

# 基于细粒度用户使用数据的智能手机电池续航时间预测模型\*

李豁然<sup>1,2</sup>, 刘譞哲<sup>1,2</sup>, 梅俏竹<sup>3</sup>, 梅宏<sup>1,2</sup>

<sup>1</sup>(北京大学 信息科学技术学院 软件研究所,北京 100871)

<sup>2</sup>(高可信软件技术教育部重点实验室(北京大学),北京 100871)

<sup>3</sup>(密歇根大学 信息学院,美国 密歇根州 安娜堡 48109)

通讯作者: 刘譞哲, E-mail: liuxuanzhe@pku.edu.cn



**摘要:** 如今,智能手机已经成为人们日常生活中重要的组成部分.然而,在智能手机软硬件能力高速发展的同时,智能手机的电池能力却未能取得突破性的进展.这导致电池的续航能力经常会成为用户使用智能手机时的体验瓶颈.为了提高用户的使用体验,一种可行的方法是为用户提供电池续航时间预测.准确的电池续航时间预测能够帮助用户更加高效地规划其使用,从而能够提高其使用体验.由于缺乏高质量数据的支持,现有电池续航时间预测方法通常比较简单,较难在真实场景下发挥实际用途.为了解决这一问题,本文基于一组细粒度大规模真实用户数据集,提出了一个智能手机电池续航时间预测模型.为了验证模型的效果,本文基于 51 名用户 21 个月内的细粒度使用数据进行了实验验证.结果显示,用户在发起查询时的使用行为、在当前会话内的使用行为、以及其历史使用习惯能够在不同程度上帮助电池续航时间预测.总体来说,本文提出的模型能够显著提升预测准确度.

**关键词:** 移动计算;电池续航时间预测;机器学习

中图法分类号: TP311

中文引用格式: 李豁然,刘譞哲,梅俏竹,梅宏.基于细粒度用户使用数据的智能手机电池续航时间预测模型.软件学报.  
<http://www.jos.org.cn/1000-9825/6199.htm>

英文引用格式: Li HR, Liu XZ, Mei QZ, Mei H. Predicting smartphone battery life by fine-grained usage data. Ruan Jian Xue Bao/Journal of Software, 2020 (in Chinese). <http://www.jos.org.cn/1000-9825/6199.htm>

## Predicting Smartphone Battery Life by Fine-Grained Usage Data

LI Huo-Ran<sup>1,2</sup>, LIU Xuan-Zhe<sup>1,2</sup>, MEI Qiao-Zhu<sup>3</sup>, MEI Hong<sup>1,2</sup>

<sup>1</sup>(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

<sup>2</sup>(Key Laboratory of High Confidence Software Technologies of Ministry of Education (Peking University), Beijing 100871, China)

<sup>3</sup>(School of Information, University of Michigan, Ann Arbor, Michigan 48109, USA)

**Abstract:** Smartphones and smartphone apps have undergone an explosive growth in the past decade. However, smartphone battery technology hasn't been able to keep pace with the rapid growth of the capacity and the functionality of devices and apps. As a result, battery has always been a bottleneck of a user's daily experience of smartphones. An accurate estimation of the remaining battery life could tremendously help the user to schedule their activities and use their smartphones more efficiently. Existing studies on battery life prediction have been primitive due to the lack of real-world smartphone usage data at scale. This paper presents a novel method that uses the state-of-the-art machine learning models for battery life prediction, based on comprehensive and real-time usage traces collected from smartphones. The method is evaluated using a dataset collected from 51 users for 21 months, which covers comprehensive and fine-grained smartphone usage traces including system status, sensor indicators, system events, and app status. We find that the battery life of a smartphone can be accurately predicted based on how the user uses the device at the real-time, in the current session, and in history. As a conclusion, the proposed model could significantly raise the prediction accuracy.

\* 基金项目: 国家杰出青年科学基金(61725201)

Foundation item: The National Science Fund for Distinguished Young Scholars (61725201)

收稿时间: 2019-05-22; 采用时间: 2020-01-02; jos 在线出版时间: 2020-12-02

**Key words:** mobile computing; battery lifetime prediction; machine learning

自从 2007 年初代 iPhone 发布以来,智能手机及智能手机应用(Application)已经得到了长足的发展.今天,智能手机已经成为了人们日常生活中的重要组成部分.如今,市场上每年都会新上市超过百种新型号的智能手机,其均会搭载能力强大的硬件,包括大尺寸的屏幕、高频率的 CPU、高容量的存储等等.在硬件能力得到极大发展的同时,智能手机应用也在蓬勃发展.如今,以苹果 Appstore 和 Google Play 为代表的应用市场上均已经包含了超过 200 万个应用.可以认为,硬件与软件的协同增长使得智能手机的地位日益重要.

但是,一直以来,相比于其它硬件,智能手机电池能力的增长始终较为缓慢.对于绝大多数智能手机,其电池只能支持至多一天的使用.当使用较为频繁时,电池的消耗速度更是会急剧加快.可以认为,较短的电池续航时间会较为严重地影响用户使用智能手机的体验,电池续航能力已经成为了智能手机各项软硬件能力中的短板.因此,用户经常需要随身携带移动电源或是其它便携式充电设备以随时进行充电,或者不得不减低使用智能手机的强度以延长续航时间.由此可见,如何减少电池能力的不足对用户使用带来的不便,是一个非常具有实际价值的研究问题.

目前,现有工作已经对这一问题进行了一定的探索,方法包括降低手机硬件的能耗<sup>[1-4]</sup>、限制不必要的资源分配等<sup>[5-7]</sup>.可以看出,多数已有工作的思路为提高电能的使用效率,以达到延长电池续航时间的目的.但是在实际当中,多数提高电能效率的方法(例如将手机调整至“省电模式”)同时伴随着限制手机功能的风险(例如在“省电模式”下无法访问网络),因而仍然不可避免地会对用户的使用体验造成影响.

为了克服上述问题,另一种思路应运而生:不试图延长电池的续航时间,而是在现有的续航时间限制下,尽可能提高用户的电量相关使用体验.一种可行的方法是在电池电量耗尽、或是消耗至某一给定值前,为用户进行预警.如果用户可以较为准确地预知手机的电池何时将会耗尽,那么用户就可以更好地安排其使用行为.例如,如果用户认为剩余的续航时间已经足够使用,那么用户无需降低其使用强度,而是可以照常使用;反之,如果用户认为剩余的续航时间不足,那么其可以有选择性地降低使用强度,或者优先完成重要任务(例如先发送重要的邮件).由此可见,准确地对电池的续航时间进行预测可以有效地提升用户的使用体验.

由于智能手机的软硬件众多,不同组件对于电池消耗速率的影响也不尽相同,因此预测智能手机电池续航时间是一件较为困难的工作.总体来说,现有的智能手机电池续航时间预测工作比较简单,难以应对复杂软硬件带来的挑战.造成这一现状的主要原因包括:一、现有工作通常基于通过受控实验收集的小规模数据集,数据集覆盖的时间范围较短,因此难以有效代表真实世界中的使用行为.二、现有工作中使用的数据集同样存在信息维度较少、粒度较粗的缺陷,因此难以有效发掘影响电池续航时间的因素.三、现有工作难以处理电池续航时间预测问题中的预测目标缺失问题.具体来说,如果用户在电池耗尽前便开始充电,那么对于该次使用,我们无法准确获知电池耗尽的具体时间,即造成了预测目标缺失.如果不对这一问题进行有效处理,则可能造成预测的结果与真实情况存在偏差.

针对上述挑战,本文基于一个长时间、细粒度的真实用户数据集<sup>[8]</sup>,设计了一个智能手机电池续航时间预测模型.具体来说,本文使用的数据集为一组真实用户的智能手机软硬件数据记录,包括系统状态、系统事件、传感器读数、应用运行记录等信息.数据集包含了三星 Galaxy S5 手机上无需 ROOT 权限即可获取的全部信息,数据维度非常丰富.该数据集涉及 51 名用户,持续时间长达 21 个月.截止至本文完稿时,本文采用的数据集是相关研究当中规模最大、粒度最细的.此外,本文同样首次关注了上述预测目标缺失问题,并且系统性地分析了这一问题对预测效果带来的影响.

本文的主要贡献如下:

1. 本文对长时间、细粒度的真实用户数据集进行了描述性分析,研究了用户在使用智能手机过程中的电量消耗模式.本文发现,电池续航时间与智能手机当前的状态(即发起预测时的状态)、智能手机在过去的使用状

态、以及用户的过往的使用习惯均存在一定的相关性。

2. 基于上述发现,本文提出了一个智能手机电池续航时间预测模型,并基于上述数据集进行了效果验证,分析了各组特征对续航时间预测起到的效果.实验结果显示,本文提出的模型最终可以平均减少 40 分钟的预测误差.当用户给定的预测目标电量为 20%时(该值为多数手机的预警阈值),本文提出模型的预测误差平均仅为 30 分钟.
3. 本文分析了智能手机电池续航时间预测中的预测目标缺失(Censoring)问题,并采用了 Concordance Index 将缺失预测目标的数据纳入到模型评价当中,从而更加准确地评价了预测模型的效果.

本文后续安排如下:第 1 节介绍相关工作.在第 2 节中,本文对智能手机电池续航时间预测问题进行形式化定义.在第 3 节中,本文介绍研究所使用的数据集,并随后在第 4 节和第 5 节分别介绍研究方法概览和实验设计.第 6 节详细分析实验结果.最终,第 7 节对全文进行总结,并展望下一步工作.

## 1 相关工作

如前文所述,电池续航能力是如今智能手机最为重要的能力之一,其会对用户的使用体验带来较大影响.因此,目前已有大量电池优化相关的研究工作.从宏观角度来看,现有工作可以被分为以下三类:能耗建模、能耗优化和续航时间预测.

### 1.1 能耗建模

能耗建模指理解智能手机的电量被哪些组件、以何种方式被消耗掉.目前,已有工作当中主要包含两种能耗建模思路.第一种思路是基于负载强度的能耗建模.其基本假设为,某一硬件的能耗与其运行负载是直接相关的,因此可以根据硬件的负载强度推断其能耗速率.例如,Shye 等人提出了一种基于系统运行状态的能耗模型<sup>[9]</sup>.该工作基于某一时刻智能手机的 CPU 频率、内存占用比等硬件运行信息,通过线性回归模型预测电池在同一时刻的耗电速率,并分析每种硬件与能耗速率之间的相关性.Dong 等人实现了一种能够以更高的频率度量手机能耗的能耗模型<sup>[10]</sup>.该模型仅依赖于手机系统提供的电池信息接口,因此不需要额外的度量手段.Zhang 等人实现了一种能耗模型,将 CPU、LCD 显示屏、GPS、网络通信模块、声音模块等硬件纳入考量,并实现了一种自动化的能耗模型建模方法<sup>[11]</sup>.Min 等人提出,智能手机的能耗速率很大程度上会受到持续传感应用(Continuous Sensing App, CSA)的影响,因为这一类应用会造成比较大的传感器负载.作者通过一系列实验验证了这一假设确实成立<sup>[12]</sup>.

第二种思路是采用有限状态自动机(Finite-State-Machine, FSM)来描述电池的耗电速率状态的变化.这一方法主要适用于网络通信相关模块,因为网络通信模块的能耗速率不仅取决于其负载强度,还受到其它多种复杂因素的影响.因此,利用有限状态自动机比较适合刻画网络通信模块的能耗状态变化.Pathak 等人将手机的系统调用视为触发器,研究了系统调用如何影响智能手机的能耗状态<sup>[13]</sup>.作者实现了一个基于系统调用的有限状态自动机,通过该自动机,可以得知当系统调用产生时,手机的能耗速率将会如何变化.另一工作 WattsOn<sup>[14]</sup>采用了同样的研究思路,用以度量 Wi-Fi 模块的能耗状态变化.

### 1.2 能耗优化

目前,已有能耗优化的核心思路通常为限制不必要的硬件调用或软件运行.具体来说,He 等人实现了 Dynamic Resolution Scaling(DRS),一种可以根据人眼与屏幕的距离自动调整屏幕分辨率的工具<sup>[5]</sup>.其思路在于,如果人眼距离屏幕比较远,则屏幕以高分辨率显示内容是没有意义的,因此可以适当降低屏幕分辨率,从而可以在不影响用户视觉体验的同时达到省电的目的.Chen 等人实现了一个名为 HUSH 的工具<sup>[6]</sup>,可以检测非必要的、在熄屏状态下运行的应用,并将其关闭.实验显示,HUSH 可以降低 15.7%的能耗.Li 等人提出了一种自动重写 Web 应用的方法,可以使得重写后的 Web 应用更加适用于 OLED 屏幕,能够更好地发挥 OLED 屏幕省电的特性<sup>[15]</sup>.

总结来说,能耗优化工作的重点在于如何识别非必要的软硬件运行,以及如何在省电与降低用户使用体验

之间进行权衡.

### 1.3 续航时间预测

如前文所述,续航时间预测作为另一种提高用户使用体验的思路,近年来也得到了学术界的部分关注.Zhao等人提出了一种基于系统运行状态的续航时间预测方法.该方法基于 CPU 运行状态、屏幕显示状态、I/O 组件负载三部分信息,通过一个线性回归模型预测系统的能耗速率,并以此为基础推断电池的剩余续航时间<sup>[16]</sup>.Kang 等人从用户的视角出发,提出了另一种思路.作者将用户的使用行为划分为若干种可能的状态,并假设用户在每种状态下时手机的耗电速率是基本恒定的.基于这一假设,作者实现了一种模型,首先估计用户在每种状态下的使用时长,然后估计每种状态下用户的耗电速率,两者结合即可得到电池续航时间的预估<sup>[17]</sup>.Kim 等人实现了一种针对单个应用的电池续航时间预估方法<sup>[18]</sup>.该工作的目标是让用户了解如果用户一直保持使用某一单一应用(例如一直看视频),其手机的续航时间为多长.具体实现时,其同样主要基于硬件的运行状态与应用执行之间的相关性进行分析.

## 2 问题定义

概括来说,本文希望解决的具体问题可以做如下描述:预测从一个用户指定的、发起查询请求的时间开始,直到手机的电量消耗到一个用户给定的阈值时(下文称该阈值为“**目标电量**”),手机能够持续使用的时长.本节将对上述问题及相关定义做明确的定义.相关符号含义及解释见表 1.

Table 1 Summary of Notations

表 1 符号含义表

| 名称         | 描述                             |
|------------|--------------------------------|
| $t_s$      | 会话开始时间                         |
| $t_e$      | 会话结束时间                         |
| $t_q$      | 查询时间                           |
| $b_n$      | 目标电量                           |
| $t_n$      | 目标时间(电量耗至 $b_n$ 的时间)           |
| $b(t)$     | 时刻 $t$ 的电量                     |
| $\Delta t$ | 续航时间( $\Delta t = t_n - t_q$ ) |

### 2.1 放电会话及电量水平

本文定义一个放电会话(简称为“会话”)为从手机断开充电器进入放电状态,直到手机重新连接上充电器开始充电的时段.换言之,一个会话实际上即为手机处于放电状态下的一段连续时间.在后文当中,本文将以会话作为基本单元进行后续分析.本文用 $[t_s, t_e]$ 表示一个会话,其中 $t_s$ 、 $t_e$ 分别表示会话开始和会话结束的时间.可以知道, $t_e$ 一定大于 $t_s$ .

本文用函数 $b(t)$ 表示在手机在时刻 $t$ 的**电量水平**.因此,一个会话在开始时的电量水平和结束时的电量水平分别可以表示为 $b(t_s)$ 和 $b(t_e)$ .同样可知,由于在会话中电池处于放电状态,所以在会话中 $b(t)$ 是随着 $t$ 的增加单调递减的.

### 2.2 查询,目标电量,续航时间

在一个会话当中,用户可能会随时对手机的续航时间发起一个查询请求.本文称该行为为一次**查询**,用 $t_q$ 表示发起查询的时间.在用户发起查询时,作为常见的查询需求是询问电池耗尽前还能够使用多长时间.但是,在某些情况下,用户可能也会希望了解电池耗至某一阈值前(例如 20%)还能够使用的时长.本文称该电量阈值为**目标电量**,用 $b_n$ 表示.在实际当中,可知 $b_n < b(t_q)$ ,即目标电量总是低于发起查询时的电量.当 $b_n = 0$ 时,可知用户实际上是在查询电池完全耗尽前的续航时间.用户也可以自行指定其他目标电量,例如 5%或 10%.这使得本文提出的模型更加灵活,能够适用于更多的场景.

本文用 $t_n$ 表示电量消耗至 $b_n$ 的时间.显然,有 $t_n > t_q$ .本文将 $\Delta t = t_n - t_q$ (即电量消耗至目标电量时距发起

查询时已经经过的时间)为**续航时间**.需要说明的是,这里定义的续航时间和一般语境下的续航时间略有区别,因为其并非仅表示**电池耗尽前**的时间,而是指**电池消耗至目标电量前**的使用时间.直至电池耗尽的续航时间实际上为本文定义的续航时间中的一个特例,即对应了 $b_n = 0$ 的情况.

上文中定义的查询、目标电量和续航时间全部都是在单个会话内定义的,即用户在某个会话中发起一次查询,查询的目标是在当前会话内电池消耗至目标电量前可以使用的续航时间.需要注意的是,在实际当中, $t_n$ 并不一定存在.例如,假如用户发起请求,希望了解电池电量在消耗至 30%之前的续航时间.但是,当用户回到家中开始给手机充电时,手机仍然有 40%的电量.那么,在这种情况下,电池在当前会话中从未消耗至 30%,因此实际上并不存在对应的 $t_n$ .如果电量消耗至目标电量这一事件实际发生了,那么本文称该数据样本为一个“**可见样本**”,否则称之为一个“**不可见样本**”.

### 2.3 续航时间预测问题定义

基于上述定义的各项概念,续航时间预测问题可以做如下形式化描述:给定一个用户 $u$ ,一个会话 $s$ ,一个发起查询的时间 $t_q$ 以及一个目标电量 $b_n$ ,预测在当前会话内从 $t_q$ 开始直至电池电量消耗至 $b_n$ 时的续航时间 $\Delta t$ .不难看出,这一问题本质上可以看做是一个回归问题,因为待预测的值(续航时间)是一个连续的实数值.因此,本文的目标即为设计一个回归模型,解决上述回归问题.回归模型的输入可以是与用户使用行为或者手机系统状态相关的多种信息,具体将在下文详述.回归模型的输出即为当次查询对应的续航时间.

## 3 数据集

为了建立预测模型,本文基于一个大规模真实用户数据集:Sherlock 数据集展开研究.Sherlock 数据集是由本·古里安大学收集并维护的一个安卓手机用户数据集.本节将会对该数据集进行详细介绍,并介绍如何从原始数据当中抽取会话.最终,本节将会对数据集进行简要的描述性分析,为后续的预测模型提供参考.

### 3.1 Sherlock数据集

Sherlock 数据集是一个数据维度较多、收集时间较长的真实安卓用户数据集.其包含的信息维度为三星 Galaxy S5 上不需要 Root 权限即可获取的全部信息.Sherlock 数据集通过一个基于 Google Funf<sup>[19]</sup> 的数据收集客户端进行收集.数据收集团队招募了一组志愿者,在志愿者知情并同意的情况下,为每一名志愿者配备了一部预装了数据收集客户端的三星 Galaxy S5 手机,要求志愿者以该手机作为日常主力手机,并持续收集用户使用过程中的数据.

总体上来说,Sherlock 数据集包含两大类别的收集方式:基于轮询的收集方式和事件驱动的收集方式.顾名思义,基于轮询的收集方式指以一定的频率,持续通过系统提供的接口读取信息,例如 CPU 读数、内存占用率、传感器读数等.事件驱动的收集方式指通过监听器监听事件,当事件发生时对相关信息进行记录,例如电话呼入、应用变更、系统广播等.总体上来说,Sherlock 数据集共包含 553 个域,持续 21 个月的时间,覆盖 51 名用户.因此,可以认为 Sherlock 数据集是一个细粒度的、长时间的、全面的真实用户数据集.更多细节介绍可以参见 Sherlock 数据集的官网和相关文献<sup>[8]</sup>.

为了保护用户隐私,Sherlock 数据集对用户数据进行了若干处理.首先,用户了解并同意数据收集的过程和数据的用途.其次,所有与用户身份有关的信息(例如 SSID、基站 ID、硬件 MAC 地址、手机号等)均进行了匿名化处理.第三,用户的具体地理位置被隐去.取而代之的,数据集仅提供了一个不含任何语义的编号来表示用户所处的大致范围.通过上述方法,用户的隐私得到了有效的保护.

### 3.2 会话抽取

为了验证 Sherlock 数据集可以用于续航时间预测模型的构建,本文首先尝试从原始数据中抽取出会话.数据集中的电池电量信息是基于轮询方式收集的,收集的频率为大约每 5 秒一次.每条数据包含 4 个维度的信息:1)用户 ID,2)收集数据时的 UNIX 时间戳,3)充放电状态(当前手机处于充电状态或放电状态),4)电量百分比.例如,数据记录(0a50e09262, 1426245782, discharge, 54)表示用户 0a50e09262 的设备在时刻 1426245782 处于放电状态,且电池电量水平为 54%.

为了抽取会话,本文首先舍弃掉了所有处于充电状态下的数据样本.对于一串连续的处于放电状态的、中间不含任何充电状态样本的样本,理论上来说,其可以被认为是一个会话.但是,在实际当中,可能存在以下两种情况使得上述论断不一定成立:

1. 相邻两条样本之间的时间间隔过长.如果用户在一段时间内关闭了手机,或者由于数据收集的故障导致一段时间内的数据没有被有效收集,都会出现相邻两条样本之间的时间间隔过长的现象.在上述情况下,这段较长时间内的电池充放电状态及电量信息是无从得知的.为了保证数据的可靠性,本文从过长的间隔处(本文选取的阈值为 10 分钟)将样本序列分割开来,将其前后分别作为一个会话处理.

2. 在连续的两条记录当中,后一条的电量水平高于前一条.前文已经讨论过,在一个会话当中电量是随着时间单调递减的,因此不可能出现电量升高的情况.唯一可能的原因是用户在两条记录之间的时间内更换了电池.在这种情况下,本文也从上述两条记录中间将样本序列分割,将其前后分别作为一个会话处理.

通过上述方法,本文可以从原始数据中抽取出会话,以供后续分析.为了使得分析更加可靠,本文去除了持续时间过短的会话,仅保留了持续时长至少为 1 小时的会话.最终,本文共从原始数据中抽取得了 37088 个会话.

### 3.3 描述性分析

在生成会话后,我们可以随即获知对应会话的若干信息,例如在该会话内用户的使用行为和系统状态等.当用户发起一个续航时间查询请求时,从会话开始的时间 $t_s$ 直到查询时刻 $t_q$ 这段时间内的上下文信息能够体现用户的使用行为和系统状态,从而很有可能为续航时间预测提供信息.为了检验这一假设是否成立,本小节基于抽取得到的会话及相关上下文信息,对数据进行简单的描述性分析.

用户在发起查询时的电量和目标电量之间的差值越大,续航时间会倾向于更长.因此,预测续航时间不能脱离上述电量差值进行,否则会失去实际意义.为了使得表述更加统一,本文取而代之地对发起查询后的电量消耗速率进行预测:

$$R_1 = \frac{b(t_q) - b(t_e)}{t_e - t_q} = \frac{b(t_q) - b(t_e)}{\Delta t}$$

需要说明的是,在 $b(t_q)$ 和 $b(t_e)$ 给定的情况下, $R_1$ 和 $\Delta t$ 呈现反比关系.因此,预测 $R_1$ 和预测 $\Delta t$ 是等价的.

为了进行描述性分析,本文对全部 37088 个会话进行了采样,从每个会话中随机选取了一个时间点作为发起查询的点( $t_q$ ),分析发起查询前的相关信息与发起查询后的耗电速率的关系.首先,从直觉上来看,电池的电量消耗速率和用户发起查询时剩余电量是有关的.为了验证这一点,本文采用 Bin-Bin 图展示了两者的关系,如图 1(a)所示.可以看出,当发起查询时电量为 30%时,查询后的耗电速率更高.在 30%之前的部分,查询时电量越低,后续耗电速率越低,说明用户会倾向于保守使用;在 30%之后的部分,查询时电量越高,后续耗电速率越低,说明在剩余电量较为充沛时用户的使用强度通常更低.

进一步地,本文希望了解在用户发起查询前后,电量消耗的速度是否相对恒定.如果电池的消耗速率整体来说较为不稳定,则续航时间预测本质上的可预测性相对较低.本文按照与计算 $R_1$ 类似的方式,计算发起查询前的平均耗电速率 $R_0$ :

$$R_0 = \frac{b(t_s) - b(t_q)}{t_q - t_s}$$

图 1(b)用 Bin-Bin 图的方式展示了 $R_0$ 和 $R_1$ 之间的相关性.可以看出,两者呈现出了比较明显的正相关,即在同一会话内,用户发起查询前后的耗电速率是相对一致的.这说明续航时间是可预测的.

接下来,本文探索哪些使用行为信息可能与 $R_1$ 紧密相关,能够对预测续航时间带来帮助.本文首先分析了用户的电池使用习惯于续航时间之间的相关性.本文定义 $\bar{R}$ 为用户在所有会话中的 $R_0$ 的平均值.该值表示了用户本身的电池使用习惯,即用户总体上习惯于重度使用还是轻度使用.图 1(c)展示了 $\bar{R}$ 与 $R_1$ 之间的关联.可以看到,两者之间也呈现出了比较清晰的正相关性,从而比较粗略地说明用户的电池使用习惯对于预测续航时间能够带来帮助.

最终,本文尝试除电量信息本身外,其它上下文信息能否对续航时间预测带来帮助.作为一个例子,本文选取了加速度传感器其中的一个读数 `Accelerometerstat_y_third_idx_fft`,分析其与 $R_1$ 的关联,结果见图 1(d).从结果上看,该读数与 $R_1$ 之间呈现出的正相关比较明显,尤其在读数较小时更加突出.因此可以认为,传感器读数与续航时间之间存在比较显著的关联,可以用于续航时间预测工作.

通过上述描述性分析可以发现,用户在发起查询时系统的状态(例:查询时电量)、当前会话内的使用行为信息(例: $R_0$ 与 $\bar{R}$ )和系统状态信息(例:传感器)都可以对续航时间预测产生帮助.

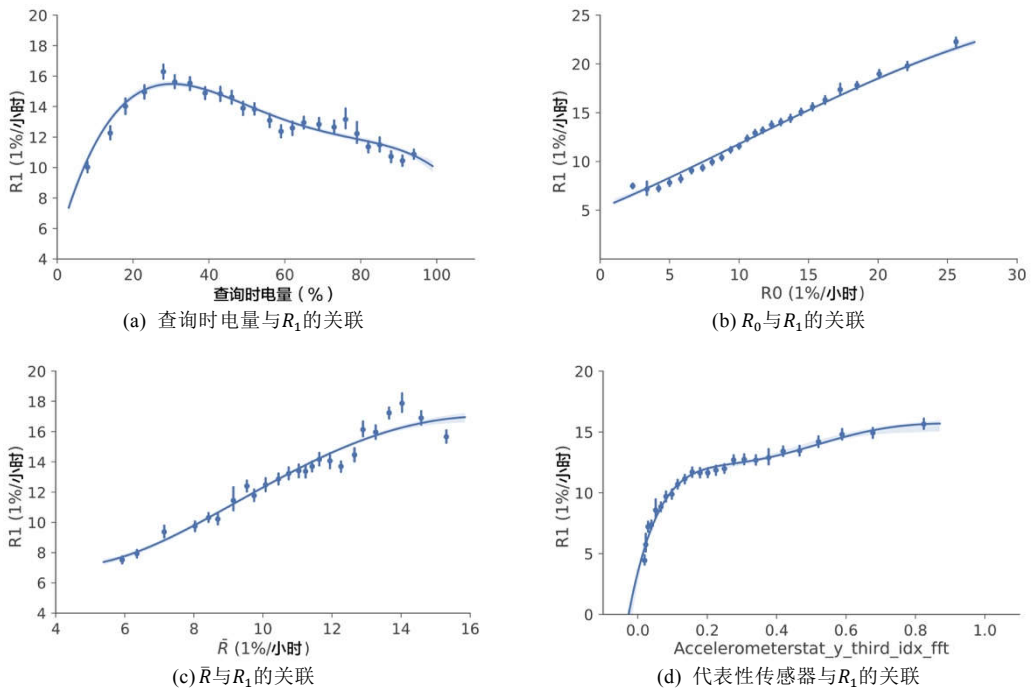


Fig.1 The correlation between  $R_1$  with current battery level at query time,  $R_0$ ,  $\bar{R}$ , and a representative sensor reading, respectively.

图 1  $R_1$ 与查询时电量、 $R_0$ 、 $\bar{R}$ 及代表性传感器的关联

## 4 研究方法概览

### 4.1 方法总体概览

本文提出模型的整体 workflow 概览如图 2 所示.整个过程包含 4 个步骤:

1. **数据收集.**该环节从原始数据中生成会话,并获取每个会话对应的查询时间.同时,该环节从原始数据中提出与会话相对应的各种上下文信息.

2. **续航时间计算与特征提取**. 对于每个会话,该环节首先根据与其对应的查询时间和目标电量计算出实际的电池续航时间,作为回归模型的输出.然后,该环节从该会话对应的上下文信息中生成与当前会话和当次查询对应的各类特征信息(将在下文详述).全体特征信息以高维向量的形式进行表示,构成特征向量,作为回归模型的输入.

3. **模型训练**. 基于上一环节中得到的模型输入(特征向量)和模型输出(续航时间),该环节训练回归模型.本文采取了一系列现有的成熟回归模型,同时覆盖了线性模型和非线性模型.最终,根据训练效果的好坏,该环节将会产生一个预测模型.

4. **预测**. 在得到预测模型后,针对一个新的会话及其对应的查询,该环节生成相关特征,然后利用上一环节训练得到的预测模型对续航时间进行预测.

#### 4.2 机器学习模型

本文选用了一组成熟的、具有代表性的机器学习模型,用以构建续航时间预测模型.具体包含以下 4 种模型:线性回归(Linear Regression)、随机森林回归(Random Forest Regression,RF)<sup>[20]</sup>、梯度提升树回归(Gradient Boosting Regression Tree,GBRT)<sup>[21]</sup> 和 XGBoost (XGB)<sup>[22]</sup>. 上述模型.同时囊括了线性模型和非线性模型,因此可以较好地捕捉特征与续航时间之间的线性关联和非线性关联.

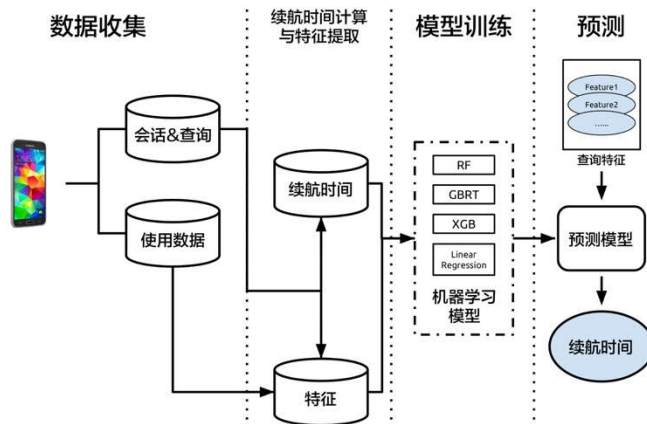


Fig.2 An overview of the prediction model.

图2 整体工作流程概览

#### 4.3 特征

特征是本文提出的预测模型的关键.本文按照以下三个类别划分特征,具体如下:

1. **查询时特征**: 查询时特征表示在用户发起查询的时刻的相关信息所构成的特征.该部分特征中,最容易想到的特征是当前电量与目标电量之间的差值,即  $b(t_q) - b_n$ .显然,该差值越大,电池的续航时间倾向于更长.除去该特征外,在  $t_q$  时刻的其它部分信息也可能与续航时间存在关联.例如,发起查询时的时间可能与续航时间之间存在关联,因为用户在一天当中的不同时刻的使用强度是不尽相同的.
2. **会话特征**: 会话特征表示从当前会话开始( $t_s$ )直至查询时刻( $t_q$ )这段时间范围内的相关信息产生的特征.该部分信息反映了在当前会话内已经产生的使用行为和系统状态,因此与电池在未来的放电速率会有比较明显的关联.例如,如前文所述,电池在查询前的耗电速率和在查询后的耗电速率呈现比较明确的正相关

、本文采用 XGBoost 库<sup>[23]</sup> 实现 XGBoost 模型,使用 Scikit-Learn 库<sup>[24]</sup>实现线性回归、随机森林回归和梯度提升树回归.



性.

3. **历史特征:**查询时特征和会话特征的信息全部局限于当前会话内.但是除此之外,用户在过去(即当前会话之前的会话)的使用习惯仍然有可能帮助续航时间预测.例如,如果某一用户的过去的会话中的耗电速率均较高,则说明该用户可能是一个重度使用用户,因此可以推测用户在当前会话中的使用也会较为重度,耗电速率也会比较高.因此,本文将用户在过去的的信息产生的特征作为第三组特征,称作历史特征.上述介绍了本文对特征的三种基本分类.具体每类特征将会在后文详细介绍.

## 5 实验设计

为了验证本文提出方法的有效性,本文基于 Sherlock 数据集训练续航时间模型,并验证模型的预测效果.本节介绍整个实验的执行流程.

### 5.1 查询生成

如前文所述,我们可以从 Sherlock 数据集的原始数据当中抽取出会话,但是在会话当中实际上并不包含用户的查询行为.因此,在基于该数据集进行实验前,我们需要在会话中模拟生成用户的查询行为.具体做法如下:前文中已经介绍过,本文从原始数据集当中抽取得了 37088 个会话.本文首先去除了会话开始时的电量 ( $b(t_s)$ ) 低于 30% 的会话,因为如果一个会话的起始电量过低,用户的使用行为会受到比较严重的限制,会明显区别于用户的正常使用.这一步骤过滤掉了大约 4% 的会话,剩余 35590 个会话.

对于每个会话,本文假设用户可能在其中的任意时间点发起查询请求.因此,对于每个会话  $[t_s, t_e]$ ,本文在时间区间  $[t_s + 2 \text{ min}, t_e - 2 \text{ min}]$  当中选取一个发起请求的时间点  $t_q$ .上述 2 分钟的边界是为了保证查询前至少已经拥有了一段时间的使用,并且在查询后会话不会立即结束.在  $t_q$  选定后,本文进一步随机选取一个目标电量  $b_n$ .本文选取  $t_q$  和  $b_n$  时额外添加了两个限制:1)  $b(t_q) \geq 10\%$ .这是为了保证查询时剩余电量不至于过低.2)  $b(t_n) \leq b(t_q) - 5\%$ .这是为了保证查询时电量和目标电量具有一定的差距,以使得查询具有实际意义.通过上述方法,每个会话都生成了一个对应的查询,每个会话可以被视为是一个数据样本.

显然,在模拟生成查询时,需要保证  $b(t_n) < b(t_q)$ ,但是本文并不要求  $b(t_n) \geq b(t_e)$ ,因为用户在发起查询时无法预知  $b(t_e)$ ,因此可能会选择一个低于  $b(t_e)$  的目标电量.按照前文定义,如果  $b(t_n) \geq b(t_e)$ ,那么表示当前会话实际消耗到了目标电量,该会话对应的数据样本为**可见样本**.否则,该会话对应的数据为**不可见样本**.

模拟生成查询数据后,在 35590 个数据样本中,14773 个为可见样本,其余 20817 个为不可见样本.本文进一步从两者当中各随机选取出 5/6 作为训练数据,其余 1/6 作为测试数据,用于后续模型训练和模型测试环节.数据统计信息见表 2.

Table 2 Summary statistics of the data

表 2 数据样本统计信息

|     | 可见样本  | 不可见样本 | 汇总    |
|-----|-------|-------|-------|
| 训练集 | 17382 | 12275 | 29657 |
| 测试集 | 3435  | 2498  | 5933  |
| 汇总  | 14733 | 20817 | 35590 |

### 5.2 特征提取

由于 Sherlock 数据集中包含的数据维度众多,为了使得模型更加有效,本文首先根据人工经验对数据进行粗粒度的筛选,从中选出了最可能与电池续航时间相关的部分信息:应用使用行为、系统运行状态、传感器数据、屏幕显示状态数据和广播数据.详细描述如下:

1. **应用使用数据.**在 Sherlock 数据集中,应用使用数据通过每 5 秒读取一次系统进程列表的方式被记录.从每次读取的记录当中,可以了解到当前哪些应用正在运行.由于在 Sherlock 数据集中一共出现过 655 个不同的应用,因此本文用一个 655 维的 0-1 向量表示某一时刻的应用运行状态.每个维度对应一个应用,该维度取值为 1

表示这一时刻该应用正在运行,取值为 0 则表示该应用未在运行。

2. **系统运行状态和传感器数据**.根据 Sherlock 数据集的分类法,系统运行状态和传感器数据被分为 4 组:基本设备状态(T1)、导航类与位置类传感器(T2)、声音类与亮度类(T3)、应用统计类(T4).通过人工筛选的方式,本文从 T1 和 T2 中各自挑选了 9 个和 150 个可能与续航时间高度相关的维度,作为该部分信息的特征。

3. **屏幕显示状态和广播数据**.屏幕显示状态和广播是通过基于事件的收集方式进行收集的.每次用户点亮或者熄灭屏幕时,会产生一条屏幕显示状态变化的记录.每次当系统发送一条广播时,该广播及其相关信息也会被记录下来。

基于以上信息,本文最终整理得到了 22 组不同的特征,详细描述见表 3(带\*的项表示采用独热编码进行表示).其中,F0 表示查询时电量与目标电量之间的差值.该信息是最重要的一组特征,是续航时间预测的基础.F1-F4 为查询时特征,F5-F18 为会话特征,F19-F21 为历史特征.为了保证不同特征处于相同的维度,本文对全部特征进行了标准正则化,即通过线性变换将特征转换为均值为 0、标准差为 1 的形式。

Table 3 Features used in our model

表 3 特征详情

| 组别   | 名称              | 特征数 | 描述  |
|------|-----------------|-----|---|
| F0   | battery_gap     | 1   | 查询时电量与目标电量之间的差值.                            |
| F1   | current_battery | 1   | 查询时电量.                                      |
| F2*  | current_hour    | 24  | 查询时的小时数(即处于一天当中的第几个小时).                     |
| F3*  | current_weekday | 7   | 查询时的星期日(即处于一周当中的第几天).                       |
| F4   | sensor_T2_last  | 150 | 150 个 T2 传感器在发起查询前的最后一次读数.                  |
| F5   | start_battery   | 1   | 会话开始时的电量.                                   |
| F6*  | start_hour      | 24  | 会话开始时的小时数(类似 F2).                           |
| F7*  | start_weekday   | 7   | 会话开始时的星期日(类似 F3).                           |
| F8   | age             | 1   | 会话从开始直至查询时经过的时间.                            |
| F9   | consumption     | 1   | 会话从开始直至查询时已经消耗的电量.                          |
| F10  | history_rate    | 1   | 会话从开始直至查询时的平均耗电速率(F10=F9/F8).               |
| F11  | naive_surv      | 1   | 假设会话始终保持 F10 的消耗速率得到的续航时间(F11=F0/F10).      |
| F12  | past_rate       | 11  | 细粒度历史耗电速率, 详见正文.                            |
| F13  | sensor_T1       | 45  | 发起查询前 1/5/10/30/60 分钟内的 T1 传感器均值.           |
| F14  | sensor_T2_5min  | 150 | 150 个 T2 传感器在发起查询前的 5 分钟内的读数均值.             |
| F15  | app_occurrence  | 400 | 发起查询前 5/10/30/60 分钟内, 50 个热门应用的使用情况(前台&后台). |
| F16  | app_usage       | 100 | 当前会话内 50 个热门应用的使用时间占比(前台&后台).               |
| F17  | screen          | 2   | 当前会话内屏幕电量次数&亮屏时间占比.                         |
| F18  | broadcast       | 86  | 当前会话内收到的广播次数.                               |
| F19* | user_index      | 51  | 用户 ID(0-50).                                |
| F20  | session_history | 8   | 历史会话中的电量消耗速率, 详见正文.                         |
| F21  | screen_history  | 8   | 历史会话中的屏幕使用记录, 详见正文.                         |

### 5.3 评价指标

为了评价续航时间预测模型的准确性,本文采取了以下三种评价指标:均方根误差(Root Mean Square Error, RMSE),Kendall's Tau 和 Concordance Index(C-Idx).三种评价指标的具体含义及选择上述指标的理由如下:

#### 5.3.1 均方根误差

最直观的衡量预测续航时间与实际续航时间的差异的方法是计算两种之间的差值.预测续航时间与实际续航时间之间的差距越小,则说明结果越准确.为了实现这一目标,本文采用均方根误差度量对上述差值进行度量.均方根误差最大的优点是具有实际意义,比较直观,易于理解.例如,如果均方根误差为 30 分钟,则说明预测续航时间与实际续航时间之间的平均差距为 30 分钟。

但是,针对于续航时间预测,均方根误差也存在局限性.首先,均方根误差对异常点比较敏感<sup>[25]</sup>.如果数据中存在个别数值极大或者极小的异常点,则均方根误差的值会受到很大的影响.其次,均方根误差对于估计过度 and 估计不足的惩罚力度是不同的<sup>[26]</sup>.例如,如果对于一次查询,实际的续航时间是 30 分钟.那么,如果预测的续航时间少于 30 分钟,均方根误差至多为 30 分钟;如果预测的续航时间多于 30 分钟,均方根误差则没有上限。

这会使得均方根误差会放大估计过度的影响,而忽视估计不足的影响。

因此,为了解决这一问题,本文进一步采用了以下两种评价指标。

### 5.3.2 Kendall's Tau

Kendall's Tau 是一种被广泛使用的排序型指标。其基本思想为,将所有查询(样本)按照其实际续航时间和预测续航时间分别进行排序,然后比较两个排序的一致程度。如果两个排序完全一致,则说明预测续航时间非常可靠;如果排序完全不一致,则说明预测续航时间完全错误。排序型指标可以较好地克服均方根误差的缺点(对噪声敏感、评价不均衡),可以作为评价指标的有效补充。

均方根误差和 Kendall's Tau 可以从不同的角度(数值角度和排序角度)衡量预测的准确性,但是两者都不能解决前文提出的不可见样本问题。换言之,两者在进行评价时均需要全体样本均具有明确的标签,对于标签不可见的样本则无法纳入评价当中。处理这一问题的最简单的方法即为舍弃不可见样本,仅针对可见样本进行评价。但是由于可见样本与不可见样本之间的分布可能存在差异,仅在可见样本上进行评价得到的结果可能与在全集上进行评价的结果具有较大偏差。因此,需要考虑如何克服这一问题。

### 5.3.3 Concordance Index

Concordance Index 是在生存分析领域常用的一种评价指标。生存分析来自于医学领域,其基本目标是研究病患的生存时间。对于一组病患,实验通常无法等待所有参与者均死亡后才结束,因此在实验结束时可能存在部分病患仍然存活。这将导致该部分病患的实际生存时间不可知。由以上简介可知,生存分析中的生存时间不可知与续航时间预测中的样本标签不可见非常类似,因此生存分析中的评价指标也可以引入至续航时间预测问题。

本文引入生存分析领域的 Concordance Index 作为评价指标。其类似于 Kendall's Tau,对于两个样本 A 和 B,其比较 A 与 B 的实际续航时间的顺序与预测续航时间的顺序是否一致。如果一致,则 A 和 B 构成一个正序对,否则构成一个逆序对。与 Kendall's Tau 不同的是,Concordance Index 能够对一些不可见样本进行处理。例如,假设会话 A 的实际续航时间为 3 小时,会话 B 的实际续航时间不可见,但是可知 B 会话结束时的使用时间已经为 5 小时。那么,可以明确地知道,会话 B 的续航时间一定长于会话 A,因此即使 B 是一个不可见样本,我们也仍然可以准确地对 A 和 B 的续航时间顺序进行比较。Concordance Index 基于以上思路,将部分不可见样本纳入到评价过程中。

综上所述,本文共采取均方根误差、Kendall's Tau 和 Concordance Index 三种评价指标对预测效果进行评价。其中,均方根误差和 Kendall's Tau 仅用于可见样本,Concordance Index 则用于全体样本。

## 6 实验结果

### 6.1 实验基准线

在分析结果之前,首先需要选定一个适当的实验基准线,作为后续实验结果的参考。根据日常生活经验,显然可知,当查询时电量与目标电量之间的差距越大时,续航时间会倾向于更长。如果假设电池的耗电速率始终恒定(显然该假设是不切实际的,仅能够作为最简单、最理想化的假设),那么电池的续航时间与查询时电量和目标电量差呈完全正比的关系。因此,本文首先仅采用查询时电量与目标电量的差(F0)作为唯一的特征训练模型,并以该模型的效果作为实验基准线。

相关结果见表 4 的前三行。上述简单模型在可见样本上的均方根误差为 149.9 分钟, Kendall's Tau 值为 0.5856。在全体样本上的 Concordance Index 的值为 0.867 左右。由于仅有一个特征,四种模型得到的效果基本是一致的。可以看出,仅采用单一特征的模型的均方根误差是不够令人满意的。将近 150 分钟的误差使得该模型的实际应用价值较差。不过,该模型得到的 Kendall's Tau 和 Concordance Index 尚可,均好于随机猜测的数值(分别为

---

– [https://en.wikipedia.org/wiki/Kendall\\_rank\\_correlation\\_coefficient](https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient)

– [https://en.wikipedia.org/wiki/Survival\\_analysis](https://en.wikipedia.org/wiki/Survival_analysis)

0 和 0.5).

均方根误差和排序指标体现出的差异说明:查询时电量与目标电量的差(F0)是和续航时间之间的正相关性比较明显,因此才能够仅依靠这一个指标得到较好的排序指标结果.但是,仅依靠单一指标还比较粗略,无法得到较为精确的具体续航时间数值.因此,还需要借助其他指标,进一步精细化续航时间预测.

## 6.2 查询时特征的效果

接上一小节思路,本小节首先研究利用查询时特征预测续航时间的效果.相关特征共包含 4 组:F1-F4.本文首先单独基于每组特征训练预测模型,以观察每组模型单独使用时的效果.结果显示,单独利用 F1-F4 预测续航时间的效果要远差于利用 F0 进行预测.这验证了 F0 的确是最重要的预测指标,其它的指标仅能够起到辅助作用.

因此,本文分别将 F1-F4 与 F0 进行组合,预测续航时间(即利用 F0+F1 预测,F0+F2 预测,以此类推),然后将 F0-F4 进行整体组合预测续航时间.结果见表 4.4 组特征当中,效果最好的是 F1,能够将 Kendall's Tau 从 0.5856 提高至 0.6230,将均方根误差从 149.9 降低至 140.1.当同时采用 4 组特征时,均方根误差可以被降低 18.5 分钟,而 Kendall's Tau 和 Concordance Index 分别可以提升 0.062 和 0.025.

Table 4 Performance of query-time features

表 4 实验结果:查询时特征

| 特征     | 评价指标              | 模型     |        |        |        | 最佳结果   |
|--------|-------------------|--------|--------|--------|--------|--------|
|        |                   | Linear | GBRT   | RF     | XGB    |        |
| F0     | 均方根误差             | 149.9  | 150.6  | 150.9  | 151.0  | 149.9  |
|        | Kendall's Tau     | 0.5843 | 0.5856 | 0.5821 | 0.5855 | 0.5856 |
|        | Concordance Index | 0.8666 | 0.8659 | 0.8652 | 0.8660 | 0.8666 |
| F0, F1 | 均方根误差             | 144.8  | 140.1  | 151.4  | 140.7  | 140.1  |
|        | Kendall's Tau     | 0.5918 | 0.6230 | 0.5684 | 0.6228 | 0.6230 |
|        | Concordance Index | 0.8701 | 0.8838 | 0.8622 | 0.8838 | 0.8838 |
| F0, F2 | 均方根误差             | 146.5  | 146.7  | 155.9  | 147.0  | 146.5  |
|        | Kendall's Tau     | 0.5954 | 0.6013 | 0.5653 | 0.6017 | 0.6017 |
|        | Concordance Index | 0.8732 | 0.8742 | 0.8559 | 0.8745 | 0.8745 |
| F0, F3 | 均方根误差             | 148.8  | 148.4  | 150.6  | 148.8  | 148.4  |
|        | Kendall's Tau     | 0.5683 | 0.5733 | 0.5618 | 0.5733 | 0.5733 |
|        | Concordance Index | 0.8646 | 0.8655 | 0.8592 | 0.8656 | 0.8656 |
| F0, F4 | 均方根误差             | 148.5  | 144.1  | 150.6  | 143.7  | 143.7  |
|        | Kendall's Tau     | 0.5793 | 0.6015 | 0.5440 | 0.6037 | 0.6037 |
|        | Concordance Index | 0.8670 | 0.8747 | 0.8491 | 0.8755 | 0.8755 |
| F0-F4  | 均方根误差             | 141.2  | 132.7  | 138.2  | 131.4  | 131.4  |
|        | Kendall's Tau     | 0.6020 | 0.6449 | 0.6180 | 0.6473 | 0.6473 |
|        | Concordance Index | 0.8738 | 0.8919 | 0.8782 | 0.8921 | 0.8921 |

## 6.3 会话特征的效果

查询时特征的效果虽然已经好于基准线,但是仍然不能令人满意.查询时特征仅包含用户发起查询时的信息,覆盖到的时间较短,信息量较少.如前文所言,用户在当前会话内发起查询前的使用行为与续航时间存在关联.因此,本小节进一步分析会话特征对续航时间预测的效果.

会话特征共包含 14 组,见表 3 中的 F5-F18.具体来说,其中包含了会话开始时的相关信息(F5-F7),使用时间与电量消耗信息(F8-F12),应用使用信息和传感器读数信息(F13-F16),以及系统事件信息(F17-F18).和上一小节类似,此处首先使用每组特征单独与 F0 进行结合,观察其是否能够提高预测效果.相关结果见表 4.

相比于查询时特征,会话特征对预测效果的提升更为明显.当采用 F10 或 F11 时,使用 GBRT 和 XGB 的提升效果非常显著.F10 是在当前会话内、查询前的平均耗电速率,F11 则是基于 F10 进行的朴素预测,即假设查询后电池的耗电速率仍然保持 F10,基于此假设预测的续航时间.换言之,F11=F1/F10.尽管上述两个特征比较简单,但是其能够产生的效果是比较明显的.最好的均方根误差被降低了 20.8 分钟,最好的 Kendall's Tau 能够提升 0.12.

如果进一步精细化与耗电速率相关的特征,则可以得到更好的预测效果.从表 4 中可以看出,效果最佳的会

话特征是 F12.F12 共包含 10 个值,分别表示从 $b(t_q) + 1\%$ 、 $b(t_q) + 2\%$ 、 $\dots$ 、 $b(t_q) + 10\%$ 消耗至 $b(t_q)$ 的平均耗电速率.距离来说,假设发起查询时的电量为 50%,那么 F12 包含的值为从 51%消耗至 50%、从 52%消耗至 50%、 $\dots$ 、60%消耗至 50%这 10 个时间段的内的平均耗电速率.F12 可以表示出查询前设备的耗电趋势.例如,如果从 60%耗至 50%的平均耗电速率较低,但是从 51%耗至 50%的平均耗电速率较高,则说明查询前的耗电速率呈现加速的趋势.结果显示,F12 可以得到目前为止单组特征的最好效果.均方根误差降至 121.4 分钟,Kendall's Tau 和 Concordance Index 分别提升至 0.7392 和 0.9224.可见,细粒度的耗电速率信息对于预测效果的提升是十分显著的.

Table 5 Performance of session features

表 5 实验结果:会话特征

| 特征          | 评价指标              | 模型     |        |        |        | 最佳结果   |
|-------------|-------------------|--------|--------|--------|--------|--------|
|             |                   | Linear | GBRT   | RF     | XGB    |        |
| F0, F5      | 均方根误差             | 147.7  | 147.2  | 151.3  | 147.3  | 147.2  |
|             | Kendall's Tau     | 0.5890 | 0.6001 | 0.5590 | 0.6002 | 0.6002 |
|             | Concordance Index | 0.8687 | 0.8706 | 0.8515 | 0.8707 | 0.8707 |
| F0, F6      | 均方根误差             | 146.7  | 147.6  | 158.8  | 147.9  | 146.7  |
|             | Kendall's Tau     | 0.5979 | 0.6031 | 0.5663 | 0.6036 | 0.6036 |
|             | Concordance Index | 0.8723 | 0.8739 | 0.8543 | 0.8739 | 0.8739 |
| F0, F7      | 均方根误差             | 148.4  | 148.6  | 153.6  | 150.1  | 148.4  |
|             | Kendall's Tau     | 0.5696 | 0.5747 | 0.5601 | 0.5743 | 0.5747 |
|             | Concordance Index | 0.8647 | 0.8656 | 0.8586 | 0.8656 | 0.8656 |
| F0, F8      | 均方根误差             | 144.9  | 143.5  | 158.3  | 144.9  | 143.5  |
|             | Kendall's Tau     | 0.6088 | 0.6132 | 0.5280 | 0.6134 | 0.6134 |
|             | Concordance Index | 0.8746 | 0.8760 | 0.8382 | 0.8761 | 0.8761 |
| F0, F9      | 均方根误差             | 149.0  | 142.7  | 152.5  | 142.9  | 142.7  |
|             | Kendall's Tau     | 0.5690 | 0.6031 | 0.5534 | 0.6031 | 0.6031 |
|             | Concordance Index | 0.8656 | 0.8790 | 0.8570 | 0.8791 | 0.8791 |
| F0, F10     | 均方根误差             | 140.4  | 128.9  | 139.7  | 129.1  | 128.9  |
|             | Kendall's Tau     | 0.6110 | 0.7055 | 0.6576 | 0.7056 | 0.7056 |
|             | Concordance Index | 0.8797 | 0.9104 | 0.8912 | 0.9105 | 0.9105 |
| F0, F11     | 均方根误差             | 140.5  | 129.7  | 141.6  | 129.9  | 129.7  |
|             | Kendall's Tau     | 0.6091 | 0.6914 | 0.6409 | 0.6919 | 0.6919 |
|             | Concordance Index | 0.8790 | 0.9036 | 0.8831 | 0.9038 | 0.9038 |
| F0, F12     | 均方根误差             | 143.7  | 121.4  | 131.3  | 121.8  | 121.4  |
|             | Kendall's Tau     | 0.6411 | 0.7392 | 0.7163 | 0.7383 | 0.7392 |
|             | Concordance Index | 0.8875 | 0.9224 | 0.9140 | 0.9222 | 0.9224 |
| F0, F13     | 均方根误差             | 148.7  | 145.0  | 152.1  | 146.4  | 145.0  |
|             | Kendall's Tau     | 0.5805 | 0.6009 | 0.5327 | 0.5996 | 0.6009 |
|             | Concordance Index | 0.8681 | 0.8756 | 0.8491 | 0.8749 | 0.8756 |
| F0, F14     | 均方根误差             | 147.0  | 141.4  | 149.6  | 141.5  | 141.4  |
|             | Kendall's Tau     | 0.5883 | 0.6087 | 0.5519 | 0.6091 | 0.6091 |
|             | Concordance Index | 0.8702 | 0.8765 | 0.8527 | 0.8771 | 0.8771 |
| F0, F15     | 均方根误差             | 144.7  | 143.7  | 149.7  | 144.8  | 143.7  |
|             | Kendall's Tau     | 0.5955 | 0.6076 | 0.5668 | 0.6063 | 0.6076 |
|             | Concordance Index | 0.8719 | 0.8762 | 0.8584 | 0.8761 | 0.8762 |
| F0, F16     | 均方根误差             | 145.7  | 141.9  | 150.3  | 141.4  | 141.4  |
|             | Kendall's Tau     | 0.6024 | 0.6144 | 0.5772 | 0.6160 | 0.6160 |
|             | Concordance Index | 0.8743 | 0.8792 | 0.8631 | 0.8797 | 0.8797 |
| F0, F17     | 均方根误差             | 148.4  | 146.6  | 152.3  | 146.5  | 146.5  |
|             | Kendall's Tau     | 0.5928 | 0.5932 | 0.5406 | 0.5926 | 0.5932 |
|             | Concordance Index | 0.8718 | 0.8737 | 0.8514 | 0.8737 | 0.8737 |
| F0, F18     | 均方根误差             | 149.4  | 143.3  | 147.0  | 143.8  | 143.3  |
|             | Kendall's Tau     | 0.6012 | 0.6071 | 0.5757 | 0.6093 | 0.6093 |
|             | Concordance Index | 0.8730 | 0.8763 | 0.8613 | 0.8770 | 0.8770 |
| F0, F10-F12 | 均方根误差             | 140.4  | 121.8  | 127.3  | 121.5  | 121.5  |
|             | Kendall's Tau     | 0.6526 | 0.7398 | 0.7296 | 0.7393 | 0.7398 |
|             | Concordance Index | 0.8920 | 0.9219 | 0.9172 | 0.9219 | 0.9219 |
| F0, F5-F18  | 均方根误差             | 131.1  | 114.8  | 121.4  | 115.0  | 114.8  |
|             | Kendall's Tau     | 0.6657 | 0.7368 | 0.7274 | 0.7407 | 0.7407 |
|             | Concordance Index | 0.8951 | 0.9213 | 0.9177 | 0.9224 | 0.9224 |

在尝试单组特征的效果后,本文进一步将全部会话特征作为输入,观察预测效果,结果见表 5.本文首先将最有效的三组特征(F10-F12)组合,发现效果与仅使用 F12 差别不大.这说明,F12 已经足够表示电池的耗电特征,F10 和 F11 能够进一步提供的帮助比较有限.然后,本文尝试了将 F5-F18 整体使用.相比于 F10-F12,均方根误差可以被进一步小幅下降.这说明其它会话特征虽然在单独使用时效果不够理想,但是在整体使用时仍然能够产生一定帮助.

#### 6.4 历史特征

会话特征体现了用户短期内的使用习惯和使用趋势,因此能够对续航时间预测带来帮助.由此可以想到,用户的长期使用习惯也应当可以对预测带来帮助.因此,本文接下来分析使用历史特征进行续航时间预测的效果.

本文采取以下三组历史特征:

1. **用户 ID(F19)**.用户 ID 表示了当前会话属于哪个用户.由于数据集当中一共有 51 名用户,因此该部分特征采用一个 51 维的独热向量进行表示.
2. **会话历史(F20)**.该部分内容包含了用户在历史会话(当前会话之前的所有会话)当中的耗电速率统计信息.共包含四类信息:所有历史会话的耗电速率、和当前会话开始于同一小时的历史会话的耗电速率、和当前会话开始于一周中同一天的历史会话的耗电速率、和当前会话开始于一周中同一天且同一小时的历史会话的耗电速率.每种信息同时考虑了平均值和中位数,所以 F20 共包含  $4*2=8$  个值.
3. **屏幕状态历史(F21)**.F21 表示了用户在历史会话中的屏幕状态变化信息,包括屏幕点亮的时间占比和屏幕点亮的次数.具体计算方法类似于 F20.

上述特征的效果见表 6.F19 和 F21 并不能有效提升预测效果.F20 能够对预测产生帮助,但是效果比 F10-F12 较差.这说明,历史特征作为长期使用习惯,能够产生一定的预测效果,但是没有短期使用习惯的效果明显.

**Table 6** Performance of user history features

表 6 实验结果:历史特征

| 特征          | 评价指标              | 模型     |        |        |        | 最佳结果   |
|-------------|-------------------|--------|--------|--------|--------|--------|
|             |                   | Linear | GBRT   | RF     | XGB    |        |
| F0, F19     | 均方根误差             | 144.8  | 144.9  | 156.3  | 145.2  | 144.8  |
|             | Kendall's Tau     | 0.5991 | 0.6087 | 0.5436 | 0.6080 | 0.6087 |
|             | Concordance Index | 0.8747 | 0.8769 | 0.8468 | 0.8766 | 0.8769 |
| F0, F20     | 均方根误差             | 141.2  | 138.2  | 145.5  | 138.6  | 138.2  |
|             | Kendall's Tau     | 0.6227 | 0.6400 | 0.5953 | 0.6385 | 0.6400 |
|             | Concordance Index | 0.8777 | 0.8845 | 0.8662 | 0.8841 | 0.8845 |
| F0, F21     | 均方根误差             | 146.1  | 142.4  | 149.6  | 141.4  | 141.4  |
|             | Kendall's Tau     | 0.5995 | 0.6092 | 0.5633 | 0.6112 | 0.6112 |
|             | Concordance Index | 0.8742 | 0.8778 | 0.8571 | 0.8787 | 0.8787 |
| F0, F19-F21 | 均方根误差             | 137.6  | 134.0  | 135.1  | 134.3  | 134.0  |
|             | Kendall's Tau     | 0.6363 | 0.6550 | 0.6272 | 0.6550 | 0.6550 |
|             | Concordance Index | 0.8861 | 0.8911 | 0.8778 | 0.8911 | 0.8911 |

#### 6.5 综合使用所有特征

最终,本文分析同时采用全部特征的预测效果.结果如图 7 所示.整体结论和前文结果类似:查询时特征能够带来的帮助不大;当引入会话特征时,预测效果的提升比较明显;历史特征则可以进一步带来小幅提升.最终,模型将均方根误差从 149.9 降低至 110.4,将 Kendall's Tau 从 0.5856 提升至 0.7486,将 Concordance Index 从 0.8666 提升至 0.9254.

#### 6.6 讨论

为了验证本文的实验结果在统计意义上的显著性,本文首先采用 Bootstrap test and the shifted method 方法<sup>[27]</sup>对实验结果进行了显著性检验.检验结果见表 8(\*对应  $p\text{-value} \leq 0.1$ ,\*\*对应  $p\text{-value} \leq 0.05$ ,\*\*\*对应

p-value $\leq 0.01$ ).从表中可以看出,查询时特征对均方根误差的提升的显著性不足,但是在另外两种评价指标下的提升是统计显著的.这说明均方根误差的确存在其局限性,因此排序型评价指标的引入是有价值的.会话特征和历史特征在所有评价指标下的提升则都是显著的.

**Table 7** Performance if we put features together

表 7 实验结果:综合使用所有特征

| 特征     | 评价指标              | 模型     |        |        |        | 最佳结果   |
|--------|-------------------|--------|--------|--------|--------|--------|
|        |                   | Linear | GBRT   | RF     | XGB    |        |
| F0-F4  | 均方根误差             | 141.2  | 132.7  | 138.2  | 131.4  | 131.4  |
|        | Kendall's Tau     | 0.6020 | 0.6449 | 0.6180 | 0.6473 | 0.6473 |
|        | Concordance Index | 0.8738 | 0.8919 | 0.8782 | 0.8921 | 0.8921 |
| F0-F18 | 均方根误差             | 128.9  | 115.0  | 118.5  | 114.1  | 114.1  |
|        | Kendall's Tau     | 0.6665 | 0.7409 | 0.7395 | 0.7430 | 0.7430 |
|        | Concordance Index | 0.8954 | 0.9229 | 0.9209 | 0.9235 | 0.9235 |
| F0-F21 | 均方根误差             | 124.7  | 110.6  | 120.3  | 110.4  | 110.4  |
|        | Kendall's Tau     | 0.6745 | 0.7486 | 0.7450 | 0.7478 | 0.7486 |
|        | Concordance Index | 0.9000 | 0.9251 | 0.9227 | 0.9254 | 0.9254 |

**Table 8** Statistic tests of experiment results

表 8 实验结果显著性检验

| 特征          | 最佳<br>均方根误差 | 最佳<br>Kendall's Tau | 最佳<br>Concordance Index |
|-------------|-------------|---------------------|-------------------------|
| F0-F4       | 131.4       | 0.6473***           | 0.8921***               |
| F0, F5-F18  | 114.8**     | 0.7407***           | 0.9224***               |
| F0, F19-F21 | 134.0       | 0.6550***           | 0.8911***               |
| F9-F21      | 110.4**     | 0.7486***           | 0.9254***               |

除了统计显著性之外,本文同时关心模型带来的效果提升是否具有足够的实际价值.在这里,本文从最易于直观理解的均方根误差入手,即希望回答如下问题:均方根误差从 150 分钟降低至 110 分钟是否具有实际价值?从数值上看,降低 40 分钟均方根误差是非常显著的提升.用户可以利用该 40 分钟时间更加合理地安排使用.但是降低后误差仍有 110 分钟,即接近两小时的时间,误差仍然较大.经过分析,本文认为这是由于本文随机生成的查询行为与用户的真实查询行为之间存在偏差导致的.本文随机生成查询时,会随机选取查询时间和目标电量.如果查询时电量和目标电量之间的差距较大,则实际的续航时间会更长,因此预测误差相对较大是正常的.但是在实际场景下,用户会倾向于在电量接近用尽时预测续航时间,查询时电量与目标电量的差距也会相对较小.在这样的场景下,预测误差应当会相应降低.为了验证这一点,本文通过图 3(a)展示查询时电量与目标电量的差( $b(t_q) - b_n$ )与均方根误差的关联,并通过图 3(b)展示目标电量( $b_n$ )与均方根误差的关联.从图中可以看出,当查询时电量与目标电量较为接近时,预测误差实际上是比较小的.因此在实际场景下,用户更易于得到一个远低于 110 分钟的预测误差.当目标电量为 20%时(手机的大致电量报警阈值),均方根误差仅为 30 分钟.这说明本文提出的模型具有较好的实际应用价值.

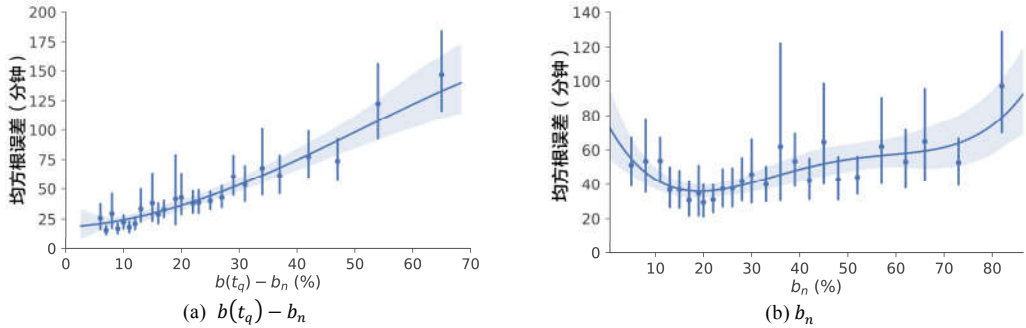


Fig.3 The correlation between RMSE with  $b(t_q) - b_n$  and  $b_n$ , respectively.

图3 均方根误差与 $b(t_q) - b_n$ 和 $b_n$ 的关联

## 7 总结

本文基于现有机器学习模型和一个长时间、细粒度的真实用户数据集,提出了一种智能手机电池续航时间预测模型.此外,本文首次引入了生存分析领域中的 Concordance Index 作为评价指标,从而解决了不可见样本对评价结果造成的偏差.

实验证明,本文提出的模型能够有效应用于实际场景,显著提升电池续航时间预测的效果.数据显示,查询时特征、会话特征和历史特征能够从不同程度上帮助续航时间预测.其中,查询时特征的效果相对最弱;会话特征作为用户的短期使用行为数据,效果最好,尤其与电池耗电速度直接相关的特征则最为有效;历史特征作为长期用户行为数据,也能够发挥一定的效果.最终,本文提出的模型可以将预测准确度提升 40 分钟.通过讨论,本文论证了模型得到的效果在实际场景下具有足够意义.

下一步,本文计划从两方面进行进一步探索.其一,本文希望能够尝试更多维度的特征,例如更多的传感器读数和更加细粒度的应用使用行为数据.其二,本文希望进一步尝试更加前沿的机器学习模型,尤其是希望能够设计一种生存分析模型,将不可见样本纳入到训练过程当中,以进一步提升预测的效果.

## References:

- [1] Balasubramanian N, Balasubramanian A, Venkataramani A. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement. ACM, 2009: 280-293.
- [2] Puustinen I, Nurminen J K. The Effect of Unwanted Internet Traffic on Cellular Phone Energy Consumption. 2011 4th IFIP International Conference on New Technologies, Mobility and Security. IEEE, 2011: 1-5.
- [3] Rosen S, Nikravesh A, Guo Y, et al. Revisiting Network Energy Efficiency of Mobile Apps: Performance in the Wild. Proceedings of the 2015 Internet Measurement Conference. ACM, 2015: 339-345.
- [4] Shen H, Balasubramanian A, LaMarca A, et al. Enhancing Mobile Apps to Use Sensor Hubs Without Programmer Effort. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015: 227-238.
- [5] He S, Liu Y, Zhou H. Optimizing Smartphone Power Consumption Through Dynamic Resolution Scaling. Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. ACM, 2015: 27-39.
- [6] Chen X, Jindal A, Ding N, et al. Smartphone Background Activities in the Wild: Origin, Energy Drain, and Optimization. Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. ACM, 2015: 40-52.
- [7] Draa I C, Tayeb J, Niar S, et al. Application Sequence Prediction for Energy Consumption Reduction in Mobile Systems. 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE, 2015: 23-30.



- [8] Mirsky Y, Shabtai A, Rokach L, et al. Sherlock vs Moriarty: A Smartphone Dataset for Cybersecurity Research. Proceedings of the 2016 ACM workshop on Artificial intelligence and security. ACM, 2016: 1-12.
- [9] Shye A, Scholbrock B, Memik G. Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures. Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2009: 168-178.
- [10] Dong M, Zhong L. Self-Constructive High-Rate System Energy Modeling for Battery-Powered Mobile Systems. Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM, 2011: 335-348.
- [11] Zhang L, Tiwana B, Qian Z, et al. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. ACM, 2010: 105-114.
- [12] Min C, Yoo C, Hwang I, et al. Sandra Helps You Learn: the More You Walk, the More Battery Your Phone Drains. Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. ACM, 2015: 421-432.
- [13] Pathak A, Hu Y C, Zhang M, et al. Fine-Grained Power Modeling for Smartphones Using System Call Tracing. Proceedings of the sixth conference on Computer systems. ACM, 2011: 153-168.
- [14] Mittal R, Kansal A, Chandra R. Empowering Developers to Estimate App Energy Consumption. Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012: 317-328.
- [15] Li D, Tran A H, Halfond W G J. Making Web Applications More Energy Efficient for OLED Smartphones. Proceedings of the 36th International Conference on Software Engineering. ACM, 2014: 527-538.
- [16] Zhao X, Guo Y, Feng Q, et al. A System Context-Aware Approach for Battery Lifetime Prediction in Smart Phones. Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, 2011: 641-646.
- [17] Kang J M, Seo S, Hong J W K. Personalized Battery Lifetime Prediction for Mobile Devices Based on Usage Patterns. Journal of Computing Science and Engineering, 2011, 5(4): 338-345.
- [18] Kim D, Chon Y, Jung W, et al. Accurate Prediction of Available Battery Time for Mobile Applications. ACM Transactions on Embedded Computing Systems (TECS), 2016, 15(3): 48.
- [19] Aharony N, Pan W, Ip C, et al. Social fMRI: Investigating and Shaping Social Mechanisms in the Real World. Pervasive and Mobile Computing, 2011, 7(6): 643-659.
- [20] Breiman L. Random forests. Machine Learning, 2001, 45(1): 5-32.
- [21] Friedman J H. Stochastic gradient boosting. Computational Statistics & Data Analysis, 2002, 38(4): 367-378.
- [22] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine. Annals of statistics, 2001: 1189-1232.
- [23] Chen T, Guestrin C. Xgboost: A Scalable Tree Boosting System. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.
- [24] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-Learn: Machine Learning in Python. Journal of machine learning research, 2011, 12(Oct): 2825-2830.
- [25] Armstrong J S. Evaluating Forecasting Methods[M]//Principles of forecasting. Springer, Boston, MA, 2001: 443-472.
- [26] Hyndman R J, Koehler A B. Another Look at Measures of Forecast Accuracy. International journal of forecasting, 2006, 22(4): 679-688.
- [27] Smucker M D, Allan J, Carterette B. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, 2007: 623-632.