Novel recursive inference algorithm for discrete dynamic Bayesian networks

Huange Wang a,b,*, Xiaoguang Gao a, Chris P. Thompson b

a School of Electronics and Information, Northwestern Polytechnical University, Xi’an 710072, China
b School of Engineering, Cranfield University, Bedford MK43 0AL, UK

Received 14 August 2008; received in revised form 7 October 2008; accepted 16 October 2008

Abstract

Dynamic Bayesian networks (DBNs) can effectively perform modeling and qualitative reasoning for many dynamic systems. However, most of its inference algorithms involve complicated graphical transformations that are hard to program and time-consuming to compute. This article proposes a new recursive inference algorithm, which is a purely numerical method derived from probability theory and the characteristics of Bayesian networks to do both on-line and off-line inferences in discrete DBNs. The most prominent advantages of this novel approach include: (1) it is an exact inference algorithm, thus its accuracy and stability can be guaranteed, (2) it avoids the complex graphical transformation so as to remarkably improve the inference speed, and (3) its concise recursive formulae facilitate programming of both forwards and backwards pass. All of these issues are verified by accurate mathematical derivation as well as a couple of application examples with comparison between the new algorithm and the two most prevailing inference approaches of discrete DBNs – the interface algorithm and the forwards-backwards algorithm.

1. Introduction

Time-series data modeling is quite significant to many theoretical and engineering problems. It is known that state-space models are much better than the classical time-series modeling approaches in many respects [1,2]. Moreover, compared with the two most common state-space models, i.e., the Hidden Markov model and the Kalman filter model, a more general state-space model called dynamic Bayesian network (DBN) is much more powerful in representation [3]. Thus, since the mid-1990s DBNs have been widely accepted for dynamic systems modeling and reasoning, especially the study of discrete DBNs has become a topic of international focus [3–5].

However, most of the literature concerning DBNs is devoted to the applied research, such as figure tracking [6], speech recognition [7], bio-sequence analysis [8,9], and so on, while few are related to the theoretical study. Moreover, most of the existing inference algorithms of DBNs are based on complicated graphical transformation [10–13], which is hard to program. In addition, inference efficiency is another crucial issue, especially for the fast evolving dynamic domains, e.g., the BAT mobile where the inference computation is under time pressure. Refs. [14,15] propose two approaches that could efficiently perform the reasoning, but Ref. [14] cannot obtain the exact posterior probability of a target state, while Ref. [15] is still based on the graphical transformation.
This article focuses on the study of the inference algorithm and its novel contribution lies in presenting a purely numerical method, which is based on the conditional probability equation, the conditional independence assumption and recursive computation, to implement probabilistic reasoning in a certain kind of discrete DBN models. Compared with the most commonly used inference approaches – the interface algorithm of DBNs [3,10] and the forwards–backwards (FB) algorithm of HMMs [3], the new recursive algorithm could perform both on-line and off-line inferences, but avoid the complex graphical transformation so as to have a much faster computing speed. Its correctness and advantages have been verified by a couple of examples.

2. The direct inference algorithm of discrete DBNs

For a discrete DBN with \( T \) time-slices, \( n \) hidden nodes and \( m \) observational nodes, the joint probability of hidden nodes under the observational condition is

\[
P(X^1, X^2, \ldots, X^n_1, \ldots, X^n_T, X^2_T, \ldots, X^m_T | y^1_1, y^2_1, \ldots, y^n_1, \ldots, y^2_1, y^2_1, \ldots, y^m_1)
\]

where \( X^i_t \) and \( y^i_t \) respectively are the \( i \)th hidden variable and \( j \)th observational datum on time-slice \( t \).

Any inference of Bayesian networks (BNs) is based on the conditional probability equation and the conditional independence assumption [16]:

\[
\text{Conditional probability equation } P(X|Y) = \frac{P(XY)}{P(Y)} = \frac{P(XY)}{\sum_X P(X|Y)} \quad (1)
\]

\[
\text{Conditional independence assumption } P(X) = \prod_i P(X^i|Pa(X^i)) \quad (2)
\]

where, \( X = \{X^1, \ldots, X^n\} \) is the set composed of all the variables in a BN, \( P(X) \) the joint probability of set \( X \), \( Pa(X) \) the parent nodes of \( X^i \), and \( P(X^i|Pa(X^i)) \) the local conditional probability.

Since discrete DBNs also satisfy the conditional independence assumption, i.e.,

\[
P(X^1_1, \ldots, X^n_1, \ldots, X^n_t, \ldots, X^n_T, Y^1_1, \ldots, Y^m_1) = \prod_i P(X^i_1|Pa(X^i)) \prod_j P(Y^j_1|Pa(Y^j))
\]

\[
t \in [1, T], \ i \in [1, n], \ j \in [1, m] \quad (3)
\]

Then from Eqs. (1) and (3), we could obtain

\[
P(X^1_1, \ldots, X^n_1, \ldots, X^n_T, \ldots, X^n_T) = \prod_i P(X^i_1|Pa(X^i)) \prod_j P(Y^j_1|Pa(Y^j))
\]

\[
= \sum_{X^1_1, \ldots, X^n_1, \ldots, X^n_T} \prod_i P(X^i_1|Pa(X^i)) \prod_j P(Y^j_1|Pa(Y^j)) \quad (4)
\]

On-line inference (also called filtering) of discrete DBNs is to compute the conditional probabilities of a current hidden variable when given all the observational data up to the present time. i.e., \( P(X^i_t|y^1_{1:t}) \) for any \( 1 \leq t \leq n \). Off-line inference (also called fixed-interval smoothing) is to compute the conditional probabilities of a past hidden state when given the observations over all the time-slices. i.e., \( P(X^i_t|y^1_{1:T}) \) for any \( 1 \leq i \leq n \) and \( 1 \leq t \leq T \).

3. The recursive inference algorithm of exclusive top-dependent nets

We define a certain kind of DBNs, in which only the hidden variable at the top of the net has the conditional dependence between each pair of neighboring time-slices, like the exclusive top-dependent nets (ETDNs). For any time-slice \( t \), the top hidden node is denoted by \( Z_t \), other \( n \) hidden and \( m \) observational variables are respectively \( X^1_t, \ldots, X^n_t \) and \( Y^1_t, \ldots, Y^m_t \). In the following arguments, we will restrict ourselves to the inference of ETDNs.

3.1. The forwards pass

After obtaining the observational data of the first time-slice, in terms of Eq. (4), we have

\[
P(Z_1; X^1_1, \ldots, X^n_1, y^1_1, \ldots, y^m_1) = \frac{\prod_{t=1}^n P(y^1_t|Pa(y^1_t)) \prod_{i=1}^n P(X^i_1|Pa(X^i))P(Z_1)}{\sum_{x^1_1, \ldots, x^n_1} \prod_{t=1}^n P(y^1_t|Pa(y^1_t)) \prod_{i=1}^n P(X^i_1|Pa(X^i))P(Z_1)} \quad (5)
\]

\[
P(Z_1; y^1_1, \ldots, y^m_1) = \sum_{x^1_1, \ldots, x^n_1} P(Z_1; X^1_1, \ldots, X^n_1, y^1_1, \ldots, y^m_1) \quad (6)
\]

In order to keep the coherence of all the expressions, the conditional probability of the second time-slice is given as below:

\[
P(Z_2; X^1_2, \ldots, X^n_2, y^1_2, \ldots, y^m_2) = \frac{\prod_{t=1}^2 P(y^1_t|Pa(y^1_t)) \prod_{i=1}^n P(X^i_2|Pa(X^i))P(Z_2|Z_1)}{\sum_{x^1_2, \ldots, x^n_2} \prod_{t=1}^2 P(y^1_t|Pa(y^1_t)) \prod_{i=1}^n P(X^i_2|Pa(X^i))P(Z_2|Z_1)} \quad (7)
\]
Here, we introduce the prior probability of $Z_t$

$$P(Z_t|y_1^m, \ldots, y_m^m) = \sum_{z_t} P(Z_t, Z_1|y_1^1, \ldots, y_m^m)$$

$$= \sum_{z_t} P(Z_t|Z_1)P(Z_1|y_1^1, \ldots, y_m^m)$$

(8)

Then a slight modification can be made to the right side of Eq. (7):

$$P(Z_t, X_1^T, \ldots, X_m^T|y_1^1, \ldots, y_m^m)$$

$$= \prod_j P(y_j|P(a_j)) \prod_j P(X_j|P(a_j)) P(Z_t|y_1^1, \ldots, y_m^m)$$

$$= \sum_{z_t, x_1^T, \ldots, x_m^T} \prod_j P(y_j|P(a_j)) \prod_j P(X_j|P(a_j)) P(Z_t|y_1^1, \ldots, y_m^m)$$

(9)

In Eq. (9) $P(Z_t|Z_1)$ is replaced by the prior probability of variable $Z_t, P(Z_t|y_1^1, \ldots, y_m^m)$. This alteration could not only eliminate the underlying impact imposed by the previous time-slice, i.e., $Z_1$, but also reserve the observational information coming from the previous time-slice, i.e., $y_1^1, \ldots, y_m^m$. So, in fact, the left side of Eq. (9) should be $P(Z_t, X_1^T, \ldots, X_m^T|y_1^1, \ldots, y_m^m)$. Now the general recursive computation formulae of the forwards pass (on-line inference) can be inferred as follows:

$$P(Z_t, X_1^t, \ldots, X_m^t|y_1^1, \ldots, y_m^t)$$

$$= \prod_j P(y_j|P(a_j)) \prod_j P(X_j|P(a_j)) P(Z_t|y_1^1, \ldots, y_m^1)$$

$$= \sum_{z_t, x_1^t, \ldots, x_m^t} \prod_j P(y_j|P(a_j)) \prod_j P(X_j|P(a_j)) P(Z_t|y_1^1, \ldots, y_m^1)$$

(10)

where $P(Z_t|y_1^1, \ldots, y_m^m) = \sum_{z_t} P(Z_t, Z_1|y_1^1, \ldots, y_m^m) = \sum_{x_1^T, \ldots, x_m^T} P(Z_t, X_1^t, \ldots, X_m^t|y_1^1, \ldots, y_m^m) = \sum_{x_1^T, \ldots, x_m^T} P(Z_t, X_1^t, \ldots, X_m^t|y_1^1, \ldots, y_m^m)$.

3.2. The backwards pass

The aim of off-line inference is computing $P(Z_{T-1}, X_{T-1}^1, \ldots, X_{T-1}^m|y_1^1, \ldots, y_m^m)$, where $1 \leq t \leq T$. When $t = T - 1$,

$$P(Z_{T-1}, X_{T-1}^1, \ldots, X_{T-1}^m|y_1^1, \ldots, y_m^m) = \frac{1}{P(y_1^1, \ldots, y_T^1)}$$

$$= \sum_{z_{T-1}, x_1^{T-1}, \ldots, x_m^{T-1}} P(Z_{T-1}, X_{T-1}^1, \ldots, X_{T-1}^m, Z_{T-1}|y_1^1, \ldots, y_m^m)$$

(11)

If we denote

$$P(X_1^1, \ldots, X_m^1, y_1^1, \ldots, y_m^1, Z_{T-1})$$

$$= \prod_j P(X_j^1|P(a_j)) \prod_j P(y_j^1|P(a_j)) P(Z_{T-1}|y_1^1, \ldots, y_m^1)$$

$$= \sum_{z_{T-1}, x_1^{T-1}, x_m^{T-1}} P(X_1^1, \ldots, X_m^1, y_1^1, \ldots, y_m^1, Z_{T-1}|y_1^1, \ldots, y_m^1)$$

(12)

Then Eq. (11) can be rewritten as

$$P(Z_{T-1}, X_{T-1}^1, \ldots, X_{T-1}^m|y_1^1, \ldots, y_m^m)$$

$$= \frac{1}{P(y_1^1, \ldots, y_T^1)}$$

$$= \sum_{z_{T-1}, x_1^{T-1}, \ldots, x_m^{T-1}} P(X_1^1, \ldots, X_m^1, y_1^1, \ldots, y_m^1, Z_{T-1}|y_1^1, \ldots, y_m^1)$$

(14)

Correspondingly, we have

$$P(y_1^1, \ldots, y_m^1, y_1^1, \ldots, y_m^1|Z_{T-2})$$

$$= \sum_{Z_{T-2}} P(y_1^1, \ldots, y_m^1|Z_{T-2})$$

(15)

$$P(Z_{T-2}, X_{T-2}^1, \ldots, X_{T-2}^m|y_1^1, \ldots, y_m^1)$$

$$= \frac{1}{P(y_1^1, \ldots, y_T^1)}$$

$$= \sum_{Z_{T-2}} P(y_1^1, \ldots, y_m^1|Z_{T-2})$$

(16)

Finally, the general recursive computation formulae of the backwards pass can be concluded as

$$P(y_1^1, \ldots, y_m^1, y_1^1, \ldots, y_m^1|Z_t)$$

$$= \frac{1}{P(y_1^1, \ldots, y_T^1)}$$

$$= \sum_{Z_{t+1}} P(y_1^1, \ldots, y_m^1, y_1^1, \ldots, y_m^1|Z_{t+1})$$

(17)

where

$$P(y_1^1, \ldots, y_m^1, y_1^1, \ldots, y_m^1|Z_t)$$

$$= \sum_{Z_{t+1}} P(y_1^1, \ldots, y_m^1, y_1^1, \ldots, y_m^1|Z_{t+1})$$

$$= \sum_{Z_{t+1}} P(X_1^1, \ldots, X_m^1, y_1^1, \ldots, y_m^1, Z_{t+1}|y_1^1, \ldots, y_m^1, Z_{t+1}|Z_t)$$

4. Experimental results and analysis

When all the hidden nodes of a DBN are discrete, two representative approaches are usually adopted to perform the inference. One is the FB algorithm with firstly converting the DBN to HMM; the other is unrolling the DBN to a static network and then using the junction tree algorithm (also called clique tree propagation), which is developed on the basis of variable elimination. However, at the present time, the interface algorithm tends to replace the junction tree algorithm, because it proposes the concept of forward interface/backward interface so as to construct junction trees based on 1-slice DBN, which makes it possible to directly apply the junction algorithm to the DBN without the unrolling process.

Fig. 1 shows two simple but typical ETDNs. We apply the recursive algorithm, the interface algorithm as well as the FB algorithm of HMMs to both of them separately, and then compare the results to validate the correctness of the novel algorithm and find its advantages.

Supposing all the variables in Fig. 1(a) are binary, the relevant parameters have been set as indicated in Table 1.
Now, we try to compute $P(Z_2 = 2|Y_1 = 1, Y_2 = 1)$ for on-line inference and $P(Z_2 = 1|Y_1 = 1, Y_2 = 1, Y_3 = 2)$ for off-line inference. We run the Matlab programs of the FB algorithm, the interface algorithm and the recursive algorithm on the same computer, which is with Intel(R) Core(TM)2 CPU T5600 1.83 GHz and 0.99 GB memory. Their inference results and corresponding time consumption are shown in Table 2.

Similarly, all the variables in Fig. 1(b) are assumed to be binary, and the parameters are given in Table 3.

Here, the on-line inference task is to compute $P(X_3 = 2|Y_1 = 2, Y_2 = 1, Y_3 = 2)$, and the off-line inference task is to compute $P(X_3 = 1|Y_1 = 1, Y_2 = 2, Y_3 = 2)$. After running the Matlab programs of the FB algorithm, the interface algorithm and the recursive algorithm on the same computer mentioned above, we put all the results in Table 4.

As the FB algorithm, the interface algorithm and the novel recursive algorithm all belong to the exact inference algorithm of DBNs, the accurate probabilities of target events computed by any of them should be exactly the same. This has been fully verified by the results of examples above. Furthermore, we can also find that the inference speed of the recursive algorithm is much faster than the one of the FB algorithm or the interface algorithm. The main reasons for this phenomenon are that the FB algorithm needs extra time to do the conversion from DBN to HMM, while the interface algorithm has to carry out the complicated graphical transformation as its first step, i.e., constructing a junction tree for each 1-slice DBN. It is worth noting that the FB algorithm is faster than the interface algorithm for models with small scale, such as the examples we adopt here, but exponentially slower for networks with many variables (>6 binary hidden nodes).

### Table 2

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Results</th>
<th>Elapsed time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(Z_2 = 2</td>
<td>Y_1 = 1, Y_2 = 1)$ (on-line inference)</td>
<td>FB algorithm of HMM</td>
</tr>
<tr>
<td></td>
<td>Interface algorithm</td>
<td>0.04919786096257 2.0160</td>
</tr>
<tr>
<td></td>
<td>Recursive algorithm</td>
<td>0.04919786096257 0.0310</td>
</tr>
<tr>
<td>$P(Z_2 = 1</td>
<td>Y_1 = 1, Y_2 = 1, Y_3 = 1)$ (off-line inference)</td>
<td>FB algorithm of HMM</td>
</tr>
<tr>
<td></td>
<td>Interface algorithm</td>
<td>0.90018805149718 2.4380</td>
</tr>
<tr>
<td></td>
<td>Recursive algorithm</td>
<td>0.90018805149718 0.1560</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Results</th>
<th>Elapsed time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(Z_1 = 1)$</td>
<td>FB algorithm of HMM</td>
<td>0.23261264109752 1.7500</td>
</tr>
<tr>
<td></td>
<td>Interface algorithm</td>
<td>0.23261264109752 2.0160</td>
</tr>
<tr>
<td></td>
<td>Recursive algorithm</td>
<td>0.23261264109752 0.1570</td>
</tr>
<tr>
<td>$P(X_1 = 1</td>
<td>Y_1 = 1, Y_2 = 2)$ (on-line inference)</td>
<td>FB algorithm of HMM</td>
</tr>
<tr>
<td></td>
<td>Interface algorithm</td>
<td>0.66594382234776 2.0780</td>
</tr>
<tr>
<td></td>
<td>Recursive algorithm</td>
<td>0.66594382234776 0.1720</td>
</tr>
</tbody>
</table>

### Table 4

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Results</th>
<th>Elapsed time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(X_3 = 2</td>
<td>Y_1 = 2, Y_2 = 1, Y_3 = 2)$ (on-line inference)</td>
<td>FB algorithm of HMM</td>
</tr>
<tr>
<td></td>
<td>Interface algorithm</td>
<td>0.04919786096257 2.4380</td>
</tr>
<tr>
<td></td>
<td>Recursive algorithm</td>
<td>0.04919786096257 0.1560</td>
</tr>
</tbody>
</table>

### 5. Conclusion

This article proposes a novel recursive inference algorithm for a certain kind of discrete DBNs which is called ETDNs, and then applies it as well as another two exact inference algorithms of DBNs, namely the FB algorithm and the interface algorithm, to a couple of models. Experimental results computed by the three different approaches are exactly the same, which fully verifies the accuracy and stability of the new algorithm. Furthermore, as a result of avoiding the complex process of graphical transformation, i.e., converting a DBN into an HMM or a junction tree, the recursive algorithm performs much faster than the other two methods in both on-line and off-line inferences. Thus, it can be concluded that the novel recursive algorithm is more applicable for rapid inference of many practical problems.

### Acknowledgement

This work was supported by National Natural Science Foundation of China (Grant Nos. 90205019 and 6077406).

### References


