Department of Computer Science
Series of Publications A
Report A-2018-2

Crowdsensed Mobile Data Analytics

Ella Peltonen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public examination in Auditorium PIII, Porthania, City Center, Helsinki on February 26th, 2018 at 12 o'clock noon.

University of Helsinki
Finland

Supervisor
Prof. Sasu Tarkoma, University of Helsinki, Finland
Dr. Petteri Nurmi, University of Helsinki, Finland

Pre-examiners
Prof. Mika Ylianttila, University of Oulu, Finland
Prof. Cristian Borcea, New Jersey Institute of Technology, USA

Opponent
Prof. Nicholas Lane, University of Oxford, United Kingdom

Custos
Prof. Sasu Tarkoma, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/
Telephone: +358 2941 911, telefax: +358 9 876 4314

ISSN 1238-8645
ISBN 978-951-51-4051-7 (paperback)
ISBN 978-951-51-4052-4 (PDF)
Computing Reviews (1998) Classification: H.2.8, H.1.1, H.1.2
Helsinki 2018
Unigrafia

Crowdsensed Mobile Data Analytics

Ella Peltonen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
ella.peltonen@cs.helsinki.fi
https://www.cs.helsinki.fi/u/peltoel/

PhD Thesis, Series of Publications A, Report A-2018-2
Helsinki, February 2018, 100+91 pages
ISSN 1238-8645
ISBN 978-951-51-4051-7 (paperback)
ISBN 978-951-51-4052-4 (PDF)

Abstract

Mobile devices, especially smartphones, are nowadays an essential part of everyday life. They are used worldwide and across all the demographic groups - they can be utilized for multiple functionalities, including but not limited to communications, game playing, social interactions, maps and navigation, leisure, work, and education. With a large on-device sensor base, mobile devices provide a rich source of data. Understanding how these devices are used help us also to increase the knowledge of people's everyday habits, needs, and rituals. Data collection and analysis can thus be utilized in different recommendation and feedback systems that further increase usage experience of the smart devices.

Crowdsensed computing describes a paradigm where multiple autonomous devices are used together to collect large-scale data. In the case of smartphones, this kind of data can include running and installed applications, different system settings, such as network connection and screen brightness, and various subsystem variables, such as CPU and memory usage. In addition to the autonomous data collection, user questionnaires can be used to provide a wider view to the user community. To understand smartphone usage as a whole, different procedures are needed for cleaning missing and misleading values and preprocessing information from various sets of variables. Analyzing large-scale data sets - rising in size to terabytes - requires understanding of different Big Data management tools, distributed computing environments, and efficient algorithms to perform suitable data analysis

and machine learning tasks. Together, these procedures and methodologies aim to provide actionable feedback, such as recommendations and visualizations, for the benefit of smartphone users, researchers, and application development.

This thesis provides an approach to a large-scale crowdsensed mobile analytics. First, this thesis describes procedures for cleaning and preprocessing mobile data collected from real-life conditions, such as current system settings and running applications. It shows how interdependencies between different data items are important to consider when analyzing the smartphone system state as a whole. Second, this thesis provides suitable distributed machine learning and statistical analysis methods for analyzing large-scale mobile data. The algorithms, such as the decision tree-based classification and recommendation system, and information analysis methods presented in this thesis, are implemented in the distributed cloud-computing environment Apache Spark. Third, this thesis provides approaches to generate actionable feedback, such as energy consumption and application recommendations, which can be utilized in the mobile devices themselves or when understanding large crowds of smartphone users. The application areas especially covered in this thesis are smartphone energy consumption analysis in the case of system settings and subsystem variables, trend-based application recommendation system, and analysis of demographic, geographic, and cultural factors in smartphone usage.

Computing Reviews (1998) Categories and Subject Descriptors:
H.1.1 Information Systems, Value of information
H.1.2 User/Machine Systems, Human factors
H.2.8 Information Systems, Data mining

General Terms:
Crowdsensing, Mobile Devices, Data Analytics

Additional Key Words and Phrases:
Data Cleaning, Machine Learning, Large-scale Data Analysis

Acknowledgements

The funded PhD position of Doctoral Programme in Computer Science (DoCS) has made it possible for me to focus full time on my PhD research and travel to important conferences of my research area. I also extend my gratitude to Nokia Foundation that awarded me the Scholarships in 2015 and 2016. These external fundings made it possible for me to visit University College London, UK, during the academic year 2015 – 2016.

I would like to thank my supervisors, Professor Sasu Tarkoma and Dr Petteri Nurmi, who have supported me through the ups and downs of my PhD process. Dr Eemil Lagerspetz, with whom I have worked from my undergraduate traineeships, has been an important co-worker during all these years. I am also grateful to my co-authors: Professor Stephan Sigg from Aalto University, Dr Mirco Musolesi and Dr Abhinav Mehrotra from University College London, and Jonatan Hamberg and all the other research assistants of the Carat project. Thank you for your brilliant ideas and precious discussions.

For me, one of the best parts of becoming a researcher has been meeting so many outstanding people around the world. There are too many names to list them all, but I would like to especially thank the following people for their support and encouragement: Professor Cecilia Mascolo and Dr Eiko Yoneki from the University of Cambridge, UK, Dr Aarathi Prasad from Skidmore College, US, and Dr Denzil Ferreira and Dr Susanna Pirttikangas from the University of Oulu, Finland. N^2 Women and ACM-W Europe have provided me with insightful networks for discussion and meeting great researchers.

Many co-workers, colleagues, and friends of mine have supported me beyond reason - I will always remember you warmly. At the University of Helsinki, Dr Pirjo Moen and the Niklander family have offered me invaluable help on every kind of practicalities and everyday problems. I would also like to extend my thanks to everyone who hosted me during my research visits and trips all around the world, especially the people in the Intelligent Social Systems Lab at University College London, the Computer Laboratory at

the University of Cambridge, and the Insight Centre at University College Cork, Ireland, together with the Jokela and Gibbs families in the UK.

My own family in Finland has supported me beyond all expectations, with warmth, trust, and purrs. Last, all my love and gratitude belongs to my husband Iivari. *This is hard, but that's how I wanted it.*

In Cork, Ireland, January 30, 2018

Ella Peltonen

Contents

Chapter 1

Introduction

1.1 Motivation

Mobile devices, especially smartphones, are nowadays an important part of everyday life. Different mobile applications support work life, well-being, education, and leisure time. Because smartphones are flexible and easy to carry, they have replaced multiple single-purpose devices, such as regular mobile phones, pocket cameras, gaming consoles, maps, and navigators. To enable all these multipurpose functionalities, smartphones have to implement different sensing capabilities on their programming interface. Because of this, smartphones provide a rich source of different types of data available: sensor readings, running applications, system settings, and different subsystem variables, such as CPU and memory usage. This information, especially collected from multiple devices, can provide important insights in how people behave and what kind of needs they have in their everyday life.

Guo et al. [1] define *crowdsensing* as a large-scale sensing paradigm based on user-companioned everyday devices, including, for example, mobile phones, tablets, and many wearable devices. In the future, many new household devices, such as smart TVs, fridges, and cars, will join this Internet-connected crowd. Crowdsensing is based on collaboration of a heterogeneous *crowd* of smart devices. Analysis of that kind of data collected from multiple devices can provide novel insights and help to consider what is normal in the device community. Sometimes the term *crowdsourcing* is used in the same meaning, but often it involves human-provided input, whereas crowdsensing indicates an autonomous process where a crowd of devices is used as self-supporting sensors [2].

Ganti et al. [3] remind us that there are challenges, but also a lot of new opportunities in crowdsensing applications. Smartphones and other

mobile devices have become efficient with computational power, storage space, and communication capabilities. Mobile devices are largely carried along everywhere people go and whatever they do. These features also make smartphones different than traditional sensor networks, where sensor functionality and location were often considered for a single purpose only.

Often a cloud or single virtual machines are used for back-end processes, such as managing data collection, data cleaning and processing, and the actual analysis phase. Because smart devices produce easily large amounts of data in a comparably short period of time, also techniques and technologies related to Big Data processing and distributed computing environments have to be considered. The data analysis output, for example, feedback, visualizations, and recommendations, can thus be sent back to the devices from the back-end service.

This thesis focuses on crowdsensing for smart devices, especially smartphones. It will cover three key topics: crowdsensed data collection, data cleaning and processing procedures, and it will present three example cases of how crowdsensed data analytics can be utilized. These example cases are the following: First, we show how system settings and subsystem variables of the smartphones can be adjusted to save energy and provide longer battery life. Second, we analyze application trends and present a methodology to improve application recommendations based on the actual success of different applications. Third, we analyze mobile users worldwide and suggest mobile usage as a novel cultural factor to define cultural boundaries between countries.

1.2 Problem Statement

Holistic understanding of smartphone crowdsensed data is an important open research topic. Complex interdependencies between application usage, system settings, and different subsystem variables, together with a need for real-life data, make holistic analysis challenging. This thesis aims to provide techniques and methods for analyzing mobile usage in the wild and generating actionable recommendations for optimizing smartphone functionalities, such as energy efficiency, recommendation of suitable applications, and understanding smartphone usage as a whole.

Jagadish et al. [4] define challenges for Big Data processing, which are relevant to the crowdsensing applications especially taking into account the amount of data smartphones are capable of producing in a short period of time. Four of these challenges that are especially covered in this thesis, are:

- **Data acquisition.** The programming interfaces of the smartphones

usually provide a wide set of sensors and other readings also for third-party developers. These can be utilized for data collection. In Section 2 we discuss in more detail for which purposes mobile data have been collected.

- **Information extraction and cleaning.** Crowdsensed data is only rarely usable directly, but there is a need for preprocessing and cleaning procedures. In Section 4 we present attributes that are easy to collect from smartphone platforms, and what kind of cleaning procedures we have applied to these attributes.

- **Modeling and analysis.** The large scale of crowdsensed mobile data sets is own challenge alone. In Section 5 we discuss distributed systems and algorithms used to scope performance and effectiveness of the analysis procedures. We also give examples of how these methodologies have been utilized in our work.

- **Interpretation.** Understanding the analysis results is crucial when aiming to provide recommendations that are of real utility back to the devices. In Section 6, we present use cases for actionable, human-readable recommendations and decision making based on the crowdsensed data analysis.

Taking into account these challenges, the research questions considered in this thesis can be listed as the following:

RQ1. How do different data attributes have to be cleaned and preprocessed to produce a reliable picture of the system state?

RQ2. How can crowdsensed data be used to present crucial factors of a smartphone's system state?

RQ3. What are the effects of subsystem variables, system settings, and their combinations to smartphone energy consumption?

RQ4. How can smartphone energy consumption be improved by recommending better system state and subsystem variables?

RQ5. How can mobile recommendation systems be improved by analyzing application popularity?

RQ6. What can be learned about mobile application usage and popularity in real-life crowdsensed data?

RQ7. How does mobile application usage reflect differences in user population?

RQ8. What can be learned about cultural, demographical, and geographical differences in crowdsensed smartphone usage?

Figure 1.1 presents how the research questions are covered in the publications listed below in Section 1.4 and also shortly summarizes methodologies involved in each research question. The first four research questions closely relate to smartphone energy analysis, even if findings and methodologies may be useful also in other application areas. RQ1 reflects a need for real-life data to understand actual usage cases and environments when studying smartphone usage and, for example, energy consumption. RQ2 studies how data gathered by a crowdsensed system need to be preprocessed and cleaned to produce reliable results. RQ3 derives analysis of complex interdependencies between system settings and subsystem variables, and RQ4 presents how these interdependencies can be modeled to generate actionable, human-understandable energy recommendations.

RQ5 and RQ6 relate to application usage analysis. First, RQ5 manages application popularity based on real-life crowdsensed data and answers the question, what happens after applications are installed to the device? Second, RQ6 focuses on the question how usage information can be utilized for application recommendation systems. RQ7 and RQ8 aim to deepen the understanding of smartphone usage in the wild. RQ7 delivers information about the effect of culture and demography in smartphone application usage, and RQ8 aims to describe smartphone usage as a modern cultural factor in benefit of the research community.

1.3 Methodology

Machine learning algorithms and statistical tests are crucial to understand interdependencies and relationships in the crowdsensed data. To generate actual value out of the analysis output, we have to consider how these results are presented in a human-readable, understandable and actionable way. The aims of large-scale crowdsensed data analysis include providing useful information out of the data to be used, for example, making decisions, generating recommendations, and showing helpful visualizations based on the data.

In the continuous sensing process, better usage suggestions on the device side would also generate back to the data and its analysis process. This phenomenon can be called the *continuous feedback loop*. Figure 1.2 presents

Figure 1.1: Research questions and their matching publications along with the methodology used.

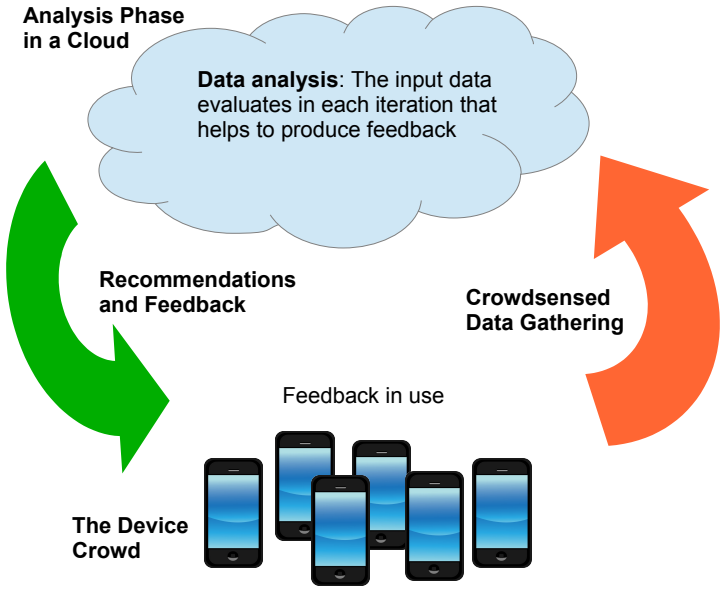


Figure 1.2: An example of a continuous feedback loop for crowdsensing applications.

an example of the continuous feedback loop, where data collected from a crowd of mobile devices is evaluated in the cloud back-end, and learning output is sent back to the devices as recommendations and feedback.

Figure 1.3 visualizes the whole process required for crowdsensed systems applying machine learning procedures and actionable feedback loop, where devices are used not only to collect the data, but also benefit the analysis output. The main phases of the system can be listed as the following, numbers of the list matching the ones in Figure 1.3:

1. A smartphone application developed for data readings and collection to perform the actual crowdsensing phase.

2. A back-end service or a cloud computing environment to manage load balancing, data storage, and the data cleaning and analysis procedures, which are next given in more detail.

3. **Data cleaning and preprocessing** to handle missing data items, unexpected values, and develop further information from attribute combinations and their interdependencies. For example, this thesis

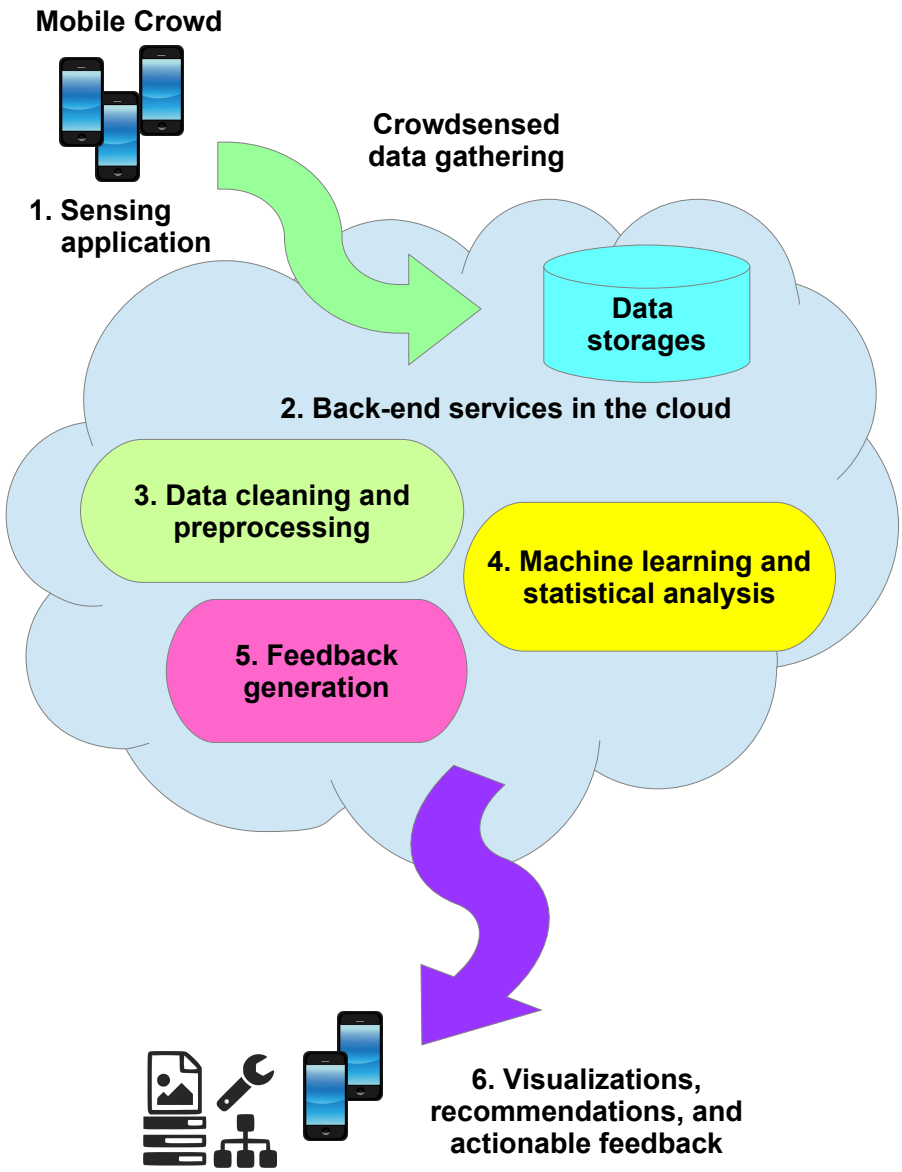


Figure 1.3: Example of a crowdsensing system that utilizes machine learning and actionable feedback.

gives approaches to clean system settings and subsystem variables by defining their reasonable operation ranges, developing general categorized usage of running applications, and present country based on network and timezone information.

4. **Machine learning algorithms** to provide statistical information, data models, and novel knowledge from the data. For example, this thesis uses information analysis - mutual and conditional mutual information - to present statistical associations, decision trees to model transactions between system states, retention rates and trend filters to understand application popularity, and the Kullback-Leibler divergence to analyze differences in application usage.

5. **Post-processing of algorithms' output** to provide actionable recommendations, feedback, visualizations, etc, to the devices and analysis environments. For example, this thesis presents how to provide energy recommendations based on system settings and subsystem variables, how to improve application recommendations based on the trend filtering, and what can be learned about cultural, demographical, and geographical differences in mobile usage.

6. The devices and other end-users, such as developers and researchers, utilizing the output of the data analysis.

The main contributions of this thesis are to give approaches for (i) the crowdsensed data cleaning and preprocessing, which is challenging with the data collected from real-life conditions, (ii) providing suitable machine learning and statistical analysis procedures that can handle large amounts of data in a sufficient period of time, and (iii) generating actionable feedback, such as recommendations and human-readable analysis results, that can be utilized in the mobile devices themselves or when understanding large crowd of smartphone users.

1.4 Thesis Contributions

The author of this work contributes the following published articles and manuscripts under revision. When referring to *the author*, it indicates the author of this thesis. These publications and manuscripts also construct the outline of this thesis, and the main focus has been given to the work the author has contributed herself.

Publication I: Energy Modeling of System Settings: A Crowdsourced Approach. Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. Published in the Proceedings of the IEEE International Conference on Pervasive Computing and Communications, PerCom '15, St. Louis, MO, USA, March 23-27, 2015.

Contribution: The author was in the lead of the planning of the publication, implementing necessary distributed data mining and statistical analysis algorithms, analyzing the data, and writing the publication. The data collection itself is based on the earlier work done in the Carat project lead by Dr Eemil Lagerspetz. Dr Petteri Nurmi and Prof. Sasu Tarkoma gave important contributions to the planning and writing processes of the publication.

Publication II: Constella: Crowdsourced System Setting Recommendations for Mobile Devices. Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. Published in Pervasive and Mobile Computing, Volume 26, February 2016, pages 71 - 90.

Contribution: The publication extends Publication I with a novel recommendation system for energy consumption of system settings and subsystem variables. Some parts of the work is based on the author's Master's Thesis published in 2013 at the University of Helsinki[1]. The author was responsible for implementing the decision tree-based recommendation system, perform the data analysis procedures, and write the publication. Dr Eemil Lagerspetz, Dr Petteri Nurmi, and Prof. Sasu Tarkoma contributed to the planning and writing process of the publication.

Manuscript I: Exploiting Usage to Predict Instantaneous App Popularity: Trend Filters and Retention Rates. Stephen Sigg, Eemil Lagerspetz, Ella Peltonen, Petteri Nurmi, and Sasu Tarkoma. A preprint is available in `https://arxiv.org/abs/1611.10161`. Under submission and review to a journal publication.

Contribution: The publication was lead by Prof. Stephan Sigg who delivered the main ideas, methodology, and structure of the publication. The author contributed by participating in the planning of the publication, and implementing and running the application recommendation system for the validation and use case of the trend filter analysis. The author also

[1] `http://hdl.handle.net/10138/40924`

gave comments through the process and participated in the writing of the publication together with other authors.

Manuscript II: The Hidden Image of Mobile Usage: Uncovering the Impact of Geographic and Demographic Factors. Ella Peltonen, Eemil Lagerspetz, Jonatan Hamberg, Abhinav Mehrotra, Mirco Musolesi, Petteri Nurmi, and Sasu Tarkoma. Under submission and revision to a journal publication.

Contribution: The publication started in collaboration between the author and researchers at University College London, Dr. Mirco Musolesi and Dr. Abhinav Mehrotra. Most of the ideas that lead to the publication were delivered through the author's research visit to University College London. The author was in the lead of the data analysis work, planning the additional data gathering, such as the user background questionnaires, and constructing the publication. Jonatan Hamberg and Dr Eemil Lagerspetz contributed significantly to the implementation of the questionnaire and data collection system, and together with Dr Petteri Nurmi and Prof. Sasu Tarkoma, they participated by sharing ideas and in the writing process.

The thesis is organized as follows: Section 2 provides the state of the art for mobile crowdsensing, presents the mobile dataset used as a source of the analysis of the listed articles, and considers ethical issues related to the crowdsensing mobile data. Section 4 discusses data cleaning procedures and techniques, and presents the main attributes available in mobile devices without complicated permission policies. Section 5 discusses distributed machine learning and statistical analysis techniques used to generate the results in the listed articles. Section 6 presents the main use cases of this work, including actionable feedback and recommendation systems for smartphones. Finally, Section 7 concludes the thesis with a summary of the main findings, discussion of limitations, and possibilities for relevant future work.

To summarize, the contributions of this thesis are the following:

- The thesis provides an approach for the **crowdsensed mobile data cleaning and preprocessing**, which is challenging with the data collected from real-life conditions. This thesis shows how interdependencies and relationships between different context factors are important to consider when analyzing mobile usage and aims to understand the smartphone system state as a whole.

- This thesis provides suitable **distributed machine learning and statistical analysis** procedures that can handle large amounts of data in a sufficient period of time. The algorithms, such as the decision tree-based classification and recommendation system, and information analysis methods presented in this thesis, are implemented in the distributed cloud-computing environment Apache Spark.

- This thesis provides approaches to **generating actionable feedback**, such as recommendations and human-readable analysis results, which can be utilized in the mobile devices themselves or when understanding large crowds of smartphone users. Understanding smartphone usage as a whole provides insights in how people use their devices and which kind of needs they have for, for example, better battery life and finding new and more successful applications.

Chapter 2

Background: Crowdsensing for Mobile Devices

Mobile devices, such as smartphones, tablets, and smart watches, are nowadays an important part of everyday life[1]. Mobile devices are nowadays used instead of several previous hand-held devices, such as cameras, navigators, and gaming consoles. In addition to applications, smart devices come with a set of various sensors, settings, and other functionalities sometimes hidden from the user. Always carried along and interacted with around 60 times per day [5], they provide a rich source of information on the everyday habits of their users.

Crowdsensing mobile usage data from large sets of users worldwide provides an access to the real everyday life of people. No laboratory simulations can provide such detailed and well covered information, because the amount of possible usage combinations of different applications and system settings rises to incalculable. On the other hand, application programming interfaces of modern smartphone platforms provide various sets of easy to access attributes. Indeed, smart device usage information can be increasingly collected through non-obtrusive instrumentation of the device. For example, the Carat [6][2] and Device Analyzer projects [7, 8][3] have collected smartphone crowdsensed data worldwide.

Experiments conducted through a combination of laboratory measurements, such as power meter measurements, and a large-scale analysis of crowdsourced measurements demonstrate that the crowdsensing method-

[1]Newzoo ranked top 50 countries by the number of smartphone users, with average smartphone penetration of 39.4% or total 2.4 bn smartphone users: `https://newzoo.com/insights/rankings/top-50-countries-by-smartphone-penetration-and-users/`.

[2]The Carat project: `http://carat.cs.helsinki.fi/`

[3]The Device Analyzer project: `https://deviceanalyzer.cl.cam.ac.uk/`

ology is capable of constructing models that accurately capture complex interdependencies between system settings, sensors, and usage contexts, providing an accurate view of the *system state* of the device. In contrast with previous works, which have predominantly focused on capturing the effects of specific sensors, system settings or applications [9, 10], a methodology presented in this thesis focuses on interdependencies and the device as a whole.

2.1 Mobile Crowdsensing

This thesis and multiple previous projects consider mobile devices and its system state as a sensor. A wide sensor base of mobile devices makes crowdsensing possible to be utilized for multiple purposes, and all the possible application areas are impossible to list. A great part of previous work has focused on analyzing device- or user-specific patterns, for example, identifying potential malware infections on the smartphones [11], analyzing network traffic and what it can reveal about the device and its user [12], or identifying and characterizing the current user of the device [13].

As carry-on devices, smartphones are easy to utilize as sensors in various conditions. One of the popular application areas is transportation mode sensing, which often utilizes sensors like accelerometer, location information, cell tower availability, and other network signals. For example, Koukoumidis et al. [14] present a system called SignalGuru that uses a smartphone's camera to predict and analyze traffic signals on roads. Hemminki et al. [15] use accelerometer and GPS location points to detect current transportation mode, such as bus, train, or walking.

Mobile devices work as sensors also indoors in contrast to, for example, GPS and network signals possible unaccessible or weak indoors. For example, images captured by camera may be used to deliver information about the usage context. Radu et al. [16] monitor indoor Wi-Fi networks, Gao et al. [17] model indoor structures and landmarks, and Chon et al. [18] present a methodology to deliver information of the place from images and audio files collected by mobile crowdsensing.

A great interest has been given to recommendation systems that help users, for example, to gain a longer battery life or choose more useful applications. In general, analyzing large-scale smartphone usage data provides an access to a rich source for knowledge. Next, we consider the state of the art in the mobile crowdsensing application areas that are especially focused on in this thesis.

2.2 Data Cleaning and Processing

The term *data cleaning* describes a process where errors, inconsistencies, and missing items in the data set are removed, replaced, or otherwise handled [19]. Data cleaning aims to improve data quality and remove misleading values, for example, unnecessary default values that may affect the reliability of the statistical distributions significantly. Data cleaning is often mentioned as one of the key challenges when analyzing and processing Big Data [20] and especially the data automatically collected from sensing devices [21, 22, 23].

Based on the study of Strong et al. [24] from the year 1997, the data quality has been an important issue at least the last twenty years. Rahm and Do [19] provide an early review for data cleaning and preprocessing procedures. They list, for example, the following challenges and problems that are especially relevant for cleaning crowdsensed smartphone data:

- *Cryptic values and abbreviations* are common in smartphone environments where any spare data transmission should be reduced due to the network costs (in terms of both energy consumption and money). That may lead to shortened values and presenting nominal values as integers, for example. In the data analysis phase, interpretation of the data values should be considered right, and possible varying presentation forms standardized so that comparison between different device models is possible.

- *Illegal values* are, for example, min and max values should not be outside reasonable or permissible range. For example, the battery temperature cannot be very high or very low due to the sensor capability to read the lithium battery, and CPU usage should be given between 0 and 1, or respectively, 0% to 100%.

- *Misspellings and the like* can appear in user-changeable settings, for example, a wrongly selected timezone setting can be considered such. A reasonable amount of system settings is adjusted automatically or the user can only choose from the limited range of options, such as screen brightness setting is often adjusted by a slider. Thus, the risk of totally inconsiderable user-based inputs is quite small.

- *Missing values* can appear in the data due to a technical error, limited access to the resource, or the presence of a default value that may indicate a missing value. The missing values have to be recognized, removed, and at least, not included in the data processing and analyzing phases.

- *Varying value representations* can appear due to, for example, different manufacturers' own changes in the API. Especially missing values can be indicated as, for example, null, NaN, none, 0, or by a default value. These values have to be recognized and combined, so that their value can be considered as the same.

- *Violated attribute dependencies* mean situations where two or more data factors should be corresponding, but for some reason they are not. For example, that may be the case when the time between two samples does not match the distance traveled between them, for example, it is not possible to travel hundreds of kilometers in several minutes.

Data cleaning and management for different sensor readings have been covered in some previous literature. They focus especially on sensor readings in unreliable or noisy environments [25, 26]. To mention some relevant examples, Williamson et al. [22] study data cleaning for wearable devices, and Tong et al. [27] propose the CrowdCleaner for web-based crowdsensed data.

The sensor-based readings are often proposed to be cleaned by machine learning or other statistic approaches. Park et al. [21] use data cleaning methods for accelerometers and light sensors using thresholds to prevent outliers, episode dictionaries to model expected measurements, and the longest common subsequences to detect errors and noise in the data. Also Jeffery et al. [23, 28] present methodologies to manage missed and unreliable data readings. Several database repairing schemes are also studied and presented in the literature [29, 30].

In some cases human input is required for successful cleaning. Chu et al. [31] use crowdsourcing to validate appropriate patterns in the data. More often human work is involved to set parameters and threshold values [32], if they are not possible to learn by statistical and other autonomous methods. In our approaches, we prefer combining autonomous and human-driven approaches, for example, setting "natural" thresholds whenever available but validating findings by statistical methods.

2.3 Generating Recommendations

Recommendations are a way to introduce users to better usage policies and help them to learn hidden features of their smart devices. Great interest has been given to help users understand their devices' energy consumption in terms of gaining a longer battery life. Another important topic considers choosing the right applications out of millions of them available in the app

markets. Next, this thesis covers the current state of the art related to these topics.

2.3.1 Energy Recommendations

Mobile energy profiling refers to the process of characterizing the energy consumption of a mobile device, including running applications, system settings, sensors, and other subsystem variables and hardware components. Energy profiling is typically carried out by constructing one or more statistical models that can be correlated with specific system states with energy consumption patterns. The goal of the energy modeling is to identify energy bottlenecks at runtime and to provide actionable recommendations on how the lifetime can be improved.

The previous research provides some insights in how people consider their device's battery life and how they tend to charge the device. Banerjee et al. [33] conduct an user study showing that, for example, users tend to leave their smartphones charging overnight or whenever it is otherwise possible. They also provide a method to save energy especially focusing on the screen brightness. Rahmati et al. [34, 35] study how people interact with their device's batteries and show that people can be divided into two groups: those who charge regularly once or more a day regardless of the battery level, and those who follow notifications and feedback given by a battery manager. Ferrera et al. [36] study how understandable different battery interfaces are, and note that users tend to have very limited knowledge what to do when they face battery problems.

Improving the user's understanding of the battery lifetime of their devices requires human-readable energy recommendations. These recommendation systems can provide warnings of bug-behavior applications, which for example, Banerjee et al. [37] suggest in their study. Ma et al. [38] present a system called eDoctor that monitors battery drain and gives suggestions about possible energy-hungry applications and suspicious system events, such as heavy network traffic. Pathak et al. [39] focus on monitoring the operation system and especially abnormal CPU usage of the device. Shye et al. [40] also focus on analyzing the effect of CPU and screen brightness on the battery life.

The measurements for constructing energy models can be gathered either using specialized hardware in laboratory conditions, such as the Monsoon power monitor [4] or BattOr [41], or through the battery interface of the device [6, 42, 43]. Benefits of the data-driven approaches include capability

[4]Monsoon Power Monitor: `https://www.msoon.com/LabEquipment/PowerMonitor/`

to catch a large variety of real-life use cases. For example, Falaki et al. [10] conduct an analysis of smartphone usage patterns, revealing that usage patterns contain significant variation across users and that personalized application usage models are essential for accurate prediction of battery drain.

Agarwal et al. [44] build in MobiBug a data-driven approach for energy diagnosis. The DeviceAnalyzer project [8, 45] is gathering rich measurements of mobile device state, but the data has not yet been used for large-scale analysis, and its high sampling cycle (even 100,000 per day from a single device) can itself lead to unexpected and increased energy consumption.

The Carat application [6] is known as the first collaborative energy profiler that performs its analysis with large-scale crowdsensed data. To the best of our knowledge, the Constella model [46, 47] that bases on the data collected in the Carat project, is the first model capable of constructing fine-grained energy effects from crowdsourced measurements.

2.3.2 Application Recommendations

Choosing the most suitable applications out of millions available is becoming a popular topic in the recommendation research field. Most application markets integrate some version of recommendation systems by themselves, for example, Google Play supports both personalized recommendations and country-specific "featured" and most popular application listings. Also, several academic and commercial recommendation systems that focus on suggesting new applications to the end users have been proposed. These systems typically operate exclusively on top of a cloud back-end, requiring large amounts of teaching data, and relying on computationally intensive matrix factorization methods [48].

Most application recommendation systems operate directly on the marketplace and rely on application popularity, such as installation counts or ratings to generate recommendations [49, 50]. However, studies on mobile usage have shown that ratings and installation counts are often a poor indicator of user interest. Users tend to try out several applications without necessarily ever using them again [51, 52]. Some users may not uninstall unnecessary applications but rather keep them, even if they are tried only once. The same holds for ratings which do not necessarily reflect true user interest. For example, many users give a one star rating for apps that do not function properly on their device [52], and some applications, especially games, even repay for higher ratings. It has been shown that usage patterns are highly contextualized, with many applications only being used in specific contexts [53], for example, tourism or transportation apps in a visited city.

Some popular app recommendation systems include, for example, AppJoy [52] that considers a weighted model where recency, frequency, and duration of interactions are taken into consideration. Other recommendation systems, such as GetJar [54] and Djinn [55], operate on binary usage patterns. AppJoy relies on a constantly running background process that monitors app use, while both GetJar and our technique can be used with crowdsourced, infrequently sampled data. Also other works on integrating context information, such as location or timing, as part of app recommendations have been proposed [53, 56, 57, 58, 59, 60]. Recently, commercial app recommendation systems, such as Aptoide[5] and Cydia[6], have emerged.

Our work in [61] uses application usage collected by crowdsensing from real users and real use cases. It focuses on adapting classic content-based and collaborative filtering techniques for mobile usage. Information learned from the trend analysis can be further used to improve the existing application recommendation systems.

2.4 Analyzing Mobile Usage

In addition to recommendations systems, there are also other essential possibilities for benefiting crowdsensed data from mobile devices. Before this, the full picture of how mobile devices have been used worldwide needs to be covered. Various previous projects have focused, for example, presenting the effect of context, timing, and location on smartphone usage. The main challenges and limitations in these works is related to the lack of worldwide, large-scale data, but in general, they give a picture how and why mobile devices are used.

Ferreira et al. [62] present that social and spatial context have a strong influence on application usage in general. They show that mobile applications are more often used at home and alone, and a large part of interactions with the phone can be considered as a "micro-usage", such as checking notifications or just killing time. Hiniker et al. [63] show that app usage reflects both instrumental (for some purpose) and ritualistic (more habitual) behavior. The instrumental use can be, for example, looking up opportunities and utilities, tracking sport or health activity, or getting in touch with other people. The ritualistic usage includes different kinds of "time killing" activities such as browsing blogs or news, playing games, or checking social media.

[5]The Aptoide meta-store: `http://m.aptoide.com/`

[6]The Cydia package management software for jailbroken iPhones: `https://www.cydiaios7.com/`

Multiple studies show that application usage reflects diurnal and daily variation. Falaki et al.[10] perform a statistical analysis and show the existence of the diurnal patterns with significantly risen activity during daytime hours compared to nighttime hours. On the other hand, they note that the exact patterns of individual users vary. Xu et al. [64] show that news apps are the most popular in the early morning and sports apps in the evening. Böhmer et al. [65] also note the risen popularity of news as well as the built-in music app in the morning hours, Google Maps in the early evening hours, and several games and e-readers in the late evenings. Both studies agree on the risen application usage when moving around, with not only traveling applications and maps, but also video and multimedia apps. The same effect might be seen in the risen energy consumption when moving around instead of staying stationary [46]. On the other hand, smartphones are still widely used for communication purposes and the communication apps are used evenly during the day [65]. Also, Jones et al. [66] study how often the apps are revisited and show that the usage patterns depend on the application and its functionality.

Verkasalo [60] shows that the location has significant correlation how smartphones are used. Xu et al. [64] study geographical differences in application usage in the US and show that 20% of applications can be considered local. They also present that the US users tend to have multiple applications for the same purpose, for example, several news applications. Petsas et al. [67] show the similar effect that the most popular apps gain the most downloads, and the users tend to have several apps from the same categories. In general, user preferences for application usage seem to be highly clustered.

Several studies show that there are also demographic and cultural boundaries in application usage. Seneviratne et al. [68] demonstrate that application usage reflects the user's gender and age. Zhao et al. [69] study over 100.000 Chinese smartphone users and find out that they can be clustered to descriptive groups, such as, "evening learners", "young parents", "financial users", and "cat lovers". They show that there is correlation between gender, age, and income level to the application usage. Lim et al. [70] analyze application download decisions across countries, finding the importance of pricing, reviews, and app descriptions to vary across countries. Kang et al. [71] compare the US and Korean smartphone users in terms of culture and basic need, such as belongingness and self-actualization.

Mobile usage can also be used to identify cognitive or personal states. Chittaranjan et al. [72] present that smartphone usage correlates with the users' Big Five personality traits. A system called MoodScope uses

applications usage patterns and other smartphone sensors to identify the user's mood [73]. Lathia et al. [74, 75] present the EmotionSense system that uses smartphones to track human behavior and changes in it. Sandstrom et al. [76] use smartphone-based crowdsensing to show that people's feelings vary in different locations and situations.

In addition to everyday mood and emotions, smartphones may help with mental illnesses. Gruenerbl et al. [77] show that smartphone sensors can be used to aid even psychiatric diagnosis. They use an accelerator to measure physical motion and GPS traces to detect travel patterns and aim to predict manic episodes of bipolar disorder patients. The MoodScope system's results are also shown to correlate with the PhQ-9 depression scores [78].

Understanding mobile usage may provide researchers and other parties valuable information of people's daily life patterns and their common needs and preferences [79]. Obviously, that kind of knowledge also benefits marketing and consumer targeting.

2.5 Large-Scale Data Analysis

Because of computational power and especially battery lifetime are limited in smartphones, a current popular approach is to collect and analyze crowdsensed data on the back-end servers, which often means introducing cloud-computing services or a cluster of virtual machines. Large-scale data processing power has become available for many users, developers, and researchers thanks to the new cloud-computing environments that do not require heavy hardware investments, but only a credit card. Amazon Web Services [7] and Microsoft Azure [8] are examples of this kind of popular cloud-computing services. The newest addition to the easy-to-access data analysis family is Gluon [9], a collaboration project between Amazon and Microsoft.

Even if these cloud-based computing resources are well available, there are challenges in implementing effective machine learning support for mobile crowdsensing. Understanding distributed environments and implementation of scalable analysis algorithms becomes crucial, when data size and diversity increase rapidly. Distributed environments require new paradigms compared to the traditional single-machine computing. MapReduce [80, 81, 82] has been seen as a leading new computational paradigm of the field, implemented

[7] `https://aws.amazon.com/`

[8] `https://azure.microsoft.com/`

[9] `https://github.com/gluon-api/gluon-api/`

in Hadoop [10] and often used together with its machine learning libraries, for example, Mahout [11] and SystemML [83].

Apache Spark [12] [84] provides a fast programming interface and supplementary features to the MapReduce paradigm together with its machine learning library MLlib [85] and programming interface MLbase [86]. These machine learning platforms implement many of the key functionalities for data analysis, such as, statistical tools for hypothesis testing and machine learning algorithms for classification, regression, clustering, recommendation making, topic modeling, and association analysis, and so on.

Users' reluctance to participate in crowdsensing projects is seen as a challenge, as well as researchers' lack of skills for mobile development [87]. Systems like AWARE [88] help researchers to launch their crowdsensing projects on a single platform without deep knowledge of smartphone app development for multiple platforms. Also, systems like AWARE already have a user base available, which reduces marketing and user acquisition costs.

The Carat application [6] uses its own data collection procedures and performs the analysis in the AWS Elastic Compute Cloud (EC2) service[13]. We implement our algorithms with the Spark platform whenever there is no library algorithm available or for some reason it does not fit the purpose intended. For example, information metrics used in our work, such as mutual information and conditional mutual information presented in Section 5.1, are not currently part of the MLlib library. From user point of view, the Carat provides actionable feedback from their battery life, which might have been a crucial element for gathering such a large user base.

[10] `http://hadoop.apache.org`
[11] `http://mahout.apache.org`
[12] `http://www.spark-project.org`
[13] `https://aws.amazon.com/ec2/`

Chapter 3

The Carat Project

Launched in June 2012 and still operating, the Carat application [6, 46] has been used to collect worldwide mobile usage data from Android and iOS devices. The project has been started in collaboration between the University of Helsinki, Finland, and University of California, Berkeley, USA. To the best of our understanding, it is currently one of the most comprehensive crowdsensed mobile data sources available including over 200 million samples from over 780,000 users.

To participate in data collection, users are not required to do anything else except download the application from a stock market: Google Play, App Store, or a separate Android package from the project website [1]. The data is collected to the Amazon EC2 cloud service and stored to the Amazon S3 data storage. Based on the analysis results, the clients show users actionable recommendations that help them to increase their battery life [89].

3.1 Collecting Large-scale Mobile Data

The Carat data collection includes multiple attributes available without extreme permissions. They are, for example, lists of the installed and running applications, user-changeable system settings, such as screen brightness and network type, and subsystem variables, such as CPU usage, memory state, and battery level. Also, user-specific hash identifier (referred to as the user's Carat id), timestamp, device model, and operating system version are recorded among others. Different mobile platforms offer varying list of system attributes, and some Android manufacturers may have included their own limitations to the programming interface. For these reasons, the

[1] The Carat project website: `http://carat.cs.helsinki.fi/`

amount and quality of items in the data may vary by manufacturer and operating system[2].

Because some of the features have been included in the system later than others, information available from specific years can vary. The newest addition is the mobile country code, which has been collected since March 2016. New data items are collected all the time, so that the system can also capture new device models, applications, and other changes in the market.

Originally designed for energy consumption research, the Carat sampling procedure takes a sample every time 1% of battery has been drained. This makes the data collection process very energy-efficient itself, but it also increases the length of time spent between two samples, especially when the smartphone is staying mainly idle. This may set some challenges in the cases where the Carat dataset is used for other than energy-efficiency research, for example, studying usage.

To respect user privacy, the Carat system does not collect any personal or contact information, such as phone numbers, calls or text messages, or exact location information. Ethical considerations are later discussed in Section 3.5. Altogether, country information can be delivered when certain factors of network and timezone are known, as we show in Section 4.5. Preliminary efforts to publish the Carat data for application developers and researchers have also been done [90], and the subset of the data consisting of system settings and subsystem variables has already been published as a part of our work [46]. This dataset is available on our website [3].

3.2 The Carat Data Statistics

Table 3.1 summarizes the statistics of the Carat data in June 2017. The entire Carat data has over 784,000 distinct user records. 48.8% of these were Android devices and 51.2% iPhones. There are more registrations to the system, over 864,000, but it might be that some users never opened the application again, so no samples have been sent to the back-end service. There are almost 215 million samples, and more is coming to the system all the time.

Different mobile platforms provide different context factors for third-party applications depending on their policies. As an open-sourced platform, Android provides the widest range of factors available and utilized by

[2]The full description of the data collection protocol can be find in `https://github.com/carat-project/carat/blob/master/protocol/CaratProtocol.thrift`

[3]The Carat context factor dataset is available in: `http://carat.cs.helsinki.fi/#Research`

Registered users	864,079
Users with samples	784,165
Android users	382,667 (48.8%)
iOS users	401,498 (51.2%)
Samples	214,931,177
Android applications	603,854
iOS applications	167,482
Raw data size	1.2 TB
Compressed data size	315GB

Table 3.1: The Carat data statistics 2nd June 2017.

application developers. Thus, in most cases of this thesis and in our previous work we consider the Android devices and a subset of the Carat data.

For example, we present an energy analysis of system settings and subsystem variables in Section 6.1 based on containing around 11.2 million samples from 150,000 active Android users. Modeling these energy combinations is based on our previous work [46], as well as a recommendation system Constella delivered on the basis of these energy models [47].

In another example that we later discuss in Sections 5.3 and 6.3, we perform a large-scale comparison of application usage in different countries. There we consider a subset of 5.65 million samples from Android devices. For those samples, we can validate the country of origin by a method later described in Section 4.5. To summarize, we compare the mobile country code obtained by the network to the country that is indicated by the timezone attribute. This procedure helps us to detect the country even when the exact GPS or Wi-Fi based location is not available for privacy reasons. The subset contains 25,323 Android users associated with 114 country codes, out of which 44 countries have a significant number of users (100 or more). Figure 3.1 presents the distribution of users whose country of origin can be tested by our methodology. The majority of the users are from the USA, but there is also a strong user base in Finland, India, Germany, and the United Kingdom among others.

3.3 User Background Questionnaire

Understanding who the users are, can provide important new insights to the smartphone usage. To collect more detailed information about the Carat users' demographic background, we sent a voluntary questionnaire within the Carat app to all active Android users. The questionnaire includes basic

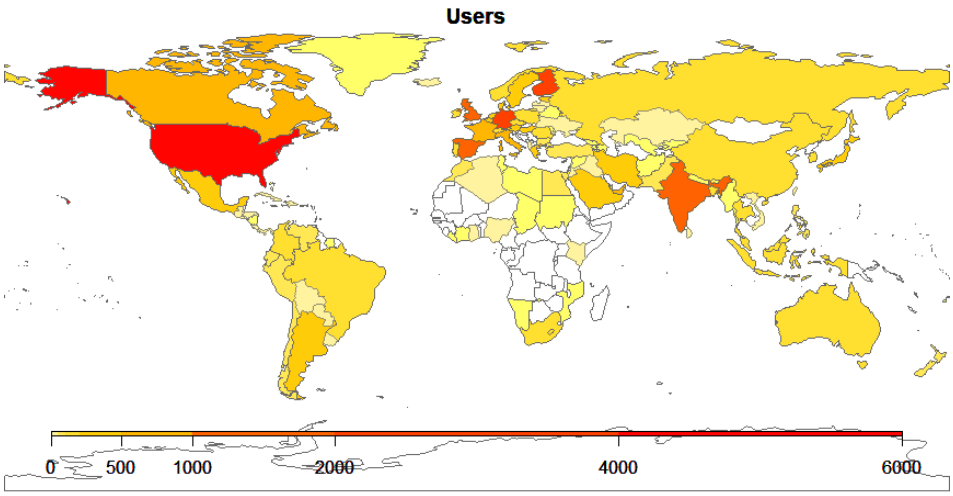


Figure 3.1: Distribution of the Carat users whose country of origin can be validated through their network's mobile country code.

background information, such as gender and age group, and socio-economic status, such as questions related to household situation and annual income. The questionnaire also records the current GPS location of the user if a permission were granted. Only adults (18 years or older) have been able to answer the questionnaire.

The following information has been collected (each question as a single choice):

1. Gender: female, male, or other;

2. Age group: 18-24, 25-34, 35-44, 45-64, or over 65 years old;

3. Current occupation: manager, professional, technician or associate professional, clerical support, sales or services, agricultural or forestry or fishery, craft and trade or plant and machine operations, entrepreneur or freelancer, student, staying at home, retired, or no suitable option;

4. Highest completed education: elementary school or basic education, high school or sixth form or other upper secondary level, vocational school or trade school or other education leading to a profession, undergraduate or lower university degree (Bachelor's or equivalent), professional graduate degree or higher university degree (Master's or equivalent), research graduate degree (PhD or equivalent);

5. Household situation: living alone, living with other adult(s), living alone with under-aged kid(s) (under 18 years old), living with other adult(s) and kid(s);

6. Annual income, compared to the user's country average: much lower, lower, about the same, higher, or much higher;

7. Debt, as percentage of monthly income need to cover it: no debt, or 10%, 25%, 50%, or most of the income;

8. Savings, as a number of months possible to live off it: less than a month, 1-3 months, 4-6 months, 7-12 months, or over a year;

9. Current coarse location, if user agrees to measure it: yes or no, measured automatically if agreed.

The users' answers can be linked to their application usage through their Carat id, a unique hash code generated automatically for each user. The questionnaire received 3,293 responses from individuals in 44 countries. This corresponds to 14.3% of active Carat users that have the latest Carat version and thus the questionnaires available.

In comparison to the results from a prior questionnaire from 2013 [89], the demographic distributions are quite similar with the exception of user locations, where the majority now coming from Finland instead of the United States. This can be caused by the marketing bias together with the research lead switching from UC Berkeley to University of Helsinki between the studies. Another bias considers gender: 10% of answers come from female and around 87% from men. On the other hand, user questionnaires performed by mobile applications have been reported to have high gender biases before [91].

In terms of occupations, the most represented are professionals (34%), technicians or associate professionals (14%), students (12%), and managers (10%), so our questionnaire respondents are well employed. That may also reflect the general picture of owners of mobile devices. Even if they have become much cheaper in present years, there may still be financial considerations in buying such a device. The distribution of education of the respondents reflects this, too: 35% have an undergraduate degree, 30% have a Master's degree or equivalent, and 5% even have a PhD or research graduate degree. 36% of the answers report their yearly salary is higher than their country's average and 7% that it is much higher. On the other hand, age groups are evenly distributed: 12% of age 18 – 24, 30% of age 25 – 34, 28% of age 35 – 44, 27% of age 46 – 64 and 4% 65 years or older.

Section 6.3.1 later discusses the analysis of how people in different demographic groups use their smartphones. Utilizing also the country attribute, Section 6.3.2 provides comparison between different demographic and geographic influences on the mobile usage.

3.4 Limitations of the Carat Dataset

As discussed before, the Carat user population – or at least those who voluntarily take also the questionnaires – seems to be biased towards well-educated and affluent males. Since the Carat application itself does not collect any background data, it is hard to say how well these distributions represent the Carat user population in general. Because it has been mainly marketed as an energy-saving application, the user base might be biased towards people having energy issues in their smartphones.

The sampling period of the Carat application is set based on the energy consumption: whenever the battery level changes, the system collects a sample. These samples are sent to the cloud only if the actual Carat application has been opened to avoid unwanted and potentially costly and energy-influencing network traffic. This data collection method means that the time distance between two samples is unpredictable and may vary a lot between different users, usage cases, and device models. This makes utilizing certain sensors, such as accelerometer and gyroscope, mostly impossible and the Carat application does not collect this kind of data features requiring more dense and interval-based sampling.

Some limitations, such as missing items and misleading default values, can be managed by the data cleaning technologies we later discuss in Section 4. On the other hand, these methodologies are never absolutely complete, for example, in the case the default value given by the device manufacturer seems to be coherent.

3.5 Ethical Considerations

Privacy and data security have become important issues for the crowdsensed data analysis [3]. User-accompanied devices may reveal users' daily routines and locations of home and workplace, also for malicious purposes, and unwanted marketing may become irritating in some cases. This is why we take especially care of user privacy when working with the automatically collected crowdsensed data.

The Carat system only considers aggregate-level data which contains no personally identifiable information, such as exact location, calls, text

messages, or phone numbers. Instead of the GPS location, only a distance between two successive samples is stored to the database. Even if application data and other possibly revealing information is collected, they are not trusted to any third parties without the full consent of how the data will be used. Our previous work [90] discusses our possible data sharing policies and plans in more detail. For example, application names can be hashed or displaced with descriptive categorical names, such as "game" or "flashlight", when the data is studied by third-party researchers or developers. It is also possible, that developers can only gain access to the data collected from their own application.

The privacy protection mechanisms of Carat are detailed in our previous work [6]. The data collection of the Carat application is also a subject to the IRB process of University of California, Berkeley. Users of Carat are informed about the collected data and give their consent from their devices when installing the application from the app market.

User questionnaires performed as a part of understanding the background of the Carat users have been approved on 14 June 2016 by the IRB process of the University of Helsinki, Finland. Participation in the study has been voluntary and the users have been informed about the data collection and management procedures. During the questionnaires, the exact location of the user or some other privacy-sensitive information, such as mental state and personality tests, have been collected but only with the consent of the user.

Chapter 4

Cleaning and Preprocessing Crowdsensed Mobile Data

Crowdsensing for smart devices is based on automatized data gathering processes. Hence, there is a possibility for unsuccessful readings and errors, for example, in case the device itself is in an incapable state, or the manufacturer or network operator limits accesses to certain factors. A good example of this kind of behavior can be seen when Apple closed access to the list of running processes from third-party developers on the iOS version 9 [1]. The operation system Sailfish in the Jolla phones is claimed to support also Android applications with an emulator, but in reality, most of the sensor readings though the emulator were unsuccessful [2]. Instead of coherent values, manufacturers and network operators may provide different default values, replacements, or empty fields. In addition, there is always a risk of programming bugs especially in autonomous processes.

Smartphones are considered highly privacy sensitive devices, as discussed before in Section 3.5. For that reason, there is a lack of some information, for example, in the Carat data no exact location information has been collected. Some useful information may be missing due to technical features, such as application information provides only a slight view of the actual functionality of the application. Thus, we need methods for developing new information from existing data attributes.

Mobile devices provide a rich source for different settings, applications, and other features that describe the usage context of the device. Some of them are only possible to collect when special permissions are received,

[1] Preventing *sysctl()* call in iOS 9: `https://developer.apple.com/videos/play/wwdc2015/703/`

[2] Jolla cannot provide compatibility: `https://jolla.zendesk.com/hc/en-us/articles/201440787`

Context Factor	Mean	Std	Median
CPU use	75%	33%	91%
Distance traveled	680.5 m	53.23 km	0 m
Distance (> 0)	867.06 m	2.66 km	5.85 m
Battery voltage (V)	3.78	0.61	3.84
Screen brightness	61.82	87.96	-1
Screen brightness (0-255)	128.03	85.71	109
Temperature (°C)	29.27	5.75	30
Wi-Fi signal strength (dBm)	-61.29	13.02	-61

Table 4.1: Summary statistics of selected context factors. Previously published [46, 47].

some of them are more easily available. In this work, we are interested in features, later called also *context factors*, which does not require heavy permission policies or come with standard permission routines. Together, these factors define the *system state* of the device.

4.1 Nominal and Ordinal Attributes

Context factors consist of both nominal and ordinal attributes. For nominal variables we use the different possible values as the categories, such as network type that indicates information of Wi-Fi or mobile, and applications come with their process names along other information, such as the human-readable name and information whether they are running background or foreground. Most of the context factors, such as screen brightness, battery temperature, and CPU use, are ordinal-valued. Managing different data types at the same time requires preprocessing, for example, discretization of the ordinal-valued factors.

Another challenge is set by default and missing values, that may seem obscure, for example, large negative values when considered missing battery temperature or screen brightness provided out of normal setting range. Some context factors come with possible calculation mistakes, for example, distance traveled between two samples may seem to be thousand of kilometers because of missing or default value in the location information of another sample.

For nominal variables we use all the different values as categories. To simplify the comparison of the context factors, we discretize ordinal-valued into categories using an equal frequencies procedure, in other words, each factor is divided into categories containing approximately the same number

of values. The number of categories is determined empirically and based on observations reported in previous studies related to the field. Summary statistics of selected context factors are given in Table 4.1 and the different categories are next discussed together with descriptions of each factor.

To be specific, we only discuss attributes given by the Android devices in this section. That is because in most of the cases our research only covers the Android devices due to the crucial differences compared to the iOS platform.

4.2 User-changeable System Settings

System settings are collected via the Android programming interface. Characteristically, they are visible to the user via system setting menus and the user has control over them. This also sets the main challenge for managing system settings: there are no proofs that users have adjusted them wisely. At the same time, system settings out of reasonable range can easily be considered as defaults, misreadings, or errors, because users can only control them inside the allowed ranges.

The systems settings considered in this work are the following:

Network type is a categorical attribute describing the current method used by the phone for Internet connectivity, for example, none, Wi-Fi, mobile, or WiMAX, depending on the technology used. When the network type equals mobile, detailed information about the connectivity type is given by the attribute mobile network type. The user can modify the setting by choosing the preferred networking strategy, such as allowing mobile data connection or connecting to the preferred Wi-Fi available. Some general settings, such as flight mode, affect the network type by suppressing all the network connectivity.

Mobile data status describes the current status of the mobile data interface. It is given as a categorical attribute and has one of the following values: connected, disconnected, connecting, or disconnecting. The user can modify the status allowing or disallowing the mobile data connection.

Mobile network type is a categorical attribute that specifies the mobile data transfer standard currently being used on the phone. Examples of values it can take include LTE, HSPA, GPRS, EDGE, and UMTS. The list of possible mobile networks is broad and depends on the technologies available in each country and by each operator. The user has the best

control over the mobile network type setting when choosing a data plan, which are widely marketed under the generalized names of 3G, 4G, and so on. Some devices also allow the user to choose the preferred technology later on, for example, in the case of lacking network coverages in rural areas.

Roaming describes whether mobile network traffic outside of the own operator network is allowed. The value of the attribute is either disabled or enabled, given as a binary attribute (0 or 1). The setting is possible to modify in the networking settings, but also the network operator may disallow it or it can be disabled in the customer data plan.

Screen brightness refers either to a manually adjusted brightness value, given as a numeric value between 0 and 255 where a larger value implies brighter screen, or automatic setting, given by value -1. Some devices provide 256 as the largest value (full screen brightness), but all the other values out of range [-1, 256] can safely be disregarded. The automatic setting can vary in some devices, for example, based on the sensing of the outside light. Therefore, knowing the setting parameter may not give the actual brightness value that is currently used.

Based on the Carat data, when screen brightness is manually controlled, the mean is around 128, or the exact midpoint. The distribution of the values, shown in 4.1, indicates that almost the entire range of screen brightness values is used, making it difficult to categorize screen brightness values in a meaningful way. While small brightness values generally have lower energy impact than higher values, or even automatic settings, they usually occur only in specific situations, such as at night or while reading a book in a dimly lit room. As these values are encountered very infrequently, their overall energy benefits are small compared to using automatic setting. Based on these observations, we opt for a binary split into manual and automatic brightness especially when studying energy consumption. For some other application areas, a different kind of split might be more useful.

4.3 Subsystem Variables

Subsystem variables are not directly available as a user-modifiable system setting, but can give information about the state of the smartphone. For example, if we notice a decreased Wi-Fi link speed or signal strength, we can recommend that the user try to use the mobile network instead of Wi-Fi in this context. In the energy analysis, these factors provide important insights into what happens inside the smartphone.

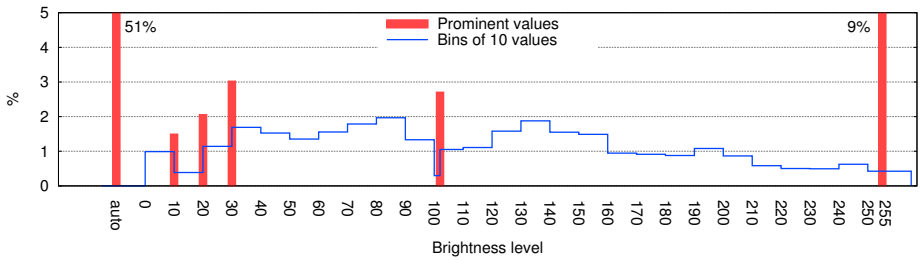


Figure 4.1: The frequency of all screen brightness settings. Previously published [47].

Misreadings, defaults, and missing data points set the most important challenge when managing and analyzing subsystem variables. At least, they should come from a reasonable range and match the given Android API description. Because manufacturers may set their own defaults, missing values or unsuccessful readings, for example, it is not straightforward to define "good" values.

The subsystem variables considered in this work are the following:

Battery health is a categorical attribute determined by the smart battery interface of the Android device. Values of the attribute are vendor-specific, with examples of common values being Good, Bad, Overheat, and Unknown failure. The value Good is the most common value in the Carat data. It is even so common that any other value might be considered an abnormal behavior.

Battery temperature is the temperature of the battery given in Celsius degrees. We only considered positive values, because in normal conditions the battery should not freeze under zero degrees. Also, some device manufacturers seem to provide high negative values when readings of the sensor are not available. With median 30 °C and mean 29.27 °C, presented in Table 4.1, we consider discretization over this value: over or under. Depending on the research objectives, a more sophisticated split might provide more information.

Battery voltage describes the current battery capacity in Volts. The safe operating voltage of a smartphone Lithium-Ion battery is in the range 3 - 4.2V. The nominal voltage of such batteries is typically 3.7V. The mean, the median, and the standard deviation presented in Table 4.1 reflect this very closely. We consider three categories for voltage: Low (0 - 3V), Medium

(3 - 4.2V), High (4.2V+). Values lower than 0 and greater than 5 were considered defaults or unsuccessful measurements.

CPU usage is a percentage (0-100%) that describes the fraction of the CPU currently used. We consider measurements that reflect the percentage of time the CPU is active. The CPU usage should be given by a value between 0 and 1, and all the other values were considered defaults or missing values. The mean and median in Table 4.1 indicate that CPUs are mostly active. We split the CPU use around the mean, resulting in three categories: Low (0 - 42%), Medium (43 - 85%), and High (86 - 100%).

Distance traveled is a location-based measurement between two samples, given in meters. For privacy reasons, the Carat application does not gather the exact location of the user, but uses distance measurements to determine whether the device has been moving or not, for example, in a car. Most of the values are during stationary periods or with little movement, especially when taking into account around 100 meters standard error of coarse localization services. Based on this observation and large statistical standard deviations, we consider a split between stationary (less than 100 meters, later referred to just as 0 meters) and non-stationary behavior (100 meters or more, later referred to as greater than 0).

Mobile data activity describes how the mobile data interface is being used. The value of this categorical attribute is one of the following: none, out, in, or inout. Mobile data activity has cross-effect on multiple settings that allow data connectivity in general, but in contrast to them, this factor describes the actual occurrence. For example, the user might have allowed the mobile data connection, but for some reason it may not be available.

Wi-Fi link speed is given in Mbps and is determined by the Android API. The attribute does not provide the actual speed used, but the capability of the closest cell tower. Thus, it better describes the maximum capacity available than real usage, and we do not consider it much in our work. The Wi-Fi link speed attribute might be useful if it is known that two or more devices are connected to the same cell tower and share the connection and thus the bandwidth, too.

Wi-Fi signal strength is given in dBm and is determined by the Android API. We only consider RSS values in the range [-100, 0] due to the technical limitations. Good Wi-Fi signal strength values are normally between -30 and

-10dBm, and the worst, while still being connected, is -95dBm. We consider four categories: Bad (-100 to -75dBm), Average (-74 to -61dBm), Good (-60 to -49dBm) and Excellent (-49 to 0dBm). The mean RSS is between the Average and the Good levels, and the Excellent and the Bad levels are within one standard deviation. These values are in line with typical values used in Wi-Fi positioning literature, and they were also restrictively tested to match four "bars" in the user interface.

4.4 Energy Measurements

Energy impact of system settings and subsystem variables is an important new research field. To measure the energy consumption of the device, we consider timestamps and battery levels reported by Carat and develop *energy rates*. These reflect normalized energy consumption per time unit, more formally defined as:

$$\text{Energy rate} = \Delta\text{battery level}/\Delta\text{time} \tag{4.1}$$

The methodology used to derive rates and the validity of using energy rates as a measurement for battery consumption has been validated and presented in previous work by Oliner et al. [6].

The energy rate distribution coarsely follows power distance: fewer rates of high energy consumption, in other words, only hours of total battery life, and most of them indicating discharge level considerable normal. We compare discharge rates routinely as consumption per second, but they can also be interpreted to more human-readable format, as hours of battery life in the given system state, as follows:

$$h = \frac{\frac{100}{rate}}{3600} \tag{4.2}$$

The difference between two different system states can thus be denoted as *battery life gain*. It measures how changes in context factors influence the lifetime of a device on average. We usually give the battery life gain as percentages compared to the average, but also actual hours of battery lifetime left in the given combination of context factors might be considered.

As an example, Table 4.2 presents battery life gains of selected subsystem variables and system settings we have studied in our work [46]. High CPU use obtains the highest energy loss. The benefit of maintaining a balanced CPU load is significant, as medium CPU use produces +5.72% energy benefit compared to the average use. For screen brightness, the automatic setting seems to improve battery life significantly, providing even +6.29%

Context Factor	Value	BL Gain
CPU use	Low (0–42%)	+3.24%
CPU use	Medium (43–85%)	+5.72%
CPU use	High (86–100%)	-2.48%
Distance traveled	None	-0.76%
Distance traveled	>0	+8.20%
Battery voltage	Low (0–3V)	-16.60%
Battery voltage	Medium (3–4.2V)	-0.76%
Battery voltage	High (4.2V+)	+69.08%
Screen brightness	Manual	-4.96%
Screen brightness	Automatic	+6.29%
Wi-Fi signal strength	Bad (-100 – -75 dBm)	-2.29%
Wi-Fi signal strength	Average (-74 – -61 dBm)	+4.00%
Wi-Fi signal strength	Good (-60 – -49 dBm)	+6.29%
Wi-Fi signal strength	Excellent(-48 – 0 dBm)	+7.63%

Table 4.2: Battery life gains of selected context factors. Previously published [46, 47].

better battery life compared to the average. Manual brightness, in contrast, shows a major loss of battery life (-4.97%). Also, Wi-Fi signal strength has a dominant effect on the energy consumption. When the Wi-Fi signal strength is considered Bad, there is -2.29% power loss compared to the average. Moving to the area of at least the average signal strength helps to gain more battery life.

4.5 Detecting Country

All the useful information is not possible to read directly from the Android API, but is derived from other collected factors. To protect user privacy, the Carat system does not gather any location information. Instead, Carat collects different attributes about the network usage, especially Mobile Country Code (MCC) as well as the current timezone. In our work [92], we propose a method to detect the country of the user without exact location information, but only using the MCC and timezone attributes.

A mobile country code (MCC) is a three-digit value tied to a mobile network. Each MCC corresponds to a single two-letter IANA country code[3]. Unfortunately, the MCC is not available on Wi-Fi-only devices, such as

[3] `http://www.iana.org/time-zones`

tablets, and some CDMA networks. From the beginning of March 2016 until May 2017, the Carat dataset has 5.65 million samples with valid MCCs.

There are 69.7 million samples with the timezone information available in the Carat dataset. The Android devices follow the IANA timezone database format and give the timezones presented as the continent and the closest big city, for example, America/New_York or Europe/London. These values can be further translated to the two-letter country codes (later referred to as CC) similar to the MCC codes.

Both mobile country codes and timezone-based country codes can sometimes have errors or they can be misconfigured. We compare the MCC and CC codes, and find that out of 5.83 million samples with valid MCC and CC values, these two indicate the same country in 97% of the samples. In those 3% of samples where MCC and CC indicate a different country, there are distinct neighbor countries such as small European states, and nearby countries in the same timezone. Difference in MCC and CC may be caused by cross-border usage of the network infrastructures in neighboring countries. Because some devices allow the user to adjust the timezone, there is a possibility of misleading selections. For example, both Europe/Athens and Europe/Helsinki represent the GMT+2 timezone but Athens is shown first in the alphabetical list, so it might be chosen also by users outside Greece for convenience.

Together with the automatic data collection in Carat, there are several volunteering questionnaires run, as described in Section 3.3. 1153 users have shared us their GPS location coordinates (latitude and longitude). We compare these locations to the user's sample history, take the MCC codes from all the user's samples, and find that, for 97% of the users, the coordinates match the most common MCC among all the samples. This means these people have been inside a single country for most of the time, and help to trust the MCC and CC analysis as a country information source.

For large-scale comparison of application usage in different countries, we consider a subset of 5.65 million samples in which the timezone-based CC and MCC fields match. The MCC is obtained from the cellular network infrastructure, and automatically converted to a two-character country code. We compare MCC with the country that the city of the timezone field corresponds to. This procedure increases the reliability of detecting the country of the user, when the exact GPS or Wi-Fi-based location is not available for privacy reasons. The subset contains 25,323 users associated with 114 country codes, out of which 44 countries have a significant number of users (100 or more). The majority are based in the USA, with strong user bases also in Finland, India, Germany, and the United Kingdom.

4.6 Applications

Running and installed applications can be seen as a fundamental factor of smartphone usage. Different applications provide functionalities for well-being, education, and leisure. There are plenty of applications available on the app markets, for example, 2.2 million applications in the Google Play store and 2 million in AppStore[4]. Different features can impact the choice of the application, for example, search results, user recommendations, and application description, and in the longer run, energy usage, performance, and user experience. Without applications and their wide functionalities, even smartphones would remain as regular phones. Interesting research questions include application usage comparison, for example, between individual users or user groups, countries, cultural areas, and so on.

The Carat application collects the following information from the running and installed applications in the device: A package name is the real name of the application, for example, `com.facebook.katana` is the package name for the main Facebook application. Also this human-readable application name will be collected, together with information if the app is a system app or update to a system app, it's human-readable version code, signatures of the app from `PackageInfo.signatures`, and package that installed the application, for example, in case of service and library applications. Importance is an attribute to describe whether the application is running foreground or background, or if it's status is something else provided by Android API, such as visible, service, or empty.

Application information collected by Carat has previously been used to study their energy consumption [6, 89] and malware prediction [11]. In this work, we present two example cases of application usage analysis: first, we analyze application usage trends in Section 6.2 that is based on Manuscript I attached to this thesis [61], and second, we present demographic, geographic, and cultural boundaries in mobile application usage in Section 6.3 that is based on Manuscript II attached to this thesis [92].

There are several challenges of analyzing application usage. First, any application seems similar to the system, and separating system applications from applications actually installed and used by the user is not always straightforward. In some cases there is the system application status provided by an importance attribute, but it is not mandatory and sometimes the difference between system app and another functionality app may be difficult to define. For example, many manufacturers and network operators

[4]Numbers of existing applications are estimates: `http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/`

provide their own, preinstalled applications for messaging, file management, picture processing, calendars, and emails. Second, there is no knowledge of the application's functionality provided in the data. The name of the application, and categories and descriptions given by the app market, may give good guesses for which purpose the app is meant, but this information first has to be gathered and processed. Thus, we next to consider this functionality or labeling problem in more detail.

4.7 Application Categories

Often it is more useful to study what applications do or are used for than the usage of single applications alone. For example, there are several messaging applications with basically the same functionalities but different popularity and language bases around the world. To avoid language and marketing biases, we can consider applications through the categories they belong to, such as communication, social apps, and different game genres.

There are two ways to label applications: by hand, which is clearly a very ineffective method, or by using categories already provided by the application markets. To obtain this categorization, we fetch the application descriptions of all the applications existing in the Carat dataset as HTML files from the Google Play store. Then we map the application names to the corresponding categories. This way each user's category usage can be detected. In October 2016, there were 55 categories on Google Play. The Carat dataset contains 97,000 different applications including system processes, out of which 54,776 applications are available in Google Play with at least one category assigned. Some apps can have multiple categories, such as family oriented action games may belong to categories *Family pretend* and *Action games*. To avoid inconvenience, these apps were considered once in every category they belong in.

There are some challenges regarding using Google Play as a source for category detection. Google Play is not available in China, so Chinese users cannot be studied. In addition, not all the applications are available in Google Play at all. For the same reason, the iPhone devices have to be excluded, even if there is plenty of iOS data in the Carat dataset. It is not possible to fetch application descriptions or categories from App Store in the same way as we can do with Google Play.

The number of categories of Google Play has increased over the years, but some application categories may still be too broad. For example, the *Tools* category contains a lot of general applications, for example, keyboards for different alphabets, flashlight apps, and other utilities such

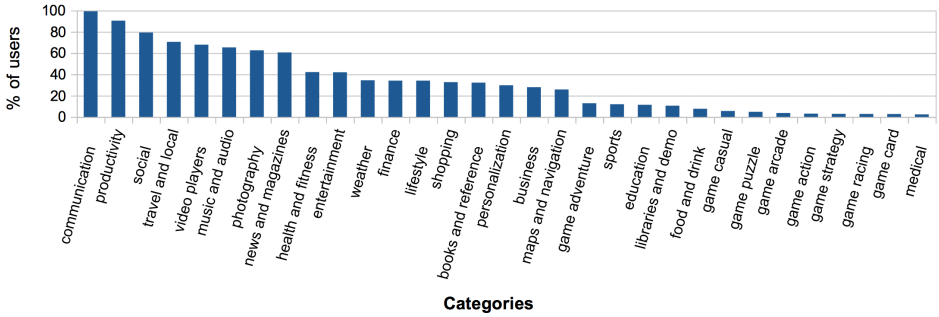


Figure 4.2: Distribution of users (%) between the top 31 (out of 55) Google Play categories. Previously published [92].

as Carat, with very different use cases. Similarly, almost every user uses the *Communication* category, since it contains common messaging apps, some of which are preinstalled on most smartphones, for example, Google's Hangouts, Facebook Messenger, and WhatsApp. When looking at the most used categories, leading categories are these two large ones: *Tools* (100% of users used) and *Communications* (99% used). Figure 4.2 shows the next popular categories, including, for example, other wide topics such as *Productivity*, *Social*, and *Travel and Local*.

Chapter 5

Methodology for Analyzing Crowdsensed Data

Data analysis methodology targets finding novel insights to the crowdsensed data. When successful, these methods are capable of constructing models that can capture complex interdependencies between the context factors. Challenges on analyzing crowdsensed data are highly related to the way data have been collected: there is often a need for cleaning and preprocessing, as discussed in Section 4.

On the other hand, mobile crowdsensing systems are capable to produce large amounts of multidimensional data in comparably short time, depending on how many attributes have been collected and the sampling period. This causes a need to perform effective machine learning procedures in a distributed environment. This Section focuses on methodological approaches to analyze large-scale mobile crowdsourced data in a suitable environment.

Several key statistical methods and algorithms essential to our work will be introduced. We introduce information metrics from statistical tools to measure the association between attributes, and methodologies to understand popularity, trend, and application usage. The output of this analysis procedure can be later utilized when making decisions or building recommendations systems, as will be discussed later in Section 6.

5.1 Information Metrics

Information metrics are used to measure the strength of statistical association between different context factors, such as system settings, subsystem variables, application usage, or any other nominal or discretized ordinal context factor available. Results of the information metrics can be used to

rank features and provide insights to understand the effect of a given context factor or a set of factors to measurable value, such as, energy consumption, popularity, or usage.

Mutual information. To measure statistical association, we consider the *mutual information* (MI) between two context factors. For assessing the influence of a single context factor X to the target factor Z, the MI is formally defined as:

$$MI(X,Z) = \sum_{z \in Z} \sum_{x \in X} p(x,z) \cdot log \left(\frac{p(x,z)}{p(x) \cdot p(z)} \right). \tag{5.1}$$

Conditional Mutual Information. For higher order combinations containing two or more context factors (denoted as X and Y in case of two factors) and the target factor (denoted as Z), the *conditional mutual information* (CMI) is formally defined as follows:

$$CMI(X,Y|Z) = \sum_{z \in Z} \sum_{y \in Y} \sum_{x \in X} p(x,y,z) \cdot \log \left(\frac{p(z) \cdot p(x,y,z)}{p(x,z) \cdot p(y,z)} \right) \tag{5.2}$$

By using CMI to analyze the impact of context factors combinations, we can identify combinations that are as informative as possible while at the same time minimizing redundancy between the different factors. Accordingly, this usage style of information metrics can be understood analogously to the use of (conditional) mutual information for the feature selection techniques in machine learning research [93].

5.1.1 Energy Impact of System Settings and Subsystems

Next, we demonstrate these information metrics by examining how context factors (system settings and subsystem variables in this case) affect the energy consumption of the mobile devices. The work has been previously published in two Publications attached to this thesis [46, 47]. We derive a ranking for different factors based on their mutual information values, presented in Table 5.1. The results of the MI analysis are well in line with previous research [10, 40]. In particular, the major individual impact of CPU use and traveled distance on battery consumption is clearly observable. The results also contain some exceptions to the findings in the previous studies. The most prominent example is screen brightness, which is commonly considered as the most battery-heavy feature. In our analysis, screen

Context Factor	MI Estimate
CPU use	1.330
Distance traveled	1.069
Battery temperature	0.143
Battery voltage	0.099
Screen brightness	0.030
Mobile network type	0.019
Network type	0.018
Wi-Fi signal strength	0.014
Wi-Fi link speed	0.014
Mobile data status	0.013
Mobile data activity	0.005
Battery health	0.004
Roaming	0.0002

Table 5.1: Context factors' impact on energy consumption, ordered by the mutual information estimate. Previously published [46].

brightness results in a lower score than many other attributes. This may be due to the fact that screen brightness often happens to be high for some reason, for example, the device has been used to play a game with heavy graphics assistance.

Similarly, we derive an energy effect for the context factor pairs by considering the conditional mutual information. The results are listed in Table 5.2. Compared to the results of individual context factors, the combination of two factors gives more accurate explanations of the battery consumption. CPU use gains significantly higher impact when combined with another factor than when considered alone. Also factors related to network connection, such as Wi-Fi signal strength and network type, are more prominent when considered in conjunction with another context factor. Capturing this kind of nuances in consumption is particularly beneficial when giving suggestions to the end user on how to improve battery life. As an example, we can observe that changing another system setting can help to improve battery life in cases where high CPU use is mandated, such as, when playing a game.

The top context factors according to energy consumption are battery voltage, CPU use, battery temperature, and movement (distance traveled) of the device, or combinations of these context factors. The effects of these factors are mediated by other factors, which in turn can cause significant increases or decreases in the energy consumption.

Context Factors		**CMI**
Battery voltage	CPU use	4.29
CPU use	Screen brightness	2.17
Battery temperature	CPU use	2.07
CPU use	Distance traveled	1.81
CPU use	Wi-Fi signal strength	1.69
Battery voltage	Distance traveled	1.53
Battery temperature	Distance traveled	1.28
Distance traveled	Screen brightness	1.26
CPU use	Wi-Fi link speed	1.12
Battery voltage	Screen brightness	1.08
Wi-Fi link speed	Wi-Fi signal strength	0.99
Mobile data status	Network type	0.95
Network type	Wi-Fi signal strength	0.85
CPU use	Mobile network type	0.80
Battery temperature	Screen brightness	0.79
Distance traveled	Wi-Fi signal strength	0.75
Network type	Wi-Fi link speed	0.64
Mobile data status	Wi-Fi signal strength	0.60
Battery temperature	Battery voltage	0.56
Distance traveled	Wi-Fi link speed	0.54
Battery voltage	Wi-Fi signal strength	0.53
Mobile data status	Wi-Fi link speed	0.46
CPU usa	Network type	0.42
Distance traveled	Mobile network type	0.37
CPU use	Mobile data status	0.32
Battery voltage	Wi-Fi link speed	0.27
CPU use	Mobile data activity	0.27
Screen brightness	Wi-Fi signal strength	0.26
Distance traveled	Network type	0.20

Table 5.2: Top of the conditional mutual information estimates for pairs of context factors for energy consumption rates. Previously partially published [46].

5.2 Trend Mining

Application popularity and trend analysis is an important part of understanding how smartphones are used. Some studies on mobile application usage have characterized factors that drive download decisions [67, 94, 95] without being able to determine what happens once the app has been installed. Some previous works have focused on overall usage and how that is influenced by contextual factors [53, 96, 97]. Some analytics companies use retention rates[1] to describe successfulness of the apps. To the best of our understanding, our study in attached Manuscript I [61] is the first to independently analyze, what happens once the application has been installed.

Retention Rates Retention rate on day d is defined as the percentage of users that continue using the application d days after the first usage. To estimate retention rates, we identify for each user and application the first and last time the user launched the application.

Retention rates of the first week are presented in Figure 5.1. First day retention rates for applications with at least 10 users are close to 50%, compared to 80% reported by many analytics companies[2]. For applications with at least 1000 users the retention rate rises to 62%. For the most popular 100 applications, the first day retention rate is even as high as 68% and after 7 days the retention rate remains higher than 50%. This analysis shows that the retention rates largely depend on the initial number of users, and popular apps stay healthy in terms of user base for longer periods of time.

Trend Analysis. While retention rate reflects the long-term attractiveness of an application to individual users, it does not cover instantaneous popularity, usage trends, or seasonal patterns. Figure 5.2 presents usage patterns of some automatically selected (based on the peak detection algorithm) applications from first day of usage up until 100 days of usage. The example indicates that application usage patterns do not always follow a simple falloff pattern suggested by retention rates. Several rising trends and again falling trends can be seen in Figure 5.2. For some and possibly various reasons, these apps have became substantive again.

[1] `http://info.localytics.com/blog/the-8-mobile-app-metrics-that-matter`

[2] `http://andrewchen.co/new-data-shows-why-losing-80-of-your-mobile-users-\is-normal-and-that-the-best-apps-do-much-better/`

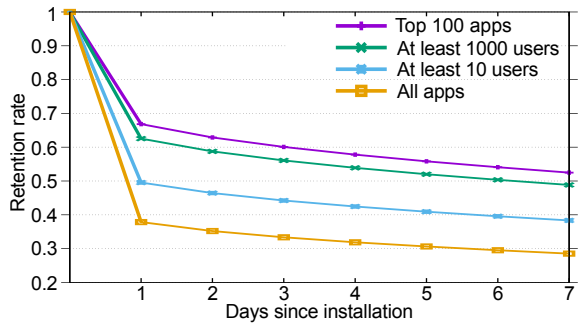


Figure 5.1: Retention rates of the first week. Previously published [61].

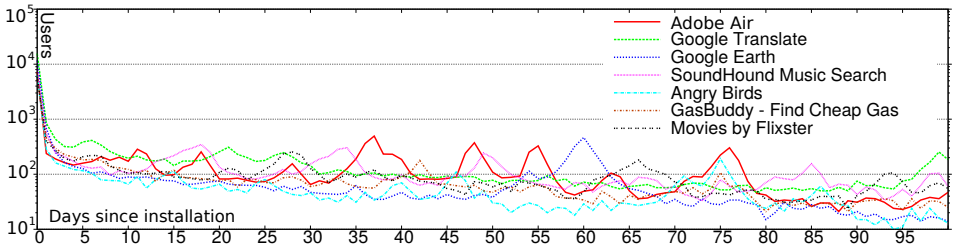
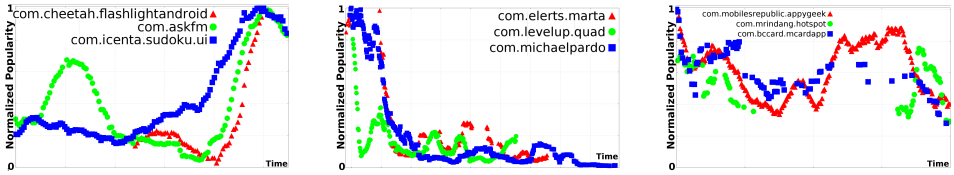


Figure 5.2: App usage patterns up to 100 days. Previously published [61].

Based on these observations, we develop in [61] a methodology to determine the application life cycle. We characterize application trend patterns to the following classes: *Flop*, *Hot*, *Dominant*, or *Marginal* apps. Figure 5.3 presents example cases of the Flop, Hot, and Dominant applications. When applying this trend analysis to the Carat dataset, we find that 40% are *Marginal* apps with a very limited user base in general. In the remaining 60%, the following patterns can be found: 0.4% are *Dominant* or gaining constantly high popularity, 1% are *Flops* or falling in continuously popularity, and 7% are *Hot* or continuously rising in popularity. These findings can be utilized in application recommendation systems, as later presented in Section 6.2.

5.3 Analyzing Similarity of Usage

Analyzing application usage data from a large population of people can provide important insights in how applications and smartphones are used in general in the wild. Because smartphones are largely considered to be important daily life devices, this kind of data analysis opens the doors to the routines, habits, and rituals people practice in their daily life.



(a) Hot apps. (b) Flop apps (c) Dominant apps

Figure 5.3: Applications trend patterns with example apps. Previously published [61].

The Carat dataset contains around 55,000 applications. To get a generally representative picture of their functionality, we map them to the Google Play categories (currently 55 possible) as presented in Section 4.7, but the same metric may also be used with an application-based approach.

Usage Vectors. We generate binary category vectors for each user considering whether that user has used a category or not (1 for usage, 0 for none). For each country (or any similar group of users), we construct the probability distribution of category usage within the country, represented as the fraction of users in the country having used that category.

Formally, for each category $c_i \in C, C = c_1, c_2, ..., c_k$ where k is a number of categories, we define the probability of its use within a country n as

$$c_{i,n} = \frac{\sum_j u_{i,j} \in U_n}{|U_n|} \tag{5.3}$$

where U_n is the set of users in country n and $u_{i,j}$ is 1 if user j used category i and 0 otherwise.

Now $C_n = c_{1,n}, c_{2,n}, ..., c_{k,n}$, is the category use probability vector for country n.

5.3.1 Demographic Usage Differences

To understand demographic differences in the application usage, we need to benefit from the definition of the category usage vectors above and the mutual information metric described in Section 5.1. We consider different demographic attributes collected from the Carat user by the questionnaire described in Section 3.3: age group, gender, current occupation, highest completed education, household situation, and the following economic factors: amount of debt, amount of savings, and level of monthly salary. In addition to these, we consider country as a similar factor affecting smartphone usage.

Attribute	Mutual Information Gain
Country	4.60
Occupation	2.78
Education	2.14
Savings	2.12
Debt	1.99
Salary	1.96
Age	1.94
Household	1.57
Gender	0.59

Table 5.3: Demographic attributes sorted by information gain against application usage. Previously published [92].

To use categorical application usage as a target factor for the mutual information metric, we consider the usage vector as a single data element. Thus, the mutual information can be considered between each demographic attribute and the application usage vectors. Table 5.3 describes the results of the analysis, sorted by the information gain. We can see that the country attribute is characterized by the highest information gain compared to the other attributes, such as gender and age, whose information is significantly lower in comparison. The high information gain for country strongly motivates to detect in more detail how countries differ in terms of mobile use.

Similar analysis can be performed conversely between individual application categories and the demographic attribute as a target factor. Table 5.4 shows the results of this analysis. For country, the category of the *Weather* applications gives the best information gain, probably because weather is more predictable in some countries than in others. Occupation is related, for example, to the *Business* and *Finance* applications, that is probably caused by academic and professional workers benefiting from the mobile techniques in their work. The household attribute that indicates whether the person is living alone, with other adults, or with kids, is best described by categories of family-related applications, such as *Family music videos*, *Parenting*, and *Dating*, the last probably for those living alone.

5.3.2 Geographic Usage Differences

Understanding how different attributes affect application usage provides us with interesting insights, but is needed also to compare and cluster users and user groups together. Comparison of application usage between users,

Attribute	App categories with highest information gain
Country	Weather, Game action, Finance, Family pretend
Occupation	Business, Game adventure, Finance, Family pretend
Education	Finance, Game adventure, Shopping, Music and audio
Savings	Game adventure, Game simulation, Entertainment, Personalization
Debt	Finance, Books and references, Game simulation, Family music video
Salary	Game adventure, Business, Game casual, Game simulation
Age	Game adventure, Weather, Business, Family music video
Household	Family music video, Dating, Family action, Parenting
Gender	Business, Game casual, Personalization, Books and reference

Table 5.4: Application categories that gain the highest information against each demographic attribute. Previously published [92].

or a set of users, requires a suitable similarity metric. We use there the Kullback-Leibler divergence (KL), which is a relative entropy metric used to detect how a probability distribution diverges from another.

Kullback-Leibler Divergence. To compare the usage vectors with each other, we use the Kullback-Leibler divergence (KL). For two probability vectors it is formally defined as

$$KL(C_n||C_m) = \sum_{i=1}^{k} C_n(i) \log \left(\frac{C_n(i)}{C_m(i)} \right) \tag{5.4}$$

However, since the KL divergence is not symmetric and it does not satisfy the triangle inequality, in our analysis we use the logarithmic sum of two-way KL divergences as a distance metric, so that the distance between two user countries is given by

$$dist(C_n, C_m) = \log \left(KL(C_n||C_m) + KL(C_m||C_n) \right) \tag{5.5}$$

As an example case, we present how the KL divergence is used to compare usage in different countries. The work is presented in the attached Manuscript II [92]. Figure 5.4 gives a dendrogram presentation based on the KL divergence between 44 countries well represented in the Carat

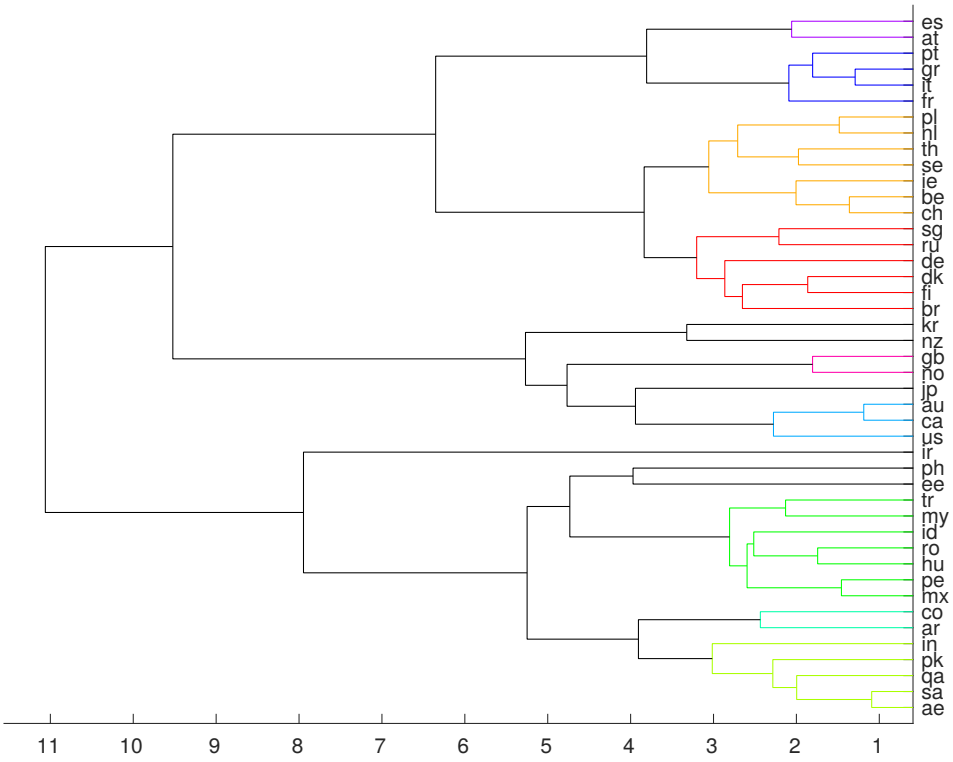


Figure 5.4: A dendrogram visualization of the Kullback-Leibler divergence between countries. Previously published [92].

data. There, we can see three main branches in the dendrogram. The topmost group contains mostly European countries. Its subgroups roughly correspond to southern (the top five countries), central (the next seven), and eastern Europe (the last six in the group). Brazil (br) may be included due to the effect of language or history.

The middle branch in the dendrogram or the next group contains English-speaking countries such as the USA, Australia, Canada, New Zealand, the United Kingdom, and other countries with early adopters of the Carat app, such as South Korea (kr) and Japan (jp). Norway (no) may be included because of its location near the United Kingdom. The latter three countries may also be included because the Carat user questionnaires have been only presented in English, so those familiar with English applications may have answered the questionnaire more readily than others.

The third main branch or group consists of the rest of the countries, with some meaningful demographical or geographical groups, such as Columbia (co) and Argentina (ar) in South America, and the Arab Emirates (ae),

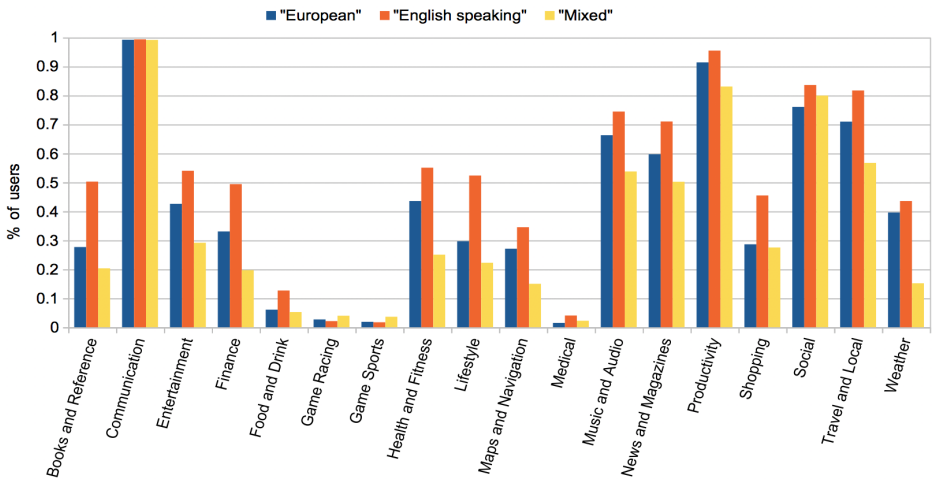


Figure 5.5: Usage of three main country clusters in several statistically significant categories. Previously published [92].

Saudi Arabia (sa), Qatar (qa), Pakistan (pk) and India (in) in Asia. Iran (ir), the Philippines (ph) and Estonia (ee) were not grouped close to other countries.

These three main groups visible in Figure 5.4 follow certain geographical, cultural, and language boundaries. These differences between application usage are also visible when looking at the application category data in greater detail. In Figure 5.5, we compare application category usage in certain categories strongly correlated with different cultural value factors. In general, the "English speaking" group uses a wider set of applications, and it seems to be statistically significantly high in almost every application category.

The "Mixed" group is characterized by lower application usage across the board, but higher than the other groups in two categories: *Sports* and *Racing games*. Some categories, such as *Food and Drink*, *Medical*, and *Shopping* are almost equally popular in both of the groups "non-English European" and "Mixed", but surpassed by the "English-speaking" group. *Weather* apps are, on the other hand, popular in the groups "non-English European" and "English-speaking", but less used in the "Mixed" group. That may be because the weather clearly is more predictable in some areas than others.

Communication apps are very popular in all the groups. Although the "Mixed" group has low usage in most categories, it also has very high

usage of the *Productivity* and *Social* applications. On the other hand, the "English-speaking" group has the highest usage in almost all categories. This may be due to the fact that almost all apps have an English version, and many services, retailers, restaurants, and public places in Europe and the USA have dedicated apps[3].

Demographic, cultural, geographical, and other differences in application usage may be utilized by marketing, social research, and many other areas. Later in Section 6.3 we analyze deeper these differences with different use cases, and compare application usage to the existing cultural factor model.

[3] McDonalds France is available in English: `https://play.google.com/store/apps/details?id=com.md.mcdonalds.gomcdo&hl=en`.
The city of Wien has a dedicated mobility app: `https://play.google.com/store/apps/details?id=at.wienerlinien.wienmobillab`.
Hyde Park club dedicated app: `https://play.google.com/store/apps/details?id=com.hydepark`.
The Finnish weather map in English: `https://play.google.com/store/apps/details?id=com.nordicweather_sadetutka`.
Sydney's Central Park has an app to aid sightseeing: `https://play.google.com/store/apps/details?id=com.beaconmaker.android.centralpark`.

Chapter 6

Decision Making and Actionable Recommendations

Understanding smartphone usage as a whole can provide novel insights to the user's needs for smartphone functionality. The most important target here is to provide the users with the best smartphone usage experience possible. Figure 6.1 provides an example of questions that need to be answered to improve smartphone utility. In terms of performance and energy-efficiency, there are choices to do for the network connectivity: mobile data or Wi-Fi connection? When can the user charge the device next time, and what is the overall condition of the battery? How many applications has the user installed, and how many of them are actually in use? What needs does the user have, and which applications meet these needs the best?

The user's location and occupation may affect the set of application functionalities in need: local transportation apps to support commuting, the best utility apps for a more effective working life, and maybe some task management apps to balance work and leisure? There might be a favorite game even if it uses a lot of battery life, but the user favors continuing to play it. The users cannot always choose applications by themselves, but are supposed to use those that are popular in the community they are living in, for example, for networking and social media. And when there are new applications available in the app market, how can the user know whether they are worth installing?

To understand these needs and answer these kinds of questions, we need to focus on analyzing the context of the user - including their daily routines, functionalities they require from their smartphones, and so on - and the context of the device including, for example, the device model and operation system version as well as memory and CPU loads. Supporting the smartphone user experience requires understanding the battery life

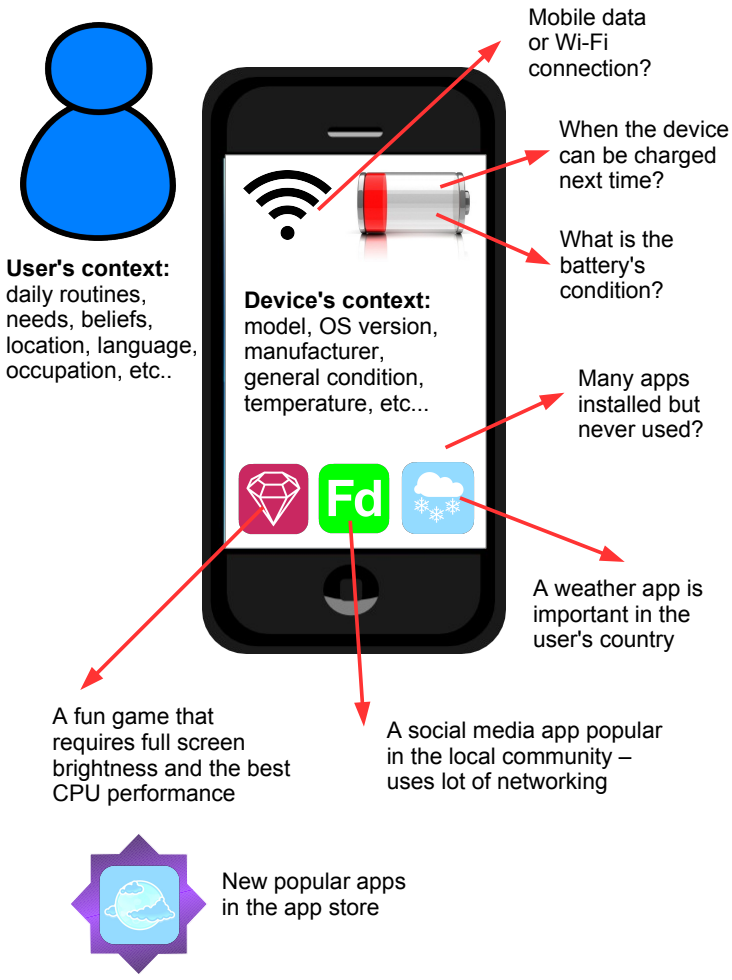


Figure 6.1: An example: Overall picture of the user's needs and smartphone usage as a whole.

limitations, user's needs, and in general the factors that have an influence on smartphones.

In this section, we present three example cases for making decisions and actionable recommendations based on the crowdsensed mobile data that has been collected in the Carat project. In every case, the methodology presented in this thesis for data cleaning and preprocessing as well as machine learning and statistical analysis of the results, are required. These examples aim to provide actionable feedback and new insights about the smartphone usage based on the crowdsensed data analytics. These example cases, with corresponding publications, can be listed as the following:

- **Energy recommendations for system settings and subsystem variables.** Many context factors of smartphones affect the device's energy consumption, but there are often complex interdependencies between them, which make it difficult to determine the optimal energy saving policy and right settings for a given situation. We present a system called Constella to provide actionable and human-readable energy recommendations for system settings and subsystem variables. The analysis work has previously been published in the attached Publication I [46] and the decision tree-based recommendation system in the attached Publication II [47].

- **Application trend analysis.** Understanding what happens after an application has been installed to the device, helps us to value the potentiality of applications. We present a novel app-usage behavior trend measure that provides instantaneous information about popularity of applications. Based on the application trends, traditional app recommendation systems can be evaluated and improved. The work is based on the attached Manuscript I [61].

- **Demographic, geographic, and cultural effects on mobile application usage.** Smart devices and functionalities they provide are nowadays an integral part of everyday life and part of modern life. Based on the user questionnaire performed for the Carat users, we study demographic, geographic, and cultural effects on smartphone usage. In addition to this, we also propose a method to use application usage data as a modern cultural factor. Understanding demographic factors, geographic differences, and cultural boundaries in application usage supports application developers, social researchers, and other people involved in the application ecosystem. The work is based on the attached Manuscript II [92].

6.1 Energy Modeling of System Settings

The processing and transmission power of smartphones continues to grow [98], while their battery technology remains largely unchanged [99]. Consequently, energy efficiency remains a high priority for current smartphone operating systems, and increasingly, for applications. The importance of energy efficiency has also been highlighted in several user studies, which have shown that users actively take measures to optimize the power consumption of their device [89, 100, 101]. Understandably, longer battery life provides better user experience and less struggle to find out the next charging possibility.

Mobile users are often forced to actively seek countermeasures to prolong the lifetime between successive recharges [33, 36]. Examples of these countermeasures include killing battery hungry applications or tasks, and manipulating context factors either through switching off specific sensors or adjusting individual system settings. Previous research has predominantly focused on the former task [6, 37, 38, 39]. In our previous work [46, 47], we demonstrate how the crowdsensed data-analysis approach can be used to obtain new insights into battery consumption. Especially, we provide a novel method to measure the energy effect of the combinations of different system settings and subsystem variables.

To demonstrate our methodology, we next provide an analysis of the energy effect of certain selected context factors. We have selected CPU use (Low, Medium, or High) and temperature (over or under 30°Celsius) from subsystem variables, and distance (motion or stationary) and screen brightness (automatic or manual) from system settings. Preprocessing of these context factors have been discussed in Sections 4.2 and 4.3. In all cases of the example, the network connection type has been a cellular data connection. Table 6.1 presents the estimated time in hours to drain the battery from 100% to 0%, while actively using a smartphone with the given context factor and value combination. With different values of CPU use, battery temperature, movement, and screen brightness, the battery life time ranges from 3.45 to 9.12 hours.

Table 6.1 demonstrates that the main deciding factor for battery life is the temperature: the lower the temperature, the longer the battery life. Traveling instead of staying still seems to increase battery life. This may be due to users driving and not using their mobile phones. After these factors, the CPU is the most dominant, and changing screen brightness brings the smallest, but still significant, battery life differences. These results show that while the CPU use alone is a good indicator of energy consumption, significant battery life gains can be obtained by considering more complex

Battery temp.	**Distance traveled**	**CPU use**	**Screen brightness**	**Battery life (h)**
Under 30°C	>0	Low	Automatic	8.83 – 9.12
Under 30°C	>0	Low	Manual	8.49 – 8.82
Under 30°C	>0	High	Automatic	8.09 – 8.24
Under 30°C	>0	Medium	Automatic	7.65 – 7.89
Under 30°C	>0	Medium	Manual	7.34 – 7.60
Under 30°C	>0	High	Manual	7.27 – 7.41
Under 30°C	None	Medium	Automatic	6.57 – 6.64
Under 30°C	None	Low	Automatic	6.28 – 6.35
Under 30°C	None	Medium	Manual	6.13 – 6.20
Under 30°C	None	Low	Manual	5.88 – 5.96
Under 30°C	None	High	Automatic	5.78 – 5.82
Over 30°C	>0	Low	Automatic	5.08 – 5.22
Under 30°C	None	High	Manual	5.00 – 5.04
Over 30°C	>0	Low	Manual	4.73 – 4.88
Over 30°C	>0	High	Automatic	4.62 – 4.69
Over 30°C	>0	Medium	Automatic	4.59 – 4.70
Over 30°C	>0	Medium	Manual	4.28 – 4.39
Over 30°C	None	Medium	Automatic	4.25 – 4.29
Over 30°C	>0	High	Manual	4.08 – 4.14
Over 30°C	None	Medium	Manual	4.06 – 4.09
Over 30°C	None	Low	Automatic	4.02 – 4.06
Over 30°C	None	High	Automatic	3.91 – 3.94
Over 30°C	None	Low	Manual	3.74 – 3.78
Over 30°C	None	High	Manual	3.45 – 3.46

Table 6.1: Estimated battery life in hours for selected combinations of four context factors. Previously published [46, 47].

context factor combinations. In addition to this, battery temperature and distance traveled can be used together to predict battery life very well.

The complex combinations of the context factors, such as those listed in Table 6.1, can be used to decide which factors to change to improve battery life, while keeping others constant. For example, while moving and playing a game, the CPU use is often high. If the phone can be kept relatively cool, 78% more battery life can be expected compared to warmer battery (increase from 4.08h to 7.27h). Further savings can be obtained by switching screen brightness from manual to the automatic setting.

Based on these observations, we deliver a recommendation system that can effectively summarize relationships between the context factor combinations and present transmission from a system state to another in a comparable easy manner. Constella [47] is a novel recommendation system for system settings and subsystem variables.

Constella relies on a decision tree-based recommendation model for capturing the energy impact of different context factors at once. Decision trees have been shown to provide a user-friendly and understandable representation for complex relationships [102], which is essential for improving users' trust in the recommendations. The decision tree model also provides a compact and compressible representation of relevant information which can be efficiently stored and used on a smartphone without a considerable impact on battery life.

The decision tree organizes context factor combinations into a logical structure and turns them into human-readable and actionable recommendations. The tree model can be learned efficiently on the cloud-computing back-end, and sent back to each client device. The clients can then generate recommendations independently by following paths of potential system state changes within the decision tree model. This makes it possible to generate energy recommendations also offline whenever the usage context changes: indoors and outdoors, with or without network connectivity, and so on. In the future, separate decision trees can be computed, for example, in the case of different applications or application combinations used.

Figure 6.2 presents an example of the Constella decision tree. Implementation details and more detailed examples are provided in Publication II attached to this thesis [47]. The example tree splits for three context factors: first the network type that splits the data into three parts. For each sub-branch, we can calculate an expected value of battery lifetime (EV). On the second level, there is a split by screen brightness after the network type "mobile" and by distance traveled after network type "Wi-Fi". If the device currently remains connected to the Internet via the mobile network with manual screen brightness, we can find at least two one-step changes to consider: if EV4 is better than EV5 (the current expected value), we can suggest switching to automatic screen brightness instead of the manual setting. If EV2 is also better than EV5, we can also show the recommendation for changing the network type. With more than one step, we can also go deeper in the tree, depending on the size and depth of the tree.

To remain clear these examples present only a limited number of the context factors. In reality, the trees will have more splits and options

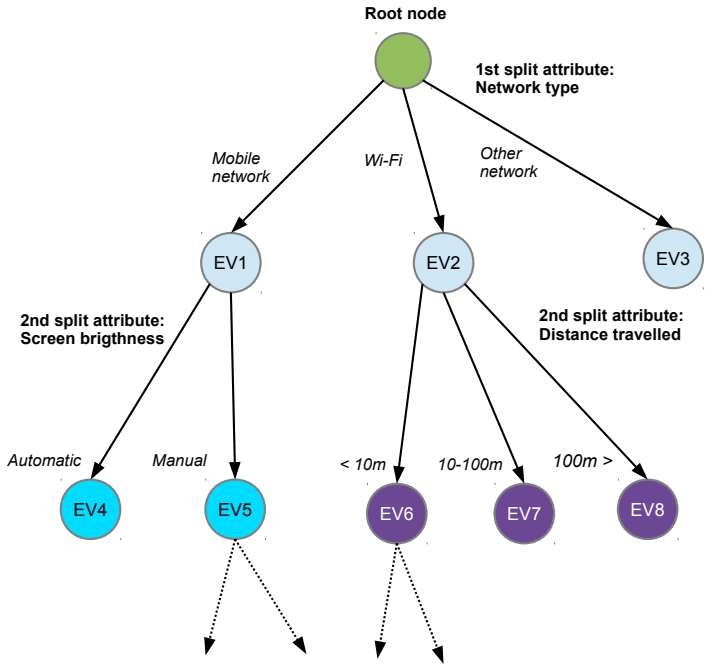


Figure 6.2: Example of the decision tree used for energy recommendations. EV = Expected value of battery lifetime in a given node. Previously published [47].

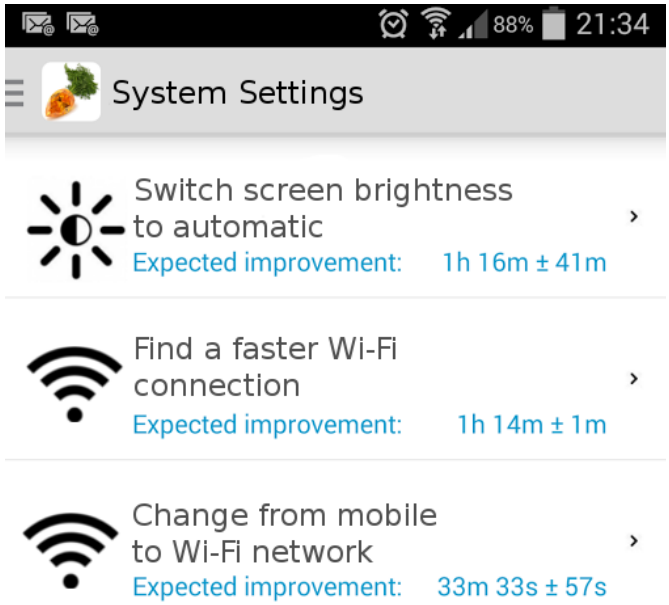


Figure 6.3: An example of the context factor energy recommendations. Previously published [47].

available. Figure 6.3 shows how the system setting and subsystem variable recommendations might be seen in the user interface. By adjusting the settings as suggested, users also participate in the continuous feedback loop with new and more informative data items to gather.

6.2 Application Trend Based Recommendations

Next, we focus on one of the fundamentals in the smartphone functionality: the applications. Choosing the right application for the right purpose is an important open research question. More generally, choosing an application that has any future in terms of upcoming new versions, security updates, and development support, is crucial. Our analysis of application trends focuses on app life cycle that characterizes usage behavior as discussed in Section 5.2. Next, we consider how trend information can affect application recommendations. The work is attached to this thesis as Manuscript I [61].

We implement the Slope One Prediction model that compares a user's profile to other users with similar usage history [103]. Slope One is considered to be a representative example of current state-of-the-art app recommenders,

and also other popular systems based on a closely similar approach [55, 104].

The Slope One Prediction and especially its well-known implementation in the AppJoy system [103] operates on so-called usage scores, which are constructed by aggregating the following information: (i) time elapsed since the last interaction with an app, (ii) frequency of the user interactions with an app, and (iii) total duration of time the user has interacted with an app. Due to the infrequent sampling period of the Carat application, we focus on the amount of interactions, in other words, how often the application has been seen in the user's sample history.

Recommendation model. The Slope One Prediction model compares a user's profile to other users with similar application usage history. Formally, we define $S(u)$ as the set of applications used by user u. Given an application i and user u, we define $R_{u,j}$ as the set of relevant applications j used by other users together with i, in other words, $R_{u,j} = \{i | i \in S(u), j \notin S(u), \#S_{i,j} > 0\}$ where $S_{i,j}$ is the set of users who have used both i and j. The relevance of application j for user u is then given by:

$$P(u_j) = \frac{1}{size(R_{u,j})} \sum_{i \in R_{u,j}} (dev_{i,j} + u_i). \tag{6.1}$$

Here *dev* is the average of the usage scores between users who have used both i and j:

$$dev_{i,j} = \sum_{w \in S_{i,j}} \frac{v_{w \vdash j} - v_{w \vdash j}}{size(S_{j,i})}. \tag{6.2}$$

Given the relevance scores $P(u_j)$, the algorithm returns the top$-N$ items with the highest score as recommendations.

We run the Slope One-based recommendation system together with our trend filter analysis for a subset of the Carat data containing $4,500$ users and their $1,000$ most frequently used applications. We select October 2014 to be the test period due to little seasonal fluctuations, and as training data we selected all the Carat data accumulated between January 2014 and September 2014. Given the test data, we used the Slope One to generate recommendations in an incremental fashion for each of the four weeks in October 2014.

We counted (i) how many recommended applications can be classified as *Flop* or *Hot* apps by the trend filter analysis, and (ii) how they compare with the total number of the *Flop* and *Hot* apps. We also calculated

Week	Rec. Hots	Rec. Flops	Total Hots	Total Flops
1	8	5	219	163
2	7	6	229	158
3	8	7	232	154
4	10	9	225	150

Table 6.2: The *Hot* and *Flop* apps in the 20 best recommendations out of top 1000 applications during a month. Previously published [61].

the following evaluation metrics proposed in the previous literature [105]: temporal diversity, novelty, and accuracy. Diversity presents how the recommendations change over time, whereas novelty describes how many new recommendations there are seen compared to the later ones. The novelty of the recommendations relates closely to the trends, because changes in the application trends should affect new recommendations. Formally these metrics are defined as follows:

$$diversity(L_1, L_2, N) = \frac{|L_2 \setminus L_1|}{N} \tag{6.3}$$

$$novelty(L_1, N) = \frac{|L_1 \setminus A_t|}{N} \tag{6.4}$$

$$accuracy(L_1, A) = \frac{size(L1 \cap A)}{size(A)} \tag{6.5}$$

Table 6.2 presents an analysis of the top-20 recommendations given to all the users during the period of four weeks. In each row there is first the number of the week (from the beginning of October), and then in order the number of recommended *Hot* apps, the number of recommended *Flop* apps, the total number of *Hot* apps in the week, and the total number of *Flop* apps in the week. Table 6.3 gives statistics of diversity, novelty, and accuracy, first considering all the given recommendations and then without the *Flop* applications.

In Table 6.2 we can see that the number of *Hot* apps recommended for each week is small and comparable to the number of recommended *Flops*. Given that we have generated in total 90,000 recommendations for 4,500 users each week, the amount of *Hot* recommended corresponds to a very small percentage of the entire set of recommendations. More than 200 applications each week can be classified as *Hot*, and about 160 as *Flop*. On average, only 3.6% *Hot* apps are recommended, compared to 4.3% *Flops*.

Week	Diversity	Novelty	Accuracy	Div. w/o Flops	Nov. w/o Flops	Acc. w/o Flops
1	-	-	0.02	-	-	0.02
2	0.80	0.98	0.03	0.90	0.90	0.12
3	0.62	0.81	0	0.54	0.73	0.10
4	0.56	0.75	0.11	0.50	0.68	0.11

Table 6.3: Diversity, novelty, and accuracy statistics of the 20 best recommendations out of top 1000 applications during a month. Previously published [61].

Table 6.3 shows that when the *Flops* are removed from the recommendations, both novelty and diversity decrease, but accuracy increases slightly. The main reason for this behavior is that the metrics used to generate recommendations require a sufficient amount of usage before an app is recommended. However, once sufficient usage has been observed, the app can already be past its "best before" date as the recommendation model does not differentiate between the *Hot* and *Flop* apps.

Integrating usage trend information as part of the recommendation process can help to overcome this issue and improve the overall quality of the application recommendations. We suggest that, among others, the applications with the *Flop* pattern might be reasonable to remove completely, and the weight given to the *Hot* applications might be increased. Thus, we can warn users for using applications that might be losing their popularity and soon becoming less supported when their developers' focus changes to the new projects. Focusing on the *Hot* applications users gain the benefits of the applications with a strong user base: security updates and developing towards new features.

6.3 Insights into Demographic, Geographic, and Cultural Factors in Mobile Usage

Sometimes mobile applications are used not because they are popular in general, but because they are popular in the users' context: in their country or among their family and friends. On the other hand, not only the application's popularity affects the usage of the applications, but also the application functionality, and the user's background, needs, and desires. It is easily understandable that people of different background and demography consider different use cases more important than others.

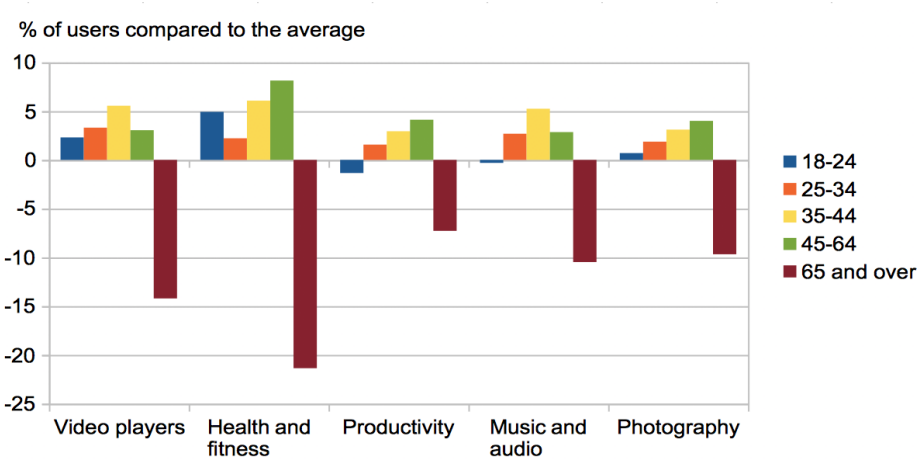


Figure 6.4: Comparison between different age groups.

Next, we show that indeed, demography, geography, and culture play an important role when considering application usage. Understanding these factors in the mobile application usage will help many stakeholders in their work, including application developers, social researchers, and other parties involved in cultural studies or mobile application ecosystem.

6.3.1 Demographic Factors

The user questionnaire run for the Carat user base (described in Section 3.3) provides background information to analyze in more detail how people of different age, education level, and so on use their smartphones. We call these features demographic factors on smartphone usage.

From the questionnaire data we choose four factors for more detailed analysis: age group, education, occupation, and household situation. Education and occupation gain a high mutual information in the analysis performed in Section 5.3.1, and the age and household situation are included because of the general interest. The country information gained the highest mutual information and it will be discussed next in Section 6.3.2. Out of all the answers, we consider those groups with ten or more responses. For example, there are hundreds of students and professionals, but only a few respondents staying at home with kids, or working in agriculture. There are 55 different application categories in Google Play. For convenience, the categories were sorted by the highest standard deviation among the usage of each answer group to highlight the differences between the groups.

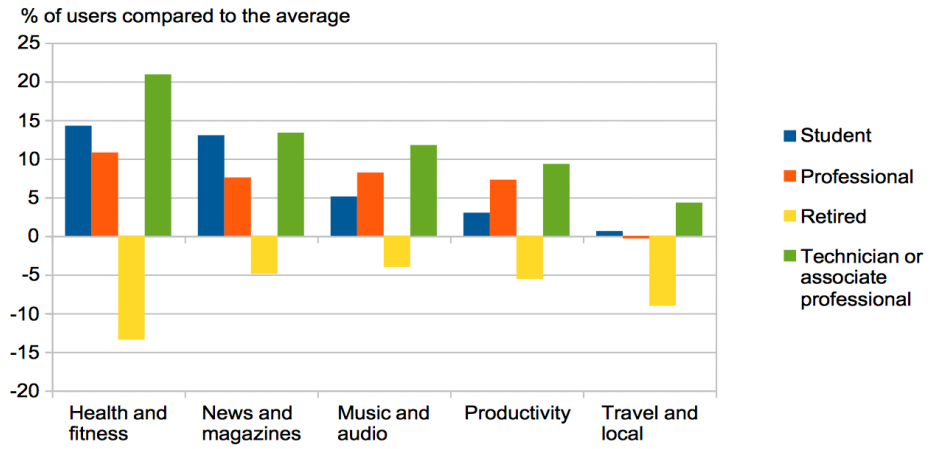


Figure 6.5: Comparison between different occupation groups.

Figure 6.4 presents the comparison between the age groups, all of them included: 18 – 24 (12% of respondents), 25 – 34 (30%), 35 – 44 (28%), 45 – 64 (27%), and over 65 years old (4%). The underage children were excluded from the study. The application categories considered interesting due to the standard deviation-based analysis are *Video players*, *Health and Fitness*, *Productivity*, *Music and Audio*, and *Photography*. The graph presents comparison the average usage, in percentages. Compared to the other user groups, elder people tend to use less of all the categories. Especially "trendy" categories, such as *Health and Fitness* do not gain popularity in this group. Elder people might be excluded from the marketing and target audience of these apps, even if caring for your health does not become less important with age. On the other hand, the health apps are the most popular in people of 45 – 64 years old. People of working age (35 – 44 years old) seem to be the most active users of *Video players* and *Music and Audio* - both categories that might be considered to gain their greatest audience from young people. It is possible that these kinds of applications provide relaxation during work days and thus gain their popularity in this age group.

Figure 6.5 compares different occupational backgrounds. Out of 13 possible choices in the questionnaire, we consider the following four groups: students (12% of answerers), professional (34%), retired (5%), and technician or associate professional (14%). The application categories considered in this case are: *Health and Fitness*, *News and Magazines*, *Music and Audio*, *Productivity*, and *Travel and Local*. Retired people follow the same pattern considered in the case of elder people in general: they seem to use less than

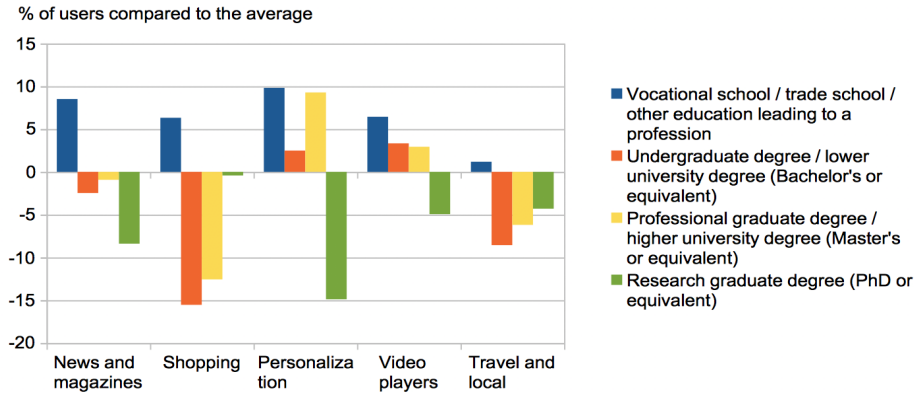


Figure 6.6: Comparison between different education backgrounds.

average in all the categories. Professional and associate professionals seems to use considerable amounts of the productivity applications, maybe to help in their work. Technicians and associate professionals have the highest usage of the traveling apps in comparison: maybe they gain more free time than students and professionals, and use it for more smartphone-oriented tasks compared to retired people.

The comparison between education levels is presented in Figure 6.6. The following groups are considered out of seven different options: vocational school or trade school or other education leading to a profession (11% of respondents), undergraduate or lower university degree (Bachelor's or equivalent) (35%), professional graduate degree or higher university degree (Master's or equivalent) (30%), and research graduate degree (PhD or equivalent) (5%). The application categories considered are the following: *News and Magazines*, *Shopping*, *Personalization, Video players*, and *Travel and Local*. Interestingly, people with vocational school or corresponding seem to use all the categories more than people with other educational backgrounds. Especially the *News and Magazines* and *Shopping* categories are more popular among them than the other groups. It seems that the highest educational group including PhD and corresponding use the *Shopping* applications as much as the average, but the lower university degree holders use them significantly less. On the other hand, having a PhD seems to reduce a need for *Personalization* apps, as well as *News and Magazines* and *Video players*.

Figure 6.7 presents the differences between household situations. There the following groups are considered: living alone (19% of respondents), living with other adult(s) (48%), living alone with under-aged kid(s) (30%),

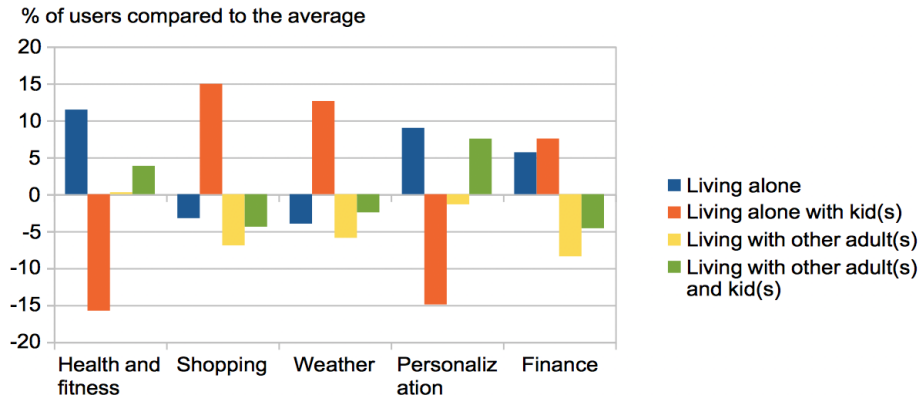


Figure 6.7: Comparison between different household situations.

and living with other adult(s) and kid(s) (3%). The following application categories are considered: *Health and Fitness*, *Shopping*, *Weather*, *Personalization*, and *Finance*. People taking care of their kids alone seems to prefer *Shopping* and *Whether* applications highly compared to the people in other household situations. At the same time, their time seems to be limited for the *Health and Fitness* and *Personalization* apps. Living alone without other adults seems to reduce a need for the *Finance* applications, maybe because there is no other person to help with financial issues. *Health and Fitness* are the most popular among people living alone, too.

6.3.2 Geographic Factors

Comparing demographic information across countries gives us an insight to application usage worldwide. Understanding how demographically speaking similar people - for example, people of the same age, education, and occupation - use their smartphones in different countries can provide important insights on geographic and cultural boundaries. As already discussed in Section 6.3.1, we study in more detail several demographic factors including occupation and education. We also include the household status to highlight some common, interesting clusters. Countries with less than 10 respondents to the questionnaire are excluded, leaving 21 in total out of 44 countries included in the comparative study in Section 5.3.2.

In Figure 6.8, we compare the four most widely represented occupations (student, professional, retired, and technician or assistant professional) within 21 countries. In Figure 6.9, we present a similar comparison between the best represented educational levels (education leading to a profession,

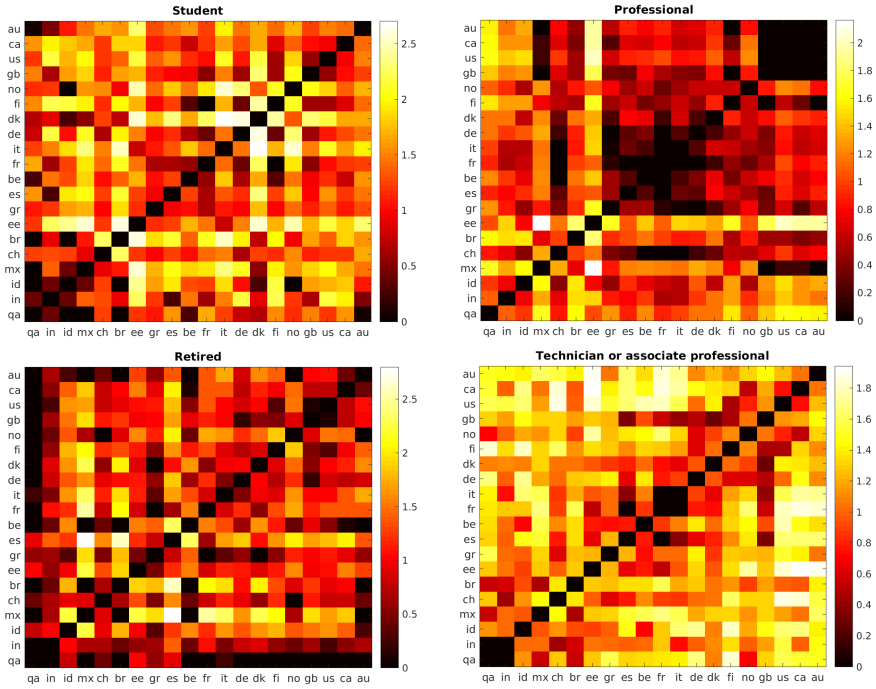


Figure 6.8: Comparison between different occupation groups in the selected countries. The colormaps are based on the KL differences.

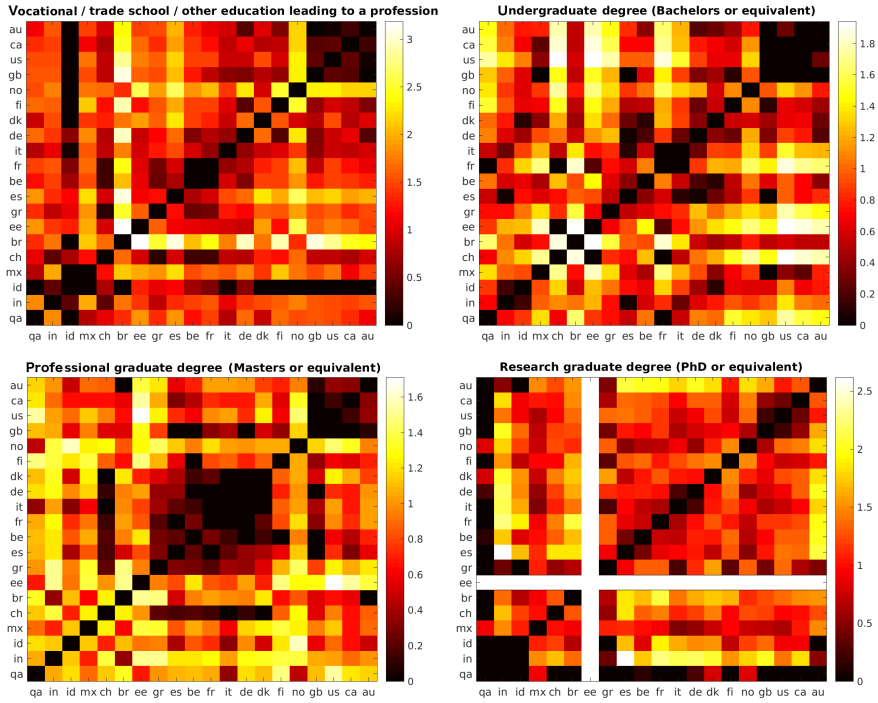


Figure 6.9: Comparison between different education backgrounds in the selected countries. The colormaps are based on the KL differences. As far as the PhD degree is concerned, the values for Estonia (ee) are missing.

Bachelor's degree, Master's degree, and PhD equivalent degree). In both figures, the darker color indicates closeness (the KL divergence between countries close to 0) and lighter color a longer distance (the higher KL divergence, see Section 5.3.2).

As seen in Figure 6.8, professionals in Australia, Canada, the USA, and the United Kingdom use application categories similarly, indicated as a dark cluster in the North-Eastern corner of the colormap. The same cluster is visible in all the educational groups in Figure 6.9, and we may conclude that highly educated people or those working as professionals seem to use their mobile devices similarly in these Western, English-speaking countries.

Another cluster is visible in the South-Western corner of the colormaps, including Qatar (qa), India (in), and Indonesia (id). Especially students and people with PhD or equivalent degree are presented in this cluster, indicating similarities in application usage of academic people in these countries. It is possible that these groups also have a higher smartphone penetration, and

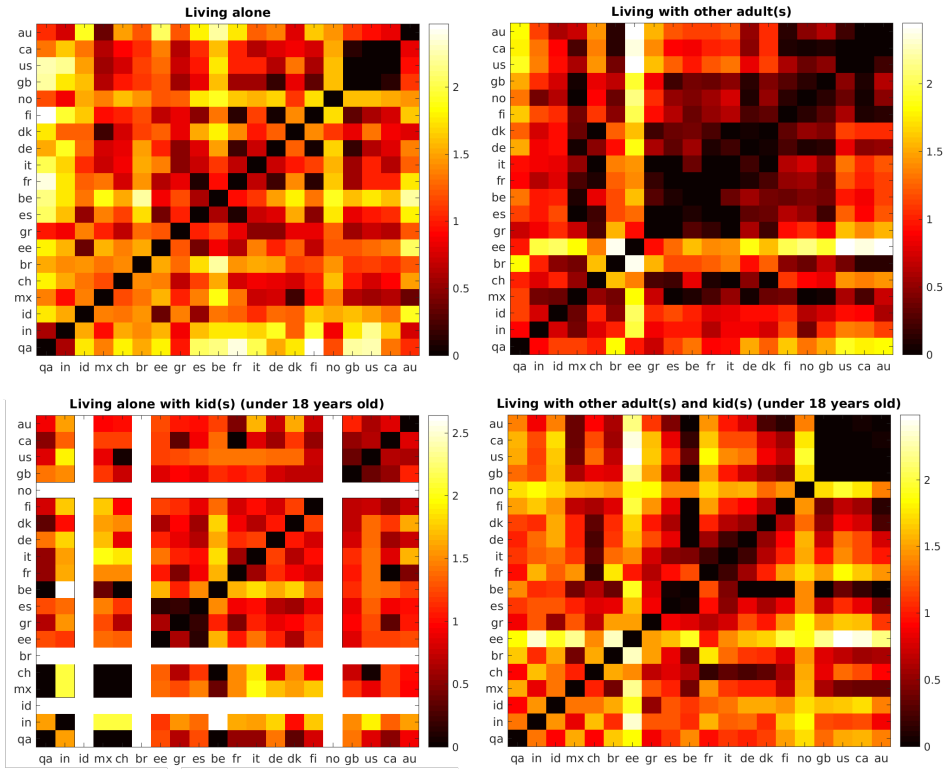


Figure 6.10: Comparison between different household situations in the selected countries. The colormaps are based on the KL differences.

the use of English may be required in studies. Highly educated people are often considered to be builders of rising societies, and they may adapt faster to new technologies such as mobile device functionalities in everyday use.

For professionals and Master's degree holders there is also a third cluster in the middle of the colormaps. This cluster includes European countries: Denmark (dk), Germany (de), Italy (it), France (fr), Belgium (be), Spain (es), and Greece (gr). The application category usage of this group is different from the previously mentioned English-speaking cluster, as also seen in Figure 5.4 presented in Section 5.3.2.

Interestingly, students seem to use applications differently in each country, as there are no clear clusters in the students' colormap. This might be because university students may travel to faraway countries to study, while, for example, vocational studies are commonly done in the same or nearby countries. The colormap of retired people is darker than the others overall, which means their application use is more similar through all the considered countries, but are few strongly similar clusters. This may be a result of people having used different sets of applications, not adopting new ones as a group. Technicians and associate professionals have the lightest colormap in comparison, indicating that the countries have the highest distances (and thus less overall similarity) in this occupational category. This may be caused by the wide range of actual professions and people with different smartphone needs in the category.

6.3.3 Cultural Factors

In addition to the demographic and geographic factors, we also study how culture affects mobile usage. Culture is a wide concept to define with some difficulties. The culture of an individual or group can encompass all aspects of life. For example, the Cambridge English Dictionary defines culture as "the way of life, especially the general customs and beliefs, of a particular group of people at a particular time." Elements of culture may include habits, rituals, and beliefs, as well as ways to perform everyday actions.

Cultural Value Model. In empirical research, Hofstede's Cultural Values Model (VSM) [106, 107] is used with wide variety to represent cultural values between countries [108]. The VSM model consists of six factors, given by a country, that are made by questionnaire studies in different countries around the world.

The VSM model has been previously used, for example, to study culture in IT corporations [109], evaluate tourist services [110], study international ethics [111], evaluate consumer decision making [112], analyze Doodle

scheduling responses [113], and model emoji usage in different countries [114]. The VSM model is not free from criticism. Especially, McSweeney [115] questions the validity of defining culture boundaries based on politically agreed national areas. Also, the model does not include minorities or subcultures inside countries, or take into account immigration and emigration in the global world, previously referred to as transnational mobility [116]. These need to be taken into account when analyzing the results: assumptions can be made only to present the measured cultural values, leaving the larger picture of culture harder to capture.

In our work attached to this thesis as Manuscript II [92], we evaluate differences between countries by comparing mobile usage to the six VSM factors[1], described in the following:

- **Power distribution (PDI)** describes whether unequal power distributions are expected and accepted in the population. Cultures with higher power distribution tend to be more hierarchical and persist more inequalities compared to the cultures with lower power dimension.

- **Individualism versus collectivism (IDV)** describes how much members of the population are supposed to take care of themselves or stay integrated to the group, such as family. In cultures with high individualism people define themselves as "I", compared to the stronger "we" feeling in countries with lower individualism.

- **Masculinity versus femininity (MAS)** describes strength of masculine and feminine roles in the population, for example, in working life. Cultures of high masculinity are more competitive, compared to the lower masculinity (higher femininity) that stands on collaboration and modesty.

- **Uncertainty avoidance (UAI)** describes whether members of the population feel either comfortable or uncomfortable in new, unstructured, or unpredictable situations. A high level of uncertainty avoidance implicates stricter codes of planning and caring for the future, compared to more relaxed cultures of the lower score of this factor.

- **Long versus short-term orientation (LTO)** describes how members of the population accept delays in either social, material, or emotional gratification. Cultures with a high score of this factor are more future-planning compared to those that score lower.

[1]The open-sourced VSM data matrix is available in: `http://www.geerthofstede.nl/dimension-data-matrix`

- **Indulgence versus restraint (IVR)** describes whether any gratifications are allowed to be relatively free (having fun by themselves) or regulated by strict norms of the population. A high indulgent score reflects higher importance of free leisure time compared to restraint cultures of a lower indulgence score.

Several methodologies to clean and process the application usage data have already been given in this thesis: Section 4.5 describes the process to collect the country information, Section 4.6 how application data is collected and cleaned, and Section 4.7 how application categories are delivered. In Section 5.3, we present how application usage data is converted to the usage vectors that are more flexible to process with different statistical methods. Countries' usage vectors are considered as the average of users, belonging to the given country, that have used a certain category.

Next, we correlate the usage of all the application category and VSM factor pairs separately. Table 6.4 summarizes the results and lists the categories that have the highest positive or negative correlation for the VSM factors.

Table 6.4 shows us several findings. For example, a low power distance that indicates low hierarchy in the culture, correlates significantly to the use of *Entertainment* applications and other leisure-related categories, such as *Travel and Local*, *Sports*, and *Music and Audio*. These same categories together with, for example, *Health and Fitness* are mostly related to individualist cultures.

Collectivist cultures, those with higher power distance, and cultures considered feminine seem to value family related categories, such as *Family create*, *Education games*, and *Family pretend*. Masculine cultures correlate with high use of *Personalization* apps. Long-term-oriented cultures seem to prefer *Sport, Casual* and *Word games*, as well as *Social* apps. In short-term-oriented cultures, there is a preference for *Role playing games* and a need for *Weather* apps as well as *Comics*.

It is noticeable that categories with high correlations differ from those with the highest usage in general (as presented in Figure 4.2), indicating that the differences are more sophisticated and complex by nature. A similar correlation analysis can also be performed reversely. There are nine categories that correlate less than 0.2 (or -0.2, similarly) to at least five VSM factors. The category *Dating* correlates slightly more (0.26) only to the Individualism versus collectivism, and the category *Events* to the Masculinity versus femininity (0.21). *Game role playing* has very low impact in five categories, but gains more than 0.3 correlation to Long versus short-term orientation. In addition to these, the category *Beaty* and a list of

	Power distance (PDI)
ρ	*Categories*
$<$ -0.5	Music & audio, Entertainment, Weather
$<$ -0.4	News & magazines, Productivity, Travel & local, Sports, Libraries & demo
$<$ -0.3	Game trivia, Photography, Finance, Communication, Auto & vehicles, Game card
$>$ 0.3	Family create
$>$ 0.4	Game action
	Individualism versus collectivism (IDV)
ρ	*Categories*
$<$ -0.4	Family create, Game action
$<$ -0.3	Game education
$>$ 0.4	Books & references, Photography, Libraries & demo, Education, Finance, Game words, Medical, Family music video
$>$ 0.5	Auto & vehicles, Productivity, Sports
$>$ 0.6	Weather, News & magazines, Travel & local, Health & fitness, Music & audio, Entertainment
	Masculinity versus femininity (MAS)
ρ	*Categories*
$<$ -0.4	Family pretend
$<$ -0.3	Game board
$>$ 0.3	Personalization
	Uncertainty avoidance (UAI)
ρ	*Categories*
$<$ -0.4	Parenting, News & magazines, Family music video, Game words
$<$ -0.3	Education, Family education, House & home, Entertainment, Books & references, Family brain games
$>$ 0.3	Family create
$>$ 0.4	Game action
	Long versus short-term orientation (LTO)
ρ	*Categories*
$<$ -0.4	Game sports
$<$ -0.3	Family music video, Game word, Social, Game casual
$>$ 0.3	Maps & navigation, Game role playing
$>$ 0.4	Comics, Weather
	Indulgence versus restraint (IVR)
ρ	*Categories*
$>$ 0.3	Sports, Photography, Communication, Game words
$>$ 0.4	Music & audio, Family music video, News & magazines, Entertainment, Books & references

Table 6.4: The best category correlations to VSM factors with 44 countries.

different game categories gain low correlation to every VSM factor. Indeed, there are certain categories that are in general more independent from the cultural model than the others. These categories can provide us with insights to the applications that are similarly important through all the studied countries.

To summarize, mobile usage reflects geographic, demographic, and cultural boundaries and at the same time, it cannot be truly explained only by those societal and cultural factors. We propose mobile usage as a novel societal factor to consider in future studies that apply smart devices worldwide.

Chapter 7

Conclusions

7.1 Summary of the Main Findings

This thesis has proposed methods and approaches to clean, analyze, and utilize crowdsensed mobile data for actionable feedback and human-readable recommendations. To summarize the main findings of the work, this section revisits the research questions provided in the beginning of the thesis in Section 1.2. Those research questions and their proposed, summarized answers are the following:

RQ1. **How do different data attributes have to be cleaned and pre-processed to produce a reliable picture of the system state?**

There is a need for cleaning mobile crowdsensed data before it is used as an input of the analysis systems: the data contains misreadings, missing values, manufacturer-specific default values, and other items that have to be considered in more detail. We show that using natural thresholds and statistical analysis of the items' value ranges we can include the valid data items in the analysis set and remove invalid ones. The cleaning procedure has to be separately defined for each crowdsensed data attribute, and that is where understanding the collected data becomes so crucial.

RQ2. **How can crowdsensed data be used to present crucial factors of a smartphone's system state?**

Our work with system setting and subsystem variable analysis has shown that these context factors, indeed, have a crucial effect on smartphone energy consumption. Our findings are in line with the

previous literature in the case of single context factors, and our results are possible to validate with the laboratory measurements, too. Indeed, the crowdsensed data provides new insights to the real-life use cases that are not even possible to model in limited laboratory conditions.

RQ3. **What are the effects of subsystem variables, system settings, and their combinations to smartphone energy consumption?**

We have shown that not only single factors affect smartphone energy consumption, but the crowdsensed data reveals that the combinations give even more detailed insights to the energy-hungry system settings and subsystem variables. We show that the most accurate energy impact is revealed when the system state of the device is analyzed as a whole, taking into account all the possible combinations that system settings, subsystem variables, and running applications can combine. In this kind of analysis, the crowdsensed data provides valuable new insights and gives possibilities to look for an almost unlimited number of real-life system states compared to the more dependent laboratory environments.

RQ4. **How can smartphone energy consumption be improved by recommending better system state and subsystem variables?**

We propose a novel energy recommendation system Constella that can take into account the whole system state of the device, including subsystem variables, system settings, and applications. Most importantly, the combinations of these context factors can be covered in the effective, decision tree-based approach. The Constella recommendation system relies on the concept of the continuous feedback loop where the data items are collected from the crowd, processed and analyzed in the back-end cloud-computing environment, and the value of the results is then sent back to the devices as human-readable, actionable recommendations. Thus, the value of the analysis will be returned to the sources of the data, also, to benefit of the future learning loops.

RQ5. **How can mobile recommendation systems be improved by analyzing application popularity?**

We suggest that the trend filtering methodology can be used to improve current application recommendation systems. We show that the traditional recommendation systems tend to favor also applications that are already falling in popularity. We show how the trend filters can be used to improve recommendation results by filtering out the *Flop* applications that are already losing their popularity or increasing the weight given to the *Hot* applications that show their potentiality by gaining a significant user base fast.

RQ6. **What can be learned about mobile application usage and popularity in real-life crowdsensed data?**

Understanding application usage in the wild provides several new insights in how applications become popular or fall in popularity. Our trend-filtering approach suggests a novel methodology to represent popularity in addition to the traditional retention rates used by the marketing analysis companies. We can characterize applications as *Hot*, *Flop*, *Marginal*, or *Dominant* based on their lifetime success in the crowd of mobile devices.

RQ7. **How does mobile application usage reflect differences in user population?**

We compare crowdsensed mobile application usage to the existing Cultural Value Model and find out that, indeed, there are correlations between mobile application usage and known cultural factors. Our study suggests that in the future, mobile application usage could be seen as an additional demographical factor or at least as a reflection of local societies. Our approach can potentially help different social research areas and application developers targeting international markets.

RQ8. **What can be learned about cultural, demographical, and geographical differences in crowdsensed smartphone usage?**

In addition to the knowledge of how applications are used as a whole, different demographic, geographic, and cultural factors have shown

to have a significant effect to the mobile usage patterns. We study worldwide crowdsensed application usage and show that, indeed, there are differences and similarities between certain areas. For example, we can categorize 44 countries into three groups, including mainly the English-speaking countries, the continental European countries, and the mixed group of various Asian and Middle-Eastern countries. Our research suggests that in addition to the demographic factors, such as age group, gender, occupation, or education, the country is an essential source of information when studying mobile usage in the wild.

7.2 Implications of the Research

The research in this thesis has shown that the crowdsourced data can and should be utilized in the cases where previously only laboratory measurement would be considered. Especially when studying real-life effects and use cases in the wild, the large-scale crowdsensed mobile data provides essential insights impossible or too expensive to model alone in laboratory conditions. This is especially important in an energy consumption perspective, where various different usage situations need to be included in reliable models.

The crowdsensed approach allows modeling real-life system state combinations and reveals important interdependencies between different context factors. This is especially seen in our energy modeling work, where we show that system setting and subsystem variable combinations provide more complex information about the battery life and overrun any single sensor-based approaches to understanding mobile energy consumption.

The crowdsensed mobile data provides a possibility for independent, large-scale studies that are not related to the marketing companies or mobile manufacturers. For example, we have presented an independent study of retention rates that provides information about application popularity. We suggest novel methods to better understand application popularity and trend patterns, and mobile usage all around the world, which provides important insights for multiple parties involved in the mobile ecosystem.

We have shown that mobile application usage, indeed, reflects demographic, geographic, and cultural factors, which is crucial for application developers to take into account when targeting their products worldwide. There are several clear design implications: Understanding the target audience, their needs, and habits regarding the mobile usage helps to design suitable system features. Knowing the popularity of certain categories in different areas eases the definition of general terms for the applications and

make them easier to find. Understanding cultural differences in application use helps to both target or generalize applications in the worldwide market.

7.3 Limitations

Even if the crowdsensed approaches can provide beneficial information to utilize in many application areas, there are still existing challenges. Even if many easy to use systems have been introduced, in most of the cases, the best suitable crowdsensing application requires mobile development skills and understanding of large-scale computing paradigms and distributed machine learning algorithms. Not every researcher or developer has time or opportunity to study all the necessary skills.

When it comes to mobile crowdsensing, data cleaning still has challenges not fully solved. For example, every new addition to the context factor set has to be studied independently to understand its natural value thresholds and statistical distributions. When new data items have been collected and, for example, new and more effective device models introduced, also the statistical distributions and features used in the old models may become dated. This means not only the learning phase but also the data cleaning procedures have to be updated regularly.

The data collection itself sets its own challenges. Even if it is known that the crowdsensing systems can provide large amounts of data in a comparably short period of time, there still exists the cold start problem to first gain the user base and then get enough data to start the analysis phases. Many existing machine learning applications are based on previously collected data sets and only rarely fully online-based learning systems are proposed. Without sufficient training data, starting a new crowdsensing project and implementing necessary machine learning procedures may be difficult.

User acquisition has often been seen as an important challenge where there really is no silver bullet to solve it. The Carat system relies on energy recommendations it gives back to the users as an additional value. The truth still seems to be that not many people want to participate in research projects without any other benefit. Giving out money or gift cards may be out of the budget for many research teams, and even then, gaining a representative user population might be challenging. First, when considering mobile crowdsensing, only users with an appropriate smartphone can be studied. There is always a group of people left outside, for example, in the case of the Carat project, only Android and iPhone users can be considered and even there, iOS provides significantly less information out of the device compared to the Android system.

Second, people owning a suitable smart device and volunteering to participate in the research task do not necessarily correspond to the full population using these devices. Based on the user questionnaires run for the Carat population, there is a high bias towards well-educated men working in professional occupations. Women and less educated people consist of a clear minority. Interestingly, the age group seems to be the least biased attribute, because even elder people are well represented in the Carat questionnaires. Without fully working recruiting strategies, it seems hard to gain a well representative user population.

Crowdsensing smartphone data gives a great responsibility to researchers managing such privacy-sensitive information. Users should trust the policies and storage strategies involved in the analysis process, and in the case of industrial applications, also trust that no information learned from the data is used harmfully or unpleasantly. Privacy and security questions still need to be covered in more detail in the future.

7.4 Future Work

This thesis has presented several use cases for crowdsensed mobile data analytics. On the other hand, there are still open questions and novel application areas where the mobile devices and the Carat data can be utilized. For example, the energy analysis can be enlarged to cover combinations of different applications together with system settings and subsystem variables. The Constella recommendation system should be analyzed in the wild with its recommendations sent back to the user community. Thus, the real-life effects of this kind of recommendation systems could be tested and evaluated.

This thesis focuses on a single energy decision tree constructed from the entire dataset, but it could be more beneficial to consider each device model separately, or based on the user profiles. Together with understanding the usage context and application usage history, also the energy recommendations could be improved even further. For example, the network infrastructures differ between countries and different user populations have different needs for their smartphones, so it is reasonable to consider that also their energy profiles vary. Personal energy plans might be one of the next topics to investigate. Characterizing users and their needs for their smartphones could be used to generate more accurate application recommendations.

The demographic, cultural, and geographical analyses of this thesis have focused on a limited number of data attributes available in the questionnaire run for the Carat user base. More information about the user context could

be collected, including but not limited to, for example, the user's personality traits, mental state, and daily habits and routines. From a cultural point of view, also different personal beliefs, religion, and political opinions might be considered, as well as identification with minorities and subcultures. There is previous work about these topics, but the Carat user base provides a special opportunity to study these topics worldwide with real users and usage cases in the wild.

The user context analysis together with understanding of global and local application trends provide a rich input also for new application recommendation systems. The current trend analysis presented in this thesis focuses on the global trends, but in the future also local trends and popularity inside certain demographical subgroups or geographical areas would provide fruitful results.

7.5 Conclusion

This thesis has presented how crowdsensed mobile data can be used in benefit of energy diagnosis, application popularity analysis, and demographic insights. Smartphones have become a crucial part of modern everyday life and it is clear that they and corresponding new smart devices in the future will continue this trend. People have become used to being connected and relying on a single device of multiple integrated functionalities for fun and leisure as well as for work and education.

The crowdsensed data can be utilized for several application areas, but it still also has challenges to solve. For example, the autonomous data collection requires intensive cleaning strategies for functional information retrieval. Large-scale data collection sets challenges for running effective machine learning procedures and statistical analysis also in the cloud-computing environment. Analysis results of the learning algorithms have to be converted to human-readable form and visualizations to fully utilize their value in the future. For example, recommendation systems and decision-making tools can utilize crowdsensed data effectively.

This thesis has proposed methodologies to collect, clean, analyze, and form the value out of the crowdsensed data. As an opinion of the author, there are more pros than cons in the crowdsensed mobile data analytics. The open challenges of the field only help to make applications and methodologies stronger and easier to utilize in the future.

References

[1] Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. From participatory sensing to mobile crowd sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 593–598. IEEE, 2014.

[2] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones. *Biological Cybernetics*, 24(3):1732–1744, 2016.

[3] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11), 2011.

[4] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[5] Daniel Hintze, Philipp Hintze, Rainhard D. Findling, and René Mayrhofer. A large-scale, long-term analysis of mobile device usage characteristics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1(2):13:1–13:21, June 2017.

[6] Adam J. Oliner, Anand P. Iyer, Ion Stoica, Eemil Lagerspetz, and Sasu Tarkoma. Carat: Collaborative energy diagnosis for mobile devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 10:1–10:14, New York, NY, USA, 2013. ACM.

[7] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device Analyzer: Large-scale mobile data collection. In *Big Data Analytics workshop (in conjunction with ACM Sigmetrics 2013)*. ACM, 2014.

[8] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device analyzer: Understanding smartphone usage. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 195–208. Springer, 2013.

[9] Andrew Rice and Simon Hay. Decomposing power measurements for mobile devices. In *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 70–78. IEEE, 2010.

[10] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010)*, pages 179–194. ACM, 2010.

[11] Hien Truong, Eemil Lagerspetz, Petteri Nurmi, Adam Oliner, Sasu Tarkoma, and N. Asokan. The company you keep: Mobile malware infection rates and inexpensive risk indicators. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 39 – 50. ACM, 2014.

[12] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Morley Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference*, IMC '11, 2011.

[13] Pascal Welke, Ionut Andone, Konrad Blaszkiewicz, and Alexander Markowetz. Differentiating Smartphone Users by App Usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, 2016.

[14] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 127–140. ACM, 2011.

[15] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[16] Valentin Radu, Lito Kriara, and Mahesh K Marina. Pazl: A mobile crowdsensing based indoor wifi monitoring system. In *Network and Service Management (CNSM), 2013 9th International Conference on*, pages 75–83. IEEE, 2013.

[17] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Tao Wang, and Xiaoming Li. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 249–260. ACM, 2014.

[18] Yohan Chon, Nicholas D Lane, Fan Li, Hojung Cha, and Feng Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 481–490. ACM, 2012.

[19] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4):3–13, 2000.

[20] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[21] Kyungseo Park, Eric Becker, Jyothi K Vinjumur, Zhengyi Le, and Fillia Makedon. Human behavioral detection and data cleaning in assisted living environment using wireless sensor networks. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, page 7. ACM, 2009.

[22] James Williamson, Qi Liu, Fenglong Lu, Wyatt Mohrman, Kun Li, Robert Dick, and Li Shang. Data sensing and analysis: Challenges for wearables. In *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pages 136–141. IEEE, 2015.

[23] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. Declarative support for sensor data cleaning. *Lecture Notes in Computer Science*, 3968:83–100, 2006.

[24] Diane M Strong, Yang W Lee, and Richard Y Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110, 1997.

[25] Eiman Elnahrawy and Badri Nath. Cleaning and querying noisy sensors. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 78–87. ACM, 2003.

[26] Dimitrios Lymberopoulos, Athanasios Bamis, and Andreas Savvides. Extracting spatiotemporal human activity patterns in assisted living using a home sensor network. *Universal Access in the Information Society*, 10(2):125–138, 2011.

[27] Yongxin Tong, Caleb Chen Cao, Chen Jason Zhang, Yatao Li, and Lei Chen. Crowdcleaner: Data cleaning for multi-version data on the web via crowdsourcing. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1182–1185. IEEE, 2014.

[28] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. A pipelined framework for online cleaning of sensor data streams. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 140–140. IEEE, 2006.

[29] Xu Chu, Ihab F Ilyas, and Paolo Papotti. Holistic data cleaning: Putting violations into context. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 458–469. IEEE, 2013.

[30] Maksims Volkovs, Fei Chiang, Jaroslaw Szlichta, and Renée J Miller. Continuous data cleaning. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 244–255. IEEE, 2014.

[31] Xu Chu, John Morcos, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, and Yin Ye. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1247–1261. ACM, 2015.

[32] Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. Nadeef: a commodity data cleaning system. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 541–552. ACM, 2013.

[33] Nilanjan Banerjee, Ahmad Rahmati, Mark D. Corner, Sami Rollins, and Lin Zhong. Users and batteries: Interactions and adaptive energy management in mobile systems. In John Krumm, Gregory D. Abowd, Aruna Seneviratne, and Thomas Strang, editors, *UbiComp 2007:*

Ubiquitous Computing, volume 4717 of *Lecture Notes in Computer Science*, pages 217–234. Springer Berlin Heidelberg, 2007.

[34] Ahmad Rahmati, Angela Qian, and Lin Zhong. Understanding human-battery interaction on mobile phones. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, pages 265–272. ACM, 2007.

[35] Ahmad Rahmati and Lin Zhong. Human–battery interaction on mobile phones. *Pervasive and Mobile Computing*, 5(5):465–477, 2009.

[36] Denzil Ferreira, Eija Ferreira, Jorge Goncalves, Vassilis Kostakos, and Anind K. Dey. Revisiting human-battery interaction with an interactive battery interface. In *Proceedings of Ubicomp 2013*. ACM, 2013.

[37] Abhijeet Banerjee, Lee Kee Chong, Sudipta Chattopadhyay, and Abhik Roychoudhury. Detecting energy bugs and hotspots in mobile apps. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, pages 588–598, New York, NY, USA, 2014. ACM.

[38] Xiao Ma, Peng Huang, Xinxin Jin, Pei Wang, Soyeon Park, Dongcai Shen, Yuanyuan Zhou, Lawrence K. Saul, and Geoffrey M. Voelker. eDoctor: automatically diagnosing abnormal battery drain issues on smartphones. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*, pages 57–70, Berkeley, CA, USA, 2013. USENIX Association.

[39] Abhinav Pathak, Abhilash Jindal, Y. Charlie Hu, and Samuel P. Midkiff. What is keeping my phone awake?: Characterizing and detecting no-sleep energy bugs in smartphone apps. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 267–280, New York, NY, USA, 2012. ACM.

[40] Alex Shye, Benjamin Scholbrock, and Gokhan Memik. Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 168–178, New York, NY, USA, 2009. ACM.

[41] Aaron Schulman, Thomas Schmid, Prabal Dutta, and Neil Spring. *Demo: Phone Power Monitoring with BattOr*, 2011. ACM Mobicom 2011. Available at http://www.stanford.edu/ aschulm/battor.html.

[42] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the 8th IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 105–114, New York, NY, USA, 2010. ACM.

[43] Fengyuan Xu, Yunxin Liu, Qun Li, and Yongguang Zhang. V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, pages 43–56, Berkeley, CA, USA, 2013. USENIX Association.

[44] Sharad Agarwal, Ratul Mahajan, Alice Zheng, and Victor Bahl. Diagnosing mobile applications in the wild. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 22:1–22:6, New York, NY, USA, 2010. ACM.

[45] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device Analyzer: Large-scale mobile data collection. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):53–56, 2014.

[46] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. Energy modeling of system settings: A crowdsourced approach. In *the 2015 IEEE International Conference on Pervasive Computing and Communications*, PerCom '15, pages 37–45, March 2015.

[47] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. Constella: Crowdsourced system setting recommendations for mobile devices. *Pervasive and Mobile Computing*, 26:71 – 90, 2016. Thirteenth International Conference on Pervasive Computing and Communications (PerCom '15).

[48] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[49] Fredrik Boström, Petteri Nurmi, Akos Vetek, Patrik Floréen, Péter Boda, Tianyan Liu, and Tiina-Kaisa Oikarinen. Capricorn - An intelligent user interface for mobile widgets. In *Proceedings of the 10th*

International Conference on Human-Computer Interaction (Mobile-HCI), pages 328–330. ACM, 2008.

[50] Andrea Girardello and Florian Michahelles. Appaware: Which mobile applications are hot? In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 431–434, New York, NY, USA, 2010. ACM.

[51] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 179–194, New York, NY, USA, 2010. ACM.

[52] Bo Yan and Guanling Chen. Appjoy: Personalized mobile application discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 113–126, New York, NY, USA, 2011. ACM.

[53] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. Falling asleep with angry birds, facebook and kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 47–56, New York, NY, USA, 2011. ACM.

[54] Kent Shi and Kamal Ali. Getjar mobile application recommendations with very sparse datasets. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 204–212, New York, NY, USA, 2012. ACM.

[55] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. Climbing the app wall: Enabling mobile app discovery through context-aware recommendations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2527–2530, New York, NY, USA, 2012. ACM.

[56] Stefano Mizzaro, Marco Pavan, Ivan Scagnetto, and Ivano Zanello. A context-aware retrieval system for mobile applications. In *Proceedings of the 4th Workshop on Context-Awareness in Retrieval and Recommendation*, CARR '14, pages 18–25, New York, NY, USA, 2014. ACM.

[57] Christoffer Davidsson and Simon Moritz. Utilizing implicit feedback and context to recommend mobile applications from first use. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '11, pages 19–22, New York, NY, USA, 2011. ACM.

[58] Wolfgang Woerndl, C. Schueller, and R. Wojtech. A hybrid recommender system for context-aware recommendations of mobile applications. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 871–878, April 2007.

[59] Konglin Zhu, Lin Zhang, and Achille Pattavina. Learning geographical and mobility factors for mobile application recommendation. *IEEE Intelligent Systems*, 32(3):36–44, 2017.

[60] Hannu Verkasalo. An international study of smartphone usage. *International Journal of Electronic Business*, 1/2:158–181, 2011.

[61] Stephen Sigg, Eemil Lagerspetz, Ella Peltonen, Petteri Nurmi, and Sasu Tarkoma. Exploiting usage to predict instantaneous app popularity: Trend filters and retention rates. *Under review, preprint available at* `https://arxiv.org/abs/1611.10161`.

[62] Denzil Ferreira, Jorge Gonçalves, Vassilis Kostakos, Louise Barkhuus, and Anind K. Dey. Contextual experience sampling of mobile application micro-usage. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, 2014.

[63] Alexis Hiniker, Shwetak N. Patel, Tadayoshi Kohno, and Julie A. Kientz. Why would you do that? predicting the uses and gratifications behind smartphone-usage behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 634–645, New York, NY, USA, 2016. ACM.

[64] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 329–344. ACM, 2011.

[65] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. Falling asleep with Angry Birds, Facebook and

Kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 2011.

[66] Simon L Jones, Denzil Ferreira, Simo Hosio, Jorge Goncalves, and Vassilis Kostakos. Revisitation analysis of smartphone app use. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1197–1208. ACM, 2015.

[67] Thanasis Petsas, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, and Thomas Karagiannis. Rise of the planet of the apps: A systematic study of the mobile app ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 277–290, New York, NY, USA, 2013. ACM.

[68] Suranga Seneviratne, Aruna Seneviratne, Prasant Mohapatra, and Anirban Mahanti. Your installed apps reveal your gender and more! *Mobile Computing and Communications Review*, 18:55–61, 2014.

[69] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. Discovering different kinds of smartphone users through their application usage behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16.

[70] Soo Ling Lim, Peter J. Bentley, Natalie Kanakam, Fuyuki Ishikawa, and Shinichi Honiden. Investigating country differences in mobile app user behavior and challenges for software engineering. *IEEE Transactions on Software Engineering*, 41:40–64, 2014.

[71] Seok Kang and Jaemin Jung. Mobile communication for human needs: A comparison of smartphone use between the us and korea. *Computers in Human Behavior*, 35:376 – 387, 2014.

[72] Gokul Chittaranjan, Jan Blom, and Daniel Gatica-Perez. Who's who with big-five: Analyzing and classifying personality traits with smartphones. In *Proceedings of the 2011 15th Annual International Symposium on Wearable Computers*, ISWC '11, pages 29–36, Washington, DC, USA, 2011. IEEE Computer Society.

[73] Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. Moodscope: Building a mood sensor from smartphone usage patterns. In *Proceeding of the 11th Annual International Conference on Mobile*

Systems, Applications, and Services, MobiSys '13, pages 389–402, New York, NY, USA, 2013. ACM.

[74] Neal Lathia, Veljko Pejovic, Kiran K. Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. Smartphones for large-scale behavior change interventions. *IEEE Pervasive Computing*, 12(3):66–73, 2013.

[75] Neal Lathia, Kiran K. Rachuri, Cecilia Mascolo, and Peter J. Rentfrow. Contextual dissonance: Design bias in sensor-based experience sampling methods. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 183–192, New York, NY, USA, 2013. ACM.

[76] Gillian M Sandstrom, Neal Lathia, Cecilia Mascolo, and Peter J Rentfrow. Putting mood in context: Using smartphones to examine how people feel in different locations. *Journal of Research in Personality*, 69:96–101, 2017.

[77] Agnes Gruenerbl, Venet Osmani, Gernot Bahle, Jose C. Carrasco, Stefan Oehler, Oscar Mayora, Christian Haring, and Paul Lukowicz. Using smart phone mobility traces for the diagnosis of depressive and manic episodes in bipolar patients. In *Proceedings of the 5th Augmented Human International Conference*, AH '14, pages 38:1–38:8, New York, NY, USA, 2014. ACM.

[78] G Drake, E Csipke, and T Wykes. Assessing your mood online: acceptability and use of moodscope. *Psychological medicine*, 43(7):1455–1464, 2013.

[79] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K Dey. Who are the smartphone users?: Identifying user groups with apps usage behaviors. *GetMobile: Mobile Computing and Communications*, 21(2):31–34, 2017.

[80] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proceedings of the Sixth Symposium on Operating Systems Design and Implementation*, OSDI '04, 2004.

[81] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[82] Jeffrey Dean and Sanjay Ghemawat. MapReduce: A flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.

[83] Amol Ghotingm, Rajasekar Krishnamurthy, Edwin Pednault, Berthold Reinwald, Vikas Sindhwani, Shirish Tatikonda, Yuanyuan Tian, and Shivakumar Vaithyanathan. SystemML: Declarative machine learning on MapReduce. In *Proceedings of IEEE 27th International Conference on Data Engineering*, ICDE '11. USENIX Association, 2011.

[84] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of NSDI '12: 9th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '12, pages 15–28. USENIX Association, 2012.

[85] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.

[86] Tim Kraska, Ameet Talwalkar, John Duchi, Rean Griffith, Michael F. Franklin, and Michael Jordan. MLbase: A distributed machine-learning system. In *Proceedings of 6th Biennial Conference on Innovative Data Systems Research*, CIDR '13, 2013.

[87] Yu Xiao, Pieter Simoens, Padmanabhan Pillai, Kiryong Ha, and Mahadev Satyanarayanan. Lowering the barriers to large-scale mobile crowdsensing. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, page 9. ACM, 2013.

[88] Denzil Ferreira, Vassilis Kostakos, and Anind K Dey. Aware: mobile context instrumentation framework. *Frontiers in ICT*, 2:6, 2015.

[89] Kumaribaba Athukorala, Eemil Lagerspetz, Maria von Kügelgen, Antti Jylhä, Adam J. Oliner, Giulio Jacucci, and Sasu Tarkoma. How Carat Affects User Behavior: Implications for Mobile Battery Awareness Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, New York, NY, USA, 2014. ACM.

[90] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. Too big to mail: On the way to publish large-scale mobile analytics data. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 2374–2377. IEEE, 2016.

[91] José S Marcano Belisario, Jan Jamsek, Kit Huckvale, John O'Donoghue, Cecily P Morrison, and Josip Car. Comparison of self-administered survey questionnaire responses collected using mobile apps versus other methods. *Cochrane Database of Systematic Reviews*, (7), 2015.

[92] Ella Peltonen, Eemil Lagerspetz, Jonatan Hamberg, Abhinav Mehrotra, Mirco Musolesi, Petteri Nurmi, and Sasu Tarkoma. The hidden image of mobile usage: Uncovering the impact of geographic and demographic factors. *Under review.*

[93] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, December 2004.

[94] Mark Harman, Yue Jia, and Yuanyuan Zhang. App store mining and analysis: Msr for app stores. In *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, MSR '12, pages 108–111, Piscataway, NJ, USA, 2012. IEEE Press.

[95] Hee-Woong Kim, Hyun Lyung Lee, and Jung Eun Son. An exploratory study on the determinants of smartphone app purchase. In *The 11th International DSI and the 16th APDSI Joint Meeting, Taipei, Taiwan*, 2011.

[96] Mark de Reuver, Harry Bouwman, Nico Heerschap, and Hannu Verkasalo. Smartphone Measurement: do People Use Mobile Applications as they Say they do? In *Proc. International Conference on Mobile Business (ICMB)*, 2012.

[97] Lenin Ravindranath, Jitendra Padhye, Sharad Agarwal, Ratul Mahajan, Ian Obermiller, and Shahin Shayandeh. Appinsight: Mobile app performance monitoring in the wild. In *OSDI*, volume 12, pages 107–120, 2012.

[98] C. H. (Kees) van Berkel. Multi-core for mobile phones. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '09)*, pages 1260–1265, 2009.

[99] D. Linden and T. B. Reddy. *Handbook of Batteries.* McGrew-Hill Professional, 2001. 3rd edition.

[100] Khai N. Truong, Julie A. Kientz, Timothy Sohn, Alyssa Rosenzweig, Amanda Fonville, and Tim Smith. The design and evaluation of a

task-centered battery interface. In *Proceedings of the12th International Conference on Ubiquitous Computing*, 2010.

[101] Denzil Ferreira, Anind K. Dey, and Vassilis Kostakos. Understanding human-smartphone concerns: A study of battery life. In *Proceedings of the 9th International Conference on Pervasive Computing*, 2011.

[102] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2119–2128, New York, NY, USA, 2009. ACM.

[103] Bo Yan and Guanling Chen. Appjoy: Personalized mobile application discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 113–126, New York, NY, USA, 2011. ACM.

[104] Kent Shi and Kamal Ali. Getjar mobile application recommendations with very sparse datasets. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 204–212, New York, NY, USA, 2012. ACM.

[105] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 210–217, New York, NY, USA, 2010. ACM.

[106] S. Ronen and O. Shenkar. Clustering countries on attitudinal dimensions: A review and synthesis. *Academy of Management Review*, 10:435–454, 1985.

[107] Viv J. Shackleton and Abbas H. Ali. Work-related values of managers: A test of the hofstede model. *Journal of Cross-Cultural Psychology*, 21:109–118, 1990.

[108] Bradley L Kirkman, Kevin B Lowe, and Cristina B Gibson. A quarter century of culture's consequences: A review of empirical research incorporating hofstede's cultural values framework. *Journal of International Business Studies*, 37(3):285–320, 2006.

[109] I. P. L. Png, B. C. Y. Tan, and Khai-Ling Wee. Dimensions of national culture and corporate adoption of it infrastructure. *IEEE Transactions on Engineering Management*, 48(1):36–45, Feb 2001.

[110] John C. Crotts and Ron Erdmann. Does national culture influence consumers' evaluation of travel services? a test of hofstede's model of cross-cultural differences. *Managing Service Quality: An International Journal*, 10(6):410–419, 2000.

[111] Richard A. Bernardi and Steven T. Guptill. Social desirability response bias, gender, and factors influencing organizational commitment: An international study. *Journal of Business Ethics*, 81(4):797–809, Sep 2008.

[112] Kendall Goodrich and Marieke de Mooij. How 'social' are social media? a cross-cultural comparison of online and offline purchase decision influences. *Journal of Marketing Communications*, 20(1-2):103–116, 2014.

[113] Katharina Reinecke, Minh Khoa Nguyen, Abraham Bernstein, Michael Näf, and Krzysztof Z Gajos. Doodle around the world: Online scheduling behavior reflects cultural differences in time perception and group decision-making. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 45–54. ACM, 2013.

[114] Xuan Lu, Wei Ai, Xuanzhe Liu, Qian Li, Ning Wang, Gang Huang, and Qiaozhu Mei. Learning from the ubiquitous language: An empirical analysis of emoji usage of smartphone users. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 770–780, New York, NY, USA, 2016. ACM.

[115] Brendan McSweeney. Hofstede's model of national cultural differences and their consequences: A triumph of faith - a failure of analysis. *Human relations*, 55(1):89–118, 2002.

[116] Janet Vertesi, Silvia Lindtner, and Irina Shklovski. Transnational HCI: Humans, computers, and interactions in transnational contexts. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 61–64. ACM, 2011.

Included Publications and Manuscripts

Research Theme A: Mobile Energy Consumption

Research Paper I

Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma

Energy Modeling of System Settings: A Crowdsourced Approach

Published in the Proceedings of the IEEE International Conference on Pervasive Computing and Communications, PerCom '15, St. Louis, MO, USA, March 23-27, 2015.

Contribution: The author was in the lead of the planning of the publication, implementing necessary distributed data mining and statistical analysis algorithms, analyzing the data, and writing the publication. The data collection itself is based on the earlier work done in the Carat project lead by Dr Eemil Lagerspetz. Dr Petteri Nurmi and Prof. Sasu Tarkoma gave important contributions to the planning and writing processes of the publication.

Energy Modeling of System Settings: A Crowdsourced Approach

Ella Peltonen*, Eemil Lagerspetz*, Petteri Nurmi*†, Sasu Tarkoma*†
†Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki
*Department of Computer Science, University of Helsinki, PO Box 64, FI-00014, University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi

***Abstract*—The question "Where has my battery life gone?" remains a common source of frustration for many smartphone users. With the increased complexity of smartphone applications, and the increasing number of system settings affecting them, understanding and optimizing battery use has become a difficult chore. The present paper develops a novel approach for constructing *energy models* from *crowdsourced* measurements. In contrast to previous approaches, which have focused on the effect of a specific sensor, system setting or application, our approach can simultaneously capture relationships between multiple factors, and provide a unified view of the energy state of the mobile device. We demonstrate the validity of using crowdsourced measurements for constructing battery models through a combination of large-scale analysis of a dataset containing battery discharge and system state measurements and hardware power measurements. The results indicate that the models captured by our approach are both in line with previous studies on battery consumption and empirical measurements, providing a cost-effective way to construct energy models during normal operations of the device. The analysis also provides several new insights about battery consumption. For example, our analysis shows the energy use of high CPU activity with automatic screen brightness is actually higher (resulting in around 9 minutes shorter battery lifetime on average) than with a medium CPU load and manual screen brightness; a Wi-Fi signal strength drop of one bar can result in a battery life loss of over 13%; and a smartphone sitting in the sun can experience over 50% worse battery life than one indoors in cool conditions.**

***Index Terms*—Mobile, Subsystems, Energy**

I. Introduction

Samsung sold over 40 million Galaxy S4 smartphones in six months in 2013[1]. Apple sold over 43 million iPhones in the second quarter of 2014[2]. Most of us use a smartphone daily, for work, entertainment and a variety of other purposes besides communication [3]. The processing and transmission power of smartphones continues to grow [21], while their batteries remain largely unchanged [10]. Consequently, energy efficiency remains a high priority for current smartphone operating systems, and increasingly, for applications. The importance of energy efficiency has also been highlighted in several user studies, which have shown that users actively take measures to optimize the power consumption of their device [5], [19].

Modern smartphones incorporate several mechanisms for optimizing battery consumption. On the operating system level, complex on-demand resource optimization strategies are used to reduce battery consumption [20]. However, the effectiveness of these policies is highly dependent on the context of the user, and there often are complex interdependencies that make it difficult to determine the optimal policy for a given situation. Ensuring the effectiveness of these policies requires fine-grained models that can characterize how different contexts and device features influence the power consumption of the device.

Fig. 1. The number of system settings available on current smartphones can be overwhelming.

The alternative to automated policies is to give users control over specific system settings, such as whether to prefer Wi-Fi or cellular networks, which screen brightness to use, and when to turn off the screen after inactivity. Indeed, contemporary smartphones have interfaces that allow this kind of operations with little effort. With an increasing number of user-controllable system settings, keeping track of each setting's energy impact becomes unmanageable. To illustrate this problem, Fig. 1 depicts some of the system settings available on the Samsung Galaxy S4. Over 20 different settings are visible, most of which have a significant effect on energy. To fully understand the impact of all of the settings would require a considerable amount of learning. Furthermore, the total energy consumption of the smartphone is not simply the sum of the energy impacts of enabled system settings. Some subsystems, such as Bluetooth and Wi-Fi, are integrated on the same chip, and can be enabled simultaneously at much less than the sum of their combined individually measured energy impacts. Another example is the integration of accelerometers and gyroscopes on the same chip to provide energy savings for activity monitoring applications.

[1]http://www.theinquirer.net/inquirer/news/2302755/samsung-disappointed-with-galaxy-s4-sales-despite-hitting-40-million-milestone

[2]http://www.businesswire.com/news/home/20140423006671/en/Apple-Reports-Quarter-Results#.U17IYnEZscl

Enabling users to make optimal decisions would thus require fine-grained information about how different settings influence the overall battery consumption of the device in a given setting.

The present paper contributes by developing a novel approach for constructing *energy models* from crowdsourced battery discharge measurements. Such information can be increasingly collected through non-obtrusive instrumentation of the device [13], which in turn enables capturing battery information across a wide range of usage contexts and devices. Experiments conducted through a combination of power meter measurements and a large-scale analysis of crowdsourced discharge measurements demonstrate that our method is capable of constructing models that accurately capture complex interdependencies between system settings, sensors, and usage contexts, providing an accurate view of the *state* of the device. This contrasts with previous works, which have predominantly focused on capturing the effect of a specific sensor, system setting or application [4], [16]. Results from our evaluation provide novel insights about battery usage, demonstrating how complex the relationships between different factors and battery discharge are. For example, our analysis shows the energy use of high CPU activity with automatic screen brightness is actually higher (resulting in around 9 minutes fewer battery lifetime on average) than with a medium CPU load and manual screen brightness; a Wi-Fi signal strength drop of one bar can result in a battery life loss of over 13%; and a smartphone sitting in the sun can experience over 50% worse battery life than one indoors in cool conditions.

The contributions of the paper are summarized as follows:

- We develop a novel approach for constructing energy models from crowdsourced measurements. The models constructed by our approach can capture the *combined* effects of multiple factors simultaneously, providing a characterization of the *energy state* of a mobile device.
- Experiments carried out through a combination of power meter measurements and a large-scale analysis of crowdsourced discharge measurements demonstrate that our approach can capture the state of the device accurately and cost-effectively, even in the presence of complex interdependencies between context factors. Through our analysis, we also reveal new insights, highlighting the complexity of factors that influence battery consumption.
- We make available a large-scale data set of 11.2 million data points from around 150.000 active Android users[3].

II. Related Work

Mobile device energy profiling has been studied in the past [8], [14], [24], [25]. Most previous works take a holistic view of the device and its energy use [24], while some target application energy use specifically [11]–[14]. There are a number of systems that take into account the hardware subsystems of the device [25]. Some of these monitor inside the device [8] while others utilize a support server [1]. Most of these systems target specific sensors, such as location or Wi-Fi, or a subset of the sensors integrated on a typical smartphone. However, the complete state of the device requires considering all subsystems that can use energy. The BattOr [17] system can be used to monitor a mobile device in the wild, but it needs to be manually connected to the device prior to monitoring, and has a limited operating time. The DeviceAnalyzer project [22] is gathering rich measurements of mobile device state, but the data has not yet been used for large–scale analysis. To our best knowledge, no previous works are capable of constructing fine-grained energy models from crowdsourced measurements. Also, the effect of different system settings on battery consumption remains under-explored.

Sensor power profiling focuses on constructing power consumption models for individual sensors or sensor combinations. One of the earliest works in this vein was introduced by Rice and Hay [16], who examine fine-grained hardware power measurements and their causes, and attribute energy drain to the networking stack version, packet size, and Wi-Fi handshake behavior. König et al. [9] measure power consumption of different sensors using a hardware power monitor. Kjærgaard et al. [7] use conditional functions, manually constructed from empirical power measurements, to represent power consumption of different sensors. Kjærgaard and Blunk [8] propose using genetic algorithms for learning the conditional functions in an unsupervised manner. Contrary to our work, none of these approaches are capable of capturing complex interdependencies in the energy consumption of different components.

Another alternative is to construct statistical models that characterize the *overall* battery consumption of a device. These approaches consider how application usage patterns, workload and other system level parameters, such as screen brightness and data transfer rate influence battery discharge. The estimated discharge rate of the device can then be used to predict the remaining lifetime of the device's battery. Wen et al. [23] propose constructing a reference curve of the battery consumption under different workloads. Once the reference curve has been constructed, a regression model is used to compare current discharge with an estimate calculated using the reference curve. The deviations from the reference curve can then be used to refine estimates of remaining battery lifetime. Instead of considering workload, Kang et al. [6] predict discharge behavior from application usage patterns. Zhao et al. [26] predict battery discharge using a regression model that considers multiple different system variables (e.g., CPU utilization, I/O rate and LCD backlight brightness). In addition to predicting battery discharge, Ravi et al. [15] predict when the user is likely to have the next charging opportunity and how much battery power is needed for maintaining essential functionality until then. If the system detects that the battery is likely to run out before the next charging opportunity, the system pro-actively provides a warning to the user instead of waiting for the battery to be nearly depleted. Falaki et al. [4] conduct an analysis of smartphone usage patterns, revealing that usage patterns contain significant variation across users and that personalized application usage models are essential for accurate prediction of battery drain. In contrast to our approach,

[3]http://carat.cs.helsinki.fi/research

which can capture how changes in device state influence battery discharge, these approaches can only provide aggregate level information of power usage.

III. Background: Dataset

We consider a large-scale dataset of crowdsourced battery discharge measurements collected from a collaborative energy diagnostic system Carat [13]. The application has collected data from around 725,000 Android and iOS devices since summer 2012. We consider a subset, which we have made publicly available for research purposes[4], of the data containing 11.2 million samples from around 150,000 active Android devices. The data includes information about the device's operating system and model, the current battery level, the set of currently active applications, and information about different system settings such as network connections and screen brightness and subsystem variables such as the CPU use and the distance traveled since the last measurement. We refer to these system settings and subsystem variables collectively as *context factors*.

As a baseline for energy consumption, we consider the *energy rates* reported by Carat, which reflect normalized energy consumption per time unit, i.e., energy rate = Δ battery / Δ t. The methodology used to derive rates and the validity of using energy rates as a measure for battery consumption has been shown in previous work by Oliner et al. [13].

We focus on 13 different context factors, including 5 user-changeable system settings and 8 other pieces of subsystem state information. These were selected based on previous studies on energy-efficiency, which have shown them to be dominant factors explaining battery consumption. We consider the status defined by all 13 factors as the *state* of the device. The five system settings that we have collected via the Android API and that we utilize in this work are:

- **Mobile data status**, either connected, disconnected, connecting, or disconnecting,
- **Mobile network type**, such as LTE, HSPA, GPRS, EDGE, or UMTS,
- **Network type** used for Internet connectivity, either none, Wi-Fi, mobile, or wimax,
- whether **Roaming** is enabled or disabled, and
- **Screen brightness**, 0-255 or "automatic" (-1).

We have also collected information about 8 subsystem variables. These are not directly available as a user-modifiable system setting, but can give information about the state of the smartphone. For example, if we notice a decreased Wi-Fi link speed or signal strength, we can recommend that the user try to use the mobile network instead of Wi-Fi in this context. This state information includes:

- **Battery health**, determined by the smart battery API of the Li-Ion battery of the Android device,
- **Battery temperature** in degrees Celsius,
- **Battery voltage** in Volts,
- **CPU use** in percent,
- **Distance traveled** between two samples in meters,
- **Mobile data activity**, one of none, out, in or inout,
- **Wi-Fi link speed**, in Mbps, and
- **Wi-Fi signal strength** in dBm.

[4]http://carat.cs.helsinki.fi/research

Context Factor	Mean	Std	Median
CPU use	75%	33%	91%
Battery voltage (V)	3.78	0.61	3.84
Screen brightness (0-255)	128.03	85.71	109
Temperature (°C)	29.27	5.75	30
Wi-Fi signal strength (dBm)	-61.29	13.02	-61

TABLE I
SUMMARY STATISTICS OF SELECTED CONTEXT FACTORS.

Most context factors are nominal-valued. To simplify comparison of these factors, we have discretized them into categories using an equal frequencies procedure, i.e., each factor was divided into categories containing approximately the same number of values. The number of categories was determined empirically and based on observations reported in previous battery usage studies. Summary statistics of selected context factors are given in Table I and the different categories are detailed below. For categorical variables (such as network type), we have considered the different possible values as categories.

CPU use: We consider measurements that reflect the percentage of time CPU is active. The mean and median in Table I indicate that CPUs are mostly active. We split the CPU use around the mean, resulting in three categories: Low (0 - 42%), Medium (43 - 85%), and High (86 - 100%).

Distance traveled: Most values are during stationary periods or with little movement. Based on this observation, we consider a split between stationary and non-stationary behavior.

Battery voltage: The safe operating voltage of a smartphone Li-Ion battery is 3 - 4.2V. The nominal voltage of such batteries is typically 3.7V. The mean, the median, and the standard deviation reflect this very closely. We consider three categories for voltage: Low (0 - 3V), Medium (3 - 4.2V), High (4.2V+).

Screen brightness: When screen brightness was manually controlled, the mean was around 128, or the exact midpoint. The standard deviation of the values suggests that almost the entire range of brightness settings is used, making it difficult to categorize the screen brightness values. Accordingly, we consider a binary split into manual and automatic brightness.

Wi-Fi signal strength: We consider RSS values in the range [-100, 0]. Good Wi-Fi signal strength values are normally between -30 and -10dBm, and the worst, while still being connected, is -95dBm. We consider four categories: Bad (-100 to -75dBm), Average (-74 to -61dBm), Good (-61 to -49dBm) and Excellent (-49 to 0dBm). The mean RSS is between the Average and the Good levels, and the Excellent and the Bad levels are within one standard deviation. These values are in line with typical values used in Wi-Fi positioning literature.

Ethical Considerations: We consider only aggregate level data which contains no personally identifiable data. The privacy protection mechanisms of Carat are discussed in detail by the authors [13]. Data collection by Carat is subject to the IRB process of UC Berkley. Users of Carat are informed about the collected data and give their consent from their devices.

IV. Battery Modeling from Crowdsourced Measurements

Battery consumption has traditionally been based on empirical models taken either directly on the battery level [17] or through system-level APIs. The former requires specialized measurements tools, limiting the contexts where measurements can be taken. The latter, on the other hand, has been shown to result in inaccuracies in the resulting models [25]. In this section we demonstrate the validity of using crowdsourced battery discharge measurements for constructing battery consumption models. Our approach provides a cost-effective alternative for modeling battery consumption, and, as we later demonstrate, our approach can capture complex interdependencies affecting battery consumption in everyday use.

A. Methodology

We construct battery models by measuring the strength of statistical association between context factors and battery discharge rates. To measure statistical association, we consider two complementary metrics. As our first metric, we consider gain in battery life, denoted *BL Gain*, which measures how changes in context factors influence the lifetime of a device on average. As our second measure, we consider the *conditional mutual information* (CMI) between context factors and energy rates. For assessing the influence of a single context factor X and energy rate Z, the CMI is equivalent to the *mutual information* (MI) given by:

$$MI(X,Z) = \sum_{z\in Z}\sum_{x\in X} p(x,z)\cdot log\left(\frac{p(x,z)}{p(x)\cdot p(z)}\right).$$

For higher order combinations containing two or more context factors (denoted X and Y), the CMI is defined as follows:

$$CMI(X,Y|Z) = \sum_{z\in Z}\sum_{y\in Y}\sum_{x\in X} p(x,y,z)\cdot \log\left(\frac{p(z)\cdot p(x,y,z)}{p(x,z)\cdot p(y,z)}\right).$$

The battery life gain measurements provide information about *absolute differences*, whereas the (C)MI measurements can be used for *relative* comparison between different context factors. The two metrics provide complementary ways to analyze strength of associations, and in applications the choice of metric depends on the scenario being considered.

B. Individual Context Factors

We demonstrate the validity of using battery discharge measurements for constructing energy models by examining the mutual information between context factors and energy rates. We derive a ranking for different factors based on their mutual information values, and demonstrate that this ranking is in line with findings from empirical studies on battery consumption.

Estimations by mutual information MI of context factors and energy consumption are given in Table II. The results of the MI analysis are well in line with previous results [4], [18]. In particular, the major individual impact of CPU use and traveled distance on battery consumption is clearly observable,

Context Factor	MI Estimate
CPU use	1.330
Distance traveled	1.069
Battery temperature	0.143
Battery voltage	0.099
Screen brightness	0.030
Mobile network type	0.019
Network type	0.018
Wi-Fi signal strength	0.014
Wi-Fi link speed	0.014
Mobile data status	0.013
Mobile data activity	0.005
Battery health	0.004
Roaming	0.0002

TABLE II
CONTEXT FACTORS' IMPACT ON ENERGY CONSUMPTION, ORDERED BY MUTUAL INFORMATION ESTIMATE.

Context Factor	Value	BL Gain
CPU use	Low (0–42%)	+3.24%
CPU use	Medium (43–85%)	+5.72%
CPU use	High (86–100%)	-2.48%
Distance traveled	None	-0.76%
Distance traveled	>0	+8.20%
Battery voltage	Low (0–3V)	-16.60%
Battery voltage	Medium (3–4.2V)	-0.76%
Battery voltage	High (4.2V+)	+69.08%
Screen brightness	Manual	-4.96%
Screen brightness	Automatic	+6.29%
Wi-Fi signal strength	Bad (-100 – -75 dBm)	-2.29%
Wi-Fi signal strength	Average (-74 – -61 dBm)	+4.00%
Wi-Fi signal strength	Good (-61 – -49 dBm)	+6.29%
Wi-Fi signal strength	Excellent(-48 – 0 dBm)	+7.63%

TABLE III
THE EXPECTED ENERGY USE TYPICAL VALUES OF CONTEXT FACTORS. BL GAINS CAN BE COMPARED TO STATISTICS IN TABLE I.

and the ordering of the settings is similar to those derived through explicit battery measurements.

The results also contain some exceptions to the findings of previous studies. The most prominent example of these is screen brightness, which is commonly considered the most battery heavy feature. In our analysis, screen brightness results in a lower score than many other attributes. In the next section, we demonstrate that the absolute energy impact of screen brightness actually is high. However, as mutual information effectively looks at the correlation between battery discharge and context factors, the changes are affected by other context factors. In most use contexts screen use is correlated with battery voltage and CPU use, both of which have a large impact on the battery drain, and hence also on the mutual information values. The main effect we observe for screen brightness comes from switching to automatic brightness.

C. Energy Consumption of Context Factors

We next consider how typical values of context factors influence battery discharge. We focus on the five factors discussed in Section IV-A (CPU use, battery voltage, screen brightness, temperature, and Wi-Fi signal strength) and consider expected gain in battery life as our evaluation measure. The results of this evaluation are shown in Table III.

In line with the results of mutual information analysis, the worst battery life is obtained for high CPU use. The benefit

of maintaining a balanced CPU load is significant, as medium CPU use produces +5.72% energy benefit compared to average use. For screen brightness, the automatic setting of the device usually improves battery life, providing even +6.29% better battery life compared to the average. Manual brightness, in contrast, shows a major loss of battery life (-4.97%).

We can make a number of other observations from the results. First, higher battery voltage results in improved battery life. This is partially explained by voltage correlating with battery health and capacity. In addition, the rates reported by Carat are based on changes in battery percentage, which tend to follow voltage changes linearly. This contrasts with actual discharge, which is nonlinear, particularly when the battery is close to full charge. The results also suggest that battery life tends to be higher for mobile than for stationary users. Studies on application usage have shown that interactions with applications are common during mobility, with web browsing, news, music/video players, and gaming being the dominant application categories [3]. Hence, the difference is likely a result of shorter interaction periods rather than avoidance of energy intensive operations. Finally, a high Wi-Fi signal strength leads to better battery life, as the phone needs to spend less energy for receiving and sending data. Bad signal availability can lead to situations where the device has to reconnect to the network repeatedly, further increasing battery consumption.

V. Context Factor Combinations

The results of our analysis thus far show that estimates given by our method are in line with observations made in studies carried out in laboratory conditions with specialized hardware measurement tools, providing a strong indication of the potential of using our approach as a cost-effective mechanism to construct models of battery consumption. However, a limitation of our analysis thus far has been the focus on individual factors' impact on battery life, without considering the state of the device as a whole. As an example, consider the case of screen brightness, which according to previous studies is one of the main battery hogs on a smartphone. In terms of expected battery gain, our results also support this observation, indicating over 5% deviations from average consumption patterns. In actual application contexts, screen usage is highly correlated with interactions on the device, which in turn require CPU and network usage, suggesting that screen brightness is not the *dominant* factor explaining battery usage. To capture such nuanced differences in consumption, we argue that models reflecting the *state* of the smartphone are required. In the following we demonstrate the validity of considering combinations of context factors as part of battery models. We also compare our results against empirical power models constructed using a hardware power monitor, demonstrating that our approach can capture more fine-grained differences in battery consumption than empirical power models.

A. Statistical Results of Context Factor Combinations

We first demonstrate the complexity of battery consumption patterns by considering how pairs of context factors influence consumption. Similarly to the previous section, we derive a ranking for the different pairs by considering the conditional mutual information between each pair, and rank the pairs in descending order of CMI values. The results of these estimations are listed in Table IV.

Context Factors		CMI
Battery voltage	CPU use	4.29
CPU use	Screen brightness	2.17
Battery temperature	CPU use	2.07
CPU use	Distance traveled	1.81
CPU use	Wi-Fi signal strength	1.69
Battery voltage	Distance traveled	1.53
Battery temperature	Distance traveled	1.28
Distance traveled	Screen brightness	1.26
CPU use	Wi-Fi link speed	1.12
Battery voltage	Screen brightness	1.08
Wi-Fi link speed	Wi-Fi signal strength	0.99
Mobile data status	Network type	0.95
Network type	Wi-Fi signal strength	0.85
CPU use	Mobile network type	0.80
Battery temperature	Screen brightness	0.79
Distance traveled	Wi-Fi signal strength	0.75
Network type	Wi-Fi link speed	0.64
Mobile data status	Wi-Fi signal strength	0.60
Battery temperature	Battery voltage	0.56
Distance traveled	Wi-Fi link speed	0.54
Battery voltage	Wi-Fi signal strength	0.53

TABLE IV
Top of the conditional mutual information estimates for pairs of context factors for energy consumption rates.

Compared to the results of individual context factors' impact (see Table II), the combination of multiple factors gives more accurate explanations of the battery consumption. A prominent example is CPU use, for which we can observe significantly higher impact when combined with another factor than when considered alone. Also factors related to network connection, such as Wi-Fi signal strength and network type, differ clearly from the MI analysis. Both have lower MI values in Table II, but are more prominent when considered in conjunction with another context factor. Wi-Fi link speed and Wi-Fi signal strength have a combined MI of 0.99, which is higher than they get separately (0.014 each). Capturing this kind of nuances in consumption is particularly beneficial when giving suggestions to the end user on how to improve battery life. For example, from the results we can observe that changing the other system setting can help to improve battery life in cases where high CPU use is mandated, for example, when playing a game.

The top context factors according to energy consumption seem to be battery voltage, CPU use, battery temperature, and movement (distance traveled) of the device, or combinations thereof. The effects of these factors are mediated by other factors, which in turn can cause significant increases or decreases in consumption. Accordingly, providing an accurate view of the battery consumption of a device requires models that can capture both the effects of multiple context factors and the effects of their interdependencies.

B. Battery Consumption of Context Factor Combinations

To further illustrate the complexity of battery consumption patterns, we consider how selected context factor combinations

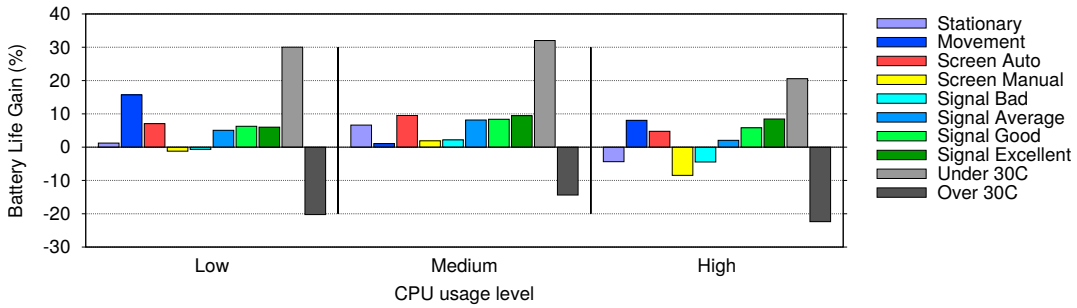


Fig. 2. Battery gain from CPU use combined with another context factor.

affect battery consumption. We chose combinations with high CMI values (see Table IV), and measured their expected battery life gain. As an example case, we consider the impact of CPU use and another factor on battery consumption. The results of this analysis are shown in Fig. 2. The y-axis shows the battery life gain in percentages when compared to the average expected battery life in our dataset. The different columns represent different values of context factors.

From the figure we can observe that the influence of context factor combinations on battery consumption is rather complex. For example, as long as the phone can observe average Wi-Fi signal strength, improving the connection will not provide significant savings unless CPU use is very high. In terms of screen brightness, as shown previously, the main effect results from switching to automatic brightness. However, this effect is most beneficial for moderate CPU use, and during high use, other factors can provide more pronounced changes. Another important factor is battery temperature, which can result in a loss of up to 50% battery life. The effects of temperature are consistent across all CPU use categories, indicating CPU use is not necessarily the (sole) cause for high battery temperature.

C. Power Meter Validation

The proposed approach of using crowdsourced measurements for constructing battery consumption models has been intended as a cost-effective way to capture fine-grained and nuanced differences in battery consumption. As we have demonstrated, these differences can have a significant impact on battery consumption and need to be accounted for to provide accurate estimates of the actual battery usage. We next compare our approach against empirical power models constructed using a hardware power monitor. We demonstrate that our approach is better at capturing the effects of changes in device state on battery consumption. In particular, our analysis indicates that empirical models are dominated by *instantaneous* effect, which tends to overestimate overall power consumption.

We consider measurements collected using a Samsung Galaxy S2 phone that was connected to a Monsoon Power Monitor[5]. The Power Monitor was set to output a constant voltage of 4.0V. The phone battery was still inserted, with the + and - terminals blocked, letting the phone start up normally. Each experiment run lasted 10 minutes. We connected the phone to the cellular network and turned the phone screen on for the duration of each experiment run. Wi-Fi was enabled and Bluetooth disabled for all our experiment runs. Automatic updating of applications was disabled. All applications were closed. Before each experiment run, we let the power consumption stabilize for several minutes to avoid the impact of background activity on the experiment. The experiment runs consisted of the following configurations:

- Full screen brightness, 30% CPU use, and bad (1); average (2); and good Wi-Fi signal strength (3).
- Full screen brightness, 60% CPU use, and bad (4); average (5), and good Wi-Fi signal strength (6).
- Full screen brightness, 100% CPU use (two tightly looping threads), and bad (7); average (8); and good Wi-Fi signal strength (9).
- Automatic screen brightness, average Wi-Fi signal strength, and 30% (10); 60% (11); 100% CPU use (12).

The impact of considering combinations instead of individual factors can be assessed by examining the relative standard deviations (i.e., ratio between standard deviation and mean) of the power monitor measurements. These are illustrated in Fig. 3. In the figure we consider separately the combined effect of CPU and Wi-Fi, and that of CPU and screen brightness. We also consider how decomposing these factors into categories influences the measurements.

The column groups in the figure correspond to the Wi-Fi signal strength range, and an average value for all of the values (signal all). Respectively, for screen brightness we consider automatic and manual, and average over all values (screen both). The columns in each group, from left to right, are CPU All (average over all use levels), Low, Medium, and High CPU use level. From the figure we can observe that the relative standard deviations for combined CPU use (i.e., columns with CPU All) are much higher than those of individual use levels by a factor of at least 1.5. The same observation applies for signal strength and screen brightness, indicating that considering individual factors is much less accurate at explaining battery consumption than the combination of multiple factors.

[5]https://www.msoon.com/LabEquipment/PowerMonitor/

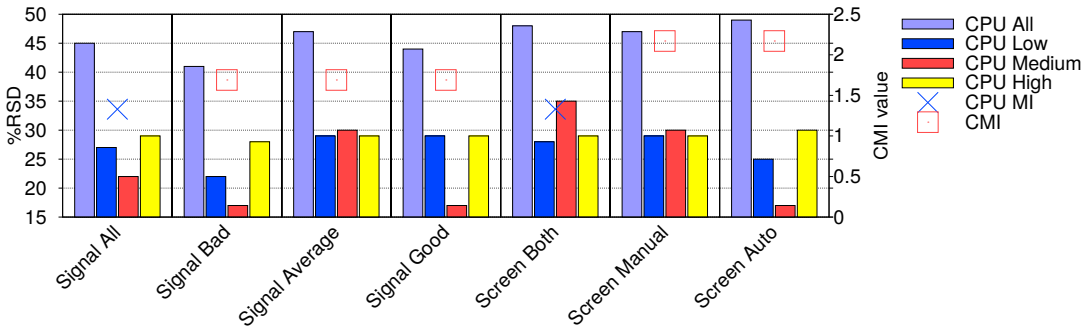


Fig. 3. Relative Standard Deviation (%RSD) of various CPU use and Wi-Fi signal strengths and screen brightnesses, along with their MI and CMI values.

CPU Use	All	Low	Medium	High
Experiment				
All Wi-Fi	2.60	4.77	2.23	1.90
Bad Wi-Fi	2.51	4.52	2.20	1.93
Average Wi-Fi	2.60	4.99	2.27	1.91
Good Wi-Fi	2.53	4.82	2.21	1.87
All screen br.	2.69	5.40	3.27	1.92
Manual screen br.	2.60	4.99	2.27	1.91
Auto screen br.	2.78	5.89	2.50	1.92
Crowdsourced data				
All Wi-Fi	5.24	5.53	5.54	5.53
Bad Wi-Fi	5.12	5.21	5.36	5.01
Average Wi-Fi	5.45	5.51	5.67	5.35
Good Wi-Fi	5.57	5.57	5.68	5.54
All screen br.	5.24	5.53	5.54	5.53
Manual screen br.	4.98	5.18	5.34	4.80
Auto screen br.	5.57	5.61	5.74	5.49

TABLE V
COMPARISON OF AVERAGE BATTERY LIFETIME IMPROVEMENT (IN HOURS) BETWEEN POWER MONITOR ESTIMATES AND OUR APPROACH.

We next compare battery life estimates between power monitor measurements and our approach. The results of this analysis are shown in Table V. From the results we can make two important observations. First, for low CPU use, the power monitor measurements are closely in line with the estimates provided by our approach. However, as CPU use increases, the estimates provided by power monitor measurements indicate significant decrease, whereas the estimates provided by our approach are much less affected. Current smartphones have several mechanisms to adjust CPU use and consistently high use is rare. Accordingly, while the power monitor estimates reflect actual battery consumption, they are over-zealous and overfit on the instantaneous consumption. In contrast, our approach can average over different usage contexts, providing a more realistic estimate of the practical impacts on battery consumption. Second, the results demonstrate that power monitor measurements are poor at identifying the relative importance of different context factors. For low CPU use, the difference between manual and automatic screen brightness is clearly observable. However, as CPU use increases, the power monitor models overfit on the high power consumption of CPU, providing limited information about the importance of other context factors. In contrast, the impacts of changing screen brightness to automatic and the impact of improving Wi-Fi signal level remain observable across all CPU use levels for our approach. In Fig. 3, we can observe that these differences are even more observable from the conditional mutual information.

D. Highlights and Example Cases

As the final step of analysis, we demonstrate how our approach can also be used to obtain new insights into battery consumption. Examples of selected context factors' impact on battery consumption are listed in Table VI. We have selected CPU use and temperature from subsystem variables, and distance (motion or stationary) and screen brightness from system settings. In all examples, connection type has been a cellular data connection. Table VI presents the estimated time to drain the battery from 100% to 0%, while actively using a smartphone with the given context factor and value combination. With different values of CPU use, battery temperature, movement, and screen brightness, the battery life can range from 3.45 hours up to 9.12 hours.

The table is sorted by the time to drain the battery, descending. We can see that the main deciding factor for battery life is the temperature of the battery. With a lower temperature, we get a higher battery life. After that, traveling instead of staying still seems to increase battery life. This may be due to users driving and not using their mobile phones while mobile. After these factors, the CPU is the most dominant, and changing screen brightness brings the smallest, but still significant, battery life differences. These results show that while CPU use alone is a good indicator of energy consumption, significant gains can be obtained by considering more complex combinations. In addition to this, battery temperature and distance traveled can be used together to predict battery life.

Complex combinations of factors, such as those listed in Table VI, can be used to decide which factors to change to improve battery life, while keeping others constant. For example, while moving and playing a game, the CPU is often high. If the phone can be kept relatively cool, 78% more battery life can be expected compared to warmer battery (increase from 4.08h to 7.27h). Further savings can be obtained by switching screen brightness to automatic.

Battery Temperature	Distance Traveled	CPU Use	Screen Brightness	Estimated Battery Life (h)
Under 30°C	>0	Low	Automatic	8.83 – 9.12
Under 30°C	>0	Low	Manual	8.49 – 8.82
Under 30°C	>0	High	Automatic	8.09 – 8.24
Under 30°C	>0	Medium	Automatic	7.65 – 7.89
Under 30°C	>0	Medium	Manual	7.34 – 7.60
Under 30°C	>0	High	Manual	7.27 – 7.41
Under 30°C	None	Medium	Automatic	6.57 – 6.64
Under 30°C	None	Low	Automatic	6.28 – 6.35
Under 30°C	None	Medium	Manual	6.13 – 6.20
Under 30°C	None	Low	Manual	5.88 – 5.96
Under 30°C	None	High	Automatic	5.78 – 5.82
Over 30°C	>0	Low	Automatic	5.08 – 5.22
Under 30°C	None	High	Manual	5.00 – 5.04
Over 30°C	>0	Low	Manual	4.73 – 4.88
Over 30°C	>0	High	Automatic	4.62 – 4.69
Over 30°C	>0	Medium	Automatic	4.59 – 4.70
Over 30°C	>0	Medium	Manual	4.28 – 4.39
Over 30°C	None	Medium	Automatic	4.25 – 4.29
Over 30°C	>0	High	Manual	4.08 – 4.14
Over 30°C	None	Medium	Manual	4.06 – 4.09
Over 30°C	None	Low	Automatic	4.02 – 4.06
Over 30°C	None	High	Automatic	3.91 – 3.94
Over 30°C	None	Low	Manual	3.74 – 3.78
Over 30°C	None	High	Manual	3.45 – 3.46

TABLE VI
BATTERY LIFE IN HOURS FOR SELECTED COMBINATIONS OF FOUR CONTEXT FACTORS.

With respect to the worst possible configuration, moving to a cooler place (45% battery life gain) and changing screen brightness without changing the CPU use can result in a battery life increase from 3.45h to 5.78h (68%). Our results in Table VI and in Fig. 2 show that the battery temperature is not always directly related to CPU use. High battery temperature can be caused, for example, by the ambient temperature in warmer countries, battery misbehavior or a battery bug, or because the smartphone has been forgotten under the windshield inside a car on a sunny day. Battery temperature alone can shorten the battery lifetime even by 50%. If cooling the device is not possible, because of the ambient climate, for example, re-configuring other context factors can help to improve the battery lifetime.

With low CPU use and a cool battery, no movement, and manual screen brightness, we can obtain an active battery life of 6 hours, which improves to almost 9 hours by only changing movement. That behavior can be caused by the users mostly walking or driving a car and not using their smartphones while moving from place to another. It is also possible, that energy saving policies activate as movement requires re-connections to the cellular base stations. As Table IV shows in Section V-A, distance traveled is ranked high together with CPU use. It is possible that the most CPU heavy actions, such as gaming, are only done in longer periods while stationary.

VI. DISCUSSION AND SUMMARY

The present paper has provided three contributions. Our first contribution has been the development of a novel approach for constructing energy models using crowdsourced battery discharge measurements. Contrary to previous works on energy modeling, our approach is not restricted to capturing the effects of individual sensors, features or system settings, but can capture complex interdependencies between all of these. As we have experimentally demonstrated, estimates provided by our approach are in line with battery meter measurements, providing an accurate view of the energy state of the device. The second contribution is a large-scale analysis of the influence of different system settings on battery consumption. Our analysis validated our method and confirmed findings in previous studies. It also provided novel insights about battery consumption and quantified their effects. For example, we demonstrated that a Wi-Fi signal strength drop of one bar can result in a battery life loss of over 13% and that a smartphone sitting in the sun can experience over 50% worse battery life than one indoors in cool conditions. As our third contribution, we have made available the large-scale (anonymized) dataset used in our analysis[6].

Energy models that can accurately capture the energy state of a device and that can estimate how system state changes influence energy, are beneficial for several reasons. Our approach can be used to bootstrap and support battery management interfaces developed to support end users. Instead of merely allowing users to switch off (or on) different settings, our approach can estimate how these changes are expected to influence device lifetime. Our approach can be used to construct device-specific resource optimization strategies that can estimate changes in battery use more accurately. Our approach could be used to construct empirical energy models for comparing and evaluating energy-effectiveness of different sensing strategies.

In terms of battery management interfaces, an interesting avenue of investigation are task-based recommendations that provide actionable feedback to the user on how to preserve battery for her current tasks. For example, if the user intends to perform high CPU use activities, they can save battery life by setting screen brightness to automatic or moving to an area

[6]http://carat.cs.helsinki.fi/research

with a better Wi-Fi signal. Such recommendations can also help to increase the user's knowledge over time, familiarizing them with the inner workings of their smartphone. Similarly, if personal measurements would be available, our approach could be used to identify "bad" behaviors for a user and provide guidance on how to mitigate these. Another benefit of our approach is the capability to construct device or OS-specific energy models with minimal effort. The energy consumption of sensors and system settings can contain significant variations across platforms, e.g., Bhattacharya et al. [2] reported over 200% differences for GPS power consumption on two different Nokia smartphone models. Accordingly, we can tailor the guidance given to the user according to the model and operating system version of her device.

The results presented in this paper are also potentially beneficial for understanding long-term effects of sensor and battery management strategies on battery life. The comparison of battery life estimates between our approach and power monitor measurements showed that our approach can average effects over different usage contexts, whereas empirical power models tend to focus on instantaneous effects. As the overall state of a smartphone is complex, and in constant flux, instantaneous estimates tend to result in overestimates of battery consumption. Assessing the benefits of using crowdsourced battery models for these purposes is another interesting venue for future investigations.

ACKNOWLEDGMENTS

The work of Ella Peltonen has been supported by Doctoral School of Computer Science (DoCS). This research was partially supported by the Academy of Finland grant 277498. The publication only reflects the authors' views. The authors are grateful to Dr Stephan Sigg, Teemu Pulkkinen, and Samuli Hemminki for comments on earlier versions of the paper.

REFERENCES

[1] S. Agarwal, R. Mahajan, A. Zheng, and V. Bahl. Diagnosing mobile applications in the wild. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 22:1–22:6, New York, NY, USA, 2010. ACM.

[2] S. Bhattacharya, H. Blunck, M. Kjærgaard, and P. Nurmi. Robust and energy-efficient trajectory tracking for mobile devices. *IEEE Transactions on Mobile Computing*, 99:1, 2014.

[3] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 2011.

[4] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010)*, pages 179–194. ACM, 2010.

[5] D. Ferreira, A. K. Dey, and V. Kostakos. Understanding human-smartphone concerns: A study of battery life. In *Proceedings of the 9th International Conference on Pervasive Computing*, 2011.

[6] J.-M. Kang, C.-K. Park, S.-S. Seo, M.-J. Choi, and J. Hong. User-centric prediction for battery lifetime of mobile devices. In *Proceedings of the 11th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 531–534. Springer, 2008.

[7] M. B. Kjærgaard, S. Bhattacharya, H. Blunck, and P. Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys 2011)*, pages 307–320, New York, NY, USA, 2011. ACM.

[8] M. B. Kjærgaard and H. Blunck. Unsupervised power profiling for mobile devices. In *Proceedings of the 8th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous)*, 2011.

[9] I. Koenig, A. Q. Memon, and K. David. Energy consumption of the sensors of smartphones. In *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013)*, pages 1–5, Aug 2013.

[10] D. Linden and T. B. Reddy. *Handbook of Batteries*. McGrew-Hill Professional, 2001. 3rd edition.

[11] Y. Liu, C. Xu, and S. C. Cheung. Where has my battery gone? Finding sensor related energy black holes in smartphone applications. In *2013 IEEE International Conference on Pervasive Computing and Communications, PerCom 2013, San Diego, CA, USA, March 18-22, 2013*, pages 2–10, 2013.

[12] X. Ma, P. Huang, X. Jin, P. Wang, S. Park, D. Shen, Y. Zhou, L. K. Saul, and G. M. Voelker. eDoctor: Automatically diagnosing abnormal battery drain issues on smartphones. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*, pages 57–70, Berkeley, CA, USA, 2013. USENIX Association.

[13] A. J. Oliner, A. P. Iyer, I. Stoica, E. Lagerspetz, and S. Tarkoma. Carat: Collaborative energy diagnosis for mobile devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, pages 10:1–10:14, New York, NY, USA, 2013. ACM.

[14] A. Pathak, Y. C. Hu, and M. Zhang. Where is the energy spent inside my app? Fine grained energy accounting on smartphones with Eprof. In *Proceedings of the 7th ACM European Conference on Computer Systems*, pages 29–42, New York, NY, USA, 2012. ACM.

[15] N. Ravi, J. Scott, L. Han, and L. Iftode. Context-aware battery management for mobile phones. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 224–233, 2008.

[16] A. Rice and S. Hay. Decomposing power measurements for mobile devices. In *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 70–78. IEEE, 2010.

[17] A. Schulman, T. Schmid, P. Dutta, and N. Spring. *Demo: Phone Power Monitoring with BattOr*, 2011. ACM Mobicom 2011. Available at http://www.stanford.edu/ aschulm/battor.html.

[18] A. Shye, B. Scholbrock, and G. Memik. Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 168–178, New York, NY, USA, 2009. ACM.

[19] K. N. Truong, J. A. Kientz, T. Sohn, A. Rosenzweig, A. Fonville, and T. Smith. The design and evaluation of a task-centered battery interface. In *Proceedings of the12th International Conference on Ubiquitous Computing*, 2010.

[20] N. Vallina-Rodriguez and J. Crowcroft. Energy Management Techniques in Modern Mobile Handsets. *IEEE Communications Surveys and Tutorials*, 15:179–198, 2013.

[21] C. H. K. van Berkel. Multi-core for mobile phones. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '09)*, pages 1260–1265, 2009.

[22] D. T. Wagner, A. Rice, and A. R. Beresford. Device Analyzer: Large-scale mobile data collection. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):53–56, 2014.

[23] Y. Wen, R. Wolski, and C. Krintz. Online prediction of battery lifetime for embedded and mobile devices. In *Proceedings of the 3rd International Workshop on Power-Aware Computer Systems (PACS)*, pages 131–138. Springer, 2005.

[24] F. Xu, Y. Liu, Q. Li, and Y. Zhang. V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, pages 43–56, Berkeley, CA, USA, 2013. USENIX Association.

[25] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the 8th IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 105–114, New York, NY, USA, 2010. ACM.

[26] X. Zhao, Y. Guo, Q. Feng, and X. Chen. A system context-aware approach for battery lifetime prediction in smart phones. In *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC)*, pages 641–646. ACM, 2011.

Research Theme A: Mobile Energy Consumption

Research Paper II

II

Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma

Constella: Crowdsourced System Setting Recommendations for Mobile Devices

Published in Pervasive and Mobile Computing, Volume 26, February 2016, pages 71 - 90.

Contribution: The publication extends Publication I with a novel recommendation system for energy consumption of system settings and subsystem variables. Some parts of the work is based on the author's Master's Thesis published in 2013 at the University of Helsinki. The author was responsible for implementing the decision tree-based recommendation system, perform the data analysis procedures, and write the publication. Dr Eemil Lagerspetz, Dr Petteri Nurmi, and Prof. Sasu Tarkoma contributed to the planning and writing process of the publication.

Pervasive and Mobile Computing 26 (2016) 71–90

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Constella: Crowdsourced system setting recommendations for mobile devices

Ella Peltonen [b,*], Eemil Lagerspetz [b], Petteri Nurmi [a,b], Sasu Tarkoma [a,b]

[a] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland
[b] Department of Computer Science, University of Helsinki, PO Box 64, FI-00014, Finland

ARTICLE INFO

Article history:
Available online 30 October 2015

Keywords:
Mobile sensing
Energy-awareness
Energy modeling
System settings

ABSTRACT

The question "Where has my battery gone?" remains a common source of frustration for many smartphone users. With the increased complexity of smartphone applications, and the increasing number of system settings affecting them, understanding and optimizing battery use has become a difficult chore. The present paper develops a novel approach for constructing *energy models* from *crowdsourced* measurements. In contrast to previous approaches, which have focused on the effect of a specific sensor, system setting or application, our approach can simultaneously capture relationships between multiple factors, and provide a unified view of the energy state of the mobile device. We demonstrate the validity of using crowdsourced measurements for constructing battery models through a combination of large-scale analysis of a dataset containing battery discharge and system state measurements, and hardware power measurements. The results indicate that the models captured by our approach are both in line with previous studies on battery consumption and empirical measurements, providing a cost-effective way to construct energy models during normal operations of the device. The analysis also provides several new insights about battery consumption. For example, our analysis reveals the combined effect of high CPU activity and automatic screen brightness to be higher (resulting in 9 min shorter battery lifetime on average) than the effect of medium CPU load and manual screen brightness; a Wi-Fi signal strength drop of one bar can shorten battery life by over 13%; and a smartphone sitting in direct sunlight can witness over 50% shorter battery life than one indoors in cool conditions. Based on the crowdsourced energy models, we develop Constella, a novel recommender system for system settings. Constella provides actionable and human-readable recommendations on how to adjust system settings in order to reduce overall battery drain. We validate the effectiveness of Constella through a hardware power measurement experiment carried out using three application case studies. The results of the evaluation demonstrate that Constella is capable of generating recommendations that can provide up to 61% improvements in battery life.

1. Introduction

Smartphones have become commodity devices, with most of us using one daily for work, entertainment and a variety of other purposes besides communication [1]. The processing and transmission power of smartphones continues to grow [2],

* Corresponding author.
E-mail addresses: ella.peltonen@cs.helsinki.fi (E. Peltonen), eemil.Lagerspetz@cs.helsinki.fi (E. Lagerspetz), petteri.Nurmi@cs.helsinki.fi (P. Nurmi), sasu.Tarkoma@cs.helsinki.fi (S. Tarkoma).

http://dx.doi.org/10.1016/j.pmcj.2015.10.011
1574-1192/

Fig. 1. The number of system settings available on current smartphones can be overwhelming.

while their batteries remain largely unchanged [3]. Consequently, energy efficiency remains a high priority for current smartphone operating systems, and increasingly, for applications. The importance of energy efficiency has also been highlighted in several user studies, which have shown that users actively take measures to optimize the power consumption of their device [4,5].

Modern smartphones incorporate several mechanisms for optimizing battery consumption. On the operating system level, complex on-demand resource optimization strategies are used to reduce battery consumption [6]. However, the effectiveness of these policies is highly dependent on the context of the user, and there often are complex interdependencies that make it difficult to determine the optimal policy for a given situation. Ensuring the effectiveness of these policies requires fine-grained models that can characterize how different contexts and device features influence the power consumption of the device.

The alternative to automated policies is to give users control over specific system settings, such as whether to prefer Wi-Fi or cellular networks, which screen brightness value to use, and when to turn off the screen after inactivity. Indeed, contemporary smartphones have interfaces that allow this kind of operations with little effort. With an increasing number of user-controllable system settings, keeping track of each setting's energy impact becomes unmanageable. To illustrate this problem, Fig. 1 depicts some of the system settings available on the Samsung Galaxy S4. Over 20 different settings are visible, most of which have a significant effect on energy. To fully understand the impact of all of the settings would require a considerable amount of learning about the device. Furthermore, the total energy consumption of the smartphone is not simply the sum of the energy impacts of enabled system settings. Some subsystems, such as Bluetooth and Wi-Fi, are integrated on the same chip, and can be enabled simultaneously at much less than the sum of their combined individually measured energy impacts. Another example is the integration of accelerometers and gyroscopes on the same chip to provide energy savings for activity monitoring applications. Enabling users to make optimal decisions would thus require fine-grained information about how different settings influence the overall battery consumption of the device in their current context.

In the present paper we develop a novel approach for constructing *energy models* from crowdsourced battery discharge measurements [7]. Such information can be increasingly collected through non-obtrusive instrumentation of the device [8], which in turn enables capturing battery information across a wide range of usage contexts and devices. Experiments conducted through a combination of power meter measurements and a large-scale analysis of crowdsourced discharge measurements demonstrate that our method is capable of constructing models that accurately capture complex interdependencies between system settings, sensors, and usage contexts, providing an accurate view of the *state* of the device. This contrasts with previous works, which have predominantly focused on capturing the effect of a specific sensor, system setting or application [9,10]. Results from our evaluation provide novel insights about battery usage, demonstrating how complex the relationships between different factors and battery discharge are. For example, our analysis reveals the combined energy use of high CPU activity and automatic screen brightness to be higher (resulting in 9 min shorter battery lifetime on average) than the effect of medium CPU load and manual screen brightness; a Wi-Fi signal strength drop of one bar can shorten battery life by over 13%; and a smartphone exposed to direct sunlight can witness over 50% shorter battery life than one indoors in cool conditions.

Based on the crowdsourced models, we develop Constella, a novel recommender system for system settings, subsystem variables, and other context factors. Constella analyzes the current state of the device and compares it against our energy models. Based on the analysis, Constella provides suggestions on how to adjust system settings or how to manipulate sensor

states in order to improve the energy-efficiency of the device. We validate the effectiveness of Constella using hardware power measurement experiments of three carefully selected test applications. The results of our evaluation demonstrate that, in each of the three application scenarios, Constella is capable of identifying system states or subsystem variables that result in reduced battery drain.

The contributions of the paper are summarized as follows:

- We develop a novel approach for constructing energy models from crowdsourced measurements. The models constructed by our approach can capture the *combined* effects of multiple factors simultaneously, providing a characterization of the *energy state* of a mobile device.
- Experiments carried out through a combination of power meter measurements and a large-scale analysis of crowdsourced discharge measurements demonstrate that our approach can capture the state of the device accurately and cost-effectively, even in the presence of complex interdependencies between context factors. Through our analysis, we reveal novel insights about battery consumption which highlight the complexity of factors that influence battery consumption.
- We develop Constella, a novel recommender system for providing suggestions on how to optimize the energy footprint of context factors in a given context. Experiments carried out using power meter measurements and three application case studies demonstrate that Constella provides recommendations that can improve the active lifetime of the device by up to 61%.
- We make available a large-scale dataset of 11.2 million data points from around 150,000 active Android users,[1] a subset of the Carat data.

2. Related research

Our work focuses on developing models for characterizing the energy consumption of system settings based on crowdsourced data, and on using these models to provide users with actionable recommendations on how to prolong the operational time of their mobile device. The former task is an example of *mobile energy profiling*, whereas the Constella recommender system proposed for the second task is an example of a *mobile energy diagnostics system*. In the following we summarize previous research on these topics.

2.1. Mobile energy profiling

Mobile energy profiling refers to the process of characterizing the energy consumption of a mobile device, including installed applications, hardware, and other subsystem components. Energy profiling is typically carried out by constructing a statistical model that can correlate specific system states with energy consumption patterns. The measurements for constructing these models can be gathered either using specialized hardware, such as the Monsoon power monitor or BattOr [11], or through the battery interface of the device [8,12,13].

Approaches for energy profiling of mobile devices can be categorized into two based on the target of the modeling process. In *sensor-level* models, the goal is to characterize the energy consumption of a specific set of sensors. Overall energy consumption can then be estimated by combining the model with usage statistics of different sensors. In one of the earliest works, Rice and Hay [9] perform fine-grained hardware power measurements of the network interface and analyze their causes, attributing energy drain to the networking stack version, packet size, and Wi-Fi handshake behavior. König et al. [14] measure power consumption of different sensors using a hardware power monitor. Kjærgaard et al. [15] use conditional functions, manually constructed from empirical power measurements, to represent power consumption of different sensors. Kjæ rgaard and Blunk [16] propose using genetic algorithms for learning the conditional functions in an unsupervised manner. Contrary to our work, these approaches are capable of capturing complex interdependencies in the energy consumption of different components. Moreover, the resulting models are necessarily inaccurate as the complete state of the device requires considering all subsystems that can use energy.

Instead of focusing on individual subsystems, *device-level* models characterize the overall battery consumption of a device. These models consider how application usage patterns, workload and other system level parameters, such as screen brightness and data transfer rate influence battery discharge. The estimated discharge rate of the device can then be used to estimate remaining battery lifetime. Wen et al. [17] propose constructing a reference curve of the battery consumption under different workloads. Once the reference curve has been constructed, a regression model is used to compare current discharge against an estimate given by the reference curve. Deviations from the reference curve can then be used to refine estimates of the remaining battery lifetime. Instead of considering workload, Kang et al. [18] predict discharge behavior from application usage patterns. Liu et al. [19] analyze application runtime to detect sensor misbehavior. Pathak et al. [20] model resource usage (networking, user tracking, game rendering, etc.) of a set of applications. Zhao et al. [21] predict battery discharge using a regression model that considers multiple different system variables (e.g., CPU utilization, I/O rate and LCD backlight brightness). In addition to predicting battery discharge, Ravi et al. [22] predict when the user is likely to

[1] http://carat.cs.helsinki.fi/research

have the next charging opportunity and how much battery power is needed for maintaining essential functionality until then. If the system detects that the battery is likely to run out before the next charging opportunity, the system pro-actively provides a warning to the user instead of waiting for the battery to be nearly depleted. Falaki et al. [10] conduct an analysis of smartphone usage patterns, revealing that usage patterns contain significant variation across users and that personalized application usage models are essential for accurate prediction of battery drain. In contrast to our approach, which can capture how changes in device state influence battery discharge, these approaches can only provide aggregate level information of energy usage.

Agarwal et al. build in MobiBug [23] a data-driven approach for energy diagnosis. The Device Analyser project [24] is gathering rich measurements of mobile device state, but the data has not yet been used for large-scale analysis, and a high sampling cycle (even 100,000 per day from a single device) can lead to unexpected and increased energy consumption. To our best knowledge, no previous works are capable of constructing fine-grained energy models from crowdsourced measurements.

2.2. Mobile energy diagnostics system

Studies on battery charging behavior have shown that users actively seek countermeasures to prolong the operational lifetime of their device, but at the same time have difficulties in identifying which countermeasures are the most appropriate in a given situation [25–28]. To facilitate the users in selecting the appropriate countermeasures, several so-called mobile energy diagnostic systems have been proposed. The goal of these systems is to identify energy bottlenecks at runtime and to provide suggestions on how the lifetime can be improved.

Existing work on energy diagnostics has predominantly focused on individual features or applications. For example, Pathak et al. [29] developed a methodology to detect applications that prevent the device to sleep or idle. Banerjee et al. [30] present an approach for detecting energy bugs in applications and propose a tool to notify developers about the bugs. Ma et al. [31] develop an application called eDoctor which attempts to detect abnormal battery drain by tracing system calls. Another example is Carat, which uses collaborative analytics to identify misbehaving or otherwise problematic applications [8]. Our research is the first to focus on the effects of system settings and subsystem variables, and to consider the entire state of the device.

Instead of suggesting users countermeasures, some previous works have focused on automatically optimizing individual features to preserve battery. As an example, Shye et al. [32] name screen brightness and CPU load as major energy consumption features, and present a system that slowly reduces the value of screen brightness and CPU frequency over time. While this kind of approach offers to the users an easy way to control energy usage, it limits user control and is not suited for situations where high brightness or CPU load are needed. In contrast, our approach preserves the user's control over her device and provide alternative ways to reconfigure the device to preserve battery.

Several academic and commercial recommendation systems that focus on suggesting new applications to the end users have been proposed. In contrast to our work, these works typically operate exclusively on top of a cloud backend, requiring data connectivity and relying on computationally intensive matrix factorization methods [33]. Furthermore, system settings may need to be adjusted more often than new applications downloaded. For example, when user changes location from outdoors to indoors, that may lead to remarkable changes in network connection types available, and changes in ambient light may make it reasonable to reconfigure screen brightness. Rapidly changing context of smartphone usage makes it important to produce also recommendations in real time, without redundant computing operations or network traffic to the supporting server. Our approach in the Constella takes these questions into account.

3. Background: Dataset

We consider a large-scale dataset of crowdsourced battery discharge measurements collected from a collaborative energy diagnostic system Carat [8]. The application has collected data from over 800,000 Android and iOS devices since summer 2012. We consider a subset of the data containing 11.2 million samples from around 150,000 active Android devices. The dataset used in our work has been made publicly available for research purposes.[2] The data includes information about the device's operating system and model, the current battery level, the set of currently active applications, and information about different system settings such as network connections and screen brightness and subsystem variables such as the CPU use and the distance traveled since the last measurement. We refer to these system settings and subsystem variables collectively as *context factors*.

As measure of energy consumption, we consider *energy rates* reported by Carat. These reflect normalized energy consumption per time unit, i.e., energy rate $= \Delta$ battery$/\Delta t$. The methodology used to derive rates and the validity of using energy rates as a measure for battery consumption has been shown in previous work by Oliner et al. [8].

2 http://carat.cs.helsinki.fi/research

Table 1
Summary statistics of selected context factors.

Context factor	Mean	Std	Median
CPU use	75%	33%	91%
Battery voltage (V)	3.78	0.61	3.84
Screen brightness (0–255)	128.03	85.71	109
Temperature (°C)	29.27	5.75	30
Wi-Fi signal strength (dBm)	−61.29	13.02	−61

Context factors

We focus on 13 different context factors, including 5 user-changeable system settings and 8 subsystem variables. The factors were chosen based on previous studies on energy-efficiency, which have shown them to be dominant factors in explaining battery consumption. We consider the status defined by all 13 factors as the *state* of the device. The 5 system settings that we have collected via the Android API and that we utilize in this work are:

- **Mobile data status** describes current status of the mobile data interface. It is given as a categorical attribute and has one of the following values: connected, disconnected, connecting, or disconnecting.
- **Mobile network type** is a categorical attribute that specifies the mobile data transfer standard currently being used on the phone. Examples of values it can take include LTE, HSPA, GPRS, EDGE, and UMTS. The list of possible mobile networks is broad and depends on the technologies available in each country and by each operator.
- **Network type** is a categorical attribute describing the current method used by the phone for Internet connectivity, for example, none, Wi-Fi, mobile, or WiMAX, depending on the technology used. When the network type equals mobile, detailed information about the connectivity type is given by the attribute mobile network type.
- **Roaming** describes whether mobile network traffic outside of the own operator network is allowed. The value of the attribute is either disabled or enabled, given as a binary attribute (0 or 1).
- **Screen brightness** refers either to a manually adjusted brightness value, given as a numeric value between 0 and 255 where larger value implies brighter screen, or automatic setting, given by value -1. The automatic setting can vary in some devices, for example, based on the sensing of the outside light. Therefore, knowing the setting parameter may not give the actual brightness value being used currently.

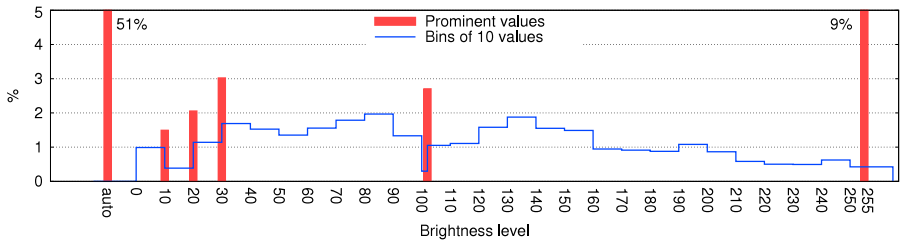
We also consider 8 subsystem variables, which are not directly available as a user-modifiable system setting, but can give information about the state of the smartphone. For example, if we notice a decreased Wi-Fi link speed or signal strength, we can recommend that the user try to use the mobile network instead of Wi-Fi in this context. The subsystem variables we consider are:

- **Battery health** is a categorical attribute determined by the smart battery interface of the device. Values of the attribute are vendor-specific, with examples of common values being Good, Bad, Overheat, and Unknown failure. The value Good is the most common value in our data.
- **Battery temperature** is the temperature of the battery given in degrees (Celsius).
- **Battery voltage** describes the current battery capacity in Volts.
- **CPU use** is a percentage (0%–100%) that describes the fraction of CPU currently used.
- **Distance traveled** is a location-based measurement between two samples, given in meters. For privacy reasons, the Carat application does not gather the exact location of the user, but uses distance measurements to determine whether the device has been moving or not, for example, in a car.
- **Mobile data activity** describes how mobile data interface is being used. The value of this categorical attribute is one of the following: none, out, in, or inout.
- **Wi-Fi link speed** is given in Mbps and is determined by the Android API.
- **Wi-Fi signal strength** is given in dBm and is determined by the Android API.

Discretization into categorical variables

Most of the context factors are ordinal-valued. To simplify the comparison of these factors, we discretize them into categories using an equal frequencies procedure, i.e., each factor is divided into categories containing approximately the same number of values. The number of categories was determined empirically and based on observations reported in previous battery usage studies. Summary statistics of selected context factors are given in Table 1 and the different categories are detailed below. For nominal variables (such as network type), we have used the different possible values as the categories.

- **CPU use:** We consider measurements that reflect the percentage of time CPU is active. The mean and median in Table 1 indicate that CPUs are mostly active. We split the CPU use around the mean, resulting in three categories: Low (0%–42%), Medium (43%–85%), and High (86%–100%).
- **Distance traveled**: Most of the values are during stationary periods or with little movement. Based on this observation, we consider a split between stationary and non-stationary behavior.



Setting	%
Automatic	51%
10	1%
20	2%
30	3%
102	2%
255	9%
Other	32%

Fig. 2. The frequency of all screen brightness settings. Values other than those listed in the table to the right are shown in bins of 10 values.

- **Battery voltage**: The safe operating voltage of a smartphone Li-Ion battery is 3–4.2 V. The nominal voltage of such batteries is typically 3.7 V. The mean, the median, and the standard deviation reflect this very closely. We consider three categories for voltage: Low (0–3 V), Medium (3–4.2 V), High (4.2 V+).
- **Screen brightness**: When screen brightness was manually controlled, the mean was around 128, or the exact midpoint. The distribution of the values, shown in Fig. 2, indicates that almost the entire range of screen brightness values is used, making it difficult to categorize screen brightness values in a meaningful way. While small brightness values generally have lower energy impact than higher values, or even automatic settings, they usually occur only in specific situations, such as at night or while reading a book in a dimly lit room. As these values are encountered very infrequently, their overall energy benefits are small compared to using automatic setting. Based on these observations, we opt for a binary split into manual and automatic brightness.
- **Wi-Fi signal strength**: We consider RSS values in the range [−100, 0]. Good Wi-Fi signal strength values are normally between −30 and −10 dBm, and the worst, while still being connected, is −95 dBm. We consider four categories: Bad (−100 to −75 dBm), Average (−74 to −61 dBm), Good (−60 to −49 dBm) and Excellent (−49 to 0 dBm). The mean RSS is between the Average and the Good levels, and the Excellent and the Bad levels are within one standard deviation. These values are in line with typical values used in Wi-Fi positioning literature.

Ethical considerations

In our analysis, we only consider aggregate-level data which contains no personally identifiable information. The privacy protection mechanisms of Carat are detailed in [8]. Data collection by Carat is subject to the IRB process of University of California, Berkeley. Users of Carat are informed about the collected data and give their consent from their devices.

4. Battery modeling from crowdsourced measurements

Battery consumption has traditionally been based on empirical models taken either directly on the battery level [11] or through system-level APIs. The former requires specialized measurements tools, limiting the contexts where measurements can be taken. The latter, on the other hand, has been shown to result in inaccuracies in the resulting models [12]. In this section we demonstrate the validity of using crowdsourced battery discharge measurements for constructing battery consumption models. Our approach provides a cost-effective alternative for modeling battery consumption, and, as we later demonstrate, our approach can capture complex interdependencies affecting battery consumption in everyday use.

4.1. Methodology

We construct battery models by measuring the strength of statistical association between context factors and battery discharge rates. To measure statistical association, we consider two complementary metrics. As our first metric, we consider gain in battery life, denoted *BL Gain*, which measures how changes in context factors influence the lifetime of a device on average. As our second measure, we consider the *conditional mutual information* (CMI) between context factors and energy rates. For assessing the influence of a single context factor X and energy rate Z, the CMI is equivalent to the *mutual information* (MI) given by:

$$MI(X, Z) = \sum_{z \in Z} \sum_{x \in X} p(x, z) \cdot \log \left(\frac{p(x, z)}{p(x) \cdot p(z)} \right).$$

For higher order combinations containing two or more context factors (denoted X and Y) and energy rates (denoted Z), the *conditional mutual information* (CMI) is defined as follows:

$$CMI(X, Y|Z) = \sum_{z \in Z} \sum_{y \in Y} \sum_{x \in X} p(x, y, z) \cdot \log \left(\frac{p(z) \cdot p(x, y, z)}{p(x, z) \cdot p(y, z)} \right).$$

Table 2
Context factors' impact on energy consumption, ordered by mutual information estimate.

Context factor	MI estimate
CPU use	1.330
Distance traveled	1.069
Battery temperature	0.143
Battery voltage	0.099
Screen brightness	0.030
Mobile network type	0.019
Network type	0.018
Wi-Fi signal strength	0.014
Wi-Fi link speed	0.014
Mobile data status	0.013
Mobile data activity	0.005
Battery health	0.004
Roaming	0.0002

By using CMI to analyze the impact of combinations, we can identify combinations that are as informative as possible while at the same time minimizing redundancy between the different factors. Accordingly, our methodology can be understood analogously to the use of CMI for feature selection in machine learning research [34].

The battery life gain measurements provide information about *absolute differences*, whereas the (conditional) mutual information measurements can be used for *relative* comparison between different context factor combinations. The two metrics provide complementary ways to analyze strength of associations, and in applications the choice of metric depends on the scenario being considered.

4.2. Individual context factors

We demonstrate the validity of using battery discharge measurements for constructing energy models by examining the mutual information between context factors and energy rates. We derive a ranking for different factors based on their mutual information values, and demonstrate that this ranking is in line with findings from empirical studies on battery consumption. Estimations by mutual information *MI* of context factors and energy consumption are given in Table 2. The results of the *MI* analysis are well in line with previous results [10,32]. In particular, the major individual impact of CPU use and traveled distance on battery consumption is clearly observable, and the ordering of the settings is similar to those derived through explicit battery measurements.

The results also contain some exceptions to the findings of previous studies. The most prominent example is screen brightness, which is commonly considered as the most battery heavy feature. In our analysis, screen brightness results in a lower score than many other attributes. In the next section, we demonstrate that the absolute energy impact of screen brightness actually is high. However, as the mutual information effectively looks at the correlation between battery discharge and context factors, the changes are affected by other context factors. In most use contexts screen use is correlated with battery voltage and CPU use, both of which have a large impact on the battery drain, and hence also on the mutual information values. The main effect we observe for screen brightness comes from switching to automatic brightness instead of using a manual setting.

4.3. Energy consumption of context factors

We next consider how typical values of context factors influence battery discharge. We focus on the five factors discussed in Section 4.1 (CPU use, battery voltage, screen brightness, temperature, and Wi-Fi signal strength) and consider expected gain in battery life as our evaluation measure. The results of this evaluation are shown in Table 3.

In line with the results of mutual information analysis, the worst battery life is obtained for high CPU use. The benefit of maintaining a balanced CPU load is significant, as medium CPU use produces +5.72% energy benefit compared to average use. For screen brightness, the automatic setting of the device usually improves battery life, providing even +6.29% better battery life compared to the average. Manual brightness, in contrast, shows a major loss of battery life (−4.97%).

We can make a number of other observations from the results. First, higher battery voltage results in improved battery life. This is partially explained by voltage correlating with battery health and capacity. In addition, the rates reported by Carat are based on changes in battery percentage, which tend to follow voltage changes linearly. This contrasts with actual discharge, which is nonlinear, particularly when the battery is close to full charge. The results also suggest that battery life tends to be higher for mobile than for stationary users. Studies on application usage have shown that interactions with applications are common during mobility, with web browsing, news, music/video players, and gaming being the dominant application categories [1]. Hence, the difference is likely a result of shorter interaction periods rather than avoidance of energy intensive operations. Finally, a high Wi-Fi signal strength leads to better battery life, as the phone needs to spend less energy for receiving and sending data. Bad signal availability can lead to situations where the device has to reconnect to the network repeatedly, further increasing battery consumption.

Table 3
The expected energy use typical values of context factors. BL gains can be compared to statistics in Table 1.

Context factor	Value	BL gain
CPU use	Low (0%–42%)	+3.24%
CPU use	Medium (43%–85%)	+5.72%
CPU use	High (86%–100%)	−2.48%
Distance traveled	None	−0.76%
Distance traveled	>0	+8.20%
Battery voltage	Low (0–3 V)	−16.60%
Battery voltage	Medium (3–4.2 V)	−0.76%
Battery voltage	High (4.2 V+)	+69.08%
Screen brightness	Manual	−4.96%
Screen brightness	Automatic	+6.29%
Wi-Fi signal strength	Bad (−100–75 dBm)	−2.29%
Wi-Fi signal strength	Average (−74–61 dBm)	+4.00%
Wi-Fi signal strength	Good (−60–49 dBm)	+6.29%
Wi-Fi signal strength	Excellent (−48–0 dBm)	+7.63%

5. Context factor combinations

The results of our analysis thus far show that estimates given by our method are in line with observations made in studies carried out in laboratory conditions with specialized hardware measurement tools, providing a strong indication of the potential of using our approach as a cost-effective mechanism to construct models of battery consumption. However, a limitation of our analysis thus far has been the focus on individual factors' impact on battery life without considering the state of the device as a whole. As an example, consider the case of screen brightness, which according to previous studies is one of the main battery hogs on a smartphone. In terms of expected battery gain, our results also support this observation, indicating over 5% deviations from average consumption patterns. In actual application contexts, screen usage is highly correlated with interactions on the device, which in turn requires CPU and network usage, suggesting that screen brightness is not the *dominant* factor explaining battery usage. To capture such nuanced differences in consumption, we argue that models reflecting the *state* of the smartphone are required. In the following we demonstrate the validity of considering combinations of context factors as part of battery models. We also compare our results against empirical power models constructed using a hardware power monitor, demonstrating that our approach can capture more fine-grained differences in battery consumption than empirical power models.

5.1. Statistical results of context factor combinations

We first demonstrate the complexity of battery consumption patterns by considering how pairs of context factors influence consumption. Similarly to the previous section, we derive a ranking for the different pairs by considering the conditional mutual information between each pair, and rank the pairs in descending order of the CMI values. The results of these estimations are listed in Table 4.

Compared to the results of individual context factors' impact (see Table 2), the combination of multiple factors gives more accurate explanations of the battery consumption. A prominent example is CPU use, for which we can observe significantly higher impact when combined with another factor than when considered alone. Also factors related to network connection, such as Wi-Fi signal strength and network type, differ clearly from the MI analysis. Both have lower MI values in Table 2, but are more prominent when considered in conjunction with another context factor. Wi-Fi link speed and Wi-Fi signal strength have a CMI value of 0.99, which is higher than they get separately (0.014 each). Capturing this kind of nuances in consumption is particularly beneficial when giving suggestions to the end user on how to improve battery life. As an example, we can observe that changing another system setting can help to improve battery life in cases where high CPU use is mandated, such as, when playing a game.

The top context factors according to energy consumption seem to be battery voltage, CPU use, battery temperature, and movement (distance traveled) of the device, or combinations thereof. The effects of these factors are mediated by other factors, which in turn can cause significant increase or decrease in consumption. Accordingly, providing an accurate view of the battery consumption of a device requires models that can capture both the effects of multiple context factors and the effects of their interdependencies.

5.2. Battery consumption of context factor combinations

To further illustrate the complexity of battery consumption patterns, we consider how selected context factor combinations affect battery consumption. We focus on combinations with high CMI values (see Table 4), and measured their expected battery life gain. As an example case, we consider the impact of CPU use and another factor on battery consumption. The results of this analysis are shown in Fig. 3. The y-axis shows the battery life gain in percentages when compared to the average expected battery life in our dataset. The different columns represent different values of context factors.

Table 4
Top of the conditional mutual information estimates for pairs of context factors for energy consumption rates.

Context factors		CMI
Battery voltage	CPU use	4.29
CPU use	Screen brightness	2.17
Battery temperature	CPU use	2.07
CPU use	Distance traveled	1.81
CPU use	Wi-Fi signal strength	1.69
Battery voltage	Distance traveled	1.53
Battery temperature	Distance traveled	1.28
Distance traveled	Screen brightness	1.26
CPU use	Wi-Fi link speed	1.12
Battery voltage	Screen brightness	1.08
Wi-Fi link speed	Wi-Fi signal strength	0.99
Mobile data status	Network type	0.95
Network type	Wi-Fi signal strength	0.85
CPU use	Mobile network type	0.80
Battery temperature	Screen brightness	0.79
Distance traveled	Wi-Fi signal strength	0.75
Network type	Wi-Fi link speed	0.64
Mobile data status	Wi-Fi signal strength	0.60
Battery temperature	Battery voltage	0.56
Distance traveled	Wi-Fi link speed	0.54
Battery voltage	Wi-Fi signal strength	0.53

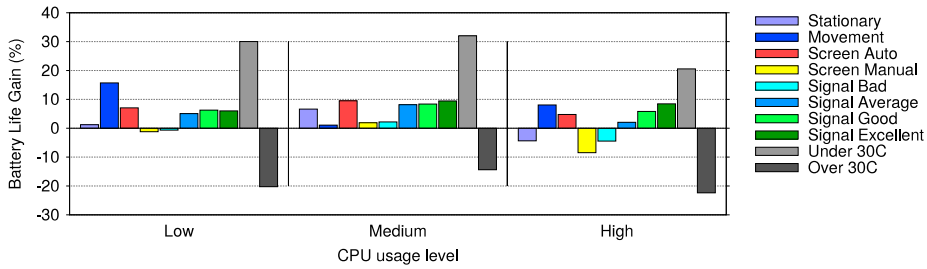


Fig. 3. Battery gain from CPU use combined with another context factor.

From the figure we can observe that the influence of context factor combinations on battery consumption is rather complex. For example, as long as the phone can observe average Wi-Fi signal strength, improving the connection will not provide significant savings unless CPU use is very high. In terms of screen brightness, as shown previously, the main effect results from switching to automatic brightness. However, this effect is most beneficial for moderate CPU use, and during high use, other factors can provide more pronounced changes. Another important factor is battery temperature, which can result in a loss of up to 50% battery life. The effects of temperature are consistent across all CPU use categories, indicating CPU use is not necessarily the (sole) cause for high battery temperature.

5.3. Power meter validation

The proposed approach of using crowdsourced measurements for constructing battery consumption models has been intended as a cost-effective way to capture fine-grained and nuanced differences in battery consumption. As we have demonstrated, these differences can have a significant impact on battery consumption and need to be accounted for to provide accurate estimates of the actual battery usage. We next compare our approach against empirical power models constructed using a hardware power monitor. We demonstrate that our approach is better at capturing the effects of changes in device state on battery consumption. In particular, our analysis indicates that empirical models are dominated by *instantaneous* effect, which tends to overestimate overall power consumption.

We consider measurements collected using a Samsung Galaxy S2 phone that was connected to a Monsoon Power Monitor.[3] The Power Monitor was set to output a constant voltage of 4.0 V. The phone battery was still inserted, with the + and - terminals blocked, letting the phone start up normally. Each experiment run lasted 10 min. We connected the phone to the cellular network and turned the phone screen on for the duration of each experiment run. Wi-Fi was enabled and Bluetooth disabled for all our experiment runs. Automatic updating of applications was disabled. All applications were

[3] https://www.msoon.com/LabEquipment/PowerMonitor/

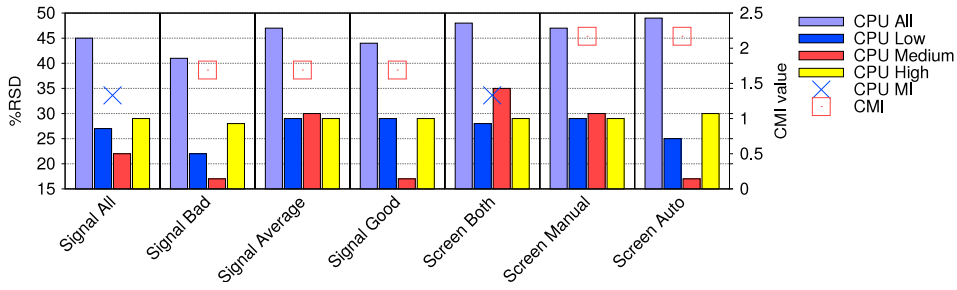


Fig. 4. Relative Standard Deviation (%RSD) of various CPU use and Wi-Fi signal strengths and screen brightnesses, along with their CMI values.

Table 5
Comparison of average battery lifetime improvement (in hours) between power monitor estimates and our approach.

CPU Use	All	Low	Medium	High
Experiment				
All Wi-Fi	2.60	4.77	2.23	1.90
Bad Wi-Fi	2.51	4.52	2.20	1.93
Average Wi-Fi	2.60	4.99	2.27	1.91
Good Wi-Fi	2.53	4.82	2.21	1.87
All screen br.	2.69	5.40	3.27	1.92
Manual screen br.	2.60	4.99	2.27	1.91
Auto screen br.	2.78	5.89	2.50	1.92
Crowdsourced data				
All Wi-Fi	5.24	5.53	5.54	5.53
Bad Wi-Fi	5.12	5.21	5.36	5.01
Average Wi-Fi	5.45	5.51	5.67	5.35
Good Wi-Fi	5.57	5.57	5.68	5.54
All screen br.	5.24	5.53	5.54	5.53
Manual screen br.	4.98	5.18	5.34	4.80
Auto screen br.	5.57	5.61	5.74	5.49

closed. Before each experiment run, we let the power consumption stabilize for several minutes to avoid the impact of background activity on the experiment. The experiment runs consisted of the following configurations:

- Full screen brightness, 30% CPU use, and bad (1); average (2); and good Wi-Fi signal strength (3).
- Full screen brightness, 60% CPU use, and bad (4); average (5), and good Wi-Fi signal strength (6).
- Full screen brightness, 100% CPU use (two tightly looping threads), and bad (7); average (8); and good Wi-Fi signal strength (9).
- Automatic screen brightness, average Wi-Fi signal strength, and 30% (10); 60% (11); 100% CPU use (12).

The impact of considering combinations instead of individual factors can be assessed by examining the relative standard deviations (i.e., ratio between standard deviation and mean) of the power monitor measurements. These are illustrated in Fig. 4. In the figure we consider separately the combined effect of CPU and Wi-Fi, and that of CPU and screen brightness. We also consider how decomposing these factors into categories influences the measurements.

The column groups in the figure correspond to the Wi-Fi signal strength range, and an average value for all of the values (signal all). Respectively, for screen brightness we consider automatic and manual, and average over all values (screen both). The columns in each group, from left to right, are CPU All (average over all use levels), Low, Medium, and High CPU use level. From the figure we can observe that the relative standard deviations for combined CPU use (i.e., columns with CPU All) are much higher than those of individual use levels by a factor of at least 1.5. The same observation applies for signal strength and screen brightness, indicating that considering individual factors is much less accurate at explaining battery consumption than the combination of multiple factors.

We next compare battery life estimates between power monitor measurements and our approach. The results of this analysis are shown in Table 5. From the results we can make two important observations. First, for low CPU use, the power monitor measurements are closely in line with the estimates provided by our approach. As CPU use increases, the estimates provided by power monitor measurements indicate significant decrease, whereas the estimates provided by our approach are much less affected. Current smartphones have several mechanisms to adjust CPU use and consistent high use is rare. Accordingly, while the power monitor estimates reflect actual battery consumption, they are over-zealous and overfit on the instantaneous consumption. In contrast, our approach averages over different usage contexts, providing a more realistic estimate of the practical impacts on battery consumption. Second, the results demonstrate that power monitor measurements are poor at identifying the relative importance of different context factors. For low CPU use, the difference

Table 6
Battery life in hours for selected combinations of four context factors.

Battery temperature	Distance traveled	CPU use	Screen brightness	Estimated battery life (h)
Under 30 °C	>0	Low	Automatic	8.83–9.12
Under 30 °C	>0	Low	Manual	8.49–8.82
Under 30 °C	>0	High	Automatic	8.09–8.24
Under 30 °C	>0	Medium	Automatic	7.65–7.89
Under 30 °C	>0	Medium	Manual	7.34–7.60
Under 30 °C	>0	High	Manual	7.27–7.41
Under 30 °C	None	Medium	Automatic	6.57–6.64
Under 30 °C	None	Low	Automatic	6.28–6.35
Under 30 °C	None	Medium	Manual	6.13–6.20
Under 30 °C	None	Low	Manual	5.88–5.96
Under 30 °C	None	High	Automatic	5.78–5.82
Over 30 °C	>0	Low	Automatic	5.08–5.22
Under 30 °C	None	High	Manual	5.00–5.04
Over 30 °C	>0	Low	Manual	4.73–4.88
Over 30 °C	>0	High	Automatic	4.62–4.69
Over 30 °C	>0	Medium	Automatic	4.59–4.70
Over 30 °C	>0	Medium	Manual	4.28–4.39
Over 30 °C	None	Medium	Automatic	4.25–4.29
Over 30 °C	>0	High	Manual	4.08–4.14
Over 30 °C	None	Medium	Manual	4.06–4.09
Over 30 °C	None	Low	Automatic	4.02–4.06
Over 30 °C	None	High	Automatic	3.91–3.94
Over 30 °C	None	Low	Manual	3.74–3.78
Over 30 °C	None	High	Manual	3.45–3.46

between manual and automatic screen brightness is clearly observable. However, as CPU use increases, the power monitor models overfit on the high power consumption of CPU, providing limited information about the importance of other context factors. In contrast, the impacts of changing screen brightness to automatic and the impact of improving Wi-Fi signal level remain observable across all CPU use levels for our approach. In Fig. 4, we can observe that these differences are even more observable from the conditional mutual information.

5.4. Highlights and example cases

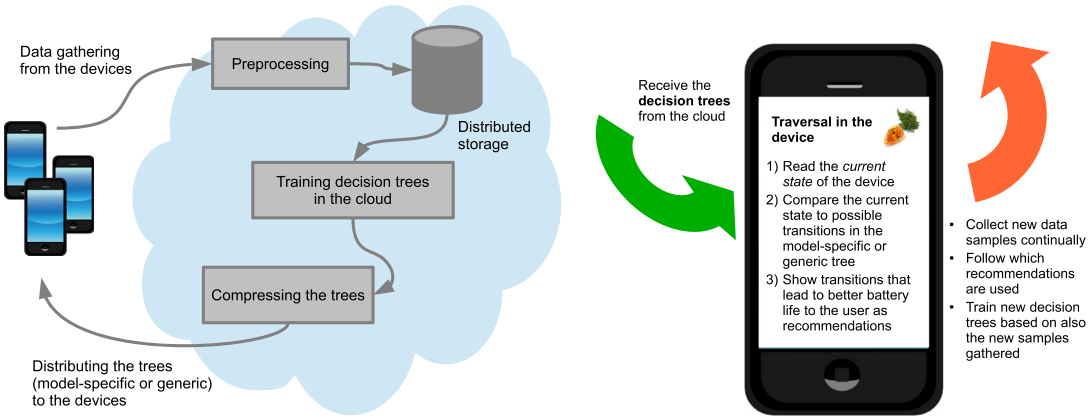
As the final step of analysis, we demonstrate how our approach can also be used to obtain new insights into battery consumption. Examples of selected context factors' impact on battery consumption are listed in Table 6. We have selected CPU use and temperature from subsystem variables, and distance (motion or stationary) and screen brightness from system settings. In all examples, connection type has been a cellular data connection. Table 6 presents the estimated time to drain the battery from 100% to 0%, while actively using a smartphone with the given context factor and value combination. With different values of CPU use, battery temperature, movement, and screen brightness, the battery life can range from 3.45 h up to 9.12 h.

The table is sorted by the time to drain the battery, descending. We can see that the main deciding factor for battery life is the temperature of the battery. With a lower temperature, we get a higher battery life. After that, traveling instead of staying still seems to increase battery life. This may be due to users driving and not using their mobile phones while mobile. After these factors, the CPU is the most dominant, and changing screen brightness brings the smallest, but still significant, battery life differences. These results show that while CPU use alone is a good indicator of energy consumption, significant gains can be obtained by considering more complex combinations. In addition to this, battery temperature and distance traveled can be used together to predict battery life.

Complex combinations of factors, such as those listed in Table 6, can be used to decide which factors to change to improve battery life, while keeping others constant. For example, while moving and playing a game, the CPU is often high. If the phone can be kept relatively cool, 78% more battery life can be expected compared to warmer battery (increase from 4.08h to 7.27h). Further savings can be obtained by switching screen brightness to automatic.

With respect to the worst possible configuration, moving to a cooler place (45% battery life gain) and changing screen brightness without changes in CPU use can prolong battery life by 68% (from 3.45h to 5.78h). Our results in Table 6 and in Fig. 3 show that the battery temperature is not always directly related to CPU use. High battery temperature can be caused, e.g., by the ambient temperature in warmer countries, battery misbehavior or a battery bug, or because the smartphone is exposed to direct sunlight, e.g., due to being forgotten under the windshield inside a car on a sunny day. Battery temperature alone can shorten the battery lifetime as much as 50%. If cooling the device is not possible, e.g., because of the ambient climate re-configuring other factors can help to improve the battery lifetime.

With low CPU use and a cool battery, no movement, and manual screen brightness, we can obtain an active battery life of 6 h, which improves to almost 9 h by only changing movement. That behavior can be caused by the users mostly walking



(a) The system overview. (b) The client side.

Fig. 5. Overview of the Constella context factors recommendation system.

or driving a car and not using their smartphones while moving from place to another. It is also possible, that energy saving policies activate as movement requires re-connections to the cellular base stations. As Table 4 shows in Section 5.1, distance traveled is ranked high together with CPU use. It is possible that the most CPU heavy actions, such as gaming, are only done in longer periods while stationary.

6. Constella: Recommender system for context factors

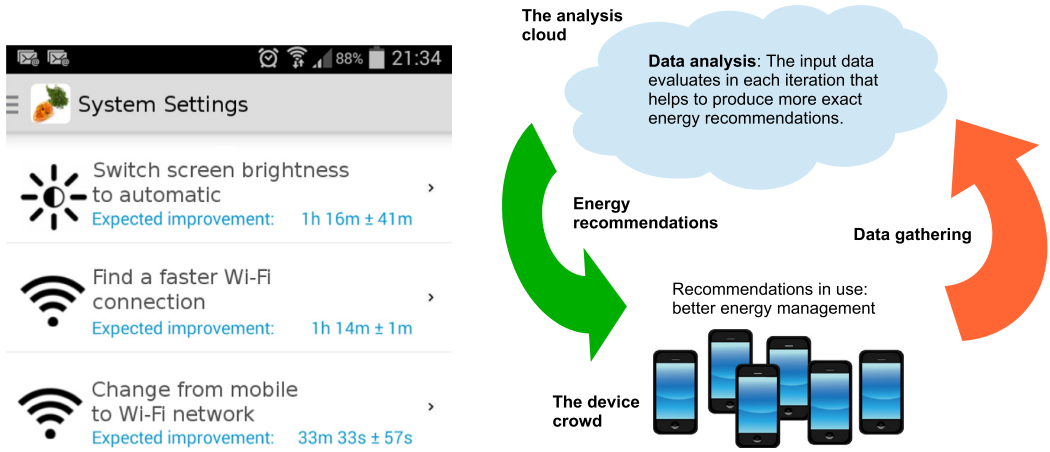
Mobile users are often forced to actively seek countermeasures to prolong the lifetime between successive recharges [25,28]. Examples of these countermeasures include killing battery hungry applications or tasks, and manipulating context factors either through switching off specific sensors or adjusting individual system settings. Previous research has predominantly focused on the former task [8,29–31]. However, in many situations the most battery hungry application is the one that the user is currently using, making the manipulation of context factors the only feasible way to save battery. As we have shown, the overall impact of context factors on battery is significant, but difficult to optimize as there are complex interdependencies between the different factors. As our final contribution, we develop Constella, a novel recommender system for context factors. In the following we describe Constella in detail and use battery monitor measurements to demonstrate that Constella provides context factor recommendations that help to significantly save battery.

6.1. Constella system overview

Constella is a recommender system for context factors that has been developed on top of Carat [8]; see Fig. 5(a) for a general overview of the Constella recommender system. The Carat application runs on mobile clients and collects samples whenever the battery level of the device changes, e.g., when the on-screen indicator changes from 99% to 98%. It does this by registering to the *BATTERY_CHANGED* Android Intent. The sample contains the battery level, timestamp, the running applications list, and the state of the context factors as described in Section 3. The data is stored locally until the user opens the Carat application. It is then sent over to the cloud, where we preprocess the data, construct a recommendation model, and perform other types of analyses on the data. Prior to Constella, Carat already supports finding anomalies in the energy consumption of applications, and recommends users to restart or kill applications that behave anomalously on their devices. The work in this paper extends the analytics functionalities of Carat to provide recommendations on how to optimize context factors. The anomaly detection and other analyses rely on the Spark [35] cluster computing system which allows analysis of large datasets in a high-performance distributed computing framework.

In Constella, we first preprocess the samples by filtering out samples obtained during charging or when the phone was switched off, and by removing samples from iOS devices. We also filter out samples that do not contain the full set of context factors that are used by Constella. The preprocessed samples are saved to distributed storage where they can be accessed by the computing nodes of the analysis system. We then learn a recommendation model, which in our case corresponds to a set of decision trees (see the next section), one for all devices, and one for the device model of the user, which are compressed and stored on the cloud side. Carat clients retrieve a copy of either the model-specific tree, if one is available, or the generic tree the next time they connect to our servers. Recommendations are generated on the client-side as they depend on the current state of the device. User interactions with the presented recommendations are logged and sent to the analytics backend where they can be used to further refine the recommendation models.

Fig. 5(b) shows how Constella looks from the perspective of the client. The device receives the decision tree from the cloud and reads its current system state from the Android API, analogous to the way Carat samples are taken. The client



(a) Example context factor recommendations.

(b) The crowdsourcing in use and the data flow.

Fig. 6. The details of the Constella.

then compares the current system state against the tree, traversing the nodes in the tree while building a list of possible transitions that lead to better expected battery lifetime. This list is shown to the user as a list of actionable context factor recommendations. Examples of these recommendations are shown in Fig. 6(a). When a user clicks the recommendation, they are taken to the relevant settings page to adjust the setting in question.

The system architecture of Constella follows an iterative data-analysis model, illustrated in Fig. 6(b). Client devices running Carat provide new samples for Constella, which uses them to update the recommendation models. By continually refining the learned models, the recommendations provided by Constella adapt to changes in the smartphone operating environment. Examples of these changes include the emergence of novel usage patterns, internal changes in smartphone operations, e.g., due to operating system updates, and emergence of new devices on the market. Analogously, user interactions with the recommendations are interpreted as feedback which is used to optimize the suitability of recommendations shown to the user and to learn about the usage contexts of recommendations.

6.2. Recommendation model

Constella relies on a decision tree based recommendation model for capturing the energy impact of different context factors. Decision trees have been shown to provide a user-friendly and understandable representation for complex relationships [36] which is essential for improving users' trust in the recommendations. The tree model also provides a compact and compressible representation of relevant information which can be efficiently stored and used on a smartphone without a considerable impact on battery life. The decision tree organizes context factor combinations into a logical structure and turns them into human-readable and actionable recommendations. The tree model can be learned efficiently using a Spark cluster, and shared to each client device. The clients can then generate recommendations independently of the cluster by following paths of potential system state changes within the tree model. This makes it possible to generate energy recommendations even in offline mode whenever the usage context changes. Trees are easy to visualize, so that users can plan their own energy profiles, for example, by selecting branches in the tree.

Since smartphones have wildly different hardware and settings, with different energy consumptions, Constella generates model-specific decision trees for smartphone models for which we have enough data, along with the generic tree for all of the data. This way users will benefit from more accurate recommendations for their device, while we can avoid the so-called cold start (or bootstrap) problem for devices with little or no data by using the generic tree.

We consider k-ary decision trees, where a node is labeled by the name of a specific context factor. Edges to child nodes are labeled by the value categories of that context factor. Therefore each node can have as many children as there are categories for the corresponding context factor. For example, a node labeled *screenBrightness* would have two children in our models, *manual* and *automatic*, whereas the node *mobileNetworkType* could have up to 15 children, corresponding to all the different mobile network types from *GPRS* to *LTE*.

To construct the tree, we need to choose the order in which different context factors appear in the tree. In particular, as our goal is to generate recommendations on how to improve battery, the ordering of the factors should be such that the factors with highest overall influence are at the top of the tree. To accomplish this, we first categorize data samples into three classes (using an equal samples procedure) based on their energy consumption. These classes are: (i) *low*: over 24 h expected battery lifetime (and no need to charge the battery during a weekday); (ii) *medium*: expected lifetime 8–24 h; and (iii) *high*: expected lifetime less than 8 h. For each context factor, we then calculate an *impurity metric* by considering how well the factor splits

the device's energy state into the different classes. As the impurity metric we currently use entropy. Minimizing entropy when choosing the next context factor for splitting the data leads to a decision tree where information gain is maximized. As the entropies are calculated according to different energy profiles, this implies that the most important context factors, in terms of energy impact, appear at the root of the tree and the importance (and energy impact) of the factors decreases as we go deeper in the tree. Note that we consider the energy classes only for determining how to generate the tree, and the leaf nodes of the tree, which are responsible for generating the recommendations, consider the expected improvement in battery lifetime in hours.

We build on a novel and efficient distributed decision tree algorithm for Spark which takes advantage of the hierarchical structure of the tree. After a split is generated, child nodes are independent and can be computed in parallel. As the depth of the tree increases, computation becomes exponentially more parallel. This enables our method to handle even large amounts of crowdsourced data very efficiently. Different context factor combinations are represented as a path in the tree which consist of splits performed until the current node. An example is the path *"Network type = Mobile, Screen brightness = Automatic"* shown on the left-hand side of Fig. 7.

We use paths in the tree as the keys for the (key, value) model of the data in Spark's grouping functions *groupBy* and *groupByKey*. They collect the path-related data points to one computation node in the cluster, which then computes that portion of the work, for example, impurity measurements. Therefore, instead of training the tree as a recursive algorithm, we can better benefit from data distribution and parallel computation.

Our distributed algorithm trains the decision tree as follows:

- Group all data items by every single context factor as a key. Save the result to a working structure, Spark's Resilient Distributed Dataset (RDD). Then iterate in the loop:

1. **Generate** all possible paths (or, data splits) for all nodes of the current level. Level means the leaves of the tree, e.g., the longest paths found until then.
2. **Evaluate goodness values** of all possible splits for all the nodes of the current level. Evaluation happens by comparing splits to the energy usage classes using an impurity metric, for example, entropy. The best split minimizes the entropy after the split, and increases the information gain in the tree.
3. **Choose best splits** for all current nodes, and generate new paths for their children (new leaves, and paths one step longer). Then group the data by the new paths, and save new paths to the working structure for next iteration.

- End the loop when there are no longer enough data items left to perform a new split for any of the leaf nodes, or all of the values of all the system settings are already part of all of the paths in the tree. Save the tree to the output file for latter compression and distribution to the devices.

The output of the algorithm is a list of paths from the root node to inner nodes and leaves, which can also be represented as a tree structure, where each node knows its children and parent. The tree format can be easily traversed on the client side. First, the client reads the current status of the device, for example, by looking at the latest Carat sample and its system state description. Then, it will traverse the tree starting from the root, finding the best matching path from the tree using the context factors that the system state contains. This will be used as the baseline for the energy consumption. Next, each possible step in the tree (to a child node, sibling, or parent) in the tree is considered. If the context change results in lower energy consumption – or, higher expected battery lifetime – this item will be saved as a possible recommendation. After collecting all possible energy saving changes in the tree, the list is sorted by energy benefit, and the best items are presented as recommendations to the user. We can combine multiple steps into a single recommendation. For example, switching from Wi-Fi to a mobile network connection might not be beneficial in the current context, but if the mobile network type is changed from GPRS to HSPA the energy benefit may be significant. So, we can recommend a combination action such as "Switch to the mobile network and find a good 3G signal to gain 1h 10 m ± 5 m battery life".

Fig. 7 shows an example of the decision tree. There are splits for three context factors: at first, for network type, which splits the data to three parts, and for each part of the data, we can calculate an expected value of battery lifetime (EV). In the second level, there is a split by screen brightness after the network type "mobile", and a split by distance traveled after network type "Wi-Fi". If the device currently remains connected to the Internet via the mobile network with manual screen brightness, we can find at least two one-step changes to consider: if EV4 is better than EV5 (the current expected value), we can suggest to switch to automatic screen brightness instead of the manual setting. If EV2 is also better than EV5, we can also show the recommendation of changing the network type. With more than one step, we can also go deeper in the tree, depending on the size and depth of the tree.

6.3. Experimental setup and test cases

We evaluate Constella through power monitor measurements carried out on carefully selected test case applications. To carry out our evaluation, we first constructed a decision tree using a dataset with 4,798,715 samples. This data is slightly larger than our published dataset, containing measurements from three additional months. We only considered samples which met the following criteria: (i) the samples are from Android devices; (ii) for each context factor considered in the tree, each sample has at least a default value; (iii) the samples have not been taken during charging (recognized by AC or USB charger being plugged in); (iv) expected battery lifetime (EV) between two samples is higher than (or at least) zero

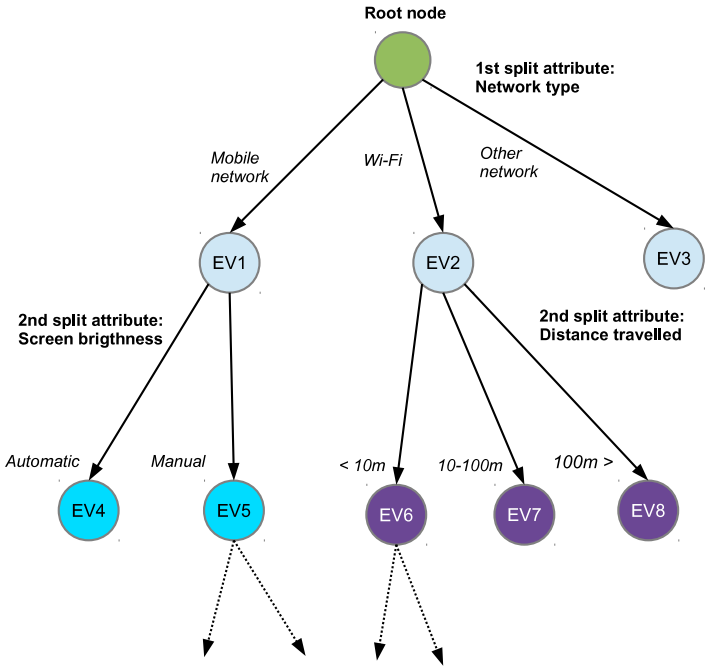


Fig. 7. Example of the decision tree used for energy recommendations. EV = Expected value of battery lifetime in a given node.

(charging samples without recognized charger); and (v) battery consumption in percentages per seconds is less than 0.05%/s, i.e., fully draining the battery must take at least an hour. Once the tree was constructed, we generated recommendations for three test applications and used a Monsoon power monitor to measure how the recommendations impact battery drain. The test cases were selected to cover popular application categories [1] where the battery drain is typically high. Our test cases are summarized as follows:

Playing a game

During gameplay CPU load is typically high and the screen settings are fixed by the application, e.g., high brightness and full screen mode are required. We choose the game Angry Birds Space (free version), which is on the top 200 most energy-intensive applications on the Carat dataset.[4] It has 1565 users in the Carat data, who can on average can save 241 min when not using this application. This game includes a space physics engine, that, for example, demonstrates gravity of different objects, and thus poses high requirements to CPU usage. The free version of the game also shows advertisements during the game. Our measurements were taken while playing different rounds (in easy level) of the game, and watching any shown advertisements between the rounds. GPS was disabled during the measurements.

Using a map

Navigation and browsing locations using Google Maps or another similar application requires constant GPS usage and hence tends to induce high battery drain. In the Carat data, Google Maps is within the 200 most energy-intensive applications. It is also highly popular, having 255,548 users. The average lifetime improvement from not using this application is 107 min in the Carat data. For carrying out our measurements, we consider a case where the user is (i) identifying her current location; (ii) searching for another location; and (iii) searching for an optimal route between them.

Streaming a video

Streaming videos requires constant network connection and heavy data transmissions, which are a major source of battery drain. We used the Youtube application to capture typical behavior for this type of applications. Our measurements were carried out when the application were buffering a video. All chosen videos had high resolution, bright colors, and music or other sounds within. To control for energy drain of advertisements, GPS was disabled during the measurements. While Youtube is not among the 200 most energy-intensive applications, it is often pre-installed to the devices and has a wide user population.

4 http://carat.cs.helsinki.fi/statistics/

Table 7
Measurements from three test cases: gaming (Angry Birds Space), finding an optimal route between map locations using Google Maps, and watching videos on Youtube. Values are hours of expected battery lifetime.

Settings	Game		Maps		Youtube	
	S3	S4	S3	S4	S3	S4
Wi-Fi good, screen full	3.92	4.44	3.82	3.63	4.40	4.92
Wi-Fi good, screen low	4.74	4.70	4.16	4.50	5.04	5.80
Wi-Fi good, screen auto	4.15	4.49	4.30	4.20	5.30	5.15
Wi-Fi bad, screen full	4.08	3.52	3.36	3.33	5.04	5.87
Wi-Fi bad, screen low	4.40	5.19	4.64	6.18	6.41	7.13
Wi-Fi worse, screen auto	4.50	4.46	4.33	5.04	6.89	5.91
Network off, screen full	5.22	3.80				
Network off, screen low	5.77	4.45				
Network off, screen auto	5.60	3.80				

For each test case, we performed power monitor measurements using the following experimental scenarios: (i) Wi-Fi signal strength was good or average (from three to four bars); (ii) Wi-Fi signal strength was worse than average (one bar in case of playing the game, two bars for Maps and Youtube since these applications failed to load anything from the Internet with worse Wi-Fi); (iii) or networking was switched off; (iv) screen brightness was set to be full; (v) low; (vi) or automatic. For the case of playing a game, all six scenarios were considered. For the Google Maps and Youtube applications we consider all scenarios except where networking is switched off since these applications require an active Internet connection. The tests were carried out on Samsung Galaxy S3 (model number GT-I9300, with battery size 2100 mAh) and S4 (model number GT-I9505, with a battery size 2800 mAh) devices. Both devices had an activated SIM card inside the device. Ground truth data about battery consumption was obtained using a Monsoon power monitor.

6.4. Results

The power monitor measurements from our experiments are summarized in Table 7. In line with the results described earlier in the paper, screen brightness has a clear impact on energy consumption, with both automatic and low settings reducing battery drain. The impact of the quality of network connection is more complex. In most cases better quality connection leads to improved battery consumption, but in the short term there are some exceptions. In particular, in video streaming worse signal quality seems to improve battery at the cost of user experience as the application is mainly stuck at buffering videos instead of showing them. To better understand the overall impacts of the different scenarios and test cases, we next look at the results from the perspective of the decision tree.

Fig. 8 presents the first four layers of nodes from a background decision tree for Android devices. The overall depth of the tree is 8, but we have chosen to show only the first 4 levels for improved visual clarity. The first split has been made by battery temperature (more or less than the median temperature, 30 °C), which had also a clear impact on energy consumption as shown in Fig. 3. The next split is Wi-Fi status (enabled or not) for both of the branches, and then screen brightness (automatic or manual setting). After that, the splits start to diverge. There are different networking features, such as Wi-Fi signal strength, and the battery health. We can see from the tree in Fig. 8 that even a combination of the three most influential context factors is not enough to narrow down the impact on the battery life of the device. After traversing the edges *battery Temperature* <30 °C, *wifiStatus = enabled*, and *screenBrightness = manual*, the EV of 57 h can still swing down to 30 h if the mobile network connection is worse than 3G (GPRS or EDGE). Note that Wi-Fi status refers to status of the Wi-Fi setting (on or off), and does not guarantee that Wi-Fi is used for Internet connectivity. In our data, network type describes which connection type is used to connect to the Internet.

We also investigated some model-specific decision trees, namely those of the Samsung Galaxy S3 and S4. The tree for S3 has an identical structure to the generic tree shown in Fig. 8, with differences at the 4th level. The S4 tree is more interesting. It has the same first split, battery temperature, but the second one for battery temperature less than 30 °C is network type. S4 users with these battery temperatures were connected to the Internet 94.5% of the time. Of that, 58.3% was via Wi-Fi and 36.2% via a cellular network connection. The battery lifetimes were 101 h for Wi-Fi, 71 h for cellular, and 90 h for other states, such as while disconnecting, connecting, or doing a connection handover. This shows also that the battery lifetimes for individual models can be longer than in the background model, which is an important consideration when detecting anomalies in a smartphone's energy consumption.

We next consider how Constella can reduce battery consumption through context factor recommendations. Based on the decision tree given in Fig. 8, recommendations for our three cases would be:

Playing a game

The high CPU usage of Angry Birds Space is likely to cause some heating for the battery. Without asking the user to end the gaming, it is not possible to avoid this effect. Hence, we traverse in the tree along the path corresponding to higher battery temperature. The next split is for Wi-Fi status, which is followed by a recommendation to keep screen brightness

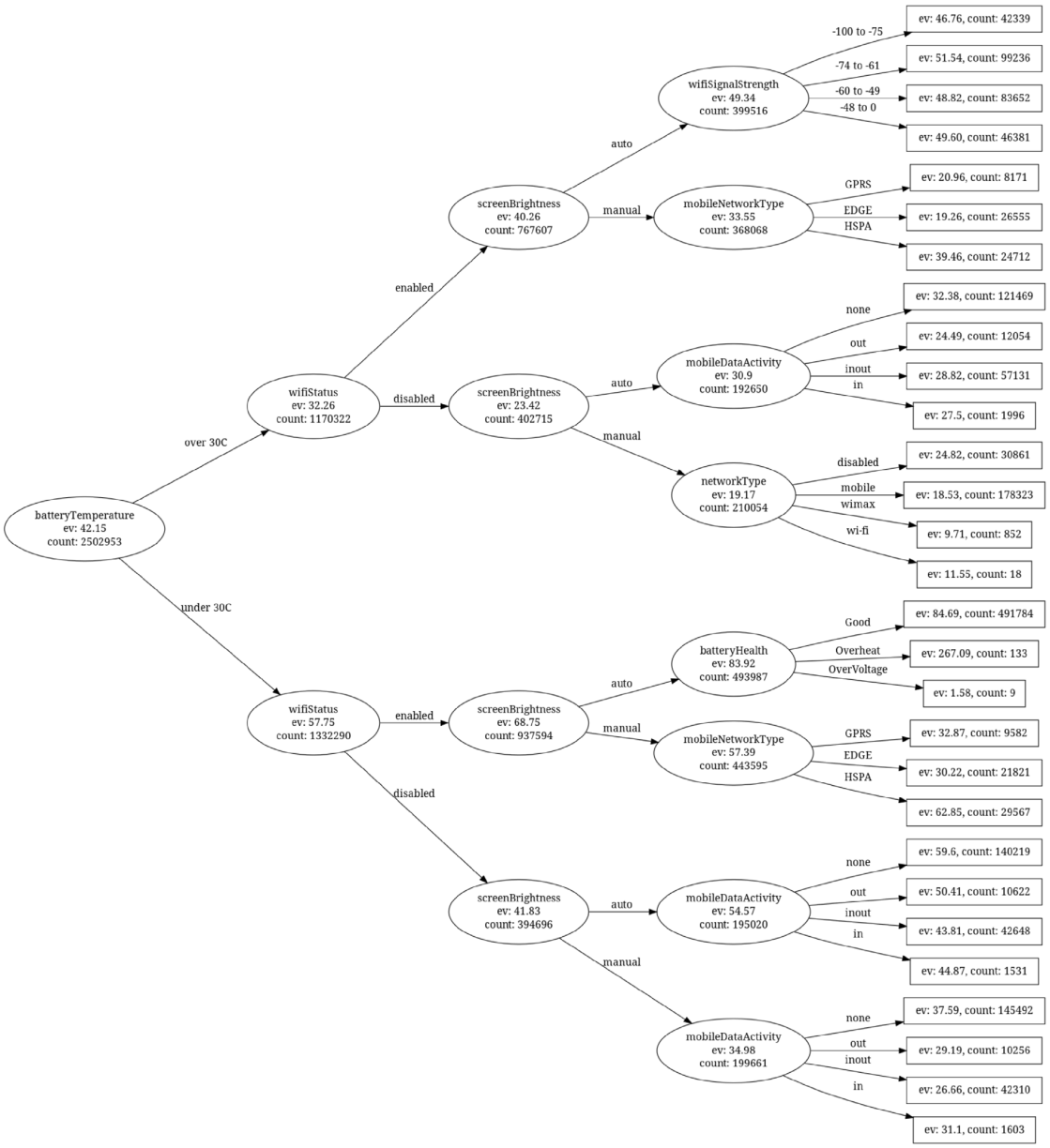


Fig. 8. Decision tree of depth four for the Carat dataset. The tree has been truncated from the full depth of 8.

automatic instead of manual setting. Hence, we can recommend switching Wi-Fi off and adjusting the display brightness. Further recommendations can be found by traversing deeper within the tree. For example, following the uppermost path of the tree (higher temperature, Wi-Fi enabled, and automatic screen brightness), we can make a comparison between the different Wi-Fi signal strength values and give a recommendation to look for a location with better Wi-Fi connection. By correlating these recommendations with the power monitor measurements in Table 7 we can observe that, on both device models, significant improvements in battery life are achieved when screen brightness is reduced. Switching off Wi-Fi similarly reduces battery drain.

Recommendations to give for game playing:

- Avoid long gaming sessions because of heavy CPU usage, or replace the app with another game.
- Let the device cool time by time.

- Set screen brightness to be automatic, or use lower value for manual setting.
- Choose a place for a good Wi-Fi, if the game requires network connectivity.

Using Maps with GPS

In the case of Google Maps, the analysis is more complicated as GPS and Wi-Fi are required for operating the application. Similarly to the mobile game scenario, the best split is obtained for battery temperature, and the next split is whether the screen brightness is manual or automatic. Accordingly, a recommendation to adjust the screen brightness would be given. In line with the game test case recorded by power monitor, the provided recommendations thus clearly help to improve battery life. Networking is required for the Google Maps search, and that motivates to find the best network connectivity available.

Recommendations to give for map search:

- If using a Wi-Finetwork, try to find an area of good signal strength (three out of four bars).
- If using a mobile network, prefer HSPA technology over GPRS or EDGE.

Streaming a video using Youtube

In the case of watching video, the main culprit is the data connection. Optimizing for energy, however, is rather complicated in this case as lowering energy naïvely can decrease usability. For example, the power monitor measurements suggest that decreasing Wi-Fi link quality reduces energy load, but this is due to reduced video quality and increased buffering as packets are lost. Accordingly, the recommendations for this case would correspond (again) to optimizing the display or changing to a more energy-efficient network interface. Allowing continuous Wi-Fi connectivity is acceptable regarding energy consumption, even if somebody might consider switching off the connection after buffering a video. The status Wi-Fi enabled has a significantly better energy estimated than in case of the disabled Wi-Fi, possible because of reconnection time by time.

Recommendations to give for video streaming:

- Choose a stable and continuous network, whether a good Wi-Fi connection or if not available, a mobile network with HSPA technology.
- Dim the screen brightness if possible, or use automatic setting.

7. Discussion and summary

The present paper has provided four contributions. Our first contribution has been the development of a novel approach for constructing energy models using crowdsourced battery discharge measurements. Contrary to previous works on energy modeling, our approach is not restricted to capture the effects of individual sensors, features or system settings, but can capture complex interdependencies between all of these. As we have experimentally demonstrated, estimates provided by our approach are in line with power meter measurements, providing an accurate view of the energy state of the device. The second contribution is a large-scale analysis of the influence of different system settings on battery consumption. The analysis validated the use of crowdsourced measurements for modeling energy, and confirmed findings in previous studies. It also provided novel insights about battery consumption and quantified their effects. For example, we demonstrated that a Wi-Fi signal strength drop of one bar can result in a battery life loss of over 13% and that a smartphone sitting in the sun can experience over 50% worse battery life than one indoors in cool conditions. As our third contribution, we have presented the Constella recommender for context factors. Constella offers a way to present complex interdependencies of the different context factor combinations and gives user-understandable and actionable energy recommendations from system settings and subsystem variables. We have empirically demonstrated that these recommendations can significantly reduce battery drain, achieving even up to 61% improvements in battery lifetime. Finally, as the fourth contribution of the paper, we have made available for research purposes the large-scale (anonymized) dataset used in our analysis.[5]

Energy models that can accurately capture the energy state of a device and that can estimate how system state changes influence energy, are beneficial for several reasons. Our approach can be used to bootstrap and support battery management interfaces developed to support end users. Instead of merely allowing users to switch off (or on) different settings, our approach can estimate how these changes are expected to influence device lifetime. Our approach can be used to construct device-specific resource optimization strategies that can estimate changes in battery use more accurately. Our approach could be used to construct empirical energy models for comparing and evaluating energy-effectiveness of different sensing strategies.

The results presented in this paper are also potentially beneficial for understanding long-term effects of sensor and battery management strategies on battery life. The comparison of battery life estimates between our approach and power monitor measurements showed that our approach can average effects over different usage contexts, whereas empirical power models tend to focus on instantaneous effects. As the overall state of a smartphone is complex, and in constant flux, instantaneous estimates tend to result in overestimates of battery consumption. Assessing the benefits of using crowdsourced battery models for these purposes is another interesting venue for future investigations.

[5] http://carat.cs.helsinki.fi/research

Acknowledgments

The work of Ella Peltonen has been supported by Doctoral School of Computer Science (DoCS). This research was partially supported by the Academy of Finland grant 277498. The publication only reflects the authors' views. The authors are grateful to Dr Stephan Sigg, Teemu Pulkkinen, and Samuli Hemminki for comments on earlier versions of the paper.

References

[1] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, G. Bauer, Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage, in: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, 2011.
[2] C.H.K. van Berkel, Multi-core for mobile phones, in: Proceedings of the Conference on Design Automation and Test in Europe (DATE '09), 2009, pp. 1260–1265.
[3] D. Linden, T.B. Reddy, Handbook of Batteries, third ed., McGrew-Hill Professional, 2001.
[4] K.N. Truong, J.A. Kientz, T. Sohn, A. Rosenzweig, A. Fonville, T. Smith, The design and evaluation of a task-centered battery interface, in: Proceedings of the12th International Conference on Ubiquitous Computing, 2010.
[5] D. Ferreira, A.K. Dey, V. Kostakos, Understanding human-smartphone concerns: A study of battery life, in: Proceedings of the 9th International Conference on Pervasive Computing, 2011.
[6] N. Vallina-Rodriguez, J. Crowcroft, Energy management techniques in modern mobile handsets, IEEE Commun. Surv. Tutor. 15 (2013) 179–198.
[7] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, Sasu Tarkoma, Energy modeling of system settings: A crowdsourced approach, in: Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on, PerCom'15, IEEE, 2015, pp. 37–45.
[8] A.J. Oliner, A.P. Iyer, I. Stoica, E. Lagerspetz, S. Tarkoma, Carat: Collaborative energy diagnosis for mobile devices, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, New York, NY, USA, 2013, pp. 10:1–10:14. http://dx.doi.org/10.1145/2517351.2517354.
[9] A. Rice, S. Hay, Decomposing power measurements for mobile devices, in: Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2010, pp. 70–78.
[10] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin, Diversity in smartphone usage, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010), ACM, 2010, pp. 179–194.
[11] A. Schulman, T. Schmid, P. Dutta, N. Spring, Demo: Phone Power Monitoring with BattOr, ACM Mobicom 2011. Available at http://www.stanford.edu/~aschulm/battor.html (2011).
[12] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R.P. Dick, Z.M. Mao, L. Yang, Accurate online power estimation and automatic battery behavior based power model generation for smartphones, in: Proceedings of the 8th IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, ACM, New York, NY, USA, 2010, pp. 105–114. http://dx.doi.org/10.1145/1878961.1878982.
[13] F. Xu, Y. Liu, Q. Li, Y. Zhang, V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics, in: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, Berkeley, CA, USA, 2013, pp. 43–56.
[14] I. Koenig, A.Q. Memon, K. David, Energy consumption of the sensors of smartphones, in: Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013), 2013, pp. 1–5.
[15] M.B. Kjærgaard, S. Bhattacharya, H. Blunck, P. Nurmi, Energy-efficient trajectory tracking for mobile devices, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys 2011), ACM, New York, NY, USA, 2011, pp. 307–320. http://dx.doi.org/10.1145/1999995.2000025.
[16] M.B. Kjærgaard, H. Blunck, Unsupervised power profiling for mobile devices, in: Proceedings of the 8th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous), 2011.
[17] Y. Wen, R. Wolski, C. Krintz, Online prediction of battery lifetime for embedded and mobile devices, in: Proceedings of the 3rd International Workshop on Power-Aware Computer Systems (PACS), Springer, 2005, pp. 131–138.
[18] J.-M. Kang, C.-K. Park, S.-S. Seo, M.-J. Choi, J. Hong, User-centric prediction for battery lifetime of mobile devices, in: Proceedings of the 11th Asia-Pacific Network Operations and Management Symposium (APNOMS), Springer, 2008, pp. 531–534.
[19] Y. Liu, C. Xu, S.C. Cheung, Where has my battery gone? Finding sensor related energy black holes in smartphone applications, in: 2013 IEEE International Conference on Pervasive Computing and Communications, PerCom 2013, San Diego, CA, USA, March 18–22, 2013, 2013, pp. 2–10. http://dx.doi.org/10.1109/PerCom.2013.6526708.
[20] A. Pathak, Y.C. Hu, M. Zhang, Where is the energy spent inside my app? Fine grained energy accounting on smartphones with Eprof, in: Proceedings of the 7th ACM European Conference on Computer Systems, ACM, New York, NY, USA, 2012, pp. 29–42. http://dx.doi.org/10.1145/2168836.2168841.
[21] X. Zhao, Y. Guo, Q. Feng, X. Chen, A system context-aware approach for battery lifetime prediction in smart phones, in: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC), ACM, 2011, pp. 641–646. http://dx.doi.org/10.1145/1982185.1982327.
[22] N. Ravi, J. Scott, L. Han, L. Iftode, Context-aware battery management for mobile phones, in: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2008, pp. 224–233.
[23] S. Agarwal, R. Mahajan, A. Zheng, V. Bahl, Diagnosing mobile applications in the wild, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, ACM, NY, USA, 2010, pp. 22:1–22:6. http://dx.doi.org/10.1145/1868447.1868469.
[24] D.T. Wagner, A. Rice, A.R. Beresford, Device Analyzer: Large-scale mobile data collection, ACM SIGMETRICS Perform. Eval. Rev. 41 (4) (2014) 53–56.
[25] N. Banerjee, A. Rahmati, M.D. Corner, S. Rollins, L. Zhong, Users and batteries: Interactions and adaptive energy management in mobile systems, in: J. Krumm, G.D. Abowd, A. Seneviratne, T. Strang (Eds.), UbiComp 2007: Ubiquitous Computing, in: Lecture Notes in Computer Science, vol. 4717, Springer Berlin, Heidelberg, 2007, pp. 217–234. http://dx.doi.org/10.1007/978-3-540-74853-3_13.
[26] A. Rahmati, A. Qian, L. Zhong, Understanding human-battery interaction on mobile phones, in: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services, 2007.
[27] A. Rahmati, L. Zhong, Human battery interaction on mobile phones, Pervasive Mob. Comput. 5 (2009) 465–477.
[28] D. Ferreira, E. Ferreira, J. Goncalves, V. Kostakos, A.K. Dey, Revisiting human-battery interaction with an interactive battery interface, in: Proceedings of Ubicomp 2013, ACM, 2013.
[29] A. Pathak, A. Jindal, Y.C. Hu, S.P. Midkiff, What is keeping my phone awake?: Characterizing and detecting no-sleep energy bugs in smartphone apps, in: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys'12, ACM, New York, NY, USA, 2012, pp. 267–280. http://dx.doi.org/10.1145/2307636.2307661.
[30] A. Banerjee, L.K. Chong, S. Chattopadhyay, A. Roychoudhury, Detecting energy bugs and hotspots in mobile apps, in: Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2014, ACM, New York, NY, USA, 2014, pp. 588–598.
[31] X. Ma, P. Huang, X. Jin, P. Wang, S. Park, D. Shen, Y. Zhou, L.K. Saul, G.M. Voelker, eDoctor: automatically diagnosing abnormal battery drain issues on smartphones, in: Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, USENIX Association, Berkeley, CA, USA, 2013, pp. 57–70. http://dl.acm.org/citation.cfm?id=2482626.2482634.
[32] A. Shye, B. Scholbrock, G. Memik, Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures, in: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, ACM, New York, NY, USA, 2009, pp. 168–178. http://dx.doi.org/10.1145/1669112.1669135.
[33] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37. http://dx.doi.org/10.1109/MC.2009.263.
[34] Fleuret François, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. 5 (2004) 1531–1555.

[35] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M.J. Franklin, S. Shenker, I. Stoica, Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, in: Proceedings of NSDI '12: 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI '12, USENIX Association, 2012, pp. 15–28.
[36] B.Y. Lim, A.K. Dey, D. Avrahami, Why and why not explanations improve the intelligibility of context-aware intelligent systems, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, ACM, New York, NY, USA, 2009, pp. 2119–2128. http://dx.doi.org/10.1145/1518701.1519023.

Research Theme B: Mobile Application Usage

Research Paper III

Stephen Sigg, Eemil Lagerspetz, Ella Peltonen, Petteri Nurmi, and Sasu Tarkoma

Exploiting Usage to Predict Instantaneous App Popularity: Trend Filters and Retention Rates

Manuscript under submission.

III

Contribution: The publication was lead by Prof. Stephan Sigg who delivered the main ideas, methodology, and structure of the publication. The author contributed by participating in the planning of the publication, and implementing and running the application recommendation system for the validation and use case of the trend filter analysis. The author also gave comments through the process and participated in the writing of the publication together with other authors.

Exploiting usage to predict instantaneous app popularity: Trend filters and retention rates

STEPHAN SIGG, Aalto University
EEMIL LAGERSPETZ, University of Helsinki
ELLA PELTONEN, University of Helsinki
PETTERI NURMI, University of Helsinki
SASU TARKOMA, University of Helsinki

The popularity of mobile apps is traditionally measured by metrics such as the number of downloads, installations, or user ratings. A problem with these measures is that they reflect actual usage only indirectly. We propose to exploit actual app usage statistics. Indeed, retention rates, i.e., the number of days users continue to interact with an installed app have been suggested to predict successful app lifecycles. We conduct the first independent and large-scale study of retention rates and usage trends on a database of app-usage data from a community of 339, 842 users and more than 213,667 apps. Our analysis shows that, on average, applications lose 65% of their users in the first week, while very popular applications (top 100) lose only 35%. It also reveals, however, that many applications have more complex usage behavior patterns due to seasonality, marketing, or other factors. To capture such effects, we develop a novel app-usage behavior trend measure which provides instantaneous information about the popularity of an application.

Our analysis shows that roughly 40% of all apps never gain more than a handful of users (*Marginal* apps). Less than 0.4% of the remaining 60% are constantly popular (*Dominant* apps), 1% have a quick drain of usage after an initial steep rise (*Expired* apps), and 7% continuously rise in popularity (*Hot* apps). From these, we can distinguish, for instance, trendsetters from copycat apps. We conclude by demonstrating that usage behavior trend information can be used to develop better mobile app recommendations.

CCS Concepts: •**Information systems** → **Information retrieval;** *Content analysis and feature selection; Users and interactive retrieval; Probabilistic retrieval models; Specialized information retrieval;*

Additional Key Words and Phrases: Mobile Analytics; Application Popularity; Trend Mining

ACM Reference format:
Stephan Sigg, Eemil Lagerspetz, Ella Peltonen, Petteri Nurmi, and Sasu Tarkoma. 2016. Sovereignty of the Apps: There's more to Relevance than Downloads. 1, 1, Article 1 (January 2016), 21 pages.
DOI: `0000001.0000001`

1 INTRODUCTION

With the popularity of mobile apps continuing rapid growth, judging on the potential of an individual app has become far from straightforward. Studies on app marketplaces have shown that overall rating and the nature of user reviews are key drivers in application download decisions [19, 24]. Recommendations, though, are biased towards apps with a

large user base. High-potential successful apps with a smaller community of users, have a higher risk to vanish in this digital cornucopia. In particular, as detailed in [43], a Matthew effect is fostered by rating systems in current app stores. Furthermore, ratings are vulnerable to spam and rating fraud [11, 21] and downloads can be misleading as users install apps simply to try them out [3, 7] and rate apps negatively for a wide range of reasons, such as technical problems or lack of features [23].

Retention rate has been proposed as a usage-based metric to measure the success of mobile apps[1]. Retention rate reflects the number of days that users continue to interact with an application after installing it. Reports[2] suggest that retention rates of most applications are low. However, thus far no independent information about retention rates of mobile apps is published.

As our **first contribution**, we present results from the first independent study of retention rates in the wild. The accuracy and generality of previous reports is unclear since the studies have not been independent[1,2]. We perform our study on usage behavior data recorded in the frame of the Carat project [31] from 339,842 Android devices over a period of three years (June 2012 – July 2015). Our results indeed confirm that, on average, applications lose 70% of their users within a week from first use. However, these effects are mediated by the number of users an application has. For applications with 10 or more users, first day retention rates below 30% are rare, and retention rates increase with better known apps. However, retention rates alone are not a sufficient measure of an app's success as they ignore fluctuation in instantaneous usage, the effects of seasonality, external factors, and other long-term usage behavior trends.

Such information could provide developers feedback about instantaneous popularity of their application, so that they can apply timely changes as countermeasure. Furthermore, such fine-grained trend information can also be used for marketplace analytics to support advertising, or, on the other hand, trend status can be used as an additional metric to clean the store of irrelevant copycat apps.

As our **second contribution**, we therefore present a novel app-usage trend filter which can capture and quantify these effects. It describes the relative popularity of apps based on daily use, indicates behavior trends regardless of absolute volume and categorizes apps into *App trend patterns* that predict the future relevance of an app.

We validate our app-usage filter through a large-scale analysis of mobile app usage trends on the above mentioned data set over the period June 2012 – July 2015. We consider selected applications, which are widely acknowledged as successful or unsuccessful apps, and demonstrate that the usage behavior trend-filter can correctly identify them. We further apply our filter to all apps in the dataset, characterizing their popularity. Our results show that 40% never ever acquire more than a handful of users (*Marginal* apps), and in the remaining 60%, only 0.4% are popular over a continuous period (*Dominant*), 1% are drastically drained in their usage after an initial steep rise (*Expired*), and 7% are continuously rising in popularity (*Hot*).

As a **practical use case**, we finally analyze the performance of a state-of-the-art mobile app recommender [45] with respect to trends. Our analysis shows that only 3.6% of the recommendations are for apps which are currently rising in popularity, and that overall recommendations have low novelty and temporal diversity. We also demonstrate that the accuracy of the recommendations can be improved by considering trend information.

[1] `http://info.localytics.com/blog/the-8-mobile-app-metrics-that-matter`
[2] `http://andrewchen.co/new-data-shows-why-losing-80-of-your-mobile-users-\is-normal-and-that-the-best-apps-do-much-better/`

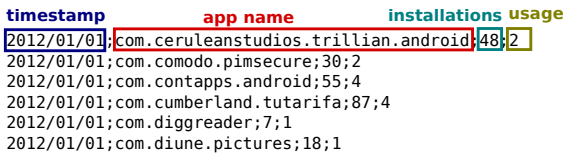


Fig. 1. App-usage data from Android: process name, installation count, and daily usage.

2 DATASET

Our work considers data recorded with Carat [31][3], a stock Android and iOS mobile app designed to offer personalized recommendations to improve a device's battery life. Carat uses energy-efficient and non-invasive instrumentation to record the state of the device, including the process list, and active apps. Carat has been deployed on over $800,000$ smartphones, roughly half of which are Android devices. The community of users that contribute data to Carat is spread all over the world, with users in roughly 200 countries, and a strong presence in USA, most of Europe, India, and Japan.

For this paper, we consider a subset of the data covering a period of three years (June 2nd, 2012 – July 14th, 2015) that contains measurements from 339,842 Android devices. We limit our analysis to Android devices as the data obtained on Android devices can be uniquely mapped into individual applications [40], whereas data obtained on iOS devices requires more complicated processing as the app names obtained from iOS in Carat are not human-readable (apps are presented by IDs only, and IDs differ for different versions of the app).

As part of our analysis, we assess the popularity of apps within categories. To carry out this analysis, we combine the crowdsourced data with category information from Google Play. The resulting dataset includes a total of 13,779,666 app usage records from 339,842 users and 213,667 apps in 47 categories. We only use the leaf categories of Google Play. *Games* and *Family*, for instance, are groups of categories, while two of their individual categories are *Games: Racing* and *Family: Action*. The scale of the dataset we consider in our analysis is an order of magnitude larger than in previous works. For example, Harman et al. [19] considered a dataset containing reviews from $30,000$ apps, whereas Böhmer et al. [5] considered measurements from $4,125$ users and $22,626$ applications. Our proposed approaches are able to estimate the popularity of apps with respect to a given measurement interval (June 2nd, 2012 – July 14th, 2015). Results achieved for other measurement periods might differ as a result of different lifecycle states of an app.

Ethical Considerations: We analyse aggregate-level data which contains no personally identifiable information. The privacy protection mechanisms of Carat are detailed in [31]. Data collection by Carat is subject to the IRB process of University of California, Berkeley. Users of Carat are informed about the collected data and give their consent to use data from their devices.

Figure 1 details a small sample of the data we utilized for our study on usage trends. Note that only anonymized data is exploited (e.g. timestamp, app name, usage) and that even the installation count was necessary only for our comparison to a state-of-the-art recommendation system in Section 6. For the calculation of retention rates in Section 3.1, anonymized user IDs have been necessary in addition. Consequently, in contrast to other methods exploiting personal information and usage information, our approach is applicable to open application at large scale. Usage frequency data is potentially discontinuous (especially for apps resembling the *Marginal* pattern), and absolute values of different apps vary. To meaningfully compare app usage trends, normalization with respect to the total usage count (relative popularity over

[3] `carat.cs.helsinki.fi`

time) and within maximum usage of the app (popularity within a particular day) is therefore necessary. Missing data is interpolated for discontinuous usage patterns.

We identified the following potential limitations due to the particular data set utilized. First, data was collected using a custom mobile application, which itself is prone to the studied usage trends and retention. Furthermore, Carat collects measurements continuously in the background of the mobile device, but only sends the data when launched. Accordingly, as long as the user launches the application *once* after a sufficiently long period from initial use, we obtain sufficient data to carry out our analysis. In order to minimize this effect, for our results on general properties of the dataset, we considered all uploads in the period (June 2nd, 2012 – July 14th, 2015) made until June 2nd 2017. To further limit potential biases caused by users stopping to use Carat, we only considered users who had used Carat over a sufficiently long period, e.g. a month.

While comparing the popularity of apps within a category, we relied on category information extracted directly from Google Play. On Google Play, the categorization of an app is the responsibility of the developer, and consequently similar apps are likely to contain variations in their categorizations. An alternative would be to rely on topic models to derive a categorization of the apps; e.g., Gorla et al. [16] have demonstrated the use of Latent Dirichlet Allocation (LDA) for mining categories from app store data. Alternatively, trend information could be integrated as part of the topic models together with additional factors, such as number and nature of ratings, and contents of user reviews.

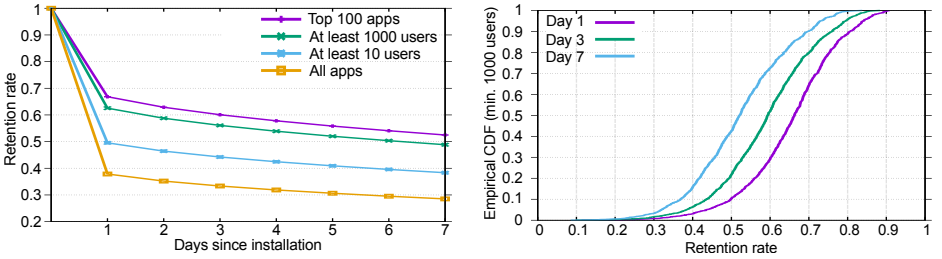
3 ANALYSIS OF RETENTION RATES

The few existing academic studies on mobile app usage have characterized factors that drive download decisions [19, 24, 33] without being able to determine what happens once the app has been installed on the device. While some studies have relied on measurements taken on the handsets, they have focused on overall usage and how that is influenced by contextual factors [5, 13, 37], leaving commercial reports by mobile analytics companies the only source of information about what happens once the app has been downloaded. Such reports suggest that usage dwindles significantly after installation (i.e., low retention), and even 80% of users stopping to use the application is common. In this section we present the first *independent* and large-scale study to investigate whether this indeed is the case.

3.1 Retention Rate

Retention rate on day d is defined as the percentage of users that continue using an application d days after first usage. To estimate retention rates, we identify for each user and application the first and last time the user launched the application. To ensure usage behavior is correctly captured for retention rate of up to d, for each app we only consider those users who had not been using the app within d days of the last measurement day (July 14th 2015).

The retention rates of mobile apps in our dataset are illustrated in Figure 2a. From the figure we can observe that, while retention rates of many applications indeed are low, there are several apps with a healthier usage lifecycle. Overall, for all apps, the first day retention is as low as 0.36 with 7 day retention falling below 0.3. However, our results also suggest that this effect is mainly due to many apps receiving only few users. Indeed, retention rates for apps with at least 10 users until the last measurement day show much healthier behavior, with first day retention being close to 0.5. For apps with at least 1000 users the same figure rises to 0.62. For the most popular 100 apps, first day retention is even as high as 0.68 and after 7 days the retention remains higher than 0.5. In summary, our analysis supports the view that retention rates of apps tend to be low, with the usage witnessing a steep decline particularly after the first use. However, our analysis also calls into contention some of the claims made by analytics companies, indicating that the number of users mediates the retention of



(a) Retention rate largely depends on the initial number of users, with popular apps staying healthy for longer periods of time.

(b) Empirical cumulative distribution of retention rate for Apps with at least 1,000 users.

Fig. 2. Retention rates and their distribution

applications, with even apps with as few as 10 users over the whole time window considered witnessing significantly better retention rates.

To further shed light on retention patterns, Figure 2b illustrates the empirical cumulative distribution function of retention rates for apps with at least $1,000$ users. In the plot we separately consider the retention rates of day 1, day 3 and 7. These days were chosen for our results to be comparable with results published[1]. From the plot we can see that high retention rates are rare. Indeed, only $10-35\%$ of apps have retention rates of 0.7, and a mere 3% of apps is able to achieve 0.75 retention rate on day 7. However, from the figure we also observe that extreme drops are rare, with less than 5% of applications having retention rates below 0.3, i.e., the 80% drop reported in the literature is not common for apps that have been able to attract a sufficient user base.

We observed similar patterns for apps with less than $1,000$ users. However, since there are orders of magnitude more apps with only a handful of users, as opposed to, for instance, hundreds of users, the retention rate of the entire data is then biased. Apps with 100 users or less are very volatile in terms of retention rate, since a drop of a single user already decreases retention by 1% or more. The plot for apps with at least 10 users within the whole measurement period follows a similar, but more jagged pattern, and rises much faster with retention rates around 10% lower than in Figure 2b.

This also further supports our earlier finding of retention being mediated by the size of the user base and reports by others[2]. To verify this, we used Spearman correlation to assess the statistical dependency between usage counts and retention rates. To limit potential biases and noise in the retention rate estimates, we only considered apps with at least 10 users within the complete measurement period. The resulting analysis revealed the correlation to be statistically significant for all days ($d = 1, \rho = 0.199, p < .001; d = 3, \rho = 0.185, p < .001; d = 7, \rho = 0.165, p < 0.001$). For applications with higher usage count, correlations were slightly lower, but remained consistently significant.

For apps with only $10-15$ users, some fraction of the retention could be potentially explained by developers of the apps continuing to test and use their app. Unfortunately identifying these users from the Carat data is not possible.

We conclude that retention is indeed an indicator of the overall usage trend an app might experience over its lifetime. However, as we demonstrate next, it does not provide a full picture of app usage. In particular, we propose the use of trend filters that are able to compare apps regardless of their user base or absolute usage count for instantaneous trend patterns and seasonal effects.

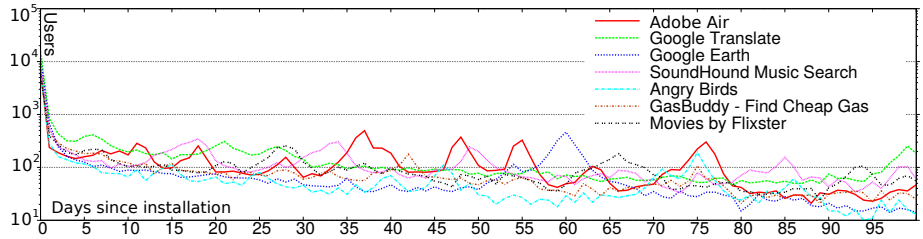
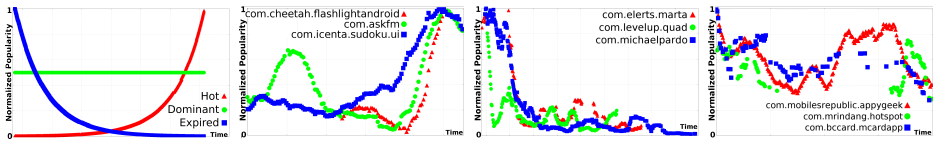


Fig. 3. App usage patterns do not always follow a simple falloff graph as suggested by retention rate.



(a) Prototype patterns (b) Example *Hot* apps (c) Example *Expired* apps (d) *Dominant* apps

Fig. 4. App trend patterns with example apps.

3.2 Beyond Retention Rates

While retention is able to indicate the long-term usage trend of an app, it does not cover instantaneous popularity, trends or seasonal patterns. Furthermore, since app-usage is also conditioned on external context and occasion [6], apps with seasonal usage patterns, such as recommendation of lunch places, nearby gas stations or vacation-related are unfairly treated by retention rates. This is in particular true when the time window is short, such as the often cited 1-day/3-day/7-day retention characteristics. For instance, Figure 3 depicts usage patterns of exemplary applications from first day of usage until 100 days of usage. The selection of applications was done automatically using a peak detection algorithm that identifies significant peaks in usage after the initial slide in usage.

In the figure, *GasBuddy* is a representative example with clear seasonal pattern. It compares fuel prices at nearby gas stations. The use of the application dwindles after day 1, but has recurrent spikes at biweekly and monthly intervals. Other examples include the *Adobe Air* game store/platform (regular peaks), or *Angry Birds* (many small peaks). Utilities, such as the music identifying *SoundHound*, *Google Translate*, and *Google Earth* (both irregular peaks) are used on-demand as their importance to users is situational.

To capture these effects and to provide a more accurate view of the usage of an application, we next develop a novel trend mining approach for capturing application lifecycles.

In particular, in this study, we are interested in the four trends 'rising in popularity' (*Hot*), 'falling in popularity' (*Expired*), 'constant high popularity' (*Dominant*) and 'no popularity' (*Marginal*) (cf. Figure 4). These prototype patterns can be prominently found in the Carat dataset (cf. Section 5.2). We expect that the basic trends *Hot, Dominant*, and

Expired will resemble the three characteristic patterns illustrated in Figure 4a[4]. Figure 4b, 4c and 4d depict exemplary app usage evolutions for specific apps filtered with one of these prototype trend patterns.

4 MINING REAL-TIME TRENDS

We propose instantaneous application usage as a novel metric to accurately characterise the momentary popularity of mobile apps. Existing metrics, such as aggregated installation counts, co-installed apps and user reviews, often contain noise and biases making them an unreliable measure of an app's success or failure. Moreover, these measures ignore effects of temporal and other seasonal or external factors.

Our trend filter, in particular, abstracts from absolute user count or usage and, instead, is able to compare applications based solely on their momentary trend potential. In this way, also apps with a small user base can compete with well known competitors. As initial transformations for any specific app $\mathcal{A}$ with a usage time series $U(\mathcal{A}) = u_{\mathcal{A},1}, ..., u_{\mathcal{A},n}$, the trend filter interpolates missing values $u_{\mathcal{A},j}$ as

$$u_{\mathcal{A},j} = \frac{u_{\mathcal{A},k} - u_{\mathcal{A},i}}{k-i}; i < j < k \in \mathcal{N} \tag{1}$$

and then operates on the normalized absolute usage

$$\hat{U}(\mathcal{A}) = \hat{u}_{\mathcal{A},1}, \ldots, \hat{u}_{\mathcal{A},n} \tag{2}$$

$$\hat{u}_{\mathcal{A},i} = \frac{u_{\mathcal{A},i}}{\max_j(u_{\mathcal{A},j})}, i = 1..n \tag{3}$$

In the following, we describe four modules constituting various operations regarding the trend app usage time series $U(\mathcal{A})$.

4.1 Module A: Group apps with respect to specific trend pattern

As an initial pre-processing for some other modules and also in order to identify apps that follow specific prototype trend patterns (e.g. *Dominant, Expired, Hot*, and *Marginal* (cf. Figure 4)), a set of apps is clustered according to their similarity (Euclidean distance in the feature space) to these prototypes. Clusters are achieved with k-means clustering according to the input pattern's similarity to specific prototype patterns. We utilise k-means as it allows to specify a fixed but arbitrary number of cluster heads. For instance, these cluster heads could constitute the four respective prototypical trend patterns *Hot, Expired, Dominant*, and *Marginal*.

As a measure to compare normalized usage patterns via k-means, we measure similarity via their Euclidean distance in a feature space spanned by the *Area Under the Curve* (AUC), *Relative Peak location* (PEAK), *Slope* (SLOPE) and

[4]Apps with very low usage (*Marginal* pattern) closely resemble a straight line with constant value '1'. This is due to our pre-processing of applications to make them comparable: Normalization with respect to the highest observed daily use followed by the interpolation of missing values (cf. Section 4)

Variance (VAR).

$$\text{AUC}\left(\hat{U}(\mathcal{A})\right) = \sum_{i=1}^{n} \hat{u}_{\mathcal{A},i} \tag{4}$$

$$\text{PEAK}\left(\hat{U}(\mathcal{A})\right) = \arg\max_{j}\left(\hat{u}_{\mathcal{A},j}\right) \tag{5}$$

$$\text{SLOPE}\left(\hat{U}(\mathcal{A})\right) = \frac{\hat{u}_{\mathcal{A},n} - \hat{u}_{\mathcal{A},1}}{n-1} \tag{6}$$

$$\text{VAR}\left(\hat{U}(\mathcal{A})\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{u}_{\mathcal{A},i} - \hat{\mu}_{\mathcal{A}}\right)^2 \tag{7}$$

where $\hat{\mu}_{\mathcal{A}}$ describes the arithmetic mean of $\hat{U}(\mathcal{A})$.

The choice of these features is motivated by the nature of the four trend patterns to distinguish. *PEAK* and *SLOPE* are able to distinguish *Hot* or *Expired* patterns, whereas the *AUC* distinguishes those from the other two constant patterns which have a much larger *AUC*. Finally, *VAR* and *AUC* are able to distinguish between *Dominant* and *Marginal* as the latter will have low *VAR* and high *AUC*.

Let N be the number of points, D the number of dimensions, and K the number of centers. Based on the number of distance calculations, the time complexity of k-means is $\mathcal{O}(NKI)$. The space complexity of k-means clustering is then $\mathcal{O}\left(N(D+K)\right)$[20]. We note that variants of k-means exist with a tight asymptotic bound on the expected run-time complexity of $\mathcal{O}(\log k)$ [41].

4.2 Module B: Extract representative patterns

In order to identify a single representative usage trend pattern for a group of apps clustered to $\mathbf{C} = \{\hat{U}(\mathcal{A}), \ldots, \hat{U}(\mathcal{Z})\}$, this module allows us to calculate a consensus $\hat{\mathcal{C}}(\mathbf{C})$ from all apps in the cluster. We exploit this module, for instance, to identify a representative pattern of a Google Play category by computing the consensus for all apps in the category. It can also be applied to arbitrary groups of apps, for instance, to compare their average popularity (e.g. different groups of games or applications from a specific developer)

A challenge we address with this module is that app lifecycles are not are not synchronized with respect to their absolute occurrence time. For instance, for the *Expired* pattern, the peak point for each app in a respective cluster might be uniformly distributed over time.

We therefore first synchronize all apps with respect to their peak as

$$\bar{u}_{\mathcal{A},l} = \hat{u}_{\mathcal{A},i} \tag{8}$$

$$l = i - \arg\max_{j}\left(\hat{u}_{\mathcal{A},j}\right) \tag{9}$$

From all usage timeseries $\hat{U}(\mathcal{A}), ..., \hat{U}(\mathcal{Z})$ in one cluster $\mathbf{C} = \{\hat{U}(\mathcal{A}), \ldots, \hat{U}(\mathcal{Z})\}$, a consensus timeseries $\mathcal{C}(\mathbf{C})$ is constructed as the mean over all time series in that cluster as

$$\mathcal{C}(\mathbf{C}) = \hat{c}_1, \ldots, \hat{c}_n \tag{10}$$

$$\hat{c}_i = \frac{\sum_{\hat{U}(\mathcal{I})\in\mathbf{C}} \hat{u}_{\mathcal{I},i}}{|\mathbf{C}|}; i \in [1,n]. \tag{11}$$

Let $\hat{U}(\mathcal{I}) \leq \hat{U}(\mathcal{J}), \forall \hat{U}(\mathcal{J}) \in \mathbf{C}$ be the longest usage time series in the cluster $\mathbf{C}$ with $|\mathbf{C}| = m$ and $|\hat{U}(\mathcal{I})| = n$. Then, the time and space complexity of this module is $\mathcal{O}(m \cdot n)$.

4.3 Module C: Determining the App Lifecycle

Many apps found in app stores don't show complete lifecycles but, instead, find themselves in the middle of a respective lifecycle. This means that apps not only follow distinct lifecycle patterns but, in particular, show incomplete lifecycles from their usage history patterns. For instance, an app might be in the beginning (initial stage), middle (rising) or end (past the peak) of a lifecycle. In order to find the stage of an app within a particular lifecycle, we apply alignment approaches. In particular, representative trend patterns for prototypical lifecycles $L = l_1, \ldots, l_o$ are aligned to the normalized observed app usage history patterns $\hat{U}(\mathcal{A})$ [34]. The alignment found constitutes a sequence $\tilde{L} = \tilde{l}_i, \ldots, \tilde{l}_j$ that originates from L and is similar to the usage pattern $\hat{U}(\mathcal{A})$. $\tilde{L}$ possibly omits leading and trailing samples of L and may feature additional gap-symbols which are inserted via integer programming to minimize the difference between $\tilde{L}$ and $\hat{U}(\mathcal{A})$. In particular, a $n \times o$ cost matrix M, spanned by $\hat{U}(\mathcal{A})$ and L is generated by calculating all possible matchings between the l_i and $\hat{u}_{\mathcal{A},j}$ with respect to a distance cost function $c(l_i, \hat{u}_{\mathcal{A},j}) \rightarrow \mathcal{R}$ and a gap cost d:

$$M_{i,j} = \min\left(M_{i-1,j-1} + c(l_i, \hat{u}_{\mathcal{A},j}), M_{i-1,j} + d, M_{i,j-1} + d\right) \tag{12}$$

The M_{ij} constitute the minimum cost to align $l_1 \ldots l_i$ with $\hat{u}_{\mathcal{A},1} \ldots \hat{u}_{\mathcal{A},j}$. The optimal alignment $\tilde{L}$ is then found by traversing the minimum-cost path through M. Note that we set $M_{1j} = 0$ to allow $\tilde{L}$ to start at any position within L. Assuming a maximum sequence length of n, time and space complexity of this module are $\mathcal{O}(n^2)$.

4.4 Module D: Identifying apps that drive the trend

Some groups of apps, for instance, categories, are dominated by individual overly popular apps. This may happen when users of an overly popular apps try other, similar apps in the same category. Due to this, the observed usage of several apps in a category might be affected by individual popular apps and multiple apps in that category rise in popularity. We are then interested in the usage trend normalized by the overall trend of the category or group of apps. In this way we are able to distinguish those apps that drive and indeed exceed the category's trend performance from others that are underperforming with respect to the trend performance of the overall category.

This is achieved by normalizing the performance of individual apps against the performance of the category. In particular, given a group of apps or category $\mathbf{C}$, we first compute the consensus $\mathcal{C}(\mathbf{C})$ of $\mathbf{C}$ (cf. module B). Then, for each app $\mathcal{A}$, we normalize its usage pattern $\hat{U}(\mathcal{A})$ with respect to $\mathcal{C}(\mathbf{C})$ as

$$\underline{\hat{u}_{\mathcal{A},i}} = \hat{u}_{\mathcal{A},i} - \hat{c}_i, i = 1..n \tag{13}$$

The resulting pattern $\underline{\hat{U}(\mathcal{A})}$ displays the performance of the app with respect to all other apps in the same category. A positive slope in $\underline{\hat{U}(\mathcal{A})}$ indicates that the app is performing better than its category while a negative slope indicates under-performance.

Assuming a total of $|\mathbf{C}| = m$ apps within the group of apps or category $\mathbf{C}$ with maximum pattern length $|\hat{U}(\mathcal{A})| = n; \mathcal{A} = \arg\max_{\mathcal{B} \in \mathbf{C}} \left(|\hat{U}(\mathcal{B})|\right)$, the time complexity of this module is $\mathcal{O}(n \cdot m + n + m)$. The space complexity is $\mathcal{O}(n \cdot m)$.

5 EMPIRICAL EVALUATION

To demonstrate the value of the trend filter, we carried out experiments using the Carat dataset (cf. Section 2). We first show that our trend filter is capable to identify known success and failure stories. Afterwards, we analyse the accuracy of our approach for the prediction of the respective trends by computing the distance of the predicted trend to the ground

Fig. 5. Exemplary popular apps chosen for comparison in our empirical evaluation

truth. Next, we investigate the frequency in which trends occur within different application categories and finally, we discuss the impact individual apps can have on the usage performance of similar apps.

5.1 Validation with exemplary well known apps

We examine eight popular apps (cf. Figure 5), four of which are believed to be exceeding expectations and four which have been extremely popular once but significantly dwindled since.

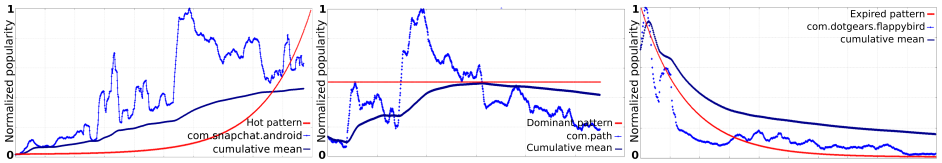
Grouping these apps with our trend filter approach according to the four representative trend patterns (cf. Figure 4, and Module A in Section 4), three of the popular apps, Vine, Evernote, and Snapchat are grouped with the *Hot* pattern, whereas Whatsapp is grouped as *Dominant* type. From the underperforming apps, three (*Flappy Bird*, *Weibo* and *QQ*) are grouped as *Expired* whereas *Path* is grouped in the *Dominant* cluster.

This grouping means that the employed k-means clustering groups these apps to the *Hot, Dominant* and *Expired* clusters according to the Euclidean distance of their observed usage pattern to the respective trend pattern. The Euclidean distance to one of the prototype clusters expresses the confidence of our trend filter on the respective categorization. In each of the four dimensions *AUC, PEAK, SLOPE* and *VAR* (cf. Section 4.1), the Euclidean distance ranges from 0 to 1. Consequently, the overall Euclidean distance ranges from 0 to 2 with an average Euclidean distance of 1.0. As further detailed in Section 5.2, we apply a confidence threshold of 0.4. For apps with an Euclidean distance to the closest trend pattern smaller than the confidence threshold, our trend filter indicates a high confidence that the observed pattern follows the associated trend. For apps with distances to their closest trend pattern that exceed the confidence threshold, the closest trend is indicated but no trend estimation is made. Table 1 illustrates the distance of each app to the nearest of the four representative trend patterns and the trend prediction.

From the apps that exceed expectation, the trend of *Evernote* indeed is predicted as *Hot* and the trend of *WhatsApp* is identified as *Dominant*. Among the apps that are trailing expectations, the trend of *Flappy bird, Weibo* and *QQ* are identified as *Expired*. These trend categorizations confirm the typical public perception.

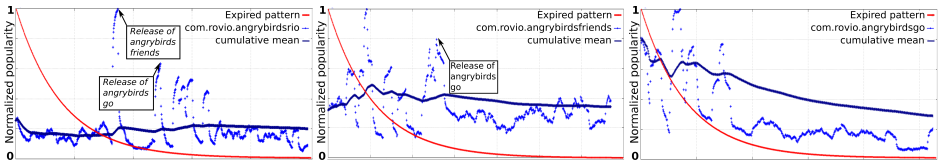
Table 1. Categorization of example applications.

App	Category	closest Pattern	Distance	Trend
Vine	Entertainment	*Hot*	0.4220	–
Evernote	Productivity	*Hot*	0.3956	*Hot*
Snapchat	Social	*Hot*	0.5399	–
WhatsApp	Communication	*Dominant*	0.1186	*Dominant*
Path	Social	*Dominant*	0.4467	–
Flappy Bird	Game - Arcade	*Expired*	0.0575	*Expired*
Weibo	Social	*Expired*	0.2343	*Expired*
QQ	Social	*Expired*	0.2854	*Expired*



(a) Trend evolution of Snapchat (b) Trend evolution of Path (c) Trend evolution of Flappybird

Fig. 6. Trend evolution, cumulative mean closest prototype trend pattern for example apps



(a) Trend of Angrybirds Rio (b) Trend of Angrybirds Friends (c) Trend of Angrybirds Go

Fig. 7. Trend evolution, cumulative mean closest prototype trend pattern for example Angry Birds apps correlated in their usage performance

For the remaining apps (*Vine, SnapChat, Path*), the trend filter has low confidence and therefore does not suggest a trend. The low confidence for assigning a trend to *SnapChat* reflects also the lower app-store ratings it has received compared to *Evernote and WhatsApp*. The low confidence for the *Path* app indicates that it is still in the transition between lifecycle states. We can observe this from Figure 6. While *Snapchat* resembles the *Hot* trend most closely, *Flappybird* is in its complete trend evolution closer to the *Dominant* trend than to the *Expired* trend.

We would further like to stress that the *Expired* pattern should not be interpreted as negative. On the contrary, it indicates that the app was successful to gather a huge user base, but experienced quick loss in users thereafter, i.e., it has low retention rate. As discussed in Section 3.1, it is a natural matter of retention that the usage drain is significant, and expired simply means that the app has surpassed its 'best before' date. Consider, for instance, the Angry Birds series of apps. As indicated in Figure 7, the patterns of most Angry Birds apps closely resemble the expired pattern (shifted by their respective release date) even if most of them can be considered to be exceptionally successful. Furthermore, we can

Table 2. Percentage of *Marginal*, *Expired*, *Dominant*, and *Hot* apps for 17 exemplary categories

	Books and Reference	Business	Comics	Communication	Education	Entertainment	Family	Finance	Game (Action, Adventure, Arcade, Board)	Health and Fitness	Lifestyle	Media and Video	News and Magazines	Personalization	Productivity	Tools	Travel and Local
Marginal apps(%)	43.28	38.16	31.94	41.2	38.78	38.63	49.81	47.2	38.62	42.54	40.3	46.1	41.35	40.15	44.78	43.97	39.62
From the rest:																	
Hot apps(%)	1.33	2.65	4	7.44	.81	2.33	4.65	4.4	1.73	3.92	3.17	6.5	4.74	2.25	6.78	5.98	2.95
Dominant apps(%)	.04	.06	0	.14	0	.03	0	.03	0	0	.07	.11	.07	.02	.06	.07	0
Expired apps(%)	.25	.74	.8	1.91	.03	.52	.9	.63	.52	.36	.37	0	.95	.74	2	1.51	.64
Mean Eucl. Dist.	.974	.967	.914	.906	.996	.958	.921	.961	.938	.967	.957	.947	.926	.961	.926	.925	.958
Variance	.010	.012	.018	.027	.008	.013	.020	.016	.016	.013	.014	.018	.023	.015	.023	.024	.012

observe that a series of apps in the same theme or product family has the potential to benefit each other as a compounding effect. We see in Figure 7a and Figure 7b how the much older *Angry Birds Rio* also experiences a rise in popularity at the time the *Angry Birds Friends* and *Angry Birds Go* are released. These peaks can also be observed at very similar times for the original *Angry Birds* and the older *Angry Birds Seasons*.

5.2 Evaluation of the quality of the clustering

We next analyse the quality of the trend prediction with respect to the distance to the respective trend pattern (Figure 4a). Table 2 summarizes the frequency of the four trend patterns in exemplary Google Play categories. Throughout all categories, about 40% or more of the apps are marginal. This means that from all apps in Google Play, less than 60% ever gather more than a handful of users.

Of the apps associated with one of the four main trend patterns, less than 0.1% gather a constantly high user base (*Dominant* apps). Fewer than 1% are *Expired*, and apps associated to the *Hot* category account for about 2 to 7% of all relevant apps. The mean Euclidean distance of all individual apps to the associated (i.e. closest) trend pattern is in the order of 0.95, with a variance of 0.2 as detailed in the table.

However, we only predict a particular trend pattern, when it has the closest Euclidean distance of all trend patterns and when this distance to the respective cluster centroid (*Hot*, *Expired*, *Marginal* or *Dominant*) falls below 0.4. These apps, for which a particular trend evolution is predicted are significantly closer to the respective cluster centroid as detailed below. The value of 0.4 was chosen empirically as (1) a sphere of radius 0.4 covers approximately[5] 10% of the total volume in a 4D unit space (spanned by the four feature values, each in [0,1]) and (2) since it was able to clearly separate the apps belonging to one of the four patterns from those that do not belong to it (cf. Figure 8). Note that it is challenging to define an absolute threshold as no ground truth on the respective app trends exists. In the figure, we have calculated the mean Euclidean distance for groups of apps clustered to the same of the four respective trend patterns to their nearest cluster centroid. The figure separates apps with a Euclidean distance of greater than 0.4 from those with a larger Euclidean distance.

For the apps associated with one of the trend patterns (i.e. Euclidean distance smaller than 0.4), the *Marginal* apps are most similar to their respective trend pattern and the mean Euclidean distance for *Hot*, *Dominant* and *Expired* apps is for all categories sharply concentrated around 0.3. As we have seen in Table 2 already, the distance of the remaining apps to their nearest trend pattern is significantly higher. This means that they might follow other, more complex patterns or

[5] A more exact approximation would be 0.377, which we rounded up to 0.4 to achieve a slightly larger noise tolerance. Results achieved for 0.4 and 0.377 are nearly identical

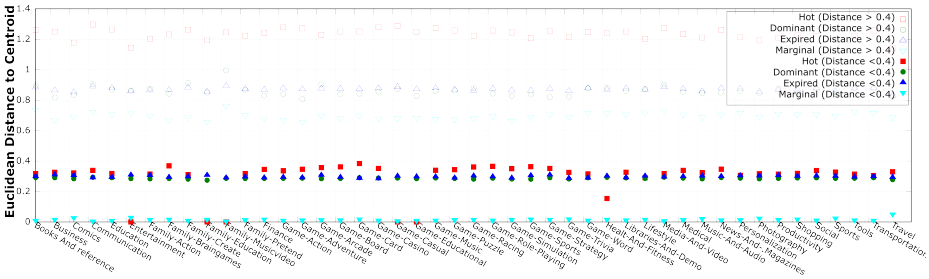


Fig. 8. Mean Euclidean distance to the respective cluster centroid (*Hot*, *Dominant*, *Expired*, *Marginal*) for apps associated with the respective trend pattern (low Euclidean distance) compared to those not associated with it (high Euclidean distance)

Table 3. Distribution of the arithmetic mean of clusters found (rounded to 1000 for space constraints)

$\mathbf{C}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\overline{\mathbf{C}}$ $[10^3]$	629	820	952	1079	1163	1285	1342	1385	1474	1550	1707	1857	2023	2180	2302	2608	3159	3566	4482

experience constant fluctuation. While we focus in this study on *Hot*, *Dominant*, *Expired*, and *Marginal* patterns, our approach can be generalized to arbitrary other patterns in the data.

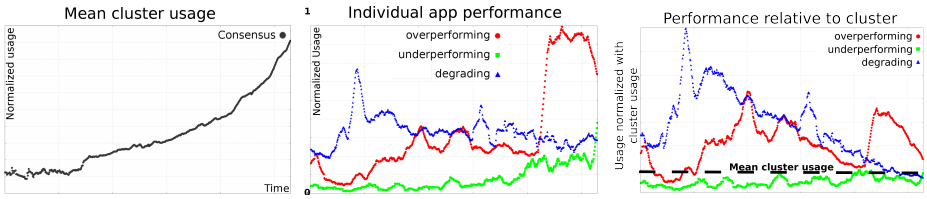
5.3 Distribution of distinct trend patterns

We have investigated the count and frequency of diverse patterns in the carat data. In particular, we are concerned with the number of different relevant trend patterns that can be found as the *Hot, Dominant, Expired* and *Marginal* patterns constitute only part of the patterns present. In order to understand this, we clustered the lifecycle patterns found for all apps in the carat dataset with k-means clustering and larger values of k. In all cases, as discussed in Section 5.2 and visible from Table 2, an overwhelming share of apps follow the *Marginal* cluster. For the remaining clusters observed, we found that few prominent clusters dominate. For instance, Table 3 displays for $k = 20$ the sorted arithmetic mean of the non-*Marginal* clusters found among the apps in the carat dataset after 10 runs of k-means. The *Hot*, *Expired*, and *Dominant* trend have been prominently found among the clusters computed in all runs of the k-means algorithm. Note from the figure, that the three largest clusters jointly represent about $\frac{1}{3}$ of the apps with a long tail of small clusters. This shows that, although more than the *Hot*, *Expired*, and *Dominant* (and *Marginal*) patterns can be found in the data, the relevance of these four clusters is significant.

5.4 Impact of Dominant Apps within a Category

We are further interested in the popularity of specific apps with respect to others. However, apps can fall into various categories and might not be comparable, such as, for instance, categories *Game* and *Business*. In particular, in the Google Play Store, some categories might increase in popularity while others decrease. As detailed in Section 4, this overall trend of the category might affect other apps within the category, so that an unbiased comparison across categories requires prior normalization (cf. Module D in Section 4).

Overly successful apps have a significant impact in the trend of their category. For instance, the Facebook app dominates the performance of its category. Compared to the consensus of all apps in this category, *com.facebook.katana*



(a) Consensus of the cluster, describing the average popularity of apps in this cluster

(b) Normalized performance of three individual apps from this cluster

(c) Relative performance of the apps with respect to the cluster

Fig. 9. Performance of individual apps relative to the performance in their cluster.

features with 0.2105 the smallest Euclidean distance in the feature space to the consensus pattern. Smaller apps might then appear to follow a rising trend but in reality they have merely been benefiting from the overall popularity of the category. For a fair comparison of apps that belong to different categories, the overall trend of the categories should therefore be subtracted out.

For recommendation or analysis purpose, apps that drive the trend are especially interesting over copycat-apps which are worse than the trend. To objectively measure the performance of an app, free of the influence of its category, we calculate its performance relative to the performance of its category (cf. Module D in Section 4). To illustrate that normalization against the overall trend in a category is beneficial to identify the actual instantaneous performance of an app consider Figure 9 for three exemplary apps in one category. Figure 9a plots the consensus of the app category. Looking at an individual app's performance (Figure 9b), their trend is hardly visible. However, after normalization with the cluster's consensus pattern in Figure 9c, the app represented with ● is over-performing as it regularly scores above the cluster's performance while the app labeled ■ is constantly under-performing. Finally, the third app (labeled ▲) is dominating in the beginning while then constantly degrading in performance. Observe that, in contrast to this normalized evolution, considering the apps individually, as in Figure 9b, the under-performing app actually appears to be rising in popularity while the degrading app appears to remain stable.

6 PRACTICAL USAGE OF APP TRENDS

To demonstrate the practical value of our work, we now consider how trend information can affect app recommendations. We have implemented AppJoy [45] as a representative example of current state-of-the-art app recommenders, and compared the recommendations provided by AppJoy against recommendations indicated by trend patterns. AppJoy operates on so-called usage scores, which are constructed by aggregating (i) the time elapsed since the last interaction with an app (v_R), (ii) the number of times the user interacted with an app (v_F), and (iii) total duration of interaction time (v_D). Accordingly, AppJoy bases its recommendations on information sources that correspond to metrics which are used by handset-based mobile analytics tools, such as Google Mobile Analytics and Countly.

AppJoy uses a prediction model that compares a user $\mathfrak{U}$'s profile to other users with similar application usage history. Let $\mathbf{S}(\mathfrak{U})$ be the set of applications used by $\mathfrak{U}$. Given an application $\mathcal{A}$ and $\mathfrak{U}$, we define $\mathbf{R}_{\mathfrak{U},\mathcal{B}}$ as the set of relevant applications $\mathcal{B}$ used by other users together with $\mathcal{A}$, so that

$$\mathbf{R}_{\mathfrak{U},\mathcal{B}} = \{\mathcal{A} | \mathcal{A} \in \mathbf{S}(\mathfrak{U}), \mathcal{B} \notin \mathbf{S}(\mathfrak{U}), size(\mathbf{S}_{\mathcal{A},\mathcal{B}}) > 0\} \tag{14}$$

Table 4. Statistics of the best 20 recommendations for the top 1000 applications during October 2014.

Week	Rec. Hot	Rec. Expired	Total Hot	Total Expired	Div.	Nov.	Acc.	Div. w/o expired	Nov. w/o expired	Acc. w/o expired
1	8	5	219	163	-	-	0.02	-	-	0.02
2	7	6	229	158	0.80	0.98	0.03	0.90	0.90	0.12
3	8	7	232	154	0.62	0.81	0	0.54	0.73	0.10
4	10	9	225	150	0.56	0.75	0.11	0.50	0.68	0.11

where $\mathbf{S}_{\mathcal{A},\mathcal{B}}$ is the set of users who have used both $\mathcal{A}$ and $\mathcal{B}$. The relevance or occurence probability of $\mathcal{B}$ for $\mathfrak{U}$ is then given by:

$$P(\mathfrak{U}_{\mathcal{B}}) = \frac{1}{size(\mathbf{R}_{\mathfrak{U},\mathcal{B}})} \sum_{\mathcal{A} \in \mathbf{R}_{\mathfrak{U},\mathcal{B}}} (dev_{\mathcal{A},\mathcal{B}} + \mathfrak{U}_{\mathcal{A}}). \tag{15}$$

In the above equation, $dev_{\mathcal{A},\mathcal{B}}$ denotes the average of the usage scores between users who have used both $\mathcal{A}$ and $\mathcal{B}$:

$$dev_{\mathcal{A},\mathcal{B}} = \sum_{\mathfrak{O} \in \mathbf{S}_{\mathcal{A},\mathcal{B}}} \frac{v_{\mathfrak{O} \vdash \mathcal{B}} - v_{\mathfrak{O} \vdash \mathcal{A}}}{size(\mathbf{S}_{\mathcal{A},\mathcal{B}})}. \tag{16}$$

Here, $v_{\mathfrak{O} \vdash \mathcal{B}}$ defines the usage score for application $\mathcal{B}$ used by user $\mathfrak{O}$:

$$v_{\mathfrak{O} \vdash \mathcal{B}} = \omega_R v_R + \omega_F v_F + \omega_D v_D, \tag{17}$$

weighted by ω_R, ω_F and ω_D. Given $P(\mathfrak{U}_{\mathcal{B}})$, AppJoy returns the apps with highest score as recommendation Φ.

To illustrate the value of trend and lifecycle information, we ran the AppJoy recommender and our trend analysis for a subset of the data containing $4,500$ users and $1,000$ most frequently used applications in the dataset. As our test period we selected October 2014, due to little seasonal fluctuations, and as training data we selected all data accumulated between January 2014 and September 2014. Given the test data, we used AppJoy to generate recommendations in an incremental fashion for each week. In particular, we generated recommendations for the first week, then included the data from this period in the training data and generated recommendations for the second week, and so on. We also generated trends for every week, taking into account the lifecycles of the past year, starting from January 1st 2014. We counted (i) how many recommended applications are grouped as *Expired* or *Hot* apps, and (ii) how these compare with the total number of *Expired* and *Hot* apps in the top 1000 applications. We also calculated temporal diversity, novelty, and accuracy for the recommendation lists [26]. Diversity presents how the recommendations change over time, whereas novelty describes how many new recommendations there are seen compared to the later ones. Novelty of the recommendations relates closely to the trends, because changes in trends should affect new recommendations. Given two sets $\mathbf{A}$ and $\mathbf{B}$ of apps and the set Φ of all recommended apps, as well as depth N, these metrics are defined as

$$diversity(\mathbf{A}, \mathbf{B}, N) = \frac{|\mathbf{B} \setminus \mathbf{A}|}{N} \tag{18}$$

$$novelty(\mathbf{A}, N) = \frac{|\mathbf{A} \setminus \Phi|}{N} \tag{19}$$

$$accuracy(\mathbf{A}, \Phi) = \frac{size(\mathbf{A} \cap \Phi)}{size(\Phi)}. \tag{20}$$

Results of our analysis are shown in Table 4 for the top-20 recommendations given to all users. The results indicate that the number of *Hot* apps recommended for each week is small and comparable to the number of *Expired* apps recommended in the same time. Given that we have generated in total $90,000$ recommendations for $4,500$ users each week, the amount of *Hot* recommended corresponds to a very small percentage of the entire set of recommendations. Within the top 1000 apps, more than 200 applications each week can be classified as *Hot*, and about 160 applications

as *Expired*. On average, only 3.6% *Hot* apps are recommended, compared to 4.3% *Expired* apps. When *Expired* apps are removed from the recommendations, both novelty and diversity decrease, but accuracy increases slightly. The main reason for this behavior is that the metrics used by AppJoy to generate recommendations require sufficient amount of usage before an app is recommended. However, once sufficient usage has been observed, the app can already be past its "best before" date as the recommendation model does not separate between *Hot* and *Expired* apps. Integrating usage trend information as part of the recommendation process can help to overcome this issue and improve the overall quality of recommendations.

In summary, our analysis clearly indicates that recommendations provided by AppJoy do not reflect dynamics in actual application usage. Other state-of-the-art recommenders, such as Djinn [22] and GetJar [38], are based on similar usage information and are hence likely to exhibit similar patterns compared to usage trends. To facilitate users to discover up and coming applications, and to help them avoid apps that are long past their popularity peak, the trend information could be integrated as part of the recommendation process, for instance, by considering it as part of the usage scores used by AppJoy or considering more complex dynamics models, for example, as part of latent factor models [25].

6.1 Application Potential

We have demonstrated the benefits of considering mobile app trend information for mobile analytics and app recommender systems. Another use for trend information is providing developers early feedback about the current popularity of their applications, which they can then use to take countermeasures against negative popularity fluctuations. Trend state can further be correlated with other factors, such as usability gathered through interaction metrics [37], to provide more detailed feedback of the possible reasons in popularity fluctuations.

Beyond providing app developers with tools to understand the state of their app, trend can also be used for marketplace analytics to support advertising strategies. On the device side, trend status can be used as an additional metric to identify most redundant applications for removal to reduce clutter on the user interface. The app-filter also enables detecting apps that are rapidly gaining in popularity, which could be used, for instance, by in-app advertisers to entice new app developers as customers or for dynamic pricing models.

Application trends are also potentially a powerful source of information for characterizing and understanding user interactions, and trend information can be used to support user modeling. For example, users with consistently many *Hot* applications are continually shifting their application usage, whereas those with many *Expired* or *Marginal* apps are likely to remain faithful to the apps they originally chose.

7 RELATED WORK

In this section we discuss recent advances in trend mining and trend detection as well as mobile app recommendation systems and how our proposed technique exceeds this state-of-the art.

Trend prediction in numerical time series data is an important and well studied field in time series forecasting [18]. A time series is a series of, often real-valued numbers that occur over a discrete time. The analysis of time series traditionally assume a time-invariant generation function so that the time series shows a stationary behavior. To arrive at such behavior, a typical approach is to detect and remove trend and seasonal components from the time series [8]. Typical methods stem, for instance, from linear regression in order to model linear or polynomial trend behavior [9]. In addition, also statistical methods, such as moving average or splines techniques are employed [1].

In contrast to these approaches, the assumption of an underlying time-invariant generation function does not hold in our case as app popularity is conditioned on external factors and is also subject to aging. In addition, trend in time series analysis describes a linear, polynomial or exponential behavior whereas we are interested to describe the trend as a lifecycle of an app, potentially also reflecting past behavior, such as a steep rise followed by a drastic loss in popularity (*Expired* pattern). Therefore, we instead propose VAR, AUC, PEAK and SLOPE features to describe the shape of a usage or trend pattern.

Commercial trend mining systems include Google Trends[6], which monitors the frequency of words in search queries related to real-world events, and the trending topics list of Twitter, which uses the frequency of hashtags and noun expressions to determine popular topics. Related academic works include detecting emerging trends in real-time from Twitter [4, 10, 30], mining of news discussions or other text documents for trends [35, 39], and analysis of web behavior dynamics [36], which are all indirect and subjective measures that might be subject to fraud. Our work is capable of operating solely on app usage information whereas these works operate on co-frequency patterns between words or n-grams.

By exploiting actual usage (in contrast to downloads, likes, ratings and similar measures), our approach can identify and compare trends of apps regardless of their absolute user count, downloads or installations. In this way, well-known apps may show an inferior trend performance can be compared to less well known newcomers and, hence, exploiting our trend filter, promising future stars are potentially spotted earlier.

Mobile app recommendation systems utilise a multitude of features to rate the relevance or popularity of a respective app. Among these, user reviews are a prominent source for app recommendation systems [14]. However, empirical studies have shown that reviews typically contain several topics, which are seldom reflected by the overall rating [23, 32]. Motivated by these studies, several works on using sentiment analysis and summarization techniques for mining app reviews have been proposed. Chen et al. [12] identify reviews that are most informative to developers, whereas Guzman and Maalej [17] use sentiment analysis and language processing to extract user opinions for different features in a mobile app. Recommendations, however, are in general prone to fraud and are inaccurate and noisy as they are based on free-form textual descriptions. Our trend filter is not affected by such subjective and biased information as recommendations, as it capitalizes on actual usage trend. Jovian et al. point out that version information should further impact the recommendation score as the popularity of an app might be affected by the change in version [28]. We remark that the function of popularity conditioned on the version number is not necessarily monotonic or even increasing. Our trend detection approach, however, is able to detect such changes implicitly whenever changes in the usage trend result from a version update. In addition, Lim et al. point out that user behaviour is country specific [27], so that recommendation systems should adapt to such properties. Our proposed trend filter is also able to filter usage trends in a specific population, such as geospatial, age or gender.

In order to improve the above mentioned global solutions in which recommendations are identical for all users, individual preferences are considered. Peifeng et al. argue that an app recommendation system has to take into account also the set of already installed apps [46]. They compare a 'tempting' value of a new application to a 'satisfactory' value of already installed applications of the same type. Another example is AppJoy [45], which employs item-based collaborative filtering to recommend apps based on personalized usage patterns. AppBrain[7] compiles recommendations within the same category by monitoring the installation history of apps. Also, AppAware [15] provides recommendations

[6] **google.com/trends**
[7] **http://www.appbrain.com**

by integrating the context information of mobile devices. Such personalized recommendations are also possible exploiting our trend filter by applying it to a persoanlized subset of apps. Moreover, users often use several apps within the same category [47] and the overall usage session times tend to be short, and depend on a wide range of contextual factors [5, 13]. Other approaches consider, for instance, the user's privacy expectations on a given app-type for recommendation [29]. In addition, co-usage of apps can be exploited for app recommendation as detailed in [42]. Responding to this observation, the AppTrends approach was proposed to base the recommendation on frequency of co-usage of apps [2]. In contrast to our work, AppTrends does not exploit usage trends of individual apps but instead co-usage with other installed apps. Our trend filter could be applied in addition to further improve app recommendations. Indeed, combinations of these individual recommendation systems to form multi-objective app recommendations have the potential to further improve accuracy in the recommendations [44].

Also, Petsas et al. [33] demonstrated that user preferences tend to be highly clustered and following various trends over time, with users showing interest in a small set of app categories at a time. Our work complements existing solutions by providing mechanisms for analysing and understanding application usage relative to the dynamics of the app's instantaneous popularity in a marketplace.

Some commercial app analytics tools, such as Google Mobile Analytics[8] and Countly's Mobile Analytics[9], follow a similar path by focusing on monitoring statistics of individual apps, covering information about usage session frequencies, lengths of usage sessions, extent of in-app purchases, and so forth. In contrast to our work, these solutions do not consider popularity of apps conditioned on whether they follow specific trend patterns.

The importance of app popularity has recently been reported also in [48]. The authors proposed a HMM-structure in order to model and predict app popularity. In contrast to our work, the authors exploit temporal observations of rankings, user ratings and user reviews, rather than actual app-usage statistics and thereby directly accesses app popularity.

8 SUMMARY AND CONCLUSION

We have presented the first ever independent study of retention rates in the wild. Our analysis shows that, on average, applications lose 65% of their users in the first week, but the effect is mediated by overall user count as applications with over $1,000$ users show much higher retention rates. We also demonstrated that, contrary to reports in the literature, severe losses in usage are rare, with less than 10% of apps losing over 80% of users in the first week. We demonstrate that retention rates are an insufficient metric of an application's success as they ignore effects of seasonality and external factors. In particular, we demonstrated that applications follow different trend patterns which are not captured by retention.

As second contribution, we proposed a novel app-filter that can categorize applications according to their currently followed usage trend. We focused on four characteristic trends: *Marginal* apps with only few users, *Dominant* applications of permanent high popularity, *Hot* apps with rapidly increasing popularity, and *Expired* apps with drastic drop of the usage. We observed that about 40% of the apps are *Marginal*. We analyzed application categories from Google Play and show that, for example, during the year 2014, 7.5% of communication apps have been *Hot*, only 0.1% were *Dominant*, and almost 2% were *Expired* apps. This kind of analysis can, in the future, lead application developers to follow needs and desires of the users in faster pace.

As a practical use case of our work, we considered how our trend-filter can benefit mobile app recommenders by enabling recommendations to focus on those apps that are rising in popularity. Trend pattern analysis can be used to

[8] `http://www.google.com/analytics/mobile`
[9] `http://www.count.ly`

strengthen existing heuristics such as interaction rate, download counts, and reviews, and even give more direct way to produce in the wild recommendations taking into account the usage history and trend-pattern of the application. Our analysis shows that only 3.6% of the recommendations are for apps which are currently rising in popularity, and that overall recommendations have low novelty and temporal diversity. We also demonstrate that the accuracy of the recommendations can be improved by considering trend information.

Another prominent issue in recommender systems is the cold start problem when insufficient data has been collected on a new user. The trend filter is not affected by this problem with regard to new users as usage trends are build by other users. However, we remark that for a new application, trend estimation is volatile over the first days as usage data first has to be collected first. Although exploiting trend filters, apps of smaller user base are empowered to content with highly popular apps, note that the potential risk of recommending badly maintained apps is low as such apps likely boast a less satisfied user population and hence feature an inferior usage trend pattern compared well maintained apps with a satisfied user population. Summarizing, using our proposed trend filter has the potential to increase the success probability in finding good and reliable apps and that poorly maintained apps will not feature a positive trend for long.

REFERENCES

[1] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE transactions on automatic control* 19, 6 (1974), 716–723.

[2] Donghwan Bae, Keejun Han, Juneyoung Park, and Mun Y Yi. 2015. AppTrends: A graph-based mobile app recommendation system using usage history. In *Big Data and Smart Computing (BigComp), 2015 International Conference on*. IEEE, 210–216.

[3] Ricardo Baeza-Yates, Di Jiang, Fabrizio Silvestri, and Beverly Harrison. 2015. Predicting The Next App That You Are Going To Use. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. ACM, New York, NY, USA, 285–294.

[4] James Benhardus and Jugal Kalita. 2013. Streaming trend detection in Twitter. *International Journal on Web Based Communities* 9 (2013), 122 – 139.

[5] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 47–56.

[6] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*. ACM, 47–56.

[7] Matthias Böhmer and Antonio Krüger. 2013. A Study on Icon Arrangement by Smartphone Users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2137–2146.

[8] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[9] Peter J Brockwell and Richard A Davis. 2013. *Time series: theory and methods*. Springer Science & Business Media.

[10] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. 2010. Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*.

[11] Rishi Chandy and Haijie Gu. 2012. Identifying Spam in the iOS App Store. In *Proceedings of the 2Nd Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality '12)*. ACM, New York, NY, USA, 56–59.

[12] Ning Chen, Jialiu Lin, Steven C. H. Hoi, Xiaokui Xiao, and Boshen Zhang. 2014. AR-miner: Mining Informative Reviews for Developers from Mobile App Marketplace. In *Proceedings of the 36th International Conference on Software Engineering (ICSE)*.

[13] Mark de Reuver, Harry Bouwman, Nico Heerschap, and Hannu Verkasalo. 2012. Smartphone Measurement: do People Use Mobile Applications as they Say they do?. In *Proc. International Conference on Mobile Business (ICMB)*.

[14] Bin Fu, Jialiu Lin, Lei Li, Christos Faloutsos, Jason I. Hong, and Norman M. Sadeh. 2013. Why people hate your app: making sense of user feedback in a mobile app store. In *Proceedings of The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[15] Andrea Girardello and Florian Michahelles. 2010. AppAware: Which Mobile Applications Are Hot?. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 431–434.

[16] Alessandra Gorla, Ilaria Tavecchia, Florian Gross, and Andreas Zeller. 2014. Checking App Behavior Against App Descriptions. In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. ACM, New York, NY, USA, 1025–1035.

[17] Emitza Guzman and Walid Maalej. 2014. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. In *Proceedings of the IEEE International Requirements Engineering Conference (RE)*.

[18] James Douglas Hamilton. 1994. *Time series analysis*. Vol. 2. Princeton university press Princeton.

[19] Mark Harman, Yue Jia, and Yuanyuan Zhang. 2012. App Store Mining and Analysis: MSR for App Stores. In *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories (MSR '12)*. IEEE Press, Piscataway, NJ, USA, 108–111.

[20] Xin Jin and Jiawei Han. 2010. *K-Means Clustering*. Springer US, Boston, MA, 563–564.

[21] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*. ACM, New York, NY, USA, 219–230.

[22] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 2012. Climbing the App Wall: Enabling Mobile App Discovery Through Context-aware Recommendations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. ACM, New York, NY, USA, 2527–2530.

[23] Hammad Khalid, Emad Shihab, Meiyappan Nagappan, and Ahmed E. Hassan. 2015. What Do Mobile App Users Complain About? *IEEE Software* 32 (2015), 70–77.

[24] Hee-Woong Kim, Hyun Lyung Lee, and Jung Eun Son. 2011. An exploratory study on the determinants of smartphone app purchase. In *The 11th International DSI and the 16th APDSI Joint Meeting, Taipei, Taiwan*.

[25] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

[26] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal Diversity in Recommender Systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. ACM, New York, NY, USA, 210–217.

[27] S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden. 2015. Investigating Country Differences in Mobile App User Behavior and Challenges for Software Engineering. *IEEE Transactions on Software Engineering* 41, 1 (Jan 2015), 40–64. DOI:**http://dx.doi.org/10.1109/TSE.2014.2360674**

[28] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2014. New and improved: modeling versions to improve app recommendation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 647–656.

[29] Rui Liu, Jiannong Cao, Kehuan Zhang, Wenyu Gao, Lei Yang, Junbin Liang, and Ruiyun Yu. 2016. Understanding Mobile Users' Privacy Expectations: A Recommendation-based Method through Crowdsourcing. *IEEE Transactions on Services Computing* (2016).

[30] Michael Mathioudakis and Nick Koudas. 2010. TwitterMonitor: Trend Detection over the Twitter Stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*.

[31] Adam J. Oliner, Anand P. Iyer, Ion Stoica, Eemil Lagerspetz, and Sasu Tarkoma. 2013. Carat: Collaborative Energy Diagnosis for Mobile Devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, Article 10, 10:1–10:14 pages.

[32] Dennis Pagano and Walid Maalej. 2013. User feedback in the appstore: An empirical study. In *Proceedings of the 21st IEEE International Requirements Engineering Conference*.

[33] Thanasis Petsas, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, and Thomas Karagiannis. 2013. Rise of the Planet of the Apps: A Systematic Study of the Mobile App Ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. ACM, New York, NY, USA, 277–290.

[34] P. A. Pevzner. 2000. *Computational molecular biology – An algorithmic approach*. MIT Press.

[35] Alexandrin Popescul, Gary William Flake, Steve Lawrence, Lyle H Ungar, and C Lee Giles. 2000. Clustering and identifying temporal trends in document databases. In *Advances in Digital Libraries, 2000. Proceedings. IEEE*. IEEE, 173–182.

[36] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. 2012. Modeling and Predicting Behavioral Dynamics on the Web. In *Proceedings of the 21st International Conference on World Wide Web*.

[37] Lenin Ravindranath, Jitendra Padhye, Sharad Agarwal, Ratul Mahajan, Ian Obermiller, and Shahin Shayandeh. 2012. AppInsight: Mobile App Performance Monitoring in the Wild.. In *OSDI*, Vol. 12. 107–120.

[38] Kent Shi and Kamal Ali. 2012. GetJar Mobile Application Recommendations with Very Sparse Datasets. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 204–212.

[39] Olga Streibel and Rehab Alnemr. 2011. Trend-based and Reputation-versed Personalized News Network. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents (SMUC '11)*. ACM, New York, NY, USA, 3–10.

[40] Hien Truong, Eemil Lagerspetz, Petteri Nurmi, Adam Oliner, Sasu Tarkoma, and N. Asokan. 2014. The company you keep: mobile malware infection rates and inexpensive risk indicators. In *Proceedings of the 23rd international conference on World wide web (WWW)*. ACM, 39 – 50.

[41] Sergei Vassilvitskii. 2007. *K-means: Algorithms, Analyses, Experiments*. Ph.D. Dissertation. Stanford, CA, USA. Advisor(s) Motwani, Rajeev. AAI3281968.

[42] Fei Wang, Zhe Zhang, Hailong Sun, Richong Zhang, and Xudong Liu. 2013. A cooperation based metric for mobile applications recommendation. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, Vol. 3. IEEE, 13–16.

[43] T. Wang, D. Wu, J. Zhang, M. Chen, and Y. Zhou. 2016. Measuring and Analyzing Third-Party Mobile Game App Stores in China. *IEEE Transactions on Network and Service Management* 13, 4 (Dec 2016), 793–805.

[44] Xiao Xia, Xiaodong Wang, Jian Li, and Xingming Zhou. 2014. Multi-objective mobile app recommendation: A system-level collaboration approach. *Computers & Electrical Engineering* 40, 1 (2014), 203–215.

[45] Bo Yan and Guanling Chen. 2011. AppJoy: Personalized Mobile Application Discovery. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*. ACM, New York, NY, USA, 113–126.

[46] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. 2013. App recommendation: a contest between satisfaction and temptation. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 395–404.

[47] Nan Zhong and Florian Michahelles. 2013. Google Play is Not a Long Tail Market: An Empirical Analysis of App Adoption on the Google Play App Market. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*.

[48] Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, and Enhong Chen. 2015. Popularity modeling for mobile apps: A sequential approach. *IEEE transactions on cybernetics* 45, 7 (2015), 1303–1314.

Research Theme B: Mobile Application Usage

Research Paper IV

Ella Peltonen, Eemil Lagerspetz, Jonatan Hamberg, Abhinav Mehrotra, Mirco Musolesi, Petteri Nurmi, and Sasu Tarkoma

The Hidden Image of Mobile Usage: Uncovering the Impact of Geographic and Demographic Factors

Manuscript under submission.

Contribution: The publication started in collaboration between the author and researchers at University College London, Dr. Mirco Musolesi and Dr. Abhinav Mehrotra. Most of the ideas that lead to the publication were delivered through the author's research visit to University College London. The author was in the lead of the data analysis work, planning the additional data gathering, such as the user background questionnaires, and constructing the publication. Jonatan Hamberg and Dr Eemil Lagerspetz contributed significantly to the implementation of the questionnaire and data collection system, and together with Dr Petteri Nurmi and Prof. Sasu Tarkoma, they participated by sharing ideas and in the writing process.

The Hidden Image of Mobile Usage: Uncovering the Impact of Geographic and Demographic Factors

ELLA PELTONEN, EEMIL LAGERSPETZ, and JONATAN HAMBERG, University of Helsinki, Finland
ABHINAV MEHROTRA, University College London, United Kingdom
MIRCO MUSOLESI, University College London and The Alan Turing Institute, United Kingdom
PETTERI NURMI and SASU TARKOMA, University of Helsinki and Helsinki Institute for Information Technology HIIT, Finland

Mobile applications have become an integral part of everyday life with popular marketplaces offering millions of them for practically any purpose. Despite the prevalence of apps, the factors governing application usage have thus far been understudied. Indeed, questions such as how app usage differs across countries, and how these differences reflect demographic factors and cultural values have not been investigated. Answering these questions is not only of academic interest, but provides essential information to support app developers, social scientists, and other people involved in the mobile ecosystem. In this paper, we study and answer these questions through a large-scale analysis of app usage data from 25, 323 Android users from 44 countries and 54, 776 apps in 55 categories. We demonstrate that there are significant differences in usage of different app categories across countries, and that these differences, to a large extent, reflect geographic boundaries, and correlate with demographic factors and cultural values. We compare mobile usage to the value survey model, a widely used model for comparing cross-cultural differences, and demonstrate that mobile usage is a factor that helps to understand and analyze contemporary everyday life. Our technical contributions focus on development of methods for identifying boundaries across different countries and regions based on mobile application use.

CCS Concepts: • **Social and professional topics** → **Cultural characteristics**; • **Human-centered computing** → *User studies*;

Additional Key Words and Phrases: Mobile Applications; Usage modeling; Cultural Factors

ACM Reference Format:
Ella Peltonen, Eemil Lagerspetz, Jonatan Hamberg, Abhinav Mehrotra, Mirco Musolesi, Petteri Nurmi, and Sasu Tarkoma. 2018. The Hidden Image of Mobile Usage: Uncovering the Impact of Geographic and Demographic Factors. 1, 1 (January 2018), 29 pages. https://doi.org/0000001.0000001

Authors' addresses: Ella Peltonen; Eemil Lagerspetz; Jonatan Hamberg, University of Helsinki, P.O. 68, Helsinki, FI-00014, Finland; Abhinav Mehrotra, University College London, Gower Street WC1E 6BT, London, United Kingdom; Mirco Musolesi, University College London and The Alan Turing Institute, Gower Street WC1E 6BT, London, United Kingdom; Petteri Nurmi; Sasu Tarkoma, University of Helsinki and Helsinki Institute for Information Technology HIIT, P.O. 68, Helsinki, FI-00014, Finland.

1 INTRODUCTION

Modern societies have assigned an important role to technology in everyday life, as demonstrated, for example, in market success of hand-held devices[1]. Different applications can support well-being, education, and leisure time, resulting in smartphones and tablets (partially) replacing multiple single-purpose devices, such as small pocket cameras, gaming consoles, maps, and navigators. There are plenty of applications available on the market, i.e., 2.2 million applications in the Google Play store and 2 million in AppStore[2]. Interestingly, thanks to wide pricing options, these devices are popular in many countries and afforded by a large percentage of the population. This opens new possibilities to study people's mobile usage in their everyday life all over the world. In our work, we study how usage of mobile devices - communication, games, or in other terms, applications - reflects geographic and demographic factors, in addition to the personal choices and preferences. To obtain further insights into the potential role of geographic factors, we also study the relationship between application usage and cultural values using the value survey model of Hofstede [14], an established and widely used model of cross-cultural differences. In our work, we utilize large-scale application usage data collected by 25,323 Android users from 44 countries distributed in Asia, Europe, Americas, and Oceania[3]. To avoid language and marketing biases, we consider applications through the categories they belong to, such as communication, social apps, and different game genres. Our approach extends the typical characteristics of application usage to the evaluation of the demographic and geographic factors and cultural values behind application choices. We show that there is a strong connection between geographic factors and application usage, and that this connection is only partially explained by demographic differences. To complement these observations, we explore usage differences within demographic groups in the form of case studies.

The information uncovered by our studies is beneficial for several reasons. It can be used by researchers, e.g., in social sciences and mobile computing to understand similarities and differences in app usage across geographic areas. Our work is also relevant to several practical applications, such as can be used to enrich application recommendation systems and better tailor application markets for their users. We note that some prevalent application markets, such as the Google Play store can already provide naive recommendations based on user location and other unambiguous attributes. However, the extent to which these systems can effectively utilize the spectrum of demographic, geographic, and value differences in application usage remains unclear. Thus, we consider ours as the first independent study in the area.

The dataset we use in this study has been originally created for technical analysis of mobile devices and, as such, imposes a variety of challenges related to extraction of demographic features from machine oriented data. Due to the limitations in our data, we also run a volunteer user questionnaire study that focuses on user demographics and values, and obtain responses from 3,293 participants.

To summarize, the main contributions of our paper are the following:

(1) We analyze relationship between mobile application usage on category level, and geographic and demographic factors. We demonstrate that statistically significant relationships can be found with country impacting app usage. We also demonstrate that, within the 44 countries considered in our analysis, clear geographic and cultural clusters can be identified based on differences in app usage.

[1]Newzoo ranked top 50 countries by the number of smartphone users, with average smartphone penetration of 39.4% or total 2.4 bn smartphone users https://newzoo.com/insights/rankings/top-50-countries-by-smartphone-penetration-and-users/.
[2]http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/
[3]We have notable exceptions such as China, because of the different app market structure, and Africa, due to lack of a significant number of installations. Please see the Appendix for a list of countries considered in our work.

(2) Based on our methodology, we show that mobile application category usage constitutes a unique societal factor that contrasts and complements the Hofstede's Cultural Values Model (VSM) [14].
(3) Comparing the information gain from different geographic and demographic attributes, such as country, age and education, we can determine which attributes reveal the most information about application use. We show that, indeed, the application usage reflects the country of an individual.
(4) As a case study, we analyze application usage in 44 countries and derive three main clusters out of them. We also compare countries together with information of different demographic features, such as occupation and educational background.

2 RELATED WORK

Academic research on analyzing app usage has predominantly focused on characterizing dominant usage patterns and the contexts where usage occurs without examining demographic and geographic factors or cultural values influencing it. Especially the effect of the country information can be seen as understudied. These studies thus are essential for understanding how individuals use apps, but provide no insights about the collective dynamics of the usage. For instance, Xu et al. [37] analyze network traffic caused by apps. The authors find app usage to follow diurnal patterns, as well as to be dependent on spatial context. In a related study, Verkasalo [34] show location to have significant correlation with app usage. Falaki et al. [8] study installation and usage patterns, showing that the number of apps installed and used by users contains significant variation. Böhmer et al. [4] also demonstrate strong diurnal variations in app use. They show that usage session times tend to be short, and that they depend on contextual factors. Hintze et al. [13] report average of 60 interactions with a smartphone during a day, lasting 107 seconds on average and 57 seconds as a median. Ferreira et. al. [9] investigate characteristics of short-term usage sessions, finding social and spatial context to have strong influence on application usage, in addition to application functionality.

Analyzing application installation patterns using data gathered from app marketplaces has been an active research area. Petsas et al. [25] demonstrate that user preferences are highly clustered, and that users generally show interest in a small set of app categories at a time. In a related study, Zhao et al. [38] demonstrate that clusters with salient features can be extracted and that specific user demographics can be associated to each cluster. Examples of clusters include "evening learners", "young parents", and "night communicators". The work by Lim et al. [20] is probably the closest to ours: in their study the authors analyze app download decisions across countries, finding the importance of pricing, reviews, and app descriptions to vary across countries. Our work extends these studies by comparing app usage between countries instead of a single market or administrative area. In addition to this, we focus on geographic differences and societal and economical demographic factors, such as education, occupation, and household status of smartphone users.

Our work is also related to previous projects that consider apps as a sensor. Most of them have focused on analyzing device or user specific patterns, for example, identifying potential malware infections [33], energy issues [23], or identifying the current user of the device [36]. Beyond these projects, recently some papers have used apps to identify cognitive or personal states. For example, Seneviratne et al. [30] demonstrate that app usage can reveal user's gender and age. In addition to this, app usage patterns can be exploited to identify mood [17–19], personality [5], and certain mental conditions [11]. Hiniker et al. [12] show that app usage reflects both instrumental (for some purpose) and ritualistic (more habitual) behavior. We complement the findings of these papers by demonstrating that app usage also strongly reflects geographical patterns.

Indeed, the focus of our work is on uncovering regional dynamics from data collected on mobile devices. To the best of our knowledge, our work is the first to study mobile app usage across countries from around the world with the goal of understanding the underlying demographic and geographic factors. Existing projects of this area have predominantly focused either on analyzing mobility patterns extracted from cellular data records (CDR) obtained through network operators or information acquired through location-based social media. For example, Silva et al. [32] uncover geographic differences from FourSquare check-ins to restaurants. Instead, Kendall et al. [10] study cultural effects of social media on consumer decision-making. They show that information sources that influence online purchase decisions strongly varies by culture. Reinecke et al. [28] study usage of the Doodle scheduling software worldwide and present differences in response times. Qiu et al. [27] study usage of Facebook and a Chinese app with similar functionality called Renren, and find cultural differences, naming the Renren community more collectivist. Kang et al. [15] study mobile usage differences in the USA and South Korea, but limit their analysis in these two countries. In our paper, we analyze application usage in 44 different countries distributed around the world.

3 MOBILE USAGE DATASETS

We investigate how application category usage reflects geographic and demographic factors among Android smartphone users. We show how usage of mobile application categories is associated to different countries and areas. Straightforwardly, location can have a strong impact on usage of certain applications. For example, public transportation apps focused on a city area are hardly ever used outside the city borders. Observing categories, we are able to compare *general usage* of a given type of application, for example, transportation apps in general. This allows us to compare countries and study general cultural perspectives instead of particular applications used in a specific city or region.

In this study, we consider different sources of data summarized in Table 1. Mobile application data collected from Android applications from 44 countries is described in detail in Section 3.1. In addition, we map these Android applications to their corresponding Google Play categories (currently 55 different categories). Based on these data sources, we determine how much applications belonging to each category have been used in each country, and we can correlate their usage with the demographic factors and the cultural value model. There are also notable differences between the mobile usage behavior, demographic factors, and the cultural values model described in Section 4.2. Section 3.2 describes the background questionnaire used to take demographic features into account, and Section 3.3 discusses the nature of our data and questionnaire bias. Finally, the set of cultural values from 111 countries, determined by Hofstede's Cultural Values Model (VSM) [14], is described in Section 3.4.

3.1 Carat Application Data

Since 2012, the Carat application [23, 24] has been used to collect mobile usage data from Android and iOS devices from multiple different countries around the globe. The data collection includes applications, system settings, and subsystem variables such as CPU usage and battery level. Because some of the features have been included in the system later than others, information available from specific years can vary. The entire Carat data at the moment of writing has 864,079 distinct user records.

The data collected by the Carat application includes various features from the device, such as package names of the applications currently running in the device. Originally designed for energy consumption research, Carat takes a sample every time 1% of battery has been drained. In addition to application names, we collect the following features that are relevant for this work: user specific identifier (referred as the user's Carat id), timestamp, device model, operating system version code, time zone, and mobile country code (MCC). The newest addition is the mobile country code, which

Data set	Attributes	Date(s)	Size
Carat mobile data set	user id, applications, timestamp, time zone	March 2016 to April 2017	25,323 users
Background questionnaire	user id, gender, age group, current occupation, highest completed education, household situation, yearly income, debt, savings, current location	June 2016 to May 2017	3,293 users
Google Play categories	1-2 categories for each application	October 2016	54,776 applications
Cultural Values Model (VSM)	6 cultural factors: power distribution, individualism versus collectivism, masculinity versus femininity, uncertainty avoidance, long versus short-term orientation, indulgence versus restraint	2015, downloaded September 2016	111 countries
VSM questionnaire	VSM questionnaire (24 items) presented to Carat users	June 2016 to May 2017	634 users

Table 1. Summary of data sources of this work.

has been collected since March 2016, from where the period of data in this study starts and continues until the end of April 2017.

For large-scale comparison of application usage in different countries, we consider a subset of 5.65 million samples collected by the Android application, in which the time zone and MCC fields match. The MCC is obtained from the cellular network infrastructure, and automatically converted to a two-character country code. We compare MCC with the country that the city of the time zone field corresponds to. This procedure increases the reliability of detecting the country of the user, when the exact GPS or Wi-Fi based location is not available for privacy reasons. We will discuss details of this implementation in Section 4.2. The subset contains 25,323 users associated with 114 country codes, from which 44 countries have a significant number of users (100 or more). Figure 1 shows how users are distributed over the countries. The majority are based in the USA, with strong user bases also in Finland, India, Germany, and the United Kingdom among others.

To obtain categorization of applications, we fetch the application descriptions of all the applications in the Carat dataset as HTML files from the Google Play store, and map application names to the corresponding categories. This way each user's category usage can be detected. In October 2016, there were 55 categories on Google Play. Our dataset contains 97,000 different applications including system processes, from which 54,776 applications are available from Google Play with at least one category assigned. Some apps have multiple categories, such as family oriented action games may belong to categories *Family pretend* and *Action games*.

An example of differences in application category usage between countries is presented in Figure 2a, which shows usage distributions of some categories from selected countries: Finland, the United Kingdom, Germany, the USA, Canada, Mexico, Brazil, Columbia, Australia, India, Japan, Korea, and Saudi-Arabia. For example, the popularity of education apps varies between countries, which may be a result of their availability in different languages. We also observe that lifestyle apps are highly popular in South Korea, Australia, and North America. The *Finance* category as well as *Maps and Navigation* are used often in English-speaking countries, Brazil, Japan and South Korea. Compared to Columbia,

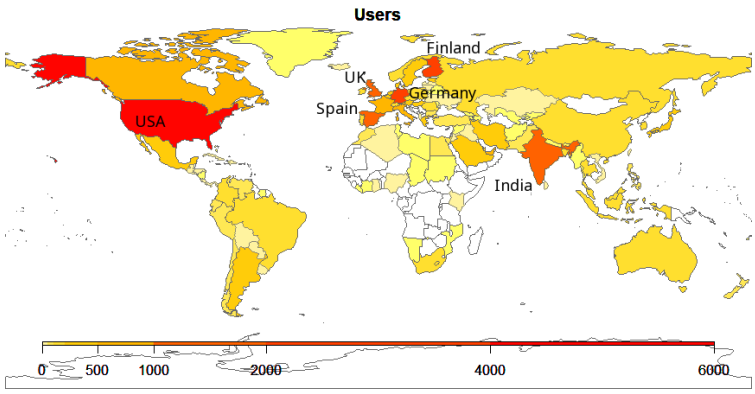


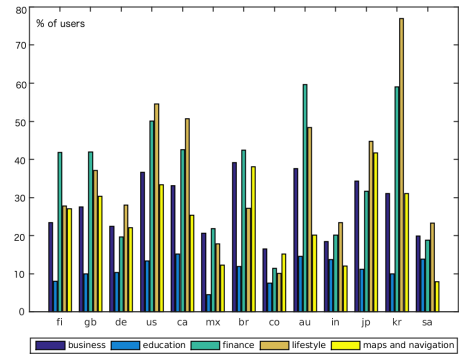
Figure 1. User distribution of the mobile usage data. The colors scale indicates number of users.

Finance apps are three or four times more popular in Brazil, even though the countries are geographically closely located.

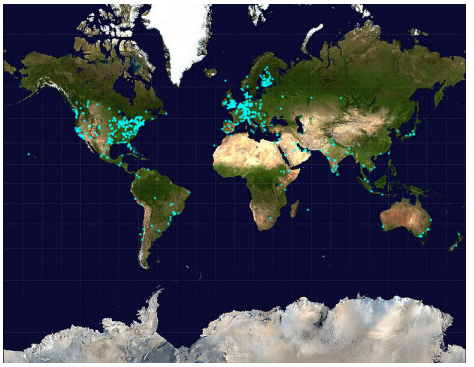
3.2 User Demographic Questionnaire

We have conducted a questionnaire within the Carat app, and it has received 3, 293 responses from individuals in 44 countries. Because of the strict privacy policy of the application, the Carat dataset has some limitations, for example, location information is available only at country level. Also, there is no personal information in the dataset. To obtain demographic information, we sent a voluntary questionnaire to all active Carat users. We are able to link their answers to their application usage through their Carat id, a unique hash code generated automatically for each user. Questions include basic background information and socio-economic status. We also collect the current location of a user with their permission. Only adults have been able to answer the questionnaire, defined as over 18 years old people. The questionnaire includes the following questions (single choice):

(1) Gender: female, male, or other;
(2) Age group: 18-24, 25-34, 35-44, 45-64, or over 65 years old;
(3) Current occupation: manager, professional, technician or associate professional, clerical support, sales or services, agricultural or forestry or fishery, craft and trade or plant and machine operations, entrepreneur or freelancer, student, staying at home, retired, or no suitable option;
(4) Highest completed education: elementary school or basic education, high school or sixth form or other upper secondary level, vocational school or trade school or other education leading to a profession, undergraduate or lower university degree (Bachelor's or equivalent), professional graduate degree or higher university degree (Master's or equivalent), research graduate degree (PhD or equivalent);
(5) Household situation: living alone, living with other adult(s), living alone with under-aged kid(s) (under 18 years old), living with other adult(s) and kid(s);
(6) Yearly income, compared to the user's country average: much lower, lower, about the same, higher, or much higher;



(a)



(b)

Figure 2. Participants visualized by application category use and location. (a) Category usage varies widely in different countries. (b) Users' GPS coordinates are distributed around the world, with the US and Europe better covered.

(7) Debt, as percentage of monthly income need to cover it: no debt, or 10%, 25%, 50%, or most of the income;
(8) Savings, as a number of months possible to live off it: less than a month, 1-3 months, 4-6 months, 7-12 months, or over a year;
(9) Current coarse location, if user agrees to measure it: yes or no, measured automatically if agreed.

We have received a total of 3,293 individual answers. This corresponds to 14.3% of active Carat users that have the latest Carat version and thus the questionnaires available. Comparing the results to a prior questionnaire from 2013 [1], we find out that the demographic distributions are similar with the exception of user locations, where the peak had shifted from United States to Finland. This can be explained considering the fact that the Carat research project has been transferred from University of California, Berkeley, USA, to University of Helsinki, Finland, during the time between the two user studies.

Out of all respondents, 1715 agreed to share their location and 1153 valid GPS coordinates have been collected. Figure 2b shows these locations in a world map. As Carat is not officially available in Chinese app stores, we have no location points there, even though some users have been able to sideload the application through unofficial distribution channels; see also the Discussion section.

In addition to the demographic questionnaire, a 24-item Value Survey Model questionnaire (described in Section 3.4) was also presented to Carat users who had already answered the demographic questionnaire. In total, 634 users answered this questionnaire. Thus, these users form a subset of the total 3,293 users.

Ethical Considerations. We only consider aggregate-level data which contains no personally identifiable information. The privacy protection mechanisms of Carat are detailed in [23]. Data collection by Carat is subject to the IRB process of University of California, Berkeley. Users of Carat are informed about the collected data and give their consent from their devices. Collection of location information and the user questionnaire performed for this work have been approved on 14 June 2016 by the IRB process of University of Helsinki, Finland. Participation in the study have been voluntary and the users have been informed about the data collection and management procedures.

3.3 Representativeness of Mobile Usage Data

Different countries are well represented in our questionnaire. All age groups are represented. However, respondents are mostly male and biased towards higher education and professionals. The Carat users are located to 44 countries in Asia, Europe, Americas, and Oceania. Notable limitations include the lack of African countries and China due to low user counts. The full list of countries is given in the Appendix at the end of the paper.

Originally designed for diagnosing the energy consumption of smartphone applications, the Carat application performs sampling every time the battery has been drained 1%, which can mean a sampling interval from minutes to hours depending on battery drain, Android's Doze, device sleeping, and so on. For user interaction research, this makes the data sparse. As a result, some running applications are missed, even if they have an impact on energy use. Statistically, the longer a user uses Carat, the higher the chance that more running apps are captured. For this reason, we consider only users with 100 samples or more. On the other hand, many applications use or support other apps by background services that can be active for longer than the foreground UI of the application. Carat also captures those applications.

Carat can record updates, removals, and new installs of applications. Many off-the-shelf devices come with pre-installed applications, such as common messengers and tools. In this study, we only use applications that have been in the running state in order to better mirror the set of applications with confirmed interactions. This way any pre-installed but never opened applications will not be considered. Applications that are currently under development are filtered out by choosing only those that can be found from the Google Play store.

From the user's perspective, Carat is an energy profiling application, and thus it is possible that mainly people with energy problems download it. This can cause bias in the user population as well as in the device population. Nevertheless, the data provides a unique insight to the mobile usage throughout a large population of users in many different countries.

In the background questionnaire, 10% of answers come from female and around 87% from men, which causes a noteworthy gender bias in the results. On the other hand, user questionnaires performed by mobile apps has been reported to have high gender biases before [2]. When studying countries, the application data is managed as aggregates, not as individual users.

In terms of occupations, the most represented are professionals (34%), technicians or associate professionals (14%), students (12%), and managers (10%), so our questionnaire respondents are well employed. That may also reflect the general picture of owners of the mobile devices. Even if they have become much cheaper in present years, there may still be financial considerations in buying such a device. The distribution of education of the respondents reflects this, too: 35% have undergraduate degree, 30% have the Master's degree or equivalent, and 5% even have PhD or research graduate degree. 36% of the answers report their yearly salary is higher than their country's average and 7% that it is much higher. On the other hand, age groups are evenly distributed: 12% of age 18 – 24, 30% of age 25 – 34, 28% of age 35 – 44, 27% of age 46 – 64 and 4% 65 years or older.

3.4 Cultural Value Survey Model

Hofstede's Cultural Value Survey Model (VSM) is used in a wide variety of empirical research [16] to present cultural differences between countries. The VSM model has been previously used, for example, to study culture in IT corporations [26], evaluate tourist services [6], study international ethics [3], evaluate consumer decision making [10], Doodle scheduling responses [28], and model emoji usage in different countries [21]. The VSM model has also been criticized. For example, McSweeney [22] questions the validity of defining culture boundaries based on politically agreed national areas. The VSM model does not include minorities or subcultures, or take into account immigration and emigration in the global world, previously referred to as transnational mobility [35]. In our work we merely analyze the relationship between the VSM model, as an established and widely used model of cross-cultural differences, and mobile app usage without making claims about culture-specific differences across countries.

As a study of the country differences, VSM has been validated earlier on [29, 31]. In our study, we compare mobile usage to the VSM and analyze how application usage compares to this model. VSM consists of six factors that were gathered using questionnaire studies in different countries around the world. It includes questions about attitudes and appreciations people have in different countries. The questions cover topics about, for example, ideal job options, such as chances for promotion and attitude towards colleagues and superiors. Also, questions about private life preferences, such as desires, services, and general happiness, are considered.

The public version of the VSM dataset[4] consists of six cultural factors from 111 countries. The VSM dataset contains partial factors for Saudi-Arabia and Qatar, which are observed in our mobile usage dataset. For the rest of the countries in the Carat dataset (in Asia, Europe, Americas, and Oceania) the VSM dataset contains a full set of cultural factors. The six factors are defined as follows:

Power distribution (PDI) describes whether unequal power distributions are expected and accepted in the population. Cultures with higher power distribution tend to be more hierarchical and persist more inequalities compared to the cultures with lower power dimension.

Individualism versus collectivism (IDV) describes how much members of the population are supposed to care themselves or stay integrated to the group, such as family. In cultures with high individualism people define themselves as "I", compared to the stronger "we" feeling countries with lower individualism.

Masculinity versus femininity (MAS) describes strength of masculine and feminine roles in the population, for example, in work life. Cultures of high masculinity are more competitive, compared to lower masculinity (higher femininity) cultures that emphasize on collaboration and modesty.

[4] http://www.geerthofstede.nl/dimension-data-matrix

Factor name	Carat VSM Questionnaire	VSM Model
Power distribution (PDI)	0.55	0.91
Individualism versus collectivism (IDV)	0.77	0.85
Masculinity versus femininity (MAS)	0.79	0.94
Uncertainty avoidance (UAI)	0.91	0.95
Long versus short-term orientation (LTO)	0.94	0.91
Indulgence versus restraint (IVR)	0.61	0.86

Table 2. Comparison of the reliability test (the Cronbach's α) for 1) the VSM model, 2) the Carat-based VSM questionnaire.

Uncertainty avoidance (UAI) describes whether members of the population feel either comfortable or uncomfortable in new, unstructured, or unpredictable situations. High level of uncertainty avoidance implicates stricter codes of planning and caring for the future, compared to more relaxed cultures of the lower score of this factor.

Long versus short-term orientation (LTO) describes how members of the population accept delays in either social, material, or emotional gratification. Cultures with a high score of this factor emphasize planning for the future more than those that score lower.

Indulgence versus restraint (IVR) describes whether any gratifications are allowed to be relative free (having fun by themselves) or regulated by strict norms of the population. High indulgent score reflects higher importance of freedom leisure time compared to restraint cultures of lower indulgence score.

In addition to the general VSM model, we run the similar questionnaire to the Carat users and obtained 634 answers from 44 countries. In total, 20 countries have more than 10 users. Hofstede uses the Cronbach's α to test reliability of the factors among the countries. We follow this procedure and run the Cronbach's α of the VSM factors produced by the Carat user questionnaire answers. The results are presented in Table 2. To compare, also the Cronbach's α calculated for the published Hofstede's Cultural Value Survey Model (VSM) are included.

From Table 2 we note that four questionnaire-based factors gain the α larger than 0.7 which is generally considered a threshold for acceptable results [7]. In the case of factors *Uncertainty avoidance* and *Long versus short-term orientation* the Carat-based VSM questionnaire gains excellent reliability that is also in line with the original VSM model. On the other hand, the factors *Power distribution* and *Indulgence versus restraint* indicate only poor or questionable reliability, which might be caused by the small number of users answered to the Carat VSM questionnaire, and by a bias in the limited set of 20 countries present in our dataset. In general, these results help to present the representativeness of the Carat user data when comparing mobile application usage to the Hofstede's VSM model.

4 DECOMPOSING APPLICATION USAGE

In this section, we present an automatic methodology to detect categories for applications using the Google Play application store, and to detect the country from smartphone by time zone information and network country code without exact location permissions. We give a metric to compare application category usage of countries based on the Kullback-Leibler divergence.

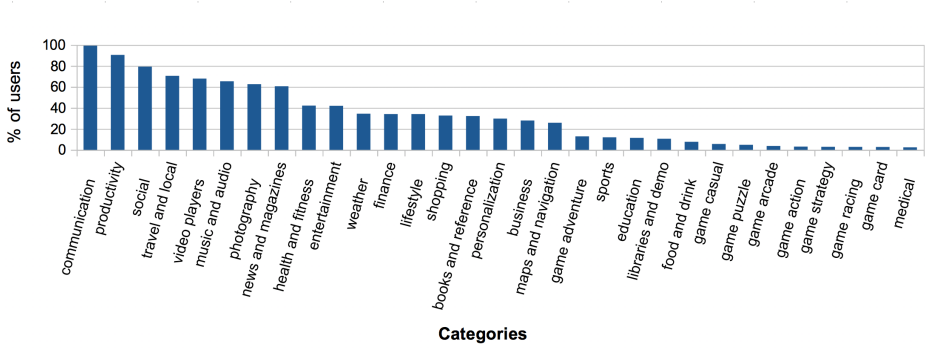


Figure 3. Distribution of users (%) between the top 31 (out of 55) Google Play categories.

4.1 Applications and Categories

While different applications are used in different countries, application categories such as one called *Travel and Local* and *Sports* are common between countries. Next, we show that the application category is more representative of smartphone usage across countries.

The dataset contains around 55, 000 applications. To get a generally representative picture of their functionality, we map them to the Google Play categories (55 were available at the time of writing). A distribution of the users who have downloaded at least one application from a certain category is presented in Figure 3. Every user has at least one application from category *Tools* and almost everyone uses *Communication*, a sign of basic functionalities of smartphones. Categories of *Productivity*, *Social*, and *Travel and local*, among others, are popular among the examined countries. In turn, certain games and minor categories have less users. Naturally, categories of most apps also gather most of the users.

For each user, we take all the applications running on the device, either foreground or background. Applications that are installed or pre-installed but never opened have not been considered. Then we map all the applications to their representative Google Play category. If the application belongs to two different categories, we consider it twice, once in each category. Next, we generate binary category vectors for each user considering whether that user has used a category or not. We detect the country of each user, a process described in detail in Section 4.2.

For each country, we construct the probability distribution of category usage within the country, represented by the fraction of users in the country having used that category. Formally, for each category $c_i \in C, C = c_1, c_2, ..., c_k$ where k is a number of categories, we define the probability of its use within a country n as

$$c_{i,n} = \frac{\sum_j u_{i,j} \in U_n}{|U_n|},$$

where U_n is the set of users in country n and $u_{i,j}$ is 1 if user j used category i and 0 otherwise.

Now $C_n = c_{1,n}, c_{2,n}, ..., c_{k,n}$, is the category use probability vector for country n. To compare these vectors with each other, we use the Kullback-Leibler divergence (KL). For two probability vectors it is defined as

$$KL(C_n||C_m) = \sum_{i=1}^{k} C_n(i) \log\left(\frac{C_n(i)}{C_m(i)}\right).$$

However, since the KL divergence is not symmetric and does not satisfy the triangle inequality, in our analysis we use the sum of two-way KL divergences as a distance metric, so that

$$dist(C_n, C_m) = \log\left(KL(C_n||C_m) + KL(C_m||C_n)\right).$$

4.2 Detecting Country

We use the mobile country code (MCC) and time zone of the smartphone to detect the country of the user. To protect user privacy, the Carat system does not collect any location information. Instead, Carat collects different information about the network used, especially Mobile Country Code (MCC) as well as the time zone. We show that using the MCC and the time zone we can detect the country of the user without exact location information.

A mobile country code (MCC) is a three-digit value tied to a mobile network. Each corresponds to a single two-letter IANA country code[5]. MCC is not available on WiFi-only devices and some CDMA networks, but from the beginning of March 2016 until May 2017, our dataset has 5.65 million samples with valid MCCs. For the time zone, Android devices follow the IANA time zone database format and give the time zones presented as the continent and the closest big city, for example, America/New_York or Europe/London. These values can be further translated to the two-letter country codes. There are 69.7 million samples with time zone information available in our dataset.

Both MCC and time zone based country codes (later referred to as CC) can sometimes have errors or they can be misconfigured. We compare the MCC and CC codes of samples and find that out of 5.83 million samples with valid MCC and CC values, the two indicate the same country in 97% of the samples. In those 3% of samples where MCC and CC indicate a different country, there are distinct neighbor countries such as small European states, and nearby countries in the same time zone. Difference in MCC and CC may be caused by cross-border usage of the network infrastructures in neighboring countries. Because the time zone can be changed by the user, there is a possibility of misleading selections, for example, both Europe/Athens and Europe/Helsinki represent the GMT+2 time zone but Athens is shown first in the alphabetical list, and can be chosen by users outside Greece for convenience.

The questionnaire responses contain GPS location coordinates (latitude and longitude) from 1153 users. We compare these locations to the user's sample history, take MCC codes from all these samples, and find that in the case of 97% of the users the coordinates match to the most common MCC among all samples. This means these people have been inside a single country for most of the time, and indicate that we can trust MCC as a country information source.

5 GEOGRAPHIC AND DEMOGRAPHIC FACTORS, AND CULTURAL VALUES IN MOBILE APP USAGE

We begin our analysis by investigating the *overall* effect of geographic and demographic factors, and cultural values. We consider 44 countries presented in the Carat mobile usage dataset (Sec. 3). First, we compare countries as their application category usage aggregates and find groups of countries with similar mobile usage. Second, we analyze demographic differences across countries, and find possible reasons behind them. Third, we compare the importance

[5]http://www.iana.org/time-zones

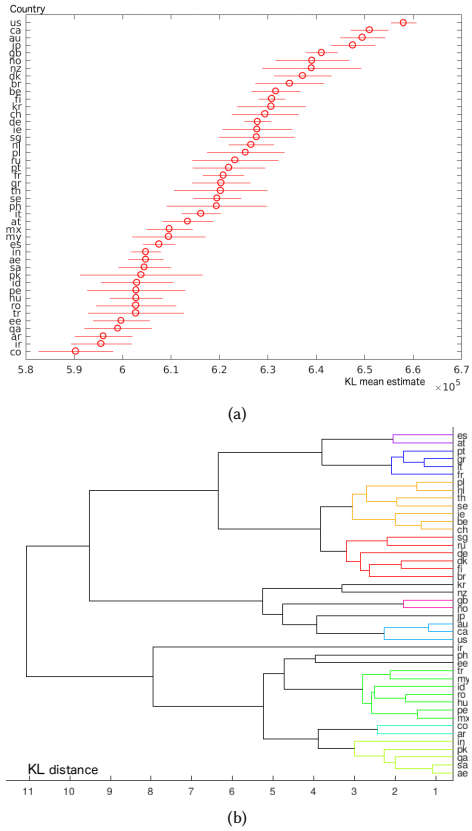


Figure 4. Application usage between different countries: (a) The multiple comparison of the country means and their standard errors by the Kruskal-Wallis test. (b) Visualization of KL divergence between countries. See the Appendix for the full names of countries.

of geographic and demographic factors by measuring the information they provide about app usage. Our results demonstrate that the country of the participant is almost twice as important than any of the demographic factors. Finally, we examine the influence of cultural values on app usage by comparing the value survey model (VSM) of Hofstede with differences in app usage in the Carat dataset. We show that differences in mobile usage partially explain cultural values, as explained by the VSM model, but also have important differences.

5.1 Application Category Usage Divides Countries Into Meaningful Groups

We first demonstrate that there are statistically significant differences between application category usage of different countries presented in the Carat dataset. To estimate this, we consider users' binary usage vectors in the corresponding country groups, and compare the distributions using a Kruskal-Wallis test, a nonparametric version of the classic

ANOVA test. We find that, indeed, there are significant differences in application category usage across countries ($p < .001$, $\chi^2 = 6792.4$, df = 40, $\eta^2 = 0.309$). Figure 4a shows post-hoc comparisons for the median estimates and their standard errors for each country. With significance level $\alpha = 0.05$ (using Tukey-Kramer correction), the medians of most countries are statistically significantly different, as can be observed from the small number of overlap for each country in the figure.

Besides differences, we are interested in understanding similarities between countries. To compare countries, we perform KL divergence analysis, as described in Section 4.1. Figure 4b illustrates the similarities between countries through a dendrogram that is created from the KL divergences. In the figure, the closer the branches are, the closer are the countries in terms of application usage. The topmost countries in the figure mostly consist of continental European countries, all relatively close to each other. Spain (es) and Austria (at) are presented together, and also close to some other Southern and Central European countries: Portugal (pt), Greece (gr), Italy (it), and France (fr). The next branch contains countries from Central and Northern Europe, including Poland (pl), Netherlands (nl), Sweden (se), Ireland (ie), Belgium (be), and Switzerland (ch). Thailand (th), a popular holiday destiny for Europeans joins to this group. The third sub-branch of this group also contains another Asian country, Singapore (sg), but also Russia (ru). The last there are Northern countries Denmark (dk) and Finland (fi) together with Germany (de) - and possible because of the language and connections with Portugal, also Brazil (br).

The next bigger branch in the middle of the dendrogram contains English-speaking countries such as the USA (us), Australia (au), Canada (ca), New Zealand (nz), the United Kingdom (gb), and other countries with early adopters of the Carat app, such as South Korea (kr) and Japan (jp). Norway (no) may be included because of its location near the United Kingdom. The latter three countries may also be included because the questionnaire has been only presented in English, so those familiar with English applications may have answered the questionnaire more readily than others.

The third main branch consists of the rest of the countries presented in the Carat dataset, with some meaningful cultural or geographical groups, such as Columbia (co) and Argentina (ar) in South America, and the Arab Emirates (ae), Saudi Arabia (sa), Qatar (qa), Pakistan (pk) and India (in) in Asia. Iran (ir), the Philippines (ph) and Estonia (ee) were not grouped close to other countries.

Our study shows that there are statistically significant differences between mobile application usage in different countries. These differences roughly follow the geographical differences, as we can see in Figure 4b. Later in Section 6.1 we present the use case where the main three groups of the dendrogram are considered in more detail and we discuss possible reasons behind this division.

5.2 Demographic Factors Are Related With Application Categories

Statistically, the country attribute gives us more information about the users' mobile behavior than the demographic features considered in this study, such as occupation, education, age group, gender, and certain household and socio-economic attributes. In the other direction, the usage of an application category can explain some demographic groupings, for example, *Parenting* and *Dating* applications help determine household status and *Weather* applications are used in some countries more than in others.

In this section, we study the combined impact of country and demographic factors on mobile usage. We separately analyze selected demographic factors in detail in the next section. To detect impact of different features on the mobile usage, we use the information gain metric known to work well with categorical data of mobile devices [24]. The information gain can be used to evaluate impact and contribution of any attribute considering the mobile usage. We sort users' application categories alphabetically and view them as a vector of ones and zeros, corresponding to the user

Attribute	Info gain
Country	4.60
Occupation	2.78
Education	2.14
Savings	2.12
Debt	1.99
Salary	1.96
Age	1.94
Household	1.57
Gender	0.59

Table 3. Demographic attributes sorted by information gain against application usage (3,293 users).

Attribute	App categories with highest information gain
Country	Weather, Game action, Finance, Family pretend
Occupation	Business, Game adventure, Finance, Family pretend
Education	Finance, Game adventure, Shopping, Music and audio
Savings	Game adventure, Game simulation, Entertainment, Personalization
Debt	Finance, Books and references, Game simulation, Family music video
Salary	Game adventure, Business, Game casual, Game simulation
Age	Game adventure, Weather, Business, Family music video
Household	Family music video, Dating, Family action, Parenting
Gender	Business, Game casual, Personalization, Books and reference

Table 4. Application categories that gain highest information against each demographic attribute (3,293 users).

having used that category, or not, respectively. Thus, we can detect information gain for each demographic factor and country against these category use vectors.

Table 3 presents attributes of the Carat background questionnaire (described in Section 3.2), sorted by the information gain. The dataset consists of 3,293 users that have answered the background questionnaire. We can see that the country attribute is characterized by the highest information gain compared to the other attributes, such as gender and age. Indeed, the effect of country is almost twice as high than other features, with socio-economic status being the second most important factor.

Table 4 shows which application categories best predict each attribute, separately, based on the information gain. For country, the category of the *Weather* applications gives the best information gain, probably because weather is more predictable in some countries than in others. Occupation is related, for example, to the *Business* and *Finance* applications, that is probably caused by academic and professional workers benefiting mobile techniques in their work. The household attribute that indicates whether the person is living alone, with other adults, or with kids, is best described by categories of family related applications, such as *Family music videos*, *Parenting*, and *Dating*, the last probably for those living alone. Different demographic factors indicating high usage of different categories shows that demographic factors have to be taken into account. Later we discuss their combined effect with country information in Section 6.2.

5.3 Association between Mobile Usage and Cultural Value Model

As we have shown, the country of the user has an impact on mobile application usage. Next, we study whether the differences are (at least partially) explained by cultural values by considering the relationship between app usage and the value survey model of Hofstede. We stress that we are not making claims about cultural factors directly influencing mobile usage, but merely investigating their relationship with app usage. Indeed, here we are considering a culture factor model and not trying to understand application usage with respect to the cultures of a country, which might be extremely complex and hard to define and capture quantitatively. We use the Value Survey Model (VSM), presented in Section 3.4, as reference dataset that has been used in previous studies to analyze country differences.

The VSM model consists of six numerical factors for each country as presented in Section 3.4. For each factor, we compute a difference matrix, and correlate them to the similar application usage matrix made by using KL divergence between countries, as presented in Section 4.1. Following the Mantel test procedure, we use the Pearson correlation coefficient between the VSM factor matrices and our mobile usage matrix using $100,000$ permutations of the order of countries. All the VSM factors have a 0.3 - 0.4 correlation coefficient with the category use matrix, indicating intermediate correlation.

Next, we correlate the use of all application category and VSM factor pairs separately. Table 5 summarizes the results, listing the categories with the highest positive and negative correlations for each VSM factor. For example, a low power distance that indicates low hierarchy in the culture, correlates significantly to the use of *Entertainment* applications and other leisure related categories, such as *Travel and Local*, *Sports*, and *Music and Audio*. These same categories together with, for example, *Health and Fitness* are mostly related to individualist cultures. Collectivist cultures, those with higher power distance, and cultures considered feminine seem to value family related categories, such as *Family create*, *Education games*, and *Family pretend*. Masculine cultures correlate with high use of *Personalization* apps. Long-term oriented cultures seem to prefer *Sport, Casual* and *Word games*, as well as *Social* apps. In short-term oriented cultures, there is a preference for *Role playing games* and a need for *Weather* apps as well as *Comics*. It is noticeable that categories with high correlations differ from those with the highest usage in general (as presented in Figure 3), indicating that the differences are more sophisticated and complex by nature.

Reversely, the application categories that do not correlate to any of the VSM factors can provide us insights to the applications that are similarly important through all the studied countries. We take a closer look to the categories that correlate less than 0.2 (or -0.2, similarly) to the VSM factors. There are nine categories that correlate less than the given threshold to at least five VSM factors. The category *Dating* correlates slightly more (0.26) only to the Individualism versus collectivism, and the category *Events* to the Masculinity versus femininity (0.21). *Game role playing* has very low impact against five factors, but gains more than 0.3 correlation to Long versus short-term orientation. In addition to these, the category *Beauty* and many types of games have low correlation against every factor: *Game arcade*, *Game casino*, *Game music*, *Game simulation*, and *Game strategy*. To summarize, there are certain categories that are in general more independent from the VSM factors than the others.

Our results reflect that mobile usage and the VSM have complex dependencies. We show that there are certain application categories that correlate well to the VSM factors, but the VSM factors cannot alone explain our findings in mobile usage as a whole, even if the mobile usage reflects the VSM factors in the case of certain categories.

6 USE CASES OF GEOGRAPHIC AND DEMOGRAPHIC FACTORS IN MOBILE APP USAGE

	Power distance (PDI)
ρ	*Categories*
< -0.5	Music & audio, Entertainment, Weather
< -0.4	News & magazines, Productivity, Travel & local, Sports, Libraries & demo
< -0.3	Game trivia, Photography, Finance, Communication, Auto & vehicles, Game card
> 0.3	Family create
> 0.4	Game action
	Individualism versus collectivism (IDV)
ρ	*Categories*
< -0.4	Family create, Game action
< -0.3	Game education
> 0.4	Books & references, Photography, Libraries & demo, Education, Finance, Game words, Medical, Family music video
> 0.5	Auto & vehicles, Productivity, Sports
> 0.6	Weather, News & magazines, Travel & local, Health & fitness, Music & audio, Entertainment
	Masculinity versus femininity (MAS)
ρ	*Categories*
< -0.4	Family pretend
< -0.3	Game board
> 0.3	Personalization
	Uncertainty avoidance (UAI)
ρ	*Categories*
< -0.4	Parenting, News & magazines, Family music video, Game words
< -0.3	Education, Family education, House & home, Entertainment, Books & references, Family brain games
> 0.3	Family create
> 0.4	Game action
	Long versus short-term orientation (LTO)
ρ	*Categories*
< -0.4	Game sports
< -0.3	Family music video, Game word, Social, Game casual
> 0.3	Maps & navigation, Game role playing
> 0.4	Comics, Weather
	Indulgence versus restraint (IVR)
ρ	*Categories*
> 0.3	Sports, Photography, Communication, Game words
> 0.4	Music & audio, Family music video, News & magazines, Entertainment, Books & references

Table 5. The best category correlations to VSM factors with 44 countries.

In the previous section we demonstrated that both geographic factors and cultural values correlate with mobile app usage, and that the overall effect of the country information is larger than those of demographic factors. To better understand these differences, we next turn our attention into use cases where we analyze individual factors and countries in detail. Among others, we show that application category usage is similar in certain geographic areas, and that highly educated respondents used application categories in a similar fashion across country boundaries.

6.1 Mobile Usage Respects Geographic Boundaries

In Section 5.1 we analyzed differences and similarities in application category usage throughout countries, demonstrating that we can roughly discover three larger clusters from the 44 countries considered in our analysis. These clusters follow certain geographical and language boundaries. The similarities between application usage therefore extend

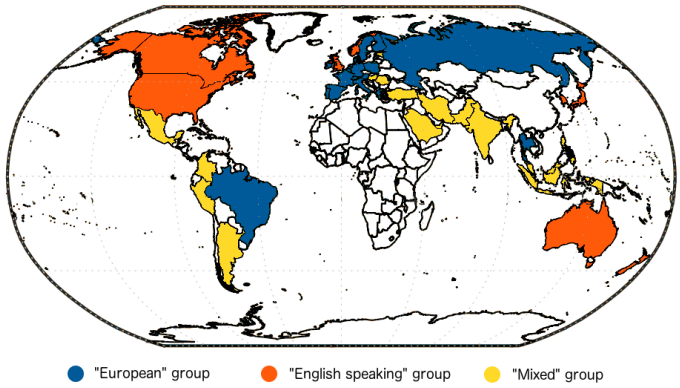


Figure 5. Countries colored by main clusters in the KL divergence analysis (see Figure 4b).

beyond country lines. To illustrate this, we show these clusters on a world map in Figure 5. The *"English-speaking"* group (red) includes countries like the USA, Canada, the United Kingdom, and Australia, but also South Korea and Japan that are traditionally considered closer to their Asian neighbors. The *"Non-English speaking European"* group, blue in Figure 5) consists of mainly countries in the continental Europe, but also includes Brazil and Thailand that may be considered as popular tourist destinations for Europeans. The *"Mixed"* group (yellow) includes various countries from Latin America, Middle Eastern countries, and many Asian countries.

These differences between application usage can also be quantified by looking at the application category data in greater detail. In Figure 6, we compare application category usage in certain categories strongly correlated with the VSM cultural value model (see Table 5). In general, the "English speaking" group uses more wider set of applications, which is statistically significantly high in almost every category, as shown in Figure 6.

The "Mixed" group is characterized by lower application usage across the board, but higher than the other two categories in *Sports* and *Racing games*. Some categories, such as *Food and Drink*, *Medical*, and *Shopping* are almost equally popular in both of the groups "non-English European" and "Mixed", but surpassed by the "English-speaking" group. *Weather* apps are, on the other hand, popular in the groups "non-English European" and "English-speaking", but less used in the "Mixed" group.

Communication apps are very popular in all groups. Although the "Mixed" group has low usage in most categories, it also has very high usage of the *Productivity* and *Social* apps. On the other hand, the "English-speaking" group has the highest usage in almost all categories. This may be due to the fact that almost all apps have an English version, and many services, retailers, restaurants, and public places in Europe and the USA have dedicated apps[6].

[6] McDonalds France is available in English: https://play.google.com/store/apps/details?id=com.md.mcdonalds.gomcdo&hl=en. The city of Wien has a dedicated mobility app: https://play.google.com/store/apps/details?id=at.wienerlinien.wienmobillab. Hyde Park club dedicated app: https://play.google.com/store/apps/details?id=com.hydepark. The Finnish weather map in English: https://play.google.com/store/apps/details?id=com.nordicweather_sadetutka. Sydney's Central Park has an app to aid sightseeing: https://play.google.com/store/apps/details?id=com.beaconmaker.android.centralpark.

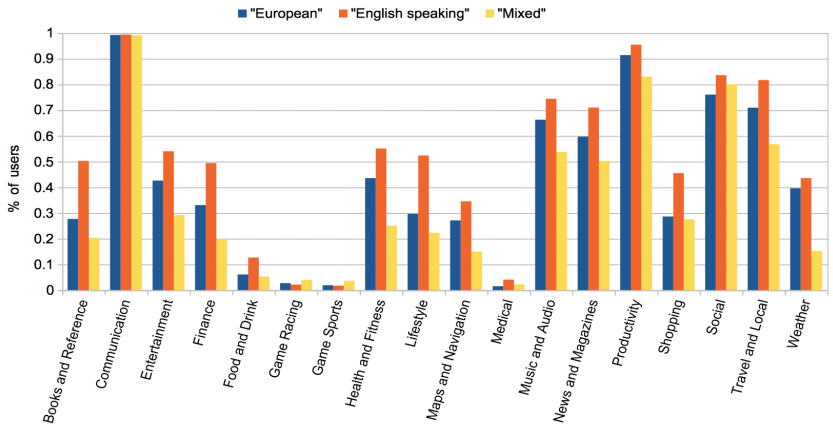


Figure 6. Usage of three main country clusters in several statistically significant categories.

6.2 Demography Forms Subgroups in Application Usage

Comparing demographic information - especially societal and economical factors - across countries can give us new insights regarding the Android application usage in different geographic areas. We study *occupation* and *education*, which have the highest information gain against application category usage (see Section 5.2). We also include *household status* to highlight some common demographic and geographic clusters and to endorse the view to societal and economical demographic factors in app usage. Our results indicate that professionals in Australia, Canada, the USA, and the United Kingdom use application categories in a similar fashion. This can also be observed in highly educated respondents.

Out of all questionnaire answers, we consider those that with ten or more responses. For example, there are hundreds of students and professionals, but only few respondents staying at home with kids, or working in agriculture. Also, countries with less than 10 respondents are excluded. In Figure 7, we compare the four most widely represented occupations (student, professional, retired, and technician or assistant professional) within 21 countries. In Figure 8, we present a similar comparison between the best represented educational levels (education leading to a profession, Bachelor's degree, Master's degree, and PhD equivalent degree). In both figures, darker color indicates closeness (the KL divergence between countries close to 0) and lighter color farther distance (the higher KL divergence).

As seen in Figure 7, professionals in Australia, Canada, the USA, and the United Kingdom use application categories similarly, indicated as a dark cluster in the North-Eastern corner of the colormap. The same cluster is visible in all the educational groups in Figure 8, and we may conclude that highly educated people or those working as professionals seem to use their mobile devices similarly in these Western, English-speaking countries.

Another cluster is visible in the South-Western corner of the colormaps, including Qatar (qa), India (in), and Indonesia (id). Especially students and people with PhD or equivalent degree are presented in this cluster, indicating similarities in application usage of academic people in these countries. It is possible that these groups also have a higher smartphone penetration, and the use of English may be required in studies.

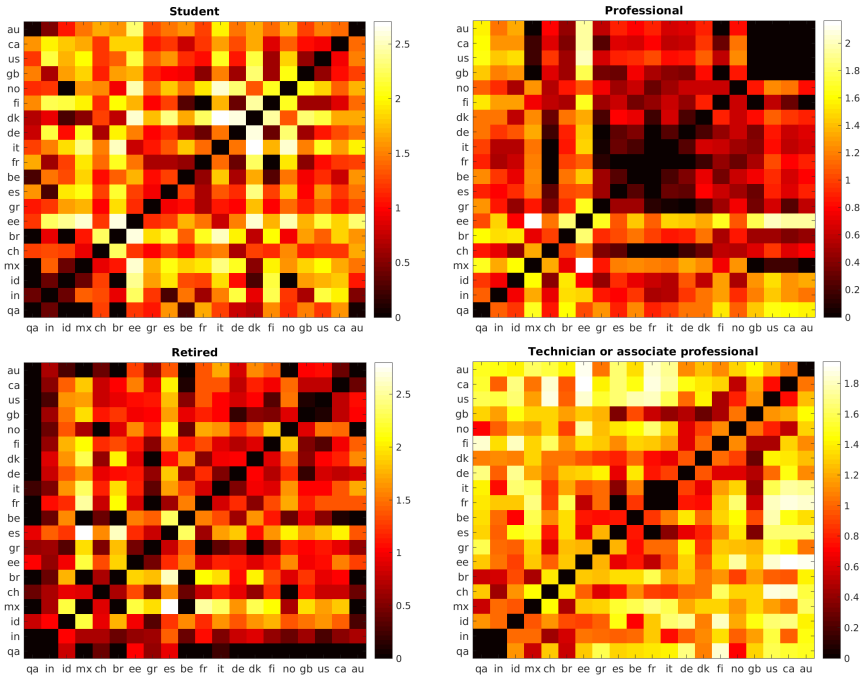


Figure 7. Colormaps of the KL differences by category usage in different countries and occupation groups. See the Appendix for the full names of countries.

For professionals and Master's degree holders there is also a third cluster in the middle of the colormaps. This cluster includes European countries: Denmark (dk), Germany (de), Italy (it), France (fr), Belgium (be), Spain (es), and Greece (gr). The application category usage of this group is different from the previously mentioned English-speaking cluster, as also seen in Figure 4b.

Interestingly, students seem to use applications differently in each country, as there are no clear clusters in the students' colormap. This might be because university students may travel to faraway countries to study, while e.g. vocational studies are commonly done in the same or nearby countries. The colormap of retired people is darker than the others overall, which means their application use is more similar through all the considered countries, but are few strongly similar clusters. This may be a result of people having used to different sets of applications, not adopting new ones as a group. Technicians and associate professionals have the lightest colormap in comparison, indicating that the countries have the highest distances (and thus less overall similarity) in this occupational category. This may be caused by the wide range of actual professions and people with different smartphone needs in the category.

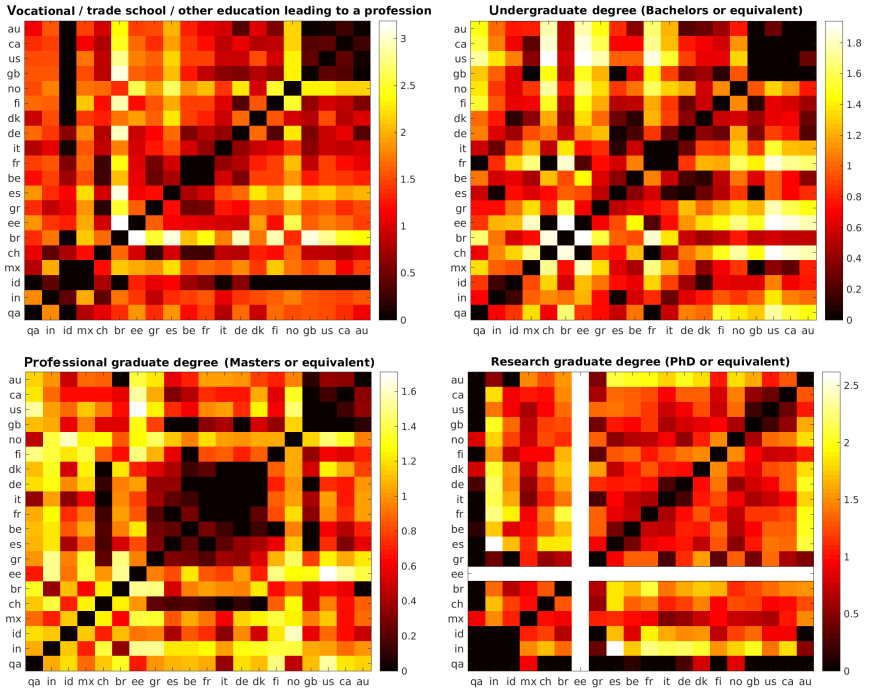


Figure 8. Colormaps of the KL differences in category usage in different countries and education groups. As far as the PhD degree is concerned, the values for Estonia (ee) are missing. See the Appendix for the full names of countries.

Respondents who report their household status are also similar to others with the same status (see Figure 9). Most statuses have a cluster of English-speaking countries in the upper right-hand corner of the figure, with a larger cluster of similar application usage for respondents living with other adults within European and English-speaking countries. The darker area encompassing Finland, Norway, and the United Kingdom may also be interpreted as its own cluster. This may result from vacationing or immigration between these countries.

To summarize, new clusters can be seen when both country and socio-economic information is taken into account. People in the same demographic group, especially professionals and well-educated people tend to have similar patterns in their application usage, and these similarities are strongest in English-speaking countries and, on the other hand, continental European countries. Interestingly, students' mobile application usage seems not to follow country boundaries, perhaps due to mobility during studies, or using the apps they are familiar with also when studying abroad.

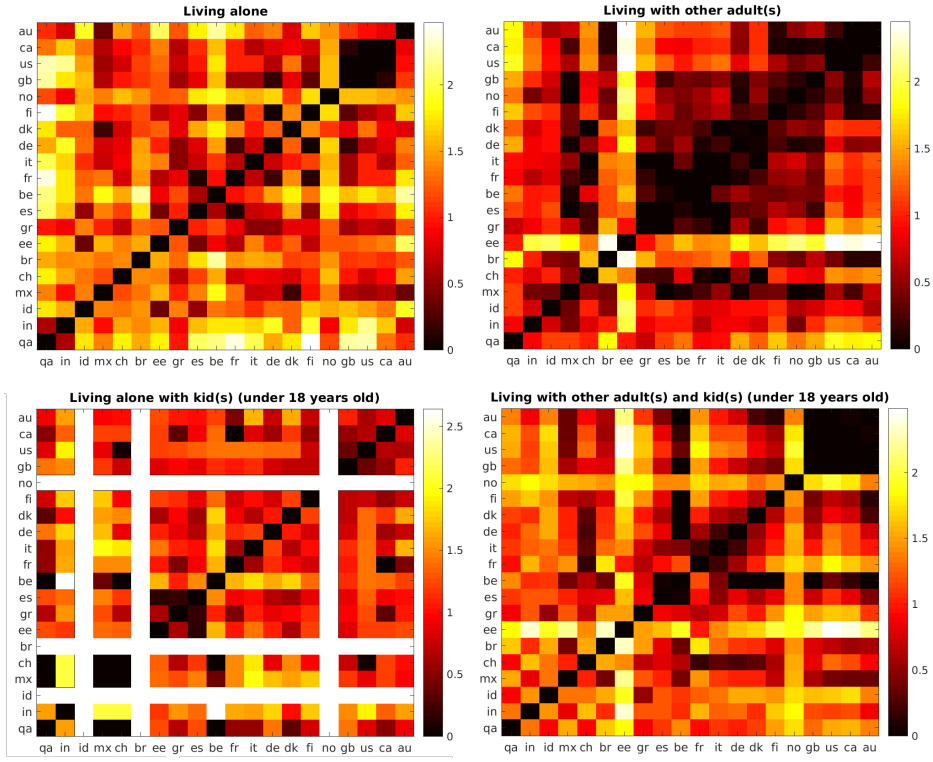


Figure 9. Colormaps of the KL differences by category usage in different countries and household statuses. White bars are indicating missing values.

7 DISCUSSION

Differences of application usage between countries may be influenced by many external factors. First, language can have a strong impact, such as English in the USA, Canada, and Australia, and Arabic languages in Qatar, Saudi-Arabia, and the Arab Emirates. On the other hand, the questionnaire has been presented only in English, and the Carat application is available only in English, Finnish, and Italian, so the respondents and indeed Carat users as a whole are most probably biased towards those familiar with English. This may also have reduced the number of questionnaire respondents in Asian countries. Second, close geographical and political relationships may also bring application usage closer between countries in Southern and continental Europe, for example, between Spain, France, Austria, Italy, and Greece. Germany, Denmark, and Finland also share an old trade route.

The VSM gives us an insight to value-based boundaries between countries in addition to geographical and historical similarities. Countries with collectivist and feminine values seems to prefer family related applications. Countries with

low power distance, that also indicates a low hierarchy, prefer applications from categories that are used for leisure and hobbies, such as *Music and Audio*, *Entertainment*, and *Travel and Local*. Those same categories are popular in countries with individualist values. There are natural interconnections within the VSM factors, and seems that they are also mirrored in application usage to a large degree.

We have shown that application category usage is similar in certain geographical regions. Similarly, the country attribute has highest information gain among all demographic factors, and models combining country with other factors can represent application category usage even better. However, demography features on their own seems not to be sufficient. This makes application category use complementary to the VSM model, and a stronger indicator of application usage behavior than the demographic factors that have been considered in this paper.

We have found that certain user groups, most notably students, did not seem to use applications in a similar fashion between countries. We can observe the same while examining respondents by age; younger respondents under 25 years of age are dissimilar between countries, while within age groups from 25 years to 64 years of age, clusters within central European countries and the English-speaking group form and grew more similar. However, respondents over 65 years of age broke this pattern, and clusters form between country pairs such as Germany and Denmark, Estonia and Greece, and Switzerland and Brazil, possibly indicating immigration or retirement destinations.

The application category use of respondents with the same household status has also been found to be similar. Respondents may be using applications targeted to singles, couples, or families, which are not present in the data of respondents with a different household status. Parenting, dating, and family applications have high information gain for household status. Also, the other Google Play *Family* subcategories see higher usage in households with children. The similar application usage of members of the same age, household, education, or profession group also motivates to study application category usage in more detail. Together with these societal, economical, and demographical factors, it can provide groupings across country boundaries.

In terms of limitations, our dataset collected using the Carat application relies on category information extracted from Google Play. We could not find a categorization with similarly extensive coverage for other markets than that of Google Play. Using another application market or platform would therefore require extensive labeling efforts, which would render our study infeasible to conduct. This limits the geographic coverage of our dataset as there are different application distribution channels in certain countries. For example, China uses local appstores instead of Google Play making China the largest omission from our dataset. We also observed the dataset to contain only few users from Africa, limiting our focus to other continents. We call for special focus on these missing regions in future data collection and studies.

Our questionnaire delivered through the Carat application has a bias towards male professionals. That might represent the distribution of Carat user population as a whole, or people using Android energy analysis apps in general. As in the larger dataset, some countries were underrepresented in the questionnaire, especially in Africa, and for that reason we decided to focus on a subset of 44 countries around the world. Finally, we excluded countries without VSM factors, which reduced the coverage of this study especially in the Middle East.

Another limitation of our study is the lack of data from iPhone users. To our best knowledge, Android is also the only large-scale mobile platform to provide a list of running applications via its public API since iOS no longer allows this[7] and neither does Windows Phone. Consequently both iOS and Windows Phone had to be excluded from the study. Some categories in Google Play may be too broad even though their number has increased over the years. For

[7] Apple WWDC 2015: "Ultimately, [in] the iOS security model the apps are isolated. They live within their own sandbox, protecting them from other apps and processes … apps on iOS are not privileged to see other apps' information." - https://developer.apple.com/videos/play/wwdc2015/703/

example, almost every user the *Communication* category, since it contains common messaging apps, some of which are preinstalled on most smartphones (e.g. Facebook).

Our dataset does not include user locations. For this reason, we use MCC to detect the country, which might introduce potential noise in border areas connecting neighboring countries. To mitigate this, we make sure the country information based on time zone matches the MCC. However, this information only provides the current country of each user, but not their nationality. We cannot effectively eliminate the effect of people travelling and immigrating between countries and causing deviations, but, even if this is hard to quantify, we believe that the errors introduced by these factors are relatively small overall.

8 CONCLUSIONS AND FUTURE WORK

In this work, we have compared mobile application use in 44 different countries with over 25, 000 smartphone users. We have shown that differences in mobile application category usage are statistically significant between countries. Our automatic methodology creates clusters of similar application category usage from widely collected mobile application usage data. We have shown that these similar usage clusters also reflect geographic and demographic factors and cultural values between countries, and factors from Hofstede's VSM model can be strongly associated to popularity of specific application categories.

Based on these results, we have given examples of mobile usage of Android users in the 44 different countries. For example, there are marked differences between the *non-English speaking European* countries (Russia and most of Europe and Brazil), and the *English-speaking* countries (the USA, Canada, Australia, the United Kingdom), and the *Mixed* countries (South American, Middle Eastern and South-East Asian countries). Particularly, the English-speaking group uses all categories in a more diverse fashion compared to the other groups, while the Mixed group uses more sports and racing games than the others.

We have also conducted a demographic questionnaire with 3, 293 respondents from 44 countries, and shown that the country of an individual is a stronger explanatory factor for application usage than any other demographic attribute. We have examined the education and profession of respondents and found that specific groups have similar application usage in certain countries. For example, professionals in Australia, Canada, the USA and the United Kingdom use application categories in a similar fashion, while students are diverse in their application category usage. The similar application usage of members of the same age, household, education, or profession group also motivates the use of application category use as a cultural factor. Together with demographics, it can provide groupings across country boundaries.

We have demonstrated that application category usage reflects geographical boundaries and demographic factors. We have found that the country of an individual is a stronger indicator of application usage than other demographic factors, such as occupation, education, and gender, and combining country with demographic factors we can further deepen our understanding of application usage. However, demography features on their own may not be sufficient enough. This makes application category usage complementary to the VSM model when analyzing countries and other groups of people.

Our results indicate that there is a strong relationship between application category usage data in our dataset and geographic and demographic factors, suggesting that when studying mobile usage data, these different factors should be taken into account. Our results can be used to better target mobile applications in different countries. The relationship between application category usage and demographic factors and cultural values can be used to determine the optimal

categorization for an app in a given country. When recommending applications, it is possible to emphasize categories that are widely used in the target country.

Our results can be used to take into account geographic and demographic variations when studying mobile users, but also for app recommendations. As different application categories are popular in different countries, it is possible to build a cultural value aware recommendation engine that recommends more apps from categories more likely to be used in the target country's value profile. In terms of application design, developers can benefit from our findings to help them choose the right categories for their apps in different countries and when targeting specific audiences, such as young professionals in the English-speaking world. Different audiences may be interested in different kinds of apps, and proper categorization can help users find them. Vice versa, a specific application might be targeted to specific countries and demographic groups according to where it would gain the largest audience.

In addition, the results indicate that further research into demographic subgroups within countries could yield commonalities in terms of application usage. For example, adults living with children choose different applications because of their household status, but do they choose the same apps across countries? This paper discussed category-level differences, which set the stage for deeper, app-level studies.

In addition to studying countries, the exact location could give city-level distribution of users to allow studying, for example, the differences in application use between different areas. Application usage between cities of various sizes, of the same or different country, could provide important novel insights. As we have location coordinates only from a part of the questionnaire respondents, we are not able to study this effect in this work. Also, our study could not cover influence of subcultures and minority groups within the countries.

REFERENCES

[1] Kumaribaba Athukorala, Eemil Lagerspetz, Maria von Kügelgen, Antti Jylhä, Adam J. Oliner, Giulio Jacucci, and Sasu Tarkoma. 2014. How Carat Affects User Behavior: Implications for Mobile Battery Awareness Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA.

[2] José S Marcano Belisario, Jan Jamsek, Kit Huckvale, John O'Donoghue, Cecily P Morrison, and Josip Car. 2015. Comparison of Self-administered Survey Questionnaire Responses Collected Using Mobile Apps Versus Other Methods. *Cochrane Database of Systematic Reviews* 7 (2015).

[3] Richard A. Bernardi and Steven T. Guptill. 2008. Social Desirability Response Bias, Gender, and Factors Influencing Organizational Commitment: An International Study. *Journal of Business Ethics* 81, 4 (01 Sep 2008), 797–809.

[4] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 47–56.

[5] Gokul Chittaranjan, Jan Blom, and Daniel Gatica-Perez. 2011. Who's Who with Big-Five: Analyzing and Classifying Personality Traits with Smartphones. In *Proceedings of the 2011 15th Annual International Symposium on Wearable Computers (ISWC '11)*. IEEE Computer Society, Washington, DC, USA, 29–36.

[6] John C. Crotts and Ron Erdmann. 2000. Does National Culture Influence Consumers' Evaluation of Travel Services? A Test of Hofstede's Model of Cross-cultural Differences. *Managing Service Quality: An International Journal* 10, 6 (2000), 410–419.

[7] Robert F De Vellis and L Suzanne Dancer. 1991. Scale development: theory and applications. *Journal of Educational Measurement* 31, 1 (1991), 79–82.

[8] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. 2010. Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. ACM, 179–194.

[9] Denzil Ferreira, Jorge Gonçalves, Vassilis Kostakos, Louise Barkhuus, and Anind K. Dey. 2014. Contextual Experience Sampling of Mobile Application Micro-usage. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*.

[10] Kendall Goodrich and Marieke de Mooij. 2014. How 'Social' are Social Media? A Cross-cultural Comparison of Online and Offline Purchase Decision Influences. *Journal of Marketing Communications* 20, 1-2 (2014), 103–116.

[11] Agnes Gruenerbl, Venet Osmani, Gernot Bahle, Jose C. Carrasco, Stefan Oehler, Oscar Mayora, Christian Haring, and Paul Lukowicz. 2014. Using Smart Phone Mobility Traces for the Diagnosis of Depressive and Manic Episodes in Bipolar Patients. In *Proceedings of the 5th Augmented Human International Conference (AH '14)*. ACM, New York, NY, USA, Article 38, 38:1–38:8 pages.

[12] Alexis Hiniker, Shwetak N. Patel, Tadayoshi Kohno, and Julie A. Kientz. 2016. Why Would You Do That? Predicting the Uses and Gratifications Behind Smartphone-usage Behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp*

'16). ACM, New York, NY, USA, 634–645.

[13] Daniel Hintze, Philipp Hintze, Rainhard D. Findling, and René Mayrhofer. 2017. A Large-Scale, Long-Term Analysis of Mobile Device Usage Characteristics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 2, Article 13 (June 2017), 13:1–13:21 pages.

[14] Geert Hofstede. [n. d.]. *Culture's Consequences: Comparing Values, Behaviors, Institutions, and Organizations Across Nations.*

[15] Seok Kang and Jaemin Jung. 2014. Mobile communication for human needs: A comparison of smartphone use between the US and Korea. *Computers in Human Behavior* 35 (2014), 376 – 387.

[16] Bradley L Kirkman, Kevin B Lowe, and Cristina B Gibson. 2006. A Quarter Century of Culture's Consequences: A Review of Empirical Research Incorporating Hofstede's Cultural Values Framework. *Journal of International Business Studies* 37, 3 (2006), 285–320.

[17] Neal Lathia, Veljko Pejovic, Kiran K. Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. 2013. Smartphones for Large-Scale Behavior Change Interventions. *IEEE Pervasive Computing* 12, 3 (2013), 66–73.

[18] Neal Lathia, Kiran K. Rachuri, Cecilia Mascolo, and Peter J. Rentfrow. 2013. Contextual Dissonance: Design Bias in Sensor-based Experience Sampling Methods. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 183–192.

[19] Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. 2013. MoodScope: Building a Mood Sensor from Smartphone Usage Patterns. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA, 389–402.

[20] Soo Ling Lim, Peter J. Bentley, Natalie Kanakam, Fuyuki Ishikawa, and Shinichi Honiden. 2014. Investigating Country Differences in Mobile App User Behavior and Challenges for Software Engineering. *IEEE Transactions on Software Engineering* 41 (2014), 40–64.

[21] Xuan Lu, Wei Ai, Xuanzhe Liu, Qian Li, Ning Wang, Gang Huang, and Qiaozhu Mei. 2016. Learning from the Ubiquitous Language: An Empirical Analysis of Emoji Usage of Smartphone Users. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 770–780.

[22] Brendan McSweeney. 2002. Hofstede's Model of National Cultural Differences and their Consequences: A Triumph of Faith - a Failure of Analysis. *Human relations* 55, 1 (2002), 89–118.

[23] Adam J. Oliner, Anand P. Iyer, Ion Stoica, Eemil Lagerspetz, and Sasu Tarkoma. 2013. Carat: Collaborative Energy Diagnosis for Mobile Devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*. ACM, New York, NY, USA, Article 10, 10:1–10:14 pages.

[24] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. 2015. Energy Modeling of System Settings: A Crowdsourced Approach. In *the 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom '15)*. 37–45.

[25] Thanasis Petsas, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P. Markatos, and Thomas Karagiannis. 2013. Rise of the Planet of the Apps: A Systematic Study of the Mobile App Ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. ACM, New York, NY, USA, 277–290.

[26] I. P. L. Png, B. C. Y. Tan, and Khai-Ling Wee. 2001. Dimensions of National Culture and Corporate Adoption of IT Infrastructure. *IEEE Transactions on Engineering Management* 48, 1 (Feb 2001), 36–45.

[27] Lin Qiu, Han Lin, and Angela K.-y. Leung. 2013. Cultural Differences and Switching of In-Group Sharing Behavior Between an American (Facebook) and a Chinese (Renren) Social Networking Site. *Journal of Cross-Cultural Psychology* 44, 1 (2013), 106–121.

[28] Katharina Reinecke, Minh Khoa Nguyen, Abraham Bernstein, Michael Näf, and Krzysztof Z Gajos. 2013. Doodle around the World: Online Scheduling Behavior Reflects Cultural Differences in Time Perception and Group Decision-making. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. ACM, 45–54.

[29] S. Ronen and O. Shenkar. 1985. Clustering Countries on Attitudinal Dimensions: A Review and Synthesis. *Academy of Management Review* 10 (1985), 435–454.

[30] Suranga Seneviratne, Aruna Seneviratne, Prasant Mohapatra, and Anirban Mahanti. 2014. Your Installed Apps Reveal Your Gender and More! *Mobile Computing and Communications Review* 18 (2014), 55–61.

[31] Viv J. Shackleton and Abbas H. Ali. 1990. Work-related Values of Managers: A Test of the Hofstede Model. *Journal of Cross-Cultural Psychology* 21 (1990), 109–118.

[32] Thiago Silva, Pedro Vaz De Melo, Jussara Almeida, Mirco Musolesi, and Antonio Louriero. 2014. You are What you Eat (and Drink): Identifying Cultural Boundaries by Analyzing Food & Drink Habits in Foursquare. In *Proceedings of the 8th AAAI International Conference on Weblogs and Social Media (ICWSM '14)*. Ann Arbor, Michigan, USA.

[33] Hien Truong, Eemil Lagerspetz, Petteri Nurmi, Adam Oliner, Sasu Tarkoma, and N. Asokan. 2014. The Company You Keep: Mobile Malware Infection Rates and Inexpensive Risk Indicators. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. ACM, 39 – 50.

[34] Hannu Verkasalo. 2011. An International Study of Smartphone Usage. *International Journal of Electronic Business* 1/2 (2011), 158–181.

[35] Janet Vertesi, Silvia Lindtner, and Irina Shklovski. 2011. Transnational HCI: Humans, Computers, and Interactions in Transnational Contexts. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 61–64.

[36] Pascal Welke, Ionut Andone, Konrad Blaszkiewicz, and Alexander Markowetz. 2016. Differentiating Smartphone Users by App Usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*.

[37] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Morley Mao, Jeffrey Pang, and Shobha Venkataraman. 2011. Identifying Diverse Usage Behaviors of Smartphone Apps. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference (IMC '11)*.

[38] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. 2016. Discovering Different Kinds of Smartphone Users Through Their Application Usage Behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16).*

A LIST OF COUNTRIES CONSIDERED IN THE STUDY

Code	Country name
ae	United Arab Emirates
ar	Argentina
at	Austria
au	Australia
be	Belgium
br	Brazil
ca	Canada
ch	Switzerland
co	Colombia
de	Germany
dk	Denmark
ee	Estonia
es	Spain
fi	Finland
fr	France
gb	United Kingdom
gr	Greece
hu	Hungary
id	Indonesia
ie	Ireland
in	India
ir	Iran
it	Italy
jp	Japan
kr	South Korea
mx	Mexico
my	Malaysia
nl	Netherlands
no	Norway
nz	New Zealand
pe	Peru
ph	Philippines
pk	Pakistan
pl	Poland
pt	Portugal
qa	Qatar
ro	Romania
ru	Russia
sa	Saudi Arabia
se	Sweden
sg	Singapore
th	Thailand
tr	Turkey
us	United States

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FI-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports are available on the e-thesis site of the University of Helsinki.

A-2013-1 M. Timonen: Term Weighting in Short Documents for Document Categorization, Keyword Extraction and Query Expansion. 53+62 pp. (Ph.D. Thesis)

A-2013-2 H. Wettig: Probabilistic, Information-Theoretic Models for Etymological Alignment. 130+62 pp. (Ph.D. Thesis)

A-2013-3 T. Ruokolainen: A Model-Driven Approach to Service Ecosystem Engineering. 232 pp. (Ph.D. Thesis)

A-2013-4 A. Hyttinen: Discovering Causal Relations in the Presence of Latent Confounders. 107+138 pp. (Ph.D. Thesis)

A-2013-5 S. Eloranta: Dynamic Aspects of Knowledge Bases. 123 pp. (Ph.D. Thesis)

A-2013-6 M. Apiola: Creativity-Supporting Learning Environments: Two Case Studies on Teaching Programming. 62+83 pp. (Ph.D. Thesis)

A-2013-7 T. Polishchuk: Enabling Multipath and Multicast Data Transmission in Legacy and Future Interenet. 72+51 pp. (Ph.D. Thesis)

A-2013-8 P. Luosto: Normalized Maximum Likelihood Methods for Clustering and Density Estimation. 67+67 pp. (Ph.D. Thesis)

A-2013-9 L. Eronen: Computational Methods for Augmenting Association-based Gene Mapping. 84+93 pp. (Ph.D. Thesis)

A-2013-10 D. Entner: Causal Structure Learning and Effect Identification in Linear Non-Gaussian Models and Beyond. 79+113 pp. (Ph.D. Thesis)

A-2013-11 E. Galbrun: Methods for Redescription Mining. 72+77 pp. (Ph.D. Thesis)

A-2013-12 M. Pervilä: Data Center Energy Retrofits. 52+46 pp. (Ph.D. Thesis)

A-2013-13 P. Pohjalainen: Self-Organizing Software Architectures. 114+71 pp. (Ph.D. Thesis)

A-2014-1 J. Korhonen: Graph and Hypergraph Decompositions for Exact Algorithms. 62+66 pp. (Ph.D. Thesis)

A-2014-2 J. Paalasmaa: Monitoring Sleep with Force Sensor Measurement. 59+47 pp. (Ph.D. Thesis)

A-2014-3 L. Langohr: Methods for Finding Interesting Nodes in Weighted Graphs. 70+54 pp. (Ph.D. Thesis)

A-2014-4 S. Bhattacharya: Continuous Context Inference on Mobile Platforms. 94+67 pp. (Ph.D. Thesis)

A-2014-5 E. Lagerspetz: Collaborative Mobile Energy Awareness. 60+46 pp. (Ph.D. Thesis)

A-2015-1 L. Wang: Content, Topology and Cooperation in In-network Caching. 190 pp. (Ph.D. Thesis)

A-2015-2 T. Niinimäki: Approximation Strategies for Structure Learning in Bayesian Networks. 64+93 pp. (Ph.D. Thesis)

A-2015-3 D. Kempa: Efficient Construction of Fundamental Data Structures in Large-Scale Text Indexing. 68+88 pp. (Ph.D. Thesis)

A-2015-4 K. Zhao: Understanding Urban Human Mobility for Network Applications. 62+46 pp. (Ph.D. Thesis)

A-2015-5 A. Laaksonen: Algorithms for Melody Search and Transcription. 36+54 pp. (Ph.D. Thesis)

A-2015-6 Y. Ding: Collaborative Traffic Offloading for Mobile Systems. 223 pp. (Ph.D. Thesis)

A-2015-7 F. Fagerholm: Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments. 118+68 pp. (Ph.D. Thesis)

A-2016-1 T. Ahonen: Cover Song Identification using Compression-based Distance Measures. 122+25 pp. (Ph.D. Thesis)

A-2016-2 O. Gross: World Associations as a Language Model for Generative and Creative Tasks. 60+10+54 pp. (Ph.D. Thesis)

A-2016-3 J. Määttä: Model Selection Methods for Linear Regression and Phylogenetic Reconstruction. 44+73 pp. (Ph.D. Thesis)

A-2016-4 J. Toivanen: Methods and Models in Linguistic and Musical Computational Creativity. 56+8+79 pp. (Ph.D. Thesis)

A-2016-5 K. Athukorala: Information Search as Adaptive Interaction. 122 pp. (Ph.D. Thesis)

A-2016-6 J.-K. Kangas: Combinatorial Algorithms with Applications in Learning Graphical Models. 66+90 pp. (Ph.D. Thesis)

A-2017-1 Y. Zou: On Model Selection for Bayesian Networks and Sparse Logistic Regression. 58+61 pp. (Ph.D. Thesis)

A-2017-2 Y.-T. Hsieh: Exploring Hand-Based Haptic Interfaces for Mobile Interaction Design. 79+120 pp. (Ph.D. Thesis)

A-2017-3 D. Valenzuela: Algorithms and Data Structures for Sequence Analysis in the Pan-Genomic Era. 74+78 pp. (Ph.D. Thesis)

A-2017-4 A. Hellas: Retention in Introductory Programming. 68+88 pp. (Ph.D. Thesis)

A-2017-5 M. Du: Natural Language Processing System for Business Intelligence. 78+72 pp. (Ph.D. Thesis)

A-2017-6 A. Kuosmanen: Third-Generation RNA-Sequencing Analysis: Graph Alignment and Transcript Assembly with Long Reads. 64+69 pp. (Ph.D. Thesis)

A-2018-1 M. Nelimarkka: Performative Hybrid Interaction: Understanding Planned Events across Collocated and Mediated Interaction Spheres. 64+82 pp. (Ph.D. Thesis)