

基于类推和灰色模型的软件阶段成本预测

王 勇¹ 李 逸¹ 王丽丽¹ 朱晓燕²

(中国海洋大学信息科学与工程学院 青岛 266100)¹ (西安交通大学电子与信息工程学院 西安 710049)²

摘 要 准确预测软件成本是软件工程领域最具挑战性的任务之一。软件开发固有的不确定性和风险性,使得仅仅在项目早期预测总成本是不够的,还需要在开发过程中持续预测各个阶段的成本,并根据变化趋势重新分配资源,以确保项目在规定的的时间和预算内完成。由此,提出一种基于类推和灰色模型的软件阶段成本预测方法——AGSE (Analogy & Grey Model Based Software Stage Effort Estimation)。该杂交方法通过合并两种方法的预测值得到最终的预测结果,避免了单独使用其中一种方法预测时存在的局限性。在真实的软件项目数据集上的实验结果表明,AGSE 的预测精度优于类推方法、GM(1,1)模型、GV 方法、卡尔曼滤波和线性回归,显示出较大的潜力。

关键词 类推,灰色模型,阶段成本预测,软件项目管理

中图法分类号 TP311 文献标识码 A

Software Stage Effort Prediction Based on Analogy and Grey Model

WANG Yong¹ LI Yi¹ WANG Li-li¹ ZHU Xiao-yan²

(School of Information Science and Engineering, Ocean University of China, Qingdao 266100, China)¹

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)²

Abstract Accurate software effort prediction is one of the most challenging tasks in the software engineering domain. Due to the inherent uncertainty and risk of software development process, it is insufficient to predict the whole effort just at the early stage of the project. In contrast, it is important to predict the effort of each stage during the software development process. This enables the managers to reallocate resources according to the variation of the project development and ensures the project to be completed with the prescribed schedule and under the budget. Therefore, this paper presented a new method for software physical time stage-effort prediction based on both analogy method and grey model. The proposed hybrid method obtains prediction results by combining the values predicted by both analogy and grey model. At the same time, this method can avoid the limitations of using either of them. The experimental results on real world software engineering dataset indicate that the prediction accuracy obtained by the proposed method is better than that obtained by analogy method, GM (1,1) model, GV, Kalman filter and linear regression, showing great potential.

Keywords Analogy, Grey model, Stage effort prediction, Software project management

1 引言

软件成本预测是软件项目管理的重要组成部分。在软件开发过程中,由于缺乏有效的成本预测方法而导致进度延期、预算超支或开发失败一直是一个严重的问题^[1-2]。Standish Group 在 2015 年公布的 CHAOS 报告中指出,来自全球各领域的 50000 个项目中,有多达 52% 的项目延期交付或预算超支^[3]。因此,采取合适的方法解决软件开发超期、超预算等问题势在必行。准确的软件成本预测是解决项目失控的重要措施,不仅有助于管理者控制开发进度,还有助于保障项目的顺利完成,提高产品质量。

软件成本预测是指对软件开发过程中需要耗费的成本进行估算^[1,4],包括人力、设备等无形和有形的成本。其中,人

力成本是核心部分,占比最大。软件工作量是指开发软件所需的人力,通常以人小时、人天等单位。人力成本的大小主要取决于软件工作量^[5]。在很多文献中,“软件成本”和“软件工作量”这两个术语可以交替使用^[6-7]。本文预测的重点是软件工作量,以人小时为工作量计算的基本单位。

目前,软件成本预测的研究主要集中在总成本预测方面^[1,8-15]。总成本预测是指在项目早期对软件开发的全部成本做出预测,以便制定合理的项目计划,并为管理者的投资决策等提供强有力的证据。由于软件开发是逐步推进的过程,因此其成本也是持续演化的。成本预测需要根据获得的新信息实时调整预测值,以便重新分配资源。然而,最初的总成本预测因预测值不变,无法应对开发中的变化。在软件开发过程中,尤其是中后期阶段,项目的实际成本和最初的预测值往

本文受国家自然科学基金(61170312, 61402355, 61502378), 软件工程国家重点实验室(SKLSE2012-09-14), 中央高校基本科研基金(XJJ2014050)资助。

王 勇(1971—),男,博士,副教授,CCF 会员,主要研究方向为软件工程经济学、数据挖掘与软件工程, E-mail: wangyong@ouc.edu.cn (通信作者);李 逸(1993—),男,硕士生,主要研究方向为软件工程经济学;王丽丽(1993—),女,硕士生,主要研究方向为软件工程经济学;朱晓燕(1982—),女,博士,讲师,主要研究方向为经验软件工程、数据挖掘。

往有很大不同,管理者无法一直参照早期的总成本调整开发进度^[16],这迫使我们需要在项目进程中逐步、分阶段地预测软件成本,及时更新项目进展,重新分配资源,从而确保项目在规定的的时间和预算内完成。阶段成本预测是指在软件开发过程中不断预测下一个阶段的成本。阶段通常是指软件生命周期的各个阶段,如需求分析、设计、编码、测试等^[17]。显然,不同项目间生命周期阶段的名称相同,但含义并不完全相同;并且阶段间没有清晰的界限,彼此重叠,基于某一项目发现的阶段变化规律不一定适用于其他项目。事实上,软件开发通常是按物理时间如周、月或季记录工作量。物理时间阶段是统一的,企业管理者很容易根据它们计划和控制项目进展^[2]。由于本研究中所使用的原始数据集中的项目以月作为工作量记录的基本单位,因此本文的研究是以月为单位的软件阶段成本预测问题,一个月即为一个阶段。也就是说,对一个待开发项目,我们研究的目标是,根据已知信息和在开发中获得的新信息,预测项目第一个月,第二个月,……,第 i 个月,……,直至最后一个月月的成本。

软件开发是动态变化、不断更新的活动,存在着诸多无法预知的因素影响着软件成本,这些因素导致成本存在较大的不确定性。同时,随着开发方法和技术等的更新换代,软件开发周期缩短,导致项目阶段数减少,因此阶段成本预测必须考虑数据缺乏和不确定因素的影响。邓聚龙教授于 1982 年首次提出灰色系统理论^[18],重点解决“小样本”不确定性系统的预测问题。通过对自身数据的研究发现事物发展的内在关系,实现对其发展规律的预测。该理论只需少量数据就可以建立用于预测的灰色模型,解决了传统预测方法^[10,19-20]在数据缺乏和不确定环境下难以准确建模和预测的问题,已广泛应用于图像处理^[21]、移动通讯^[22]、能源管理^[23]等多个领域。更为重要的是,灰色系统理论已经在软件总成本预测方面取得成功^[11-13]。但是在预测阶段的成本时,由于前几个阶段的数据需要用于建模,灰色模型无法预测最早期的几个阶段的成本。对于软件开发来说,前期阶段的成本非常重要,若不能预测则会给管理者带来诸多不便。因此,单独使用灰色模型预测软件阶段成本存在局限性。

软件开发有其普遍规律,相似项目的成本变化可能存在类似的规律。因此,可以考虑利用从历史项目中归纳得到的规律预测新项目的成本。类推是一种利用历史数据解决问题的方法,已由 Shepperd 等归纳为著名的类推预测^[14]。在软件成本预测过程中,该方法从历史项目中检索相似项目,进而推导出新项目的预测值,已成功应用于总成本预测^[14-15,24-25]。由此,我们考虑将类推方法用于软件阶段成本预测中。不过,基于阶段成本的类推预测有其特殊性。我们有理由相信,如果项目相似,不仅项目属性应相似,由前期阶段成本构成的序列也应相似。此外,应用类推方法时存在一定的限制,即如果不存在历史项目,类推预测就无法进行;即便存在历史项目,如果其与新项目的相似度不高,即使选择其中最相似的项目进行类推,也会导致较大的预测误差。

综上所述,类推方法和灰色模型各有优势,均可用于成本预测,但单独使用都存在一定的不足。因此,我们尝试将两种方法合并,避免各自的局限性,即灰色模型不能预测前期阶段成本的问题,可用类推寻找相似项目来解决;类推方法因缺失历史项目而无法进行预测时,可使用灰色模型对项目自身的

数据建模实现预测。由此,本文提出基于类推和灰色模型相结合的阶段(月)成本预测方法——AGSE。新方法的预测值包含两部分:1)基于相似项目得到的“类推预测值”;2)基于前期阶段成本得到的“灰色预测值”。这两部分按合成系数合并,从而得到最终的预测值。我们在真实的软件项目上测试 AGSE 的性能,实验结果表明,新方法有优良的预测性能,预测精度优于类推方法、GM(1,1)、GV 方法、卡尔曼滤波和线性回归。

本文第 2 节介绍相关工作;第 3 节介绍类推和灰色模型的基本概念;第 4 节详细解释 AGSE 的实现过程;最后总结全文。

2 相关工作

虽然已有大量关于软件成本预测的研究^[1,8,17,24],但是阶段成本预测问题却极少引起研究者的关注。其中,只有王勇等^[26]的研究使用了物理时间单位,其余均是针对软件生命周期阶段^[27-32]。MacDonell 等^[28]利用生命周期中相关性较高的阶段建立线性回归模型,使用前期阶段预测后续阶段的成本,例如根据需求阶段预测设计阶段。结果表明,线性回归与专家方法相结合可以提高预测精度。Kulkarni 等^[29]为每个生命周期阶段建立转换矩阵,这个矩阵可以将某一阶段的输入转换为输出,同时作为下一个阶段的输入。通过将不同阶段的矩阵串联成一个整体的模型,只需提供第一个阶段的输入,便可确定最后阶段的输出。该实验数据只是用于一个军方项目的演示,没有提及预测精度。Ohlsson 等^[30]的研究是使用前期阶段结束后获得的新信息改进原有预测值,例如用设计阶段中流程图的个数作为模型的输入,得到新的成本预测值。但是由于模型的输入与成本间的相关性不显著,该方法并没有提高预测精度。Tsunoda 等^[31]利用生命周期中早期阶段的成本作为回归模型的解释变量,来预测软件总成本。结果表明,该方法可以提高预测的准确度。Ferrucci 等^[32]使用线性回归建立多个预测模型,分别预测软件生命周期阶段的成本。实验结果表明,基于前期(生命周期)阶段成本的模型能比基于功能点的模型更准确地预测后续阶段的成本。

利用类推方法预测软件成本的研究已有很多。Shepperd 等^[14]提出模型 ANGEL,利用属性寻找相似项目,类推得到新项目的总成本预测值,结果显示类推优于逐步回归方法。Azzeh^[33]将人工蜂群算法与类推方法相结合来预测软件总成本,明显改善了预测精度,这说明类推可与其他方法结合来共同预测软件成本。但是,据我们所知,类推方法的研究主要集中在总成本预测^[14,33]和方法的改进方面^[34],还没有关于阶段成本预测的研究。

宋擒豹等^[11]首次将灰色系统理论用于软件成本预测。他们使用灰色关联分析构建新模型 GRACE,对软件总成本做出预测。王勇等^[26]首次提出按月划分软件的开发周期,使用灰色系统理论中的 GM(1,1)和 Verhulst 建模预测后续月份的工作量,实验结果优于线性回归和卡尔曼滤波。

与以上研究相比,本文所提方法——AGSE 具有以下差异和特征。

(1)AGSE 采用物理时间——月作为阶段单位预测软件成本,与文献^[26]中的研究一致,不同于文献^[27-32]中的生命周期阶段;

(2) AGSE 将类推方法用于软件阶段成本预测,不同于文献[14,33]中的总成本预测;

(3) AGSE 中使用的灰色模型和文献[11]中的灰关联分析属于两种不同的方法;

(4) 文献[26]中的研究使用了两种灰色模型,即 GM(1,1) 和 Verhulst,但是本文首次将灰色模型和类推两种明显不同的方法结合,避免了只使用灰色模型存在的局限。

从上面的介绍可以看出,阶段成本预测的相关研究较少,更没有将类推和灰色模型相结合进行此类预测的研究。总之,不同于先前的研究,文中提出的 AGSE 方法是解决软件阶段成本预测问题的新方法,有潜力作为一种可选方法。

3 基于类推和灰色模型的阶段成本预测

AGSE 结合类推和灰色模型进行阶段成本预测。本节将详细介绍类推方法和灰色模型的原理以及 AGSE 方法的实现过程。

3.1 类推方法

类推是基于案例推理的一种表现形式^[24],在软件工程领域,类推方法因具有杰出的预测性能而被广泛使用^[33]。类推预测的关键思想是:识别新项目;在历史数据中寻找相似项目;重复使用从相似项目中获得的知识来解决问题。这种预测方法会带来两个需要着重解决的问题。

(1) 选取合适的属性描述项目。类推需要属性对比寻找相似项目,因此选取合适的属性描述项目至关重要。项目属性既有数值值,如开发团队的规模(常用人数表示);也有分类值,如项目类型(新开发项目、维护型项目等)。在选取属性时应选择客观真实的属性来准确地描述项目,如开发方法、行业类型和应用领域等。在 Shepperd^[14]的研究中,选用了项目类型、功能点个数、不同实体的引用数量等具有真实值的项目属性。

(2) 评价相似性。历史项目的相似度可以帮助我们选择合适的项目进行类推预测。评价相似性的方法有很多^[35],其中最典型的是最近邻居法(Nearest Neighbor Algorithms)^[36]。项目的 n 个属性组成一个 n 维空间向量,相似性定义为向量间的距离,距离越小则相似度越高。计算已完成项目与新项目间相似度的方法如式(1)所示:

$$Sim(C_1, C_2, P) = \frac{1}{\sqrt{\sum_{j=1}^n FD(C_{1j}, C_{2j})}} \quad (1)$$

其中, $Sim(C_1, C_2, P)$ 是相似度, C_1 是新项目, C_2 是已完成项目, P 是属性的集合, n 是属性的个数, $C_{1j}, C_{2j} \in P$, $FD(C_{1j}, C_{2j})$ 是 C_1 与 C_2 间属性的差异,可由式(2)计算得到:

$$FD(C_{1j}, C_{2j}) = \begin{cases} (C_{1j} - C_{2j})^2, & \text{数值值} \\ 0, & \text{分类值且 } C_{1j} = C_{2j} \\ 1, & \text{分类值且 } C_{1j} \neq C_{2j} \end{cases} \quad (2)$$

在实际应用中,需要在式(1)的分母中加上一个很小的值,以避免出现分母为零的情况。

当对新项目的工作量进行类推预测时,可由式(1)找出与新项目相似的 N 个历史项目,然后将相似度高的前 k 个项目的工作量合成为新项目工作量的预测值。其中,项目个数 k 可用机器学习方法学习得到, k 一般取 1, 2, 3 等^[10],表示用 1

个、2 个或 3 个相似项目的工作量计算得到预测值。同时,在合成预测值时也可对每个相似项目的工作量分配一个权重 w 。

3.2 GM(1,1)模型

在大多数情况下,原始成本序列的离散性很大,难以用函数拟合。但是,在对原始序列做累加生成后,生成的序列会显示出明显的准指数规律,可以利用指数曲线拟合,进而预测其发展情况。灰色系统理论包括多种灰色模型^[37],其中 GM(1,1)模型的应用较为广泛,主要用于预测准指数序列。但是,若原始序列的离散性过大,累加生成后依然无法拟合,则需要对其进行变换处理,以减小离散性。本节主要介绍 GM(1,1)模型建模的预测过程、光滑性检验和平移变换等。

3.2.1 GM(1,1)模型的预测过程

假设 $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$ 是非负初始序列,使用 GM(1,1)模型对其进行建模的步骤如下。

第 1 步 对 $X^{(0)}$ 作一阶累加生成(1-AGO),得到累加序列:

$$X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)) \quad (3)$$

其中:

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), k=1, 2, \dots, n \quad (4)$$

第 2 步 对 $X^{(1)}$ 作紧邻均值生成,得到序列:

$$Z^{(1)} = (z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(n)) \quad (5)$$

其中:

$$z^{(1)}(k) = \frac{1}{2}(x^{(1)}(k) + x^{(1)}(k-1)), k=2, 3, \dots, n \quad (6)$$

第 3 步 根据 $x^{(0)}(k)$ 和 $z^{(1)}(k)$ 建立灰色微分方程,该方程即为 GM(1,1)模型的基本形式,如式(7)所示:

$$x^{(0)}(k) + ax^{(1)}(k) = b \quad (7)$$

其中,参数 a 和 b 的值可以运用最小二乘法估计确定:

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T Y \quad (8)$$

其中, B 和 Y 分别为:

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \dots & \dots \\ -z^{(1)}(n) & 1 \end{bmatrix}, Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \dots \\ x^{(0)}(n) \end{bmatrix} \quad (9)$$

第 4 步 将式(8)、式(9)代入式(7),可得式(7)的解为:

$$\hat{x}^{(0)}(k) = (1 - e^a)(x^{(0)}(1) - \frac{b}{a})e^{-a(k-1)}, k=2, 3, \dots, n \quad (10)$$

第 5 步 令 $k=n+1$,由式(10)可得序列 $X^{(0)}$ 第 $n+1$ 个点的预测值 $\hat{x}^{(0)}(n+1)$ 。

3.2.2 光滑性检验与平移变换

如果原始序列的波动非常剧烈,使用 GM(1,1)直接建模将导致较大的预测误差。因此,在建模前需要检验原始序列是否符合光滑性条件,确定序列是否适合建模,如不符合则需要对其进行变换处理。

(1) 光滑性检验

灰色序列的光滑性定义如下^[33]:

设原始序列 $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$,

$x^{(0)}(k) \geq 0, k=1, 2, \dots, n$, 称:

$$\rho(k) = \frac{x^{(0)}(k)}{\sum_{i=1}^{k-1} x^{(0)}(i)}, k=2, 3, \dots, n \quad (11)$$

为序列 $X^{(0)}$ 的光滑比。

光滑比使用序列 $X^{(0)}$ 中第 k 个数据与其前 $k-1$ 个数据之和的比值 $\rho(k)$ 来考查序列 $X^{(0)}$ 中的数据变化是否平稳。光滑比越小, 序列变化越平稳。

如果序列 $X^{(0)}$ 的光滑比 $\rho(k)$ 满足以下两个条件:

$$\begin{aligned} 1) & \frac{\rho(k+1)}{\rho(k)} < 1, k=2, 3, \dots, n-1 \\ 2) & \rho(k) \in [0, 0.5], k=3, 4, \dots, n \end{aligned} \quad (12)$$

则称 $X^{(0)}$ 为准光滑序列, 可以建立灰色模型。

(2) 平移变换

如果原始序列不满足光滑性条件, 则需要对其进行变换处理, 使之变得光滑。序列变换的方法有很多^[38], 文中以平移变换为例进行说明。平移变换是指原始序列 $X^{(0)}$ 中的所有数据均加上平移量 C , 得到新序列:

$$X = (x^{(0)}(1) + C, x^{(0)}(2) + C, \dots, x^{(0)}(n) + C)$$

原始序列 $X^{(0)}$ 经过平移变换后的效果如图 1 所示。其中, A 为原始序列, 波动情况十分剧烈。

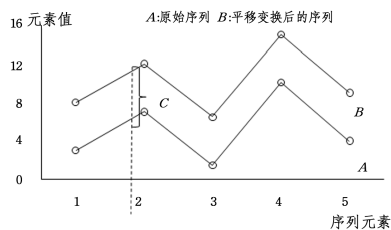


图 1 平移变换前后的序列形状

如图 1 所示, 如果对序列 A 中的每个元素都加上一个平移量 C , 则 A 向上平移变成序列 B 。由式 (11) 可知, 光滑比 $\rho(k)$ 的计算公式中包含分母和分子两部分, 其中分母是多个序列元素之和, 而分子只有一个元素。当序列 A 发生平移变换时, 分母增加的程度远大于分子, 光滑比 $\rho(k)$ 减小。虽然图 1 中 A 和 B 两个序列的形状相同, 但是从灰色模型的观点来看, 序列的光滑性得到了改善。通过选择 C 的大小, 总可以使 $\rho(k)$ 足够小, 从而使序列满足光滑性条件。

利用式 (11) 和式 (12) 可以计算得到平移量 C 的取值范围:

$$C > \frac{0.5 \sum_{i=1}^{k-1} x^{(0)}(i) - x^{(0)}(k)}{1.5 - 0.5k} \quad (13)$$

原始序列经过平移变换后得到的新序列可用于 GM(1, 1) 模型进行建模预测。

3.3 基于类推和灰色模型的阶段成本预测

AGSE 是类推和灰色模型两种方法的组合, 可以预测新项目的阶段(月)成本, 也可以在没有历史项目的情况下做出预测。在应用类推方法时, 除了使用项目属性外, 还可以使用前期阶段成本来帮助判断历史项目的相似性, 基本思想是: 两个项目相似, 不仅具有相似的属性, 前期阶段成本也应该是相似的。

新方法的预测过程包含两条路线: 1) 从历史数据集中选出 k 个与新项目属性和前期阶段成本均相似的项目, 然后用

相似项目的成本类推得到新项目阶段成本的“类推预测值”, 即 *analogy_pre*; 2) 利用新项目自身的前期阶段成本构建灰色序列, 如前 3 个、前 4 个阶段等, 建立 GM(1, 1) 模型, 预测后续阶段的成本, 得到“灰色预测值”, 即 *grey_pre*。最后, 按经验合成规则, 得到新项目的阶段成本预测值。利用 AGSE 方法预测新项目第 n 个阶段成本的过程如图 2 所示。

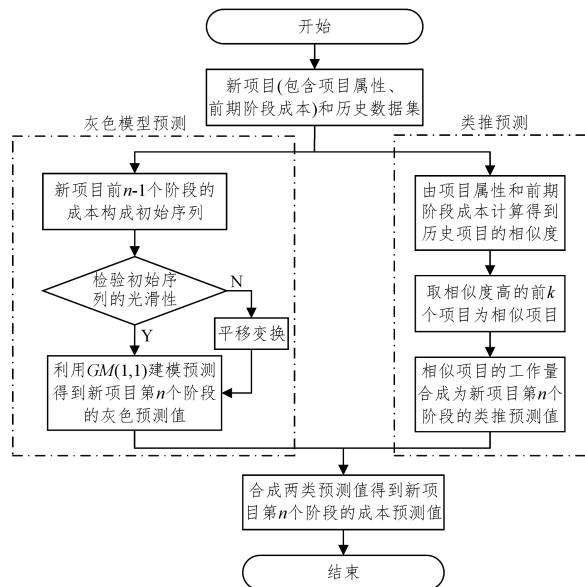


图 2 AGSE 方法的预测过程

详细步骤如下:

(1) 数据准备

AGSE 建模预测需要新项目的属性、前期阶段成本以及历史数据集(不是必须)作为模型的输入。其中, 新项目 and 历史项目均可以表示为属性和阶段成本的集合, 如式 (14) 所示。

$$[pf(1), \dots, pf(a), sc(1), \dots, sc(b)] \quad (14)$$

其中, pf 表示项目属性, sc 表示阶段成本。当然, 不同项目间属性的个数 a 和阶段数 b 各不相同。

(2) 灰色模型预测

1) 构建序列: 新项目前 $n-1$ 个阶段的成本构成初始序列 $X^{(0)}$ 。对于建模所需的最少阶段数, 还没有明确的标准。不过, 目前绝大部分的研究集中于使用 3 至 6 个阶段的数据建模^[18, 37]。为了使可预测的阶段数尽可能得多, 本节最少使用 3 个月的数据建模预测, 即 $n \geq 4$ 。在开发初期, 新项目的阶段数较少, 不足以建立灰色模型, 则可以使用类推预测得到的阶段成本来建立初始序列。

2) 前期成本序列是否光滑: 检验新项目前期成本序列的光滑性, 确定序列是否适合建模, 需要时进行平移变换处理, 详见 3.2.2 小节。

3) 灰色模型预测: 依照 3.2 小节 GM(1, 1) 模型建模预测的步骤, 计算得到第 n 个阶段的“灰色预测值”, 即 *grey_pre*。

(3) 类推预测

1) 描述项目: 在预测过程中, 项目处于不断变化的状态, 根据每个月的情况选择不同的属性应能得到更优的预测结果, 然而, 目前还没有数据集可以提供属性的动态记录。因此, 根据 3.1 节中的介绍, 本研究选取 20 个属性来描述项目(详细属性信息见 4.2 节)。相比于总成本预测, 使用类推方法进行阶段成本预测更加复杂。在计算相似度时, 除使用项

目属性外,还需要考虑前期阶段成本。

2)计算历史项目的相似度:根据式(1)和式(2)计算得到各个历史项目的相似度,其中前期阶段成本按属性数字值的方式处理。

3)相似项目类推预测:首先,历史项目按相似度由大到小排序,然后依照 3.1 节合成相似项目的工作量,其中项目个数 k 和权重 w 均取 1。也就是说,取相似度最高的项目的第 n 个阶段的成本作为“类推预测值”,即 $analogy_pre$ 。

(4)合成预测值

AGSE 的预测值合成过程分为以下 3 种情况:

1)当 $n \leq 3$ 时,灰色模型无法预测,类推预测值即为 AGSE 方法的最终预测结果,即 $\alpha = 1$ 。

2)当 $n > 3$ 时,将灰色预测值和类推预测值代入式(15),计算得到新项目第 n 个阶段的成本预测值。

$$P = \alpha \times analogy_pre + (1 - \alpha) \times grey_pre \quad (15)$$

其中, P 为新项目第 n 个阶段的成本预测值, $\alpha \in [0, 1]$ 是合成系数,可通过在训练集上学习取得最优值。当然,参数 α 的最优值会随着数据集的变化而改变。

3)当没有历史项目时,则直接使用新项目自身的成本数据建立灰色模型,计算得到预测值,即 $\alpha = 0$ 。

由以上实验过程可知,AGSE 方法建模预测的局限性较小。只有在既没有历史项目也没有前期阶段成本的情况下,AGSE 方法才无法工作。然而,在这种情况下,现有的模型均难以预测软件阶段成本。此时,可以考虑通过专家估算^[9]等对前期阶段的成本进行初步预测。

4 实验与结果

4.1 原始数据集概述

本文所用的数据集来源于 EDS 公司所维护的软件数据仓库,其已被用于多项研究^[2,26]。在实验研究中,我们对数据仓库中的项目进行了深入的分析,发现每个项目包含 100 多个属性,例如:行业包括通信、军事、金融、政府、制造、航天等;项目实施地包括美国、英国、澳大利亚、加拿大等;项目的开发语言包括 Visual Basic, C, C#, Java, SQL 等。通过复杂的记录方法,不仅项目属性被详细地记录下来,而且成本信息的记录更为具体。在原始数据集中,软件开发的每个月都有各种并发开发活动进行。其中,开发活动的成本每月记录一次。为了更直观地分析和预测软件成本,本文基于月成本(某月各项活动成本的总和)进行实验。

通过对原始数据集的分析,我们发现其中存在大量噪声数据,如不合理的极小值。因此,在实验之前需要对原始数据进行预处理。下一节将详细介绍数据的预处理过程。

4.2 数据预处理和实验数据集

4.2.1 项目属性描述

如 3.1 节所述,类推依靠属性对比寻找相似项目,因此我们需要在原始数据集 100 多个属性中选取合适的属性用于类推预测。Chen 等^[39]提出一种基于实验的属性选择方法。其基本思想是,首先用全部属性进行实验,然后逐个删减属性,检查预测精度是否改变,最后确定最佳属性集。在本文的实验中,采用该方法选定与项目相关的属性共 20 个,既包括数

字值也包括分类值,如表 1 所列。

表 1 项目属性

编号	属性	编号	属性
1	Proj-Id	11	Client-Participation
2	Proj-Type	12	Tool-Experience
3	Work-Type	13	Info-Tech-Experience
4	Industry	14	Language-Experience
5	Concurrency	15	Computer-Experience
6	Tech-Innovation	16	Methods-Experience
7	Bus-Innovation	17	Proj-Mgt-Experience
8	Appl-Innovation	18	System-Experience
9	Team-Complexity	19	Est-FP
10	Client-Complexity	20	Est-Duration

表 1 中,项目编号(Proj-Id)仅用于标识项目。项目类型(Proj-Type)包括新开发类、项目改进类、生产支持类和维护类等。工作类型(Work-Type)包括应用实施类、项目管理类和项目支持类等。Concurrency 是指项目的并发性,即不具备并发性、硬件和应用程序并发、硬件和软件并发等。创新性(Innovation)代表项目的新颖程度,例如团队曾开发过类似项目、同行业其他团队曾开发过类似项目和从未开发过此类项目等。复杂性(Complexity)由个人或团队独立开发、多个团队协同开发和承包给第三方等表示。经验(Experience)以年限为单位,即不到一年、一至三年和三年以上。在原始数据集中,有些团队已经对功能点个数(Function Point)和项目持续时间(Duration)进行了初步的估计。在实验预测过程中,此类数据可以作为参考。

在选取属性的过程中我们发现,原始数据集中存在着属性记录不完整的情况,属性缺失问题十分严重,这会对相似性度量产生不利影响。目前,已经有很多方法(例如删除方法和插值补偿方法)可以用于解决软件成本预测中的数据缺失问题^[40]。鉴于大规模数据集和数据缺失的随机性,我们采用删除方法^[41]。为此,按属性对原始数据集中的所有项目逐一检查,删除属性不完整的项目。

4.2.2 数据预处理

依照类推属性集和 GM(1,1)建模预测的要求进行数据预处理,最后余下 437 个项目,具体步骤如下:

(1)由于实验数据集中的项目需要记录完备的开发活动,因此其必须包含完整的软件生命周期。

(2)实验数据集中的项目要包含完整的属性,删除属性不完整的项目。

(3)原始数据集中包含的开发方法有 Com-Iterative, System Life Cycle 3 和 Standard System Development 等。在这些开发方法中,使用 Com-Iterative 方法开发的项目多达 61%。因此,本文着重研究开发方法为 Com-Iterative 的项目,即实验数据集中的所有项目都有相同的开发方法学类型。

(4)由于不同项目所包含的开发活动的类型和数量等各不相同,因此按月汇总同一项目的各类成本,生成以月份为间隔、项目持续月数为长度的成本序列。同时,删除成本记录不完整和全职工作人员数小于或等于 0 的项目。

(5)灰色系统理论中的 GM(1,1)模型,至少需要连续 3 个月的数据建模用于预测下一个月的成本。虽然 AGSE 不需要项目月数超过 3 个月,但是为了与其他基于灰色模型的方法比较,本文删除数据集中工作量小于或等于 3 个月的项目。

(6)项目开发要结合实际情况,避免非正常值对实验结果的影响,删除某月总工作量小于或等于 10 小时的项目。

(7)原始数据集中包含某月工作量突然变小而下一月又恢复的情况,此类项目为数据记录异常,将其删除。

(8)通过比较余下的项目发现,存在一个工作量持续 33 个月的项目,若该项目在实验中被选为测试集,则无法进行类推预测,因此删除该项目,剩余项目组成实验数据集。

4.2.3 实验数据集概述

如上所述,经过数据预处理可以得到实验数据集,其中包括 437 个项目。表 2 总结得出实验数据集中的项目信息,单位为人小时。

表 2 实验数据集的描述性统计

项目数	属性数	平均值	标准差	最小值	最大值
437	20	638.72	924.15	10.50	7143.75

从表 2 可以看出,项目数为 437,属性数为 20,每月工作量从 10.50 到 7143.75 人小时不等,标准差为 924.15 人小时。这说明,实验数据集中的项目种类丰富,可以充分测试新方法的性能,得出更为可靠的结论。

4.3 评价标准

在最近的研究中,MMRE 及其相关评价指标被广泛批评^[42-45]。目前,在软件成本预测领域广泛应用的评价标准为绝对误差(Absolute Error),其基本定义如下:

$$Absolute\ Error_i = |\epsilon_i - \hat{\epsilon}_i| \tag{16}$$

其中, ϵ_i 为实际值, $\hat{\epsilon}_i$ 为预测值。

表 3 AGSE 在数据集 A 上的预测结果

(单位:人小时)

预测方法	GM(1,1)					AGSE					Analogy	
	α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
MdAE		114.15	104.06	95.46	90.14	83.29	77.59	72.80	69.83	71.61	73.63	85.00
MAE		196.07	178.55	162.86	148.99	137.57	129.79	126.72	128.35	133.75	142.76	156.20

表 3 中,参数 α 是式(15)中的合成系数。根据式(15),当 $\alpha=0$ 时,AGSE 转化成纯灰色模型方法; $\alpha=1$ 时,AGSE 转化成纯类推方法; $0<\alpha<1$ 时,AGSE 为类推和灰色模型相结合的预测方法。为了确定 AGSE 方法的合成系数 α ,使用表 3 中 MdAE 和 MAE 的值做折线图。

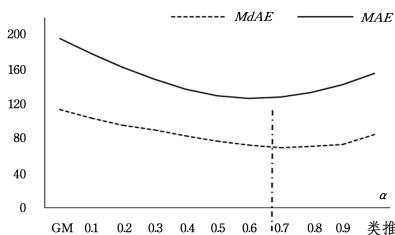


图 3 MdAE 和 MAE 折线图

由图 3 可以发现,曲线两边高,中间低。综合各项指标,当 $\alpha=0.7$ 时,AGSE 方法的预测效果最好。

从表 3 可以看出,随着历史项目的引入(从 0 向 0.7 变化),类推预测开始影响 AGSE 的预测结果,衡量指标中 MdAE 和 MAE 的值不断减小,预测结果的准确性越来越好。随着项目自身数据的影响逐渐被消除(从 0.7 向 1 变化),灰色预测成分减少,两个指标均在恶化,预测结果的准确性越来越差。AGSE 方法比 GM(1,1)模型和类推(Analogy)预测在

为了充分检验所提方法的性能,采用两个基于绝对误差的衡量指标,即绝对误差的中位数(Median of Absolute Errors, MdAE)和绝对误差的平均值(Mean of Absolute Errors, MAE),MdAE 和 MAE 的值越接近 0,表示预测结果越好。

4.4 交叉验证

为了充分利用实验数据集中的每个项目,使用 leave-one-out 验证方法^[46]来评价方法的性能。leave-one-out 方法是从实验数据集中随机取出一个项目,用剩余的 $n-1$ 个项目预测该项目的成本;然后将该项目放回,再取出另一个项目重复预测过程,直至数据集中的所有项目均被预测过一次;最后,在 n 次预测中计算 4.3 小节中衡量指标的结果。这种方法符合实际应用情况:现实预测通常是根据历史项目来预测新项目的成本,这就类似于从数据集中挑选出一个项目作为测试集,剩余项目组成训练集。

4.5 实验与分析

本文的实验分为两个阶段:第一个阶段得到 AGSE 的参数 α 值;第二个阶段选用类推方法(Analogy)、GV^[26]、GM(1,1)、线性回归(LR)^[19]和卡尔曼滤波(KF)^[47]作为基准方法,以评价 AGSE 的预测性能。值得注意的是,若某一数据集已经用于训练得到参数 α 的最优值,则该数据集就不能再次用于测试 AGSE 方法的性能。为此,我们将 4.2.3 小节得到的 437 个项目随机分为数据集 A 和 B 后分别用于这两个阶段的实验。其中,数据集 A 包含 218 个项目,B 包含 219 个项目。

依照 3.3 节中 AGSE 的实现过程,在数据集 A 上进行实验,所得 MdAE 和 MAE 的值如表 3 所列。

MdAE 上改进了 63% 和 22%,在 MAE 上改进了 53% 和 22%。

由以上分析可知,单纯用类推方法或者灰色模型的预测效果都不好,只有将二者相结合才能得到更准确的预测结果,因此 AGSE 优于类推方法和灰色模型。

为了更全面地检验 AGSE 方法的性能,使用 AGSE($\alpha=0.7$)、类推(Analogy)、GV、GM(1,1)、卡尔曼滤波(KF)和线性回归(LR)在数据集 B 上进行实验,实验结果如表 4 所列。

表 4 AGSE, Analogy, GV, GM(1,1), KF 和 LR 在数据集 B 上的预测结果

(单位:人小时)

预测方法	MdAE	MAE
AGSE	68.46	126.49
Analogy	76.75	149.34
GV	113.85	196.60
GM(1,1)	121.21	204.64
KF	140.89	223.90
LR	165.00	258.64

在本实验中,参数 α 的值取 0.7。从表 4 中可以得出,在同样的数据集上,AGSE 比 Analogy, GV, GM(1,1), KF 和 LR 在 MdAE 上分别改进了 12%, 66%, 43%, 106% 和 141%;在 MAE 上改进了 18%, 55%, 62%, 77% 和 104%。

由上述结果可知,AGSE的预测结果优于其他5种方法,预测精度更高,有能力准确地预测软件阶段成本。

为了更清楚地分析这6种方法在MdAE和MAE方面的差异,对其进行显著性检验^[48]。因为Absolute Errors不满足正态分布,所以我们采用Wilcoxon检验来检查AGSE相比其他5种方法在MdAE和MAE方面是否存在显著改进。检验结果如表5所列。

表5 统计检验的结果

方法对比	p -value
AGSE vs Analogy	$10e-3$
AGSE vs GV	$10e-3$
AGSE vs GM(1,1)	$10e-3$
AGSE vs KF	$10e-3$
AGSE vs LR	$10e-3$

由表5可知,所有检验结果的 p 值均小于0.001,这意味着AGSE的预测精度明显优于类推、GV、GM(1,1)、卡尔曼滤波和线性回归。

箱图(Box-plot)是显示数据分布情况的统计图。本文采用箱图描绘AGSE、Analogy、GV、GM(1,1)、KF和LR每月预测结果的绝对误差分布情况,如图4所示。

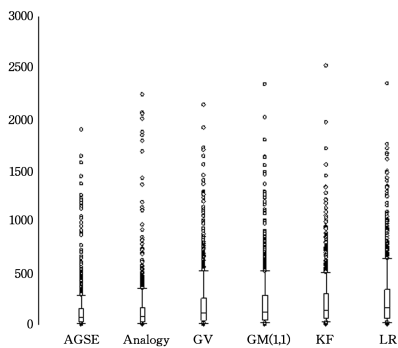


图4 AGSE、Analogy、GV、GM(1,1)、KF和LR的月误差绝对值分布情况

从图4可以看出:1)AGSE方法的箱体最小,说明其预测结果的误差分布更加集中;2)AGSE方法的箱体向下偏斜,意味着有更多的高精度预测值;3)AGSE方法的中位数比其他5种方法都小,说明至少有一半的预测结果优于其他方法;4)AGSE方法的极值点小且数量少,意味着其有更小的方差,这表示AGSE方法的预测结果更稳定。简而言之,在真实的软件项目上的月误差绝对值箱图表明,AGSE方法的预测性能优于类推方法(Analogy)、GV方法、GM(1,1)模型、卡尔曼滤波和线性回归。

结束语 软件阶段成本预测可使管理者及时发现开发过程中的潜在问题,提前采取预防措施,保障项目开发的顺利进行。本文首次将类推方法和灰色模型结合,提出了一种全新的预测方法——AGSE,用于解决软件阶段成本预测方面的问题。新方法以月为阶段单位,一个月即为一个阶段,不仅可以通过项目属性和前期阶段成本检索相似项目,得出“类推预测值”,同时还可以利用灰色系统理论中的GM(1,1)模型,在新项目的前期阶段成本中寻找规律,建模预测得“灰色预测值”;然后,将两类预测值合并,即为新项目的阶段成本预测值。AGSE可以通过在训练集上学习,自动平衡类推和GM(1,1)的贡献,甚至可以切换成纯类推方法或GM(1,1)模型进行预测。

本文实验是在真实的软件项目数据集上进行的。实验结果表明,AGSE方法可以达到较高的预测精度。同时,使用类推方法、GV方法、GM(1,1)、卡尔曼滤波和线性回归作为基准方法来测试新方法的性能。结果表明,AGSE在Median of Absolute Errors和Mean of Absolute Errors方面的表现均优于基准方法,新方法的预测性能更好,准确度更高,有潜力作为解决软件阶段成本预测问题的备选方法。

在实际预测过程中,项目处于不断变化的状态,可以根据每个阶段(周、月等)的具体情况选取不同的属性或者参数值。

未来,我们准备在其他数据集上进行实验,同时也将尝试使用更多的方法和技术等来扩展我们的研究。

参考文献

- [1] JORGENSEN M, SHEPPERD M. A systematic review of software development cost estimation studies[J]. IEEE Transactions on Software Engineering, 2006, 33(1): 33-53.
- [2] WANG Y, SONG Q B, SHEN J Y. Grey learning based software stage-effort estimation[C]// 2007 International Conference on Machine Learning and Cybernetics. IEEE, 2007: 1470-1475.
- [3] HASTIE S, WOJEWODA S. Standish Group 2015 Chaos Report Q&A with Jennifer Lynch[J]. Retrieved, 2015, 1(15): 2016.
- [4] BOEHM B W. Understanding and Controlling Software Costs [J]. IEEE Transactions on Software Engineering, 1988, 14(10): 1462-1477.
- [5] 贾经冬, 林广艳. 软件项目管理[M]. 北京: 高等教育出版社, 2012.
- [6] AZZEH M, NASSIF A B. A hybrid model for estimating software project effort from Use Case Points[J]. Applied Soft Computing, 2016, 49: 981-989.
- [7] CHOU J S, WU C C. Estimating software project effort for manufacturing firms[J]. Computers in Industry, 2013, 64(6): 732-740.
- [8] MOLOKKEN K, JORGENSEN M. A review of software surveys on software effort estimation[C]// 2003 International Symposium on Empirical Software Engineering, 2003(ISESE 2003). IEEE, 2003: 223-230.
- [9] HUGHES R T. Expert judgement as an estimating method[J]. Information and Software Technology, 1996, 38(2): 67-75.
- [10] SRINIVASAN K, FISHER D. Machine learning approaches to estimating software development effort[J]. IEEE Transactions on Software Engineering, 1995, 21(2): 126-137.
- [11] SONG Q B, SHEPPERD M, MAIR C. Using Grey Relational Analysis to Predict Software Effort with Small Data Sets[C]// IEEE International Symposium on Software Metrics. IEEE, 2005: 35.
- [12] HSU C J, HUANG C Y. Comparison of weighted grey relational analysis for software effort estimation [J]. Software Quality Journal, 2011, 19(1): 165-200.
- [13] HUANG S J, CHIU N H, CHEN L W. Integration of the grey relational analysis with genetic algorithm for software effort estimation[J]. European Journal of Operational Research, 2008, 188(3): 898-909.
- [14] SHEPPERD M, SCHOFIELD C. Estimating software project effort using analogies[J]. IEEE Transactions on Software Engi-

- neering,1997,23(11):736-743.
- [15] MUKHOPADHYAY T, VICINANZA S S, PRIETULA M J. Examining the feasibility of a case-based reasoning model for software effort estimation[J]. *MIS quarterly*,1992,16(2):155-171.
- [16] AZZEH M, COWLING P I, NEAGU D. Software stage-effort estimation based on association rule mining and Fuzzy set theory [C]// 2010 IEEE 10th International Conference on Computer and Information Technology (CIT). IEEE,2010:249-256.
- [17] BOEHM B W. *Software engineering economics*[M]. Englewood Cliffs (NJ):Prenticehall,1981.
- [18] DENG J L. Control problems of grey systems[J]. *Systems & Control Letters*,1982,1(5):288-294.
- [19] MONTGOMERY D C, PECK E A, VINING G G. *Introduction to linear regression analysis* [M]. John Wiley & Sons,2015.
- [20] HEIAT A. Comparison of artificial neural network and regression models for estimating software development effort[J]. *Information and Software Technology*,2002,44(15):911-922.
- [21] HUANG H, HUANG S, CHEN J, et al. An image information hiding algorithm based on grey system theory[J]. *International Journal of Communication Systems*,2014,27(10):2426-2442.
- [22] SU S L, SU Y C, HUANG J F. Grey-based power control for DS-CDMA cellular mobile systems[J]. *IEEE Transactions on Vehicular Technology*,2000,49(6):2081-2088.
- [23] GUO J J, WU J Y, WANG R Z. A new approach to energy consumption prediction of domestic heat pump water heater based on grey system theory[J]. *Energy and Buildings*,2011,43(6):1273-1279.
- [24] KEUNG J. Software development cost estimation using analogy:a review[C]// 2009 Australian Software Engineering Conference. IEEE,2009:327-336.
- [25] AZZEH M, NEAGU D, COWLING P I. Analogy-based software effort estimation using Fuzzy numbers[J]. *Journal of Systems and Software*,2011,84(2):270-284.
- [26] WANG Y, SONG Q B, MACDONELL S, et al. Integrate the GM (1,1) and Verhulst models to predict software stage effort [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*,2009,39(6):647-658.
- [27] YANG Y, HE M, LI M, et al. Phase distribution of software development effort [C]// Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM,2008:61-69.
- [28] MACDONELL S G, SHEPPERD M J. Using priorphase effort records for reestimation during software projects[C]// Ninth International Software Metrics Symposium. IEEE,2003:73-86.
- [29] KULKARNI A, GREENSPAN J B, KRIEGMAN D A, et al. A generic technique for developing a software sizing and effort estimation model[C]// Twelfth International Conference on Computer Software and Applications,1988 (COMPSAC88). IEEE,1988:155-161.
- [30] OHLSSON M C, WOHLIN C. An empirical study of effort estimation during project execution[C]// Sixth International Software Metrics Symposium. IEEE,1999:91-98.
- [31] TSUNODA M, TODA K, FUSHIDA K, et al. Revisiting software development effort estimation based on early phase development activities[C]// *Mining Software Repositories*. IEEE,2013:429-438.
- [32] FERRUCCI F, GRAVINO C, SARRO F. Exploiting prior-phase effort data to estimate the effort for the subsequent phases:a further assessment[C]// Proceedings of the 10th International Conference on Predictive Models in Software Engineering. ACM,2014:42-51.
- [33] AZZEH M, ELSHEIKH Y, ALSEID M. An Optimized Analogy-Based Project Effort Estimation[J]. *International Journal of Advanced Computer Science & Applications*,2014,5(4):6-11.
- [34] SIGWENI B, SHEPPERD M. Feature weighting techniques for CBR in software effort estimation studies;a review and empirical evaluation[C]// Proceedings of the 10th International Conference on Predictive Models in Software Engineering. ACM,2014:32-41.
- [35] KOLODNER J. *Case-based reasoning*[M]. Morgan Kaufmann,2014.
- [36] AHA D W. Case-based learning algorithms[C]// Proceedings of the 1991 DARPA Case-Based Reasoning Workshop. 1991:147-158.
- [37] 刘思峰,杨英杰,吴立丰. *灰色系统理论及应用(第7版)*[M].北京:科学出版社,2014.
- [38] 崔立志,刘思峰. 基于数据变换技术的灰色预测模型[J]. *系统工程*,2010(5):104-107.
- [39] CHEN Z, MENZIES T, PORT D, et al. Finding the right data for software cost modeling[J]. *IEEE Software*,2005,22(6):38-46.
- [40] LITTLE R J A, RUBIN D B. *Statistical analysis with missing data*[M]. John Wiley & Sons,2014.
- [41] STRIKE K, EL EMAM K, MADHAVJI N. Software cost estimation with incomplete data[J]. *IEEE Transactions on Software Engineering*,2001,27(10):890-908.
- [42] SHEPPERD M, MACDONELL S. Evaluating prediction systems in software project estimation[J]. *Information and Software Technology*,2012,54(8):820-827.
- [43] WHIGHAM P A, OWEN C A, MACDONELL S G. A baseline model for software effort estimation[J]. *ACM Transactions on Software Engineering and Methodology(TOSEM)*,2015,24(3):20.
- [44] MITTAS N, MAMALIKIDIS I, ANGELIS L. A framework for comparing multiple cost estimation methods using an automated visualization toolkit[J]. *Information and Software Technology*,2015,57:310-328.
- [45] KITCHENHAM B, MADEYSKI L, BUDGEN D, et al. Robust statistical methods for empirical software engineering[J]. *Empirical Software Engineering*,2018,22(2):579-630.
- [46] KOHAVI R. A study of cross-validation and boot-strap for accuracy estimation and model selection[C]// *IJCAI*. 1995:1137-1145.
- [47] KALMAN R E. A New Approach to Linear Filtering and Prediction Problems[J]. *Journal of Basic Engineering Transactions*,1960,82:35-45.
- [48] KITCHENHAM B A, PICKARD L M, MACDONELL S G, et al. What accuracy statistics really measure[J]. *IEEE Proceedings-Software*,2001,148(3):81-85.