软件众包参与度影响因素分析及预测模型①

安思锦[1,2], 翟 健[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: 随着“众包”这种商业模式的快速发展, 越来越多的互联网公司选择以“众包”的形式发布软件任务. 然而, 软件任务因其高门槛、高复杂度、长周期等特性, 面临着严重的低参与度问题. 本文结合全球最大的软件众包平台 TopCoder 的数据, 对软件众包任务的参与度进行研究. 首先, 使用多元回归分析了影响软件众包参与度的因素; 接着, 综合数据挖掘领域的多种分类预测算法, 探讨软件众包参与度的预测模型. 希望通过本实证研究, 为发包方、众包平台降低软件众包的低参与风险提供参考.

关键词: 软件众包; 参与度; 影响因素; 分析; 预测

Participation Analysis and Predication Model of Crowdsourcing Software Tasks

AN Si-Jin[1,2], ZHAI Jian[1]

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

Abstract: With the rapid development of crowdsourcing, more and more Internet companies choose to crowdsource their software tasks. However, software tasks have their own characteristics, such as high threshold, high complexity, and long period, which make them face serious problem of fewer participants. In this paper, using the data on TopCoder, which is the world's largest crowdsourcing platform for software, we carefully researched the quantity of participants of crowdsourcing software tasks. Firstly, we analyzed the factors affecting participation of crowdsourcing software tasks by multiple regression method. Then, participation prediction model was studied with classification algorithms in data mining area. We hope that this empirical study could help companies or crowdsourcing platforms reduce the risk of low participation in crowdsourcing software tasks.

Key words: crowdsourcing software tasks; participation; factors; analysis; predication

1 引言

2006 年, Jeff 首次提出“众包”一词, 它指的是一个公司或机构把过去由员工执行的工作任务, 以自由自愿的形式外包给非特定的网络大众的做法[1]. “众包”的参与主体有三个: 发包方、接包方和众包平台. 发包方即任务的发布者, 通常是公司或机构; 接包方是为任务做出贡献的非特定网络大众; 众包平台则是发包方与接包方联系的中介.

“众包”可以使发包方以更短的时间、更低的成本获得更高质量的产出[2]. 随着“低成本, 高回报”的优势越来越多的展现在人们面前, “众包”这种商业模式迅速扩张, 并在软件工程领域获得了广泛应用. 国外著名的软件众包平台包括 TopCoder、App Store、uTest、GetACoder、Freelancer 等, 国内也出现了任务中国、猪八戒等平台众包软件开发任务.

“如何吸引用户参与”是“众包”模式面临的一个重大挑战[3–4], 这个问题在软件众包的过程中表现得更加突出. 一方面, 软件众包任务相比于一般众包任务技术要求更高, 往往需要参与者有编程背景; 另一方面, 软件众包任务复杂度更高[5], 决定了参与者要投入更

① 基金项目:国家自然科学基金(61304237)
收稿时间:2015-02-05;收到修改稿时间:2015-04-02

多精力; 再者, 软件众包任务的周期更长, 意味着参与者即使提交了作品, 也需要较长的周期才能拿到奖金. 软件众包任务的这些特征加大了用户参与的难度.

目前, 关于软件众包参与度的研究比较匮乏, 但是存在一些非软件众包任务用户参与动机、参与行为的研究[6]. 通过对 Amazon Mechanical Turk 众包任务参与者的调查, Silberman 等发现奖金对用户的激励作用最大[7], Chandler 等指出用户更愿意参加有意义的任务[8]; Brablham 在对 iStockphoto 社区用户调研后得出: 奖金、提升个人技能和兴趣是用户参与的最大动机[9]; Yang 等结合任务中国众包任务的用户参与数据提出奖金、期限、描述长度、工作量等因素会影响参与人数[10]; Shao 等基于猪八戒众包任务数据分析了奖金、难度、期限、竞争等因素对参与度的影响[11]; Walter 等人的研究认为奖金、期限、平台成熟度等因素不一定会按照预期对众包任务发挥作用[12].

我们结合全球最大软件众包平台 TopCoder 的任务数据, 对软件众包任务的参与度进行研究, 回答以下三个问题: 1.哪些因素会对软件众包任务的参与度产生影响, 如何影响? 2.软件任务发布前, 怎样提前预估任务的参与度? 3.发包方或众包平台应采取怎样的措施, 降低软件众包任务的低参与风险?

本文组织结构如下: 第二部分介绍 TopCoder 平台和实验数据集. 第三部分提出软件众包参与度影响因素的分析方法及实验结果. 第四部分给出建立软件众包参与度预测模型的方法及实验结果. 第五部分总结研究得出结论.

2　经验数据集

2.1 TopCoder 简介

TopCoder 社区成立于 2001 年, 注册用户超过 730,000 人, 定期举行算法竞赛的同时, 以“众包”形式为 Google、Facebook、Amazon、IBM、Microsoft 等客户发布软件任务, 获取盈利.

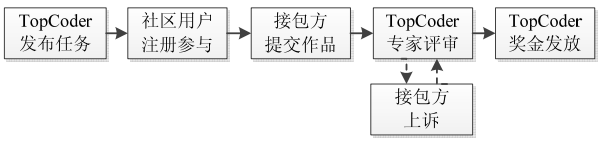


图 1　TopCoder 软件众包任务流程

TopCoder 软件众包流程如图 1. 首先, TopCoder 根据客户需求发布软件众包任务; 接着, 社区用户选择感兴趣的任务注册参与, 本研究将软件众包任务的参与度定义为此阶段的注册人数; 然后, 注册用户依据任务要求在规定的时间内提交软件作品; 任务提交截止后, TopCoder 会安排 2–3 名有经验的专家, 对所有作品进行评审, 接包方如果对评审结果有异议, 可在评审结果发布的 24 小时内上诉; 评审结束后, 优胜的接包方将获得任务奖金.

TopCoder 上软件众包任务的组织架构如图 2. 软件应用的研发过程分为需求分析、体系结构、组件设计、组件开发、组件集成、应用测试等阶段, 前阶段的最优产出作为后阶段输入. 每个阶段不同类型的任务被发布, 如需求分析阶段, 主要发布 Specification 和 Conceptualization 类型的软件众包任务.

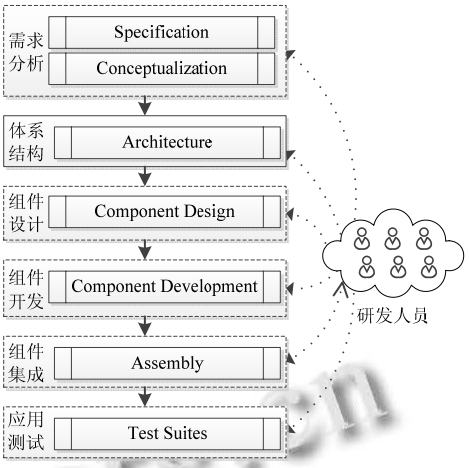


图 2　TopCoder 软件众包任务框架

2.2 数据获取

2003 年 9 月至今, TopCoder 共发布了 3061 个组件设计任务和 3172 个组件开发任务. 其中 1336 个可重用组件, 包括 836 个 Java 组件, 500 个.Net 组件, 设计和开发阶段的产出大部分被 TopCoder 公开.

由于不同阶段软件任务参与度影响因素的度量方法不同, 同时, TopCoder 公开组件数据的开放性最高, 故以 TopCoder 公开组件的开发任务为经验数据集, 研究软件众包任务的参与度.

基于 scrapy 框架, 我们编写爬虫程序爬取了所有公开组件的信息, 包括各个组件对应的设计任务、设计产出、开发任务、开发产出数据. 因为一些组件设计阶段或者开发阶段数据缺失, 共爬取到 932 个组件. 为了计算这些组件对应的平台、竞争等维度数据, 使

用爬虫爬取了 TopCoder 上所有组件开发任务的数据, 一些组件开发任务参数的缺失, 导致共爬取到 3122 个组件开发任务.

3 软件众包参与度影响因素分析

3.1 分析方法

软件众包参与度影响因素分析方法的框架如图 3.

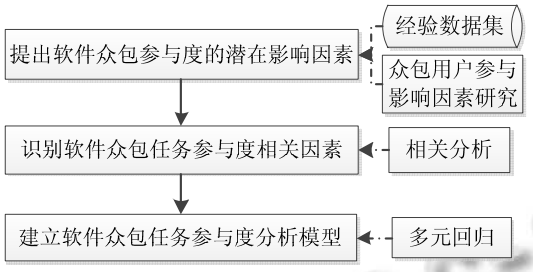


图 3 软件众包参与度影响因素分析方法

首先, 分析经验数据集并结合一般众包任务用户参与影响因素的研究, 从任务参数、任务复杂度、前阶段产出、平台活跃度、同期竞争五个维度提出可能对软件众包任务参与度产生影响的潜在因素.

接着, 计算各潜在因素与参与度间的相关系数, 识别与软件众包任务参与度显著相关的因素. 描述两个变量间相关关系的统计量主要包括 Pearson 相关系数和 Spearman 相关系数[13]. Pearson 相关系数适用于联合分布为二维正态分布的两个随机变量, Spearman 相关系数对变量分布没有要求. 本研究中潜在因素与参与度的联合分布不一定满足正态分布, 因此使用 Spearman 相关系数度量各潜在因素与参与度之间的相关性. Spearman 相关系数计算时需先对数据排秩, 其计算公式如下:

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{1}$$

其中, d_i 表示两个变量第 i 个观测值的秩之差, n 为样本容量.

最后, 使用多元回归方法建立各影响因素与参与度之间的分析模型. 以前一阶段得到的与软件众包参与度显著相关的因素为自变量, 参与度为因变量, 建立如下多元回归模型:

$$\text{参与度} = \beta_0 + \beta_1 \times \text{影响因素}_1 + \beta_2 \times \text{影响因素}_2 + \ldots + \beta_k \times \text{影响因素}_k \tag{2}$$

考虑到与软件众包参与度显著相关的因素间有可能出现彼此相关, 即因素间不独立的情况, 如果直接使用这些因素建立多元回归模型, 模型中将出现多重共线性, 造成回归结果混乱. 为了消除多重共线性对分析结果的干扰, 我们采用具有最优变量筛选效果的逐步回归方法[14,15]建立分析模型. 逐步回归不仅使模型变得更加简单, 还使自变量对因变量的影响清晰地展现出来, 结果更可信, 也更容易解释. 使用逐步回归方法建立软件众包参与度分析模型的过程中自变量逐个引入, 边引入边检查有没有可能剔除某个变量. 第一次变量筛选时, 对 k 个影响因素分别拟合参与度的一元线性回归模型, 共 k 个, 然后找出 F 统计量的值最大, 也即是使得残差平方和减少最显著的模型及影响因素 $_i$, 并将其首先引入模型. 第二次变量筛选时, 在已经引入模型的影响因素 $_i$ 的基础上, 再分别拟合模型外 $k–1$ 个影响因素的二元线性回归, 即自变量组合为影响因素 $_i$+影响因素 $_1$、...、影响因素 $_i$+影响因素 $_{i–1}$、影响因素 $_i$+影响因素 $_{i+1}$、…、影响因素 $_i$+影响因素 $_k$, 因变量为参与度的二元线性回归. 然后挑选出具有最大 F 统计量的模型, 假设其中的两个自变量为影响因素 $_i$ 和影响因素 $_j$, 如果影响因素 $_j$ 对参与度的影响是显著的(变量的 t 统计量检验通过), 那么影响因素 $_j$ 被选入模型. 如果除影响因素 $_i$ 外的 $k–1$ 个影响因素中没有一个是统计上显著的, 则运算终止, 参与度分析模型中的自变量只包括影响因素 $_i$. 如果第二次变量筛选后, 计算未终止, 从第三次变量筛选开始, 变量引入的规则与前面相同(F 统计量最大的模型中的变量被引入), 同时, 会考虑引入一个影响因素后, 之前被引入的某个影响因素是否对模型的贡献变得不显著. 如果是, 这个不显著变量就会被剔除出模型. 逐步回归就是按此方法不停地增加变量并考虑剔除以前增加的变量的可能性, 直至增加变量已经不能导致模型的残差平方和显著减少(模型的 F 统计量检验不通过)或增加任一变量, 该变量对参与度的影响均不显著(变量的 t 统计量检验不通过).

3.2 实验结果

3.2.1 潜在影响因素

我们对 TopCoder 软件任务进行分析, 从任务参数、任务复杂度、前阶段产出、平台活跃度、同期竞争五个维度提出 31 个可能影响软件众包任务参与度的潜在因素. 各因素的度量方法及统计描述属性如表 1 所示, 第 4 列有效 N 表示对应因素上具有有效值的数据量.

3.2.2 Spearman 相关分析

软件众包任务参与度与各潜在影响因素之间的 Spearman 相关系数计算结果如表 1. 在 0.05 的显著性水平下, 与软件众包任务参与度显著相关的因素包括: 类型(如果任务类型为 Java, 值为 1; 任务类型为.Net, 值为 0)、名称长度、描述长度、技术要求数、前阶段参与度、前阶段提交人数、前阶段通过人数、前阶段获胜者排名、活跃提交用户数、近期任务平均参与度、同期同类任务数.

3.2.3 多元逐步回归

经分析, 11 个与软件众包参与度显著相关的因素中, 存在 2 个或 2 个以上因素彼此不独立. 例如, 前阶段提交人数与前阶段通过人数两个因素就高度相关, 二者之间的 Spearman 相关系数为 0.959, 显著性水平小于 0.001.

结合逐步回归方法建立软件众包参与度的分析模型, 共进行了 9 次变量筛选. 前 8 次变量筛选时, 模型中依次引入了前阶段参与度、技术要求数、类型、名称长度、近期任务平均参与度、活跃提交用户数、同期同类任务数、描述长度等 8 个因素, 且没有变量被剔除. 第 9 次变量筛选时, 尝试引入前阶段提交人数、前阶段通过人数、前阶段获胜者排名 3 个因素中的任何一个时, 该因素对参与度均无统计显著性, 即变量的 t 检验不通过, 因此模型建立过程终止, 结果如表 2. 回归模型的总体 P 值小于 0.001, 说明可用多元线性回归分析各因素对参与度的影响; 8 个自变量的 P 值均小于 0.05, 认为这 8 个因素对软件任务参与度的影响是显著的; 各因素的方差膨胀因子(VIF)均小于 1.5, 故模型中不存在多重共线性.

表 1 软件众包任务参与度潜在影响因素

维度	潜在影响因素	度量方法	有效 N	极小值	极大值	均值	标准差	Spearman 相关系数	Spearman 显著性
任务参数	奖金	第一名奖金额度	932	0.00	2500.00	542.70	275.16	-0.038	0.250
	注册期限	发布到注册截止之间天数	932	1.00	7.00	2.81	0.57	-0.056	0.086
	提交期限	发布到提交截止之间天数	932	0.00	14.00	6.78	1.66	0.027	0.415
	类型	开发平台: Java/.Net	932	0.00	1.00	0.60	0.49	0.262	0.000
任务复杂度	版本	任务版次	932	1.00	10.00	1.16	0.59	-0.020	0.549
	名称长度	名称包含单词数	932	1.00	8.00	2.95	1.16	-0.220	0.000
	描述长度	描述包含单词数	932	1.00	311.00	90.52	43.07	-0.066	0.045
	技术要求数	要求的技术数目	914	1.00	8.00	2.04	1.15	-0.298	0.000
	类图数	类图数目	827	1.00	9.00	1.80	1.19	0.029	0.413
	用例图数	用例图数目	820	1.00	5.00	1.19	0.56	-0.012	0.735
	时序图数	时序图数目	784	1.00	45.00	5.35	5.10	0.009	0.793
	需求文档长度	需求文档页数	922	2.00	51.00	3.84	2.33	-0.056	0.090
	规格说明文档长度	规格说明文档页数	828	3.00	64.00	11.92	6.11	-0.040	0.246
	依赖的组件数	依赖的组件数目	932	0.00	19.00	2.84	3.07	0.030	0.359
	工作量	获胜者注册到提交之间天数	926	0.00	470.00	53.05	108.70	0.063	0.054
前阶段产出	前阶段参与度	设计阶段注册人数	931	0.00	38.00	8.92	4.89	0.497	0.000
	前阶段提交人数	设计阶段作品提交人数	931	0.00	26.00	2.70	2.08	0.292	0.000
	前阶段通过人数	设计阶段评审通过的作品数	931	0.00	26.00	2.55	2.00	0.261	0.000
	前阶段产出质量	设计阶段获胜者得分	924	63.06	100.00	90.54	6.59	0.029	0.374
	前阶段获胜者排名	设计阶段获胜者排名	916	402.00	3250.00	1670.45	661.75	0.080	0.016
	前阶段工作量	设计获胜者注册到提交之间天数	921	0.00	469.00	52.39	105.88	0.049	0.140
平台活跃度	活跃任务数	发布前 90 天内, 完成的组件开发任务数	932	6.00	193.00	109.49	46.37	-0.043	0.185
	活跃注册用户数	发布前 90 天内, 注册过组件开发任务的用户数	932	92.00	2988.00	1811.81	690.94	-0.025	0.443
	活跃提交用户数	发布前 90 天内, 提交过组件开发作品的用户数	932	10.00	765.00	450.28	192.93	-0.087	0.008
	近期任务平均参与度	发布前 30 天内, 已完成的组件开发任务平均注册人数	931	2.80	43.38	17.47	4.35	0.197	0.000
	平台成熟度	2003 年 9 月到发布之间月数	932	1.00	99.00	38.86	15.32	0.015	0.643
同期竞争	同期同类任务平均奖金	发布时, 进行中的同类组件开发任务平均奖金	925	96.00	2133.33	560.29	189.85	0.037	0.263
	同期同类任务平均注册期限	发布时, 进行中的同类组件开发任务平均注册期限	932	0.00	24.00	8.66	5.14	-0.130	0.406

同期同类任务平均提交期限	发布时, 进行中的同类组件开发任务平均提交期限	925	1.00	4.57	2.84	0.50	-0.027	0.287
同期同类任务平均工作量	发布时, 进行中的同类组件开发任务平均工作量	925	3.14	30.33	7.23	2.09	-0.035	0.747
同期同类任务数	发布时, 进行中的同类组件开发任务数	925	2.00	464.43	53.43	106.19	0.011	0.000

表 2 回归系数表(P 值 <0.001)

模型	β	t	P 值	VIF
(常量)	10.372	5.404	0.000	
前阶段参与度	0.884	12.931	0.000	1.245
技术要求数	-1.533	-5.597	0.000	1.122
类型	3.886	5.815	0.000	1.196
名称长度	-1.491	-5.411	0.000	1.152
近期任务平均参与度	0.281	3.879	0.000	1.117
活跃提交用户数	0.008	4.330	0.000	1.296
同期同类任务数	-0.197	-2.955	0.003	1.327
描述长度	-0.017	-2.391	0.017	1.051

由多元回归结果可知: 影响软件众包任务参与度的因素包括类型、名称长度、描述长度、技术要数、前阶段参与度、活跃提交用户数、近期任务平均参与度、同期同类任务数. 其中, Java 任务比.Net 任务更受欢迎, 如果软件类型是 Java, 参与度将提高 3.886; 任务名称每增加 1 个英文单词, 参与度降低 1.491; 任务描述每增加 1 个英文单词, 参与度降低 0.017; 技术要求数目每增加 1, 参与度降低 1.533; 前阶段参与度每增加 1, 参与度将提高 0.884; 活跃提交用户数增加 1 时, 参与度提高 0.008; 近期任务平均参与度增加 1 时, 参与度提高 0.281; 同期同类任务数增加 1 时, 参与度降低 0.197.

研究发现, 软件众包参与度的影响因素并未包括一般众包任务用户参与行为的相关文献[7-12]中提出的奖金、期限、工作量等因素. 这是因为: 一方面, 发包方发布任务时, 会根据实际情况对各因素进行合理设置. 如奖金会随着任务的紧迫程度、工作量进行相应的调整. 另一方面, TopCoder 软件任务的奖金、周期相对稳定. 932 个组件开发任务中, 57.9%的任务奖金为\$500, 76.5%的任务注册期限为 3 天, 74.0%的任务提交期限为 7 天.

4 软件众包参与度预测模型研究

4.1 预测方法

软件众包参与度预测模型的建立过程如图 4:

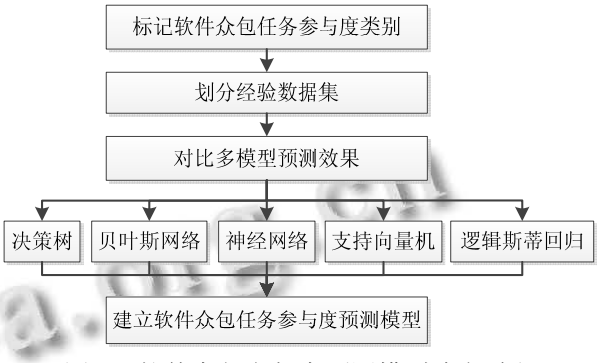


图 4 软件众包参与度预测模型建立过程

首先, 标记软件众包任务参与度所属类别. 注册参与软件众包任务的用户不一定能够在规定的时间内提交软件作品. 众包模式下, 软件作品的提交类似于商品交易中的投标, 中华人民共和国招投标法规定投标人数不应少于三个[16], 本文依据不同参与度条件下软件众包任务平均投标人数的分布情况对参与度进行分类.

然后, 划分经验数据集, 将数据集划分为训练集和检验集, 以便对模型的预测效果进行评估. 我们使用十折交叉验证[17]的方法估计模型正确率, 因此, 把经验数据集随机划分成十个互不相交的子集, 每个子集大小大致相等, 参与度分布基本一致.

接着, 结合数据挖掘领域的分类预测模型对软件众包任务的参与度进行预测, 对比预测效果. 数据分类的基本技术有决策树、贝叶斯网络、神经网络、支持向量机、逻辑斯蒂回归等. 决策树是一种类似于流程图的树结构, 每个内部节点表示在一个属性上的测试, 每个分枝代表一个测试输出, 每个树叶节点存放一个类标号[17], 常用的决策树算法包括 C5.0、Quest、CART 和 CHAID; 贝叶斯网络是朴素贝叶斯分类方法的扩展, 基于贝叶斯定理预测类标号; 神经网络是一组连接的输入、输出单元, 每个连接都有一个权重, 由一个输入层、多个隐藏层和一个输出分类结果的输出层组成; 支持向量机将原始数据映射到更高的维, 在新的维上, 建立最佳分离超平面完成数据分类; 逻辑斯蒂回归基于逻辑斯蒂分布计算分类结果. 评估各模

型的预测效果时, 使用十折交叉验证法, 经过十次迭代, 每次迭代用十个数据子集中的一个作为检验集, 其他子集一起作为训练集来训练模型, 使用训练得到的模型预测检验集中软件任务的参与度. 十次迭代完成后, 正确分类的数据量占总数据量的比例即为该模型的预测正确率.

最后, 以具有最优预测效果的模型为基础建立软件众包任务参与度的预测模型. 如果依据基本数据挖掘算法得到的预测模型较为复杂, 考虑模型简化的方法, 使模型具有更高的实用价值.

4.2 实验结果

4.2.1 参与度分类

TopCoder 公开组件开发任务参与度的分布如图 5. 从图中可以看出, 软件任务的注册参与人数从 0 到 93 不等, 主要集中在 0 到 32 之间, 参与度大于 32 的软件众包任务较少.

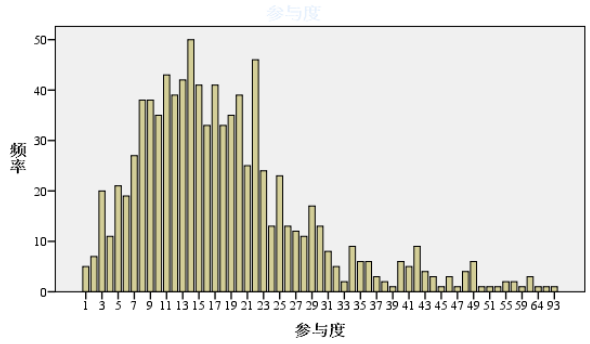


图 5　软件众包参与度频率分布

对不同参与度条件下软件众包的平均投标人数进行分析, 结果如图 6:

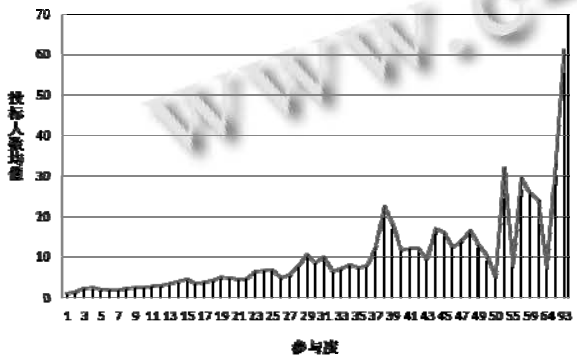


图 6　不同参与度的软件任务竞标人数

当软件众包任务的参与度在 0 到 32 之间变化时, 随着参与度的增加, 投标人数也有不断增加的趋势; 当软件众包任务的参与度大于 32 时, 投标人数的变化趋势波动较大, 这是因为参与度大于 32 的软件众包任务较少, 导致平均投标人数的估计存在偏差. 由于参与度小于等于 11 的软件众包任务平均投标人数均在三个以下, 而参与度大于 11 的软件众包任务的平均投标人数也都大于三个, 因此, 将参与度大于 11 的任务标记为“高”参与度, 记为 C_h, 小于等于 11 的任务标记为“低”参与度, 记为 C_l.

4.2.2 数据划分

剔除在 8 个参与度影响因素上存在缺失值的软件任务, 共得到 912 个组件数据, 随机将其划分成十个互不相交的子集或“折”, 如表 3. 从表中可以看出, 各折数据量分配均匀, C_h 类、C_l 类任务在每“折”数据中的比率与其在总体数据中的比率基本一致.

表 3　数据划分表

折号	C_h 任务数 (比率%)	C_l 任务数 (比率%)	任务总数
1	58(65.17)	31(34.83)	89
2	69(77.53)	20(22.47)	89
3	60(68.18)	28(31.82)	88
4	69(71.88)	27(28.13)	96
5	54(65.85)	28(34.15)	82
6	73(76.04)	23(23.96)	96
7	65(71.43)	26(28.57)	91
8	53(73.61)	19(26.39)	72
9	63(66.32)	32(33.68)	95
10	84(73.68)	30(26.32)	114
合计	648(71.05)	264(28.95)	912

4.2.3 多模型预测对比

结合多种分类模型对众包模式下软件任务的参与度进行预测, 使用十折交叉验证法评估各模型的预测正确率, 结果如下:

表 4　多模型预测效果

模型	正确率(%)	C_h 识别率(%)	C_l 识别率(%)
C5.0 决策树	77.96	81.08	66.32
CART 决策树	70.61	76.76	49.01
QUEST 决策树	70.72	74.71	48.94
CHAID 决策树	70.72	75.71	49.12
贝叶斯网络	73.90	79.80	55.80
神经网络	74.78	79.44	58.42
支持向量机	75.00	78.53	60.23
逻辑斯蒂回归	73.57	77.84	56.35

从图 7 所有预测模型效果的对比可以看出, C5.0 决策树具有最优预测效果, 支持向量机、神经网络、逻辑斯蒂回归模型次之, CART 决策树、QUEST 决策树、CHAID 决策树模型的预测效果最差.

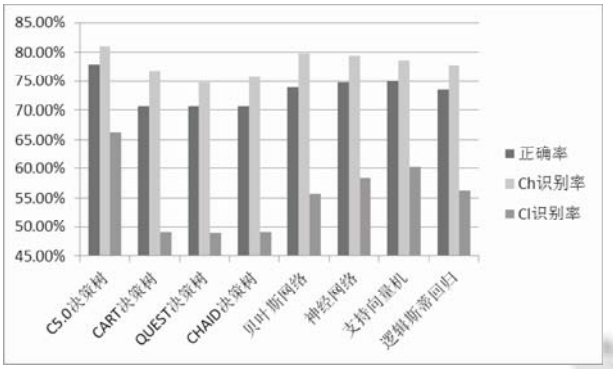
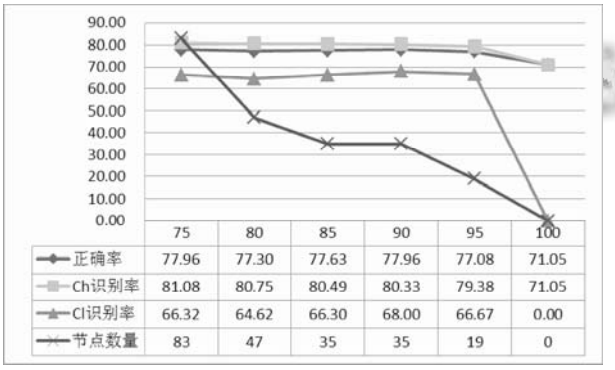


图 7 多模型预测效果对比

4.2.4 预测模型建立

鉴于C5.0决策树在软件众包任务参与度预测效果上表现最优, 在其基础上建立软件众包任务参与度的预测模型.

C5.0 算法在默认条件下构建的决策树复杂度较高、模型难于理解, 因此, 需要对决策树剪枝. 通过提高 C5.0 决策树的修剪纯度, 可获得更小更简洁的决策树[18]. 修剪纯度的取值范围为[0,100], C5.0 决策树修剪纯度的默认值为 75, 提高其值, 对比不同修剪程度下决策树的复杂度、C_h 识别率、C_l 识别率和预测正确率.



	75	80	85	90	95	100
正确率	77.96	77.30	77.63	77.96	77.08	71.05
Ch识别率	81.08	80.75	80.49	80.33	79.38	71.05
Cl识别率	66.32	64.62	66.30	68.00	66.67	0.00
节点数量	83	47	35	35	19	0

图 8 不同剪枝纯度下节点数量及预测效果

由图 8 可知, 随着剪枝纯度的不断增加直至 95, 预测正确率、C_h 识别率、C_l 识别率基本不变, 决策树的节点数量不断减少, 复杂度明显降低. 当剪枝程度增至 100 时, C_l 识别率骤然下降至 0, 且总体预测正确率及 C_h 识别率明显下降.

为了得到尽可能简单、易于理解的参与度预测模型, 将 C5.0 决策树的修剪纯度设定为 95, 得到如图 9 所示决策树预测模型: 从软件众包任务参与度的预测模型可以看出, 各因素对软件众包任务参与度的影响与使用多元回归所得结果基本一致: ①Java 任务比.Net 更受欢迎; ②名称长度、描述长度、技术要求数、同期同类任务数对参与度产生负面影响; ③前阶段参与度、活跃提交用户数、近期任务平均参与度对参与度产生正面影响.

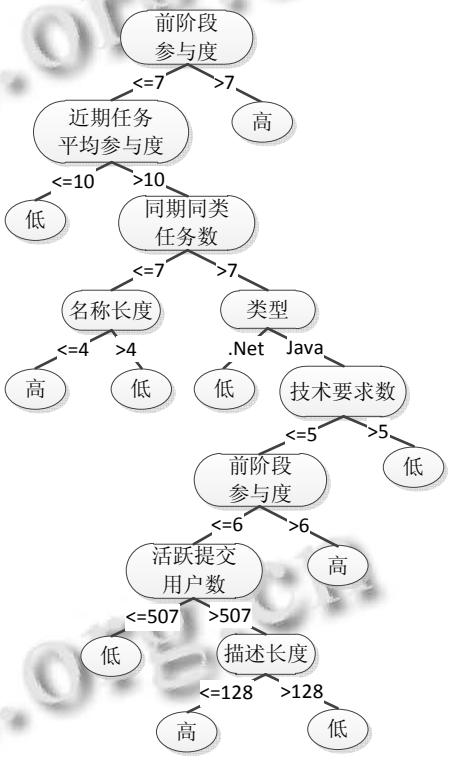


图 9 软件众包参与度预测模型

软件众包任务参与度的预测模型使发包方或众包平台可以在软件任务发布前预估参与度, 减小低参与度风险. 结合软件众包任务参与度的预测模型, 给发包方或众包平台提出如下建议: ①软件应用如果没有平台要求的话, 相比于 C#语言开发, 优先选择发布 Java 任务; ②减少任务名称的长度, 尽量使其小于等于 4 个英文单词; ③尽量用精简的语言把任务描述清楚, 描述单词数控制在 128 个以内; ④尽量降低任务的技术要求数目, 控制在 5 个技术以内, 可通过任务拆分等方式实现; ⑤提高前阶段任务的参与度; ⑥ 在活跃提交用户数大于 507 时发布任务; ⑦选择在任务平

均参与度大于 10 时发布任务; ⑧为了降低同期竞争, 可选择同期同类任务数量小于 7 时发布任务.

5 结论

本文对众包模式下软件任务参与度的影响因素进行了分析, 得到了 8 个影响软件众包任务参与度的因素: 任务类型、名称长度、描述长度、技术要求数、前阶段参与度、活跃提交用户数、近期任务平均参与度、同期同类任务数. 接着, 我们使用分析得到的影响软件众包任务参与度的因素研究了软件众包任务参与度的预测模型, 使得软件众包的发包方或众包平台可以提前预估任务参与度, 并针对降低软件众包任务的低参与风险给出了参考建议.

参考文献

1 Howe J. The rise of crowdsourcing. Wired Magazine, 2006, 14(6): 1–4.

2 Schenk E, Guittard C. Towards a characterization of crowdsourcing practices. Journal of Innovation Economics & Management, 2011 (1): 93–107.

3 Doan A, Ramakrishnan R, Halevy AY. Crowdsourcing systems on the world-wide web. Communications of the ACM, 2011, 54(4): 86–96.

4 Simula H. The rise and fall of crowdsourcing? HICSS ’13 Proc. of the 2013 46th Hawaii International Conference on System Sciences. Washington DC: IEEE Computer Society. 2013. 2783–2791.

5 Wu W, Tsai WT, Li W. Creative software crowdsourcing: from components and algorithm development to project concept formations. International Journal of Creative Computing, 2013, 1(1): 57–91.

6 Yuen MC, King I, Leung KS. A survey of crowdsourcing systems. 2011 IEEE 3rd International Conference on Privacy, Security, Risk and Trust (Passat); 2011 IEEE 3rd International Conference on Social Computing (Socialcom). Boston. IEEE. 2011. 766–773.

7 Silberman M, Irani L, Ross J. Ethics and tactics of professional crowdwork. XRDS: Crossroads, The ACM Magazine for Students, 2010, 17(2): 39–43.

8 Chandler D, Kapelner A. Breaking monotony with meaning: Motivation in crowdsourcing markets. Journal of Economic Behavior & Organization, 2013, 90: 123–133.

9 Brabham DC. Moving the crowd at iStockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application. First Monday, 2008, 13(6).

10 Yang Y, Chen PY, Pavlou P. Open innovation: An empirical study of online contests. ICIS 2009 Proceedings, 2009, 13.

11 Shao B, Shi L, Xu B, et al. Factors affecting participation of solvers in crowdsourcing: an empirical study from China. Electronic Markets, 2012, 22(2): 73–82.

12 Walter TP, Back A. Towards measuring crowdsourcing success: An empirical study on effects of external factors in online idea contest. Proc. from the 6th Mediterranean Conference on Information Systems (MCIS). AIS Electronic Library. 2011. 1–12.

13 高详宝,董寒青.数据分析与 SPSS 应用.第 1 版.北京:清华大学出版社,2007:197–201.

14 马逢时,周暐,刘传冰.六西格玛管理统计指南.第 1 版.北京:中国人民大学出版社,2007:288–297.

15 贾俊平.统计学.第 2 版.北京:清华大学出版社,2006:416–420.

16 琬钟.中华人民共和国招标投标法释义与适用指南.第1版.北京:中国人民公安大学出版社,1999.

17 Jiawei H, Kamber M. Data mining: concepts and techniques. 3rd ed. San Francisco:. Morgan Kaufmann, 2011: 330–350, 370–371.

18 谢邦昌.数据挖掘 Clementine 应用实务.第 1 版.北京:机械工业出版社,2008:195–198.