

# 数据驱动的移动应用用户接受度建模与预测\*

陆璇<sup>1,2</sup>, 陈震鹏<sup>1,2</sup>, 刘讚哲<sup>1,2</sup>, 梅宏<sup>1,2</sup>



<sup>1</sup>(高可信软件技术教育部重点实验室(北京大学), 北京 100871)

<sup>2</sup>(北京大学 软件工程研究所, 北京 100871)

通讯作者: 刘讚哲, E-mail: liuxuanzhe@pku.edu.cn

**摘要:** 应用市场(app market)已经成为互联网环境下软件应用开发和交付的一种主流模式,相对于传统模式,应用市场模式下,软件的交付周期更短,用户的反馈更快,最终用户和开发者之间的联系更加紧密和直接.为应对激烈的竞争和动态演变的用户需求,移动应用开发者必须以快速迭代的方式不断更新应用,修复错误缺陷,完善应用质量,提升用户体验.因此,如何正确和综合理解用户对软件的接受程度(简称用户接受度),是应用市场模式下软件开发需考量的重要因素.近年来兴起的软件解析学(software analytics)关注大数据分析技术在软件行业中的具体应用,对软件生命周期中大规模、多种类的相关数据进行挖掘和分析,被认为是帮助开发者提取有效信息、作出正确决策的有效途径.从软件解析学的角度,首先论证了为移动应用构建综合的用户接受度指标模型的必要性和可行性,并从用户评价数据、操作数据、交互行为数据这3个维度给出基本的用户接受度指标.在此基础上,使用大规模真实数据集,在目标用户群体预测、用户规模预测和更新效果预测等典型的用户接受度指标预测问题中,结合具体指标,提取移动应用生命周期不同阶段的重要特征,以协同过滤、回归融合、概率模型等方法验证用户接受度的可预测性,并讨论了预测结果与特征在移动应用开发过程中可能提供的指导.

**关键词:** 用户接受度;应用市场;移动应用;软件解析学;数据驱动

**中图法分类号:** TP311

中文引用格式: 陆璇,陈震鹏,刘讚哲,梅宏.数据驱动的移动应用用户接受度建模与预测.软件学报,2020,31(11):3364–3379. <http://www.jos.org.cn/1000-9825/6106.htm>

英文引用格式: Lu X, Chen ZP, Liu XZ, Mei H. Data-driven modeling and prediction of user acceptance for mobile apps. Ruan Jian Xue Bao/Journal of Software, 2020,31(11):3364–3379 (in Chinese). <http://www.jos.org.cn/1000-9825/6106.htm>

## Data-driven Modeling and Prediction of User Acceptance for Mobile Apps

LU Xuan<sup>1,2</sup>, CHEN Zhen-Peng<sup>1,2</sup>, LIU Xuan-Zhe<sup>1,2</sup>, MEI Hong<sup>1,2</sup>

<sup>1</sup>(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

<sup>2</sup>(Software Engineering Institute, Peking University, Beijing 100871, China)

**Abstract:** With the popularity of mobile Internet and smart mobile devices in recent years, the app market mode has become one of the main modes of software release. In this mode, app developers have to update their apps rapidly to keep competitive. In comparison with traditional software, the connection between end users and developers of mobile apps is closer with quicker release of software and feedback of users. Understanding and improving user acceptance of mobile apps inevitably becomes one of the main goals for developers to improve their apps. Meanwhile, there is a wealth of data covering different stages of the software cycle of mobile apps in the

\* 基金项目: 广东省重点领域研发计划(2020B010164002); 民航旅客服务智能化应用技术重点实验室开放课题; 国家自然科学基金(J1924032)

Foundation item: Key-area Research and Development Program of Guangdong Province (2020B010164002); Key Laboratory of Intelligent Application Technology for Civil Aviation Passenger Services, CAAC; National Natural Science Foundation of China (J1924032)

收稿时间: 2020-02-07; 修改时间: 2020-05-06; 采用时间: 2020-06-12; jos 在线出版时间: 2020-7-27

app-market-centered ecosystem. From the view of software analytics, with techniques such as machine learning and data mining, valuable information could be extracted from data including operation logs, user behavior sequence, etc. to help developers make decisions. This article first demonstrates the necessity and feasibility of building a comprehensive model of user acceptance indicators for mobile apps from a data-driven perspective, and provides basic indicators from three dimensions of user evaluation, operation, and usage. Furthermore, with large-scale datasets, specific indicators are given in three user acceptance prediction tasks, and features from different stages of the software cycle of mobile apps are extracted. With collaborative filtering, regression models, and probability models, the predictability of user acceptance indicators is verified, and the insight of the prediction results in the mobile app development process is provided.

**Key words:** user acceptance; app market; mobile app; software analytics; data driven

随着计算机软件的运行环境从单机环境发展到互联网环境,软件的发布模式也发生了重要改变.近年来,随着移动互联网的兴起和智能终端设备的大量普及,应用市场模式逐渐成为主要的软件发布模式之一.2019 年第一季度,苹果 App Store 应用市场的应用数量达到 220 万,谷歌 Google Play 应用市场的应用数量则超过了 250 万.与此同时,应用市场模式<sup>[1-4]</sup>以及这一模式下的移动应用开发也成为学术界的研究热点,甚至已经成为软件工程领域各大学术会议中的重要主题.在应用市场模式下,逐渐形成了包括应用市场、应用开发者和用户这 3 类主要角色的生态系统.其中,应用市场作为应用的分发平台,支持开发者提交、管理应用,并提供统计数据、反馈信息等服务;支持用户下载应用,同时提供评价和反馈的渠道.在这一生态系统中,不同角色的活动和交互产生了覆盖软件生命周期多个阶段的丰富数据,包括移动应用的安装包、元信息、开发者信息、用户行为数据等.

相对于传统软件,在应用市场模式下,移动应用的更新迭代周期明显缩短,用户反馈更快,最终用户和开发者之间的联系更加紧密和直接.为了应对激烈的市场竞争,同时满足动态演变的用户需求,移动应用开发者必须以快速迭代的方式不断更新应用,修复错误缺陷,完善应用质量,提升用户体验.因此,如何正确和综合地理解用户对应用的接受程度,即用户接受度,是应用市场模式下软件开发需要考量的重要因素.换言之,对用户接受度的考量,能够为移动应用的开发提供相对具体的指导.例如,通过分析用户评论,可以更好地定位并修复软件中的错误<sup>[5]</sup>;通过量化不同设备型号用户卸载应用的情况,可以帮助开发者优先关注应用在特定设备上的适配性问题<sup>[6]</sup>.开发者以用户为导向进行应用的需求分析、设计实现和发布,以更有针对性、更为高效地实现用户期望的功能和效果.

在应用市场模式下,如何帮助开发者更好地以用户为导向改进应用,是本文关注的问题.近年来兴起的软件解析学(software analytics)<sup>[7,8]</sup>即是针对软件生命周期中产生的大量的、各种类型的数据,运用机器学习、数据挖掘、信息可视化以及大规模数据处理等技术,以数据驱动的方式帮助开发者提取有用信息,做出正确决策<sup>[9]</sup>.基于软件解析学的思想,以移动应用市场生态系统中的大量数据为出发点,可以为移动应用开发者提供改进应用的有效洞察.

用户接受度可以表示用户为满足特定需求接受或使用一个应用的程度.作为移动应用评价指标,用户接受度与传统软件的相关指标存在一定的联系.例如,传统软件的可信性问题关注软件是否安全稳定、能否成功运行并给用户提供服务,包括用户(主体)和软件(客体)两个角度<sup>[10]</sup>.其中,主体角度的定义侧重用户主观感受,强调用户对软件行为的信任、是否符合用户的期望.从满足用户需求的角度看,用户对移动应用的接受度是软件可信性的重要方面.在应用市场模式下,用户接受度的来源数据维度更为丰富,度量和分析大量应用的用户接受度,也为理解移动应用生态下的软件可信性提供了新的视角.

现有研究中,对用户接受度的度量主要依赖于用户评分与评论等相对主观的数据,而对用户操作、交互行为等相对客观的因素考虑不足,导致度量不够翔实和充分,因而难以对应用进行更具针对性的改进.而移动应用市场生态系统中存在的用户评分、评论、管理操作、交互行为等多个维度的数据,都能够体现用户接受度的不同方面.因此,为了更全面地理解用户接受度,从而帮助开发者更好地改进应用,有必要在解决具体的应用改进问题之前,综合考虑应用市场中的各类与用户相关的数据,并进行抽象、分类和建模,建立综合的用户接受度指标模型.在此基础上,本文选取应用开发不同阶段的典型问题,包括需求分析阶段的目标用户群体预测问题、设计开发阶段的用户规模预测问题和版本发布阶段的更新效果预测问题,进行用户接受度指标的提取.从移动应

用市场生态中的丰富数据中选取可能影响用户接受度的特征,建立用户接受度的预测模型以验证用户接受度指标的可预测性,并讨论在不同场景中可能为开发者提供的指导.

本文的主要贡献在于:

1. 针对现有用户接受度指标较为片面和主观的问题,提出数据驱动的用户接受度建模方法,即从应用市场中大规模、多维度的真实用户数据出发,从评价数据、操作数据和交互行为数据中提取具体的用户接受度指标,构造了一个相对综合和全面的用户接受度模型.
2. 选取应用开发不同阶段的典型问题,结合大规模的应用市场数据集,介绍在具体问题中提取用户接受度指标的方法,并结合具体指标选取对应不同开发阶段的重要特征,以协同过滤、回归融合、概率模型等方法建立用户接受度预测模型,验证了用户接受度的可预测性.

## 1 相关工作

用户对移动应用的接受度可以通过用户的评分、评论、管理操作、交互行为等数据体现.现有工作主要将应用评分与评论作为主要的评价指标来源,而对用户行为的研究则多停留于描述性分析层面,即,通过统计检验、可视化等方式试图揭示用户行为中的规律,但未从中提取出度量指标以用于进一步的研究.本节介绍相关工作.

### 1.1 用户评分与评论

在应用市场中,用户可以通过打分、评论机制反馈应用质量和使用感受.本文将评分与评论统称为用户评价.一般地,对于每个应用,应用市场列出不同分数的人数以及加权平均分,同时以一定的规则显示评论,每条评论与对应的分数绑定.相应地,开发者可以对评论进行回复.

以用户评价作为应用接受度的度量指标,目前较常见的做法是结合评分从评论文本中获取有效信息<sup>[15,11]</sup>,在改进应用后再以评分为指导分析改进的效果.然而,用户评价在数据代表性方面存在着局限性:一方面,虽然部分应用市场支持用户对应用的某个版本进行评分,但展现出来的通常是总评分,即用户对所有历史版本的整体评价,这使得研究中无法将评分与版本进行准确对应;另一方面,用户发表评论的倾向并不相同<sup>[12]</sup>,参与评分的用户对于整体来说较为稀疏<sup>[13]</sup>且存在“幸存者偏差”.进一步地,由于应用市场可能覆盖多个国家的用户,用户评论中涉及不同的语言,研究者通常选取自己相对了解的某种语言进行分析.例如,由于英语的自然语言处理技术相对成熟,许多研究者专注于分析英文评论.一些第三方分析机构(如 App Annie)提供对不同语言评论的翻译服务,但翻译质量也受到机器翻译技术的影响.可见,评价机制本身的特性使得可获取的用户样本不能很好地代表整体用户,现有研究方法的局限性则可能进一步加剧这一问题.

研究用户评论通常需要运用自然语言处理技术进行文本处理,而评论文本本身的特性也为研究带来了挑战:首先,与网页、书籍等传统文本形式相比,应用评论基本属于短文本<sup>[13,14]</sup>,而针对短文本的自然语言处理技术依然有很大的提升空间;其次,包含这些短文本在内的用户评论所包含的信息量整体较低,例如“好”“不错”“不好”等,只有小部分用户评论包含可以帮助开发者提升应用的有效信息<sup>[15]</sup>;再次,应用市场中存在一定比例的垃圾评论,其内容不能代表用户的真实想法,甚至与应用本身无关.其中,部分负面评论可能是出自竞争对手,目的是对应用声誉产生负面影响;部分好评则可能来自“地下”推广服务<sup>[16]</sup>,以诱导用户下载应用.现有研究往往对评论文本进行预处理,过滤垃圾评论和无效信息,以提高分析的准确性和有效性.实际上,预处理的效果也极大依赖于自然语言处理技术.

### 1.2 用户操作与交互

在用户主动提交的评价信息之外,用户在操作应用过程中产生的数据(包括应用的下载、卸载等)、在应用中的行为轨迹以及随之产生的衍生数据(例如网络流量)等,能够更为客观地体现用户对应用的接受度.应用市场的公开数据通常只包括应用下载量,因此,通过用户操作与交互行为理解用户接受度,主要依赖于所能获取的相关数据.

用户操作与交互行为数据的获取来源主要可分为 3 类:一是在目标应用内收集,二是通过第三方应用收集,三是通过网络流量数据获取.相较而言,对于第一类情况,开发者可以在权限范围内根据需求定制数据类型,包括用户在应用内的操作路径、点击坐标、操作时间、网络状况等多种粒度的数据,因此,通过该方式可收集的数据粒度最细.

第 2 类数据来源可进一步分为两种.

- 其一是研究者开发专门的工具收集用户数据.早期相关研究一般针对志愿者进行数据收集<sup>[17-20]</sup>,要求志愿者安装工具或使用安装了工具的手机,所获数据规模相对较小.后来,一些研究者<sup>[21-25]</sup>将收集工具通过应用市场发布供用户下载,以获取较大规模的用户数据.
- 其二是借助已有的工具实现对具体数据的获取<sup>[26]</sup>.其中,有些收集行为是与该应用本身功能相关的,如豌豆荚应用市场曾在其历史版本中记录应用的流量使用情况,用于帮助用户检测最耗流量的应用.

第 3 类数据来自网络运营商.目前,绝大多数应用在运行时联网,通过以太网或蜂窝网与服务器产生数据交换.通过分析移动网络运营商存储的流量数据,可以推测用户的应用使用情况,以此分析用户使用应用的行为特性<sup>[27]</sup>.但此类数据也存在一定的局限性,不能简单地将其等同于用户行为的记录.由于有些应用有时可在无网状态下使用,有时会在后台运行时产生网络流量,再加上移动设备所处网络环境较为多样,因此,由某个网络运营商记录的网络流量数据与用户在设备中的交互行为并非一一对应.若要通过这类数据理解用户的应用操作与交互行为,需要对数据集进行深入的分析,给出符合限制条件的结论.

总体而言,对用户行为的研究多停留于描述性分析层面,即,通过统计检验、可视化等方式试图揭示用户行为中的规律,如用户行为的分布特征<sup>[6,28-31]</sup>、用户对应用的使用偏好<sup>[18,20,32]</sup>、不同类型用户的行为差异<sup>[19,33,34]</sup>等.这类描述性分析能够在不同程度上为开发者提供调整与优化的建议,例如,Zhong 等人<sup>[28]</sup>发现,在付费应用中,便宜的应用相对更为成功,同时,昂贵的专业应用会获得不成比例的高下载量,因此建议开发者要为应用找准市场定位.Liu 等人<sup>[6]</sup>在研究中发现,某些应用在特定设备上的卸载率明显较高,因此建议开发者关注应用在这些设备上的适配性问题.然而,由于没有使用具体的应用接受度指标进行进一步的量化分析,此类研究的发现尚不能为开发者提供更为具体的开发指导.

## 2 数据驱动的用户接受度建模

如前所述,由于用户评价数据本身的局限性,有必要同时考虑用户的客观行为数据,为移动应用构建综合的用户接受度模型,从而为开发者提供较为具体的开发与改进指导.移动应用的用户接受度,即用户接受或使用一个移动应用的程度,可以表示为多个指标的集合:

$$User\ Acceptance = \{u_1, u_2, \dots, u_n\}.$$

其中, $u_1, u_2, \dots, u_n$  表示不同维度的用户接受度指标.本文的用户接受度建模过程,就是从不同数据中抽象出多种具体指标,构造用户接受度指标集合的过程.基于这一综合的用户接受度模型,在具体研究中,可以根据研究目标选取不同的指标,以体现用户态度的不同方面,从而为分析软件过程中的具体问题提供指导.需要说明的是:由于数据维度复杂、研究目标多样,能够提取出的用户接受度指标以及开发者关心的用户接受度指标一时间难以穷尽.因此,本文并不试图构建出一个完备的用户接受度指标集合,而是指出综合考虑不同维度用户接受度指标的重要性与必要性,并根据现有应用市场生态中的已知数据给出建模方法和思路.

考虑到应用市场生态中存在大量用户数据,本文采用数据驱动的视角进行用户接受度建模.如前所述,在移动应用市场中,可能反映出用户接受度的用户数据主要包括评价数据、操作数据和交互行为数据.以下分别基于这 3 类数据给出建模方法和思路.

### 2.1 基于评价数据的用户接受度指标

用户评价,指用户显式提供的对应用的评价信息,包括评分和评论,二者通常相互绑定.其中,评分是一种定量的评价表达方式,较为直观且可直接用于不同应用间的比较;评论则通常以自然语言的形式出现,除了能表达用户对应用的整体态度外,还能够表达对应用某个具体方面的看法和需求.相比于传统的在软件内部设置评价

反馈渠道、向部分用户进行访谈或发放调查问卷等方法,应用市场为移动应用开发者提供了大规模获取用户评分与评论的渠道.因此,用户评价数据是一种获取用户接受度的重要来源.

### 2.1.1 评分指标

在不同的应用市场中,用户评分通常以两种模式实现.

- 一种是以 Google Play 为代表的 5 分模型,即用户选择一星~五星作为评分;在这种评分模式下,通常将所有用户评分的平均值作为应用的整体评分.
- 另一种是二元模型,用户可以为应用标记“喜欢(like)”或“不喜欢(dislike)”.由于二元模型无法直接体现喜欢程度或不喜欢程度的差异,可以定义好评率(like ratio)这一指标,表示所有标记用户中标记“喜欢”的用户的比例.好评率计算公式为

$$Like\ Ratio = \frac{number\ of\ likes}{number\ of\ likes + number\ of\ dislikes}.$$

通常情况下,应用市场展示的评分是应用发布以来的历史整体评分.当研究移动应用版本变化相关问题时,使用整体评分作为用户接受度指标存在局限性:一方面,用户给出的往往是对当前使用版本的评分,历史整体评分不能反映对当前版本的接受度<sup>[1]</sup>;另一方面,即便对于稳定的、成熟的、未发生版本更新的应用,其评分也不稳定<sup>[35]</sup>,因而评分的变化无法直接体现用户对不同版本应用接受度的变化.此外,由于用户对应用的更新偏好不同,加之应用市场支持开发者进行灰度更新,不同用户可能同时使用不同版本的应用,因此,简单以版本更新时间作为评分与版本对应的依据亦不准确.为此,本文提出评分趋势作为一种新的用户接受度指标.评分趋势表示应用评分在一段时间内的趋势.若将每天同一时间的评分为点连接为曲线,评分趋势可以通过曲线的斜率表示.根据不同的拟合方法,可以计算出评分趋势的具体数值.当评分趋势发生变化时,可以认为该变化可能与一定时间范围内的版本更新行为有关.对评分趋势的计算涉及到对拟合曲线的转折点检测问题,因而取决于选用的具体算法.本文在第 3.3.2 节中给出一种具体的计算方法.

### 2.1.2 评论指标

基于现有研究,可以从情感、信息量、群体代表性等维度从评论数据中提取用户接受度指标.

1. 情感维度.研究发现,应用评论通常体现出明显的情感倾向,而短评论几乎都是由情感相关单词组成.因此,对评论文本进行情感分析能帮助理解用户对应用的接受度.根据文本,可以定性地将用户态度分为正面、负面、中性等类别.一种常见的做法是从评论中提取关键词,例如,“good”“great”“love”能够代表积极的态度,而“bad”“hate”“terrible”则可以表示消极的态度.同时,也可以结合评分数据,以评分的高低辅助对情感极性的判断.
2. 信息量维度.应用评论中存在一定比例的垃圾评论,因此,过滤常常是进一步分析之前的必经步骤.与此同时,应用市场中的评论所包含的信息量整体较低<sup>[15]</sup>.为减轻这一现状对用户的影响,应用市场也采取了一定的应对措施.例如,Google Play 根据用户对评论的评价对应用评论进行排序.对研究者来说,一方面可以借助应用市场中用户辅助提供的排序结果;另一方面,也可以通过调研归纳<sup>[15]</sup>等方式找到对开发者有效的信息,进而计算每条评论的信息含量.当然,也可以通过主题模型等方法分析评论中的具体信息.进一步地,通过对不同信息的权重分析,可以获得更为具体的发现.例如,基于对不同功能提及概率的分布,可以得知用户更关心哪些功能.
3. 群体代表性维度.除了文本内容之外,用户信息可以体现评论者的分布情况,从而体现不同群体对应用的接受度情况.例如,通过分析评论用户的设备机型,可以获取不同机型用户的接受度情况,从而为选择测试机型提供参考<sup>[36]</sup>.当此类数据可获得时,还可以从不同用户群体入手,细化对情感和信息量的分析.

实际上,现有的基于评论的研究通常并不以抽取定量指标为目的,即便在分析过程中采取一些定量的方法,其目标往往是为开发者提供更为直接的洞察或建议,例如哪些 Bug 出现频率较高,哪些功能需要尽快完善.当然,在研究中如有实际需要,可以结合问题目标和数据质量,通过不同的算法实现对以上维度指标的度量.

## 2.2 基于操作数据的用户接受度指标

操作数据,指记录用户对应用的下载、更新、卸载等管理行为的数据.基于操作行为数据的用户接受度指标可以来自以下 3 个角度.

### 1. 对某一操作进行直接计数.

例如,下载量指应用被下载的次数,是一种常见的用户接受度度量指标.应用市场中显示的应用下载量通常是该应用的历史累积下载量,常以数量级的方式展现.进一步地,基于此类计数结果,可以计算所有用户中不同用户群体所占的比例,以代表这些用户群体间应用接受度的相对大小.例如,对指定应用,可将不同用户群体所占下载量的比例定义为“下载比例(download share)”,计算公式为

$$\text{Download Share}(G_i) = \frac{\text{Downloads}(G_i)}{\sum \text{Downloads}(G_i)}.$$

其中, $\text{Downloads}(G_i)$ 为群体  $G_i$  的下载量, $\sum \text{Downloads}(G_i)$  为所有群体的下载量. $G_i$  的下载比例越高,表明相对于其他群体, $G_i$  对该应用下载量的贡献越大.

当需要根据某一指标对不同应用进行排序时,除了绝对数值,有时需要对相关指标进行转换.例如,考虑到应用市场中应用的下载量分布符合幂率(power law)<sup>[37,38]</sup>,在预测任务中,长尾分布不符合某些评价指标的要求,此时可以用下载排名百分位数(download percentile against rankings,简称 download percentile)表示一个应用在一组应用的下载量排名中所处的相对位置.具体而言,将一组  $n$  个应用按下载量从高到低排序,对于应用  $i$ ,其排名为  $\text{rank}_i$ ,则其下载排名百分位数为

$$\text{Download Percentile}(i) = \frac{n - \text{rank}_i}{n}.$$

在一组应用中,一个应用的下载排名百分位数越高,代表其下载量相对越大.

### 2. 计算不同操作数量的相对规模.

对于某些指标(例如卸载量)来说,由于不同应用的用户规模不同,在比较不同应用的用户接受度时,不宜使用其绝对数值.使用卸载率(U-I ratio)这一相对指标可以表示应用卸载量占下载量的比值,其计算公式为

$$U-I \text{ Ratio} = \frac{\text{number of uninstalls}}{\text{number of installs}}.$$

同理,当一个应用发布新的版本后,可以通过用户的更新率(update ratio)表示进行更新的用户比例.

$$\text{Update Ratio} = \frac{\text{number of updates}}{\text{number of installs}}.$$

### 3. 在基于数量的指标之外,还可以从用户的操作序列中提取用户接受度指标.

例如,“用户在下载后经多次更新最后卸载”与“下载后很短时间内进行卸载”这两种操作序列可以体现出不同的用户接受度.从序列中提取指标的方法,亦取决于具体的问题、研究思路与模式定义方法.

当需要考虑不同时间段或不同版本应用的用户接受度变化时,可以在以上 3 类指标的基础上定义新的指标,例如下载趋势、卸载率变化等.

## 2.3 基于交互行为数据的用户接受度指标

考虑到用户下载应用后可能将其闲置,用户对应用的管理操作行为也不能完全代表用户的实际使用情况.因此,相较而言,用户在应用中的交互行为能够更加直接地体现用户对应用的接受度.然而,相对于可直接通过应用市场公开数据获得的评分、下载量等数据,应用内交互行为数据较难获取,且现有研究基本局限于描述性分析,未考虑进一步从中提取出指标.由于用户交互行为维度非常多样,更难以将可从中提取的用户接受度指标一一列举.本节基于基本的交互行为数据类型,给出若干可用于提取具体指标的方向,在后文的实例中,再根据具体的问题和数据集给出具体指标的定义.

如图 1 所示,用户交互行为的维度非常丰富.基本的交互行为包括应用的启动、运行和退出.当用户启动一个应用时,表明需要该应用来实现某种需求;而退出此应用,则表明当前需求已完成.应用运行既可以在前台也

可以在后台,因此前台运行情况、后台运行情况以及前后台切换情况都能够从不同侧面反映用户的实际使用情况.当应用在前台运行时,用户在应用内的交互行为能够反映用户对功能实现、界面设计等方面的接受程度.相关行为包括点击、滑动、在不同页面中的停留情况、在不同页面间的切换情况等.应用为实现其功能往往需要获取用户设备的若干权限,而用户对权限的许可程度可以反映出对应用要求权限的需要程度以及对其合理性的认可程度.对联网权限这一重要的权限而言,应用运行在前台、后台时的联网时长和流量消耗可以体现用户对该应用的依赖程度.应用也常要求获取用户的推送权限,从而为用户推送消息.对此,用户可以选择推送权限的开或关,在收到推送时,可以点击进入应用,并阅读相关内容.对于使用账号系统的应用来说,用户会进行账号的登录、切换和退出.对于应用内广告,用户可能进行隐藏、点击或进一步的消费.有些应用还会有应用内商品供用户购买.

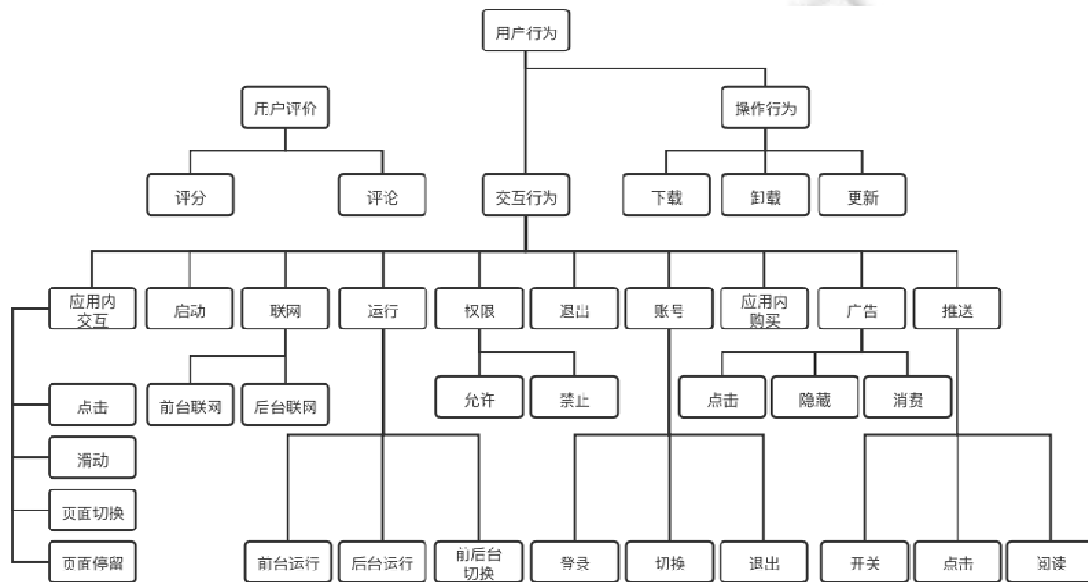


Fig.1 Main user data types in app market

图1 应用市场中的主要用户数据类型

在研究中使用的应用交互行为数据的粒度主要由收集者决定,相应地,能够提取的用户接受度指标取决于具体能够获取的数据.以本文后文中所使用的豌豆荚数据集为例,豌豆荚客户端为帮助用户进行流量统计,在获取权限的前提下,记录设备中不同应用的流量使用情况.该数据从3个维度进行记录,包括前台/后台、Wi-Fi/蜂窝网、时长/流量,共以8个字段表示,具体为前台Wi-Fi联网时长、前台Wi-Fi联网流量、前台蜂窝网联网时长、前台蜂窝网联网流量、后台Wi-Fi联网时长、后台Wi-Fi联网流量、后台蜂窝网联网时长和后台蜂窝网联网流量.基于这些数据,可以提取出不同的用户接受度指标.例如,将前台Wi-Fi联网时长与前台蜂窝网联网时长相加,可以得到前台联网时长(或在线使用时长).对于需要联网的应用来说,这一指标与前台使用时长相近,可以代表用户使用应用、与应用进行交互的程度.

### 3 移动应用用户接受度预测典型问题

数据驱动的用户接受度建模技术,简单而言即是基于大规模、多维度的数据提取出可用于指导应用开发的用户接受度指标,能够根据开发者的实际开发需求提供具有指导价值的用户接受度指标.前文基于评价数据、操作数据和交互行为数据的建模方案即是给出了提取用户接受度指标的方法和思路.与此同时,正是由于应用市场生态中可用于评价用户接受度的数据的多维性与复杂性,从中能够提取的用户接受度指标难以一一列举.实际上,脱离了实际需求的用户接受度指标缺乏实际意义.对开发者来说,当目标接受度指标未知时,可以通过

建立预测模型构建接受度指标与不同因素之间的关联,从而为改进开发过程提供指导.因此,本文根据移动应用开发者在开发不同阶段可能面临的典型问题,选取合适的用户接受度指标,通过问题描述、指标提取、指标预测等步骤给出 3 个实例.基于真实数据集,有针对性地进行用户接受度指标的提取,并通过验证用户接受度指标的可预测性,讨论这些指标可能为应用开发带来的改进与帮助.

### 3.1 目标用户群体预测

#### 3.1.1 问题描述

在应用开发之初,开发者需要对应用的目标用户群体进行分析和理解,从而确定应用的功能定位,并为后续的开发和测试工作提供指导.实际上,在开发测试中遇到的问题很可能与应用的目标用户群体有关.例如,Android 应用开发者通常需要考虑应用在不同 Android 设备上的适配性问题.由于 Android 系统的开源特性,不同的生产厂家生产出大量不同型号的 Android 设备,在操作系统版本、硬件配置、调用接口设计等方面千差万别.根据 Open Signal 的报告,2015 年,市场上已有超过 2.4 万种 Android 设备.在这一背景下,开发者往往需要对不同机型上的适配性问题进行穷举并逐个解决.然而,由于经济、人力、时间等各方面成本的限制,开发者通常无法穷尽列举所有的情况.因此,若能在大量 Android 设备中定位出主要的设备,尽可能多地覆盖目标用户,则能够帮助开发者显著减少开发与测试成本.

目前,应用开发者最常用的做法是根据市场情况,选取销量最好的若干种设备用于测试.这一做法存在明显的问题,即不同应用的目标用户群体不一定相同.换言之,销量最好的设备型号不一定能覆盖某个特定应用的主要用户群体.因此在应用上线之前,需要一种更好的方案为移动应用进行主要用户群体的预测.具体而言,对于一个特定应用,需要提前预测出其主要用户所使用的若干种设备型号,从而在开发测试阶段尽可能解决可能遇到的适配性问题.

#### 3.1.2 指标选择

判断使用某种设备的用户是否属于应用的主要用户群体,其标准亦取决于开发者所关注的用户接受度指标.当选定该指标后,可以通过预测其在不同机型用户中的分布情况来判断不同机型用户对应用的重要程度.在线广告投放策略中,一种常见的理念可供借鉴,即用户在页面中停留的时间越长,其点击广告的概率越大.换言之,不同用户的在线使用时长能够有效衡量其对应用的重要程度.同时,考虑到用户遇到适配性问题的几率可能与其使用时长相关,因此选取在线使用时长作为用户接受度指标.在线使用时长指用户在前台使用该应用且访问互联网的时长,此处联网包括 Wi-Fi 和蜂窝网络.对于应用  $A$ ,由设备  $D$  贡献的在线使用时长定义如下.

$$Time(D \rightarrow A) = \sum Time(d_i \rightarrow A).$$

其中, $D$  代表一款特定的机型, $d_i$  代表属于机型  $D$  的一个具体设备, $d_i \in D$ .

为了体现不同机型对在线使用时长的贡献程度,以每款机型所占在线使用总时长的比例作为不同机型重要性的度量指标,简称为“时长比例(time share)”,定义如下.

$$Time Share(G_j) = \frac{Time(D_j \rightarrow A)}{\sum Time(D_j \rightarrow A)}.$$

其中, $Time(D_j)$  为机型  $D_j$  的在线使用时长, $\sum Time(D_j)$  是所有机型的在线使用时长(也即所有用户的在线使用时长). $D_j$  的时长比例越高, $D_j$  的用户在此应用中消耗的在线时间越长.这样,不同机型用户群体对应用的接受度通过在线使用时长来度量.据此抽象得到的时长比例,则表示不同用户群体对应用的重要程度.

#### 3.1.3 指标预测

此处数据集为豌豆荚收集的 238 231 个应用在 2014 年 7 月 1 日~9 月 30 日期间的用户行为数据,覆盖 4 775 293 名用户和 16 602 种 Android 设备.在这一数据集中,选用前台 Wi-Fi 联网时长和前台蜂窝网联网时长计算不同应用的用户接受度.初步分析显示:在线使用时长在不同机型上的分布情况具有帕累托分布的特征,即少量机型可以覆盖大量的交互行为数据.如图 2 所示,在游戏类应用 11 538 种机型中,要覆盖使用总时长的 90%,只需要其中至多 4.3% 的机型.要覆盖 80% 的时长,需要的机型种类的中位数低于 100.这证实了为应用开发者显



著缩小机型选择空间的可能性.

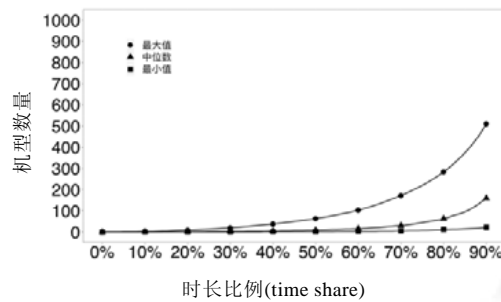


Fig.2 Number of device models covering X% time share of game apps

图 2 覆盖游戏类应用 X% 时长比例需要的机型数量

由于应用上线之前没有足够的用户数据,无法根据应用自身的历史信息预测未来的情况,运用协同过滤思想,使用相似应用的用户数据来预测目标应用的用户数据.根据具有相同特征的应用中不同机型用户的接受度分布,预测该应用的用户接受度分布情况.具体而言,对于应用 A,选取同类别的  $K-1$  个应用,根据在这  $K-1$  个应用中不同机型用户接受度的分布情况预测应用 A 的  $N$  个主要机型,采用  $K$  折交叉验证(此处等同于留一验证<sup>[39]</sup>)算法验证预测效果.在  $K$  个应用中,选择某个应用作为应用 A,对其余的  $K-1$  个应用,将同一机型的在线使用时长进行加总,计算出不同机型的在线使用时长,并选取排名前  $N$  的机型,预测其为应用 A 的主要机型.以机型命中数、时长覆盖率、平均精准度作为评价指标,其中,机型命中数表示在实际的  $N$  种主要机型中命中的数量,时长覆盖率表示预测出的  $N$  种机型的在线使用时长占实际时长的比例,平均精准度(average precision)则同时考虑了  $N$  种机型的排序情况.每次选取不同的应用,将这一过程重复  $K$  次,并对以上 3 种评价指标进行统计.选定  $K=100, N=10$ ,对豌豆荚数据集中包含的 14 类应用实验结果见表 1,其中,指标分别为机型命中数的中位数、时长覆盖率的中位数和平均精准度的平均数(mean average precision).

Table 1 Prediction results of main device models for different app categories

表 1 不同类别应用主要机型预测结果

应用类别	机型命中数(中位数)			时长覆盖率(中位数)			平均精准度(平均数)		
	算法 1	算法 2	算法 3	算法 1	算法 2	算法 3	算法 1	算法 2	算法 3
商务	5	4	5	82.52	69.01	74.29	0.49	0.29	0.40
媒体	8	5	6.5	96.7	77.84	90.85	0.72	0.39	0.54
购物	7	5	5	91.68	76.44	85.64	0.62	0.33	0.47
通讯	6	5	5	85.92	73.31	81.71	0.53	0.33	0.44
母婴	5	4	5	76.81	62.87	73.24	0.47	0.27	0.41
社交	7	5	5	91.61	71.84	83.65	0.61	0.33	0.45
财务	7	5	6	90.58	77.11	84.43	0.63	0.38	0.49
新闻	7	4	5	90.31	73.13	79.14	0.58	0.32	0.43
工具	7	5	6	92.09	78.4	87.23	0.55	0.35	0.48
游戏	8	5	7	97.35	76.36	92.09	0.75	0.39	0.62
效率	4.5	4	4	61.05	57.62	60.75	0.40	0.27	0.36
旅行	8	5	6	94.77	80.59	88.34	0.69	0.41	0.50
生活	7	5	6	93.30	80.18	87.90	0.61	0.40	0.48
教育	7	5	6	93.20	73.43	85.70	0.65	0.38	0.52

作为对比,将上述预测算法计作算法 1,同时选用两种基于市场占有率的机型预测方法,分别计作算法 2 和算法 3,同样计算 3 项指标.其中,算法 2 选用 AppBrain 在同一季度的 Android 设备销量排名前 10 的机型,对每款应用均以此 10 种机型预测其排名前 10 的机型,即对任一应用 A,预测其排名前 10 的机型依次是 AppBrain 公布的销量排名前 10 的机型.算法 3 则使用豌豆荚数据集统计出的市场占有率排名前 10 的机型,以消除数据集不同带来的影响.同样,对每款应用均以此 10 种机型进行预测.

结合 3 种有效性指标可知,算法 3 的预测效果优于算法 2.即使对于机型命中数相同的应用类别,如购物、通讯、社交,算法 3 的时长覆盖率和平均精准度均明显高于算法 2.这一结果表明,即使命中的机型数相同,算法 3 命中的机型排名更为靠前.这两种算法的主要区别在于统计市场占有率的数据集来源不同,因此预测目标用户群体时,应考虑数据集来源因素.

将算法 1 和算法 2 的预测效果进行比较可以发现,在所有应用类别上,算法 1 的机型命中数、时长覆盖率、平均精准度均优于算法 2.与算法 3 比较可知,算法 1 的预测效果依然更优.即使在商务和母婴类别中,算法 1 的机型命中数(中位数)与算法 3 相同,均为 5 种,但算法 1 的时长覆盖率和平均精准度依然高于算法 3.这表明算法 1 命中的机型更为重要,能够帮助开发者定位出更为重要的目标用户群体.

比较算法 1 在不同应用类型上的表现可知,算法 1 在大部分应用类型中可以取得较好的预测效果.例如在游戏类应用中,算法 1 能够准确预测前 10 种主要机型中的 8 种,时长覆盖率达到 97.35%,平均精准度为 0.75,均明显优于两种基准算法.对于 14 种应用类别中的 10 种,算法 1 可以达到的时长覆盖率均超过 90%.从应用接受度的角度看,覆盖越多使用时长的机型(用户群体)更为重要,因此,算法 1 能够有效帮助这些类别中应用的开发者预测其主要用户群体.相对而言,算法 1 在效率类应用中的预测效果较差,这表明在该类别内部,不同应用在不同机型上的在线使用时长分布的相似性较低,不易通过同类应用的数据分布情况进行预测.

在本实例中,以在线使用时长作为用户接受度指标,通过相似应用在不同设备型号上的用户接受度分布情况,以协同过滤的思想预测新应用的用户群体分布情况,从而有效地识别出应用的目标用户群体.通过这一方法,可以帮助开发者在给定的资源下更好地选择设备机型进行开发和测试,从而优化资源配置,优先解决覆盖更多用户的适配性问题,提升用户体验.

## 3.2 用户规模预测

### 3.2.1 问题描述

以下载量为主要度量指标的用户规模,常常是体现应用成功度的重要方面.对应用开发者来说,越早了解应用可能获取的用户规模,在软件开发过程中越能够灵活地进行调整和改进.然而当前的应用发布模式下,开发者需要在发布应用之后才能获取用户反馈数据.具体而言,开发者在应用开发过程中常常面临不同的决策问题,包括是否选用某种设计、是否实现某一功能、以何种方式实现等等.一种常见的做法是以 A/B 测试的方式度量用户的反馈情况,即同时实现并发布不同版本的应用,通过分析用户接受度指标的数值及其变化来决定采取哪种方案.这一做法基于应用上线后获取的用户数据进行.然而,即便可以同时上线不同的版本,依然需要开发者对大量因素进行决策,只保留有限的变量.例如,同时上线的两个版本只在某个特定功能的布局上有所不同,其他因素完全相同,从而通过用户对不同版本的反馈情况对功能布局方案进行选择.这一过程中,对其他需要确定的因素的决策依然在很大程度上依赖于开发人员的直觉与经验.同时,每个版本的实现都需要需求分析、设计、开发、审核等多方面的成本.

因此,若能在应用上线之前对其可能获得的用户规模进行预测,则能够大大降低成本和对开发人员的决策依赖.通过对可控因素与用户接受度关联性的挖掘,可以在开发阶段即对应用进行有针对性的改进.实际上,用户规模预测也是研究者关心的问题<sup>[40]</sup>.

### 3.2.2 指标选择

开发者所关心的用户规模通常以下载量作为主要度量指标.由于影响下载量的因素非常复杂,预测其绝对值具有相当的难度.本例选取下载量的相对值,即在同类应用中的排名情况作为接受度指标之一.与此同时,仅以下载量作为评价维度也存在局限性:一方面,用户在下载应用后可能卸载,因此引入卸载率,将用户的负面态度纳入考量;另一方面,基于评价数据的指标依然为大部分开发者和研究者所关心,因此,同时选用好评率作为指标的补充.

### 3.2.3 指标预测

在确定目标指标后,需要提取应用上线之前可获得的数据特征来预测应用的用户接受度.这些数据特征来自应用市场中可获得的安装包和描述信息,属于开发者可控因素.具体而言,从安装包大小、代码复杂度、代码

质量、库依赖、设备/用户要求、能耗处理、界面复杂度、应用描述、市场指标这 9 个维度选出共计 108 项数据特征,采用多种回归模型融合的方法对用户接受度进行预测,并挖掘数据特征与用户接受度之间的线性与非线性关联。

同样选用豌豆荚数据集,包括 2014 年 5 月 1 日~9 月 30 日用户的下载、更新、卸载等操作行为数据.在爬取对应的安装包并进行反编译等处理后,获取包括第三方库<sup>[41]</sup>在内的代码相关指标.进一步爬取应用的网页描述信息,并提取数据特征和用户接受度指标.最终获得了 9 824 个应用的完整有效信息,用于预测模型的训练和测试.将数据集随机分为 3 个子集,即一个包含 9 000 个应用的训练集,一个包含 500 个应用的测试集和一个包含 324 个应用的验证集。

在成功提取 108 种特征之后,一个简单的做法是直接这些特征训练模型.考虑到一些特征可能彼此之间高度相关,从而导致共线性等问题,严重影响模型的鲁棒性和个体特征的可解释性,因此在训练预测模型之前,先对特征进行相关性分析,并以此为基础进行过滤,最后得到 79 个特征用于后续预测。

为了达到指导开发的目的,需要预测模型具有较好的可解释性,因此选取经典的机器学习算法,包括 Lasso 回归、岭回归和随机森林算法,来预测应用的用户接受度.其中,Lasso 回归<sup>[42]</sup>和岭回归<sup>[43]</sup>是标准的线性回归模型,采用相同的损失函数,但使用不同的正则项.模型学到的变量系数能够体现相应特征的重要性.由于在训练过程中,Lasso 回归模型会对特征进行选择,而岭回归模型不会,因此二者生成的变量系数会有所不同.但由于两种回归模型产生的系数的正负符号基本一致,因此对符号的判断更为可靠.进一步地,由于有些特征可能会与用户接受度指标之间存在非线性关系,因此在线性模型之外引入随机森林算法<sup>[44]</sup>,在解决非线性回归问题的同时,保持变量的可解释性.均以均方误差(mean square error,简称 MSE)和肯德尔等级相关系数(Kendall's tau<sup>[45]</sup>,简称  $\tau$  值)作为模型评价指标。

作为比较,采用随机猜测算法作为基线算法.例如,对于测试集中的每个应用,可以简单猜测其卸载率为 0,并计算相应的 MSE;同样地,也可以简单猜测卸载率为 0.5 或 1.在不同的猜测策略中,选择效果最好的一种(即产生最低的 MSE)作为比较基线.在排名效果的比较方面,对测试集中的应用进行随机排序,计算  $\tau$  值后,与实际观测值比较.实验结果可见表 2。

Table 2 Prediction results of user acceptance

表 2 用户接受度预测结果

预测模型	下载量(download percentile)		卸载率(U-I ratio)		好评率(like ratio)	
	$\tau$ 值	MSE	$\tau$ 值	MSE	$\tau$ 值	MSE
Lasso 回归	0.236	0.080	0.280	0.028	0.150	0.054
岭回归	0.220	0.084	0.314	0.027	0.135	0.061
随机森林	0.316	0.070	0.331	0.025	0.252	0.047
基线算法	-0.011	0.091	-0.016	0.075	0.010	0.075

从实验结果来看,MSE 越低、 $\tau$ 值越高,表明算法效果越好.因此 Lasso 回归、岭回归和随机森林算法的模型效果均明显优于基线算法,这验证了基于应用特征建立的用户接受度预测模型的有效性.其中,从下载量、卸载率和好评率来看,随机森林算法的效果均优于两种线性回归模型。

在通过预测模型建立数据特征与用户接受度指标之间的关联后,可以通过识别单个特征对每个指标的影响来验证特征与指标之间的相关性.具体而言,在使用 R 语言训练随机森林模型后,可以通过 `rf$importance` 指令得到每个特征的 `%IncMSE` 数值,表明当该特征发生重新排列时,MSE 发生的变化.`%IncMSE` 值越大,表明相应变量对模型输出越为重要.对于随机森林模型选出的重要变量,若该变量也被 Lasso 模型选择,则可以进一步判断其相关性方向.通过这一方法,为 3 个用户接受度指标选出了包括 SDK 版本、调用 Android API 的数量、要求权限的数量、代码耦合程度、描述文本长度等 44 个特征.通过进一步分析每项特征与具体指标之间的相关性,可以为开发者提供改进应用的建议.例如,应用市场中介绍图片的数量、文本描述的长度、描述标签的个数都与用户下载量之间存在正相关,这表明开发者可以使用更多的图片、更为丰富的描述信息来展示应用的功能特性.又如,调用 Android API 的数量、目标 SDK 的版本都与用户接受度之间存在正相关,开发者可以考虑更多地

使用 Android 原生 API 并开发面向较新 SDK 版本的应用。

当然,通过上述方法挖掘出的是特征与用户接受度指标之间的相关性,而非因果性.要得到因果性结论依然需要进行严谨的实验分析.然而,考虑到可能影响用户接受度的因素规模,通过构建预测模型发现潜在因素的方法,已经为开发者大幅缩小了考虑范围,提供了较为具体的分析方向.

### 3.3 更新效果预测

#### 3.3.1 问题描述

在应用市场生态下,移动应用常常需要快速迭代,以满足用户的需求,提升应用的竞争力.通过应用市场进行应用更新时,开发者在市场中发布新版本的安装包,并更新应用名称、描述等信息.然而,应用的更新策略,包括应用发布时间、更新描述等,都有可能影响用户对新版本的接受度.特别地,当需要更新的内容已确定时,如何选择最优的发布时间以达到相对最优的更新效果,是应用开发者关心的问题.

#### 3.3.2 指标选择

当应用发生版本更新后,不同维度的用户接受度都有可能发生变化.本节从评分的角度刻画用户对不同版本的接受度.考虑到评分本身存在不稳定性<sup>[35]</sup>,为减少不稳定性带来的影响,选取应用在一段时间内的评分趋势作为用户接受度指标,并只考虑评分趋势发生显著变化时的情况.即进行一次更新后,只有在一定时间范围内出现了评分趋势的显著变化,才认为此次更新行为对用户接受度产生了影响.以根据这一规则选取的更新作为研究对象,预测更新行为可能带来的评分趋势的变化.

具体而言,将同一应用每一天的评分连接得到的评分曲线拟合为首尾连接的若干条线段,每条线段的斜率表示相应时间内的评分趋势,连接点则表示评分趋势发生了转折.将此类连接点称为显著转折点(turning points).为检测出显著转折点,可以构造出一个分割问题,即通过用若干个片段最优地表示向量  $T$ <sup>[46,47]</sup>,并使用自顶向下在线算法为  $T$  训练模型<sup>[48]</sup>并给出最优的分割方案.图 3 即展示了一个应用的评分趋势转折点检测结果.

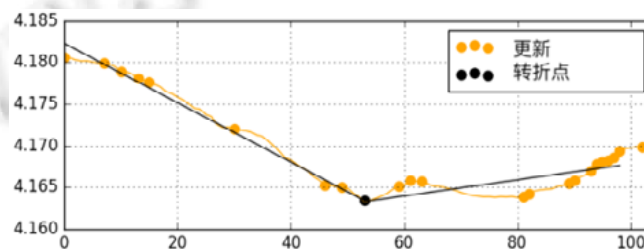


Fig.3 An example of turning points in rating trends

图 3 评分趋势中的显著转折点(示例)

当应用发布新版本后,用户需要时间在自己的设备上更新,并熟悉新的特性,因此对新版本的接受度需要一定时间才能够通过评分等方式体现出来.所以,应用的评分曲线与更新曲线之间存在时滞效应.换言之,当显著转折点已知时,需要找到转折点之前的一段时间区间,在该区间内发生的应用版本的更新可能与评分的转折有关.这段区间称为时滞区间(time-lag slot),而区间内的更新称为显著更新.对于显著更新,计算对应转折点前后的评分拟合曲线斜率之差以度量评分趋势的变化.

为了识别时滞效应,首先将更新点和评分抽象为不同的序列向量,并进行平滑处理以去除噪音;其次,使用皮尔逊积矩相关系数(Pearson product-moment correlation coefficient,简称 PPCC)<sup>[49]</sup>计算在不同时滞区间下版本更新与评分之间的相关程度.对每个应用进行时滞区间检测<sup>[50]</sup>,并计算每一对向量之间的 PPCC 值.当  $p \leq 0.01$  时,认为更新向量与评分向量之间存在显著相关.通过计算每个应用在不同时滞区间下显著相关向量的百分比,发现在区间长度为 4 天时这一比值最高,达到 48%.这验证了更新行为与评分的反应之间存在时滞效应,同时也表明,应用评分的变化有可能与变化发生之前,时滞区间内的更新行为相关.

### 3.3.3 指标预测

要建立用户接受度指标预测模型,需要先提取出可能影响指标的因素.此处考虑 3 类因素:更新间隔、更新目的和市场排名.现有研究通常以更新频率<sup>[51]</sup>来刻画应用更新的时间,然而两个应用即便更新频率相同,其更新时间的分布也可能明显不同<sup>[52]</sup>.因此,选用更新间隔表示发布新版本的日期距前一次发布的天数.这样,每个应用的更新历史可以表示为更新间隔的序列.更新目的,顾名思义,指发布新版本的目的.常见的目的包括修复 Bug 和增加新功能<sup>[35]</sup>.用户对修复已有问题或增改功能特性的迫切程度可能不同,因而应用版本的更新目的有可能影响更新的效果.通过处理更新描述文本(what's new),用 TF.IDF 模型<sup>[53]</sup>生成检索词向量,可以表示每次更新目的的分布情况.市场排名指应用在同类应用中所处的排名,该排名由应用市场综合评估给出.考虑到版本更新会影响应用市场对该应用的评估,市场因素对更新效果的影响不容忽视,而市场评估规则对一般开发者来说并不透明,因此以可获得的排名来反映市场因素.

此处使用的数据集来自 Google Play,包含来自 33 个类别的 17 820 个应用,覆盖 2016 年 11 月 25 日~2017 年 3 月 10 日.为获取到具有较为丰富的版本更新历史和可靠评分信息的应用,进而减轻应用取样问题<sup>[54]</sup>,本数据集选择的应用是每个类别中排名前 540 的应用.对每个应用爬取排名、评分、更新描述以及版本信息.当版本信息发生变化时,认为应用发生了一次版本更新.最终,数据集中共包含 41 028 次更新.

以应用评分趋势作为用户接受度的度量,以发布间隔、发布目的、市场排名以及应用当前的评分趋势为自变量,建立用户接受度预测模型.模型分为 3 部分.

- 首先,建立特征变量与发布效果的关联模型.考虑到特征变量的离散性,选择多项式朴素贝叶斯(multinomial naive Bayes, 简称为 MNB)模型<sup>[55]</sup>.以  $E$  表示一次显著发布的效果,  $E$  为浮点数,其正负代表发布效果的方向;以  $r$  表示发布时应用的市场排名;以  $t$  表示计划的发布间隔;以  $s$  表示在此次发布前检测到的评分曲线片段的斜率,代表当前的评分趋势;以  $C$  表示描述文本中检索词的重要性向量.模型可表示为

$$E_i = \text{MultinomialNB}(r_i, t_i, s_i, C_i) (i \in I).$$

- 其次,预测用户接受度的变化.为验证模型可靠性,通过五折交叉验证(5-fold cross validation)<sup>[39]</sup>预测应用更新效果的方向,实验表明,准确率可达到 76%.在基线算法中,使用只包含检索词向量  $C$  为特征的模型,准确率达到 54%;进一步使用包含检索词向量  $C$ 、市场排名  $r$ 、当前评分趋势  $s$  为特征的模型,准确率达到 70%.本模型的 76% 准确率显著高于基线算法,表明综合考虑包括更新间隔在内的特征对预测更新效果的重要性.
- 最后,优化发布策略.开发者通常需要一个合适的时机来发布新版本的应用.在不考虑外部因素的情况下,上述模型可以帮助开发者选择更新间隔以优化发布策略.具体地,给定应用当前的排名  $r$ 、评分趋势  $s$  和检索词向量  $C$ ,需要找到使得评分趋势变化  $E$  最高的间隔  $t$ .此优化任务可形式化表示为

$$t_{best} = \max_{t' \in \{0, 1, \dots, \infty\}} \{\text{MultinomialNB}(r, t', s, C)\} (i \in I).$$

其输出  $t_{best}$  代表在给定排名、评分趋势、更新目的的情况下,使得评分变化最优的更新时间间隔.

当然,应用的更新策略是开发者综合多种因素进行权衡的结果,很多因素,包括市场环境、应用品质、开发资源等,也可能影响实际的更新效果.通过预测模型揭示更新行为与更新效果之间的潜在关联之后,优化策略的实际效果还需要在真实环境下进行进一步的实验验证.

## 4 总结与展望

在应用市场生态下,由于与应用开发者之间紧密和直接的联系,并且产生了大量用户数据可供收集和分析,用户在移动应用的开发、测试、发布等不同阶段扮演着重要的角色.在用户主动提交的评价信息之外,如何同时利用用户的操作信息和交互行为信息,从不同角度刻画用户对应用的接受度,构建综合的应用接受度指标,进而为开发过程提供多方位的指导,是本文探讨的主要问题.本文从应用市场生态中的用户数据出发,构建了一个综合的用户接受度模型,并介绍了具体指标的计算方法.进一步地,基于软件解析学的思想,运用应用市场生态

中的丰富数据,刻画了移动应用软件生命周期不同阶段的重要特征.通过应用开发不同阶段的3个典型问题,讨论了针对具体问题如何提取合适的用户接受度指标,并使用协同过滤、机器学习、概率模型等方法建立用户接受度预测模型.使用大规模的真实数据,通过实验验证了这些指标的可预测性,分析了指标对开发过程的指导作用.当然,由于应用市场中存在的用户数据类型繁多,开发者需要解决的问题多样,同时实际可获得的数据集有限,本文给出的用户接受度指标模型尚不完备,选取的实例也未必能够充分代表开发者所遇到的实际问题.因此,本文提出的用户接受度指标模型仍需要在实践中进一步扩展与完善.

## References:

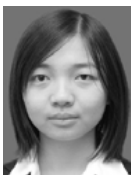
- [1] Jansen S, Bloemendal E. Defining app stores: The role of curated marketplaces in software ecosystems. In: Proc. of the Int'l Conf. of Software Business. Berlin, Heidelberg: Springer-Verlag, 2013. 195–206.
- [2] Basole RC, Karla J. On the evolution of mobile platform ecosystem structure and strategy. *Business & Information Systems Engineering*, 2011,3(5):313–322.
- [3] Martin W, Sarro F, Jia Y, Zhang Y, Harman M. A survey of app store analysis for software engineering. *IEEE Trans. on Software Engineering*, 2016,43(9):817–847.
- [4] AlSubaihain A, Sarro F, Black S, Capra L, Harman M. App store effects on software engineering practices. *IEEE Trans. on Software Engineering (Early Access)*, 2019. [doi: 10.1109/TSE.2019.2891715]
- [5] Sun Y, Guo B, Ouyang Y, Yu ZW, Wang Z. Strategies for mobile app evolution by using crowdsourced data. *Journal of Frontiers of Computer Science and Technology*, 2020,14(1):40–50 (in Chinese with English abstract).
- [6] Liu X, Li H, Lu X, Xie T, Mei Q, Feng F, Mei H. Understanding diverse smartphone usage patterns from large-scale appstore-service profiles. *IEEE Trans. on Software Engineering*, 2018,44(4):384–411.
- [7] Zhang D, Dang Y, Lou JG, Han S, Zhang H, Xie T. Software analytics as a learning case in practice: Approaches and experiences. In: Proc. of the Int'l Workshop on Machine Learning Technologies in Software Engineering. 2011. 55–58.
- [8] Zhang D, Han S, Dang Y, Lou JG, Zhang H, Xie T. Software analytics in practice. *IEEE Software*, 2013,30(5):30–37.
- [9] Zhang DM, Han S, Lou JG, Dang YN, Zhang HD, Xie T. Software analytics: Essence and practices. *Communications of the CCF*, 2014,10(3):29–36 (in Chinese with English abstract).
- [10] Shen GH, Huang ZQ, Xie B, Zhu YQ, Liao LL, Wang F, Liu YL. Survey on software trustworthiness evaluation: Standards, models and tools. *Ruan Jian Xue Bao/Journal of Software*, 2016,27(4):955–968 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5024.htm> [doi: 10.13328/j.cnki.jos.005024]
- [11] Hu TY, Jiang Y. Mining of user's comments reflecting usage feedback for APP software. *Ruan Jian Xue Bao/Journal of Software*, 2019,30(10):3168–3185 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5794.htm> [doi: 10.13328/j.cnki.jos.005794]
- [12] Lim SL, Bentley PJ, Kanakam N, Ishikawa F, Honiden S. Investigating country differences in mobile app user behavior and challenges for software engineering. *IEEE Trans. on Software Engineering*, 2015,41(1):40–64.
- [13] Hoon L, Vasa R, Schneider JG, Grundy J. An analysis of the mobile app review landscape: Trends and implications. Technical Report, Faculty of Information and Communication Technologies, Swinburne University of Technology, 2013.
- [14] Hoon L, Vasa R, Martino GY, Schneider JG, Mouzakis K. Awesome! Conveying satisfaction on the app store. In: Proc. of the Australian Computer-Human Interaction Conf.: Augmentation, Application, Innovation, Collaboration. 2013. 229–232.
- [15] Chen N, Lin J, Hoi SC, Xiao X, Zhang B. AR-miner: Mining informative reviews for developers from mobile app marketplace. In: Proc. of the 36th Int'l Conf. on Software Engineering. 2014. 767–778.
- [16] Xie Z, Zhu S. AppWatcher: Unveiling the underground market of trading mobile app reviews. In: Proc. of the 8th ACM Conf. on Security and Privacy in Wireless Mobile Networks. 2015. 1–11.
- [17] LiKamWa R, Liu Y, Lane ND, Zhong L. Moodscope: Building a mood sensor from smartphone usage patterns. In: Proc. of the 11th Annual Int'l Conf. on Mobile Systems, Applications, and Services. 2013. 389–402.
- [18] Rahmati A, Zhong L. Studying smartphone usage: Lessons from a four-month field study. *IEEE Trans. on Mobile Computing*, 2013, 12(7):1417–1427.

- [19] Falaki H, Mahajan R, Kandula S, Lymberopoulos D, Govindan R, Estrin D. Diversity in smartphone usage. In: Proc. of the 8th Int'l Conf. on Mobile Systems, Applications, and Services. 2010. 179–194.
- [20] Do TM, Gatica-Perez D. By their apps you shall understand them: Mining large-scale patterns of mobile phone usage. In: Proc. of the 9th Int'l Conf. on Mobile and Ubiquitous Multimedia. 2010. 1–10.
- [21] Girardello A, Michahelles F. AppAware: Which mobile applications are hot? In: Proc. of the 12th Conf. on Human-Computer Interaction with Mobile Devices and Services. 2010. 431–434.
- [22] McMillan D, Morrison A, Brown O, Hall M, Chalmers M. Further into the wild: Running worldwide trials of mobile systems. In: Proc. of the 8th Int'l Conf. on Pervasive Computing. 2010. 210–227.
- [23] Henze N, Poppinga B, Boll S. Experiments in the wild: Public evaluation of off-screen visualizations in the Android market. In: Proc. of the 6th Nordic Conf. on Human-Computer Interaction: Extending Boundaries. 2010. 675–678.
- [24] Sigg S, Lagerspetz E, Peltonen E, Nurmi P, Tarkoma S. Exploiting usage to predict instantaneous app popularity: Trend filters and retention rates. *ACM Trans. on the Web*, 2019,13(2):1–25.
- [25] Zuniga A, Flores H, Lagerspetz E, Nurmi P, Tarkoma S, Hui P, Manner J. Tortoise or hare? Quantifying the effects of performance on mobile app retention. In: Proc. of the World Wide Web Conf. 2019. 2517–2528.
- [26] Patro A, Rayanchu S, Griepentrog M, Ma Y, Banerjee S. Capturing mobile experience in the wild: A tale of two apps. In: Proc. of the 9th Conf. on Emerging Networking Experiments and Technologies. 2013. 199–210.
- [27] Xu Q, Erman J, Gerber A, Mao Z, Pang J, Venkataraman S. Identifying diverse usage behaviors of smartphone apps. In: Proc. of the ACM SIGCOMM Conf. on Internet Measurement. 2011. 329–344.
- [28] Zhong N, Michahelles F. Google play is not a long tail market: An empirical analysis of app adoption on the Google Play app market. In: Proc. of the 28th Annual ACM Symp. on Applied Computing. 2013. 499–504.
- [29] Garg R, Telang R. Inferring app demand from publicly available data. *MIS Quarterly*, 2013,37(4):1253–1264.
- [30] Petsas T, Papadogiannakis A, Polychronakis M, Markatos EP, Karagiannis T. Rise of the planet of the apps: A systematic study of the mobile app ecosystem. In: Proc. of the ACM SIGCOMM Conf. on Internet Measurement. 2013. 277–290.
- [31] Li H, Lu X, Liu X, Xie T, Bian K, Lin FX, Mei Q, Feng F. Characterizing smartphone usage patterns from millions of Android users. In: Proc. of the ACM SIGCOMM Conf. on Internet Measurement. 2015. 459–472.
- [32] Böhmer M, Hecht B, Schöning J, Krüger A, Bauer G. Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. In: Proc. of the Int'l Conf. on Human-computer Interaction with Mobile Devices and Services. 2011. 47–56.
- [33] Rahmati A, Tossell C, Shepard C, Kortum P, Zhong L. Exploring iPhone usage: The influence of socioeconomic differences on smartphone adoption, usage and usability. In: Proc. of the 14th Int'l Conf. on Human-computer Interaction with Mobile Devices and Services. 2012. 11–20.
- [34] Zhao S, Ramos J, Tao J, Jiang Z, Li S, Wu Z, Pan G, Dey AK. Discovering different kinds of smartphone users through their application usage behaviors. In: Proc. of the 2016 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing. 2016. 498–509.
- [35] Martin W, Sarro F, Harman M. Causal impact analysis for app releases in Google Play. In: Proc. of the 24th ACM SIGSOFT Int'l Symp. on Foundations of Software Engineering. 2016. 435–446.
- [36] Khalid H, Nagappan M, Shihab E, Hassan AE. Prioritizing the devices to test your app on: A case study of Android game apps. In: Proc. of the SIGSOFT Int'l Symp. on the Foundations of Software Engineering. 2014. 610–620.
- [37] Li H, Ai W, Liu X, Tang J, Huang G, Feng F, Mei Q. Voting with their feet: Inferring user preferences from app management activities. In: Proc. of the 25th Int'l Conf. on World Wide Web. 2016. 1351–1362.
- [38] Newman ME. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 2005,46(5):323–351.
- [39] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence, 1995. 1137–1143.
- [40] Zhang YX, Guo B, Ouyang Y, Wang Z, Yu ZW. Evolutionary pattern analysis and prediction of mobile app. *Journal of Frontiers of Computer Science and Technology*, 2019,13(12):1981–1994 (in Chinese with English abstract).
- [41] Wang HY, Guo Y, Ma ZA, Chen XQ. Automated detection and classification of third-party libraries in large scale Android apps. *Ruan Jian Xue Bao/Journal of Software*, 2017,28(6):1373–1388 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5221.htm> [doi: 10.13328/j.cnki.jos.005221]

- [42] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, 1996,58(1):267–288.
- [43] Hoerl AE, Kennard RW. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 1970,12(1):55–67.
- [44] Ho TK. Random decision forests. In: *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition, Vol.1*. 1995. 278–282.
- [45] Kendall MG. A new measure of rank correlation. *Biometrika*, 1938,30(1/2):81–93.
- [46] Sugiura N, Ogden RT. Testing change-points with linear trend. *Communications in Statistics-Simulation and Computation*, 1994, 23(2):287–322.
- [47] Ge X, Smyth P. Segmental semi-Markov models for endpoint detection in plasma etching. *IEEE Trans. on Semiconductor Engineering*, 2001,259:201–209.
- [48] Chua L, Deng AC. Canonical piecewise-linear modeling. *IEEE Trans. on Circuits and Systems*, 1986,33(5):511–525.
- [49] Stigler SM. Francis Galton's account of the invention of correlation. *Statistical Science*, 1989,4(2):73–79.
- [50] Ruiz EJ, Hristidis V, Castillo C, Gionis A, Jaimes A. Correlating financial time series with micro-blogging activity. In: *Proc. of the 5th ACM Int'l Conf. on Web Search and Data Mining*. 2012. 513–522.
- [51] McIlroy S, Ali N, Hassan AE. Fresh apps: An empirical study of frequently-updated mobile apps in the Google Play store. *Empirical Software Engineering*, 2016,21(3):1346–1370.
- [52] Möller A, Michahelles F, Diewald S, Roalter L, Kranz M. Update behavior in app markets and security implications: A case study in Google Play. In: *Proc. of the 3rd Int'l Workshop on Research in the Large*. 2012. 3–6.
- [53] Manning CD, Raghavan P, Schütze H. Scoring, term weighting and the vector space model. In: *Introduction to Information Retrieval*. 2008. 109–133.
- [54] Martin W, Harman M, Jia Y, Sarro F, Zhang Y. The app sampling problem for app store mining. In: *Proc. of the IEEE/ACM 12th Working Conf. on Mining Software Repositories*. 2015. 123–133.
- [55] Rennie JD, Shih L, Teevan J, Karger DR. Tackling the poor assumptions of naive bayes text classifiers. In: *Proc. of the 20th Int'l Conf. on Machine Learning*. 2003. 616–623.

#### 附中文参考文献:

- [5] 孙悦,郭斌,欧阳逸,於志文,王柱.基于群智数据的移动 APP 演化策略研究. *计算机科学与探索*,2020,14(1):40–50.
- [9] 张冬梅,韩石,楼建光,党映农,张海东,谢涛.软件解析学——要点与实践. *中国计算机学会通讯*,2014,10(3):29–36.
- [10] 沈国华,黄志球,谢冰,朱羿全,廖莉莉,王飞,刘银陵.软件可信评估研究综述:标准、模型与工具. *软件学报*,2016,27(4):955–968. <http://www.jos.org.cn/1000-9825/5024.htm> [doi: 10.13328/j.cnki.jos.005024]
- [11] 胡甜媛,姜瑛.体现使用反馈的 APP 软件用户评论挖掘. *软件学报*,2019,30(10):3168–3185. <http://www.jos.org.cn/1000-9825/5794.htm> [doi: 10.13328/j.cnki.jos.005794]
- [40] 张艺璇,郭斌,欧阳逸,王柱,於志文.移动 APP 演化模式分析与预测. *计算机科学与探索*,2019,13(12):1981–1994.
- [41] 王浩宇,郭耀,马子昂,陈向群.大规模移动应用第三方库自动检测和分类方法. *软件学报*,2017,28(6):1373–1388. <http://www.jos.org.cn/1000-9825/5221.htm> [doi: 10.13328/j.cnki.jos.005221]



陆璇(1991—),女,博士,主要研究领域为软件解析学,数据挖掘.



陈震鹏(1994—),男,博士生,CCF 学生会会员,主要研究领域为软件解析学,数据挖掘.



刘讓哲(1980—),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,系统软件.



梅宏(1963—),男,博士,教授,博士生导师,CCF 会士,主要研究领域为软件工程,系统软件.