第32卷 第19期 计 算 机 工 程 2006年10月
Vol.32 № 19 Computer Engineering October 2006
·软件技术与数据库· 文章编号：1000—3428(2006)19—0071—03 文献标识码：A 中图分类号：TP311.5

基于TSP度量模型的FCA扩展应用

李 旭[1]，刘宗田[1]，强 宇[1,2]
（1. 上海大学计算机工程与科学学院，上海 200072；2. 蚌埠坦克学院计算机系，蚌埠 233013）

摘 要： TSP开发过程强调用数据说话，要求较高的精确度，这对于大多数软件企业难以达到，因此应遵循一种“适度度量”的策略。对过程数据的分析不仅可以减少度量的工作量，还可为后续的开发及过程的改进提供参考和建议。该文提出了将形式概念分析（FCA）应用于TSP度量模型中，通过基于概念格的关联规则，挖掘出了有价值的信息。通过实验项目验证了该方法的有效性和实用性。

关键词： TSP；度量；形式概念分析；内涵缩减；关联规则

Patulous Application of FCA Based on TSP Measurement Model

LI Xu[1], LIU Zongtian[1], QIANG Yu[1,2]
(1. School of Computer Engineering and Science, Shanghai University, Shanghai 200072;
2. Department of Computer, Bengbu Tank College, Bengbu 233013)

【Abstract】 The development process of TSP emphasizes that data is important and requests for full-scale metrics, but it is difficult for most of software enterprise, so it needs a strategy so-called “moderate metrics”. The analysis of data reduces workload of metrics and provides suggestions and references for the latter developments or process improvement. This paper puts forward the application of formal concept analysis to TSP metrics model, achieves goal of “moderate metrics” and gains valuable process improvement information by association rules mining based on concept lattice. Some experimental projects prove validity and practicability of application of FCA in TSP measurement model.

【Key words】 Team software process(TSP); Measurement; Formal concept analysis; Intention reduction; Association rules

当前，越来越多的组织采用TSP软件开发过程，显著提高了组织的能力成熟度以及产品的质量[1,3]。TSP强调对所有过程及产品数据的积累以及全面的精确度量。

在TSP的过程度量中，如果能够发现各个度量元之间的潜在联系，在某一度量元状态已知时能推断与其有联系的其他度量元的状态，那么不仅可以减少度量工作量，而且可以有效地提高过程质量以支持持续的过程改进[2]。

本文扩展了原有的TSP度量模型，将形式概念分析理论融入了TSP度量模型当中，不仅可以做到“适度度量”的策略，在有限的进度和成本下交付高质量的产品，而且提出了一种挖掘度量元之间潜在关系的方法，为后续的过程改进提供了有效的建议。

1 基本概念

1.1 概念格

形式概念分析通常由形式背景这一基本概念开始。在形式概念分析中，形式背景被定义为一个三元组(U,D,R)，其中U和D是集合而I是U和D间的二元关系，即I⊆U×D；U和D的元素分别被称为对象和属性。

在形式背景K中，在U的幂集和D的幂集之间可以定义两个映射f和g如下：

$\forall O1\subseteq U: f(O1)=\{ d \mid \forall x\in O1(xRd) \}$

$\forall D1\subseteq D: g(D1)=\{ x \mid \forall d\in D1(xRd) \}$

来自P(U)×P(D)的二元组(O1,D1)如果满足两个条件：O1=g(D1)和D1=f(O1)，则它被称为是形式背景K的一个形式概念。K的所有形式概念的集合被标记为CS(K)。CS(K)上最重要的结构是由亚概念-超概念关系产生的，其定义如下：如果形式概念(O1,D1)是(O2,D2)形式概念的亚概念，记为(O1,D1)≤(O2,D2)。通过这个关系，得到一个有序集CS(K)=(CS(K),≤)，这是一个完全格，被称为形式背景K的概念格[4]。

1.2 模糊概念格

模糊集合是一种特殊定义的集合，隶属度函数反映了模糊集合中的元素属于该集合的程度。根据模糊集合论，可将模糊引入形式背景，使背景从标准变为模糊，背景的每列定义为一个模糊语言表示的模糊集合。

其他定义与1.1节中概念格定义相同，基于模糊形式背景建立起的就是一个模糊概念格。

1.3 内涵缩减

概念格节点的内涵缩减用来表示保持节点外延值不变性的最小属性集。概念的内涵集体现了一种最大性，内涵缩减正是其最大性的对应物，它以最小的内涵集合（最小特征集）刻画了格节点，是对节点内涵的一种最精简表达[4]。

对于一个给定的概念$C=(O_1, D_1)$，如果属性集合D_2满足下述两个条件：

(1)$g(D_1)= g(D_2)=O_1$；

(2)对于任意的$D_3\subseteq D_2$有$g(D_3)\supseteq g(D_2)=O_1$；

则它为C的一个内涵缩减。概念C的所有内涵缩减的族集称

基金项目： 上海市高等学校科学技术发展基金资助重点项目“软件开发质量管理与控制平台研究”(02AZ86)
作者简介： 李 旭(1981－)，男，硕士，主研方向：数据挖掘，软件工程；刘宗田，教授、博导；强 宇，博士
收稿日期： 2005-12-29 **E-mail：** colorfuldays_lee@sina.com

为 C 的内涵缩减集，记为 INT_RED(C)。

2 形式概念分析在 TSP 软件开发过程中的应用

2.1 内涵缩减理论在 TSP 度量模型中的应用及算法描述

内涵缩减用来表示保持节点外延值不变性的最小属性集。概念的内涵集体现了一种最大性，内涵缩减正是其最大性的对应物，它以最小的内涵集合（最小特征集）刻画了格节点。

如果将度量元作为属性，将度量元集合的每一次考察作为一个对象，可以生成一个关于度量元的概念格。在此基础上考察频繁格节点，对该格节点进行内涵缩减操作就得到了原度量元集合的一个约减。

下面给出整个约减过程的算法描述：

(1)确定需要考察的初始度量元集合，记为 D。

(2)考察历史过程数据，对前期度量工作进行总结和评审。将对度量元集合的每一次考察作为一个对象，每个度量元作为一个属性，就生成了一张关于度量元集合的形式背景。对于（o_i,d_j）取１还是０，规定如下：对应于o_i的这次考察，利用事后判断法，如果度量元d_j的指标高，那么对这个度量元d_j进行计算是必要的，它在形式背景中的值为１，否则为０。

(3)由生成的形式背景构造度量元概念格。

(4)考查概念格中的频繁节点，需要设定频繁节点选择门限μ，若节点其外延集的数目不小于(即大于或等于)|U|*μ（|U|表示该节点所含的对象的数目），则认为它就是一个频繁节点，并记内涵为D_i。

(5)计算节点内涵的缩减集。

(6)选择内涵缩减集中含有属性个数最少的内涵缩减，记为 red。

(7)将内涵缩减，把上节点不含有的属性作为最终的约减集合。

RESULT=red∪{D-D_i}

(8)得到最终属性集合 RESULT，完成。

通过本算法计算得出的结果，在后期的度量工作当中，只需要考察结果中含有的度量元就可以了。

2.2 基于概念格的规则提取在 TSP 度量模型中的应用

挖掘度量元潜在关系不仅使度量工作有的放矢，而且由某一度量元的状态可推断其他度量元的状态，以减少工作量。另外，还可以研究影响因素之间的联系，这些都可以提高开发效率并为过程改进提供建议。

本节给出了在 TSP 编码阶段中错误类型之间的规则挖掘，以及 6 个度量元状态之间的规则挖掘，主要涉及：规模，配置改变，缺陷，风险管理，小组士气，进度。它们都是 TSP 度量模型中的几个基本度量元。

2.2.1 数据预处理

表 1 中记录了 8 个测试模块具有的各种缺陷类型。在表中将打勾的位置变为 1，其余位置变为 0，就得到了一张初始的形式背景。各属性的含义为：d_1为代码错；d_2为参数类型错；d_2为逻辑错；d_1为循环错；d_5为递归错；d_6为输出错；d_7为计算错；d_8为指针错；d_9为变量错。

对 6 个基本度量元分别进行状态划分，并将度量元每一个状态作为形式背景中的一个属性，如表 2 所示。表 3 为考察缺陷、小组士气和进度 3 种度量元间关联关系。计算得到的模糊形式背景，各属性含义见表 2，隶属度函数由用户设定。

表 1 缺陷类型形式背景

模块号	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
1	√	√					√		
2	√	√					√	√	
3	√	√	√				√	√	
4	√		√				√	√	√
5	√	√		√		√			
6	√	√	√	√		√			
7	√		√	√	√				
8	√		√	√		√			

表 2 6 种基本度量元的状态划分

度量元	状态	度量元	状态
规模	规模大d_1	引入缺陷	缺陷多d_9
	规模中d_2		缺陷少d_{10}
	规模小d_3	小组士气	士气低d_{11}
配置改变	配置改变多d_4		士气中d_{12}
	配置改变少d_5		士气高d_{13}
风险问题	风险大d_6	项目进度	未拖延d_{14}
	风险中d_7		拖延d_{15}
	风险小d_8		

表 3 模糊形式背景

对象＼属性	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
1	0.9	0.1	0.7	0.3	0.0	0.8	0.2
2	0.8	0.2	0.6	0.3	0.1	0.7	0.3
3	0.8	0.2	0.9	0.1	0.0	0.7	0.3
4	0.7	0.3	0.8	0.2	0.0	0.6	0.4
5	0.6	0.4	0.8	0.1	0.1	0.8	0.2

2.2.2 概念格生成及关联规则挖掘

SQCP平台采用改进的渐进式算法[4,5]及模糊格生成算法[7]来生成概念格，表 2 对应生成的概念格如图 1 所示。

基于概念格和模糊概念格的关联规则挖掘算法，这里不再赘述。在基于模糊概念格的规则挖掘中，最后的结果还要进行进一步的筛选。

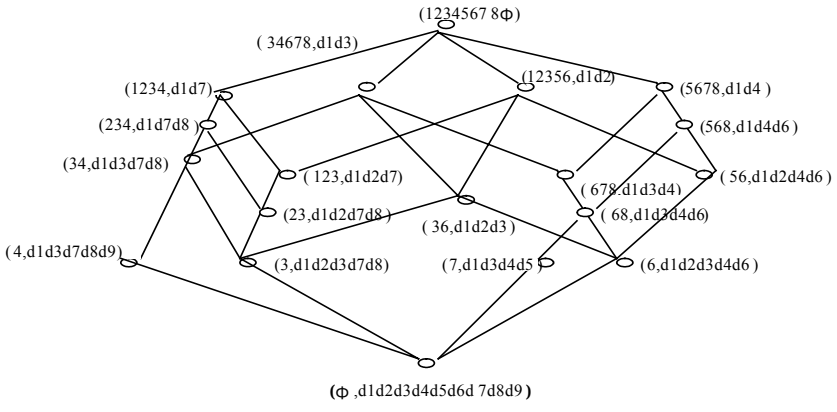

图 1 缺陷类型对应的概念格

每一条模糊规则中只能含有同一度量元的一个状态属

性。例如规则：

“规模小”∧“小组士气低”⇒“规模大”

其中，包含了属于规模因素的两个状态，这显然是一条没有意义的规则，应从结果集中清除。

2.2.3 挖掘的结果和评估

对于图 1 所示的概念格，应用规则挖掘算法[6]，并将置信度设为 70%，则可得到候选规则，如表 4 所示。由于例子中形式背景包含信息不够丰富（只含 8 条记录），因此挖掘出的规则还不能很好地反映现实中的特征。但随着过程数据的积累，挖掘结果将逐步改善。

表 4 缺陷类型挖掘结果

候选关联规则	置信度	含义
$d_1 \wedge d_4 \Rightarrow d_6$	3/4=75%	如代码错且循环错，则输出错
$d_1 \wedge d_6 \Rightarrow d_4$	3/3=100%	如代码错且输出错，则循环错
$d_4 \wedge d_6 \Rightarrow d_1$	3/3=100%	如循环错且输出错，则代码错
$d_4 \Rightarrow d_1 \wedge d_6$	3/4=75%	如循环错，则代码错且输出错
$d_6 \Rightarrow d_1 \wedge d_4$	3/3=100%	如输出错，则代码错且循环错

对于由表 3 生成的模糊概念格，应用模糊关联规则挖掘算法[8]，并将置信度设为 90%，则可得到一条候选规则：

$d_{11} \wedge d_{15} \Rightarrow d_9$ 支持度：66.7% 置信度：91.7%

含义 如果小组士气低且项目拖延，则引入缺陷多。

这样的挖掘结果可以引起问题的原因。任务量加剧了小组成员技术上存在的困难，导致了工作压力增大，这样就造成了成员的士气低落并在开发过程中不断的引入错误，对完成开发任务没有信心，继而无法保质保量地完成开发任务，致使整个项目开发进入恶性循环。

3 实验与分析

作为试验项目，挑选了 12 名学生，并把分为 A、B、C 3 个小组（每组 4 人），小组 A 采用结构化的开发方法，小组 B 和小组 C 采用 TSP 开发方法，其中小组 C 的开发平台为 SQCP（本文的度量约减和规则挖掘分别是过程管理和周期评价子系统中的两个功能）。

如表 5 所示，可以看出小组 C 的度量元数目和度量所用的工作量都比小组 B 有了显著的减少。当然，随着项目的数据的积累，度量元集合数不会经约减后无限减小，它会稳定于某个数目，但是被约减后的度量元集合将会越来越精确地刻画初始度量元集合。

表 5 度量元集合约减统计

	B 的度量元数	C 的度量元数	B 工作量（h）	C 工作量（h）	度量元约减百分比	工作量约减百分比
周期 1	24	17	9.0	6.5	29.1%	27.7%
周期 2	24	16	8.4	6.2	33.3%	34.1%
周期 3	24	16	8.6	6.6	33.3%	26.7%

图 2、图 3 分别显示了在 12 周内 3 个小组的进度曲线（完成百分比）以及产品各个组件在代码检查阶段统计的错误数，可以看出小组 C 在开发过程中，质量和进度都明显优于小组 A 和 B。

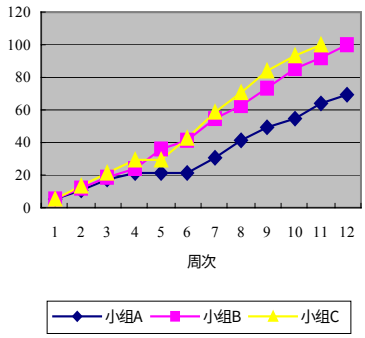

图 2 3 个小组进度曲线

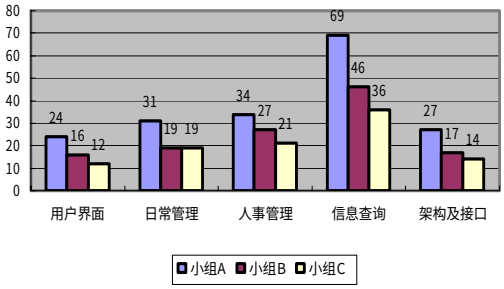

图 3 3 个小组产品的质量统计

4 结论

在TSP软件开发过程中会产生大量的分布式过程数据，对数据的分析可以得到有价值的信息，本文介绍了形式概念分析方法在TSP度量模型中的应用。当然，概念格也存在自身的问题，由于其描述的信息很完备而导致数据存储过大，在实际应用中生成的格结构可能会很复杂。尤其在对海量数据分析和处理工作时，问题将愈发突出。目前人们已就此开展研究，提出了不少有益的方法[9]。在下一步的工作中，也将针对大规模TSP过程数据处理问题进行深入研究，使形式概念分析方法更好地应用于TSP软件开发过程，以支持持续的过程改进，提高组织的软件能力成熟度。

参考文献

1 Davis N, Mullaney J. The Team Software Process in Practice: A Summary of Recent Results[EB/OL]. http://www.sei.cmu.edu/publications/documents/03.reports/03tr014.html, 2003.

2 International Function Point Users Group. 方德英译. IT度量——专家实践[M]. 北京: 清华大学出版社, 2003.

3 Hunmphrey W. Introduction to the Team Software Process[M]. Texas, USA: Addison Wesley Longman, 2000.

4 谢志鹏. 概念格及扩展模型研究[D]. 合肥: 合肥工业大学计算机学院, 2000: 12-20.

5 谢志鹏, 刘宗田. 概念格的快速渐进式构造算法[J]. 计算机学报, 2002, 25(5): 490-496.

6 谢志鹏, 刘宗田. 概念格与关联规则[J]. 计算机研究与发展, 2000,37(12): 1415-1421.

7 强 宇, 刘宗田. 模糊概念格在知识发现的应用及一种构造算法[J]. 电子学报, 2005, 33(2): 350-353.

8 强 宇, 刘宗田. 一种模糊概念格构造算法研究[J]. 计算机工程与应用, 2004, 40(4): 13-19.

9 刘宗田. 容差近似空间的广义概念格模型研究[J]. 计算机学报, 2000, 23(1): 66-70.