使用贝叶斯网评估基础软件平台的互操作性

兰雨晴 赵 同

(北京航空航天大学 计算机学院, 北京 100191)

摘 要: 互操作性是当前软件最重要的特性之一. 通过分析问题域,结合贝叶斯网的特征域,提出使用贝叶斯网来解决基础软件平台的互操作性评估问题. 首先根据问题域选取贝叶斯算法,并收集实际数据以引入与问题相关的领域知识. 根据所选取的算法构造互操作性的贝叶斯网结构,并且进一步学习此结构的参数. 在此过程中,对选取的 K2算法进行改进. 然后,利用贝叶斯推理来根据互操作性的结构和参数得出评估对象的互操作性等级. 最后,一个实例讲述了方法具体的应用过程. 实验结果证明了方法的合理性.

关 键 词: 贝叶斯网; 互操作性; 基础软件平台; K2搜索算法

中图分类号: TP 311. 5

文献标识码: A **文 章 编 号**: 1001-5965(2009)09-1148-04

Evaluate interoperability of foundational software platform by Bayesian networks

Lan Yuqing Zhao Tong

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: Interoperability is one of the most important features of current software. Bayesian networks was proposed to solve the problem of evaluating interoperability of foundational software platform after analyzing both the problem domain and the Bayesian networks feature domain. First of all, Bayesian algorithm was chosen according to the problem domain, and actual data were gathered for importing domain knowledge. Bayesian structure was formed by the selected algorithm and used to learn its parameters. During this process, the chosen K2 algorithm was improved. After that, Bayesian reasoning was used to evaluate the grade of interoperability of foundational software platform according to the established Bayesian structure and parameters. At last, an application showed how to use this brand-new model to evaluate the interoperability of foundational software platform. The experimental result proved the reasonableness and usefulness of the model.

Key words: Bayesian network; interoperability; foundational software platform; K2 algorithm

20世纪后期以来,世界经济进入全球化和飞速发展阶段,急需一种可操作途径来实现计算资源的整合和信息共享. 于是,开放标准和互操作性逐渐成为软件工业发展的核心问题[1-2]. 互操作性指一个软件系统与另一个软件系统间具有接收、处理并共享所发送信息的能力[3]. 解决软件的互操作性问题已成为当务之急,并形成当前软件产业发展中的一股技术潮流.

互操作性是所有软件共同的一个特性,但是对于某些软件形态来说尤为重要,最具有代表性的便是基础软件平台. 基础软件平台 (FSP, Foundational Software Platform)是指由操作系统、中间件、数据库、安全产品以及办公套件等通用性软件组建的一系列应用支撑平台. 它是软件学科不断发展的新产物,是软件形态多样化的结果. 由于对其上层应用起着支撑作用,并且极大地影响到其上层应用的质量,基础软件平台这种软件形态正变得越来越重要.

收稿日期: 2008-09-24

基金项目: 国家 863计划资助项目 (2009AA012406)

作者简介: 兰雨晴 (1969-),男,内蒙呼和浩特人,博士生, lanyuqing@buaa.edu.cn.

研究基础软件平台的互操作性既能有效地指导用户来挑选互操作性好的平台,指导厂商有针对性地改进平台的互操作性,又能为研究其它软件类型的互操作性提供极大的参考价值.

针对这种情况,本文提出用贝叶斯网络来解决基础软件平台的互操作性评估问题. 此方法的主要思想是,先利用贝叶斯学习来构建互操作性的贝叶斯网,然后再利用贝叶斯推理,评估所给平台的互操作性.

1　研究基础

贝叶斯网 (BN, Bayesian Networks,又称为贝叶斯信念网或 BBN),主要用于解决在不确定环境中知识表示、问题推理[4]. 它介于联合概率分布和朴素贝叶斯分类器之间,既表示变量之间的依赖关系又作出了一些合理的独立性假设[5].

贝叶斯网的研究集中在 2个方面:推理和学习. 贝叶斯网学习是根据数据和专家知识建立贝叶斯网. 现有许多经典的贝叶斯网学习算法,如: Russell提出的基于梯度上升的参数学习算法[6], Cooper和 Herskovits建立的著名的基于打分函数和爬山法搜索算法的 K2算法[7]等. 贝叶斯网的推理则是根据学习的结果对预期目标进行推理和预测的过程. 理论上,对于任意贝叶斯网的概率推理都是一个非确定性多项式 (NP, Non-deterministic Polynomial)难题[8]. 但在实践中很多推理方法都是有效的[9],并且还出现了一些实时的推理方法[10].

在应用方面,贝叶斯网近年来已经成为人工智能领域在不确定环境中进行知识表示和推理的一种有效工具,被应用于各种专家系统和诊断系统,也有一些学者将贝叶斯网应用到软件工程领域[11-13].

由于基础软件平台的互操作性是一个不确定性问题,通过分析贝叶斯网的特征域之后,本文决定采用贝叶斯网来解决此问题. 图 1为问题域与贝叶斯特征域的对比.

为了更好地理解下一节的算法,下面给出贝叶斯网相关的一些基本概念:

1) 贝叶斯网:利用一组条件独立性假设和一组局部条件概率集合来表示一组变量在联合空间中的联合概率分布. 它由 2个部分组成:结构 B_S 和参数 B_Θ. 其中, B_S表示贝叶斯网的结构; B_Θ则表示贝叶斯网的参数.

$P_{A_i} = p_{a_i}$, P_{A_i}表示贝叶斯网中随机变量 X_i的前驱节点集合, p_{a_i}则表示 P_{A_i}的取值.

$p(X_i = x_i \mid p_{a_i}) = p(x_i \mid p_{a_i})$表示贝叶斯网中随机变量 X_i取值为 x_i的条件概率.

$D = \{X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n\}$表示训练数据.

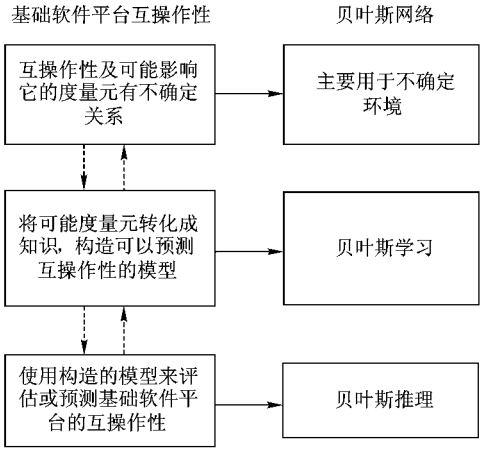

图 1　问题域与贝叶斯特征域对比

2) 3种连接方式:在贝叶斯网的有向无环图中,随机变量 b与其在连接路径上最邻接的 2个变量 a和 c之间存在 3种连接方式:线性连接、会聚连接和分叉连接,见图 2

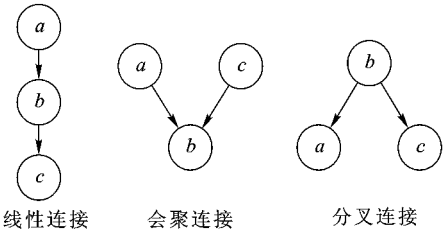

图 2　3种连接方式

3) 领域知识:通常来说,一个人对事物 a的主观判断取决于其自身的知识 K,记为 $P(a \mid K)$.

2　方　法

本文提出用贝叶斯网来解决基础软件平台互操作性的评估问题. 主要思路是:通过贝叶斯学习来构造互操作性的预测模型 B_S,并学习该模型的参数 B_Θ;然后通过贝叶斯推理来对给定的基础软件平台实例进行评估. 其中, B_S的构造是收集足够的经验数据,加上已有的领域知识,通过贝叶斯网学习算法得出. 主要步骤如图 3所示. 方法主要包含了 3个部分:准备工作、数据处理和贝叶斯学习.

2.1　准备工作

准备工作主要包含 2个部分,贝叶斯学习算法的选择以及领域知识的收集.

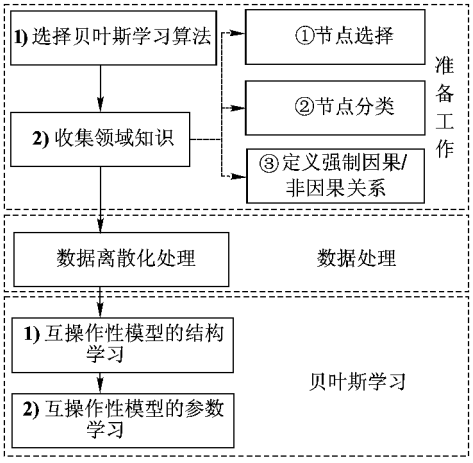

图 3 方法流程图

2.1.1 选择贝叶斯学习算法

本文选择了离散型贝叶斯学习算法中的经典算法[12]. 选择离散型的原因主要是现有的连续型贝叶斯网学习技术要求数据满足正态分布,而暂时还未发现互操作性的度量有满足正态分布的趋势.

2.1.2 收集领域知识

由于基础软件平台的研究刚刚起步,贝叶斯训练又需要大量数据,所以经验数据很难保证. 在经验数据较少的情 下,领域知识就在很大程度上可以往好的方向引导训练结果. 本文从以下几个方面收集领域知识.

1) 节点选择: K2算法的一个重要输入是有序节点集 N_s. 为使节点集更可靠,选择了 4个方面的共 50名专家来调查整理可能影响基础软件平台互操作性的度量元和原因节点. 专家意见被整合并调整,来控制节点数量,最终 13个节点被定义,其中前 10个节点是与基础软件平台相关的外部、内部质量度量,最后 3个是可能导致基础软件平台互操作性变化的原因. 相关节点及其定义为: ①安全:平台保护信息和数据的能力,以使未授权的人员或系统不能阅读或修改这些信息和数据,而不拒绝授权人员或系统对其操作; ②适应性:平台适应不同指定环境的能力; ③系统兼容性:平台内部的和谐运作,能和硬件、软件、通信插件之间实现内部融合的能力; ④容错性:在出现故障的情况下,平台维持规定的性能级别的能力; ⑤独立性:平台及其构件与环境无关,毋需采用额外的活动或手段就可适应不同环境的能力; ⑥可安装性:平台作为一个整体,在制定环境中被安装的能力; ⑦可取代性:平台中的任一构成软件能够被同类软件取代,且不影响平台整体质量的能力; ⑧数据可交换性:平台最终用户不能与其他软件产品之间交换数据的频率; ⑨接口依从性:平台对于相关标准、公约或规定的依从能力,如发生依从性问题接口数量等; ⑩存取可控性:对平台进行访问的可控程度,以及用户访问平台的完整程度.

由于文章篇幅限制,这里不对各节点的详细内部、外部度量元和计算公式等一一解释.

2) 节点分类:根据节点的发生时间或逻辑关系来初步划分原因节点和结果节点.

分析 N_s 不难发现,数据可交换性,接口依从性,以及存取可控性是结果节点,其余 10个均为原因节点.

3) 定义强制因果/非因果关系: ①变化原因相关的 3个节点间不存在因果关系; ②技术方面的调整对平台的存取可控性不会产生较大影响; ③平台适应性与系统兼容性之间肯定存在关联.

2.2 数据处理

由于选择了离散型贝叶斯网,这一步主要是完成数据的离散化处理. 对 N_s 中的度量节点进行等级划分,列举部分函数如表 1所示. 其中, H表示高, L表示低, M则表示中.

表 1 等级函数

度量	等级	范围
安全	H	[0.7, 1)
	L	(0, 0.7)
适应性	H	[0.5, 1)
	L	(0, 0.5)
系统兼容性	H	[0.8, 1)
	M	(0.3, 0.8)
	L	(0, 0.3]
容错性	H	[0.8, 1)
	L	(0, 0.8)
独立性	H	[0.6, 1)
	M	(0.3, 0.6)
	L	(0, 0.3]

2.3 贝叶斯学习

1) 贝叶斯结构学习. 本文选择的是由 Cooper等提出的学习算法. 该算法利用模型选择来实现结构学习,涉及到 2个关键问题:模型选择准则和搜索算法.

根据贝叶斯方法, Cooper选择具有最大后验概率的网络结构 B_S 作为最佳假设. 当后验概率最大时,认为模型"最好".根据贝叶斯公式,后验概率为

$$p(B_S \mid D) = \frac{p(D \mid B_S)\, p(B_S)}{p(D)} \qquad (1)$$

对于搜索算法,本文选择的是 K2启发式搜索算法. 其基本思路是:开始时所有节点的父节点集合都为空,利用贪婪搜索的方法,搜索对目标值产生最大影响的操作,直至没有一个操作能够增

加目标值. K2算法使用式(2)中的函数作为搜索的评价函数.

$$g(i,\pi_i) = \prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}+\alpha_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (2)$$

一系列试验结果表明,此方法直接应用的部分结果可能与已有的成熟领域知识冲突. 经过分析发现,一个重要原因是此方法假设贝叶斯网结构的先验概率相等,所以无法将一些可以由专家判断的知识,特别是与贝叶斯网结构相关的领域知识体现在结果中.

因此,上一步中的强制因果/非因果关系被引入,用来增强结果的可信度. 结构学习的结果如图4所示.

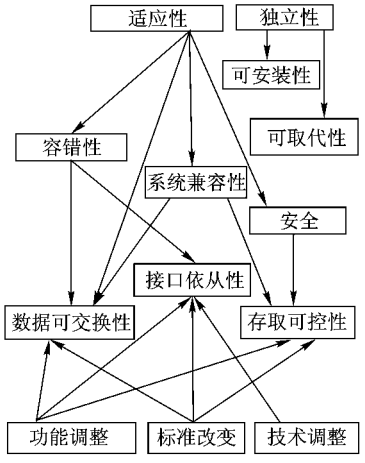

图 4　贝叶斯网结构

2) 参数学习. 在构造贝叶斯模型之后,就可以通过 Cooper和 Herskovits的方法中提供的方法来得到贝叶斯网的条件概率表,即完成参数学习.

3　应用和结果分析

在第 2节,利用贝叶斯网来建立基础软件平台的互操作性评估和预测模型. 用户可以根据第2节的结果收集节点数据,然后利用贝叶斯推理得到所需节点的值. 这时也可充分体现贝叶斯的另一个优点:在原因节点收集不完整的情况下,仍可以使用它进行评估.

有许多工具支持贝叶斯网的推理,如微软研究院的 MSBNx[14]和 KSU的 BNJ[4]. 本文选用前者.

选择中标软件 2006年 11月 18日发布的 NeoShine Linux Server 3.0来检验本文结果. 迄今为止,该产品实例被广泛应用于政府部门和商业机构.

此平台实例度量结果如下:(安全,0.87),(适应性,0.66),(系统兼容性,0.64),(容错性,0.75),(独立性,0.48),(可安装性,0.90),(可取代性,0.43).

根据度量值利用本文模型进行推理,结果如表 2.

表 2　贝叶斯推理的结果

变化原因	结果	推理值		
功能调整	数据可交换性	L	M	H
		0.69	0.12	0.19
	接口依从性	L	-	H
		0.76	-	0.24
	存取可控性	L	-	H
		0.82	-	0.18
标准改变	数据可交换性	L	M	H
		0.30	0.34	0.36
	接口依从性	L	-	H
		0.09	-	0.91
	存取可控性	-	L	H
		-	0.80	0.20
技术调整	数据可交换性	L	M	H
		0.15	0.12	0.73
	接口依从性	L	-	H
		0.70	-	0.30
	存取可控性	L	H	-
		0.77	0.23	-

根据表 2的预测结果可知,本文选用的平台实例互操作性为:①当发生功能调整时,NeoShine Linux Server 3.0的主要特性发生大变化的概率比较小;②当标准改变时,该产品的接口依从性不稳定;③技术调整可能导致数据可交换方面的问题.

根据结果还知,其它情况下,该产品的性能通常能得到保证. 这与对不同领域用户进行调查的结果相符.

4　结 束 语

软件行业出现了合理、有效的基础软件平台评估的迫切需求,应该加强相关的投入和研究,促进基础软件平台的进一步发展.

本文重点研究了基础软件平台的互操作性评估相关问题. 后续工作主要是收集企业数据,验证并改进此模型. 解决基础软件平台互操作性评估问题,可以为基础软件平台整体的质量评估起到极大参考作用.

(下转第 1160页)

表 2 点目标质量评估结果

坐标/(m,m)	方位分辨率/m	距离分辨率/m	PSLR/dB		ISLR/dB	
			方位向	距离向	方位向	距离向
0,-500	0.91	0.66	-13.04	-13.27	-10.35	-10.20
-2000,0	0.91	0.66	-13.03	-13.29	-10.34	-10.12
2000,0	0.91	0.66	-13.13	-13.30	-10.31	-10.15

注: PSLR—峰值旁瓣比; ISLR—积分旁瓣比.

4 结束语

为保证图像的方位向分辨率,通常的 ECS算法中的子孔径分割其要求重频大于点目标的多普勒带宽,而该文将 deramp处理应用于 ECS算法,降低了子孔径处理中对脉冲重复频率的要求,减轻了聚束式 SAR系统设计中对重频 PRF的限制,另外通过子孔径处理也提高了算法的效率.

参考文献 (References)

[1] Riccardo Lanari, Manlio Tesauro, Eugenio Sansosti, et al. Spotlight SAR data focusing based on a two-step processing approach [J]. IEEE Transactions on Geoscience and Remote Sensing, 2001, 39(9): 1993 - 2004

[2] 王国栋,周荫清,李春升. 高分辨率星载聚束式 SAR的 Deramp Chirp Scaling成像算法 [J]. 电子学报, 2003, 31(12): 1784 - 1789
Wang Guodong, Zhou Yinqing, Li Chunsheng. A deramp chirp scaling algorithm for high-resolution spaceborne spotlight SAR imaging[J]. Acta Electronic Sinica, 2003, 31(12): 1784 - 1789(in Chinese)

[3] Ding Zegang, Long Teng, Zeng Tao, et al. Deramp range migration processing for space borne spotlight synthetic aperture radar [J]. Advances in Space Research, 2008, 41: 1822 - 1826

[4] Moreira Alberto, Mittermayer Josef, Scheiber Rolf. Extended chirp scaling algorithm for air-and spaceborne SAR data processing in stripmap and scanSAR imaging modes[J]. IEEE Transactions on Geoscience and Remote Sensing, 1996, 34(9): 1123 - 1136

[5] Mittermayer Josef, Moreira Alberto. Spotlight SAR processing using the extended chirp scaling algrithm[C]//IEEE International Conference on Geoscience and Remote Sensing [S.l]: IEEE, 1997, 4: 2021 - 2023

[6] Sun Jinping, Wang Jun, Yuan Yunneng, et al. Extended chirp scaling algorithm for spotlight SAR[J]. Chinese Journal of Aeronautics, 2002, 15(2): 103 - 108

[7] 王国栋,周荫清,李春升. 高分辨率星载聚束式 SAR改进的 ECS成像算法 [J]. 宇航学报, 2003, 24(1): 65 - 70
Wang Guodong, Zhou Yinqing, Li Chunsheng. Refined ECS algorithm for high resolution spaceborne spotlight SAR imaging [J]. Journal of Astronautics, 2003, 24(1): 65 - 70(in Chinese)

[8] 黄岩,李春升,陈杰,等. 高分辨率聚束 SAR改进 chirp scaling成像算法 [J]. 电子学报, 2000, 28(3): 35 - 38
Huang Yan, Li Chunsheng, Chen Jie, et al. Refined chirp scaling algorithm for high resolution spaceborne SAR imaging[J]. Acta Electronic Sinica, 2000, 28(3): 35 - 38(in Chinese)

(责任编辑:娄 嘉)

(上接第 1151页)

参考文献 (References)

[1] Lu Shouqun. Discussion on interoperability of open-source software [J]. Software World, 2007, 32(6): 1 - 50

[2] Komatsoulis G A. Introduction to interoperability and compatibility [EB/OL]. National Cancer Institute Centre for Bioinformatics, 2006. https://cabig.nci.nih.gov/events/2006/2006-AnnualMeeting-Day-2/Intro-Interoperability-Compatibility-Komatsoulis.pdf

[3] PIT Ltd. PIT Philosophy [EB/OL]. 2007[2007-08-30]. http://www.qpit.ltd.uk/media/papers/Defining-Quality.pdf

[4] 高晖. 软件体系结构层次的软件适应性度量和预测研究 [D]. 北京:北京航空航天大学计算机学院, 2008
Gao Hui. Research on flexibility metrics in software architecture level[D]. Beijing: School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, 2008(in Chinese)

[5] Herskovits E. Computer-based probabilistic-network construction[D]. CA: Stanford University, 1991

[6] Dagum P, Luby M. Approximating probabilistic reasoning in Bayesian belief networks is NP-hard [J]. Artificial Intelligence, 1993, 60(1): 51 - 91

[7] Mitchell T M. Machine learning [M]. Beijing: China Machine Press, 2003: 86 - 154

[8] Guo H, Hsu H. A survey of algorithms for real-time Bayesian network inference [EB/OL]. 2002. http://citeseer.ist.psu.edu/guosurvey.html

[9] Fan C F, Yu Y C. BBN-based software project risk management [J]. Journal of Systems Software, 2004, 73: 193 - 203

[10] Stamelos I, Angelis L, Dimou P, et al. On the use of Bayesian belief networks for the prediction of software productivity[J]. Information and SoftwareTechniques, 2003, 45: 51 - 60

[11] Fenton N E, Neil M, Marsh W, et al. Predicting software defects in varying development lifecycles using Bayesian nets [J]. Information and Software Technology, 2007, 49: 32 - 43

[12] Cooper G, Herskovits E. A Bayesian method for the induction of probabilistic networks from data [J]. Machine Learning, 1992, 9: 309 - 347

[13] Microsoft Research. MSBNx [CP/DK]. 2006. http://research.microsoft.com/adapt/MSBNx

[14] KSU Probabilistic Reasoning Group. BNJ[CP/DK]. 2004. http://bnj.sourceforge.net

(责任编辑:吴 琼)