1] takes the i^{th} value in the dataset given X[t] being in the j_{th} state. For each position (i, j) in the matrixes, the elements S_{ijt}(i, j = 0, 1, . . .,N; t = 0, 1, . . . , T) also form a value sequences. So we must consider the problem of how to estimate theses values of the elements S_{ijt} in the subsequent slices ($t > T$) that do not exist in the current value sequence.

The paper expends the general process of Bayesian probability statistical method to estimating the value of element S_{ijt} in posterior time slice t (t>T). Now the above problem can be transformed into estimating the value of element S_{ijt} in the $(T+1)_{th}$ time slice through these values of elements S_{ijt} in foregoing T time slices. Some definitions are defined as follows:
Definition 3 The elements sequence S_{ij0},…S_{ijT} which is composed of elements S_{ijt} of matrix S in T time slices is represented by the variables set X=($X_{1,\ldots\ldots}X_T$).It is assumed that a stochastic sample D={ $X_{1,\ldots\ldots}X_T$ } can be gotten from the physical joint probability distribution of **X**. D's a element X_i denotes sample's an observed value. X_i is called a case.
Definition 4 If random variable set X's m kinds of states correspond to m kinds of model structure, and according to assumptions of m kinds of model structure decomposed X's physical joint probability distribution is expressed by m^h, p(m) is called as probability distribution of m kinds of model structure.
Definition 5 If there is a variable Θ_m which takes vector value corresponding to a parameter vector θ_m for every kind of model structure m ,uncertainty of θ_m is expressed by a prior probability density function p(θ_m|m).

Given a stochastic sample set D, the posterior distribution of m and θ_m will be computed with Bayesian probability statistics method as follows:

$$p(m|D) = \frac{p(m)\,p(D|m)}{\sum_{m'} p(m')\,p(D|m')} \qquad (6)$$

$$p(\theta_m|D,m) = \frac{p(\theta_m|m)\,p(D|\theta_m,m)}{p(D|m)} \qquad (7)$$

Where $p(D|m) = \int p(D|\theta_m,m)p(\theta_m|m)d\theta_m$

It is assumed that the physical joint probability distribution of variables set $X=(X_{1,\ldots\ldots}X_T)$ can be applied to certain model structure m.

$$p(x|\theta_m, m) = \prod_{i=1}^{n} p(x_i|pa_i, \theta_i, m) \quad (8)$$

Where $p(x_i|pa_i, \theta_i, m)$ is the power series. The pa_i represents the modeling of the corresponding variable x_i's parent variables. The θ_i represents local possible parameter of variable x_i. When every $X_i \in X$ is discrete variable, it has r_i possible values $x_i^1, \ldots x_i^{ri}$, and every kind of local possibility is a polynomial collection. A joint distribution of X to each modeling pa_i is:

$$p(x_i^k|pa_i^j, \theta_i, m) = \theta_{i,j,k} > 0 \quad (9)$$

Where $pa_i^1, \ldots, pa_i^{q_i}$ $(q_i = \prod_{r_i \in pa_i} r_i)$ expresses q_i kinds of modeling pa_i of the variable x_i, θ_i is parameter and

$$\theta_i = ((\theta_{ijk})_{k=2}^{r_i})_{j=1}^{q_i}$$

For convenient description, the following definition is made:

$\theta_{ij} = (\theta_{ij1}, \theta_{ij2}, \ldots, \theta_{ijr_i})$

$(i=1, 2, \ldots, T; j = 1, 2, \ldots, q_i)$

After the above local distribution function is given, two assumptions are needed so that the posterior distribution $p(\theta_m| D, m)$ can be computed in closed form. They are described as follows:

(1) The stochastic sample set D is complete.

(2) Each parameter vectors θ_{ij} is independent of others.

So every parameter vector θ_{ij} can be updated independently. It is assumed that every parameter θ_{ij} has prior the Dirichlet distribution Dir $(\theta_{ij} | \alpha_{ij1}, \alpha_{ij2}, \ldots, \alpha_{ijri})$. So posterior distribution can be gotten as follows:

$$p(\theta_{ij}|D,m) = Dir(\theta_{ij}|\alpha_{ij1} + \alpha_{ij2}, \ldots, + \alpha_{ijr_i} + N_{ijr_i}) \quad (10)$$

Where N_{ijk} is the number of cases in D when $X_i=x_i^k$ and $Pa_i=pa_i^j$.

Now some important predications can be gotten using the mean of θ_m. For example, we estimate the value of the element S_{ijt} in the $(T+1)_{th}$ time slice as follows:

$$p(X_{N+1}|D,m) = \underset{P(\theta_m|D,S^k)}{E}\left(\prod_{i=1}^{r_i} \theta_{ijk}\right) \quad (11)$$

The mathematics expectation can be computed as follows by use of the parameter θ_m which is independent of D:

$$p(X_{N+1}|D,S^k) = \int \prod_{i=1}^{n} \theta_{ijk} p(\theta_{ij}|D,S^k) d\theta = \prod_{i=1}^{n} \int \theta_{ijk} p(\theta_{ij}|D,S^k) d\theta_{ij} \quad (12)$$

The final result is:

$$p(X_{N+1}|D,S^k) = \prod_{j=1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \quad (13)$$

Where $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}, N_{ij} = \sum_{k=1}^{r_i} N$

Because unconstraint multinomial distribution belongs to the family of index, the computation of the equation (13) is simple. So the value of the element S_{ijt} in the $(T+1)_{th}$ time slice can be gotten, and then the complete condition frequency matrix S_t of the $(T+1)_{th}$ time slice can be gotten through collecting all estimated values of these elements $S_{ijt\ (i,j=0,1\ldots N, t=T+1)}$ .At last, we can learn the time variant transition probabilities with both current sufficient statistics and estimated sufficient statistics.

5. Experimental Analysis

For validating the effect of the new methods, a preliminary simulation experiment for the battlefield situation assessment has been made. The battlefield situation simulated by using the battlefield scenario editor software which is developed by us is shown in Fig. 3.

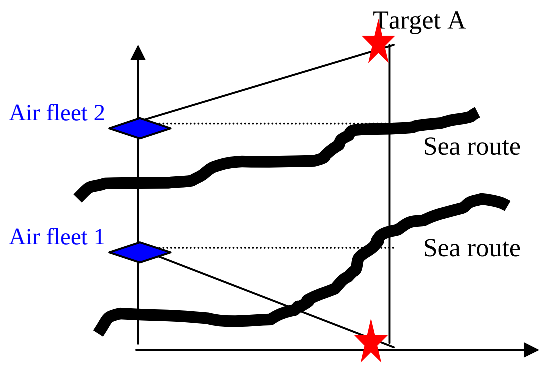


Fig. 3 The simulation scenario for battlefield situation

In Fig. 3 the blue rhombic frame represents enemy air fleet, the red star represents the target which we need to protect from being assaulted, and the black bold line represents the sea route of the enemy naval ship. For recognizing the tactics intention of enemy in emulation environment, the dynamic Bayesian network model for situation assessment is constructed by using the improved method proposed in the paper, which composes of the two kinds of state variables: character events and classified assumptions of battlefield situation, and then the DBN's approximate reference mechanism is applied to identify the situation classified assumption with the most posterior probability. The part of this situation assessment model is shown in Fig. 4, in which the node E represents the character event and the node S, P and R represent three different situation assumptions: assault, feint and recovery respectively. Here, the change of the character event node E's states are caused by the change of the hidden nodes' states, and there is not direct relationship among these character event nodes that are all conditionally dependent on three situation classified assumptions nodes above respectively.

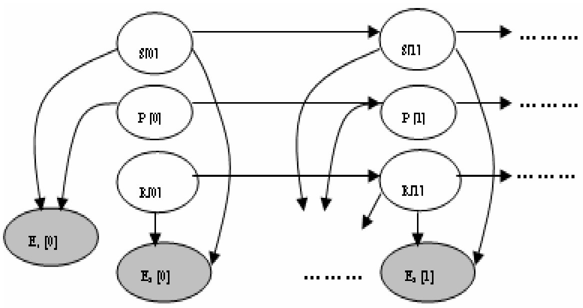
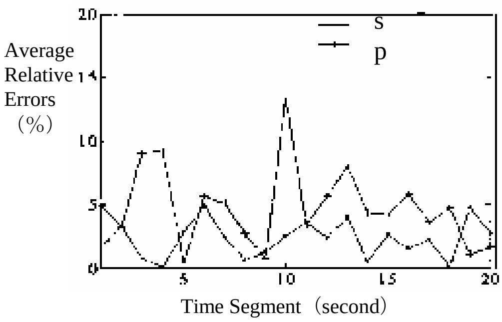
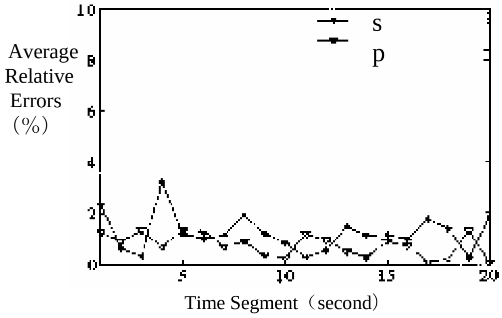


Fig. 4 The dynamic Bayesian network model for situation assessment

We choose two random variables S(joint assault), P(feint), which have important value for assessing the whole battlefield situation, to be made the error analysis. As is shown in figure5 and figure 6, the random variable S means the enemy air fleet 1 and 2 will assault target A or B simultaneously; the random variable P means an enemy air fleet pretends to assault our target A., after puzzling our commanders, the enemy air fleet tries to assault target B together with the other air fleet. The random variable S has three states: assaulting target A simultaneously, assaulting target B simultaneously and assaulting target A and target B simultaneously respectively; The random variable P has two states: assaulting target A but pretending to assault target B and assaulting target B but pretending to assault target A. The results of the two variables' error analysis are shown in Fig. 5.



(a) 100 situation samples



(b) 500 situation samples

Fig. 5 The average relative errors of state variable S and P

Fig. 5(a) shows the average relative errors of the edge probabilities of variable S and P using the above algorithms when 100 situation samples are chosen. Fig. 5(b) shows the average relative error of the edge probabilities of variable S and P when 500 situation samples are chosen. This experiment shows that when the more samples are chosen, the better effect of the new methods is obtained and the errors in every time segment can be endured in principle. Meanwhile the wavy scope of errors becomes smaller and smaller with time running and the speed of convergence becomes faster when more samples are chosen. This indicate exactly that when our commanders know more and more information of enemy air fleets with evolvement of the battlefield situation, they can recognize enemy intention more accurately.

The experiment shows that extended DBN can model effectively complex battlefield situation, which builds stability for the application of DBN in other complex system. But some problems are also found in experiment, including that how to speed up the convergence of average relative error to predict enemy intention with less time, how to solve the problem of the large errors with only a few samples, and whether a general multiple linear regression model exists, which describes the relationship of the number of sample, average relative errors and time slices.

6. Experimental Analysis

Almost all the current studies on DBN share the two common assumptions, Markov assumption and time invariant transition probability assumption. The paper proposes the solution to modeling the complex system with DBN when the two assumptions are not satisfied. These two assumptions, to great degree, have constrained the application scenes of DBN. This paper makes efforts to relax the two assumptions so as to construct Markov models with the extended hidden variables and build time-variant transition probability models using Bayesian probability statistics method. Meanwhile an effective structure learning algorithm for DBN with extended hidden variables is also proposed. Finally a preliminary experiment designed to test the soundness of the methods proposed.

The experiment shows that improved DBN model can effectively model the complex battlefield situation and realize the prediction of enemy intention in advance. The research work of the paper makes a little contribution to the application of DBN in other complex systems. The future work is to research further several qualitative and quantitative problems found in experiment and do experimental analysis with real complex systems.

References

[1] F. V. Jensen. An Introduction to Bayesian Networks. London: UCL Press, 1996

[2] Eugene Santos, Jr. and Joel D. Young. Probabilistic temporal networks: A unified framework for reasoning with time and uncertainty. Intl. J. of Approximate Reasoning, 20(3):191-216, 1999.

[3] C. F. Aliferis and G. F. Cooper. A Structurally and Temporally Extended Bayesian Belief Netwok Model: Definitions, Properties and Modeling Techniques. In UAI, 1995.

[4] P. Dagum and A. Galper. Time-series prediction using belief network models. Intl. J. of Human-Computer Studies, 42:617-632, 1995.

[5] L. Rabiner and B. Juang. An introduction to hidden Markov models. IEEE Acoustics, Speech & Signal Processing, 1986.

[6] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. Proc. 14th Conf. On Uncertainty in Artificial Intelligence, pages 139-147, 1998.

[7] G. Zweig and S. Russell. Speech recognition with dynamic bayesian networks. In AAAI98, 1998.

[8] A. E. Nicholson and J. M. Brady. The data association problem when monitoring robot vehicles using dynamic belief networks. 10th European Conference on Artificial Intelligence Proceedings, pages: 689-693, 1992.

[9] Gregory F. Cooper, Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. In Machining Learning 1992.9, 309-347.

[10] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In ICML97, 1997. Vanderbilt University, Morgan Kaufmann Publishers

[11] F. Tian, Y. Lu, and C. Shi. Learning bayesian networks with hidden variables using the combination of em and evolutionary algorithm. PAKDD 2001, Hong Kong, China, Aril, 2001 proceeding, pages 568–574, 2001.

[12] W. Lam and F. Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. Computational Intelligence, 10:269–293, 1994.

[13] F. Tian, H. Zhang, Y. Lu, and C. Shi. Inference and modeling of multiply sectioned bayesian networks. IEEE TENCON02, Beijing, China, October 28-31, 2002 proceeding, pages 683-686, 2002.

[14] Tian Fengzhan and Zhang Hongwei, "Learning and Approximate Inference of DBNs", Proceedings of the 4h World Congress on Intelligent Control and Automation, June 10-14, 2002.