中国科学院软件研究所

杰出青年人才发展专项计划年度进展报告书

（2011 年度）

姓名	杨叶	课题名称	经验软件工程方法研究
资助金额	160 万	课题起止时间	2009.9-2013.9
资助类别		所在部门	基础软件中心

研究工作主要进展和阶段性成果（含论文、研究生培养等）

（一）主要工作进展

由于软件开发环境天然的易变性（开发过程、新技术、新工具等），软件工程领域的很多预测和估算模型，如缺陷预测模型、可靠性模型、成本估算模型等都会遇到概念漂移的问题，即新收集的数据所包含的模式相对于旧数据不断发生变化的现象，需要结合软件工程研究的具体特点加以解决，否则会导致已训练的模型对新数据的预测能力下降。

本项目 2011 年度的工作围绕以软件成本估算和缺陷预测为代表的经验软件工程研究领域的预测模型和方法，对典型预测模型的性能评价、校准和优化问题进行改进和创新研究。研究结果有助于提高典型软件工程预测模型在模型/训练数据选择、模型维护、预测结果的准确度和可信度，从而确保经验软件工程研究结果的稳定性和可重复性。主要工作成果包括以下几个部分：

- 成本估算模型校准中的局部偏差度量及处理方法
- 基于需求演化的规模估算方法
- 缺失数据处理技术
- 跨组织数据在软件缺陷预测中的可适用性
- 基于本体的软件估算经验提取方法

1、成本估算模型校准中的局部偏差度量及处理方法

参数估算模型依赖于使用历史数据进行校准，以确定参数的恰当取值。这种模型分为两类。一是通用模型，校准它的参数时使用来自各个组织的数据。二是本地模型，校

准它的参数时使用来自特定组织的数据，并基于一个通用模型。另外，本地模型所能校准的参数集合是通用模型的子集。常见的实践是各个组织在采纳通用模型的升级版本后，使用本地数据再校准得到本地模型的升级版本，最后以此版本作为本地估算的模型。原因是通用模型的估算精度通常要低于本地模型。不过通用模型的校准使用来自多个组织的数据，是导致其估算精度较低的一个原因。实际上，各个组织提供的数据是带有本地偏差的，并非完全符合通用模型的定义。使用这种带有偏差的数据来校准通用模型，自然会使得估算精度较低。可想而知，如果能够消除数据中的本地偏差，通用模型的估算精度当会有所提高。即便不能消除偏差，如果能够发现数据中的较大偏差，亦可在校准通用模型时提出警告。

以 COCOMO II 2000 成本估算模型为例，该模型的默认版本（general model）在 2000 年前收集的 CII2000 数据集上能达到很好的预测精度（Pred(30%)=75.2%），但对 2000 年以后收集的项目数据（After2000 数据集）的预测精度急剧下降（Pred(30%)=37.0%）。因此，需要在模型的使用过程中，不断的利用新收集的数据对模型进行校准，使估算模型更能好的匹配当前项目开发环境的特点，保证模型的稳定性和精确性。

对于 COCOMO 模型的本地偏差处理方法基于如下假设：本地偏差越大的数据，代表的估算模式与当前模型的模式差异越大，在使用该数据进行模型校准时应赋以更小的权重。在进行模型校准时，先按所在的组织将新收集的数据切分为多个子集，每个子集包含一个组织的所有数据。按照假设，需要寻找一种权重分配函数，为各个子集进行权重分配。该函数以各子集的局部偏差为输入，输出各子集的权重：Weight=F（LocalBias）。权重分配函数应满足如下约束：

1） 当 LocalBias=0 时，Weight=1：当新收集的数据本地偏差为 0 时（即新数据与原模型包含的模式完全一致时），新数据的权重应当为 1。

2） 当 LocalBias 趋近于正无穷时，Weight 应趋近于 0：当新数据包含的模式与原模型完全不同时，在模型校准时应剔除该数据的影响（Weight=0）。

3） F 是[0, +∞) 区间上的单调递减函数：按照假设，本地偏差越大的数据，分配的权重应当越小。

实验中共比较了 3 种不同的权重分配函数：

1) **Function-1**: $Weight=\frac{1}{1+\ln(LocalBias)}$

2) **Function-2**: $Weight=\frac{1}{LocalBias+1}$

3) **Function3**: $Weight=\frac{1}{e^{LocalBias}}$

这三种函数均满足约束，仅下降速率不同，代表本地偏差对权重分配的影响不同。在 COCOMO II 2000 模型和 CII2010 数据集上的实验结果表明，三种权重分配函数均能不同程度的降低新收集的数据中本地偏差的负面影响，提高了数据的可用性。

2、基于需求演化的规模估算方法

大部分的软件成本估算模型需要软件规模作为输入，因为产品规模是决定软件开发工作量及成本的一个重要的因素。然而在项目初期软件的代码行或功能点是无法获得的，准确地估算项目规模一直是一个富有挑战性的问题。

解决这个问题的方法往往取决于项目初期我们能获取怎样的项目高层次信息。通常情况下，在项目初始阶段，我们能获取的信息主要是项目的高层次抽象目标。一般而言，此类目标定义的内容比较抽象，不足以具体到能够直接估算出软件规模或工作量。因此如果我们想利用这些信息来估算软件规模，则需要了解更加详细的信息。其中一种解决的办法就是去研究这些顶层抽象信息与详细的低层次信息之间有着怎样一层层递增的映射关系。

随着项目开发的不断推进，其需求可以以多种形态表示，例如：目标、功能、用例、用例步、代码行等，它们有着不同的详细程度。在此语境下的需求演化指的是项目需求从一开始的高层次信息（目标）到最后的低层次信息（代码行）之间的逐步细化。演化的抽象过程可以用演化因子来度量。两个相邻需求层次之间的演化因子可以定义为它们数量上的一个比率。例如：如果用例与用例步之间的演化因子为 7，则表示一个用例平均包括 7 个用例步。同样地，如果用例步与代码行之间的演化因子值为 15，就表示一个用例步平均被 15 行代码实现。

每个项目都有它自己的演化进程。演化因子的数量取决于需求可以表现出来的层次。通常情况下，如果一个项目需求可以分为 n 层，则演化因子会有 n-1 个。为了帮助软件开发组织更好地了解需求演化过程，在项目早期得到尽可能准确的规模及工作量估算，我们同中国一中小型软件公司进行合作，用该公司的案例项目历史数据，主要包括需求文档，需求用例及代码，针对这一问题进行了经验研究。案例项目的开发主要是重用开发，每个新的版本都是对前一个版本的复用和维护，因此对案例项目开发过程中需求演化及规模估算的研究能够帮助我们进一步确定重用软件开发过程中项目演化的特点，帮助我们更好地进行重用工作量估算的研究。

我们深入细致的研究了现有的软件成本估算方法，分析并对比了这些方法的应用范围、特点和局限性，在此基础上提出了一种新的基于用例的重用软件成本估算方法。我们研究的目的在于分析增加、修改、不变的用例数对重用工作量的影响，并利用实际工业数据，回归出一个适用于需求阶段的重用工作量估算方法，用于指导中小规模公司的项目的重用工作量估算。我们提出的方法能够同时应用于新开发和重用类型项目的成本估算，首先充分利用企业组织的历史项目数据，挖掘软件需求中的用例信息，为目标项目的成本估算建立基于用例的模型，然后在此模型基础之上，引入需求细化因子（requirements elaboration factors）来估算目标项目的用例规模，使得工作量估算能够在软件生命周期的早期阶段进行。

基于案例项目数据对以上模型进行交叉验证，计算得到 MRE 平均值为 32.19%，MRE 中值为 22.51%，在实际的项目估算中，这样的估算准确度是比较令人满意的，回归方法的误差可接受的范围之内。另一方面，此估算方法的执行效率相对于其它方法会更高，仅要求估算者掌握历史数据，不需要与开发人员或专家进行沟通，不需要收集环境因素等。对于一些需要短时间制定管理计划，或难以收集环境因子的组织来说会更实用。

3、缺失数据处理技术

软件工作量可信是软件可信性能评价重要方面。当软件工作量数据被用于工作量估算，其背后基本的假设是软件工作量历史数据能够被用于建构软件工作量预测模型（统计学模型如线性回归模型，或者机器学习模型如神经网络模型）。然而，目前研究人员面临的一个难题就是软件工作量历史数据中存在大量的缺失数据和不完整数据。通常，由于软件工作量历史数据集合较小，经典统计学常用的数据预处理方法—去除缺失数据—的做法，往往会导致在预处理过后的数据集上建立的预测模型存在较大的偏差（Bias），进而影响预测模型的精度。由此，处理软件工作量数据缺失已成为软件工程领域一个比较活跃的研究方向。

- **软件工作量数据的缺失机制**

缺失数据中实际上存在两种缺失机制：结构性缺失和非结构性缺失。前者指在数据收集的过程中，软件项目本来就不具备某种或某类特征；后者指软件项目存在数据收集过程中给定的项目特征，但实际数据测量不准确或者没有被收集上来。我们以 ISBSG 和 CSBSG 为研究对象，分析了软件工作量数据中的缺失机制。首先，我们假定数据中的缺失服从某种缺失机制，然后利用在此种机制下的缺失数据处理方法对缺失数据进行修正并进行软件项目工作量的预测。当缺失数据被假定为结构性缺失条件下，最大间距分类（Max-Margin Classification）被用来处理缺失的工作量数据并更根据历史数据预

测软件工作量。当缺失数据被假定为非结构性缺失，MINI（mean imputation based k nereast neighbour hot-deck imputation）方法，类别均值修复方法（class mean imputation,CMI）和均值修复（Mean Imputation，MI）被用于修复工作量中的缺失数据。然后利用支持向量机（Support Vector Machine， SVM）在修复后的数据基础上进行工作量预测。图 1 显示了以上缺失数据处理方法在 ISBSG 和 CSBSG 数据集上的软件工作量预测性能对比。

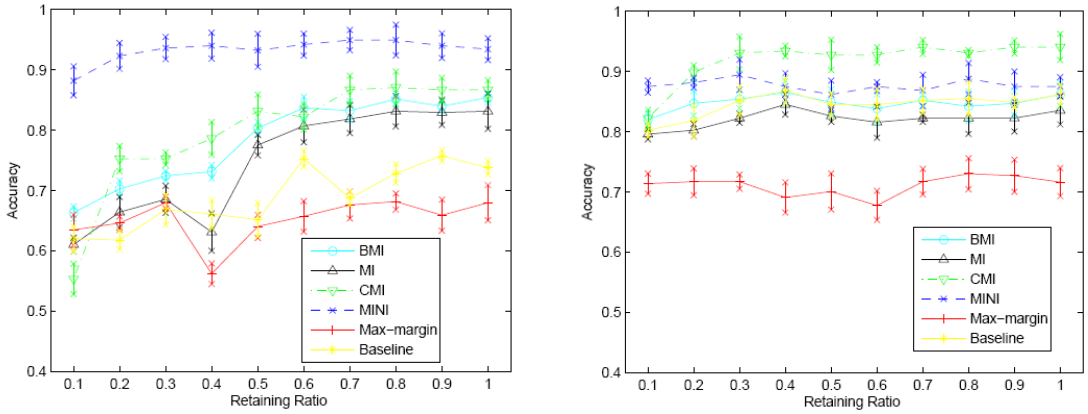

图 1 结构性和非结构性缺失数据处理方法在 ISBSG 数据集（左）和 CSBSG 数据集（右）上进行工作量类别预测的性能对比

实验结果表明，无论对于 ISBSG 数据集还是 CSBSG 数据集，建立在非结构性数据缺失假设基础上的缺失数据处理方法比建立在结构性数据缺失假设基础上的软件工作量预测性能要好。而且，在非结构性数据缺失处理方法内，在 ISBSG 上，MINI 方法要比 CMI 方法要好，而在 CSBSG 上，其结论恰好相反。我们将其解释为 CSBSG 数据集上不同工作量类别的数据集之间的方差较小，而 ISBSG 上的数据则方差较大。

- **基于 Naïve Bayesian 和 EM 算法的软件工作量缺失数据处理方法**

本研究提出了一种基于朴素贝叶斯模型和 EM（Expectation Maximization）算法的工作量缺失数据处理方法：容忍数据缺失方法（EM-T）和修复缺失数据方法（EM-I），并进行了对比研究。并利用标杆数据集 ISBSG 和 CSBSG 数据验证了这种缺失数据处理在不完全数据上进行工作量预测的有效性。 软件工作量数据缺失是软件工作估算的一个难题。本研究在朴素贝叶斯模型和 EM（Expectation Maximization）算法的基础上，提出了针对软件工作量数据缺失的处理方法。图 2 显示了容忍数据缺失方法，修复数据缺失方法和基准方法分别在 ISBSG 和 CSBSG 数据集上所建立的类别预测模型的性能比较。

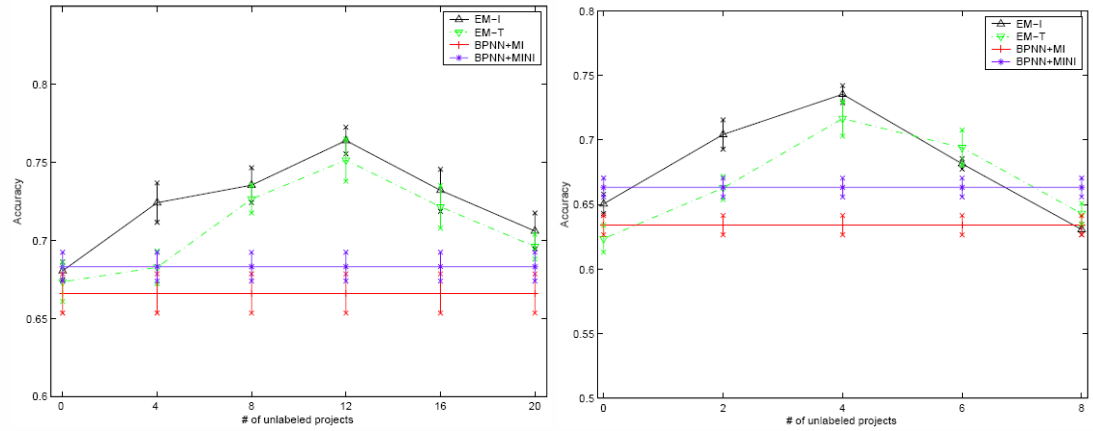

图 2 结构性和非结构性缺失数据处理方法在 ISBSG 数据集（左）和 CSBSG 数据集（右）上进行工作量类别预测的性能对比

实验结果表明，无论对于 ISBSG 数据集还是 CSBSG 数据集，建立在非结构性数据缺失假设基础上的缺失数据处理方法比建立在结构性数据缺失假设基础上的软件工作量预测性能要好。而且，在非结构性数据缺失处理方法内，在 ISBSG 上，MINI 方法要比 CMI 方法要好，而在 CSBSG 上，其结论恰好相反。我们将其解释为 CSBSG 数据集上不同工作量类别的数据集之间的方差较小，而 ISBSG 上的数据则方差较大。

4、跨组织数据在软件缺陷预测中的可适用性

经验研究表明，软件缺陷的分布通常满足“2-8”原则，即大部分的缺陷发生在一小部分模块中。因此，在测试开始前需要对缺陷的分布进行预测，以便更有效地分配测试资源，提高产品质量。已有的缺陷预测方法通常从软件的历史版本缺陷数据中学习出模型，然后对新开发版本的缺陷进行预测。然而在实际的项目环境中，这种历史版本的数据（即预测模型的训练数据）不一定总是可用，可能的原因有：1）对新开发的项目，不存在历史版本数据；2）历史版本的数据没有进行有效地收集和整理，或收集的数据质量太差。在缺乏历史版本的数据的情况下（即缺乏本地训练数据），一种可能的缺陷预测途径是从其它项目的历史数据中训练出预测模型，然后对本地项目进行缺陷预测，即进行跨项目缺陷预测。但由于不同项目的开发环境不同，缺陷数据中包含的模式也不同，跨项目缺陷预测的精度难以保证，因此，需要对跨项目缺陷预测的可行性进行深入研究。具体的研究问题包括：

1) 从其它项目历史数据中训练出得模型能否提供可接受的预测结果（准确率>50%，召回率>70%）；
2) 项目的本地训练数据提供的预测结果是否一定比其它项目的训练数据提供的预测结果要好；
3) 如果跨项目缺陷预测可行，如何自动的从大量的其它项目历史数据中挑选出合适的训练数据。

实验共分析了 10 个项目共计 34 个版本的缺陷数据，采用支持向量机(SVM)、C4.5 决策树、决策表、朴素贝叶斯、Logistic 回归 5 种不同的机器学习算法构建缺陷预测模型。预测模型以模块的静态代码属性作为输入，使用召回率、精确率和 F 值作性能评价指标，性能评价过程使用 5-折交叉验证。实验结果表明：

1) 在最优的情况下，来自其它项目的训练数据能提供可接受的预测结果；
2) 在最优的情况下，来自其它项目的训练数据的预测效果要优于本地训练数据；
3) 在随机选取训练数据的情况下，跨项目缺陷预测的成功几率很低，仅为 3%左右。

基于以上结果，进一步设计和试验了一种基于数据集分布特征和决策树算法的跨项目缺陷预测模型训练数据自动选择方法。实验结果表明，通过该方法选取的跨项目训练数据能提供和本地训练数据相当的预测能力。

总体而言，在仔细进行训练数据的情况下，跨项目缺陷预测是可行的；所提出的基于数据集分布特征的跨项目训练数据自动选择方法能有效的处理本地训练数据不可用问题，扩大缺陷预测模型的应用范围。下一步计划研究更多更有效的训练数据自动选择方法，并将其扩展应用到其它预测模型，如成本估算模型等。

5、基于本体的软件估算经验提取方法

随着软件工程的不断发展，经验软件工程扮演着越来越重要的角色。Systematic Literature Review (系统化调研，以下简称 SLR)作为一种经验方法，在软件工程领域得到了广泛的应用。通过对大量的文献资料进行系统、科学、全面的调研，SLR 能够对特定的研究问题给出公正的令人信服的回答。SLR 包含“寻找初始文献集合，筛选文献，评估筛选出的文献质量，相关数据抽取，综合调研结论”五个步骤。由于整个过程需要专家经验判断，做一次 SLR 需要非常大量的人力投入。尤其是第二步骤“筛选文献”，可能要从成百上千篇文章中筛选出几十篇左右的文献，这部分开销是十分可观的。这也给实施 SLR 带来了普遍的困难。因此很多学者都在研究能不能用自动化的方法辅助 SLR 以提高效率。我们也从语义分析的角度，提出了一种辅助 SLR 的方法。具体的内容包括：

- 提出一种辅助 SLR 的方法，在成本估算文献集合上作一个特定应用；
- 从文献摘要入手，使用基于规则的句法分析方法，把成本估算文献的“非结构化摘要”转化为“结构化摘要”。将摘要分成“背景介绍，主体内容，结论”有序的三部分。其中背景介绍部分包含的信息噪音较多，予以舍弃。主要使用主题内容和结论两个部分；
- 使用语义分析方法，从这两部分中抽取实体信息。具体地，从主体内容部分抽取有意义的实体名词组。从结论部分，抽取出简单的带有感情色彩的简单结论逻辑关系。用这两部分的实体信息作为标签来代表一篇论文的实际内容。

- 对成本估算论文作者给出的关键字进行分析整理，建立出“方法，度量元，成本估算词汇”三个 Ontology 结构。每篇成本估算文章都拥有从摘要中抽取出来的“方法，度量与，成本估算，简单结论”四种实体信息。把这些实体信息一一映射到建立好的 Ontology 中去，建立几种 Ontology 之间的连接网络，作为成本估算的知识库。

我们使用了 347 篇成本估算文章，从这些文章的摘要中抽取出“主体介绍”与“结论”两个部分，能够达到 73%左右的准确率。接下来我们从主体介绍部分抽取出实体名词，召回率范围在 72.42 到 84.28%左右。使用这些信息，我们构建了 COSONT 成本估算 Ontology，用以辅助成本估算领域的 SLR。在方法验证方面，针对 SLR 第二步“筛选文献”,我们同时使用 COSONT 的自动化辅助方法与人工判断方法,并对结果进行对比。结果表明，自动化方法能够节省大量的时间，并能够达到和人工判断基本一致的调研结果。

6、国际交流与合作

2011 年 4 月：参与组织在北京召开的 Boehm Symposium，就软件成本估算、软件风险管理、系统与软件过程等相关研究问题与到会人员进行了探讨。

2011 年 6 月：参与 ENASE 2011 国际会议，并报告两篇学术论文。此外，与澳大利亚 NICTA 的张贺研究员联合组织 1st International Workshop on Evidential Assessment of Software Technologies，与 ENASE 2011 大会同址，旨在推动系统化、规范化的经验软件工程研究及评价方法。

2011 年 9 月：参加经验软件工程国际周 ESEIW 2011，并在其中的 PROMISE 2011 大会报告学术论文。

邀请来访:澳大利亚 La Trobe 大学 Richard Lai 教授和美国 Iowa State University 的 David Weiss 教授，分别进行软件过程建模和软件过程度量方面的交流与合作研讨。

（二）阶段性成果

1、论文（完成国际期刊论文 1 篇，国际会议论文 6 篇;）

1) He Zhimin; Shu Fengdi; Yang Ye; Li Mingshu; Wang Qing. An Investigation On The Feasibility Of Cross-project Defect Prediction. Accepted by Journal of Automated Software Engineering. 录用未刊登。

2) Ye Yang, Lang Xie, Zhimin He, Qi Li, Vu Nguyen, Barry W. Boehm, Ricardo Valerdi: Local bias and its impacts on the performance of parametric estimation models. PROMISE 2011: 14. Best paper award.

3) Wen Zhang, Ye Yang, Qing Wang: On the Predictability of Software Efforts

using Machine Learning Techniques. ENASE 2011: 5-14

4) Wen Zhang, Ye Yang, Qing Wang: Network analysis of OSS evolution: an empirical study on ArgoUML project. EVOL/IWPSE 2011: 71-80

5) Yan Ku, Jing Du, Ye Yang, Qing Wang: Estimating software maintenance effort from use cases: An industrial case study. ICSM 2011: 482-491

6) Qi Li, Barry W. Boehm, Ye Yang, Qing Wang: A value-based review process for prioritizing artifacts. ICSSP 2011: 13-22

7) Wen Zhang, Ye Yang, Qing Wang: Handling missing data in software effort prediction with naive Bayes and EM algorithm. PROMISE 2011: 4

2、提交专利申请 1 份，完成软件著作权 3 份；
3、培养硕士生 3 名；

下一年度工作计划，包括国内外合作与交流计划

2012 年度的主要研究计划围绕以下几方面的内容展开：

1、软件工程预测模型的概念漂移问题

在系统调研的基础上，对常见软件成本估算模型和软件缺陷预测模型的概念漂移现象进行汇总和分类。结合已有的概念漂移检测和处理方法，列出处理每种概念漂移现象可能的策略。

2、参数化模型校准中的局部偏差处理方法

计划结合机器学习领域对概念漂移现象进行检测和处理的方法（如样本选择、系综学习），来进一步提升 COCOMO 模型的稳定性和精确性。

3、基于本体的软件估算经验提取方法

进一步完善经验软件工程本体和规则库的构建，完善 COSONT 原型工具。

年度经费使用情况概要

2011 年度经费拨付 40 万元，支出如下：

差旅费：3709 元；国际交流费：4687 元；其他科研业务费：3000 元。

存在的问题、建议及其他需要说明的情况
无。

备注：1、杰出青年人才入选者应实事求是地填写此报告，禁止弄虚作假；
2、此报告作为专项计划跟踪、管理的主要依据，每年报送人力资源处；
3、此报告将在所网站对外发布。