

编号_____

西安科技大学

毕业设计（论文）

（ 2022 届）

题 目	App 用户使用行为模式和量化分析研究
学生姓名	刘丽君
学 号	18408010203
专业班级	软件工程 1802
指导教师	刘晓建
所在学院	计算机科学与技术学院
日 期	2022 年 5 月 29 日

西安科技大学

学位论文诚信声明书

本人郑重声明：所呈交的学位论文（设计）是我个人在导师指导下进行的研究（设计）工作及取得的研究（设计）成果。除了文中加以标注和致谢的地方外，论文（设计）中不包含其他人或集体已经公开发表或撰写过的研究（设计）成果，也不包含本人或其他人在其它单位已申请学位或为其他用途使用过的成果。与我一同工作的同志对本研究（设计）所做的任何贡献均已在论文中做了明确的说明并表示了致谢。

申请学位论文（设计）与资料若有不实之处，本人愿承担一切相关责任。

学位论文（设计）作者签名：刘丽君 日期： 2022 年 6 月 1 日

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：在校期间所做论文（设计）工作的知识产权属西安科技大学所有。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文（设计）被查阅和借阅；学校可以公布本学位论文（设计）的全部或部分内容并将有关内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存和汇编本学位论文。

保密论文待解密后适用本声明。

学位论文（设计）作者签名：刘丽君 指导教师签名：刘晓建
2022 年 6 月 1 日

分 类 号

学校代码 10704

密 级

学 号 18408010203

西安科技大学

学 士 学 位 论 文

题目：App 用户使用模式和量化分析研究

作者：刘丽君

指导教师：刘晓建

学科专业：软件工程

专业技术职称：副教授

申请学位日期：2022 年 6 月

摘 要

随着 5G 时代的到来，智能手机已经成为人们生活不可或缺的一部分，目前智能手机端主要搭载的操作系统是 Android 系统，在这系统下的应用种类也越来越多。基于此，本文主要研究 Android App 的用户使用行为模式，通过挖掘用户使用 App 的行为数据，并进行量化分析，对软件测试和软件产品新功能开发具有重要意义。

本文的主要工作包括如下三个方面：

1. 研究了基于录制重放的用户使用 App 的轨迹获取方法。分别尝试了使用 Android 自动化测试工具 MonkeyRunner 和 UIAutomator，编写脚本获取用户操作 App 的行为轨迹。

2. 研究了基于自埋点的用户使用 App 行为数据获取方法。设计并实现了基于装饰器模式和面向切面编程思想（Aspect-oriented）的自埋点算法，并且研究了如何使用第三方 SDK 进行代码埋点。

3. 研究了用户使用 App 的行为模式的量化分析方法。建立了四个用户行为模型：界面访问路径模型、界面热力图模型、行为事件分析模型、漏斗模型对用户行为模式进行量化。

为了验证上述方法，以淘宝 App 和自开发地图导航 App 两个应用为例，获取其用户行为数据后，对其用户行为模式进行量化分析并可视化，验证了上述方法的可行性。以上研究方法和工作为测试人员和产品设计人员提供了指导性建议。

关键词：Android App 测试；用户行为模式；自埋点；SDK；量化分析

ABSTRACT

With the advent of the 5G era, smart phones have become an indispensable part of people's lives, and the main operating system on the current smartphone side is the Android system, and there are more and more types of applications under this system. Based on this, this paper mainly studies the user behavior pattern of Android App, and is of great significance for software testing and software product new feature development by mining the behavior data of users using App and conducting quantitative analysis.

The main work of testing includes the following three aspects:

1. The track acquisition method based on the user's use of the App based on the recording and playback. I tried to use the Android automated testing tools MonkeyRunner and UIAutomator respectively to automatically obtain the script of the entire process of the user operating the App.

2. The method of acquiring behavior data of users using App based on buried points. Design and implement a self-embedding algorithm based on the decorator pattern and aspect-oriented programming, and study how to use the third-party SDK to embed code.

3. Quantitative analysis method of the behavior pattern of users using the App. Four user behavior models are established: interface access path model, interface heat map model, behavioral event analysis model, and funnel model to quantify user behavior patterns.

In order to verify the above methods, taking Taobao App and self-developed map navigation App as examples, after obtaining the user behavior data, the user behavior patterns are quantitatively analyzed and visualized, which verifies the feasibility of the above methods. The above research methods and work provide guiding recommendations for testers and product designers.

Key Words: Android app test; User behavior pattern; Burial site by oneself; SDK; Quantitative analysis

目 录

1 绪论.....	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	1
1.2 国内外研究现状	2
1.3 本文主要研究内容	3
1.4 本论文组织结构	3
2 相关理论和技术	4
2.1 用户行为模型和分析方法	4
2.2 安卓自动化测试工具	4
2.3 代码埋点	5
2.4 第三方 SDK	6
2.5 本章小结	7
3 用户 App 使用行为获取	8
3.1 基于录制重放的用户使用 App 的轨迹获取	8
3.1.1 MonkeyRunner 安装和使用	9
3.1.2 UIAutomator2 安装和使用	11
3.2 使用 Android SDK 埋点	13
3.2.1 Maven 自动集成友盟+SDK	13
3.2.2 集成校验测试	14
3.3 自埋点的设计和实现	14
3.3.1 总体设计	14
3.3.2 具体实现	18
3.3.3 结果分析	20
3.4 本章小结	21
4 用户行为模式量化分析	22

4.1 分析处理公开数据集	22
4.1.1 数据收集	22
4.1.2 数据清洗	22
4.2 用户行为模型的建立	26
4.2.1 界面访问路径模型	26
4.2.2 界面热力图模型	26
4.2.3 行为事件分析模型	27
4.2.4 漏斗模型	28
4.3 本章小结	28
5 实验设计与结果展示	29
5.1 电商类 App 用户使用行为模型	29
5.1.1.分析用户访问情况	29
5.1.2.构建漏斗模型	29
5.1.3.用户行为习惯分析	30
5.2 地图导航类 App 用户使用行为模型	34
5.2.1 获取用户行为轨迹	35
5.2.2 分析用户行为数据建立模型	36
5.3 本章小结	38
6 结论与展望	39
6.1 本文工作总结	39
6.2 未来展望	39
致谢	40
参考文献	41

1 绪 论

1.1 研究背景及意义

1.1.1 研究背景

随着 5G 时代的到来，网络蓬勃发展，移动平台已经成为个人计算和网络未来的主要发展趋势，智能手机的普及，方便了人们生活的方方面面。为了满足广大消费者的需要，移动应用的数量日益增多，应用复杂度也提高。目前智能手机使用的操作系统主要有两个：IOS 和 Android。据相关资料表明^[1-3]，Android 系统已经是移动市场的主流，所以本文致力于研究 Android App 的用户使用行为模式。

随着开发技术不断提升，一些自动化测试方法应运而生。但是这些思路方法并没有在实际工作中广泛使用，因为它们更关注的是破坏性测试，致力于使代码的覆盖率最大化或者试图发现尽可能多的系统崩溃，而且生成的测试用例与应用功能有差异，无法有效对安卓应用进行全面的功能性测试。

然而，功能性测试较为复杂，工作量较大。为了解决这个问题，用户轨迹的概念被提出。在特定情况下用户与应用程序进行交互，交互产生的数据称为用户轨迹。基于这些用户轨迹，提取应用程序的用户行为模式有助于了解 Android 应用程序的功能，辅助测试，但其工作量大，测试周期长，而且用户轨迹并没有对它们的内部关系进行深入分析。为此，一些学者提出了用户行为模型的概念，这就解决了用户轨迹不能描述各用户行为之间关系的问题，并且它可以提取用户共同的行为特征。

1.1.2 研究意义

基于用户角度，随着用户在智能手机上下载安装越来越多的 App，用户查找自己当前需要用到的 App 时，需要花费更多的时间。应用的数目增加会占据更多的系统内存，这就导致手机 App 打开耗费时间。因为对于一些手机自带应用程序，执行延迟约为几秒钟，但对于一些复杂应用程序，执行延迟通常超过 10 秒。因此，越来越多的研究致力于如何提高应用程序搜索和启动的效率，将应用程序缓存在内存中是减少应用程序运行时间的直观解决方案。然而，移动设备的内存有限。要想在内存中缓存大量大型应用程序是很困难的。而当我们能够挖掘到用户行为模式，从软件性能角度提高用户体验感，就可以缓解该类问题的出现。因此，如何对 App 用户使用行为模式进行量化分析是一个值得研究的课题。App 用户使用行为模式的量化对提高软件测试的效率和有效性，优化程序提高用户体验，以及帮助理解应用程序的功能具有十分重要的研究价值和意义。

基于产品的角度，建立用户行为模型，能够帮助产品设计师更准确地判断用户行为路径的转变，做好产品的修改，了解新功能对原有产品的影响，帮助运营人员设计准确的营销策略，有效评估营销效果等。

1.2 国内外研究现状

对于用户轨迹获取，有基于插桩埋点的用户轨迹获取和基于记录/重放的用户轨迹获取两种方法。埋点作为一种最简单直接的获取用户行为轨迹的方式，包括代码埋点、全埋点、可视化埋点。传统的代码埋点效率较低，并且和源代码耦合性高。而且，App 界面元素种类较多，用户操作类型各种各样，埋点成本高，维护起来也需要消耗人力。金小俊^[4]等人提出了一种基于数据和状态的可视化埋点方法，以页面为单位进行埋点配置，页面以类名作为唯一标识，埋点的发生可由数据和状态的变化来驱动。并将该方法应用于实际 App 后验证其埋点方法的高效性。蔡榆榕^[5]运用埋点技术，对高校的应用服务平台的用户行为进行跟踪和捕获，得到单位变量 PV（访问量）和 UV（独立访客），并实现了用户活跃度分析。吴荣灿^[6]等人基于大数据分析技术，设计并实现了用户行为分析平台。该平台采用无埋点自定义的检测方案用于采集用户行为数据，并通过对数据进行统计分析，为产品设计人员提供方案。基于记录/重放，有学者^{[7][8]}使用 Android 自动化测试工具，通过其提供的 UI 元素查看器和录制重放功能，编写脚本，对用户使用 App 行为进行获取。

对于用户行为模式挖掘，主要有三类：单应用的用户行为模式挖掘、移动用户轨迹模式挖掘、Web 浏览模式挖掘。毛群^[9]提出了一种跨 Android 应用程序挖掘和重用用户行为模式的方法，对于被测试的安卓应用，使用特定的序列模式挖掘技术，提取用户行为模式。在此基础上，他们还提出了两个策略：基于语义的模糊事件相互匹配策略以及基于被测安卓应用 GUI 模型的最优路径选择策略，实现了相似安卓应用之间一种用户行为模式的重用。郭景怡^[10]对 Mach-O 二进制分析与功能场景特征挖掘展开研究，提出一种 iOS 应用场景 API 序列模式挖掘方法。该方法使用符号执行技术从 Mach-O 二进制中提取出 iOS 应用的执行路径，从路径中提取出场景实例集合并对其进行模式挖掘，最终获得场景的 API 序列模式。Wu Tingting^[11]针对电子商务领域的后台日志，构建了一个基于电子商务用户行为数据分析的用户行为挖掘模型。基于网络用户行为的理论，分析了基于互动内容的用户行为。在大数据背景下，对传统的数据挖掘算法进行了进一步的优化，大大提高了算法的效率。

1.3 本文主要研究内容

本文主要研究 Android App 的用户使用行为模式，通过挖掘用户使用 App 的行为数据，建立用户行为模型并进行可视化。主要包括三个方面：第一，利用录制重放思想，使用 Android 自动化测试工具 MonkeyRunner 和 UIAutomator 获取用户使用 App 的行为轨迹。第二，基于装饰器模式和面向切面编程思想，设计实现了自埋点算法，从而对自开发 App 用户行为数据进行获取。第三，本文总结了四个用户行为模型，并以淘宝 App 和自开发的地图导航类 App 为例，获取其用户行为数据，建立用户行为模型，并进行量化分析。

1.4 本论文组织结构

第一章：绪论。本章介绍了 App 用户行为模式的研究背景以及对用户行为模式量化分析的研究意义，并分析了国内外对于用户 App 使用行为获取和量化分析的研究现状。

第二章：相关理论和技术。本章介绍本文用到的理论和技术，包括用户行为模型和分析方法，安卓自动化测试工具，代码埋点，第三方 SDK。

第三章：用户 App 使用行为获取。本章介绍了两个用户 App 使用行为获取的方法：使用安卓自动化测试工具的录制重放功能获取和基于装饰器模式和面向切面编程思想设计实现自埋点算法。

第四章：用户行为模式量化分析。本章使用 SQL 语句对已有公开数据集进行用户行为模式量化，建立用户行为模型并进行可视化。

第五章：实验设计与结果展示。本章对电商购物类和地图导航类两类 App 进行用户行为模式分析和量化。

第六章：结论与展望。总结了本文的主要研究成果和工作内容，并对之后的工作进行计划展望。

2 相关理论和技术

2.1 用户行为模型和分析方法

用户行为是指用户在某个时间，某个地点与某人员进行了某些交互。用户行为分析可以更好地了解某类用户的使用 App 的行为习惯，发现产品后续发展中存在的问题。在日常用户行为分析中，有六种常用的分析方法见下表 2-1。

表 2-1 用户行为分析方法

分析方法	主要功能	表现形式
用户行为事件分析	研究某行为事件的发生对产品的影响	可以通过埋点实现，并用数据结构和图表表示
页面点击分析	显示页面区域元素点击密度	可以用热力图形式呈现
用户行为路径分析	展示用户访问路径和页面滑动轨迹	可以通过图示（流程图或知识图谱展示）
用户健康度分析	体现产品运营情况	通过 E-R 图展示
漏斗模型分析	体现每个环节的转化率或者用户流量	通过转化漏斗图展示
用户画像分析	根据用户属性抽象得到用户标签	通过脑图或者 Profile 软件

2.2 安卓自动化测试工具

表 2-2 移动端自动化测试工具的功能

工具名称	功能描述
Monkey	随机测试、压力测试，生成伪随机的用户事件流
MonkeyRunner	可以发送特定命令和事件，从而用于控制设备
UIAutomator	可以用于 Android、iOS 应用的 UI 测试、单元测试、监控事件
Appium	可以用于测试移动端和 web 应用

表 2-3 移动端自动化测试工具的区别

	MonkeyRunner	UIAutomator	Appium
Android 系统	支持	支持	支持
iOS 系统	不支持	不支持	支持
Web 应用	不支持	不支持	支持
使用编程语言	Python	Java, Python	支持多种
跨应用测试	不支持	支持	不支持

Monkey 是最早较多使用的自动化测试工具。它最大的特点就是可以进行随机测试。它不仅能在 Android 设备上运行也能在模拟器上运行，在运行过程中可以生成一系列随机事件和用户事件。然而，Monkey 生成的测试事件流是随机的，事件流的序列也相对

冗余，测试效率较低。

而 MonkeyRunner 工具提供了一个 API，编写相关代码后，系统可以通过该 API，发送一系列特定命令和事件，以便控制模拟器或 Android 设备。该 API 主要包括三个模块：MonkeyRunner、MonkeyDevice、MonkeyImage，对每一个模块具体的功能介绍见下表 2-4：

表 2-4 MonkeyRunne API 模块	
模块名	功能
MonkeyRunner	用于连接 MonkeyRunner 和设备或模拟器
MonkeyDevice	用于安装和卸载包、可以发送按键和触摸事件、运行测试包
MonkeyImage	用于屏幕截图

UIAutomator 是一种 Android App 界面的自动化测试工具，能够进行点击、长按、滑动、查找等一系列操作。UIAutomator 的具体工作原理如图 2-1 所示。

- Python-uiautomator2 主要分为两个部分：Python 客户端和移动设备。
- Python 端：负责运行 Python 测试脚本，并且向移动设备发送对应的 HTTP 请求。
 - 移动设备：负责运行已经封装好的 UIAutomator2 的 HTTP 服务，解析发来的请求，并且转化为对应的代码。

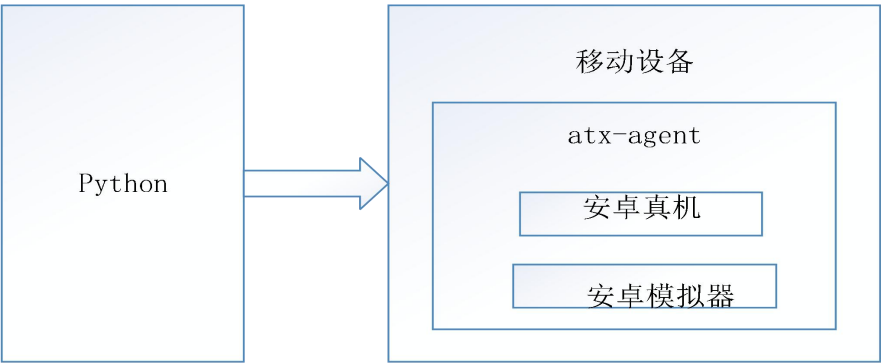


图 2-1 UIAutomator 工作原理

2.3 代码埋点

代码埋点也叫自定义埋点，一般代码埋点使用两种方法，一种是通过调用第三方 SDK 进行植入。具体过程为：初始化 SDK 之后，将 SDK 植入 App 源代码中，当某个事件发生后，系统会调用相关接口，将触发的事件保存在一种特定数据模型中，之后将数据模型转化成 JSON 串，存储在本地数据库中。然后，按照特定的发送策略将数据发送到指定的服务端^[12]。另一种方法是自埋点，一般自埋点采用 AOP（面向切面编程）思想，具体过程为：在需要进行埋点的事件处添加代码，在切面处进行埋点后，用 log

日志输出，上传到服务器，用本地库存储。

自埋点采用的 AOP 思想是面向对象编程的补充。它利用一种“横切”的技术，代表一种横向技术。AOP 技术将与业务无关但是各个业务模块共同调用的逻辑封装起来，减少系统中重复的代码，实现高耦合低内聚。下图 2-2 为 AOP 实现的原理。

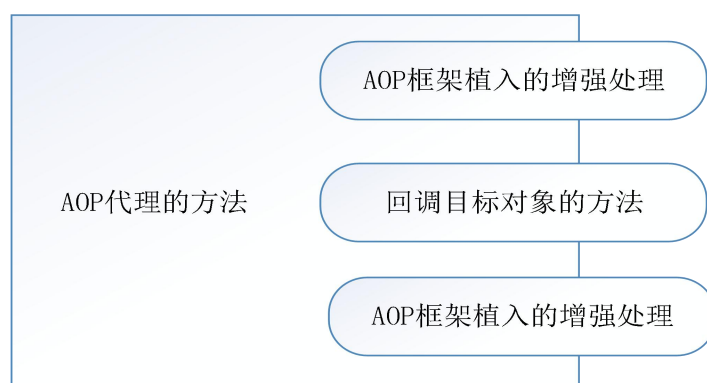


图 2-2 AOP 实现原理图

AOP 代理是由 AOP 框架动态生成的一个对象，该对象可作为目标对象使用。

2.4 第三方 SDK

软件开发工具包（SDK）是由第三方服务提供商提供的工具包，用于实现软件产品的某些功能。目前市场上有很多分析应用用户行为的第三方工具，这些工具一般为开发者提供两种 Android SDK 集成方式：自动集成和手动集成^[13]。自动集成是指 Android 的 maven 依赖，手动集成是指在下载页下载 SDK 后导入 App 工程。

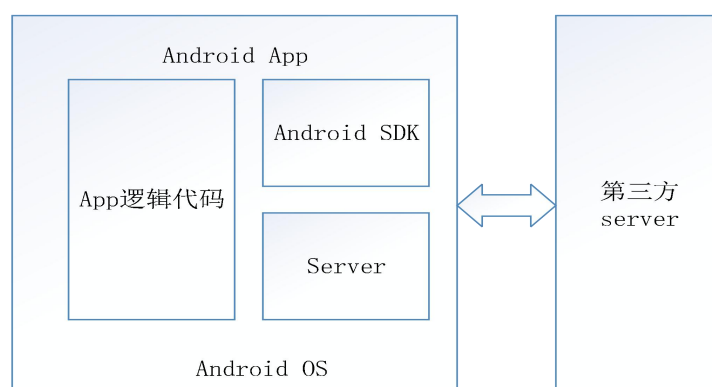


图 2-3 第三方 SDK 在 Android 系统的运行机制

本文介绍的第三方 SDK 植入，采用自动集成方法。在源代码中进行集成之后，将

用户使用 App 行为数据传入服务器，之后第三方服务对其进行可视化数据分析建模。

2.5 本章小结

在本章节，主要对于系统中使用到的技术进行了介绍说明，包含用户行为模型和分析方法、移动端自动化测试工具使用方法、代码埋点技术、第三方 SDK 等。除此之外，对这些方法技术在本课题中的功能和选择原因进行了阐述和介绍。

3 用户 App 使用行为获取

3.1 基于录制重放的用户使用 App 的轨迹获取

为了获取 App 的用户使用行为轨迹，本文借助 MonkeyRunner 和 UIautomator 这两个测试工具。整体思路如下图：

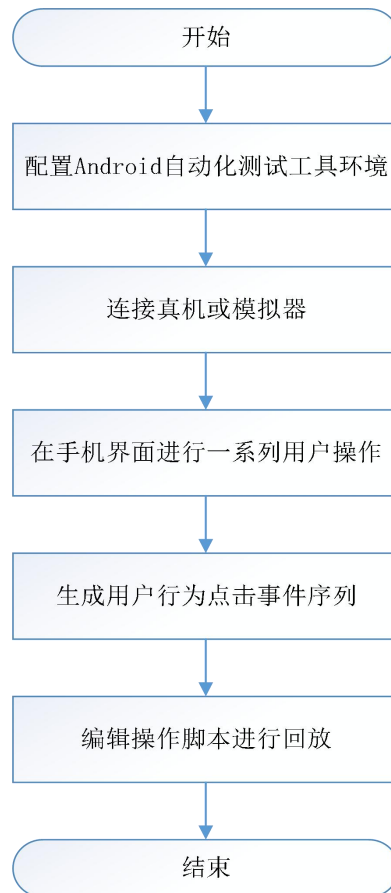


图 3-1 Android 自动化测试工具获取用户行为轨迹流程图

首先需要配置好对应环境，连接真机或者模拟器。真机连接可以通过 USB 或者 WIFI 连接。其次，当用户在手机界面进行一系列操作后，会被工具记录在脚本中。最后，编辑自动化测试工具生成的脚本，运行后可以进行重放。

实施该方法所需要的硬件环境是：Windows10 系统，软件环境是：Python 编译器、JDK、Android SDK 环境。

下面介绍环境搭建流程：

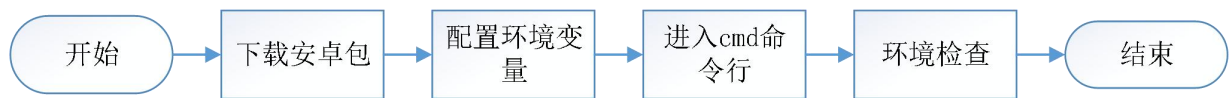


图 3-2 Android 自动化测试工具环境搭建流程

在官网下载需要的安装包，进行环境配置后，进入 cmd 命令行中，输入对应的检查命令，验证环境配置是否完成。

3.1.1 MonkeyRunner 安装和使用

1.MonkeyRunner 环境搭建

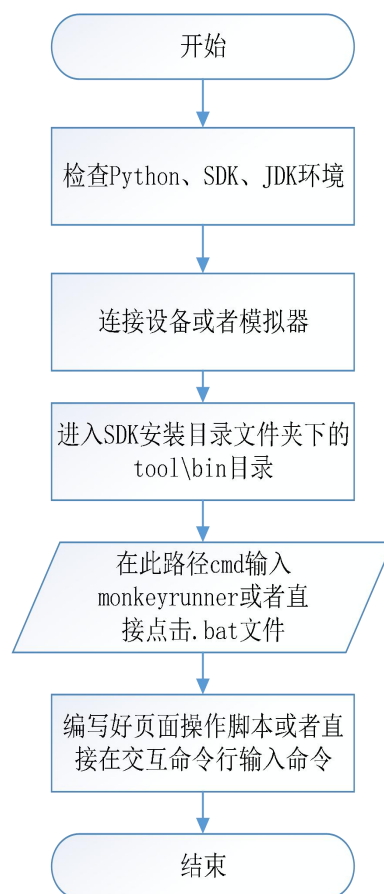


图 3-3 MonkeyRunner 环境搭建流程图

首先检查 Python、SDK、JDK 环境是否配置好，连接设备或者模拟器，如果是手机设备，需要打开手机开发者模式，模拟器可以使用夜神模拟器或雷电模拟器，本文使用夜神模拟器 V.7.0.2.5 版本。

其次，进入到 Android SDK 的安装目录，在 tool\bin 目录进入 cmd 命令行，输入

MonkeyRunner 或者点击 monkeyrunner.bat 文件，出现如下界面则搭建成功。

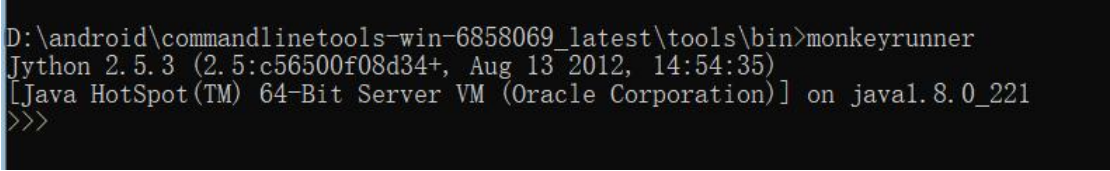


图 3-4 SDK 环境搭建成功截图

接下来进入就可以输入 MonkeyRunner 命令。

2.具体实验操作

本文主要使用 MonkeyRunner 的屏幕录制功能，下面对该功能使用方法做一介绍。

(1) 下面是 monkey_recorder.py 的代码

```
1.  # -*- coding: UTF-8 -*-
2.  From com.android.monkeyrunner import MonkeyRunner as mr
3.  From com.android.monkeyrunner  import MonkeyDevice as md
4.  #from com.android.monkeyrunner import MonkeyImage as mi
5.  from com.android.monkeyrunner.recorder import MonkeyRecorder as recorder
6.  device=mr.waitForConnection()
7.  recorder.start(device)
```

本文将此代码保存在路径：D:\android\commandlinetools-win-6858069_latest\tools\bin\monkey_recorder.py。

(2) cmd 命令行中输入命令：monkeyrunner D:\android\commandlinetools-win-6858069_latest\tools\bin\monkey_recorder.py

Enter 回车后会看到工具界面：



图 3-5 MonkeyRunner 运行界面图

该窗口可以自动显示手机当前界面，并且实时刷新，每一个按钮的功能如下：

- wait: 设置等待时间，即就是下一个操作的响应等待时间，单位是 seconds 秒。

- Press a Button: 确定需要点击的按钮，包括菜单、主键、查找等。
- Type Something: 输入内容到输入框。
- Fling: 可以向上、下、左、右拖动并且可以设置操作的位置范围。
- Export Actions: 用来导出脚本。
- Refresh Display: 当界面卡顿时，可以刷新手机界面。

(3)当在这个工具上点击时，真机也会跟着执行，同时会显示我们每一步点击的位置的坐标。

点击“Export Actions”导出脚本文件，打开该文件可以看到如下脚本,该脚本中记录了点击事件的点击位置，但是每一次点击完需要设置等待时间，以便软件响应，这也是该自动化测试工具的不便之处。

```
1. TOUCH|{'x':400,'y':100,'type':'downAndUp',}
2. WAIT|{'seconds':1.0,}
3. TOUCH|{'x':509,'y':2317,'type':'downAndUp',}
4. WAIT|{'seconds':1.0,}
5. TOUCH|{'x':390,'y':1290,'type':'downAndUp',}
6. WAIT|{'seconds':1.0,}
7. TOUCH|{'x':850,'y':2317,'type':'downAndUp',}
8. TYPE|{'message':'HELLO WORLD!',}
9. PRESS|{'name':'HOME','type':'downAndUp',}
10. WAIT|{'seconds':1.0,}
11. TOUCH|{'x':661,'y':897,'type':'downAndUp',}
12. WAIT|{'seconds':1.0,}
13. TOUCH|{'x':661,'y':933,'type':'downAndUp',}
```

3.1.2 UIAutomator2 安装和使用

本文使用 UIAutomator2，当连接好移动手机设备后，在联网条件下，系统会自动安装 atx-agent 这个 App，它会及时启动 UIAutomator2 服务。

1.UIAutomator2 环境搭建

(1) 安装 UIAutomator2

使用 pip 安装：进入 Python 安装目录下，cmd 命令行中输入 `pip install -u uiautomator2`。

(2) 安装 Weditor

Weditor 可以在浏览器界面中显示用户界面元素的具体定位。由于 UIAutomator 是资源独占的，而且 ATX 不能与 UIAutomator Viewer 同时使用，为了避免此类问题，本文使用 Weditor，用以减少频繁开关 ATX 的需要。

使用 pip 安装：`pip install -u weditor`，若无法安装成功，可以在 Pycharm 配置中安装

或安装 Weditor 指定的低版本。

2. 具体实验操作

(1) 本文使用 Python+UIAutomator2 完成自动化测试，使用 Pycharm 编写 Python 代码，版本为 PyCharm Community Edition 2019.3.1。在使用之前需要在 Pycharm 中配置 UIAutomator2，下载 package。

File->setting->project->project interpreter->点击右上方+>下载 UIAutomator2
之后连接夜神模拟器，在 Pycharm 中编写代码进行测试。

(2) 运行 Weditor

打开夜神模拟器，在 Python 安装目录下，进入 cmd 命令行界面，输入 `python -m weditor` 启动 Weditor，出现图 3-6，则表示正在启动。

```
C:\Windows\System32\cmd.exe - python -m weditor
Microsoft Windows [版本 10.0.19044.1645]
(c) Microsoft Corporation。保留所有权利。

D:\Python37>python -m weditor
listening on http://192.168.105.21:17310
```

图 3-6 cmd 命令行启动 Weditor 截图

随后进入浏览器界面，在文本框内输入 127.0.0.1:62001 连接模拟器，得到如下图 3-7:

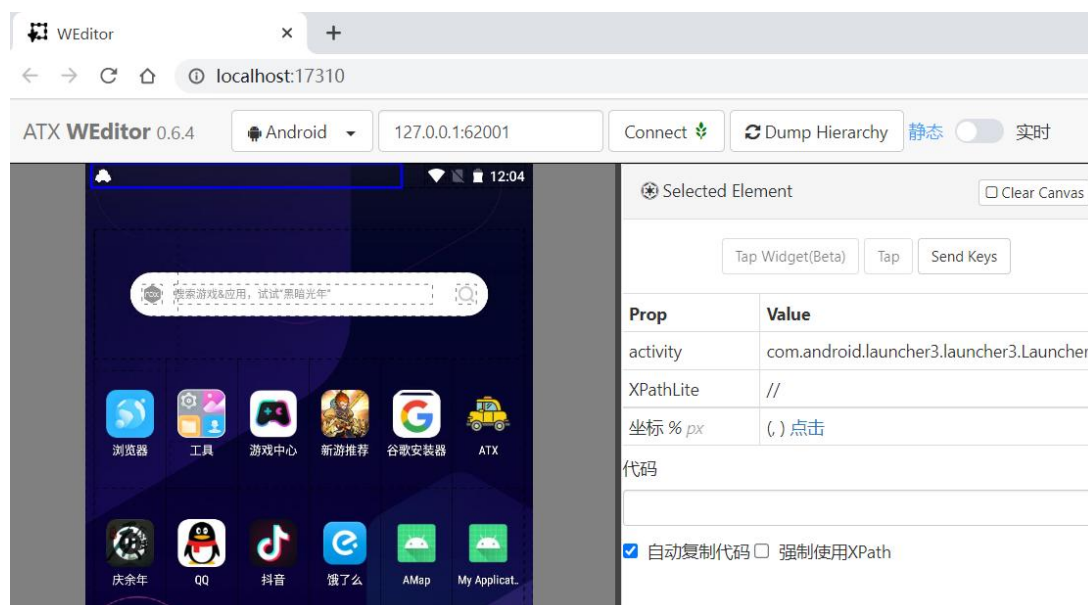


图 3-7 Weditor 操作界面

之后根据 Weditor 的提示代码编写 Python 脚本，进行自动化测试。

3.2 使用 Android SDK 埋点

3.2.1 Maven 自动集成友盟+SDK

本文选用第三方工具友盟，友盟+是国内第三方全域数据平台，为企业的开发人员、产品经理、运营人员等服务。本文使用友盟+SDK，选择移动统计 SDK，并借助其辅助工具，排查集成问题，验证集成结果。

为了方便起见，本文选择 Maven 进行自动集成，在安卓项目的 build.gradle 配置脚本中选择添加 Maven 在线依赖项，并导入 SDK。

下面介绍 Maven 依赖配置方法步骤：

1. 添加仓库地址

在 APP 的项目目录下找到工程文件 build.gradle，在配置脚本中找到文件 buildscript 和 allprojects，在其中添加 Maven 仓库地址。

2. 添加依赖

在工程下对应文件中的 build.gradle 配置脚本下找到 dependencies 字段，添加统计 SDK 库依赖。

```
1. dependencies {
2.     implementation fileTree(include:['*.jar'], dir:'libs')
3.     implementation 'com.umeng.umsdk:common:9.4.7'
4.     implementation 'com.umeng.umsdk:asms:1.4.1'// 必选
5.     implementation 'com.umeng.umsdk:apm:1.5.2'
6. }
```

3.集成测试配置

在 manifest.xml 文件中进行集成测试配置，需要在其中找到响应 android.intent.action.main 的 activity（也就是应用可以被外部唤起的第一个 Activity），加上如下代码，让应用能被指定连接 scheme 唤起。

在《我的产品》中找到自己的 appkey

替换如下

```
1. <intent-filter>
2.     <action android:name="android.intent.action.VIEW" />
3.     <category android:name="android.intent.category.DEFAULT" />
4.     <category android:name="android.intent.category.BROWSABLE" />
5.     <data android:scheme="um.该 app 对应的 appkey" />
6. </intent-filter>
```

4.代码初始化

在项目的 mainactivity.java 类中，添加如下代码：

```
UMConfigure.setLogEnabled(true);
UMConfigure.init(this, "59892f08310c9307b60023d0", "Umeng",
UMConfigure.DEVICE_TYPE_PHONE, "");
```

3.2.2 集成校验测试

打开 App 后页面显示如下图 3-8 所示，即为集成成功。

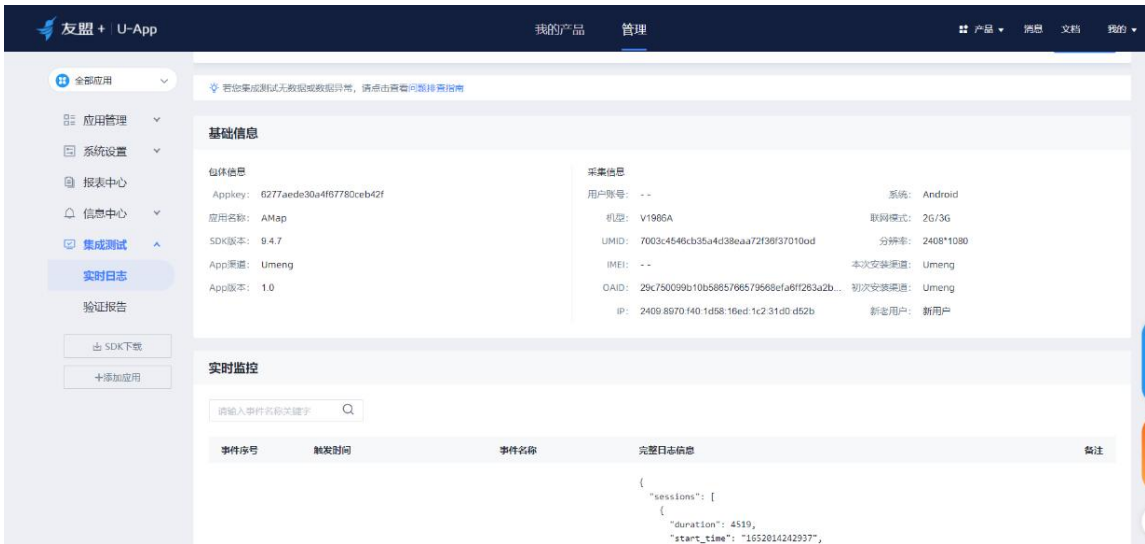


图 3-8 SDK 集成成功界面图

3.3 自埋点的设计和实现

本章以只有登录注册界面的简单 Demo App 为例介绍自埋点的设计实现。

3.3.1 总体设计

1.代码埋点组织结构

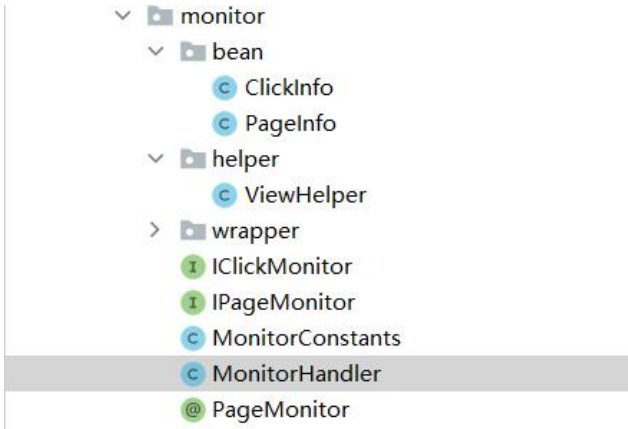


图 3-9 自埋点组织结构图

Monitor 包下面是自埋点的算法代码，wrapper 包下为装饰类。

2.设计模式

本文主要使用装饰器模式^[14]这一设计模式思想，借助面向 AOP（面向切面编程）的思想。

装饰器模式在代码埋点中的具体应用：

A 类中的监听事件方法替换成 B 类中的埋点逻辑 + A 类中的监听事件方法。思路如下：

A implements Interface (接口方法 doSth)

B implements Interface{ B(A a){} doSth{a.doSth() + 埋点逻辑}}

然后 View 获取已设置好的监听事件，通过反射获取。借用装饰器设置到 B 类的监听事件方法中，并将其重新设置给目标控件。

3. 类图

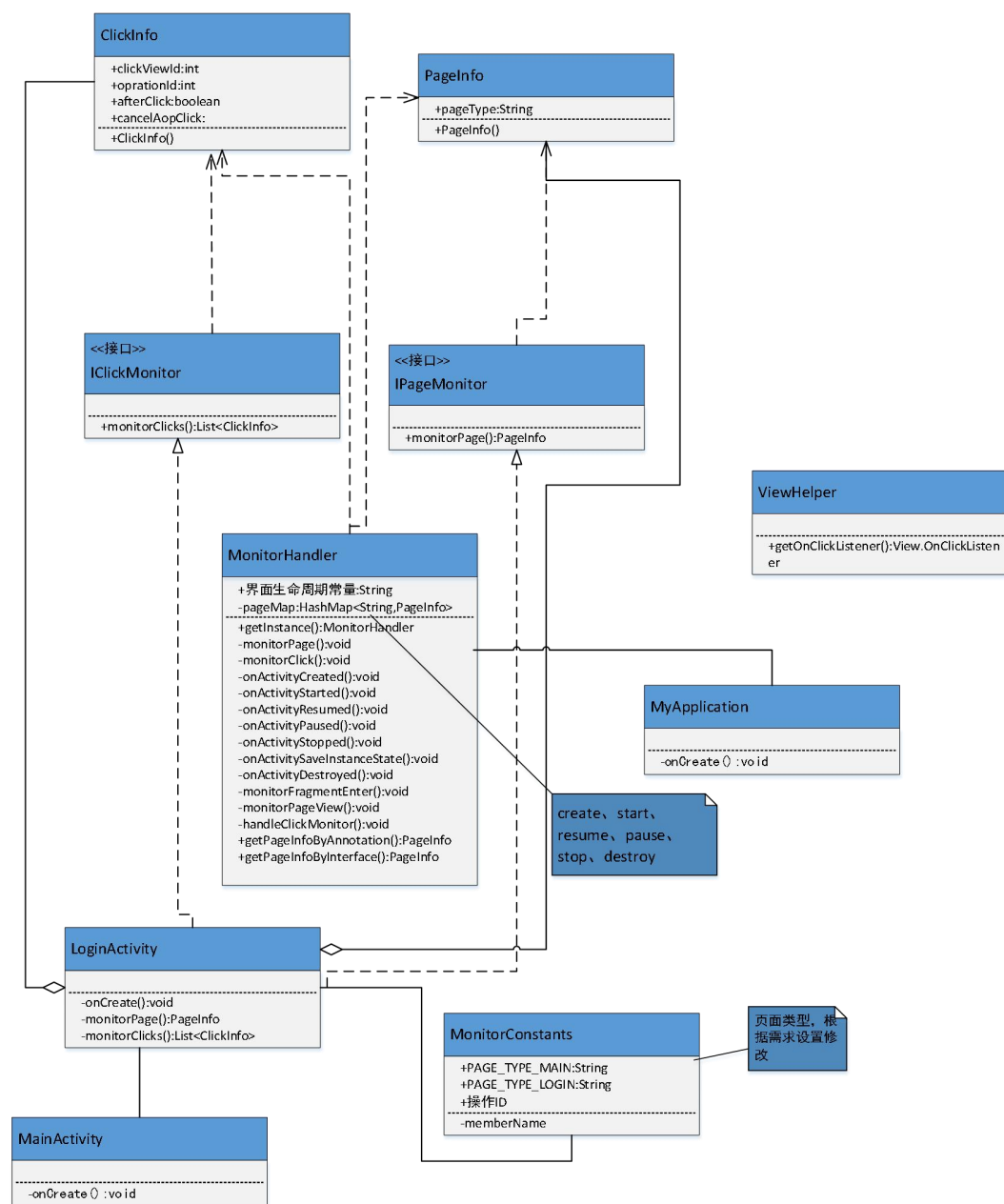


图 3-10 自埋点设计类图

Clickinfo 类和 PageInfo 类存储数据结构类型。装饰器模式的核心类 MonitorHandler 类，IClickMonitor 和 IPageMonitor 为两个接口，分别表示界面类型和点击事件。MonitorConstants 类中存储页面类型的变量以及操作 ID 的常量。PageMonitor 为注释，标注界面类型。

4. 自埋点程序流程图

(1) 通过注解获取界面信息：

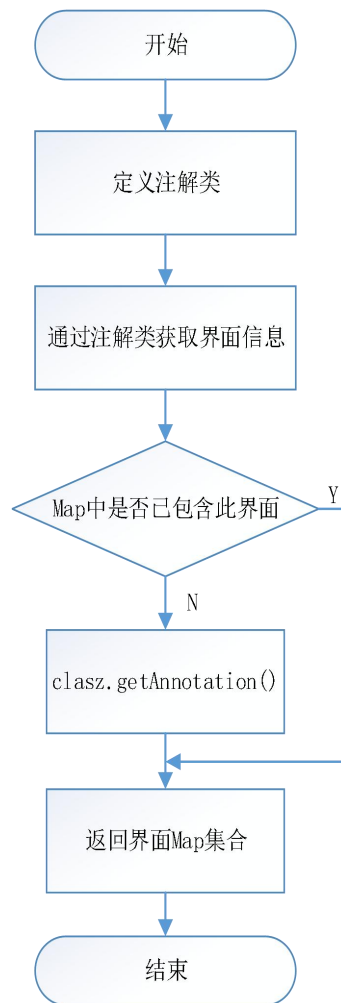


图 3-11 通过注解获取界面信息流程图

(2)通过装饰器包装类实现:

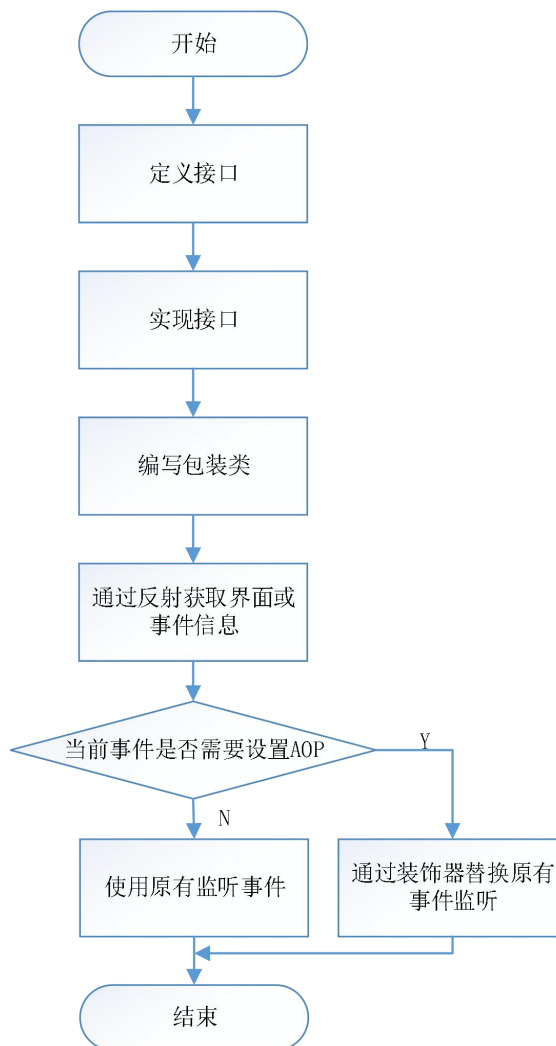


图 3-12 装饰器包装类实现流程图

3.3.2 具体实现

1. 定义数据结构

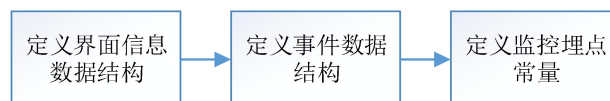


图 3-13 自埋点数据结构设计图

定义点击事件数据结构如下：

表 3-1 点击事件数据结构表

变量名	数据类型	含义
ClickViewId	int	执行点击事件的控件 id
operationId	Int	操作 id
afterClick	boolean	AOP 位置是否再点击之后
cancelAopClick	boolean	是否取消点击监控（AOP）

2. 装饰器实现

（1）定义接口：

```

1. //点击事件监控接口
2. public interface IClickMonitor {
3.     List<ClickInfo> monitorClicks();
4. }
5. //定义页面监控接口
6. public interface IPageMonitor {
7.     PageInfo monitorPage();
8. }
```

（2）具体对象：针对需要实现的方法做初始化操作，即基本的实现核心代码如下：

```

1. public class MainActivity extends AppCompatActivity implements
    AMap.OnPOIClickListener,
2.     AMap.OnMyLocationChangeListener, View.OnClickListener,
3.     GeocodeSearch.OnGeocodeSearchListener,
    NavigationView.OnNavigationItemSelectedListener,
4.     PoiSearch.OnPoiSearchListener, IClickMonitor {
5. // 埋点监控界面的接口实现
6. public PageInfo monitorPage() {
7.     return new PageInfo(MonitorConstants.PAGE_TYPE_LOGIN);
8. }
9. //埋点监控界面 View 点击事件的接口实现
10. public List<ClickInfo> monitorClicks() {
11.     List<ClickInfo> list = new ArrayList<>();
12.     list.add(new ClickInfo(R.id.tv_login, MonitorConstants.OPERATE_LOGIN));
13.     return list;
14. }
15. }
```

（3）装饰类：定义点击监控包装类。

```

1. public abstract class ClickListenerWrapper implements View.OnClickListener {
2.     public View.OnClickListener clickListener;//View 中原始点击监听器
3.     public boolean afterClick;//AOP 位置是否再点击之后
4.     public ClickListenerWrapper(View.OnClickListener clickListener, boolean afterClick) {
5.         this.clickListener = clickListener;
6.         this.afterClick = afterClick;
7.     }
8.     @Override
9.     public void onClick(View v) {
10.         if (afterClick) {
11.             clickListener.onClick(v);
```

```

12.         aopClick(v);
13.     } else {
14.         aopClick(v);
15.         clickListener.onClick(v);
16.     }
17. }
18. protected abstract void aopClick(View view);
19. }

```

(4) 其他具体装饰类实现自己特性的需求。

3.通过注解方式实现埋点

(1) 定义注解

```

1. @Target(ElementType.TYPE)
2. @Retention(RetentionPolicy.RUNTIME)
3. public @interface PageMonitor {
4.     String pageType() default "";
5. }

```

(2) 在接听界面对应的 activity 类使用

```

1. @PageMonitor(pageType = MonitorConstants.PAGE_TYPE_MAIN)
2. public class MainActivity extends AppCompatActivity {
3.     @Override
4.     protected void onCreate(Bundle savedInstanceState) {
5.         super.onCreate(savedInstanceState);
6.         setContentView(R.layout.activity_main);
7.         findViewById(R.id.tv_main).setOnClickListener(v -> {
8.             startActivity(new Intent(MainActivity.this, LoginActivity.class));
9.         });
10.    }
11. }

```

3.3.3 结果分析

输出结果包括点击事件、长按事件、输入事件三个事件类型，以及当前运行界面的生命周期，在 logout 栏下，将输出的 log 日志保存于本地数据库。

表 3-2 logout 日志分析部分数据表

时间	操作日志
2022-05-27 14:56:38.	E/TAG: 界面类型=Main -- 当前执行的生命周期=onResume
2022-05-27 14:56:41	E/TAG: 当前点击操作 ID=12
2022-05-27 14:56:45	E/TAG: 当前点击操作 ID=14
2022-05-27 14:56:51	E/TAG: 界面类型=Main -- 当前执行的生命周期=onPause
2022-05-27 15:00:43	界面类型=Main -- 当前执行的生命周期=onResume

3.4 本章小结

本章介绍了用户 App 使用行为获取的两个思想：一个为基于录制重放思想，使用安卓自动化测试工具获取用户行为轨迹，其中详细 MokeyRunner 和 UIAutomator 两个测试工具的原理和具体使用方法。另一个是使用代码埋点方法获取用户行为数据，分别介绍了代码埋点的两种方式，使用友盟+SDK 实现自动化埋点和基于装饰器设计模式和面向 AOP 编程思想实现自埋点。其中对自埋点算法设计和核心代码进行了详细介绍。

4 用户行为模式量化分析

4.1 分析处理公开数据集

4.1.1 数据收集

数据来源：阿里云天池^[15]中的淘宝用户行为数据集。

该数据集数据过大，其中所包含的约 100 万随机用户的行为数据，均采自 2017 年 11 月 25 日到 2017 年 12 月 3 日，因此可采用 UltraEdit 打开，数据结构如下：

表 4-1 淘宝用户行为数据集数据结构

字段	字段名称	注释
User_id	用户 ID	区分不同用户
Item_id	商品 ID	二级分类
Category_id	商品类别 ID	一级分类
	PV	页面浏览量
Behavior_type	Buy	购买
	Cart	加购
	fav	收藏
timestamp	时间戳	行为发生的时间戳

表 4-2 数据集大小说明

维度	数据量
用户数量	987,994
商品数量	4,162,024
商品类目数量	9,439
行为数量	100,150,807

4.1.2 数据清洗

1. 数据导入

首先用 Navicat 导入数据集，存入 Mysql 数据库，先建立数据库后，选择导入前 10 万行数据。

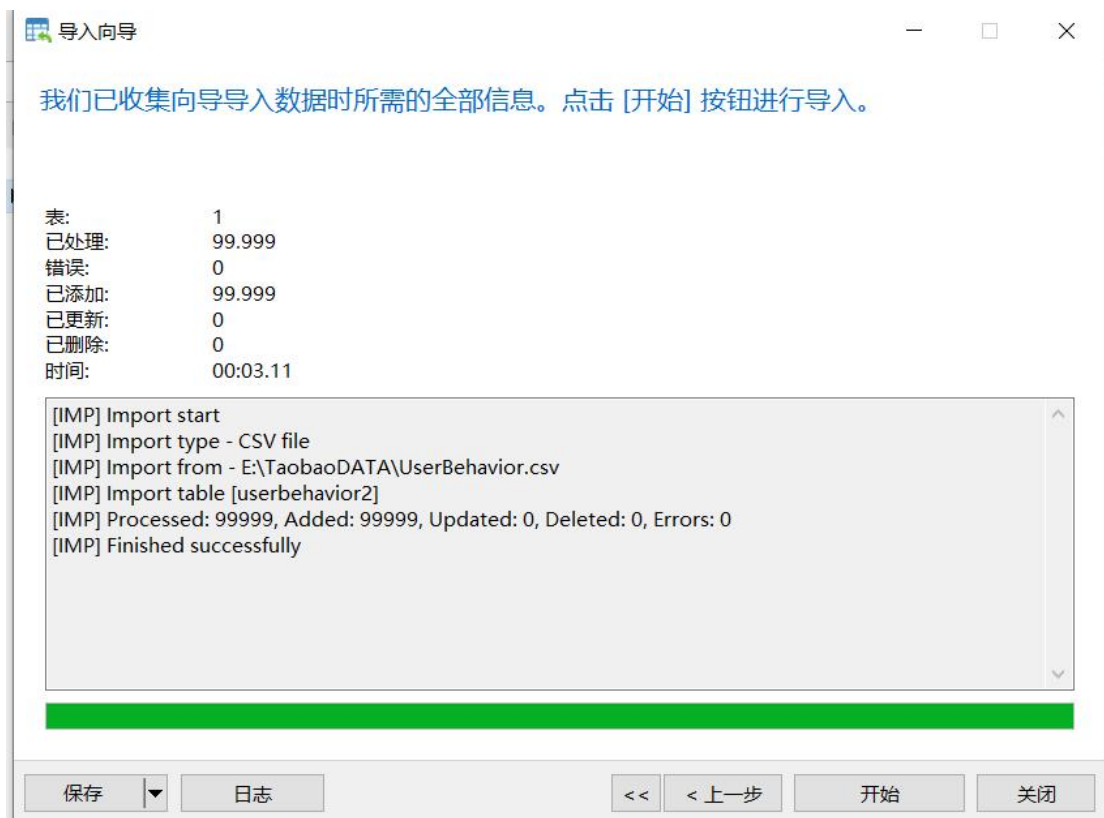


图 4-1 导入数据截图

数据库中建表格式如下表名为 userbehavior:

对象						
test @taobao (127.0.0.1) - ... userbehavior @taobao (127... userbehavior @taobao (12						
保存 添加字段 插入字段 删除字段 主键 上移 下移						
字段	索引	外键	触发器	选项	注释	SQL 预览
名						类型 长度 小数点 不是 null 键
user_id						varchar 255 0 <input checked="" type="checkbox"/>
item_id						varchar 255 0 <input checked="" type="checkbox"/>
category_id						varchar 255 0 <input checked="" type="checkbox"/>
behavior						varchar 255 0 <input checked="" type="checkbox"/>
timestamps						varchar 255 0 <input checked="" type="checkbox"/>

图 4-2 数据库表设计截图

2. 数据处理^[16]

(1) 重复值处理

```
SELECT * FROM userbehavior GROUP BY
user_id,item_id,category_id,behavior,timestamps
HAVING COUNT(*)>1;
```

查找有无重复记录，结果如下图 4-3，可以看到无重复值，那么不需要删除



图 4-3 重复值处理结果

(2) 缺失值处理

查找缺失值记录:

```
SELECT * from userbehavior WHERE user_id IS NULL OR item_id IS NULL
OR behavior IS NULL OR timestamps IS NULL;
```

运行结果如下图 4-4:

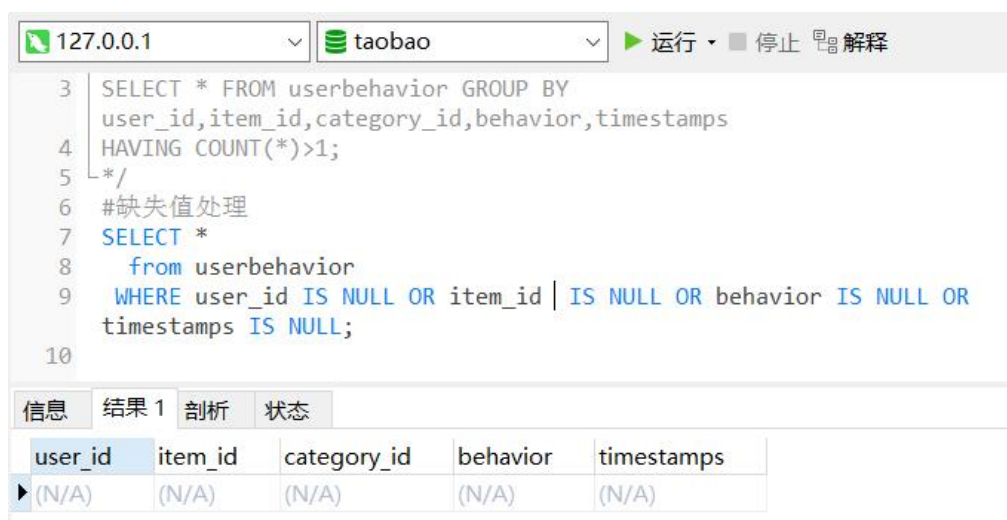


图 4-4 缺失值处理结果

查询语句结果为空，说明数据没有缺失值

(3) 数据一致化处理

用 Mysql 中的 from_unixtime 函数.将时间戳格式转换为标准日期时间:

```
ALTER TABLE userbehavior ADD dates VARCHAR(255);
ALTER TABLE userbehavior ADD hours VARCHAR(255);
UPDATE userbehavior SET dates=FROM_UNIXTIME(timestamps,"%Y-%m-%d");
UPDATE userbehavior SET hours=FROM_UNIXTIME(timestamps,"%H-%i-%s");
```

结果如下图 4-5，可以看到已经将时间戳转换为年月日，时分秒格式。

user_id	item_id	category_id	behavior	timestamps	dates	hours
1	2333346	2520771	pv	1511561733	2017-11-25	06-15-33
1	2576651	149192	pv	1511572885	2017-11-25	09-21-25
1	3830808	4181361	pv	1511593493	2017-11-25	15-04-53
1	4365585	2520377	pv	1511596146	2017-11-25	15-49-06
1	4606018	2735466	pv	1511616481	2017-11-25	21-28-01
1	230380	411153	pv	1511644942	2017-11-26	05-22-22
1	3827899	2920476	pv	1511713473	2017-11-27	00-24-33
1	3745169	2891509	pv	1511725471	2017-11-27	03-44-31
1	1531036	2920476	pv	1511733732	2017-11-27	06-02-12
1	2266567	4145813	pv	1511741471	2017-11-27	08-11-11
1	2951368	1080785	pv	1511750828	2017-11-27	10-47-08

图 4-5 时间标准化结果

(4)异常值处理

官网数据集中的日期是在 2017-11-25 至 2017-12-3，所以异常值处理则转为查询日期是否在规定时间内。

```
select * from userbehavior2 where dates<'2017-11-25' or dates>'2017-12-04';
```

user_id	item_id	category_id	behavior	timestamps	dates	hours
1	2333346	2520771	pv	1511561733	2017-11-25	06-15-33
1	2576651	149192	pv	1511572885	2017-11-25	09-21-25
1	3830808	4181361	pv	1511593493	2017-11-25	15-04-53
1	4365585	2520377	pv	1511596146	2017-11-25	15-49-06
1	4606018	2735466	pv	1511616481	2017-11-25	21-28-01
1	230380	411153	pv	1511644942	2017-11-26	05-22-22
1	3827899	2920476	pv	1511713473	2017-11-27	00-24-33

图 4-6 异常值处理结果

对图 4-6 中分析得到的异常值进行删除处理：

```
DELETE from userbehavior2 where dates<'2017-11-25' or dates>'2017-12-04';
```


4.2 用户行为模型的建立

4.2.1 界面访问路径模型

界面访问路径模型用桑基图表示，示例图见下图 4-7。它可以理解为一种特定类型的流程图，从每一个节点延伸的分支的宽度表示数据流量的大小。每条边表示上一节点到该节点的流量。

具体建立过程如下：将自埋点程序运行后，对 log 日志中的数据进行分类，对于界面类型，将页面对应的类名转化为 App 界面对应的界面名称；对于按钮事件，将 log 日志输出的按钮 ID 根据最初设定的规则，转换为 App 界面的按钮名称。之后通过使用 Excel 筛选分析统计得到界面访问流程。使用在线绘制工具,导入 EXCEL 数据表格，即可生成桑基图。

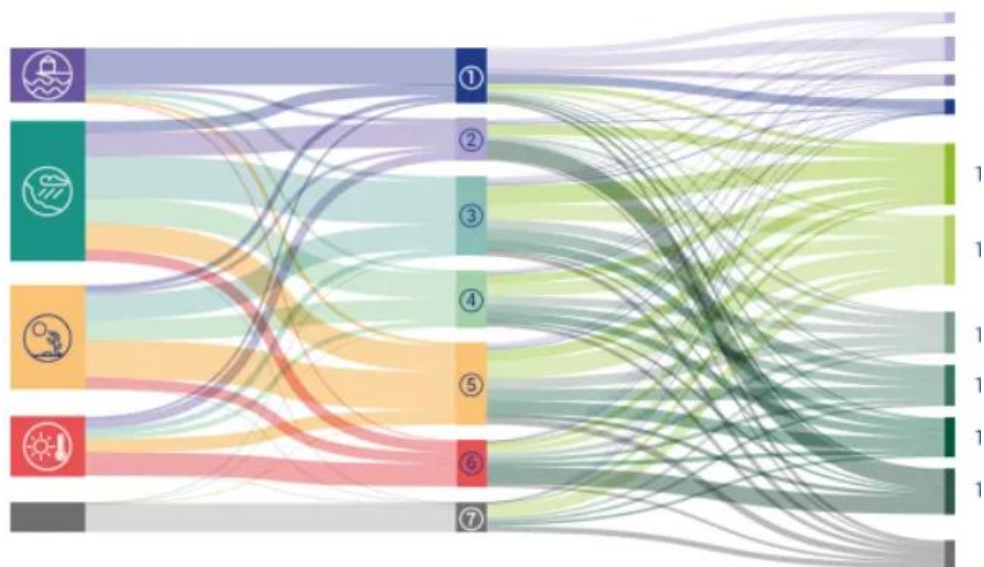


图 4-7 桑基图示例图

4.2.2 界面热力图模型

界面热力图可以展示界面位置的点击频率，颜色深浅代表其点击次数多少。如下图 4-8 为热力图实例。具体方法如下：统计 log 日志中的按钮点击，长按事件，将 log 日志输出的按钮 ID 根据最初设定的规则，转换为 App 界面的按钮名称。之后将 log 日志导出后进行文本分析，以相同事件控件出现次数作为每一个按钮的点击频率，统计分析后绘制 Excel 表格，转化为简洁版界面点击热力图。

对于更加直观的界面事件热力图，需要借助在线热力图生成工具，本文使用灰大胜热力图生成工具。

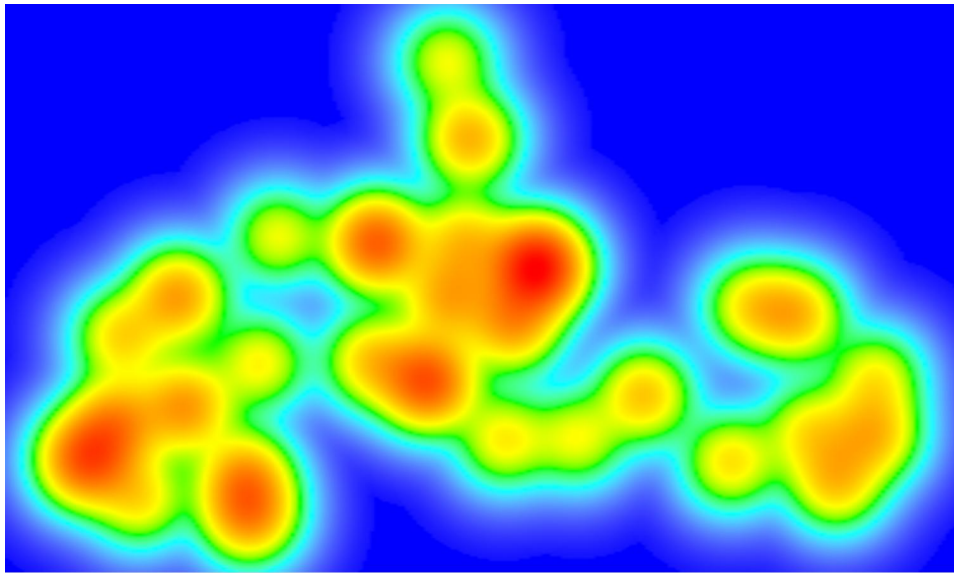


图 4-8 界面热力图

4.2.3 行为事件分析模型

利用 SQL 语句量化行为事件情况。表 4-3 为行为事件分析的五个指标和其简述。

表 4-3 行为事件分析表

指标	描述
PV	页面总访问量
UV	网页访问用户总数
每日活跃情况	用户行为数量随时间变化情况
每时活跃情况	一天内用户行为数量随时间变化情况
time	用户行为发生时间

对应 SQL 代码如下：

1. 计算 PV

```
SELECT COUNT(*) FROM 表名 where 行为='PV';
```

2. 计算 UV

```
SELECT COUNT (DISTINCT 用户账号) from 表名 as UV;
```

3. 每日活跃情况

创建视图，统计各种类型的行为在同一天数量的数量，代码如下：

1. CREATE VIEW 用户活跃时间分布（天） AS SELECT
2. dates 日期,
3. COUNT(DISTINCT user_id) 访客数,
4. sum(CASE WHEN behavior = 'pv' THEN 1 END) 行为 A,
5. sum(CASE WHEN behavior = 'fav' THEN 1 END) 行为 B,
6. sum(CASE WHEN behavior = 'cart' THEN 1 END) 行为 C,
7. sum(CASE WHEN behavior = 'buy' THEN 1 END) 行为 D
8. FROM 表名 GROUP BY 日期 ORDER BY 日期;

4. 每时活跃情况

与每日活跃情况代码相似，只需修改日期为小时。

5. 用户行为发生时间

Select 时间 from 表名 where 行为=“行为 A”;

4.2.4 漏斗模型

漏斗模型也是一种方法论或思维模式，示例图见下图 4-9。漏斗模型更多反映的是用户的流失和转化，帮助产品经理和运营更好做用户增长和产品运营。具体方法如下：例如对于电商类 App，计算点击率、收藏率、加购率、下单率、回购率、计算转化比后，使用 Tableau 工具或者 EXCEL 绘制用户转化漏斗。

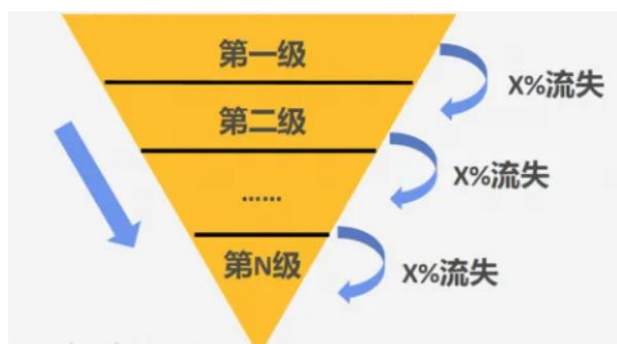


图 4-9 漏斗模型示例图

4.3 本章小结

本章分别介绍了使用 SQL 语句对公开数据集中的用户行为数据进行分析的方法以及建立的四个 App 用户行为使用行为模式的量化分析模型：界面访问路径模型、界面热力图模型、行为事件分析模型、漏斗模型，并且具体介绍了每一种模型的建立方法。

5 实验设计与结果展示

5.1 电商类 App 用户使用行为模型

5.1.1.分析用户访问情况

利用 SQL 语句计算网页访问用户总数 (UV)、计算页面总访问量 (PV)、页面流量。页面流量是指每个访客在这个界面的平均浏览次数, PV/UV 即为页面流量。语句如下:

```
SELECT COUNT (DISTINCT user_id) as UV,  
(SELECT COUNT(*) FROM userbehavior='PV') as PV,  
((SELECT COUNT(*) FROM userbehavior='PV')/COUNT(DISTINCT user_id) as  
'PV/UV'  
FROM userbehavior;
```

得到结果如下: UV:983,PV:89664;PV/UV:91.2146

页面流量高达 91 次/人,由此推得在这 9 天内, 平均每个用户每天访问界面十次, 因此, 得出结论: 用户访问淘宝的频率很高。

5.1.2.构建漏斗模型

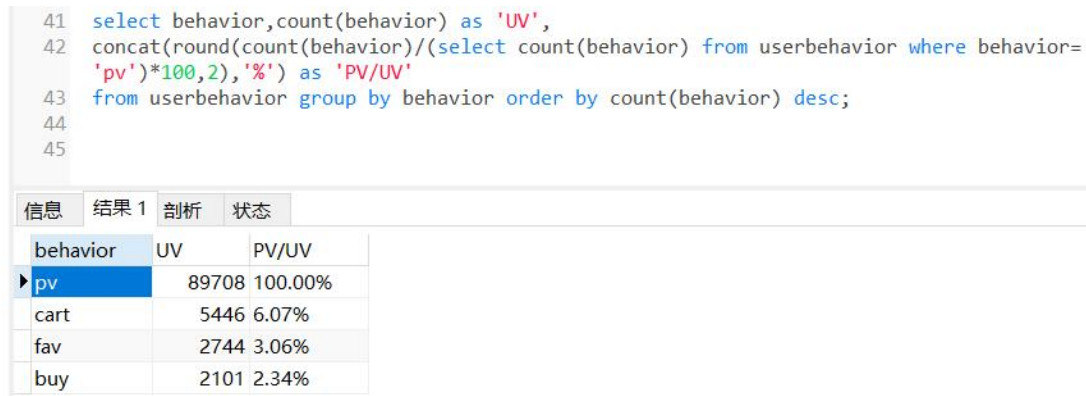


图 5-1 购买行为转化率结果

由此将四个行为分为三个环节, 见下表 5-1:

表 5-1 行为转化环节表

第一环节	第二环节	第三环节
点击->加购	加购->收藏	收藏->购买

根据表 5-3 使用 EXCEL 绘制转化率图, 如图 5-2 所示。

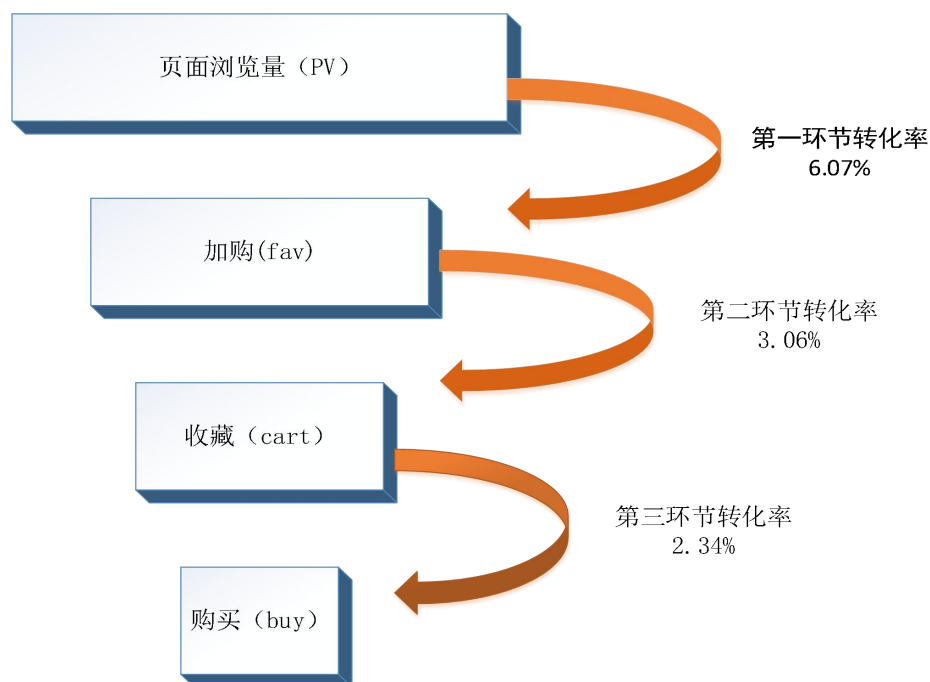


图 5-2 转化漏斗图

观察转化漏斗可知：

页面浏览量到加购环节转化率 6.07%，显然在这一过程中，用户流失率很高，加购概率较低。但是从加购收藏到购买的转化率相对较高。

5.1.3.用户行为习惯分析

1.每日活跃情况

```
CREATE VIEW 用户活跃时间分布（天） AS SELECT
dates 日期,
COUNT( DISTINCT user_id ) 访客数,
sum( CASE WHEN behavior = 'pv' THEN 1 END ) 点击,
sum( CASE WHEN behavior = 'fav' THEN 1 END ) 收藏,
sum( CASE WHEN behavior = 'cart' THEN 1 END ) 加购,
sum( CASE WHEN behavior = 'buy' THEN 1 END ) 购买
FROM userbehavior GROUP BY dates ORDER BY dates;
```

创建得到的视图如下图 5-3 所示：

日期	访客数	点击	收藏	加购	购买
2017-11-25	63	874	24	56	15
2017-11-26	67	871	27	49	18
2017-11-27	62	786	24	51	29
2017-11-28	70	1168	35	62	24
2017-11-29	68	1035	49	59	22
2017-11-30	71	1036	34	52	14
2017-12-01	67	793	18	48	10
2017-12-02	90	1190	29	80	20
2017-12-03	90	1229	58	80	27

图 5-3 Mysql 中的用户日活视图

根据 Mysql 中导出的日活视图，在 EXCEL 中对各指标随时间变化进行可视化。

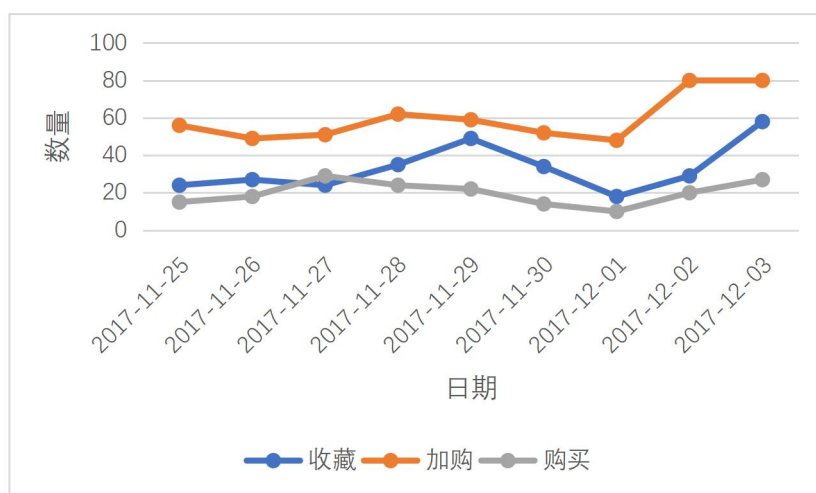


图 5-4 各行为指标随日期变化趋势

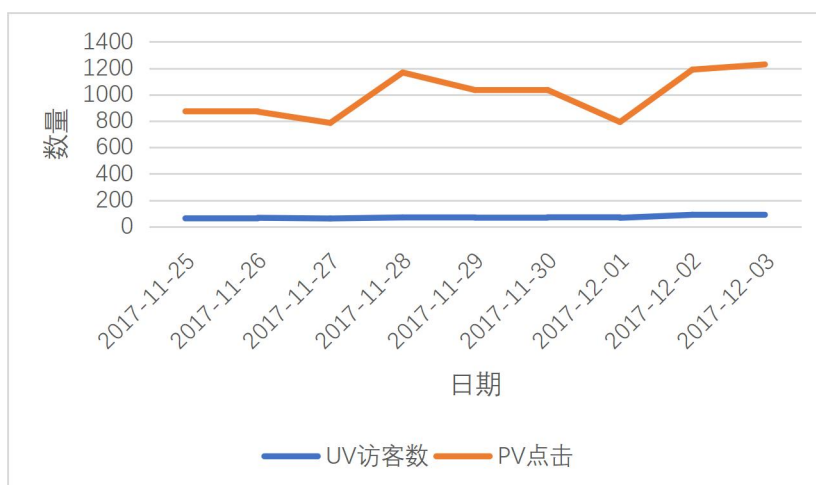


图 5-5 用户日活图

根据图 5-4、图 5-5 可以看到，在整个日期范围内用户活跃程度不高，仅在 11 月 28 日和 12 月 2 日有明显的增长。12 月 2 日-3 日 PV、UV、收藏数、加购数等等都有明显的增长,但购买量增加的很少。

2. 每时活跃情况

创建用户每小时活跃情况的视图，SQL 语句如下：

```
CREATE VIEW 用户活跃时间分布(时) AS SELECT
h 时间,
COUNT( DISTINCT user_id ) 访客数,
sum( CASE WHEN behavior = 'pv' THEN 1 END ) 点击,
sum( CASE WHEN behavior = 'fav' THEN 1 END ) 收藏,
sum( CASE WHEN behavior = 'cart' THEN 1 END ) 加购,
sum( CASE WHEN behavior = 'buy' THEN 1 END ) 购买
FROM userbehavior GROUP BY h ORDER BY h;
```

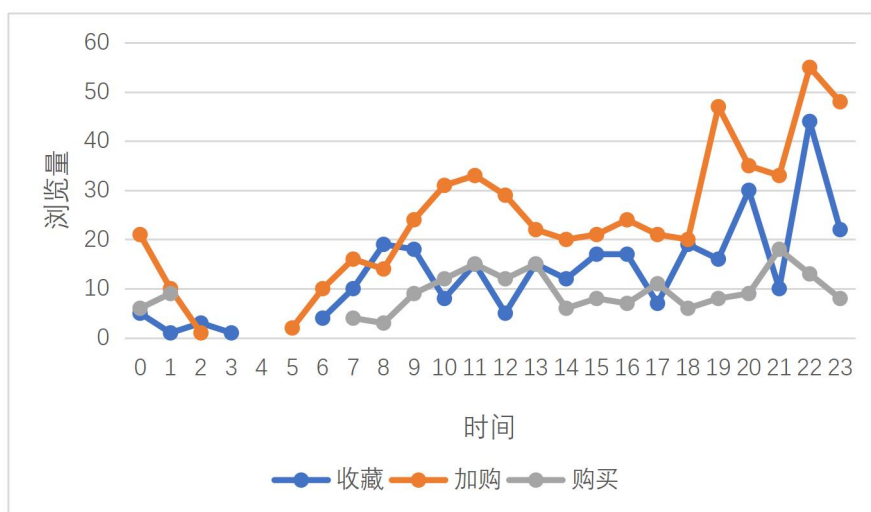


图 5-6 各行为指标随小时变化

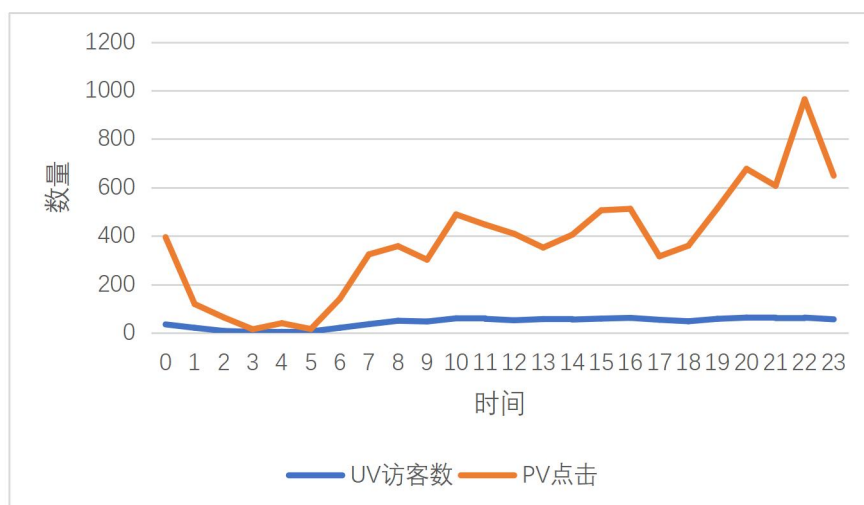


图 5-7 用户时活图

通过可视化用户行为在一天中的分布，可以发现用户活动从每天 0:00 到 4:00 迅速下降到最低值，符合用户日常作息，在中国时区内，这个时间段，大多数人都在休息。从 5:00 到 10:00 开始快速上升，从 10:00 到 18:00 趋于稳定。18:00 后，用户活动数量开始迅速上升，晚上 1:00 达到一天中用户活动的最高值，可以看到，夜间用户逛淘宝的频率较高。而且夜间下单率，加购数量也急速上升，一方面由于这一时间段内，用户数量较多，另一方面是由于夜间用户容易冲动消费。

3. 用户在每天不同时间段的活跃度周内周末对比

SQL 语句获取小时：

```
ALTER TABLE userbehavior ADD hours_ VARCHAR(255);
```

```
UPDATE userbehavior SET hours_ = HOUR(timestamps);
```

结果如下：

对象		用户活跃时间分布 (时) @taobao						userbehavior @taobao (127...	
开始事务		文本		筛选	排序	导入		导出	
user_id	item_id	category_id	behavior	timestamps	dates	hours	h		
1	2333346	2520771	pv	1511561733	2017-11-25	06-15-33	06		
1	2576651	149192	pv	1511572885	2017-11-25	09-21-25	09		
1	3830808	4181361	pv	1511593493	2017-11-25	15-04-53	15		
1	4365585	2520377	pv	1511596146	2017-11-25	15-49-06	15		
1	4606018	2735466	pv	1511616481	2017-11-25	21-28-01	21		
1	230380	411153	pv	1511644942	2017-11-26	05-22-22	05		
1	3827899	2920476	pv	1511713473	2017-11-27	00-24-33	00		
1	3745169	2891509	pv	1511725471	2017-11-27	03-44-31	03		
1	1531036	2920476	pv	1511733732	2017-11-27	06-02-12	06		

图 5-8 获取小时的 SQL 语句运行结果图

获取用户在周内某天的活跃度：

```
SELECT hours_,COUNT(*) FROM userbehavior WHERE dates='2017-11-28' AND behavior='PV' GROUP BY hours_ ORDER BY hours_ ASC;
```

获取用户在周末的活跃度：

```
SELECT hours_,COUNT(*) FROM userbehavior WHERE dates='2017-12-02' AND behavior='PV' GROUP BY hours_ ORDER BY hours_ ASC;
```

导出结果后，在 EXCEL 中绘制周内和周末 24 小时的用户浏览量。见下图 5-9：

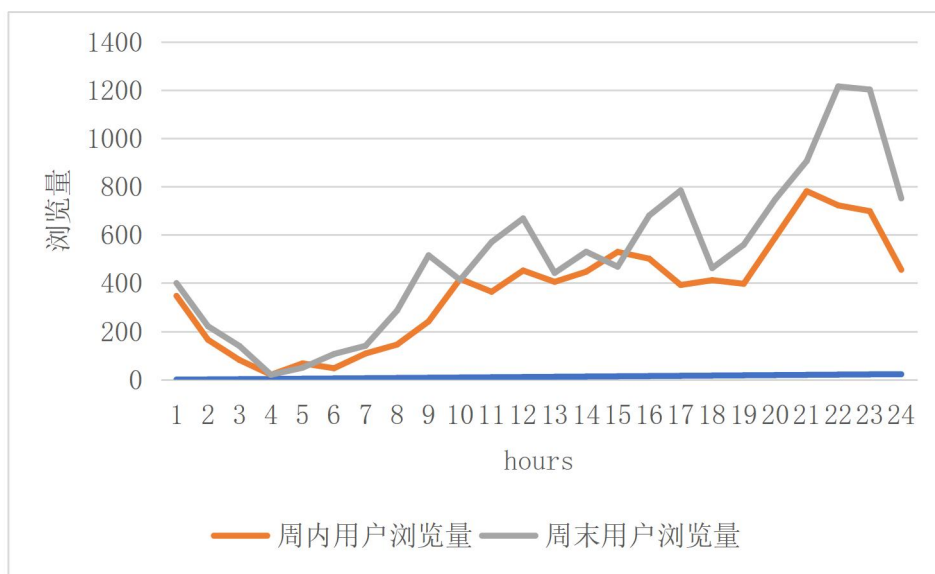


图 5-9 周内和周末 24 小时的用户浏览量

图 5-9 中展示了 11 月 28 日和 12 月 2 日每小时的活跃数据，二者有相同的趋势和特点。

从结果可以看出，在 1-8 点内用户活跃度较低，符合用户的作息规律；9-19 点时段内活跃度有小幅上升；19 点过后活跃度曲线迅速上升，直到 22 点到达顶峰。可见，在 19 点至 23 点的这段时间是一天中最活跃的时间段。而且整体来看，周末用户浏览量都是高于周内的。

5.2 地图导航类 App 用户使用行为模型

本文以开源 App 模仿高德地图，使用代码埋点方法，得到用户使用地图导航类 App 的行为模型。下图是整个实验过程流程图：

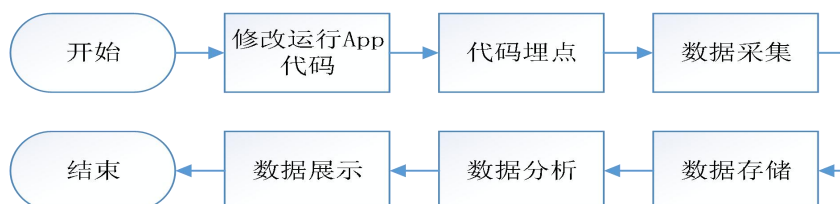


图 5-10 用户使用行为建模流程图

首先编写 App 代码，利用前文提到的代码埋点方法进行埋点；其次通过 log 日志输出的文件对用户行为数据进行采集，导出后存到数据库中；再次，对数据进行分析，之后使用前文建立的模型对其进行可视化。

5.2.1 获取用户行为轨迹

将 3.3 模块中的技术点应用到已有的自写的 App 中，对几个界面进行埋点。

1.埋点核心方法

A implements Interface (接口方法 doSth)

B implements Interface{ B(A a){} doSth{a.doSth() + 埋点逻辑}

具体步骤如下：

- (1) View 获取已设置好的 AListener。反射获取
- (2) 通过装饰器设置到 BListener 中，并将 BListener 重新设置给目标控件
- (3) 其他类似 OnClickListener 监听比如上拉刷新下拉加载控操作的监听、其他监听等都可以仿照 (1) (2) 中步骤处理。例如 CustomView 有设置上拉刷新监听 setOnRefreshListener (OnRefreshListener refreshListener)，在埋点时要获取到该 refreshListener 对象，先看 CustomView 是否提供了获取 refreshListener 对象的方法，若有则直接调用，否则通过反射获取 refreshListener，然后通过 (2) 中所讲的装饰器替换即可

具体如下：

- (1) MonitorConstants 类中自定义操作 ID 常量和界面类型
- (2) MainActivity 类需要继承包装类的两个接口，实现其方法，并且添加注解。
- (3) MyApp 类中做监控类的初始化。

2.日志分析

将埋好点的 APK 安装在 10 个用户手机上，或者写个 HTTP 接口，App 请求接口把数据发送到带有公网 ip 的服务器，收集到的日志保存在 Mysql，将设备 ID 作为保存每一个用户的唯一 ID，保存 log 日志文件，部分 log 日志截图如下：

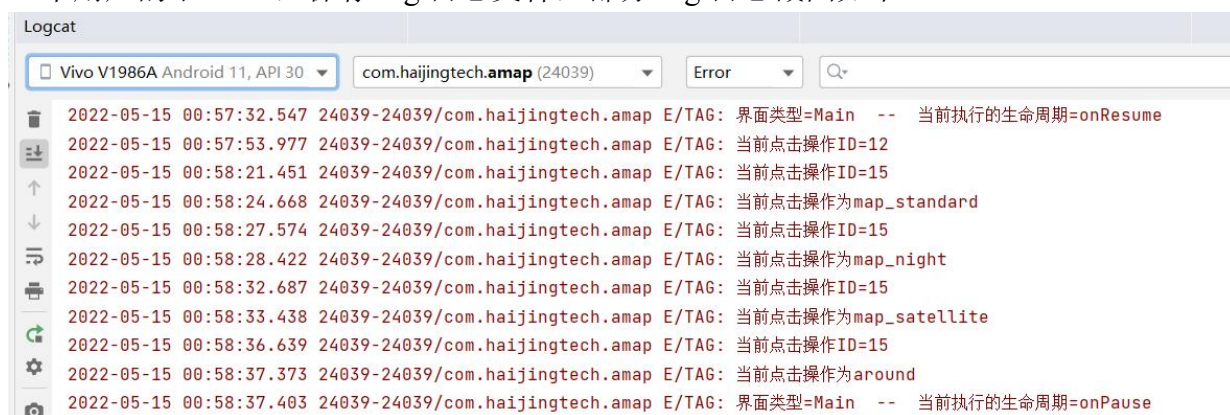


图 5-11 log 日志截图

可以看到，log 日志会输出当前界面的类型名称和执行的生命周期，并且打印当前

的事件操作。

5.2.2 分析用户行为数据建立模型

1. 界面访问路径模型

本文使用在线工具 <https://www.zxgj.cn/g/sankey> 绘制桑基图。通过前文提到的数据分析方法，绘制界面访问路径图如下：

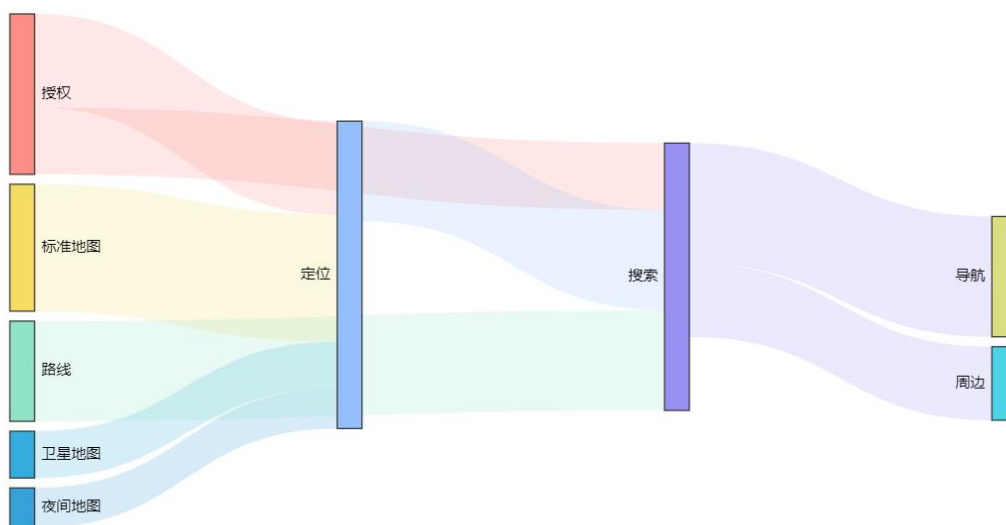


图 5-12 用户轨迹桑基图

从图中可以看到，用户进入 App 后，主要点击授权，地图模式选项，路线事件，而授权事件一般为新用户需要进行的操作。再进行定位操作后，用户数据流主要流向搜索栏，最终主要流向导航控件和周边页面。

2. 界面热力图模型

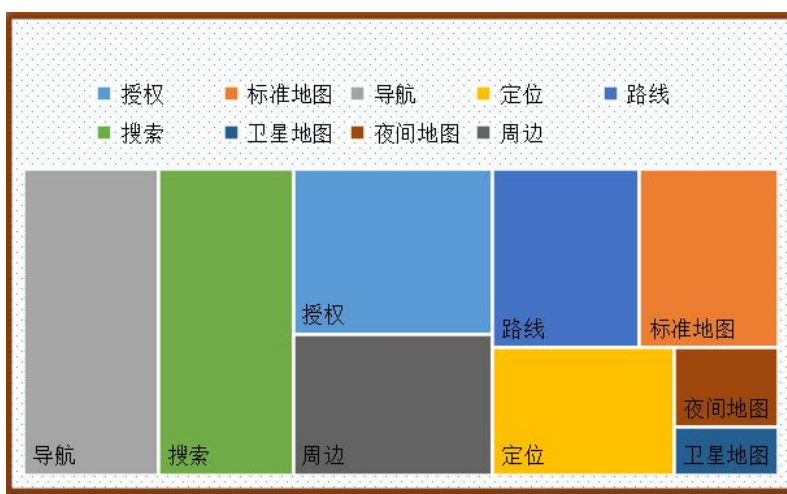


图 5-13 界面点击热力图

使用 EXCEL 绘制界面点击热力图，面积的大小表示控件点击频率，可以看到，导航和搜索控件使用频率最高，而夜间地图和卫星地图界面使用频率较低。原因是该 App 的主要受众人群是仅将地图用于日常生活的人群，所以卫星地图用的不多。

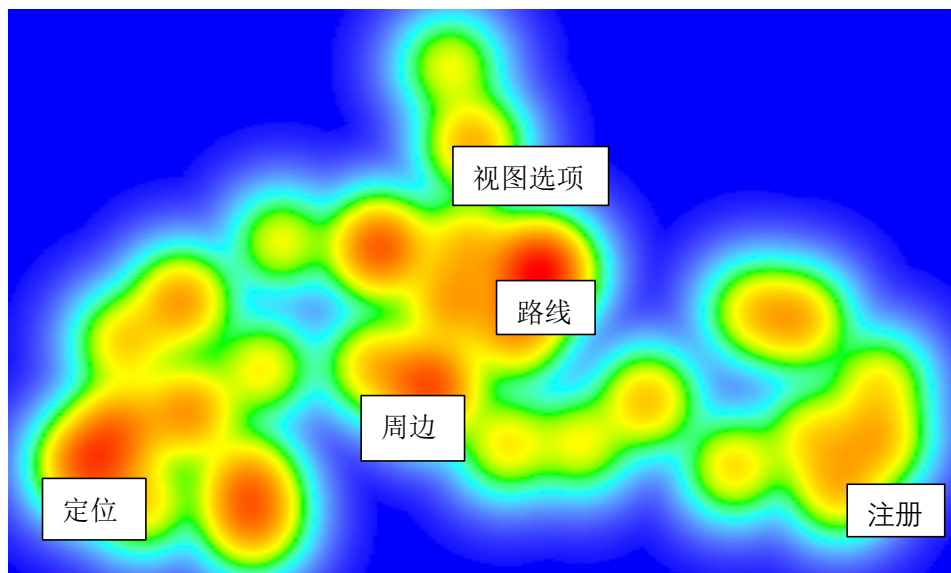


图 5-14 时间点击热力图

将数据加工导入 VOSViewer，绘制界面点击热力图^[17]，面积的大小表示控件点击频率，可以看到，导航和搜索控件使用频率最高，而夜间地图和卫星地图界面使用频率较低。原因是该 App 的主要受众人群是仅将地图用于日常生活的人群，所以卫星地图用的不多。

3.行为事件分析模型

通过对已经进行量化过的 log 日志文件的时间进行分析，将同一个界面生命周期内的时间间隔作为用户在某个界面停留时间长度。使用 TableAU 绘制用户行为时间流量图。

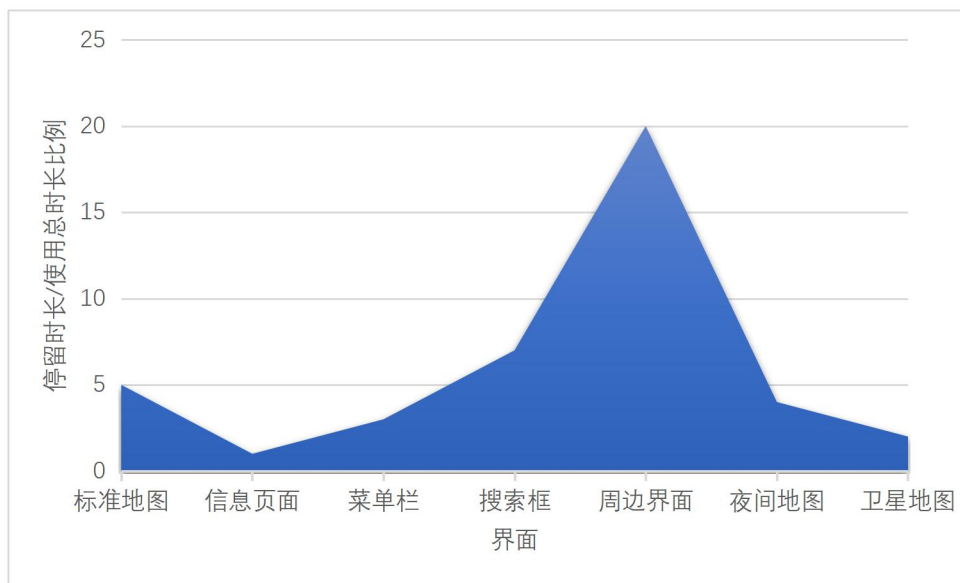


图 5-15 用户行为时间流量图

从图中可以看到，用户在周边界面停留时间最长，不难理解大多数用户使用地图会关注周边吃喝玩乐。

5.3 本章小结

本章对两个具体 App 进行实验,针对第一个实验,使用 SQL 语句对公开数据集进行分析,从而探索用户的行为模式。针对第二个案例，以自开发地图导航类 App 进行自埋点后，根据用户行为数据建立用户使用行为模式并可视化。

6 结论与展望

6.1 本文工作总结

随着互联网的发展，智能手机普及，目前智能手机端主要的操作系统中，Android 操作系统占比份额最大，因此，本文研究 Android 应用的用户使用行为模式，通过挖掘用户使用行为数据，并进行量化分析。

对于 App 用户使用行为数据的获取和挖掘，本文详细介绍了两种方法：基于录制重放的用户使用应用轨迹获取模式和基于埋点的 App 用户使用行为数据。对于前者，本文介绍了用安卓自动化测试工具 MonkeyRunner 和 UIAutomator 获取 App 用户行为轨迹的使用方法；对于后者，本文设计并实现了基于装饰器模式和面向切面编程思想的自埋点算法，并且阐述了如何使用第三方 SDK 进行代码埋点。对于用户行为使用模式的量化分析，本文总结了四个用户行为模型：界面访问路径模型、界面热力图模型、行为事件分析模型、漏斗模型，并详细阐述了模型建立的方法。

基于提出的研究方法，本文以淘宝 App 和自开发地图导航 App 两个应用为例，对其用户使用行为进行获取，对其用户行为数据建立行为模型并可视化，验证了本文研究方法的可行性，对致力于 App 用户使用行为获取研究的人员提供了思路，有助于应用的测试开发人员和产品设计人员工作的展开。

6.2 未来展望

对于安卓 App 用户使用行为轨迹获取，本文通过大量实验后，选取两个具有代表性的 App 进行实验，后续工作可以将埋点方案运用于多个开源 App 上，并用 Python 的 flask 做一个接口，挂到服务器后，收集用户行为数据，以用户手机标识为唯一 ID 保存在数据库。对于自开发 App 大量用户数据行为进行分析。

对于安卓 App 用户使用行为模式量化，本文在自开发 App 中仅建立静态的用户行为模式可视化图，而实际上这些模型是随时间用户使用情况实时变化的。后续工作中可以将已有部分用户使用的自写 App 与第三方 SDK 集成，调用第三方已有的服务生成动态实时的用户行为模型。并且对已有的用户行为模型用于同类相似 App 中进行复用，从而指导后期软件测试和设计维护。

致 谢

我与西科的故事，始于 2018 年的秋天，止于 2022 年的夏天。这四年来，从青涩懵懂到慢慢成熟，从害怕彷徨到勇敢坚定，要感谢的人太多太多。

首先，我要感谢的是我的导师刘晓建老师。从大二第一次上老师的计组课，老师很多次不管多忙都耐心解答我课下遇到的问题到大三被老师讲的有趣的操作系统课吸引，再到大四不管是学习上，思想上，还是毕业设计，老师都耐心给我指导和建议。我很庆幸自己能够成为老师的学生，感谢老师这三年的帮助和鼓励。

其次，我要感谢的是李立红老师和史晓楠老师，我印象很深，大一第一门专业课 C 语言课，李立红老师每一次都很温柔解答我的疑惑，大二另一门 VC++ 课程我遇到编码问题，老师在机房给我一遍又一遍讲解，改错误。史晓楠老师指导我发表大学里的第一篇学术论文，在我考研路上给予我很多帮助，在我遇到挫折的时候给我鼓励开导，在我的劳动成果不被外界认可的时候，老师为我出面作证。还有理学院的数学建模团队的老师，尤其是张守刚老师，是他的数学建模公开课，让我从此喜欢上了数学建模，在我比赛心态奔溃的时候不断鼓励我开导我。我很感谢两位老师让我能在最难熬的那段日子坚持下来，走出那段无法释怀的阴影。我还要感谢我的几位辅导员老师，感谢他们对我的帮助和指导。感谢大学期间遇到的诸多亦师亦友的老师，是他们的认真负责，有趣的授课，让我学习到了新的知识和学习方法。

再次，我要感谢我的家人们、我的朋友们，他们的陪伴是我负重前行最大的动力，他们永远是我坚强的后盾。我要感谢我的大学同学们，我亲爱的舍友们，是她们陪伴我整个四年，关心我，包容我，帮助我。我要感谢我大学期间所有遇到的学长学姐，给我很多的帮助和鼓励。感谢这群可爱善良的人们陪我长大。感谢网络上社交群里帮我解决问题的热心网友们，感谢食堂的叔叔阿姨，感谢宿管阿姨，感谢校医院医生，感谢校园的小动物，感谢校园广播站，感谢图书馆门卫叔叔，感谢常常偶遇到的晚霞，感谢小吃街……感谢我遇到的所有温暖与善良。

最后，感谢自己，在所有选择面前，勇敢选择了那条具有挑战的路，感谢自己没有忘记初心，感谢自己在不被外界认可的时候依然相信自己。

这一路走来，跌跌撞撞，磕磕碰碰，迷茫过，崩溃过，绝望过，怀疑过，但从未放弃过。我相信，所有的阻力都将推动我更加勇敢坚定的前进。肆载寒暑，无论是喜悦还是酸楚，所有经历，于我都是礼物；所有相遇，于我都是宝藏。肆年里的所有羁绊，或许做不到一生铭记，但绝对一生感恩。我深知，以后的路任重而道远，我将以更勇敢的我去迎接未来更大的挑战。

参考文献

- [1] 中国信通院.国内手机产品通信特性与技术能力监测报告(2020 年第一期)[EB/OL].http://www.caict.ac.cn/kxyi/qwfb/qwsj/202004/t20200427_280405.htm,2022-1-21.
- [2] Number of available applications in the Goole Play Store from December 2009 to December 2020[EB/OL].<https://www.statista.com/statistics/266210/number-of-available-application-in-the-google-play-store/>,2022-01-10.
- [3] Gartner Says Global Smartphone Sales Declined 20% in Second Quarter of 2020 [EB/OL].<https://www.gartner.com/en/newsroom/pree-releases/2020-8-25-gartner-says-global-smartphone-sales-declined-20—in->,2022-1-10.
- [4] 金小俊,孙艳霞,赵化等.基于数据和状态的移动应用可视化埋点方法[J].电脑知识与技术,2022,3(2):19-21.
- [5] 蔡榆榕.运用埋点技术实现高校应用服务平台过程监控[J].中国管理信息化,2021,24(15):187-188.
- [6] 吴荣灿,罗嘉龙,郑伙群等.基于大数据技术的用户行为分析平台的设计与实现[J].软件导刊,2021,17(12):40-41.
- [7] 宋海涛,韦大伟,汤光明等.基于模式挖掘的用户行为异常检测算法[J].小型微型计算机系统,2016,(2):221-225.
- [8] 毛群.跨安卓应用的用户行为模式挖掘及复用研究[D].北京:北京化工大学,2021.
- [9] Qun Mao,Weiwei Wang,Feng You etal.User behavior pattern mining and reuse across similar Android apps[J].The Journal of Systems & Software,2022,183(2022):1-9.
- [10] 郭景怡.iOS 应用场景 API 序列模式挖掘技术的研究和实现[D].北京:北京邮电大学,2016.
- [11] Wu Tingting.Research on User Access Pattern Mining Based on Web Lo[J].Asia-pacific Journal of Convergent Research Interchange,2018,22(3):135-148.
- [12] 张力平.Web 用户行为模式挖掘及其在 E-Learning 系统中的应用[J].软件导刊,2014,6(13):117-119.
- [13] D.Anandhi1,M.S.Irfan Ahmed.Prediction of user's type and navigation pattern using clustering and classification algorithms[J].Cluster Comput,2017,122(2):2-5.
- [14] 薛群群.基于数据挖掘的 Web 用户使用模式生成方法研究[D].北京:北京交通大学,2017.

- [15]天池小喵喵.User Behavior Data from Taobao for Recommendation[OL].
<https://tianchi.aliyun.com/dataset/dataDetail?dataId=649&userId=1>,2022.
- [16]Typhone.淘宝用户行为分析[OL].<https://zhuanlan.zhihu.com/p/187194560>,2022.
- [17]刘洪辉,吴岳芬.用户行为模式挖掘问题的研究[J].计算机技术与发展,2006(05):50-52,55.