维普资讯 http://www.cqvip.com

软件系统层次的数据挖掘方法*)

钟 智[1] 尹云飞[2] 张师超[1,3]

(广西师范大学计算机系 桂林541004)[1] (广西师范学院数计系 南宁530001)[2]

(悉尼理工大学信息技术学院 澳大利亚 悉尼)[3]

摘 要 论文在软件数据中挖掘聚类模式的研究基础上,进一步提出了在软件层次上的数据挖掘方法。对于解决软件工程中项目代价的估算和评测具有重要的参考价值。首先收集不同类型软件数据,接着根据Halstead软件科学从它们中间抽取不同的特征,以此来标识不同的软件;然后将这些软件归为不同的类别,对于同一类中的软件可以认为它们具有相似的软件代价或相似的结构,可以用于病毒特征检测和预测,对于在不同类中的软件可以发现二者存在差异的决定"相异因素";最后给出了对5414个实际软件系统挖掘的实验结果。结果表明这种软件层次的数据挖掘方法是可行而有效的。

关键词 软件系统,数据挖掘,聚类,知识表示,偏离度

A Data Mining Method on Software System

ZHONG Zhi[1] YIN Yun-Fei[2] ZHANG Shi-Chao[1,3]

(College of Computer Science, Guangxi Normal University, Guilin 541004)[1]

(Department of Mathematics and Computer Science, Guangxi Normal College, Nanning 530001)[2]

(Faculty of Information Technology, University of Technology, Sydney, Australia)[3]

Abstract This paper brings forward a data mining method on software system on the basis of the former clustering data mining software data. This way is valuable for reference to solve the estimation of project cost in software engineering. Firstly, collect all kinds of different software data; the following thing we have done is drawing different features from these data according to the Halstead software science, which are used to identify different softwares; then, classify the data to different classes, to the softwares in the same class, we regard them as the same software cost or similar structures, which can be used to check or pretest the virus features, to the softwares in different classes, we try to find the "essential different factors" among them; finally, we give the experiment results of 5414 actual software systems. The results show that this kind of data mining based on software data is feasible and effective.

Keywords Software system, Data mining, Clustering, Knowledge representation, Deviation degree

1 引言

如何有效地评估软件产品?如何快速处理软件数据?长期以来都是软件工程中一件极其复杂而繁琐的工作。例如甲、乙两人,分别用不同风格实现了同一个功能要求的软件,那么谁的软件更好(可靠性)一些呢?严格来说,这个问题实际上很难准确回答。鉴于此,文[1]提出了软件成熟度模型作为评估软件的统一标准。该模型需要收集大量的、可信度较高的数据作为对比依据。文[2]提出在软件生存周期的早期阶段收集必要的信息,借此预测未来软件之代价。这确实是一个有效的方法,然而在软件生存周期的早期阶段,可供使用的数据极其匮乏。如何有效地提高有限数据的利用率已成为人们追求的目标。为此,有人提出了人工智能的方法、遗产算法、数据挖掘法、经济科学法等。

数据挖掘方法是近年来较热门的一种研究方法,例如文[3]研究了负关联规则的挖掘;文[4]提出了因果关联规则的挖掘方法;文[5,6]提出了将数据挖掘应用于文本分类中;文[7]构建了多媒体数据关联挖掘的模型;文[8,9]涉及了时序数据和序列数据的挖掘;所有这些方法对于软件数据的处理和软件产品代价的评估都有一定的借鉴意义。我们的研究正是在这种背景下进行的。

2 软件工程数据的处理

2.1 当代软件工程面临的困惑

软件工程在计算机发展和应用中的地位是非常重要的,它对软件产业的形成和发展起着决定性的推动作用,是人类进入信息化社会和网络互连社会的支持。

文[10]提出开发复杂的大型的软件项目遇到的很多问题,由于找不到解决的办法,致使问题积累起来,形成了日益尖锐的矛盾,这些问题归结起来有:

怎样比较软件?模块的划分标准是什么?设计程序的风格技巧是什么?怎样预测软件代价?软件开发无计划性、主观盲目地制定计划,执行起来和实际情况有很大差距,致使常常突破经费预算。软件需求不充分,使一些问题在开发后期集中暴露。软件开发过程无规范:开发过程没有统一的、公认的方法和规范指导,参加的人员各行其是。资料很不完整,忽视了每个人工作与其它人的接口部分。软件产品无评测手段:未能在测试阶段充分做好检测工作,提交用户的软件质量差。软件设

*)澳大利亚ARC基金资助项目,国家自然科学基金项目(60075016)。

维普资讯 http://www.cqvip.com

计需要的好方法、好的分析策略是什么?如何分解一个软件才能得到最佳的模块组合呢?操作所依赖的数据结构是什么?它们是如何实现的?如何减低模块之间的耦合度?用户界面应当具备什么质量特性?为什么程序的正确性不是对程序质量的唯一要求呢?怎样评估传统程序与当代程序。进行软件维护的原因是什么?在目前情况下,设计、编码、测试中的哪一方面可以改进?有些维护资源应该有但是却没有?工作中主要的或次要的障碍是什么?从维护申请的类型来看是否应当有预防性维护?哪些度量适合于过程和产品?如何使用收集到的数据?在以往的项目中,软件开发生产率如何?生产出的软件产品的质量又怎样?等等。

2.2 数据挖掘中的聚类技术

数据挖掘中的聚类技术来源于现实生活中"物以类聚,人以群分"的生活理念。1965年 L. A. Zadeh 提出模糊集合的概念后,将聚类的方法推向了一个实用的高度。在聚类中最大的问题是评价的标准,例如:经典聚类理论认为如果 A 与 B 相似度大于指定阈值,B、C 以及 A、C 的相似度也大于指定阈值,那么 A、B、C 一定属于同一类。这在某些情况下会出现问题,比如:甲与乙、乙与丙、甲与丙分别是同学,那么甲、乙、丙不一定是同学。因为当甲、乙是小学同学,乙、丙是中学同学,甲、丙是大学同学,那么显然甲、乙、丙不是同学。从集合论的观点来看,以下情况甲、乙、丙并没有共同的交集:

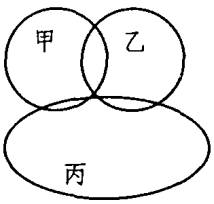

图1 甲、乙、丙无交集

也就是说,相似度计算的公式成为决定聚类质量好坏的唯一标准。这实际上又归结为特征的抽取问题,抽取什么样的特征就可以标定对象,既没有冗余而带来的繁重计算问题,又不会因为特征较少而使对象描述失真。

在数据挖掘中常用的聚类方法有以下几种:

(1)划分方法。划分方法是较常用的聚类方法。它是指在一个给定的对象或数据集中,利用一定的分类原则将其划分为 k 个子集。其中每一个子集均代表一个聚类。对每个聚类的标定一般采用两种方法,一种是 k-means 方法,该方法中每个聚类由其所有对象的均值来表示[7];另一种是 k-medoids 方法,该方法中每个聚类用相应聚类中离聚类中心最近的对象来表示[8]。

(2)模型法。其基本思想是:首先组建数据模型,然后将给定数据与该模型拟合。该方法要求首先提出假设,然后去检验假设。例如,神经网络聚类法就是通过首先将每个聚类描述成一个例证(模型),新的对象与哪个例证最相似就将它分派到相应的聚类中;在聚类结束时,每个聚类被认为是一个新特征集。

(3)密度法。它是一种基于对象间距离来进行聚类的。设定一定的密度域值,在参考点附近不断增长所获得的聚类直到"临近"密度小于指定的阈值为止,这种方法可以发现任意形状的聚类。

(4)模糊法。将模糊数学的理论引入聚类中。它的显著特点是可以软化聚类边界,使获得的聚类更加切合实际。它分为以下几步来进行:首先,根据实际情况,按某种标准对指定集合中的两两对象之间进行赋值,其值取0~1之间,即所谓的相似系数。它的大小表示对应两个对象之间彼此接近或相似的程度。然后,对得到的相似系数组成相似矩阵。该矩阵具有自反性、对称性,但未必满足传递性,这就需要我们进行"求传递闭包运算"。以此得到模糊等价矩阵。接着用水平截集来进行分类,记下不同的相似系数所关联的对象,将它们作为一个聚类。最后,输出聚类结果,供进一步抽取、挖掘使用。

2.3 软件系统层次数据挖掘理论的可行性

2.3.1 *系统简要描述* 假若一个工程项目 W 含有 n 个工程文件 $p_1, p_2, \cdots, p_n$,每个工程文件 p_i 含有 t_i 个不同的模块:$m_{i1}, m_{i2}, \cdots, m_{it_i}$,则自底向上构建项目工程。

我们对项目的评估是从基本的最简单的单位——模块入手。在这里,若将每个模块视为一个项集(ItemSet),那么不同的项集之间必然存在某些关联和内在规律。发现这些模式和知识,这就是数据挖掘的任务。

2.3.2 *系统的处理流程* 首先针对某个工程文件,正确标度模块。标度的有效性取决于抽取的特征,根据 Halstead 软件科学和"部分重复性理论"。在既不导致冗余又不致使信息失真的情况下,我们选择了一个模块所包含的代码行数、不同操作数、不同操作符、操作数总数、操作符总数组成五维空间来标度一个模块对象。然后考虑工程文件内的模块相似性,并作出聚类、简化。接着就考虑不同工程文件之间的聚类。最后,得出整个工程项目的评测模型或代价模型。

2.3.3 *系统应用背景* 众所周知,在软件生产实践中,任务(模块)的分配是一个让项目管理人员头痛的事,因为它直接关系到软件的开发成本和开发进度问题。一个完备的任务分配不仅可节省人力、物力资源,还可以提高项目研发组的开发效率,增进团结协作,充分发挥每一个工作人员的潜在能力;有效协助人力资源部门聘任所需人才,最大限度抑制人员的流失对整个项目进度的影响,也就是将这种波动控制在局部调整内,从而保证整个软件按时完工。相反,一个无计划、无依据的任务模块分配,必然导致无谓的时间浪费和人员冗余,对软件项目带来的损失是不可想象的。

"以文档为中心"的开发理念早已灌注于每一个研发者心中,相似的模块进行归类合并意味着:服务于其上的软件人员、物资的合并、组合。但是哪些人员应该组合、哪些资源应该统一调配呢?数据挖掘确实可以根据历史数据做出关联分析,来帮助项目管理人员解决这个问题。

挖掘软件数据可以对项目的预测和软件质量的度量做出极具意义的一步。通过模块的合并和分化使项目维持一个基本固定的动态规模,使得项目成本的预算和项目时间的预测成为可能。它提供的五维空间定量化的方法,对软件质量的价值进行估计,可以得到了软件质量的比较精确的估计值,与传统的人工方法比较,可控性、可管理性、有效性都有较大的突破。

3 软件系统层次数据挖掘的概念及方法

3.1 基本概念及步骤

定义1(中心知识) 中心知识是形如:

$\{\vec{S}_1, \vec{S}_2, \cdots, \vec{S}_n\}$

有限集合的一个聚类结果。其中,$\vec{S}_1, \vec{S}_2, \cdots, \vec{S}_n$ 是 n 个五维向量。

维普资讯 http://www.cqvip.com

定义2(知识的正确度) 知识的正确度是一个0～1之间的小数,用于衡量不同分类的可信程度。

定义3(知识的浮动域) 知识的浮动域是指一组对象(包括向量)所有可能的分类结果。

定义4(偏离度) 偏离度是软件结构优化程度的标度,其计算公式为:

$$\sigma=|N-(\eta_1\log_2^{\eta_1}+\eta_2\log_2^{\eta_2})|$$

其中,N 表示程序的代码行数,η_1 表示程序中不同的操作符数,η_2 表示程序中不同的运算对象数。

在软件工程中,不同的程序设计风格和技巧决定了软件的代价和生存能力,盲目主观地进行开发,必然导致开发出来的软件和实际情况有较大的差距。既然软件工程能够反映自然规律和人类社会的特性,决定了它具有极大的复杂性和不确定性。因此面对大量的软件数据,我们如何寻找潜在的规律和知识,成为研究的焦点。

首先,是数据的提取。它的基本原则是在保证不使客观对象失去真实性的前提下,选取尽可能少的特征。信息冗余、重复是数据提取必须回避的一个问题。特征选取过少影响对事物的刻画,选取过多又会造成信息冗余,处理起来耗时费力,得到的结果意义也不大。在此,我们选择每个程序所包含的代码行数 N,不同运算符个数 η_1,不同运算对象个数 η_2,运算符总数 $N1$,运算对象总数 $N2$这五个数据来标度每个程序。

其次是数据的正规化。其理论依据是模糊聚类法要求相似系数在[0,1]区间内。在这里我们采用了极值标准化公式来进行正规化:$r=\frac{x-x_{min}}{x_{max}-x_{min}}$,其中 x 为某类数据(向量的一个分量)的一个赋值,x_{min}为该类数据的最小值,x_{max}为该类数据的最大值。利用该公式分别对代码行数、不同运算符的个数、不同运算对象的个数、运算符总数、运算对象总数进行处理。

最后,抽样生成模型。面对庞杂繁多的软件数据,对其全集进行处理显然不太现实。因为:(1)随着软件工程工作的进行,会产生大量的软件数据。就是说,软件数据的集合是在不停地增长着的。(2)抽样技术能有效地减少异常数据对组建模型的影响。由于软件数据中不可避免地存在着噪声、不完整甚至是不一致的数据。而预处理只能除去表层上的噪声数据,对于那些深深隐藏的数量较少的"假数据"无法识别。这就需要我们通过多次抽样技术来减少这些数据对建模的影响。(3)虽然抽样生成的"部分重复性模型"是以效率换取准确性,但是我们可以通过提高抽样的正确度阈值来缓解这种偏差。

3.2 软件系统层次数据挖掘的方法

3.2.1 *生成"中心知识"的方法* 第一步:获取软件数据集。在软件数据集中每条记录占一行,每条记录包括五个字段,它们分别代表程序的代码行数、操作符数、运算对象数、总的操作符数、总的运算对象数。如果我们将每条记录视为一个向量,那么它就有5个分量,或者说它是一个5维向量。所以每个程序就能用一个5维向量来刻画和描述了,程序之间的关系也就可以用向量之间的定量关系来标定了。

第二步:抽样。由于软件数据集相当庞大,为了提高处理效率我们采取了抽样技术,从软件数据集中随机抽出若干数据组成一个抽样集。

第三步:正规化。对抽样集中的数据进行正规化处理。在这里我们采用了极值标准化公式来进行正规化:$r=\frac{x-x_{min}}{x_{max}-x_{min}}$,其中 x 为某类数据(向量的一个分量)的一个赋值,x_{min}为该类数据的最小值,x_{max}为该类数据的最大值。利用该公式分别对抽样集中的代码行数、不同运算符的个数、不同运算对象的个数、运算符总数、运算对象总数进行处理。

第四步:求模糊关系。假若有两个向量 i,j 则其模糊关系为:$r_{i,j}=1-\sqrt{\frac{1}{m}\sum_{k=1}^{m}(x_{jk}-x_{ik})^2}$,$m$ 表示两个向量的维数,x_{jk}表示向量 j 的第 k 个分量的正规化值,同理 x_{ik}表示向量 i 的第 k 个分量的正规化值。

第五步:产生"中心知识"。将 n 个向量的模糊关系组成一个 $n\times n$ 的相似关系矩阵,显然,这是一个对称方阵。然后用 λ 水平截集去对所有向量进行分类。分在同一类中的向量所代表的程序具有相同的软件代价和复杂性。可以选择结构清晰易懂的程序作为该类的代表,或者计算出该类的中心来标度该类程序。

3.2.2 *"知识浮动域"的生成方法* 第一步:计算偏离度。对于抽样集中的所有程序计算它们的偏离度。偏离度计算公式为:$\sigma=|N-(\eta_1\log_2^{\eta_1}+\eta_2\log_2^{\eta_2})|$,其中 N 表示该程序的代码行数,η_1表示不同操作符数,η_2表示不同运算对象数。偏离度衡量了一个程序结构优化的程度。

第二步:求相似关系。假若两个向量 i,j 的偏离度分别为 σ_i、σ_j,则其相似关系为:

$r_{i,j}=w_{ij}(1-\sqrt{\frac{1}{m}\sum_{k=1}^{m}(x_{jk}-x_{ik})^2})$,其中 $w_{ij}=1+Q\frac{(\sigma_i-\sigma_j)}{\sqrt{\sigma_i^2+\sigma_j^2-\sigma_i\sigma_j}}$,$Q$ 是一个0～1之间的小数,作为衡量 σ_i 与 σ_j 关系的经验系数,$-1\leqslant\frac{(\sigma_i-\sigma_j)}{\sqrt{\sigma_i^2+\sigma_j^2-\sigma_i\sigma_j}}\leqslant 1$(推导略)。对于抽样集中的所有向量,求两两之间的相似关系并将其组织成 $n\times n$ 的对称方阵。

第三步:产生浮动域。当给定 Q 一个值时会产生一个 $n\times n$ 的相似矩阵。求出其 λ 截集的分类集合;当 Q 取不同的值时会产生不同的分类集合。

3.2.3 *"知识正确度"的求法* 知识的正确度C,或者说每种分类结果的可信度,可以按照以下公式求:

$$C=\frac{|f(\vec{S})\cap E(\vec{S})|}{|f(\vec{S})|}\geqslant min_correctness$$

其中,$f(\vec{S})$表示"中心知识"对应的分类,$E(\vec{S})$表示浮动域对应的分类,$f(\vec{S})\cap E(\vec{S})$表示"中心知识"与"浮动域"的交集;$min_correctness$ 为最小正确度域值。

因此,给定不同的 Q、$min_correctness$ 将会形成对应的浮动域和正确度因子,而且能够计算任何一个不在浮动域中的分类的可能性(正确度)。

4 实验与评估

4.1 实验步骤

算法的实现采用 Microsoft 公司的 Visual C++6.0来完成,可以对 ANSI C 及 ANSI C++程序进行读取分析。我们所用的测试数据来源于 Borland、Microsoft 等公司的商用软件包。

整个实验分为五步:

(一)预处理:主要是填补遗漏数据、消除异常数据、平滑噪声数据以纠正不一致的数据;也就是说,删除"代码行数"为0的数据、删除"操作符数"或"运算对象数"为0的数据,一些空白的字段填上该字段所有取值的平均值;

(二)抽样:从软件数据集中随机抽取50～1000条数据,组成抽样集;

维普资讯 http://www.cqvip.com

(三)挖掘"中心知识":即产生一个基于距离的分类结果;

(四)产生"浮动域"和"正确度因子":给定一组 Q 和 *min_correctness* 将会产生对应的"浮动域"和"正确度"因子;

(五)重复(二)~(四)步。

4.2 实验结果

在此,我们列出了进行上述实验的部分结果,表1、表2、表3分别显示了五次不同抽样的结果。

表1 第一次随机抽样结果 ($n=201, \lambda=0.8$)

经验系数 Q	0.6	0.8	0.9	0.98
中心知识 $K(X)$	6类	6类	6类	6类
浮动域 $E(X)$	6类	6类	6类	6类
最小正确度 *min_correctness*	0.5	0.5	0.5	0.5
实际正确度 C	(0.60, 0.65,1,1, 1,0.5)	(0.60, 0.68,0.93, 1,1,0.5)	(0.60, 0.68,0.93, 1,1,0.5)	(0.60, 0.68,0.93, 1,1,0.5)

表2 第二次随机抽样结果 ($n=158, \lambda=0.8$)

经验系数 Q	0.6	0.8	0.9	0.98
中心知识 $K(X)$	6类	6类	6类	6类
浮动域 $E(X)$	4类	4类	4类	4类
最小正确度 *min_correctness*	0.66	0.66	0.66	0.66
实际正确度 C	(0.87, 0.87,0.67, 1,1,1)	(0.87, 0.87,0.67, 1,1,1)	(0.87, 0.87,0.67, 1,1,1)	(0.87, 0.87,0.67, 1,1,1)

表3 第三次随机抽样结果 ($n=99, \lambda=0.8$)

经验系数 Q	0.6	0.8	0.9	0.98
中心知识 $K(X)$	4类	4类	4类	4类
浮动域 $E(X)$	7类	7类	7类	7类
最小正确度 *min_correctness*	0.6	0.6	0.6	0.6
实际正确度 C	(0.61,1, 0.85,1)	(0.61,1, 0.85,1)	(0.61,1, 0.85,1)	(0.61,1, 0.85,1)

说明:n 表示抽样的样本个数;λ 表示聚类的 λ 截集水平,λ 取不同值时会对整个分类以及"部分重复性模型"产生非常大的影响,但是,在这里我们仅讨论 λ 取定值(0.8)时"部分重复性模型"的变化情况。

结论 文章讨论了一种软件层次的数据挖掘方法:首先收集不同的软件系统,然后从中抽取特征来标示它们,最后用聚类的方法将它们归类。目前,对软件数据进行处理是软件工程学目前研究的热点和难点,本文试图将数据挖掘的技术和方法应用于软件工程,并从中寻找有用的知识。这一探索对于全面、深入进行软件数据挖掘的研究提供了有益的思路和可借鉴的方法。大量仿真实例表明,这种软件层次的数据挖掘方法对于软件代价预测、复杂度评估是非常有效的。

参考文献

1 Paulk M C. Capability Maturity Model Version1.1[J]. IEEE Software,10 4,July 1993

2 Boehm B. Software Engineering Economics[J]. Englewood Cliffs, NJ,Prentice-Hall,1981

3 Zhang C Q,Zhang S C. Association Rule Mining Models and Algorithms[M]. Springer-Verlag,Berlin Heidelberg,2002

4 Zhang S C,Zhang C Q. Discovering Causality in Large Databases [J]. Applied Artificial Intelligence,2002

5 Feldman R,Hirsh R. Finding Associations in Collections of text. Machine Learning and Data Mining: Methods and Applications [M]. John Wiley Sons,1998.223~240

6 Salton G. Automatic Text Processing[M]. Addison-Wesley,1989

7 Za O R,Li Z N,Chiang J Y,Chee S. Multimedia-miner:A System Prototype for Multimedia Date Mining[C]. ACM-SIGMOD Conf. On Management of Data,1998.581~583

8 Han J,Dong G,Yin Y. Efficient Mining of Partial Periodic Patterns in Time Series Database[C]. In:proc. ACM-SIGMOD Int. Conf. Management of Data,1997.553~556

9 Bettini C,Wang X S,Jajodia S. Mining Temporal Relationships with Multiple Granularities in Time Sequences[J]. Data Engineering Bulletin,1998,21:32~38

10 Martin T,McClure C. Software Maintenance:The Problem and Its Solutions[M]. Prentice-Hall,Inc.,1993

(上接第181页)

识处理领域的研究尚处于起步阶段。

本文的主要工作是在考察直觉模糊集基本运算的基础上,引入两个典型的作用于直觉模糊集的时态逻辑算子,研究了直觉模糊集上的若干扩展运算及其性质,将这些性质归结为一组定理,并给出详细的证明过程,从而进一步夯实直觉模糊时态逻辑算子的理论基础,深化了直觉模糊集理论在知识处理领域中的应用研究。

参考文献

1 Zadeh L A. Fuzzy sets [J]. Information and Control,1965,8 (3): 338~353

2 Atanassov K. Intuitionistic fuzzy sets [J]. Fuzzy Sets and Systems,1986,20 (1) : 87~96

3 Atanassov K. More on intuitionistic fuzzy sets [J]. Fuzzy Sets and Systems,1989,33 (1) : 37~46

4 Atanassov K. New operations defined over the intuitionistic fuzzy sets [J]. Fuzzy Sets and Systems,1994,61 (1) : 137~142

5 Atanassov Krassimir T,Janusz K,Eulalia S,et al. On Separability of Intuitionistic Fuzzy Sets [J]. Lecture Notes in Artificial Intelligence,2003,2715 : 285~292

6 Vaughan P. Chu spaces as a semantic bridge between linear logic and mathematics [J]. Theoretical Computer Science, 2003, 294 (3) : 439~471

7 雷英杰,王宝树. 直觉模糊逻辑的语义算子研究[J]. 计算机科学,2004:31(11):4~6

8 刘新. 直觉模糊时态逻辑算子及其性质[J]. 吉林师范大学学报(自然科学版),2003,24 (2) : 37~38,54

9 雷英杰,王宝树. 拓展模糊集之间的若干等价变换[J]. 系统工程与电子技术,2004,26(11):1414~1418