跨组织数据的软件成本估算方法①

梁 然[1,2], 杨 达[1]

[1](中国科学院软件研究所 基础软件国家工程研究中心, 北京 100190)
[2](中国科学院大学, 北京 100190)

摘 要: 软件成本估算一直是软件项目管理的重要部分. 经过半个多世纪的研究和工业实践, 成本估算方法、模型得到了极大的丰富. 这些方法、模型也衍生出了各种成本估算工具. 但是, 成本估算方法和模型的基础是历史项目数据. 没有历史项目数据的公司和组织只能利用其他公司或组织的数据来进行自己项目的成本估算. 如何利用跨组织数据进行有效的成本估算成为更具现实意义的问题. 针对这一问题, 提出了一种有效利用跨组织数据进行成本估算的方法, 并通过实验说明了方法的有效性.

关键词: 软件估算; 成本估算; 工作量估算

Software Cost Estimation Method with Cross-Company Data

LIANG Ran[1,2], YANG Da[1]

[1](Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Science, Beijing 100190, China)

Abstract: Software cost estimation has been an important part of the software project management. Afterover half a century of research and industrial practice, cost estimation methods and models have been greatly enriched. A variety of cost estimation tools are derived from these methods and models. However, cost estimation methods and models are based on historical project data. Companies and organizations without historical project data can only use data from other companies or organizations to do the cost estimates of their own projects. How to effectively do cost estimationwith cross-company data becomes more realistic. Aiming at this problem, we propose a new cost estimation method to use cross-company data effectively. The experiments show that the method could improve the accuracy.

Key words: software estimation; costestimation; effort estimation

1 引言

软件成本估算一直是软件项目管理的重要部分. Barry Boehm 指出软件成本模型和估算方法的四个目的[1]: 1)项目预算; 2)权衡和风险分析; 3)项目计划和控制; 4)软件改进投资分析. 从以上总结的四个目的不难看出软件成本估算的重要作用. 而且, 软件成本估算的不精确性会导致有限资源的不合理分配. 资源的不合理分配会进而导致一连串的不良后果[2], 影响软件项目的进程. 因此, 大量的成本估算的研究集中在高精确度的估算方法和模型上.

为了提高软件成本估算的精确度, 研究人员提出了各种各样的估算方法和模型. 这些方法和模型大体可以分为以下四类[4]: 1)通用模型, 如, COCOMO 系列模型[3]、SLIM[4]、PRICE-S[5]; 2)统计分析方法; 3)机器学习方法; 4)专家判断. 随着越来越多的方法和模型被提出, 有研究者在比较各种方法和模型的精确性时, 发现同一个估算方法在不同的数据集上表现有差异, 并且当某种方法在数据集上估算精确性较差时, 其他方法反而会有较高的精确性[6]. 这种现象导致了组合多种方法的组合估算成为成本估算的研究热点.

① 基金项目:国家自然科学基金(91218302,61073044,61003028,71101138,61100071);北京市自然科学基金(4122087);核心电子器件、高端通用芯片及基础软件产品项目(2012ZX01039-004)
收稿时间:2014-03-04;收到修改稿时间:2014-04-11

并且, EkremKocaguneli 等人也指出, 虽然没有最好的单一的估算方法, 但是存在最好的估算方法的组合[7].

虽然现在已经存在大量关于成本估算方法和模型的研究, 但是得到可接受精确度的成本估算仍然是有挑战性的. Martin Sheppard 等人为我们指出了一个原因, “我们越来越关注合适的数据的可用性”[8]. 数据是学习系统的基础. 没有数据, 估算模型不能被训练和本地校准. 有许多组织并没有积累自己的历史项目数据, 它们只能通过利用别的组织的历史数据(“跨组织数据”)来估算自己的软件成本. 但由于组织间的差异, 直接利用跨组织数据来训练现有估算方法和模型并不能得到较好的估算效果. 在这种应用场景下, 如何利用跨组织数据进行有效的成本估算成为更具现实意义的问题.

为了解决上述问题, 本文提出了一种有效利用跨组织数据进行成本估算的方法. 我们方法的基本假设是, 如果一种估算方法在与目标项目相似的项目集上的估算精确性高, 那么这种方法也应该也能很好地应用于目标项目. 严格的说, 我们的方法并没有真正意义上提出一种新的成本模型或估算方法, 而是提供了如何利用已有的跨组织数据来有效选择已知的估算方法(“基本估算方法”). 我们通过实验说明了我们的假设是成立的. 并且, 实验结果也显示了我们提出的方法的有效性.

2 相关工作

2.1 软件成本估算及其常用的估算模型和方法

从 20 世纪 60 年代末期开始, 以大量软件项目进度延期、预算超支和质量缺陷为典型特征的软件危机开始逐渐被人们所认识. 其中, 造成软件危机的原因之一就是人们对软件成本估算不足[9,10].

软件成本估算, 就是确定软件项目开发时间和开发成本的过程. 一般软件项目的绝大多数成本是人力成本. 由于人力成本主要是通过工作量确定的, 所以对软件项目成本的估算主要转化为估算项目开发的工作量. 于是, 成本估算和工作量估算这两个概念很多时候是交替使用的.

进行软件成本估算的方法和模型多种多样, 有基于专家的估算, 如专家判断, 有基于数据的估算, 如 COCOMO 模型、统计分析方法、机器学习方法. 本文考虑的方法和模型都是基于数据的估算. 本节将简单介绍其中最为常用的两种模型和方法, 也是本文实验中涉及到的成本估算方法: COCOMO 模型和类比估算.

2.1.1 COCOMO 模型

COCOMO 模型(COnstructiveCOst Model[3])是由南加州大学的 Barry Boehm 教授在上个世纪八十年代开始研究的成本模型, 经历了近三十年代的发展和不断完善. 无论是最初的COCOMO 81模型[3], 还是二十世纪九十年代中期提出的逐步成熟完善的 COCOMO II[11,12], 所解决的问题都具有当时软件工程实践的代表性. COCOMO 模型也是目前应用最广泛、得到学术界与工业界普遍认可的软件估算模型之一.

COCOMO 模型的通用的表达形式如下:

$$PM=A*(\sum Size)^{\sum B}*\prod EM$$

其中, PM 是工作量, 单位通常是人月(Person Months); A 是校准因子; $Size$ 是对工作量呈可加性影响的软件模块的功能尺寸的度量(如, 代码行、功能点等); B 是对工作量呈指数或非线性影响的比例因子; EM 为影响软件开发工作量的工作量乘数.

2.1.2 类比估算

类比估算(Estimation By Analogy, EBA[13])是一种基于实例推理的机器学习方法, 即通过对一个或多个已完成的项目与目标项目的对比来估算目标项目的成本. 它的基本假设是软件项目的描述特征越相似, 则它们所需的成本也越相近.

采用类比估算主要的关键活动有: 1)描述项目特征; 2)度量相似程度和选择相似项目; 3)根据相似的项目数据得到最终估算值. 这些活动中有很多决策选择来决定类比估算是否能得到较好的估算结果, 这也是研究者普遍关注的问题[14-16]. 比如, 如何选择项目特征?如何计算相似程度?选择多少个相似的项目?如何处理/综合各相似项目的成本值来得到目标项目的成本值?

2.2 组合估算

在面对各种各样的成本估算方法和模型时, 成本估算人员往往感到很为难: 该如何选择适应特定场景下的估算方法呢. 更为重要的, 大量研究[6,7]表明并没有哪种估算方法和模型一定最好. Stephen G. MacDonell 等人早期在进行估算方法组合的研究时指出, “在实验中, 我们的数据集上采用多个估算方法会带来潜在的优势”[6]. EkremKocaguneli 等人在最近的

研究实验中组合了 90 个单一方法(solo-methods), 他们的实验结果表明, 虽然没有最好的单一的估算方法, 但是存在最好的估算方法的组合[7]. 这些结论在一定程度上支持了组合估算的后续研究.

组合估算的研究都是利用静态组合策略来进行估算方法的组合. 静态组合估算策略指定不同的估算方法组合的方式, 它对不同的目标项目不具备差异性. 这种策略不能充分考虑和利用历史项目和目标项目之间的信息. 我们的方法从本质上来说是一种组合估算. 它是基于动态选择策略来从多个已知的估算方法中选择适合的估算方法. “动态选择”[17,18]的概念来自于多分类器系统. 在多分类器系统中, 它是一种组合多个分类器的算法. 动态分类器选择试图从已知的多种分类器中选择一个对给定样本具有潜在最好的分类效果的分类器.

2.3 跨组织数据的成本估算

正如 2.1 节描述的, 除了专家判断是不需要依赖数据的估算方法(其实也需要数据, 只是数据已经内化为专家的知识和经验), 其他的估算方法和模型都是需要数据来进行训练和校验. 很多研究比较了基于本地数据和基于跨组织数据进行估算的差异. 基于本地数据的估算是基于公司和组织本身的历史项目数据(“本地数据”)来进行成本估算的方法. 基于跨组织数据的估算是基于跨组织数据的成本估算的方法, 这主要是针对缺乏本地数据的公司和组织.

正如在是否存在最好的单一的估算方法一样, 基于本地数据的估算和基于跨组织数据的估算哪个精确性高也是一直争论的问题. 不过, Kitchenham, B.A 等人在调研了大量研究文献后, 给我们提供了如下建议, “基于我们调研的结果和我们的经验, 我们建议在采用跨组组数据的成本估算时要考虑跨组织数据中与目标项目相似的项目, 也要考虑公司自身的特征”[19]. 我们的方法中也考虑了跨组织数据中和目标项目相似的项目.

3 方法介绍

本节详细介绍了我们提出的有效利用跨组织数据的成本估算方法.

3.1 基本要求

正如本文前面介绍的, 我们的方法是针对缺乏历史项目数据的公司和组织提出的, 并不是通用的成本估算方法. 在使用我们的方法时, 首先要考虑如下基本要求:

需要具有跨组织项目数据, 不管是来源于开放数据集或非公开公司数据集;

需要具有至少两种基本成本估算方法;

跨组织数据集有足够的数据量, 能够满足成本估算方法和模型的训练和校验的需要.

对于缺乏历史项目数据的公司和组织, 以上基本要求是能够满足的.

3.2 核心思想

我们方法主要包括三个基本步骤, 如下:

采用留一交叉验证方法(Leave-One-Out Cross Validation, LOOCV)评估每种基本估算方法对跨组织数据中每个项目成本的估算的精确性;

利用 K-最临近方法(k-Nearest Neighbor Algorithm, KNN)在跨组织数据中找出与目标项目最相似的 K 个项目;

基于 LOOCV 的结果, 选择在相似项目集上表现最好的估算方法, 并利用它来进行估算. 方法的流程图如图 1.

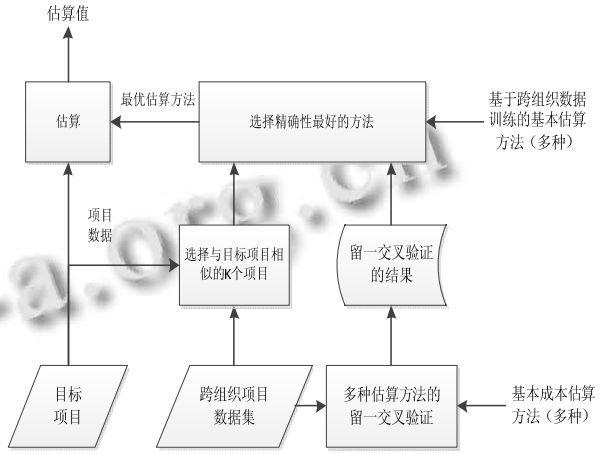

图 1 方法流程

图 1 中多种基本估算方法在跨组织项目数据上的留一交叉验证的结果保存在 $m*n$ 的表中(其中, m 表示跨组织项目数据的项目数, n 表示使用的基本估算方法的数目, 表中每个值表示实际工作量与估算工作量之间的差距度量值, 比如, 相对误差(Relative Error, RE)), 如图 2.

我们方法的核心是在与目标项目相似的跨组织项目数据上评估各种基本估算方法的精确性, 并根据这

个评估结果选出潜在最优的估算方法来进行实际的估算. 因此, 我们的方法不需要本地数据, 并能充分利用跨组织数据中的信息.

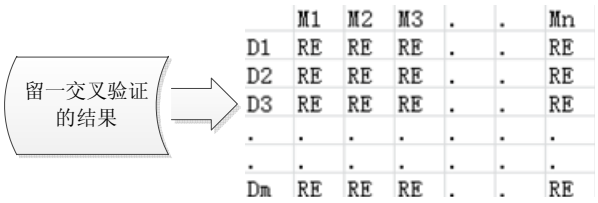

图 2 留一交叉验证 LOOCV 的结果表示

4 实验及分析

COCOMO 模型和类比估算是两种常用的软件估算模型和方法. 在分析我们方法的有效性时, 我们在实验中实现这两种方法. 由于本文方法的核心就是有效利用跨组织数据来选择潜在最优的基本估算方法, 在实验中我们关注是否选择了的最好的估算方法, 并总体上提高了精确性. 在实验中, 我们选取了相对误差作为精确性度量指标[20,21].

4.1 实验数据

实验中使用的数据集是根据 COCOMO81 模型收集整理的两个不同年份的数据集, COCOMO81 数据集和 NASA93 数据集. 这两个数据集可以从 PromiseData[22]上获取.

表 1 简单描述了这两个数据集中项目的规模和工作量的统计信息(其中, 规模单位: 千代码行(KLOC), 工作量单位: 人月(PM)).

表 1 COCOMO81 和 NASA93 数据集的统计信息

数据集	属性名	均值	中值	最小值	最大值
COCOMO81 (63 条)	规模	77.21	25	1.98	1150
	工作量	683.53	98	5.9	11400
NASA93 (93 条)	规模	94.02	47.5	0.9	980
	工作量	624.41	252	8.4	8211

由于数据是根据 COCOMO81 模型进行收集整理的, 这两个数据集中数据属性包括了模型中的 15 个工作量乘子、1 个项目规模和 1 个实际工作量. 由于篇幅有限, 我们在这里就不具体介绍各个工作量乘子及其定义描述, 可以参考[3].

4.2 估算方法实现

实验选取两种比较常用的估算方法作为我们方法中的基本估算方法. 在这里, 成本估算实际上估算的是项目的工作量值, 即根据项目数据来得到项目工作量的过程.

正如第 2 节中简单介绍的, 在应用类比估算时有很多决策选择, 比如: 如何计算相似程度? 选择多少个相似的项目? 如何处理各相似项目的成本值? 在实现类比估算时, 根据不同的决策选择, 实现了四种不同的类比估算. 根据前期对类比估算的认识, 我们实现了: 两种相似程度的计算(规模距离[规模的差]、欧式距离[各属性值差的平方和, 然后取平方根]), 两种相似项目的数目取值(n=1, n=3), 一种合并相似项目的工作量值的方法(基于距离的加权和).

表 2 中展示了实验中实现的估算方法.

表 2 实验中实现的估算方法

序号	方法名	描述
1	RegressionCocomo	COCOMO81
2	SimpleAnalogySize	EBA, n=1, 规模距离
3	SimpleAnalogyEDistance	EBA, n=1, 欧式距离
4	AnalogySize	EBA, n=3, 规模距离
5	AnalogyEDistance	EBA, n=3, 欧式距离

4.3 实验描述

在实验中, COCOMO81 数据集被当作是跨组织数据, 而 NASA93 数据集是目标项目数据集. 根据第 3 节中描述的方法步骤, 实验的流程如图 3.

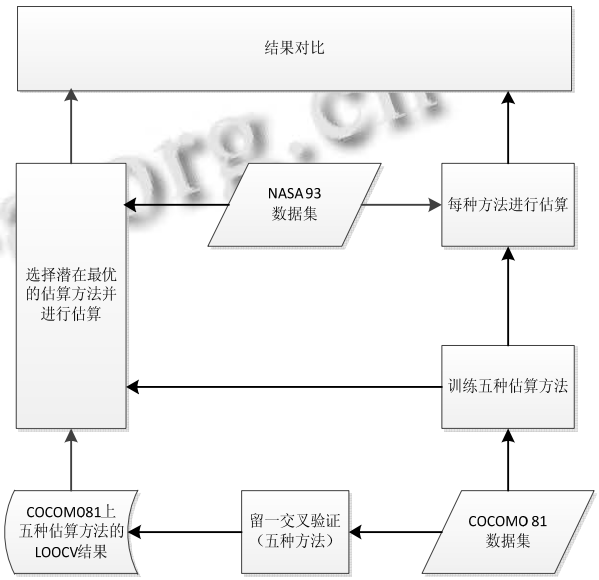

图 3 实验流程

4.4 实验结果分析

根据以上描述的实验流程, 基于 COCOMO81 数据集和 NASA93 数据集, 我们对本文提出的方法进行

了实验分析. 实验的主要目的是回答我们的方法是否可以有效利用跨组织数据提供的信息来选择潜在最优的估算方法.

图 4 中展示了实现的五种方法在 COCOMO81 数据集上进行留一交叉验证的实验结果. 从图中可以明显看出同一个方法在不同的项目数据上, 方法的表现差异(RE 波动)很大. 比如, 方法 SimpleAnalogySize 在几个项目上 RE 值相当大, 说明它的精确度很低, 但它也有 RE 较小的一些点, 说明它在这些项目上的精确度较高.

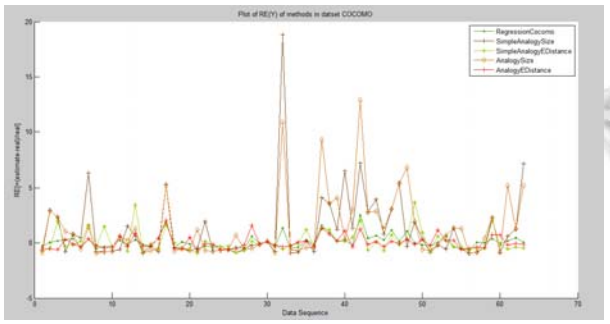

图 4 COCOMO81 数据集上五种方法 LOOCV 的结果

当然, 图 4 中波动现象出现的原因也是由于留一交叉验证中对每个项目进行估算时, 估算方法的训练数据本身有差异造成的. 图 5 展示了五种方法在 COCOMO81 数据集上进行训练, 然后对 NASA93 项目进行估算的结果. 不难看出这种波动现象仍然存在. 这就说明了, 确实没有最好的单一的成本估算方法. 并且, 从图 5 中可以看出, 这种波动现象在不同的方法之间不是完全同步的. 这就说明, 对于不同的项目选择不同的估算方法来进行估算是可以从某种程度上来提高估算精确性的.

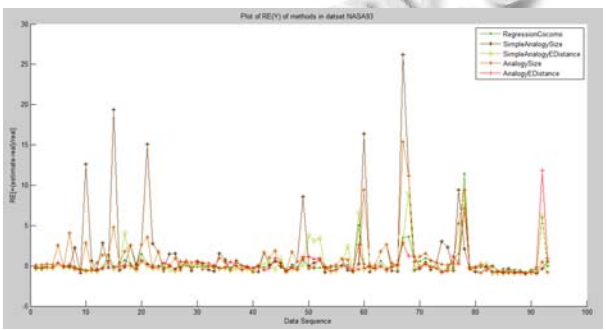

图 5 NASA93 数据集上五种方法估算精确性结果

在图 5 中, 方法 RegressonCocomo 和方法 Analogy EDistance 总体来看是表现优于其他方法. 两者在不同的数据上仍然存在着差异. 我们期望我们方法对每个项目进行估算时, 能每次都选出最优的方法, 进而提升估算精确度.

在实验中, 调整 KNN 的 k 取值会影响估算精确性. 图 6 中给出了估算精确性随 k 值变化的结果.

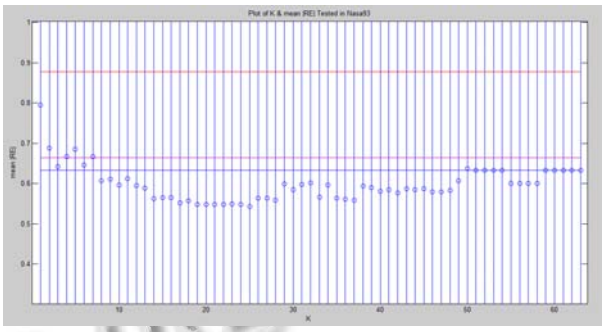

图 6 K 值变化对估算结果的影响

在图 6 中, 圆点(蓝色)表示 K 取不同的值时(取值区间由跨组织项目数据的大小决定)我们的方法在 NASA93 上的估算相对误差的绝对平均值. 图中三条水平线表示五个估算方法的相对误差的绝对平均值, 另外两条水平线的值远远大于 1, 超出了图所表示的范围内(由于五个估算方法的估算值和 k 的取值无关, 为了比较方便, 我们将它们用水平线表示, 并横跨 k 的整个取值范围).

根据图 6 的结果, 当 k 取值在[12,28]时, 我们方法的效果很明显(基本能提升 10%). 虽然我们方法的估算效果取决于具体度量指标、k 的取值以及选择的基本估算方法, 但通过对前面实验结果的观察, 我们的方法确实能超过基本估算方法的估算精确度.

通过实验及其对结果的分析, 我们的方法的确能有效利用跨组织数据的信息, 进而选择潜在最优的估算方法.

5 讨论

本文提出的方法和第 2 节介绍的类比估算类似, 在使用方法时有很多决策选择问题, 比如, 如何计算相似程度? 选择多少个相似的项目? 根据相似的项目如何评估各成本估算? 这些问题一直是也将长期存在, 因为这些问题本身和实际项目情况和数据是直接相关的. 比如, 在本文实验中, 我们了解 K 的变化与估算结果之间的影响, 但却不能下结论说, 对别的数据别的环境, 它们之间的关系仍然是这样, 我们只能说 K 的变化确实会改变我们的方法从跨组织数据中获得的信息, 进而影响评估和选择潜在的估算方法.

本文实验选择的两个数据集都是在几十年前收集的, 但这并不意味着它们过时、没用了. 况且, 我们的方法并不依赖于具体的数据集. 这两个数据集本身在成本估算领域中具有较高的认可度, 无疑从某种程度

上保证了数据的真实性. 这一点对于我们的实验无疑是很重要的考虑因素. 关于数据中软件项目的开发环境和开发实践和当前的差别很大, 但这可以通过变更我们使用的基本估算方法来避免, 比如数据的属性和 COCOMO81 模型有关, 我们就可以在基础估算项目中加入COCOMO81估算模型. 这也进一步说明, 我们的方法本身对数据没有依赖性. 当然, 我们也会在今后的研究工作中将本文的方法应用于实践, 用实践来检验我们方法的有效性.

最后, 基于跨组织数据的成本估算与基于本地数据估算是两种不同的估算场景. 当公司/组织有足够的本地数据, 当然应该充分利用本地数据. 但是当没有本地数据的时候, 公司/组织不得不使用跨组织数据来进行估算方法和模型的训练和校准. 从这个角度来说, 哪种效果好不好是没有太大意义的.

6 结语

本文针对如何利用跨组织数据进行有效的成本估算的问题, 提出了一种有效利用跨组织数据进行成本估算的方法. 方法的基本假设是, 如果一种估算方法在与目标项目相似的项目集上的估算精确性高, 那么这种方法也应该也能很好地应用于目标项目. 文中通过实验说明了本文方法的有效性. 最后, 希望方法能够促进软件估算实践活动.

参考文献

1 Boehm BCA, Chulani S. Software development cost estimation approaches--A survey. Annals of Software Engineering, 2000, 10(1): 177–205.

2 Lederer AL, Prasad J. Nine management guidelines for better cost estimating. Commun. ACM, 1992, 35(2): 51–59.

3 Boehm BW. Software Engineering Economics. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

4 Putnam L, Myers W. Measures for Excellence, 1992, Yourdon Press Computing Series.

5 Park R. The Central Equations of the PRICE Software Cost Model, Park R, 4th COCOMO Users' Group Meeting, November 1988.

6 MacDonell SG, Shepperd MJ. Combining techniques to optimize effort predictions in software project management. Journal of Systems and Software, 2003, 66(2): 91–98.

7 Kocaguneli E, Menzies T, Keung J. On the value of ensemble effort estimation. IEEE Trans. on Software Engineering, 2011, 99: 1.

8 Shepperd M, Cartwright M. Predicting with sparse data. IEEE Trans. on Software Engineering, 2001, 27(11): 987–998.

9 李明树,何梅,杨达,舒风笛,王青.软件成本估算方法及应用. 软件学报,2007,18(4):775–795.

10 Glass RL. Facts and Fallacies of Software Engineering. Boston: Addison-Wesley, 2003.

11 Boehm BW, Clark B, Horowitz E, Westland C. Cost models for future software life cycle processes: COCOMO 2.0. Annals of Software Engineering, 1995, 1: 57–94

12 Boehm B, Abts C, Brown AW, Chulani S, Clark BK, Horowitz E, Madachy R, Reifer D, Steece B. Software Cost Estimation with COCOMO II. Prentice Hall, 2000

13 Shepperd M, Schofield C. Estimating software project effort using analogies. IEEE Trans. on Software Engineering, 1997, 23(11): 736–743.

14 Kocaguneli E, et al. Exploiting the essential assumptions of analogy-based effort estimation. IEEE Trans. on Software Engineering, 2012, 38(2): 425–438.

15 Azzeh M. A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation. Empirical Software Engineering, 2012, 17(1): 90–127.

16 Azzeh M, Neagu D, Cowling P. Fuzzy grey relational analysis for software effort estimation. Empirical Software Engineering, 2010, 15(1): 60–90.

17 Huang YS, Suen CY. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1995, 17(1): 90–94.

18 Woods K, Kegelmeyer WP Jr, Bowyer K. Combination of multiple classifiers using local accuracy estimates. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1997, 19(4): 405–410.

19 Kitchenham BA, Mendes E, Travassos GH. Cross versus within-company cost estimation studies: A systematic review. IEEE Trans. on Software Engineering, 2007, 33(5): 316–329.

20 Kitchenham BA, et al. What accuracy statistics really measure [software estimation]. Software, IEEE Proceedings, 2001, 148(3): 81–85.

21 Foss T, et al. A simulation study of the model evaluation criterion MMRE. IEEE Trans. on Software Engineering, 2003, 29(11): 985–995.

22 Menzies T, Caglayan B, Kocaguneli E, Krall J, Peters F, Turhan B. The PROMISE repository of empirical software engineering data. West Virginia University, Department of Computer Science. http://promisedata.googlecode.com. 2012.