one of the other parts in the assembly, in which only one sequence is possible.

VI. Conclusion

Virtual assembly is a pilot project of a much bigger vision on "manufacturing in the computer." The interactive approach presented in this paper creates a way of introducing human expertise into assembly planning and a mechanism for integrating robot programming with sequence planning. Virtual assembly helps to identify and resolve issues related to the construction of an integrated virtual manufacturing environment that could enhance all levels of manufacturing decision and control.

References

[1] J. Bander, "A heuristic-search algorithm for path determination with learning," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 131–134, Jan. 1998.

[2] K. Gupta, "The sequential framework for practical motion planning for manipulator arms: Algorithm and experiments," in *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, K. Gupta and A. del Pobil, Eds. New York: Wiley, 1998, pp. 9–31.

[3] K. Gupta and A. del Pobil, "Successes, failures, challenges, and future directions of robot motion planning," in *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, K. Gupta and A. del Pobil, Eds. New York: Wiley, 1998, pp. 351–354.

[4] S. Jayaram, H. Connacher, and K. Lyons, "Virtual assembly using virtual-reality techniques," *Computer Aided Design*, vol. 29, pp. 575–584, Aug. 1997.

[5] J. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[6] S. Mok, K. Ong, and C. Wu, "Automatic generation of assembly instructions using STEP," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2001, pp. 313–318.

[7] S. Mok, C. Wu, and D. Lee, "Modeling automatic assembly and disassembly operations for virtual manufacturing," *IEEE Trans. Syst., Man, Cybern. A*, vol. 31, pp. 223–232, May 2001.

[8] H. Sun, X. Yuan, G. Baciu, and Y. Gu, "Direct virtual–hand interface in robot assembly programming," *J. Vis. Lang. Comput.*, vol. 10, no. 1, pp. 55–68, 1999.

[9] R. Wilson and J. Latombe, "Geometric reasoning about mechanical assembly," *Artif. Intell.*, vol. 71, pp. 371–396, Dec. 1994.

[10] C. Wu and N. Kim, "Modeling of part-mating strategies for automating assembly operations for robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 1065–1074, July 1994.

[11] X. Yuan, "Interactive assembly planning in virtual environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nov. 2000, pp. 1462–1467.

Encoding Probability Propagation in Belief Networks

Shichao Zhang and Chengqi Zhang

***Abstract*—Complexity reduction is an important task in Bayesian networks. Recently, an approach known as the linear potential function (LPF) model has been proposed for approximating Bayesian computations. The LPF model can effectively compress a conditional probability table into a linear function. This correspondence extends the LPF model to approximate propagation in Bayesian networks. The extension focuses on encoding probability propagation as a polynomial function for a class of tractable problems.**

***Index Terms*—Approximating reasoning, Bayesian network, belief network, encoding technology, probabilistic reasoning.**

I. Introduction

Probabilistic reasoning with *Bayesian networks* (or *belief networks*) [2] is based on conditional probability matrices. A conditional probability matrix $M_{Y\,|\,X}$ is used in belief networks to describe causality of the form $X \rightarrow Y$. Let the domain of X be $R(X) = \{x_1, x_2, \ldots, x_m\}$ and the domain of Y be $R(Y) = \{y_1, y_2, \ldots, y_n\}$. $M_{Y\,|\,X}$ is defined as $M_{Y\,|\,X} = P(y\,|\,x) = P(Y = y\,|\,X = x) = [p(y_j\,|\,x_i)]_{m\times n}$, where $p(y_j\,|\,x_i) = p(Y = y_j\,|\,X = x_i)$ are conditional probabilities of $Y = y_j$, given $X = x_i, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$. Given an observation $X = a = (p(a_1), p(a_2), \ldots, p(a_m))$, we can obtain $Y = b$ of the form $(p(b_1), p(b_2), \ldots, p(b_n))$ as

$$b = aM_{Y\,|\,X}. \tag{1}$$

Equation (1) is a well-known probability propagation in Bayesian networks. Computation with Bayesian networks has been proven to be NP-hard [1], [6]. By building an encoding technique, Santos [4], [5] advocated an efficient approach, known as the *linear potential function (LPF)*, to circumvent this problem for Bayesian networks. This approach compresses a conditional probability table into a linear function. The encoding technique provides a new way to perform probability computations in Bayesian networks.

By extending Santos' approach, this correspondence presents a *polynomial approximating function (PAF)* model for approximating propagation in Bayesian networks. To demonstrate the extensibility of the LPF model, this correspondence focuses on propagation in Bayesian networks for only a class of tractable problems.

For the rule $X \rightarrow Y$, we can divide the causality between X and Y into two cases.

Case I A polynomial relationship between X and Y.

For example, for a free-falling object, the total distance X traveled is directly proportional to the square of the time Y of travel. In particular, Example 1 (shown in Section II) presents a typical linear relationship between X and Y, which constitutes the simplest form of causality.

Case II A nonpolynomial relationship between X and Y.

Manuscript received February 8, 2001; revised August 22, 2001, December 10, 2001, and January 26, 2002. This paper was recommended by Associate Editor E. Santos.

S. Zhang is with the Faculty of Information Technology, University of Technology, Sydney NSW 2007, Australia, and also with the School of Mathematics and Computing, Guangxi Normal University, Guilin, China (e-mail: zhangsc@its.uts.edu.au).

C. Zhang is with the Faculty of Information Technology, University of Technology, Sydney NSW 2007, Australia (e-mail: chengqi@it.uts.edu.au).

Digital Object Identifier 10.1109/TSMCA.2002.804784

For example, let X be *years*, and Y be *the population of China*. There is a nonpolynomial relationship between *the population of China* and *years*.

Case I certainly refers to a class of *tractable problems* in Bayesian networks. Also, there are potentially polynomial functions that can be used to approximate some nonpolynomial causal relations (Case II). These nonpolynomial causal relations refer to tractable problems in Bayesian networks, as well. In this correspondence, we construct polynomial functions to approximate the propagation for these tractable problems.

Descriptions of the concepts used in this correspondence are the same as those in [4].

Our goal is to develop techniques for

1) encoding the vector a to an integer $E_X(a)$;
2) encoding the vector b to an integer $E_Y(b)$, where $b = aM_{Y\,|\,X}$;
3) finding a polynomial function $E'_Y(b) = k_2E^2_X(a)+k_1E_X(a)+k_0$ close to (1), where $|E'_Y(b) - E_Y(b)| < \varepsilon$ and $\varepsilon\ (>0)$ is small enough;
4) finding a unique decoding function to extract b from $E'_Y(b)$.

To find the polynomial function $E'_Y(b) = k_2E^2_X(a)+k_1E_X(a)+k_0$, the key issues are how to construct the encoding and decoding functions for a pair of vectors, a and b, and to find k_2, k_1, and k_0.

This correspondence tackles the aforementioned problems. Section II demonstrates the effectiveness of the encoding system. Section III proposes the encoding and decoding techniques. Section IV constructs the PAF model. Sections V and VI contain a performance evaluation and conclusion, respectively.

II. Effectiveness of Encoding

In the quest for reducing the complexity of the networks, let us now examine existing models.

The first solution is the matrix model presented in (1), which has been proven to be NP-hard.

We now illustrate the use of this solution with an example borrowed from [2, pp. 151–153].

Example 1: In a certain trial there are three suspects, one of which has definitely committed a murder. The murder weapon, showing some fingerprints, has been found by police. Let X identify the last user of the weapon, namely, the killer. Let Y identify the last holder of the weapon, i.e., the person whose fingerprints were left on the weapon, and let Z represent the possible readings that may be obtained in a fingerprint laboratory examination.

The relationship between these three variables would normally be expressed by the chain $X \rightarrow Y \rightarrow Z$.

Let $\mathrm{suspect}_1$, $\mathrm{suspect}_2$, and $\mathrm{suspect}_3$ be the three suspects. To represent common-sense knowledge $(X \rightarrow Y)$ that the killer is normally the last person to hold the weapon, we use a 3 × 3 conditional probability matrix

$$M_{Y\,|\,X} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Now, let evidence be $x = (0.7, 0.1, 0.2)$. Then we have

$$[0.7\ 0.1\ 0.2] \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} = [0.59 \quad 0.17 \quad 0.24].$$

Therefore, $y = [0.59, 0.17, 0.24]$ is the result that we want, or 0.59 is the probability that $\mathrm{suspect}_1$ is the last holder of the weapon, 0.17 is the probability that $\mathrm{suspect}_2$ is the last holder of the weapon, and 0.24 is the probability that $\mathrm{suspect}_3$ is the last holder of the weapon.

The second solution is the LPF model [4], which uses an approximation function to capture all the values in a conditional probability table of a Bayesian network.

Consider a simple table consisting of only one random variable, say A. Suppose $R(A) = \{$red, green, yellow, blue, purple$\}$ and the table is $P(A = \text{red}) = 0.02$, $P(A = \text{green}) = 0.50$, $P(A = \text{yellow}) = 0.00$, $P(A = \text{blue}) = 0.37$, and $P(A = \text{purple}) = 0.11$.

The LPF model encodes the colors to $E_A(\text{yellow}) = 0$, $E_A(\text{red}) = 0.2$, $E_A(\text{purple}) = 1.1$, $E_A(\text{blue}) = 3.7$, and $E_A(\text{green}) = 5$. Then, the probabilities are described by points (0, 0.00), (0.2, 0.02), (1.1, 0.11), (3.7, 0.37), and (5, 0.50) which are able to approximately fit in a line. Hence, we can replace the information in the table with a simple continuous real function according to Santos' LPF model.

The third solution is our PAF proposal, which extends Santos' approach to encode propagation in belief networks for a class of tractable problems.

Consider some probabilities of X and the corresponding values of Y in Example 1 as follows.

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(y_1)$	$p(y_2)$	$p(y_3)$
0.9	0.1	0	0.73	0.17	0.1
0	0.7	0.3	0.1	0.59	0.31
0.6	0	0.4	0.52	0.1	0.38
0.1	0.1	0.8	0.17	0.17	0.66

Intuitively, there is no direct linear relationship between $a = (p(x_1), p(x_2), p(x_3))$ and $b = (p(y_1), p(y_2), p(y_3))$ in the above graph. Now we encode them using the following functions:

$$E_X(a) = p(x_1)10^2 + p(x_2)10^4 + p(x_3)10^6$$
$$E_Y(b) = p(y_1)10^2 + p(y_2)10^4 + p(y_3)10^6.$$

Then, the earlier graph can be transformed into the following form:

$E_X(a)$	$E_Y(b)$
1090	101773
307000	315910
400060	381052
801010	661717

Now we take $E_X(a)$ and $E_Y(b)$ as an ordered pair of the form $(E_X(a), E_Y(b))$. The concrete forms of (1090, 101 773), (307000, 315 910), (400060, 381 052), and (801010, 661 717) are fitted by the line

$$E_Y(b) = 0.7E_X(a) + 101\,010.$$

From Example 1, for $a = (0.7, 0.1, 0.2)$, $E_X(a) = 0.7 * 10^2 + 0.1 * 10^4 + 0.2 * 10^6 = 201\,070$. From the function $E_Y(b) = 0.7E_X(a) + 101010$, $E_Y(b) = 0.7 * 201\,070 + 101\,010 = 241\,759$. After decoding, $b = (0.59, 0.17, 0.24)$, which is exactly the same as the result from the first solution. (The encoding and decoding functions are described in Section III.)

From the above observations, probability propagation with matrices is NP-hard. LPF can effectively compress probability tables. PAF can effectively reduce the complexity of propagation.

III. Encoding and Decoding Techniques

Given a rule $X \rightarrow Y$ with matrix $M_{Y\,|\,X}$ and an observation pair $X = a$ and $Y = b$, this correspondence endeavors to construct a polynomial function of the form: $F(a) = k_nE^n_X(a) + \cdots + k_1E_X(a) + k_0$, close to (1). Without loss of generality, we suppose $n = 2$ for the purpose of this correspondence. Ideally, this function should be expected to satisfy $F(a) = E_Y(b) = E_Y(aM_{Y \mid X})$. That is

$$k_2 E_X^2(a) + k_1 E_X(a) + k_0 = E_Y(aM_{Y \mid X}). \tag{2}$$

In (2), we need to determine the constants k_2, k_1, k_0, and the mappings (encoders) E_X and E_Y. First, we construct the encoders E_X and E_Y. Then we construct the decoder. However, the techniques of determining k_2, k_1, and k_0 will be demonstrated in Section IV.

Let $R(X) = \{x_1, x_2, \ldots, x_k\}, R(Y) = \{y_1, y_2, \ldots, y_m\}$, and the state space of X be

$$S(X) = \left\{ (p_1, p_2, \ldots, p_k) \mid 1 \le i \le k, 0 \le p_i \le 1, \sum_{i=1}^{k} p_i = 1 \right\}.$$

The encoder E_X is defined as

$$E_X(p_1, p_2, \ldots, p_k) = 10^d p_1 + 10^{2d} p_2 + \cdots + 10^{kd} p_k$$

where $d > 0$ is a positive integer.[1] d is determined as $d = r + 1$, if the decimal places demanded is r in an application; otherwise $d = n + 1$ when $10^{n-1} < \mathrm{Max}\{|R(X)|, |R(Y)|\} \le 10^n$.

Theorem 1: The above encoder E_X is a one-to-one mapping.

Proof: Reduction to absurdity. Let the encoder of $(p_1, p_2, \ldots, p_k)$ be equal to the encoder of $(p_1', p_2,' \ldots, p_k')$. That is, $E_X(p_1, p_2, \ldots, p_k) = E_X(p_1', p_2,' \ldots, p_k')$, or

$$10^d p_1 + 10^{2d} p_2 + \cdots + 10^{kd} p_k = 10^d p_1' + \cdots + 10^{kd} p_k'$$

or

$$10^d (p_1 - p_1') + 10^{2d}(p_2 - p_2') + \cdots + 10^{kd}(p_k - p_k') = 0.$$

We know from the above suppositions, $p_i - p_i'$ $(i = 1, 2, \ldots, k)$ must not all be equal to 0. Assume that when $i = j_1, j_2, \ldots, j_m, p_i - p_i' \ne 0$, where $j_1 < j_2 < \cdots < j_m$. Then, the above formula can be rewritten as $10^{j_1 d}(p_{j_1} - p_{j_1}') + \cdots + 10^{j_m d}(p_{j_m} - p_{j_m}') = 0$, or

$$10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) = -10^{j_m d}\left(p_{j_m} - p_{j_m}'\right).$$

In other words

$$\begin{aligned} &\left\| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \right\| \\ &\quad = \left\| 10^{j_m d}\left(p_{j_m} - p_{j_m}'\right) \right\| \end{aligned}$$

or

$$\begin{aligned} &\Big| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d} \\ &\quad \times \left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \Big| \Big| = 10^{j_m d}\left\| p_{j_m} - p_{j_m}' \right\|. \end{aligned}$$

Because $0 \le p_i \le 1$, thus $10^{j_m d}\|p_{j_m} - p_{j_m}'\| \ge 10^{j_m d} 10^{-d} = 10^{(j_m - 1)d}$, according to the encoder. On the other hand, according to suppositions $1 \le i \le k, 0 \le p_i \le 1, \sum_{i=1}^{k} p_i = 1$ and $1 \le i \le k, 0 \le p_i' \le 1, \sum_{i=1}^{k} p_i' = 1$ we have, $\|p_{j_1} - p_{j_1}'\| + \cdots + \|p_{j_m} - p_{j_m}'\| \le 1$ and $\|p_{j_m} - p_{j_m}'\| \ne 0$. So $\|p_{j_1} - p_{j_1}'\| + \cdots + p_{j_{m-1}} - p_{j_{m-1}}'\| < 1$. Also, note the power of ten of each operand when we have

$$\begin{aligned} &\left\| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \right\| \\ &\quad \le \left\| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) \right\| + \cdots \\ &\qquad + \left\| 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \right\| \\ &\quad < 10^{j_{m-1} d}. \end{aligned}$$

Because $j_{m-1} < j_m$, so $j_{m-1} \le j_m - 1$, then the above inequation can be reduced to

$$\begin{aligned} &\left\| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \right\| \\ &\qquad < 10^{j_{m-1} d} \le 10^{(j_m - 1)d}. \end{aligned}$$

Hence

$$\begin{aligned} &\left\| 10^{j_1 d}\left(p_{j_1} - p_{j_1}'\right) + \cdots + 10^{j_{m-1} d}\left(p_{j_{m-1}} - p_{j_{m-1}}'\right) \right\| \\ &\qquad < 10^{j_m d}\left\| p_{j_m} - p_{j_m}' \right\|. \end{aligned}$$

This contradicts the previous assumptions. Accordingly, the above encoder is a one-to-one mapping. ■

We now present the decoder of $E_Y(b)$ for (2).

If we can obtain $F(a) = k_2 E_X^2(a) + k_1 E_X(a) + k_0$ as an approximation function of (1) for the propagation, we can solve $b_1, b_2, \ldots, b_m$ from $F(a)$ by decoding. That is

$$b_i = \left(\mathrm{INT}\left(F(a) \big/ 10^{(i-1)d}\right) - \mathrm{INT}(F(a)/10^{id}) * 10^d\right) \Big/ 10^d$$

$i = 1, 2, \ldots, m$, where $\mathrm{INT}(\,)$ is an integer function. For example, for $d = 2$ in Example 2 (shown in Section IV), if $F(a) = 241\,759$, then $b_1 = (241\,759 - 241\,700)/100 = 0.59, b_2 = (2417 - 2400)/100 = 0.17$, and $b_1 = (24 - 0)/100 = 0.24$. To assure the probability significance level of the results, the final results are

$$\begin{aligned} &b_1 := \mathrm{Max}\{0, 1 - (b_2 + b_3 + \cdots + b_m)\}, \\ &b_i := b_i/(b_1 + b_2 + \cdots + b_m), \quad \text{if } b_2 + \cdots + b_m > 1 \end{aligned}$$

where $i = 2, \ldots, m$. To consider computing errors, we assume the errors only impact on the last digits. For our model, suppose that only the last d digits are impacted upon. That is, only the probability of $p(y_1)$ is changed by the computing errors. Therefore, $b_1 \leftarrow \mathrm{Max}\{0, 1-(b_2 + b_3 + \cdots + b_m)\}$ when we decode.

The above decoder of $E_Y(b)$ is actually the reversion of the encoder.

IV. PAF Model for the Propagation

In this section, we will first determine parameters k_2, k_1, and k_0 of (2) and then illustrate the use of the PAF model by examples.

A. Determining Parameters of the Function

Our goal is to compress (1) into an approximation function; therefore, for all states in $S(X)$, or $a \in S(X)$, they must satisfy: $\|k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y \mid X})\| < \varepsilon$, where $\varepsilon > 0$ is small enough, or

$$\begin{aligned} &f'(k_2, k_1, k_0) \\ &\quad = \sum_{a \in S(X)} \left(k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y \mid X})\right)^2 \end{aligned}$$

[1] Note that this encoder only extends the LPF model to the propagation for a class of tractable problems. When the precision required is too large, the problem is intractable.

where the value of $f'(k_2, k_1, k_0)$ must be the *minimum*. Or, for $\Omega(A) \subset S(A)$

$$f(k_2, k_1, k_0) = \sum_{a\in\Omega(X)} \left(k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y\mid X})\right)^2$$

where the value of $f(k_2, k_1, k_0)$ must be the *minimum* and $f(k_2, k_1, k_0)$ is a restriction of $f'(k_2, k_1, k_0)$ on $\Omega(A)$. Hence, we can obtain the following theorem.

Theorem 2: The minimal solutions to the above formula for constants k_2, k_1, k_0 are

$$\begin{aligned} k_2 &= \frac{\eta_1(E_Y)\xi_4(E_X) - \eta_2(E_Y)\xi_2(E_X)}{\xi_1(E_X)\xi_4(E_X) - \xi_2(E_X)\xi_3(E_X)}, \\ k_1 &= \frac{\eta_1(E_Y)\xi_3(E_X) - \eta_2(E_Y)\xi_1(E_X)}{\xi_2(E_X)\xi_3(E_X) - \xi_1(E_X)\xi_4(E_X)}, \\ k_0 &= 1/\Im(\Omega(X)) \left(\sum_{a\in\Omega(X)} E_Y(aM_{Y\mid X}) - k_2 \sum_{a\in\Omega(X)} E_X^2(a) - k_1 \sum_{a\in\Omega(X)} E_X(a) \right). \end{aligned}$$

where

$$\begin{aligned} \xi_1(E_X) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} E_X^4(a) - \left(\sum_{a\in\Omega(X)} E_X^2(a)\right)^2 \\ \xi_2(E_X) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} E_X^3(a) - \Im(\Omega(X)) \times \sum_{a\in\Omega(X)} E_X^2(a) \\ \xi_3(E_X) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} E_X^3(a) - \sum_{a\in\Omega(X)} E_X(a) \times \sum_{a\in\Omega(X)} E_X^2(a) \\ \xi_4(E_X) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} E_X^2(a) - \Im(\Omega(X)) \times \sum_{a\in\Omega(X)} E_X(a) \\ \eta_1(E_Y) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} \left(E_Y(aM_{Y\mid X}) E_X^2(a)\right) - \sum_{a\in\Omega(X)} E_X^2(a) \sum_{a\in\Omega(X)} (E_Y(aM_{Y\mid X})) \\ \eta_2(E_Y) &= \Im(\Omega(X)) \sum_{a\in\Omega(X)} (E_Y(aM_{Y\mid X}) E_X(a)) - \sum_{a\in\Omega(X)} E_X(a) \sum_{a\in\Omega(X)} (E_Y(aM_{Y\mid X})). \end{aligned}$$

Proof: By using the principle of extreme value in mathematical analysis, we can find the *minimum* by taking the partial derivatives over $f(k_2, k_1, k_0)$ with respect to $k_2, k_1,$ and k_0. We must determine this, and then set these derivatives to 0. That is

$$\begin{cases} \frac{\partial f}{\partial k_2} = 2 \sum_{a\in\Omega(X)} \left(\left(k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y\mid X})\right) E_X^2(a)\right) = 0 \\ \frac{\partial f}{\partial k_1} = 2 \sum_{a\in\Omega(X)} \left(\left(k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y\mid X})\right) E_X(a)\right) = 0 \\ \frac{\partial f}{\partial k_0} = 2 \sum_{a\in\Omega(X)} \left(k_2 E_X^2(a) + k_1 E_X(a) + k_0 - E_Y(aM_{Y\mid X})\right) = 0 \end{cases}.$$

By solving the above equation group, we can obtain the solutions in Theorem 2. ■

Hence, we can take the above formula, $F(a) = k_2 E_X^2(a) + k_1 E_X(a) + k_0$, as an approximation function of (1) for the propagation in belief networks.

Generally, for rules of the form $X_1 \wedge X_2 \wedge \cdots \wedge X_n \rightarrow Y$, PAFs can be easily constructed in the same way as above.

For (2), when $k_2 = 0$, the causal relation is linear, which captures the simplest causality in Case I; when $k_2 \neq 0$, it can be used in the polynomial causality in Case I, and it can also be used to approximate the causality in Case II. The use of the approximation function is illustrated by the following examples.

B. Examples

To demonstrate the use of the above approximation function model, we use two examples for causality Case I and one example for causality Case II.

Example 2: Example 1 demonstrates a typical linear causality between X and Y. This means $k_2 = 0$. Suppose the encoders of X and Y are both the same as those in Section III, and $d = 2$. According to Theorem 2, k_0 and k_1 are determined by samples in $S(X)$ and $S(Y)$ as follows: $k_0 = 101010, k_1 = 0.7$. Then, the approximating function is as follows:

$$F(a) = 0.7E_X(a) + 101\,010.$$

In the above example, we directly use the encoder method from Section III. We now present a different example in which the order of the point values of variables need to be rearranged.

Example 3: For the rule in Example 2, let $d = 2$ and

$$M_{X\mid Y} = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \\ 0.8 & 0.1 & 0.1 \end{bmatrix}.$$

For this matrix, before X is encoded, the order of its point values needs to be rearranged as follows:

$$x_3, x_1, x_2.$$

We can rename the values as $z_1 = x_3, z_2 = x_1, z_3 = x_2$. Then, the state space is $S(X) = \{(p(z_1) = a_1, p(z_2) = a_2, p(z_3) = a_3) \mid a_1 + a_2 + a_3 = 1\}$, and the encoder is as $E_X(a) = E_X(a_1, a_2, a_3) = 10^d a_1 + 10^{2d} a_2 + 10^{3d} a_3$, or $E_X(a) = 10^d p(x_3) + 10^{2d} p(x_1) + 10^{3d} p(x_2)$, and $E_Y(b) = 10^d p(y_3) + 10^{2d} p(y_1) + 10^{3d} p(y_2)$.

Now we can solve the approximation function with the above encoder as follows:

$$F(a) = 0.7E_X(a) + 101\,010.$$

Given an observation $a = (p(x_1) = 0.2, p(x_2) = 0.1, p(x_3) = 0.7)$, the probabilities of the point values of Y_P can be gained by using the propagation for Bayesian networks

$$p(y_1) = 0.59, \quad p(y_2) = 0.24, \quad p(y_3) = 0.17.$$

The corresponding state of this observation is (0.7, 0.2, 0.1) and the encoder of the state is $E_X(a) = 107\,020$. If it is substituted into the above approximation function, we have $F(a) = 175\,924$. According to the decoding method in Section III, we can obtain the probabilities of Y_Z from $F(a)$ as follows:

$$p(y_1) = 0.59, \quad p(y_2) = 0.24, \quad p(y_3) = 0.17.$$

Examples 2 and 3 have shown that the causality in Case I can be perfectly fitted by the above function $F(a)$. Now we can demonstrate the case when $k_2 \neq 0$.

Example 4: Let the conditional probability matrix of a node be

$$M_{Y\,|\,X} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}.$$

Suppose the encoders of X and Y are both the same as those in Section III, and $d = 2$. According to the PAF model, k_0, k_1. and k_2 are determined by samples in $S(X)$ and $S(Y)$ as follows: $k_0 = 302\,693.63, k_1 = 0.101\,286\,29, k_2 = -1.267\,17 \times 10^{-10}$. Then, the approximating function is as follows:

$$F(a) = -1.267\,17 \times 10^{-10} E_X^2(a) + 0.101\,286\,29 E_X(a) + 302\,693.63.$$

Given an observation $a = (p(x_1) = 0.1, p(x_2) = 0, p(x_3) = 0.9)$, the probabilities of the point values of Y_P can be gained by the propagation for Bayesian networks as follows: $p(y_1) = 0.23, p(y_2) = 0.38, p(y_3) = 0.39$. The encoder of this observation is $E_X(a) = 900\,010$. If it is substituted into the above approximation function, we have $F(a) = 393\,749.66$. According to the decoding method in Section III, we can obtain the probabilities of the point values of Y_Z from $F(a)$ as follows:

$$p(y_1) = 0.49, \quad p(y_2) = 0.37, \quad p(y_3) = 0.39.$$

In order to assure the probability significance level of the results, the final results are

$$p(y_1) = 0.24, \quad p(y_2) = 0.37, \quad p(y_3) = 0.39.$$

Thus, we have $\|P(Y_Z) - P(Y_P)\| = \|p_Z(y_1) - p_P(y_1)\| + \cdots + \|p_Z(y_n) - p_P(y_n)\| = 0.02$.

Again, we can construct an approximating function for $d = 3$ with the same samples as follows:

$$k_0 = 300246064, \quad k_1 = 0.100\,216\,9, \\ k_2 = -6.699\,27 \times 10^{-14}.$$

Then, the approximating function is as follows:

$$F(a) = -6.699\,27 \times 10^{-14} E_X^2(a) + 0.100\,216\,9 E_X(a) + 300\,246\,064.$$

Now, given an observed value $p(x_1) = 0.1, p(x_2) = 0, p(x_3) = 0.9$, the probabilities of the point values of Y_P can be gained in Pearl's plausible inference model as follows: $p(y_1) = 0.23, p(y_2) = 0.38, p(y_3) = 0.39$. The encoder of this observed value is $E_X(a) = 900\,000\,100$. If it is substituted into the above approximation function, we have $F(a) = 390\,386\,000$. According to the given formulae numbers, we can obtain the probabilities of the point values of Y_Z from $F(a)$ as follows:

$$p(y_1) = 0.00, \quad p(y_2) = 0.386, \quad p(y_3) = 0.39.$$

In order to assure the probability significance level of the results, the final results are

$$p(y_1) = 0.224, \quad p(y_2) = 0.386, \quad p(y_3) = 0.39.$$

Thus, we have $\|P(Y_Z) - P(Y_P)\| = 0.012$.

These results show $\|P(Y_Z) - P(Y_P)\|$ is decreased from 0.02 to 0.012 for the observation $(p(x_1) = 0.1, p(x_2) = 0, p(x_3) = 0.9)$. However, the influencing range of the computational error is not reduced as d is enlarged.

V. Performance Evaluation

To study effectiveness and efficiency, we have performed several experiments for the proposed approach. Our algorithms are implemented on DELL-Pentium III machines, using Java++.

Using our PAF model, the storage space of a rule $X \rightarrow Y$ can be reduced from $O(|R(X)| * |R(Y)|)$ to $O(|R(X)| + |R(Y)|)$ and its running time can be decreased from $O(|R(X)| * |R(Y)|)$ to $O(|R(X)| + |R(Y)|)$. Our experiments in this section focus on the effectiveness and efficiency of the PAF model, compared with the matrix model [2].

A. Error

In our experiments, 20 random matrices for $|R(X)| = 2$ and $|R(Y)| = 3$, and 20 random matrices for $|R(X)| = 3$ and $|R(Y)| = 3$ are selected. For each matrix, given 50 random samples of $S(X)$, 50 states of $S(Y)$ are first generated in (1). Then, the approximating function of this matrix can be constructed by using the proposed method from the 50 groups of data. Let $Y_P = (p(y_1), p(y_2), \ldots, p(y_n))$ be the result by using (1), and $Y_Z = (p(y_1'), p(y_2'), \ldots, p(y_n'))$ be the result of using the approximating function constructed. For the sake of simplicity, let $d = 2$ in our experiments. In practical applications, $d = 4$ is more appropriate in relation to the computing complexity and the accuracy of results. For each approximating function constructed, we use 100 random samples in $\Omega(X)$ $(\Omega(X) \subset S(X))$ to check the effectiveness of the PAF model by $\|Y_Z - Y_P\|$, where $\|Y_Z - Y_P\| = \|p(y_1') - p(y_1)\| + \|p(y_2') - p(y_2)\| + \cdots + \|p(y_n') - p(y_n)\|$.

Our methods for checking the effectiveness of the PAF model by $\|Y_Z - Y_P\|$ in the experiments are as follows:

1) Maximum error (ME):

$$\underset{a \in \Omega(X)}{\mathrm{Max}} \{\|Y_Z - Y_P\|\}$$

2) Average error (AE):

$$\frac{1}{100M} \sum_{a \in \Omega(X)} \{\|Y_Z - Y_P\|\}$$

where M is the number of matrices checked, and 100 is the number of each matrix checked.

TABLE I
EFFECTIVENESS OF APPROXIMATING FUNCTIONS

Proc	Ftype	$\|R(X)\|$	$\|R(Y)\|$	No1	No2	ME	AE
$k_2 = 0$	LF	2	3	20	100	0.00	0.00
$k_2 = 0$	LF	3	3	5	100	0.00	0.00
$k_2 \neq 0$	PF	3	3	7	100	0.04	0.023
$k_2 \neq 0$	NPF	3	3	8	100	0.07	0.058

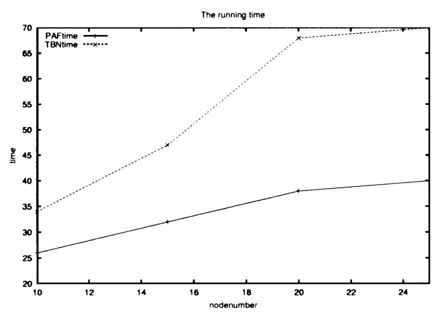

Fig. 1. Comparison on time.

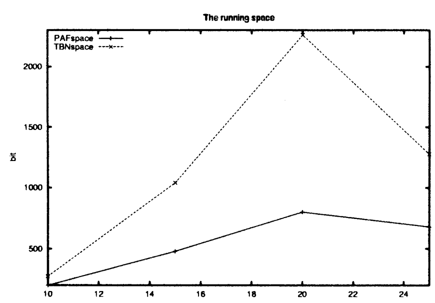

Fig. 2. Comparison on space.

The effectiveness of the PAF model by $||Y_Z - Y_P||$ in the experiments is listed in Table I.

In Table I, "Proc" is a procedure for either $k_2 = 0$ or $k_2 \neq = 0$, "Ftype" is the type of function fitted, "LF" stands for linear function, "PF" stands for polynomial function, "NPF" indicates nonpolynomial function, "No1" is the number of matrices, "No2" is the checking numbers, "ME" is the maximum error, and "AE" is the average error.

B. Efficiency of the PAF Model

For comparison, we select four groups of data from real-world investment applications. For propagating probabilities, data in each group forms a belief network (a poly-tree). The main properties of the data sets are as follows. The first group consists of ten objects (nodes), in which the average range of objects is 5 and the biggest range among the objects is 9. The second group consists of 15 objects, in which the average range of objects is 8 and the biggest range among the objects is 16. The third group consists of 20 objects, in which the average range of objects is 10 and the biggest range among the objects is 24. The fourth group consists of 25 objects, in which the average range of objects is 7 and the biggest range among the objects is 14. Comparison of our PAF model with traditional Bayesian networks (TBNs) [2] on running time and space are illustrated in Figs. 1 and 2.

C. Analysis

It can be seen that the model based on encoding functions provides good approximations of probability propagation in Bayesian networks. The functions constructed for these matrices are without error. However, the matrices in Case II also become simple by optimizing, and the errors are certainly afforded in applications.

This correspondence has focused on a class of tractable problems in Bayesian networks. Most of the matrices in the above four groups for checking time and space were selected to match Case I of causality. Only two to four matrices in each group are in Case II, and the functions for these matrices contain low errors, not over 0.01. Therefore, the propagation errors in the poly-trees are apparently acceptable for applications.

Because TBNs depend on operations on matrices to propagate probabilities, the computation is nonlinear. The proposed approach propagates probabilities by approximating polynomial functions. Therefore, running time and space for the proposed approach are less than those in TBN models.

VI. CONCLUSION

The LPF model has been proposed for compressing a conditional probability table in Bayesian networks into a linear function [4]. In this correspondence, we have extended the model to encode propagation in Bayesian networks for a class of tractable problems. We have developed techniques for 1) encoding the state a of a variable X to an integer $E_X(a)$ (shown in Section III); 2) finding a polynomial function $F(a) = k_2 E_X^2(a) + k_1 E_X(a) + k_0$ close to propagation (shown in Section IV); and 3) finding a unique decoding function to extract b from $E'_Y(b)$ (shown in Section III). The experiments have shown that our PAF model can approximately fit propagation in Bayesian networks.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor, M. Santos, and the anonymous reviewers for their detailed constructive comments on the first version of this correspondence. They would also like to thank G. Webb for his comments and help.

REFERENCES

[1] G. Cooper, "Probabilistic inference using belief networks is NP-hard," Stanford Univ., Stanford, CA, Rep. TR KSL-87-27, 1987.
[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
[3] E. Santos, Jr. and S. Shimony, "Belief updating by enumerating high-probability independence-based assignments," in *Proc. UAI*, 1994, pp. 506–513.
[4] E. Santos, Jr., "On linear potential functions for approximating Bayesian computations," *J. ACM*, vol. 43, pp. 399–430, 1996.
[5] ——, "On multiple spline approximations for Bayesian computations," *Ann. Math. Artif. Intell.*, vol. 20, no. 1–4, pp. 267–300, 1997.
[6] S. Shimony, "The role of relevance in explanation—Part I: Irrelevance as statistical independence," *Int. J. Approx. Reason.*, vol. 6, pp. 281–324, 1993.