

基于 Bayes 网的软件失效预测模型^{*}

白成刚¹ 俞蒙槐² 胡上序² 蔡开元¹

(北京航空航天大学自动控制系 北京100083)¹ (浙江大学智能信息研究所 杭州310027)²

Software Failure Predication Model Based on Bayesian Network

BAI Cheng-Gang¹ YU Meng-Huai² HU Shang-Xu² CAI Kai-Yuan¹

(Dept. of Automatic Control, Beijing University of Aeronautics and Astronautics, Beijing 100083)¹

(Intelligent Information Institute, Zhejiang University, Hangzhou 310027)²

Abstract Owing to the complexity of software development, the software reliability model should not only have the capability of dealing with multiple complex factors, but also provide the function of flexibility in construction. So far, no software reliability model is universally applicable. The main reason for this is of too many conditions, thus making software reliability models introvert. Bayesian network is a powerful tool for solving this problem, which exhibits strong adaptability in dealing with problems involving complex variant factors. In the paper, software failure predication model based on Markov. Bayesian network is established and analyzed thoroughly. Then a method of solving the model is given. Finally, through an example the validity of the model is validated.

Keywords Software failure, Bayesian network, Markov Bayesian network

1 引言

随着航空电子设备向着综合化和数字化方向发展,计算机已成为现代航空电子系统的核心。机载计算机的用量迅速增长,软件的复杂性变得越来越高。20世纪60年代改型的FB-111战斗轰炸机采用的软件只有10万条指令,70年代研制的F-16战斗机的软件有25万条指令,新一代的战斗机F-22具有700万条指令(采用Ada语言)。美国空军发现军用飞机中任务关键的系统出现的故障70%与软件有关,F/A-18飞控系统在20,000小时的试验中,软件故障309次,而硬件故障271次。更为重要的是在诸如电传操纵系统之类的关键系统中,软件失效可能造成机毁人亡的事故,如瑞典的JAS39战斗机就发生过。

由此可见,研究如何合理预测软件失效就具有紧迫的现实性。迄今为止,人们已提出了众多的软件失效预测模型。文[1~3]做了详尽的描述。目前广泛使用的是概率类模型。但是为了保证模型的可解性,常常规定一些假设条件,而在实际情况中这些条件是很难得到满足的^[4,5]。在模型的可解性和模型对问题表示的充分性之间存在的矛盾严重制约了概率类模型进一步发展。因此,人们开始寻求建立非概率类模型。有代表性是神经网络方法^[6,7]。但是,神经网络方法也存在着问题:一是结构选取的随意性,无法反映各输入量之间的内在关联;二是需要输入数据的完整性,即如果实际问题中缺少若干的输入结点的数据,神经网络就难以对此进行有效的处理。还有,对于软件失效这样一个非常复杂的问题,我们必须考虑到一切能考虑到的因素,并且尽可能地利用各种先验信息和专家的经验 and 知识。

现在一种较新的技术 Bayes 网使得对问题求解的范围大大拓宽了。Bayes 网使许多相互关联的因素及其这种关联充

分表示出来了。它把用概率分布表示的科学严格性与用专家经验表示的先验信息有机结合起来,能够很好地处理不确定问题。Bayes 网越来越广泛地应用于各个领域^[8~12]。本文试图将 Bayes 网应用到软件失效预测中去。

2 Bayes 网

有关 Bayes 网的定义有许多种。其基本形式为^[13]: Bayes 网是一种有向概率图(Probability Graph),它将感兴趣的变量用有向弧连接起来,这种连接表达了变量之间的条件关联。下面是较严格的定义:

一个 Bayes 网定义为一个三元集合 $\{X, S, P\}$, 其中:

(1) X 是一个变量集合, $X = \{X_1, \dots, X_n\}$ 。

(2) S 是一个条件概率分布集合, $S = \{p(X_1 | \text{Parents}(X_1)), \dots, p(X_n | \text{Parents}(X_n))\}$, $\text{Parents}(\cdot)$ 是一个从一结点到一结点集合的映射, $\text{Parents}(X_i) \subset X$ 它表示了 X_i 的所有父结点, $p(X_i | \text{Parents}(X_i))$ 表示变量 X_i 的条件分布。

(3) P 是一个边沿概率分布集合, $P = \{p_1(X_1), \dots, p_n(X_n)\}$, $p_i(X_i)$ 表示变量 X_i 的概率分布。

在 Bayes 网中,变量用来表示现实世界的事件或对象。通过观察研究这些变量的表现就可以把该问题模型化。一般是首先计算(或通过专家经验确定)各变量的概率分布和变量之间的条件概率分布,然后通过变量的概率分布和变量之间条件概率分布得到所有变量的联合概率分布,从而在已知某些变量的情况下对另外某些感兴趣的变量作出推断。

Bayes 网建立在强大的 Bayes 统计理论基础上,和其它传统的数据分析方法相比它有以下的特点:

(1) Bayes 网可以联结所有感兴趣的变量,所以它可以尽可能地增加分析的精度。并且能够处理不完全数据问题。比如某一问题需要两个输入量,一个输出量,而两个输入量具有相

^{*} 国家杰出青年科学基金和国家攀登计划资助。

关性。如两个输入量的值都能被观察到,预测输出量的值,这是大多数模型都能胜任的。但是,当其中之一未被观察到时,大多数模型就会给出不精确的预测,因为它们未能利用这两个输入量具有的相关性。而 Bayes 网则可以一种比较自然的方式度量这种相关性。

(2) Bayes 网可以学习变量之间的因果关系。这能使我们了解感兴趣变量的相互关系,以便进行有效的预测。比如,一个市场分析员可能想为了增加某项产品的销售量,是否值得加大某项广告的宣传力度。为了回答这个问题,市场分析员需要知道广告是否是增加销售的一个因素,以及在何种程度上。Bayes 网可以帮助回答类似的问题。

(3) Bayes 网采用 Bayes 统计方法能够充分地把问题领域的有关知识和获取的当前数据相结合。任何一位从事过实际问题开发的人员都知道先验知识或问题领域知识的重要性,尤其是当数据非常缺少或获取数据的代价很昂贵时。

3 基于 Bayes 网的软件失效预测模型

令 $X_i (1 \leq i \leq n)$ 表示发生第 i 次失效时软件中所含的缺陷数, $Y_i (1 \leq i \leq n)$ 表示第 i 次失效距第 $i-1$ 次失效的时间段。当软件发生失效后,软件调试人员会排除缺陷。在排除过程中,既有可能使软件中的缺陷数减少,也有可能引入新的缺陷, X_{i+1} 是依赖于 X_i 的。另外,软件失效发生时间的间隔是受软件中的缺陷数影响的。如果设 $p(y_i | x_i)$ 和 $p(x_{i+1} | x_i)$ 与时间无关,我们用马尔可夫 Bayes 网^[14]表示这一模型。网络结构如图1:

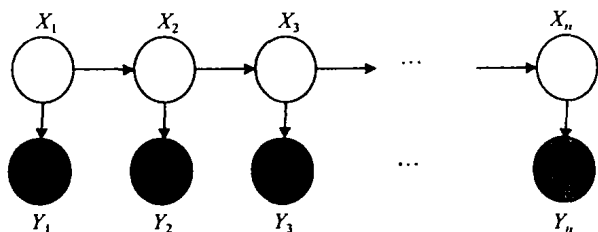


图1 马尔可夫 Bayes 网

4 有关分布的确定

要用基于马尔可夫 Bayes 网的软件失效预测模型进行推断,首先要确定三个分布:软件缺陷数的初始分布 $p_0(x)$;在已知软件内部缺陷数条件下,软件失效时间的分布;在发生失效的情况下,软件缺陷的排除数分布。

(1) 软件缺陷数的初始分布 $p_0(x)$ 可以根据软件规模通过 Halstead 的软件科学法来估计。

(2) 在软件内部缺陷数为 x_i 时,当发生失效时,软件开发人员就要排查缺陷,既有可能把相应缺陷排除掉,也有可能引入新的缺陷,即缺陷数有可能增加也有可能减少,但总的趋势应是减少。可以假设 x_{i+1} 服从期望为 βx_i ($\beta > 0$), 根方差为 σ 的正态分布。参数 β 决定于缺陷的排除情况, σ 决定于缺陷排查人员的水平离散程度。因为一般软件可靠性是增长的,所以一般应有 $\beta < 1$ 。

(3) 在软件的内部缺陷数为 x_i 时,我们来分析软件的失效情况。假设软件的可执行行数为 N , 则知平均一个执行行的缺陷数为 x_i/N 。再假设软件在单位时间内执行的执行行数为 h 。而软件每执行一个软件缺陷时,软件失效的概率为 p 。则软件在单位时间内失效的概率为:

$$1 - (1 - p)^{hx_i/N} = 1 - e^{(hx_i/N) \ln(1-p)} = 1 - e^{-\alpha x_i}$$

其中, $\alpha = -\frac{h}{N} \ln(1-p) > 0$

则有软件在单位时间内失效的概率,即失效率为 $1 - e^{-\alpha x_i}$,

所以可假定在软件的内部缺陷数为 x_i 时,软件的失效分布为指数分布:

$$p(y_i | x_i) = (1 - e^{-\alpha x_i}) \cdot e^{-(1 - e^{-\alpha x_i}) y_i}$$

5 软件失效预测

对于如图1的马尔可夫 Bayes 网, $X_1, \dots, X_n, Y_1, \dots, Y_n$ 的联合概率密度为:

$$p(x_1, \dots, x_n, y_1, \dots, y_n) = p(x_1) \prod_{i=1}^n p(y_i | x_i) \prod_{i=1}^{n-1} p(x_{i+1} | x_i) \quad (1)$$

$X_1, \dots, X_n$ 的边沿联合概率密度为:

$$p(x_1, \dots, x_n) = p(x_1) \prod_{i=1}^{n-1} p(x_{i+1} | x_i) \quad (2)$$

已知 $y_1, \dots, y_n$ 时, $X_1, \dots, X_n$ 的条件联合概率密度为:

$$p(x_1, \dots, x_n | y_1, \dots, y_n) = \frac{p(x_1, \dots, x_n, y_1, \dots, y_n)}{p(y_1, \dots, y_n)} \quad (3)$$

其中, $p(y_1, \dots, y_n)$ 是 $Y_1, \dots, Y_n$ 的边沿联合概率密度。由(3)式就可对结点 $X_1, \dots, X_n$ 作出估计,得到 $\hat{x}_1, \dots, \hat{x}_n$ 。然后由分布 $p(x_{n+1} | \hat{x}_n)$ 得到 $\hat{x}_{n+1}$, 进而再由分布 $p(y_{n+1} | \hat{x}_{n+1})$ 得到 $\hat{y}_{n+1}$, 此即软件第 $n+1$ 个失效时刻的预测值。

因为(3)形式比较复杂,直接计算有一定的难度,可采用 Gibbs 抽样技术来进行。

由 Gibbs 抽样法的理论^[15]可以知道,在我们的问题中关键在于对分布 $p(x_k | x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n, y_1, \dots, y_n)$ 的抽样。

$$p(x_k | x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n, y_1, \dots, y_n) = \frac{p(x_1, \dots, x_n, y_1, \dots, y_n)}{p(x_1, x_{k-1}, x_{k+1}, \dots, x_n, y_1, \dots, y_n)} \quad (k=2, \dots, n-1) \quad (4)$$

(4)式的分母:

$$\begin{aligned} I &= \int_{x_k} p(x_1, \dots, x_n, y_1, \dots, y_n) dx_k = \int_{x_k} p(x_1) \prod_{i=1}^n p(y_i | x_i) dx_k \\ &= p(x_1) \prod_{i=1}^{k-1} p(y_i | x_i) \prod_{i=k+1}^n p(y_i | x_i) \prod_{i=1}^{k-2} p(x_{i+1} | x_i) \\ &\quad \prod_{i=k+1}^{n-1} p(x_{i+1} | x_i) \cdot \int_{x_k} p(y_k | x_k) p(x_k | x_{k-1}) p(x_{k+1} | x_k) dx_k \end{aligned} \quad (5)$$

(5) 式中 $\int_{x_k} p(y_k | x_k) p(x_k | x_{k-1}) p(x_{k+1} | x_k) dx_k$ 在已知 $p(y_k | x_k), p(x_k | x_{k-1}), p(x_{k+1} | x_k)$ 的情况下是可以计算的。

(4)式的分子:

$$\begin{aligned} II &= p(x_1, \dots, x_n, y_1, \dots, y_n) = p(x_1) \prod_{i=1}^n p(y_i | x_i) \prod_{i=1}^{n-1} p(x_{i+1} | x_i) \\ &= p(x_1) \prod_{i=1}^{k-1} p(y_i | x_i) \prod_{i=k+1}^n p(y_i | x_i) \prod_{i=1}^{k-2} p(x_{i+1} | x_i) \\ &\quad \prod_{i=k+1}^{n-1} p(x_{i+1} | x_i) \cdot p(x_k | x_{k-1}) p(x_{k+1} | x_k) \end{aligned} \quad (6)$$

(6) 式只有 $p(y_k | x_k) p(x_k | x_{k-1}) p(x_{k+1} | x_k)$ 与 x_k 有关。

对于 $k=1$ 时可以做类似推导。所以 $p(x_k | x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n, y_1, \dots, y_n)$ 是可以抽样的。

6 实例

作者在参与仪征化纤集团仿真培训系统(简称 FZPS)开发时,搜集的失效数据如表1所示:

表1 FZPS 软件失效数据

失效序号	天数	累积天数
1	5	5
2	9	14
3	14	20
4	6	26
5	10	36
6	3	39
7	8	47
8	11	58
9	9	67
10	7	74
11	2	76
12	5	81
13	4	85
14	12	97
15	11	108
16	3	111
17	6	117
18	10	127
19	13	140
20	14	154
21	18	172
22	27	199

采用表1中的 FZPS 失效数据对本文提出的模型进行了估计和预测,然后和 Jelinski-Moranda 模型(简称 JM 模型), Goel-Okumoto NHPP 模型(简称 GO 模型)进行了比较(见图2, n 为失效序号, $t(n)$ 为失效时刻)。JM 模型和 GO 模型是两个经典的软件可靠性模型,具有代表性,常被用作研究的参照模型。通过计算结果可以看出, Bayes 网模型的估计和预测能力是较强的。

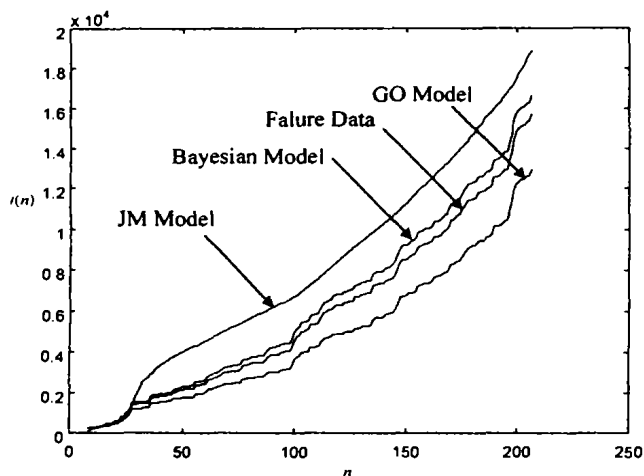


图2 Bayes 网模型和 JM 模型、G-O 模型预测失效时刻的比较情况

结论 本文讨论了基于 Bayes 模型的软件失效预测模型。通过文中实例,可以看出该模型的预测能力是较强的。其原因在于 Bayes 模型利用了更多的信息,如软件缺陷数的初始分布。对于软件这种极为复杂的逻辑系统,多考虑一些因素是有必要的。

参考文献

- Goel A L. Software reliability models: Assumptions, limitations, and applicability. IEEE Transactions on Software Engineering, 1985, SE-11: 1411~1423
- Malaiya Y K, Srimani P K. Software reliability models: Theoretical Developments, Evaluation and Application. IEEE CS Press, 1991
- Xie M. Software reliability models: a selected bibliography. Journal of Software Testing, Verification and Reliability, 1993, (3): 32~38
- Bishop P, Pullen F D. Probabilistic modeling of software failure characteristics. In: Proc. IFAC. Workshop SAFECOMP'88, 1988. 87~93
- Littlewood B. Software reliability modeling: Achievements and limitations. In: Proc. COMPEURO'91, Bologna, 1991
- Khoshgoftaar T M, Pandya A S, More H B. A neural network approach for predicting software development faults. In: Proc. of the Third Intl. Symposium on Software Reliability Engineering, IEEE Computer Society Press, Oct. 1992. 83~89
- Karuanithi N, Witeley D, Malaiya Y K. Prediction of software reliability using neural network. In: Proc. 1991 IEEE Int. Symp. on Soft. Rel. Eng., May 1991. 124~130
- John S B, Russ B. Automating computer bottleneck detection with belief nets. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1995. 36~45
- Bickmore, Timothy W. Real-time sensor data validation: [NASA Contractor Report 195295]. National Aeronautics and Space Administration, April, 1994
- Huang T, Koller D, Malik J, et al. Automatic symbolic traffic scene analysis using belief networks. In: Proc. of National Conf. on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1994
- David M, Krzysztof M, Russell A G. Explanation in belief networks. to appear in Journal of Computational and Graphical Statistics, 1996
- Kazuo J E, Til S. Fraud/uncollectible debt detection using a Bayesian network based learning system: a rare binary outcome with mixed data structures, computer bottleneck detection with belief nets. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1995. 157~166
- Pearl J. Fusion, propagation and structuring in belief network. Artificial Intelligence, 1986, 29: 241~288
- Jordan M I. An introduction to variational methods for graphical models. To appear: Jordan, M. I. (ed.), Learning in Graphical models, Kluwer Academic Publishers
- Gelfand A, Smith A. Sampling based approaches to calculating marginal densities. JASA 85: 385~409