UNIVERSITY OF CALIFORNIA, SAN DIEGO

Behavioral Context Recognition In the Wild

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal & Image Processing)

by

Yonatan Vaizman

Committee in charge:

Gert Lanckriet, Chair
Lawrence Saul
Mohan Trivedi
Nuno Vasconcelos
Nadir Weibel

2018

The dissertation of Yonatan Vaizman is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2018

DEDICATION

This dissertation is dedicated to my wonderful husband, Ran Goldblatt, who kept motivating me, encouraging me, and giving me an excellent example of a hard working academic.

I love you, Ran!

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Table of Contents . . . v

List of Figures . . . viii

List of Tables . . . ix

Acknowledgements . . . xi

Vita . . . xii

Abstract of the Dissertation . . . xiv

Chapter 1 Introduction . . . 1
- 1.1 Sensors and contexts . . . 3
  - 1.1.1 Sensing modalities . . . 5
  - 1.1.2 Naturally used devices . . . 7
  - 1.1.3 Describing behavior — context-labels . . . 8
  - 1.1.4 In this work . . . 9
- 1.2 Data collection . . . 10
  - 1.2.1 Towards in-the-wild . . . 12
  - 1.2.2 In this work . . . 17
- 1.3 Artificial intelligence (AI) and machine learning (ML) . . . 19
- 1.4 Contributions . . . 20

Chapter 2 Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches . . . 25
- 2.1 Introduction . . . 26
  - 2.1.1 Related work . . . 27
  - 2.1.2 Our work . . . 31
- 2.2 Context recognition system . . . 33
  - 2.2.1 Sensor fusion . . . 34
- 2.3 Data collection . . . 35
- 2.4 Evaluation and results . . . 40
  - 2.4.1 Why does sensor fusion help? . . . 41
- 2.5 User personalization . . . 45
- 2.6 Conclusions . . . 46
- 2.7 Future directions . . . 47
- 2.8 Supplementary material . . . 48

2.8.1 Mobile app . . . 48
2.8.2 Data collection procedure . . . 50
2.8.3 Sensor measurements . . . 52
2.8.4 Extracted features . . . 54
2.8.5 Label processing . . . 57
2.8.6 Classification methods . . . 61
2.8.7 Performance evaluation . . . 62
2.8.8 User personalization assessment . . . 64
2.9 Detailed results tables . . . 65
2.9.1 5-fold cross validation evaluation . . . 65
2.9.2 Leave-one-user-out evaluation . . . 69
2.10 Acknowledgements . . . 69

Chapter 3 Context-Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification . . . 74
3.1 Introduction . . . 75
3.2 The ExtraSensory Dataset . . . 78
3.3 Related work . . . 81
3.3.1 Off-the-shelf tools in controlled studies . . . 82
3.3.2 Getting out of the lab . . . 83
3.3.3 Transfer learning . . . 84
3.3.4 Missing input data . . . 85
3.3.5 Measuring recognition performance . . . 86
3.4 Our contribution . . . 87
3.5 Methods . . . 89
3.5.1 Multi-task multiple layer perceptron . . . 89
3.5.2 Data preparation . . . 91
3.6 Experiments and results . . . 92
3.6.1 Multi-task MLP . . . 92
3.6.2 The performance gain . . . 94
3.6.3 Transfer learning for new labels . . . 99
3.6.4 Missing sensors . . . 101
3.6.5 Interpreting the trained MLP . . . 104
3.6.6 External validation . . . 105
3.7 Discussion . . . 108
3.7.1 Future Improvements . . . 110
3.8 Conclusions . . . 111
3.9 Supplementary material . . . 112
3.9.1 Missing label information . . . 112
3.9.2 Results per-label . . . 117
3.9.3 Interpreting the multi-task MLP . . . 120
3.10 Acknowledgements . . . 125

Chapter 4 ExtraSensory App: Data Collection In-the-Wild with Rich User Interface to Self-Report Behavior . . . 128
4.1 Abstract . . . 129
4.2 Introduction . . . 129
4.3 Related work . . . 131
4.3.1 Camera-based Approaches . . . 131
4.3.2 Self-Reporting In-Situ . . . 132
4.3.3 Self-Reporting by Recall . . . 133
4.3.4 Mixed Self-Reporting Approaches . . . 133
4.4 The ExtraSensory Mobile App . . . 134
4.4.1 Recording Sensors . . . 135
4.4.2 Reporting Context Labels . . . 136
4.4.3 Additional Visual Features . . . 140
4.5 User Deployment, Analysis and Results . . . 143
4.5.1 Quantitative Analysis . . . 144
4.5.2 Qualitative analysis . . . 148
4.6 Discussion . . . 154
4.6.1 Revised ExtranSensory App . . . 157
4.6.2 Future directions . . . 158
4.7 Conclusion . . . 159
4.8 Acknowledgements . . . 159

Chapter 5 Discussion and future directions . . . 160
5.1 Sensors and contexts . . . 161
5.1.1 Multi-modal sensors . . . 161
5.1.2 Multi-label contexts . . . 163
5.1.3 Labeling resolution . . . 166
5.2 Data collection . . . 169
5.2.1 Potential hybrid methods . . . 169
5.2.2 Utilizing real-time automated context-recognition . . . 171
5.2.3 Measuring label reliability . . . 173
5.3 Artificial intelligence (AI) and machine learning (ML) . . . 174
5.3.1 Overcoming the curse of dimensionality . . . 175
5.3.2 Training with irregular data . . . 176
5.3.3 Open problems for future improvements . . . 178
5.4 Conclusion . . . 179

Bibliography . . . 180

LIST OF FIGURES

Figure 1.1: Behavioral context recognition overview 4
Figure 1.2: Data collection approaches — trade-off between reliable labels and in-the-wild behavior 16
Figure 1.3: Data collection approaches, in previous works and this work — trade-off between reliable labels and in-the-wild behavior 22
Figure 1.4: Automation feedback-loop for data collection 24

Figure 2.1: Context recognition system 36
Figure 2.2: Screenshots from the ExtraSensory mobile application (iPhone version). (A) History page. (B) Selecting labels from a menu. (C) Active feedback. (D) Notification. 39
Figure 2.3: Overall performance of the single-sensor classifiers (Acc, Gyro, WAcc, Loc, Aud and PS) and the sensor-fusion classifiers (EF, LFA, LFL and EF-LOO) 42
Figure 2.4: Why sensor fusion helps recognition 43
Figure 2.5: User adaptation performance 46

Figure 3.1: Dimesionality reduction. Comparing reducing dimension by PCA to a hidden layer of a multi-task MLP. 98
Figure 3.2: Activities of Daily Living (ADL) dataset 107
Figure 3.3: Sensor-features and hidden node activation 121
Figure 3.4: Context-labels and hidden node activation 123
Figure 3.5: Sensing modalities and behavioral aspects. Balanced accuracy (in %) scores of MLP (16,16) either with (green) or without (blue) sensor-dropout training 124

Figure 4.1: Label-reporting user interface, with flow marked in purple shapes and arrows 138
Figure 4.2: Notification flow with possible example scenarios 139
Figure 4.3: Watch — confirmation notification 141
Figure 4.4: Watch — open-ended notification 142
Figure 4.5: Sensor recordings for two users 145
Figure 4.6: Distribution of label-reporting over the minutes in the dataset 146
Figure 4.7: Distribution of label-reporting mechanisms for selected labels 147
Figure 4.8: Label-reporting mechanisms over time 149
Figure 4.9: Label reporting patterns 150

Figure 5.1: Three aspects of research in behavioral context recognition, and the relations between them 162
Figure 5.2: Data collection approaches, in previous works, this work, and possible future methods — trade-off between reliable labels and in-the-wild behavior . . . 170
Figure 5.3: Automation feedback-loop for data collection, with detour for practical applications that use context-recognition 173
Figure 5.4: Recognition performance with a linear model *vs.* multi-layer perceptrons with various dimensions of hidden layers 177

LIST OF TABLES

Table 1.1: Previously collected datasets for context recognition 18

Table 2.1: Statistics over the 60 users in the dataset (SD: standard deviation). 38
Table 2.2: The sensors in the dataset 38
Table 2.3: 5-fold evaluation performance (BA) of the different classifiers on each label. Part 1 of the labels 65
Table 2.4: 5-fold evaluation performance (BA) of the different classifiers on each label. Part 2 of the labels 66
Table 2.5: 5-fold evaluation performance (F1) of the different classifiers on each label. Part 1 of the labels 67
Table 2.6: 5-fold evaluation performance (F1) of the different classifiers on each label. Part 2 of the labels 68
Table 2.7: Leave-one-user-out evaluation performance (BA) of the different classifiers on each label. Part 1 of the labels 69
Table 2.8: Leave-one-user-out evaluation performance (BA) of the different classifiers on each label. Part 2 of the labels 71
Table 2.9: Leave-one-user-out evaluation performance (F1) of the different classifiers on each label. Part 1 of the labels 72
Table 2.10: Leave-one-user-out evaluation performance (F1) of the different classifiers on each label. Part 2 of the labels 73

Table 3.1: Recognition scores reported for baseline system (LR — logistic regression per-label), and for the multi-task MLP (either linear or with the dimensions of the hidden layers in parenthesis) 95
Table 3.2: Effect of instance-weighting. LR and MLP with two hidden layers of sixteen nodes, for each — performance with and without instance-weighting. 96
Table 3.3: Effect of non-linearity and hidden layers. In the per-label experiments, each label has a separate MLP model. 97
Table 3.4: Multi-task MLPs with node-wise architecture that is comparable to an MLP-per-label system. 97
Table 3.5: Transfer learning to a new set of context-labels 102
Table 3.6: Handling missing sensors 104
Table 3.7: Relations between sensors and contexts 106
Table 3.8: Label counts in the dataset, before and after regarding to missing label information (MLI) 115
Table 3.9: Logistic regression performance. Training without and with missing labels information. Performance scores reported with old and new metrics (without and with missing labels information, respectively). 116
Table 3.10: Balanced accuracy per label (part 1) 118
Table 3.11: Balanced accuracy per label (part 2) 119
Table 3.12: Relations between sensors and context-labels 1–25 126

Table 3.13: Relations between sensors and contexts-labels 26–51 127

ACKNOWLEDGEMENTS

I thank all my collaborators and coauthors. Special thanks to Katherine Ellis for being a wonderful partner in the ExtraSensory data collection effort. Special thanks to Nadir Weibel, for providing me with guidance in the fields of ubiquitous computing and human-computer interaction. Thanks to Brian McFee and Emanuelle Coviello for mentoring me in research on Music Information Retrieval in the first few years at UCSD. I would like to thank all the participants in the data collection for their dedicated time and effort.

Chapter 2, in full, is a reprint of the material as it appears in IEEE Pervasive Computing, 16(4):62–74, October-December 2017, Y. Vaizman, K. Ellis, and G. Lanckriet. The dissertation author was a co-primary investigator and co-author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), 1(4), December 2017, Y. Vaizman, N. Weibel, and G. Lanckriet. The dissertation author was the primary investigator and co-author of this paper.

Chapter 4, in full, is a reprint of the material as it will appear in the ACM Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018), April 2018, Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel. The dissertation author was the primary investigator and co-author of this paper.

VITA

2007	B. S. in Computer Science and Computational Biology with division in Mathematics, the Hebrew University in Jerusalem, Israel
2009–2011	Graduate Teaching Assistant, the Hebrew University in Jerusalem, Israel
2014	M. S. in Electrical Engineering (Signal & Image Processing), University of California, San Diego, USA
2014–2018	Graduate Teaching Assistant, University of California, San Diego, USA
2018	Ph. D. in Electrical Engineering (Signal & Image Processing), University of California, San Diego, USA

PUBLICATIONS

Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, Nadir Weibel, "ExtraSensory App: Data Collection In-the-Wild with Rich User-Interface to Self-Report Behavior", *Proceedings of the 2018 Conference on Human Factors in Computing Systems (CHI 2018)*. ACM, April 2018.

Yonatan Vaizman, Nadir Weibel, Gert Lanckriet, "Behavioral Context In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification", *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, Volume 1, No. 4. December 2017.

Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, "Recognizing Detailed Human Context In-the-Wild from smartphones and smartwatches", *IEEE Pervasive Computing*, Volume 16, No. 4. October-December 2017.

Brian King, I-Fan Chen, Yonatan Vaizman, Yuzong Liu, Roland Mass, Sree Hari Krishnan Parthasarathi, Bjorn Hoffmeister, "Robust Speech Recognition Via Anchor Word Representations", *Proceedings of Interspeech 2017*. August, 2017.

Yonatan Vaizman, Brian McFee, Gert Lanckriet, "Codebook Based Feature Representation for Music Information Retrieval", *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Volume 22, Issue 10. October 2014.

Emanuele Coviello, Yonatan Vaizman, Antoni B. Chan, Gert Lanckriet, "Multivariate Autoregressive Mixture Models for Music Autotagging", *13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, October 2012.

Yonatan Vaizman, Roni Y. Granot, Gert Lanckriet, "Modeling Dynamic Patterns for Emotional Content in Music", *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. October 2011.

Roie Kliper, Yonatan Vaizman, Shirley Portuguese, Daphna Weinshall, "Evidence for depression and schizophrenia in speech prosody", *Second ISCA Tutorial and Research Workshop on Experimental Linguistics — ExLing (ExLing 2010).* Athens Greece, 2010.

ABSTRACT OF THE DISSERTATION

Behavioral Context Recognition In the Wild

by

Yonatan Vaizman

Doctor of Philosophy in Electrical Engineering (Signal & Image Processing)

University of California, San Diego, 2018

Gert Lanckriet, Chair

The ability to automatically recognize a person's behavioral context (including where they are, what they are doing, who they are with, *etc.*) is greatly beneficial in health monitoring, aging care, personal assistants, smart homes, customized entertainment, and many other domains. For all of these different applications to succeed on a larger scale, the context-recognition component must be unobtrusive and work smoothly, without making people adjust their behavior. It is important for research to validate context-recognition systems in the real world — under the same conditions in which such applications will eventually be deployed. In this thesis, I promote context recognition *in-the-wild*, capturing people's authentic behavior in their natural environments using natural, everyday devices.

In Chapter 1, I introduce the field of behavioral context recognition, and describe three parts of research in the field: defining the problem (what are the inputs and what are the outputs), collecting data, and artificial intelligence (AI) / machine learning (ML) methods.

In Chapter 2, I present the problem of behavioral context recognition and the challenges of addressing behavior in-the-wild. I introduce the ExtraSensory Dataset, which was collected from 60 participants in-the-wild, and is publicly available at http://extrasensory.ucsd.edu. I describe simple machine learning methods and demonstrate that smartphones and smartwatches can be used to successfully recognize diverse contexts in regular life (*e.g.* walking, sleeping, at school, on a bus, cooking, shower, phone in pocket).

In Chapter 3, I specifically address machine learning solutions that facilitate training classifiers with irregular data from the wild — highly unbalanced, with occasions of missing labels or sensors, and potentially collected in phases addressing different sets of labels.

In Chapter 4, I address the challenge of collecting labeled data in-the-wild and describe our self-reporting solution – the ExtraSensory App. I analyze the collected data and subjective feedback from the participants to gain insights about user-interface design to engage users to contribute labels about their own behavior. A revised version of the app, with improvements based on this dissertation, is publicly available at http://extrasensory.ucsd.edu/ExtraSensoryApp.

In Chapter 5, I discuss the progress this work makes in the field of behavioral context recognition, and suggest directions for future improvements.

Chapter 1

Introduction

Recent technological developments in mobile devices and the increasing popularity of smartphones and smartwatches allow for new opportunities in context recognition. Smartphones today are equipped with a variety of sensors and have strong computational abilities, which enable rich information about the user to be collected and processed easily and efficiently. This opens the possibility for context recognition systems that use sensor measurements from smartphones and smartwatches (*e.g.* accelerometer, gyroscope, GPS, microphones, *etc.*) to automatically recognize various aspects of a user's situation (*i.e.* their location, environment, activity, mood, or intentions). Such a system should combine these diverse sources of data to gain better recognition of what the user is doing or experiencing — their *behavioral context*.

Context recognition systems have the potential to serve a wide variety of applications. In particular, there are a variety of medical applications where patient monitoring can be useful. A system that can provide automatic logging and monitoring of a patient's behavior (including environmental exposure) can provide more information for the clinician for better diagnosis, treatment and follow up. For example, physicians or researchers might be interested in monitoring diet and physical activity for diabetes patients. Cognitive impairment in older adults can be monitored by tracking activities such as reading, talking and social interactions. Similar logging applications can help public health researchers conduct studies about designing intervention programs to encourage physical activity. By using an automated context recognition system, physical activity logging can be done objectively and without any extra effort by the research subjects. Monitoring social activities or mood can provide insights for patients with depression.

In addition, commercial applications can take huge advantage of user context recognition — to improve recommendation systems (media, shopping, web search, *etc.*) by making them context-relevant, or to boost personal assistant systems by fitting the assistant better to the current situation of the user. Even digital communication networks will benefit from automated context recognition, for instance, if the system recognizes that many users are gathered together in the same place (*e.g.*, a sport event), the usage of the network infrastructure can be optimized

accordingly (*e.g.*, by avoiding redundant communication or using peer to peer communication).

In this work, I address the problem of behavioral context recognition, and specifically promote context recognition systems to work well *in-the-wild*, meaning in real work applications (as opposed to controlled conditions). This chapter gives an introduction to behavioral context recognition and surveys previous works in the field, to set the ground for my research work.

Behavioral Context Recognition

Behavioral context recognition is a type of artificial intelligence (AI) task, where a system uses signals from sensors to automatically recognize what is going on with a person at a given time — their behavioral context. The recognized context may include physical activity, mental or social activity, as well as other aspects of behavior, like location or interaction with other people. Figure 1.1 presents an overview of the different components in research and development of behavioral context recognition systems. In this chapter, I go through these different components and survey how previous studies in the field addressed them. This field was sometimes referred to as human activity recognition (HAR), and relevant research surveys about HAR were written by Lara *et al.* [43] and Bulling *et al.* [11]. I recently gave an "introduction to Behavioral Context Recognition" talk as part of a "Ubiquitous Computing" class, taught by Nadir Weibel at the University of California, San Diego; this talk was video-recorded and is available at https://www.youtube.com/watch?v=2cuhvEQZ_sI.

1.1 Sensors and contexts

The sensors being used and the contexts being recognized describe the input and output of the system, respectively, and as such, they define the problem as a function from input to output (see Figure 1.1 (B,D)). The choice of sensing modality and sensing-device placement

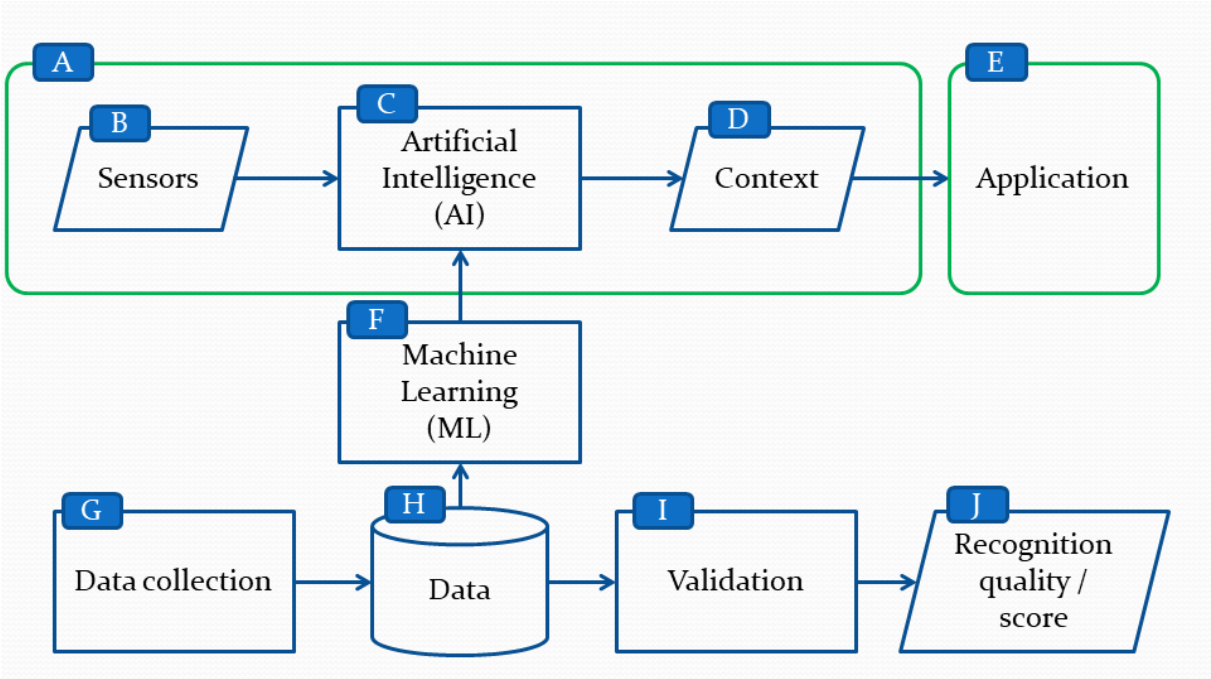

Figure 1.1: Behavioral context recognition overview. A context-recognition system (A) is a system whose input comes (typically) from sensors (B) and whose output is the recognized context (D). The system can serve various practical applications (E). The internal mechanism of the system can be viewed as performing artificial intelligence (C) and it usually requires training with machine learning methods (F). Data (H) is needed both to train the system with machine learning tools, and to validate the system's quality (I) to report some score (J). The methods for collecting the data (G) have significance to the validity of the system and to its potential to work well in the wild.

were usually motivated/directed by the targeted context-labels that researchers were interested in recognizing.

1.1.1 Sensing modalities

Previous research works used a wide variety of sensing modalities to perform context recognition.

Some studies used **stationary sensors** and deployed an array of sensors in a fixed test environment (*e.g.* a home or office). This can include simple and cheap binary state-change sensors (switches for state change, *e.g.* refrigerator opened, window closed, light switch turned, toilet flushes, electrical appliance turned on/off *etc.*) [87, 99, 33, 64]. It can also include sensors of different physical modalities, like chemical reactors [24, 95], proximity (RFID) or remote motion sensors [88, 99], and water pressure sensors [44]. These studies attempted to detect typical activities that happen in a home environment, and were sometimes motivated by applications for assisted living, aging at home, and efficient power usage. These approaches demonstrated how an ensemble of cheap sensors can be very informative for monitoring behavior in a home, while being unobtrusive and maintain natural behavior. However, they are limited to tracking the person in a specific environment, and they require a complicated setup and installation.

A long line of work was done to use **body-worn inertial sensors**, like accelerometers and gyroscopes. Such sensors were usually placed on different positions of the participant's body (*e.g.* arms, legs, wrist, chest, hip, pocket) and were used to track body posture and ambulation (lying down, sitting, standing, walking, running), other physical activities (like climbing stairs, cycling, rowing, carrying weight, hopping, *etc.*), and specific-motion activities, like brushing teeth or typing [55, 4, 70, 35, 86, 84, 38, 23].

Some studies added **GPS** (global positioning system) sensing to track location for recognizing physical activities as well as transportation modes [73, 42, 20].

Bio-sensors, like heart-rate monitors, skin temperature and electromyography (EMG),

have also been used to track body movement and specific activities, like walking, running, or eating [65, 9, 86, 42].

Some studies designed systems that would recognize the context based on **audio**. Several studies explored auditory scene analysis methods to recognize context from ambient sounds. These studies attempted to recognize various environments (like restaurant, office, bus, marketplace, *etc.*) [67, 22] or also live entities (*e.g.* speech, dog) and inanimate objects (*e.g.* washing machine) [76, 75]. Other studies targeted context related to the person themselves, for example recognizing bodily sounds of laughing, talking, eating *etc.* from microphone on the neck [102, 71]; detecting eating activity from ambient audio recorded from the wrist [90]; and assessing stress level from the person's voice as recorded from their phone [50, 1].

In some studies, the suggested modality was **vision**, where people would wear a camera facing out, and the system would use the scene-images from the camera to automatically recognize various daily activities, like driving, cooking, cleaning, watching TV, riding a bike, *etc.* [12, 68]. In some studies a camera was facing the subject, and researchers used computer vision methods to analyze images or video of the person's face/head for tasks like assessing stress level [57], or inferring intention of a driver [19]. This approach to a computer-vision-based recognition system is distinguished from other studies that used body-worn cameras for the sole purpose of annotating data from other sensors with context-labels [102, 89, 20].

Several studies designed systems with a **combination of different sensing modalities**. Some of these studies conducted feasibility experiments with many diverse sensors (including acceleration, audio, light, air pressure, humidity, heart rate, skin temperature, and more) at different positions on the body [65, 21, 46, 42]. Although such studies promoted knowledge of processing multi-modal sensor signals and intuition about which modalities are informative for different contexts, the experimental sensing apparatus in those studies was unnatural and possibly uncomfortable to wear/carry (sometimes involving a sensor-box and wires). Other studies addressed the issue of convenience and practicality of the system and designed small,

lightweight devices with multi-modal sensors and wireless communication [69, 14], or designed sensors that are embedded into textile for more natural integration to everyday life [13].

The focus on specific context-labels was often motivated by certain applications (*e.g.* fall detection for supporting aging at home [59]), and different researchers suggested various choices of sensors that may carry informative signal about the behavior of interest. For the task of eating-detection, researchers suggested systems based on diverse sensing-modalities, including proximity sensors on the ear [6], microphone on the neck [102], electromyography around the neck [2], wrist motion [18, 89], and ambient sound recorded from the wrist [90].

1.1.2 Naturally used devices

Smartphones have become a natural agent for behavioral context recognition in the past decade. People carry around their phone almost everywhere, and even if it is not within arm's reach, it is usually close by [66, 17]. In addition, smartphones have become more and more equipped with a diverse set of sensors, including inertial sensors (accelerometer, gyroscope, magnetometer), GPS, and ambient sensors (microphone, light, humidity, air pressure, temperature, proximity).

Researchers have used accelerometers and gyroscopes, as well as other built-in modalities (*e.g.* audio, air pressure, light, GPS) in phones for some time to track body movement activities [77, 8], activities of daily living (*e.g.* watching TV, cooking, eating, vacuuming) [25, 37, 59], transportation modes [73, 29], and even estimate heart or breathing rate [30] or stress level [50]. Although such studies promoted utilizing everyday devices to recognize context, they often required that the phone be positioned in a particular place on the body (*e.g.* in a front pant pocket). The **device placement** has shown to have a significant effect on recognition [40]. Some studies specifically compared different positions [51, 30, 59], designed features to be less sensitive to placement of the phone [29], or even tried to infer whether the phone is in or out of a pocket [60].

Smartwatches are an ideal addition to phones; unlike phones, the regular usage of a

watch has a very consistent body positioning — the watch is restricted to the wrist. This gives opportunity to get a less noisy movement signal, while maintaining natural usage of the device. Previous studies have used smartwatches and wrist-worn sensors [56, 26, 82].

1.1.3 Describing behavior — context-labels

Behavioral context recognition is an artificial intelligence (AI) task, and as such it is usually defined with some simplified structure. Specifically, the recognized context (the output of the system) has some structure — the researcher/designer of the system decides how to describe behavior in a simplified manner.

In most previous works, the approach was a **multi-class** description of behavior, where a single label (typically describing "human activity") is applied at any given time, and the AI task is formulated as a multi-class classification. For example: a single ambulatory state out of the options {lying down, sitting, standing, walking}, a single transportation mode out of the options {walking, bus, train, metro, tram} [73, 29], or a single activity of daily living out options like {brushing teeth, washing dishes, watching TV, cooking, eating, driving, ...} [25, 68, 37]. While this approach is very simple and enables usage of generic machine learning methods, it can sometimes become too simplistic, and fail to capture the full richness of behavior in the wild. For example, there are different flavors of running (you can run on a treadmill, indoors; you can run outside; alone or with friends, *etc.*) and they will all elicit different signal patterns in the sensors. Depending on the application, researchers may be interested in different aspects of the complex multi-dimensional behavior: one study may only care about the physical exercise of running, while another may be interested to track the person's exposure to fresh air or social activities while they were running.

Some studies addressed this by adding more categories to the multi-class set (*e.g.* having both "running" and "running on a treadmill" [37]). However, this solution scales exponentially with the amount of detail/dimensions that we want to capture. A way to overcome this issue

is to represent behavior in a combinatorial manner, by adding more dimensions to the description of behavior: for example, in addition to the selection of activity adding indications of indoors/outdoors and eating/not-eating [21]. Similarly, in [68], researchers used a **multi-label** formulation to annotate images from body-worn camera with multiple objects that appear in the scene (although, for the higher-level task of activity-recognition they still used a multi-class formulation — assigning to each image one out of eighteen activities).

The use of strict multi-class description of behavior has implications also on the experimental part — the data collection with actual participants. When participants collect data from their natural life [25, 37], they may find situations where it is not clear which is the more appropriate option to describe the behavior (*e.g.* when moving around in the house, is it considered "walking"?). There can also be situations where someone is engaged in multiple activities simultaneously, like running on a treadmill while watching TV. The multi-class approach imposes labels that are objectively defined by researchers, thus may cause participants to adjust their behavior in order to conform to one of the activities in the target list.

1.1.4 In this work

In this work, I wish to promote the applicability of behavioral context recognition to *in-the-wild* situations. For defining the problem, this has implications on both the input (sensors) and output (context) of the system. For the input, I use **everyday devices** — smartphones and smartwatches. Furthermore, each participant used their own personal phone and carried it without any restrictions of device placement. The goal was to evaluate how well we can recognize a person's context from the everyday regular usage of their own phone. The added smartwatch is natural to wear and adds little burden. For the output, I use the **multi-label** approach. In order to cover many situations, we included a large menu of context-labels of different behavioral aspects (activity, location, phone position, company) and we let the person subjectively describe their own behavior by selecting appropriate combinations of context-labels.

1.2 Data collection

The methods of data collection have an influence on the validity of experimental results and on the utility of trained classifiers to work in-the-wild; models that are trained on simulated or prescribed activities may generalize poorly to real-life situations [36]. There is a trade-off between consistency/regularity of the data and ecological validity of the data; lab-controlled experiments sacrifice ecological validity but gain consistent and convenient data, and in-the-wild settings emphasize ecological validity but result in irregular data that is harder to work with.

Since behavioral context recognition is in the realm of *supervised* machine learning (the AI task is a function from input to the output context-labels), a key component for data collection is **assigning context-labels** to the sensor measurements, to act as ground truth (or reference). These context-labels are used for both training the system (Figure 1.1 (F)) and for testing it (Figure 1.1 (I)) to quantify its performance. Unlike AI tasks that imitate some human cognitive function (*e.g.* object recognition from images, speech recognition), behavioral context recognition typically uses sensing modality that humans are not naturally used to process. This makes it not practical to use offline annotation of the collected examples: for instance, a researcher cannot observe a plot of phone-accelerometer measurements and determine whether the phone-user was walking, cooking, or in the shower at the time. This makes label-collection an important challenge and studies have approached it in different ways.

In Chapter 2 (originally published in [91]), I define four **in-the-wild conditions of behavior**. Those are meant to be guidelines for designing context-recognition systems that should work well in-the-wild in actual real-life applications. These conditions are also guidelines for collecting research data that simulates real-life conditions as best as possible. These four conditions are:

1. **Naturally used devices (NUD).** The system should make use of convenient, unobtrusive devices that are natural for the user to use. These can be everyday-devices (like phones,

watches, clothes) or special devices that blend in well with the user's environment, in a way that doesn't alter their natural behavior.

2. **Unconstrained device placement (UDP).** The user should be able to use the devices naturally — in any way convenient to them. Again, the point is to maintain the natural behavior of the person.

3. **Natural environment (NE).** This is obvious for real applications, but it is a challenging requirement for collecting research data — the data should be collected in the person's own natural environment and on their own regular schedule.

4. **Natural behavioral content (NBC).** In order to validate what may happen in real life, the collected/recorded behavior should be as natural for the participant as possible. This means, without a script or specific instructions of what to do or how to do it. Ideally, this would also be the original, authentic behavior for each individual participant (which may be different activities for different people).

Table 1.1 summarizes several related datasets for context-recognition, and indicates for each dataset whether it fulfilled the four in-the-wild conditions, its scale, and whether or not it was publicly available.

Many studies collected data under **controlled conditions in a lab** [14, 74, 26, 82, 59]. In these studies, participants would come to a designated location on scheduled time (violating NE) to collect the data. The researchers would then place devices in specific body positions (violating UDP) and instruct the participant to perform some tasks according to a scripted protocol (violating NBC). These studies gave the researcher more control over the collected data, making it easier to generate consistent and well balanced datasets, which are important for developing signal processing methods to detect events/activities from sensor measurements. The down side of such approaches is that the recorded behavior does not cover the full richness and variability of behavior in-the-wild and the simulated behavior in the lab may be unnatural. In

order to generalize better to real-life situations, it is important to validate methods also with data that is collected in more realistic conditions.

1.2.1 Towards in-the-wild

To promote ecological validity, some studies collected data **outside of the lab**, with various degrees of realistic settings. As stated in the NBC in-the-wild condition, the participants' recorded behavior will be more natural and realistic when there is no researcher observing them, and of course no researcher instructing them what to do. Collecting data in the natural environment (the person's own home, workplace, *etc.*) further makes the recorded behavior more authentic. These additions of ecological validity come at a cost — it is much harder to assign relevant context-labels to the correct sensor-measurements.

Some studies applied a **camera approach** to annotating data, where the person wore a camera device that took images of the scene, and these images would later help annotate what was going on. Pirsiavash and Ramanan [68] used a GoPro camera mounted around the chest and collected data from twenty participants from their own homes, performing home-activity tasks from a prescribed list, but on their own time. In that work, the camera had two roles: it was designed to be the sensor for the future context-recognition system (a computer vision based system), and it was also used to annotate the ground truth labels of the objects in the scene and the activities. Ellis *et al.* [20] used a SenseCam device hung around the neck. In their work, the camera was only augmented for the purpose of offline annotation of the context. The actual sensors for context-recognition were accelerometers and GPS. Both these works [68, 20] involved offline annotation done by research assistants, which may compromise the privacy of the participant and their surrounding. Other studies shifted the annotation effort to the participants themselves, to self-report their own context-labels based on the images taken by a worn camera [12, 89]. This allowed for the participants to control their privacy better, but it also added much load to them, so special software tools were designed to help people label their

own images as quickly and easily as possible.

So far, these works [68, 20, 12, 89], which used cameras for collecting labels, promoted some conditions of in-the-wild behavior (NE, NBC), but at the expense of other in-the-wild conditions (NUD, UDP): on one hand, people contributed data from their natural environments, on their own time, and without instructed scripts, but on the other hand, researchers introduced a foreign, unnatural, and sometimes uncomfortable device (GoPro, SenseCam, phone camera) and forced that device's placement (around the neck, on chest), which can greatly affect the authenticity of the behavior. In Thomaz's work [89] (see two rows in table 1.1), this trade-off is clearly exhibited: their pilot study had a lab part, where they told the participants what to do from a script and the participant was video-recorded for later annotation, but they only had to wear the watch-accelerometer; they also had an in-the-wild part, where participants roamed freely without observation, but they had to also wear a camera and annotate their own behavior. As technology progresses, wearable cameras are sure to become more lightweight and convenient, so it will be much more natural to incorporate them in data collection. Future solutions for privacy will also help use cameras for annotating behavioral data.

Several research works relied on **self-reporting in-situ**, where the participant is in charge of describing their own behavior in real time. One way to do it is to use the experience sampling method (ESM, sometimes referred to as ecological momentary assessment — EMA [81]), where the system initiates prompting the user to fill a form or answer short questions describing their current situation or feelings [98, 96]. Another way is to let the participant initiate reporting about their activity when they start it, using some user interface (on a PDA [21] or phone) that allows to mark start and end time of the activity, and possibly select the activity from a short list [18, 25, 37]. When self-reporting context in-situ, the reported labels can be trusted as reliably conveying what the person is doing or experiencing at the moment, because the context is fresh in the person's memory. However, the action of reporting in-situ may interrupt or alter the natural behavior of the person.

An alternative is **self-reporting by-recall**, where the participant describes their own behavior after-the-fact, by recalling. Trying to remember your earlier context (what you experienced a few hours ago) can be difficult, so studies that used recall-based self-reporting suggested various solutions to help the participant recall. This includes observing snapshots from body-worn camera [12, 89], using the day reconstruction method (DRM) [90], or utilizing automatic recognition of basic activities (like walking, bicycling) to help recall and fix recognition-errors [15]. The eating-detection work mentioned earlier, by Thomaz *et al.* [89] exhibits another trade-off — between the authenticity of behavior and the ease of acquiring larger scale data with detailed labels (see two rows in table 1.1): in their lab part, it was easier to collect data from more participants (twenty), and they were able to track detailed labels of nine specific eating gestures; on the other hand, in their in-the-wild part of the study, they only acquired data from seven participants and only tracked a single binary variable — eating or not.

The main challenge in data collection for behavioral context recognition is the **trade-off between reliable labels and in-the-wild behavior**. On one hand, we want to have data with reliable labels (avoiding noise or mistakes in labeling), and on the other hand, we want the recorded behavior to be as natural and authentic as possible. These two objectives, which we want to maximize, typically compete with one another. Figure 1.2 demonstrates this trade-off by placing the different approaches in the 2-axis system. On one extreme, we have the very controlled conditions of a lab setting, where it is easy to obtain very reliable labels (by instructing the participant what to do, or by video-recording the session and annotating it later), but the behavior is not reflecting in-the-wild behavior (simulated, repeated, constrained device placement) [25, 82, 14, 59]. On the other extreme, self-reporting by recall, allows participants to behave freely during the day, maintaining in-the-wild conditions, but when they report about it later, they may forget the exact context they were in earlier, or the exact time they did or experienced something. Recall with contextual-cues [72] or semi-automated context logging [15] can ease some of that memory bias, and help reporting by-recall provide less noisy

labels. In between, other current methods offer different ways to balance the two objectives. Using a wearable camera enables participants to contribute data from their natural environment (NE), and potentially engage in their own regular behavior (NBC), but camera-based studies so far used devices and device placement that are not natural and may harm the manner of behavior [68, 20, 12, 89]. Experience sampling method (ESM) [98, 96] allows for people to behave freely most of the time, and once in a while to answer a few questions, prompted by the system. Since the ESM questions appear in-situ, the person can typically have an accurate report about what they are currently doing, experiencing or feeling, so the labels can be fairly reliable. However, the mere act of filling a form or answering a question may interrupt the in-the-wild nature of behavior, or affect the person's context (*e.g.* by causing stress [72]). In the timer-approach [21, 25, 37], the participant initiates reporting, so they are free to start reporting when convenient to them, typically right before starting some activity of interest; this can reduce interruptions and support a bit more natural behavior. On the other hand, in this approach, the participant may forget to mark the stopping of an activity at the right time, resulting in time-segmentation errors.

One of my main goals in this work is to ease this trade-off, and provide data collection methods that enable getting reliable (and detailed) labels about a person's behavior, while reducing the interference in the natural course of the person's life (maintaining in-the-wild behavior). I aim towards filling the top-right area of the trade-off plot in Figure 1.2.

Besides "how to collect context-labels?", another challenging issue in data collection in-the-wild is **privacy**. In lab studies, there is little privacy concern, so participants are more agreeable to publish data that they contributed, and even augment them with some identifiable pieces of information, like weight, height, and gender [59]. However, in in-the-wild studies, the more authentic the behavior, participants would be more concerned with protecting their privacy. This puts limitations on what can be shared as publicly available data, as well as on the recorded data available to the researchers (*e.g.* not recording full audio of phone conversations).

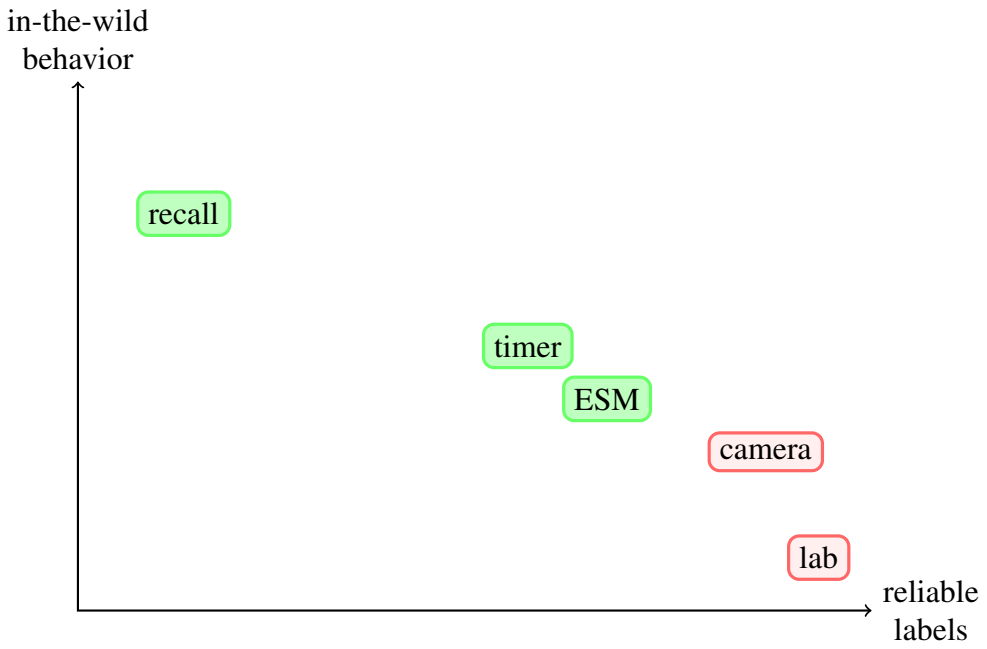

Figure 1.2: Data collection approaches — trade-off between reliable labels and in-the-wild behavior. Approaches that rely on self-reporting using everyday devices (phone, watch) are marked in green: ESM = experience sampling method, timer (when the participant initiate reporting labels in-situ and marks start and stop time of a selected activity), and recall-based self-reporting. Other approaches are marked in red: data collection in a lab, and data collection outside of the lab, using a wearable camera. Current approaches trade-off between the two objectives, and the optimal area to aspire to is the top-right part of this space.

1.2.2 In this work

For our data collection, we emphasized maintaining in-the-wild conditions. Naturally used devices: the devices we used were everyday devices — smartphones and smartwatches; furthermore, each participant used her own personal smartphone, to increase the authenticity of behavior. Unconstrained device placement: there was no constraint on device placement — the watch was naturally worn on the wrist (the wrist each participant preferred, for convenience), but participants were free to take it off when they wanted, and they were free to use their phone in any way convenient to them. Natural environment: each participant contributed data from their own natural environments (home, work, school, commute, *etc.*) and on their own regular schedule. Natural behavioral content: participants engaged in their regular routine, without any list of tasks or activities that were objectively targeted by us, researchers.

Our solution for collecting context-labels in-the-wild relied on self-reporting and was designed as a mobile app — the ExtraSensory App — with a rich user-interface to self-report behavior. The interface combined multiple methods from previous studies, including in-situ reporting (both by user active-feedback and by system-initiated notifications) and recall-based reporting (with a daily history page). It also included many features to make it quick and easy to cover long periods of behavioral time and report detailed multi-label descriptions with little interaction. This solution is fully described in Chapter 4 and will soon be published in [92]. A revised version of the ExtraSensory App is publicly available for free at http://extrasensory.ucsd.edu/ExtraSensoryApp.

The resulting dataset — the ExtraSensory Dataset — was first introduced in [91] (presented in this dissertation in Chapter 2). It contains over 300,000 minutes from sixty participants and is labeled with detailed descriptions in the form of combinations of over fifty diverse context labels. The ExtraSensory Dataset is publicly available for free at http://extrasensory.ucsd.edu.

Table 1.1: Previously collected datasets for context recognition. In-the-wild columns indicate the manner of collecting the data, relevant to the four in-the-wild conditions: NUD = naturally used devices; UDP = unconstrained device placement; NE = natural environment; NBC = natural behavioral content. The scale of the dataset is specified in terms of: number of participants (n_p), number of examples (n_e, sometimes indicated as number of minutes or hours collected), and number of labels (n_l, where just a number indicates multi-class, and a number with asterisk * indicates multi-label). The last column, "pub" indicates whether the data is publicly available. Positive indications are marked with ✓, negative indications are marked ✗. Some cells indicate intermediate relevance with ✓✗, for example, Khan *et al.* [37] gave more options for device placement, but still instructed how to do it. The last row describes the data collected in this work — the ExtraSensory Dataset — where the data collection complied with all the in-the-wild conditions, is fully publicly available, and has larger scale than previous datasets.

		in-the-wild				scale			
year	study	NUD	UDP	NE	NBC	n_p	n_e	n_l	pub
2010	Ganti [25]	✓	✗	✓	✓	8	80 h	8	✗
2012	Reiss [74]	✗	✗	✗	✗	9	10 h	18	✓
2012	Han [27]	✓	✓	✓✗	✗	10	10k	8	✓
2012	Pirsiavash [68]	✗	✗	✓	✓✗	20	10 h	18, 42*	✓
2013	Hemminky [29]	✓	✗	✓	✓✗	16	150 h	6	✗
2014	Khan [37]	✓	✓✗	✓	✓	30	33k	15	✓✗
2014	Ellis [20]	✗	✗	✓	✓	40	103k m	5, 12*	✗
2015	Thomaz [89]-lab	✓	✓	✗	✗	20	600 m	9	✓
2015	Thomaz [89]-wild	✗	✗	✓	✓	7	2k m	2	✓
2015	Castro [12]	✗	✗	✓	✓	1	40k	19	✗
2015	Shoaib [82]	✓	✗	✗	✗	10	340 m	13	✓
2016	Vavoulas [94]	✓	✗	✗	✗	57	2.5k	13	✓
2017	Micucci [59]	✓	✗	✗	✗	30	12k	17	✓
2017	ExtraSensory [91]	✓	✓	✓	✓	60	308k m	51*	✓

1.3 Artificial intelligence (AI) and machine learning (ML)

Behavioral context recognition is an artificial intelligence (AI) task. The internal mechanism of a context recognition system is an AI component (Figure 1.1 (C)), typically composed of two main parts:

1. Feature extraction. The raw sensor measurements are processed and relevant features are extracted, depending on the sensing modality, for example, time-domain statistics from acceleration measurements [37] or Mel frequency cepstral coefficients (MFCCs) from audio [25].

2. Classification. The feature vector is given as input to a classifier, like support vector machine, k-nearest-neighbors, random forest, or multi-layer perceptron [53, 41, 69, 26, 82].

This routine is typically repeated for every short time-segment (*e.g.* every minute), and further post-processing is sometimes applied to the time-sequence of recognized contexts [20, 89].

Many studies suggested improvements or adjustments to the computational methods, like extracting the appropriate features [23]; reducing power consumption by using integer operations [3] or by feature selection [37]; improving time segmentation [31]; smoothing the recognition over time [20]; using different sensors dynamically as a decision tree [27, 29]; utilizing low-level recognition of events/objects to detect higher level of activity [89, 68], and more. In most works, as mentioned in 1.1.3, the output of the recognition task was defined in a multi-class way, so the AI component was a multi-class classifier, meaning one that outputs a single selected label (class) for every example.

In this work, I specifically address issues that arise in-the-wild and how they affect the design of the AI and ML methods. As described in 1.1.4, to capture the variability in behavior in-the-wild, I use a multi-label formulation, so the classifier that I apply to the sensor-data is a multi-label classifier, meaning one that has multiple outputs; it can be seen as multiple simultaneous binary classifications — one for each label. In Chapter 2 (and in the original

paper [91]), I use a separate model per-label, and describe simple baseline methods, based on linear classifiers, with early and late fusion of sensors; I demonstrate why fusing complementary sensing-modalities is important, especially in-the-wild: when behavior is not scripted/instructed and the phone is not constrained to a particular position, there are situations where one modality is "blind" and additional sensors are required to resolve the context.

Chapter 3 in this dissertation (originally published in [93]) is specifically dedicated to the AI and ML aspects — developing ML methods that are appropriate for in-the-wild data and to be used in real-time in-the-wild applications. In Chapter 3, I describe a single model that addresses many relevant issues: it combines all labels to a single multi-task model (with hidden layers), to reduce parameters and improve generalization; it is trained with special instance-weighting, to overcome the imbalance and missing label information in in-the-wild data; it demonstrates transfer learning to a new set of target labels; and it shows how to make the system more resilient to missing sensors.

1.4 Contributions

This work deals with the problem of behavioral context recognition and promotes solving this problem *in-the-wild*, by addressing the three parts described in this introduction:

1. **Sensors and contexts**. This work suggests formulating the problem in a way that is appropriate for application in-the-wild. The input of a context-recognition system should come from sensors of everyday-devices, like phones and watches, in the regular way that people use them. The output of the system (recognized contexts) should describe in-the-wild behavior — having rich variability, multi-dimensional aspects, and subjectively described by each individual person; a multi-label formulation enables this richness, while still allowing for practical AI solutions.

2. **Data collection**. In this work, I provide guidelines for maintaining in-the-wild behavior

when collecting data: naturally used devices, unconstrained device placement, natural environment, and natural behavioral content. I also fully describe my solution for how to collect data in-the-wild. The user-interface described here combines multiple methods to self-report context-labels, and helps overcome the trade-off between acquiring reliable labels and maintaining in-the-wild behavior. Figure 1.3 demonstrates that the combination of multiple method in the ExtraSensory App moves data collection towards the optimal area:

- The server-guesses and the user-reported details throughout the day make it easier later that day to remember earlier context — making reporting by-recall more reliable.
- Participants can utilize active feedback to mark start time, without specifying too many details and without foresight too long into the future. They know that they can later add finer detail, or extend their reported context (if it remains the same). This makes user-initiated in-situ reporting less intrusive than previous timer approaches, and more focused on reporting about time close to the present — avoiding errors of trying to label too far into the future.
- The option of responding to a confirmation-notification with the watch greatly reduces the intrusiveness of ESM — making system-initiated in-situ reporting maintain in-the-wild behavior.

3. **Artificial intelligence and machine learning**. The machine learning evaluations in this work demonstrate that behavioral context recognition is possible even when maintaining in-the-wild conditions of behavior (in spite of the greater variability and irregularity of the data). I specifically address properties that may arise when collecting data in-the-wild, from many participants: inconsistent/irregular data with missing sensors, missing labels, highly unbalanced labeling, data collection in phases, and the general great variability of behavior and of sensor-measurements in the wild. I provide models and machine learning

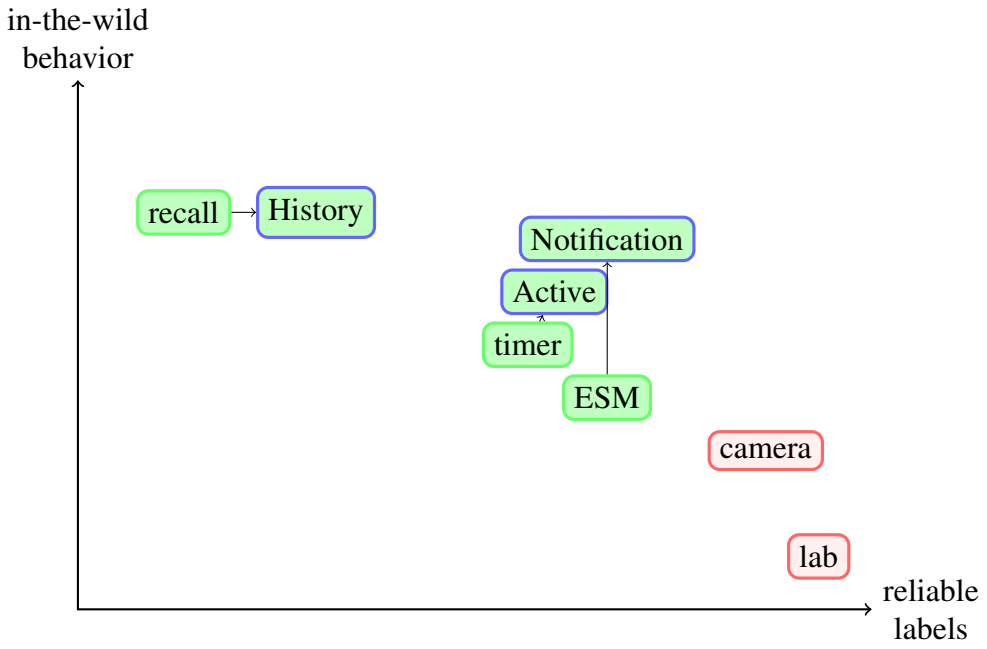

Figure 1.3: Data collection approaches, in previous works and this work — trade-off between reliable labels and in-the-wild behavior. Approaches that rely on self-reporting using everyday devices (phone, watch) are marked in green: ESM = experience sampling method, timer (when the participant initiate reporting labels in-situ and marks start and stop time of a selected activity), and recall-based self-reporting. Other approaches are marked in red: data collection in a lab, and data collection outside of the lab, using a wearable camera. The self-reporting methods in this work are based on previous methods marked with blue frames: history page, active-feedback and notifications. The added features of the ExtraSensory App and the combination of multiple methods helped ease the trade-off and achieve better balance between reliable labels and in-the-wild behavior.

methods to overcome these issues and train systems that are robust, generalize well to unseen people, and practical for real-time applications.

In addition to the written publications, my work on this thesis produced two major practical contributions:

1. **The ExtraSensory Dataset**. The dataset includes sensor measurements and context-labels from over 300k minutes recorded by sixty participants in-the-wild. It has rich labeling (on average, a minute is tagged with 3.8 labels from a vocabulary of more than 50 context-labels). This dataset can be used as a common benchmark to compare algorithms for context recognition. The dataset is publicly available for free at http:/extrasensory.ucsd.edu.

2. **The ExtraSensory App**. A mobile Android application (with a Pebble watch component). This app was originally used to collect the ExtraSensory Dataset in-the-wild. The published version is an improved revision, which includes adjustments based on our experience and suggestions from the participants. The app can be used for various purposes:

   - Collecting data, including sensor-measurements and context-labels self-reported by participants about their own behavior.
   - Tool for real-time context-recognition. By simply running ExtraSensory App in the background of the phone (and communicating with a server) the app provides recognition of 51 context-labels and these can be used by other applications (like context-aware music streaming, healthy lifestyle monitoring, *etc.*).

   The app is publicly available for free at
   http://extrasensory.ucsd.edu/ExtraSensoryApp. The full source code (Android phone code, Pebble watch code, and server-side code) is available. Researchers can download it, install the server-side on their own server and deploy the app with study participants. Researchers are free to adjust the code and suggest improvements. They can also train their own

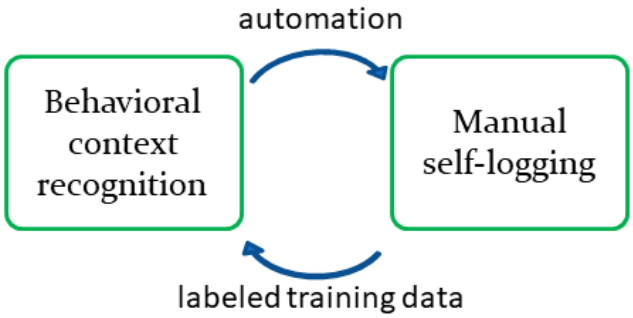

Figure 1.4: Automation feedback-loop for data collection. Data that was collected with manual labeling by participants is used to train better context-recognition systems (better classifiers). The improved classifier can be plugged-in to the server-side of the ExtraSensory App and provide better real-time recognition. This improved automated recognition, in turn, will make it easier for the next generation of participants to provide more labeled data. This cycle will go on supporting itself.

classifiers (*e.g.* by using the ExtraSensory Dataset) and plug them into their deployment's server.

The ExtraSensory Dataset and the ExtraSensory App will server the research community in promoting development of behavioral context recognition systems, and specifically addressing data collection and application in-the-wild. These two components can help jump-start a longer feedback-loop that will improve context-recognition, alternating between improving the server-side real-time classifier and collecting more data (see Figure 1.4).

Chapter 2

Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches

Abstract

The ability to automatically recognize a person's behavioral context can contribute to health monitoring, aging care and many other domains. Validating context recognition *in-the-wild* is crucial to promote practical applications that work in real-life settings. We collected over 300k minutes of sensor data with context labels from 60 subjects. Unlike previous studies, our subjects used their own personal phone, in any way that was convenient to them, and engaged in their routine in their natural environments. Unscripted behavior and unconstrained phone usage resulted in situations that are harder to recognize. We demonstrate how fusion of multi-modal sensors is important for resolving such cases. We present a baseline system, and encourage researchers to use our public dataset to compare methods and improve context recognition in-the-wild.

2.1 Introduction

The ability to automatically recognize a person's context (*i.e.*, where they are, what they are doing, who they are with, *etc.*) is greatly beneficial in many domains. Health monitoring and lifestyle interventions have traditionally been based on manual, subjective reporting [80], sometimes by end-of-day recalling [5]. These can improve with automatic (frequent, effortless, and objective) detection of behaviors like exercise, eating, sleeping, or mental states like stress. Just-in-time interventions (*e.g.* for addiction) often prompt the patient at arbitrary times of the day, possibly missing times when the patient needs support the most [61]. Automatically recognizing context will help detect critical times and offer immediate support (*e.g.* an alcoholic patient may be in high risk of craving or lapse when the context is "at a bar, with friends").

The biomedical research community acknowledges the effects of behavior, lifestyle and environment on health, disease and treatments [32]. Automatic context recognition tools will be essential to incorporate behavioral and exposure aspects into large scale studies and to tailor

appropriate treatment for patients. The range of measured exposures should be broad and cover diverse life style and environmental aspects. Commercial tools that offer superficial recognition (*e.g.* of walking, running, and driving) will not suffice. Personal assistant systems can adjust to context and better serve the user. Aging care programs can use automated logging of older adults' behavior to detect early signs of cognitive impairment, monitor functional independence, and support aging at home [45].

In order for such applications to succeed in large scale, the context recognition component has to be unobtrusive and to work smoothly, without requiring the person to adjust their behavior. It is important that research emulates real-world settings, where such applications will eventually be deployed. In this work we promote context recognition *in-the-wild*, meaning capturing people's authentic behavior in their natural environments, with the use of every-day devices — smartphones and smartwatches. We address the difficulty that in-the-wild conditions add, and show how multi-modal sensors can help.

2.1.1 Related work

It is common for people to have their phone close to them most of the time [17]. This growing trend, and the variety of built-in sensors, make phones popular agents for recognizing human behavior. Smartwatches are a useful sensing addition. While capturing informative signals about hand and arm motion, they remain very natural to wear and don't add any burden to the user.

Previous works have shown the advantage of fusing sensors of different modalities, from smartphones and smartwatches, to improve recognition of basic movement activities [26] and more complex activities, like smoking or drinking coffee [82]. However, most past works have collected data under heavily controlled conditions, with researchers instructing subjects to perform scripted tasks. Fitting models to recognize prescribed activities may result in poor generalization to real life scenarios [36]. To promote real-life working applications, we argue

that research has to be done in natural and realistic settings, satisfying four *in-the-wild* conditions:

1. Naturally used devices. Introducing a foreign device to the user adds a burden and harms natural behavior. Ideally, subjects would use their own phone, and possibly additional convenient devices, like watches.

2. Free device placement. It has been shown that the placement and orientation of sensors have a great influence on the success of recognition [40]. However, this does not mean we should avoid this difficulty by forcing specific placement — a practical real-world application cannot restrict users to keep their phone in pant pocket for the recognition to work. Instead, research should address the variability in device placement as a challenge, and provide solutions to overcome it.

3. Natural environment. The recorded behavior should be in the subjects' natural environment and on their own free time. They should not be instructed where or when to perform their activity.

4. Natural behavioral content. In many works the researchers instructed subjects to perform scripted tasks [26, 82]. The recorded behavior was then simulated, and not natural. Other works let the subjects behave on their own time, but still prescribed a list of targeted activities [25, 68], which may cause the subject to perform actions they are not used to, like "vacuum cleaning". In-the-wild studies should record the behavior that is natural to each individual subject.

A major challenge is acquiring labels of the behavioral context. Attaining in-the-wild conditions usually trades off with other aspects of the data collection effort, resulting in fewer labeled examples, smaller range of interest labels or compromised privacy of the subjects.

Previous research addressed some aspects of in-the-wild data collection in different ways. Han *et al.* [27] designed a decision-tree architecture that activates predetermined sensors to

differentiate eight ambulatory and transportation states. Such a hand-crafted system is hard to scale to more contexts. They validated their system with an observer that followed a single user. Ordonez *et al.* [64] installed a set of state-change sensors around a home to detect daily home activities. While such sensors are un-obtrusive and maintain natural behavior, the complicated device setup limits the deployment of data collection and practical applications. It also cannot track the person outside of the monitored environment. Dong *et al.* [18] targeted eating periods and used an unnatural setup of having a smartphone bound to the wrist. Subjects had to mark start times of eating, and after data collection they reviewed and corrected their markings. This resulted in 449 hours of data with 116 eating periods from 43 subjects. Rahman *et al.* [72] compared different approaches for subjects to self-report their stress level (immediately or by recalling later). They suggested a compromise approach where the subject can report on their own time but with the help of cues (like location or surrounding sound level) to remember how they felt at specific times of the day.

Choudhury *et al.* [14] designed a system to address the requirements for a practical context recognition system, including unobtrusive lightweight devices, long battery life and multi-modal sensing. However, most of their validation was done on controlled data, collected in specific locations, with constrained positioning of device, and with a sequence of 8 activities that was scripted, observed, and repeated by 12 subjects. Consolvo *et al.* [15] utilized the same system (trained on the controlled data) in a field study of an application to promote physical activity. The mobile app used a combination of the automated recognition (of walking, cycling, *etc.*) and manual editing of a daily journal.

Hemminki *et al.* [29] targeted detecting transportation modes and specifically designed features that would be less sensitive to placement of the phone. Ganti *et al.* [25] gave eight subjects a Nokia N95 phone for a period of eight weeks and asked them to go about their regular routine and use the phone for recording whenever they can, in any location or time-of-day. The phone was constrained to be in the pocket or pouch. The interface allowed selecting an

activity from a set of eight activities and marking when you start and when you finish. They collected a total of 80 hours. Khan *et al.* [37] targeted recognition of 15 activities. To collect measurements and annotations, they handed a NEXUS phone to subjects for a month. Subjects were free to perform the prescribed activities on their own time and they used the phone to mark the beginning and end of the selected activity. They collected about 3000 examples per activity from 30 subjects, plus a follow-up validation with eight subjects using the trained real-time recognition system.

In Yatani *et al.*'s [102] out-of-lab study, five subject wore a phone around their neck to take egocentric snapshots that were later used to label the activity. A similar label acquisition strategy was taken in large scale by Ellis *et al.* [20] with 40 subjects who recorded hip-mounted accelerometer and GPS data from routine behavior in natural environment for several days. The subjects wore a SenseCam device around their neck, which took snapshots periodically, and the thousands of images were later used by research assistants to annotate the activity. Pirsavash *et al.*[68] used a GoPro video camera for both sensor measurements and ground truth labels. The subjects wore the device around the chest in a single morning at their own home, and were prescribed a list of home activities to perform with no extra specifications. They recorded over 10 hours of video from 12 people and later annotated household objects and activities for about 30k frames (every second). Their dataset is publicly available. While the camera approach may generate more reliable labels in certain cases, the unnatural and uncomfortable equipment compromises natural behavior. Furthermore, offline annotation of images is costly, making it hard to scale, and violates the privacy of the subjects and people around them. The alternative of self reporting has the advantage of collecting labels when a camera is not present (*e.g.* "shower"), when the context is not clearly visible in the image (*e.g.* "singing"), or when the subject knows best what is happening (*e.g.* "with family" *vs.* "with friends").

2.1.2 Our work

In this work we use smartphone and smartwatch sensors to recognize detailed situations of people in their natural behavior. We collected labeled data from over 300k minutes from 60 subjects. Every minute has multi-sensor measurements and is annotated with relevant context labels. To the best of our knowledge, this dataset, which is publicly available, is far larger in scale compared to others collected in the field. Similar to [18, 25, 37], we rely on self-report. Unlike those works, our data collection app offers an extensive menu of over 100 context labels and the ability to select combinations of relevant labels. This facilitates natural behavior from the subjects, *e.g.* they are free to "run on a treadmill", while "watching TV" if it is natural to them (in [25, 37], subjects were forced to choose one activity, possibly causing them to act unnaturally). This also provides rich descriptions of context, as combinations of different aspects, like environment, activities, company, body posture. Similar combinatorial representations were previously used to describe objects and actions in images [68] and locations, objects, humans, and animals in sound clips [76]. In those cases annotation was done offline, but in our case, attaining detailed labeling by self-reporting requires attention and effort from the subjects. To mitigate it, our app's interface offers many reporting-mechanisms to minimize interaction time. Subjects can report the start of activity (as in [18, 25, 37]). They can manually edit events in a daily calendar that included automatically recognized contexts (similar to [15]). We treat only the manual corrections or additions as ground truth; the automated predictions act as cues, to help the subjects recall their context (as suggested in [72]).

The main contribution of this work is the emphasis on *in-the-wild* conditions, as mentioned in "previous work":

1. Naturally used devices. Subjects used their own personal phones, and a smartwatch that we provided.

2. Free device placement. Subjects were free to carry their phone in any way that was

convenient to them.

3. Natural environment. Subjects collected data in their own regular environment for about a week.

4. Natural behavioral content. No script or tasks were given. We did not target a specific set of activities. Instead, the context labels that we analyze came *from the data*, as the subjects engaged in their routine and applied any relevant labels (from the large menu) that fit what they were doing.

Recognizing context in-the-wild is more challenging, compared to controlled conditions, because of the large variability in real-life. Diversity in phone devices and sensor hardware has an effect on the measurements [85]. Our data represents both iPhone and Android, including many varieties of devices. Variability in behavioral content is clearly visible in the ground truth labels of out data, including combinations like {Running, Outside, Exercise, Talking, With friends}, {Running, Indoors, Exercise, At the gym, Phone on table}, {Sitting, Indoors, At home, Watching TV, Eating, Phone on table}, {Sitting, At a restaurant, Drinking (alcohol), Talking, Eating}, {Sitting, On a bus, Phone in pocket, Talking, With friends}, {On a bus, Standing}. Such variability was missed in works that defined behavior with a small set of mutually exclusive activities. Variability in manner or style (*e.g.* different gaits) is less visible, but is still captured in our sensor measurements. Such variability can easily be missed in scripted experiments or if restricting how to use devices. Our analysis demonstrates the difficulty in resolving context in-the-wild, and the importance of using complementary sensing modalities. We show that everyday devices, in their natural usage, can capture information about a wide range of behavioral attributes.

2.2 Context recognition system

Figure 2.1 illustrates the flow of our recognition system. The system is based on measurements from five sensors in a smartphone: accelerometer (Acc), gyroscope (Gyro), location (Loc), audio (Aud), and phone state (PS), as well as accelerometer measurements from a smartwatch (WAcc). For a given minute, the system samples measurements from these six sensors and the task is to detect the combination of relevant context labels (Figure 2.1 (A)), *i.e.* declare for each label l a binary decision: $y_l = 1$ (the label is relevant to this minute) or $y_l = 0$ (not relevant).

For this paper we opted for simple computational methods, based on linear classifiers and basic heuristics for sensor fusion. We model each label separately and treat every minute as an independent example. We include time-of-day as part of the PS features, but we do not model the behavioral time-series throughout the day. The goal of this paper is to show the potential of context recognition *in-the-wild*, and to establish a baseline. Future papers will use non-linear methods, dynamic-context models, and interaction among labels.

Single-sensor classifiers use sensor-specific features and help us understand how informative each sensor can be, independently of the other sensors, for a given context label (Figure 2.1 (B)). We use logistic regression — a linear classifier that outputs a continuous value (interpreted as probability) in addition to the binary decision. This is helpful for sensor fusion. The following procedure was performed for a given sensor s and a given label l: (a) For each example, compute a d_s-dimensional feature vector x_s. Each sensor has a different set of relevant features. (b) Standardize each feature by subtracting mean and dividing by standard deviation (these statistics are estimated on the training set). (c) Learn a d_s-dimensional logistic regression classifier from the training set. (d) Apply the logistic regression classifier to a test example to obtain a binary classification y_l and probability value $P(y_l = 1|x_s)$. To overcome the imbalance between the positive class and the negative class we applied balanced class weights (inversely proportional to the class frequency in the training set).

At this point, it is possible introduce some domain knowledge and assign appropriate sensors to certain labels. For example, the watch accelerometer can be a good indicator for specific hand-motion activities, like "washing dishes", while audio might better predict environmental contexts like "in class" or "at a party". These design decision are not always obvious, so we continue with sensor-fusion methods that can learn the best predictors from data.

2.2.1 Sensor fusion

Our system further combines information from N different sensors. We propose three alternative ways.

Early fusion (EF) classifiers combine information from multiple sensors prior to the classification stage (Figure 2.1 (C)). The following procedure was performed for a given label l: (a) Start with the sensor-specific feature vectors $\{x_s\}_{s=1}^{N}$. (b) Concatenate the (standardized) sensor-specific feature vectors into a single vector x of dimension $d = \sum_{s=1}^{N} d_s$ (c) Learn a d-dimensional logistic regression classifier from the training set. (d) Apply the logistic regression classifier to a test example to obtain a binary classification y_l and probability value $P(y_l = 1|x)$.

Late fusion classifiers. We use ensemble methods to combine the predictions of the N single-sensor classifiers. We chose to combine the probability outputs $P(y_l = 1|x_s)$, and not the binary decisions, to take into account the "confidence" of each of the N classifiers and avoid over-influence of irrelevant sensors. We explore two methods for late fusion:

Late fusion using average probability (LFA) (Figure 2.1 (D)) applies a simple bagging heuristic and averages the probability values from all the single-sensor classifiers to obtain a final "probability" value, *i.e.*, $P(y_l = 1|x_1, x_2, \ldots, x_N) = \frac{1}{N}\sum_{s=1}^{N} P(y_l = 1|x_s)$. LFA declares "yes" if the average probability is larger than half. No additional training is performed after the single-sensor classifiers are learned. This method grants equal weight to each sensor, hoping that informative sensors will classify with higher confidence (probability close to 0 or close to 1) and will influence the final decision more than irrelevant sensors (which will hopefully predict with probability

close to 0.5).

As mentioned earlier, some sensors may be consistently better suited for certain labels. As a flexible alternative to deciding apriori how to assign sensors to labels, we can let sensor-weights be learned from data. **Late fusion using learned weights (LFL)** (Figure 2.1 (E)) is a second type of late fusion that places varying weight on each sensor. This method learns a second layer of N-dimensional logistic regression model. The second layer's input is the N probability outputs of the single-sensor models, and the output is a final decision y_l.

2.3 Data collection

For the purpose of large-scale data collection, we developed a mobile application (app) called *ExtraSensory*, with versions for both iPhone and Android smartphones, and a companion application for the Pebble smartwatch that integrates with both. The app was used to collect both sensor measurements and ground truth context labels. Every minute the app records a 20sec window of sensor measurements from the phone and watch. Within that window, the time samples of different sensors are not guaranteed to be exactly aligned. The flexible user interface provided the user with many mechanisms to self-report the relevant context labels and cover long behavioral time with minimal effort and time of interaction with the app (Figure 2.2). On the history tab (Figure 2.2 (A)), the user can see a daily log of activities and contexts. The server sends real-time body-state predictions (based on preliminary training data from two iPhone users — the researchers). These predictions appear with question marks and help the user organize the log and recall when their activity may have changed. The user can update the history records' labels, add secondary labels like "at home" and "eating", merge consecutive records into a longer period, and split records. In addition, the label selection menu is indexed by topics and a "frequently used" link to make it easier for the user to select quickly (Figure 2.2 (B)). Using the "active feedback" button the user can report they will be engaged in a specific context starting

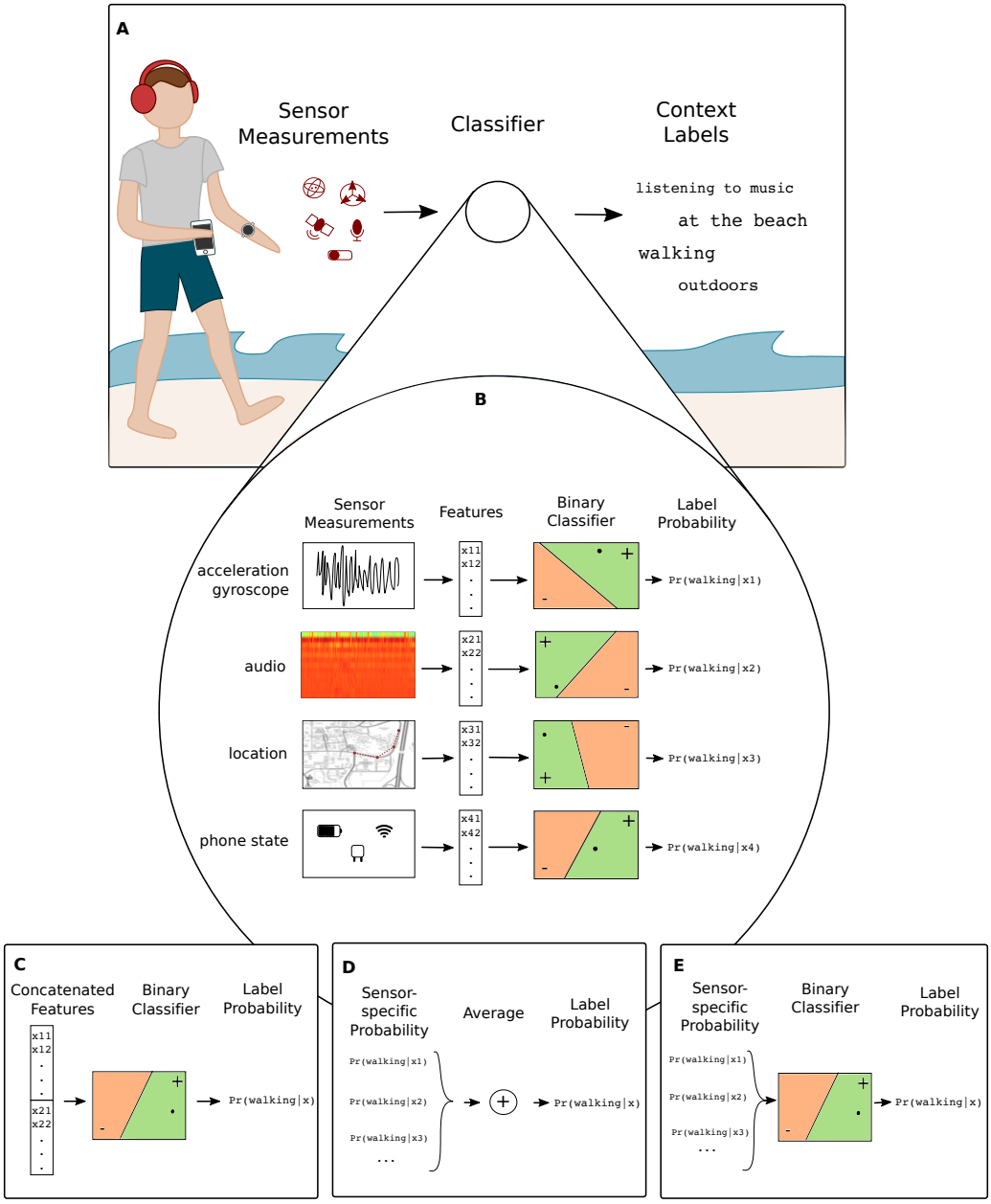

Figure 2.1: Context recognition system. (A) While the person is engaged in natural behavior the system uses sensor measurements from the smartphone and smartwatch to automatically recognize the person's detailed context. (B) Single-sensor classifiers. Appropriate features are extracted from each sensor. For a given context label, classification can be done based on each sensor independently. (C) Under Early fusion (EF), features from multiple sensors are concatenated to form a long feature vector. (D) Late fusion using averaging (LFA) simply averages the output probabilities of the single-sensor classifiers. (E) Late fusion with learned weights (LFL) learns how much to "listen" to each sensor when making the final classification.

immediately and valid for a set period of time (Figure 2.2 (C)). Periodic notifications remind the user to provide labels (Figure 2.2 (D)). If the user is engaged in the same context as they recently reported they simply need to reply "correct" to the question. If any element of the context has changed they can press "not exactly" and be directed to a screen where they can update the labels of the recent time period. These notifications appear on the watch as well, which enables easier responses.

Sixty subjects (users) were recruited using fliers posted around the UC San Diego campus and campus-based email lists. 34 of the subjects were iPhone users, with iPhone devices of generations from iPhone4 to iPhone6 and with operating system (iOS) versions 7, 8 and 9. 26 subjects were Android users, with various devices (Samsung, Nexus, Motorola, Sony, HTC, Amazon Fire-Phone, and Plus-One). The subjects were from diverse ethnic backgrounds (self-defined), including Chinese, Mexican, Indian, Caucasian, African-American, and more. The majority of the subjects (93%) were right handed, and chose to wear the smartwatch on their left wrist. The dataset is homogeneous with regard to occupation; almost all the subjects were students or research assistants. 34 subjects were female and 26 were male. Table 2.1 describes additional subject characteristics. We installed the app on each subject's personal phone and provided the watch to the subject (56 out of the 60 agreed to wear the watch). The subject then engaged in their usual behaviors for approximately a week, while keeping the app running in the background on their phone as much as possible and convenient. The subject was asked to report as many labels as possible without interfering too much with their natural behavior. They were free to remove the watch whenever they wanted and were asked to turn off the watch-app when they were not wearing it. Basic compensation of $40 was given to each subject, with additional incentive of up to $35 that depended on the amount of labeled data they provided.

The resulted *ExtraSensory Dataset*, contains a total of 308,320 labeled examples (minutes) from sixty users. Table 2.1 details statistics (over 60 subjects) about the amount of data collected. Not all the sensors were available at all times. Table 2.2 specifies details on the sensors.

The dataset is publicly available and researchers are encouraged to use it for developing and comparing context recognition methods: http://extrasensory.ucsd.edu.

Table 2.1: Statistics over the 60 users in the dataset (SD: standard deviation).

	Range	Mean (SD)
Age (years)	18–42	24.7 (5.6)
Height (cm)	145–188	171 (9)
Weight (kg)	50–93	66 (11)
Body mass index (kg/m^2)	18–32	23 (3)
Labeled examples	685–9706	5139 (2332)
Additional unlabeled examples	2–6218	1150 (1246)
Average applied labels per example	1.1–9.7	3.8 (1.4)
Participation duration (days)	2.9–28.1	7.6 (3.2)

Table 2.2: The sensors in the dataset. For each sensor, details of the raw measurements, the number of examples with measurements from that sensor and the number of users with measurements from that sensor. "Core" represents examples that have measurements from all six core sensors that are analyzed in this paper (Acc, Gyro, WAcc, Loc, Aud and PS). "1pe" means sampled once per example. "var" means variable sampling rate — gathering updates whenever the value changes.

Sensor	raw measurements	examples	users
Accelerometer	3-axis (40Hz)	308,306	60
Gyroscope	3-axis (40Hz)	291,883	57
Magnetometer	3-axis (40Hz)	282,527	58
Watch Accelerometer	3-axis (25Hz)	210,716	56
Watch Compass	heading angle (var)	126,781	53
Location	long-lat-alt (var)	273,737	58
Location precomputed	location variability (1pe)	263,899	58
Audio	13MFCC (46ms frames)	302,177	60
Audio power	1pe	303,877	60
Phone State	1pe	308,320	60
Low frequency sensors	1pe	308,312	60
Core		176,941	51

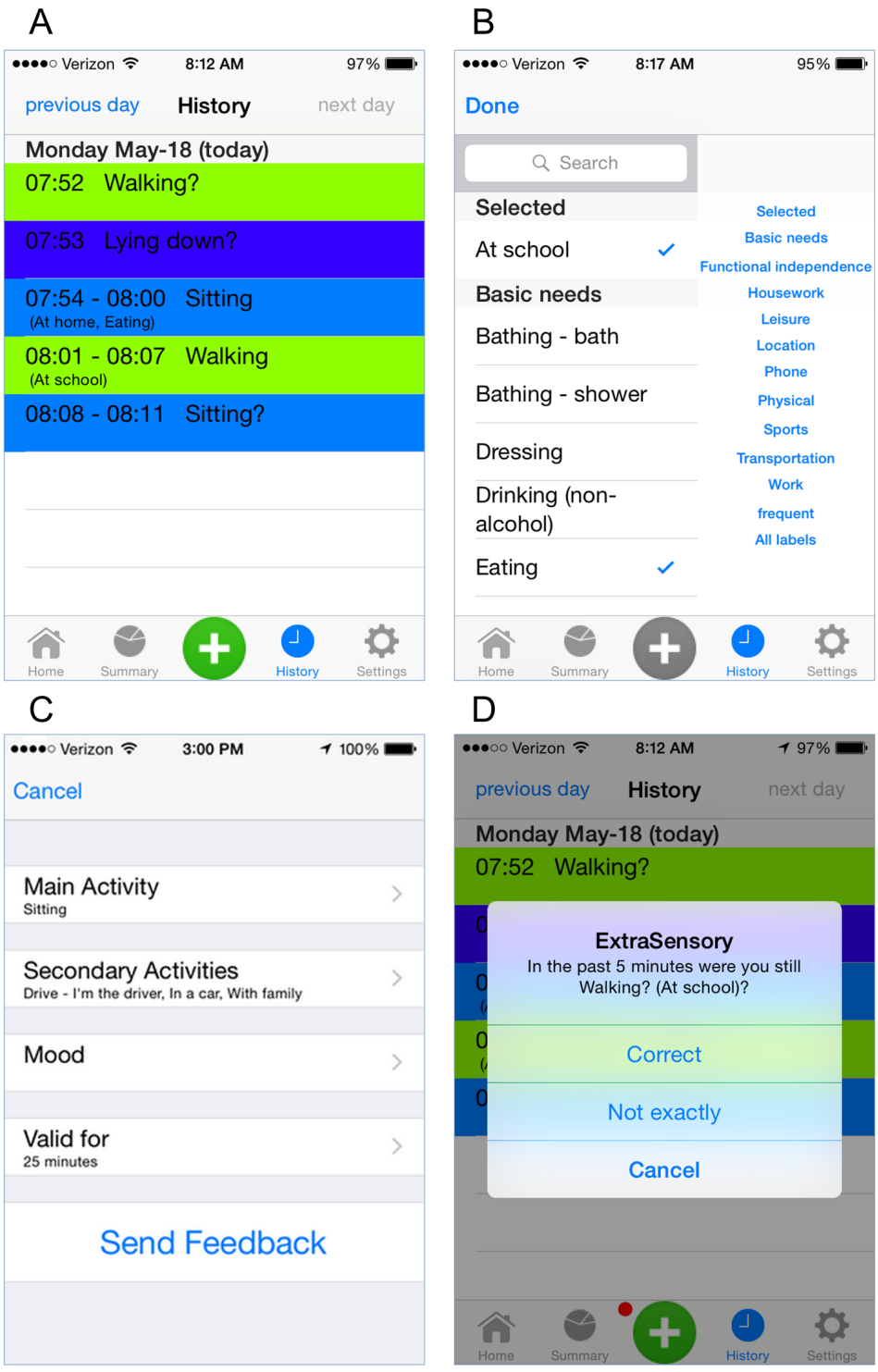

Figure 2.2: Screenshots from the ExtraSensory mobile application (iPhone version). (A) History page. (B) Selecting labels from a menu. (C) Active feedback. (D) Notification.

2.4 Evaluation and results

We evaluated classification performance using five-fold cross validation: each fold has 48 users in the training set and the other twelve users in the test set. We also conducted leave-one-user-out experiments (LOO). To measure performance, classification accuracy is a misleading metric because of imbalanced data; for a rare label that appears in 1% of the test set, a trivial classifier that always declares "no" will achieve 99% accuracy but is completely useless. It is important to consider competing metrics, like sensitivity and specificity. A common approach is to observe sensitivity (recall) against precision, or to calculate their harmonic mean (F1). However, precision and F1 are less fitting, since they are very sensitive to how rare labels are. Chance level can be arbitrarily small, and when averaging precision or F1 over many labels, certain labels will unfairly dominate the score. Additionally, the self-reported data may be noisy, possibly including cases where a label was actually relevant, but was not reported by the subject. Precision and F1 will be too sensitive to such cases. Unlike F1, the balanced accuracy, BA=0.5*(sensitivity+specificity), does not suffer from these issues, and can serve as a convenient objective that fairly balances competing metrics.

First, we assess the potential of single sensors. Figure 2.3 (A) shows some specific context labels for which relatively few examples were collected. If we pick our first (sometimes second) guess of relevant sensor we can achieve reasonable recognition of these contexts.

Next, to see if we can do better, we evaluate the three sensor-fusion methods described in "Sensor fusion" and compare them to single-sensor classifiers. Figure 2.3 (B) shows performance for 25 labels from diverse context domains. In most cases sensor-fusion managed to match the best fitting single-sensor. The system learned from data how to best utilize the different sensors, without the need of a human to guide it, which can be useful for scalable systems, where the researcher does not necessarily know which sensor to trust for which label. Furthermore, in many cases sensor-fusion improved performance, compared to the best single-sensor, meaning

that there is complementary information in different sensors. We see the overall advantage of multi-sensor systems over single-sensor systems, shown by the average performance of the different systems in Figure 2.3 (C). The three sensor-fusion alternatives seem to perform similarly well, with LFL slightly ahead. The selection of a sensor-fusion method can be guided by the training data available to the researcher. When having plenty of labeled examples that have all six sensors available, the simple EF system can work. Otherwise, late fusion will be more fitting, still having plenty of data to train each single-sensor classifier alone. Leave-one-user-out results are consistent with 5-fold evaluation (figure 2.3 shows LOO results for the EF system, marked "EF-LOO"). For some labels, like "running", the system benefited from the larger training set in the LOO evaluation. Full per-label results are provided in supplemental material.

2.4.1 Why does sensor fusion help?

The performance of single-sensor classifiers on selected labels (Figure 2.4 (A)) demonstrates the advantage of having sensors of different modalities. As expected, for detecting sleeping, the watch is more informative than the phone's motion sensors (Acc, Gyro) — during sleep the phone may be lying motionless on a nightstand, while the watch records wrist movements. Similarly, contexts such as "shower" or "in a meeting" have unique acoustic signatures (running water, voices) that allow the audio-based classifier to perform well. When showering, it is reasonable that the phone will be in a different room than the person, in which case the watch is an important indicator of the activity. Figure 2.4 (A) demonstrates that the LFL method assigns reasonable weights to the six sensors — sensors that perform more strongly for a given label are given higher weight.

Investigating where misclassification occurs helps to understand the predictive ability of the system. Figure 2.4 (B–G) shows confusion matrices that depict misclassification rates between related context labels. For example, a classifier using the phone's motion sensors (Acc and Gyro) (Figure 2.4 (B)) to discriminate between body movement/posture states confuses

A

Label	examples	Sensor	BA	Sensor	BA
Stairs - going up	399	Gyro	0.73	Acc	0.70
Stairs - going down	390	Gyro	0.73	Acc	0.71
Elevator	124	Gyro	0.76	Aud	0.71
Cleaning	1839	WAcc	0.71	Gyro	0.64
Laundry	473	WAcc	0.66	Acc	0.65
Washing dishes	851	WAcc	0.70	Aud	0.60
Singing	384	Aud	0.68	Loc	0.61
At a party	404	Aud	0.81	Acc	0.74
At the beach	122	Loc	0.72	PS	0.70
At a bar	520	PS	0.93	Gyro	0.66

C

	classifier	accuracy	sensitivity	specificity	BA	precision	F1
	p99	0.50	0.50	0.50	0.50	0.11	0.13
	Acc	0.73	0.64	0.73	0.68	0.17	0.22
	Gyro	0.70	0.64	0.69	0.66	0.16	0.20
	WAcc	0.73	0.67	0.72	0.70	0.18	0.22
	Loc	0.71	0.63	0.70	0.67	0.17	0.22
	Aud	0.75	0.65	0.75	0.70	0.18	0.22
	PS	0.76	0.74	0.76	0.75	0.20	0.24
	EF	0.87	0.67	0.87	0.77	0.24	0.30
	LFA	0.84	0.76	0.83	0.80	0.23	0.29
	LFL	0.85	0.76	0.85	0.80	0.24	0.30
	EF-LOO	0.86	0.69	0.86	0.78	0.24	0.30

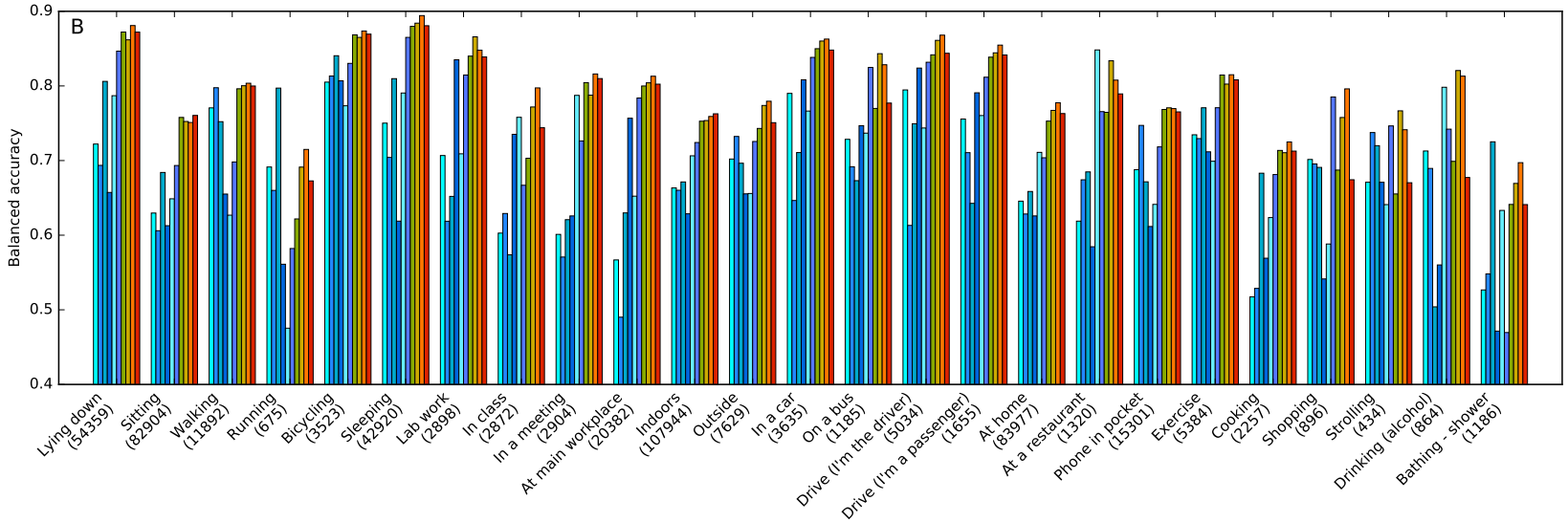

Figure 2.3: Overall performance of the single-sensor classifiers (Acc, Gyro, WAcc, Loc, Aud and PS) and the sensor-fusion classifiers (EF, LFA, LFL and EF-LOO). (A) Specific labels that had few examples with first and second guess of sensors that intuitively seem relevant and the BA score of the corresponding single-sensor classifiers. (B) BA scores for selected labels from diverse domains (number of examples in parenthesis). Color legend is in table (C). (C) Average performance metrics over the 25 context labels from B. All average scores were well above the p99 value, which marks the 99th percentile of random score — scores above the p99 value have less than 1% probability of being achieved randomly (p99 was estimated from 100 random simulations).

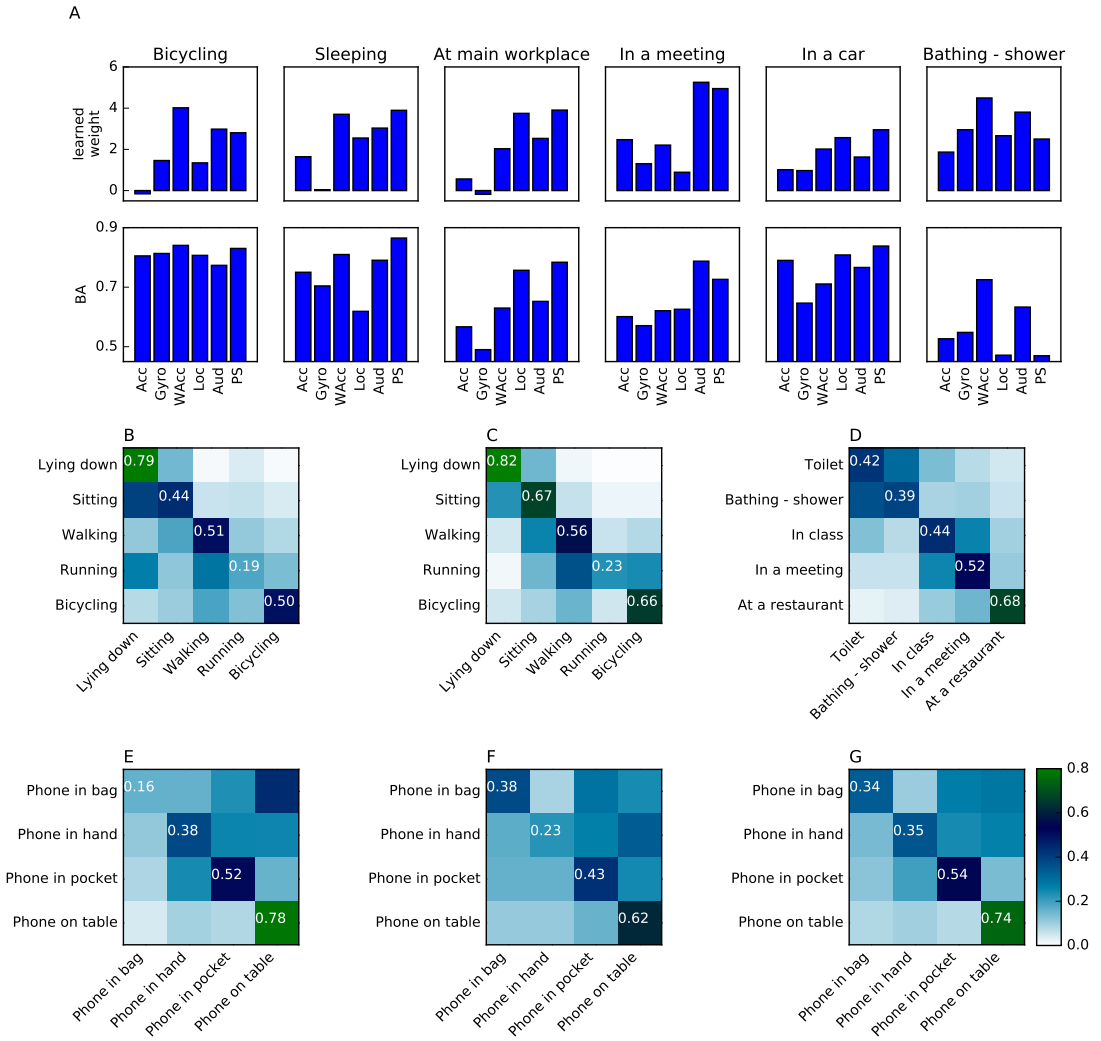

Figure 2.4: Why sensor fusion helps recognition. (A) The bottom row shows the overall performance (BA) of each single-sensor classifier and the top row shows the weights that the LFL classifier learned to assign to each sensor (taken from the first cross validation fold). (B–G) Confusion matrices for subsets of mutually exclusive labels. Rows represent ground truth labels and columns represent predicted labels. Rows are normalized so that a cell in row i and column j displays the proportion of examples of class i that were assigned to class j. The correct classification rates (main diagonal) are also marked numerically. The sensors used are: (B) Acc and Gyro; (C) Acc, Gyro and WAcc; (D) Aud; (E) Acc, Gyro and WAcc; (F) Aud; (G) Acc, Gyro, WAcc and Aud.

even dissimilar labels ("running" *vs.* "lying down"). Such errors arise in natural, unconstrained behavior; in-the-wild, people do not always carry their phone in their pocket — subjects were sometimes running on a treadmill with their phone next to them, motionless. The watch can help in such situations — when the watch accelerometer features were added to the classifier (Figure 2.4 (C)), the confusion between activities was reduced.

The audio signal from the smartphone (Figure 2.4 (D)) is informative for labels related to the environmental context. We see a hierarchy of misclassification: while there is some confusion between labels that share similar acoustic properties ("toilet" *vs.* "shower", "class" *vs.* "meeting"), there is a sharper distinction between label groups from different domains ("toilet or shower" *vs.* "class or meeting" *vs.* "restaurant").

The phone placement itself provides cues about the user's activity; when the phone is lying on a table it is more likely the user is showering than walking to work. The ability to recognize the phone's position will improve overall context recognition. A single modality is not sufficient to fully identify phone position. A classifier based on motion sensors is sensitive to movement, so when the phone was in a bag (possibly motionless) it was often mistaken for being on a table (Figure 2.4 (E)). On the other hand, a classifier using audio alone is more sensitive to whether the phone is enclosed or exposed to environmental sounds, so with this classifier cases of "phone in bag" were mistaken for "phone in pocket" and "phone in hand" was often mislabeled as "phone on table" (Figure 2.4 (F)). However, by combining motion and audio modalities, the classifier synthesized these two dimensions of discrimination to better recognize phone position (Figure 2.4 (G)). These examples demonstrate the large variability in behavior in-the-wild and highlight the utility of fusing multi-modal sensors.

2.5 User personalization

People move, behave and uses their phone in different manners. A system that is fine-tuned to its specific user may outperform a more general model. To explore the potential of personalization we performed experiments with a single test user. We compared three models: (1) universal (trained on data from other users), (2) individual (trained on half of the data from the same test user) and (3) adapted (merges both). We tested the three models on the same unseen data.

Figure 2.5 shows the results of these experiments. The universal model demonstrates good performance. This shows the basic ability of a trained system to work well for a new unseen user. As suspected, the individual model performed better than the universal model for labels that had many individual examples ("lying down", "sitting", "sleeping", "at home", "computer work", and "at main workplace"). However, the individual user is missing data for many context labels. For other labels there are only a limited number of examples a new user can acquire in a few "training" days, which risks over-fitting to these few examples. In such cases a universal model is better, having been trained on plenty of data from many users. The optimal solution is to benefit from both universal and individual data: the user-adapted model shows overall improvement in recognition performance, even among the labels that had over 300 examples for the test user. LFA is a simple heuristic that manages to demonstrate this advantage. For each label, when there is not enough data to train an individual model, the adapted model relies only on the universal model. When there is enough data to train an individual model, the adapted model "listens" to the universal and individual models, in some cases achieving better performance than either model on its own (*e.g.* "sleeping", "at home").

In practical systems, the logistics of implementing personalization may not be an obvious task. For medical applications, the clinician or patient may decide that their cause is important and worth dedicating some effort to provide individual labeled data for a few days, in order

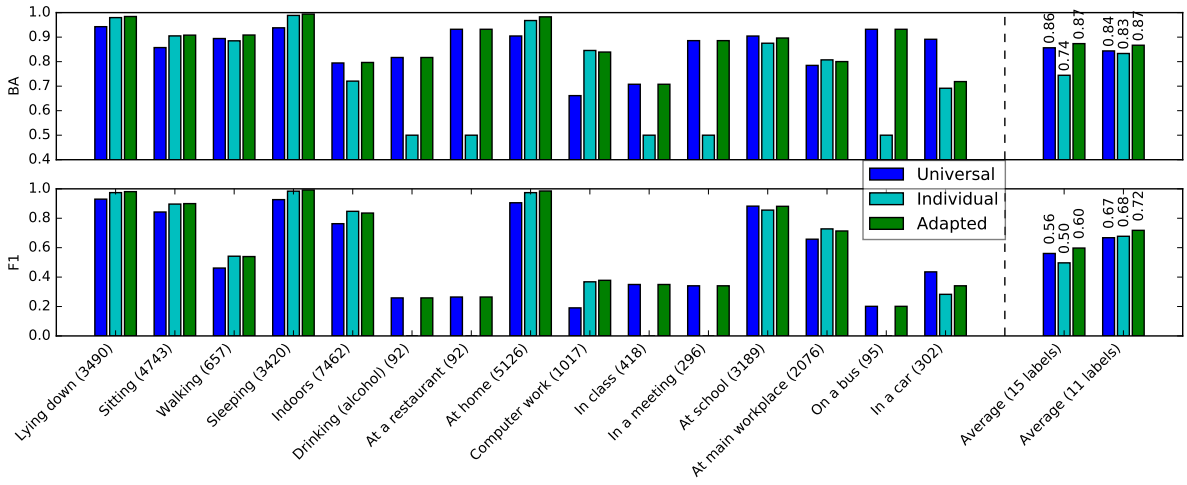

Figure 2.5: User adaptation performance. Balanced accuracy and F1 score were assessed for a single user. The x-axis denotes the labels with the total number of examples for the user. The "universal" model was trained on data from other users, the "individual" model was trained on data from the same user and the "adapted" model combines the universal and individual models using LFA. The bars on the right hand side of each plot present the scores averaged over the 15 tested labels, and averaged over 11 labels that had over 300 examples.

to better adapt the model. However, in commercial applications the users (clients) may not be motivated to invest the extra effort in labeling. In such cases semi-supervised methods can still be used to make the most of unlabeled data from the individual user and personalize the model.

2.6 Conclusions

Our novel data collection brings about behavioral variability in-the-wild that is under-represented in controlled studies. This makes context recognition a harder challenge compared to previous scenarios, hence accuracy levels in-the-wild are lower than those reported in experiments that had some restrictions on behavioral conditions. We demonstrate that everyday devices, in their normal unconstrained usage, carry information about the person's natural behavioral context. We describe a baseline system, suggest three simple methods for sensor fusion, and reinforce previous findings that showed the advantage of fusing multi-modal sensors. We demonstrate how the sensing modalities complement each other, and help resolve contexts that

arise with uncontrolled behavior (*e.g.* running on treadmill with phone on table, motionless).

Combinatorial representation of behavior is very flexible. A well trained system has the potential to correctly recognize a new specific situation (combination of labels) that did not appear in the training. To broaden the range of contexts, researchers can either use supervised methods and focus on newly added target labels when collecting extra data, or use unsupervised methods to discover complex behaviors as common combinations or sequences of simpler contexts [79]. The labels in our work were interpreted in a subjective manner. The same location may be considered as "school" for one subject and "workplace" for another. We did not tell the subjects how we define "walking" or "eating" in order to capture the full scope of what people consider eating. Domain-expert researchers may decide to define labels clearly to subjects or use more specific labels like "eating a meal" and "snacking".

2.7 Future directions

New technologies and original solutions for collecting labels in-the-wild are required to reduce annotation load from study subjects and increase reliability of labeling. Online learning can be used to keep improving real-time recognition, which will require less label-correcting effort from new research subjects. Active learning can be utilized to collect data in scale, while sparsely probing subjects to provide annotations. In parallel, semi-supervised methods can be used to make the best out of plenty unlabeled data (which is easy to collect) and reduce the dependence on labeled examples to a minimum.

The public dataset we collected provides a platform to develop and evaluate these methods, as well as explore feature extraction, inter-label interaction, time series modeling and other directions that will improve context recognition.

2.8 Supplementary material

Supplementary material has technical details about the following components of the work:

- Mobile app
- Data collection procedure
- Sensor measurements
- Extracted features
- Label processing
- Classification methods
- Performance evaluation
- User personalization assessment
- Detailed results tables

2.8.1 Mobile app

For the purpose of data collection in a large scale we developed a mobile application called *ExtraSensory App*, with versions for both iPhone and Android smartphones, and a companion application for the Pebble smartwatch that integrates with both. The app was used for supervised data collection, meaning it collects both sensor measurements and ground truth context labels. The app is scheduled every minute to automatically record measurements for 20 seconds from the sensors. Sensors are sampled in frequencies appropriate for their domain, and include motion-responsive sensors, location services, audio, environment sensors, as well as bits of information about the phone's state. When the watch is available (within Bluetooth range and

paired with the phone) measurements from the watch are also collected by the app during the 20 second recording session. More details about the sensors are provided in "Sensor measurements". At the end of the 20 second recording session the measurements are bundled in a zip file and sent through the internet (if a WiFi network is available) to our lab's server, which runs a quick calculation and replies with an initial prediction of the activity (*e.g.* sitting, walking, running). All communication between the app and the server is secure and encrypted, and identified only by a unique universal identifier (UUID) that was randomly generated for each user.

In addition to collecting sensor measurements, the app's interface provides several mechanisms for the user to report labels about their context. This was a crucial part of the research design and we had to overcome a basic trade-off: on one hand we wanted to collect ground truth labels for as many examples (minutes) as possible and with much detail (combination of all the relevant context labels). On the other hand we did not want the subject to interact with the app every minute to report labels, both because it would be an extreme inconvenience for the subject and because it would impact the natural behavior of the subject and miss the point of collecting data in-the-wild. To balance this trade-off, we designed a flexible interface that helps minimize the user-app interaction time. The following label-reporting mechanisms were included:

- A history journal presents the user's activities chronologically as a calendar and enables the user to easily edit the context labels of time ranges in the past (up to one day in the past). The user can easily merge a sequence of minutes to a single "event" with the same context labels, or split a calendar event to describe a change in context. See Figure 2.2 (A). The real-time predictions from the server assist the user to recall when their activity changed — consecutive minutes with the same prediction from the server are merged to a single item on the history calendar.

- The user can also initiate active feedback by reporting labels describing their context in the immediate future (starting "now" for up to half an hour in the future). See Figure 2.2 (C).

- Every x minutes (by default, x is 10 minutes, but can be set by the user) the app presents a notification to remind the user to provide labels. If the the user has recently provided labels, the notification asks whether the user was still engaged in the same activities — allowing for a quick and easy response if the answer is "yes". See Figure 2.2 (D).

- The notifications also appear on the smartwatch, allowing for an easier response with a click of a button on the watch, without using the phone itself.

- When selecting labels from the menu, a side-bar index allows quick search of the relevant labels, either by categories (*e.g.* sports, work, company) or through a "frequently used labels" menu, which presents labels that the user has applied in the past. The category in which a label was presented in the menu does not matter, and a label can appear under different categories (*e.g.* "skateboarding" appears under "sports", "leisure" and "transportation") — the only reason for these categories is to make it easy for the user to find the relevant label quickly. See Figure 2.2 (B).

2.8.2 Data collection procedure

The study's research plan and consent form were approved by the university's institutional review board (IRB). Human subjects were recruited for the study via fliers across campus, university mailing lists and word of mouth. Every subject read and signed the consent form. The researchers installed the app on each subject's personal phone (to maximize authenticity of natural behavior). The subject then engaged in their usual behaviors for approximately a week, while keeping the app running in the background on their phone as much as was possible and convenient. The subject was asked to report as many labels as possible without interfering too much with their natural behavior. Subjects varied in their level of rigorousness with respect to providing labels: some wanted to be very precise (with specific detailed combinations of labels, and trying to keep minute-to-minute precision) and others tended to be less specific and

to dedicate less effort. The subjects who used the watch, which we supplied them with, were told that it is fine to get it wet (wash hands, shower) but not submerge it (swimming). They were also asked to turn off the watch app whenever they removed the watch from their wrist and to turn it back on when they wore the watch — so we can generally assume that whenever watch measurements are available they were taken from the subject's wrist.

Using the app consumes the phone's battery more quickly than normal. To make up for this, the researchers provided participants with an external portable battery, which provides one extra charge during the day. The researchers also provided the subject with the Pebble smartwatch (56 of the subjects agreed to use the watch). The external battery and the smartwatch were returned at the end of the study. Each subject was compensated for their participation. The basic compensation was in the amount of $40, and an additional amount was calculated based on the amount of labeled data that the subject contributed (as an incentive to encourage reporting many labels). The total compensation per subject was between $40 and $75.

Technical difficulties

During the development of the iPhone app, there were releases of new iOS versions that caused the app to not work well and required us to change the code.

Since subjects used their personal phones, the app had to handle different devices, and in the Android case, different makers. For some of the Android users when we installed the app we noticed it didn't work well. In three cases the workaround was to install a slightly different version of the app that didn't use the gyroscope. After installing the changed app and making sure it works those users began collecting records (without gyroscope measurements).

On top of the dataset's 60 subjects, there were four more subjects that participated and received the basic compensation, but whose data was not included in the dataset. For two of them the app didn't work well on their devices. The other two were too stressed or otherwise occupied during participation, and produced too little and un-reliable labels, so we decided to discard their data.

2.8.3 Sensor measurements

Raw sensor measurements are provided in the publicly available dataset.

High frequency measurements:

Each sensor (and pseudo-sensor) in the following list was sampled at 40Hz during the ∼20 second recording session to produce a time series of ∼800 time points. The sampling relies on the design of the phone's hardware and operating system and the sampling rate was not guaranteed to be accurate (especially for the Android devices). For that reason the time reference of each sample in a time series was also recorded; the differences between consecutive time references were approximately 25 milliseconds.

- Accelerometer. Time series of 3-axis vectors of acceleration according to standard axes of phone devices.
- Gyroscope. Time series of 3-axis vectors of rotation rate around each of the phone's 3 axes.
- Magnetometer. Time series of 3-axis vectors of the magnetic field.
- Processed signals. Both iPhone and Android operating systems provide processed versions of the signals: The raw acceleration is split to the gravity acceleration (estimated direction of gravity at every moment, the magnitude is always 1G) and the user-generated acceleration (subtraction of the gravity signal from the raw acceleration). For the gyroscope the OS has a calibrated version that attempts to remove drift effects. For the magnetometer the OS has an unbiased version that subtracts the estimated bias of the magnetic field created by the device itself.

In this paper we used the raw acceleration signal (which includes the effects of gravity) and the calibrated version of the gyroscope signal. Acceleration is reported in units of G (gravitational

acceleration on the surface of the Earth) on iPhone and in units of m/s^2 on Android. Before extracting features we converted the Android acceleration measurements to units of G.

Watch measurements: From the Pebble smartwatch we collected signals from the two available sensors—accelerometer and compass. Acceleration was sampled at 25Hz and describes a 3-axis vector of acceleration (in units of mG) relative to the watch's axes-system. The compass does not have a constant sampling rate; it was requested to provide an update of the heading whenever a change of more than one degree was detected. The compass takes some time to calibrate before providing measurements, so some examples that have watch acceleration measurements are missing compass measurements.

Location measurements: Both iPhone and Android provide location services (based on a combination of GPS, WiFi and cellular communications). The app samples location data in a non-constant rate, as the location service updates each time a movement is detected. This creates a time series of varying length (sometimes just a single time point in a recording session, sometimes more than 20 points) of location updates. Each update has a relative time reference and the estimated location measurements: latitude coordinate, longitude coordinate, altitude, speed, vertical accuracy and horizontal accuracy (these accuracies describe the range of reasonable error in location). Some of these values may be missing at times (*e.g.* when the phone is in a place with weak signals). In addition to the time series of location updates, the app calculates on the phone some basic heuristic location features: standard deviation of latitude values, standard deviation of longitude values, total change of latitude (last value minus first value), total change of longitude, average absolute latitude derivative and average absolute longitude derivative (as proxy to the speed of the user).

To protect our study subjects' privacy (collected examples with label "at home" that also include the exact location coordinates may reveal the subject's identity) the app has an option to select a location (typically their home) they would like to disguise. For the subjects that opted to use this option, whenever they were within 500 meters of their chosen location, the app would

not send the latitude and longitude coordinates from the current recording (but it would send the other estimated location values such as altitude, speed, as well as the basic heuristic location features).

Low frequency measurements: These measurements were sampled just once in a recording session (approximately once per minute). Some of them describe the phone state (PS): app state (foreground/background), WiFi connectivity status, battery status (charging, discharging), battery level, or phone call status. Other low frequency measurements are taken from sensors built in to the phone, if available: proximity sensor, ambient light, temperature, humidity, air pressure.

Audio data: Audio was recorded from the phone's microphone in 22,050 Hz for the duration of the recording session ($\sim$20 seconds). Audio was not available for recording when the phone was being used for a phone call. In order to maintain the privacy of the subjects, the raw audio recording was not sent to the server. Instead, standard audio processing features (Mel Frequency Cepstral Coefficients (MFCC)) were calculated on the phone itself and only the features were sent to the server. The MFCCs were calculated for half-overlapping windows of 2048 samples, based on 40 Mel scaled frequency bands and 13 cepstral coefficients (including the 0^{th} coefficient). As part of the preprocessing of the recorded audio the raw audio signal was normalized to have maximal magnitude of 1 (dividing by the maximum absolute value of the sound wave). This normalizing factor is also sent as a measurement separately from the calculated MFCC features.

2.8.4 Extracted features

For the experiments in this work we focused on six sensors: accelerometer, gyroscope, watch accelerometer, location, audio and phone state. Other sensors' measurements are available in the public dataset. Every sensor measures different physical or virtual properties and has a different form of raw measurements. Therefore we designed specific features for each sensor. The published dataset includes files with these features pre-computed for all the users.

Accelerometer and Gyroscope (26 features each): Since in natural behavior the phone's position is not controlled we cannot assume it is oriented in a particular way, and it also may be changing its axes-system with respect to the ground (and with respect to the person). For that reason we had little reason to assume that any of the phone's axes will have a particular coherent correspondence to many behavioral patterns, and we extracted most of the features based on the overall magnitude of the signal. We calculated the vector magnitude signal as the euclidean norm of the 3-axis acceleration measurement at each point in time, *i.e.*, $a[t] = \sqrt{a_x[t]^2 + a_y[t]^2 + a_z[t]^2}$. We extracted the following features:

- Nine statistics of the magnitude signal: mean, standard deviation, third moment, fourth moment, 25^{th} percentile, 50^{th} percentile, 75^{th} percentile, value-entropy (entropy calculated from a histogram of quantization of the magnitude values to 20 bins) and time-entropy (entropy calculated from normalizing the magnitude signal and treating it as a probability distribution, which is designed to detect peakiness in time—sudden bursts of magnitude).

- Six spectral features of the magnitude signal: log energies in 5 sub-bands (0–0.5Hz, 0.5–1Hz, 1–3Hz, 3–5Hz, >5Hz) and spectral entropy.

- Two autocorrelation features from the magnitude signal. The average of the magnitude signal (DC component) was subtracted and the autocorrelation function was computed and normalized such that the autocorrelation value at lag 0 will be 1. The highest value after the main lobe was located. The corresponding period (in seconds) was calculated as the dominant periodicity and its normalized autocorrelation value was also extracted.

- Nine statistics of the 3-axis time series: the mean and standard deviation of each axis and the 3 inter-axis correlation coefficients.

Watch accelerometer (46 features): From the watch acceleration we extracted the same 26 features as from the phone accelerometer or gyroscope. Since the watch is positioned in

a more controlled way than the phone (it is firmly fixed to the wrist), its axes have a strong meaning (*e.g.* motion along the x-axis of the watch describes a different kind of movement than motion along the z-axis of the watch). Hence we added 15 more axis-specific features—log energies in the same 5 sub-bands as before, this time calculated for each axis' signal separately. In addition, to account for the changes in watch orientation during the recording we calculated 5 relative-direction features in the following way: we first calculate the cosine-similarity between the acceleration directions of any two time points in the time series (value of 1 meaning same direction, value of -1 meaning opposite directions and value of 0 meaning orthogonal directions). Then we averaged these cosine similarity values in 5 different ranges of time-lag between the compared time points (0–0.5sec, 0.5–1sec, 1–5sec, 5–10sec, >10sec).

Location (17 features): In this work we used location features that were based only on relative locations, and not on absolute latitude/longitude coordinates. This was in order to avoid over-fitting to our location-homogeneous training set that will not generalize well to the outside world (*e.g.*, mistakenly learning that a specific location in the UCSD campus always corresponds to "at work"). Six features were calculated on the phone — this was in order to have some basic location features in cases where the subjects opted to hide their absolute location. These quick features included standard deviation of latitude, standard deviation of longitude, change in latitude (last value minus first value), change in longitude, average absolute value of derivative of latitude and average absolute value of derivative of longitude.

The transmitted location measurements were further processed to extract the following 11 features: number of updates (indicating how much the location changed during the 20 second recording), log of latitude-range (if latitudes were transmitted), log of longitude-range (if longitudes were transmitted), minimum altitude, maximum altitude, minimum speed, maximum speed, best (lowest) vertical accuracy, best (lowest) horizontal accuracy and diameter (maximum distance between two locations in the recording session, in meters).

Audio (26 features): From the time series of 13-dimensional MFCC vectors (typically

around 400 time frames) we calculated the average and standard deviation of each of the 13 coefficients.

Phone State (34 features): For this work we used only the discrete phone state measurements. We represented them with a 26-dimensional one-hot representation—for each property we created a binary indicator for each of the possible values the property can take, plus one indicator denoting missing data. This representation is a redundant coding of the phone state, but it facilitates the use of simple, linear classifiers over this long binary vector representation. The keys were: app state (3 options: active, inactive, background), battery plugged (3 options: AC, USB, wireless), battery state (6 options: unknown, unplugged, not charging, discharging, charging, full), in a phone call (2 options: false, true), ringer mode (3 options: normal, silent no vibrate, silent with vibrate) and WiFi status (3 options: not reachable, reachable via WiFi, reachable via WWAN).

In addition, we added a set of features indicating time-of-day information. We used the timestamp of every example and (using San Diego local time) extracted the hour component (one out of 24 discrete values). In order to get a flexible, useful representation we defined 8 half-overlapping time ranges: midnight-6am, 3am-9am, 6am-midday, 9am-3pm, midday-6pm, 3pm-9pm, 6pm-midnight and 9pm-3am. Then we represented each example's hour with an 8-bit binary value, where 2 bins will be active for 1 relevant time range.

2.8.5 Label processing

Since the labels are obtained by subjects self-reporting their own behaviors, the reliability of annotation is not perfect. In some cases, this was the result of the subject reporting labels some time after the activity had occurred and mis-remembering the exact time. More common are cases where the subject neglected to report labels when relevant activities occurred (perhaps because the subject was distracted, did not have time to specify all the relevant labels, or was not aware of another relevant label in the vocabulary). As part of cleaning the data, we created

adjusted versions for some labels using two methods: based on location data and based on other labels.

Location adjusted labels. We collected absolute location coordinates of the examples that had location measurements (selecting the location update with best horizontal accuracy from each example) and visualized them on a map. This made it easier to correct some labels which were clearly reported incorrectly. In examples without location data the original label was maintained.

- "At the beach". According to the few examples that reported being at the beach we marked areas that should be regarded as beach (and manually verified their validity by viewing them on a map). We then adjusted the label by applying "At the beach" to any example with a location within these areas.

- "At home". For each subject we identified the location of their home (by visualizing on a map all locations of examples where the subject reported being at home) and marked the coordinates of a visual centroid. This was only done when it was clear that we had indeed identified a location of a home. Three subjects reported being at home in two different houses, in which case we marked the two locations as locations of home. Two subjects never reported being at home but it was clear from their location to identify their location of residence. Some subjects had none or very few examples of "at home" with location data, so no home location was noted and their original reported labels were used. To define the adjusted version of the label "at home", whenever a subject's location was within 15 meters of their marked home location (or either of the two marked home locations), the adjusted value was set to "true"; whenever a subject's location was farther than 100 meters from all the subject's marked home locations the adjusted value was set to "false". In other cases (when the location was between 15 and 100 meters from a home location, or when there was no location data available) we retained the subject's originally reported value for "at home". This adjustment removed some obviously false reports of "at home" (*e.g.*,

when the subject was clearly on a drive on a freeway). The adjustment manifested mostly by adding the missing label "at home" to many examples where the subject was clearly at home but failed to report it.

- "At main workplace". Similarly to home label we identified for each subject (if they used the original label "at work") the centroid location of their main workplace and created a new label — "at main workplace" — in a similar way. Some subjects reported being at work in different locations, so the original label "at work" is still valid for analysis and may have a different meaning than "at main workplace" (which was designed to capture behavioral patterns typical to the most common place that a person works in). This adjustment removed some examples where the label "At work" was probably incorrectly reported, but more importantly, it added the missing label in cases where the subject was clearly present at their most common workplace.

Labels corrected using other labels. We used reported values of other labels to adjust some labels. In a few cases it was clear that the reported labels were mistakes (because the combination of labels was unreasonable). In other cases a relevant label was simply not reported, even though it clearly should be relevant according to the other reported labels.

- "Walking". We identified a few cases where subjects reported walking together with labels related to driving. In cases where location data was available, it was clear on the map that the correct activity was the drive and not the walk. In the adjusted version of "walking" we changed the value to "false" whenever the subject reported "on a bus", "in a car", "drive (I'm the driver)", "drive (I'm a passenger)", "motorbike", "skateboarding" or "at the pool".

- "Running". The adjusted version was set to "false" for the same activities as in the adjusted "walking" label, plus in cases where the subject reported "playing baseball" or "playing frisbee". Although these cases are likely still valid (because the subject decided to report they were running during these playing activities), we decided to create the adjusted

"running" version to represent a more coherent running activity (assuming that the playing activities involve a mixture of running, walking and standing intermittently). While the adjusted versions of "walking" and "running" may have a few misses (*e.g.*, some minutes during a baseball game when the subject was purely running), these misses don't harm the integrity of the multi-class experiments, which used only examples that had positive labels of "running", "walking", "sitting", *etc.*.

- "Exercise". The adjusted version was set to "true" whenever the subject reported "exercising", "running", "bicycling", "lifting weights", "elliptical machine", "treadmill", "stationary bike" or "at the gym". This adjustment boosted the representation of the exercise behavior and also took advantage of reported specific activities without enough examples to be analyzed on their own.

- "Indoors". The adjusted version was set to "true" whenever the subject reported "indoors", "sleeping", "toilet", "bathing — bath", "bathing — shower", "in class", "at home", "at a bar", "at the gym" or "elevator". It is reasonable that many subjects simply did not bother to report being indoors every time they did an activity indoors.

- "Outside". The adjusted version was set to "true" whenever the subject reported "outside", "skateboarding", "playing baseball", "playing frisbee", "gardening", "raking leaves", "strolling", "hiking", "at the beach", "at sea" or "motorbike".

- "At a restaurant". In the adjusted version we changed the value to "false" whenever the subject reported "on a bus", "in a car", "drive (I'm the driver)", "drive (I'm a passenger)" or "motorbike".

2.8.6 Classification methods

Our system uses binary logistic regression classifiers (with a fitted intercept). Logistic regression provides a real-valued output, interpreted as the probability of the relevance of the label (value larger than 0.5 yielding a decision of "relevant"). For each context label we used an independent model. We first randomly partitioned the training examples into internal training and validation subsets, allocating one third of the training examples for the validation subset, while maintaining the same proportion of positive *vs.* negative examples in both subsets. We then used grid search to select the cost parameter C for logistic regression: for each value (out of $\{0.001, 0.01, 0.1, 1, 10, 100\}$) we trained a logistic regression model on the internal train subset and tested the model on the validation subset. We selected the value of C that resulted in highest F1 measure on the validation subset. We then re-trained a logistic regression model with the selected value on the entire training set. For the leave-one-user-out experiment with the EF system we simplified the procedure and only trained the logistic regression models with value of $C = 1$ (instead of performing grid search). The learned weights from LFL for a set of selected labels that are presented in Figure 2.4 (A) are taken from the first (of five) training set of the cross validation evaluation. To look at misclassifications and to produce the confusion matrices in Figure 2.4 (B–G) we used the multiclass (one-versus-rest) version of logistic regression, with a fixed cost value of $C = 1$. Each multiclass experiment used the set of examples annotated with exactly one label from the examined label subset and with data from all of the sensors of interest (so an experiment with only accelerometer and gyroscope sensors might have more examples than an experiment with accelerometer, gyroscope and watch accelerometer). These experiments were more fitting than binary classification in cases where missing labels are common. For example, labels describing the phone's position were not always consistently annotated. A binary classifier will use all negative examples to learn a decision boundary, including examples the subject forgot to label, which may skew the results if there are many missing labels.

2.8.7 Performance evaluation

In order to make a fair comparison among the different sensors, evaluation was done on the subset of examples with data from all six core sensors available ($\sim$177k examples from 51 subjects). In the training phase, however, a single-sensor classifier was allowed to use all examples available (*e.g.*, all examples in the dataset had phone state data, so the PS single-sensor classifier was trained with all examples). While the early fusion system benefited from the advantage of modeling correlations between features of different sensors, it was limited to being trained only on examples with all sensor data available. The late fusion systems, on the other hand, had the advantage of using single-sensor classifiers that were trained on many more examples.

Classifier performance was evaluated using 5-fold cross validation. The subjects were randomly partitioned once into 5 folds, while equalizing the proportion of iPhone *vs.* Android users between folds (To keep a fair evaluation it was important to partition the subjects, and not randomly partition the pool of examples, in order to avoid having examples from the same subject appear in both the training set and the test set). The cross validation procedure repeats the following for each fold: (1) hold out the selected fold to act as the test set (2) train a classifier on the remaining folds (3) apply the classifier to the held out test set. For each fold and for each label, we counted the numbers of true positives (TP. Examples that were correctly classified as positive), true negatives (TN. Examples that were correctly classified as negative), false positives (FP. Examples that were wrongfully classified as positive) and false negatives (FN. Examples that were wrongfully classified as negative). At the end of the 5-fold procedure we summed up the total numbers of TP, TN, FP and FN over the entire evaluation set and calculated the following performance metrics:

- *Accuracy* is the proportion of correctly classified examples out of all the examples. This metric is sensitive to imbalanced label proportion in the data.

- *True positive rate* (TPR, also called sensitivity or recall) is the proportion of positive examples that were correctly classified as positive: recall=TPR=TP/(TP+FN).

- *True negative rate* (TNR, also called specificity) is the proportion of negative examples that were correctly classified as negative: TNR=TN/(TN+FP).

- *Precision* (prec) is the proportion of correctly classified examples out of the examples that the classifier declared as positive: precision=TP/(TP+FP).

- *Balanced accuracy* is a measure that accounts for the tradeoff between true positive rate and true negative rate: BA=(TPR+TNR)/2.

- The *F1* measure is another such measure, which takes the harmonic mean of precision and recall: F1=(2*TPR*prec)/(TPR+prec).

While the balanced accuracy is easy to interpret (chance level is always 0.5 and perfect performance is 1) the F1 measure is very sensitive to how rare the positive examples are, so for each label a typical F1 value is different. The 5-fold subject partition is available with the dataset, and we encourage researchers using the dataset to evaluate new methods to use the same 5-fold partition, in order to promote fair comparisons.

Random chance. To assess the statistical significance of the performance scores we achieved, we evaluated a distribution of performance scores achieved by a random classifier. The random classifier declares "relevant" with probability 0.5 independently for each example and for each label. To estimate the distribution of scores that such a classifier obtains, we ran 100 simulations (each time the classifier randomly assigned binary predictions and the performance scores were calculated over the entire evaluation dataset). Chance level (expected value of the random classifier) of balanced accuracy is 0.5 for every label. For the F1 measure the chance level for each label is dependent on the proportion of positive and negative examples in the data. For each performance measure and for each label we estimated a value which we call *p99*, the

99^{th} percentile among the 100 scores achieved in the 100 simulations. Hence the probability of obtaining a score greater than p99 by chance is less than 1%. For average (over a set of labels) scores the p99 value was calculated similarly (computing the average score for each of the 100 simulations).

2.8.8 User personalization assessment

To assess the advantages of user personalization, we selected a single test subject that had provided relatively many examples and many labels. We partitioned this user's examples into the first half and second half of the examples (according to their recording timestamps), to simulate an adaptation training period (the first half) and a deployment period (the second half). We used early fusion (EF) classifiers to combine the features from all 6 sensors. The *universal model* was the one used in previous experiments, taken from the fold where the test user was part of the cross validation test set (so the universal model was trained on 48 other users). The *individual model* was trained only on data from the test user, taken from the first half of the subject's examples. The *adapted model* was a combination of both the universal and individual models using the LFA method (*i.e.*, averaging the probability outputs of both models). All three models were tested on the same set of unseen test examples (the second half of the subject's examples). For some labels, an insufficient number of examples to train an individual classifier resulted in a trivial classifier (always declaring the same answer). In those cases the performance was reported as chance level (BA of 0.5 and F1 of 0).

2.9 Detailed results tables

2.9.1 5-fold cross validation evaluation

Table 2.3: 5-fold evaluation performance (BA) of the different classifiers on each label. Part 1 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Lying down	54359	47	0.50	0.72	0.69	0.81	0.66	0.79	0.85	0.87	0.86	**0.88**
Sitting	82904	50	0.50	0.63	0.61	0.68	0.61	0.65	0.69	**0.76**	0.75	0.75
Walking	11892	50	0.51	0.77	0.80	0.75	0.66	0.63	0.70	0.80	0.80	**0.80**
Running	675	19	0.52	0.69	0.66	**0.80**	0.56	0.48	0.58	0.62	0.69	0.71
Bicycling	3523	22	0.51	0.81	0.81	0.84	0.81	0.77	0.83	0.87	0.87	**0.87**
Sleeping	42920	40	0.50	0.75	0.70	0.81	0.62	0.79	0.87	0.88	0.88	**0.89**
Lab work	2898	8	0.51	0.71	0.62	0.65	0.84	0.71	0.81	0.84	**0.87**	0.85
In class	2872	13	0.51	0.60	0.63	0.57	0.74	0.76	0.67	0.70	0.77	**0.80**
In a meeting	2904	34	0.51	0.60	0.57	0.62	0.63	0.79	0.73	0.80	0.79	**0.82**
At main workplace	20382	26	0.50	0.57	0.49	0.63	0.76	0.65	0.78	0.80	0.80	**0.81**
Indoors	107944	51	0.50	0.66	0.66	0.67	0.63	0.71	0.72	0.75	0.75	**0.76**
Outside	7629	36	0.51	0.70	0.73	0.70	0.66	0.66	0.73	0.74	0.77	**0.78**
In a car	3635	24	0.51	0.79	0.65	0.71	0.81	0.77	0.84	0.85	0.86	**0.86**
On a bus	1185	24	0.52	0.73	0.69	0.67	0.75	0.74	0.82	0.77	**0.84**	0.83
Drive (I'm the driver)	5034	24	0.51	0.79	0.61	0.75	0.82	0.74	0.83	0.84	0.86	**0.87**
Drive (I'm a passenger)	1655	19	0.51	0.76	0.71	0.64	0.79	0.76	0.81	0.84	0.84	**0.85**
At home	83977	50	0.50	0.65	0.63	0.66	0.63	0.71	0.70	0.75	0.77	**0.78**
At a restaurant	1320	16	0.52	0.62	0.67	0.68	0.58	**0.85**	0.77	0.76	0.83	0.81
Phone in pocket	15301	31	0.50	0.69	0.75	0.67	0.61	0.64	0.72	0.77	**0.77**	0.77
Exercise	5384	36	0.51	0.73	0.73	0.77	0.71	0.70	0.77	0.81	0.80	**0.81**
Cooking	2257	33	0.51	0.52	0.53	0.68	0.57	0.62	0.68	0.71	0.71	**0.72**
Shopping	896	18	0.52	0.70	0.70	0.69	0.54	0.59	0.79	0.69	0.76	**0.80**
Strolling	434	8	0.53	0.67	0.74	0.72	0.67	0.64	0.75	0.66	**0.77**	0.74
Drinking (alcohol)	864	10	0.52	0.71	0.69	0.50	0.56	0.80	0.74	0.70	**0.82**	0.81
Bathing - shower	1186	27	0.52	0.53	0.55	**0.73**	0.47	0.63	0.47	0.64	0.67	0.70
average			0.50	0.68	0.66	0.70	0.67	0.70	0.75	0.77	0.80	0.80

Table 2.4: 5-fold evaluation performance (BA) of the different classifiers on each label. Part 2 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Cleaning	1839	22	0.51	0.63	0.64	**0.71**	0.41	0.60	0.51	0.60	0.70	0.68
Laundry	473	12	0.52	0.65	0.66	**0.66**	0.38	0.53	0.65	0.58	0.63	0.63
Washing dishes	851	17	0.52	0.40	0.52	0.70	0.58	0.60	0.57	0.65	0.70	**0.70**
Watching TV	9412	28	0.51	0.61	0.54	0.56	0.56	0.64	0.67	0.65	**0.70**	0.68
Surfing the internet	11641	28	0.50	0.56	0.55	0.60	0.54	0.60	0.57	0.59	0.63	**0.63**
At a party	404	3	0.53	0.74	0.71	0.49	0.54	**0.81**	0.56	0.54	0.76	0.75
At a bar	520	4	0.53	0.45	0.66	0.53	0.60	0.49	**0.93**	0.50	0.61	0.66
At the beach	122	5	0.55	0.62	0.48	0.47	**0.72**	0.58	0.70	0.50	0.71	0.70
Singing	384	6	0.53	0.57	0.64	0.46	0.61	**0.68**	0.59	0.50	0.65	0.53
Talking	18976	44	0.50	0.60	0.61	0.60	0.54	0.65	0.65	0.65	0.67	**0.67**
Computer work	23692	38	0.50	0.57	0.56	0.62	0.63	0.61	0.68	0.68	**0.71**	0.70
Eating	10169	49	0.51	0.59	0.58	0.60	0.51	0.61	0.62	**0.66**	0.65	0.65
Toilet	1646	33	0.51	0.57	0.51	0.58	0.57	0.64	0.59	0.65	0.66	**0.66**
Grooming	1847	25	0.51	0.44	0.49	0.62	0.59	0.63	0.58	0.60	**0.63**	0.63
Dressing	1308	27	0.52	0.51	0.52	0.64	0.54	0.65	0.61	0.64	**0.67**	0.67
At the gym	906	6	0.52	0.50	0.55	0.58	0.57	0.65	**0.70**	0.54	0.64	0.61
Stairs - going up	399	17	0.53	0.70	**0.73**	0.65	0.55	0.55	0.51	0.58	0.69	0.67
Stairs - going down	390	15	0.53	0.71	**0.73**	0.66	0.55	0.55	0.51	0.58	0.71	0.66
Elevator	124	8	0.55	0.72	**0.76**	0.44	0.54	0.71	0.51	0.49	0.73	0.73
Standing	22766	51	0.50	0.60	0.59	0.67	0.54	0.59	0.63	**0.68**	0.67	0.68
At school	25840	39	0.50	0.59	0.59	0.59	0.66	0.64	0.68	0.70	**0.70**	0.70
Phone in hand	8595	37	0.51	0.65	**0.68**	0.56	0.59	0.59	0.61	0.64	0.67	0.66
Phone in bag	5589	22	0.51	0.59	0.56	0.55	0.59	0.64	0.69	0.67	0.68	**0.69**
Phone on table	70611	43	0.50	0.60	0.61	0.56	0.53	0.55	0.61	0.61	0.62	**0.62**
With co-workers	4139	17	0.51	0.57	0.57	0.61	0.58	0.68	0.67	0.69	0.71	**0.72**
With friends	12865	25	0.50	0.55	0.58	0.53	0.54	0.60	0.60	0.55	**0.61**	0.58
average			0.50	0.59	0.60	0.59	0.56	0.62	0.62	0.60	0.67	0.66

Table 2.5: 5-fold evaluation performance (F1) of the different classifiers on each label. Part 1 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Lying down	54359	47	0.38	0.61	0.59	0.71	0.55	0.69	0.78	0.81	0.79	**0.82**
Sitting	82904	50	0.49	0.58	0.58	0.67	0.59	0.62	0.72	**0.75**	0.75	0.74
Walking	11892	50	0.12	0.38	0.38	0.31	0.22	0.19	0.22	0.38	**0.39**	0.38
Running	675	19	0.01	0.03	0.03	**0.04**	0.01	0.01	0.01	0.03	0.04	0.04
Bicycling	3523	22	0.04	0.19	0.16	0.23	0.18	0.12	0.17	**0.31**	0.25	0.26
Sleeping	42920	40	0.33	0.57	0.53	0.65	0.44	0.63	0.75	0.79	0.77	**0.81**
Lab work	2898	8	0.03	0.08	0.06	0.06	0.11	0.09	0.11	**0.21**	0.16	0.19
In class	2872	13	0.03	0.05	0.06	0.04	0.07	0.10	0.06	0.13	0.12	**0.14**
In a meeting	2904	34	0.03	0.05	0.04	0.05	0.05	0.11	0.07	**0.17**	0.10	0.14
At main workplace	20382	26	0.19	0.23	0.18	0.28	0.41	0.31	0.42	0.49	0.47	**0.50**
Indoors	107944	51	0.55	0.74	0.73	0.70	0.68	0.75	0.71	0.78	**0.79**	0.78
Outside	7629	36	0.08	0.20	0.20	0.18	0.15	0.15	0.20	0.23	**0.26**	0.25
In a car	3635	24	0.04	0.15	0.07	0.10	**0.27**	0.13	0.16	0.23	0.22	0.23
On a bus	1185	24	0.01	0.04	0.03	0.03	0.07	0.04	0.06	0.07	**0.07**	0.06
Drive (I'm the driver)	5034	24	0.06	0.21	0.09	0.15	**0.37**	0.16	0.23	0.31	0.31	0.31
Drive (I'm a passenger)	1655	19	0.02	0.07	0.04	0.04	**0.15**	0.07	0.08	0.14	0.12	0.12
At home	83977	50	0.49	0.66	0.65	0.64	0.63	0.70	0.67	0.74	0.76	**0.77**
At a restaurant	1320	16	0.02	0.02	0.03	0.03	0.02	0.08	0.05	**0.11**	0.07	0.09
Phone in pocket	15301	31	0.15	0.28	0.33	0.26	0.21	0.25	0.28	**0.38**	0.36	0.37
Exercise	5384	36	0.06	0.21	0.18	0.24	0.14	0.13	0.19	**0.27**	0.26	0.25
Cooking	2257	33	0.03	0.03	0.03	0.05	0.03	0.04	0.06	**0.09**	0.07	0.08
Shopping	896	18	0.01	0.03	0.03	0.02	0.01	0.02	0.04	**0.04**	0.04	0.04
Strolling	434	8	0.01	0.02	0.02	0.01	0.01	0.01	0.02	0.03	**0.03**	0.03
Drinking (alcohol)	864	10	0.01	0.03	0.02	0.01	0.01	0.04	0.03	**0.07**	0.07	0.06
Bathing - shower	1186	27	0.01	0.01	0.02	0.04	0.01	0.02	0.01	0.04	0.04	**0.05**
average			0.13	0.22	0.20	0.22	0.22	0.22	0.24	0.30	0.29	0.30

Table 2.6: 5-fold evaluation performance (F1) of the different classifiers on each label. Part 2 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Cleaning	1839	22	0.02	0.05	0.05	0.05	0.01	0.03	0.02	0.05	**0.06**	0.05
Laundry	473	12	0.01	0.01	0.01	0.01	0.00	0.01	0.01	**0.02**	0.01	0.02
Washing dishes	851	17	0.01	0.01	0.01	0.03	0.01	0.02	0.01	0.03	0.03	**0.04**
Watching TV	9412	28	0.10	0.14	0.11	0.12	0.12	0.17	0.18	0.21	0.22	**0.22**
Surfing the internet	11641	28	0.12	0.15	0.14	0.17	0.13	0.17	0.15	0.18	0.19	**0.20**
At a party	404	3	0.01	0.01	0.01	0.00	0.01	0.03	0.01	**0.04**	0.03	0.04
At a bar	520	4	0.01	0.00	0.01	0.01	0.01	0.00	**0.09**	0.00	0.03	0.06
At the beach	122	5	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	**0.02**	0.02
Singing	384	6	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.00	**0.01**	0.01
Talking	18976	44	0.18	0.25	0.26	0.24	0.21	0.29	0.27	0.29	0.30	**0.30**
Computer work	23692	38	0.21	0.26	0.25	0.30	0.31	0.30	0.35	0.38	0.39	**0.39**
Eating	10169	49	0.11	0.15	0.14	0.15	0.11	0.16	0.15	**0.19**	0.18	0.17
Toilet	1646	33	0.02	0.03	0.02	0.03	0.02	0.03	0.03	**0.04**	0.04	0.04
Grooming	1847	25	0.02	0.02	0.02	0.03	0.03	0.04	0.03	0.05	0.04	**0.05**
Dressing	1308	27	0.02	0.02	0.02	0.03	0.02	0.03	0.03	**0.04**	0.04	0.04
At the gym	906	6	0.01	0.01	0.01	0.02	0.01	0.02	0.02	0.03	0.03	**0.03**
Stairs - going up	399	17	0.01	**0.02**	0.02	0.01	0.01	0.01	0.00	0.01	0.02	0.01
Stairs - going down	390	15	0.00	**0.02**	0.02	0.01	0.01	0.01	0.00	0.01	0.02	0.01
Elevator	124	8	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	**0.01**
Standing	22766	51	0.21	0.28	0.27	0.35	0.23	0.27	0.29	**0.36**	0.34	0.35
At school	25840	39	0.23	0.30	0.29	0.30	0.38	0.34	0.39	0.41	0.41	**0.41**
Phone in hand	8595	37	0.09	0.17	**0.17**	0.11	0.13	0.13	0.13	0.16	0.17	0.16
Phone in bag	5589	22	0.06	0.10	0.08	0.07	0.09	0.11	0.12	**0.15**	0.14	0.14
Phone on table	70611	43	0.45	0.58	0.58	0.51	0.50	0.51	0.56	0.56	**0.59**	0.58
With co-workers	4139	17	0.05	0.06	0.06	0.07	0.06	0.11	0.08	0.13	0.12	**0.13**
With friends	12865	25	0.13	0.15	0.17	0.14	0.15	0.18	0.18	0.15	**0.19**	0.18
average			0.08	0.11	0.11	0.11	0.10	0.11	0.12	0.13	0.14	0.14

2.9.2 Leave-one-user-out evaluation

Table 2.7: Leave-one-user-out evaluation performance (BA) of the different classifiers on each label. Part 1 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Lying down	54359	47	0.50	0.73	0.69	0.81	0.65	0.79	0.84	0.87	0.86	**0.88**
Sitting	82904	50	0.50	0.63	0.61	0.68	0.61	0.65	0.69	**0.76**	0.75	0.75
Walking	11892	50	0.51	0.77	0.80	0.75	0.66	0.63	0.71	0.80	0.80	**0.81**
Running	675	19	0.52	0.69	0.69	**0.80**	0.56	0.50	0.57	0.67	0.72	0.76
Bicycling	3523	22	0.51	0.81	0.81	0.85	0.80	0.76	0.80	**0.87**	0.86	0.87
Sleeping	42920	40	0.50	0.75	0.70	0.81	0.62	0.80	0.85	0.88	0.88	**0.89**
Lab work	2898	8	0.51	0.69	0.61	0.65	0.82	0.70	0.84	0.84	**0.84**	0.84
In class	2872	13	0.51	0.61	0.63	0.58	0.74	0.77	0.72	0.74	0.79	**0.81**
In a meeting	2904	34	0.51	0.62	0.59	0.62	0.66	0.78	0.73	0.81	0.80	**0.82**
At main workplace	20382	26	0.50	0.55	0.49	0.64	0.76	0.65	0.80	0.80	0.81	**0.82**
Indoors	107944	51	0.50	0.67	0.66	0.68	0.63	0.70	0.72	**0.76**	0.75	0.76
Outside	7629	36	0.51	0.72	0.74	0.70	0.66	0.67	0.71	0.75	0.78	**0.79**
In a car	3635	24	0.51	0.79	0.66	0.71	0.82	0.76	0.83	0.85	0.86	**0.87**
On a bus	1185	24	0.52	0.74	0.69	0.68	0.72	0.72	0.80	0.78	**0.83**	0.82
Drive (I'm the driver)	5034	24	0.51	0.80	0.62	0.75	0.83	0.75	0.84	0.84	0.86	**0.87**
Drive (I'm a passenger)	1655	19	0.51	0.76	0.70	0.64	0.80	0.77	0.82	**0.84**	0.83	0.84
At home	83977	50	0.50	0.65	0.63	0.66	0.62	0.72	0.72	0.76	0.77	**0.77**
At a restaurant	1320	16	0.52	0.62	0.68	0.69	0.57	**0.84**	0.74	0.79	0.84	0.84
Phone in pocket	15301	31	0.50	0.69	0.75	0.67	0.61	0.64	0.71	0.77	0.77	**0.77**
Exercise	5384	36	0.51	0.74	0.73	0.77	0.71	0.70	0.75	0.81	0.81	**0.81**
Cooking	2257	33	0.51	0.52	0.55	0.67	0.57	0.62	0.68	0.71	0.72	**0.72**
Shopping	896	18	0.52	0.71	0.69	0.68	0.53	0.57	**0.79**	0.67	0.75	0.78
Strolling	434	8	0.53	0.64	0.73	0.70	0.63	0.62	0.71	0.67	0.74	**0.75**
Drinking (alcohol)	864	10	0.52	0.72	0.70	0.54	0.56	0.79	0.54	0.68	**0.79**	0.77
Bathing - shower	1186	27	0.52	0.50	0.54	**0.74**	0.48	0.63	0.48	0.64	0.69	0.72
average			0.50	0.68	0.67	0.70	0.66	0.70	0.74	0.78	0.80	0.81

2.10 Acknowledgements

We thank Rafael Aguayo and Jennifer Lu for their help implementing the ExtraSensory App.

Chapter 2, in full, is a reprint of the material as it appears in IEEE Pervasive Computing, 16(4):62–74, October–December 2017, Y. Vaizman, K. Ellis, and G. Lanckriet. The dissertation

author was a co-primary investigator and co-author of this paper.

Table 2.8: Leave-one-user-out evaluation performance (BA) of the different classifiers on each label. Part 2 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Cleaning	1839	22	0.51	0.62	0.63	**0.73**	0.42	0.62	0.49	0.70	0.71	0.70
Laundry	473	12	0.52	0.67	0.65	0.66	0.35	0.52	**0.78**	0.60	0.68	0.70
Washing dishes	851	17	0.52	0.36	0.48	**0.69**	0.54	0.61	0.54	0.66	0.67	0.68
Watching TV	9412	28	0.51	0.61	0.54	0.57	0.57	0.67	0.66	0.69	**0.72**	0.71
Surfing the internet	11641	28	0.50	0.55	0.58	0.59	0.56	0.60	0.57	0.61	**0.62**	0.62
At a party	404	3	0.53	0.73	0.71	0.48	0.70	**0.84**	0.67	0.52	0.79	0.76
At a bar	520	4	0.53	0.53	0.69	0.50	0.64	0.62	**0.88**	0.52	0.71	0.68
At the beach	122	5	0.55	0.66	0.51	0.52	**0.71**	0.58	0.69	0.57	0.71	0.71
Singing	384	6	0.53	0.56	0.62	0.46	**0.70**	0.68	0.60	0.48	0.68	0.53
Talking	18976	44	0.50	0.61	0.61	0.61	0.55	0.66	0.64	0.66	0.68	**0.68**
Computer work	23692	38	0.50	0.59	0.57	0.62	0.65	0.59	0.67	0.69	**0.71**	0.69
Eating	10169	49	0.51	0.59	0.58	0.60	0.53	0.61	0.63	0.66	**0.66**	0.66
Toilet	1646	33	0.51	0.57	0.52	0.57	0.57	0.63	0.56	0.65	**0.65**	0.65
Grooming	1847	25	0.51	0.46	0.53	0.62	0.60	0.65	0.53	0.63	0.64	**0.66**
Dressing	1308	27	0.52	0.51	0.54	0.66	0.53	0.67	0.55	0.66	0.67	**0.68**
At the gym	906	6	0.52	0.55	0.56	0.67	0.51	0.67	**0.70**	0.58	0.67	0.67
Stairs - going up	399	17	0.53	0.68	**0.76**	0.65	0.57	0.57	0.48	0.59	0.69	0.66
Stairs - going down	390	15	0.53	0.70	**0.75**	0.66	0.54	0.55	0.48	0.57	0.69	0.63
Elevator	124	8	0.55	0.68	0.70	0.56	0.57	**0.70**	0.54	0.50	0.62	0.61
Standing	22766	51	0.50	0.60	0.59	0.67	0.54	0.59	0.62	**0.68**	0.66	0.67
At school	25840	39	0.50	0.60	0.59	0.59	0.68	0.66	0.70	**0.72**	0.71	0.71
Phone in hand	8595	37	0.51	0.66	**0.68**	0.56	0.58	0.58	0.59	0.63	0.67	0.66
Phone in bag	5589	22	0.51	0.60	0.56	0.56	0.59	0.69	0.69	0.72	0.71	**0.73**
Phone on table	70611	43	0.50	0.60	0.61	0.56	0.52	0.56	0.61	0.61	**0.63**	0.62
With co-workers	4139	17	0.51	0.55	0.57	0.61	0.61	0.68	0.71	0.69	0.73	**0.74**
With friends	12865	25	0.50	0.56	0.57	0.54	0.55	0.62	0.59	0.58	**0.63**	0.61
average			0.50	0.59	0.60	0.60	0.57	0.63	0.62	0.62	0.68	0.67

Table 2.9: Leave-one-user-out evaluation performance (F1) of the different classifiers on each label. Part 1 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Lying down	54359	47	0.38	0.62	0.59	0.71	0.54	0.69	0.76	0.81	0.79	**0.82**
Sitting	82904	50	0.49	0.58	0.58	0.68	0.58	0.62	0.71	**0.75**	0.75	0.74
Walking	11892	50	0.12	0.38	0.37	0.32	0.21	0.19	0.22	0.38	**0.39**	0.39
Running	675	19	0.01	0.03	0.02	0.04	0.01	0.01	0.01	0.04	0.04	**0.04**
Bicycling	3523	22	0.04	0.19	0.15	0.23	0.16	0.12	0.15	**0.30**	0.24	0.26
Sleeping	42920	40	0.33	0.56	0.53	0.65	0.44	0.64	0.74	0.79	0.76	**0.80**
Lab work	2898	8	0.03	0.07	0.06	0.06	0.11	0.08	0.11	**0.21**	0.15	0.18
In class	2872	13	0.03	0.05	0.06	0.04	0.08	0.11	0.07	0.13	0.12	**0.14**
In a meeting	2904	34	0.03	0.06	0.04	0.05	0.06	0.11	0.07	**0.17**	0.11	0.15
At main workplace	20382	26	0.19	0.22	0.19	0.29	0.41	0.31	0.43	0.49	0.48	**0.52**
Indoors	107944	51	0.55	0.75	0.73	0.71	0.68	0.75	0.71	**0.79**	0.79	0.79
Outside	7629	36	0.08	0.21	0.20	0.18	0.16	0.16	0.20	0.23	**0.26**	0.25
In a car	3635	24	0.04	0.15	0.08	0.10	**0.27**	0.13	0.16	0.23	0.22	0.23
On a bus	1185	24	0.01	0.04	0.03	0.03	0.05	0.04	0.05	0.07	**0.07**	0.07
Drive (I'm the driver)	5034	24	0.06	0.21	0.09	0.16	**0.38**	0.15	0.21	0.31	0.31	0.31
Drive (I'm a passenger)	1655	19	0.02	0.07	0.04	0.04	**0.15**	0.07	0.08	0.13	0.12	0.11
At home	83977	50	0.49	0.67	0.65	0.65	0.63	0.71	0.69	0.75	0.76	**0.76**
At a restaurant	1320	16	0.02	0.03	0.03	0.03	0.02	0.07	0.04	**0.11**	0.07	0.10
Phone in pocket	15301	31	0.15	0.29	0.34	0.26	0.22	0.25	0.27	**0.38**	0.37	0.37
Exercise	5384	36	0.06	0.21	0.16	0.22	0.14	0.13	0.15	0.26	**0.26**	0.24
Cooking	2257	33	0.03	0.03	0.03	0.05	0.03	0.04	0.05	**0.08**	0.07	0.07
Shopping	896	18	0.01	0.03	0.02	0.02	0.01	0.01	**0.04**	0.04	0.04	0.04
Strolling	434	8	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.02	**0.02**	0.02
Drinking (alcohol)	864	10	0.01	0.03	0.02	0.01	0.01	0.04	0.01	0.05	**0.06**	0.05
Bathing - shower	1186	27	0.01	0.01	0.02	0.04	0.01	0.02	0.01	0.04	0.04	**0.05**
average			0.13	0.22	0.20	0.22	0.22	0.22	0.24	0.30	0.29	0.30

Table 2.10: Leave-one-user-out evaluation performance (F1) of the different classifiers on each label. Part 2 of the labels. For each label n_e is the number of examples and n_s is the number of subjects in the testing (possibly more examples participated in the training). p99 marks the 99^{th} percentile of random scores — a score above the p99 value has less than 0.01 probability to be achieved randomly. For each label the score of the highest performing classifier is marked in bold.

	n_e	n_s	p99	**Acc**	**Gyro**	**WAcc**	**Loc**	**Aud**	**PS**	**EF**	**LFA**	**LFL**
Cleaning	1839	22	0.02	0.04	0.04	0.06	0.01	0.03	0.02	**0.07**	0.06	0.06
Laundry	473	12	0.01	0.01	0.01	0.01	0.00	0.01	0.02	0.02	0.02	**0.02**
Washing dishes	851	17	0.01	0.00	0.01	0.02	0.01	0.02	0.01	**0.03**	0.03	0.03
Watching TV	9412	28	0.10	0.14	0.11	0.12	0.12	0.19	0.17	0.23	0.22	**0.24**
Surfing the internet	11641	28	0.12	0.14	0.16	0.16	0.14	0.17	0.15	**0.20**	0.19	0.19
At a party	404	3	0.01	0.01	0.01	0.00	0.01	0.03	0.01	0.02	0.03	**0.04**
At a bar	520	4	0.01	0.01	0.01	0.01	0.01	0.02	**0.06**	0.01	0.04	0.05
At the beach	122	5	0.00	0.00	0.00	0.00	0.01	0.00	0.01	**0.05**	0.02	0.02
Singing	384	6	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.00	**0.01**	0.01
Talking	18976	44	0.18	0.25	0.25	0.25	0.21	0.30	0.26	0.30	0.30	**0.30**
Computer work	23692	38	0.21	0.28	0.26	0.30	0.32	0.28	0.34	0.39	**0.39**	0.39
Eating	10169	49	0.11	0.15	0.14	0.15	0.11	0.16	0.15	**0.18**	0.18	0.18
Toilet	1646	33	0.02	0.03	0.02	0.03	0.02	0.03	0.02	**0.04**	0.04	0.04
Grooming	1847	25	0.02	0.02	0.02	0.04	0.03	0.04	0.02	0.05	0.04	**0.05**
Dressing	1308	27	0.02	0.02	0.02	0.03	0.02	0.03	0.02	**0.04**	0.04	0.04
At the gym	906	6	0.01	0.01	0.01	0.02	0.01	0.02	0.03	**0.04**	0.03	0.04
Stairs - going up	399	17	0.01	0.01	**0.02**	0.01	0.01	0.01	0.00	0.01	0.02	0.01
Stairs - going down	390	15	0.00	0.01	**0.02**	0.01	0.01	0.01	0.00	0.01	0.02	0.01
Elevator	124	8	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Standing	22766	51	0.21	0.28	0.27	0.35	0.23	0.27	0.29	**0.35**	0.34	0.35
At school	25840	39	0.23	0.30	0.30	0.29	0.42	0.37	0.40	**0.44**	0.42	0.42
Phone in hand	8595	37	0.09	0.17	**0.17**	0.11	0.13	0.12	0.12	0.16	0.17	0.16
Phone in bag	5589	22	0.06	0.11	0.08	0.08	0.08	0.13	0.11	**0.16**	0.15	0.16
Phone on table	70611	43	0.45	**0.59**	0.58	0.51	0.48	0.51	0.56	0.55	0.59	0.58
With co-workers	4139	17	0.05	0.06	0.06	0.07	0.07	0.11	0.09	0.13	0.12	**0.14**
With friends	12865	25	0.13	0.16	0.17	0.14	0.15	0.20	0.17	0.18	**0.20**	0.20
average			0.08	0.11	0.11	0.11	0.10	0.12	0.12	0.14	0.14	0.14

Chapter 3

Context-Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification

Abstract

Automatic recognition of behavioral context (location, activities, body-posture *etc.*) can serve health monitoring, aging care, and many other domains. Recognizing context *in-the-wild* is challenging because of great variability in behavioral patterns, and it requires a complex mapping from sensor features to predicted labels. Data collected in-the-wild may be unbalanced and incomplete, with cases of missing labels or missing sensors. We propose using the multiple layer perceptron (MLP) as a multi-task model for context recognition. Based on features from multi-modal sensors, the model simultaneously predicts many diverse context labels. We analyze the advantages of the model's hidden layers, which are shared among all sensors and all labels, and provide insight to the behavioral patterns that these hidden layers may capture. We demonstrate how recognition of new labels can be improved when utilizing a model that was trained for an initial set of labels, and show how to train the model to withstand missing sensors. We evaluate context recognition on the previously published *ExtraSensory Dataset*, which was collected in-the-wild. Compared to previously suggested models, the MLP improves recognition, even with fewer parameters than a linear model. The ability to train a good model using data that has incomplete, unbalanced labeling and missing sensors encourages further research with uncontrolled, in-the-wild behavior.

3.1 Introduction

The behavioral context of a person can be described by various aspects: where is the person? what kind of activities is the person doing? who is with the person? what is the body-posture state? and so on. Automatic recognition of behavioral context can serve many applications, such as monitoring physical activity [20], logging older adults' functional independence to promote aging at home [45], and context-adaptive personal assistant systems. In order for such applications to work well *in-the-wild*, meaning in real life settings, the context

recognition component should be seamlessly integrated. Ideally, people will conduct their regular daily behavior, while the non-interfering system recognizes what is going on using unobtrusive sensors, and every-day devices, like smartphones. If the system requires wearing an uncomfortable or unnatural sensor, it may cause the person to act differently, thus missing the goal of recognizing *natural* behavior.

In-the-wild human behavior has great variability and the system should not fail if the person takes the phone out of the pocket, exits a monitored room, or enters an elevator. Behavioral context is rich and complex: people walk, eat, or interact with their phones in different manners and typically do not focus on a single activity, like watching TV; they may watch TV while eating, cooking, or hanging out with friends. An activity like running can have different flavors: outside, indoors on a treadmill, at the gym, at home, alone, with friends, and so on. Applications that monitor physical activity will have to overcome this variability and recognize that people are running in all these different cases. Other in-the-wild applications may focus on social interaction, exposure to fresh air, or other aspects of daily life.

Many studies promoted great progress in processing sensor measurements to recognize basic human activity. However, most of these studies conducted controlled experiments. They handed foreign devices to research participants, with instructions for how to use them. Participants came to designated locations (typically a lab) on scheduled time and researchers observed them conducting scripted tasks. Unfortunately, such studies missed the great variability of natural behavior by scripting what to do and by forcing specific positioning on the body of devices (*e.g.* phone in the pocket or accelerometer on the hip). In these cases, the repeated simulations typically resulted in little variability among participants, making the recognition task easier than it should be. This means that models that worked well with simulated activities may fail in-the-wild [36]. In addition, many of the suggested systems in these works were not practical for in-the-wild usage because of inconvenient sensing apparatus or classification methods that are not fit for real-time or mobile applications.

This is why it is important that research in the field validates context recognition *in-the-wild* — in the same setting where real applications will eventually be deployed. In our previous work [91] we stated that when collecting data, researchers can promote natural behavior by maintaining four in-the-wild conditions: (1) naturally used devices, (2) unconstrained device placement, (3) natural environment, and (4) natural behavioral content.

We strongly believe that when analyzing data and suggesting methods, researchers should consider models that are appropriate for working in-the-wild. Behavioral models should be able to learn complex mappings from sensor measurements to the predicted contexts, while avoiding over-fitting to the training data. These models should be efficient enough to work on real-time, mobile applications. They should also work well with data that is not controlled and may be irregular, incomplete, and unbalanced: when relying on many participants to collect data from their daily lives, not everyone will contribute examples of cooking, driving, and so on; some participants may provide information about their activity, while ignoring other aspects like their environment. In addition, in-the-wild models should be able to work even when some of the sensors are missing.

In this paper, we introduce the use of multiple layer perceptron (MLP) as a multi-task model to recognizes rich descriptions of context, including details about environment, activities, body-posture, company, and more. We evaluate context recognition using the *ExtraSensory Dataset*, which is publicly available at http://extrasensory.ucsd.edu, a publicly available large-scale in-the-wild dataset that we previously described in [91]. We demonstrate how our model is useful in practical scenarios. The contribution of this paper is the introduction of a context-recognition model that improves recognition in-the-wild compared to the baseline suggested in [91], and addresses the following important considerations for in-the-wild research:

- The output of our MLP model is **multi-label**, meaning multiple context-labels can be relevant simultaneously. This allows for holistic descriptions of context that may include combinations of activities (like watching TV while eating), as well as environment, body-

posture, and other aspects. This is a more appropriate way to describe behavior than the previous multi-class approach, where the model predicted a single activity at any given time. We discuss the advantage of sharing parameters in a unified **multi-task** model.

- Our model can handle data with **incomplete and unbalanced labeling**. We achieve this by training with multi-label instance-weighting. This is important when collecting large in-the-wild data, where it is hard to control the distribution of labels.

- We demonstrate how the model facilitates **transfer learning** — it can help when collecting new data and extending the system to a new behavioral aspect. This is especially useful when there is limited data for the new labels and researchers want to take advantage of an existing system that was already trained to predict initial labels.

- We show that training with sensor-dropout can make the model resilient to **missing sensors**. This is an important property for in-the-wild systems — they should keep working smoothly, even when some of the information sources become unavailable.

3.2 The ExtraSensory Dataset

Before diving into the discussion of the recognition model we present in this paper, it is important to contextualize our work with respect to the data we use for evaluating context recognition. The data was fully described in our previous work [91], and we provide a brief description in this section. For collecting the data, we designed and implemented a mobile application (for iPhone and for Android, with an additional Pebble-watch component) that recorded a 20sec window of sensor measurements (from the phone and watch) every minute when the app was running in the background. The flexible user interface of our app provided many mechanisms for participants to self-report their context in terms of what they were doing, where they were, who they were with, where their phone was, and so on. Among the mechanisms,

they could report immediate future (up to thirty minutes) context, edit context labels for past events from the day, use the watch to respond to notifications, and more. This flexibility was important to minimize the interference that reporting labels had on the actual natural behavior. When collecting the data, we considered the four key conditions for natural behavior in-the-wild:

1. *Naturally used devices*: Participants used their own personal smartphone. We supplied an additional smartwatch, which is natural to wear and adds little burden. In [91] we showed how adding information from the watch can improve recognition. Here, we show how to reduce the reliance on extra devices, like the watch.

2. *Unconstrained device placement*: Participants were free to use their phone in any way convenient to them. We collected annotations regarding that aspect — participants sometimes reported labels describing the phone position (*e.g.* "Phone in pocket"; "Phone in hand") in addition to other contextual aspects. This allow us in this paper to jointly model device placement together with other aspects and better capture variability in-the-wild.

3. *Natural environment*: Each person participated for approximately one week. They collected data from their own environment and on their own schedule. This enables capturing diverse contexts that could not be simulated in lab experiments. This also resulted in technical challenges, like many cases when some sensors were not available. In this paper, we address such cases as examples of what a real application may have to face, and we make our model more resilient to missing sensors.

4. *Natural behavioral content*: We did not specify a list of tasks to perform or activities to focus on. Instead, we provided an extensive menu of over 100 behavioral attributes (context labels), and the option to select multiple labels simultaneously (multi-label setting). The participants engaged in their routine, and reported any labels that were relevant to their context. This method contributed to the authenticity of each individual's behavior, but it also made the data harder to manage, having incomplete and unbalanced labeling. In this

paper, we show how to manipulate the MLP in a non-standard fashion to handle this kind of irregular data.

The resulting *ExtraSensory Dataset* is publicly available and has over 300,000 labeled examples from sixty participants. Every example represents one minute and has measurements from various sensors on the phone and watch. Our intial analysis focused on six core sensors [91]: phone-accelerometer (Acc), phone-gyroscope (Gyro), phone-audio (Aud), phone-location (Loc), phone-state (PS), and watch-accelerometer (WAcc). Acc and Gyro are both 3-axial and were sampled in 40Hz. WAcc is 3-axial and was sampled at 25Hz. Audio was recorded at 22,050Hz and then processed on the phone to produce thirteen Mel Frequency Cepstral Coefficients (MFCCs) for every 46msec frame. Location was updated whenever there was a significant change. PS was sampled once a minute. From each sensing modality we extracted appropriate features, totaling 175 different features across multiple modalities. Specifically, from the motion sensors (Acc, Gyro, WAcc) we calculated simple statistics, axes-correlations, and spectral features; the audio sensor features are based on averages and standard deviations of MFCCs; from the location modality, we focused on relative-location, describing the variability of movement within a minute, plus estimates of altitude and speed; the phone state sensor features are binary indicators, specifying details like app-state, WiFi connectivity, and time-of-day.

The dataset also demonstrates a practical challenge in-the-wild: missing sensors. The watch was not worn all of the time, participants sometimes turned off location services to conserve battery, and audio was not available to our app during a phone call (to preserve privacy). Furthermore, only approximately half of the examples had all the six core sensors available.

Every labeled example in the dataset is annotated in a multi-label fashion — a combination of relevant labels, describing various aspects of the behavior, like the activity (*e.g.* "Computer work", "Cooking", "Drive — I'm the driver"), environment (*e.g.* "At home"; "At school"; "Outside"; "On a bus"), company (*e.g.* "With friends"), and body-state (*e.g.* "Lying down"; "Walking"). In addition, the dataset demonstrates a variety of label-combinations that were

reported by participants, describing detailed contexts, like “Running, Indoors, Exercise, At the gym, Phone on table” or “Sitting, On a bus, Phone in pocket, Talking, With friends”. For reporting these rich contexts, participants selected multiple relevant labels from a large menu. This method accounted for two important contributions:

1. It helped ensure that participants engaged in their individual authentic behavior, without trying to conform to any given list of activities.

2. It provides us with much richer descriptions of context and the opportunity to research different behavioral aspects and their relations.

On the other hand, this method resulted in incomplete and unbalanced labeling, with each participant contributing information about a small subset of the labels in the menu. Some labels (*e.g.* “Sleeping”) were more represented than others (*e.g.* “Washing dishes”).

In the remainder of the paper we discuss related work, describe our classification model based on MLP, and present results of applying the model to different scenarios, evaluated on the *ExtraSensory Dataset* (with validation on an additional dataset). We then discuss the results, suggest future improvements, and conclude the paper.

3.3 Related work

In this section we address previous research that used MLP as a tool for context recognition, both in controlled studies (3.3.1) and outside of the lab (3.3.2), and differentiate our work from those studies. We survey various approaches to modeling multiple aspects of context and previous methods to extend systems to new contexts (3.3.3), and we describe how previous studies have addressed missing input data (3.3.4). Finally, we regard to the important issue of performance metrics and how they impact the conclusions of research, especially with in-the-wild data that is skewed and unbalanced (3.3.5).

3.3.1 Off-the-shelf tools in controlled studies

Several studies have used MLP and other models as black-box tools to recognize activities from mobile sensors. Mantyjarvi *et al.* [53] used two acceleromenters on the waist to recognize four body movements. Kwapisz *et al.* [41] targeted six body states and used a built-in accelerometer in a smartphone that was placed in the participants' front pant pocket. They compared different models, including logistic regression, decision tree, and MLP. Guiri *et al.* [26] used more diverse sensors from a phone (placed in the pocket) and a watch, and distinguished nine physical activities. They compared five off-the-shelf models, including MLP. Pirttikangas *et al.* [69] designed a small device with multi-modal sensors. In their study they placed four such devices in specific positions on the body, plus a data collection terminal on the arm. They tested recognizing seventeen specific tasks using k-nearest-neighbors (kNN) and MLP.

Although these studies contributed greatly to methods for processing sensors across different modalities, their data was collected under heavily controlled conditions: Researchers handed foreign devices to the participants and placed them in specific body positions. Participation was done in designated locations and on scheduled time. The behavior itself was scripted and instructed, which may result in simulated, un-natural behavior with little variability among participants.

Models that fit well to such controlled data may generalize poorly in-the-wild [36]. K-nearest-neighbors is an example of a model that is not appropriate for in-the-wild applications but was still suggested. The apparent success of kNN in some controlled studies [69, 82] may rely heavily on the fact that repeated simulations of activities can be very similar: to classify a new example, kNN searches the training set for examples with similar sensor measurements; these can easily be found if all the participants repeat the same script and wear the sensors in the same position. In-the-wild, however, such a model can fail. Also, kNN requires retaining many examples and comparing them to every new example. This scales badly to larger training sets

and is not practical for mobile or real-time applications. Additionally, when conducting these controlled experiments, researchers made sure that their datasets will be nicely balanced and that all required sensors were used. For such well crafted datasets, off-the-shelf classification tools may be appropriate. That may not be the case when collecting data in-the-wild, especially at a large scale.

3.3.2 Getting out of the lab

Natarajan *et al.* [62] worked on using wearable Electrocardiography sensors to detect cocaine use. They addressed problems that arise when training classifiers on lab data and validating them on data from the field (in-the-wild). These problems include two types of distribution shifts from lab data to field data: prior probability shift, referring to class distribution ("cocaine use" *vs.* "not cocaine use") and covariate shift, referring to the distribution of sensor features. For both these problems, their solution involved training the model on the lab data using instance-weights that compensate for the distribution shifts and adapt the model to the distribution of the target domain (field). They also addressed the difficulty in collecting reliable ground truth labels in-the-wild, and decided to process the field data in a day-by-day granularity (unlike the five-minute windows in their lab data).

Ermes *et al.* [21] addressed some aspects of in-the-wild behavior. They followed their in-the-lab scripted data collection with an additional phase, where participants roamed freely and self-reported their behavior using a personal digital assistant (PDA) device. The labeling interface enabled selecting combinations of physical activity (one out of nine), location (indoors, outdoors, or vehicle), and indication of eating *vs.* not-eating. For recognizing the physical activity, they suggested a model that incorporates domain knowledge — a human-crafted decision tree structure that defined a hierarchical clustering of the activities. They added machine learning to the model — each decision in the tree was resolved with an MLP. The tree structure was able to represent similarity and grouping relations among the labels. However, hand crafting such a

structure depends on the researchers' assumptions, which may not hold in real life, especially when device placement is not controlled. Also, it is hard to scale such a structure to a wider range of labels or to contexts that involve multiple labels simultaneously, like sitting while watching TV. Finally, and most importantly, the sensing apparatus that they used was unnatural and inconvenient (it included wires that connected the sensors to a carried box).

Khan *et al.* [37] provided a phone to the participants and let them collect data in their natural environments for one month. For classification, they tried using support vector machine (SVM), Gaussian mixture models (GMM), and MLP, with or without kernel-discriminant-analysis (KDA) for feature transformation that reduces dimensionality and enhances discriminability. For KDA, they regarded within-class and between-class variability, so they relied on the multi-class formulation of behavior — where every example was assigned a single activity.

3.3.3 Transfer learning

Some studies described settings that have different types of context-labels, but did not model their interaction. Shoaib *et al.* [82] started with "simple" activities (body movements) and then extended their experiments to "complex" activities (*e.g.* smoking, typing). However, their scripted activities were repeated by participants, so they missed the rich variability of real-life behaviors; For instance, they did not simulate the combination "smoking while sitting".

Rossi *et al.* [75] collected sound clips, available on the web, that were annotated with tags describing locations (*e.g.* beach, office), inanimate objects (*e.g.* bus, washing machine), and live entities (*e.g.* speech, dog). However, they only assigned a single tag to each clip, missing combinatorial contexts, like "man speaking to dog at the beach". They also modeled each label with a separate GMM, potentially missing common relations, *e.g.* washing machine and dishwasher may produce similar sounds.

The *ExtraSensory Dataset* [91] contains labels describing different aspects and includes rich combinatorial contexts — on average every example was annotated with more than three

different labels. However, in the baseline system we initially proposed in [91], every label was modeled separately. The system was based on linear classifiers (logistic regression) and we focused the reported results on 25 labels for which recognition was successful. The separate model-per-label system missed the dependencies among related labels. This may have caused many labels with relatively few examples to have poor recognition; for instance, recognizing "Washing dishes" may be improved if it were modeled together with "At home".

Other works utilized transfer learning and explicitly modeled sharing information from one set of labels to another. Zheng *et al.* [103] used unsupervised learning to discover common behavioral patterns in a home environment, equipped with binary state-change sensors. They used a growing self-organizing map method to construct hierarchical clustering of the training examples. Such an approach can be used to expand an existing model to new data and capture new behaviors. Seiter *et al.* [79] described using topic modeling to discover new contexts — common temporal-sequences of "activity primitives". Pirsiavash *et al.* [68] collected data of daily activities from participants at their home with video recordings from a chest mounted camera. They annotated the recorded images for objects (where multiple objects can appear at the same scene) and actions (one out of eighteen). For low level recognition (object detection) they trained a separate model per object. Finally, they used predictions from these object-detectors to form a bag-of-objects histogram representation, and used it for classifying actions. In [103, 79, 68], the direction of transfer learning was clearly defined — there was explicit partition to lower-level and higher-level contexts. Such methods are less fitting in cases where there is not a clear hierarchy among labels.

3.3.4 Missing input data

Previous works evaluated recognition with different combinations of sensors by training a separate system for each combination [26, 82]. In [91], we evaluated single-sensor systems as well as sensor-fusion systems that use all six sensors. In-the-wild, it is likely that the combination

of available sensors will be constrained by the situation: the participant may decide to turn off a power-hungry resource (like location service) or to remove a watch, or one of the sensors can temporarily not work. To make sure the recognition system keeps working in such cases, a naïve option is to pre-load it with trained classifiers for different combinations of sensors, but this solution scales badly to many sensors.

Ngiam *et al.* [63] demonstrated the utility of learning a joint representation for two modalities — facial video and audio — for speech classification. They showed how a shared model can work well even with the absence of one modality. Mallidi *et al.* [52] worked on speech recognition where some of the input streams (spectral sub-bands) are distorted by noise. They showed that instead of training a separate model for each combination of streams, a single unified model, trained with stream-dropout, can capture different scenarios and work well when selecting the subset of less-noisy input streams. Lipton *et al.* [48] faced the real-world limitation of missing input data. They worked on prediction of clinical diagnosis in an intensive care unit. Their input was a collection of measurements that were taken at different rates: blood pressure was typically measured once an hour but urine samples could be taken once a day. They represented the data as time-series of hours, which resulted in many time points that had only part of the input values. They compared different ways to address the missing values, including zero-imputation, keeping the value from the most recent measurement, and adding missing-data indicators as input features.

3.3.5 Measuring recognition performance

When evaluating recognition for many labels with unbalanced data (as is the case with *ExtraSensory Dataset*), the performance metrics make a big difference. In [91] we discussed why the naïve accuracy is a misleading metric for unbalanced labels, and why it is important to balance the trade-off between competing metrics, like sensitivity and specificity. A common approach is to observe the sensitivity (recall) against precision or use their harmonic mean (F1).

However, precision and F1 are very sensitive to class skew. This makes it hard to interpret F1 since chance level itself can be very small for rare labels. Moreover, in the multi-label setting, when applying micro-average or macro-average, the summarized score can present undesirable and inconsistent trends: under-emphasizing or over-emphasizing the rare labels. This effect can result in misleading conclusions when comparing two systems [47]. Ward *et al.* [97] addressed these issues and suggested metrics that partition false negative rate (1-sensitivity) and false positive rate (1-specificity) to describe finer types of errors. In such metrics (as well as sensitivity and specificity) the denominator has ground-truth counts (*e.g.* in sensitivity it is the count of ground-truth-positive), which makes them unaffected by the class skew in the data. This is unlike precision, where the denominator is the count of predicted-positive. A convenient way to consider both sensitivity and specificity is simply to average them, resulting in the balanced accuracy metric [10], whose chance level is always 0.5. Balanced accuracy (BA) can be viewed simply as a fair (balanced) version of accuracy. As in [91], we focus on BA (averaged over labels) as a fair metric of performance.

3.4 Our contribution

In this paper, we use the MLP as a model for context recognition, but instead of using it as a black-box tool, as done in [53, 41, 26, 69], we manipulate it to fit uncontrolled in-the-wild data. We adjust the MLP's objective function in a non-standard fashion to handle unbalanced, incomplete labeling. Similar to [62], we train with instance-weighting to neutralize the effect of the skewed class distributions in the training set, but here we work with a multi-label setting, so instead of a single weight per example, we coordinate weights for each example-label pair. We also describe transfer learning by copying parts from one MLP into a new MLP to extend prediction to new labels, and sensor-dropout to make the MLP robust to missing sensors. In MLP, unlike the hand-crafted label taxonomy in [21], the relations among different labels are

not explicitly defined, but rather implicitly learned from the data and represented in the hidden layers. Unlike kNN [69, 82], MLP has a model size (and test run-time) that does not depend on the number of training examples, so it can be stored on a mobile device, and it is fitting for real-time applications.

Same as done in [91], we evaluate multi-label classification over the *ExtraSensory Dataset*. However, unlike the separate-model-per-label system we proposed in [91], here we perform multi-task learning and model all the predicted labels in a single MLP. We demonstrate the utility of sharing the same learned hidden representation among the labels. The resulting MLP works like a set of logistic regression classifiers (one per context-label) whose input is a common learned representation with reduced dimensionality (similar to [37]). The MLP's learned hidden representation is driven by supervised training, which acts as learning both the representation and the classifiers at the same time. This is different from the two-stage approach in [37], where they first used KDA to learn a representation, and then trained an SVM classifier.

Similar to [68], we demonstrate a transfer learning scenario where we start by learning good representations for predicting a basic set of labels, and then using these representations to classify another set of labels. Unlike [103, 79, 68], we offer a more flexible model: a researcher can either construct a fully multi-task MLP and train it with all labels together or can start by modeling one subset of labels and later extend to another. Unlike the unsupervised discovery of new contexts in [103, 79], we stay in the realm of supervised learning. Finally, we address the need for the system to work well with arbitrary subsets of available sensors. Our treatment of missing sensors is similar to the stream-dropout used in [52] or the simple zero-imputation used in [48], but we make sure to normalize the total contribution of the available sensors at every example. The model that we suggest in this paper addresses all these issues pertaining to research in-the-wild, making it a more fit model. To support it, we evaluate the model with data that was collected in-the-wild.

3.5 Methods

3.5.1 Multi-task multiple layer perceptron

Our recognition model is based on multiple layer perceptron (MLP) — a feed-forward neural network that has hidden layers, in addition to the input features and output labels. It processes an input feature vector $x \in \mathbb{R}^d$ with a sequence of J affine transforms, each followed by an element-wise nonlinear activation function. This allows all the sensor-features to be mixed together in a non-linear transformation (the first $J-1$ stages) to form a hidden representation, which is then shared for linearly-predicting all the labels (in the last affine transform). Unlike the model-per-label in [91], here we train a multi-task model, with multiple outputs for a whole set of L binary labels. Previous studies used MLP for the multi-class setting, where the purpose was selecting a single activity (the one with highest probability output) out of a set of mutually-exclusive options [53, 41, 26, 69, 21]. Here, we work with the richer multi-label setting, where multiple labels can be classified as positive simultaneously. For the hidden layers, we use a leaky rectified linear unit as activation: $g(v) = \max[\frac{v}{10}, v]$. For the output layer, we use the logistic function (sigmoid): $g(v) = \frac{1}{1+e^{-v}}$, to produce valid probability-outputs. The actual binary predictions are achieved by thresholding the continuous outputs by 0.5.

Formally, we can represent an MLP as a function $f : \mathbb{R}^d \to [0,1]^L$. For convenient notation, we define the function f as processing a batch of N examples, $f : \mathbb{R}^{N\times d} \to [0,1]^{N\times L}$ (although every example is processed independently of the others). This function is parametrized by the free parameters of the model — the weight matrix and bias vector of each affine transform:

$$\Theta = \{W_j, b_j\}_{j=1}^{J} \tag{3.1}$$

Training is done over a training set of N examples that have sensor features and incomplete labeling (for every example there is information about part of the labels). We denote the training

set with the feature matrix $X \in \mathbb{R}^{N \times d}$, the ground truth labels matrix $Y \in \{0,1\}^{N \times L}$, and the missing-label matrix $M \in \{0,1\}^{N \times L}$. To train the model, we define the following optimization problem:

$$\min_{\Theta} \left(\frac{1}{NL} \sum_{i=1}^{N} \sum_{l=1}^{L} \Psi_{i,l} c(f(X)_{i,l}, Y_{i,l}) \right) + \lambda \varphi(\Theta) \tag{3.2}$$

For every example i and label l, the entry's prediction cost is the traditional cross entropy loss:

$$c(\tilde{y}, y) = -\left(y \log(\tilde{y}) + (1-y) \log(1-\tilde{y})\right) \tag{3.3}$$

As a regularization term, we selected $\varphi(\Theta)$ to be the total Frobenius norm of the weight matrices of the model. This optimization problem is an instance-weighted version of maximum a posteriori probability (MAP) estimation, where $\varphi(\Theta)$ accounts for the prior.

The nontraditional element here is the instance-weighting matrix Ψ. For entries (i,l) that are regarded as "missing label" ($M_{i,l} = 1$), $\Psi_{i,l}$ is set to zero, to make sure this example-label pair contributes nothing to the total cost. The other entries are normalized for each label l by their class (positive or negative), with weights that are inversely proportional to the frequency of that class for this label in the training set. As a result, for every label, the total contribution of the positive examples is equal to the total contribution of the negative examples. Instance-weighting is a common practice when training a single-output binary classifier, where every example gets a single weight according to its class. Ψ is a generalization of that practice to multi-label — it coordinates the positive/negative balance for all the outputs. This weighting is very important because our data is very unbalanced: generally, there are more negative examples than positive examples, and for every label the ratio is different. Without the weighting matrix, the learned model tends to be trivial — always declaring "no" for most labels. The weighting matrix also incorporates the missing label information, which enables training a multi-task model when the data has incomplete labeling. In this way, for instance, an example that only provides information

about body-state ("Walking", "Sitting", *etc.*) can still contribute its share to the training: we do not have to throw it away because it does not specify environment information ("Outside", "At home", *etc.*).

In all our experiments, we used early fusion of the sensors (the input layer is the 175 features from the six sensors). Training was done using gradient descent with back-propagation, for forty epochs, with mini-batch size of 300 examples. The learning rate was linearly decreasing at every epoch, from 0.1 to 0.01. We used momentum with weight 0.5.

3.5.2 Data preparation

For evaluating our model, we use our *ExtraSensory Dataset* [91]. We follow the same evaluation as done in [91]: We perform five-fold cross validation using the same partition of the sixty participants to five folds. We use the same six core sensors and the same $d = 175$ extracted sensor-features. We standardize each feature according to the mean and standard deviation estimated from the training set.

In [91], every example-label entry was treated as either positive or negative. For this paper, we further processed the ground truth labels and added a representation of "missing label information". During data collection the participant could only report positive labels by selecting the relevant labels from the large menu. The original analysis assumed that whenever a label was not marked, it was not relevant to the example. Here, we applied several common sense rules to infer when it is better to treat an entry (example-label pair) as *missing* rather than negative (see "Missing label information" in supplementary material for details). This label-cleaning may get rid of some actual negative examples, but the resulting labeling is more reliable. This is more crucial for the labels that were reported less, and may have been overlooked by many participants. For this paper, we calculate performance metrics by counting correct classifications and errors only over non-missing entries.

3.6 Experiments and results

We begin our experiments with the full multi-task MLP — one that is trained with all the labels (3.6.1). We compare it to the baseline system (referred to as early-fusion in [91]) — a separate logistic regression model per-label; here we refer to this baseline system as LR. We analyze the recognition performance, and the size of the different models (number of parameters). We then provide additional control experiments, to examine the contribution of the various techniques employed by the system (3.6.2). Next, we present experiments of transfer learning, where we extend the model to new labels (3.6.3), and for making the MLP robust to missing sensors (3.6.4). We analyze the experiments to gain insight about how the MLP uses the different sensors to recognize different behavioral contexts (3.6.5). Finally, we validate our model on an additional dataset and report similar results (3.6.6).

3.6.1 Multi-task MLP

The basic experiments here are with a multi-task MLP that predicts $L = 51$ context labels simultaneously. In [91], the main results focused on 25 labels with successful recognition. Here, we jointly model all these labels together with additional 26 labels that got poorer results with the baseline system. We evaluate different architectures of MLP, with no hidden layers (the linear case), or with one or two hidden layers of different widths.

When training a model, to select values for the hyper-parameters, we employ an internal validation procedure: The training set is partitioned to 70% internal-training-set and 30% internal-validation-set. With a grid-search over possible hyper-parameter values, we train on the internal-training-set and test BA on the internal-validation-set. We select the hyper-parameter values that yield highest validation-BA, and use them to re-train the model over the entire 100% of the training set. For each separate label-model in the LR system, the grid-search values for the hyper-parameter C_{logist} were $\{10^{-6}, 10^{-5}, \ldots, 10^{2}\}$, and the 70%/30% partition

of the training examples was done while maintaining the ratio between positive and negative examples. For MLP with a specific architecture, the grid-search values for the hyper-parameter λ were $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$, and the $70\%/30\%$ partition of the training examples was done randomly, because there was no way to guarantee the same positive/negative ratio for all the 51 labels. In addition, we perform experiments where the depth of the MLP (one or two hidden layers) is fixed, but the grid-search is done to select both λ and the dimension of the hidden layers (width), among $\{2, 4, 8, 16, 32\}$.

Table 3.1 presents recognition performance, including the baseline system from [91] (LR). As discussed in [91], the performance metrics for the LR system show that the accuracy metric is mostly dominated by the specificity (since there are many negative examples). The accuracy almost ignores the sensitivity (since there are fewer positive examples), hence it is misrepresenting the quality of the system. Switching to the linear-MLP causes a decrease in specificity but a much greater improvement in sensitivity. The balanced accuracy (BA) metric appropriately captures this overall improvement. Both the LR and linear-MLP systems solve similar optimization problems. However, in the MLP system, we use a single unified value for the balancing hyper-parameter (λ) for all the labels, unlike LR, where we tuned the hyper-parameter (C_{logist}) separately for every label. The difference in performance can indicate that this tuning caused the LR system to over-fit (indeed, the last two columns of table 3.1 show that the LR system had a larger gap between train performance and test performance).

Adding a hidden layer to the MLP introduces nonlinearity and dimensionality reduction to the model. When the features are extremely compressed (only two hidden nodes), BA decreases. With a wider bottleneck hidden layer, the MLP can express richer relations and performance increases. Adding a second hidden layer is another way to increase the expressiveness of the MLP. When selecting an architecture (*e.g.* width and depth of MLP), researchers should observe the competing performance metrics (or use a "fair" combination, like BA), but should also consider the model's size — the number of parameters (we specify the size of each model in the

first column of table 3.1). The linear models in our experiments require tuning and representing 8,976 parameters (including weights for all the combinations of 175 features and 51 labels). A bottleneck hidden layer can greatly shrink the model: for example, with a single hidden layer of sixteen nodes (MLP (16)) the total number of parameters is 3,683 ($175 \times 16 = 2,800$ from W_1, 16 from b_1, $16 \times 51 = 816$ from W_2, and 51 from b_2), which is less than half the linear model's size. Having a smaller model is an optimization constraint that can help prevent over-fitting. MLPs with one or two hidden layers of 64 nodes are larger (have more parameters) than the linear model, and indeed we see that these large MLPs are more prone to over-fitting: their recognition performance on the training examples (train-BA column) is higher than for the other MLPs, while their test performance (BA column) is lower than for smaller MLPs. Generally, it seems that the smaller the model size, the smaller the train-test gap ("BA gap" column). This can explain the advantage of moderately-sized MLPs that have enough expressiveness to predict 51 labels, while having less parameters than the linear model.

Summary of results: These basic experiments show the superiority of the MLP over the baseline. MLP manages to capture good predictive mappings from sensors to many diverse labels, all in a concise representation. By selecting a moderately-sized architecture (large enough, but no more parameters than the linear model), we can balance the trade-off between capturing many contexts and generalizing to unseen data. The improved recognition compared to the linear system can be attributed to the nonlinearity and the dimensionality reduction in the bottleneck hidden layers. The improvement can also be explained by the fact that we model all the labels with a shared structure, unlike the separate model-per-label in the baseline.

3.6.2 The performance gain

The multi-task MLP adds nonlinearity, hidden layers, dimensionality reduction, and sharing of parameters (among labels), all in one supervised-learning framework. In this section, we describe control experiments, to better understand which of these techniques contribute to the

Table 3.1: Recognition scores reported for baseline system (LR — logistic regression per-label), and for the multi-task MLP (either linear or with the dimensions of the hidden layers in parenthesis). For each network we specify its size — the number of free parameters (including weight matrices and bias vectors). MLP (d) represents a model with a single hidden layer, where the hidden dimension is selected via internal validation among $\{2,4,8,16,32\}$. MLP (d,d) represents a model with two hidden layers, where the hidden dimension used for both layers is selected via internal validation among $\{2,4,8,16,32\}$. For MLP (d) and MLP (d,d), the architecture can be different for each test fold, so the model size is not determined. Scores are averaged over all the 51 labels. For balanced accuracy (BA), the last two columns report the score measured on the training examples (train-BA) and the gap between training and test performance (BA gap), to assess the level of over-fitting.

	size	**accuracy**	**sensitivity**	**specificity**	**BA**	**train-BA**	**BA gap**
LR	8976	0.832	0.597	0.838	0.718	0.875	0.158
MLP (linear)	8976	0.760	0.746	0.757	0.752	0.813	0.061
MLP (2)	505	0.666	0.773	0.661	0.717	0.735	0.017
MLP (4)	959	0.730	0.773	0.727	0.750	0.773	0.023
MLP (8)	1867	0.776	0.768	0.775	0.772	0.806	0.035
MLP (16)	3683	0.781	0.755	0.781	0.768	0.820	0.052
MLP (32)	7315	0.799	0.736	0.800	0.768	0.847	0.079
MLP (64)	14579	0.806	0.687	0.808	0.747	0.865	0.118
MLP (d)	?	0.799	0.736	0.800	0.768	0.847	0.079
MLP (2,2)	511	0.662	0.759	0.656	0.707	0.736	0.029
MLP (4,4)	979	0.707	0.769	0.707	0.738	0.763	0.025
MLP (8,8)	1939	0.761	0.772	0.759	0.766	0.803	0.037
MLP (16,16)	3955	0.773	0.773	0.773	0.773	0.817	0.044
MLP (32,32)	8371	0.805	0.729	0.807	0.768	0.845	0.078
MLP (64,64)	18739	0.817	0.661	0.823	0.742	0.877	0.135
MLP (d,d)	?	0.805	0.729	0.807	0.768	0.845	0.078

gain in performance.

Instance-weighting: We stated that instance-weighting is especially important when the training data is very unbalanced. Both the baseline (LR) and the multi-task MLP we presented already employ instance-weighting. To account for the importance of this technique, we conduct corresponding experiments without instance-weighting (for the MLP experiment, this means that Ψ simply has binary values — indicating for each entry if it is non-missing: $\Psi = not(M)$). Table 3.2 shows the results with and without instance-weighting. The baseline LR system (with instance-weighting, first row in tables 3.1 and 3.2) has some discrepancy: it has much better performance for negative examples (specificity) than positive examples (sensitivity). However,

when training LR without using instance-weighting, this discrepancy is much more severe, and the fair metric of balanced accuracy demonstrates this degradation. For multi-task MLP with two hidden layers of sixteen nodes, we observe a more drastic effect, demonstrating how crucial instance-weighting is for unbalanced multi-label datasets. Without instance-weighting, the resulting models optimize the raw accuracy, which makes them neglect the rare cases (positives) and produce almost-trivial classifiers.

Table 3.2: Effect of instance-weighting. LR and MLP with two hidden layers of sixteen nodes, for each — performance with and without instance-weighting.

	accuracy	**sensitivity**	**specificity**	**BA**
LR	0.832	0.597	0.838	0.718
LR, no instance-weighting	0.918	0.256	0.940	0.598
MLP (16,16)	0.773	0.773	0.773	0.773
MLP (16,16), no instance-weighting	0.935	0.145	0.959	0.552

Nonlinearity and hidden layers: To examine whether non-linearity and hidden layers are the main contributors to the improvement of the multi-task MLP, we experiment with systems that have separate MLPs per label, with one or two hidden layers of one, two, or four nodes. These systems add the richness of MLP to each label's model, allowing it to express more complicated mappings from features to the target label, but they do not share information among labels. For each label, the value of λ was selected via internal validation and grid search over $[0.0001, 0.0005, 0.001, 0.005]$.

Table 3.3 shows the results of these experiments. All of the separate MLP per-label systems performed roughly at the same level as the baseline LR system, meaning that the added non-linearity and hidden layers did not add much improvement, certainly not enough to compete with the multi-task MLP. This approach to increasing the richness of the system comes at the price of over-fitting. Without shared parameters, each label's separate model has to grow, causing the size of the whole system to blow up. This adds much richness to the overall system, as seen by the increasing BA on the training set. However, there is too much richness (too many parameters) and it causes the system to severely over-fit: BA train-test gap increases with increasing model

size.

Table 3.3: Effect of non-linearity and hidden layers. In the per-label experiments, each label has a separate MLP model.

		size	**accuracy**	**sensitivity**	**specificity**	**BA**	**train-BA**	**BA gap**
MLP (1) per label	$51 \times 178 =$	9078	0.837	0.601	0.845	0.723	0.881	0.158
MLP (2) per label	$51 \times 355 =$	18105	0.830	0.622	0.837	0.729	0.889	0.160
MLP (4) per label	$51 \times 709 =$	36159	0.843	0.576	0.850	0.713	0.919	0.206
MLP (1,1) per label	$51 \times 180 =$	9180	0.832	0.605	0.839	0.722	0.880	0.158
MLP (2,2) per label	$51 \times 361 =$	18411	0.825	0.620	0.831	0.726	0.900	0.174
MLP (4,4) per label	$51 \times 729 =$	37179	0.852	0.553	0.863	0.708	0.924	0.216

Parameter-sharing: We can compare the MLP-per-label systems to a multi-task MLP with comparable size (number of parameters), for example the "MLP (1) per label" system has a total of 9,078 parameters, close to the size of the multi-task "MLP (32,32)". According to this criterion, we already saw that multi-task MLPs with comparable sizes outperform an MLP-per-label system. Another comparison criterion is the node-wise architecture — the number of hidden layers and hidden nodes: A multi-task MLP with a hidden layer of 51 nodes (shown in table 3.4) has the same node-wise architecture as the "MLP (1) per label" system (the difference is in the connectivity among the nodes), and a multi-task MLP with two hidden layers of 51 nodes has the same node-wise architecture as the "MLP (1,1) per label" system. According to this criterion as well, when comparing a separate-MLP-per-label system to a comparable multi-task MLP, the multi-task MLP performs better. This indicates that the sharing of parameters is beneficial for the model.

Table 3.4: Multi-task MLPs with node-wise architecture that is comparable to an MLP-per-label system.

	size	**accuracy**	**sensitivity**	**specificity**	**BA**	**train-BA**	**BA gap**
MLP (51)	11628	0.803	0.719	0.806	0.762	0.855	0.093
MLP (51,51)	14280	0.803	0.712	0.805	0.759	0.857	0.099

Shared representation with dimensionality reduction: One feature of the multi-task MLPs that we analyze here is dimensionality reduction. Having a hidden layer with smaller dimension than the input-features contributes to reducing the number of parameters to get better

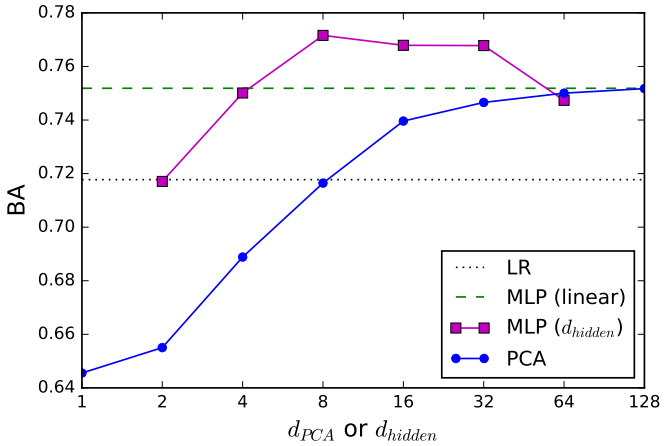

Figure 3.1: Dimesionality reduction. Comparing reducing dimension by PCA to a hidden layer of a multi-task MLP.

generalization. In the MLP, the hidden representation (with the reduced dimension) is learned together with the classifiers/output layer via supervised learning. An alternative is to learn a representation using unsupervised learning. Here, we examine the most basic unsupervised method to learn a reduced dimension representation — principal component analysis (PCA). We estimate (over the training set) the PCA projection of the features to different reduced dimensions, and then train a linear-MLP from the projected representation to the output labels (with validation-selection of $\lambda \in [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]$).

Figure 3.1 shows the results with PCA dimensionality reduction to different dimensions, as well as the corresponding multi-task MLPs with a single hidden layer of the same dimension. With increasing PCA-dimension, performance increases and reaches the linear system with the full dimension — MLP (linear). This means that dimensionality reduction alone does not contribute to gain in performance. On the other hand, training a multi-task MLP with a single hidden layer results in better reduced-dimension representations, which contributes to better performance.

Summary of results: All the different techniques combined in our suggested multi-task MLP contribute to improved performance. Instance-weighting is crucial to avoid trivial

classifications. Non-linearity (through hidden layers) alone is not enough to improve performance of separate per-label models. The sharing of parameters is important to allow for rich mappings while avoiding too many parameters. Finally, even a shared representation and dimensionality reduction are not enough to provide the full performance gain (as seen in the PCA experiment); Supervised learning of all the layers makes sure we learn a good (useful) hidden representation — a representation that carries important information for predicting the labels.

3.6.3 Transfer learning for new labels

In this section, we simulate a practical scenario that may occur in research. In the scenario, researchers first collect labeled data targeting a starting-set of labels and train a basic context recognition system to predict these labels. Later on in the scenario, the researcher wish to extend their system to recognize a new-set of labels, addressing a different behavioral aspect that they did not have in mind earlier, so they collect additional labeled data for the new-set of labels. We examine whether we can use transfer learning, to take advantage of the already-trained model, when training the new model for the new-set of labels.

In the following experiment, we first train an MLP (with two hidden layers of sixteen nodes) to predict a starting-set of L_s labels. Next, we train a new MLP (also with two hidden layers of sixteen nodes) to predict a new-set of L_n new labels (the complement set, out of the 51 labels in the paper). For the new-set we have three options:

1. Fresh: Start from scratch, meaning randomly initialize the entire network and train it.

2. Copy: Copy the first two affine transforms (W_1, b_1, W_2, b_2) from the starting-set MLP, replace the last affine transform (the one closest to the output — W_3, b_3) with a new one. Then train the entire network.

3. Copy-freeze: Construct the new network same as in the Copy option, but when training, freeze the copied components, and only update the parameters of the new last transform,

W_3, b_3.

In the multi-task MLP experiment presented earlier, with the MLP (16,16) architecture, the internal-validation procedure consistently (for the five folds) selected the value $\lambda = 0.001$, so for the transfer learning experiment, we use a fixed value of $\lambda = 0.001$. The two training phases in these experiments (starting-set and new-set) are never exposed to the explicit co-occurrence of labels from the starting-set and the new-set.

Intuitively, it may seem better to fully optimize the new network for the new labels, but if the new data is limited this can cause over-fitting. In that respect, utilizing the starting-set MLP can act as a "warm-start" regularization.

We examine three sets of labels as the new-set (for each, the starting-set is all the other labels, among the 51):

- Body-state: {"Lying down", "Sitting", "Standing", "Walking", "Running", "Bicycling"}.
- Home-activities: {"Cooking", "Cleaning", "Doing laundry", "Washing dishes", "Grooming", "Dressing"}.
- Environments: {"In class", "In a meeting", "At main workplace", "At home", "At a restaurant", "At a bar", "At a party", "At the beach", "At the gym", "At school"}.

A new-set MLP still has a multi-label output and it can still be regarded as multi-tasking. However, here we reserve the name "multi-task MLP" to the full 51-label model, which predicts many labels across different behavioral aspects.

Table 3.5 presents the results from these experiments. The basic option of training a new MLP with the new data (Fresh) achieves some improvement compared to LR. This can be the result of the added nonlinearity, dimensionality reduction, and sharing information among the labels of the new-set. The Environments set has a wider range (10 labels), which may explain the significant improvement. When utilizing the first few levels of the starting-set MLP as a starting

point for training (Copy), there is another slight improvement. However, it seems that when updating the entire new network with the new label set, the model can "forget" what it already learned before and lose the advantage of the richer starting data. The option of maintaining the previously learned hidden representation and only updating the output level (Copy-freeze) works as a stronger regularization on the training for the new-set, preventing it from losing what was already learned. Indeed, this option shows a stronger improvement in performance. The Copy-freeze option reaches similar level of performance as the full multi-task MLP (trained for 51 labels), even though the multi-task MLP had an advantage of being trained with more information — the full combinations of all the labels. Of course, the success of transfer learning relies on the assumption that there is a strong-enough statistical dependence between the starting-set and the new-set of labels; it is possible that hidden representations that are trained to be informative for human behavioral contexts will not be so useful to recognize other contexts, like weather.

Summary of results: These experiments show that there is a clear advantage of sharing a model (through the hidden layers of an MLP) among labels. The shared model helps boost recognition of new labels both when the model is trained with all the data (with the old and new labels together) and when the new labels appear in separate data. It is possible that the researchers collecting the new data do not have access to the full data from the starting-set of labels, but they only have the trained model for the starting-set (*e.g.* if they received the trained model from other researchers). In such a case, they can still take advantage of the old data indirectly, through the concise structure of the trained model.

3.6.4 Missing sensors

A practical system that works in-the-wild has to face situations where some sensors may be missing. In our data collection, we naturally encountered such cases. Most notably, the participant sometimes turned off location services to conserve battery, and sometimes removed

Table 3.5: Transfer learning to a new set of context-labels. BA scores averaged over the new set of labels. Reported scores for LR, and for MLP (with 2 hidden layers of dimension 16). MLP for the new-set (new-set MLP (16,16)) was trained with either the Fresh, Copy, or Copy-freeze option. Multi-task MLP (16,16) was trained with all 51 labels (but scores are averaged only on the new-set labels).

		new-set MLP (16,16)			
new-set	LR	Fresh	Copy	Copy-freeze	multi-task MLP (16,16)
Body-state	0.771	0.778	0.783	0.803	0.801
Home activities	0.647	0.654	0.657	0.718	0.722
Environments	0.735	0.788	0.791	0.799	0.801

the watch, for convenience. The average-probability late-fusion method presented in [91] has the potential to handle such cases by averaging the prediction outputs for the sensors that are currently available. However, the late-fusion approach misses the opportunity to model correlations between features of different sensors. In our MLP, where all the sensors' features are presented in the input layer, there is the potential for sensors to learn from each other. To handle missing sensors with the MLP, we suggest the dropout technique in a structured manner: the input features of the missing sensor(s) are set to zeros, and the features of the available sensors are multiplied by an appropriate weight to keep the total contribution of the input features in a standard level. For example, if four out of the six sensors are available, we multiply their features by $\frac{6}{4}$. During training, for every mini-batch, we randomly mask some sensors as "missing" — independently masking each example-sensor entry with probability p_{drop}.

Dropout was originally presented as a method to avoid over-fitting and train more robust networks that do not rely too much on specific nodes [83]. Traditional usage of dropout is for training alone, and it expects all the input features to be available at recognition time. In [52], the authors applied dropout also at recognition time: their system attempted different combinations of input streams, and used performance-monitoring metrics to identify the less noisy combinations. Similarly, we also wish to use a single network to handle all different scenarios of available input streams (sensors, in our case); and we also employ dropout only on the feature layer, in structured blocks of features. However, unlike [52], our motivation is to handle cases where sensors are

actually missing; and our system uses all the available sensors at recognition time, instead of selecting a less noisy subset of sensors. In [91], the early fusion classifiers were trained with only the examples that had all the six sensors. Here, the formulation of how to handle missing sensors enables us to use the full training data, including examples that were collected with missing sensors.

In this experiment, we again use the MLP (16,16) architecture, and again, we fix λ to a value of 0.001. Table 3.6 presents BA scores for training with and without sensor-dropout. The basic MLP (first row) was trained with all sensors available (hence, was limited to use only the core subset of the training examples). This MLP is not so sensitive to the lack of signal from Acc or Gyro, indicating that these two sensors carry a less unique signal, which can be recovered from other sensors. It is much more reliant on the phone-state modality (PS) — without PS there is a large drop in performance. Fortunately, this is the cheapest source of information — these indicators are readily available on the phone's operating system, so there is no practical concern of missing PS. On the other hand, it is very reasonable to be in a situation where WAcc, Loc, or Aud is missing. These three modalities also contribute important information to the system (missing one of them reduces performance).

Training with dropout (with $p_{drop} = 0.2$, second row) generates a more robust MLP, that generalizes slightly better when all six sensors are available (per-label scores for this robust MLP are provided in supplementary material). More importantly, the dropout-MLP can better withstand any missing sensor, and reach performance closer to when having input from all sensors. We also evaluated two specific scenarios, when only three sensors are available, that can reasonably occur in practical applications. Both scenarios simulate that the extra device (watch) is missing and the power-hungry location service is turned off. AGP uses only accelerometer, gyroscope, and phone-state and AAP uses only accelerometer, audio, and phone-state. In both these cases training with sensor-dropout improves the performance.

Summary of results: These experiments demonstrate that some sensors are very impor-

tant for successful recognition and show that with proper training (using sensor-dropout), the model can be more resilient to losing these sensors.

Table 3.6: Handling missing sensors. Balanced accuracy (averaged over the 51 labels) scores. Tested on the core examples (those that have all six sensors) with all sensors available, with simulating one missing sensor, and with simulating specific reasonable combinations of sensors: AGP represents Acc, Gyro, and PS; AAP represents Acc, Aud, and PS. All experiments are with MLP with 2 hidden layers of dimension 16. Models were trained either on core examples without dropout (first row), or with all training examples with dropout (second row).

		5 sensors (all except one)						3 sensors	
Training	6 sensors	Acc	Gyro	WAcc	Loc	Aud	PS	AGP	AAP
core examples, no dropout	0.773	0.771	0.770	0.753	0.763	0.746	0.737	0.704	0.733
all examples, dropout	0.780	0.778	0.777	0.764	0.770	0.763	0.757	0.730	0.748

3.6.5 Interpreting the trained MLP

It is not straight forward to interpret a trained MLP; the learned mappings from sensor-features to context-labels may be non intuitive to understand and because all the features are mixed together through multiple layers, it is not always possible to identify specific features that are used to recognize specific labels. In order to gain insight about how this model allocates its resources and uses the different parts of the sensor input to recognize the broad range of context labels, we utilize the missing-sensors experiments, described in the previous section (3.6.4). We observe the recognition performance (balanced accuracy) averaged over subsets of context-labels, each representing a different behavioral aspect, same as the label subsets we used in the transfer-learning experiments (3.6.3). For this analysis we use the MLP (16,16) model that was trained with sensor-dropout.

Table 3.7 shows how the model performs for various subsets of labels (rows) when it is presented with the full features from all the six sensors, and when it is presented with one sensor missing. The results show that when missing the watch (WAcc), performance degrades for recognizing body-state contexts (balanced accuracy drops from 80% to 77%), but recognition of body-state is almost unaffected when any of the other sensors is missing. This means that

the MLP learned to rely mostly on the watch acceleration for its recognition of the different body states. Similarly, the model relies heavily on the watch for recognizing home activities (like cooking, cleaning, or washing dishes; see the full list of labels in section 3.6.3), but it also "invests" in location and audio for recognizing these activities. For recognizing various environments, the model relies mostly on the audio and phone-state modalities. The MLP relies heavily on phone-state and audio also to recognize the position of the phone, and in a lesser degree, on gyroscope. The audio may give good indication for whether the phone is enclosed (in pocket or in bag) or exposed to external sound (in hand or on table). Interestingly, when masking the features from the watch, the MLP performed better for the phone-position context (which makes sense since the watch is a separate device than the phone); this implies that within the limited resources (the hidden nodes for internal representation) the MLP had to capture some signal from the watch and this signal adds a bit of distraction from the relevant information about the phone-position.

As seen in the overall results from the missing-sensors experiments (section 3.6.4), the phone's accelerometer and gyroscope are usually able to compensate for one another, so masking one of them does not create significant degradation in performance. This is understandable, seeing as they both measure physical properties that are based on the same thing — the motion of the phone.

See supplementary material for per-label results. We also provide node-activation analysis on a smaller MLP, to get basic intuition about the potential of MLPs to capture information from multi-modal sensors in a concise representation that is meaningful for a broad range of contexts (see "Interpreting the multi-task MLP" in supplementary material).

3.6.6 External validation

In order to further validate our suggested model, we apply it to an additional dataset. Pirsiavash and Ramanan collected the "Activities of Daily Living (ADL)" dataset and published

Table 3.7: Relations between sensors and contexts. Balanced accuracy with MLP (16,16) that was trained with sensor-dropout. For various behavioral aspects (rows), the table presents the average (over the context-labels in the relevant subset) score, tested with all the 6 sensors available, and with masking each of the 6 sensors. These results help interpret the learned MLP, and how it uses the different sensors to recognize different contexts. For each subset of labels (row), the most relied-upon sensor is highlighted in gray. The "Phone position" label-subset includes the 4 context-labels describing different phone positions (in hand, in pocket, in bag, and on table).

		5 sensors (all except one)					
Label-set	all sensors	Acc	Gyro	WAcc	Loc	Aud	PS
Body-state	0.801	0.799	0.797	0.772	0.800	0.797	0.792
Home activities	0.740	0.739	0.739	0.673	0.726	0.722	0.729
Environments	0.810	0.810	0.812	0.806	0.792	0.776	0.773
Phone position	0.760	0.756	0.751	0.769	0.756	0.744	0.733

it with their analysis paper [68]. The dataset contains images from a chest mounted camera from twenty participants engaged in free daily living activities in their own homes. The images were annotated for objects and actions (brushing teeth, making coffee, watching TV, *etc.*).

We perform the action-recognition task that they describe in [68], where each item is a pre-segmented clip annotated with a single action out of eighteen. We report average accuracy (average of the binary-recall over the eighteen actions). We repeat the same processing stages, including using the object detection scores from 26 object models, and calculating temporal pyramid to produce a 78-dimensional feature vector for each clip. As baseline, we used the original linear SVM (one *vs.* rest) multi-class classifier. We experiment with our multi-task MLP, with slight modifications to apply it to a multi-class problem: we add a "softmax" activation at the output (normalizing the eighteen output probabilities to form a categorical distribution over the eighteen actions); we represent the ground truth of a clip as a vector of 0s with only a single 1 value for the relevant action; when classifying a clip, we report the action with highest output value. For MLP, we experiment with a linear model (zero hidden layers), or with one or two hidden layers of various dimensions. We use a fixed value of $\lambda = 0.001$. Because this dataset has fewer examples than the *ExtraSensory Dataset* (203 pre-segmented clips), we replicate training examples to approximate 100 per action (same as in Pirsiavash's experiments) and use smaller

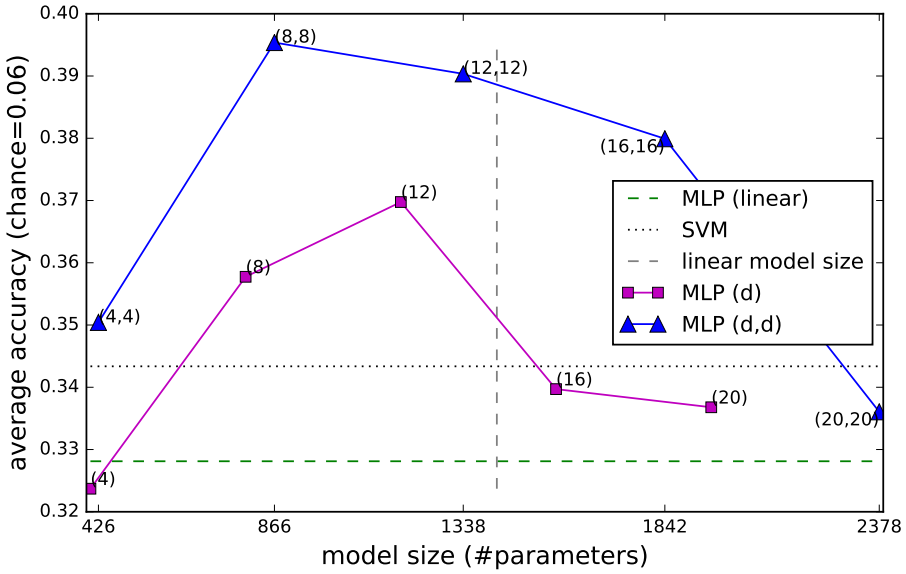

Figure 3.2: Activities of Daily Living (ADL) dataset. Performance of the baseline from [68] (SVM) and linear-MLP are presented in flat horizontal lines. Performance of multi-task MLPs with one hidden layer (purple squares) or two hidden layers (blue triangles) are shown with the hidden layer dimensions next to each point. Models are arranged on the x-axis according to their size (number of parameters). As reference, the vertical dashed line marks the size of the linear models (SVM and linear-MLP each have 1,422 parameters). The plots show that a multi-task MLP can outperform linear models, when it has "wide enough" hidden layers (8, 12 nodes), but still "narrow enough" to keep the model size smaller than the linear model.

mini-batches of 10 examples. All experiments were done with leave-one-participant-out over participants 7–20, and participants 1–6 were only used to validate the choice of hyper-parameters.

Figure 3.2 shows the results. Again, we see that a multi-task MLP can out-perform a linear model (including the SVM). We see the same trends as with the *ExtraSensory Dataset*: adding shared hidden layers increases the richness of the system, causing gain in performance, but up to a certain limit. When increasing the hidden dimension further, performance decreases (as a result of over fitting). Again, we see that a good measure for the tendency to over-fit is the model size, where a good reference point to compare to is the size of the linear model (vertical dashed line in figure 3.2). Same as in our previous experiments, here we see that the models that performed best have less parameters than the linear model.

3.7 Discussion

In order to address in-the-wild behavior, research needs to better represent it. The traditional multi-class approach, which selects a single activity from a small set of mutually-exclusive options, is a naïve way to describe behavior. On the other hand, the multi-label approach, where multiple labels can apply simultaneously, provides more richness and allows to describe behavioral context as a combination of multi-dimensional aspects: people do not just "sit" but rather "sit at school, doing computer work"; "sit at a bar with friends"; or "sit at home, with family, watching TV". Having a multi-task model, like the MLP we present here, enables modeling the complexity and richness of in-the-wild behavior.

The multi-task MLP employs various techniques that are essential to its performance gain, as seen by our extensive experiments:

- Instance-weighting is crucial to avoid trivial classifiers that neglect rare contexts.
- Non-linearity and hidden layers are not enough. They certainly add richness (increasing performance on the training set) but when applied to separate per-label models the added richness results in severe over-fitting. Parameter-sharing across labels is needed to make the overall system generalize well to unseen examples.
- Parameter-sharing and dimensionality reduction are also not enough to get the full gain. Unsupervised methods to learn a hidden representation (like PCA) only care about capturing statistics of the input-features. In the multi-task MLP, the supervised training of all the layers makes sure that the hidden representation captures relevant information for predicting the output-labels.

The multi-task MLP has flexibility to learn interesting inter-label and sensor-label dependencies without the guiding hand of the researcher. Potentially, labels with strong recognition (or with many examples) can help boost the performance of related labels that have weaker

recognition when modeled separately. Some sets of labels, like {"Indoors", "Outside"}, have clear relations, and it makes sense to jointly model them. The model may implicitly learn co-occurrence patterns, like "Sleeping" mostly occurs in conjunction with "At home". In our experiments, the MLP learned which sensors to rely on more for recognizing different contexts (*e.g.* watch accelerometer for home activities or audio and phone state for environments).

Different people engage in different contexts. Some like to cook while others prefer eating out; some drive to work every day while others bike; some hang out with friends at bars while other prefer to stay home and watch TV. The multi-task model can extract the similar patterns between different contexts and their differences. The training methods we present in this paper enable using data from different people, where each person contributes to some contexts while ignoring others.

The transfer learning experiments highlight the advantage of sharing information in a unified model, even among labels that describe different aspects of behavior (activities *vs.* environments *vs.* body posture, *etc.*). These results are also encouraging for the practice of building context recognition systems: if data collection is done in phases addressing different target labels, the new system can rely on a previously trained system — this is especially important if the new data is smaller in size compared to the starting-data. Depending on the amount of previous and new data, researchers have the flexibility to balance the impact of both data parts, using combinations of the Copy and Copy-freeze methods we describe here.

The size of a model (number of parameters) has an effect on both the compatibility of the model for practical applications and on its generalization to unseen data. Unlike k-nearest-neighbors, the MLP's size does not depend on the size of the train set, and it does not require comparing a new example to training examples. An MLP can out-perform a linear model, without increasing the size of the model. In fact, when the MLP's bottleneck is narrow enough and the total number of parameters is less than the linear model's, it contributes to better generalization of the trained model to unseen data. We see this effect in both the *ExtraSensory Dataset* and

the ADL dataset (section 3.6.6). In that respect, we can view the linear model (which assigns a separate weight parameter for each combination of input-feature and output-label) as wasteful and prone to over-fitting. This is even more severe in the LR system, when also the hyper-parameter is fitted separately for each label (adding fifty extra degrees of freedom, compared to the single shared value of λ in the multi-task MLP). A multi-task MLP distributes the limited resources (the parameters of the model — the inter-node connections and the node-biases) more efficiently. It can re-use common calculations instead of repeating them for each label, and ultimately capture more complex mappings from features to labels by using less parameters than the linear model. Having a multi-task MLP with smaller size than a linear model also means that it is fit for real in-the-wild usage. Applications can hold a fully trained multi-task MLP on a smartphone or on a web-server, and have it recognize context in real time.

In addition, with proper training, the model can withstand missing sensors, in realistic situations like when the person removes a watch or when location services are not available. This is important to make applications work seamlessly in real life. Our experiments also demonstrate using data with missing sensors for training. This encourages further collection of research data in-the-wild. Researchers can let many participants collect data from their own various environments. Even if some participants never used a watch or often turned off some sensor, all the partial data can be combined to train a single model.

3.7.1 Future Improvements

Despite the progressive steps we offer here, further improvements are still recommended to promote research in-the-wild that is even more ecologically valid. Semi-supervised methods can be used in order to exploit larger sets of unlabeled data, which is cheap and easy to collect. Our formulation of the MLP optimization problem makes a step in that direction, by allowing every example to contribute information about part of the labels, while not affecting the cost of other labels. However, further additions can be made to take advantage of examples that have no

labels at all. That would allow collecting larger scale data with less effort (since acquiring labels is the main challenge) and capturing more diverse in-the-wild cases. The reduced load on the participants would also contribute to the authenticity of behavior.

Creative solutions for collecting labels in-the-wild will make it easier on participates. Self-reporting interfaces may include a speech-to-text feature, enabling participants to easily say "From nine to ten this morning I was in a meeting with my co-workers" or "I am starting to run now, at the beach, with my dog". A free-text option will allow people to add contexts beyond the provided menu of labels. A natural language processing component will be required to clean the text or interpret spoken sentences. Such mechanisms will add richness but also increase the sparseness of the labeling. This will be further reason to utilize a multi-task model that can combine partial pieces of data together.

The ability to work with a subset of the sensors will facilitate control systems that dynamically select which sensors to activate at any given time. Such systems will help conserve power and further promote real-time applications on mobile devices.

3.8 Conclusions

Recognition of behavioral context in-the-wild poses many challenges. In this paper, we propose the usage of multiple layer perceptron (MLP) for simultaneous recognition of many context labels. This multi-task model improves performance, compared to logistic regression, thanks to nonlinearity, dimensionality reduction, and shared hidden layers that are learned via supervised training.

The hidden representation may implicitly describe inter-label or sensor-to-label associations that "make sense" or more illusive connections that are harder to interpret. All these internal "concepts" are learned from data and not designed by a researcher — this helps avoid bias of human assumptions that may not generalize to the real world. Of course, when the hidden

layers are too wide, the MLP has too much flexibility and can learn connections that are specific to the training set. To avoid this over-fitting, a good measure is the total number of parameters in the model — an MLP with less parameters than a linear model is likely to be less prone to over-fitting and generalize better than the linear model.

We show how to use the model together with data that has unbalanced and incomplete labeling, which is very likely to happen when collecting data in-the-wild. We demonstrate how an MLP can be used to transfer a learned representation from one set of labels to a new set of labels. This can help expand a system to new behavioral aspects, even with limited amount of new data. The MLP can be resilient to missing sensors, which is a great property for practical real-world systems.

The ability to learn a good model from unbalanced, sparse data — with cases of missing labels or missing sensors — is encouraging and promotes further research efforts with naturalistic behavior: data collection does not have to be strict — it is fine if each participant contributes a small part of the labels and if each example contributes part of the sensors. This relaxation can reduce the load on participants, helping them maintain natural behavior. These advantage, and future improvements, will promote medical, research, and commercial applications that work smoothly in-the-wild.

3.9 Supplementary material

3.9.1 Missing label information

We composed several heuristic rules to declare labels as missing. These rules may cause losing cases of labels that were actually correct. However, these rules leave us with cleaner labels that we can be more confident in.

1. There are examples for which the participant did not use the label reporting interface at all.

For such examples, we mark as "missing" all the labels, except labels that we adjusted based on location ("At home", "At the beach", and "At main workplace").

2. We identify subsets of labels that represent mutually-exclusive alternatives that typically cover all the possible options for a certain aspect:

   - Body posture/movement: {"Lying down", "Sitting", "Standing", "Walking", "Running", "Bicycling"}
   - Phone position: {"Phone in pocket", "Phone in hand", "Phone in bag", "Phone on table"}
   - {"Indoors", "Outside"}

   For every example, we examine each of these label subsets. If none of the labels in the set was selected, we mark all of them as missing for this example.
   For instance: if an example is not annotated with any of the body posture/movement labels, it is most likely that actually one of this subset's labels is relevant, but the participant simply did not report it. We do not want to regard all the body posture/movement labels as negative since one of them is correct, so it is better (safer) to treat them all as missing for this example.

3. For the phone position label subset, there were cases where a participant reported two of the labels (*e.g.* "Phone in hand" and "Phone in pocket"). Most likely such cases were mistakes of label-reporting. For these cases, we mark all the phone position labels as missing, since we do not know which of the reported labels is the correct one.

4. For every participant, we identify the subset of labels that were applied. We then mark all the other labels as missing for all the participant's examples. The reason behind this is that every participant typically used a small subset of labels during the days of participation. For these labels, we can treat the participant as an authority for when they are relevant

and when they are not; but for the labels that the participant never used, it is possible the participant was not aware of them in the menu or did not bother to regard to them, so we should not rely on them to be actual negative examples.

Table 3.8 shows the counts of examples per label in the dataset, before and after applying the missing label information (MLI). For most labels, the number of positive examples remained the same, and the MLI simply narrowed down the collection of examples to be considered as negative.

Table 3.8: Label counts in the dataset, before and after regarding to missing label information (MLI). These counts are out of the 176,941 core examples (those that have all six core sensors available). P_l is the number of participants with positive examples of the label. Without MLI presents the counts of examples (positive N_l^p and negative N_l^n) before applying MLI. With MLI presents the counts of examples (positive N_l^p and negative N_l^n) that remain after removing missing labels.

			without MLI		with MLI					without MLI		with MLI	
	Label	P_l	N_l^p	N_l^n	N_l^p	N_l^n		Label	P_l	N_l^p	N_l^n	N_l^p	N_l^n
1	Lying down	47	54359	122582	54359	119880	26	Cleaning	22	1839	175102	1839	90588
2	Sitting	50	82904	94037	82904	93215	27	Laundry	12	473	176468	473	54955
3	Walking	50	11892	165049	11892	164227	28	Washing dishes	17	851	176090	851	88053
4	Running	19	675	176266	675	93692	29	Watching TV	28	9412	167529	9412	100152
5	Bicycling	22	3523	173418	3523	79920	30	Surfing the internet	28	11641	165300	11641	98028
6	Sleeping	40	42920	134021	42920	124072	31	At a party	3	404	176537	404	25876
7	Lab work	8	2898	174043	2898	24384	32	At a bar	4	520	176421	520	19986
8	In class	13	2872	174069	2872	49400	33	At the beach	5	122	176819	122	20845
9	In a meeting	34	2904	174037	2904	124578	34	Singing	6	384	176557	384	15768
10	At main workplace	26	20382	156559	20382	80114	35	Talking	44	18976	157965	18976	139394
11	Indoors	51	107944	68997	107414	7099	36	Computer work	38	23692	153249	23692	125379
12	Outside	36	7629	169312	7099	80923	37	Eating	49	10169	166772	10169	158630
13	In a car	24	3635	173306	3635	104642	38	Toilet	33	1646	175295	1646	128368
14	On a bus	24	1185	175756	1185	98751	39	Grooming	25	1847	175094	1847	109353
15	Drive (I'm the driver)	24	5034	171907	5034	93827	40	Dressing	27	1308	175633	1308	117002
16	Drive (I'm a passenger)	19	1655	175286	1655	92384	41	At the gym	6	906	176035	906	32958
17	At home	50	83977	92964	83977	91065	42	Stairs - going up	17	399	176542	399	57797
18	At a restaurant	16	1320	175621	1320	87257	43	Stairs - going down	15	390	176551	390	59749
19	Phone in pocket	31	15301	161640	14658	67960	44	Elevator	8	124	176817	124	46631
20	Exercise	36	5384	171557	5384	143467	45	Standing	51	22766	154175	22766	153353
21	Cooking	33	2257	174684	2257	127535	46	At school	39	25840	151101	25840	120042
22	Shopping	18	896	176045	896	82705	47	Phone in hand	37	8595	168346	7535	79201
23	Strolling	8	434	176507	434	25234	48	Phone in bag	22	5589	171352	5201	55473
24	Drinking (alcohol)	10	864	176077	864	41955	49	Phone on table	43	70611	106330	69929	27237
25	Bathing - shower	27	1186	175755	1186	117321	50	With co-workers	17	4139	172802	4139	62410
							51	With friends	25	12865	164076	12865	81005

Table 3.9: Logistic regression performance. Training without and with missing labels information. Performance scores reported with old and new metrics (without and with missing labels information, respectively).

	metrics without MLI				metrics with MLI			
	accuracy	**sensitivity**	**specificity**	**BA**	**accuracy**	**sensitivity**	**specificity**	**BA**
LR (trained without MLI)	0.846	0.533	0.851	0.692	0.846	0.534	0.863	0.698
LR (trained with MLI)	0.828	0.587	0.824	0.705	0.840	0.588	0.846	0.717

Table 3.9 shows the effect of regarding to missing label information (MLI) in both training and testing of the logistic recognition system. Introducing MLI to the performance metrics (counting only non-missing entries) shows very slight increase in sensitivity (probably related to cases of wrong phone-position labels that are now marked missing) and larger increase in specificity (related to the many cases that were previously treated as negative and now as missing). The effect of MLI on training is a combination of slight decrease in specificity (a small sacrifice caused by getting rid of good negative examples) and larger increase in sensitivity, contributing to an overall increase in balanced accuracy.

3.9.2 Results per-label

In order to provide a complete picture, and to allow readers to examine results for different labels, we add performance scores for each of the 51 labels in tables 3.10–3.11. These tables include results with the LR baseline, and with MLP with zero–two hidden layers. The last column refers to MLP that was trained with sensor-dropout. These tables show a general trend of improvement for many labels when progressing from the baseline to an MLP with two hidden layers. The improvement is more significant for labels that started with relatively poor performance, like “Bathing — shower”, “Cleaning”, “At the beach”, and “Elevator”.

Table 3.10: Balanced accuracy per label (part 1). LR is the baseline system with separate logistic regression trained per label. The other columns refer to MLP with either 0 hidden layers (linear), or with the hidden layer dimensions specified in parenthesis. The last column is for MLP trained with dropout ($p_{drop} = 0.2$).

		LR	**linear**	**(16)**	**(16-16)**	**(16-16)DO**
1	Lying down	0.870	0.871	0.874	0.874	0.876
2	Sitting	0.757	0.764	0.767	0.765	0.770
3	Walking	0.797	0.801	0.810	0.808	0.808
4	Running	0.658	0.753	0.814	0.814	0.819
5	Bicycling	0.867	0.851	0.872	0.877	0.868
6	Sleeping	0.891	0.892	0.895	0.896	0.897
7	Lab work	0.828	0.798	0.845	0.843	0.842
8	In class	0.767	0.793	0.770	0.766	0.795
9	In a meeting	0.797	0.810	0.814	0.814	0.781
10	At main workplace	0.822	0.835	0.842	0.852	0.847
11	Indoors	0.867	0.879	0.888	0.884	0.891
12	Outside	0.856	0.869	0.876	0.881	0.885
13	In a car	0.864	0.867	0.869	0.859	0.864
14	On a bus	0.809	0.835	0.866	0.865	0.858
15	Drive (I'm the driver)	0.858	0.871	0.865	0.866	0.857
16	Drive (I'm a passenger)	0.834	0.819	0.853	0.868	0.860
17	At home	0.752	0.769	0.778	0.792	0.794
18	At a restaurant	0.770	0.839	0.820	0.833	0.846
19	Phone in pocket	0.778	0.789	0.795	0.798	0.802
20	Exercise	0.821	0.813	0.812	0.829	0.821
21	Cooking	0.712	0.722	0.728	0.737	0.747
22	Shopping	0.723	0.774	0.783	0.773	0.792
23	Strolling	0.649	0.687	0.745	0.764	0.759
24	Drinking (alcohol)	0.681	0.779	0.786	0.793	0.803
25	Bathing - shower	0.632	0.706	0.731	0.734	0.746
	Average (labels 1–25)	0.786	0.807	0.820	0.823	0.825

Table 3.11: Balanced accuracy per label (part 2). LR is the baseline system with separate logistic regression trained per label. The other columns refer to MLP with either 0 hidden layers (linear), or with the hidden layer dimensions specified in parenthesis. The last column is for MLP trained with dropout ($p_{drop} = 0.2$).

		LR	**linear**	**(16)**	**(16-16)**	**(16-16)DO**
26	Cleaning	0.624	0.693	0.721	0.731	0.740
27	Laundry	0.648	0.758	0.682	0.662	0.674
28	Washing dishes	0.606	0.704	0.729	0.761	0.793
29	Watching TV	0.639	0.690	0.713	0.711	0.734
30	Surfing the internet	0.611	0.588	0.599	0.589	0.614
31	At a party	0.765	0.640	0.773	0.738	0.794
32	At a bar	0.783	0.671	0.791	0.845	0.863
33	At the beach	0.498	0.717	0.822	0.820	0.846
34	Singing	0.524	0.514	0.501	0.529	0.663
35	Talking	0.664	0.677	0.677	0.685	0.679
36	Computer work	0.705	0.724	0.732	0.730	0.727
37	Eating	0.657	0.666	0.672	0.677	0.669
38	Toilet	0.635	0.647	0.683	0.717	0.695
39	Grooming	0.632	0.667	0.698	0.702	0.735
40	Dressing	0.660	0.683	0.710	0.737	0.749
41	At the gym	0.651	0.683	0.712	0.800	0.779
42	Stairs - going up	0.595	0.708	0.757	0.755	0.731
43	Stairs - going down	0.609	0.707	0.751	0.753	0.728
44	Elevator	0.500	0.783	0.813	0.845	0.845
45	Standing	0.679	0.678	0.677	0.668	0.667
46	At school	0.739	0.748	0.751	0.754	0.751
47	Phone in hand	0.685	0.699	0.692	0.695	0.694
48	Phone in bag	0.753	0.752	0.746	0.764	0.744
49	Phone on table	0.789	0.804	0.797	0.802	0.801
50	With co-workers	0.657	0.720	0.752	0.755	0.778
51	With friends	0.608	0.613	0.617	0.636	0.635
	Average (labels 26–51)	0.651	0.690	0.714	0.725	0.736

3.9.3 Interpreting the multi-task MLP

Interpreting a small MLP

In order to better understand how a multi-task MLP has the potential to use multi-modal sensors to recognize a broad range of context-labels, we analyze a model with a relatively small architecture — two hidden layers of two nodes (MLP (2,2)). Using the trained model from one of the cross validation folds, we process the fold's $\sim$130k training examples. We observe the activations of hidden nodes to examine what kind of examples cause a node to "turn on" (have high activation value) and try to characterize the "meaning" of the node. We focus on the two nodes in the second hidden layer (the layer right before the output). For each node, we find the $\sim$52k (40%) low-activation-examples — those that caused the lowest activation values for that node, and the $\sim$13k (10%) high-activation-examples — those that caused the highest activation.

In figure 3.3, we examine the input sensor-features (after they were standardized over the whole training set): we look at average feature values of the low-activation-examples and high-activation-examples for selected features. Features for which there is a strong contrast between the low-activation and high-activation examples give indication about what the hidden node is sensitive to — what kind of information it encodes. From the second hidden layer, node-1 (top sub-figure) seems to be activated by situations that involve relatively constant and low-magnitude motion of the phone (Acc magnitude signal has low average and standard deviation), strong watch motion only in the y-axis (*e.g.* lateral rotation of the arm while the hand keeps facing a table), low diameter of location, WiFi availability, and time-of-day between 9am and 3pm. On the other hand, node-2 (bottom sub-figure) is associated with higher and more fluctuating phone motion, strong motion in all axes of the watch, large location diameter and no WiFi availability.

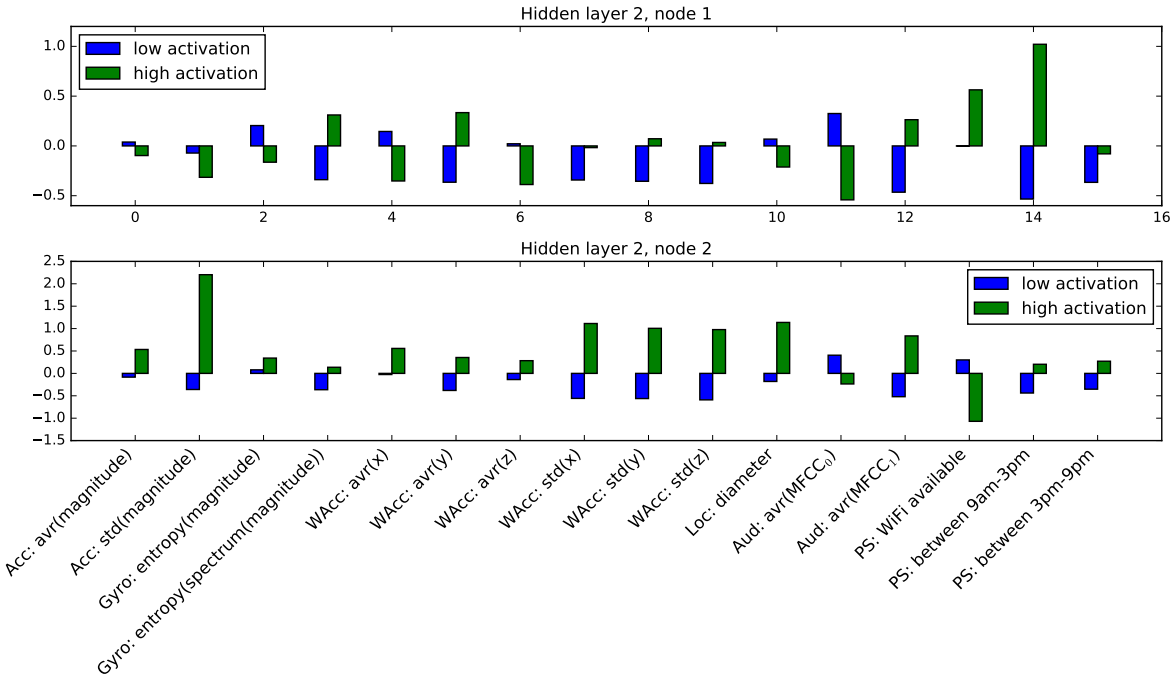

Figure 3.3: Sensor-features and hidden node activation. For each of the two hidden nodes and for a selected subset of input features, the bars describe the average standardized feature value among the examples that cause low activation in the node (blue) and among the examples that cause high activation in the node (green).

In figure 3.4, we examine the context-labels by looking at the frequency of each label among the low-activation-examples and high-activation-examples. Both node-1 and node-2 respond with low activation to many examples of lying down, sitting, sleeping, indoors, home, and phone on table. However, their activation patterns differ in some behavioral aspect: node-1 is more responsive than node-2 to sitting, in a meeting, at main workplace, computer work, at school, and phone on table. Node-2 responds more to walking, running, bicycling, outside, in a car, drive, phone in pocket, and exercise.

From these observations, we can describe the multi-layer hidden representation in a more human-interpretable manner: simplistically, node-1 detects "sedentary office-style context", and node-2 detects "moving around outside". The supervised training needs to distribute the limited resources (the hidden nodes) in a way that is most predictive for the many context-labels. MLPs with wider hidden layers (*e.g.* with four, eight, or sixteen nodes) can refine the representation and each hidden node can represent a more specific situations. This can help cover more possible contexts. If the MLP has a too wide hidden representation (*e.g.* 64 nodes), the training can result in nodes that capture too-specific cases that only occur in the training set (over-fitting), causing the MLP to make mistakes on unseen data.

Interpreting a larger MLP

Analysis of activation of internal nodes only provides intuition to the *potential* of MLPs to concisely represent information from multi-modal sensors that will be informative for recognition of a broad range of contexts. In the main text of the paper, we also provide analysis with one of the most successful models from our experiments (MLP (16,16) trained with sensor-dropout), to better understand how the successful model relies on different sensors to recognize different contexts from different behavioral aspects (see "Interpreting the MLP" in main text). The following tables (tables 3.12–3.13) present the results of this analysis broken down by specific context-labels.

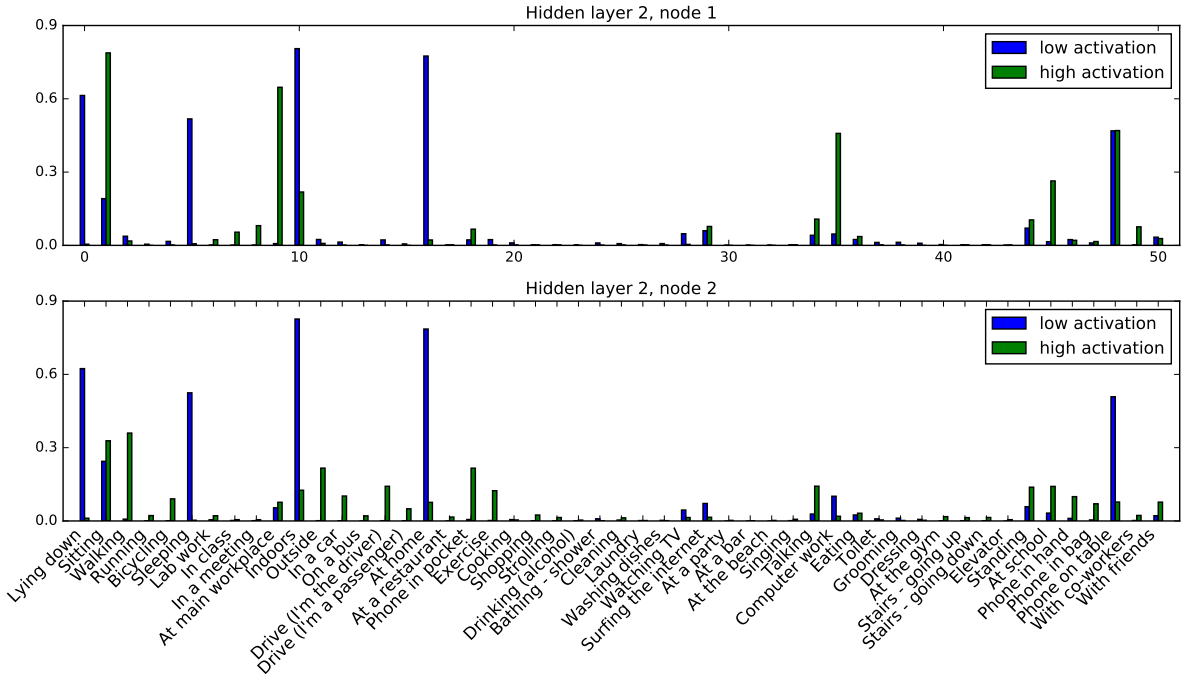

Figure 3.4: Context-labels and hidden node activation. For each of the two hidden nodes the bars describe the frequency of each context-label among the examples that cause low activation in the node (blue) and among the examples that cause high activation in the node (green).

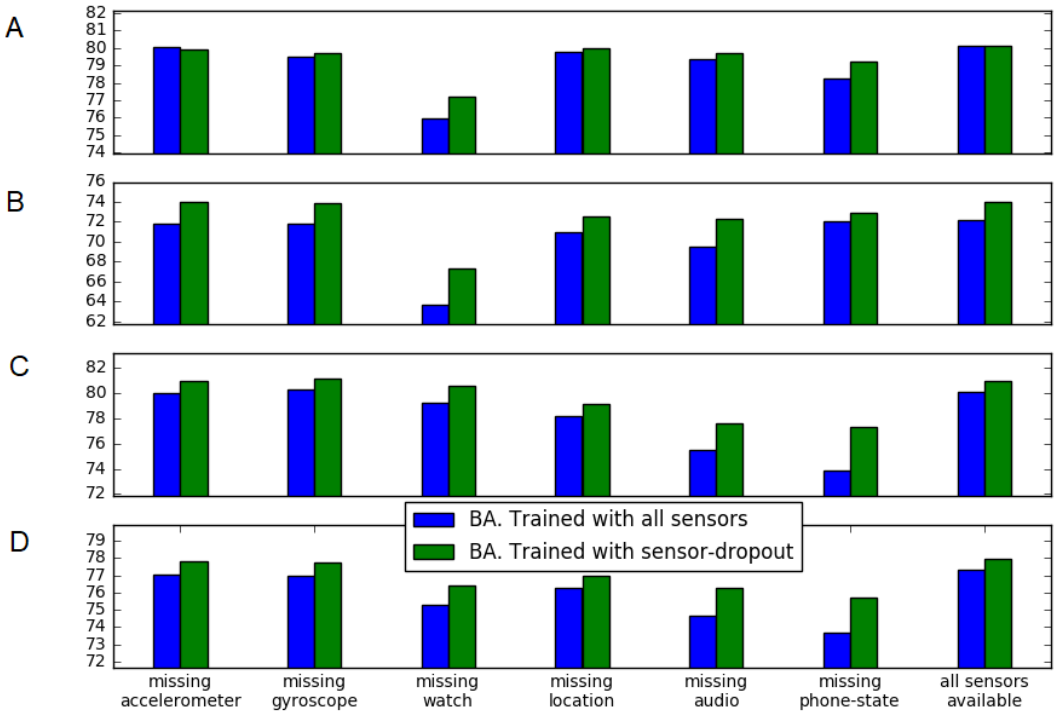

Figure 3.5: Sensing modalities and behavioral aspects. Balanced accuracy (in %) scores of MLP (16,16) either with (green) or without (blue) sensor-dropout training. The rightmost bars show scores when testing with all sensors available and the other bars show scores when testing with a single sensing modality missing. Subplot D) shows BA averaged over all 51 labels, showing overall strong reliance on watch-acceleration, audio, and phone-state (as well as significant recovery with sensor-dropout training). The other subplots show BA averaged over subsets of context-labels, referring to the three behavioral aspects mentioned in the paper. A) Body-state (Lying down, Sitting, Standing, Walking, Running, Bicycling) recognition is relying mainly on watch-acceleration, and sensor-dropout training reduces the sensitivity to missing watch. B) Home-activities (Cooking, Cleaning, Laundry, Washing dishes, Grooming, Dressing) recognition is also heavily reliant on watch-accelerometer, where sensor-dropout is also significantly improving. C) Environments (Class, Meeting, Workplace, Home, Restaurant, Bar, Party, Beach, Gym, School) recognition relies strongly on phone-state and audio. Here, as well, sensor-dropout training reduced this reliance. Finally, even when all sensors are available, the sensor-dropout trained MLP is more robust and performs better.

3.10 Acknowledgements

Chapter 3, in full, is a reprint of the material as it appears in Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), 1(4), December 2017, Y. Vaizman, N. Weibel, and G. Lanckriet. The dissertation author was the primary investigator and co-author of this paper.

Table 3.12: Relations between sensors and context-labels 1–25. Recognition scores (balanced accuracy) with the full multi-task MLP (with 2 hidden layers of 16 nodes) that was trained with sensor-dropout. Scores for specific context-labels (rows), tested with all the 6 sensors available, and with masking each of the 6 sensing modalities. These results help interpret the learned MLP, and how it uses the different sensors to recognize different contexts. For each context-label (row), the most relied-upon sensor (missing this sensor creates the strongest degradation in recognition) is highlighted in gray.

		5 sensors (all except one)					
Label	all sensors	Acc	Gyro	WAcc	Loc	Aud	PS
Lying down	0.876	0.877	0.877	0.860	0.872	0.867	0.847
Sitting	0.770	0.767	0.763	0.749	0.766	0.763	0.749
Walking	0.808	0.802	0.791	0.800	0.805	0.804	0.801
Running	0.819	0.817	0.814	0.729	0.823	0.808	0.835
Bicycling	0.868	0.867	0.869	0.864	0.870	0.864	0.852
Sleeping	0.897	0.896	0.897	0.886	0.894	0.892	0.865
Lab work	0.842	0.841	0.841	0.845	0.768	0.836	0.800
In class	0.795	0.795	0.798	0.815	0.753	0.743	0.796
In a meeting	0.781	0.779	0.778	0.782	0.760	0.731	0.760
At main workplace	0.847	0.849	0.849	0.841	0.820	0.839	0.788
Indoors	0.891	0.888	0.885	0.889	0.890	0.865	0.871
Outside	0.885	0.882	0.880	0.883	0.884	0.857	0.868
In a car	0.864	0.861	0.864	0.860	0.852	0.851	0.842
On a bus	0.858	0.857	0.861	0.863	0.858	0.845	0.813
Drive (I'm the driver)	0.857	0.853	0.857	0.853	0.840	0.862	0.824
Drive (I'm a passenger)	0.860	0.863	0.862	0.865	0.859	0.855	0.836
At home	0.794	0.795	0.796	0.790	0.775	0.754	0.767
At a restaurant	0.846	0.840	0.848	0.846	0.843	0.807	0.795
Phone in pocket	0.802	0.792	0.781	0.805	0.798	0.791	0.778
Exercise	0.821	0.820	0.824	0.807	0.816	0.811	0.795
Cooking	0.747	0.750	0.752	0.717	0.734	0.730	0.722
Shopping	0.792	0.798	0.811	0.772	0.797	0.796	0.741
Strolling	0.759	0.753	0.751	0.747	0.765	0.770	0.707
Drinking (alcohol)	0.803	0.799	0.799	0.800	0.789	0.776	0.764
Bathing - shower	0.746	0.739	0.748	0.663	0.741	0.684	0.763

Table 3.13: Relations between sensors and contexts-labels 26–51. Recognition scores (balanced accuracy) with the full multi-task MLP (with 2 hidden layers of 16 nodes) that was trained with sensor-dropout. Scores for specific context-labels (rows), tested with all the 6 sensors available, and with masking each of the 6 sensing modalities. These results help interpret the learned MLP, and how it uses the different sensors to recognize different contexts. For each context-label (row), the most relied-upon sensor (missing this sensor creates the strongest degradation in recognition) is highlighted in gray.

		5 sensors (all except one)					
Label	all sensors	Acc	Gyro	WAcc	Loc	Aud	PS
Cleaning	0.740	0.740	0.741	0.662	0.737	0.740	0.744
Laundry	0.674	0.674	0.662	0.572	0.674	0.697	0.668
Washing dishes	0.793	0.792	0.791	0.707	0.762	0.772	0.771
Watching TV	0.734	0.731	0.728	0.735	0.725	0.688	0.712
Surfing the internet	0.614	0.609	0.613	0.595	0.605	0.622	0.605
At a party	0.794	0.779	0.769	0.777	0.781	0.718	0.846
At a bar	0.863	0.859	0.863	0.861	0.861	0.866	0.758
At the beach	0.846	0.860	0.865	0.849	0.845	0.833	0.742
Singing	0.663	0.654	0.665	0.653	0.669	0.646	0.656
Talking	0.679	0.678	0.677	0.676	0.679	0.663	0.670
Computer work	0.727	0.726	0.726	0.719	0.712	0.724	0.704
Eating	0.669	0.667	0.671	0.663	0.668	0.668	0.650
Toilet	0.695	0.692	0.694	0.662	0.685	0.664	0.691
Grooming	0.735	0.736	0.741	0.703	0.714	0.682	0.718
Dressing	0.749	0.743	0.746	0.680	0.732	0.713	0.753
At the gym	0.779	0.795	0.798	0.735	0.768	0.737	0.745
Stairs - going up	0.731	0.744	0.701	0.736	0.732	0.742	0.743
Stairs - going down	0.728	0.717	0.696	0.715	0.717	0.738	0.734
Elevator	0.845	0.843	0.838	0.844	0.842	0.828	0.795
Standing	0.667	0.666	0.666	0.630	0.663	0.676	0.668
At school	0.751	0.752	0.755	0.758	0.711	0.732	0.735
Phone in hand	0.694	0.684	0.673	0.708	0.677	0.688	0.671
Phone in bag	0.744	0.753	0.767	0.755	0.748	0.711	0.713
Phone on table	0.801	0.795	0.784	0.809	0.799	0.787	0.769
With co-workers	0.778	0.773	0.774	0.777	0.742	0.774	0.741
With friends	0.635	0.632	0.635	0.641	0.638	0.580	0.636

Chapter 4

ExtraSensory App: Data Collection In-the-Wild with Rich User Interface to Self-Report Behavior

4.1 Abstract

We introduce a mobile app for collecting in-the-wild data, including sensor measurements and self-reported labels describing people's behavioral context (*e.g.* driving, eating, in class, shower). Labeled data is necessary for developing context-recognition systems that serve health monitoring, aging care, and more. Acquiring labels without observers is challenging and previous solutions compromised ecological validity, range of behaviors, or amount of data. Our user interface combines past and near-future self-reporting of combinations of relevant context-labels. We deployed the app on the personal smartphones of 60 users and analyzed quantitative data collected in-the-wild and qualitative user-experience reports. The interface's flexibility was important to gain frequent, detailed labels, support diverse behavioral situations, and engage different users: most preferred reporting their past behavior through a daily journal, but some preferred reporting what they're about to do. We integrated insights from this work back into the app, which we make available to researchers for conducting in-the-wild studies.

4.2 Introduction

The ability to automatically recognize people's behavioral context (the activities they're doing, where they are, their body posture, *etc.*) is desirable for many domains, such as health management [28, 36], aging care [39, 45] and office assistant systems [100]. Machine learning

methods to train and test context-recognition systems require data, including sensor measurements and labels describing the actual context of real people. Many activity-recognition studies validated their systems with data collected in a lab [69, 14, 26] . However, in order to develop ecologically valid systems that work well in the real world, the data used for development should be collected in-the-wild — capturing people's authentic behavior in their regular environments.

Data collection in-the-wild raises technical difficulties related to interruptions in sensor recording and diversity in phone-devices [85] and device placement [40]. The harder challenge, however, is acquiring labels when there is no researcher-observer present with study participants. Previously suggested solutions involved unnatural equipment [68, 20, 89, 12] or simple self-reporting interfaces [25, 37] and resulted in data that had limited ecological validity and labels that describe behavior in a single-dimensional manner and cover a small portion of everyday life.

Recently, we have collected the *ExtraSensory Dataset* from 60 participants using everyday devices [91]. To maintain ecological validity, participants used their *own personal phones*, without restricting phone placement, contributed data from their natural environments (home, work, commute, *etc.*), while they engaged in their natural (unscripted, unobserved, and without a prescribed list of tasks to perform) behavior, and described their own behavior in an authentic, subjective manner. We applied simple machine learning methods to the data and demonstrated successful recognition of a wide variety of everyday contexts, like sleeping, shower, on a bus, *etc.*

In this paper, we present the tool we used to collect the data — the *ExtraSensory App*, a mobile app designed to collect sensor data and engage participants to contribute detailed and frequent labels describing their behavioral context. To evaluate how the user interface enabled and affected data collection, we analyze the quantitative data from the 60 participants, as well as the qualitative feedback that they gave about their experience using the app.

The contribution of this paper is fourfold:

- Design. Our rich user interface enables self-reporting both in-situ (active-feedback and notifications) and recall-based (daily history) and has additional features to facilitate

detailed-labeling with little interaction.

- Validation. The app enabled collecting data *in-the-wild*. The resulting *ExtraSensory Dataset* is larger than previous datasets in scale (over 300,000 labeled minutes), range of behaviors (more than 50 diverse context-labels), and detail (combinations of more than three relevant labels per minute). This data was successfully used to train and test context-recognition systems [91, 93].
- Insights. Our combined analysis of the quantitative data collected in-the-wild and qualitative user-experience reports from our participants helps understand the effectiveness of the various design features. Among our findings: the rich history page facilitated reporting about long behavioral time with detail, using a watch for single-click confirmation of notifications was very helpful, and active-feedback engaged people who preferred reporting about their immediate future rather than recalling their past behavior.
- Open source code. With this paper, we also make the complete source code of the *ExtraSensory App* freely available (http://extrasensory.ucsd.edu/ExtraSensoryApp). The published app includes improvements based on the analysis in this paper and can be used either to collect labeled data or as a black-box tool for real-time behavioral context recognition.

4.3 Related work

Previous data collection studies in-the-wild exploited a variety of different approaches to acquire context labels.

4.3.1 Camera-based Approaches

In several studies, participants wore a camera that took snapshots of the scene, enabling context labels to be assigned to different times throughout the day based on the captured images.

In some studies research assistants annotated the images [68, 20], compromising the privacy of the participants and their surrounding. In other studies, the participants annotated their own images, which resulted in limiting the range of targeted behaviors (like eating detection [89]) or the number of participants (*e.g.* single person in [12]). Relying solely on camera can miss situations where context is not visible (*e.g.* phone in pocket, singing) or private situations like shower.

4.3.2 Self-Reporting In-Situ

In in-situ self-reporting, participants report their own context (*e.g.* location, activity, emotion, *etc.*) in real-time. For instance, the Experience Sampling Method (ESM) is a technique where the participant is prompted at different times to fill a short form and report their context [78]. This method samples time to estimate statistics of well-being, time-usage, or relations between activities and feelings [16]. The CrowdSignals project [98] aims to collect phone-sensor data from large crowds of users, with additional sparse probing for labels by using quick multiple-choice questions whenever the user unlocks their phone, combined with more in-depth (and less frequent) ESM questionnaires. In [101], whenever the user selected a music playlist, she was prompted to report one out of 13 activities and one out of 10 moods.

In context-recognition studies that target a specific list of activities, researchers often used in-situ self-reporting, but instead of sampling reports in arbitrary times, they let participants actively initiate reporting at relevant times. Studies that tracked a single activity (*e.g.* eating detection [18]) used a simple interface with a single button for the user to mark the start and stop times of eating. Works that targeted multiple activities (like watching TV, driving, *etc.*) added to the interface a selection of a single activity from a list [25, 37]. Commercial systems like Toggl[1] offer similar timer-based reporting.

[1] https://toggl.com

4.3.3 Self-Reporting by Recall

An alternative to in-situ self-reporting is reporting after-the-fact, by recalling. When the required time resolution is daily, silent notifications may be helpful to remind people to answer simple questions (*e.g.* "how much did you eat today?") every day [7], but for more detail, it can be hard for people to remember their daily events. The Day Reconstruction Method (DRM [34]) is a survey-based method that requires the participant to arrange the previous day in a short diary, as a sequence of episodes. By thinking of each episode as a holistic scene with different contextual aspects (location, activity, interaction with others, emotion), the person can better recall specific variables of interest, like tiredness or joy. In sensor-based context recognition studies, accurate timing of the context is important in order to align the labels with the sensor-data. DRM was used in [90] for eating detection, but the participants struggled remembering when they were eating. The researchers then listened to audio recordings and they reported that annotating eating periods based on audio was difficult.

4.3.4 Mixed Self-Reporting Approaches

Mark *et al.* [54] explored multitasking at work. They assessed productivity with end-of-day surveys, sleep using an actigraph, and monitored computer activity with a custom software.

Rahman *et al.* [72] dealt with self-assessment of stress-level and discussed the trade-off between in-situ reporting (more ecologically valid but disruptive and may cause stress) and recall-based reporting (non disruptive but introduces memory bias). They proposed a compromise solution, where participants could report on their own time, but with the aid of contextual cues like location and ambient sound level, to help them remember how they felt at specific times of the day.

Mehrotra *et al.* [58] explored people's receptivity to phone notifications. Their study combined ESM with cue-assisted recall. Four times a day, a questionnaire presented a selected

notification that the phone received in the past four hours, and asked the person what they were doing at the time, how disruptive the notification was, *etc.* They showed higher likelihood to dismiss a notification during complex ongoing tasks, and apparent connection between personality traits and responsiveness to notifications.

Consolvo *et al.* [15] designed and validated UbiFit Garden, an application to promote physical exercise, with both sensor-based automated activity recognition and user manual labeling. The activity recognition component, which was trained on controlled, scripted, and observed data [14], ran in the background and recognized activities like walking and cycling. The user could view the recognized events in a daily journal and delete, add, or change today's and yesterday's events. In addition, the visual appearance of the phone's wallpaper (graphics of flowers and butterflies) was adjusted according to the user's exercise events and was designed to incentivize the user to engage in physical activity or to correct the recognized events.

4.4 The ExtraSensory Mobile App

The solution we present in this paper — ExtraSensory — is a mobile app that automatically collects data from a range of sensors built into popular smart phones and a dedicated smart watch. In addition, it provides a rich labeling interface.

Our labeling approach uniquely combines the advantages of multiple existing solutions for self-reporting. Similar to [25, 37], our users can actively report that they are starting an activity. Similarly to the DRM, the users can look at the previous day (or today) as a journal of events and as in UbiFit Garden, this journal is filled by both automated recognition and manual editing [15]. As in the ESM studies, our app also triggers pre-scheduled prompts to ask the user to report labels [78]. Much like survey-based studies with ESM or DRM [78, 34], we address the multi-aspect nature of behavioral context and allow users to report combinations of activities, as well as location, company, body posture and more.

All the sensor-based studies mentioned above provided a study-phone to their participants and constrained the position of the phone. Contrary to that approach, to support ecological validity, we evaluated ExtraSensory with participants that used their own personal phone, in any way convenient to them. In order to broaden the options for participants, we implemented our app for both iPhone and Android. Additionally, we added support for the optional pairing of a Pebble-watch,[2] which can interact with both phone devices, and adds more sensing and user-interaction capabilities to the data collection solution.

4.4.1 Recording Sensors

When ExtraSensory is running (in either the foreground or background of the smartphone), it records a 20-second window of sensor measurements every minute and sends the measurements to a dedicated server. The measurements include 40Hz 3-axial motion sensors (accelerometer, gyroscope, and magnetometer), location coordinates, audio (the app processes the raw audio on the phone to produce 13 Mel Frequency Cepstral Coefficients [49]), and phone-state indicators (app-state, WiFi availability, time-of-day, *etc.*). During the 20-second window, the app also collects measurements from the optional watch, if it is available and used by the participant (25Hz 3-axial accelerometer and compass heading updates).

Communication with the ExtraSensory server is encrypted and users have the option to allow cellular communication or, as all our participants chose, communicate via WiFi only. In case no network is available, measurements are stored until they can be transmitted. The server has a basic activity-classifier that was trained on preliminary data from two iPhone users. When the server receives the sensor data, it responds with a guessed activity (the body posture/movement state), which in turn helps the user report their own subjective labels.

The app has a "data-collection" switch, which is on by default whenever the app is launched. The user can decide, for any reason (low battery, privacy, *etc.*), to temporarily turn

[2] https://www.pebble.com

data-collection off, in which case new recordings are suspended, but the label-reporting interface is still available.

4.4.2 Reporting Context Labels

In ExtraSensory, the description of behavior is based on two label components: *main activity* and *secondary activities*.

"Main activity" refers to the body posture/movement state — a single value out of the mutually-exclusive states: *lying down*, *sitting*, *standing in place*, *standing and moving*, *walking*, *running*, and *bicycling*. We included the label "standing and moving" with the intention of describing intermediate situations — not exactly standing in the same position and not exactly walking towards a destination, but something in between (*e.g.* when cooking or cleaning at home).

"Secondary activities" refer to any additional fine-grained attributes that apply to a situation, in a multi-label formulation (multiple labels can apply simultaneously). This includes specific sport activities, work or home activities, transportation modes, as well as other non-activity descriptors for location, phone position, and more. We also defined a multi-label "moods" component but we did not focus on collecting mood labels. The app lets the user decide which labels best describe their own behavior and the goal is to later train classifiers that are able to predict those subjective labels.

The flexible user interface provides a variety of mechanisms to help make label-reporting quick and easy, and has two modes of reporting: *past* and *near-future*.

History page (past). The main route for past reporting is through ExtraSensory's history page (Fig. 4.1 (A)) — it allows users to engage in some behavior (*e.g.* sleep, drive) and then report about it later. This page displays a daily calendar, where each row represents an "event" — a continuous time segment where the context stayed the same. The server guesses of body state appear with a question mark, to signal to the user that their own labels for this time-segment were not yet provided (*e.g.* "07:52 Walking?"). In case the server guessed the same body state

for several consecutive minutes, these minutes appear in the history as merged to a single event (*e.g.* "08:08 – 08:11 Sitting?") and the user can report the same labels to all these minutes simultaneously. By clicking on an event, the app opens the labeling form, where the user can edit the context-labels (Fig. 4.1 (B)). If the event was already labeled by the user, the existing labels are loaded and can be edited; otherwise, the server-guess is loaded to the "main activity" field and the other fields start blank. After selecting the context-labels in the labeling form and pressing "send feedback", the labels are sent to the server (or queued, waiting for network connection) and the history now displays the time-segment without a question mark, and with the added secondary labels in parenthesis (*e.g.* "07:54 – 08:00 Sitting (At home, Eating)").

The colors of the history rows correspond to the main activity (body-state), ranging from a cold blue for "lying down" to a warm red for "bicycling". The color-code was designed to roughly illustrate the intensity level of movement and help the user visually see when their activity might have changed. With finger-swipe gestures, the user can split a time-segment to separate minutes or merge consecutive rows to a longer event with constant context. Additionally, users can view previous days (by clicking the "previous day" button, at the top left), but they can only edit labels for today and yesterday, to avoid memory bias of looking back too-long ago.

Active feedback (near-future). For cases where users already know what behavior they are going to engage in, ExtraSensory enables pre-labeling immediate-future context. The main route for near-future reporting is the "active feedback" feature: at any time, users can press the green plus-symbol (bottom center, see Fig. 4.1 (A)); this opens the labeling form in the near-future mode (Fig. 4.1 (C)), where users can report their current or upcoming context. For example, a user can report that she is going to be driving a car, with family, and that this context is going to stay relevant for the next 25 minutes. After pressing "send feedback", at every new recorded minute, the same labels will automatically be sent to the server, and the user can attend to the actual activity (*e.g.* driving, without distractions). We limit the foresight time (the "valid for" field) to a maximum of 30 minutes in the future.

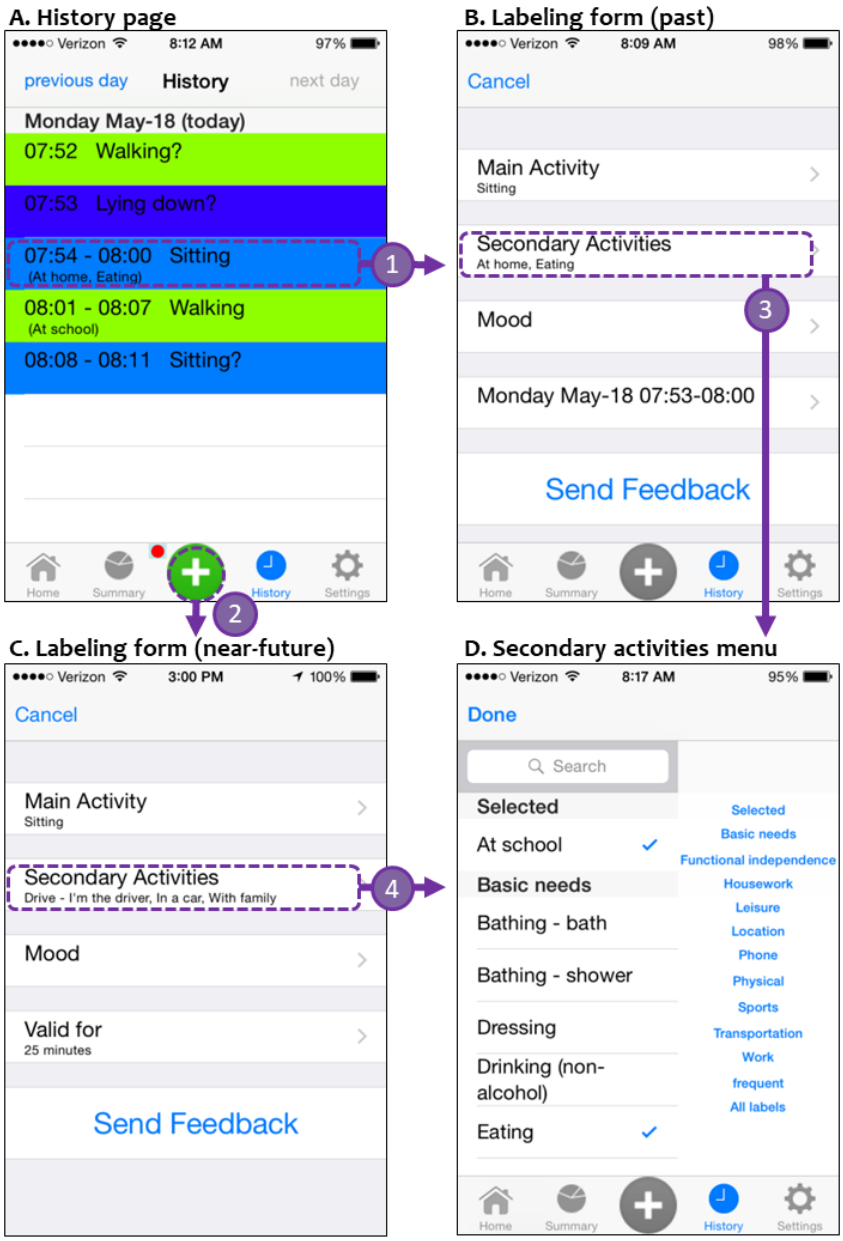

Figure 4.1: Label-reporting user interface, with flow marked in purple shapes and arrows. In the history page (A), each row represents a segment of time with constant context-labels, either with question mark (server-guess) or without (user-reported). 1) By clicking a row, the app opens the Labeling form in the past-mode (B), where the user can edit the context labels for a specific time-segment in the past. 2) By pressing the active-feedback button (green with plus symbol), the app opens the labeling form in the near-future mode (C), where the user can initiate a report of what they are about to do. 3–4) From the labeling form, pressing the "secondary activities" field opens a rich menu (D), where the user can select multiple labels, jump to a relevant topic, and see personalized frequently-used labels.

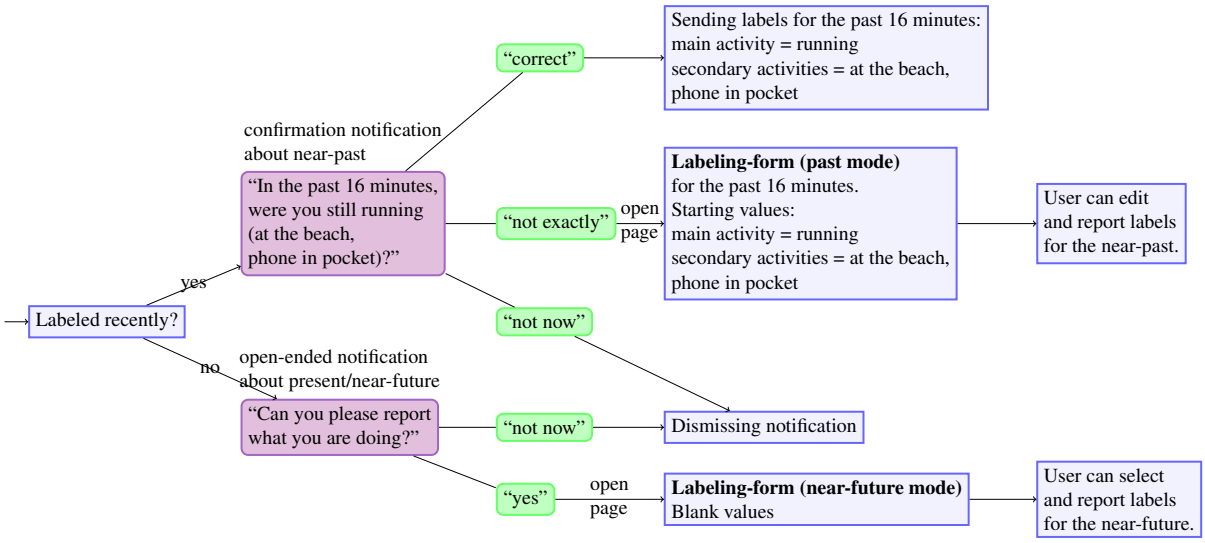

Figure 4.2: Notification flow with possible example scenarios. The flow starts in periodic intervals and first checks if there are any reported labels for any minute in the past 20 minutes. The purple rounded boxes present the notification messages displayed to the user. The green rounded boxes present the optional buttons for the user. For a confirmation-notification, the user-answer "correct" is a way to send labels for up to 20 minutes with a single click. Two possible routes lead to opening the labeling-form in two different modes: the "not exactly" user-answer enables adjusting the labels for the near-past and the "yes" user-answer enables reporting near-future context (like when pressing the active-feedback plus-symbol button).

Selecting the labels – From the labeling form (Fig. 4.1 (B)–(C)), clicking the "main activity" field opens a simple menu to select a single body-state out of the seven options (in the past-mode, there is an additional "I don't remember" option — in case the user just wants to report secondary activities). Clicking the "secondary activities" field opens a richer menu that allows selecting multiple labels from a list of over 100 labels (Fig. 4.1 (D)). To make it easier for the user to find the relevant labels quickly, the menu is organized by topics (with quick-link index in the side), like "basic needs" or "transportation." A "frequently-used" section (indexed by the link "frequent") displays the labels that the individual user previously applied, in order of usage frequency, making it quicker to find personalized relevant labels after a day or two of participation.

Notifications (past or near-future) – In addition to the participants' initiated reports, the app also triggers notifications at constant intervals (the default is every 10 minutes, but the

user can increase this up to 45 minutes). These notifications remind users to report labels and they provide a direct connection to the labeling form, in either the past or near-future modes, depending on whether the user reported any labels for the recent 20 minutes (see flow diagram in Fig. 4.2). After reporting near-future context, the next notification is re-scheduled to appear after the reported near-future period is over.

Watch Notifications and quick responses – In addition to the increased sensor recording, the optional smart watch also contributes to the interaction with the user. When a notification is triggered on the phone, it also appears on the the watch. In case the system asks whether the recent context is still the same, and if the answer is "correct", the user can actually respond on the watch by pressing the right top button on the side of the watch (see Fig. 4.3). In case of an open-ended notification (when there is no user-provided recent context), the notification on the watch serves merely as a reminder (Fig. 4.4). The visual indication is complemented with a vibration when every new notification appears on the watch (users can disable vibrations, *e.g.* when going to sleep).

4.4.3 Additional Visual Features

Besides the label-reporting mechanisms, ExtraSensory provides additional supporting features. During every 20-second recording window a red dot appears on the control bar of the app (see Fig. 4.1 (A)) and a "REC" text appears on the watch (see Fig. 4.4). Additionally, the app has a home page that acts as a dash-board to keep users informed and to help debug possible problems. The page specifies how many minutes currently have data awaiting to be sent to the server and has an icon indicating whether or not the watch is currently paired with the phone. In the iPhone version, there is an additional cartoon image that symbolizes the latest guessed main activity. This feature was originally designed to attract the user's attention and encourage them to provide their own labels. However, in preliminary experimentation, it became clear that it was more useful to keep the app on the history page rather than the home page, so we did not include

Figure 4.3: Watch — confirmation notification. The same notification from the phone scrolls on the top half of the watch app. If the user's context remained the same, they can reply "correct" by pressing the top-right button.

Figure 4.4: Watch — open-ended notification. This is only useful as a reminder; to initiate reporting labels the user has to go to the phone. During a 20-second recording window, the text "REC" is shown in the bottom half.

the cartoon in the Android version.

Similarly, the app has an additional "summary" page, which displays minute counts of each of the main activity labels, in a pie chart (iPhone) or bar plot (Android), with the same color-code as in the history page. Similar to UbiFit-Garden [15], the user can take a quick glance at this visual summary and possibly decide to report more labels, to update this visualization.

4.5 User Deployment, Analysis and Results

To evaluate ExtraSensory as a solution for data collection in-the-wild, and to collect data to develop context-recognition systems, we conducted an in-the-wild study. We recruited 60 participants (34 female, 26 male). They were mostly students and research assistants at our local university, averaged 25 years in age, and had diverse ethnic backgrounds. With each participant (user), we conducted two meetings, approximately seven days apart.

In the first meeting, we installed the app on the user's personal phone (34 were iPhone users, 26 were Android users) and provided them with a Pebble smart watch (56 users agreed to wear the watch). The user read and signed the consent form. We explained how to use the app and requested that the users keep the app running (with data-collection on) as much as convenient. We also requested that they use the different label-reporting mechanisms to provide as many labels as convenient (and as much as they can remember) without interfering too much with their natural behavior. We did not specify any targeted activities, but rather asked that they engage in their routine, and report any labels that they believe appropriately describe their context. We explained that the collected data will be de-identified and published and will serve for training systems that can measure people's activities using sensors (but we did not specify any particular application).

In the second meeting, we uninstalled the app from the user's phone, collected the watch back, and asked the user to fill out a short survey about the experience. We also compensated

users for their participation with a basic amount of US$40, plus an incentive amount of $0–35, depending on the amount of labeled data that the user contributed. Although we did not explicitly examine the compensation's impact, we can report that 39 users contributed more than enough data to reach the maximum total of $75 and the other 21 averaged $60.

We present results first from the quantitative data that was collected and then from the qualitative user-experience surveys. For both aspects, we examine how the user interface of ExtraSensory influenced the study.

4.5.1 Quantitative Analysis

During the six months of the study, we collected over 300,000 minutes from the 60 users, labeled with combinations of over 50 diverse context-labels. On average, each minute was assigned more than three labels. These detailed contexts describe over 14,000 distinct "events" (segments of constant context), with median duration of nine minutes. Although not a direct contribution of this paper, we made this dataset, titled the *ExtraSensory Dataset*, publicly available at http://extrasensory.ucsd.edu. In this section we analyze these data to gain insight about the usage of our app.

Turning on data-collection. The users had control and could decide when to turn off data-collection (*e.g.* when battery is too low or to maintain privacy). Figure 4.5 shows two users who participated for approximately seven days and had different patterns of data collection. Some users (like the one presented on the left) kept the app running and data-collection on almost continuously throughout their participation days. Other users (like the one on the right) collected data in many separate segments with gaps.

Figure 4.5 also shows that during the times that data-collection was on, not all sensors were available all the time. Most notably, the users were free to remove the watch (and turn off the watch app) so there are times when data collection was on but there are no measurements from the watch accelerometer. Similarly, users sometimes turned off location services on their

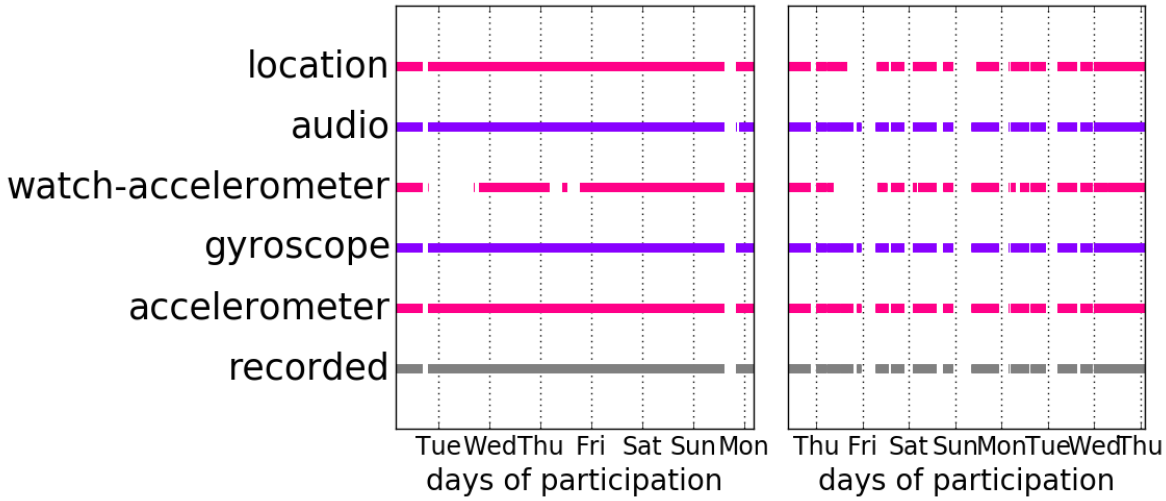

Figure 4.5: Sensor recordings for two users. The "recorded" row describes when data-collection was on and the other rows refer to recording of specific sensors. The bars indicate when, during the days of participation, data was collected. The vertical dotted grid lines indicate the time 6AM in the participation days. Watch and location were sometimes unavailable.

phone or location had a weak signal.

Keeping the app running caused faster draining of the battery. According to users' estimation, on average, they charged the phone 2.3 times a day and the watch once every 1.75 days.

Reporting labels. Figure 4.6 shows the distribution of label reporting mechanisms over the total recorded minutes in the dataset. For 82% of the recorded minutes, the users provided context-labels. History was the reporting mechanism that yielded most of the label coverage (85% of the labeled minutes). Out of the minutes that were labeled via notification, the vast majority were cases of confirmation-notification (asking whether the near-past context remained the same), and in most of those cases, the user replied "correct" using the watch.

The different reporting mechanisms helped support a variety of situations. Figure 4.7 shows the relative minute-coverage achieved by the three reporting mechanisms for reporting specific labels. Understandably, the history page was almost the only feasible way to report sleeping. Similarly, laughing was mainly report retroactively (with the history page), which is

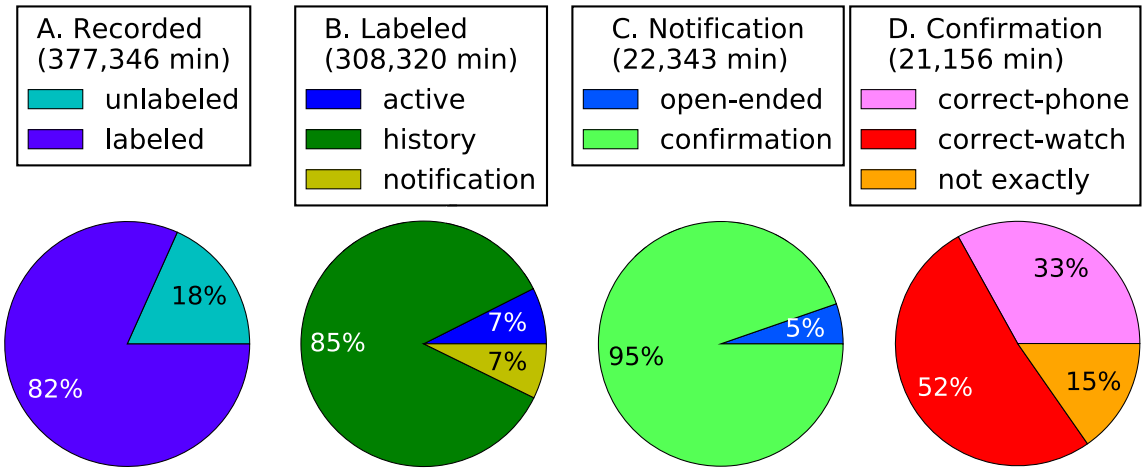

Figure 4.6: Distribution of label-reporting over the minutes in the dataset. Above each pie chart is the total number of minutes in the whole pie. A) Most of the recorded minutes were labeled. B) The vast majority of labeled minutes were labeled via the history page but active-feedback and notifications still contributed significantly. C) Almost all of the minutes that were labeled via notification were from confirmation-notifications. D) For more than half of the minutes that were labeled via confirmation-notification, the user replied "correct" through the watch. C–D) Relatively few minutes (4,500) were labeled as the result of editing the labeling form triggered by notification (either open-ended notification or when replying "not exactly" to a confirmation-notification).

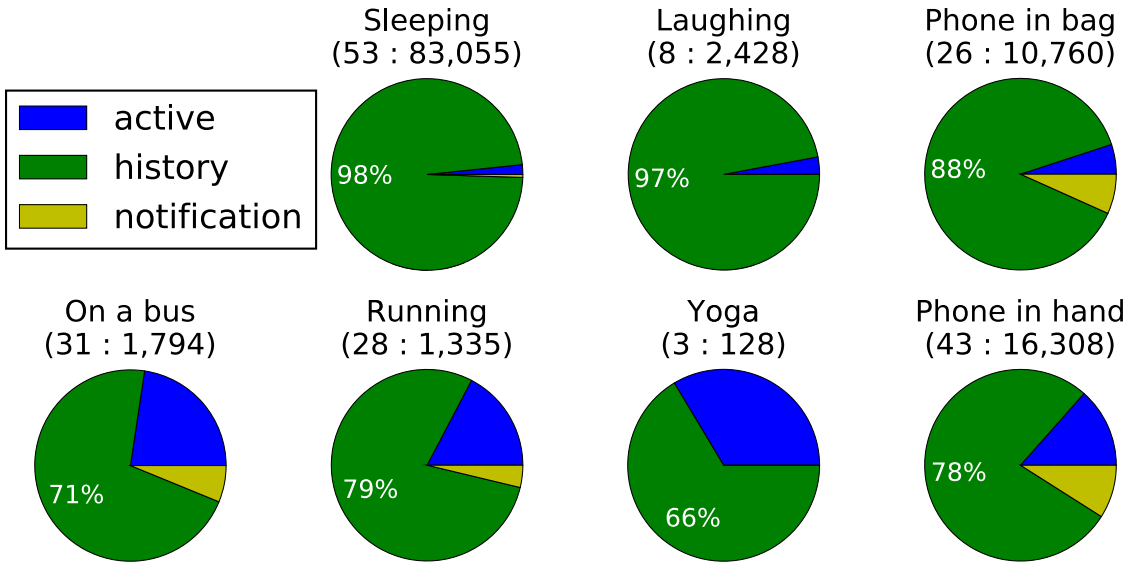

Figure 4.7: Distribution of label-reporting mechanisms for selected labels. For each label, the title above the pie indicates how many users reported this label followed by how many minutes were annotated with the label (the whole pie). The percentage of minutes for which the label was reported via history is also indicated numerically inside the pie. The top-row contexts were mostly reported via history. For the bottom-row contexts, active-feedback was utilized more.

fitting for a spontaneous action that is typically not predictable in advance. Contexts like "on a bus", "running", and "Yoga" were more planned, so users utilized active feedback more to report starting these contexts. During Yoga, users were not available to reply to notifications, so they had to either use active feedback before starting or history after they were done. When the phone was held in hand, users were more easily available to report in-situ contexts (using active feedback and notifications), but when the phone was in a bag, they had to rely more on the history and report about it after the fact.

Figure 4.8 shows the label reporting patterns across different days of the week and hours of the day. Overall, users turned on data-collection more during the work week and less in the weekends. Similarly, more data was recorded in the afternoon and evening hours. These peak hours also show increased usage of notifications and active-feedback. This makes sense, given that people are not available to interact with the app while they sleep, but they can report about it

later through the history. History covered the vast majority of labeled minutes in all days of the week and all hours of the day.

Users had different approaches to reporting labels. Figure 4.9 shows labeling patterns (including the label-reporting mechanism and the reported label combinations) from two example users, over their entire participation periods. The first user (top subplot) tended to use the history page much more than notifications or active-feedback. The reported labels seem to form long time-segments of continuous context (especially at nights, when the context involved "lying down" and "sleeping"). The reported daytimes were comprised of long, continuous contexts, like "sitting" and "at school", with additional details that changed more frequently, like short periods of walking or changing phone positions (sometimes in the pocket, sometimes on table). The second user (bottom subplot) used active-feedback more than history. The labeling of this user is comprised of many short time-segments and labels that change more frequently (compared to the first user).

4.5.2 Qualitative analysis

In this section, we summarize the feedback gathered in the surveys that the users filled after their use of the ExtraSensory App, and highlight common themes with selected quotes (with participant number in brackets). Among the questions, we asked each user to discuss their preferred and least preferred method of reporting labels, among active-feedback, history, and notifications.

Using active-feedback. Nine users selected active-feedback as their preferred method. A common reason was that it was "more accurate" or as one stated: *"It is easier to say what I'm about to do than try to recall what I did"* [P20]. Some of these users also stated that it was easier for them to remember to provide labels using active-feedback.

Nine other users had an opposite view of active-feedback and selected it as their least preferred method. Some explained it was hard for them to predict exactly what they were about

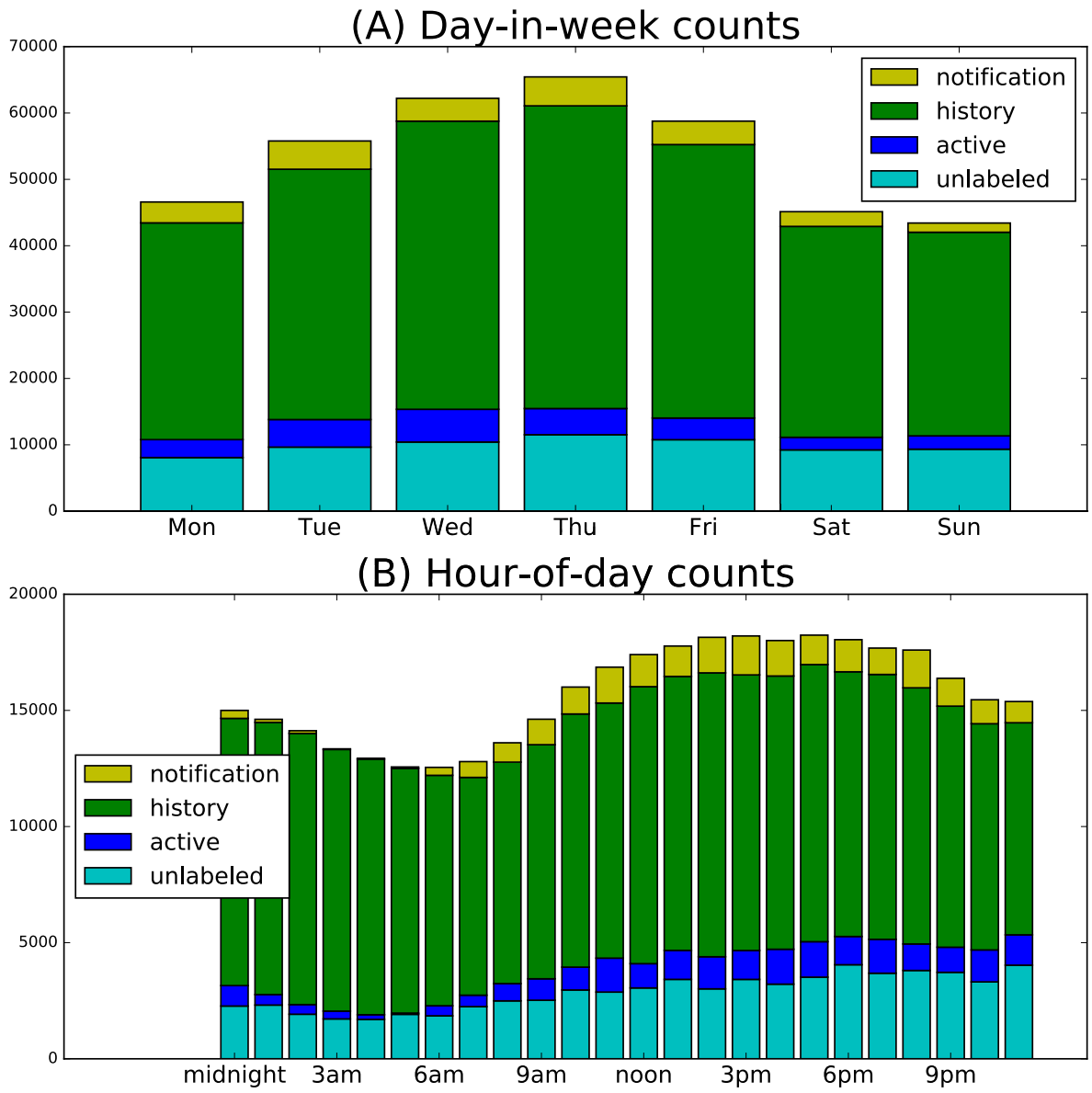

Figure 4.8: Label-reporting mechanisms over time. The color bars illustrate how many recorded minutes were unlabeled and how many were labeled via the different reporting mechanisms, across (A) days of the week and (B) hours of the day. History contributed the majority of labeled minutes in all days and hours. Night-time was covered very little by in-situ reporting (active-feedback or notification).

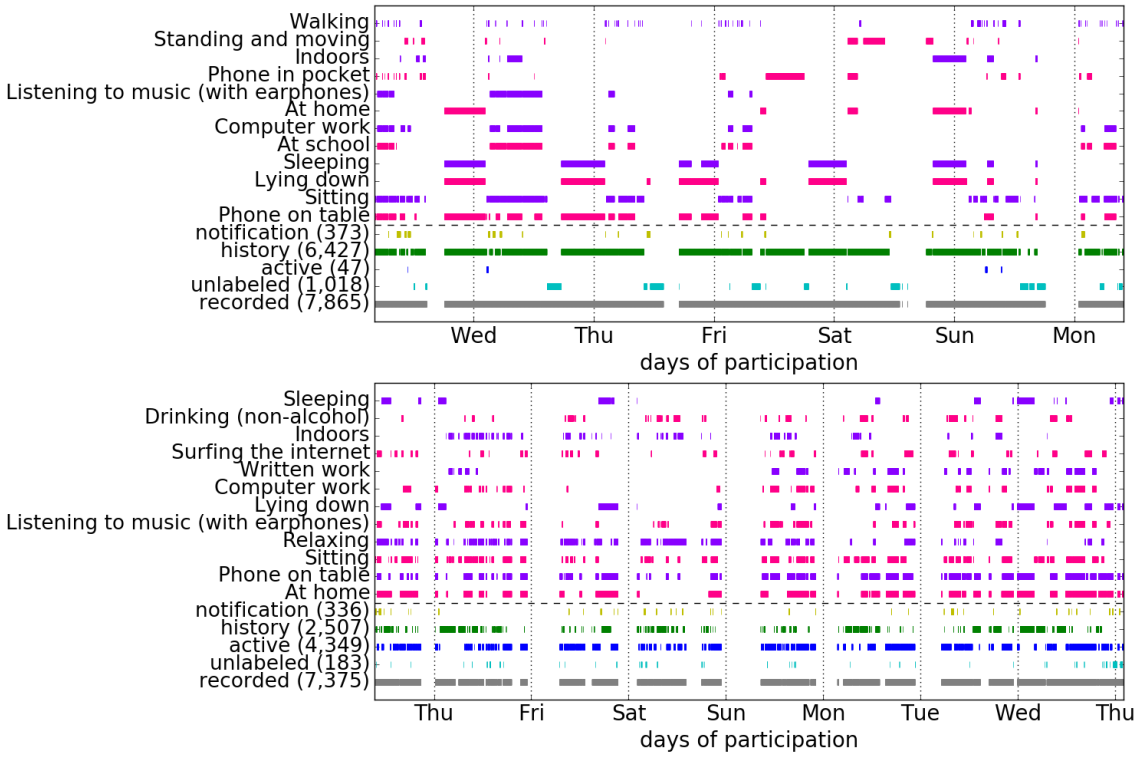

Figure 4.9: Label reporting patterns. Each subplot shows the participation time of a single user. The bottom "recorded" row shows when data-collection was on. The next four rows show for each minute which reporting mechanism was used to report labels for this minute (or if it is unlabeled). For the rows below the horizontal dashed line, the count of relevant minutes is presented in parenthesis. The rows above the horizontal dashed line show the user's most commonly used labels, with the color bars indicating the minutes for which each label was reported. The vertical dotted grid lines indicate the time 6AM in the participation days. The two users demonstrate different styles of label reporting (long time-segments *vs.* fluctuating contexts).

to do or how long it will take them; one stated *"Most of my activities are spontaneous. I found myself edit again what I reported in active-feedback"* [P6, sic]. Some users said they were too busy to use active-feedback or simply forgot to use it. In addition, active-feedback lacked desirable features that other mechanisms had, including adapting to changes in activity and chronological view of the whole day.

Using the history page. 31 users preferred the history page for reporting labels. Multiple themes arose from these users, highlighting different features of the history page:

- Server-guesses. *"Easier to see and confirm or change what was predicted"* [P45].
- Batch-report. *"Could batch-edit entries and change tasks easier"* [P51]; *"You can combine intervals, which made it easy for me"* [P56].
- Free-time interaction. *"I didn't have to be constantly on my phone. Instead I could report activities all at once"* [P23].
- Reporting fine details. *"I felt I could really pin down everything, even if it required more time to accomplish"* [P21]; *"My activity was very varied minute by minute so it was easier to adjust my data"* [P46].
- Easier to recall. *"Most convenient for retrieving relevant memories"* [P27]; *"I do almost the same things everyday so it's convenient looking through history view"* [P34]; *"It was easiest to record my activity when all the data was in front of me"* [P50]; *"I was able to manage my time by viewing history"* [P54]; *"Much easier to remember what I'd done within the view of past events"* [P58].

For 13 users, the history page was the least preferred mechanism. The most prevalent reason was that it is *"less accurate"* [P1], with specific explanations like: *"Hard to remember precise minute labels"* [P33]; *"I wasn't always certain on the minutes and changed activities so I was afraid to give incorrect data"* [P18]; *"Hard to remember when I was doing a lot of things quickly and could not use active feedback"* [P39].

Some users did not like the interaction with the history interface, stating *"Minute-by-minute is*

too specific. Every five minutes would be better" [P5]; *"tedious and inconvenient"* [P48]; or *"takes forever because it guessed something different every minute and the guesses were rarely accurate"* [P60]. These inconveniences were partly due to the real-time classifier on the server that was very basic (trained on little data from only two iPhone users); in sedentary behavior it sometimes alternated between guessing "sitting" and "lying down", which made consecutive minutes appear in the history as separate, un-merged events.

Using notifications. Seven users noted notification as their preferred label-reporting method. Most of them specifically referred to confirmation-notifications (asking whether their context remained the same): *"I mostly do the same thing for extended periods of time and the watch made it easy to respond without using my phone"* [P36]; *"Watch-confirm was very easy to press and confirm"* [P8]; *"When doing the same thing it is less intrusive"* [P28]; *"I mostly forgot or was too busy to change (correct) the context at the moment, but when the question was accurate the notification helped"* [P49].

Three users selected notification as both preferred and least preferred; P26 explained preferring it *"because I had the chance to remember to update the app"* and least preferring it *"because it was stressful."* 34 users least preferred the notification. These users did not benefit from the notifications as a reminder; many of them explained that when they were not reporting labels for a while it was because they were too busy (not because they forgot to report). Some complained about the timing of notifications being inopportune, too frequent, and not based on changes in behavior. This caused negative perception of notifications, ranging from low utility (*"Felt rather useless"* [P11]; *"Not needed much thanks to history view"* [P59]) to different levels of annoyance and stress (*"Somewhat annoying, especially while I was working"* [P41]; *"It was disturbing sometimes"* [P54]; *"Extremely annoying and I find them unkind"* [P29]; *"It was stressful"* [P26]).

Mixed preferences. Most users actually utilized the combination of reporting mechanisms and some reported two mechanisms as preferred, like five users who selected both

active-feedback and history, explaining *"active for short duration events, history for long duration events"* [P14] or *"used active to enter main activities and later whenever I got home I filled the secondary details in history"* [P57].

Uncomfortable situations. We asked the users "Were there any situations where you did not feel comfortable using the app?" Most users (33) answered "no" (one specified *"no. I was open to use the app anywhere I was"* [P13]). From the users who did raise issues, several themes arose:

- Distraction from work. *"My supervisor knows about my participation, but otherwise it would be a problem to be distracted with the phone during work"* [P4]; *"during exam time, notifications were disturbing"* [P48].
- Social politeness. *"In a meeting or out with friends, because did not want to be rude"* [P24]; *"in class — notifications would vibrate loudly"* [P31].
- Privacy concerns. *"conversations felt a bit strange when I knew it was recorded"* [P11]; *"during intimate settings (sex), but other than that no problem"* [P21]; *"I don't usually have my phone in my hand or use my phone at all times, so documenting everything was a little invasive to me"* [P39].
- Practical inconvenience. *"During the weekend, too busy to tag all the activity labels"* [P5]; *"going to sleep — kept getting notifications and had urge to either update or deal with the notification"* [P30]. For some users, the inconvenience was physical so they avoided using the watch or the app altogether: *"some nights, while sleeping, I put off the watch since it was not very comfortable for me"* [P59]; *"sleeping (I don't like sleeping with accessories)"* [P49]; *"didn't use it during the race because it is less aerodynamic"* [P60].

The label menus. To assess the coverage of behaviors that we pre-defined in the label menus, we asked "Were there any situations where you did not know what labels to select? Are there any labels you think are missing from the lists on the app?" 26 replied "no" (some described the label lists as *"comprehensive"* [P4,P6]). Many users suggested specific activities that were

missing (*e.g.* "brushing teeth" or "playing squash"), but they could find more general labels (*e.g.* "grooming" or "exercising", respectively). Some of the suggested labels never crossed our minds when we initially composed the label menus, like "my kids are using my phone" or differentiating between being "in class" as a student *vs.* as a teacher.

After getting such feedback, we added new labels to the "secondary activities" menu, so they became available to the following participants. Also, after about thirty participants, we noticed reoccurring cases of users suggesting labels they could not find although these labels were already in the menu; P30 also mentioned that the topics in the side-bar index were not clear/intuitive. Following that realization, we decided to adjust our protocol for the first meeting and dedicate a few minutes for the new participant to go over the "secondary activities" menu, including looking at the index-topics of the menu. The purpose was to keep the list in the back of their minds, possibly already noticing specific labels that they are likely to use, so that during the study it would be quick and easy for them to find the relevant labels.

In the "main activity" menu, the two versions of standing — "standing in place" and "standing and moving" — caused some confusion; some users described selecting "standing and moving" in situations that involved alternating between standing and shifting from place to place; some said it was hard to distinguish "standing and moving" from "walking." One user indicated that the list did not cover all postures, lacking "crouching" or "kneeling" and some users expected labels like "driving" and "skateboarding" to be in the "main activity" menu instead of the "secondary activities" menu.

4.6 Discussion

Collecting self-reported behavioral data in-the-wild raises the challenge of how to get plenty, detailed, and reliable labels, with minimal interference to natural behavior? We tried to overcome this challenge with the design of ExtraSensory's flexible label-reporting interface. In

this section, we discuss how our solution addressed data-collection trade-offs and outline how this can guide future designs of in-the-wild studies.

Behavioral time *vs.* interaction time. To cover plenty of behavior-time with little interaction-time, we provided mechanisms that allow reporting labels for time segments of variable durations, either in-situ (up to 30 minutes in the future or 20 minutes in the past) or by recall (for today or yesterday). Indeed, the users utilized batch-reporting of whole time segments, and managed to provide labels for the vast majority of their recorded time. Unfortunately, we did not log the duration of interaction with the app, so it is hard to measure how efficient the reporting-interface was.

Detailed labeling in-situ. We asked users for *detailed* labeling (with diverse aspects of behavior). In-situ reporting has a trade-off between labeling with detail and interfering with natural behavior (it takes time to add all the relevant specific labels). This was, however, mitigated by the confirmation-notifications, especially in cases when the recent context remained the same and the user could easily and quickly respond "correct" (either on the phone or the watch). The difficulty in entering detailed-labels was also mitigated by the "frequently-used" link, which made it easier to find labels after a few days. Users who liked the history page also used it in-situ (reporting about a few minutes ago).

Reliable labels with recall. Recall-based reporting has the risk of poorer reliability of the reported labels. To mitigate this problem, the history page combined various features to help the user recall their past context, including server guesses, visually organizing the day chronologically, and the multi-label details in the labeled events. These features, along with the ability to quickly cover long segments of time, made the history page by far the users' most preferred mechanism, which also yielded the majority of labeled minutes.

Flexible interface. Multiple reporting mechanisms significantly contributed to the labeled data, throughout all days of the week and most hours of the day. Active-feedback was used to track quick or temporary changes in behavior, like switching posture or going to the

restroom. Some users used active-feedback to mark time in-situ, and later went over the history page to fill in the details. Using the watch was very popular for confirmation-notifications: while users could also use the history page for such reporting, pressing a watch button is much less intrusive. The flexibility in the interface helped cover different situations and the results show interaction patterns with mixed mechanisms. In addition, the surveys confirm that this flexibility was important to engage users with different preferences and styles of daily behavior.

Open-ended notification perceived negatively. The reporting option of open-ended notification was especially disliked, and correspondingly, it yielded very few labeled minutes. With the advantage of the other mechanisms, there was little use for a blank-notification that acts merely as a reminder, comes at inopportune timings, and sometimes causes stress.

Types of users. Some users were very meticulous and wanted to provide the best data, so they made sure to keep reporting up-to-date labels. While this contributed much labeled data, this may have also affected the authentic nature of their behavior (it is typically not natural for a person to interact with such an app every few minutes). Other users dedicated less effort to labeling, so they contributed less detailed or less accurate labeling, but their recorded behavior was more authentic. The frequently-used labels section seems to have eased this trade-off after a few days of participation.

Our validation users were mostly students and university workers. People with attention-demanding jobs (*e.g.* child care, construction) may tend to use in-situ reporting less but can still report by-recall using the history page. However, generalization to other occupations has to be validated empirically.

Label structure. The labeling form we designed was semi-structured: it had dedicated fields for body-state (main activity) and moods, but it also included the less structured secondary-activities menu. While forcing a single body-state value helped generate a consistent behavioral dimension, it had some disadvantages: in some situations it was confusing and users were not sure which value to select; also, forcing a dedicated field for this dimension made interaction

sometimes inconvenient (more clicks), especially in cases the user did not remember the exact body-state. On the other hand, the multi-label approach that we used in the secondary-activities menu may produce labeling that is inconsistent (a user may accidentally mark both "indoors" and "outside") or incomplete (*e.g.* reporting activity-labels while ignoring environment-labels). However, multi-label can be more convenient and it enables reporting simultaneous activities (like watching TV while eating) and situations that the researcher did not have in mind — thus promoting individual authentic behavior.

Label accuracy. When relying solely on self-reporting, the labeling may be noisy and there is no direct way to assess how accurate are the labels that participants report about themselves. Furthermore, the accuracy of the labels can be inconsistent, because of multiple mechanisms of label reporting and users with different levels of rigorousness. In our previous works [91, 93], we specified simple methods to clean the labels and demonstrated successful machine learning experiments with the data. In [93], we specifically addressed training classifiers with in-the-wild data, which may be inconsistent and have highly unbalanced labeling and occasions of missing labels or sensors. The results were encouraging and facilitate future data collections that are less strict and easier on the participants.

4.6.1 Revised ExtranSensory App

Following our experience in data collection and the user-experience feedback, we revised the mobile application to addresses some of the insights raised in this discussion. We make this revised version of our mobile app publicly available at http://extrasensory.ucsd.edu/ExtraSensoryApp. This public code package includes the full code to the Android phone app, the Pebble watch component, and the server-side code, as well as a fully detailed guide for users and for researchers. We provide this package to allow other researchers to use the app (with or without adjusting its code) for further data collection.

In order to provide better inference, this revised version is based on a server-side real-

time classifier that is now trained on the full data from the 60 users, so its guesses are now more accurate and more detailed — specifying probability values for 51 diverse context-labels. These detailed guesses now appear on the history, providing more cues to recall the true context. The predicted labels also appear sorted by probability in a new "server guess" section in the secondary-activities menu, making it easier to find the relevant labels to select. The researcher can decide to combine all the study's labels (including body-states and moods) in the secondary-activities menu to make the interaction flow easier and allow for more subjective definitions of the user's behavior. Researchers can also easily edit a text file to customize the label menu and its organization by topics.

In addition, the ExtraSensory App now allows disabling open-ended or confirmation notifications, selecting which sensors to record (to reduce battery consumption and communication), selecting classifier to be used on the server, and more.

4.6.2 Future directions

More detailed logging of the user-app interaction can improve assessment of time-efficiency of the interface and of accuracy of the reported labels. Daily self-audit can help users correct their own labeling mistakes. Light-weight body-worn cameras and user-taken phone pictures can augment the self-reporting to help users remember their past context and to allow for external validation of label accuracy.

Further improvements can come from enhancing features of our app, like utilizing real-time guesses to cleverly trigger opportune notifications, or adding more functionality to the watch. Speech-recognition engines will enable voice dictation self-reporting with structured instructions (*e.g.* "start: running, with pet, at the beach, valid for: 30 minutes") and eventually intelligently processed free text (*e.g.* "I'm taking Barkley out for a walk around the block"). These additions and more will reduce the load on users and make interaction smoother and less intrusive.

4.7 Conclusion

In this paper, we introduce the *ExtraSensory App*, a mobile application for collecting data in-the-wild, including sensor measurements and self-reported detailed labels of behavioral context. We validated this app in an in-the-wild study with 60 users. The app's rich label-reporting interface was important to engage users with different behavior styles and phone-interaction preferences and to acquire detailed labels for over 300,000 minutes of diverse behavioral contexts.

ExtraSensory's history page showed to be very useful and the features it offered helped users recall their past context. The additional watch component turned out to be very helpful to keep the user-interaction from interfering with natural behavior. To maximize the utility of the watch, its prompts should be cleverly timed and require minimal reaction (single button press). Ongoing data collection and re-training of real-time classifiers will improve the server guesses and notifications and make user interaction easier and less time consuming.

We believe that the insights that we describe in this paper will inspire future designs of in-the-wild data-collection studies. The public version of the ExtraSensory App that we provide will allow for further collections of data and studies that use real-time context-recognition in-the-wild for various applications in health-monitoring, aging-care, and other domains.

4.8 Acknowledgements

Chapter 4, in full, is a reprint of the material as it will appear in the ACM Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018), April 2018, Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel. The dissertation author was the primary investigator and co-author of this paper.

Chapter 5

Discussion and future directions

In this work, I address the problem of behavioral context recognition, and specifically aim to solve this problem *in-the-wild*. In the introduction (Chapter 1), I described three main parts to research in the field ("sensors and contexts", data collection, and "AI and ML"). In this chapter, I go over each one to summarize the progress made in this work and the possible future steps. These three aspects, of course, depend on one another. Figure 5.1 illustrates some of these dependencies. In order address the variability of behavior in-the-wild, I made certain choices for how to define the context-recognition problem, like treating each minute as a single example, supporting unconstrained device placement, utilizing multi-modal sensors, and describing context using a multi-label formulation with a large vocabulary of context-labels. These decisions affect both the data collection procedures and the AI/ML solutions. The methods of data collection in-the-wild, from many participants in their own natural lives, resulted in irregular data, which the AI solutions need to support.

After my experience in actually collecting data and using that data with machine learning experiments, I can return to the planning of the framework, and suggest improvements for future work. Some of these improvements were already incorporated into the revised version of the ExtraSensory App, which is now publicly available (as full source code) for free at http:/extrasensory.ucsd.edu/ExtraSensoryApp.

5.1 Sensors and contexts

5.1.1 Multi-modal sensors

The usage of **multi-modal sensors** showed to be very important to overcome the variability of behavior in-the-wild, and resolve ambiguity of context. In our ML experiments, there were some sensing modalities that seemed particularly informative about certain aspects of behavior (*e.g.* audio to recognize environment, watch-acceleration to recognize certain home activities). Other modalities (*e.g.* phone magnetometer, watch compass) did not result in improved

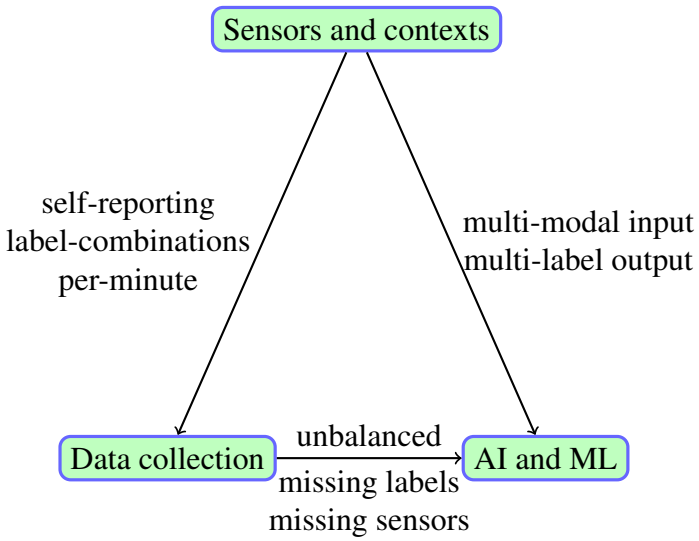

Figure 5.1: Three aspects of research in behavioral context recognition, and the relations between them. The definition of the context-recognition problem (sensors and contexts) influences data collection: it determines that we want to collect self-reported labels per-recorded-minute, with detail (combinations of context-labels). The definition of the problem also influences the proper AI models and ML solutions to solve the problem — they need to process multi-modal sensor-inputs and provide recognition of context in a multi-label formulation. The AI/ML solutions also depend on the methods of data collection: collecting data in-the-wild from many independent participants results in irregular data, with highly unbalanced labeling, missing labels, and missing sensors; the AI/ML solutions should handle that type of data.

recognition; this may be because these modalities carry less information about the examined contexts, or because we did not yet discover the "perfect" feature extraction methods for these modalities. Following these observations, I included more flexibility in sensor-recording in the revised ExtraSensory App — now researchers (or users) can select to record a subset of the sensors available on the phone, which will help reduce battery consumption and communication of data.

Future studies can augment phone and watch sensors with more modalities, like stationary sensors in a fixed environment (*e.g.* home, office). These can provide more information to the context-recognition system, and can also serve for external validation of the self-reported labels in data collection efforts; for example, binary state-change sensors can give reliable indications of opening kitchen cabinets or refrigerator, and confirm user provided labels of cooking activity. The addition of more sensor-inputs may seem to invite the "curse of dimensionality" and risk having the AI model over-fit to the specific realizations of multiple-sensor features that appear in the training data. However, this dissertation describes methods to reduce the dimensionality and ease this risk, either by training separate-sensor classifiers and using late-fusion (as described in Chapter 2), or with a single model with hidden layers of reduced dimension (as described in Chapter 3). Chapter 3 further presents methods to utilize the same model with multi-modal inputs even when some of the inputs may be missing. These methods encourage further studies with multiple sensing modalities, even if some are only available when the person is at home and some are only available when the person is wearing a watch, *etc.*

5.1.2 Multi-label contexts

The usage of **multi-label** descriptions of behavior provides a convenient way to represent a wide variety of behaviors, while keeping the representation simple enough for machines to interface with (both the context-recognition system and the application that uses the recognized

context). It enables a more elegant way to represent different flavors of activities that have something in common (*e.g.*, "running outside", "running on a treadmill") instead of having to define a special category for each of these flavors. It also provides a multi-dimensional way to represent combinations of different behavioral aspects, like activity, location, and interaction with others. These dimensions are, of course, statistically dependent on one another (the role of estimation or machine learning will be to capture those dependencies), but many combinations are feasible, like "running, at the beach, with friends", "running, at the gym, alone", *etc.* When designing a context-recognition system for very particular applications (*e.g.* tracking people's transportation modes [29]), it can be appropriate to use the multi-class approach — define a small set of mutually-exclusive categories. However, when aiming for a more general usage of a context-recognition system or to capture a broader range of everyday behaviors, the multi-class approach can be limiting, for example to represent multiple activities at the same time, like "eating while watching TV". This may also have consequence on the data collection procedure, where participants may find themselves having to choose which activity to perform, eat or watch TV. The multi-label approach, however, gives more flexibility to represent a wide variety of behaviors (including situations that the researcher never though of, but may occur in-the-wild), and helps maintain a more natural and authentic behavior in the data collection.

The choice of using multi-label representation can have a harmful effect on the data collection procedure. The action of selecting multiple relevant labels (especially when the menu is large) can sometimes be tedious and time consuming. This is why, in the design of the ExtraSensory App, we tried to add features to make it easier to quickly report relevant labels, like "frequently used" labels section or single click confirmation of continuing the same context. More features can help make the interaction easier, like having ready combinations of labels that are likely to be selected or using a voice recognition system. On the other hand, a possible advantages to using a multi-label representation is that it enables users to provide partial labeling (*e.g.* just a body-state label) in-situ, and later (when they have more free time to interact with

the app), add more fine details. In addition, the multi-label approach for self-reporting gave participants more flexibility to subjectively describe their own behavior, decide for themselves what were the more important context-labels in their mind, and possibly ignore some dimensions (*e.g.* reporting eating in a restaurant, but neglecting to mention if they are alone or with friends, or where their phone is positioned). When we designed the ExtraSensory App, we actually started with only a multi-class label menu (for the "main activity" — the body state), and then we added the "secondary activities" field to allow for multi-label selection of more details. Some participants complained that this complicates the interaction (too many clicks), and sometimes a person did not remember their body state, but they did want to report that they were eating in a restaurant; in such cases, although there was an option to select "Don't remember" as main activity, this was a waste of time. So, in hind sight, I believe it would have been better to have just a single multi-label selection menu, and to have the body-state categories (sitting, walking, running, *etc.*) included in it, as well as the mood labels. This will simplify things for the participant (less clicks to find all the relevant labels) and for the researcher (less rules and structure to think of — simply letting the user select whatever labels they see fit). Another reason to include the body-state labels in the same pool of labels with the others is that they don't necessarily describe what a person would consider their "main activity" — it is reasonable that in some context a person would mainly describe their context/activity as cooking or in a meeting, and the body posture will be considered as a secondary detail.

Deciding to use the multi-label formulation defines the context-recognition problem as a multi-label classification problem and affects the AI/ML solutions. A convenient way is to treat inference as a set of binary classification problems (for each context-label, is it relevant or not?). In this work, I use this approach for evaluating recognition performance — a score (*e.g.* balanced accuracy) is evaluated for each context-label independently, and then averaged over labels. A harder task would be to try to recognize the exact label-combination — that hard task would also suffer much more from the noisiness of the ground truth labels and from lack of

examples for each particular combination. In the estimation (or training) part, the multi-label approach adds more challenges, like the fact that the in-the-wild data tends to have incomplete labeling (some people may completely ignore some labels). To help partially overcome this, the machine learning methods I describe in Chapter 3 use common-sense rules to determine when some label-entries are better regarded as "missing" and an objective function that makes sure that these entries will not influence the learning.

The way that we collected multi-label data (with combinations of more than 3 labels from a menu of over 100 context-labels) also brings up the curse of dimensionality for estimation (learning) — the fact that the system can have so many possible outputs (exponential in the number of context-labels in the menu) suggests the risk that we cannot collect sufficient examples to reliably model these different label-combinations. However, there are strong dependencies among labels, so most "possible" output label-combinations have very low probability (or are practically infeasible) in-the-wild; the models I presented, with narrow-bottleneck hidden layers, can capture these complicated dependencies. It is also possible that with enough training examples of basic combinations (*e.g.* all the feasible combinations of two context-labels), a machine will be able to learn deeper relations and even successfully recognize new combinations of labels that it was not exposed to during learning (*e.g.* recognizing "walking, at the beach, with friends" after being trained with only examples of "walking, at the beach, alone" and "walking, indoors, with friends"). Additionally, a similar "curse of dimensionality" risk also exists with the traditional multi-class approach, when trying to capture a broad range of behaviors (or distinguish behaviors with finer resolution), so I will discuss this issue in the next topic of labeling-resolution.

5.1.3 Labeling resolution

There is a significance to the choice of the labeling-resolution in **both label-detail and time**. In this work, I aimed to capture a broad range of everyday behaviors, and with

multi-dimensional details. This decision was done for several reasons:

- This work was not motivated by one specific application. The goal was to have a unifying framework that will eventually serve many applications, in medicine/health, commercial products, or many other domains we do not yet have in mind. However, having a more concrete purpose can help direct system-design and data collection, and make research easier (*e.g.* self-reporting with much fewer options of labels) and better suited for the practical application.

- As part of our goal to capture in-the-wild behavior, we did not want to impose too much of our researcher assumptions, and we wanted to let the crowd — the actual participants — guide us with "what types of behavior are out there" (at least among the set of campus-people in our data collection). This is why we had a large label-menu and the flexibility of selecting multiple labels (perhaps participants can surprise us with situations we did not have in mind).

- We decided to collect data with fine-grained labels, with the possibility of later post-processing the labels to cluster multiple labels together in a rougher resolution. This was actually done, for example, to re-define the context-label "exercise" as a combination (with logical OR operation) of different activities (including running, playing baseball, *etc.*)

As mentioned earlier, the attempt to address a fine resolution of behaviors (with either multi-class or multi-label representations) makes estimation harder and requires the collection of many more training examples to properly model the different behaviors. In the data collection, we demonstrated the ability to deploy our methods in large scale and collect many more examples than previous studies. I also demonstrated good recognition (with generalization to unseen users) of a wide variety of context-labels, using the ExtraSensory Dataset. Of course, there is always the risk that when these systems will be validated on other sectors of the population (different occupations, age groups, places in the world), they generalize poorly.

The choice of time-resolution (I chose to regard to each minute as a separate example, with its own behavioral context labels) is also related to the end-goal applications. I tried to think of a basic time unit that is reasonable to describe people's everyday behavior in; "every minute" seemed like a good balance: fine-enough to represent transient changes (like getting up from a chair or going to the restroom for two minutes), but not too frequent to distinguish typical human behavior. Of course, this design decision was influenced by the list of possible behaviors or potential applications that I had in mind. Having a particular application can help better direct the choice of time-resolution: if a researcher is interested in analyzing people's commute modes of transportation, it may be sufficient to represent blocks of 10 minutes (to capture bike rides, car drives or train rides); in such a case you may not care about the fine details (and exact timings) within a commute, like when the person got up from a seat in the train or took a phone call, you just care that the person took the train to work that day.

Both the label-space and time resolutions affect the self-reporting interaction and may also compete with one another: it can be easy enough to track whether you are eating or not (single binary representation) every minute, or to describe your detailed context (where you are, what you did, whom with, *etc.*) for every hour, but it is harder to keep track of fine details and every minute. As I report in Chapter 4, several participants in the data collection complained that reporting details for every minute was hard — it was tedious and it was hard for them to remember their exact timing of past context. Some people decided to dedicate less effort and mostly describe their "overall" context for longer periods of time. Others were more meticulous and tried to report fine details per-minute, but it sometimes resulted in them interacting with the app too frequently (causing them to behave less naturally). In future data collections, a more focused list of labels or application can help decide on an easier time resolution and make things easier on participants and more consistent across different users.

5.2 Data collection

In the introduction of this dissertation, in the contributions section 1.4, I compared different approaches for data collection with regard to two conflicting objectives: the ability to collect reliable labels and maintaining in-the-wild (natural, authentic) behavior. An illustration of this is presented in Figure 5.2. As mentioned in the introduction, the self-reporting methods I employed in this work were based on previous approaches, and their combination helps ease the trade-off between reliable labels and in-the-wild behavior: the history page makes reporting by-recall generate more reliable labels; active-feedback makes the timer approach less intrusive (by allowing the user to mark time with partial details and add more labels or correct mistakes later) and a bit more reliable (by limiting foresight to max 30 minutes, where the user can later extend the duration); confirmation-notification using the watch is a very unobtrusive way to reply to ESM queries, and potentially report detailed labels for multiple minutes in the near past. In addition, the methods we used for data collection proved to be effective to collect plenty of labeled minutes, with detailed labels, and to support all times of day and days of the week (see Figure 4.8), cover diverse everyday situations (see Figure 4.7), and engage people with different styles of daily behavior or interaction-preferences (see Figure 4.9).

5.2.1 Potential hybrid methods

There is still room for improvements and progress. Figure 5.2 presents also examples of potential improvements. With the development of smaller wearable cameras, they can become much more comfortable to wear, or even unnoticeable. If such devices become also cheap, it will be more practical to use them in data collection efforts. In such cases, the behavior can be much more natural and authentic (compared to hanging a SenseCam around the neck [20], or GoPro mounted on the chest [68]). This will enable to collect very “in-the-wild” behavior, and still acquire detailed labels with high reliability (assuming the context is visible in the images).

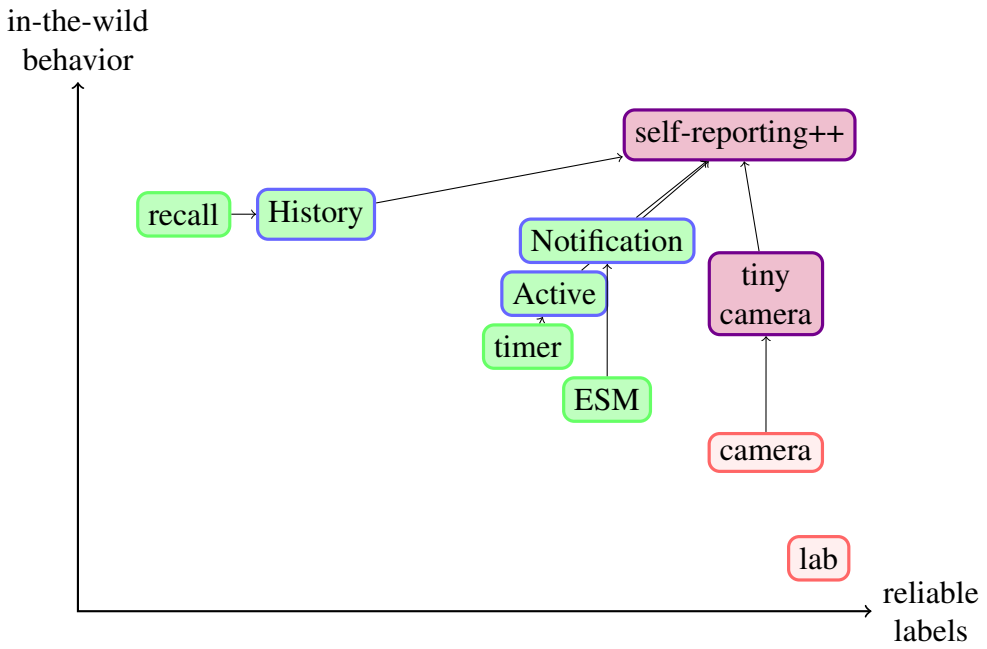

Figure 5.2: Data collection approaches, in previous works, this work, and possible future methods — trade-off between reliable labels and in-the-wild behavior. Approaches that rely on self-reporting using everyday devices (phone, watch) are marked in green: ESM = experience sampling method, timer (when the participant initiate reporting labels in-situ and marks start and stop time of a selected activity), and recall-based self-reporting. Other approaches are marked in red: data collection in a lab, and data collection outside of the lab, using a wearable camera. The self-reporting methods in this work are based on previous methods marked with blue frames: history page, active-feedback and notifications. The added features of the ExtraSensory App and the combination of multiple methods helped ease the trade-off and achieve better balance between reliable labels and in-the-wild behavior. Improvements for future data collections (marked in purple) may come from tiny, unobtrusive (and cheap), wearable cameras, or with combinations of self-reporting and other tools.

Studies will have to address the issue of privacy, either by putting the annotation load on the subject or with creative solutions, like automatic face masking. Combination of self-reporting and other methods (represented in the illustration in Figure 5.2 as "self-reporting++") can reach even better balance between reliable labels and in-the-wild behavior. The different methods can complement each other, for example, relying more on a tiny wearable camera to annotate busy times like at work, and relying more on manual self-reporting for private behaviors like toilet or shower. Other augmented tools can be voice recognition to dictate annotations, mementos like the user recording her speech or taking a snapshot to make it easier later to remember the current context. Other cues can be pictures you take on your phone regardless of participating in data collection or specific locations you visit during the day — these can be presented as extra cues in the history page to help remember earlier context.

5.2.2 Utilizing real-time automated context-recognition

As part of the conclusions of collecting the ExtraSensory Dataset, we saw that the **server-guesses** were a very helpful feature that contributed to the popularity of reporting with the history page. However, some participants complained that these guesses were often wrong or they fluctuated too much, causing them to be unhelpful or even confusing. Having better server-guesses can reduce annoyance, reduce effort, and help participants better remember their past context. Automated real-time recognition (like ExtraSensory App's server-guesses) can help also in-situ self-reporting: the interface can present the labels in a menu according to order of their guessed probability to help the user find relevant labels quickly; using the system's "belief" of the current context can help detect when the behavior is changing and initiate notifications in more clever timings.

Since watch-confirmation is such an effective tool to get labels with little effort and interruption, the notification mechanism can make more use of the automated context-recognition (the guesses) also to come up with **better questions to ask** — valuable questions that require a

binary response (single button click, yes or no) but have a lot of entropy (the user's response will supply maximal information). In our data collection, the way to do it was to ask a yes/no question about a very detailed context, but at special timing (a few minutes after this specific context was reported by the user). Generally, a question like "in the past 12 minutes were you sitting (on a bus, phone in pocket)" is very likely to get the answer "no", so it generally has very low entropy and is not useful to be asked at arbitrary times. However, if the user recently reported the context "sitting (on a bus, phone in pocket)", this evidence raises the probability that this context is still the same to be closer to half, making the question have higher entropy and have more value. When the server-guesses are stronger (more accurate, more detailed), as they are in the revised version of the ExtraSensory App, this gives an opportunity to use them to ask better single-click questions using notifications. This can be a form of active-learning[1], to get the most information from few examples.

To utilize the server-guesses in such ways, the classifier on the server needs to be trained and improved. The **automation feedback-loop** can continue supporting this (see Figure 5.3): the more data (with manual labeling) is collected, the context-recognition classifier can be improved, and the better the real-time classifier, the more helpful the server-guesses be to the next generation of data collection participants. The usage of this feedback loop should be done carefully — there's a risk that it will converge to a biased state; for example, if the server-classifier tends to give higher probability to "indoors" than to "at home" (even when both labels are perfectly relevant), and the user-interface suggests labels ordered by the guessed probability, then users may tend to only mark the higher probability labels, and the system's bias will only be reinforced instead of reduced. Some methods to avoid this can be making sure that there are additional sources for the manual reporting (not completely depending on the server-guesses) or controlling and experimenting with how the server guesses influence the user-interface.

[1] Do not confuse: in active-feedback, the *user* is actively initiating to report labels; in active-learning, the *system* is actively initiating to solicit labels from the user

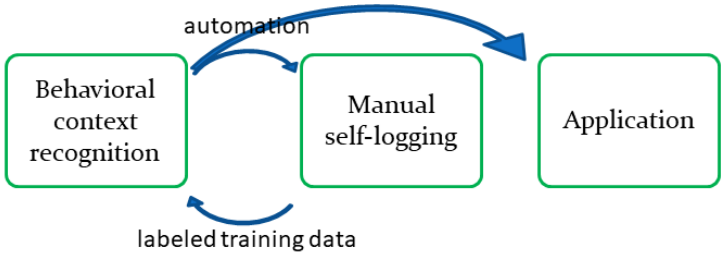

Figure 5.3: Automation feedback-loop for data collection, with detour for practical applications that use context-recognition. Data that was collected with manual labeling by participants is used to train better context-recognition systems (better classifiers). The improved classifier can be plugged-in to the server-side of the ExtraSensory App and provide better real-time recognition. This improved automated recognition, in turn, will make it easier for the next generation of participants to provide more labeled data. This cycle will go on supporting itself. Besides serving the next data collection effort, the context-recognition system can serve practical applications.

5.2.3 Measuring label reliability

Although in Figure 5.2 I illustrate the potential for reliable labels in the different methods, these are not based on quantified or objective measures. Assessing the reliability or accuracy of labels collected in-the-wild still remains a problem. Other AI tasks that enable convenient offline annotation by researchers, usually also enable measuring accuracy of the annotations or getting rid of less reliable annotations. For example, for object recognition in natural images, researchers can hire multiple annotators to label the same images, and quantify label reliability based on agreement among annotators. This also enables later auditing of the labeling, to ignore the labels from a less-reliable person or to re-check the labels of examples that showed problems in classification experiments. These methods translate to behavioral context recognition when using wearable cameras, but not when relying solely on self-reporting. In self-reporting, the subject is the only "expert" to report their own behavior, and there is no other gold-standard to compare to.

One solution to gain ways to assess label reliability will be to use wearable cameras (once they are unobtrusive and cheap enough) in addition to self-reporting. This will enable to do

external validation (with objective annotators) for a subset of the participants. Another possible way is to experiment with the self-reporting interface — manipulate it in different ways for different participants to get a sense of how it influences biases in manual reporting and possibly to catch obvious labeling mistakes. This can be done, for example, by playing with the order of suggested context-labels in the menu: if participants consistently pick the labels that appear on top (even if the order was ascending, descending, or random order of guessed-probabilities), it may mean that users are blindly following the system's recommended labels without thinking too much. Another example is manipulating the notification questions: while most of the time asking for the "valuable" (high entropy, where the server is uncertain) questions, the system can sometimes "trick" the user by asking about a label that the system is very confident is relevant or a label that the system is very confident is not relevant (to check if the user lazily responds). Those experiments can provide some indirect assessments of label reliability and also serve to compare the usefulness of different label collection methods.

Another indirect way to assess the labels' quality is to get out of the automation feedback-loop and actually use the current context-recognition system for a practical application. When there is a concrete use case, end-goal users (or customers) can provide feedback about how useful the application is or how well it works. This, of course, is a longer-term mechanism but it can still indicate that there is progress in the context-recognition systems or in the data that was used to train them.

5.3 Artificial intelligence (AI) and machine learning (ML)

In this work, my main motivation was to capture in-the-wild behavior. In defining the context-recognition problem or designing the data collection procedure, I did not focus on "making it easy for the AI solution". For example, it was important that the participants use their phone and carry it as convenient and natural to them; this makes it harder to recognize activities

(*e.g.* walking) because the sensors pick up something different when the phone is in a pocket and when the phone is in the hand. As I describe in Chapter 2, having multiple complementary modalities helped overcome the variability in behavior.

5.3.1 Overcoming the curse of dimensionality

Another difficulty that I imposed on solving the context-recognition problem is the curse of dimensionality, which I mentioned earlier. This is relevant both to the many sensor features from different modalities (in my experiments, I use 175 features from six sensors) and to the fine resolution of behaviors (there are many possible combinations of the 51 context-labels analyzed in this work). The more input or output variables you want to model (estimate their joint distribution), the exponentially more data you need. This was partially addressed before solving the AI problem:

- We collected a (relatively) large dataset from many people. We did collect many examples for contexts like sleeping or at home (but not necessarily many examples for all feasible combinations of context-labels).

- Many of the context-label combinations have very low probability to occur in-the-wild (*e.g.* walking while lying down is practically infeasible), so it is fine that we do not have sufficient (or at all) data from these situations to estimate their sensor-to-context dependencies, because the system would not have to deal with those situations in practice.

- I decided to simplify the multi-label classification problem by evaluating it as a set of (conditionally) independent binary classifications.

As for the AI solutions to overcome the curse of dimensionality, dimensionality reduction seems to be a key method. In Chapter 3, I demonstrate the advantage of having a model with hidden layers that are narrow enough to reduce the total number of free parameters in the model

(compared to a linear model). When having much fewer parameters to estimate, you can do it successfully (generalize well) with the limited data that you have. The reduced-dimension models (MLPs) that I examined seem to fit well in sensor-based context-recognition, because there are strong dependencies among sensor-features and among context-labels (*e.g.* "walking" and "running" have some similarities, "indoors" and "outside" do not co-occur, "sleeping" mostly occurs with "at home"); these dependencies were successfully captured in the hidden layers of the MLPs. Figure 5.4 presents the experimental results with the MLPs with two hidden layers, and compared to the linear model (LR — separate logistic-regression per-label). These results were also presented in Table 3.1. The gap between the train score (green) and the test score (blue) indicates the tendency of the model to over-fit to the training set. The linear model, which tries to estimate the "correlation" (a regression coefficient) for each pair of sensor-feature and context-label, has too many parameters (over 8,000) to estimate; as a result, it suffers from the curse of dimensionality and has poor generalization to unseen people (the test set). However, the MLPs that have narrow-enough hidden layers (with 32 nodes or less) have less total number of parameters in the model, and as a result have a much smaller gap between train and test performance (less over-fitting, better generalization).

5.3.2 Training with irregular data

One challenge that comes with collecting data in-the-wild, from many participants, is irregular data. Because we wanted each person to engage in their own individual, authentic behavior, the resulting dataset has examples of different contexts from different people. We also did not control how many examples we collect from each context, because we recorded regular natural behavior. This caused the ExtraSensory Dataset to have incomplete labeling (a person may have never reported the label "eating" even though they did eat during the week of participation), and highly-unbalanced labeling (for all context-labels there were many more negative examples and the positive/negative ratio had a wide range among labels).

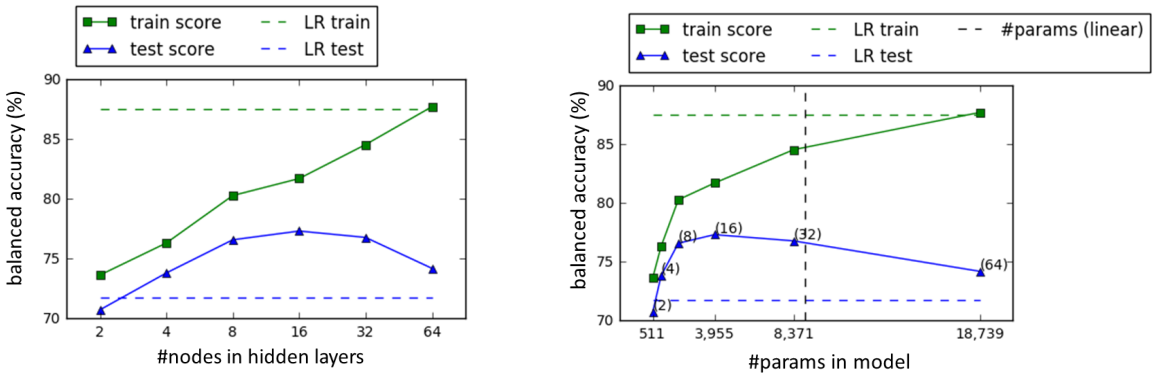

Figure 5.4: Recognition performance with a linear model *vs.* multi-layer perceptrons with various dimensions of hidden layers. Performance is measured in balanced accuracy averaged over the 51 context-labels. LR represents the logistic-regression system that has a separate model per-label. The solid line and shapes represent MLP with two hidden layers of the same dimension. The green lines mark the performance on the train set and the blue lines mark the performance on the test set. In the left plot, the x-axis represents the dimension of the hidden layers (number of nodes in the layer) and in the right plot, x-axis represents the total number of parameters in the model. The vertical dashed line marks the number of parameters in a linear model, for comparison.

To combat both these properties of the data, I utilized instance-weighting, which is a practical and convenient tool when training is done with an objective function that has a sum over instances. To get over the imbalance in the training data, the instance-weights were designed inversely proportional to the frequencies of the positive/negative class in every label. This neutralization of the positive/negative ratio in the training set fits well with the performance metric that I chose to use for evaluation — the balanced accuracy (average of true positive rate and true negative rate). A potential problem with this method is over-sensitivity to outliers — in the attempt to raise emphasis of rare cases (typically positive examples), this can give too much emphasis to examples that are not well-representing actual data or are wrongfully labeled.

I utilized the same method (instance-weighting) to address incomplete labeling as well, by inserting zero-weights to certain entries in the objective function. However, this relied on first determining or inferring when specific example-label pairs are better considered as "missing label". In Chapter 3, I describe simple common-sense rules to declare missing labels, but these rules are not complete and there are still cases where I included in the objective function

entries as negative examples, even though some of them probably reflected positive examples (*e.g.* Figure 4.9 shows an example of a user who reported the label "sleeping" during some of the days, but had two complete days without reporting this label, even though it is reasonable to believe they did sleep some of that time). Further solutions are required to address training a multi-label classifier with incomplete labeling.

Another type of irregular data is missing sensors. In Chapter 3, I address this issue for both training and inference. To practically handle missing sensor-features, the simple policy of zero-imputation (after centering features around zero) enables using many more examples for training, and sensor-dropout helped prepare the MLP to better recognize context when some of the sensors are missing. In my experiments, based on six sensing-modalities, these methods improved recognition of most behavioral aspect, and for all the six modalities as missing (see Figure 3.5). However, if the training data's sensor-features become more sparse (with many more "holes" of missing sensors), the training can be less successful. An alternative, is to train separate parts of the model for different sensors, and fuse them with late fusion, like I describe in Chapter 2; this may be a better way to address stronger separations of sensor measurements, for example if the system includes stationary sensors at home and stationary sensors at work, which never sense the person at the same time.

5.3.3 Open problems for future improvements

Further improvements of AI and ML tools can yield progress in context-recognition and in data collection. As I experienced, data collection in-the-wild is very challenging, especially acquiring labels. If semi-supervised learning methods succeed to improve recognition by using unlabeled examples, this will contribute to future data collection efforts. Same can be achieved by successful active-learning methods — these can be practically integrated into data collection procedures with the notification mechanism — probing the user to answer questions only with the most valuable questions (*e.g.* particular minutes and particular contexts where the real-time

recognition is uncertain).

More open machine learning problems are adaptation to new populations, personalization, feature learning, time-series modeling, and more.

5.4 Conclusion

To encourage further progress in behavioral context recognition, it was important for me to provide useful sources to the research community. The ExtraSensory Dataset is publicly available at http://extrasensory.ucsd.edu and can serve to develop and evaluate context-recognition or machine learning methods. The ExtraSensory App is publicly available at http://extrasensory.ucsd.edu/ExtraSensoryApp and researchers are encouraged to use it to collect more data in-the-wild, perform experiment about data collection, or use its real-time context-recognition framework for certain practical applications. I hope this work helps engage discussion, progress research in behavioral context recognition, and promotes studies in-the-wild.

Bibliography

[1] Phil Adams, Mashfiqui Rabbi, Tauhidur Rahman, Mark Matthews, Amy Voida, Geri Gay, Tanzeem Choudhury, and Stephen Voida. Towards personal stress informatics: Comparing minimally invasive techniques for measuring daily stress in the wild. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 72–79. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[2] Oliver Amft and Gerhard Tröster. Recognition of dietary activity events using on-body sensors. *Artificial intelligence in medicine*, 42(2):121–136, 2008.

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient assisted living and home care*, pages 216–223. Springer, 2012.

[4] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *Pervasive computing*, pages 1–17. Springer, 2004.

[5] Mathias Basner, Kenneth M Fomberstein, Farid M Razavi, Siobhan Banks, Jeffrey H William, Roger R Rosa, and David F Dinges. American time use survey: sleep time and its relationship to waking activities. *SLEEP-NEW YORK THEN WESTCHESTER-*, 30(9):1085, 2007.

[6] Abdelkareem Bedri, Apoorva Verlekar, Edison Thomaz, Valerie Avva, and Thad Starner. A wearable system for detecting eating activities with proximity sensors in the outer ear. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 91–92. ACM, 2015.

[7] Frank Bentley and Konrad Tollmar. The power of mobile notifications to increase wellbeing logging behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1095–1098, New York, NY, USA, 2013. ACM.

[8] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R Schmidtke, and Michael Beigl. Actiserv: Activity recognition service for mobile phones. In *Wearable Computers (ISWC), 2010 International Symposium on*, pages 1–8. IEEE, 2010.

[9] Søren Brage, Ulf Ekelund, Niels Brage, Mark A Hennings, Karsten Froberg, Paul W Franks, and Nicholas J Wareham. Hierarchy of individual calibration levels for heart rate and accelerometry to measure physical activity. *Journal of Applied Physiology*, 103(2):682–692, 2007.

[10] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 3121–3124. IEEE, 2010.

[11] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3):33:1–33:33, January 2014.

[12] Daniel Castro, Steven Hickson, Vinay Bettadapura, Edison Thomaz, Gregory Abowd, Henrik Christensen, and Irfan Essa. Predicting daily activities from egocentric images using deep learning. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 75–82. ACM, 2015.

[13] Jingyuan Cheng, Oliver Amft, and Paul Lukowicz. Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In *Pervasive Computing*, pages 319–336. Springer, 2010.

[14] Tanzeem Choudhury, Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Anthony LaMarca, Louis LeGrand, Ali Rahimi, Adam Rea, G Bordello, Bruce Hemingway, et al. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2), 2008.

[15] Sunny Consolvo, David W McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, et al. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1797–1806. ACM, 2008.

[16] Mihaly Csikszentmihalyi and Reed Larson. Validity and reliability of the experience-sampling method. In *Flow and the foundations of positive psychology*, pages 35–54. Springer, 2014.

[17] Anind K Dey, Katarzyna Wac, Denzil Ferreira, Kevin Tassini, Jin-Hyuk Hong, and Julian Ramos. Getting closer: an empirical investigation of the proximity of user to their smart phones. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 163–172. ACM, 2011.

[18] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. Detecting periods of eating during free-living by tracking wrist motion. *IEEE journal of biomedical and health informatics*, 18(4):1253–1260, 2014.

[19] Anup Doshi, Brendan Morris, and Mohan Trivedi. On-road prediction of driver's intent with multimodal sensory cues. *IEEE Pervasive Computing*, 10(3):22–34, 2011.

[20] Katherine Ellis, Suneeta Godbole, Jacqueline Kerr, and Gert Lanckriet. Multi-sensor physical activity recognition in free-living. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 431–440. ACM, 2014.

[21] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[22] Antti J Eronen, Vesa T Peltonen, Juha T Tuomi, Anssi P Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329, 2006.

[23] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and João MP Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.

[24] Jordi Fonollosa, Irene Rodriguez-Lujan, Abhijit V Shevade, Margie L Homer, Margaret A Ryan, and Ramón Huerta. Human activity monitoring using gas sensor arrays. *Sensors and Actuators B: Chemical*, 199:398–402, 2014.

[25] Raghu Kiran Ganti, Soundararajan Srinivasan, and Aca Gacic. Multisensor fusion in smartphones for lifestyle monitoring. In *2010 International Conference on Body Sensor Networks*, pages 36–43. IEEE, 2010.

[26] John J Guiry, Pepijn van de Ven, and John Nelson. Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices. *Sensors*, 14:5687–5701, 2014.

[27] Manhyung Han, Young-Koo Lee, Sungyoung Lee, et al. Comprehensive context recognizer based on multimodal sensors in a smartphone. *Sensors*, 12(9):12588–12605, 2012.

[28] Erik B Hekler, Predrag Klasnja, Vicente Traver, and Monique Hendriks. Realizing effective behavioral management of health: the metamorphosis of behavioral science methods. *IEEE pulse*, 4(5):29–34, 2013.

[29] 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.

[30] Javier Hernandez, Daniel J McDuff, and Rosalind W Picard. Biophone: Physiology monitoring from peripheral smartphone motions. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7180–7183. IEEE, 2015.

[31] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmaki, and Hannu TT Toivonen. Time series segmentation for context recognition in mobile devices. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 203–210. IEEE, 2001.

[32] Stephen Intille. The precision medicine initiative and pervasive health research. *IEEE Pervasive Computing*, 15(1):88–91, 2016.

[33] Stephen S Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer S Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Pervasive Computing*, pages 349–365. Springer, 2006.

[34] Daniel Kahneman, Alan B Krueger, David A Schkade, Norbert Schwarz, and Arthur A Stone. A survey method for characterizing daily life experience: The day reconstruction method. *Science*, 306(5702):1776–1780, 2004.

[35] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, 2006.

[36] Jacqueline Kerr, Ruth E Patterson, Katherine Ellis, Suneeta Godbole, Eileen Johnson, Gert Lanckriet, and John Staudenmayer. Objective assessment of physical activity: Classifiers for public health. *Medicine and science in sports and exercise*, 48(5):951–957, 2016.

[37] Adil Mehmood Khan, Ali Tufail, Asad Masood Khattak, and Teemu H Laine. Activity recognition on smartphones via sensor-fusion and kda-based svms. *International Journal of Distributed Sensor Networks*, 2014, 2014.

[38] AM Khan, YK Lee, and SY Lee. Accelerometer's position free human activity recognition using a hierarchical recognition model. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pages 296–301. IEEE, 2010.

[39] Abby C King, Eric B Hekler, Lauren A Grieco, Sandra J Winter, Jylana L Sheats, Matthew P Buman, Banny Banerjee, Thomas N Robinson, and Jesse Cirimele. Harnessing different motivational frames via mobile phones to promote daily physical activity and reduce sedentary behavior in aging adults. *PloS one*, 8(4):e62613, 2013.

[40] Kai Kunze and Paul Lukowicz. Sensor placement variations in wearable activity recognition. *IEEE Pervasive Computing*, 13(4):32–41, 2014.

[41] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[42] Óscar D Lara and Miguel A Labrador. A mobile platform for real-time human activity recognition. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 667–671. IEEE, 2012.

[43] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[44] Eric Larson, Jon Froehlich, Tim Campbell, Conor Haggerty, Les Atlas, James Fogarty, and Shwetak N Patel. Disaggregated water sensing from a single, pressure-based sensor: An extended analysis of hydrosense using staged experiments. *Pervasive and Mobile Computing*, 8(1):82–102, 2012.

[45] Matthew L Lee and Anind K Dey. Sensor-based observations of daily living for aging in place. *Personal and Ubiquitous Computing*, 19(1):27–43, 2015.

[46] Ming Li, Viktor Rozgica, Gautam Thatte, Sangwon Lee, Adar Emken, Murali Annavaram, Urbashi Mitra, Donna Spruijt-Metz, and Shrikanth Narayanan. Multimodal physical activity recognition by fusing temporal and cepstral information. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4):369–380, 2010.

[47] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.

[48] Zachary C Lipton, David C Kale, and Randall Wetzel. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare*, 2016.

[49] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[50] Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360. ACM, 2012.

[51] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84. ACM, 2010.

[52] Sri Harish Mallidi and Hynek Hermansky. Novel neural network based fusion for multi-stream asr. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5680–5684. IEEE, 2016.

[53] Jani Mantyjarvi, Johan Himberg, and Tapio Seppanen. Recognizing human motion with multiple acceleration sensors. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 2, pages 747–752. IEEE, 2001.

[54] Gloria Mark, Shamsi T. Iqbal, Mary Czerwinski, Paul Johns, and Akane Sano. Neurotics can't focus: An in situ study of online multitasking in the workplace. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1739–1744, New York, NY, USA, 2016. ACM.

[55] MJ Mathie, ACF Coster, NH Lovell, and BG Celler. Detection of daily physical activities using a triaxial accelerometer. *Medical and Biological Engineering and Computing*, 41(3):296–301, 2003.

[56] Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4–pp. IEEE, 2006.

[57] Daniel J McDuff, Javier Hernandez, Sarah Gontarek, and Rosalind W Picard. Cogcam: Contact-free measurement of cognitive stress during computer tasks with a digital camera. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4000–4004. ACM, 2016.

[58] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. My phone and me: Understanding people's receptivity to mobile notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1021–1032, New York, NY, USA, 2016. ACM.

[59] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

[60] Emiliano Miluzzo, Michela Papandrea, Nicholas D Lane, Hong Lu, and Andrew T Campbell. Pocket, bag, hand, etc.-automatically detecting phone context through discovery. *Proc. PhoneSense 2010*, pages 21–25, 2010.

[61] Inbal Nahum-Shani, Shawna N Smith, Ambuj Tewari, Katie Witkiewitz, Linda M Collins, Bonnie Spring, and S Murphy. Just in time adaptive interventions (jitais): An organizing framework for ongoing health behavior support. *Methodology Center technical report*, (14-126), 2014.

[62] Annamalai Natarajan, Gustavo Angarita, Edward Gaiser, Robert Malison, Deepak Ganesan, and Benjamin M Marlin. Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ecg. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 875–885. ACM, 2016.

[63] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[64] Fco Javier Ordóñez, Paula de Toledo, and Araceli Sanchis. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors*, 13(5):5460–5477, 2013.

[65] Juha Parkka, Miikka Ermes, Panu Korpipaa, Jani Mantyjarvi, Johannes Peltola, and Ilkka Korhonen. Activity classification using realistic data from wearable sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):119–128, 2006.

[66] Shwetak N Patel, Julie A Kientz, Gillian R Hayes, Sooraj Bhat, and Gregory D Abowd. Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *UbiComp 2006: Ubiquitous Computing*, pages 123–140. Springer, 2006.

[67] Vesa Peltonen, Juha Tuomi, Anssi Klapuri, Jyri Huopaniemi, and Timo Sorsa. Computational auditory scene recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–1941. IEEE, 2002.

[68] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

[69] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature selection and activity recognition from wearable sensors. In *International Symposium on Ubiquitious Computing Systems*, pages 516–527. Springer, 2006.

[70] D.M. Pober, J. Staudenmayer, C. Raphael, and P.S. Freedson. Development of novel techniques to classify physical activity mode using accelerometers. *Medicine and science in sports and exercise*, 38(9):1626–34, September 2006.

[71] Tauhidur Rahman, Alexander Travis Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. Bodybeat: a mobile system for sensing non-speech body sounds. In *MobiSys*, volume 14, pages 2–13, 2014.

[72] Tauhidur Rahman, Mi Zhang, Stephen Voida, and Tanzeem Choudhury. Towards accurate non-intrusive recollection of stress levels using mobile sensing and contextual recall. In *International Conference on Pervasive Computing Technologies for Healthcare*, 2014.

[73] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, 6(2):1–27, February 2010.

[74] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 108–109. IEEE, 2012.

[75] Mattia Rossi, Sebastian Feese, Oliver Amft, Nils Braune, Sandro Martis, and G Troster. Ambientsense: A real-time ambient sound recognition system for smartphones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 230–235. IEEE, 2013.

[76] Mirco Rossi, Gerhard Troster, and Oliver Amft. Recognizing daily life context using web-collected audio data. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 25–28. IEEE, 2012.

[77] Jason Ryder, Brent Longstaff, Sasank Reddy, and Deborah Estrin. Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 927–931. IEEE, 2009.

[78] Christie Napa Scollon, Chu-Kim Prieto, and Ed Diener. Experience sampling: promises and pitfalls, strength and weaknesses. In *Assessing well-being*, pages 157–180. Springer, 2009.

[79] Julia Seiter, Oliver Amft, Mirco Rossi, and Gerhard Tröster. Discovery of activity composites using topic models: An analysis of unsupervised methods. *Pervasive and Mobile Computing*, 15:215–227, 2014.

[80] Kelly Servick. Mind the phone. *Science*, 350(6266):1306–1309, 2015.

[81] Saul Shiffman, Arthur A Stone, and Michael R Hufford. Ecological momentary assessment. *Annu. Rev. Clin. Psychol.*, 4:1–32, 2008.

[82] Muhammad Shoaib, Stephan Bosch, Hans Scholten, Paul JM Havinga, and Ozlem Durmaz Incel. Towards detection of bad habits by fusing smartphone and smartwatch sensors. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 591–596. IEEE, 2015.

[83] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[84] John Staudenmayer, David Pober, Scott Crouter, David Bassett, and Patty Freedson. An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer. *Journal of Applied Physiology*, 107(4):1300–1307, 2009.

[85] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015.

[86] Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 37–40. IEEE, 2007.

[87] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, volume 3001, pages 158–175. Springer, 2004.

[88] Emmanuel Munguia Tapia, Stephen S Intille, Louis Lopez, and Kent Larson. The design of a portable kit of wireless sensors for naturalistic data collection. In *Pervasive Computing*, pages 117–134. Springer, 2006.

[89] Edison Thomaz, Irfan Essa, and Gregory D Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1029–1040. ACM, 2015.

[90] Edison Thomaz, Cheng Zhang, Irfan Essa, and Gregory D Abowd. Inferring meal eating activities in real world settings from ambient sounds: A feasibility study. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 427–431. ACM, 2015.

[91] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. Recognizing detailed human context in-the-wild from smartphones and smartwatches. *IEEE Pervasive Computing*, 16(4), 2017.

[92] Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, and Nadir Weibel. Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018)*. ACM, 2018.

[93] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. Behavioral context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *Under review for Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4), 2017.

[94] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. The mobiact dataset: Recognition of activities of daily living using smartphones. In *ICT4AgeingWell*, pages 143–151, 2016.

[95] Shankar Vembu, Alexander Vergara, Mehmet K Muezzinoglu, and Ramón Huerta. On time series features and kernels for machine olfaction. *Sensors and Actuators B: Chemical*, 174:535–546, 2012.

[96] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 3–14. ACM, 2014.

[97] Jamie A Ward, Paul Lukowicz, and Hans W Gellersen. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):6, 2011.

[98] Evan Welbourne and Emmanuel Munguia Tapia. Crowdsignals: a call to crowdfund the community's largest mobile dataset. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 873–877. ACM, 2014.

[99] Daniel H Wilson and Chris Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *Pervasive computing*, pages 62–79. Springer, 2005.

[100] Hao Yan and Ted Selker. Context-aware office assistant. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 276–279. ACM, 2000.

[101] Yi-Hsuan Yang and Yuan-Ching Teng. Quantitative study of music listening behavior in a smartphone context. *ACM Trans. Interact. Intell. Syst.*, 5(3):14:1–14:30, September 2015.

[102] Koji Yatani and Khai N Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 341–350. ACM, 2012.

[103] Huiru Zheng, Haiying Wang, and Norman Black. Human activity detection in smart home environment with self-adaptive neural networks. In *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pages 1505–1510. IEEE, 2008.