ISSN : 0974 - 7435 Volume 10 Issue 23

2014

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 10(23), 2014 [14558-14564]

Forwards-backwards algorithm based on parallel factors

Haiyang Chen*, Hongying Nie, Ruirui Mao
School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, (CHINA)
E-mail : chy_00@163.com

ABSTRACT

The study about highly efficient inference algorithms on dynamic Bayesian networks has become one of the focuses in the area of artificial intelligence. In order to improve the inference efficient of the improved forwards-backwards algorithm, we proposed the forwards-backwards algorithm based on parallel factors. After explaining the basic idea of the parallel factors, we defined the parallel factors, then introduced them to the forwards pass and backwards pass to realize the computation step sharing and reduce the amount of computation further. It is proved by the simulation experiments that the forwards-backwards algorithm based on parallel factors is correct and efficient.

KEYWORDS

Forwards-backwards algorithm; Dynamic Bayesian networks; Inference; Efficient.

INTRODUCTION

Bayesian networks are a combination of graph theory and probability theory. As a decision analysis tool based on the theory of probability and statistics, it's very suitable for describing the nonlinear relationship and uncertainty caused by the random phenomenon in its application field, and it has strong ability of knowledge representation and probabilistic inference. Since it was proposed by Pearl in 1986[1], Bayesian networks have been widely used in software project, agricultural system, reliability analysis, risk assessment, threat situation assessment, and so on[2-5]. The research about the basic theory of Bayesian networks involves three aspects: structure learning, parameter learning and inference. For the research about the inference algorithm, static Bayesian networks have more achievements, mainly including message propagation[1], global conditioning[6], junction tree algorithm[7], symbolic probabilistic inference[8], arc reverse/node reduction algorithm[9], random sampling algorithm[10], and so on. Because of the limitations in the inference process on Bayesian networks, that is, during the whole inference process, time factors aren't considered, so dynamic Bayesian networks are developed[11], which are the extension of Bayesian Networks in time series modeling and also a kind of compressed representation of the complex random process[12]. Compared with the inference algorithms on static Bayesian networks, there are fewer research achievements about the inference algorithms on dynamic Bayesian networks at present. The classical inference algorithms on dynamic Bayesian networks include forwards-backwards algorithm[13], frontier algorithm[14], interface algorithm[15] and so on. The advantages of dynamic Bayesian networks are that it can use evidence information on different time slices to effectively reduce uncertainties in different levels of information synthesis inference, and the inference results can reflect the objective world better, however, with the increasing of the time slices, the inference time becomes longer and longer, therefore the research about inference efficiency of dynamic Bayesian networks has important practical significance. Among the various dynamic Bayesian networks models, the Hidden Markov Model is not only the most basic model, but is one of the most widely used models in the research and application field. It is widely used in speech recognition, biological sequence analysis, on-line handwritten Chinese character recognition and on-line hand-drawn graphics recognition. Forwards-backwards algorithm is a classical inference algorithm of the Hidden Markov model. But the algorithm can only handle evidence information which can exactly tell us in which state the observation variable is, which is called the hard evidence, however, in the practical application, we often obtain soft evidence, which can't exactly tell us in which state the observation variable is, but can only provides the probability of the observation variable in a certain state. In order to overcome the disadvantages that forwards-backwards algorithm cannot handle soft evidence and also cannot be used in generalized Hidden Markov Model. So the improved forwards-backwards algorithm is proposed[16], which provides the intuitive expression in handling the soft evidence and extends the application scope of the original algorithm. The disadvantage of this algorithm is that forwards recursive computation and backwards recursive computation processes are considered as two relatively independent processes and the computation step sharing isn't considered. In addition, there is repeated computation in each of the two processes, which will affect the computing efficiency of the algorithm. So this paper proposed forwards-backwards algorithm based on parallel factors, which contains two steps: the first step is introducing the novel computing method to the forwards and backwards recursive computing processes to avoid repeated computation during the two recursive processes; the second step is to realize computation step sharing by introducing parallel factors, which can reduce the complexity of the algorithm significantly.

FORWARDS-BACKWARDS ALGORITHM BASED ON PARALLEL FACTORS

Improved forwards-backwards algorithm

The structure of dynamic Bayesian networks with T time slices is shown (see, Figure 1). There is only one hidden node X and m observation nodes $Y^1, Y^2, \cdots, Y^m$ in each time slice.

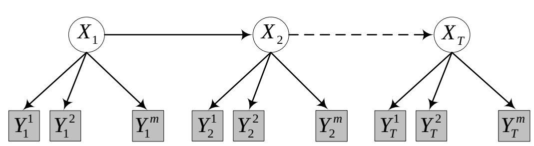

Figure 1 : Dynamic Bayesian networks

From[16], we have the recursive formula of the improved forwards algorithm as follows

$$\begin{aligned}\alpha_t(j) &= P(X_t = j \mid y_1^{1:mo}, y_2^{1:mo}, \cdots, y_t^{1:mo}) \\ &= \eta \sum_{y_t^{1s},\cdots,y_t^{ms}} \prod_{k=1}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks}) \sum_{i=1}^{n} P(X_t = j \mid X_{t-1} = i) P(X_{t-1} = i \mid y_1^{1:mo}, \cdots, y_{t-1}^{1:mo}), \qquad (1) \\ &= \eta \sum_{y_t^{1s},\cdots,y_t^{ms}} \prod_{k=1}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks}) \sum_{i=1}^{n} a_{ij} \alpha_{t-1}(i)\end{aligned}$$

where η is a normalizing constant, the subscript t in X_t and Y_t^k denotes time slices in which the variables are, the superscript k in Y_t^k denotes the serial number of the variable in the observation variable set, a_{ij} denotes the state transition probability of which is from the i'th state of X_{t-1} to the j'th state of X_t, $P(Y_t^k = y_t^{ks})$ denote probabilities of the observation variable Y_t^k which is in its s'th state, $y_k^{1:mo} = \{y_k^{1o}, y_k^{2o}, \cdots, y_k^{mo}\}$, y_k^{lo} denote the states in which the observation variable Y_k^l is.

As in Equation (1), when we sum $\sum_{y_t^{1s},\cdots,y_t^{ms}} \prod_{k=1}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks})$, there is repeated computation in $P(Y_t^k = y_t^{ks} \mid X_t = j)$ and $P(Y_t^k = y_t^{ks})$, so the forwards recursive computation needs to be optimized.

From[16], the recursive formula of the improved backwards algorithm is as follows:

$$\begin{aligned}
\beta_t(i) &= P(y_{t+1}^{1:mo}, y_{t+2}^{1:mo}, \cdots, y_T^{1:mo} \mid X_t = i) \\
&= \sum_{j=1}^{n} P(y_{t+2}^{1:mo}, \cdots, y_T^{1:mo} \mid X_{t+1} = j) P(y_{t+1}^{1:mo} \mid X_{t+1} = j) P(X_{t+1} = j \mid X_t = i), \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} \sum_{y_{t+1}^{1s},\cdots,y_{t+1}^{ms}} \prod_{k=1}^{m} P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{ks})
\end{aligned} \tag{2}$$

As in Equation (2), when we sum $\sum_{y_{t+1}^{1s},\cdots,y_{t+1}^{ms}} \prod_{k=1}^{m} P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{ks})$, there is repeated computation in $P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j)$ and $P(Y_{t+1}^k = y_{t+1}^{ks})$, so the backwards recursive computation needs to be optimized.

From above analyses, there is repeated computation in the forwards computation process and the backwards computation process of the improved forwards-backwards algorithm. In order to overcome the disadvantage of the algorithm, a novel computation method is needed to optimize these two recursive computation processes.

A new computing method

A new computing method is separately introduced into the forwards recursive formula and the backwards recursive formula to avoid the repeated computation and realize the optimization of the improved forwards and backwards algorithm.

Introduce a new computing method into the forwards and backwards formulas, we have

$$\begin{aligned}
\alpha_t(j) &= P(X_t = j \mid y_1^{1:mo}, y_2^{1:mo}, \cdots, y_t^{1:mo}) \\
&= \eta \sum_{y_t^{1s}, y_t^{2s}, \cdots, y_t^{ms}} \prod_{k=1}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks}) \sum_{i=1}^{n} a_{ij} \alpha_{t-1}(i) \\
&= \eta \sum_{y_t^{2s}, \cdots, y_t^{ms}} \prod_{k=2}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks}) \left[\sum_{p=1}^{s_1} P(Y_t^1 = y_t^{1p} \mid X_t = j) P(Y_t^1 = y_t^{1p}) \right] \sum_{i=1}^{n} a_{ij} \alpha_{t-1}(i) \quad , \\
&= \eta \sum_{y_t^{3s}, \cdots, y_t^{ms}} \prod_{k=3}^{m} P(Y_t^k = y_t^{ks} \mid X_t = j) P(Y_t^k = y_t^{ks}) \prod_{k=1}^{2} \left[\sum_{p=1}^{s_k} P(Y_t^k = y_t^{kp} \mid X_t = j) P(Y_t^k = y_t^{kp}) \right] \sum_{i=1}^{n} a_{ij} \alpha_{t-1}(i) \\
&= \eta \prod_{k=1}^{m} \left[\sum_{p=1}^{s_k} P(Y_t^k = y_t^{kp} \mid X_t = j) P(Y_t^k = y_t^{kp}) \right] \sum_{i=1}^{n} a_{ij} \alpha_{t-1}(i)
\end{aligned} \tag{3}$$

$$\begin{aligned}
\beta_t(i) &= P(y_{t+1}^{1:mo}, y_{t+2}^{1:mo}, \cdots, y_T^{1:mo} \mid X_t = i) \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} \sum_{y_{t+1}^{1s},\cdots,y_{t+1}^{ms}} \prod_{k=1}^{m} P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{ks}) \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} \sum_{y_{t+1}^{2s},\cdots,y_{t+1}^{ms}} \prod_{k=1}^{m} P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{ks}) \left[\sum_{p=1}^{s_1} P(Y_{t+1}^1 = y_{t+1}^{1p} \mid X_{t+1} = j) P(Y_{t+1}^1 = y_{t+1}^{1p}) \right] \quad , \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} \sum_{y_{t+1}^{3s},\cdots,y_{t+1}^{ms}} \prod_{k=1}^{m} P(Y_{t+1}^k = y_{t+1}^{ks} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{ks}) \prod_{k=1}^{2} \left[\sum_{p=1}^{s_k} P(Y_{t+1}^k = y_{t+1}^{kp} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{kp}) \right] \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} \prod_{k=1}^{m} \left[\sum_{p=1}^{s_k} P(Y_{t+1}^k = y_{t+1}^{kp} \mid X_{t+1} = j) P(Y_{t+1}^k = y_{t+1}^{kp}) \right]
\end{aligned} \tag{4}$$

As in Equation (3) and (4), it is easy to see that after introducing a new computing method to the improved forwards-backwards algorithm, it avoids repeated computation in the forwards recursive process and the backwards recursive process. However, the computing method still has the disadvantages because these two computing methods are independent. And computation step sharing is considered on the two processes, so it needs to be optimized.

The parallel factors below can correlate these two recursive processes to realize the computation step sharing, which can improve the inference efficiency of the algorithm further.

Parallel factors

Both forwards-backwards algorithm and the improved forwards-backwards algorithm need two inferences in different directions, that is, the information propagates forwards and backwards. The information forwards propagation is realized by the forwards recursive formula; the information backwards propagation is realized by the backwards recursive formula. There are the same computation steps during the two computation processes. So we define the parallel factors.

Definition 1

There are the same computation factors in the recursive formulas of the information forwards and backwards propagation, the same computation factors in the two recursive formulas is called the parallel factors.

The basic idea of the parallel computation factors: In the recursive computation process of the forwards algorithm, not only the computation results of α but also the results of the parallel factors are stored in each time slice. So we needn't compute the parallel factors during the recursive computation process of the backwards algorithm, which can reduce the computation amount, and improve efficiency of the algorithm significantly.

We deduce the forwards-backwards algorithm based on the parallel factors in theory.

Algorithm description

The information forwards propagation: By Equation (3), the parallel computation factor is defined $\mu_t(j)=\prod_{k=1}^{m}\left[\sum_{p=1}^{s_k}P(Y_t^k=y_t^{kp}\mid X_t=j)P(Y_t^k=y_t^{kp})\right]$, where $t\geq 2$, so the forwards recursive formula is as follows:

$$\begin{aligned}\alpha_t(j)&=P(X_t=j\mid y_1^{1:mo},y_2^{1:mo},\cdots,y_t^{1:mo})\\&=\eta\prod_{k=1}^{m}\left[\sum_{p=1}^{s_k}P(Y_t^k=y_t^{ks}\mid X_t=j)P(Y_t^k=y_t^{ks})\right]\sum_{i=1}^{n}a_{ij}\alpha_{t-1}(i),\\&=\eta\mu_t(j)\sum_{i=1}^{n}a_{ij}\alpha_{t-1}(i)\end{aligned}\tag{5}$$

where η is a normalizing constant, $j=1,2,\cdots,n$.

The information backwards propagation: During the backwards recursive computation process, we need directly use the results of the parallel factors obtained in the forwards recursive process. From Equation (4), we have the backwards recursive formula as follows:

$$\begin{aligned}\beta_t(i)&=P(y_{t+1}^{1:mo},y_{t+2}^{1:mo},\cdots,y_T^{1:mo}\mid X_t=i)\\&=\sum_{j=1}^{n}\beta_{t+1}(j)\prod_{k=1}^{m}\left[\sum_{p=1}^{s_k}P(Y_{t+1}^k=y_{t+1}^{kp}\mid X_{t+1}=j)P(Y_{t+1}^k=y_{t+1}^{kp})\right]a_{ij},\\&=\sum_{j=1}^{n}\beta_{t+1}(j)\mu_{t+1}(j)a_{ij}\end{aligned}\tag{6}$$

where $\mu_{t+1}(j)$ is the parallel factor of the $t+1$'th time slice, $i,j=1,2,\cdots,n$.

Combining Equation (5) and Equation (6), we have the forwards-backwards algorithm based on the parallel factors as follows:

$$\begin{aligned}\gamma_t(i)&=P(X_t=i\mid y_1^{1:mo},y_2^{1:mo},\cdots,y_T^{1:mo})\\&=P(X_t=i\mid y_1^{1:mo},\cdots,y_t^{1:mo},y_{t+1}^{1:mo},\cdots,y_T^{1:mo})\\&=\eta P(X_t=i\mid y_1^{1:mo},\cdots,y_t^{1:mo})P(y_{t+1}^{1:mo},\cdots,y_T^{1:mo}\mid X^t=i,y_1^{1:mo},\cdots,y_t^{1:mo}),\\&=\eta P(X_t=i\mid y_1^{1:mo},\cdots,y_t^{1:mo})P(y_{t+1}^{1:mo},\cdots,y_T^{1:mo}\mid X_t=i)\\&=\eta\alpha_t(i)\beta_t(i)\end{aligned}\tag{7}$$

The computation steps of the forwards-backwards algorithm based on the parallel factors.

Step 1 : Initialize the networks by the conditional probabilities, the evidence information and prior probabilities;

Step 2 : Recursively compute α_t, and store the computation results of the parallel factor μ_t;

Step 3 : If $t+1 \le T$, turn to Step 2, or turn to Step 4;

Step 4 : Initialize backwards recursive process, $\beta_T(i)=1$;

Step 5 : When computing β_t recursively, we use the value of the parallel factor μ_{t+1};

Step 6 : if $t-1 \ge 1$, turn to Step 5, or turn to Step 7;

Step 7 : compute γ_t;

Step 8 : if $t+1 \le T$, turn to Step 7, or end.

Complexity analysis

Let's suppose a discrete dynamic Bayesian networks with T time slices, there is a hidden node, m observation nodes in each time slice. The maximum state number of the node is N. The complexity of the improved forwards-backwards algorithm is $O\left(mN^{m+2}T\right)$, but the complexity of the forwards-backwards algorithm based on the parallel factors is $O\left(mN^3T\right)$. By comparing the complexities of these two algorithms, we can see that when the number of the observation nodes in each time slice is greater than or equal to 2, the complexity can be reduced significantly by the algorithm based on parallel factors.

EXPERIMENTS

In this paper, experiments are programmed with C++ and run on the PC-2.2 GHz.

In order to prove performance of the algorithm, the constructed dynamic Bayesian network model is shown in Figure 1, there is a hidden node and four observation nodes on each time slice. The given state transition probabilities are shown in TABLE 1, the conditional probabilities in TABLE 2, the observation data in TABLE 3, prior probabilities are $P\left(X=1,2,3,4\right)=\left(0.05,0.33,0.4,0.22\right)$.

TABLE 1 : State transition probabilities

$X_t \backslash X_{t+1}$	x^1	x^2	x^3	x^4
x^1	0.9	0.08	0.02	0.0
x^2	0.01	0.74	0.2	0.05
x^3	0.02	0.24	0.64	0.1
x^4	0.0	0.1	0.16	0.74

TABLE 2 : Conditional probabilities

X	P(Y^1|X) y^{11}, y^{12}, y^{13}	P(Y^2|X) y^{21}, y^{22}	P(Y^3|X) y^{31}, y^{32}, y^{33}	P(Y^4|X) y^{41}, y^{42}, y^{43}
x^1	0.0,0.04,0.96	0.90,0.10	0.95,0.05,0.0	0.98,0.02,0.0
x^2	0.15,0.35,0.5	0.2,0.8	0.15,0.7,0.15	0.05,0.85,0.1
x^3	0.4,0.35,0.25	0.25,0.75	0.1,0.5,0.4	0.0,0.6,0.4
x^4	0.1,0.3,0.6	0.02,0.98	0.05,0.25,0.7	0.0,0.2,0.8

TABLE 3 : Observation data

t	Y^1	Y^2	Y^3	Y^4
1	0.0,0.27,0.73	0.12,0.88	0.15,0.85,0.0	0.0,0.75,0.25
2	0.0,0.25,0.75	0.15,0.85	0.2,0.8,0.0	0.0,0.75,0.25
3	0.0,0.21,0.79	0.12,0.88	0.15,0.85,0.0	0.0,0.8,0.2
4	0.0,0.2,0.8	0.15,0.85	0.2,0.8,0.0	0.0,0.85,0.15
5	0.0,0.17,0.83	0.22,0.78	0.25,0.75,0.0	0.05,0.85,0.1
6	0.0,0.13,0.87	0.24,0.76	0.25,0.75,0.0	0.05,0.9,0.05
7	0.0,0.12,0.88	0.21,0.79	0.2,0.8,0.0	0.05,0.95,0.0
8	0.0,0.15,0.85	0.17,0.83	0.15,0.8,0.05	0.0,0.85,0.15
9	0.0,0.21,0.79	0.12,0.88	0.1,0.85,0.05	0.0,0.8,0.2
10	0.0,0.22,0.78	0.1,0.9	0.05,0.8,0.15	0.0,0.75,0.25

According to the observation data in TABLE 3, we use the improved forwards-backwards and forwards-backwards algorithm based on parallel factors to infer, respectively, and the results are given in TABLE 4, where, each group of data from the left to the right represent the probabilities of the hidden node X belonging to its x^1, x^2, x^3, x^4 states. Figure 2 is the comparison of the running time which is obtained by using the data in TABLE 3 through the two inference algorithms.

TABLE 4 : Comparison of inference results by the two inference algorithms

t	Inference results by improved forwards-backwards	Inference results by forwards-backwards algorithm based on parallel factors
1	3.81332e-005,0.823951,0.147278,0.0287332	3.81332e-005,0.823951,0.147278,0.0287332
2	9.24871e-006,0.91385,0.0746691,0.0114712	9.24871e-006,0.91385,0.0746691,0.0114712
3	5.39919e-006,0.942055,0.0521287,0.00581081	5.39919e-006,0.942055,0.0521287,0.0058108
4	7.99375e-006,0.951848,0.044432,0.00371158	7.99375e-006,0.951848,0.044432,0.00371158
5	5.26926e-005,0.956798,0.0404067,0.0027429	5.26926e-005,0.956798,0.0404067,0.0027429
6	5.71597e-005,0.960105,0.0376505,0.00218768	5.71597e-005,0.960105,0.0376505,0.0021876
7	3.84893e-005,0.959971,0.0378281,0.00216225	3.84893e-005,0.959971,0.0378281,0.0021622
8	6.69657e-006,0.947804,0.0468639,0.00532527	6.69657e-006,0.947804,0.0468639,0.0053252
9	3.48227e-006,0.924003,0.0640118,0.0119812	3.48227e-006,0.924003,0.0640118,0.0119812
10	1.81186e-005,0.855873,0.110812,0.0332968	1.81186e-005,0.855873,0.110812,0.0332968

By comparing the data in the left and right column in TABLE 4, the inference results in each time slice are the same by the two algorithms. The validity of the algorithm based on parallel factors is proved.

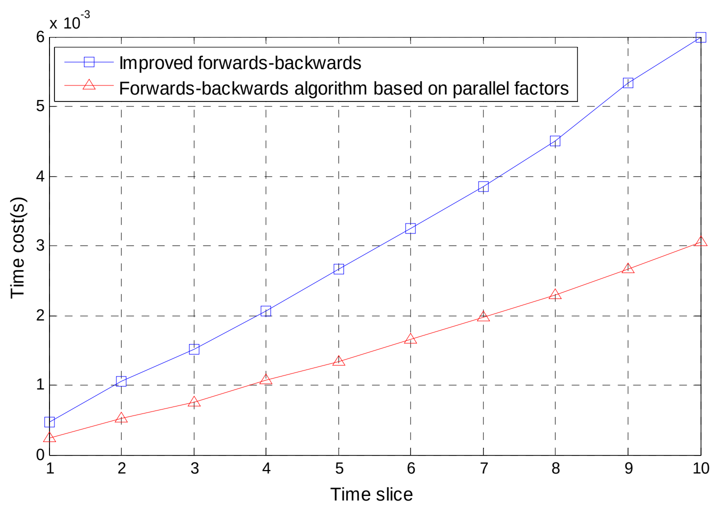

Figure 2 : Comparison of the time cost by the improved forwards-backwards algorithm and the forwards-backwards algorithm based on parallel factors

From the curves (see, Figure 2), it's easy to see that the time cost and the time slices have approximate linear relationships by these two algorithms, where, the slope of the improved forwards-backwards algorithm is bigger. As the time slices increase on dynamic Bayesian networks, the differences in the time cost between the two algorithms are more and more obvious. When the number of time slices is large, the efficiency of the algorithm based on parallel factors is shown completely.

CONCLUSION

The research about the high efficiency inference algorithm on dynamic Bayesian networks is a challenge in the area of Bayesian networks. In this paper, according to the feature of the information propagation on dynamic Bayesian networks, the parallel factors are introduced to realize the computation step sharing, which avoids extra computation and improves the computation efficiency. The algorithm is suitable for both soft evidences and hard evidences. The high efficiency of the inference can be shown significantly, which provides a new choice for generalized hidden Markov model inference.

ACKNOWLEDGEMENTS

This research is supported by Scientific Research Program from Education Department of Shanxi Provincial Government (2013JK1110).

REFERENCES

[1] J.Pearl; Fusion, propagation, and structuring in belief networks, Artificial Intelligence, **29(3)**, 241-288 **(1986)**.
[2] Y.Hu, X.Z.Zhang, E.Ngai, R.C.Cai, M.Liu; Software project risk analysis using Bayesian Networks with causality constraints, Decision Support Systems, **56**, 439-449 **(2013)**.
[3] X.Wang, Y.Qiu; Software Dependability Evaluation based on Linguistic Bayesian Network and OWA Operator, Journal of Computational Information Systems, **9(7)**, 2889-2896 **(2013)**.
[4] H.Gan, Q.Yong, X.Mu; Intelligent Monitoring and Control System for Ship Engine Room Based on Bayesian Network, Journal of Computational Information Systems, **10(10)**, 4109-4117 **(2014)**.
[5] Z.Li, J.Yang, S.Tan; Systematically Discovering Dependence Structure of Global Stock Markets using Dynamic Bayesian Network, Journal of Computational Information Systems, **9(18)**, 7215-7226 **(2013)**.
[6] R.D.Shachter, S.K.Anderson, P.Szolovits; Global conditioning for probabilistic inference in belief networks, Paper presented at the Tenth International Conference on Uncertainty in artificial intelligence, Morgan Kaufmann, San Francisco, 514-522 **(1994)**.
[7] S.L.Lauritzen, D.J.Spiegelhalter; Local computations with probabilities on graphical structures and their applications to expert systems, Journal of the Royal Statistical Society Series B, **50**, 157-224 **(1988)**.
[8] R.D.Shachter, B.D'Ambrosio, B.A.D.Favero; Symbolic probabilistic inference in belief networks, Paper presented at the Eighth National Conference on Artificial Intelligence, MIT, Boston, 126-131 (**1990)**.
[9] S.Ross; Evidence absorption and propagation through evidence reversals, Paper presented at the Fifth Annual Conference on Uncertainty in Artificial Intelligence, Amsterdam, North-Holland, 173-190 **(1990)**.
[10] P.Dagum, R.Karp, M.Luby, S.Ross; An optimal algorithm for Monte Carlo estimation, Paper presented at the 36th Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos, 142-149 **(1995)**.
[11] T.Dean, K.Kanazawa; A model for reasoning about persistence and causation, Computational Intelligence, **5**, 142-150 **(1989)**.
[12] T.L.Dean, K.Kanazawa; Probabilistic Temporal Reasoning, Paper presented at the AAAI-88 Proceedings, AAAI Press, Menlo Park, California, 524-529 **(1988)**.
[13] S.Rusell, P.Norving; A Modern Approach, Artificial Intelligence, 559-580 **(2003)**.
[14] J.Bilmes, G.Zweig; The graphical models toolkit: An open source software system for speech and time-series processing, Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Institute of Electrical and Electronics Engineers, Orlando, Florida, 3916-3919 **(2002)**.
[15] K.P.Murphy; Dynamic Bayesian Networks: Representation, Inference and Learning. PhD dissertation, University of California, USA, **(2002)**.
[16] H.Y.Chen, X.G.Gao; Ship recognition based on improved forwards-backwards algorithm, Paper presented at the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, Los Alamitos, 509-513 **(2009)**.