



软件成本估算的粒子群算法 类比模型及自助法推断

吴登生^{1,2}, 李建平¹, 蔡晨¹

1 中国科学院 科技政策与管理科学研究所, 北京 100190

2 中国科学院 研究生院, 北京 100049

摘要: 软件成本估算是控制软件进度、降低软件风险和保证软件质量的有效措施, 已引起产业界和学术界的广泛关注。为能更准确地估算软件成本, 采用粒子群算法优化加权类比估算模型(PSO 类比模型)中各个特征属性的权重, 避免权重选择的盲目性; 同时采用非参数自助法对原始数据抽样, 在自助法子样本的基础上, 分析 PSO 类比模型中的类比项目个数和成本计算方式对模型精确度的影响, 并进一步计算新项目估算成本值的可信度以及在一定置信水平下的区间值; 采用 Desharnais 数据库验证 PSO 类比模型与自助法推断方法的有效性, 根据 MMRE 和 Pred(0.25) 两个标准将 PSO 类比模型与一般类比模型、支持回归机、人工神经网络、径向基神经网络和分类回归树进行估算精度比较。研究结果表明, 采用粒子群算法优化权重的 PSO 类比估算模型能得到较高的估算精度, 且自助法能有效校正 PSO 类比模型中的相关变量并检验结果的可信度, 可以为管理者在软件项目风险分析和项目规划方面提供有益参考。

关键词: 可信软件; 软件成本估算; 类比方法; 粒子群算法; 自助法

中图分类号: C931

文献标识码: A

文章编号: 1672-0334(2010)03-0113-08

1 引言

随着软件应用范围的不断扩大以及在重大工程中应用地位的日益显著, 以产品安全性为主的关于软件可信性的讨论和研究逐渐引起人们的重视。1994年由美国多家政府组织和商业机构参与的可信软件方法学项目将软件可信性定义为软件满足既定需求的信心度, 阐述可信性对管理决策、技术决策以及既定需求集合的高度依赖性^[1]。从过程角度理解软件的可信性并保证软件的质量需求都得到可信的实现, 是解决问题、迎接挑战的一个有效途径。作为软件管理过程中的一个重要环节, 准确估算软件成本是控制软件进度、降低软件风险和保证软件质量的重要措施, 受到产业界和学术界的广泛关注。事实上, 软件开发项目中传统的质量、成本和进度三大约束条件是可以相互转化的, 软件的成本直接影响到软件的质量和进度。Boehm 指出, 理解并控制软件成本带给我们的不仅仅是更多的软件, 而且是更好的软件^[2]。The Standish Group 发布的《CHAOS Sum-

mary 2009》显示, 只有32%的项目是成功的, 即能在预算和限期内完成; 有24%的项目是失败的, 即项目未能完成或取消; 剩下44%的项目是被质疑的, 即虽然完成了但严重超支^[3]。人们一致认为对软件成本估算不足和需求不稳定是造成软件项目失控的两大主要原因^[4], 所以有必要对软件成本估算模型进行深入研究, 以提高成本估算的精确度。

2 相关研究评述

通常认为软件成本估算方法分为算法模型、专家判断、回归分析和类比估算4种^[5]。算法模型就是找到影响软件成本的各种因子, 并分析其影响程度, 运用一个成本驱动因子变量的函数估算软件成本, 如 COCOMO 方法^[6]。该方法比较直观、高效, 而且可重复, 但是它不能反映成本与项目因子之间暗含的非线性和相互影响的关系, 具有一定的局限性。专家判断是指根据专家经验、历史数据、过程指引对新项目的成本做出估算^[7], 如 Molokken 等用规划扑

收稿日期: 2009-12-13 修返日期: 2010-03-29

基金项目: 国家自然科学基金(90718042, 70531040)

作者简介: 吴登生(1984-), 男, 安徽庐江人, 中国科学院科技政策与管理科学研究所博士研究生, 研究方向: 风险管理和软件工程等。E-mail: wds@casipm.ac.cn

克方法集结专家意见,最终得到专家对新项目成本的估算值^[8];Koch等在专家判断的基础上运用投票规则估算软件成本^[9]。由于专家判断的方法过于依赖专家自身的经验,估算的精确度一般都不高。回归分析是指在已有历史数据的基础上,将相关影响因素作为自变量,项目成本作为因变量,通过一定的方法找到自变量与因变量之间的线性或非线性关系,作为估算新项目成本的依据。用回归分析估算软件成本的研究较多,如张俊光用线性回归^[10]、Heiat用人工神经网络和线性回归^[11]、Sentas等用序数回归^[12]、Kumar等利用小波神经网络^[13]、Huang等用模糊神经网络^[14]、Oliveira用支持回归机^[15]等刻画自变量与因变量之间的线性或非线性关系。线性回归和序数回归的结果容易受异常值的影响,神经网络方法对数据样本量的要求较高,现有的软件成本数据很难达到要求。类比估算分析是将新项目与多个已完成的项目进行对比,找出与新项目最相似的一个或多个项目,作为估算新项目成本的依据。该方法比较直观,具有很好的可理解性和很强的样本学习能力,且符合人们解决问题的常规思维,在软件成本类比估算中得到广泛应用^[16]。一般用欧式距离衡量各个项目与新项目的相似程度,Huang等用灰色关联度作为判断两个项目相似程度的标准是对类比方法的一种改进^[17]。现有的研究表明,用加权欧式距离作为判断标准时,能够达到较高的准确度,而在加权类比模型中选择合适的权重系数是一个非常重要的问题^[18]。常用的权重选择方法有两种,一种是由专家根据具体情况人为确定权重,这种方法主观性较强,而且权重不能适应不同数据库;另一种方法是采用启发式算法(如遗传算法)在类比的过程中计算出最优估算结果的权重。

本研究提出一种粒子群算法(particle swarm optimization, PSO)优化权重的加权类比模型估算软件成本,用加权欧式距离作为衡量项目之间相关性的标准,PSO方法用来优化类比估算中各个特征属性的权重。PSO方法保留了种群的全局搜索策略,而它采用速度-位置搜索方法,避免其他算法复杂的遗传步骤,操作简单,被认为是一种更高效的并行算法^[19]。在此基础上,采用自助法(Bootstrap)校正PSO类比方法中相关变量的选择,如类比项目个数和成本计算方式。此外,还通过Bootstrap检验新项目估算成本的可信度,并推断其在一定置信水平下的区间值。

3 软件成本估算的类比法

类比法作为基于案例推理(case-based reasoning, CBR)的一种形式,已经在软件成本估算中得到广泛应用^[19]。该方法主要是通过多个已完成的项目与新项目进行对比,找到一个或几个与新项目最相似的历史项目,以估算新项目的成本。

3.1 类比项目相似度的计算

度量新项目和历史项目相似度的标准有多种,

一般采用欧式距离较多,包括加权欧式距离和非加权欧式距离两种。非加权类比是指在计算两个项目的相似度时采用非加权欧式距离作为判断标准,即项目中每个属性是等权重的,具体计算方法见(1)式和(2)式。加权类比是指在计算两个项目的相似度时采用加权欧式距离,用不同的权重反映各属性之间的相对重要程度,具体见(1)式和(3)式。本研究采用PSO算法求各个特征属性的最佳拟合权重。

$$D(C_x, C_y) = \sqrt{\sum_{i=1}^p \text{Dis}(C_{xi}, C_{yi})} \quad (1)$$

其中, $D(C_x, C_y)$ 为 C_x 与 C_y 之间欧式距离, C_x 为新项目, C_y 为历史项目, C_{xi} 为 C_x 中第 i 个属性的值, C_{yi} 为 C_y 中第 i 个属性的值, p 为样本中含有的属性个数。 $D(C_x, C_y)$ 值越大,表示两个项目之间的距离越大,即其相似性越小。

C_{xi} 与 C_{yi} 之间欧式距离的计算公式为

$$\text{Dis}(C_{xi}, C_{yi}) = \begin{cases} (C_{xi} - C_{yi})^2 & C_{xi} \text{ 为数值} \\ 1 & C_{xi} \text{ 为名义值且 } C_{xi} \neq C_{yi} \\ 0 & C_{xi} \text{ 为名义值且 } C_{xi} = C_{yi} \end{cases} \quad (2)$$

C_{xi} 与 C_{yi} 之间加权欧式距离的计算公式为

$$\text{Dis}(C_{xi}, C_{yi}) = \begin{cases} W_i(C_{xi} - C_{yi})^2 & C_{xi} \text{ 为数值} \\ W_i & C_{xi} \text{ 为名义值且 } C_{xi} \neq C_{yi} \\ 0 & C_{xi} \text{ 为名义值且 } C_{xi} = C_{yi} \end{cases} \quad (3)$$

其中, W_i 为第 i 个属性的权重,其值由PSO算法计算得到。

3.2 项目拟合数目和数值计算方式

计算出新项目与历史项目之间的相似度后,还需要确定用几个历史项目作为新项目成本估算的依据。选择用多个相似程度较高的历史项目作为新项目成本估算的依据,还需要明确成本的计算方式,一般可以用多个项目成本的均值或中位数作为新项目成本的估算值,具体计算公式为

$$\hat{E}_{new} = \text{mean}(E_1, \dots, E_k)$$

$$\text{或 } \hat{E}_{new} = \text{median}(E_1, \dots, E_k)$$

$$k = 1, \dots, n - 1 \quad (4)$$

其中, \hat{E}_{new} 为新项目的估算成本; k 为作为计算新项目成本依据的历史项目个数,从理论上说, k 可以取值为 $1, 2, \dots, (n - 1)$ (n 为数据库中项目总数); E_1, \dots, E_k 为历史项目的成本,mean表示取 k 个项目成本的均值,median表示取 k 个项目成本的中位数。本研究在比较非加权类比法和PSO类比法时, k 值取 $1, 2$ 和 3 三种情况;在利用Bootstrap校正PSO类比法中类比项目个数 k 和成本计算方法时, k 取值为 $1, 2, \dots, 20$ 。

3.3 评价标准

本研究采用平均相对误差(mean magnitude related

error, MMRE) 和 $Pred(0.25)$ 两个标准衡量模型的准确度^[20], $MMRE$ 和 $Pred(0.25)$ 标准都是建立在相对误差值 (magnitude related error, MRE) 的基础上。定义 MRE_i 为数据集中第 i 个项目估算的成本值与实际成本值的误差, 计算公式为

$$MRE_i = \frac{|E_i - \hat{E}_i|}{E_i} \quad (5)$$

其中, E_i 为数据集中第 i 个项目的实际成本值, \hat{E}_i 为数据集中第 i 个项目的估算成本值。在 MRE_i 的基础上, 用平均相对误差标准衡量该模型在数据集中的整体误差, 相当于将数据集中每个项目的估算误差进行等权重的加权和处理, 计算公式为

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i \quad (6)$$

其中, n 为该数据集中项目的总数。

在软件成本估算研究中, 还经常采用另外一种评价标准衡量模型的估算误差, 即 $Pred(0.25)$, 具体计算公式为

$$Pred(0.25) = \frac{m}{n} \quad (7)$$

其中, m 为数据集中相对误差值小于 0.25 ($MRE < 0.25$) 的项目个数。

4 基于 PSO 优化权重的类比法

4.1 PSO 算法简介

PSO 算法采用速度和位置两个参数作为搜索依据^[21]。群体中第 i 个粒子在 D 维解空间的位置为 X_i , $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$, 速度为 V_i , $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$, 当前时刻个体的极值记为 p_{ibest} , 全局极值记为 g_{best} 。在每一次迭代中, 粒子跟踪个体极值、全局极值和自己前一刻的状态来调整当前时刻的位置和速度, 迭代过程为

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot rand() \cdot [p_{ibest} - X_i(t)] + c_2 \cdot rand() \cdot [g_{best} - X_i(t)] \quad (8)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (9)$$

其中, $V(t+1)$ 为粒子下一时刻的速度; $V(t)$ 为粒子当前时刻的速度; $X(t)$ 为粒子当前时刻的位置; $X(t+1)$ 为粒子下一时刻的位置; $rand()$ 为 $[0, 1]$ 之间的随机数; c_1 和 c_2 为学习因子; w 为权重因子, 为加快收敛速度, 其值应随算法迭代的增加而自动减小, 具体定义为

$$w = w_{min} + \frac{(iter_{max} - iter) \cdot (w_{max} - w_{min})}{iter_{max}} \quad (10)$$

其中, w_{max} 为最大权重因子, w_{min} 为最小权重因子, $iter$ 为当前迭代次数, $iter_{max}$ 为总的迭代次数。

4.2 带 PSO 优化权重的类比法

在软件成本估算应用的过程中使用类比方法需要特别注意两个问题, 一是如何构建案例集, 二是如

何选择一组与新项目最相似的案例。已有的研究工作证明, 用已经完成的软件项目作为类比方法中的案例集, 能够很好地解决软件成本估算类比方法中案例集较难建立这一困难。而如何找到一个或几个与新项目最相似的历史项目, 关键是在计算新旧项目相似度时能否选择合适的属性权重。传统类比方法或者是将各属性进行等权重处理, 或者利用专家判断的方法确定权重, 严重影响模型的精确度。Huang 等采用遗传算法优化类比方法中属性权重, 虽然能提高模型的精确度, 但是其复杂的遗传步骤会影响模型的效率, 尤其是采用交叉验证方法验证模型精确度时^[18]。本研究采用 PSO 算法优化类比方法中的属性权重, PSO 方法用速度-位置搜索方法, 既保留种群的全局搜索策略, 又避免复杂的遗传步骤, 在优化权重方面已证实其精度和效率都优于遗传算法^[22]。

PSO 类比法的基本思想是在计算新旧项目相似度时, 采用 PSO 方法优化属性的权重, 根据数据的特性, 找到最优权重, 从而提高模型的精确度。计算过程中采用留一交叉验证法检验模型, 取 $MMRE$ 值作为 PSO 算法中的适应度函数值。加权类比法的权重编码成 PSO 中的一个粒子, 具体步骤如下。

步骤 1 初始化 PSO 算法模块的相关参数, 包括种群规模、惯性权重等, 为每个粒子随机分配一组位置信息和速度信息;

步骤 2 迭代次数 $iter=0$;

步骤 3 用每个粒子作为加权类比模型中的一组权重, 在软件成本数据的基础上计算 $MMRE$ 值;

步骤 4 判断是否达到退出条件, 即计算适应度函数, 若适应度函数值满足要求或达到设定迭代次数, 转到步骤 7;

步骤 5 迭代次数 $iter=iter+1$;

步骤 6 确定 p_{best} 和 g_{best} , 更新粒子位置信息和速度信息, 得到新的权重粒子群, 转到步骤 3;

步骤 7 给出最佳权重, 在历史数据的基础上计算另一个评价标准 $Pred(0.25)$, 并用得到的权重估算新软件项目的成本。

5 基于 Bootstrap 的推断

Bootstrap 法是由 Efron 在 1979 年提出的, 主要是通过样本的经验分布进行随机再抽样, 得到 Bootstrap 子样本, 然后再进行统计量的估计^[23]。在软件成本估算方面, Angelis 和 Stamelos 运用 Bootstrap 分析软件成本估算中的相关变量和结果的不确定性^[24, 25]。本研究主要采用非参数 Bootstrap 比较 PSO 类比法中类比项目个数和类比项目计算方法, 并计算新项目估算成本的可信度和置信区间值。

5.1 非参数 Bootstrap 简介

非参数 Bootstrap 过程的机制是, 首先有一个实际观测到的数据集含有 n 个观察样本, 从这个数据集中有放回地随机抽取 m 个组成一个新样本, 称之为 Bootstrap 子样本。在这个随机抽样中, 原始数据集中

的每个观察单位每次被抽到的概率相等(为 $\frac{1}{n}$)。

非参数 Bootstrap 不需假设观测数据符合某一分布形式,直接由经验分布进行抽样。在无任何参数假设的情况下,经验分布函数是观测数据分布的极大似然估计。

5.2 Bootstrap 校正类比项目个数和成本计算方式

根据软件成本估算类比模型的介绍,有两个重要的变量会影响类比估算模型的准确性,分别是类比项目个数和成本计算方式。现有的研究成果在选择类比项目个数和成本计算方式时,通常是比较不同组合情况下模型的 *MMRE* 值。而用传统方法每一种组合只能得到一个 *MMRE* 值,不能对 *MMRE* 值均值或方差进行比较,影响选择结果的科学性。

本研究在原数据的基础上采用 Bootstrap 抽样,得到 B 个 Bootstrap 子样本,用 PSO 类比模型分别计算每种类比项目个数和成本计算方式组合在 Bootstrap 子样本中的 *MMRE* 值和 *Pred*(0.25) 值。通过对所有 Bootstrap 子样本中得到的误差进行比较(通过箱线图展示比较结果),选择合适的类比项目个数和成本计算方式估算新项目成本。

5.3 Bootstrap 计算新项目估算成本的可信度和置信区间值

在估算新项目成本时,传统的类比方法只能估算一个成本值,无法进一步对估算成本值的可信度进行分析。本研究运用 Bootstrap 对原有历史数据进行抽样,生成 B 个 Bootstrap 子样本,在此基础上运用 PSO 类比法得到新项目估算成本的分布情况,并进一步计算原估算值的可信度和置信区间。

定义 \hat{E}_{new} 为用 PSO 类比法在原始数据基础上得到的估算成本。为度量 \hat{E}_{new} 可信度,定义 $E_{new}^*(b)$ 为运用 PSO 类比法在第 b 个 Bootstrap 子样本上得到的估算成本, $b = 1, \dots, B$, B 为 Bootstrap 子样本的总数。通常采用标准差 SE_{boot} 和偏差 $BIAS_{boot}$ 的绝对比值衡量估算成本 \hat{E}_{new} 可信度的大小,计算公式为

$$SE_{boot} = \sqrt{\frac{\sum_{b=1}^B [E_{new}^*(b) - E_{new}^*(\cdot)]^2}{B-1}} \quad (11)$$

其中, $E_{new}^*(b)$ 为在第 b 个 Bootstrap 子样本中得到的估算成本值; $E_{new}^*(\cdot)$ 为在全部 Bootstrap 子样本中估

算成本的均值, $E_{new}^*(\cdot) = \frac{\sum_{b=1}^B E_{new}^*(b)}{B}$ 。估算值与 B 个 Bootstrap 子样本中得到的估算均值的偏差为

$$BIAS_{boot} = E_{new}^*(\cdot) - \hat{E}_{new} \quad (12)$$

Angelis 和 Stamelos 提出用 $BIAS_{boot}$ 与 SE_{boot} 的绝对比值衡量估算成本的可信度,即

$$Accuracy = \left| \frac{BIAS_{boot}}{SE_{boot}} \right| \quad (13)$$

并认为其绝对比值小于 0.25 时,估算值 \hat{E}_{new} 较为合理。*Accuracy* 为模型估算结果的可信程度, *Accuracy* 值越大,即多个 B 个 Bootstrap 子样本的估算值相差很大,说明运用该模型实际估算软件成本时,其估算值偏离实际值的可能性较大,说明模型的可信度较小。

为了对估算成本进行验证,还可以通过计算估算成本的置信区间值来提高估算值的可信度,即对估算成本给出一个最高值、一个最低值和一个置信概率。 $(1-a)\% CI_{boot}$ 表示估算成本 \hat{E}_{new} 的一个置信概率为 $(1-a)\%$ 的区间,计算公式为

$$(1-a)\% CI_{boot} = [E_{new,\frac{a}{2}}^*, E_{new,1-\frac{a}{2}}^*] \quad (14)$$

其中, a 为概率值, $E_{new,\frac{a}{2}}^*$ 为所有估算成本值 $E_{new}^*(b)$ 的第 $100 \times \frac{a}{2}$ 分位值, $E_{new,1-\frac{a}{2}}^*$ 为所有估算成本值 $E_{new}^*(b)$ 的第 $100 \times (1 - \frac{a}{2})$ 分位值。计算估算值的置信区间值是为了让决策者对实际成本有个最悲观和最乐观的估算,提高估算成本的参考价值。

6 实证分析

本研究利用 Desharnais 数据库检验已构建模型的有效性。Desharnais 数据库有 10 个属性特征,分别是 TeamExp、ManageExp、YearEnd、Transaction、Entities、PointsAdjust、Envergure、DevEnv、PointsNonAdjust、Effort, 10 个属性都是数值型的^[26]。该数据库有 81 个样本,其中有 4 个样本属性值不全,故删去。本研究采用 Desharnais 数据库中 77 个样本数据,77 个样本中成本的最大值为 23 940,最小值为 546,均值为 4 834,标准差为 4 184。

6.1 PSO 类比法的结果分析和比较

PSO 类比法通过 MATLAB 7.0 编程实现类比计算,PSO 算法采用常用 PSOOTB 工具箱。PSO 算法的参数设置为,粒子数为 30,迭代次数为 500,学习因子 c_1 和 c_2 为 2。表 1 给出两种不同的类比法对 Desharnais 数据的估算结果。在类比项目个数上,给出 3 种不同的选择,即 $k = 1, 2, 3$,评价标准为 *MMRE* 和 *Pred*(0.25) 两个指标。

表 1 中数据显示,在 Desharnais 数据中,PSO 类比法与非加权类比法相比,精确度有了很大提高,表现为更低的 *MMRE* 值和更高的 *Pred*(0.25) 值。其中 PSO 类比中的最优 *MMRE* 结果($k=1$ 时, *MMRE*=0.407) 与非加权类比中最优 *MMRE* 结果($k=2$ 时, *MMRE* = 0.525) 相比, *MMRE* 值改进了约 22%; PSO 类比中的最优 *Pred*(0.25) 结果($k=1$ 时, *Pred*(0.25) = 0.456) 与非加权类比中最优 *Pred*(0.25) 结果($k=2$ 时, *Pred*(0.25) = 0.284) 相比, *Pred*(0.25) 值改进了约 60%。从表 1 中可知,在 Desharnais 数据中,3 种不同 k 值情况下 PSO 类比的 *MMRE* 值要低于非加权类比。当 $k=1$ 时, PSO 类比法得到最小的 *MMRE* 值为 0.407; 当 $k=2$ 时,非加权类比法得到最小的 *MMRE* 值 0.525。表 1 中结果还显示,除了当 $k=2$ 时, PSO 类比的 *Pred*(0.25) 值

略低于非加权类比,其他情况下 PSO 类比的 $Pred(0.25)$ 值都较大程度上高于非加权类比,也是在 $k = 1$ 时取得最优估算精度。综合考虑 $MMRE$ 和 $Pred(0.25)$ 两个指标可以看出,当 $k = 1$ 时,PSO 类比能够得到最佳估算精度。总体看,采用 PSO 算法优化权重的加权类比的准确度得到有效提升,不同 k 值对模型的精确度有较大影响,其中 PSO 类比模型中 $k = 1$ 时能够得到最佳估算精度。

表1 Desharnais 数据中软件成本估算模型的结果比较
Table 1 Comparison of the Software Effort Estimation Models in Desharnais Dataset

模型		$MMRE$	$Pred(0.25)$
非加权类比	$k = 1$	0.712	0.243
	$k = 2$	0.525	0.284
	$k = 3$	0.610	0.077
PSO 类比	$k = 1$	0.407	0.456
	$k = 2$	0.473	0.266
	$k = 3$	0.540	0.224
支持回归机(SVR)		0.400	0.370
人工神经网络(ANN)		0.570	0.220
径向基神经网络(RBF)		0.420	0.370
分类回归树(CART)		0.520	0.300

将 PSO 类比模型中最佳估算精度与其他模型进行比较,发现 PSO 类比模型同样具有较强的优势。与 SVR、ANN、RBF、CART 4 个模型相比,加权模型的最优 $MMRE$ 值要明显小于 ANN 和 CART,与 SVR 和 RBF 差不多。从 $Pred(0.25)$ 标准看,PSO 类比模型的最优 $Pred(0.25)$ 值比其他 4 个模型的 $Pred(0.25)$ 值都大,说明其精确度最高。

比较不同 k 值情况下 PSO 类比法的 $MMRE$ 值和 $Pred(0.25)$ 值,最后得出 $k = 1$ 时估计精度最高。但是这种比较没有涉及到均值、方差等统计变量,结果容易受历史数据的影响。本研究将采用 Bootstrap 检验这一结论,并进一步分析选择多个类比项目情况下不同的成本计算方法对精确度的影响。

6.2 类比项目数和计算方式的优化结果分析

在类比模型中,有两个重要变量影响模型的精确度,分别是类比项目个数 k 和计算类比项目的成本计算方式。理论上说,类比项目个数可以取 $1, 2, \dots, (n - 1)$ (n 为数据库中样本个数)。本研究中 k 值选择 $1, 2, \dots, 20$, 分别采用 Bootstrap 对原始数据进行抽样,运用 PSO 类比方法估算项目成本并计算其 $MMRE$ 值和 $Pred(0.25)$ 值。在计算方式上分别比较均值和中位数对估算精度的影响,采用均值法和中位数法得到不同 k 值的 $MMRE$ 值和 $Pred(0.25)$ 值的箱线图分别见图 1 和图 2。箱线图是利用数据中的最小值、第一四分位数、中位数、第三四分位数和最大值描述数据的一种方法,还可以根据分布识别极端情况下的异常值。在图 1 中,当 k 取任一值时,最上面和最下面的“+”表示是异常值,箱子上端表示第三四分位数、下端表示第一四分位数,箱子中间的线表示中位数,箱子上方的端点表示最大值、下方端点表示最小值。

从图 1 可以看出,在均值计算方式下,随着类比项目个数 k 的不断增加,PSO 类比模型的精确度呈不断下降的趋势,表现为 $MMRE$ 值越来越大,而 $Pred(0.25)$ 值越来越小。

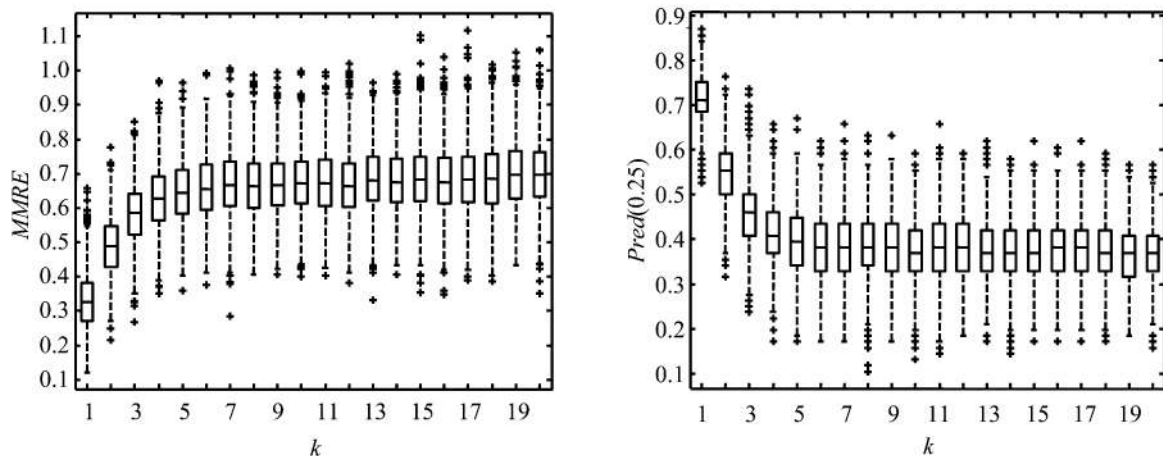


图1 采用均值法不同 k 值的 $MMRE$ 值和 $Pred(0.25)$ 值

Figure 1 Distributions of $MMRE$ and $Pred(0.25)$ for Various Numbers of Analogies by Mean Method

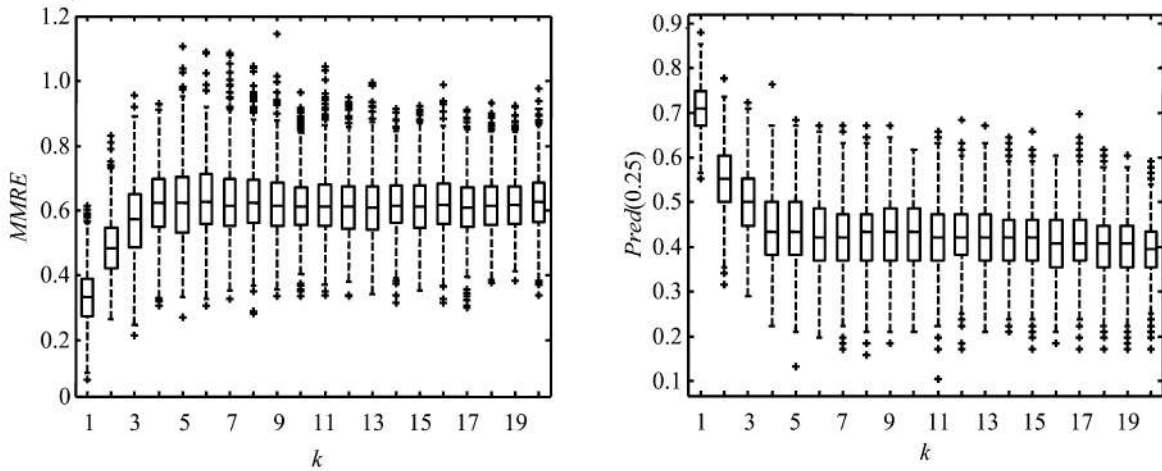


图2 采用中位数法不同 k 值的MMRE 值和Pred(0.25) 值

Figure 2 Distributions of MMRE and Pred(0.25) for Various Numbers of Analogies by Median Method

图2中的结果显示,采用中位数方法计算软件类比项目成本时,模型的精确度也是随着k值的增加而递减。综合图1和图2的结果可以看出,采用PSO方法得到的权重能够很好地刻画不同特征属性之间的相对重要性,在这种情况下,取一个相似度最大的类比项目(k=1)就能很好的反应新项目的情况,而取多个类比项目时(k>1)相似度值较小的类比项目带来的是冗余信息,会降低新项目估算成本的精确度。

6.3 新项目估算成本精确度的结果分析

PSO类比方法只能给出新项目估算成本的具体值,而不能给出置信概率和对应的区间值。本研究采用Bootstrap方法进行抽样,从而得到新项目成本的分布情况以及置信概率和对应的区间值。现假设有一个需要开发的软件项目,其对应的9个属性(TeamExp、ManageExp、YearEnd、Transaction、Entities、PointsAdjust、Envergure、DevEnv、PointsNonAjust)的数值分别是2、3、86、11、168、119、287、27、271。在Desharnais数据中,采用PSO类比方法得到的估算值 $\hat{E}_{new} = 6\,699$ 。采用非参数Bootstrap法以Desharnais数据为基础生成1000个子样本,并用PSO类比方法(类比项目个数为1)估算每一个Bootstrap子样本下新项目的成本。1000个Bootstrap样本下得到的新项目成本的柱状图见图3。

通过对1000个新项目估算成本的计算,其均值和方差分别为 $E_{new}^*(\cdot) = 7\,015$, $SE_{boot} = 2\,668$ 。根据偏差计算公式, $BIAS_{boot} = 316$,从而得到估算值 \hat{E}_{new} 的准确度 $Accuracy = 0.119 < 0.25$ 。根据Angelish和Stamelos的观点,该项目估算值 \hat{E}_{new} 的Accuracy值小于0.25,说明其准确度和可信度较高。

根据置信区间值的定义,可以计算某一概率下估算成本的区间值。在得到的1000个项目估算值中,2.5%(第25个估算值)和97.5%(第975个估算

值)分位值组合成95%置信水平下的区间 $95\% CI_{boot} = [2\,896, 12\,667]$ 。该区间范围可能过大,提供的决策信息有限,一种折中方法是减小置信水平,得到一个相对窄小的区间值,如 $50\% CI_{boot} = [5\,086, 9\,065]$ 。

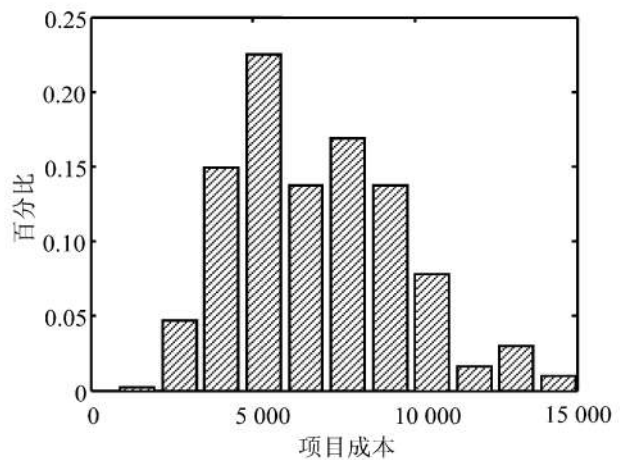


图3 基于1000个Bootstrap样本的估算成本柱状图

Figure 3 Histogram of the Effort Estimations Obtained from 1000 Bootstrap Samples

通过上述实证分析结果可以看出,本研究提出的PSO类比法能有效刻画不同特征属性之间的相对重要性,较大程度上提高软件成本的估算精度。而通过Bootstrap抽样方法对PSO类比法中的两个参数(类比项目个数k和类比项目的成本计算方式)进行校正,能有效描述不同的类比项目个数和类比项目的成本计算方式对最终结果的影响。采用Bootstrap方法得到新项目成本的分布情况以及估算成本的置信概率和对应的区间值,可以为管理者在规划分析项目时提供有益参考。

7 结论

本研究提出用 PSO 优化权重的加权类比估算方法估算软件项目的成本,并运用 Bootstrap 在原始数据基础上进行抽样,以校正 PSO 类比模型中的类比项目个数和成本计算方式这两个变量,同时计算新项目估算成本的可信度和置信区间值。在 Desharnais 数据中的应用结果显示,PSO 类比方法的估算精度明显高于非加权类比方法,并优于一些其他常用的估算模型,表现在评价指标上就是更低的 *MMRE* 值和更高的 *Pred(0.25)* 值。Bootstrap 方法的校正结果表明,在类比项目个数为 1 的情况下可以得到最优估算精度。说明 PSO 类比方法能很好地剔除属性之间的冗余信息,用一个类比项目($k=1$)的成本作为新项目成本的估算值就能够得到满意的结果。本研究运用 Bootstrap 得到新项目估算成本的可信度和置信区间,可以为软件项目风险分析和项目规划提供有益参考。

参考文献:

- [1] Amoroso E, Taylor C, Watson J, Weiss J. A Process-Oriented Methodology for Assessing and Improving Software Trustworthiness [C] // Proceedings of the 2nd ACM Conference on Computer and Communications Security. Virginia, 1994:39-50.
- [2] Boehm B W, Papaccio P N. Understanding and Controlling Software Costs [J]. IEEE Transaction on Software Engineering, 1988, 14(10):1462-1477.
- [3] The Standish Group. CHAOS Summary 2009[EB/OL]. http://www.standishgroup.com/newsroom/chaos_2009.php, 2009.
- [4] Bannerman P L. Risk and Risk Management in Software Projects: A Reassessment [J]. Journal of Systems and Software, 2008, 81(12):2118-2133.
- [5] Magne J, Martin S. A Systematic Review of Software Development Cost Estimation Studies [J]. IEEE Transactions on Software Engineering, 2007, 33(1):33-53.
- [6] Boehm B W, Valerdi R. Achievements and Challenges in Cocomo-Based Software Resource Estimation [J]. IEEE Software, 2008, 25(5):74-83.
- [7] Jorgensen M. Practical Guidelines for Expert-Judgment-Based Software Effort Estimation [J]. IEEE Software, 2005, 22(3):57-63.
- [8] Molokken-Ostfold K, Haugen N C, Benestad H C. Using Planning Poker for Combining Expert Estimates in Software Projects [J]. Journal of Systems and Software, 2008, 81(12):2106-2117.
- [9] Koch S, Mitlohner J. Software Project Effort Estimation with Voting Rules [J]. Decision Support Systems, 2009, 46(4):895-901.
- [10] 张俊光, 吕廷杰. 基于规模的软件工作量估计方法研究[J]. 中国管理科学, 2008, 16(5):118-122.
- [11] Zhang J G, Lv T J. Estimation Method of Software Project Effort Based on Software Size [J]. Chinese Journal of Management Science, 2008, 16(5):118-122. (in Chinese)
- [12] 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.
- [13] Sentas P, Angelis L, Stamelos I, Bleris G. Software Productivity and Effort Prediction with Ordinal Regression [J]. Information and Software Technology, 2005, 47(1):17-29.
- [14] Kumar K V, Ravi V, Carr M, Raj Kiran N. Software Development Cost Estimation Using Wavelet Neural Networks [J]. Journal of Systems and Software, 2008, 81(11):1853-1867.
- [15] Huang S J, Chiu N H. Applying Fuzzy Neural Network to Estimate Software Development Effort [J]. Applied Intelligence, 2009, 30(2):73-83.
- [16] Oliveira A L I. Estimation of Software Project Effort with Support Vector Regression [J]. Neurocomputing, 2006, 69(13-15):1749-1753.
- [17] Mittasa N, Athanasiadesa M, Angelis L. Improving Analogy-Based Software Cost Estimation by a Resampling Method [J]. Information and Software Technology, 2008, 50(3):221-230.
- [18] 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.
- [19] Huang J S, Chiu N H. Optimization of Analogy Weights by Genetic Algorithm for Software Effort Estimation [J]. Information and Software Technology, 2006, 48(11):1034-1045.
- [20] Chiu N H. An Early Software-Quality Classification Based on Improved Grey Relational Classifier [J]. Expert Systems with Applications, 2009, 36(7):10727-10734.
- [21] Maxwell K, Wassenhove L V, Dutta S. Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation [J]. Management Science, 1999, 45(6):787-803.
- [22] 肖智, 吴慰. 基于 PSO-PLS 的组合预测方法在 GDP 预测中的应用 [J]. 管理科学, 2008, 21(3):115-121.
- Xiao Z, Wu W. The Application of Combining Forecasting Based on PSO-PLS to GDP [J]. Journal of Management Science, 2008, 21(3):115-121. (in Chinese)
- [22] Medeiros C C, Schirru R. Identification of Nuclear Power Plant Transients Using the Particle Swarm Optimization Algorithm [J]. Annals of Nuclear Energy,

- 2008,35(4):576-582.
- [23] Efron B. Bootstrap Methods: Another Look at the Jack-knife[J]. The Annals of Statistics,1979,7(1):1-26.
- [24] Angelis L, Stamelos I. A Simulation Tool for Efficient Analogy Based Cost Estimation [J]. Empirical Software Engineering, 2000,5(1):35-68.
- [25] Stamelos I, Angelis L. Managing Uncertainty in Project Portfolio Cost Estimation [J]. Information and Software Technology, 2001,43(13):759-768.
- [26] Li Y F, Xie M, Goh T N. A Study of Project Selection and Feature Weighting for Analogy Based Software Cost Estimation [J]. Journal of Systems and Software, 2009,82(2):241-252.

Analogy Estimation Based on Particle Swarm Optimization and Bootstrap Inference for Software Effort

WU Deng-sheng^{1,2}, LI Jian-ping¹, CAI Chen¹

1 Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China

2 Graduate University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: Wide attention has been attracted on software effort estimation by both software industry and academic community owing to its high ability in controlling software schedule, reducing software risk and guaranteeing software quality. This paper investigates the improvement effects of estimation accuracy in analogy-based model when particle swarm optimization method is adopted to optimize the feature weights. Meanwhile, nonparametric bootstrap was employed to generate samples for calibrating the number of the most similar projects and the project adaptation in PSO analogy model. Reliability of estimated value and confidence interval were calculated by nonparametric bootstrap too. Experiments were carried out using software projects from Desharnais dataset in order to verify the effectiveness of PSO analogy model and bootstrap inference method. Estimation accuracy of the PSO analogy model was compared with ordinary Analogy, SVR, ANN, RBF and CART in terms of the error measure which is *MMRE* and *Pred(0.25)*. The empirical results show that applying particle swarm optimization method to optimize the feature weights is a feasible approach to improve the accuracy of software effort estimation. Moreover, nonparametric bootstrap can calibrate the variables in PSO analogy model and calculate estimation accuracy and confidence interval effectively. Results of the proposed model are beneficial to assess estimation accuracy, analyze risk and plan project.

Keywords: trustworthy software; software effort estimation; analogy method; particle swarm optimization; bootstrap

Received Date: December 13th, 2009 **Accepted Date:** March 29th, 2010

Funded Project: Supported by the National Natural Science Foundation of China(90718042,70531040)

Biography: WU Deng-sheng, an Anhui Lujiang native(1984 -), is a Ph. D. candidate in the Institute of Policy and Management at Chinese Academy of Sciences. His research interests include risk management and software engineering, etc. E-mail:wds@casipm.ac.cn □