Cyclic Bayesian Network for Software Project Iterative Process

Guoping Jiang and Mingkui Ren

1,2Department of Management Engineering and Equipment Economy, Naval Engineering University, Wuhan Hubei, China, 430033

*Corresponding author

Abstract— Aim at the Cyclic Bayesian Network, discusses the convergence of the ring’s probability distribution. Simplify a cycle to a Simple-Cycle via node elimination operation, probability convergency of the cycle is proved. Based on cycle’s probability convergency, probability propagation method is proposed.

Keywords—bayesian network; cycle; convergency; probability propagation

I. INTRODUCTION

Software engineering practice shows[1], software project’s risks root in the uncertainties of software project process. Modeling software project process to abstract, represent and analyze the software process, can identify risk factors, estimate risk expense and construct risk relaxation program.

A Bayesian Network[2] (BN) is a graphical model that encodes relationship among variables of interest. At the same time, a BN is also a probability inference technique to deal with conditional uncertainty among different knowledge components, and it offers an ideal mathematic formalization for representation of uncertainty. A BN is the integration of graph theory and probability theory. A BN is a complicated evaluated causality graph intuitively, each node denotes a variable in the domain, namely event. The arc between nodes denotes the direct causality between events. Generally speaking, a BN is a directed acyclic graph (, , ) where is the set of nodes and is the set of arcs. For any , , , , , and are named as input nodes of this cycle. Every node has its conditional probability distribution reflecting the dependence relationship between its parent nodes and itself. A BN represents joint probability distribution of involved variables in the interested domain essentially. Due to the conditional independence, joint probability distribution of the network can be obtained by the chain equation (1). Lots of probability inference methods[3] have been exploited based on equation (1).

\[ P = \pi^u_0 \cdot P(x_i | pa(x_i)) \]  (1)

For the firm mathematic theoretic base, BN is deemed as the ideal tool for knowledge representation, inference and estimation in uncertain environment[4]. In the software engineering domain, BN has been used to estimate software’s quality and faults successfully[5,6]. But when we model software project process as a BN to estimate and evaluate software project risk, there are always directed cycles in the BN, for doing poorly done work over again, iterative development and etc. are possible in software project process. Probability inference of BN with directed cycles can not be carried out as Equation (1) shows. In this paper, we will discuss the problem of cyclic BN.

An outline of the remainder of the paper is as follows. In Section 2, we discuss how the cyclic BN can be simplified. In Section 3, we discuss the probability convenency for directed cycle. In section4, we discuss probability propagation for cyclic BN and carry out a case study. Finally, in Section 5, we conclude with a summary and a statement of future research.

II. SIMPLIFY THE CYCLIC BAYESIAN NETWORKS

A. Definition

Firstly, we suppose that all the nodes in this paper are discrete variables, and the number of their values is limited. The relative conceptions are listed as follows:

**Cyclic Bayesian Network[7]** (Cyclic BN): if any directed cycles existed in a Bayesian Network (, ), then (, ) is named as a Cyclic Bayesian Network. Directed cycle is marked as (), where is the set of nodes on the cycle, and is the set of directed arcs.

**Input of a cycle**: A node on a cycle which have parent nodes and don’t belong to the cycle are named as input nodes of this cycle and as an Inceptor-Node.

**Inquired-Node**: An Inquired-Node is the node whose values and probability distribution demand to be obtained. For a directed cycle, the node possessing descent nodes which don’t belong to this cycle, can be treated as an Inquired-Node.

**Node-Elimination**: the operation of eliminating some node on a cyclic BN, at the same time changing the probability distribution of relative nodes conforming to certain rules, and having no influence on the probability propagation, is named as Node-Elimination.

**Normal-Node**: a node on a directed cycle is named as an Ordinary-Node, when it isn’t an evidence node nor an Inquired-Node.
Simple-Cycle: a cycle possessing just two nodes is named as a Simple-Cycle.

B. Simplify the Directed Cycle

Lemma 1 For a cycle possessing no inputs, the Ordinary-Nodes can be eliminated one by one until the cycle is converted into a Simple-Cycle.

Lemma 2 For a cycle possessing inputs, the Normal-Nodes can be eliminated one by one until the cycle is converted into a Simple-Cycle.

Proof: Let us consider a cycle possessing three nodes. There are three cases according to different Inquired-Node:

1) if the Inquired-Node is an evidence node

As Figure 1 shows, Inceptor-Node is an evidence node. After node B is eliminated, the topological structure is the right figure in Figure 1. Calculate each node’s probability distribution as follows:

\[ P(C) = \sum_{A \in E} P(A, E, C) = \sum_{A \in B \in E} P(A, B, C, E) = \sum_{A \in B \in E} P(A \mid E) P(E) \sum_{B \mid A} P(B \mid A) P(C \mid B) \]

Then \( P(C \mid A) = \sum_{B \mid A} P(B \mid A) P(C \mid B) \), \( P(A \mid C) = P(A \mid C) \) and \( P(A \mid E) = P(A \mid E) \).

2) If the Inceptor-Node is an Inquired-Node

As Figure 2 shows, Inceptor-Node is an Inquired-Node. After node B is eliminated, the topological structure is the right figure in Figure 2. Calculate the probability distribution as follows:

\[ P(C) = \sum_{A \in E} P(A, E, C) = \sum_{A \in B \in E} P(A, B, C, E) = \sum_{A \in B \in E} P(A \mid E) P(E) \sum_{A \mid B} P(B \mid A) P(C \mid B, E) \]

3) If the Inceptor-Node is an Ordinary-Node

As Figure 3 shows, Inceptor-Node B is an Ordinary-Node. After node B is eliminate, the topological structure is the right figure in Figure 3. Calculate the probability distribution as follows:

\[ P(C) = \sum_{A \in E} P(A, E, C) = \sum_{A \in B \in E} P(A \mid E) P(E) P(C \mid A, E) = \sum_{A \in B \in E} P(A \mid E) P(B \mid A, E) P(C \mid B) = \sum_{A \in B \in E} P(A \mid E) \sum_{B \mid A} P(B \mid A) P(C \mid B) \]

Then \( P(C \mid A, E) = \sum_{B \mid A} P(B \mid A) P(C \mid B) \).

Synthesize the above three instances, Lemma 2 is proved.

Theorem 1 Any directed cycle can be simplified to a Simple-Cycle via Node-Elimination operation.

Lemma 1 indicates that no-inputs cycle can be simplified to a Simple-Cycle, and Lemma 2 indicates the cycle possessing inputs can also be simplified to a Simple-Cycle, so Theorem 2 is tenable. That is to say, every directed cycle on a cyclic BN can be simplified to a Simple-Cycle.

III. CONVERGENCY ANALYSIS FOR DIRECTED CYCLE

Lemma 3 If every node on a Simple-Cycle has just two values, then the cycle’s probability distribution is convergent.

Proof: The Simple-Cycle’s structure is showed in Figure 4, node A and B are both two-valued nodes, \( A(A_1, A_2) \), \( B(B_1, B_2) \).

Let \( P(A_i \mid B_j) = \alpha_i \), \( P(B_i \mid A_j) = \beta_j \) where \( i = 1, 2 \) and \( j = 1, 2 \), and the initial probability state is \( B(B_1, B_2) \).

1) After the first propagation, probability states of each node are \( A(A_1, A_2) \cdot P(A_i, A_2) \) and \( B(B_1, B_2) \cdot P(B_i) \):

\[ P_{i1} = P_{i1} \alpha_1 + P_{i2} \alpha_2, \]
\[ P_{i2} = P_{i1} \alpha_2 + P_{i2} \alpha_1, \]
\[
P_n = P_0 + \sum_{i=1}^{n} P_i,
\]

where \(P_i = P_{i-1} + \sum_{j=1}^{i} P_{j-1} P_{j-1}\), and

\[
P_n = w + sP_{n-1},
\]

where \(|s| < 1\). Then, \(\lim_{n \to \infty} (1 + s + \ldots + s^n) = \frac{1}{1-s}\), \(\lim_{n \to \infty} s^n = 0\).

So

\[
P_n = \lim_{n \to \infty} P_n = \frac{w}{1-s}.
\]

2) After probability propagate \(n\) times, the probability distribution of node \(B\)'s state \(\beta_i\) is:

\[
P_n = w + sP_{n-1},
\]

where \(\alpha_i \beta_i + \alpha_i \beta_i \Delta x\),

\[
(\alpha_i \beta_i + \alpha_i \beta_i) - (\alpha_i \beta_i + \alpha_i \beta_i) \Delta x,
\]

\[
P_n = w + sP_{n-1},
\]

where \(n = \alpha_i \beta_i + \alpha_i \beta_i \leq 1\), \(|s| = (\alpha_i \beta_i + \alpha_i \beta_i) - (\alpha_i \beta_i + \alpha_i \beta_i) \Delta x\), \(|s| \leq 1\).

If \(s < 1\), then: \(\lim_{n \to \infty} (1 + s + \ldots + s^n) = \frac{1}{1-s}\), \(\lim_{n \to \infty} s^n = 0\).

So

\[
P_n = \lim_{n \to \infty} P_n = \frac{w}{1-s}.
\]

The probability distribution of each node on the cycle is:

\[
P_n = w + sP_{n-1},
\]

where \(\alpha_i \beta_i + \alpha_i \beta_i \Delta x\), and

\[
P_n = w + sP_{n-1},
\]

where \(n = \alpha_i \beta_i + \alpha_i \beta_i \leq 1\), \(|s| = (\alpha_i \beta_i + \alpha_i \beta_i) - (\alpha_i \beta_i + \alpha_i \beta_i) \Delta x\), \(|s| \leq 1\).

If \(s < 1\), then: \(\lim_{n \to \infty} (1 + s + \ldots + s^n) = \frac{1}{1-s}\), \(\lim_{n \to \infty} s^n = 0\).

So

\[
P_n = \lim_{n \to \infty} P_n = \frac{w}{1-s}.
\]
be the ultimate probability distribution, then:

\[ P'(b_i) = P'(B_i) \]
\[ P'(b_i) = \sum_{i} P'(A)P(b_i | A) = P'(a)P(b_i | a) + P'(a)|P(b_i | a) \quad i = 2, ..., n \quad (4) \]

2) if node A and node B are both multi-valued, then let A be \( a_1, a_2, ..., a_m \) and B be \( b_1, b_2, ..., b_n \) in a general way.

So:

Taken a, as a dummy state \( A_1 \), and \( a_2, ..., a_m \) as another dummy state \( A_2 \), then we know that the cycle possesses probability constringency as we just discussed in 1).

Let \( P'(A_1), P'(A_2), P'(b_1), ..., P'(b_n) \) be the ultimate probability distribution, where

\[ P'(b_i) = \sum_{i} P'(A_1)P(b_i | A_1) = P'(a_1)P(b_i | a_1) + P'(a_2)|P(b_i | a_2) \quad i = 1, ..., n \quad (5) \]
\[ P'(a_j) = \sum_{i} P'(b_i)P(a_j | b_i) \quad j = 1, ..., m \quad (6) \]

So, the ultimate probability state of the cycle is

\((P'(a_1), P'(b_1)) \quad i = 1, ..., m; \quad j = 1, ..., n\

Synthesize the discuss of 1) and 2), Lemma 4 is proved.

**Theorem 2** A directed cycle’s local probability distribution on a cyclic BN is convergent.

Theorem 2 can be deduced easily because Theorem 1, Lemma 3 and Lemma 4 were proved.

Any cycle on a cyclic BN must converge to some stable probability distribution \( P^* \), and \( P^* \) can be calculated by Equation (2) - (6).

**IV. PROBABILITY PROPAGATION FOR CYCLIC BAYESIAN NETWORK**

Directed cycle’s probability convergency is the theoretic base for probability propagation. Let probability distribution \( P^* \) is a directed cycle’s convergent state, then the cycle will be treated as a special node of the BN with probability distribution \( P^* \), so all the probability inference method introduced in Reference 3 can be used to update belief.

But in generally, engineering practice can not repeat endless. So, switch node is introduced to solve probability propagation problem whose position and value \( \epsilon \) is determined by engineering practice. Assuming the local probability distribution of the cycle is \( P^* \) when the switch node approach to \( \epsilon \), cut the fan-in arc or fan-out arc to destroy the cycle. It is important for software project engineering practice to destroy the cycle when local probability reaches some point, because the ultimate purpose is to reach some satisfied situation within limited budget not to perfection. For the extreme instance \( s = -1 \) as discussed in section 3, introducing switch node is the only probability inference method.

A simple example is given in Figure 5 to show how the probability distribution propagate in the cycle. Software test is an iterative process usually, expressed as figure I in Figure 5. Conditional probability of each node is known, node A is evidence node, and B’s initial state is known also.

Given the cycle L’s joint probability distribution converging to \( (P(B), P(C), P(D)) \), then as figure II shows, taking node D as evidence node, carry on probability propagation in the all the network via Equation (1).

In software project practice, when the remnant errors less than \( \epsilon \), system test will always be stopped. So, we choose node D as switch node and its threshold value is \( \epsilon \). When \( P^{+1}(D < \epsilon) < P^*(D < \epsilon) > P^{+1}(D < \epsilon) \), evaluate local joint probability distribution as \( P^* = P^8 \), and disconnect the directed arc \( D \rightarrow B \). Then as figure III shows, we can calculate the joint probability distribution for all nodes in the network according to Equation (1).

**V. CONCLUSION**

Modeling the uncertainties in the software project process by Bayesian Network, provides a useful method to identify, evaluate and reduce latent risks in software project process. Modeling the iterative development of the software project based on Bayesian Network, results in the directed cycles in the BN process model. BN is a graphical model that encodes relationships among variables of interest, and it is a directed acyclic graph which has no ability to deal with the directed cycles. In this paper, **Theorem 1** indicates that, all the ordinary nodes on the directed cycle can be eliminated one by one until the cycle is converted into a simple cycle, and **Theorem 2** indicates that any simple cycle possesses probability astringency, so all the directed cycles are probability convergent. Based on probability convergency of the directed cycle, two kinds of probability propagation route are proposed.

In this paper, we discuss cyclic BN premising the nodes are all discrete variables, how about the continuous variables? We will discuss further in the future.

**REFERENCES**


