文章编号:1003－207(2013)04－0178－09

基于广义模糊数的软件成本加权 CBR 估算研究

吴登生

(中国科学院科技政策与管理科学研究所,北京 100190)

摘 要:在软件项目开发过程中,准确估算出软件成本在提高软件质量和保障软件成功开发方面起到重要支撑作用。针对软件项目历史数据库中部分属性在项目开发初期难以给予精确数值(仅仅能给出模糊数),而已有软件成本估算模型不能很好地处理模糊信息的问题,本文在基于案例推理模型(CBR)基础上集成广义模糊数,提出了基于广义模糊数的 CBR 模型。使用基于广义模糊数的相似度度量方法代替传统 CBR 模型中采用的欧式距离等相似度度量方法,采用模糊 C 均值聚类(FCM)方法将已有软件项目历史数据库中的精确数值进行模糊化处理,以匹配新项目中的模糊数。进一步采用粒子群算法(PSO)来优化属性的权重,构建基于广义模糊数的加权 CBR 模型。最终在实验中采用 Desharnais 数据来检验构建模型的有效性。实证结果表明,在与常用的欧式距离 CBR 模型相比,构建的基于广义模糊数的加权 CBR 模型能有效提高估算精度,采用 PSO 优化属性权重能提高模型的估算精度。

关键词:软件成本估算;基于案例推理;广义模糊数;权重优化

中图分类号:TP311.5 **文献标识码**:A

1 引言

作为软件过程中的一个重要环节,软件成本估算是指在软件规划阶段,根据软件项目的相关属性特征,对其成本进行预测。成功地进行软件成本估算,做出科学的项目决策,不仅影响着软件项目的成败,很大程度上也影响着企业的成败。准确地估算软件成本在控制软件成本,提高软件质量,保障软件成功开发方面起到重要支撑作用,正如 Boehm 教授指出的"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"。软件项目的动态开发环境、需求多变、人力知识密集等特殊性加大了软件成本估算的难度,使得软件成本估算成为软件项目管理中最具挑战性的活动之一。

鉴于软件成本估算的重要意义和实际估算中存在的诸多问题,不断有新的软件成本估算模型被提出。总结软件成本估算模型,可以将其分为数据驱动和专家驱动两类。专家驱动方法是指利用专家经验、在借助其他方法和信息的基础上,完成对新项目的成本估算。早期常见的专家判定技术有 Delphi 技术和工作分解结构技术[1]。近年来,也有许多研究采用专家判断的方法来估算软件成本,主要集中在如何集结多个专家的意见,如 Molokken-Ostvold 等人用规划扑克的方法集结专家的意见[2],Koch 等人在专家判断的基础上运用选举理论来估算软件开发成本[3]。当仅有的可用信息只能依赖专家意见而非确切的经验数据时,专家方法无疑是解决成本估算问题的最直接的选择。但是这种方法过于依赖专家的经验,有时会带来较大的估算偏差。数据驱动方法是指在历史数据的基础上,采用相关算法分析数据的规律,进一步估算新项目的成本。数据驱动方法主要有 COCOMO[4]、基于案例推理模型(CBR)[5]、神经网络[6],支持向量回归机(SVR)[7]、顺序回归[8]、多重累计回归树[9]等。当有较多的历史项目数据时,数据驱动方法能有效刻画软件成本与相关属性之间的关系,得到较高的估算精度,故在研究和实践过程中被广泛采用。

数据驱动方法在估算软件成本时需要采用历史项目数据库来训练和校正模型,得到软件成本与相关属性之间的关系,进而对新软件项目的成本值进行估算。现有的软件项目数据库都是对已完成项目的描述,相关属性都是准确数字,而对于一个新开发

收稿日期:2012－07－30; **修订日期**:2013－04－30

基金项目:国家自然科学基金资助项目(71201156,91218302,70531040);中国科学院青年创新促进会基金项目

作者简介:吴登生(1984—),男(汉族),安徽庐江人,中国科学院科技政策与管理科学研究所助理研究员,博士,研究方向:风险管理.

的软件项目来说,数据库中相关属性在项目早期难以确定精确值,这就制约了数据驱动方法的适用阶段。如常用的 ISBSG 数据库[10]、Maxwell 数据库[11]和 Desharnais 数据库[12],其最重要的相关属性都是软件规模属性,一般用功能点或代码行来衡量,这两个属性在软件开发项目早期都难以给出一个精确数字。而已有数据驱动类软件成本估算模型大部分都不能处理"非精确值"。有学者提出用模糊数来描述新项目的相关属性值,对已有数据驱动方法进行适当的修改,进而估算新项目的成本[13-14],为解决这类问题提供了很好地思路。但是现有成果还存在许多问题,如能刻画属性之间模糊信息的相似度度量方法还有待进一步研究,提出的集成广义模糊数方法没有涉及到权重优化等问题。

针对这些问题,本文选择数据驱动方法中应用最为广泛的基于案例推理模型(Case Based Reasoning,CBR)来集成广义模糊数,构建基于广义模糊数的 CBR 模型(GFN-CBR),利用广义模糊数来刻画软件成本数据中的模糊特征。CBR 模型可理解性强,比较直观,符合人们解决问题的常规思维,尤其是当决策者关注软件成本估算过程时,CBR 模型的过程透明性使其具有明显的优势。Heemstra[15]对近 600 个软件组织进行调查,结果显示 CBR 模型是一个广泛应用的软件成本估算方法。Mittas 和 Angelis[16]以及 Dejaeger 和 Vevbere[17]在最近的软件成本估算研究综述性论文中,也将 CBR 方法视为一种重要的估算方法。在 GFN-CBR 模型中,新软件项目的相关属性值直接用广义模糊数给出,历史数据库中相关属性的准确值用模糊 C 均值聚类(FCM)方法将其转化为广义模糊数,通过构建基于广义模糊数的相似度度量标准来判断新旧项目的相似程度,进而对新项目成本值进行估算。进一步利用粒子群算法(Particle Swarm Optimization,PSO)来优化属性权重,构建基于广义模糊数的加权 CBR 模型(GFN-PSO-CBR),提高模型的估算精度。

2　软件成本估算的 CBR 模型

软件成本估算 CBR 模型的基本思想就是在历史项目数据库中,通过相似度度量方法找到一个或多个与新项目最为相似的历史项目,以这些项目的成本值为依据,估算新项目的成本值[18]。CBR 模型的基本假设就是属性越相似的项目,其成本值也是越相似。CBR 模型中通常采用欧式距离作为衡量项目之间相似度的标准[19-20]。Huang 等[21]和 Azzeh 等[22]改进了原有的方法,采用灰色关联度度量项目之间的相似度。Wu 等[12]进一步总结了 CBR 模型常用的 6 种相似度计算公式,构建了 6 类软件成本估算 CBR 模型。此外,现有的研究成果采用启发式算法优化 CBR 模型的属性权重,用以提高模型的估算精度。Chiu 等[20]采用遗传算法优化属性权重,吴登生等[23]采用粒子群算法在估算过程中对属性权重进行优化,实证结果表明采用启发式算法优化属性权重能显著提高模型的估算精度。已有的研究中,一般将 CBR 模型分成相似度度量、项目拟合数目与成本拟合方式、模型评价标准三个部分[23-25]。

在运用 CBR 模型来估算软件成本时,首先要选用一种相似度度量标准来衡量新旧项目之间的相似程度。已有研究中最常用的是采用欧式距离来衡量相似程度的高低,距离值越大表示相似度越低[20-21]。现假设历史数据库中每个项目有 p 个属性,$x_i(k)$ 表示历史项目 x_i 在第 k 个属性上的取值,$x_0(k)$ 表示新项目 x_0 在第 k 个属性的值,新项目 x_0 和历史项目 x_i 的距离如公式(1)和公式(2)所示。

$$Dis(x_0(k),x_i(k)) = w_i \times (x_0(k) - x_i(k))^2 \tag{1}$$

$$D(x_0,x_i) = \sqrt{\sum_{k=1}^{p} Dis(x_0(k),x_i(k))} \tag{2}$$

其中 $Dis(x_0(k),x_i(k))$ 表示历史项目 x_i 和新项目 x_0 在第 k 个属性上距离,$D(x_0,x_i)$ 表示历史项目 x_i 和新项目 x_0 之间的距离,w_i 是第 i 个属性的权重,可以通过相关优化算法计算得到最优值。

当确定新旧项目之间的相似度计算方式后,就可以计算新项目和历史项目数据库中每个项目之间的相似度(距离越大,相似度越低),进一步需要明确用来进行拟合的项目数以及相应的拟合方式。当确定只选择一个最为相似的历史项目作为估算新项目成本依据时,只需要将该历史项目作为新项目成本的估算值即可。已有研究表明,采用多个项目作为估算新项目成本估算依据会提高估算精度,通常建议用来拟合的项目数为 1-3 个[18]。当选择多个历史项目作为估算新项目成本的依据时,还需要确定合适的拟合方式来拟合多个历史项目的成本值,从而得到新项目的成本估算值。已有成果中采用多个项目成本的均值(Mean)、中位数(Median)和加权均值(Weighting Mean,WM)三种方式作为新项目成本的估算值[23],具体见公式(3)至公式(5),

$$\hat{E}_{new}(mean) = \text{mean}(E_1,\cdots,E_m) \tag{3}$$

$$\hat{E}_{new}(median) = \text{median}(E_1, \cdots, E_m) \tag{4}$$

$$\hat{E}_{new}(weighing\ mean) = \sum_{i=1}^{m}\left(\frac{D(x_0, x_i)}{\sum_{i=1}^{m} D(x_0, x_i)} E_i\right) \tag{5}$$

其中 $\hat{E}_{new}$ 表示新项目成本的最终估算值，E_1，…，E_m 表示最为相似的 m 个历史项目的成本值。

当通过上述方式得到新项目成本的估算值时，还需要建立合适的标准用来评价模型估计结果的精度。在软件成本估算研究领域，在评价模型的估算精度时，通常是基于历史项目计算出来。首先将一个历史项目假设成新项目，根据其相关属性值（去除成本属性）从历史数据库中找到最为相似的项目进行成本拟合，得到成本估算值，进一步用估算值和实际值进行对比，来刻画模型的估算能力。定义 MRE_i 作为数据集中第 i 个项目成本估算值与实际值之间的误差，具体见公式(6)。

$$\text{MRE}_i = \frac{|E_i - \hat{E}_i|}{E_i} \tag{6}$$

其中 $\hat{E}_i$ 为第 i 个项目估算成本值，E_i 为第 i 个项目实际成本值。在 MRE 的基础上，可以定义平均相对误差（Mean Magnitude Related Error，MMRE）、中位数相对误差（Median Magnitude of Relative Error，MdMRE）与 Pred(0.25)三个标准来评价软件成本估算模型的精度[20-21]，具体如公式(7)至公式(9)所示。

$$\text{MMRE} = \frac{1}{n}\sum_{i=1}^{n} \text{MRE}_i \tag{7}$$

$$\text{MdMRE} = median(\text{MRE}_i) \tag{8}$$

$$\text{Pred}(0.25) = \frac{l}{n} \tag{9}$$

上述公式中，l 为数据集中相对误差值小于 0.25（$\text{MRE} < 0.25$）的项目个数，n 为该数据集中所有项目的总数。

3 基于广义模糊数的加权 CBR 模型

已有的软件成本 CBR 估算模型无法刻画属性之间的模糊特征，制约了模型的应用范围。本文运用广义模糊数刻画不同属性之间的模糊特征，构建基于广义模糊数的 CBR 模型（GFN-CBR），使用基于广义模糊数的相似度度量方法代替传统 CBR 模型中采用的欧式距离等相似度度量方法。为了匹配新项目中的模糊数，采用模糊 C 均值聚类（FCM）方法将已有软件项目历史数据库中的精确数字进行模糊化处理。进一步采用粒子群算法来优化不同属性的权重，建立基于广义模糊数的加权 CBR 模型（GFN-PSO-CBR）用来估算软件成本，提高模型的估算精度。

3.1 广义模糊数基本概念

广义模糊数是定义在实数 R 上的一个模糊集，可以表示为 $A = (a,b,c,d;w)$，其中 $0 < w \leqslant 1$，a,b,c,d 为实数，广义模糊数的隶属度函数 μ_λ 需要满足以下条件[26-27]：

(1) μ_λ 定义为从 R 到闭区间 $[0,1]$ 上连续映射；

(2) $\mu_\lambda(x) = 0$，当 $-\infty < x \leqslant a$；

(3) $\mu_\lambda(x)$ 在 $[a,b]$ 是严格递增的；

(4) $\mu_\lambda(x) = w$，当 $b \leqslant x \leqslant c$；

(5) $\mu_\lambda(x)$ 在 $[c,d]$ 是严格递减的；

(6) $\mu_\lambda(x) = 0$，当 $d \leqslant x < \infty$。

广义模糊数是模糊数的一般表现形式，常用的正规梯形模糊数、区间数、实数、广义三角模糊数是其特殊形式，如当 $b = c$ 时广义模糊数演变为广义三角模糊数。

3.2 基于广义模糊数的相似度度量

在上述广义模糊数基本概念定义的基础上，给出计算两个广义模糊数相似度的公式，刻画属性之间的模糊特性，作为软件成本估算过程中衡量新旧项目相似度依据。对于两个广义模糊数 $A = (a_1, a_2, a_3, a_4; w_A)$ 和 $B = (b_1, b_2, b_3, b_4; w_B)$，其中 $0 \leqslant a_1 \leqslant a_2 \leqslant a_3 \leqslant a_4 \leqslant 1$，$0 \leqslant b_1 \leqslant b_2 \leqslant b_3 \leqslant b_4 \leqslant 1$。根据已有研究成果的定义，两个广义模糊数之间的相似度要考虑高度调整比率（Height Adjustment Ratio，HAR）、几何距离（Geometric Distance，GD）和形状调整因子（Shape Adjustment Factor，SAF）[14, 28-29]，具体如公式(10)所示。

$$S(A,B) = HAR \times \frac{1 - GD}{SAF} \tag{10}$$

其中 $S(A,B)$ 定义为广义模糊数 A 和 B 之间的相似度。

HAR 用来衡量两个广义模糊数的在高度上差异，并用来调整其相似度。当 w_A 与 w_B 差异越大时，说明广义模糊数 A 和 B 的相似度应该越小，此时用 $\min(w_A/w_B, w_B/w_A)$ 作为调整因子。进一步为了降低其敏感程度，对调整因子开平方，得到 HAR，具体见公式(11)。

$$HAR = \sqrt{\min(w_A/w_B, w_B/w_A)} \tag{11}$$

广义模糊数 A 和 B 的重心定义为 (x_A, y_A) 和 (x_B, y_B),公式(12)和公式(13)给出计算广义模糊数 A 重心 (x_A, y_A) 的过程,广义模糊数 B 重心的计算类似。在此基础上定义 GD 用来衡量广义模糊数 A 和 B 的几何距离,具体如公式(14)所示。

$$y_A = \begin{cases} \dfrac{w_A \times ((a_3 - a_2)/(a_4 - a_1) + 2)}{6} & \text{if } a_4 \neq a_1 \\ \dfrac{w_A}{2} & \text{if } a_4 = a_1 \end{cases} \tag{12}$$

$$x_A = \frac{y_A(a_3 + a_2) + (a_4 + a_1)(w_A - y_A)}{2w_A} \tag{13}$$

$$GD = \frac{1}{5}\left(\sum_{i=1}^{4} | a_i - b_i | + | x_A - x_B |\right) \tag{14}$$

在广义模糊数重心的基础上,进一步定义 SAF 来作为几何距离的调整因子,具体如公式(15)所示。

$$SAF = 1 + | y_A - y_B | \tag{15}$$

3.3　数据驱动的广义模糊数构建

在利用基于广义模糊数相似度度量方法计算出新旧项目之间的相似度时,需要事先确定项目在不同属性上的模糊数。已有的软件成本数据库中通常都是数值型或名义型数据,需要采用合适的方法进行模糊化,得到相应的模糊数。已有确定模糊数方法可以分为专家驱动和数据驱动两种,前者基于主观判断,具有一定的随意性,后者基于数据自身的特征,客观性较强。本文采用数据驱动的方法来构建软件项目属性的广义模糊数。在保证模型刻画模糊属性能力的同时,为了减少模型的复杂度,本文采用广义模糊数的一种特殊形式广义三角模糊数来刻画属性的模糊特征。属性值的模糊化过程中通过模糊 C 均值聚类(FCM)方法来计算。

FCM 算法是一种基于划分的聚类算法,它的思想就是使得被划分到同一簇的对象之间相似度最大,而不同簇之间的相似度最小。在进行 FCM 聚类时,确定的类数对聚类效果会有一定的影响,所以需要确定最优类数。本文采用 Xie 等[30]提出的 S 指数来选择 FCM 的最优类数,具体如公式(16)所示。

$$S = \frac{\sum_{k=1}^{c}\sum_{i=1}^{n} (u_{ki})^2 \| C_k - X_i \|}{n \times \min_{k,l} \| C_k - C_l \|} \tag{16}$$

式中 c 是模糊聚类的类数,C_k 是第 k 个聚类中心点向量,X_i 为第 i 个样本值,U 为模糊划分矩阵,u_{ki} 为第 i 个样本属于第 k 类的隶属度,$\| \ \|$ 表示欧式距离。指数 S 越小,表示模糊聚类的效果越好。

根据 S 指数确定最优类数后,进一步计算第 i 个样本在第 j 个属性上取值 $x_i(j)$ 的伸展值 σ_{ij},具体如公式(17)所示。

$$\sigma_{ij} = \frac{1}{c}\sum_{k=1}^{c} u_k(x_i(j)) \times \frac{x_i(j) \times \delta_{kj}}{C_{kj}} \tag{17}$$

式中 c 是模糊聚类的类数,$u_k(x_i(j))$ 表示 $x_i(j)$ 属于第 k 类的隶属度,δ_{kj} 和 C_{kj} 分别是第 j 个属性在第 k 类上的扩展值和中心值。根据计算得到的伸展值 σ_{ij},可以得到每一个属性值 $x_i(j)$ 对应的模糊数 $\tilde{x}_i(j)$,具体如公式(18)所示[14]。

$$\tilde{x}_i(j) = \left(x_i(j) - \frac{\sigma_{ij}}{2}, x_i(j), x_i(j) + \frac{\sigma_{ij}}{2}; 1\right) \tag{18}$$

3.4　基于粒子群算法的属性权重优化

属性的权重优化是基于广义模糊数加权 CBR 模型中的一个重要部分,已有研究中都是将各属性进行等权重处理,一定程度上影响了模型的估算精度。在传统的软件成本 CBR 估算模型研究领域,已有相关研究成果采用优化算法(如遗传算法、粒子群算法)来优化属性的权重,取得较好的结果[12, 20]。本文采用粒子群算法(Particle Swarm Optimization,PSO)来优化基于广义模糊数加权 CBR 模型中属性权重。PSO 算法优化权重时避免了复杂的遗传步骤,保留了种群的全局搜索策略,已被证实在权重优化方面效果明显[12]。现假设有历史数据库中有 n 个项目,每个项目有 p 个属性,$\tilde{x}_0(k)$ 表示新项目在第 k 个属性上的模糊数,$\tilde{x}_i(k)$ 表示第 i 历史项目在第 k 个属性上的模糊数。公式 19 为两个项目的加权相似度计算方法,单个属性上的相似度 $S(\tilde{x}_0(k), \tilde{x}_i(k))$ 计算过程如公式(10)至公式(15)所示。

$$S(\tilde{x}_0, \tilde{x}_i) = \sum_{k=1}^{p} w_k \times S(\tilde{x}_0(k), \tilde{x}_i(k)) \tag{19}$$

在采用 PSO 算法优化广义模糊数的加权 CBR 模型中属性权重 w_k 时,其基本思路是将属性权重编码成 PSO 中的一个粒子,通过更新粒子位置信息和速度信息,进行粒子迭代,最终找到最优权重。权重优化过程中采用留一交叉法来验证构建模型的有效

性。此外，采用评价标准 MMRE 作为 PSO 优化权重过程中的适应度函数。具体步骤如下[23]：

步骤 1：初始化 PSO 算法模块参数(种群规模、惯性权重等)，并随机分配一组位置信息和速度信息给每个粒子；

步骤 2：迭代次数 $ite = 0$；

步骤 3：用每个粒子作为加权 CBR 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，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 $ite = ite + 1$；

步骤 6：确定 pbest 和 gbest，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，用得到的权重估算新软件项目的成本。

4 案例分析

在软件成本估算领域，已有许多历史数据库用于验证构建模型的有效性。本文采用较为常用的 Desharnais 数据库作为样本数据来检验已构建模型的有效性，该数据库一共有 10 个属性特征，具体属性特征见表 1[18]。该数据库有 81 个样本，其中有四个样本属性值不全，故删去。77 个样本中成本最大值是 23940，最小值是 546，均值是 4834，标准差是 4184。

表 1 中 Desharnais 数据的相关属性是软件成本估算领域常用的属性，该数据库也被多个研究成果用来作为验证模型有效性的依据[18-21]。“Desharnais”数据的相关属性基本涵盖了软件成本估算领域考虑的要素，结果具有较强的可信性。对于软件企业来说，在估算一个新项目成本时，需要根据表 1 中的属性，结合项目实际情况，对项目属性(Effort 属性除外)进行赋值。一般是在项目初期就要进行软件成本估算，而此时相关属性无法给出精确值，如表 1 中的 YearEnd、Length、Transactions、Entities、PointsNonAjust、PointsAdjust、Envergure 等属性，只能给出一个模糊数。现假设对于一个新项目，其 Transactions 属性(事务功能点数)值为 90(事后确认)，在项目初期，软件企业在进行分析时认为该值在 80－100 之间，但是无法给出精确值，用模糊数 (80,90,95,100;1) 能够很好刻画这种不确定性。

为了消除不同属性之间量纲的差异，首先对数据库的各属性值进行归一化处理。基于广义模糊数的 CBR 模型和 FCM 数值模糊化处理都是采用 MATLAB 编程计算。PSO 算法采用基于 MATLAB 的 PSOTB 工具箱，相关参数设置为：迭代次数为 500，粒子群规模为 30，学习因子 c_1 和 c_2 分别为 2，w_{max} 和 w_{min} 值分别设置为 0.90 和 0.45。带 PSO 参数优化的模型采用三折交叉验证的方法检验模型的效果，即将样本随机分成三份，取其中的两份作为训练集，剩下的一份作为测试集。

表 2 给出了基于广义模糊数 CBR 模型(GFN-CBR)和基于广义模糊数加权 CBR 模型(GFN-PSO-CBR)在不同拟合项目数和数值拟合方式组合下估算精度，进一步分析采用 PSO 优化权重后，对 GFN-CBR 模型估算精度的改进度。

表 1 Desharnais 数据的属性特征

序号	属性名称	释义	数值化方法
1	TeamExp	团队经验	用年衡量
2	ManagerExp	项目管理者经验	用年衡量
3	YearEnd	项目完成年份	实际完成年份
4	Length	项目持续时间	用月衡量
5	Transactions	事务功能点数	个数
6	Entities	实体功能点数	个数
7	PointsNonAjust	未调整功能点数(事务功能点＋实体功能点)	个数
8	PointsAdjust	调整功能点数(根据 PointsNonAjust 调整)	个数
9	Envergure	项目复杂因子(调整项目复杂程度)	取值[0，100]
10	Effort	软件成本(用人时来衡量)	人时

表 2　基于广义模糊数 CBR 模型的软件成本估算精度

模型	拟合项目数	数值拟合方式	MMRE		MdMRE		Pred(0.25)	
			训练集	测试集	训练集	测试集	训练集	测试集
GFN-CBR	1	CP		0.868		0.472		0.221
	2	Mean		0.832		0.470		0.182
		WM		0.566		0.488		0.351
	3	Mean		0.818		0.489		0.260
		WM		0.586		0.612		0.104
		Median		0.838		0.492		0.260
PSO-GFN-CBR	1	CP	0.565	0.717	0.389	0.573	0.312	0.247
	2	Mean	0.517	0.779	0.432	0.472	0.286	0.273
		WM	0.459	0.511	0.316	0.422	0.429	0.343
	3	Mean	0.570	0.730	0.322	0.566	0.363	0.284
		WM	0.512	0.692	0.494	0.556	0.228	0.241
		Median	0.482	0.658	0.353	0.463	0.376	0.312

注:GFN-CBR 算法无需采用交叉验证方法验证,故只有测试集结果数据。

表 3　基于广义模糊数 CBR 模型与其他 CBR 模型估算精度对比分析

模型	拟合项目数	数值拟合方式	MMRE		MdMRE		Pred(0.25)	
			训练集	测试集	训练集	测试集	训练集	测试集
EUC-CBR	2	WM		0.598		0.575		0.169
GFN-CBR	2	WM		0.566		0.488		0.182
PSO-EUC-CBR	3	Mean	0.548	0.538	0.287	0.469	0.435	0.339
PSO-GFN-CBR	2	WM	0.459	0.511	0.316	0.422	0.429	0.343

注:EUC-CBR 与 GFN-CBR 算法无需采用交叉验证方法验证,故只有测试集结果数据。

表 2 中的结果显示,不同的参数组合对模型的估算结果影响较大,GFN-CBR 模型在拟合项目数为 2、数值拟合方式为 WM 时能取得最优估算结果。对于 PSO-GFN-CBR 模型来说,也是在拟合项目数为 2、数值拟合方式为 WM 时取得最优估算结果。对比两个模型的最优估算结果可以发现 PSO-GFN-CBR 模型在 MMRE 和 MdMRE 两个指标上,要优于 GFN-CBR 模型,在 Pred(0.25)指标上略逊于 GFN-CBR 模型。结果表明采用 PSO 算法优化属性权重在一定程度上提高了模型的估算精度。此外,PSO-GFN-CBR 模型的估算结果中,不同参数集对模型估算精度的影响没有 GFN-CBR 模型中明显,说明采用 PSO 算法优化属性权重的模型能降低不同参数集结果的波动幅度。

为了说明本文提出的模型的有效性,将构建的 GFN-CBR 和 PSO-GFN-CBR 模型与常用的基于欧式距离的 CBR 模型(EUC-CBR)和基于欧式距离的加权 CBR 模型(PSO-EUC-CBR)就估算结果的精度进行对比,具体见表 3。

表 3 显示了四种 CBR 模型在估算精度上的差异,对比结果显示 GFN-CBR 模型估算结果的精度要优于 EUC-CBR 模型,说明采用广义模糊数的 CBR 模型能有效刻画属性之间的模糊特征,提高了模型的估算精度。此外,表 3 中的结果还显示两个采用 PSO 算法优化属性权重 CBR 模型的结果要明显优于没有采用权重优化 CBR 模型的估算结果,说明 PSO 优化属性权重能有效提高模型的估算精度。最后对比两个采用 PSO 算法优化属性权重 CBR 模型的结果,PSO-GFN-CBR 模型的结果在一定程度上要优于 PSO-EUC-CBR 模型。为了更好地的对比不同模型的之间估算精度的差异,图 1 将四个模型在测试集中不同评价标准下估算精度进行了对比分析。

从图 1 中可以明显看出,从 MMRE 和 MdMRE 两个评价指标来看,采用广义模糊数的 CBR 模型要优于采用欧式距离的 CBR 模型,采用 PSO 优化权重的 CBR 模型要优于没有进行权重优化的模型,PSO-GFN-CBR 模型的估算精度最高。从 Pred(0.25)评价指标来看,PSO-GFN-CBR 模型要略逊于 PSO-EUC-CBR 模型。

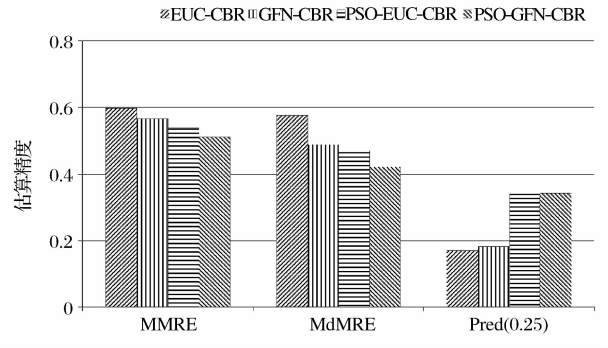

图 1 不同模型在测试集中估算精度对比情况

综合上述分析来看，本文构建的 PSO-GFN-CBR 模型一定程度上提高了模型的估算精度。虽然部分指标上没有占优，但是 PSO-GFN-CBR 模型能够处理软件项目数据库中的模糊信息，这对于在软件项目初期进行成本估算来说，显得尤为重要，一定程度上拓宽了软件成本估算模型的应用阶段，对于指导软件开发实践有着重要意义。此外，在软件成本估算过程中由于多种原因，会使得估算结果出现一定的偏差[31]。对于软件企业来说，如何在软件成本估算结果的基础上，对其偏差进行有效控制也是一个重要议题。在进一步研究中，需要将已有的软件成本估算成果和偏差控制进行有机集合，为更好地指导软件开发实践提供有益支撑。

5 结语

准确地估算软件成本对于软件项目来说至关重要，针对软件成本估算实践中无法刻画属性模糊特征的问题，本文将广义模糊数和 CBR 模型进行有效集成，利用已有的广义模糊数相似度度量标准，构建了基于广义模糊数的软件成本 CBR 模型，并采用 FCM 方法对原有数据进行模糊化处理，进一步采用 PSO 算法来优化属性的权重，形成基于广义模糊数的加权 CBR 模型。为了验证构建模型的有效性，采用 Desharnais 数据作为样本集进行实证分析。实证结果显示，与常用欧式距离 CBR 相比，集成广义模糊数 CBR 模型能够有效利用软件成本数据中的模糊特征，一定程度上能够取得更高的估算精度。此外，实证结果还表明，采用 PSO 优化属性权重是一种有效方法，在广义模糊数 CBR 模型和欧式距离 CBR 模型中都能得到更高的估算精度。本文提出的能够刻画软件项目数据模糊信息的估算模型，对于在软件开发早期进行软件成本估算来说有着重要意义。

参考文献：

[1] Lackman M. Controlling the project development cycle [J]. IEEE Engineering Management Review, 1987, 15 (3): 56－78.

[2] Molokken-Ostvold 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.

[3] Koch S, Mitlöhner J. Software project effort estimation with voting rules [J]. Decision Support Systems, 2009, 46(4): 895－901.

[4] Boehm B W, Valerdi R. Achievements and challenges in cocomo-based software resource estimation [J]. IEEE Software, 2008, 25(5): 74－83.

[5] Mittas N, Athanasiades M, Angelis L. Improving analogy-based software cost estimation by a resampling method [J]. Information and Software Technology, 2008, 50(3): 221－230.

[6] Vinay Kumar K, Ravi V, Carr M, et al. Software development cost estimation using wavelet neural networks [J]. Journal of Systems and Software, 2008, 81(11): 1853－1867.

[7] Oliveira A L I. Estimation of software project effort with support vector regression [J]. Neurocomputing, 2006, 69(13－15): 1749－1753.

[8] Sentas P, Angelis L, Stamelos I, et al. Software productivity and effort prediction with ordinal regression [J]. Information and Software Technology, 2005, 47 (1): 17－29.

[9] Elish M O. Improved estimation of software project effort using multiple additive regression trees [J]. Expert Systems with Applications, 2009, 36 (7): 10774－10778.

[10] Cheikhi, L, Abran, A, Desharnais, LM. Analysis of the ISBSG software repository from the ISO 9126 view of software product quality [C]. 38th Annual Conference on IEEE-Industrial-Electronics-Society, Montreal, Oct, 25－28, 2012.

[11] Maxwell K D. Applied statistics for software managers [M]. New Jersey: Prentice Hall, 2002.

[12] Wu Dengsheng, Li Jianping, Liang Yong. Linear combination of multiple case-based reasoning with optimized weight for software effort estimation [J]. Journal of Supercomputing, 2013, 64(3): 898－918.

[13] Nassif A B, Ho D, Capretza L F. Towards an early software estimation using log-linear regression and a multilayer perceptron model [J]. Journal of Systems

and Software, 2013, 86(1): 44－160.

[14] Azzeh M, Neagu D, Cowling P I. Analogy-based software effort estimation using fuzzy numbers [J]. The Journal of Systems and Software, 2011, 84 (2): 270－284.

[15] Heemstra F J. Software cost estimation [J]. Information and Software Technology, 1992, 34(10): 627－639.

[16] Mittas N, Angelis L. Ranking and clustering software cost estimation models through a multiple comparisons algorithm [J]. IEEE Transactions on Software Engineering, 2013, 39(4): 537－551.

[17] Dejaeger K, Verbeke W, Martens D, Baesens B. Data Mining techniques for software effort estimation: A comparative study [J]. IEEE Transactions on Software Engineering, 2012, 38(2): 375－397.

[18] Shepperd M, Schofield C. Estimating software project effort using analogies [J]. IEEE Transactions on Software Engineering, 1997, 23 (12): 736－743.

[19] 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.

[20] Chiu N H, Huang S J. The adjusted analogy-based software effort estimation based on similarity distances [J]. Journal of Systems and Software, 2007, 80(4): 628－640.

[21] 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.

[22] Azzeh M, Neagu D, Cowling P I. Fuzzy grey relational analysis for software effort estimation [J]. Empirical Software Engineering, 2010, 15(1): 60－90.

[23] 吴登生,李建平,蔡晨.软件成本估算的粒子群算法类比模型及自助法推断[J].管理科学,2010,23(3):113－120.

[24] 梁昌勇,顾东晓,范昕,陈文恩.面向不确定多属性决策问题的范例检索算法研究[J].中国管理科学,2009,17(1):131－137.

[25] 路云,吴应宇,达庆利.基于案例推理技术的企业经营决策支持模型设计[J].中国管理科学,2005,13(2):81－87.

[26] Chen S J, Chen S M. Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers [J]. IEEE Transactions on Fuzzy Systems, 2003, 11(1): 45－56.

[27] 文成林,周哲,徐晓滨.一种新的广义梯形模糊数相似性度量方法及在故障诊断中的应用[J].电子学报,2011,39(3A):1－6.

[28] Wei S H, Chen S M. A new approach for fuzzy risk analysis based on similarity measures of generalized fuzzy numbers [J]. Expert Systems with Applications, 2009, 36(1): 589－598.

[29] 张增刚,郑贤斌,李继志.基于广义模糊数相似测度风险分析方法[J].系统工程理论与实践,2010,30(4):738－743.

[30] Xie X L, Beni G. A validity measure for fuzzy clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(8): 841－847.

[31] 张俊光,杨芳芳,杨双.基于重大偏差标准的软件项目工作量管理方法研究[J].中国管理科学,2013,21(2):161－167.

Case-based Reasoning with Optimized Weight for Software Cost Estimation Based on Generalized Fuzzy Number

Wu Deng-sheng

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

Abstract: In the software development process, accurate estimation of software effort is of great significance for software project and the enterprise. In order to overcome the difficulties that there isn't an accurate number for new software project at early stages according to the attributes of history project dataset and the existing software effort estimation models can't deal with the fuzzy number effectively, case-based reasoning and generalized fuzzy number are integrated, and a case-based reasoning (CBR) model based on generalized fuzzy number is proposed for software effort estimation in this paper. The traditional similarity measure such as Euclidean distance is replaced by a new similarity measure based on generalized fuzzy number in CBR model. Furthermore, the fuzzy c-means clustering is applied to fuzz the accurate number in history project dataset. Moreover, particle swarm optimization (PSO) is employed to further optimize attrib-

ute weights of the model. Finally Desharnais data is adopted to examine the validity of the model. It is shown that the proposed generalized fuzzy numbers CBR model can improve the estimation accuracy in comparison with the commonly used Euclidean distance CBR. In addition, it is also shown that the model with optimized weight from PSO can improve the estimation accuracy.

Key words: software effort estimation; case based reasoning; generalized fuzzy number; weight optimization