Newcastle University

School of Computing Science

# Online Learning of Personalised Human Activity Recognition Models from User-Provided Annotations

*Tudor-Alin Miu*

*Submitted for the degree of Doctor of Philosophy in the School of Computing Science, Newcastle University*

January 2017

# Abstract

In Human Activity Recognition (HAR), supervised and semi-supervised training are important tools for devising parametric activity models. For the best modelling performance, large amounts of annotated personalised sample data are typically required.

Annotating often represents the bottleneck in the overall modelling process as it usually involves retrospective analysis of experimental ground truth, like video footage. These approaches typically neglect that prospective users of HAR systems are themselves key sources of ground truth for their own activities.

This research therefore involves the users of HAR monitors in the annotation process. The process relies solely on users' short term memory and engages with them to parsimoniously provide annotations for their own activities as they unfold. Effects of user input are optimised by using Online Active Learning (OAL) to identify the most critical annotations which are expected to lead to highly optimal HAR model performance gains.

Personalised HAR models are trained from user-provided annotations as part of the evaluation, focusing mainly on objective model accuracy. The OAL approach is contrasted with Random Selection (RS) – a naive method which makes uninformed annotation requests. A range of simulation-based annotation scenarios demonstrate that using OAL brings benefits in terms of HAR model performance over RS. Additionally, a mobile application is implemented and deployed in a naturalistic context to collect annotations from a panel of human participants. The deployment is proof that the method can truly run in online mode and it also shows that considerable HAR model performance gains can be registered even under realistic conditions.

The findings from this research point to the conclusion that online learning from user-provided annotations is a valid solution to the problem of constructing personalised HAR models.

# Publications

Portions of the work within this thesis have been documented in the following publications:

Miu et al. [1]: T. Miu, T. Plötz, P. Missier, and D. Roggen, "On strategies for budget-based online annotation in human activity recognition" in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, (New York, NY, USA), 2014.

Miu et al. [2]: T.Miu, P. Missier and T. Plötz, "Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning" in *Proceedings of the 14th IEEE International Conference on Ubiquitous Computing and Communications*, IUCC 2015, (Liverpool, UK), 2015

# Dedication

*To my parents who have loved and supported me unconditionally.*

# Acknowledgements

First and foremost, I am grateful to Paolo Missier, my main PhD supervisor, not only for giving me the opportunity to pursue the PhD, but also for supporting me throughout this journey. Paolo has been nothing short of an inspiration and I learned tremendously from him. I can't believe it's been four years!

I would like to thank Paul Watson for his general guidance and for always ensuring all the key resources were in place. I have worked extensively with Thomas Plötz and I grateful for his collaboration and valuable advice. I would also like to thank Daniel Roggen for working with me during his stay at Newcastle University. Casim Ladha has provided me with technical support for the sensors and that was essential, so thank you!

I had fascinating discussions (not always technical!) with my office mates Rawaa Qasha, Zhenyu Wen, Hugo Firth and Faris Llwah. They were terrific to have around and their feedback on my work proved to be very useful.

I was extremely fortunate to meet Anya Batalina, Ciprian Spătar and Adil Ibadula here in Newcastle. They are wonderful friends and I am grateful for their relentless positivity and encouragement which made it indirectly in this thesis.

The list could go on and on, so, in general, I would like to thank the crowd in the School of Computing Science for being so welcoming and engaging.

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## Contents

# Introduction

One of the key promises of Weiser's vision of pervasive computing has been the prospect of disappearing technologies that "*weave themselves into the fabric of everyday life until they are indistinguishable from it*" [3]. Tremendous progress has already been made towards making this vision a reality where smart environments, living labs, and especially mobile computing now constitute the central paradigm of this third generation of computing [4]. As an enabling technology, automatic inference of the context and especially of the activities humans are engaged in (Human Activity Recognition – HAR) plays a central role in a large plurality of pervasive and mobile computing applications.

# Health – A Case for Physical Activity Recognition

HAR is primarily an observational tool suitable for monitoring lifestyle. By identifying certain patterns, such as damaging behaviours, one has the possibility of changing them by prioritising certain lifestyle decisions. A common damaging behaviour is general physical inactivity which has been linked with serious health conditions including cardiovascular disease [5, 6], diabetes [7–9], colon cancer [10], obesity [11] or depression [12]. According to the World Health Organization, the incidence of such cases could be reduced if physical inactivity was less pervasive [13].

The prevalence of these medical conditions has far-reaching consequences on how society functions and sustains itself. For example, according to Hex et al. [14], in 2010/2011 the economic cost incurred by the UK's National Health Service (NHS) for treating diabetes stood at £9.8bn for direct treatment and an additional £13.9bn for treating complications. The cost is projected to rise by 2035/2036 to £16.9bn for direct treatment and £22.9bn for treating complications. In fact, diabetes accounts for approximately 10% of the total health expenditure for 2010/2011 and the share is expected to rise to 17% in 2035/2036, according to the same source.

Astronomical health care costs are not singular to diabetes. In the UK, the healthcare costs for major conditions partly caused by physical inactivity are still in the order of billions of pounds: cardiovascular disease cost £8.6bn in in 2009 [15], while cancer

cost £5.81bn in 2010/2011[1]. It is estimated that between 1.5% to 3% of the healthcare costs in developed countries is directly accounted for physical inactivity [16]. Physical inactivity is also correlated with obesity [17], smoking [18, 19] or hazardous alcohol consumption [20], each of which adds costs of the order of billions of pounds to NHS expenditure [21]. With the UK GDP at £1800bn[2], the cost of addressing diabetes and other serious illnesses rises to single-figure percentages of the GDP.

According to Lee et al. [22], at a worldwide level, physical inactivity is estimated to cause 6% of the cases of coronary heart disease, 7% type 2 diabetes, 10% breast cancer and 10% colon cancer. In the UK, according to the same source, the effects are more pronounced: 10.5% for coronary heart disease, 13% for type 2 diabetes, 17.9% for breast cancer and 18.7% for colon cancer. With physical inactivity being linked to serious medical conditions in double-figure percentages of all cases, it is clear that society should strive to combat physical inactivity.

### *Interventions*

People who are sedentary for long periods of time, such as office workers, face additional health risks. Sitting down for extended periods of time has been identified as a health risk promoter, even for people who are otherwise physically active [23, 24]. Prolonged sitting can affect not only by causing physical discomfort, such as, for example, muscoskeletal pain [25], but can also have serious health repercussions, such as increased cardiovascular risk [26] or coronary heart disease [27].

As a whole, the negative societal impact of these medical conditions, which is partly caused by physical inactivity, is set to increase and cannot be ignored. Therefore, at this point in time, it is justified to allocate research resources to combat these medical conditions, not only directly through treatment, but also indirectly, by addressing the causes, such as reducing physical inactivity.

[1] `https://www.gov.uk/government/publications/2003-04-to-2010-11-programme-budgeting-data` - Accessed 06.05.2015

[2] `http://www.imf.org/external/pubs/ft/weo/2015/01/weodata/weorept.aspx?pr.x=83&pr.y=16&sy=2015&ey=2015&scsm=1&ssd=1&sort=country&ds=.&br=1&c=193%2C273%2C223%2C156%2C924%2C922%2C132%2C184%2C134%2C534%2C536%2C136%2C158%2C112%2C111%2C542&s=NGDPD%2CPPPGDP&grp=0&a=` - IMF estimates; Accessed 27.05.2015

Research efforts include interventionary actions at the workplace to interrupt long sitting times and promote other less sedentary behaviour such as standing. These interventions were shown to be effective not only in changing behaviour in the short term [28, 29], but also in reducing muscoskeletal pain symptoms [30–32]. It has been suggested by Dunstan et al. [33] that short and regular interventions which break sitting times with walking may reduce cardiovascular risk.

Office workers are sedentary a great deal of time, so the office is an ideal environment to suppress physical inactivity and to promote healthier behaviours. Therefore, short breaks from sitting, either by standing or by performing some form of physical exercise, can have substantial effects in increasing physical comfort and maintaining good health. However, these interventions are naturally disruptive. While it is beneficial for office workers, for example, to regularly break their sitting times, this is not in their short-term interest and, in addition, the benefits are not immediate. Without instant gratification, it is questionable that office workers would voluntarily remember to comply with the intervention protocol.

Consequently, in order to ensure greater compliance, these interventions can be supported by appropriate technological means. We argue therefore that technology can be part of the solution. An enormous body of research, which has been dedicated to the general problem of inferring context, has shown that HAR technology and techniques can meet the needs for self-monitoring. For instance, it is now possible to measure one's expended energy [34–37] or register individual activities [37, 38]. In fact, multiple dimensions of wellbeing can be monitored in parallel, such as sleep patterns, social interaction and physical activity [39]. The user's tracked behaviour can be summarised in order to monitor progress and improve the intervention compliance. HAR can also help with tracking behaviours which are symptomatic to serious underlying conditions, such as, diabetes [40] or Alzheimer's Disease [41].

Insight into what is worth monitoring and how to do so can be gained from the adherents of the *Quantified Self* [42] community. These are individuals who track different aspects of their life including sleep, dieting or time management and some of whom are also inspired by the prospects of tracking physical activities. Choe et al. [43] show that physical exertion is one of the top tracked objectives for quantified selfers. The

reasons may lie with the desire to self-improve in different physically demanding activities, such as sports or athletics, but they may also lie with maintaining an active and healthy lifestyle and the benefits it brings.

HAR applications do not necessarily have to be strict monitors, but they can encourage certain behaviours. This was demonstrated by Consolvo et al. [44] who have deployed a system that displayed pleasant video feedback when the user attained certain physical exertion goals. The authors report that most of their user study participants thought the system would motivate physical exertion.

Once compliance is ensured and time is allocated to physical activity, then our goal is to use HAR and to maximise its effects. Performing certain physical activities, such as fitness ones, is a skill that can be improved with correct technique. Instead of personal trainers, which may be a limited and expensive resource, HAR can be used to automatically assess and provide feedback for such activities [45].

An important factor that led to the adoption and study of HAR by the research community is the large scale adoption of consumer-grade wearable technology. The wearables industry is developing to serve a considerable market of consumers. According to `uk.businessinsider.com` [46], 33 million units are going to be shipped by the end of 2015. The size of the wearables market is on an upward trend with sales expected to grow yearly by 35% until 2019, when a predicted 148 million units are going to be shipped annually.

## Personalising Wearable Devices

Given our societal need to reduce physical inactivity and the increasingly widespread use of wearable devices, HAR is becoming an enabling technological driver for personal interventions that can prevent or correct damaging behaviours. Activity recognition may be further improved if the wearable devices are *personalised*, i.e. adapted to the user's lifestyle or movement idiosyncrasies. This is explained in greater detail in Chapter 2.

The overarching concept we investigate in this thesis is *user feedback as a form of knowledge that can be used to personalise HAR models*. We make the hypothesis that

it is possible to construct fully personalised HAR models from gradual user feedback, by starting from either zero or relatively little prior knowledge.

A *HAR model* is a description of a set of a user's physical activities. The types of models we are concerned with in this thesis are *machine learning* models, as discussed in Chapter 2.4. Specifically, we focus on the user's *HAR model* which monitors the user's stream of physical activities. In this context, a user's *model* is an operational description of the user's physical activities which is used within a HAR monitoring system to estimate what activities are exerted by the user. The type of model of we investigate throughout this thesis are *classification models* – models that output estimates over a discrete set of labels. We use terms *model*, *classification model* and *classifier* interchangeably to designate the user's HAR model. In this respect, and as discussed in Section 2.2, a *personalised model* is a model which is constructed or adapted in such a way so as to better monitor the target user, without regard to a larger pool of separate users. When it comes to *training* personalised models, we apply (1) semi-supervised machine learning techniques (introduced in Chapter 2 and investigated in Chapters 3-6) to collect personalised training sets and (2) standard supervised techniques to infer the parameters of standard classification models (outlined in Chapter 2).

If this hypothesis is correct, then the user of the HAR system is in control of personalising her model. Continuous personalisation leads to improved activity monitoring relative to previous versions of the model. Because this technical improvement eventually translates into improved wellbeing to the user, the user might be motivated to participate in personalising her own HAR model. Additionally, as we demonstrate in this thesis, the user can personalise her own model without external assistance, such as a researcher or video footage of her activities. This not only frees researchers from the costly process of labelling another person's data, but also potentially allows user-led labelling at a much larger scale than would be possible with external supervision. Also, as we show in this thesis and also by means of existing research, in many situations, personalised models are more accurate than non-personalised models.

Personalising HAR models is a challenging research problem because user feedback is a valuable, but, at the same time, scarce resource. Exploiting it presents a problematic

trade-off because, on the one hand, the more user input is requested, the greater the personalisation, but, on the other hand, asking for feedback from a user is disruptive, so the user should not be interrupted excessively. The difficulty, therefore, lies in identifying the most critical inputs to request from the user and this creates two-fold complications: Firstly, the requests for feedback should be simple and timely enough for the user to answer, but, at the same time, critical enough to warrant interrupting the user. As shown in Chapter 2, we rely on the user's short-term memory to obtain personalising feedback and this entails severe time restrictions over what the user can reliably provide. Secondly, the frequency and volume of input requests, even if all are highly critical, should not cross user interruption boundaries.

In this thesis we demonstrate different solutions of obtaining feedback from a user and personalising her model in the context of several case studies. Encouraged by our positive results, we argue that our methods may be applicable to a wider range of HAR classes of applications.

## *HAR System Usage*

Previously we have argued about the benefits of personalising HAR models and proposed that personalisation could be achieved through user feedback. In this section we go into more detail. We first illustrate how general HAR systems work without personalisation. Afterwards, we propose a mechanism of personalisation and look at the consequences of our proposed mechanism of personalisation. To this end, we illustrate how we envisage the process of HAR model personalisation from user feedback and we exemplify it through a hypothetical scenario involving a fictitious character. We draw parallels between and contrast a generic non-personalisable HAR system and a personalisable one. We consider how personalisation affects the system usage and user interaction.

### System Usage – without Personalisation

Consider Anya – a typical office worker who is conscious about her health. Because her work involves long periods of physical inactivity, she uses a generic wearable physical activity tracker to monitor her levels of physical activity throughout the day. She relies

on accurate reports so that she can break long sitting times or compensate physical inactivity at the office with physical exercise after work. The monitor's functioning is completely passive. Fig. 1.1, which illustrates a schematic of its mechanics, shows that the user does not supply input – Anya merely benefits from passive monitoring.

The device uses a "one size fits all" model which gives good results for a limited set of activities and when averaged across a large population. However, the model may not be very accurate for Anya in particular because some of her favourite physical activities and her general movement idiosyncrasies are not taken into account.

Therefore, without personalisation, the model does not account for user-specific characteristics, such as idiosyncrasies or lifestyle specifics. A generic system that supports personalisation is illustrated in Fig. 1.2. Key user aspects can be captured via model personalisation and this is expected to improve the recognition accuracy.

### System Usage – with Personalisation

In this thesis we propose that HAR model personalisation is achieved through active user participation. Specifically, we envisage a HAR monitor that parsimoniously interacts with the user by requesting feedback which is then used to further personalise the user's model. The type of personalising feedback we choose to obtain from the user are *annotations*. This format, discussed in detail in Chapter 2, assigns an activity label to recorded signal data and can be used directly to personalise HAR models. We use annotations throughout the rest of the thesis to train personalised models under different scenarios and to evaluate their recognition performance.

Because the only types of HAR models investigated here are classification models, intuitively, we use recognition success rate as a measure of HAR monitoring performance. We use other similar terms, such as *model performance*, *model improvement* *classification accuracy*, interchangeably. The objective measure behind recognition success rate is F-Score and it is formally defined in Section 3.3.2[3], i.e. what insight can an expert gain from examining the parameters of such a model.

[3]Models are therefore distinguished between themselves on a *prediction*-based criterion (i.e. how successful the recognition of activities is). An alternative method of comparing models, albeit more subjective, is based on *inference* – James et al. [47].

Such a personalising system, shown in Fig. 1.3, can automatically diagnose when feedback is most needed, but can also take into account the user's propensity towards interruption so that the provision of annotations is not overly taxing on the user. Now that the system supports personalisation, this brings about a new pattern of usage of the system, as illustrated in Fig. 1.4. With the addition of the annotation step, we reroute the natural data flow of a generic HAR application to form a closed circuit so that it can support progressive model personalisation.

Continuing our example, Anya decides she is willing to respond only to a very limited number of annotation requests and, for instance, preferably evenly spread out throughout the day. We discuss in Chapter 2 the user's propensity to provide annotations on demand. We accept that the user's tolerance towards annotation interruptions is a finite resource, so annotation strategies should account for limited compliance. Therefore, we introduce the concept of a *budget of annotations* that models the sparsity of input which the user could provide. The effects of budget-constrained annotation methods are evaluated in Chapters 3 and 6.

We further propose that annotations are produced *online*, i.e. while the system is being used and the activities are being monitored. This is a crucial distinction to other proposals which focus on *offline* annotations, i.e. the annotations are attributed upon retrospective inspection of the data and other sources of information such as video footage. We argue against offline approaches in Chapter 2 because they require extensive supporting infrastructures and support from other researchers or expert annotators. Instead, by focusing on online annotations, users can easily resort to their short-term memory to parsimoniously annotate their most recent activities. As explained in Chapter 2, offline approaches are not compatible with a very short term user memory.

In our example, suppose that Anya is on a break from working at her desk and she starts performing a series of regular torso movements. This is a new exercise that has been recently suggested to her and she hasn't done it before, or has done it only a few times, and this would be insufficient for accurate monitoring. The system runs a fully automated procedure (one that does not require user feedback) to diagnose its performance at classifying the activity. Because the system has no or limited data

on this activity, it performs relatively poorly at recognising the activity, so it decides that an annotation for this activity is important for improving Anya's HAR model. As soon as the system detects the end of this problematic activity, it issues an annotation request to Anya for that activity. Anya does not have to recall into the distant past and does not have to fall back to memory aids such as video footage. Instead, she can easily remember what she was doing just seconds before the annotation request.

Going further with the example, after Anya is done with her torso exercise and provides an annotation, she heads down to the common room to make herself some tea. The system now detects she is walking and, because it already has several annotations for this more common activity, it recognises the activity with very little uncertainty. Consequently, the system decides not to ask Anya to annotate the activity, so she proceeds uninterrupted.

Anya turns on the kettle and, while the water comes to boil, she performs again the torso exercises. However, she had previously set up her HAR monitor to issue annotation requests only very infrequently while she is at work. The HAR model still diagnoses itself as not very adequate at classifying this exercise, but it suppresses its current annotation request so that Anya does not become annoyed.

Considering the mechanics of the system, because it is now interactive, the pattern of usage changes partially. In this case, the user is expected to engage occasionally to provide annotations. Anya still relies on fully automated, passive and, for the most part, non-intrusive activity recognition, but now the system occasionally asks her to provide feedback which aids model personalisation. The mechanics of how an annotation is deemed useful by the system – using active learning – is introduced in Chapter 2 and is a core concept which is evaluated in Chapters 4, 5 and 6. However, as we discuss in Chapter 2, previously studied offline active learning methods are inapplicable to our scenario because, effectively, they are at odds with our assumption of limited user memory recall. Instead, we apply an online method, called Online Active Learning (investigated by Sculley [48]), which is able to identify critical annotations even when operating over an extremely limited horizon of choices. In our case, the horizon is limited to a single potential annotation at any one time which is the most recently finished activity.

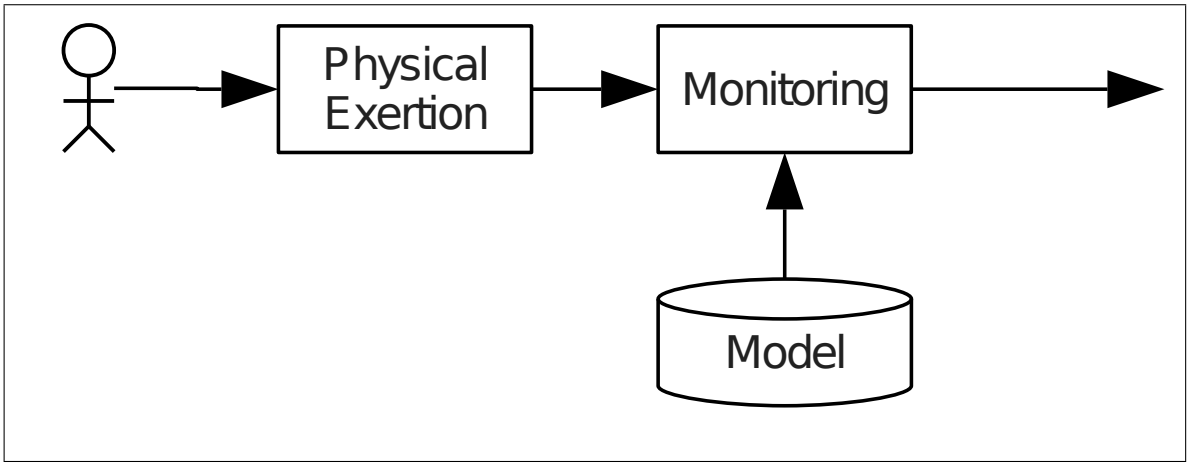



Figure 1.1: Typical HAR Monitoring.

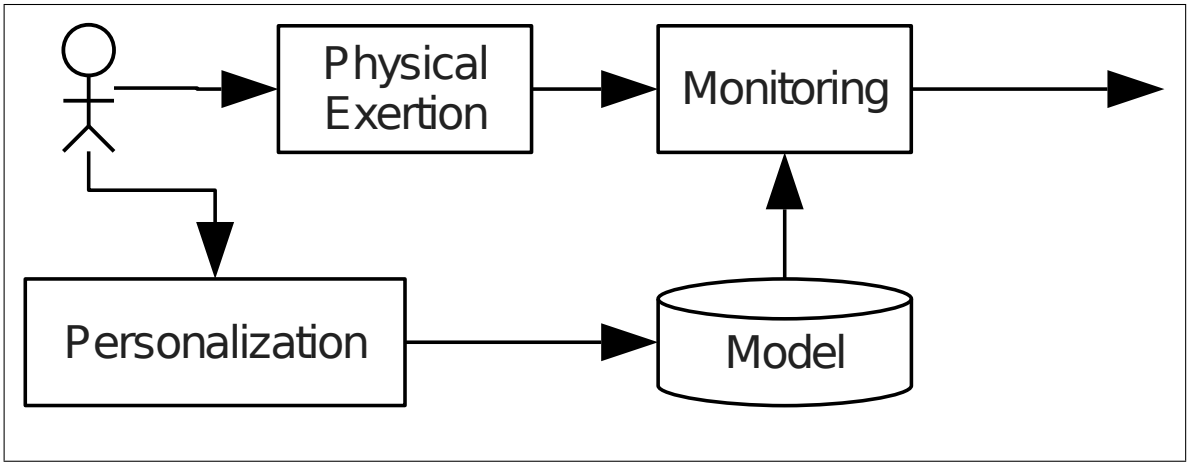



Figure 1.2: HAR Monitoring with Model Personalisation.

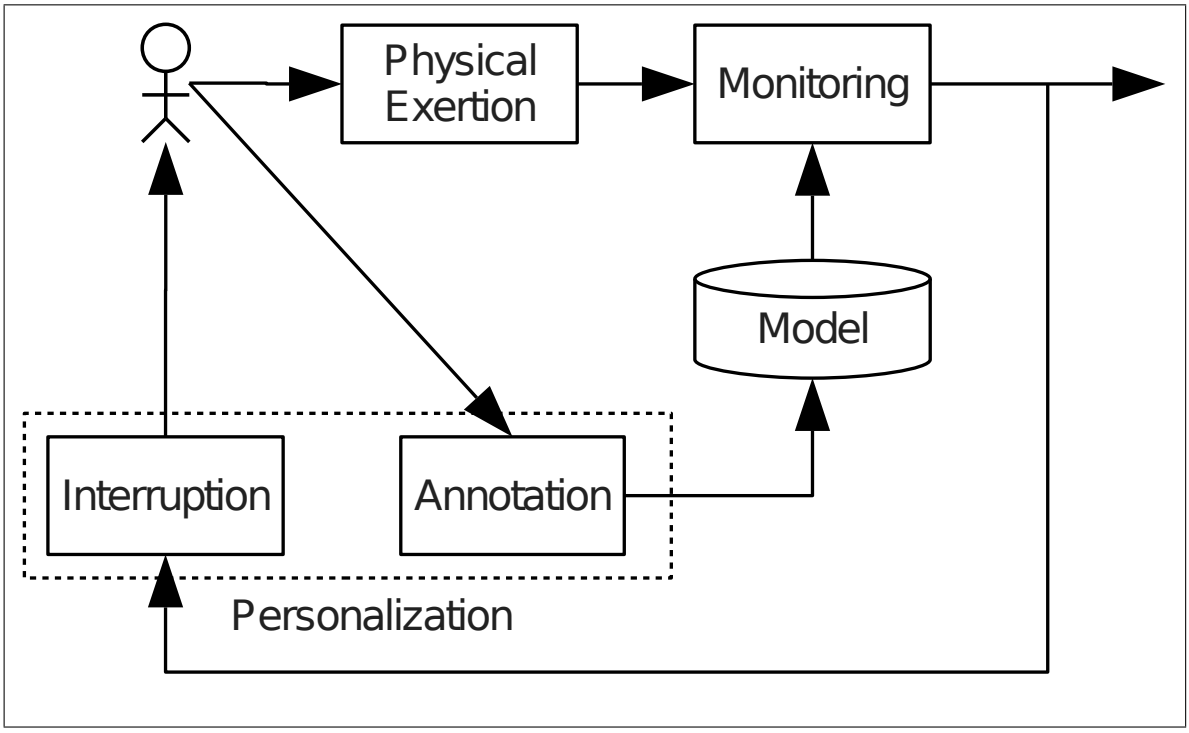



Figure 1.3: HAR Monitoring with User-Provided Annotations for Model Personalisation.

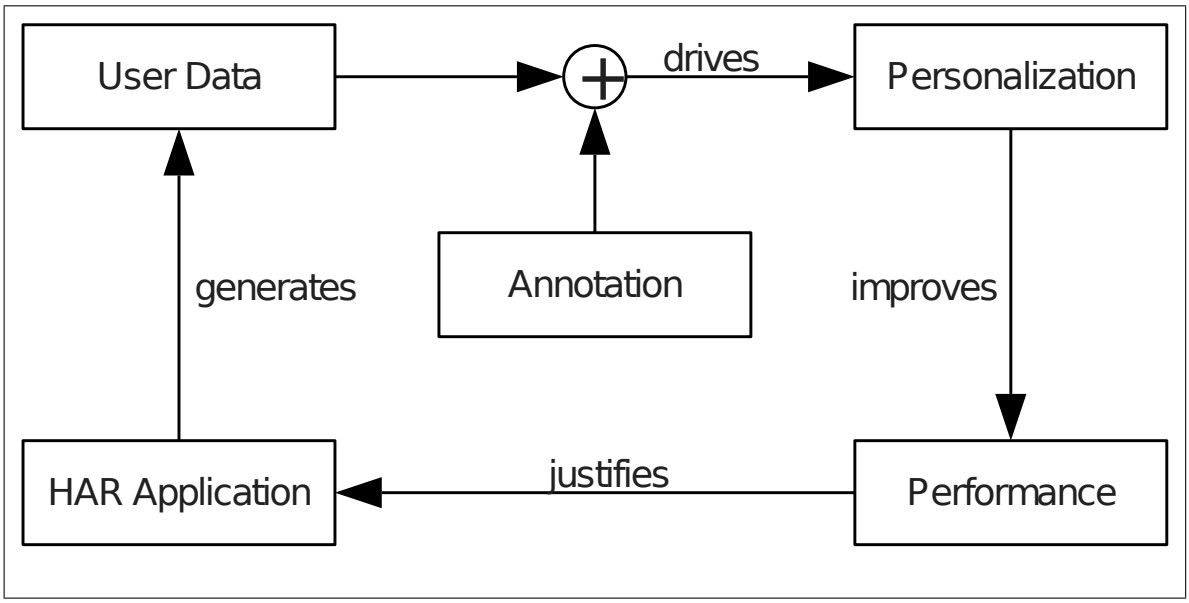



Figure 1.4: Personalisation Loop.

## Contributions

We claim that HAR models can be personalised from user-provided online annotations and throughout this thesis we use experiment-based approaches to support our claim. As underlined previously, online annotations can be more easily supported than offline annotations because online annotations can be provided by the users themselves while resorting only to their short-term memory.

In our scenarios we start either from zero knowledge, i.e. the system does not know anything about the user and it does not have any of the user's annotations, or from a relatively small corpus of annotations. With each annotation the user provides, we *bootstrap* the model – we update or reconstruct the model so that it accounts for the newly updated corpus of annotations.

We use a core concept, *Learning from Online Annotations*, to designate the process of model induction from the annotations the user parsimoniously provides while using the system. We introduce several methods and a complete system implementation and we assess different aspects of our core concept, as illustrated in Fig. 1.5.

The principal assumption of our work is a technological evolution of physical activity trackers. Even though, currently, activity trackers are usually passive sensing devices, we propose to augment them with user interaction capabilities, so that the they can engage with the user to learn from personalising feedback.

The contributions in this thesis are as follows:

Firstly, we model a user's limited and varying propensity of providing annotations by introducing the concept of a *Budget of Annotations*. We issue annotation requests strictly according to the user's predisposition towards interruptions, but without accounting for the importance of each potential annotation towards model improvement. We evaluate the impacts of the user's predisposition on the recognition performance of the user's personal HAR model in Chapter 3.

Secondly, in Chapter 4 we exploit the observation that not all annotations increase model accuracy by equal amounts. Here we shift focus from the user's predisposition entirely to the importance potential annotations have for model improvement.

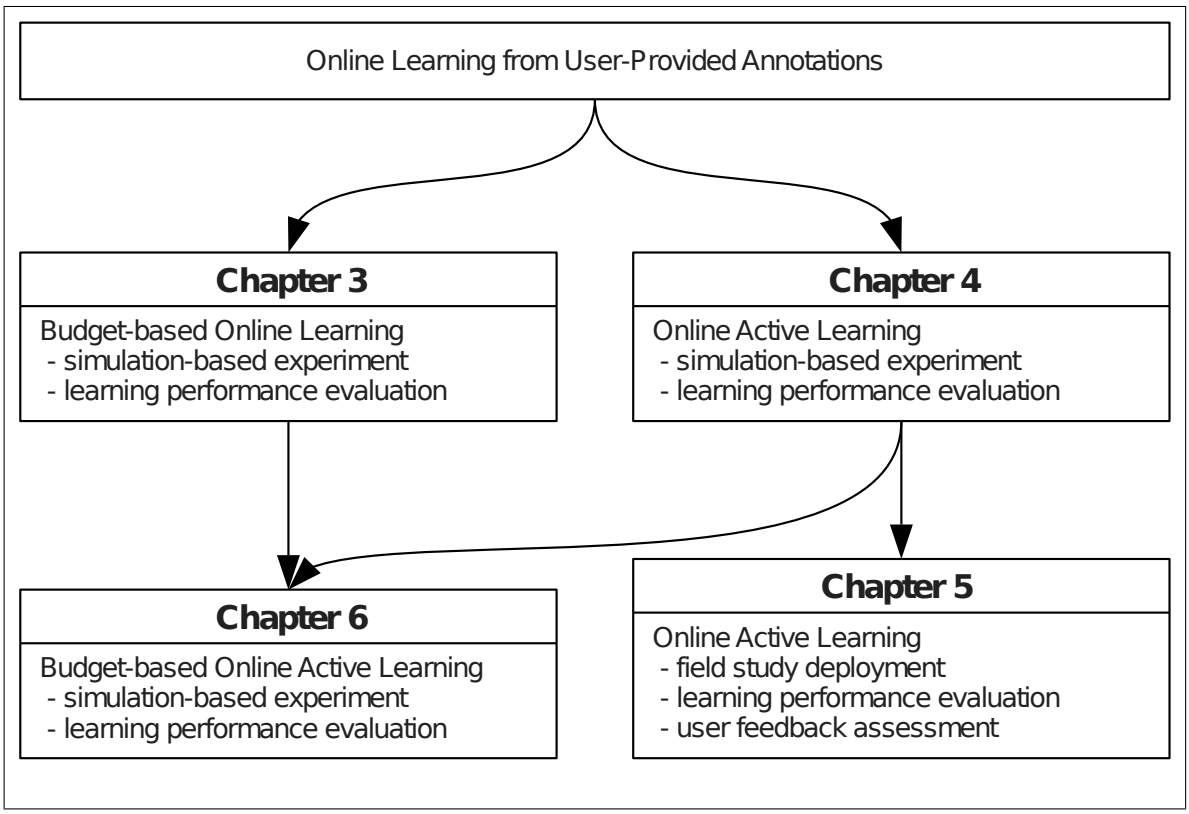



Figure 1.5: Roadmap of Contributions.

Because some annotations are more beneficial towards model accuracy improvement, we propose to use *Online Active Learning*, a method of identifying highly promising annotations (from a model improvement point of view) from a stream of activities. We obtain online annotations which clearly distinguishes our work from other Active Learning methods which typically identify offline annotations, as explained in Chapter 2. Our results demonstrate that performance gains on recognition performance can be obtained by using our Online Active Learning approach to accumulating annotations. Encouraged by these results, in Chapter 5, we provide a complete *system implementation* of our Online Active Learning method for HAR and we deploy the system within the setting of a naturalistic user study. We propose a HAR framework that incorporates a suite of methods supporting Online Active Learning and implement a mobile application for online activity recognition using body-worn wireless sensors. The results are threefold:

1. the deployment of our system in a user-centred field study shows that modern mobile platforms could support our concept implementation and, so, it serves as

proof of concept for our Online Active Learning method;

2. post-experiment analysis shows that models improve their recognition accuracy as they are increasingly personalised. Therefore Online Active Learning method is robust enough to operate under realistic conditions, not only under simulated conditions as in Chapter 4;

3. the compilation of subjective user feedback on interacting with the system reveals that Online Active Learning is disruptive and taxing on user tolerance, but not necessarily excessively, so it can arguably be adopted in moderation to drive model personalisation.

Finally, in Chapter 6, we propose a hybrid criterion of requesting input that combines the Online Active Learning method from Chapters 4 and 5 with the Budget-based online learning method from Chapter 3. Our results show that the seemingly competing Online Active Learning and Budget-based online learning methods can actually be blended to yield, on the one hand, active learning-specific performance gains while, on the other hand, still closely adhering to the limitations of an imposed budget of annotations.

Overall, as we show in Fig. 1.6, we are, roughly speaking, exploring a two-dimensional space of parameters for our annotation method of obtaining online annotations and evaluating the effects on HAR model performance.

## Thesis Organisation

The rest of the thesis is structured as follows. In Chapter 2 we provide an overview of the state of the art in HAR model personalisation and related directions of research. In Chapter 3 we present an exploration of the effects of a budget-based online learning strategy that models a user's tolerance towards interruption. After that, we temporarily depart from budget constraints and instead focus on Online Active Learning as a method to identify important annotations for model improvement. In Chapter 4 we provide simulation-based experiments to evaluate Online Active Learning, while in Chapter 5 we test the feasibility of the method on a live user study. In Chapter 6 we

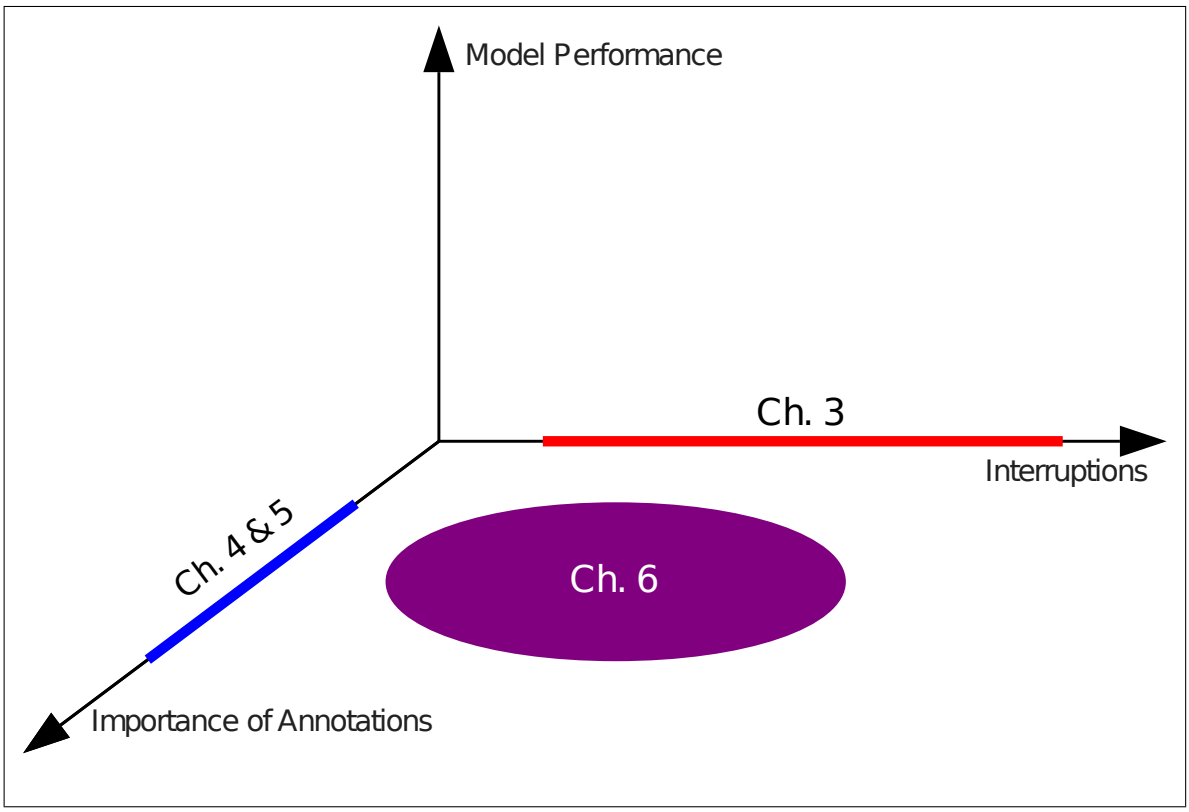



Figure 1.6: Parameter Space of the Annotation Method.

unify the budget idea from Chapter 3 and the Online Active Learning from Chapters 4 and 5. Finally, in Chapter 7 we reflect upon our contributions and place them in the context of future exploration.

# 2

# Foundations

## Contents

# Introduction

In the previous chapter we presented the motivation behind our research and a general overview of the significance of the contributions in this thesis. In this chapter we provide the research background for the rest of the thesis.

Because constructing personalised HAR models through user input only and without external supervision is a complex multi-stage problem, a series of inter-related design choices have to be made. In this chapter, for all design decisions we make in future chapters, we consider relevant related and background research.

## *Definition of an Annotation*

For a continuous timeseries of sensor readings which capture movement about physical activities, we define an *annotation as a labelled contiguous subsequence of sensor data which is representative for the underlying activity.* An annotation therefore consists, in part, of a *start timestamp* and an *end timestamp* which designate the first and, respectively, the last readings of a *segment* – the contiguous sub-timeseries that corresponds to the underlying activity. In addition, an annotation carries a segment *label* – a name of the respective activity class.

The annotations are collected into a *training set* which is then used to construct a HAR model. Further details concerning how this is achieved are presented in Section 2.4.4.

## *Our Contributions in Context*

Supervised learning is a dominant approach for HAR applications. However, obtaining reliable and sufficient ground truth annotations for training data is challenging, largely due to practical as well as ethical reasons, especially in mobile ubiquitous computing settings.

We involve the user in the annotation process and, like Intille et al. [49], we propose that users parsimoniously provide annotations for their own activities. We therefore remove expert annotators and other external sources of ground truth, such as video

footage, from the annotation process and, pass the responsibility of annotation to the prospective users of the HAR system themselves. The annotations are then used to construct fully personalised HAR models for each user in turn. This style of annotation provision and others are explicated in Section 2.2. As discussed in Chapter 1, even though user engagement entails disruption, users would arguably be motivated to interact with the system and provide annotations because this would lead to improved monitoring accuracy. User disruptions, their effects and mechanisms to alleviate annoyance are discussed in Section 2.3.

In this thesis we are concerned only with wearable HAR systems. By definition, these are attached and in close proximity to the user and, so, they can be carried across different environments. Not only do they have to provide effective monitoring by design, but the mechanism for collecting annotations should also function regardless of where the user may be. Since not all contexts will be instrumented, external sources of ground truth such as video footage are not always available and so they cannot be relied upon.

A key source of ground truth for annotations, therefore, is the user's *short-term memory*. We propose an interactive HAR system which makes use of this recent human memory in order to obtain annotations for the user's own activities. The activities at which annotation requests are aimed need to have occurred very recently; otherwise, as Eisen et al. [50] show, human memory recall deteriorates with the passage of time. In order to simplify the annotation process and to increase the robustness in user feedback, we only target the last activity the user performed. In Section 2.2.2 we use existing research to show that such timely feedback can accurately represent reality. In fact, with the benefit of hindsight, according to the results in Chapter 5, the event is so recent, that users have no problems remembering.

In Section 2.5.3 we use supporting literature to show that not all annotations are equally beneficial in terms of accuracy. Before prompting the user to provide an annotation, it is possible to estimate its potential performance gains using a learning methodology called Active Learning [51]. Learning methodologies in general, including active learning, are discussed in 2.5.

In order to exploit the user's short-term memory, which is inherently finite, and still use

Active Learning, one needs to compromise on a technical level. For example, a widely used class of active learning techniques are *pool-based* or *offline*. These require, at any one time, a relatively large corpus of potential annotations from which to choose the one which is expected to maximise the gain in model accuracy. However, any sizeable sequence of activities would not be robustly recalled by the user because it would be beyond the user's limited short-term memory. To circumvent this problem, we adopt a *stream-based* or *online* active learning technique developed by Sculley [48]. Instead of operating over a large corpus of potential annotations, this Online Active Learning approach considers only one activity at a time – the most recent one – and decides for each activity in turn whether to annotate it or not.

An interactive HAR monitor that supports collecting user-provided annotations and constructing personalised models is an entire ecosystem of automated procedures and algorithms that ultimately support its core machine learning-related functionality. Consideration to the characteristics of auxiliary functions and the technological implications of HAR systems is given in Section 2.6.

### *Chapter Outline*

Having established our contributions and the necessary context for them in Section 2.1.2, we proceed to outline the rest of the chapter:

- **Section 2.2 Obtaining Annotations** discusses different types of annotations in human activity recognition. We focus on how the acquisition mechanism of annotations affects not only the quality of the annotations, but also the interaction mechanism with the user
- In **Section 2.3 The User's Perspective** we outline key ideas in how users perceive interruptions. Additionally, in order to exploit the user's emotional variations towards interruptions, we draw inspiration from existing research on the notion of a *budget* of annotation – treating interruptions as a finite resource.
- In **Section 2.4 Machine Learning** we discuss general mechanisms of creating HAR models from annotated data. We review a pipeline of data processing and

model building which is routinely used in HAR research and which we also use in this thesis.

- **Section 2.5 Learning Methodologies for Model Learning** deals with more advanced Machine Learning concepts. While the previous section (2.4) deals with supervised model building methods, we now investigate unsupervised and semi-supervised methods. A semi-supervised method of particular importance to our thesis is Active Learning. In this section we describe the model accuracy benefits that Active Learning can bring and we also discuss how to adapt Active Learning to our proposed annotation mechanism.

- Finally, **Section 2.6 Machine Learning in HAR Applications** gives a practical perspective of applying machine learning for HAR and techniques used in this thesis.

## Obtaining Annotations

Personalising activity models, i.e. fine-tuning a model to the user being monitored, has been shown to lead to improved recognition accuracy over non-personalised models, for example by Lane et al. [52]. Personalisation, and therefore model improvement, can be obtained directly from gradually accumulating personalised annotations about a user, as Rebetez et al. [53] show, or by leveraging existing corpora of non-personalised data that can supplement the personalised annotations, like, for example, Cook et al. [54] or Stikic et al. [55]. Obtaining annotations is a critical step towards model improvement and different methodologies of collecting annotations exist.

### *Retrospective Annotations*

In HAR research, typically, while movement data is readily collected through sensors, the participants are observed by a researcher who annotates the data as the user's activities are performed, like Lester et al. [56] or Morris et al. [57], or by examining retrospective video footage of the participants, like Chavarriaga et al. [38] or Pham and Olivier [58].

An example tool for annotating used in HAR is ELAN [59]. It allows expert annotators to synchronise the video ground truth with sensor data and then to annotate the sensor data: they inspect the ground truth, establish temporal segment boundaries and declare the label for identified segments.

Retrospectively annotating activity data is suitable for one-off research investigations, such as collecting a dataset of annotations for offline analysis. However, in more realistic settings, this approach presents several limitations:

- Because of the involvement of an expert annotator, retrospective annotation does not scale well to increasing user bases because this requires increasing the team of annotators proportionally. Bagaveyev and Cook [60] and Lasecki et al. [61] have suggested crowdsourcing the annotation task. While this approach removes the bottleneck in human annotation, extensive ground truth in the form of video footage still needs to be collected, which can still limit the context of the annotations.

- The possibility to annotate is limited to the environment where the ground truth collection infrastructure is present. This greatly reduces what can be annotated and annotations may not be representative of a user's entire lifestyle. Portable cameras are a possibility (for instance, Maekawa et al. [62] suggest computer vision techniques to assist automated activity recognition), but, examining video footage is arguably tedious and time consuming, so users may not participate in the annotation process as much.

- If examined by a human annotator, the collection of video footage may be a source ground truth for annotations, but it may also be revealing in unexpected ways and so video footage can raise serious privacy concerns.

### *User-Generated Annotations*

Retrospective methods typically do not involve the users in the annotation process and instead require external assistance from expert annotators. The ubiquitous computing research community has recognised the importance of leveraging user-generated annotations. Similarly to Intille et al. [49], we also propose that users, not researchers,

occasionally provide annotations for their own activities as they happen. Engaging directly with the users (1) relieves the annotation bottleneck by expert annotators and (2) ensures that the annotations are collected in a more naturalistic context where users do not feel excessively monitored (i.e. by video footage or in the constant presence of a human annotator).

A central issue with user-provided annotation is their timeliness. A self-reporting method, Ecological Momentary Assessment (EMA), described by Smyth and Stone [63], which is also known as Experience Sampling Method (ESM) according to Intille et al. [49, 64], is successfully used in medical research to allow patients to report relevant symptoms, conditions or circumstances as and when they occur. Data integrity levels in EMA/ESM are high and Smyth and Stone [63] argue this may be due to the timeliness with which input is given. We too take advantage of this timeliness and we propose that the user takes ownership of annotating some of her own activities as they happen. In addition, we continually monitor user context and identify which activities should be annotated so that the user's participation translates in optimal model improvement.

Obtaining annotations straight from the users creates opportunities to reinforce or adapt known contexts and to augment the set of contexts. For instance, Nguyen et al. [65] propose to adapt a crowdsourced acoustic model from annotations generated (without expert supervision) by the users themselves in the environments they visit. In terms of context augmentation, *SoundSense* (Lu et al. [66]), uses the microphone on a smartphone to monitor not only predefined categories of sound (speech and music), but also a variable category of ambient sound. The authors use unsupervised learning to automatically discover frequent novel patterns in the user's monitored data. When such a pattern is identified and if it is dissimilar to previously discovered ones, the user is asked to provide an annotation for it. This approach allows the user to increase the vocabulary of contexts or activities, in similar fashion to Hossmann et al. [67].

Given the typical limited amount of annotations that can be collected through typical observational studies, Kawaguchi et al. [68] proposed collecting physical annotations on a large scale, directly from users. Aided by a smartphone app, users could opt in to generate annotations for their activities with the consent that their data would

be uploaded to a centralised HAR database. Similarly to the authors of *ActiServ* Berchtold et al. [69] or Hossmann et al. [67], the user must be conscious about her intention to execute a physical activity. Therefore, the user must first signal on the app that she is about to exert an activity and when the user is done, she must again inform the app as soon as the activity ended. While the interactions with the annotation device are very granular, the advantage of obtaining annotations in this way is that the user is ultimately in control of when annotations are provided – she is not interrupted with potentially intrusive annotations requests.

Some authors recognised that this kind of repetitive input on a smartphone can be tedious, so voice commands have been proposed instead of tactile input by Harada et al. [70] (*VoiceLabel*), Hoque et al. [71] (*Vocal-Diary*) and van Kasteren et al. [72]. van Kasteren et al. [72] report near errorless voice recognition, but Hoque et al. [71] have shown that, in a different context, the precision for some labels can drop to 80%. The added layer of voice recognition may result in additional errors in the activity model. We want to avoid such errors and we suggest that annotations be collected using an unambiguous interface, such as a tap-only interface on a mobile device.

However, using interruptions on a smartphone to require (as opposed to the previous paragraph, where annotations were merely provided by a purely benevolent user) physical activity annotations from users has been also tried before, for example, by Cleland et al. [73], Abdallah et al. [74, 75] and by Miluzzo et al. [76] in the *CenceMe* application. As discussed later, in our approach, we apply a heuristic to reduce the number of user interruptions and this distinguishes our work from Cleland et al. [73] or Miluzzo et al. [76] who do not apply heuristics to identify potential annotations which might not make the most of the users' annotation effort. The differences between our work and Abdallah et al. [74, 75] are more technical (they revolve around the timing of annotations) and they are discussed in detail in Section 2.5.3.

Reducing the number of interactions is known to be an important factor and has been researched (in Section 2.3 we discuss some key papers for our work). For example, the authors of *YouSense* Linnap and Rice [77] define a trade-off between the importance of an annotation and the cost of interrupting the user. Their work, however, is aimed at geo-contexts, so their heuristic function which values geographical coverage is not

applicable to our scenario.

### Proactive Annotations

Users may be asked to *proactively* initiate the annotation process by declaring in advance that they are going to perform an activity and provide a label when the activity will have finished. This may be done for model personalisation, as suggested by Berchtold et al. [69], or simply as part of the experimental protocol for collecting user-provided annotations, as done by van Kasteren et al. [72].

However, by essentially providing an annotation before the usefulness of obtaining that annotation can be estimated, one denies the possibility of directing annotation effort only towards the most promising annotations.

### Reactive Annotations

Not all annotations bring equal improvement to the model. This has been shown, for example, by Longstaff et al. [78] and it is a concept we leverage in Chapters 4, 5 and 6 where we seek to find the annotations that overall bring greater performance gains than randomly provided annotations. Annotations are *reactive* in the sense that the annotation process is initiated after the activity has finished.

Because reactive annotations are, by definition, aimed at activities which occurred in the past, user-driven *retrospective* annotation techniques (as explicated in Section 2.2.1) could potentially be used with the user performing the role of the annotator. However, either the user benefits from the same ground truth collection infrastructure (since memory recall deteriorates with time [50]) or steps have to be taken to simplify the annotation task.

If the user has the same access to the same collection of ground truth as an expert annotation, she may inspect the video footage and identify segment boundaries and assign activity labels. Nonetheless, the annotation process can be simplified if in addition the segment boundaries can be automatically estimated. This shields the user from the relatively complex task of delineating her own activities and, instead, has to provide only a label for the activity. As we discuss later, activity segmentation

is a difficult problem, and especially so in the absence of prior annotations. In this case, the following reactive annotation scenario is applicable:

1. The system would identify what segments are most useful for annotation (i.e. are expected to bring a relatively large gain in terms of activity monitoring accuracy).

2. The user would then be prompted with ground truth video footage for that segment and would be asked to provide a label.

The techniques that quantify how a segment is deemed "useful" are discussed at length in Section 2.5.3.

Nonetheless, if segment boundaries are detected automatically in real-time or *online* and the usefulness of annotating the most recent segment can also be estimated in real-time, as Abdallah et al. [74, 75] propose, then the user does not need an infrastructure for ground truth – she may simply use her short-term memory.

In this thesis we present several case studies where we either employ a fully automated online segmentation procedure or assume a perfect online segmentation procedure that produces segments which are suitable for online annotation by the user.

### Proactive versus Reactive Annotations

Proactive and reactive annotations differ essentially in the moment in time when an annotation is provided. Proactive annotations are provided before an activity starts, whereas reactive annotations are provided after an activity starts. The timing has important implications on the entire annotation process. Using a proactive approach, segments cannot be assessed in terms of their usefulness because the decision to annotate precedes the segment. Therefore, in the proactive case, the user needs to guide the annotation process, but this robs the system of the opportunity of annotating the most critical segments, which are the focus of reactive annotation.

However, the two annotation schemes are not exclusive of each other and could be used in tandem. Reactive annotations can be more efficient in terms of accuracy gains

(when compared to proactive annotations), but, as we discuss in Section 2.5, this is true only if there is already enough class diversity in the collected annotations. This initial class diversity can be expanded through proactive annotation, because proactive annotations for completely new activities can help increase the training set diversity which in turn boosts the effectiveness of the reactive annotations. Nonetheless, as we show, reactive annotations can bring classification accuracy improvements over proactive annotations.

### *Relation to our Work*

In Section 2.1.2, we described the outline of our proposed annotation process. As we pointed out, in our approach, the responsibility of providing annotations rests with the user who is involved in the annotation process – the user provides the activity labels for annotations of her own activities. Since we focus on a mobile scenario, we cannot rely on a heavy infrastructure for collecting ground truth which is typical of smart homes [79] and which benefits from video footage. Instead, in a mobile setting, users find themselves in uninstrumented environments and the only definitive source of ground truth is their memory. This annotation scenario, in order to be compatible with the user's memory limited power of recall, must be compatible with online processing.

Data must be processed in a timely fashion so that all annotation requests reference activities which have finished in the near past. Therefore, the EMA/ESM-style of ground truth provisioning is suitable for our scenario. Moreover, we support EMA/ESM with necessary technological means which support online activity data processing, which include the detection of activity boundaries or the construction of personalised activity models. As soon as a boundary is detected, we propose to ask the user to provide an annotation for this activity if the annotation is expected to substantially improve the HAR model.

## The User's Perspective

Tolerance to interruption while performing an activity is naturally idiosyncratic, as personal patience and inclination to collaborate come into play. Clearly explaining to

the users the purpose of the interruptions and the benefits that can be expected is one way to try and increase their tolerance. Key acceptance factors include the nature of the task, and user awareness that a device is gathering information about the task itself.

In this context, the notion of *intelligibility*, introduced by Lim and Dey [80], has been adopted by the ubiquitous computing community to measure and improve upon the capabilities of interactive systems. Intelligibility emphasises the need to explain to the user the decisions of a context-aware application. However, this works only if the user perceives the application to perform high-confidence actions, i.e. actions which the user can rely on. Therefore, the user should be made aware that her annotation effort leads to the improvement of the HAR model and that the improvement should be observable.

Meschtscherjakov [81] has shown that users become emotionally attached on different levels to their devices, so this can be leveraged to attract the user to interact with the device more often. The user's emotional involvement can be further exploited by *nudging* [82], i.e. instilling subtle desired bias in one's actions, so that users can be influenced to act in a desired way, namely to provide annotations for their own actions. For example, Consolvo et al. [44] have shown that an attractive design with seemingly pleasing and rewarding animations can change users' behaviour – in this case, even causing them to be more physically active.

### *Interrupting the User*

There exists a significant corpus of prior art that explores how appropriate it is to interrupt users. The effects of interruptions on task performance were explored by Bailey and Konstan [83], who showed that user interruption not only reduces effectiveness in performing and completing tasks, but it may also increase user annoyance. Interruptions, however, are not fundamentally or entirely negative. For example, Sahami Shirazi et al. [84] point out that, while notifications are disruptive by nature, the users do value notifications if they are about their own context. Consequently, if a user is motivated by improving her activity recogniser, then it is arguable that the user will put some effort into annotating her activities.

Assuming some notifications are deemed important, then one must also take into account whether interrupting the user at a certain point in time is appropriate. For example, Pejovic and Musolesi [85] propose using a non-disruptive method of modelling the suitability of interruption using a multidimensional mobile phone trace including current time, accelerometer data and location. In an online setting where the user's sentiment toward interruption is predicted, the authors report large variations in precision and recall, but also large discrepancies between the two. This suggests that interruption models can suit a large spectrum of preferences: from users who are strict about not being interrupted outside their preferred intervals of time to users who prefer not to miss important notifications with less regard to when they happen. Similarly, Fogarty et al. [86] leverage context cues such as video footage to model the suitability for interruption. Using audio processing, computer vision-based techniques and retrospective manual annotation, the authors construct models of suitability for interruption. Using a different approach, Kapoor and Horvitz [87] used a desktop-based application that not only monitored application use and other contextual information, but also probed the user to continually adapt the interruption model.

This general direction of research is complementary to ours because their focus is on the user's sentiment towards disruption, while ours is on maximising the performance of a personalised activity model by carefully selecting what sample data to ask the users to annotate.

### *Budget of Annotations*

Given that a user's emotional involvement is limited and that user interruptions are taxing, user interruptions are a finite resource. Taking this into account, we motivate our work on two fronts. Firstly, we introduce the *budget* concept for obtaining HAR annotations in Chapter 3, which allows for an annotation mechanism that takes into account time-varying user tolerance towards annotation.

Secondly, in Chapters 4 and 5 we attempt to make the most of the user's annotation effort and optimise the accuracy of the activity recogniser by asking the user to annotate only the most promising activities.

Finally, we combine these two concepts in Chapter 6 and maximise the accuracy of the model even if the user's inclination towards annotation changes in time. Our approach takes into account not only a finite budget of annotations, like Helmbold and Panizza [88] or Attenberg and Provost [89], but also respects a time-varying user disposition to how frequently annotations can be requested.

### *Relation to our Work*

In this thesis we accept that asking users to provide annotations is generally disruptive to them. However, we assume that users are still motivated to provide a limited number of annotations distributed according to a priorly-agreed-upon distribution. In line with the notion of *intelligibility*, we assume that if the user's HAR model can be improved and such improvements are made observable by model performance evaluation, then the user would commit to provide annotations on the established limited basis.

While we do not quantify what are the actual user's tolerance levels, we nonetheless provide a generic *budget-based* mechanism to cope with any such levels of involvement. These mechanisms prioritise the timing of annotation requests so that the resulting time distribution of annotations matches the user's expectations.

We explore budget-based annotation strategies in Chapter 3 where user tolerance, modelled as an annotation budget, is the sole criterion of requesting annotations. This approach is taken further in Chapter 6: we apply Online Active Learning as a means of acquiring critical annotations, but we also overlay the restrictions that come with an annotation budget. The end result is a mixed effect between budget-based annotation and Online Active Learning, where highly critical annotations are still identified by Online Active Learning, while the shape of the distribution of annotation requests approximates the desired one.

## Machine Learning

Machine learning lies at the heart of HAR and it is a toolbox of methods and techniques which allow the automatic monitoring of physical activities. Typically, several stages of data processing are composed to form a pipeline for model building and classification,

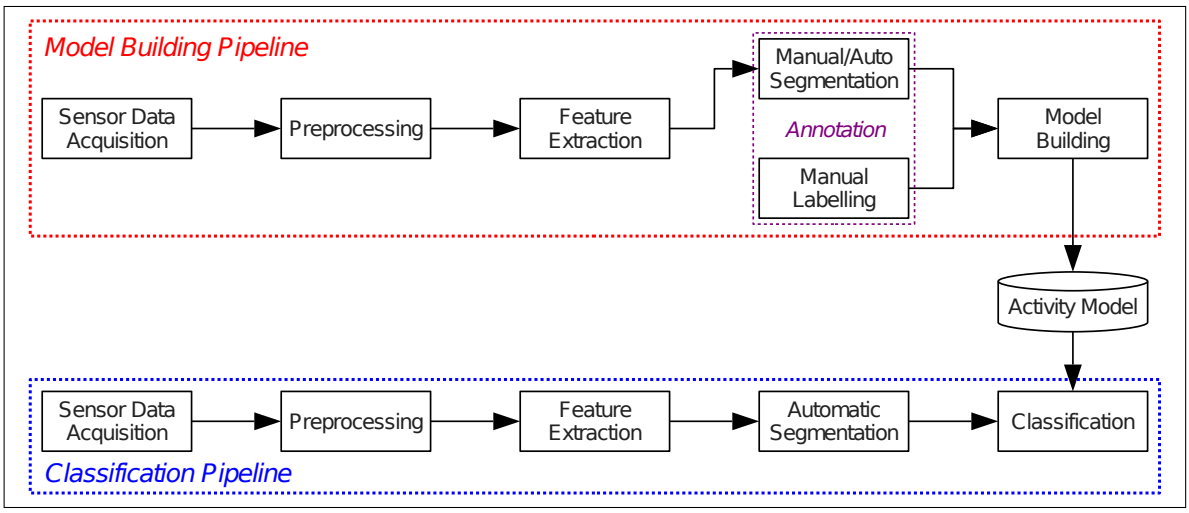



Figure 2.1: Machine Learning Classification and Model Building Pipelines.

as illustrated in Figure 2.1. The activity model, which is the central piece of a machine learning pipeline, is, essentially, a mathematical function which takes as input signals (movement data) and output activity labels (the estimated activities).

The classification pipeline deals with estimating labels for newly registered activities, whereas the model building pipeline fine-tunes the activity model which improves the end results of the classification pipeline. These first three steps in the pipelines, namely *Sensor Data Acquisition*, *Preprocessing* and *Feature Extraction*, are identical so that classification and model building are compatible. In our framework, input sensor data are partitioned into *segments*, which are examined in turn by the activity model, and the final outputs are the estimated corresponding *labels*.

### *Sensor Data Acquisition*

The prevalence of sensing hardware such as that found smartphones with embedded sensors, wearable sensors and ambient sensors has greatly enriched the sensing options HAR application designers have at their disposal.

In this thesis, we explore the mobile scenario exclusively, so our focus is on wearable sensors. These are also a widely used means of collecting data on a wearer's movement, for example, by Chavarriaga et al. [38], Kurz et al. [90] or Morris et al. [57]. Portable sensors can also be found in modern smartphones and have been used to support HAR applications, like in Berchtold et al. [69], Zhao et al. [91] or Kwapisz et al. [92].

Smartphones are an enabling technology not only in terms of potential sensing, but also in terms of connectivity and general computation. Connectivity-wise, smartphones have been used, for example by Pärkkä et al. [93] or Xuel and Jinl [94], as a base station, i.e. to collect sensor data from worn sensors. In addition, Abdallah et al. [75] have shown that smartphones pack sufficient computational resources to support a full machine learning pipeline that provides classification and model personalisation.

Also, a wide range of sensor modalities have been employed in HAR, such as accelerometers, magnetometers, gyroscopes, microphones, pressure sensors, as reviewed by Shoaib et al. [95] or Lara and Labrador [96].

### *Preprocessing and Feature Extraction*

Sensor data preprocessing is a curation and organisation step that prepares sensor data for meaningful feature extraction. The preprocessing operations include high-frequency noise filtering using a low-pass digital signal processing filter (for instance Anguita et al. [97] or Morris et al. [57]) and/or, very commonly, a sliding window, as noted by Bulling et al. [98], which splits a stream of sensor data into *windows* – contiguous chunks of sensor readings that are further processed individually.

Each window is then transformed into a *feature vector* which is meaningful for machine learning. This feature extraction step transforms a high-dimensional window into a smaller-dimensional data product which is suitable for a large class machine learning model building algorithms. For the purpose of feature extraction, numerous data transformations have been attempted in HAR. These include statistical measures [99], such as the means of the values of an axis within a window, their variance, the correlation between different axes, but also many others, such as harmonic content [99, 100] or timeseries auto-correlation features [57].

While features for classification can be manually defined, as we have shown previously, recent advances in machine learning, namely in the direction of *deep learning* [101] have demonstrated that it is possible to automatically infer high-level features from low-level data, such as individual image pixels [102] using *Deep Belief Networks* (DBNs). Plötz et al. [103] have applied the same concept to HAR. They proposed to learn

a high-level feature schema from raw sensor signals and to further use the resulting features for classification. Their results show that, for classification of human activities, deep features work relatively well when compared with other sets of manually defined features.

An alternative to extracting features from sensor data is to compare whole sensor time series. A standard algorithm for measuring dissimilarity between timeseries is Dynamic Time Warping (DTW) [104, 105]. For example, in HAR, Muscillo et al. [106] have shown that physical activities recorded with an accelerometer can be reliably recognised by a DTW-based classifier.

### *Activity Segmentation*

Numerous techniques on how to detect segment boundaries in activity streams have been developed. For example, in environments instrumented with on/off sensors, Laguna et al. [107], Krishnan and Cook [108], Chua et al. [109] and Okeyo et al. [110] have exploited discrete sensor changes to segment activities. However, these methods are not applicable to our scenario because our sensing framework is based on continuous acceleration signals.

Signals from continuously-valued sensors[1] have been automatically segmented using a plethora of methods. For example, Hidden Markov Model (HMM) methods [111] which account for the temporal dependencies in non-periodic gestures and physical activities (i.e. [38]) were used to find segment boundaries by Deng and Tsui [112]. Alternatively, Krishnan et al. [113] and Junker et al. [114] suggest modified versions of Adaboost [115] as an alternative to HMM-based segmentation. Other methods include detecting segment boundaries with Dynamic Time Warping [105] (Hsiao Ko et al. [116]), with string matching using Dynamic Programming [117] (Stiefmeier et al. [118]), or with time-series based measures such as autocorrelation-derived features (Morris et al. [57]). Some assumptions can be built into the sequence of activities (i.e. Cleland et al.[73] assume that higher energy activities are always followed by being stationary) so that a simple segmentation procedures can be used. However, these

[1]These are actually digital sensors with a resolution much higher than 1 - the resolution of the aforementioned on/off sensors.

assumptions are not generally true for a naturalistic environment where the user may engage in different activities in unpredictable order.

These methods, however, are not applicable in our scenario because they assume a prior corpus of annotations to inform the segmentation decision, which is in contrast to our assumption of bootstrapping personalised models from scratch – i.e. assuming no initial annotations exist. Instead, we draw inspiration from the video segmentation literature and adapt an online segmentation procedure devised by Cooper [119] to physical activities recorded with accelerometers. This segmentation technique is detailed in Chapter 3.

### *Model Building and Classification*

Model building, which we refer to as the application of a *supervised learning* algorithm [120], entails using the training data to approximate a *hypothesis*, i.e. mapping from the feature vectors to the corresponding labels. Typically, for a given dataset of labelled examples, supervised learning implies the search of a hypothesis which is optimal in some way like, for example, one that minimises the prediction error. A general difference between different classes of learning algorithms is how the space of hypotheses is explored to yield an optimal one.

A plethora of model builders have been reported in the HAR literature and Shoaib et al. [95] and Lara and Labrador [96] outline numerous such works. These model builders yield activity models which are subsequently used for classification, i.e. the act which entails estimating labels for continuously monitored sensor data.

### *Relation to our Work*

In this thesis, we set up a number of machine learning pipelines for different HAR contexts. In these we include standard machine learning techniques and algorithms like the ones previously discussed. Our focus is entirely on accelerometer data from wearable sensors as this sensor modality is common to wearables and smartphones. For the most part, we use a typical sliding window approach over a continuous timeseries of acceleration sensor readings. Because the machine learning pipeline serves not only to

recognise human activities, but also to acquire annotations, possibly from scratch (zero starting knowledge), the pipeline must not assume prior knowledge. Consequently, we adopt knowledge-agnostic approaches such as extracting typical statistical features (means, variances and correlations) or using Dynamic Time Warping to contrast time-series directly. Deep Belief Networks (DBNs) could be an alternative mechanism for extracting signal features, but these need a relatively large amount of unlabelled data that ideally covers all activities of interests. However, at the beginning of the bootstrapping process, not all activity classes may have occurred, so a trained DBN at this point might not be sufficiently representative of all activity classes.

As we have shown, automatic activity segmentation is still an ongoing research problem. Challenging cases which necessarily combine activity segmentation and activity recognition into one typically require a prior corpus of annotations. This is inapplicable to our scenario because we cannot assume forms of prior knowledge. Instead, we apply a knowledge-free segmentation method.

In terms of classification models for activity recognition, we applied model builders commonly used not only in HAR, but also in numerous other fields. For most cases, we enhanced the capabilities of the Naive Bayes classifier by constructing an ensemble of individual classifiers using Bootstrap Aggregation. Alternatively, when it is beneficial to retain certain characteristics of the timeseries, such as temporal structure for non-periodic activities (discussed in Chapter 4), we compared timeseries directly using Dynamic Time Warping and use a k-Nearest Neighbours classifier.

## Learning Methodologies for Model Building

Supervised learning, discussed previously in Section 2.4.4, is not the sole approach to constructing HAR models. In this section we consider *unsupervised learning*, which does not require personal annotations, and *semi-supervised learning*, which seeks to complement a supervised model with non-personalised labels, unlabelled examples or incorporating data or knowledge gained from other sources.

### *Unsupervised Learning*

Unlike supervised learning, which uses personalised annotations to construct a model that maps feature vectors to labels, *unsupervised learning* does not require the accumulation of personalised annotations to improve a user's model. For example, Chavarriaga et al. [121] use unsupervised learning to correct for variability in sensor placement and rotation. Maekawa and Watanabe [122] avoid personalised annotations altogether by constructing individual training sets based on a user's physical characteristics, such as age, height and weight.

While purely unsupervised approaches bring some improvement, ignoring supervised learning techniques does not, in general, fare well for classification performance. In fact, both supervised and unsupervised techniques can improve HAR models. As we discuss further, supervised and unsupervised techniques can be unified into semi-supervised techniques to use personalised annotations as a springboard for deriving new knowledge from auxiliary sources, including the user's own unlabelled data or other users' annotations.

### *Semi-Supervised Learning*

Obtaining personalised labels has been recognised as a difficult endeavour by numerous researchers [55, 78, 123, 124] who proposed to improve existing classifiers by leveraging large corpora of unlabelled examples which are easy to collect or corpora of data representative of similar contexts [52, 54, 90, 122, 125, 126].

In this section we explore prior work of using personalised corpora, i.e. which contain both labelled and unlabelled examples belonging to the user for which personalization is made. Several techniques, including *self-training* [78, 123], *co-training* [78, 123] and *multi-instance learning* [55, 124], utilise the user's activity model to infer labels that further the user's training set. We also look at prior work on *transfer learning* [52, 54, 90, 122, 125, 126] which attempts to adapt non-personalised examples (e.g. collected for other users). Finally, we investigate *active learning* [48, 51, 53, 60, 74, 75, 78, 88, 89, 96, 123, 126–129], another semi-supervised methodology which seeks to expand the user's training set, but, unlike self-training, co-training and multi-instance

learning, the user is asked to label some of her own unlabelled examples.

Some of these methods usually rely on an initial corpus of annotations in order to infer labels for new examples which are then used to complement the initial corpus. A notable exception is active learning which can work without any prior annotations. In particular, in the case of active learning, this is not without technical difficulties, as pointed out by Sculley [48]. At the very beginning of the bootstrapping process when there is very little or no class diversity, active learning may be misleading in terms of annotation decisions. This problem is explored in detail in Chapter 5 where we propose mechanisms to alleviate this issue.

In general, existing research has shown that improvements for HAR models can be obtained by employing these semi-supervised techniques. In the case of non-active learning methods, the direction of research is largely complementary to ours because we focus on mechanisms to obtain definitive personalised labelled examples for activities. In existing non-active learning research, these initial corpora of labelled examples can be further enriched with unlabelled data or non-personalised annotations.

#### Self-Training

One way to increase the quantity of labelled examples is to include in the training set the examples which are classified with the greatest confidence by the current classifier. These initially unlabelled examples, which are personal to the user, are labelled with the activity label inferred by the classifier. This was done by Longstaff et al. [78] and Stikic et al. [123]. Essentially, from the activity classifier's point of view, the examples with the greatest confidence in classification are assigned as ground truth labels the estimated ones from classification. We too use classifier confidence, but we do not self-train and infer labels because the latter may be incorrect. Instead, we decide whether to obtain a definitive label for an unlabelled example from the user herself.

#### Co-Training

In similar fashion, Longstaff et al. [78] and Stikic et al. [123] have used classifier confidence to infer new labels. They have split the features of labelled examples in

groups which are independent given the label. A classifier trained on each group passes to the other classifiers labels for its own most confidently classified examples. The process iterates a number of times and the most confidently classified are then assigned inferred labels. Again, this process may yield incorrect annotations and, so, is no substitute for definitive annotations.

### Multi-Instance Learning

Stikic et al. [55, 124] proposed extending known activity labels to unlabelled examples which are temporally or structurally close to the labelled examples. An unlabelled example is temporally close it was registered shortly before or after a labelled example. Similarly, a labelled example is structurally close if the feature representation is not dissimilar to a known labelled example.

### Transfer Learning

Transfer Learning, as surveyed in general by Pan and Yang [125] and by Cook et al. [54] for activity recognition, entails the existence of two data domains: a *source* domain, which abounds in data, and a similar, but not necessarily identical, *target* domain for which data from the source domain must be adapted.

Lane et al. [52] seek to personalise models by leveraging existing corpora of unlabelled examples collected from other users. In addition to an initial corpus of annotations, they exploit a set of similarity measures between users (at the level of raw sensor readings, at the level of physical body measurements and at the level of lifestyle) to decide what other examples to include in a user's training set. The use of personalised labels differentiates this approach from Maekawa and Watanabe's [122] method, described earlier, which is focused purely on inter-user physical similarity.

Transfer learning has been used by Kurz et al. [90] to adapt to changing sensor configurations that are expected to happen if sensors are discarded (for example, the user removes an article of clothing with an embedded sensor) or, conversely become worn again. They adopt a *teacher-learner* methodology within sensing networks where nodes may come alive unexpectedly. Classifiers corresponding to newly integrated sensors lack labels, so existing classifiers monitor new activities and pass estimated labels

to the new classifiers. New classifiers are "taught" labels by adopting as ground truth the predicted labels from classifiers trained with data from existing sensors.

Another example of transfer learning is outlined by Shi et al. [126]. They similarly apply labels from a source domain to construct a classifier that estimates labels for the target domain data points. However, if the classification accuracy is low, then they seek to complement transfer learning by obtaining labels from expert annotators. This latter technique falls in the realm of active learning and it is described next.

### *Active Learning*

Active learning is another semi-supervised learning paradigm. Instead of augmenting training sets with uncertain labels, as do self-training, co-training or multi-instance learning, *active learning* identifies unlabelled data points which, if annotated, are expected to bring considerable improvement to the performance of the model.

Active learning is therefore a trigger for *annotation requests*. Requests can be alternatively requested at random, something called *Random Selection*, but the mechanism behind active learning weighs possible annotation requests and selects those that are expected to bring the greatest improvement to model accuracy. Consequently, the improvement to classification due to annotations triggered with active learning has the potential to be greater than the improvement due to annotations triggered randomly [51].

Active learning is governed by a *heuristic function* that examines unlabelled data and yields decisions over whether or not to annotate those data. The function is heuristic because it does not predict what the improvement in accuracy is going to be or even if there will be an improvement. Rather, it outputs a quantitative measure which it is believed (by the designer of the system) to be positively correlated with, but not proven to guarantee, optimised performance gains. For instance, if a large quantity of unlabelled examples is available and one annotation has been made, there exist heuristics, such as the *confidence* in prediction of the classifier [51], which output the ranking of all examples corresponding to the expected gains in performance.

Because active learning requires the inspection of unlabelled data, it suits *reactive*

annotation strategies. On the contrary, random selection does not inspect the data, so it can support both *reactive* and *proactive* annotation strategies.

### Pool-Based/Offline Active Learning

As mentioned earlier, in HAR, users should only be asked to annotate limited amounts of just the most relevant data because overly frequent requests can lead to reduced user compliance. For example, in a bid to obtain sufficiently many user-provided annotations for supervised model building and evaluation, Intille et al. [49] have generated annotation requests every 15 minutes for two weeks. The resulting level of user compliance was very low and the authors believe this is due to the excessive disruption that competes with normal living. In our approach, we propose that annotation requests are informed by the user context, so that only the most beneficial activities are annotated by the user. In addition, as we show in Chapter 6 using budget-based techniques, annotation requests can be suppressed if they would occur more frequently or in a larger volume than that specified by the user.

Active Learning (AL), serves to orchestrate the accumulation of labelled segments in the training set in such a way that it improves the gains in recognition accuracy over random discovery of training data (Random Selection). In HAR, many attempts focus on *pool-based* active learning – offline datasets are used and the annotation of data is simulated by revealing one or a few labels at a time from the entire dataset or from a large subset, as done by Rebetez et al. [53], Stikic et al. [123], Longstaff et al.[78], Alemdar et al. [127], Bagaveyev and Cook [60] or Hoque and Stankovic [130].

In general, a heuristic function examines the input datasets and identifies the most promising data instance to annotate. A good choice of the heuristic function and a comprehensive view of large parts or the whole dataset promise good optimality in choosing what points to annotate, but, from a user perspective, this places unrealistic expectations on the user memory. In reality, people cannot be expected to precisely remember the individual activities which took place in the distant past or the associated exact start and end times. If the source of ground truth was the user's memory such approaches would lead to unreliable annotations.

In a mobile scenario, users do not benefit from an intricate infrastructure for collecting

ground truth, so, in order to annotate their own activities, they would have to use their short-term memory. Pool-based active learning is not directly applicable because annotation requests are not generally timely, so we have to investigate other approaches which are compatible with this user limitation.

### Stream-Based/Online Active Learning

Simulations that operate on datasets of annotations curated by researchers and experts can afford pool-based Active Learning or similar approaches. In reality, in many cases, activities unfold sequentially, so it is logical to construct a stream-based HAR system, like Abdallah et al. [74, 75] who propose that each annotation decision is aimed at clusters of potentially multiple activities. Additionally, annotation requests are aimed at recent segments in the stream of activities.

Stream-based Active Learning, which we will refer to as *Online Active Learning* because annotations are requested as a result of real-time processing, goes hand-in-hand with EMA/ESM. By using a heuristic that operates only on a user's most recent segments, stream-based active learning parsimoniously asks the user to annotate those segments.

Nonetheless, stream-based active learning is justifiable if it can outperform random selection. In this scenario, random selection corresponds to the randomly selecting recent activities for annotation, so the user's short term memory could still be used as a source of ground truth.

Abdallah et al. [74, 75] apply active learning to a stream of activities by selecting entire clusters of activities for annotation. Online Active Learning, in this application, is a means of personalising a prior model trained with non-personalized annotations. While they provide curation techniques that remove most of the outliers to keep only the predominant label in a selected cluster, the activities considered are not very diverse. In contrast, we propose to direct annotation requests at individual activities and we evaluate the system against more diverse activities. Additionally, we bootstrap personalised models, i.e. progressively training from zero knowledge or existing annotations, as opposed to adapting existing models. Furthermore, we evaluate the learning performances which are the result of the annotation strategy and we show that our

online active learning method registers performance gains over soliciting annotations at random.

To this end, we modify an existing Online Active Learning technique already elaborated for spam classification by Sculley [48], discussed in detail in Chapter 4, and adapt it for activity recognition.

Helmbold and Panizza [88] propose another Online Active Learning method. They provide theoretical guarantees provided the classification is performed by an ensemble, i.e. a group of individual classifiers whose predictions are combined into one, for example, by majority voting. However, this method is not applicable to situations when a sole single classifier is used, which is extremely common in HAR [96].

Other Online Active Learning approaches have been suggested, such as Attenberg and Provost [89]. By assuming some recurrence of unlabelled examples and a misclassification cost function, they estimate the label distribution at run-time and identify highly critical training examples which lead to classification improvement over random selection. However, not all of the authors' assumptions are valid for HAR applications. While a misclassification cost function has been used, for example, by Abidine et al. [128], to moderate activity predictions, one cannot assume that identical examples repeat. This is because, using wearable sensors, activities are registered by high resolution sensors sampled with high frequencies [131], so virtually no two activity timeseries would be identical.

While the previously mentioned work focused on streams originating from fixed distributions, Žliobaitė et al. [129] investigate how to correct the annotation behaviour of Online Active Learning when the underlying distribution changes with time, a phenomenon known as *concept drift*. The problem of concept drift has been investigated in the context of activity recognition by Smith et al. [132], but, to the best of our knowledge, we are unaware of a similar application for stream-based active learning.

Finally, Zhu et al. [133], Vlachos [134] or Laws and Schätze [135] observed that the performance due to annotation by active learning increases up to a point and then can decrease gradually. In order to prevent this degradation, they have developed *stopping criteria*, i.e. heuristics that end the annotation process. However, for HAR, we are

unaware of analogous work. In some of the experiments in this thesis we also observe such a performance decline, but we do not attempt to replicate their work on stopping criteria.

### *Relation to our Work*

In this thesis, a large portion of the contributions revolve around using active learning as a means of identifying critical annotations for constructing fully personalised HAR models. While other learning techniques have been outlined in this chapter, these can nonetheless be combined with active learning. For example, active learning can generate an initial corpus of annotations which could then be used in conjunction with other semi-supervised methods to augment activity models.

While substantial research has been done on active learning in HAR, most of the time only pool-based/offline active learning has been considered. As shown previously, in order for offline active learning methods to be applicable, they need to inspect a large corpus of potential annotations at any one time. Due to the deep disparity between the time an activity took place and the instant its annotation is requested, these methods work well only in conjunction with a definitive source of ground truth, such as video footage. However, because of the mobile context in which we operate, the only available source of ground truth is the user's short-term memory, which is insufficient for offline active learning. Therefore, we resort to an *online* variant of active learning which limits annotation requests only to the most recently finished activity – something we assume the user can remember unaided.

The main difference to numerous applications of active learning to HAR is mainly *qualitative* – we incorporate Online Active Learning over the offline counterpart. While functionally identical (offline and online active learning achieve the same thing – a corpus of annotations), non-functionally, they are incompatible. Offline active learning has virtually no temporal constraints[2]. Consequently, offline active learning, is simply inapplicable to our mobile scenario because users cannot be expected to provide annotations for historic activities without a definitive source of ground truth, like video

[2]Any substantial temporal constraint, i.e. limiting the horizon of time in which to ask for an annotation, will severely limit the effectiveness of the method.

footage. In contrast, Online Active Learning can adhere to *soft real-time constraints* [136]. This means that, as soon as enough movement data has been acquired, the system should issue an annotation request with, preferably, as little a delay as possible. Since Online Active Learning, has a different modus operandi, it fits our user memory-related limitations.

Nonetheless, as pointed out previously, regardless of the variant of active learning, it has to outperform Random Selection in terms of HAR model accuracy. Random Selection, in our case, equates to requesting annotations in an uninformed/random fashion. We underline another contribution which is now *quantitative*: As the bulk of our results in the rest of the thesis show, in terms of recognition accuracy, our variant of Online Active Learning outperforms Random Selection.

## Machine Learning in HAR Applications

### *Using Active Learning*

Active learning-based deployments typically integrate an annotation heuristic into a larger, more complex machine learning pipeline that matches the requirements for particular contexts. An example of such a complex pipeline is presented by Shi et al. [126]. As discussed previously, they seek to improve classifier accuracy primarily via transfer learning, but fall back to active learning when transfer learning is insufficient. In HAR, Abdallah et al. [75], integrate active learning into a light-weight machine learning pipeline for a mobile application that provides online monitoring and user interaction. Our proposed application is similarly restricted to a mobile platform.

Classifier confidence, as identified by Settles [51] in his survey, is a popular measure used in active learning annotation heuristics. Classifier confidence has been very popular in HAR in particular. It has been shown, for example, by Abdallah et al. [74, 75], Stikic et al. [123] or Longstaff et al. [78], that an active learning heuristic based on classifier confidence leads to model accuracy improvements when compared to random selection of annotations.

### *Mobile and Continuous Monitoring*

Activity recognisers help their users keep track of important events in their life or their lifestyle, such as the amounts of exerted physical activity. To achieve this, it is desirable for tracking to happen continuously. There has been a commercial explosion of specialised continuous activity tracking devices[3]. These devices typically work in conjunction with a mobile smartphone app, typically in order to upload data to remote servers for processing. Given the ubiquity of physical sensors in modern mobile hardware (accelerometer, gyroscope, light sensor, etc.), standalone smartphone apps leverage available sensors to deliver continuous activity recognition[4].

Real-time activity recognition is not a new idea. For instance, Tapia et al. [137] have implemented such a system on mobile hardware back in 2007. Another example of is by Lu et al. [138] who continuously collect microphone, accelerometer and GPS sensor data in order to track multiple dimensions of a user's context.

In fact, Martín et al. [139] argue that it is possible to compromise between recognition accuracy and resource consumption on a mobile device. The authors demonstrate how one can fine-tune an activity model in order to reduce the computational cost or memory usage associated with continuous activity recognition. Similarly, Abdallah et al. [75] decrease CPU usage by monitoring not individual activity instances, but rather groups of activity instances.

While the computational resources on a mobile device are nonetheless limited, for our proposed application and the hardware we used, we were not hampered by CPU power, memory or battery life, so it was not necessary to address optimising these in any special way, except for using online data processing methods only.

From a usability point of view, the ultimate purpose of continuous monitoring for physical activities is the accurate recognition of these activities. Statistics can be further derived from these estimates into reports. We argue that the plethora of techniques developed for HAR can be funnelled into applications that present reports

[3] `http://uk.pcmag.com/activity-trackers/159/guide/the-best-activity-trackers-for-fitness` - Accessed 19.03.2015

[4] `http://www.techradar.com/news/phone-and-communications/mobile-phones/10-best-fitness-apps-for-android-1145635` - Accessed 19.03.2015

to users. For example, the user's wellbeing [39] or the amount of expended energy [37], can be distilled and presented to the user so that insight into key lifestyle aspects can be gained and informed decisions can follow.

### *Relation to our Work*

As mentioned in Section 2.5.3, we propose Online Active Learning as a means of obtaining critical annotations. However, our proposed Online Active Learning method needs to be supported by a full complement of machine learning algorithms and user interaction functionality, as shown previously in Fig. 2.1.

These algorithms too need to operate in online mode and we justify why our chosen algorithms are indeed online. By online, we understand that the system does not use more computation or more memory as more movement is monitored. Firstly, we argue later in Chapter 5 that both Online Active Learning and the supporting algorithms operate in online mode because they require constant time and complexity. Finally, also in Chapter 5, we apply Online Active Learning in a realistic user deployment and we demonstrate by example that the pipeline is truly online.

A limited resource for mobile devices is electrical energy. In our user-study in Chapter 5, the energy provided by the batteries of the sensors and of the phone were enough to cover the duration of the individual experiments.

## Conclusions

In this chapter, we summarised the technical aspects of our contribution. We consolidated the logic behind design choices by making critical references to a substantial body of existing research.

Our contribution revolves around interoperating HAR monitoring with collecting annotations with the aim of constructing personalised HAR models. We exclude the possibility of any artificial sources of ground truth or external supervision and instead rely on the user to provide annotations for her activities, as they occur in time. Additionally, all annotations must be timely, i.e. directed at very recent activities;

otherwise the user might not be able to remember correctly. By self-reporting under such time constraints, we draw inspiration from EMA/ESM techniques which have a proven track record of good recall.

We enhance EMA/ESM with computational resources so that the user's context is continuously monitored. Activity boundaries are identified in real-time and, for the most recent activity, the usefulness of acquiring an annotation is computed. It is not realistic to assume that the user can be interrupted every time to provide annotations. Therefore, we only require a subset of the most critical annotations. An Online Active Learning method automatically decides whether an annotation is useful enough to justify interrupting the user. In order for this to work, the Online Active Learning method must be part of a larger data processing pipeline which combines general machine learning algorithms and user interaction capabilities.

In what follows, we review the major research contributions by others which relate to our contributions or which we adopt to support our claims.

### Obtaining Annotations

Annotations, depending on the mechanism of delivery can be *retrospective*, if an external expert annotator provides them after reviewing ground truth sources (such as video footage) or *self-reported*, if the annotations are supplied by the same user who is being monitored. Self-reported annotations can also be *proactive*, if the user provides the annotation in advance of the actual activity being exerted (i.e. she promises to perform an activity which is going to be recorded), or *reactive*, if the activity has already happened and now an annotation is requested for that activity. In this thesis, we are concerned with self-reported reactive annotations and we investigate the effects of user involvement on personalised HAR model performance.

### The User's Perspective

In this thesis we accept that, even though the user is involved in the annotation process, the user's willingness to provide annotations is limited and time-varying. Existing research [85–87] shows that it is possible to estimate when it would be appropriate to interrupt a user. Taking interruptions further, by *budgeting* them, it is possible to distribute the annotation effort in time [88, 89], so that a user's expectations with

respect to her availability can be met.

### Machine Learning

A central aspect of HAR is model building and, in particular, *supervised* model building [98]. These approaches presume a typical data processing pipeline, as shown in Fig. 2.1, which chains together preprocessing, feature extraction, segmentation and model building algorithms to create HAR models. A huge corpus of research has demonstrated how different types of data can be used to infer user context [95, 96].

### Learning Methodologies for Model Building

When supervised model building results in unsatisfactory recognition performance, models can be improved further with semi-supervised learning. In Section 2.5, we documented semi-supervised methodologies including self-training [78, 123], co-training [78, 123], multi-instance learning [55, 124] or transfer learning [54, 125]. These techniques augment the existing set of annotations with knowledge from other sources of data – unlabelled data or data from other users.

Another semi-supervised class of methods, called active learning, seek to improve model performance, not by mining external sources of data, but, instead, by discovering annotations in the user's own unlabelled data. Active learning has been investigated in HAR contexts and, typically, *pool-based* or *offline* variants [51, 53, 60, 78, 123, 127, 130] have been attempted. Pool-based active learning operates over a long history of unlabelled data and, so, annotations from this set of unlabelled data cannot be provided from the user's memory and, instead, require other sources of ground truth, such as video footage.

In order to keep an active learning-style of annotation in line with the limitations of the user's short term memory *stream-based* or *online* [48, 74, 75, 88] variants of active learning can be employed. These variants operate on a stream of activities and the only the latest activity in the stream is eligible for annotation. This means that the user can respond to annotation requests by reporting the label of her most recent activity.

In this thesis, because we focus on self-reported reactive annotations, we must create the right conditions for the user to engage with the annotation system in a suitable

way. This includes making it feasible for the user to remember the activities she would be asked to annotate. Therefore an ESM/EMA style of annotation is timely enough to match a user's memory model where the most reliable annotation, at any point in time, is for the most recent annotation. ESM/EMA annotation is supported via Online Active Learning which, in addition, attempts to improve the performance of the HAR model relative to a HAR model constructed on annotations requested at random.

**Machine Learning in HAR Applications**

Finally, in Section 2.6 we discuss critical issues on applying an Online Active Learning annotation system. One of the design choices when constructing a system for HAR is the annotation heuristic. Research has demonstrated that, for the most part, confidence-based active learning heuristics perform better (in terms of model performance) than others. Also, the issue of computing resources for continuous monitoring is also brought into discussion. We show through existing work that there exists a great deal of flexibility on what can be inferred and what resources can be dedicated to the task. For example, one can set up a compromise between resource consumption and recognition accuracy [139].

# 3

# Online Learning with a Budget

## Contents

# Introduction

As an enabling technology, automatic inference of the activities humans are engaged in plays a central role in the majority of ubiquitous and mobile computing applications. Targeting real-world scenarios, Human Activity Recognition (HAR) techniques are often developed in "field deployments", i.e. keeping prospective users in the loop from early stages of the development process. For example, model personalisation is of importance for healthcare settings, which require individual input to generate personalised feedback during physical exertion [140], or to target individual medical conditions [141]. Often, user involvement becomes a technical necessity, where user models need to be adapted or even bootstrapped *from scratch*, i.e. without having access to prior information – be it training data or existing models that could be adapted.

In contrast to lab-based developments, in such contexts it is often difficult to obtain ground truth annotations required for deriving automatic recognisers. Reasons for this can be of a very practical nature, e.g. it is often simply impossible to follow a user of mobile HAR technology for the sake of labelling sample data. More importantly, ethical restrictions often prevent direct observations aimed at obtaining ground truth annotations such as in private (smart) homes. Alternative annotation strategies engage users directly, e.g. through self-reporting of activities [142], or through experience sampling, i.e. prompting users to provide labels for current or previous activities [143]. Such user involvement is disruptive as it interferes with ongoing activities with what appears to be mundane support for a technical system – a task that is typically not the primary focus of the user. Arguably, the tolerance for such active user participation is thus limited.

In this chapter we focus on a *technical* solution that enables prompting in online annotation contexts such that the user's preferences towards interruption are taken into account. Especially for bootstrapping HAR systems this is a non-trivial endeavour. Existing approaches, such as some variants of active learning [144] or semi-supervised learning [55, 78, 123], are not applicable because they require prior knowledge about the activities to be recognised, i.e. annotated data for estimating the underlying

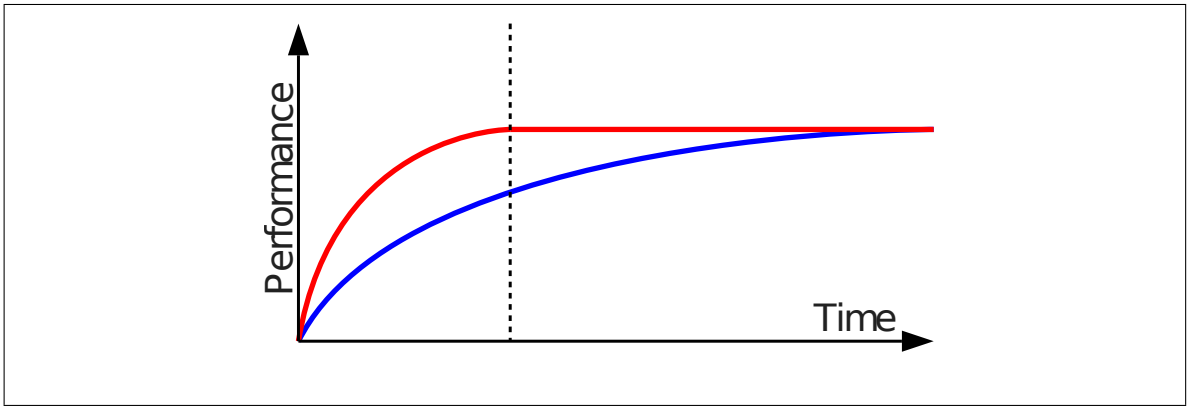



Figure 3.1: Influence of annotation strategies on online activity recognition systems (schematic): Accelerated (red) vs. slower learning (blue).

distributions, or the timeliness with which annotation requests are generated is not appropriate for our case.

Our working assumption is the existence of a *fixed budget of user provided annotations* that a HAR system may spend during its bootstrapping phase. Spending one unit of the budget corresponds to asking the user for the label of an activity. Focusing on online annotation we assume a "worst memory" scenario where users will only provide reliable information regarding their most recent activity. A request for annotation can be made at any given time as long as there is budget available and we assume the provision of reliable annotation.

With these assumptions we explore the effectiveness of possible budget allocation strategies. We aim to explore how annotation strategies impact *model performance*. Fig. 3.1 illustrates how an upfront strategy (red), one which expends the annotation budget immediately, translates in faster model performance than a uniform strategy (blue), one which distributes the annotation budget at regular instances of time. While accelerated learning (red) results in a reliable model earlier on, this may come at a cost in terms of aggravated *user tolerance* which may impact further interaction with the annotation system. The slower strategy (blue) learns more slowly but might be preferred by the user in the long run.

The main contribution of this chapter is an experimental exploration of various configurations and trade offs between budget levels and spending strategies on one side, and accuracy of the HAR models that can be learned in such settings, on the other. Our

findings serve as guidelines for designers of interactive online annotation interfaces to support them in user-centred studies. Specifically, we develop and evaluate budget-based strategies for online annotation of HAR by means of an extensive case study where we simulate online annotation scenarios. We use the Opportunity challenge dataset [38], which comprises of a blend of diverse periodic and non-periodic activities, each recorded at different levels of repetitions using multiple sensors and feature types. We use this realistic simulation to study different problem configurations in detail and in an objective, reproducible way.

Our findings suggest that effective online annotation of human activities can be achieved using a deterministic upfront budget spending strategy or a probabilistic strategy employing an exponential distribution function. Furthermore, the proposed approach extracts and annotates training examples, which also allows us to suggest realistic budget sizes for online annotation tasks. Given that Opportunity is regarded as a realistic and at the same time challenging HAR dataset, these findings are very encouraging for related real-world deployments of budget-based online annotation.

# A Budget-based Online Annotation Framework

## *Overview*

The focus of our work is on exploring strategies for online annotation of human activities, with special emphasis on mobile and ubiquitous scenarios. In such scenarios:

- Ground truth annotations are provided by the prospective user of a mobile HAR technology.
- A budget is available for annotation. Our hypothesis is the existence of a fixed budget which models limited levels of tolerance.

Experimental evaluation in mobile and ubiquitous computing applications is a challenge in itself as interactive scenarios are difficult to replicate, which poses a challenge to objective judgements and is not appropriate for in-depth exploration. In response to this, we systematically assess the effects of different budget spending strategies by

realistically simulating interactions aimed at selectively acquiring user annotations. This gives us complete control over the selection of the subsets of annotations to use, and provides a level of repeatability which would be very difficult to achieve using a field experiment (in addition to being more practical and economical overall).

### *Budget-Based Interactive Annotation Framework*

We address a HAR scenario where the system bootstraps a recogniser that is custom-made for each user, by occasionally collecting input from users while they go about their daily living. The system continuously records sensor readings and, according to a schedule (established prior to the start of monitoring), prompts the user to annotate recently identified activities. Because user compliance to interruptions is a limited resource, not everything is annotated, but, rather, a convenient budget and schedule of interruptions can be specified in advance.

We model the user's preference for the annotation requests with a *budget of annotations* defined as a triplet (*Horizon*, *BudgetSize*, *BudgetStrategy*) where, intuitively:

**The Budget Horizon** is the interval of time the user is willing to reply to *occasional* annotation requests.

**The Budget Size** is the total number of annotations the system is going to ask the user until the **Horizon** expires.

**The Budget Strategy** is a theoretical distribution of annotations over time which models how the total number of annotations **Budget Size** is distributed in time until the **Horizon** expires.

In Chapter 6 we present a mathematically rigorous definition of the budget which is needed for subsequent mathematical derivations and proofs. However, for the purposes of this chapter, the added level of detail from Chapter 6 is unnecessary and the current intuitive definition suffices.

In order to streamline the interactive bootstrapping process, we propose a data processing framework that combines standard HAR data processing and machine learning

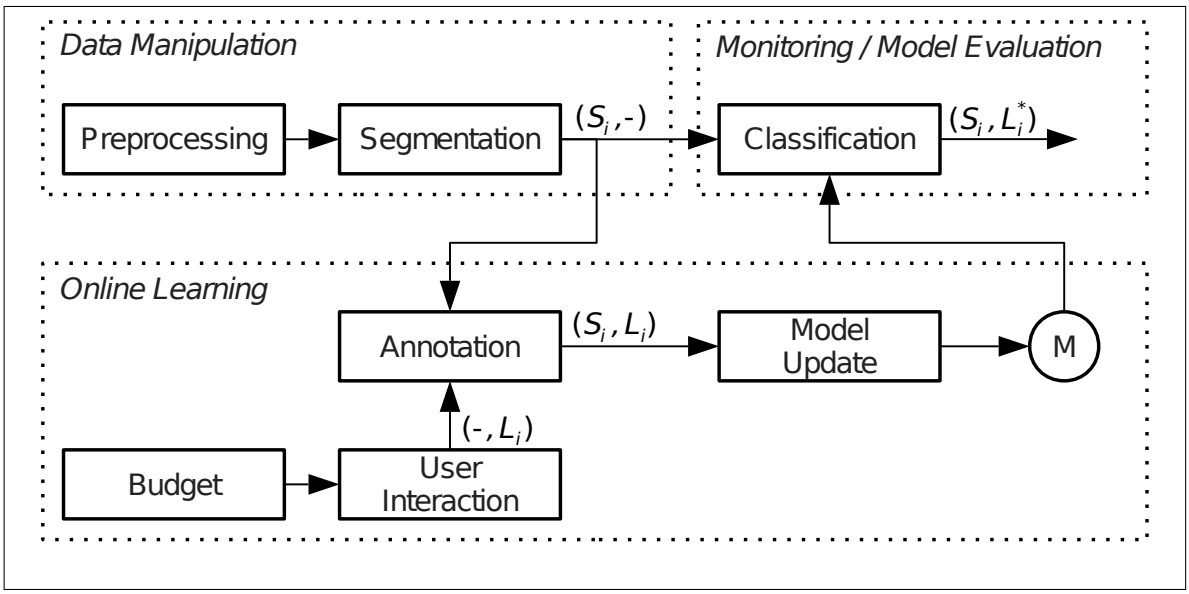



Figure 3.2: Budget-Based Interactive Annotation Framework

procedures, with the capability to collect user-provided annotations. We assume an interaction model focused on annotation requests directed at the user that drive the recogniser bootstrapping process. The interactive pipeline is inherently modular so that one has the freedom to adapt it to the specifics of the HAR application under consideration. Fig. 3.2 illustrates the design of the framework.

**Preprocessing** This step centralises automatic sensor readings and provides the core machine learning preprocessing functions such as sliding window. This involves building a vector of feature sets, by extracting a feature set from the readings in each window [98].

**Segmentation** The preprocessing step produces a sequence of frames. When a frame captures the full characteristics of an entire periodic activity such as walking or running, frames can be used as individual training examples. Composite activities such as those considered here, however, are only fully expressed across multiple frames, suggesting that training examples should consist of sequences of contiguous frames, called *segments* and denoted $S_i$. Segments are derived solely from sensor data and do not necessarily carry a label, unless they are annotated. This aspect is illustrated in Fig. 3.2, where the output to the segmentation stage is a sequence of unlabelled segments $(S_i, -)$. It is these segments that the user is asked to annotate and that are used to bootstrap the activity recogniser.

**Budget** The decision of when annotation requests should be made to the user is controlled by a budget spending strategy, as defined earlier.

**User Interaction** The interaction with the user is triggered by the Budget component and it is responsible for obtaining annotations from users in the form of labels $(-, L_i)$. Because we simulate a field study where the user's short term memory is the only available source of ground truth, the output of this component is driven entirely by simulation. Annotation requests always refer to the most recently identified segment.

**Model Update** When the training set is extended with a new training example (a segment with an associated label), the system re-trains the activity model so that its capabilities additionally reflect the latest example in the training set. As an initial model, we use a *strawman* classifier which, without consideration to the input features, randomly predicts an activity label from a uniform distribution over the activity labels.

**Classification** The classification stage takes, as input, unlabelled segments $(S_i, -)$ and produces, as output, estimated labels $L_i^*$ for the input segments. Since the activity classifier is bootstrapped using incrementally collected activity labels, classification accuracy is expected to increase with the growing size of the training set, as more labels are obtained. Thus, in addition to the final accuracy (corresponding to the point where the entire budget has been spent), in our results we also report the *learning rates*. These are the intermediary classification accuracy scores measured at every stage of the bootstrapping process, namely every time a new training example is supplied and the model is updated.

**Annotation** This component acts as a bridge between the machine learning and the interactivity parts of the pipeline. The annotation stage fuses together segments and user-provided activity labels into training examples which are further used to improve the accuracy of the activity recogniser.

In line with our argument regarding the recall capacity of users' memory, prompting is always done for the most recent segment. When a user provides an annotation, the resulting activity label is associated with the most recent segment for

which a prompt was invoked. However, if, for a given segment, an annotation is not required, then the segment is discarded.

# Experiments and Key Results

We used the previously described HAR framework in a simulated case study where we study the effectiveness of budget-based online annotation. In particular, we focus on the influence of different budget sizes and spending strategies of model performance throughout the bootstrapping process.

## *Segmentation and Budgeting*

### Segmentation

In the initial set of experiments we study the effects of budget configurations on recognition performance. We ignore possible segmentation errors, by assuming, as part of our simulation, a perfect segmentation procedure which identifies all and only the correct boundaries between segments, at the exact point in time when there is an activity change. This assumption is ideal for two reasons:

1. An output segment contains data for only one activity.
2. Segmentation does not split a contiguous activity in more than one segment.

The first assumption guarantees that, if a segment is annotated with a single activity label and if the label is correct, then no label noise is introduced, i.e. an activity label is not extended to the data of another activity. The second assumption ensures that segments are as long as possible, so that a single annotation accounts for as much of an activity as possible.

It is unreasonable to expect ideal segmentation in the real deployment, but we use it to set an upper bound against which we contrast our own realistic segmentation procedure in Section 3.4. We expect that annotating ideal segments leads to improved learning and, because of the lack of label noise, the resulting performance is maximal. In our simulations which use ideal segmentation, we segment the data according to the ground truth labels provided with Opportunity by retrospective human annotators.

### Budget Horizon

In this chapter, the time (which is used to compute budget schedules) is expressed in terms of the number of monitored segments. We therefore defined the budget horizon to be equal to the total number of segments present in the user's training dataset. The exact figures naturally depend on the particular dataset and, in our case, the details are provided later in Section 3.3.3.

### Budget Sizes

The larger the annotation budget, the more annotated segments are available to train an activity model, which results in better recognition performance. Although, realistically, the budget size may be limited by human user, context and application considerations, we are interested in studying only the relationship between recognition performance and budget.

Thus, we experiment with three budget sizes: *small* (10 annotations), *medium* (40 annotations) and *large* (100 annotations). This choice of budget sizes is purely technical as it not only provides insight into expected recognition performance, but also exemplifies how additional annotation effort translates into increased performance. For comparison, as a reference we use the theoretical best-case scenario where the entire sequence of segments is annotated. This *baseline* provides us with an upper bound in model accuracy.

### Budget Spending Strategy

Having decided on a budget size, the next design choice is how to spend the budget. The system uses an online segmentation mechanism, meaning that at any point when a segment is identified, the system must decide whether to interrupt the user to annotate the most recent segment, or to discard it.

We implement the distribution of annotation requests as distribution over a *numbered sequence of segments* with the horizon defined as the length of the sequence of segments. This approach allows us an entire pass through the user's data with each segment being monitored exactly once. As a consequence, each segment will either be

annotated once or zero times. Using a distribution over segment sequence numbers instead of over *physical time* (e.g. expressing the distribution over the duration of a day) is a consequence of the limited amount of HAR data. This limitation and the usage of physical time is discussed in Chapter 7.

For the distribution of interruptions, we use the following strategies:

**Uniform Random** The interruptions are scheduled at random within a horizon of time, according to a uniform probability density function.

**Uniform Constant** The interruptions are scheduled to occur periodically, on each occasion after a fixed interval of time.

**Upfront** The budget is spent as quickly as possible. For every detected segment, an annotation request is prompted until the budget runs out.

**Exponential** The density of interruptions is an exponentially decaying function. Interruption times are sampled from an exponential probability density function, so more interruptions are likely to happen at the beginning and very few toward the end of the horizon of time.

Strategies can be chosen such that the budget is expended as soon as possible (**Upfront**), more quickly at the beginning (**Exponential**) or more evenly across time (**Uniform Random** or **Uniform Constant**). Mathematically, the **budget size** and **budget strategy**, are used to sample individual schedule over the interval $[0, 1]$. Following this, the schedule timings is scaled linearly with respect **budget horizon** so that the annotation requests now lie within the budget horizon.

Because, in this chapter we are interested in the impact of budget strategies on recognition performance, we evaluated the budget-based bootstrapping of personalised HAR models as a *one-off* process (i.e. we consider exhausting a single iteration). The alternative approach which would entail multiple iterations is not evaluated, but it is nonetheless discussed in Chapter 7. In short, multiple iterations would allow one to more accurately approximate the distribution of activities throughout the day and to

target "interesting" activities in future iterations. For this purpose, however, we propose alternative methods in Chapters 4-6 which do not necessarily require multiple iterations of budget exhaustion.

## *Evaluation Methodology*

We use a publicly available dataset to simulate online bootstrapping of a HAR recogniser using user-provided annotations. For simulation purposes, we segment the available labelled data in segments. Consequently, a user's activity exertion is simulated by replaying segments in sequence – as if the user was performing those activities and the data were recorded as a result of continuously monitoring the user. User interaction, by means of annotation requests, is simulated by revealing segment labels from the dataset's ground truth labels (which are provided offline together with the set of sensor readings). We control (1) the number of interruptions by specifying the size of the budget and (2) the occurrences of interruptions by the strategy of spending the budget.

### Performance Measure

We measure model accuracy using a separate test set, which is itself segmented, so that testing is done at segment level. We then calculate the model's F-Score with regard to the segments in the independent test set:

$$F = \sum_i 2w_i \frac{P_i R_i}{P_i + R_i}$$

where $P_i$ and $R_i$ are the precision and recall, respectively, of the classifier on the activity class $a_i$. The weighting factor $w_i$ is defined as the relative numerosity of $a_i$, $w_i = N_i / \sum N_i$, where $N_i$ is the number of segments belonging to $a_i$ in the test set.

### Segment Shuffling

We report the *learning curves* of the classifier during all stages of the bootstrapping process. As learning curves from a single budget expenditure are very jagged, in order to reduce performance spikes or drops, we perform 50 repeated randomisations of activity segments and then report the average F-scores over all randomisations.

### *Dataset*

We use the Opportunity dataset [38] as a means of evaluating our budget-based annotation method. The Opportunity dataset is a publicly available benchmark dataset, which is widely used in current HAR research. Opportunity is known to pose hard learning problems, so it is an excellent benchmark for tools that promise to advance the state-of-the-art in terms of HAR.

We use Opportunity to perform a set of experiments on how to bootstrap recognition systems using budget-based online learning techniques. By using Opportunity and by describing our experimental setup, we have ensured that we ground our conclusions on a non-trivial classification task and that our research is reproducible.

Opportunity contains contiguous sequences of readings from a set of 23 sensors worn by the participants while they perform a vocabulary of common gestures or activities of daily living (ADLs). Opportunity contains data collected independently for four subjects. Each subject has six data files: `ADL1`, `ADL2`, `ADL3`, `ADL4`, `ADL5` and `Drill`. In our use of the dataset, we follow the gesture recognition task in the challenge definition (Task B2) set out in [38]. As specified, we use the gesture sequences in the subsets `ADL1`, `ADL2`, `ADL3`, and `Drill` as the training set from which we draw activity segments, and the sequences in subsets `ADL4` and `ADL5` as the fixed test set, by which we evaluate the classifier's accuracy at each step of the learning curve. We use a subset of the 23 body-worn sensors available in the files, namely we used signals from five tri-axial accelerometers (upper right arm, lower right arm, upper left arm, lower left arm and back), as done previously by Rebetez *et al.* [53].

Each atomic activity, or *gesture segment*, consists of a sequence of adjacent frames annotated with the same activity label, for instance "Open Fridge". In a realistic application scenario (for example in Chapter 5) we would prompt the participating subject to annotate her activities on this segment-level, i.e. the system would ask for one label per activity instance and then assign the same label to all frames this very segment subsumes. In the results presented in this section we assume that the boundaries of each activity have been identified using an existing segmentation procedure (Opportunity Task B1). In the next section, we report on experiments where a

realistic segmentation procedure has been put in place.

We follow the suggestion of Rebetez *et al.* [53] who reduced the Opportunity gestures to seven by aggregating similar ones, namely Open/Close_Fridge, Open/Close_Drawer, Open/Close_Door, Clean_Table, Open/Close_Dishwasher, Switch_Light and Drink [53]. Overall, the extracted segments represent any of Opportunity's 17 *mid-level gestures* [38].

Opportunity contains the *null class* activity label which designates any activity outside the predefined vocabulary of interest – resulting in the aforementioned 17 gestures. We chose to ignore segments labelled as null because the semantics of such segments are too loose and would not generalise well to a realistic deployment. We argue that in a real scenario the user would either give a definitive non-null answer (e.g. "Clean Table") or may simply ignore or dismiss the interruption. In our experiments, therefore, we spend budget units only on actual non-null gesture segments and our recognisers only discriminate between non-null gesture classes.

The user's data is segmented (using an ideal segmentation procedure, as in Section 3.3, or using an automated, but imperfect segmentation procedure, as in Section 3.4) and we set the *budget horizon* to the total number of resulting segments. This allows the annotation method a single pass through the entire data and each segment can either be annotated once or none at all, according to the annotation schedule sampled from the budget definition.

Finally, we note that the Opportunity dataset includes activities that are attributed to each of the four participants. All experiments presented here were performed on a per-subject basis, consistent with our target scenario where it is desired to learn a different HAR model for each of a (possibly large) set of users. We did not mix data belonging to different subjects for training, nor for testing and we did not average results across different users.

### *Classification Backend*

Given that the focus of our work is on exploring effective annotation strategies, we employ a standard analysis approach for human activity recognition, which shall be

deemed to provide reasonable classification accuracy results [98]. The overall procedure can be summarised as follows:

**Input data** In this study we focus on tri-axial accelerometer data. Note that this is not a limitation of the presented approach but rather a practical consideration, consistent with the popularity of accelerometry in contemporary HAR applications.

**Feature extraction** We employ a standard sliding window procedure (e.g. [145]) that translates the continuous stream of sensor data into a sequence of small analysis windows capturing 500ms of consecutive sensor readings, and overlapping by 50%. For every frame we then calculate the mean of each signal representing a simple yet reasonable local feature representation (and is in line with the baseline Opportunity system as described in [146]). Concretely, there are 16 sensor readings in a frame from five sensors, each with three axis, which results in a frame dimensionality of 240. On this data we perform the sliding window based pre-processing and feature extraction procedure as outlined in the previous subsection. This translates the 240-dimensional frames into 15-dimensional feature vectors, which are then fed into the classification backend.

**Classification** These feature vectors are then fed into a classification backend, for which we utilise a standard C4.5 decision tree Witten et al. [120]. In doing so we adopt the approach developed by one of the participating, very successful teams in the original Opportunity challenge [38]. When simulating the annotation of a segment, we include the feature vectors from all of its frames in the training set and then retrain the activity model from scratch. In order to classify a segment, we first classify all the frames in the segment. Afterwards we designate the segment label as the the predominant predicted label (the *mode*) across all frames in the segment. Sometimes there exist ties between two or more predicted labels. In this case we break ties by randomly choosing one of the offending labels as the segment label.

| | **Small(10)** | **Medium(40)** | **Large(100)** | **Baseline** |
|---|---|---|---|---|
| **UR** | 0.39 | 0.62 | 0.64 | 0.78 |
| **UC** | 0.39 | 0.59 | 0.65 | |
| **Upfr** | 0.39 | 0.58 | 0.65 | |
| **Exp** | 0.39 | 0.60 | 0.65 | |

Table 3.1: Final recognition accuracies (F-scores; Opportunity challenge test set) for different budget configurations under ideal segmentation.

### *Results*

We perform an initial set of experiments with the most straight-forward parametrisation, to show how budgeting works in our online setting. We report the results for the Opportunity Subject 1 (out of a total of four subjects) because, for this machine learning pipeline, this subject yielded the greatest performance. Our focus is not on performance optimisation, but rather exploratory. We are interested in exploring the effects of budgeting on performance and Subject 1 presents us with highly visible performance contrasts. This, in turn, allows us to present a large range of performance values caused by different budget configurations and to describe the effects. The presented results are representative for the whole of the Opportunity dataset, but the other participants have less visible performance contrasts.

Ideal segmentation for Subject 1 yields 383 segments, so we use a *budget horizon* of 383 segments. We refrained from analysing null-class segments (as previously explained), as well as from processing segments shorter than the length of a sliding window, and those segments containing missing sensor readings. Evaluation was done against Subject 1's fixed testing set which contains 115 segments.

The results from the initial set of experiments show that it is possible to bootstrap human activity recognisers by involving the user in an online annotation process. Table 3.1 shows the asymptotic performance that can be expected using our proposed budget configurations. As expected, more annotated segments result in better recognition ability. However, it is important to note that *gains in performance* from additional annotations decrease as the number of annotations increases. For example, with 30 additional annotations from Small(10) to Medium(40), recognition accuracy is boosted with approximately 0.2 from 0.39 to around 0.6. However, if the budget was increased

with 60 annotations, therefore advancing from Medium(40) to Large(100), the performance gain would in fact drop substantially to about 0.05, thus increasing the F-Score from approx. 0.6 to only 0.65.

It is also clear that the budget strategy does not affect the end performance, but it has an effect on the learning rate of the activity model. Figure 3.3 shows that the strategy impacts the speed with which the recogniser is bootstrapped. We have plotted the performance of all strategies for the Medium budget size (40 units) and also for the baseline. As explained before, if the budget size is strictly less than the total number of segments, not all processed segments are annotated. Figure 3.3 shows that strategies which request annotations early on, such as **Upfront** or **Exponential** cause a steeper learning rate – they reach the end performance level sooner, whereas lazier strategies such as **Uniform Random** or **Uniform Constant** delay the production of a reliable activity model.

Note that the Upfront strategy does not follow the graph of the baseline exactly. This is because, as explained earlier, during classification it is possible to have ties between two or more segment labels, in which case we randomly choose one of the offending labels. These cases have a slight impact on performance, as can be seen in Figure 3.3, but do not significantly alter reported performance.

We have isolated the learning curve of the baseline illustrated in Figure 3.3 and displayed it in Figure 3.4. This shows an exhaustive analysis of budget sizes, where the x-axis represents the budget size and the y-axis is the expected *end* recognition accuracy. The budget strategy, as we have seen, determines how quickly the end performance is going to be reached.

This pattern of analysis can be replicated prior to field deployments. Researchers may collect relatively large corpora of annotations from a few motivated participants and simulate different patterns of user interaction. The results may be used to inform budget parametrisations of subsequent field deployments.

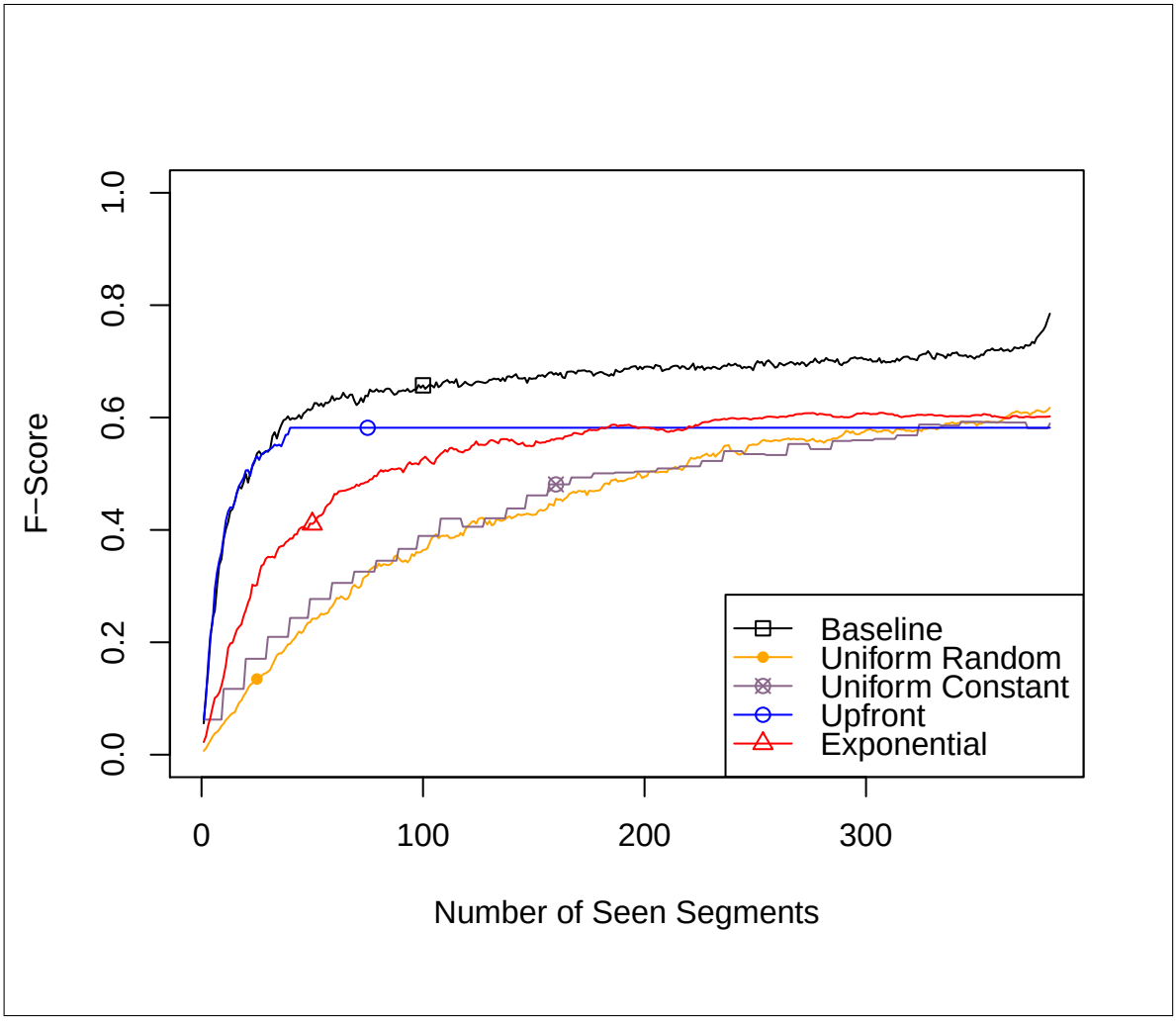



Figure 3.3: Influence of budget strategy over model bootstrapping speed.

### *Conclusions for Key Results*

So far we have shown that it is possible to bootstrap a personalised activity recogniser using online learning, where user annotations are controlled by a budget spending strategy. The results presented in this section indicate that a wide range of performance outcomes can be obtained by varying the budget size or budget distribution. In the following section, we are going to present extended results obtained by relaxing our assumptions on segmentation.

## Extended Results

In Section 3.3 we employed an ideal segmentation procedure which assumed the best case scenario when the exact boundaries of segments can always be detected. We

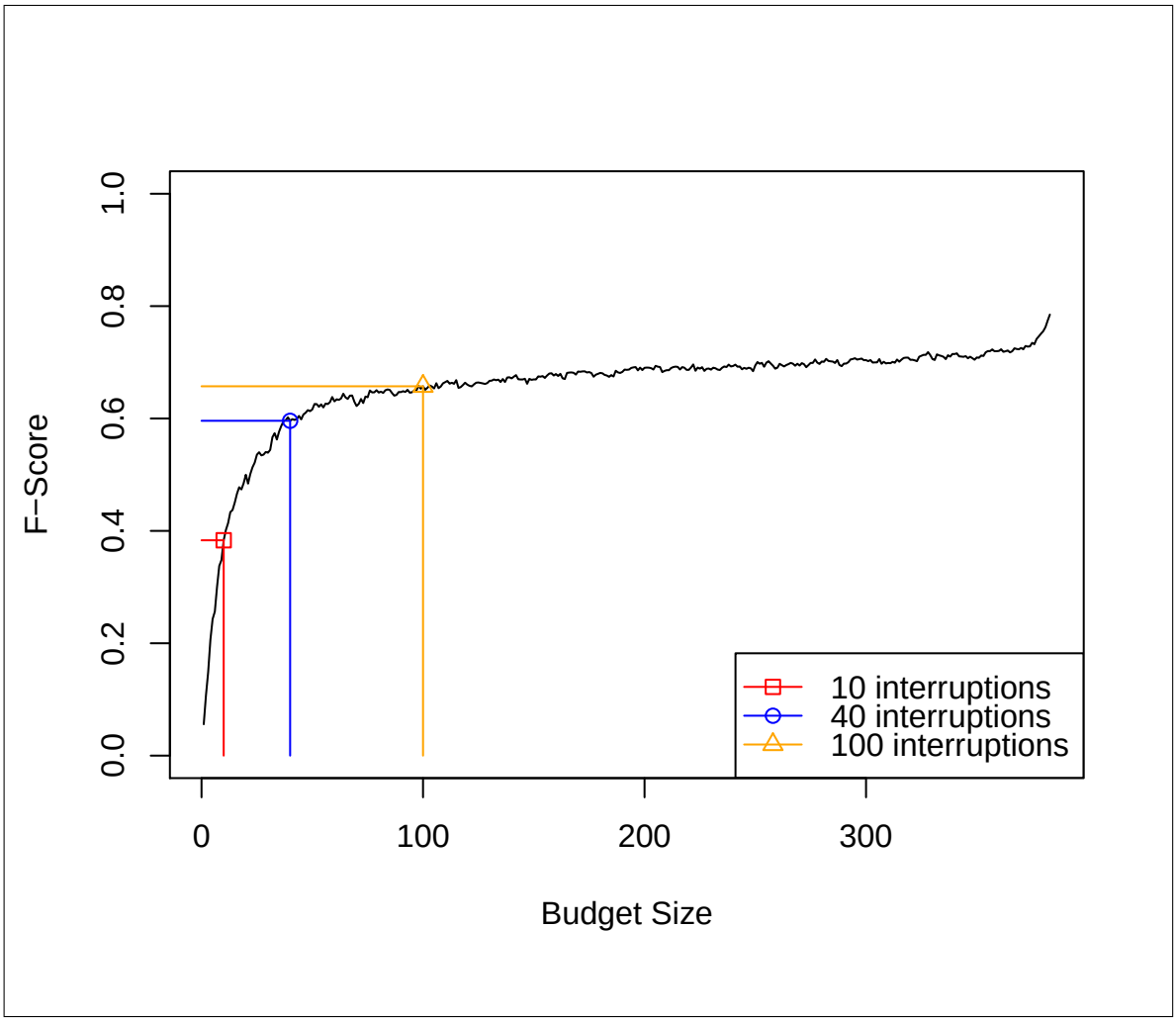



Figure 3.4: Obtaining performance estimates from baseline graph.

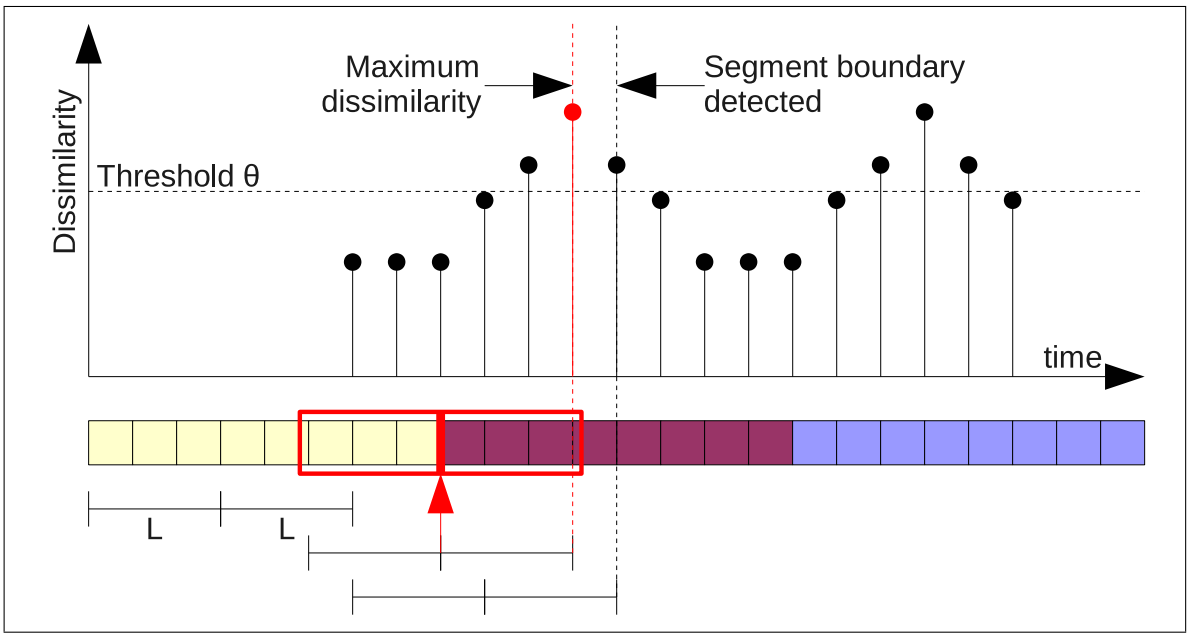



Figure 3.5: Automatic Segmentation Strategy (Schematic)

now relax this assumption and evaluate the impact of a realistic automatic segmentation procedure. We adapted this procedure, illustrated in Fig. 3.5, from the video segmentation literature [119].

### *Segmentation Procedure*

We operate a fixed-length sliding window over the stream of detected feature vectors. We compare the feature vectors in the first half of the window to those in the second half. If the registered dissimilarity between the two halves is great enough, then a change in activity is deemed to have taken place.

More precisely, we consider a window size $K = 2L$, with $L > 0$, covering the most recently produced feature vectors. We refer to the feature vectors indexed by $1, 2, ..., L$ as the first half of the segmentation window and $L+1, L+2, ..., 2L$ as the second half of the window. We then compute an *aggregate distance* defined as the mean of the pairwise dissimilarity between the vectors in the first half of the window and the vectors in the second half of the window. If the dissimilarity is greater than a predefined threshold $\theta$, then a segment boundary is signalled between the frames indexed $L$ and $L + 1$. This means that the last feature vector of the current segment is $L$ and the first feature vector of the new segment is $L + 1$. The process is repeated with every

new feature vector that becomes available. This segmentation procedure yields the sequence of segments, each of which, according to the online active learning method, the user may be asked to annotate.

The dissimilarities are computed on pairs of feature vectors scaled to the interval $[0, 1]$. This ensures that the numeric contributions to the dissimilarity value are roughly uniform across features. We therefore assume that the ranges of the features are known in advance and, so, they can be scaled online. This is a reasonable assumption because, in a real deployment, reliable range values can be trivially obtained without user intervention. For example, the minimum and maximum values for each feature form the range for that feature and, so, can be used to scale that feature. The ranges can be potentially updated with every new feature vector if an extreme value is exceeded. In fact, this is our approach when scaling a stream of feature vectors in Chapter 5 concerning the user study.

As a dissimilarity measure, we compute all pairwise Euclidean distances between the scaled feature vectors in $\{1, 2, ...L\}$ and those in $\{L+1, L+2, ..., 2L\}$ and then compute the average. Let $\{d_k\}_{k\in\mathbb{N}}$ be the sequence of average distances generated from the stream of feature vectors. A segment boundary is flagged between the frames causing $d_k$ if $d_k$ is a local maximum ($d_k > d_{k-1}$ and $d_k > d_{k+1}$) and $d_k$ is above a fixed threshold $\theta$ ($d_k > \theta$).

Our segmentation procedure is *online* because it continuously operates only on a recent sub-stream (the latest $2L$ feature vectors) in order to decide whether a segment has ended. A new segment is detected with a delay of $L + 1$ frames, as shown in Fig. 3.5, so the horizon within which users are requested to provide annotations is limited to the duration of just a few frames.

We performed an initial experiment with $L = 3$ and $\theta = 0.55$. We replayed the dataset in its original order and applied our online segmentation procedure. We chose the parameter values because they resulted in a list of 371 segments, very close to the 383 ground truth segments. For this reason we shall refer to this configuration as *best-effort segmentation*. This entails that overall this setup did not over- or significantly under-segment the data. Just as we did in the first set of experiments, we *shuffle* the order of the generated segments, apply our budget configurations, bootstrap the model

| | **Small(10)** | **Medium(40)** | **Large(100)** | **Baseline** |
|---|---|---|---|---|
| **UR** | 0.33 | 0.48 | 0.52 | 0.47 |
| **UC** | 0.32 | 0.47 | 0.52 | |
| **Upfr** | 0.32 | 0.49 | 0.51 | |
| **Exp** | 0.31 | 0.47 | 0.52 | |

Table 3.2: Recognition performance as a function of budget configuration. Real segmentation.

and evaluate its accuracy at every step.

We emphasise that only the training set data was subjected to our segmentation procedure. When evaluating the classification performance in terms of F-scores over the test set, we did not apply our segmentation procedure on the test set. Instead, we used the original ground truth segments found in Opportunity, exactly as in the previous section.

We reiterate that a segment is annotated with a single label which is passed to all its constituent frames. Imperfect boundary estimation leads to the introduction of label noise within the segments, i.e. some frames are attributed an incorrect label and the model will be trained partially from noisy data.

Table 3.2 contains the classification F-scores of the model trained on segments from our segmentation procedure. If we contrast them with the classification F-scores of the model trained on ideally segmented data from Table 3.1 we can see that the label noise indeed impacts recognition performance on the long term. The accuracy of the model plateaus early around 0.47-0.52, well below 0.78 from the previous section, when employing ideal segmentation. Figure 3.6 illustrates that all strategies encounter learning difficulties due to noise.

According to Table 3.2, end accuracy is not conditioned on the budget strategy. The rate of learning is, however, impacted by the strategy. This generalises our conclusion from the previous section to the current scenario of imperfect segmentation.

The threshold parameter $\theta$ controls the strictness with which segment boundaries are admitted. As explained, for a fixed input, lower values of $\theta$ generally cause more segments to be generated, which may lead to over-segmentation (a single activity is split into several segments), whereas higher values of $\theta$ cause less segments to be

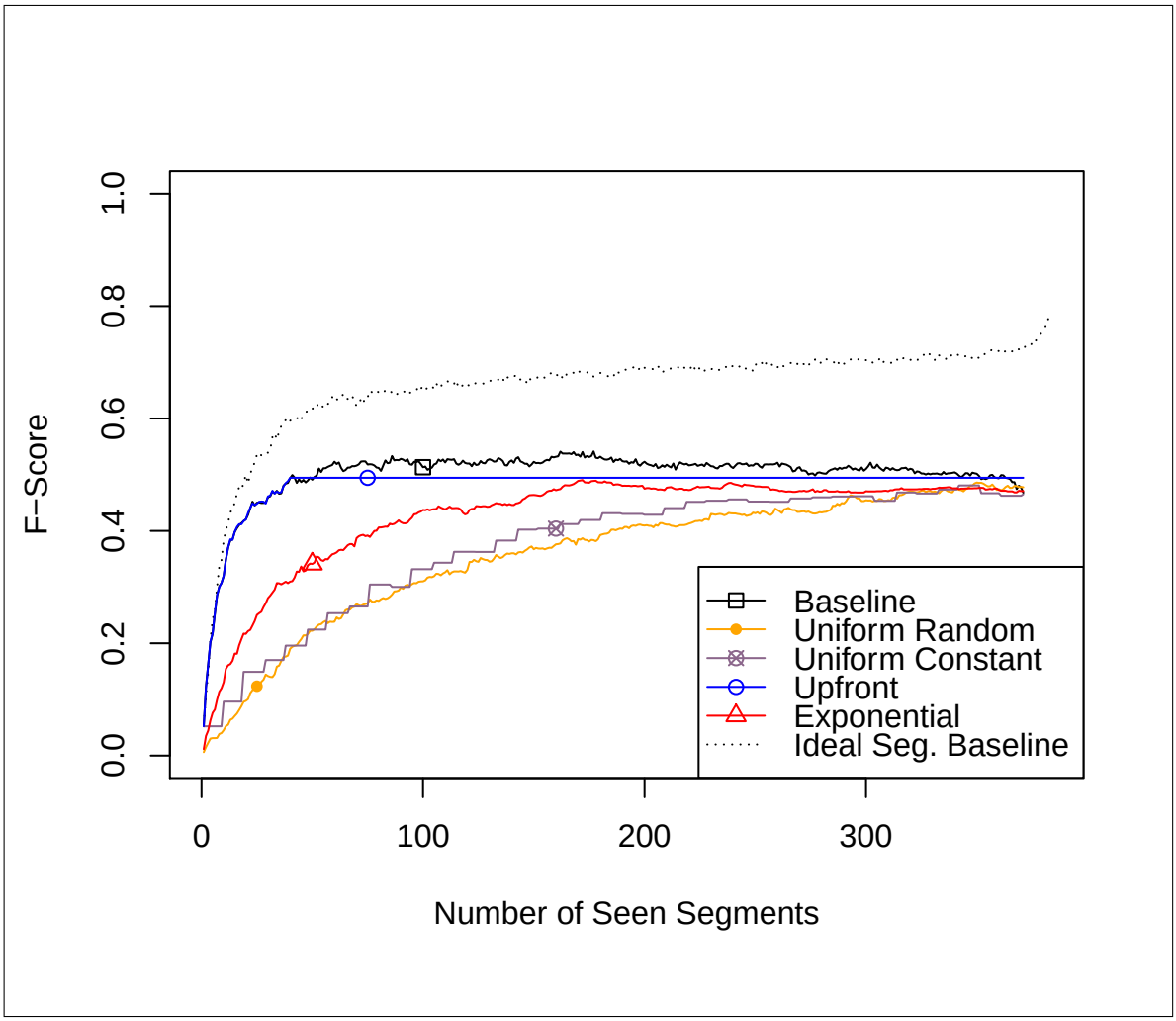



Figure 3.6: Budget strategies with best-effort segmentation.

generated, which may lead to under-segmentation (a segment contains data from more than one activity, possibly including the null activity).

Imperfect segmentation alters the quantity and quality of annotated data. Under-segmentation is likely to reduce the quality of examples because it promotes label noise, but increases data quantity, because it tends to find longer segments. Over-segmentation, on the other hand, is stricter in terms of data quality, short segments are less likely to be polluted with label noise. However, data quantity is affected because segments are forcibly shorter.

We have seen that a substantial amount of noise is introduced even in the *best-effort* configuration. We now lower the threshold to $\theta = 0.35$ and obtain 866 segments. This is well over the original 383, so this is clearly over-segmenting. While this makes segments shorter, Figure 3.7 shows that learning is not substantially impacted. On the contrary, compared to the best-effort configuration, learning is improved and even comes close to the ideal segmentation setup. We are only interested in the direct comparison with the other baselines so we only plotted the first 383 data points. The end performance for over-segmentation setup is 0.75, very close to the ideal setup presented in the previous section. Also, the learning rate is not affected by over-segmentation compared to ideal segmentation, despite the fact that the training examples are shorter.

Clearly very little noise is introduced now and this has a positive outcome on recognition accuracy. The effect on performance is noticeable since the over-segmentation learning curve comes very close to the ideal segmentation one. Furthermore, it seems that the general reduction of segment lengths bears very little impact on learning rate. Learning with over-segmentation is almost as fast as learning with ideal segmentation. We conclude that for our scenario revolving around the Opportunity dataset, the quality of data coming out from the segmentation stage is pivotal to online bootstrapping of activity models. Cautious segmentation results in shorter segments with little possible overlap between activities. When such a segment is annotated by the user, very little label noise is introduced, so the newly provided training example reflects a focused span of an activity.

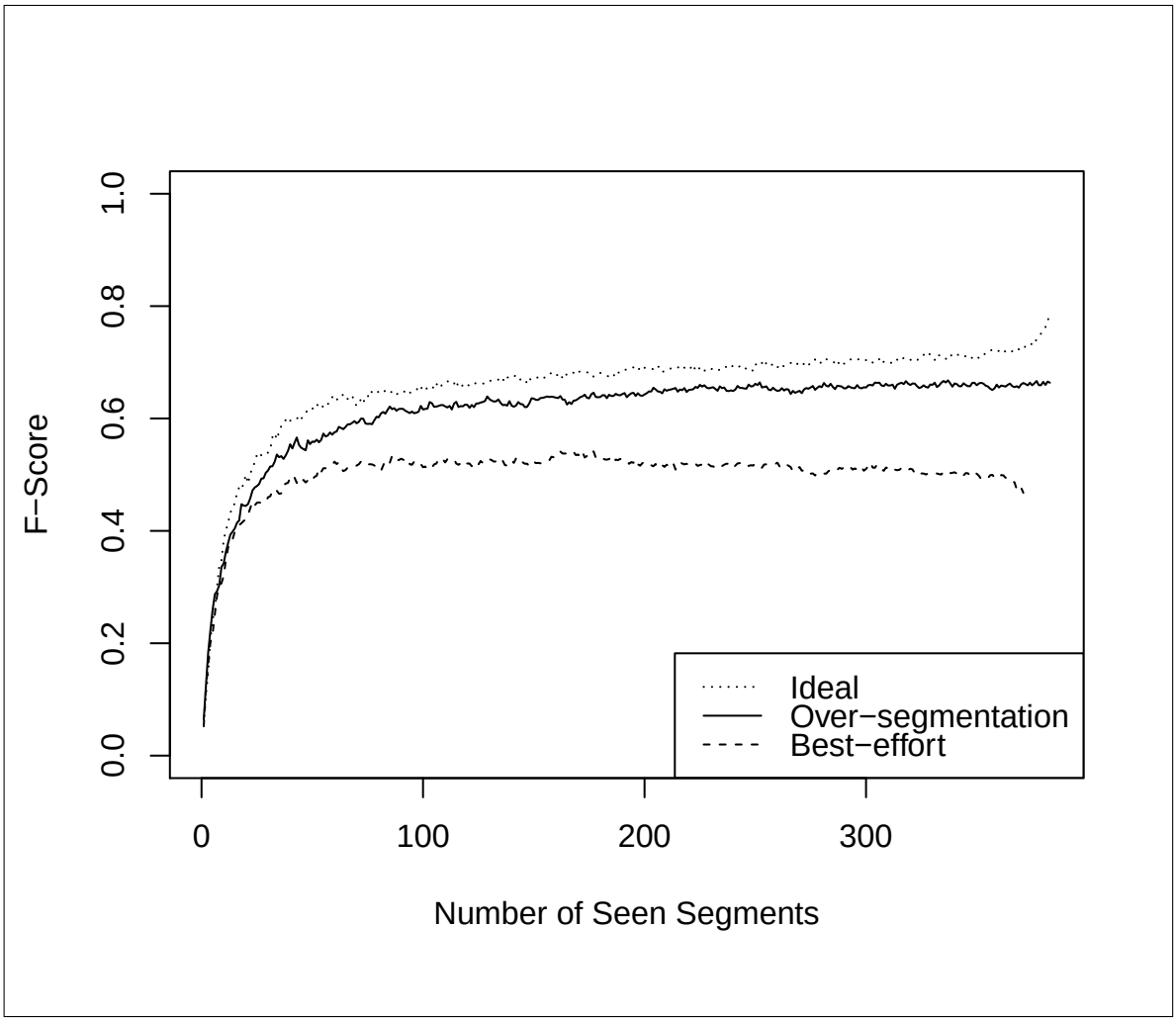



Figure 3.7: Baseline performance with different segmentation configurations.

### *Conclusions for Extended Results*

In this section we relaxed our assumption that segment boundaries can be identified perfectly. Instead, we applied a straightforward thresholding scheme in order to extract segments from sensor readings. However, the resulting segments may not be perfectly aligned with the ideal ones, leading to noise in the corresponding labels, and thus to loss in recognition accuracy. Our evidence suggests that conservative segmentation, while it may produce a higher number of smaller segments than needed, is a reasonable action to reduce the overall accuracy loss.

In the case of imperfect segmentation we have shown that, while the choice of budget spending strategy does not affect the final accuracy, the strategy still impacts the rate of learning and thus the speed with which a reliable activity model is bootstrapped. Results show that this generally holds for training examples of varying degrees of quality obtained from different segmentation setups.

## Summary and Discussion

Learning accurate Human Activity Recognition models requires training examples which are often difficult to acquire in practice. Our work is set in the context of online learning, where further challenges arise. Firstly, the labelled examples only become available incrementally, as the activities unfold. Secondly, labels must be acquired through proactive interaction with the user, who may have limited tolerance for such interruptions, as well as limited memory to recall past events. This leads to the notion of a *budget* of available user interactions, whereby the user is asked to identify the type of activity associated with the most recent sequence of gestures. Thus, a third challenge is that, for the labels themselves to be reliable, the system first needs to accurately detect the boundaries of individual activities, i.e. by properly segmenting the raw sensor data. The combination of these factors leads to a scenario where the learning process can only afford a set number of interactions, which are aimed at labelling the type of activity that is being observed, and under the assumption of imprecise segmentation.

### *Summary of Contributions*

In this chapter we have proposed a principled way of analysing the trade-offs between the number of available interactions (budget), the way the budget is spent over time (budget spending strategy), and the accuracy of the HAR models that can be improved under such budget constraints. Our approach involves extracting segments from the Opportunity challenge dataset and simulating interactions that occur during sequences of activities for an extensive set of budget configurations.

Our main contribution is an experimental method which is generally applicable to the online learning setting. Our results indicate that (i) recognition accuracy close to the baseline (the upper bound model that assumes every activity is labelled) can be achieved by using about 50% of the labels that are potentially available; (ii) the choice of budget spending strategy has little bearing on overall accuracy at the end of training, however it does affect the learning rate, which certainly has massive implications on the overall acceptability of user-involvement in online learning of HAR systems; and (iii) a simple segmentation method, which is decoupled from the recognition task, is an adequate surrogate for ideal segmentation, which is not available in a realistic setting.

### *Moving Forward: From Simulation to Field Studies*

The work presented in this chapter is based on the premise that one can simulate user interactions to explore the effects, on recognition accuracy, of various assumptions regarding the user's tolerance to interruptions and propensity to react to prompts. We now discuss how our findings may inform user-centred studies, leading to practical impact. Open questions concern the impact of imperfect segmentation on the effectiveness of user interaction, as well as the determination of realistic budget sizes and of budget spending strategies.

Firstly, it should be clear that imperfect segmentation may affect the interaction with the user. Over-segmentation, which produces more segments than necessary, may result in the user being interrupted in the middle of an ongoing activity, while under-segmentation may span multiple actual individual activities, leading to user confusion when asked to identify the most recent gesture using a single label.

A related issue is the gap between the time at which the sensor readings corresponding to the transition become available, and the time when the transition is detected. This gap is due to the length of the segmentation window, which requires subsequent readings to be made. For example, a window of size $2L$ where $L$ is three overlapping frames causes a delay of one second (one frame window of 500ms and two overlaps of 250ms each). In Chapter 5, we present the results from a real deployment that personalises HAR models from annotations provided by real human participants. We employ the same segmentation strategy and, because of longer windows in the sliding window procedure, the time delays due to activity segmentation amount to approximately 15 seconds. Subjective feedback reveals that even such a delay is not a problem for users to deal with, because they can remember the most recently finished activity.

Regarding the determination of realistic budget sizes, we expect our results to be instrumental to inform future user studies. This is a complex problem in Human Computer Interaction, where assumptions on user motivation and tolerance to interruptions are being challenged by new generations of wearable devices aimed at self-monitoring. For example, in [147] people were reminded by the monitoring device to expend physical energy after periods of inactivity. A well thought-out user interface can even make interaction with the device enjoyable [44], and techniques such as nudging [82] may be employed to try and influence user disposition to interaction. With respect to budget strategies, in this thesis (including in the current chapter and in Chapter 6) the strategies are defined as distributions over the expected sequence of segments. As noted before, this was done in order to make the best use of the data – this ensures a single pass through all the data with each segment being annotated once or none at all. However, such strategies are arguably not intuitive to the user, but rather distributions over physical time would be more easily understood. This distinction is discussed in more detail in Chapter 7.

The space of options to address usability is potentially broad. For example, one may try to determine whether the current user context is favourable for user interruption [148, 149] and therefore block annotation requests while the user is busy. Also, one may investigate adaptive strategies that attempt, heuristically, to optimise budget spending based on various factors such as the expected performance gain from individual

annotations. In Chapters 4 and 5 we investigate strictly the effects of Online Active Learning on model performance. While budget spending strategies are temporarily ignored in these chapters, in Chapter 6 we combine both Online Active Learning and budget-based spending into a single method which attempts to optimise model performance using Online Active Learning but is subjected to underlying budget spending constraints.

# 4

# Online Active Learning in the Lab

## Contents

# Introduction

In this chapter we present and evaluate an Online Active Learning technique to *bootstrap* fully personalised activity models from scratch, i.e. to start from a zero-knowledge setting and accumulate user-provided annotations that gradually improve the models. In Chapter 3 we proposed obtaining annotations according to a predefined schedule that matched given budget configurations. In contrast, in this chapter we assume there are no budget restrictions and we no longer employ a fixed annotation schedule, but rather the decisions to annotate are made while the user performs the activities. We propose an annotation method which continuously monitors the user's activities and which relies on her to occasionally provide annotations for some of her activities. Any decision to annotate an activity is deferred until that activity has finished. Because of this, we use an Online Active Learning (OAL) approach to inspect the movement data of the last identified activity and to inform the decision of whether or not to annotate that activity. We evaluate our proposed Online Active Learning method on publicly available human activity recognition datasets and results show that the accuracy of activity models bootstrapped with OAL is improved when compared to the corresponding naive annotation method, Random Selection (RS), which triggers annotation requests at random, i.e. completely uninformed[1].

Our annotation method constructs fully personalised activity models for the wearer starting from zero knowledge – no prior annotations are required. Others, such as Abdallah et al. [75], start from an existing corpus of annotations and use Active Learning to personalise an existing generic HAR model. While this is a valid approach in some scenarios, some limiting assumptions have to be made about the application of such a system. Firstly, not all target activities may be known in advance. The personalising system should not restrict the addition of new activities by the user. Also, some activities may not be of interest to the user who may never perform them. From a technical point of view, building this unnecessary knowledge into the recogniser allows for potential recognition errors – namely false positives for these superfluous

[1]Online Active Learning and Random Selection ultimately identify which annotations to request from the users. Therefore, both support the bootstrapping of fully personalised HAR classification models.

activities. Secondly, the sensor configuration indirectly plays a critical role in what activities can be recognised with prior knowledge. A prior corpus of annotations can be used directly only if the wearer's sensor placements match exactly. If the sensor locations differ, it may be possible for the the prior data to be adapted using Transfer Learning techniques, as discussed in Chapter 2, i.e. Cook et al. [54].

In this chapter we focus on the performance gains made solely by personalised annotations (i.e. the annotations provided by the user) using OAL. As mentioned in Chapter 2, models constructed from personalised annotations may be further refined with existing personal data (Self-training or Co-training [78, 123] and Multi-instance learning [55, 124]) or non-personal data or knowledge (Transfer Learning [54]), but we do not duplicate these research efforts.

## *Contributions*

The main contributions in this chapter are as follows:

- **Analysing an OAL annotation decision heuristic.** We propose an OAL annotation decision heuristic that operates over a data stream corresponding to ongoing activities. Similar to other active learning approaches, our heuristic attempts to optimise model performance through informed decisions over what annotations are requested from the user. However, in contrast to previous applications of active learning to HAR, our heuristic does not need a long history of potential annotations. Instead, it works in the severely limited case when only the most recent activity is available for annotation. This ensures that annotations can be reported from the user's short-term memory and that the HAR model performance could be improved with respect to RS.

- **Designing a framework for bootstrapping activity recognisers using Online Active Learning.** We integrated our OAL annotation decision heuristic into a machine learning framework. The framework provides multi-stage processing, with the option of specifying concrete algorithm implementations for each step, depending on the type of data being monitored. The framework con-

tinuously monitors a user's activities and bootstraps a personalised model from user-provided annotations.

- **Evaluation through simulations (in the lab).** We use public HAR datasets to simulate the acquisition of user-provided annotations, as opposed to a naturalistic user study (in the wild), which we do in Chapter 5. We evaluate OAL on both non-periodic activities, using the challenging Opportunity dataset [38], and on periodic activities, using the USC-HAD [150] and PAMAP [151] datasets. In the case of periodic activities, we additionally adopt a method for activity segmentation, which exploits the repetitive nature of the movement to identify segments (contiguous subsequences that ideally span a single activity). Our results show that OAL constructs personalised models which exhibit superior accuracy over models constructed with RS: up to 5% for non-periodic activities and up to 8.5% for periodic activities. In addition, when comparing the number of annotations from RS and OAL, equivalent levels of performance can be obtained from OAL by reducing the number of user annotations by up to 60.8% from the RS case.

  In both situations, using our annotation method shows that informed annotation decisions via OAL can accelerate the bootstrapping of a fully personalised activity recogniser. By applying our method to the non-periodic and periodic cases, we show that our method would be potentially compatible with a large corpus work on classification for HAR [95, 96]. In terms of recognition accuracy, we show that our proposed annotation method, which uses a heuristic function to inform annotation decisions, outperforms the corresponding naive method, Random Selection (RS), which annotates segments at random.

The core of our OAL approach deals with obtaining labels, meaning that we assume a user is willing to respond to annotation requests and to provide the correct labels. However, as noted in Chapter 2, annotations also comprise of segment boundaries. We address the segmentation problem separately for each non-periodic/periodic case in turn. For the non-periodic case, a zero knowledge segmentation procedure, such as that described in Section 3.4.1, is not compatible with our OAL method and our

classification pipeline in this chapter. Consequently, for the non-periodic case, we only assume a perfect segmentation procedure, which has the advantage of isolating the effects of OAL and RS on HAR model accuracy. For the periodic case, we consider both ideal segmentation and the segmentation procedure from Section 3.4.1. The results show that OAL outperforms RS regardless of whether the segmentation procedure is ideal (perfect segment boundaries) or best-effort (some segment boundaries may be misplaced, which introduces noise in the training set).

## Annotation Decision Framework

In this section we present our framework for bootstrapping personalised activity models from user-provided annotations. We propose that annotations are obtained from user feedback, similar to Intille et al. [49]. However, we also draw cues from the continuously monitored user context to identify the segments which, if annotated, would be highly beneficial for model performance improvement.

### *Online Active Learning Heuristic*

The framework detects segments in a continuous stream of activity data, and, using an Online Active Learning heuristic, segment annotations are occasionally requested from the user. Intuitively, the framework maintains a current version of the activity classifier. The confidence levels of the classifier are used to determine the probability that an annotation be requested on the current segment in the stream. Every new annotation obtained from the user is added to the training set, and used to produce a new version of the classifier.

Our annotation decision heuristic, which selects segments to be annotated is adapted, from one proposed by Sculley [48] for online spam classification. In order to adapt the heuristic to HAR, we integrate it into a framework – a multi-stage data processing pipeline with algorithm placeholders for every stage. Depending on the characteristics of the data, different framework instances can be created by plugging in concrete algorithms. The framework is illustrated in Fig. 4.1.

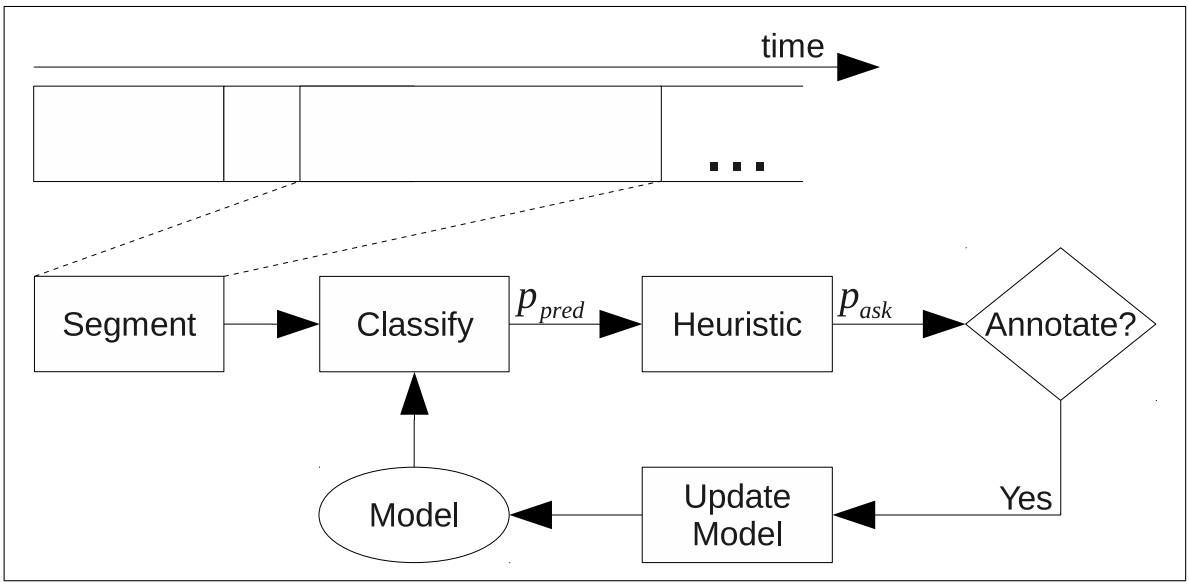



Figure 4.1: Overview of the annotation method.

The end goal of the framework is to produce updates of the HAR model for every new annotation and we evaluate the performance (recognition accuracy) of each of these models. For each newly observed segment in the stream of activities, an annotation request will be issued with probability $p_{ask}$. This value is computed from the confidence associated to class predictions for that segment, as follows. Firstly, the current segment is classified by the current version HAR model. This generates a probability $p^j_{pred}$ for each of the activity classes known to the current version of the classifier. We use $p_{conf} = \max_j p^j_{pred}$ as our measure of overall confidence in the classification. We then define the probability $p_{ask}$ of issuing a new annotation request for that segment as:

$$p_{ask} = exp(-\gamma p_{conf}) \tag{4.1}$$

In Eq. 4.1, $\gamma$ is a tunable parameter that controls the asking behaviour. As can be seen in Fig. 4.2, for a fixed $\gamma$, the probability of asking to annotate a segment increases as the classification confidence of the model decreases. This means that low confidence values $p_{conf}$ increase the likelihood that the segment will be annotated, triggering an update of the classifier. Therefore, annotation requests select only highly critical training examples. Furthermore, increasing $\gamma$ has two consequences, as shown in Fig. 4.2. Firstly, given a fixed $p_{conf}$, the probability of asking for an annotation decreases. Overall, this results in fewer annotation requests. Secondly, when $p_{conf}$

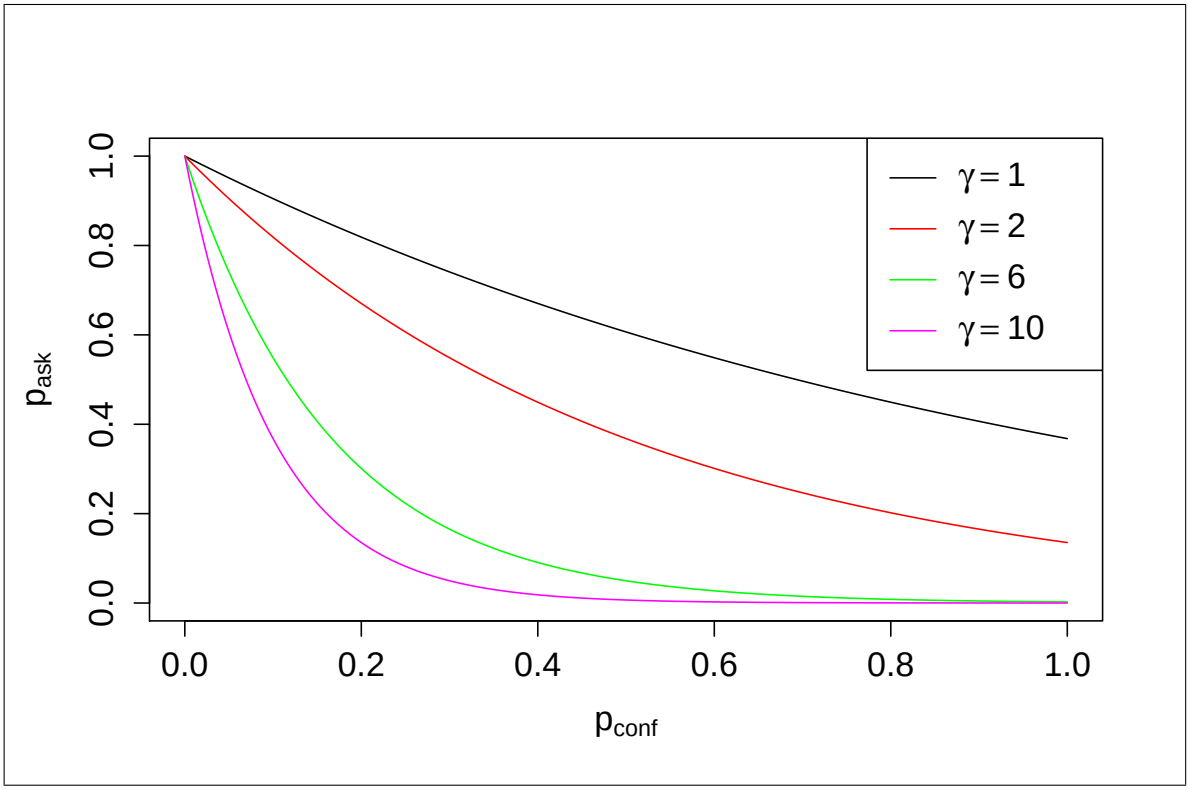



Figure 4.2: The probability of asking as a function of classification confidence.

decreases, the decline in asking probability is more pronounced with higher values of $\gamma$. Effectively, with an increased $\gamma$, segments with high $p_{conf}$ are far more likely to be ignored. Thus, the system is more likely to focus the user's annotation effort on segments with low $p_{conf}$.

Throughout this chapter, we use a *fixed value* of $\gamma = 6$ for our analysis (except on one occasion in Section 4.4.4 where we contrast results for $\gamma = 6$ and $\gamma = 2$). This particular value was chosen empirically because, on the one hand, it is large enough to reject with high probability annotating segments which are relatively confidently classified (which results in clear accuracy gains of Online Active Learning over Random Selection) and, on the other hand, it is still low enough to not reject segments extremely frequently (which allows the simulations to finish in reasonable time). Chapter 6, however, *varies* the $\gamma$ parameter extensively: we explain there how these values are calculated and what this controlled variability achieves.

Part of our proposed OAL framework, we complement the annotation decision heuristic with a data processing pipeline that automatically detects activity segments, makes informed annotation requests to the user and improves the model by propagating

back the annotations so that the activity model is updated. The overarching concept was illustrated in Fig. 2.1, whereby the annotation task is a coordination between a segmentation procedure and an annotation heuristic that obtains labels from the user. Starting with Fig. 4.3, we now present our solution pipeline that combines automated data preprocessing with user involvement in order to improve the activity model.

The **Sensor Array** includes the set of sensors which are continually monitored to infer user context. We focus on multiple sensors with a single sensing modality. This homogeneity in sensor data allows the operation of a single **Sliding Window** procedure over all sensor data streams to obtain a single stream of frames. **Feature Extraction** changes the representation of the data to a suitable one for machine learning. The resulting stream of feature vectors from the Feature Extraction stage follows dual processing. Firstly, the feature vectors are used in the **Segmentation** stage to estimate segment boundaries using the procedure detailed in Section 3.4.1. Secondly, the resulting feature vectors are used for **Classification**, where the model estimates the classification probabilities for known activities. The classification confidence is then used to decide whether a user annotation is needed in accordance with Eq. 4.1. If an annotation is needed, then the **User Notification** is invoked and the label provided by the user is used in the **Annotation** stage. Ultimately, the new annotation is used to by the **Model Improvement** stage to update the activity classifier. In this chapter we assume that the user has the technical means to provide annotations (examples of prior work about user provided annotations are given in Section 2.2). In contrast, in Chapter 5, we describe a mobile application for collecting user-provided annotations which was used in a user study.

### *General Simulation Procedure*

In this chapter, because we are analysing HAR datasets which are already collected and annotated, we simulate the interaction with the user in the annotation process. In particular, whenever an annotation is deemed necessary, the ground truth labels are revealed by the computer simulation environment and the activity model is retrained to account for the updated training set.

The model bootstrapping process is fully personalised by being done strictly for each

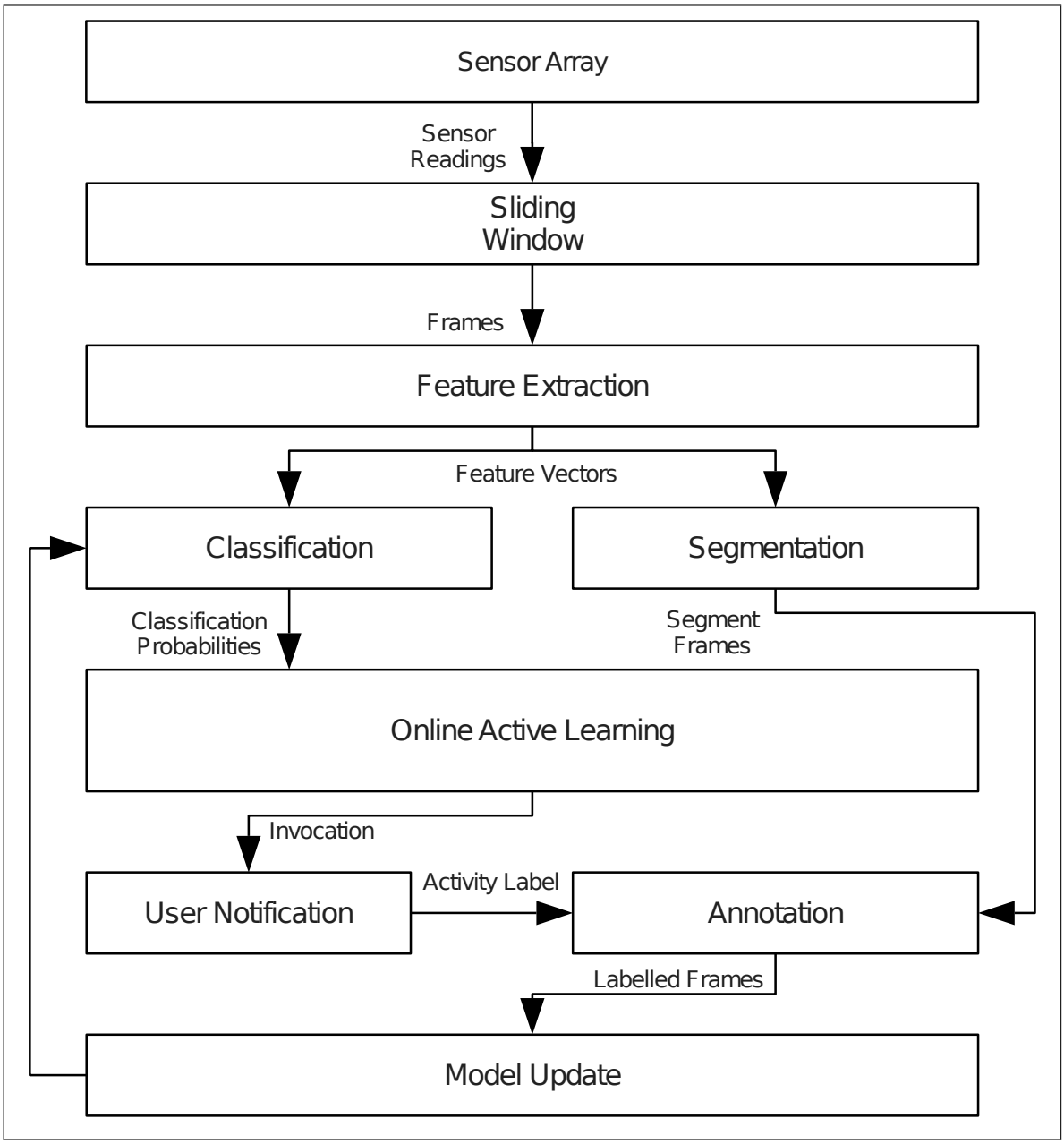



Figure 4.3: Interactive Annotation Pipeline.

subject independently (as all the datasets we consider here have data partitioned on a per-subject basis). That is, we repeat the process for every subject in the dataset and we only sample data segments collected for that subject only. Models are evaluated on a subject's own data, so we obtain a true reflection of how well the personalised model performs for the relevant subject.

We simulate annotating from a continuous stream of activities by maintaining a set of not-yet-annotated data and replaying data points from this set. Due to the limited size of the data, we replay activity segments several times until a target number of annotations is achieved. With the exception of an outlined portion of analysis in Section 4.4.4, segments which are annotated are permanently removed from the stream. This avoids duplication of data in the training set and makes model evaluation harsher and more realistic. Conversely, data segments which are not annotated are potentially "recycled", i.e. possibly re-sampled into the stream in the future.

The procedure to reach a decision of whether or not to request an annotation according to Eq. 4.1 is based on sampling using pseudo-random number generation, as illustrated in Algorithm 1.

**input :** $\gamma$ – the hyperparameter for the annotation heuristic in Eq. 4.1

$s_i$ – $i$th segment in the stream of activities

**output:** $d_{ask}$ – the decision to ask the user for an annotation

$p_{pred} = predict(s_i)$

$p_{conf} = max(p_{pred})$

$p_{ask} = exp(-\gamma \cdot p_{conf})$

$threshold = sample\_uniform\_from([0, 1])$ /* generate random threshold */

$d_{ask} = threshold < p_{ask}$

**Algorithm 1:** Simulating the annotation process.

## Non-Periodic Activities

In this section, we present the evaluation of the applicability of our OAL annotation method on the publicly available Opportunity dataset [38]. The dataset was previously described and analysed in Chapter 3. However, for the purposes of this chapter, we

diverge from that data processing and machine learning treatment and adopt a different approach aimed at exploiting the temporal structure of the activities. Because, in this chapter, we are interested in improving model recognition performance to a greater extent, we use a new machine learning pipeline which is able to work with a larger number of activity classes and which substantially increases the recognition accuracy scores.

### *Preprocessing and Segmentation*

In Chapter 3 we aggregated all 17 labels into seven labels and used a generic machine learning pipeline (sliding window, feature extraction, frame-based model builder) that yielded clear increases in performance. The choice of the machine learning pipeline was not based on maximising recognition accuracy, but it was rather based on replicating a very common sliding window-based classification analysis pattern for HAR, like, for example, Rebetez et al.[53].

However, in this chapter, we are motivated to obtain generally high classification scores and to improve them further using Online Active Learning. To this end, we no longer aggregate the labels, but, rather, we use the full vocabulary of 17 labels and, so, we evaluate how well the pipeline discriminates between all activities. Furthermore, the machine learning pipeline better accounts for the temporal dependencies which are the important characteristics of non-periodic activities.

We assume an ideal segmentation procedure – the existing ground truth was used to segment the data in the Opportunity dataset, which allowed for ideal segmentation. Accurate segmentation of non-periodic activities was shown to be possible, for example by [112–114, 116], but only if one already has a corpus of annotations to guide the segmentation process and this is in contradiction with our assumption of starting from zero knowledge. By using a perfect segmentation procedure, we better isolate and evaluate the effects of OAL on recognition accuracy. In Chapter 3 we applied the segmentation procedure from Section 3.4.1. However, for this machine learning pipeline, which relies on continuous segments (as opposed to sequences of frames in Chapter 3), this segmentation procedure is not appropriate. Segmentation procedures specialised for non-periodic activities have been documented in Section 2.4.3 (for example, Deng

and Tsui [112], Krishnan et al. [113], Junker et al. [114]), but these assume a pre-existing corpus of annotations, which is conflicting with our assumption of no initial annotations.

### *Learning Machinery*

We do not extract features from the acceleration signals, but instead classify entire activities based on their acceleration timeseries. We use k-Nearest Neighbours (kNN) [152] model to distinguish between activities and, as a dissimilarity measure of acceleration timeseries, we use Dynamic Time Warping (DTW) [153].

DTW is a method for quantifying dissimilarity between timeseries. Intuitively, DTW finds the associations between all points in one timeseries to all the points in the other timeseries according to shape characteristics (e.g. peaks are matched to peaks, troughs are matched to troughs and intermediary values to the the closest intermediary values). Shape mismatches are quantified in a distance metric which we use as a measure of dissimilarity. DTW-based classification approaches for HAR are not new; for example Muscillo et al. [106] show that physical activities recorded with an accelerometer can be reliably recognised by a DTW-based classifier. In our simulations, we used the *dtw*[2] [153] R package as a DTW implementation.

kNN is used as follows: We use the data from the five accelerometer positions used in Chapter 3 (*upper right arm*, *lower right arm*, *upper left arm*, *lower left arm* and *back*). Model building, in the case of kNN, simply means storing all individual labelled segments, called *templates*, in a *template database*. At classification time, the segment which needs to be classified is compared with all templates (a linear search within the template database). There exist template search methods [120], such as exploring a KD Tree [154], which may reduce the complexity of the search step, but, in general, these do not improve upon the correctness of linear search. The metric of comparison is the resulting DTW dissimilarity between the current segment and the template. Finally, in order to fulfil the classification, the closest $k$ templates to the segment give a mean vote on the label probabilities. kNN has been used previously in HAR with good classification accuracy results, for example, by Bicocchi et al. [155].

[2] `http://cran.r-project.org/web/packages/dtw/index.html` Accessed 12.01.2015

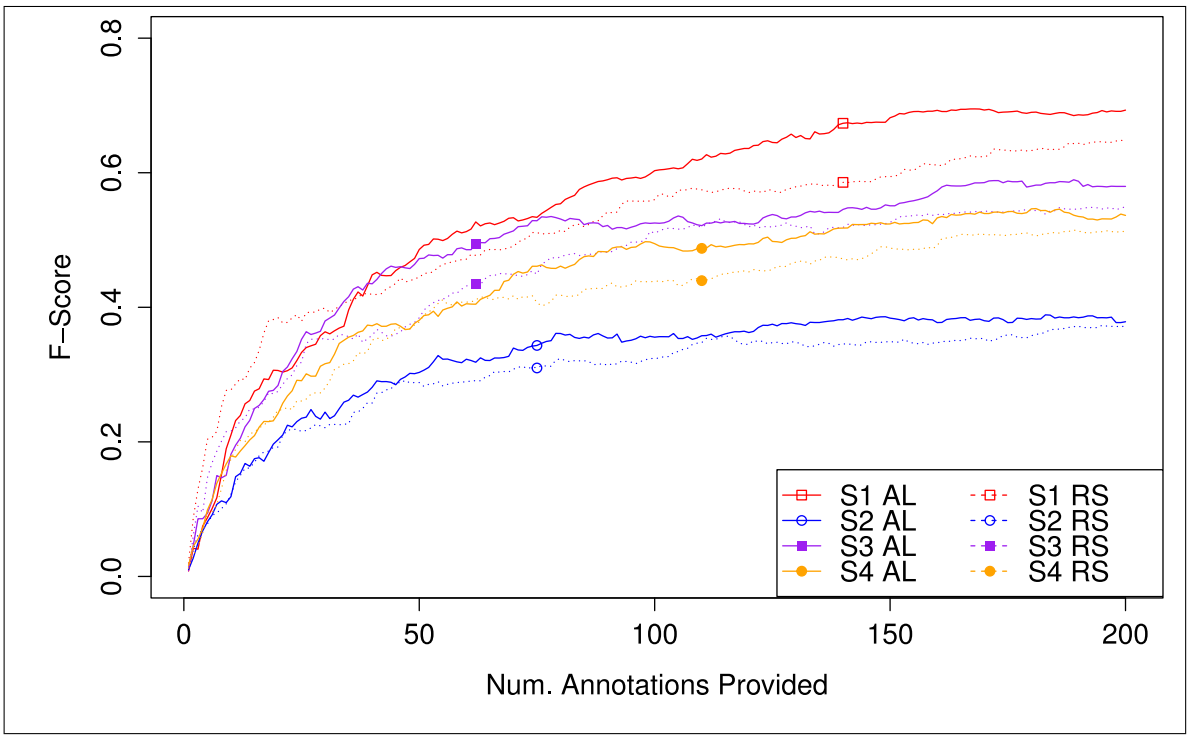



Figure 4.4: Learning Curve for Opportunity; Legend: S1 – Subject 1, AL – Online Active Learning, RS – Random Selection.

We take advantage of the standard train/test split in the Opportunity dataset and we apply the same train/test evaluation procedure we used throughout Chapter 3.

### *Results*

We have replayed segments and used them as input to the annotation method, as explained in Section 4.2.2. Segments that were marked for annotation by Eq. 4.1 were included in the template database of the kNN classifier. Conversely, segments that were not annotated at a point in time, were made available again for future annotation.

As each new segment was annotated, we evaluated the performance of the model against the standard testing set of the corresponding participant and obtained a learning curve. We repeated the annotation process 10 times and averaged the results for each participant.

Fig. 4.4 illustrates for every subject the performances accrued from Online Active Learning and Random Selection, while Fig. 4.5 shows the same performances averaged over all participants. Online Active Learning was stopped after accumulating 200

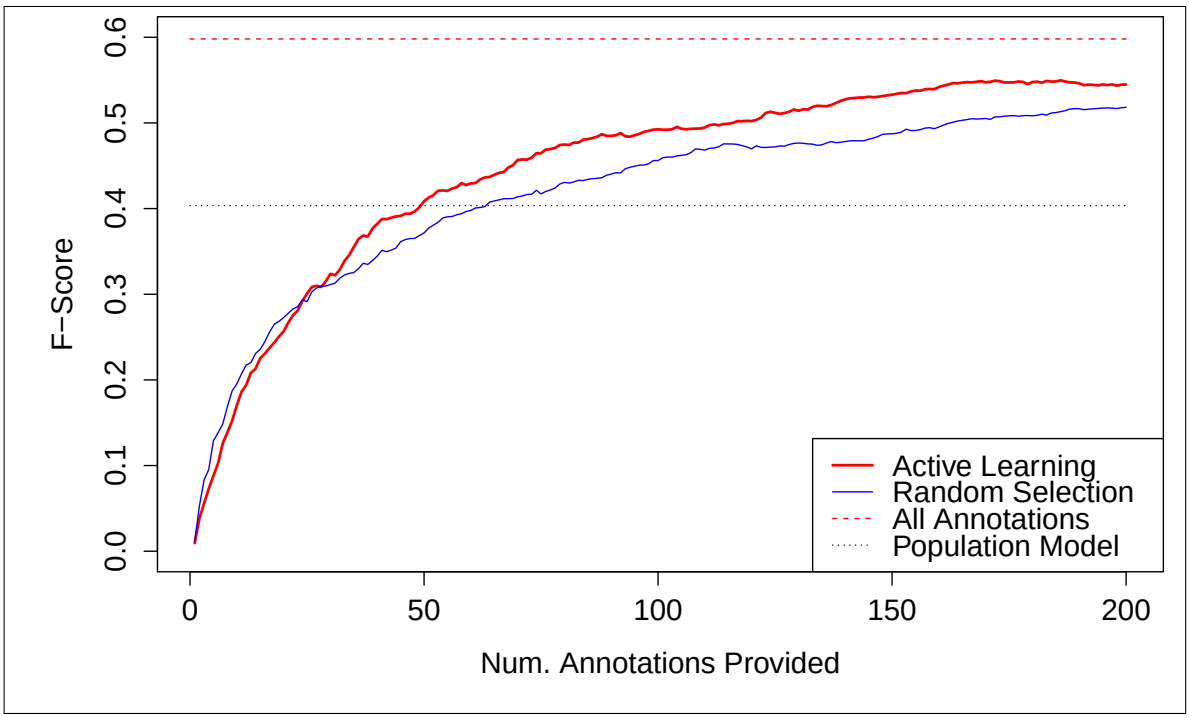



Figure 4.5: Active Learning versus Random Sampling on Opportunity. Average performances.

annotations because, after this point, most segments would be classified with very high classification confidence. This not only slowed down simulations, but, because of the constantly high classification confidence, segments could not be differentiated, so the annotation mechanism effectively became Random Selection.

Fig. 4.5 also includes two additional accuracy levels. Firstly, we included the average accuracy of all four personalised models (All Annotations – red dashed line), i.e. one model per participant, trained with all available annotations for the corresponding participant. This serves as an upper performance baseline for OAL and RS, meaning that, both these methods will eventually reach this accuracy when annotating all data. The marked difference between OAL and RS is that OAL comes closer this accuracy score sooner than RS. Secondly, we underline the importance of personalised models by including the performance level of a non-personalised population-wide model (Population Model – black dotted line) using a *leave-one-subject-out cross-validation*. This means that, for each user, we trained a model using the data from the other users and evaluated the predictive performance of the resulting model using the initial user's testing set. The results show that greater performance results can be obtained by using personalised data instead of relying exclusively on non-personalised annotations.

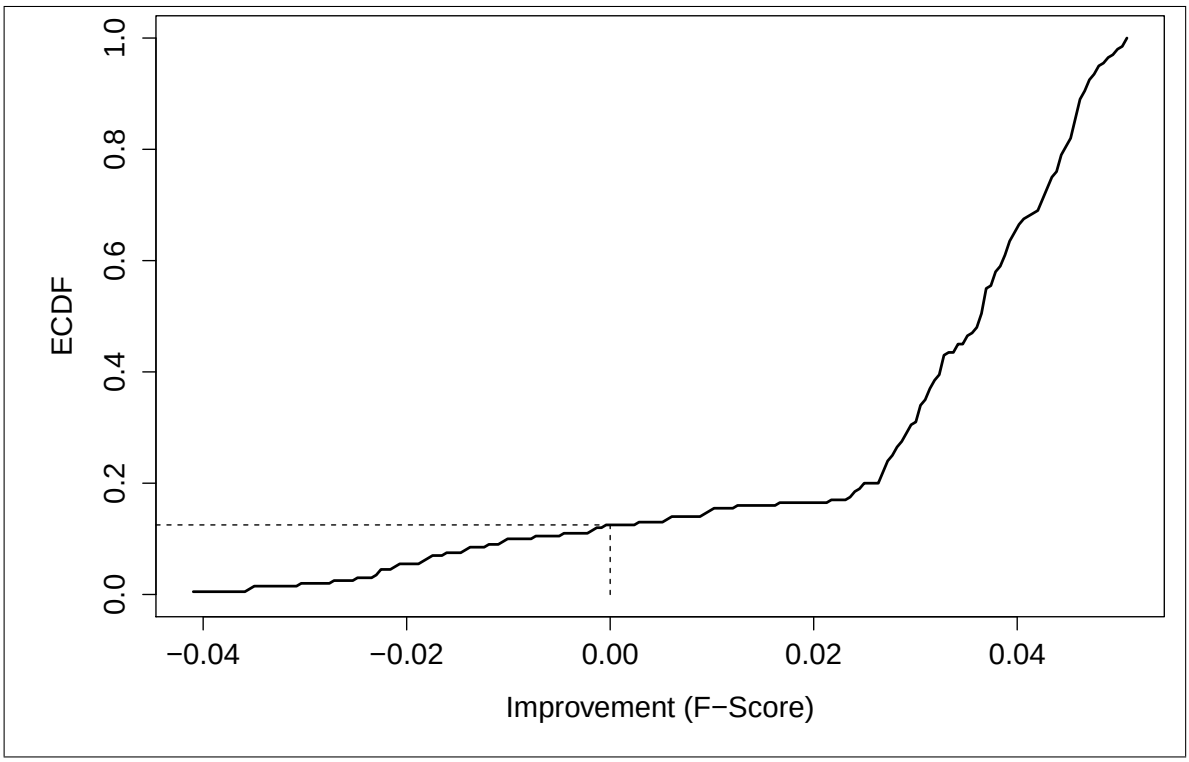



Figure 4.6: Performance gain on Opportunity due to Active Learning. Empirical Cumulative Distribution Function.

The performance gains of OAL over RS are illustrated alternatively in Fig. 4.6. It depicts the Empirical Cumulative Distribution Function (ECDF) of the difference in recognition performance between Online Active Learning and Random Sampling, for every size of the training set. We conclude that 87.5% of the points on the averaged learning curve for Active Learning exhibit performance gains of at most 5% over the corresponding points on the Random Sampling learning curve.

Figs. 4.4, 4.5 and 4.6 highlighted the *vertical* differences between OAL and RS – the differences of performance between the two methods when the training sets were the same size. Figs. 4.7-4.10 illustrate an alternative interpretation of the performance gains of OAL over RS. Specifically, they show the *horizontal* differences between OAL and RS – the differences in the number of annotations required to reach a certain performance level.

For each subject, the plots were obtained as follows: The x-axis values $n_{OAL} \in \{1, 2, ..., 200\}$ represent the sizes of the training sets obtained with OAL. For each training set size $n_{OAL}$, the performance of the model $P_{n_{OAL}}$ is computed. The y-axis value corresponding to $n_{OAL}$ is the size of the training set obtained with RS ($n_{RS}$) such

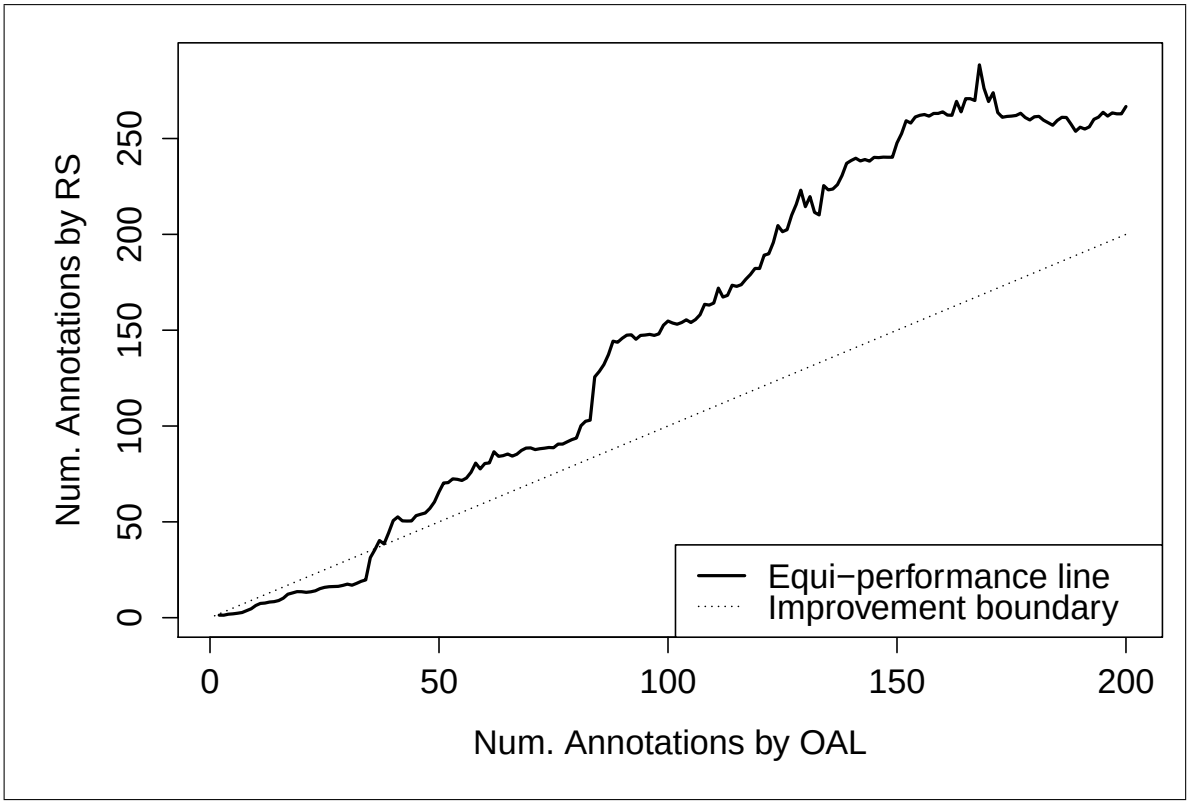



Figure 4.7: Comparison of annotation effort; Opportunity Subject 1.

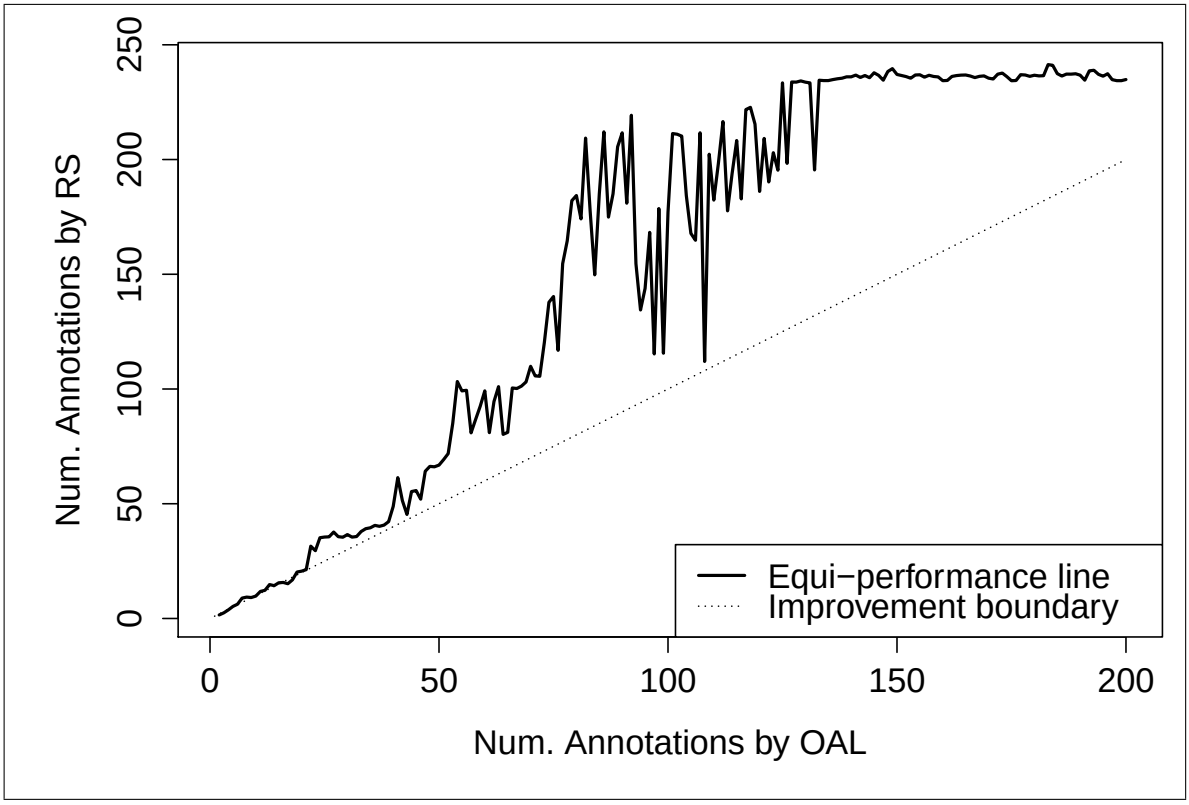



Figure 4.8: Comparison of annotation effort; Opportunity Subject 2.

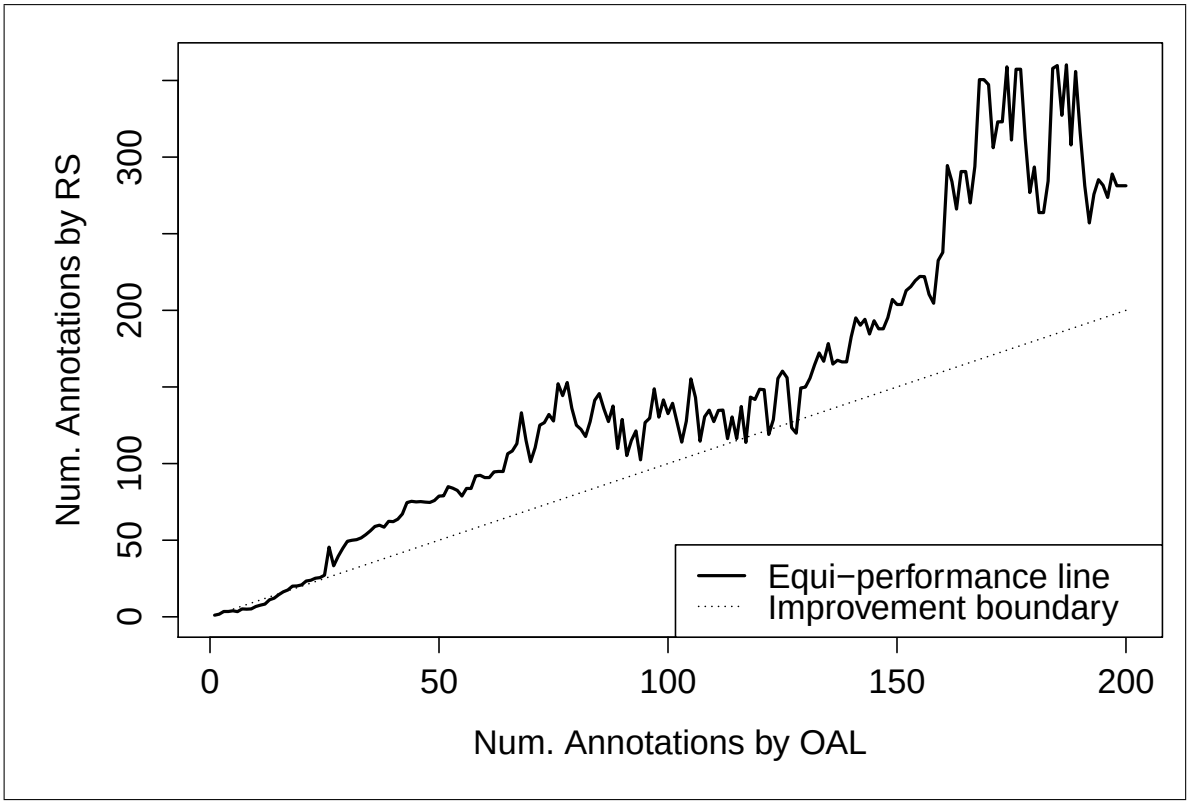



Figure 4.9: Comparison of annotation effort; Opportunity Subject 3.

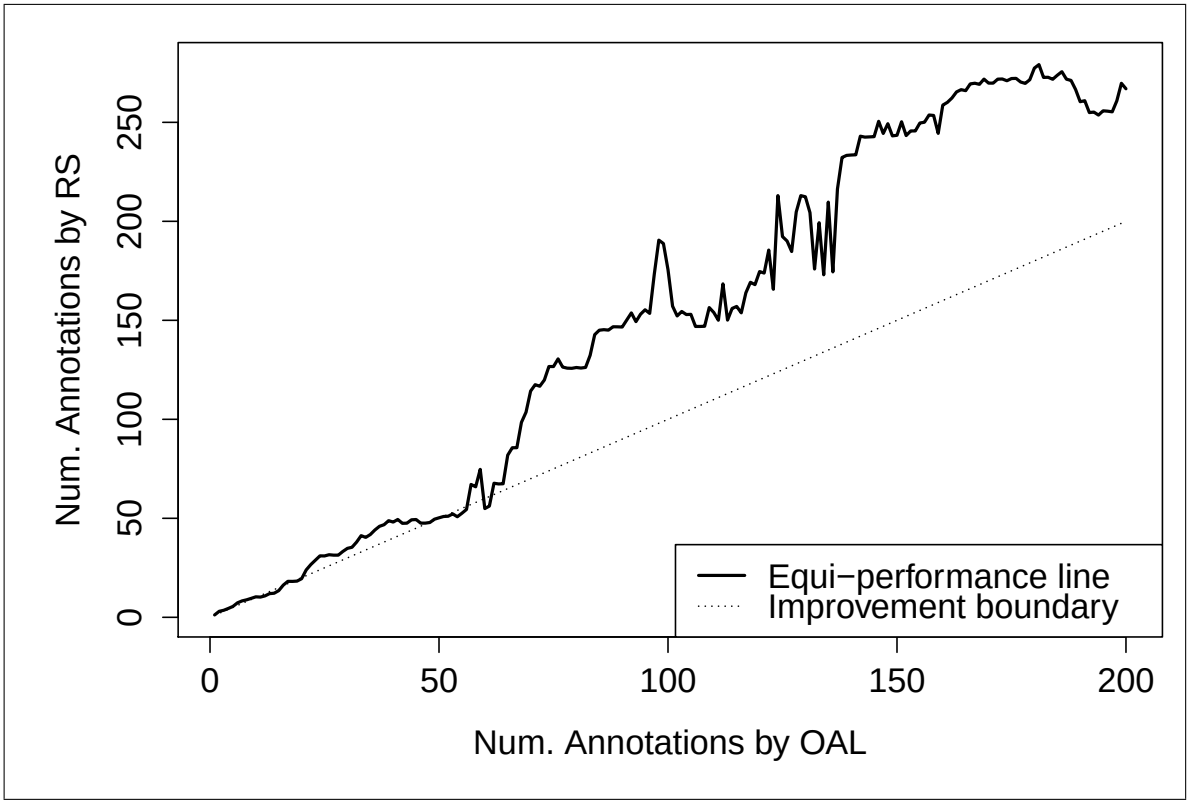



Figure 4.10: Comparison of annotation effort; Opportunity Subject 4.

| Subject | Points Above Line (%) | Mean (%) | Max (%) |
|---|---|---|---|
| Subject 1 | 164 (82%) | 47.5 (12.6%) | 120.4 (42.1%) |
| Subject 2 | 192 (96%) | 53.7 (29.7%) | 127.3 (60.8%) |
| Subject 3 | 183 (91.5%) | 47.9 (24.9%) | 184.8 (52%) |
| Subject 4 | 184 (92%) | 48.8 (26.1%) | 104.5 (48.5%) |
| Average | 180.7 (90.3%) | 49.5 | 134.2 |

Table 4.1: Summary of Opportunity Equi-Performance Lines

that the performance $P_{n_{RS}}$ of the model constructed with $n_{RS}$ is equal to $P_{n_{OAL}}$. We introduce a concept which we call the *Equi-performance line* between OAL and RS – it contrasts the annotation effort required between OAL and RS to achieve the same level of activity recognition accuracy. The hypothetical line $n_{RS} = n_{OAL}$ marks the improvement boundary (black dotted line) such that equi-performance points above it designate OAL training set sizes smaller than RS training set sizes. These points correspond to the cases when OAL reaches a performance level with fewer annotations than RS.

Figs. 4.7-4.10 show that the majority of equi-performance points lie above the improvement boundary which means that OAL-based annotation reduced user annotation effort, as summarised in Table 4.1. The Points Above Line column represents the number of equi-performance points above the improvement line. The Mean and Max columns represent the mean and, correspondingly, the maximum number of annotations the user was spared from providing by using OAL instead of RS.

In summary, for non-periodic activities, our simulation results show that, as an annotation method, Online Active Learning outperforms Random Selection not only in terms of objective accuracy scores (87.5% of the points on the learning curve are improved by up to 5%), but also in terms of the number of annotations required to reach a certain level of accuracy (user annotation effort is reduced by up to 60.8% when using OAL instead of RS).

For the majority of points on the learning curves, Online Active Learning registers performance gains over Random Sampling. We conclude that, even for a technically challenging dataset such as Opportunity, our method accelerates the bootstrapping process of activity models.

# Periodic Activities

In this section, we apply our Online Active Learning annotation method on the publicly available USC-HAD and PAMAP periodic activities datasets. The USC-HAD dataset consists of movement data collected about 12 activity classes from 14 participants. The PAMAP dataset consists of movement data collected about 12 activity classes from 9 subjects. The activities in both datasets are periodic, which is typical for healthcare and fitness applications.

## *Preprocessing*

We used only the tri-axial accelerometer data to infer activities. As noted in Chapter 3, this is common practice in activity recognition.

We applied a sliding window procedure over the acceleration data timeseries. The length of the sliding window was 5 seconds and there was no overlap between adjacent windows. For every window we extracted feature vectors characterised by the following 9 features: *X axis mean*, *Y axis mean*, *Z axis mean*, *X axis variance*, *Y axis variance*, *Z axis variance*, *X and Y axis correlation*, *Y and Z axis correlation*, *Z and X axis correlation*. This resulting feature-based representation of the data is suitable for human activity recognition via machine learning.

## *Recognition Performance Evaluation*

The USC-HAD and PAMAP datasets contain relatively limited amounts of data for each participant (at the low extreme, USC-HAD contains one of the participants with 247 labelled frames, while PAMAP contains one participant with 282 labelled frames). Two typical performance evaluation procedures are k-fold cross validation [120] and evaluating against a fixed test set [120]. In our incremental scenario, which involves repeated model evaluation across a large range of training set sizes, neither of these two procedures are suitable because none of the procedures uses all available testing data. In order to robustly evaluate the performance of an activity model across such a large spectrum, we have designed a performance evaluation procedure that makes use of all available data, at all times.

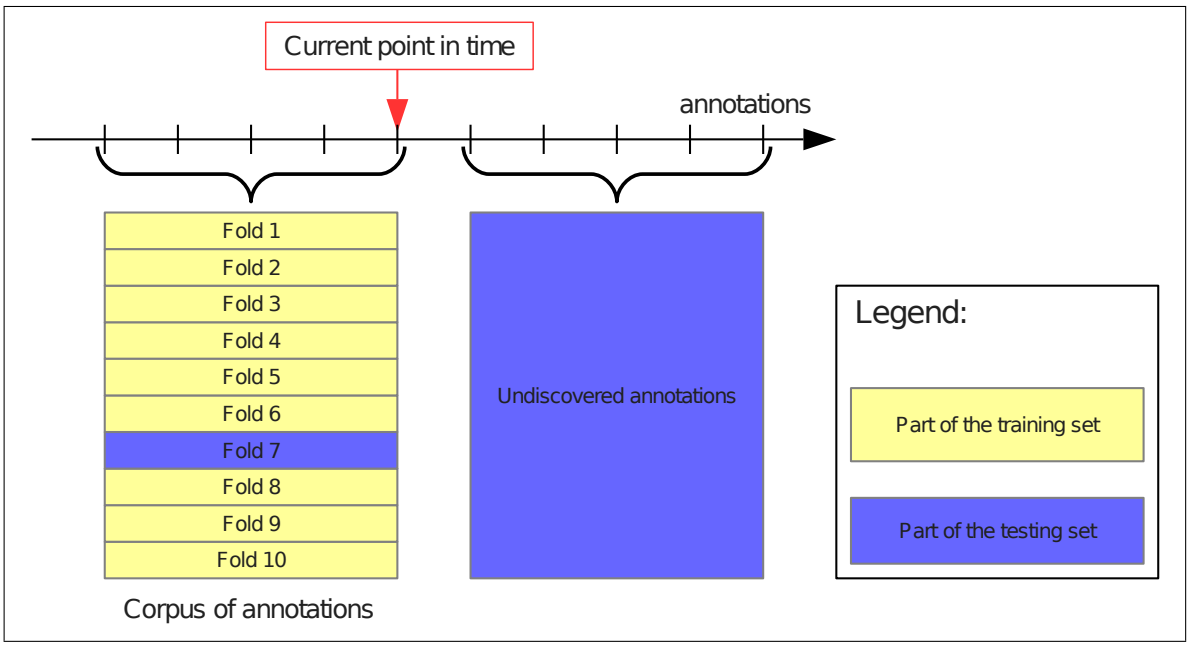



Figure 4.11: Model evaluation procedure; 7th round of cross-validation.

On the one hand, *at the beginning* of the bootstrapping process, when the size of the corpus of annotations is small, there is a relatively large amount of data which is not used for model training (the annotations which have not yet been discovered). In fixed testing set evaluation, this latter proportion of data is transformed into a relatively large test set which is arguably representative of the domain. However, k-fold cross-validation (the alternative evaluation procedure), at any one time, uses $\frac{k-1}{k}$ of the annotated data for model training and, so, only $\frac{1}{k}$ of the annotated data is available for testing the model. This latter testing set is relatively small and unrepresentative at the beginning of the bootstrapping process, so the resulting model accuracy scores are not reliable. Therefore, at the beginning of learning, fixed testing set is more robust than k-fold cross-validation.

On the other hand, *towards the end of the bootstrapping process*, the size of the corpus of annotations is large and the usefulness of the two evaluation procedures is now reversed. Only a relatively small amount of unused data is left (a relatively small number of annotations remain undiscovered) and, so, evaluating against a fixed testing set is unrepresentative. With a large corpus of annotations, however, k-fold cross-validation is now the more robust performance evaluation procedure.

To account for this volatility across training set sizes, we propose a hybrid between k-

fold cross validation and fixed testing set validation, which is exemplified in Fig. 4.11. Our proposed evaluation method blends the two very common evaluation procedures and gradually shifts focus from fixed testing set evaluation, at the beginning of the learning process, to k-fold cross-validation as new annotations are collected towards the end of the learning process. We use k-fold cross-validation and we augment each testing fold with any data which is unused in the training set. This guarantees that the size of the testing set is maximised (in general, our procedure uses more testing data than either k-fold cross-validation or fixed testing set evaluation), thereby making evaluation more stringent and realistic. Our procedure also but also ensures that there is no overlap between training and testing sets at any point during the evaluation.

### *Learning Machinery*

As an activity model builder we used a Bootstrap Aggregator [156] with 30 Naive Bayes [157] base learners. In our analysis, this model builder yielded superior performance over others commonly used in HAR (logistic regression, decision trees, k-Nearest Neighbours).

The default implementation of the Bootstrap Aggregator does not allow for *updating* (incremental learning), but instead requires complete re-building of the model with any arrival of new data. While this is a limitation in terms of applicability to online learning scenarios, we show later in Section 5.3.2 that the model builder can be modified to allow updating. This is a critical modification we bring forward in order to deploy a fully Online Active Learning pipeline in a realistic participants-based context in Chapter 5.

### *Results*

We evaluated our proposed annotation method under three different conditions. Firstly, we generate a stream of segments consisting of single frames. This scenario looks at what happens if there is no segmentation procedure and hence no segmentation noise. Annotation requests are directed at individual frames, which are automatically generated by the sliding window procedure. The results isolate the performance difference

between two annotation decision heuristics, Online Active Learning and its corresponding naive correspondent – Random Selection.

Secondly, we generate a stream of segments, each consisting of several frames and we operate the automatic segmentation procedure detailed in Section 3.4.1. The output from the segmentation procedure is a sequence of segments. A segment is associated with a single label, which is simulated to be provided by the user and which is extended to all the frames in the segment. However, because the segment delineation does not always match the ideal boundaries, some segments may contain label noise – frames from more than one activity which are assigned the label. In this scenario, we report the results of how imperfect segmentation affects the model bootstrapping process in addition to the effects of the annotation decision heuristic.

Finally, we generate a stream of segments, but we additionally alter the activity class distribution in the stream. Specifically, we simulate a more sedentary lifestyle by oversampling sedentary activities. In this scenario, we focus on how this class imbalance affects the annotation process in terms of not only model accuracy, but also class distribution in the training set. The results in this final case reveal how Online Active Learning reduces user involvement by distributing annotation requests more evenly (compared to Random Selection) across activity classes.

#### First Scenario: 1-Frame Segments

We begin by evaluating our annotation method on single-frame segments. This allows us to emphasise the merits of OAL by controlling every individual frame which is selected for annotation. Fig. 4.12 shows that our annotation method improves the recognition performance over Random Selection on the USC-HAD dataset. Specifically, for 95.5% of the points on the learning curve, the difference in F-Score between OAL and RS is positive and peaks at 7.4%. Fig. 4.12 also shows that unpersonalised models (the Population Model line) perform substantially worse than fully personalised ones (the Online Active Learning and Random Selection curves).

The gain in performance illustrated in Fig. 4.12 was obtained by parametrising Eq. 4.1 with $\gamma = 6$. In our analysis, this was a relatively high value for the parameter and sustained highly informed accumulation of training data. Fig. 4.13 contrasts the

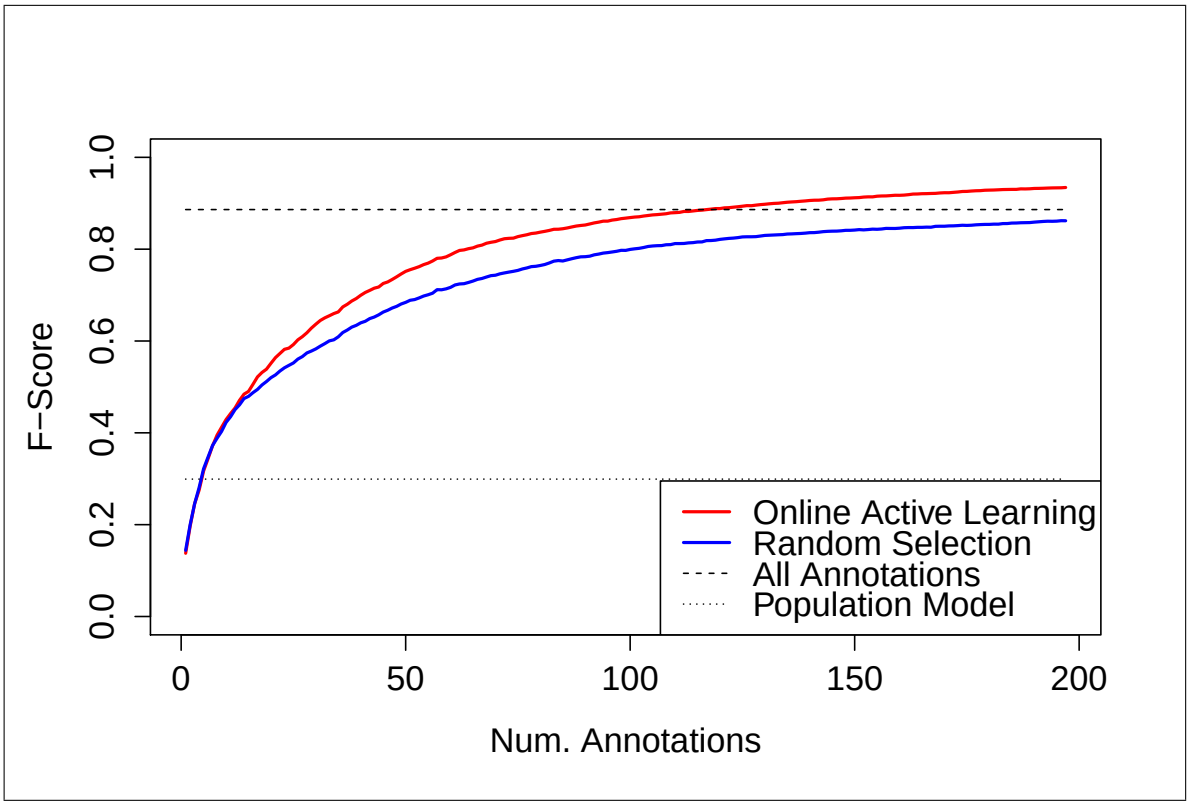



Figure 4.12: Active Learning versus Random Sampling on USC-HAD. Average performance for 1-frame segments.

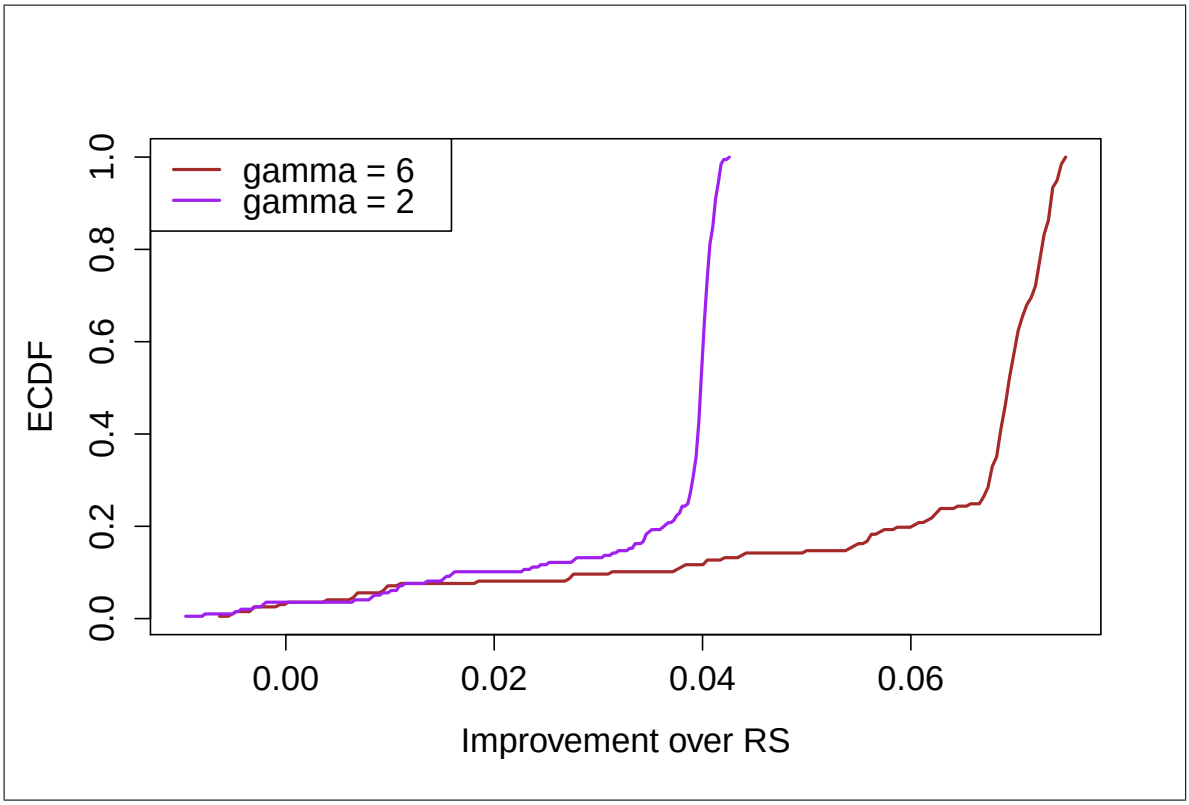



Figure 4.13: Active Learning versus Random Sampling on USC-HAD. Performance ECDF comparison between $\gamma = 2$ and $\gamma = 6$.

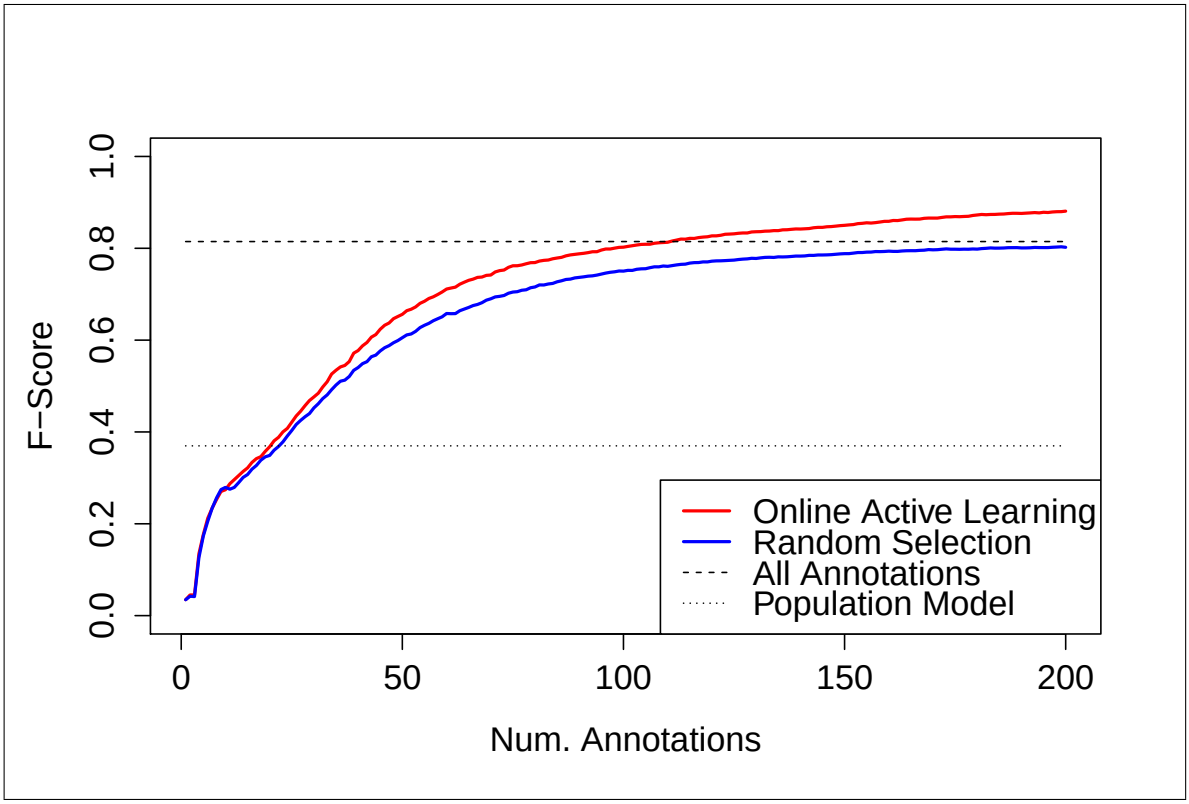



Figure 4.14: Active Learning versus Random Sampling on PAMAP. Average performance for 1-frame segments.

distribution of performance gains over Random Selection (F-Score of OAL minus F-Score of RS for all points on the learning curve) for $\gamma = 6$ and, separately, for $\gamma = 2$. Higher values for the $\gamma$ parameter make the annotation process more informed and the results show that the performance gains of Online Active Learning over Random Selection can be reduced if low values for $\gamma$ are used.

Fig. 4.12 additionally illustrates a known phenomenon, namely that Online Active Learning may reach a peak performance and, if training is continued with additional annotated data beyond this point, the performance will gradually decrease. As noted in Chapter 2, this phenomenon was also observed by Zhu et al. [133], Vlachos [134] or Laws and Schätze [135]. In our particular case, this is evident by the performance of Online Active Learning which overshoots the performance obtained by k-fold cross-validation on the entire set of annotations (which is ultimately obtained when all annotations in the dataset are discovered).

For the PAMAP dataset, the performance contrast between OAL and RS is pictured in Fig. 4.14 which shows that, for 95.5% of the points on the learning curve, OAL

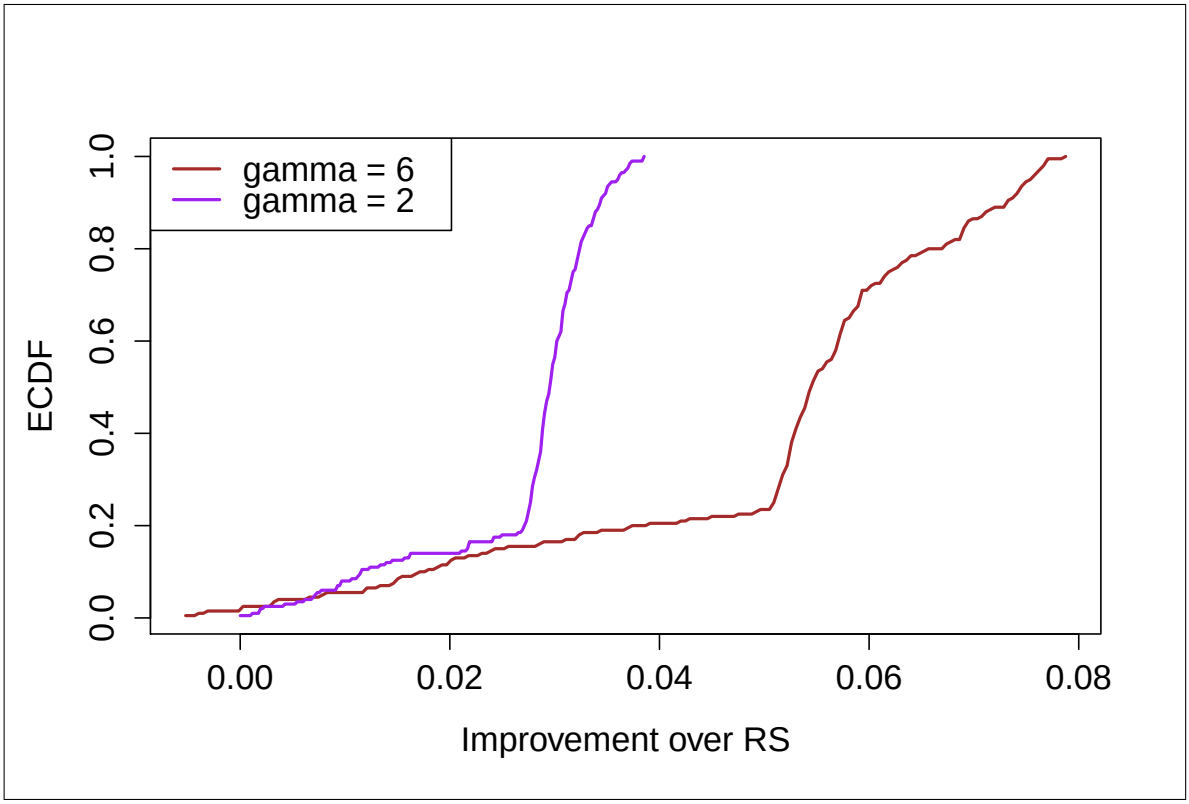



Figure 4.15: Active Learning versus Random Sampling on PAMAP. Performance ECDF comparison between $\gamma = 2$ and $\gamma = 6$.

outperforms RS by up to 7.8% in terms of F-Score. The effect of the $\gamma$ parameter is illustrated in Fig. 4.15. Similar to the USC-HAD dataset, the higher the value of the $\gamma$ parameter, the greater the overall improvement in F-Score of OAL over RS.

Overall, when annotating 1-frame segments of periodic activities, our results show HAR model F-Score improvements of Online Active Learning over Random Selection by up to 7.4% for the USC-HAD dataset and 7.8% for the PAMAP dataset. Additionally, we have also shown that higher values of the $\gamma$ parameter result in greater performance gains than smaller values.

### Second Scenario: Imperfect Segmentation

While noise-free segments are always preferable, in a realistic deployment it is not clear how these can be consistently detected. 1-frame segments, which are automatically detected by a sliding window procedure, can be used (as was done in our previous set of analyses), but these segments are very short, so the user's annotation effort is inefficiently used. Instead, in the following, we investigate what happens when anno-

tating longer segments – their constituent frames are annotated with a single label. We present results for periodic activities when we integrate the automatic segmentation procedure described in Section 3.4.1. This is a realistic best-effort activity stream segmentation method, but, as explained, the segmentation procedure may unintentionally introduce noise in the training set.

We generate a stream of activity data by continually concatenating *ground truth* segments of 3-6 frames to the end of the activity stream[3]. At the same time we operate the automatic segmentation procedure detailed in Chapter 3 and discover *estimated* segments, which are annotated. The estimated segments boundaries and ground truth segments boundaries may not align perfectly and, so, a degree of label noise may be introduced.

As new segments are detected we decide whether or not to annotate these. If an annotation is requested, we simulate receiving a response to the annotation request by calculating the *mode* of the frame labels[4] in the detected segments (as was done in Section 3.4) and using this as the segment label.

Fig. 4.16 illustrates the impact imperfect segmentation has on recognition accuracy. Due to the resulting label noise, general recognition performance drops – both the OAL and RS learning curves are lowered with respect to the cases corresponding to ideal segmentation. Nonetheless, for 92.5% of the points on the learning curve, the difference of performance between OAL and RS is positive and up to 8%. Additionally, Fig. 4.17 contrasts Figs. 4.12 and 4.16 and shows that annotating whole segments at once, although at the sacrifice of asymptotic F-Score, converts user involvement (number of provided annotations) more quickly into F-Score gains.

We have included the results for Online Active Learning with automatic segmentation for the PAMAP dataset in Fig. 4.18. These exhibit clear improvement of up to 8.5% for 92.5% of the points on the learning curve and reinforce our conclusion that, for periodic activities, Online Active Learning outperforms Random Selection. Similarly for the PAMAP dataset, Fig. 4.19 shows that annotating whole segments results in faster performance gains than when annotating single-frame segments, even though

[3]The length of the segments is limited due to the data size limitations in the datasets.
[4]The predominant label in the segment.

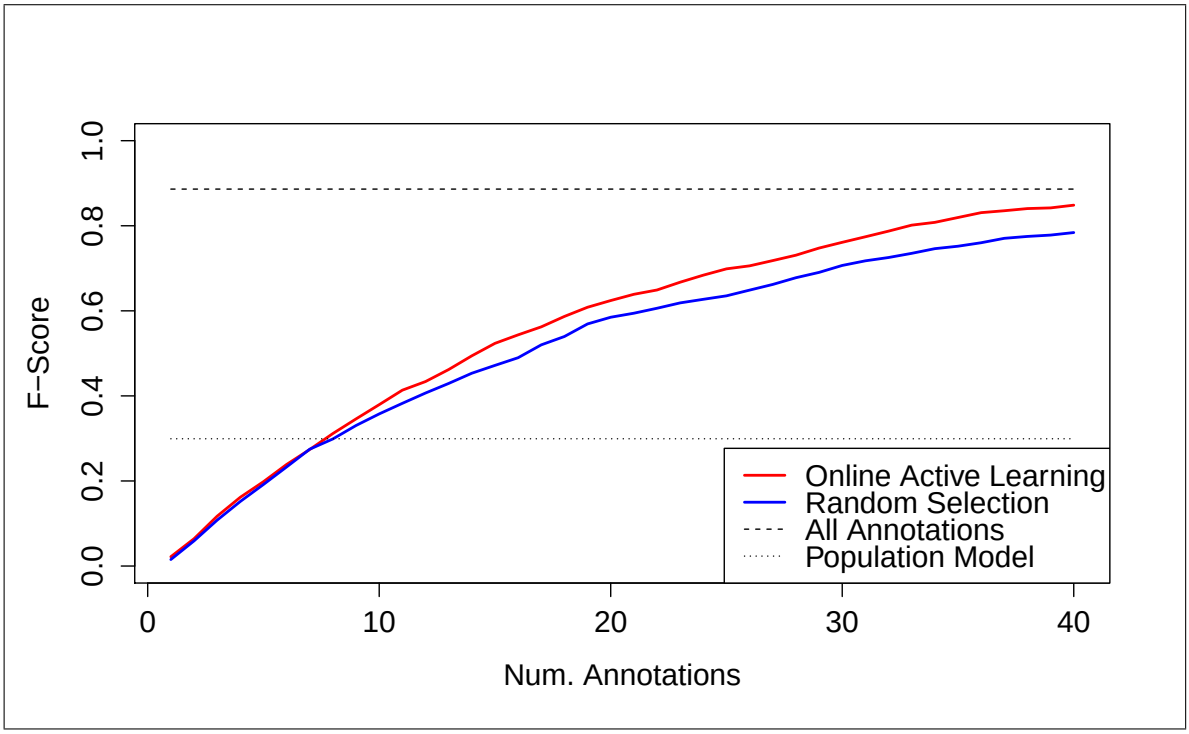



Figure 4.16: Active Learning versus Random Sampling on USC-HAD. Average performance for automatically delineated segments.

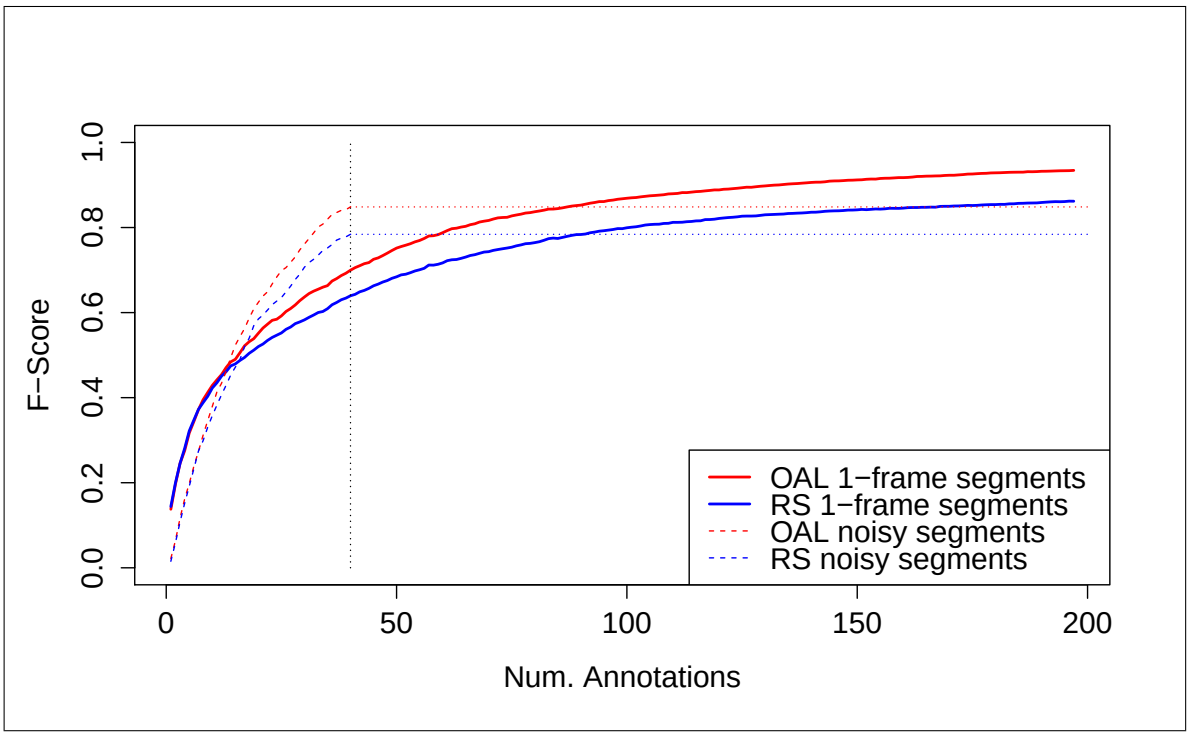



Figure 4.17: Active Learning versus Random Sampling on USC-HAD. Performance contrast between 1-frame segments and imperfect segments.

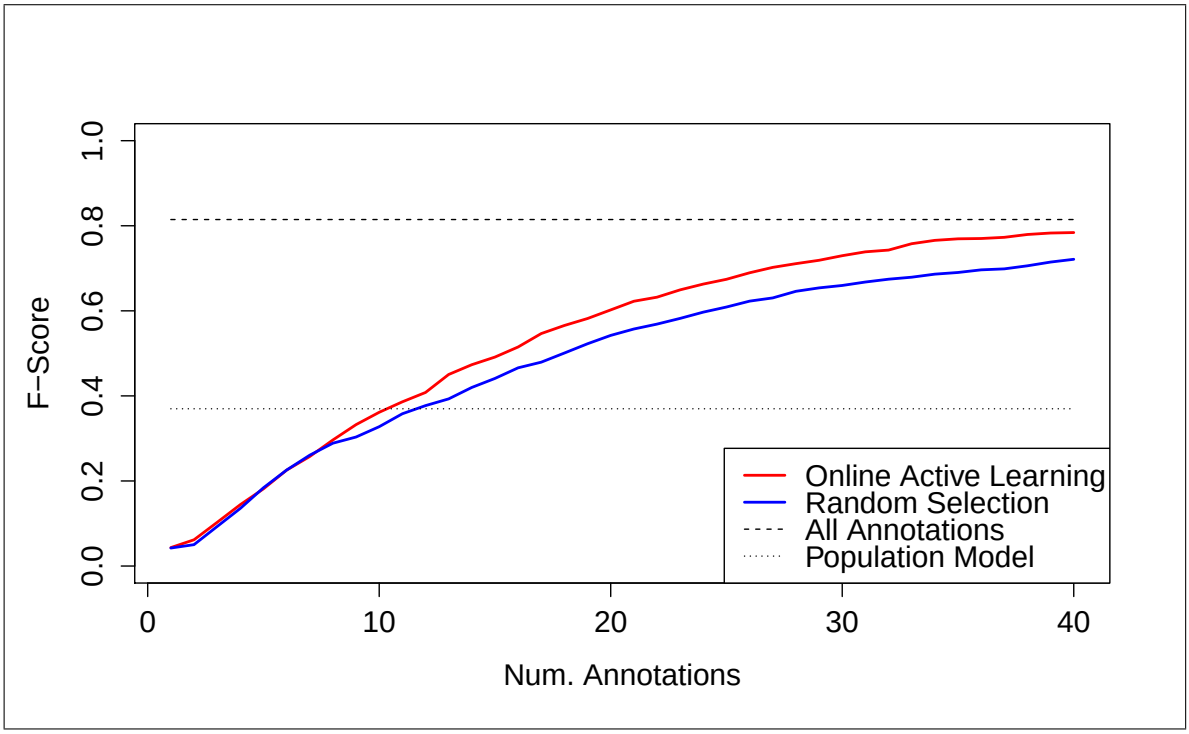



Figure 4.18: Active Learning versus Random Sampling on PAMAP. Average performance for automatically delineated segments.

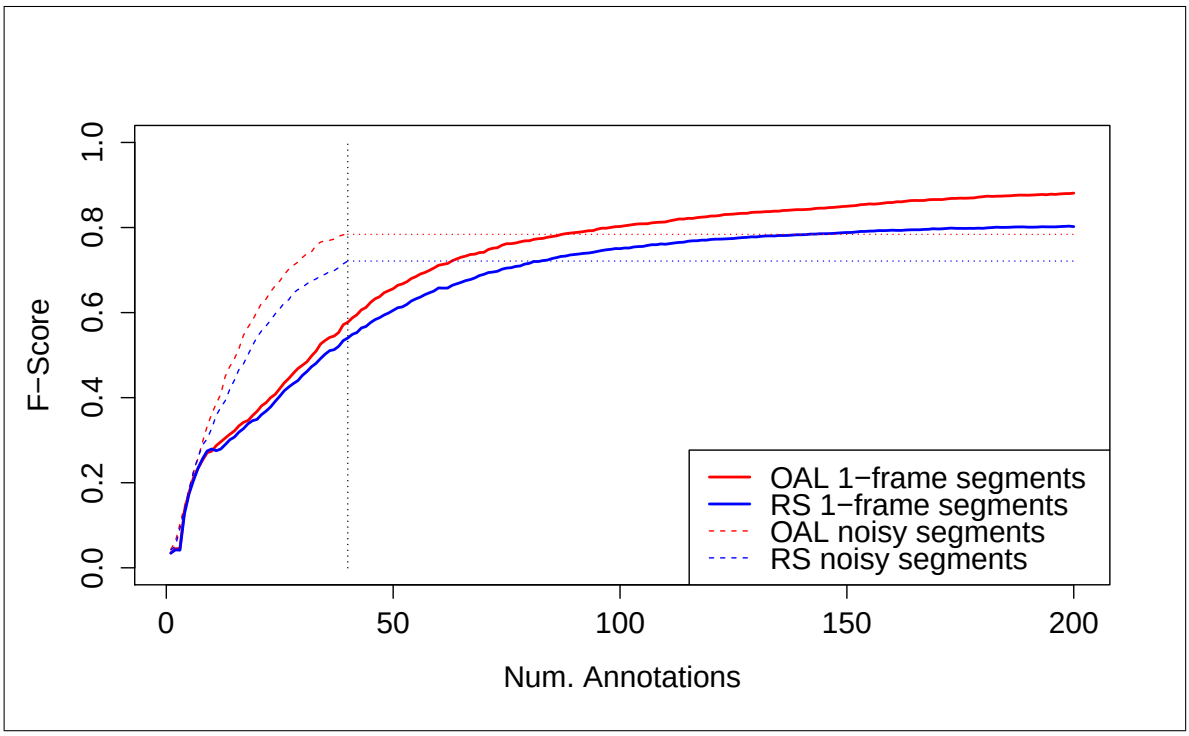



Figure 4.19: Active Learning versus Random Sampling on PAMAP. Performance contrast between 1-frame segments and imperfect segments.

some end accuracy is lost due to segmentation noise.

The reduction in performance is primarily attributed to label noise arising from imperfect segmentation. Segmentation introduces a degree of label noise because of the proposed annotation mechanism which assumes that a segment can have only one label (the one provided by the user) which is distributed to all the frames constituting the segment.

**Third Scenario: Long Activity Sequences**

In this section, we change the simulation conditions by no longer removing segments from the stream of activities when they are annotated. Instead, the segments are periodically replayed for potential annotation. By removing the previous data replay restrictions, we can get a better understanding of (1) how the performance gap between OAL and RS can be affected by label imbalances in the sequence of activities and (2) what activity labels the annotation process is going to favour.

For each user we simulate the monitoring of long activity sequences – we repeatedly replay a fixed sequence of activity segments sampled from the dataset. We consider two scenarios. In the first scenario, we split the data in perfectly delineated segments and replay these in the same order repeatedly[5]. Each segment is then presented to the OAL heuristic which may or may not annotate it. In the second scenario, we additionally duplicate the number of sedentary activities in the day by a factor of 10. We train the models regardless of the duplication in the training set, but, in order to fairly quantify the effects on model performance, all duplication is removed from the training set (this was the case throughout all analyses in this chapter). We perform this process only for the PAMAP dataset because the activity instances are roughly evenly distributed among the activity classes and only two of the 12 activities can be considered sedentary (*lie* and *sit*). We could not create equivalent experimental conditions with the USC-HAD dataset because it is not clear whether some activities are sedentary or non-sedentary.

[5]The previous results demonstrate the merits of imperfect segmentation. However, for the purpose of analysing the label distribution of the annotated data, not introducing label noise better isolates the effects.

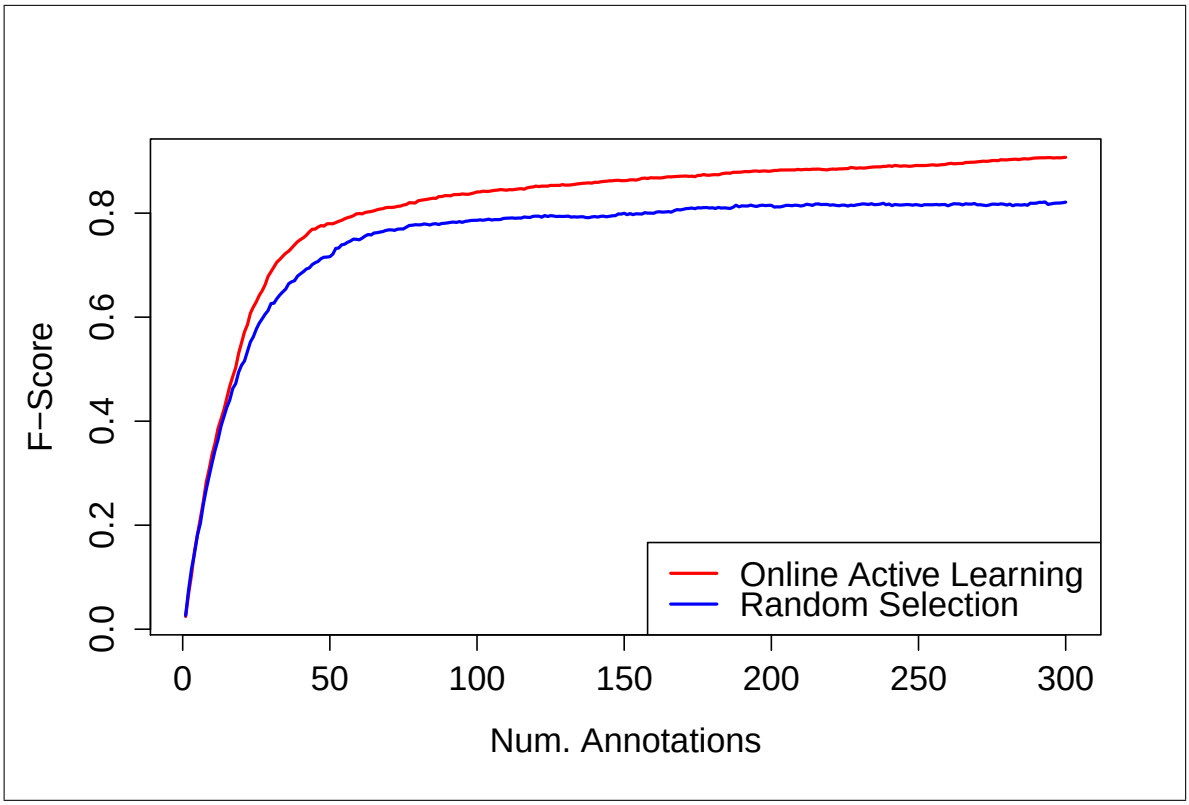



Figure 4.20: Active Learning versus Random Sampling on PAMAP. Average performance for balanced activity classes (initial class distribution).

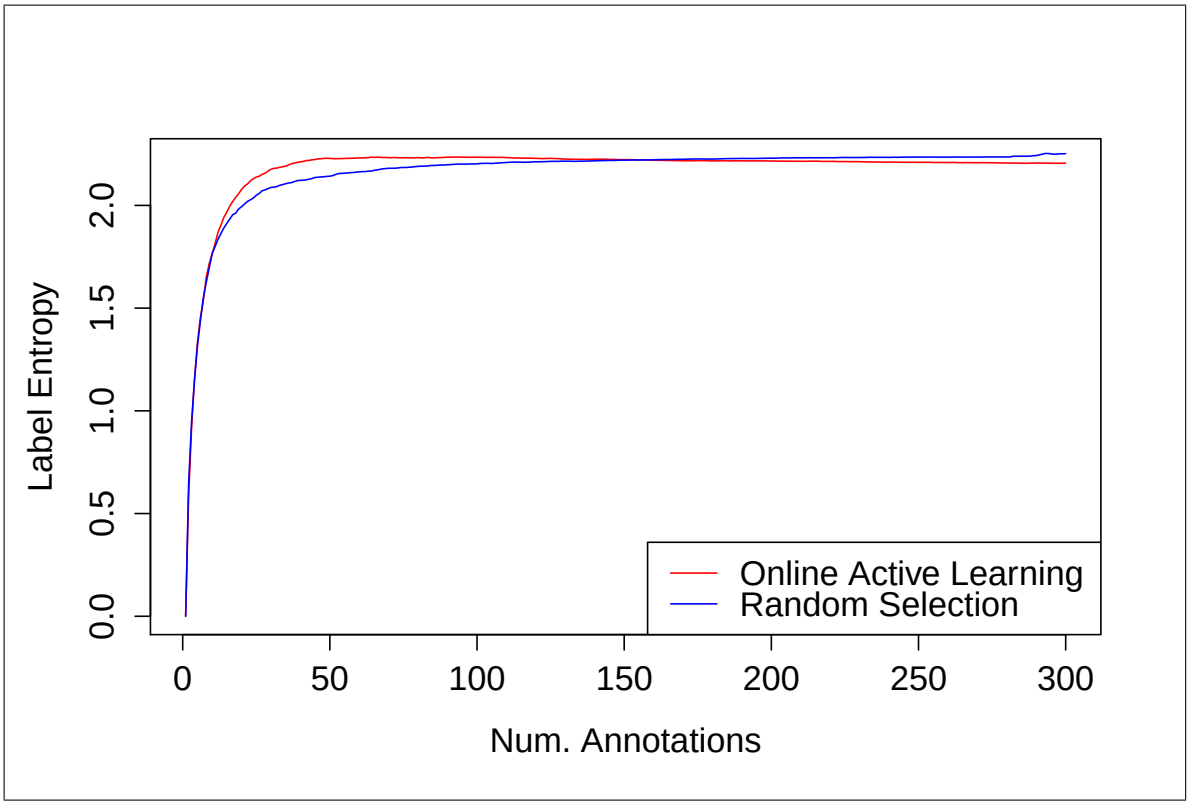



Figure 4.21: Active Learning versus Random Sampling on PAMAP. Training set class label entropy for balanced activity classes (initial class distribution).

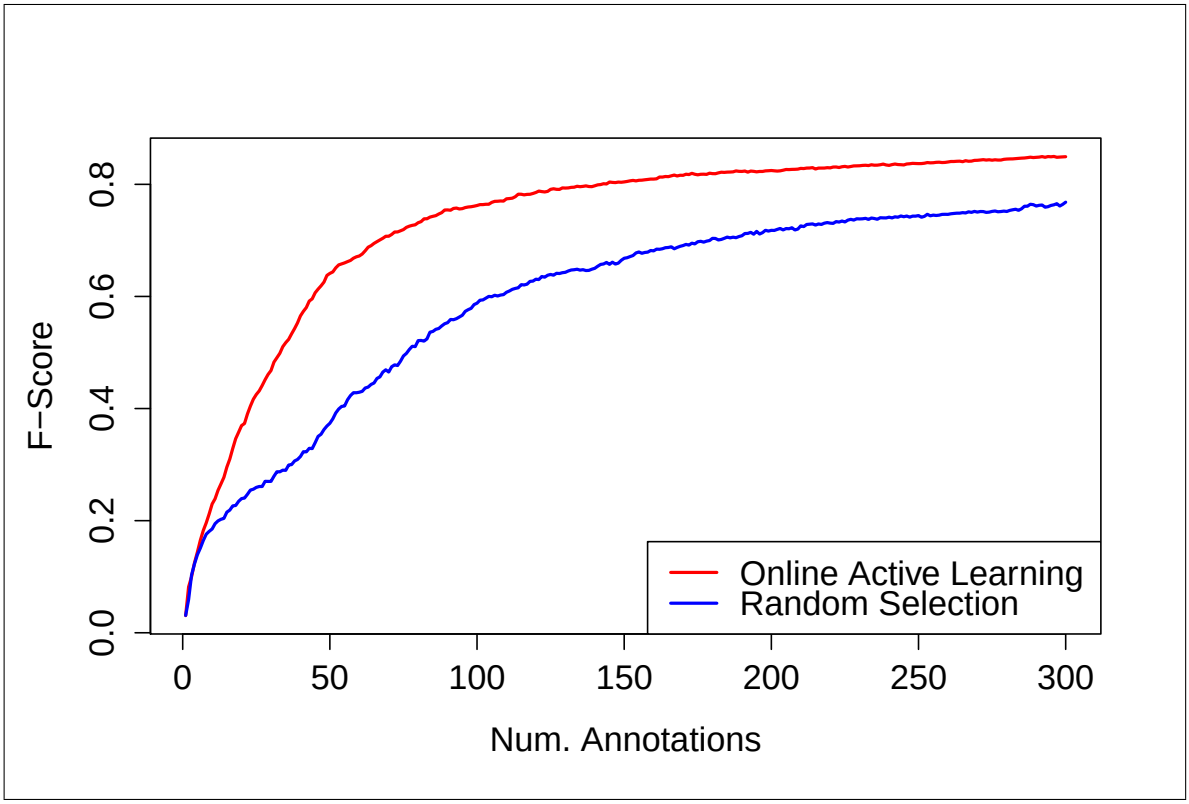



Figure 4.22: Active Learning versus Random Sampling on PAMAP. Average performance for imbalanced activity classes (artificially imbalanced class distribution).

For the first scenario, Fig. 4.20 represents the performance contrasts between OAL and RS for the PAMAP dataset. As expected, OAL-driven annotation maintains a performance margin over RS. We quantify the label diversity in the training set by calculating the *entropy* of the numerosity of the activity classes. Fig. 4.21 shows the contrast of entropy between OAL and RS for the PAMAP dataset when the class distribution is virtually non-existent. This implies that both OAL and RS register roughly equal training set class entropy showing that with the current class distribution in the activity stream OAL did not induce imbalances in the training sets, relative to RS.

For the second scenario, Fig. 4.22 represents the performance contrasts between OAL and RS. As the class distribution in the sequence of activities is more imbalanced (in this case, sedentary activities are 10 times more prevalent than in the previous scenario), then the benefits of employing OAL become more apparent. In this case, the probability of RS annotating rarer activities is lower, and this has clear penalties on model accuracy because other activities are neglected. The effect of what activities are

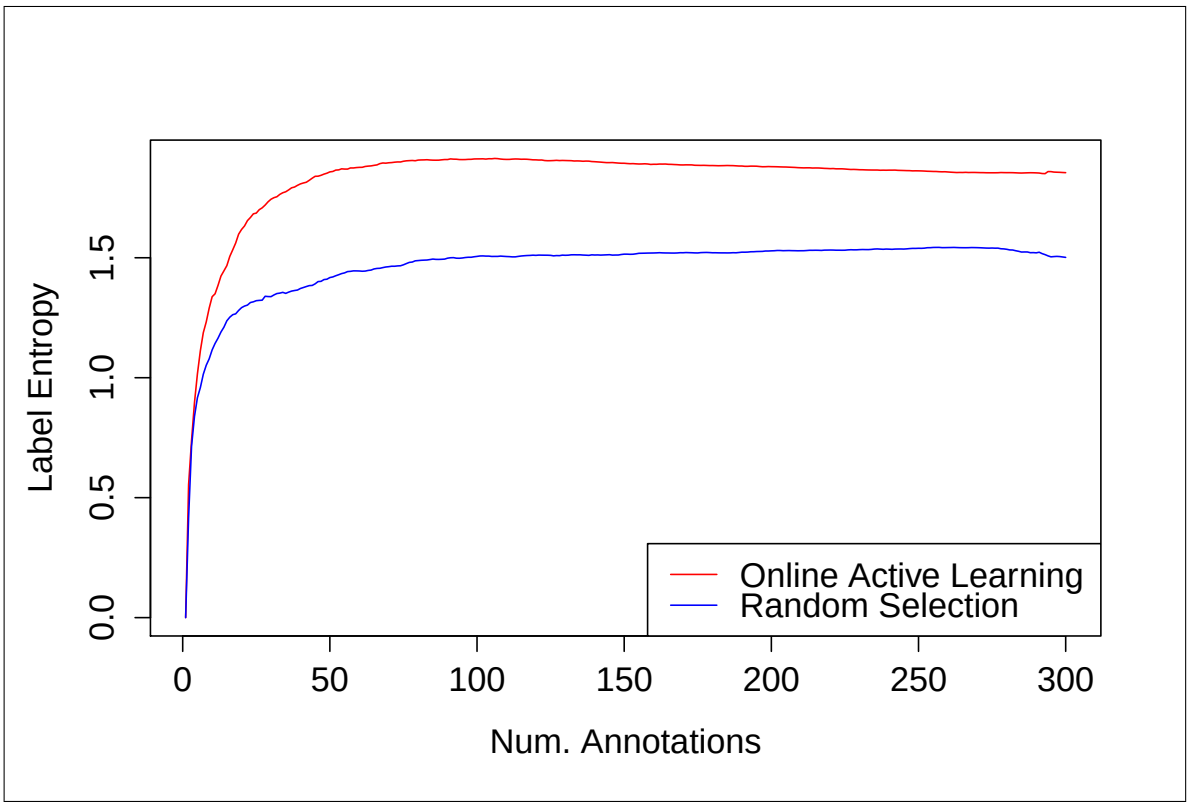



Figure 4.23: Active Learning versus Random Sampling on PAMAP. Training set class label entropy for imbalanced activity classes (artificially imbalanced class distribution).

targeted is illustrated in Fig. 4.23 which shows that OAL corresponds to higher class entropy (more balanced distribution of labels in the training set) than RS. While the class diversity corresponding to RS is ultimately influenced by the relative numerosity of each activity class in the stream of activities, OAL tends to reinforce difficult classes and, so, it distributes the annotation effort more evenly (relative to how RS does it) across all classes.

Because Online Active Learning overshoots the peak Random Selection performance point, we did not repeat the Equi-performance analysis from Section 4.3.3 (some points on the OAL curve are never matched or surpassed in value by any other points on the RS curve). However, because OAL outperforms RS vertically for the majority of time, it also reduces annotation effort relative to RS.

## Conclusions

In this chapter we continued our exploration of bootstrapping personalised HAR models from user-provided annotations. Using the motivation from Chapter 2, where we discussed the limitations concerning the user's memory, and similarly to Chapter 3, we employed an online approach to collecting annotations and constructing a personalised HAR model from these. However, unlike Chapter 3, in this chapter we are not concerned with the budget of individual annotations, but, instead, focus entirely on how to bring performance improvements the HAR model. We proposed an Online Active Learning decision heuristic for annotation which operates on an online stream of activities and identifies activities which, if annotated, are expected to bring considerable improvement to the accuracy of the HAR model.

We apply this annotation method and simulate its use in two different contexts – one focused on periodic activities, which are frequent in healthcare scenarios, and non-periodic activities, which are specific to more specialised applications, such as monitoring daily activities in a smarthome. Even though annotation decisions are informed only by a very limited recent history of the activity stream, results show that the recognition performance of a personalised model can be enhanced by using our Online Active Learning method over Random Selection – a naive annotation method which selects annotation segments at random.

We additionally proposed a data processing framework (a directed graph of algorithm placeholders) which includes stages for machine learning (sensor data preprocessing, feature extraction, classification, model updating, activity segmentation), a function for interacting with the user and a heuristic for deciding on what annotations to request. Actual algorithms can be plugged into the framework and create framework instances which can be used to obtain annotations from users in order to bootstrap a personalised HAR model. Using this framework, we evaluated the influence of the annotation heuristic on HAR model performance. The significance of our results in this chapter is four-fold:

1. For the same number of annotations, in most cases, the recognition performance

of a model trained from OAL-based annotation requests is greater than that of a model trained from RS-based annotation requests.

2. In order to obtain the same performance using OAL and RS, typically one requires fewer annotations through OAL than RS.

3. In case the monitored stream of activities is highly imbalanced, i.e. one or more activity classes greatly outnumber the rest (as is the case, for instance, of sedentary lifestyles where sedentary activities are more prevalent than non-sedentary ones), OAL favours annotating the rarer activities more than RS. Random Selection is ultimately influenced by the numerosity of each activity class in the stream and it is therefore less likely to annotate rarer activities. In addition, by annotating from a more imbalanced stream of activities, the performance gap between OAL and RS is increased compared to when the activities are more balanced.

4. For periodic activities we additionally investigated the effects of a realistic segmentation method by plugging it into our annotation framework. We argue that this framework setup is sufficient for bootstrapping personalised HAR models (and we demonstrate its application in a real user study in Chapter 5). Results show that the label noise due to imperfect segmentation has a negative but nonetheless small effect on accuracy (relative to the case when no noise is introduced). In this scenario as well OAL outperforms RS in terms of recognition accuracy.

Our annotation method works in an online setting, which is, in terms of computational complexity, compatible with a realistic deployment involving accelerated bootstrapping of a personalised activity model outside of an instrumented environment. We use the results in this chapter as motivation to further our investigation by enacting a user study which revolves around an actual deployment of the annotation system described in Chapter 5.

# 5

# Online Active Learning in the Wild

## Contents

# Introduction

In this chapter we build upon the results of Chapter 4 where we proposed Online Active Learning (OAL) as a method to collect annotations that accelerate the bootstrapping of personalised HAR models. There, we evaluated OAL on a public dataset and demonstrated clear objective performance gains. In this chapter, we apply the previously analysed methods for collecting annotations in a field deployment involving a panel of participants. We describe the implementation of a complete system using only online methods including Online Active Learning, sensor data preprocessing, automatic segmentation, model building and classification. The implementation itself is proof that it is feasible to construct a mobile and autonomous Online Active Learning deployment.

We embody the system in a mobile app which allows us to collect real-time data wirelessly from wearable sensors and to process the data using an OAL-based machine learning pipeline, similar to our proposition in Chapter 4. We set up a field deployment involving a panel of participants and deploy the app in a naturalistic environment. This allows us to evaluate not only the objective performance of OAL in a naturalistic setting, but also the subjective impressions the participants experienced while using the system.

## *Contributions*

The contributions of this chapter are as follows:

Firstly, we enhance the machine learning pipeline from Chapter 4 by incorporating an additional module, the *Novel Activity Detector* (NAD) that further accelerates the learning process by discovering new activities or activities with unusual executions. We detail why the Online Active Learning method detailed in Chapter 4 is sometimes too slow to improve the model and therefore important annotations may be missed. We show that the NAD can impact the quantity of useful annotations that can be obtained during the limited exposure of the user to the annotation method.

Secondly, we demonstrate that model boostrapping from genuine user-provided online annotations of activities performed in a naturalistic environment leads to clear

model improvement: When compared with a *strawman* classifier (a simplistic classifier that systematically predicts the most numerous activity class in the training set), the performance due to OAL surpasses the strawman by 38-60% in F-Score. We did not perform Random Selection because, as we explain later in this chapter, due to the nature of the experiment protocol, for most target activities, an estimated 43% of annotations would have been lost if Random Selection had been used instead of Online Active Learning. This complements our findings in Chapter 4, which were based on simulations of online methods on previously collected data, with this chapter's results based on annotations collected under live and realistic conditions and without expert supervision. Results suggest that only a few annotations for each activity are enough to register improvement to a fully personalised activity model, relative to a simple strawman.

Finally, we gauge the perceptions of the participants who provided the annotations as part of the user study. We present insight gained from analysing user feedback, at key stages of the user study, reflecting their view on how the system behaves. We show how this subjective feedback can be used to to "close the loop" and to alter the design so that users' expectations are better met. Finally, we compile summary opinions from our participants and explain how using an OAL-based system affects them and, also, how it could be applied to other contexts.

## *Overview of Field Study*

In this chapter we explain the design and implementation of a fully personalised interactive mobile activity monitor which, at its core, uses Online Active Learning to collect user-provided annotations for automatically segmented activities as they occur. We explain how we deployed it in an office environment as part of a user study involving 10 participating office workers. We finally present quantitative and qualitative results regarding user interaction with the activity monitor and how these results impact potential applications in the area of physical activity exertion at the office.

In order to support our claims and enact personalised activity recognition for office workers, we have set up a field study that incorporates the following elements:

**Experiment Design** We designed a user-centred field study which involved participants performing a set of light physical activities while providing annotations on demand, according to an OAL-based asking heuristic. We explain why the experiment protocol is in line with our assumptions on self-provided annotations and why this in turn validates the applicability of the methods and techniques to the context of office workers in a office environment.

**Machine Learning Pipeline** We applied a modified version of the machine learning pipeline from Section 4.4, which now accommodates an extra module – the NAD –, which works in parallel to the OAL module and which accelerates learning further by soliciting some annotations more quickly than OAL alone.

**Mobile App** We implemented a mobile app which implemented the online machine learning pipeline and which acted as the point of user interaction for answering annotation requests. We describe the architecture of the mobile app we employed in the user study to monitor users and collect annotations. We focus on how we integrate purely online modules in our data processing pipeline and how this supports the central feature of requesting relevant annotations through Online Active Learning.

**Objective Performance Evaluation** We derive objective recognition performance measures for individual participants. While the quantity of annotations is smaller than for the datasets used in Chapter 4, we demonstrate that online learning from live annotations in a naturalistic environment is a valid solution to the problem of bootstrapping a personalised model. Our results show that live on-demand annotations are suitable for bootstrapping a personalised activity recognizer and that the resulting personalised models exhibit improvements in recognition performance.

**Subjective Usability Questionnaires** Participants were asked to fill in questionnaires at different stages of the experiment where they would express their subjective views on the usability of the system as a whole and potential of the annotation method in the current and other contexts. We used the feedback to not only reflect how some design choices affect usability, but also to respond to

a criticised issue – the NAD module– and address it by changing its behaviour to better meet user expectations.

In terms of quantitative analysis, the user study was aimed not at replicating the type of results in Chapters 3 and 4, but rather the purpose of the study was two-fold. Firstly, we aimed to obtain insight into how users respond to this class of notifications and how they can integrate the level of disruption in their daily routines. This is discussed further in 5.6. Secondly, we aimed to show that, although the quantity of annotations is more limited compared to other offline datasets like Opportunity or USC-HAD or PAMAP, it is possible to accumulate useful self-reported annotations and to conclude that these lead to increased activity recognition accuracy.

## Experimental Design

We have highlighted in Chapter 1 not only that sedentary behaviour is detrimental to health, but also that prolonged sitting, which is typical of many office workers, carries additional health risks. We envisage a mobile scenario where office workers combat excessive sitting by using a smartphone app which monitors their activities at the office and which improves by parsimoniously asking for annotations.

Our goals are two-fold: Firstly, we explore the technical feasibility and objective merits of the suite of methods we employed in a realistic deployment. The learning techniques used in this experiment are largely those used in Section 4.4, but they are now applied to a live setting where annotation decisions are informed by live data streams generated by sensors worn by our participants. We create a mobile app to integrate the data processing infrastructure with interactive functionality to collect labels for activities as they occur. This is in direct contrast to the simulations from Chapter 4 where offline data was simply sequentially replayed in order to artificially create the restriction of a data stream.

Secondly, we examine the subjective usability of such an interactive activity monitor aimed at office workers. We therefore introduce written questionnaires as a means of collecting the participants' impressions at different stages of the experiment. With this

feedback we are able to better understand how users perceive this type of interaction and how useful this type of adaptive application would be in the current and different contexts. We also use the feedback to alter the application to correct a problematic aspect of the interaction with the user.

To these ends, we have designed a compatible experimental setup which consists of a set of activities with an associated context in which they are performed and an experiment protocol which emphasises key manifestations of the context.

### *Activities*

We target light physical activities that could realistically be performed at the office: *sitting*, *standing*, *sitting knee raises*, *walking*, *squats*, *calf raises*, *torso side to side*, *torso twists* and *torso back to forward*.

This is a diverse set of activities that is arguably suitable for an office environment. All activities require relatively little energy expenditure and, in retrospect, none of the participants mentioned any difficulty in performing them. Also, no special equipment or areas are needed.

Movement data was collected using accelerometers strapped in four locations on the participants' bodies: the right foot, the right lower leg, the right upper leg and the chest. These locations were chosen because they capture key movements for the proposed activities – these predominantly consist of movement from the lower limbs and from the torso. The accelerometer data was transmitted wirelessly to an Android smartphone where our app coordinated data processing with user interactions.

### *Experiment Protocol*

In terms of participants, we recruited 10 colleagues (2 females, 8 males, aged 23-30) from our department who were not affiliated with our research. We demonstrated the target activities and asked the participants to include 8-10 repetitions of each activity in their daily routine at the office, each time for a duration of at least 30-60 seconds. Participants were informed that they could execute the activities in any order, at any time and could take breaks. The participants were not supervised while the

experiment was under way in order to ensure that there was no interference in how the activities were performed or what or how annotations were provided. There is arguably a risk of noise being introduced in the data, but, with the benefit of hindsight, the results show that noise levels are low enough not to deny the accumulation of relevant annotations. The participants were also informed that the app would not prompt or guide them to perform activities in any way, but rather would simply react to registered activities. This ensures that the annotation framework is decoupled from the experimental protocol and, so, it could be applied in similar contexts, without the user having to observe a certain protocol.

We divided the duration of the experiments in two parts, each with its own annotation request mechanism. In the first part of the experiment only informed annotation requests were generated. In the second part, after the *switch-over time*, some random annotation requests were added in order to obtain additional annotations for performance evaluation.

Overall, we aimed for a naturalistic and minimally obtrusive environment where the participants would perform the experiment by executing a diverse set of activities and occasionally providing their own annotations as directed by a mobile app. We did not follow the participants and we did not collect video footage so that participants did not feel compelled to perform the activities in an unnatural way and so that the environment was not restricted to where video footage was collected. Therefore, the participants' own annotations are the main factors in performance results. In total, the participants were monitored by our mobile app for 55 hours[1] and, during this time, they annotated 3 hours and 20 minutes' worth of sensor data.

Besides the physical exertion component of the experiment, participants were also asked to fill in questionnaires as a direct means of obtaining subjective judgements regarding the usability of the app. We asked the participants to fill in three questionnaires at different key stages of the experiment:

**Pre-experiment questionnaire** The participants were asked to fill in this questionnaire after an explanation of what the experiment involved, but before commenc-

[1]Including downtime due to occasional app crashing.

ing the physical activity data acquisition. The purpose of this questionnaire was to make the participant create a personal expectation regarding the amount of interruptions and, after the end of the physical experiment, to realise whether this expectation was met or not.

**Post-experiment questionnaire 1** As soon as the participants finished the physical exertion part of the experiment and they surrendered the smartphone with the onboard database of movement data, we asked them to complete the first of two post-experiment questionnaires. At this point we did not present any results or discuss recognition performance metrics with the participants. Without additional information from us, but having experienced a day at the office interacting with the app, participants were asked to reflect on usability aspects and also whether they thought the app learned from their input.

**Post-experiment questionnaire 2** Using the participant's movement data and annotations from the smartphone, we generated model performance charts which were essentially the same as Figures 5.8, but illustrating only the current participant's data. We showed the participant the performance curve and explained to her what the curve represents. However, we refrained from expressing our opinion on whether the performance curve designates learning or stagnation. In this final questionnaire, since the participant was now in possession of objective performance indicators, we repeated the question of whether she thought the system learned from her feedback. In addition, we asked the participant to reflect on the potential of using the app for longer periods of time or in different contexts.

The consent form completed and signed by every participant is found in Appendix A. The questionnaire structure can be found in Appendix B and the questionnaire answers completed by the participants are in Appendix C. Given the nature of the data collected from the participants, their tasks and the loose supervision of the participants by the research team, the user study did not require explicit ethical approval from the university. It is extremely improbable that the questionnaire and movement data collected can be used by another party to identify the participants or with other

unethical intentions.

## Learning Machinery

As the activities are periodic, we use a frame-based approach to preprocessing, machine learning and segmentation similar to that used in Section 4.4 for the simulations using the period USC-HAD and PAMAP datasets. Namely, for each of the four tri-axial accelerometers we set up the automated data processing pipeline as follows:

- We use a 5s sliding window with a 50% overlap. We included the overlap because this would improve the segmentation of movement data. Since the start and end of an activity is no longer under the control of a computer simulation, it can happen at any time, including in the middle of a sliding window. Therefore, the 50% overlap better segregates the adjacent frames corresponding to neighbouring activities.

- As features, for each of the three acceleration signal axes we compute the mean (three features), variance (three features) and for each unordered pair of axes (XY, YZ, ZX) we compute the inter-axis correlation coefficients (three features). This results in a 9-dimensional feature vector.

- For classification, we use a Bootstrap Aggregator of 30 Naive Bayes classifiers. As discussed in Section 4.4, this type of ensemble was the best performing model builder. We obtain classification metrics (predicted label, classification probabilities) at the level of individual frames and we average these in order to obtain corresponding segment-level metrics.

- We apply the segmentation procedure from Section 4.4 to partition the stream of feature vectors into individual activities comprising contiguous sequences of frames.

There are two differences between the current machine learning pipeline and the one used in Section 4.4. Firstly, the current pipeline includes an additional module (the NAD) and this is discussed in detail in Section 5.3.1. Secondly, the sliding window

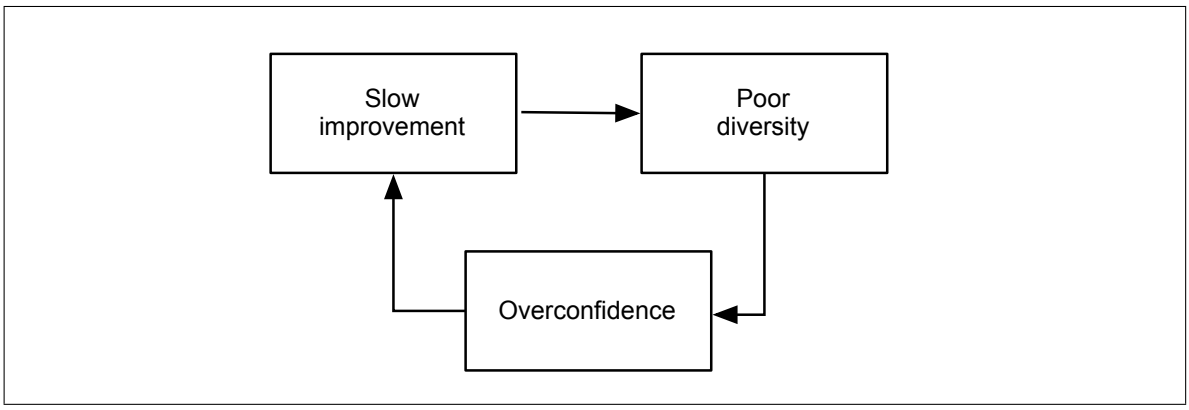



Figure 5.1: The Ignorant Classifier Problem.

procedure now has an overlap between adjacent frames. This was done to increase the number of annotated frames and provide more data for model evaluation. As opposed to Section 4.4, where we specifically avoided overlaps in order to reduce duplication between training and testing sets when evaluating models, the motivation for doing so is no longer valid. Here we are no longer interested in extremely precise performance estimations, but rather we simply seek to identify whether a model registers improvement. Having an overlap in the sliding window does not fundamentally affect the presence of improvement.

### *NAD: Novel Activity Detector*

We prompt the user of the system to provide labels according to Eq. 4.1. The mechanism uses the confidence in prediction of a bootstrapped model to issue the probability $p_{ask}$ of asking the user for annotation.

While Online Active Learning yields gains in recognition performance, the speed with which initial annotations are requested is very low. The problem arises with the initial annotated segment which results in a training set with a single label. At this stage, this training set leads the classifier to systematically predict that label for all new frames and with 100% confidence. The issue, which we call the *Ignorant Classifier Problem*, is illustrated in Fig. 5.1. Little diversity in the training set triggers classifier overconfidence which, in turn, causes very slow improvement. This behaviour can persists for many iterations afterwards as long as more diverse labels are still not discovered. Therefore, initially, the classifier is misguided to confidently but incorrectly

classify new segments. Overconfidence is what makes Eq. 4.1 ineffective.

If there is a limit of time within which to conduct Online Active Learning, then it is clear that very little input would be generated at the beginning of the monitoring, thereby wasting time. However, the *Ignorant Classifier* was not a problem in the simulations in Chapter 4 because there was no imposed limit of time within which to finish the bootstrapping process. Technically, the simulation environment would simulate physical activity by generating numerous data points without necessarily generating annotation requests as well (due to Eq. 4.1). Nonetheless, some annotations would eventually be requested because the asking probability is nonetheless non-zero,. Therefore, the training set would eventually diversify and more informed decisions would follow. However, initial overconfidence becomes problematic in the context of a realistic deployment because of the time constraints in our field experiments. Specifically, for each participant, the monitoring would last only several hours. Therefore, obtaining annotations even at the beginning of the bootstrapping process is instrumental to collecting a large enough dataset for model performance evaluation.

In order to address the Ignorant Classifier problem, we introduce another annotation heuristic called the Novel Activity Detector (NAD) which complements Online Active Learning. The NAD works in parallel to Online Active Learning and generates annotation requests of its own based on a different mechanism. The NAD therefore aims to break the vicious circle illustrated in Fig. 5.1 and to increase label diversity in the early stages of learning. In our user study we experimented with two NAD versions. The first version, the *Speculative* NAD, favours a high throughput of annotation requests, but it can be intrusive to users. The second version, the *Restrained* NAD, limits the number of annotation requests to one per activity class, but this version carries the risk of not discovering some labels.

### Novel Activity Detector – Speculative Version

In the initial version, we use a Bag of 30 Naive Bayes classifiers. Normally, each Naive Bayes classifier would output a vector of *probability scores* for each label. These scores would be transformed into actual probabilities by scaling them such that they sum to 1. The unscaled probability scores are proportional to the scaled probabilities,

so that they too are representative of the model's prediction confidence. However, for the 1-label Ignorant Classifier case, the confidence is always 100% and hides the potential variation of the corresponding unscaled probability score. We want to detect changes in confidence even if the training set contains only one label or very little label diversity. Therefore, we propose Eq. 5.1 as a NAD formula that uses the unscaled prediction confidence $p_{conf}^{unscaled}$ to generate its own probability $p_{ask}$ of asking the user for an annotation. Because of the lack of scaling, the NAD works equally well for any number of classes known to the model, including for the one class case.

$$p_{ask} = exp(-\gamma \cdot \ln p_{conf}^{unscaled}) \tag{5.1}$$

The unscaled probability scores are extremely sensitive to the high dimensionality of the input space – small variability in the input may lead to disproportionately large variations in unscaled probability scores. We have taken two steps to limit this variability. Firstly, we reduced the dimensionality of the input space by using only a subset of the original features and, secondly, we applied a logarithmic factor to further reduce variability down to a more manageable range.

The resulting asking probability is given by Eq. 5.1 which is linearly scaled to $[0, 1]$[2]. This annotation mechanism is similar to the main one used in Eq. 4.1. However, as Fig. 5.2 shows, the NAD focuses high asking probabilities only in the region of very small unscaled probability scores. High $\gamma$ parameter values result in, practically, a hard threshold, whereas lower $\gamma$ parameter values provide a more attenuated decline in the probability of raising annotation requests.

We used a small dataset collected offline and concluded that $\gamma = 0.02$ would be a good value to highlight novel activities while ignoring known labels. However, the first participants whose app used this NAD implementation had noted the large number of *sitting* activities they were asked to annotate. We traced the cause of annotation requests to the NAD which was too sensitive. The NAD would trigger annotation requests for known activities that were executed slightly differently even if this was natural variability and this is the cause of high throughput we noted earlier.

[2]The Weka implementation of the Naive Bayes classifier protects against numeric underflow by enforcing a minimum unscaled probability of $10^{-75}$. We use this minimum value to scale $p_{ask}$.

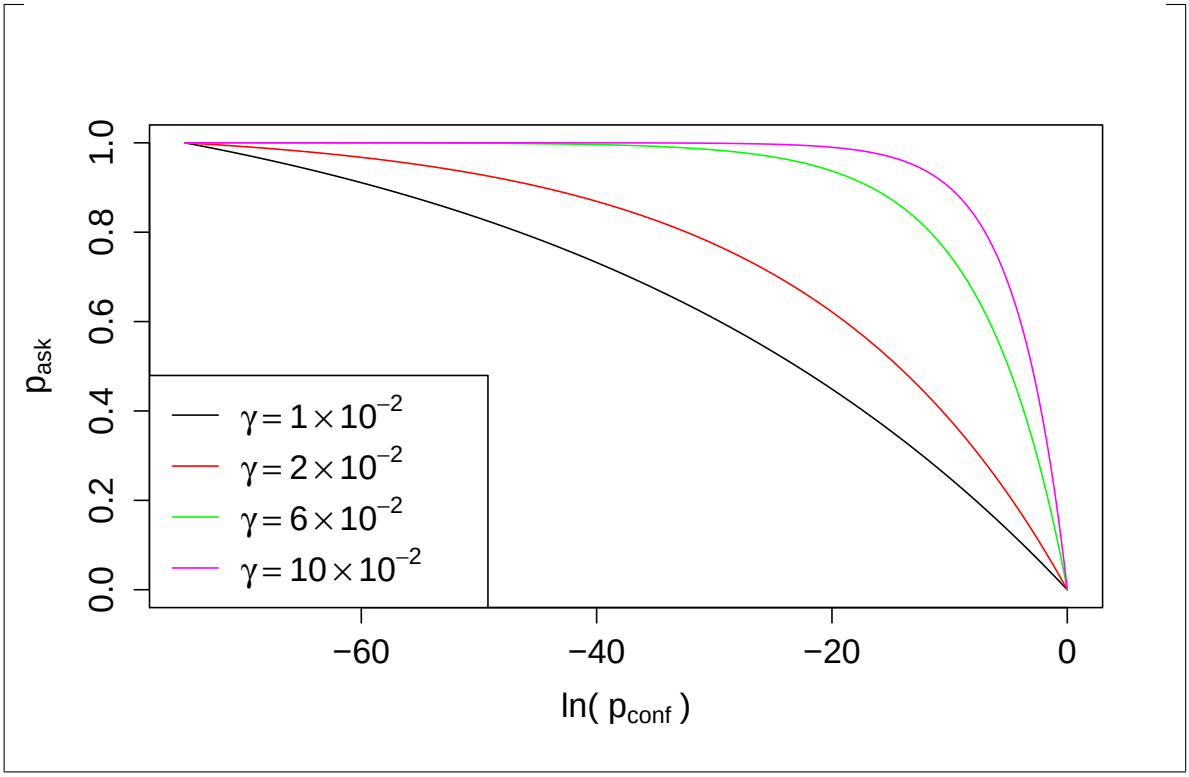



Figure 5.2: NAD Probabilities of Generating Annotation Requests (Logarithmic Scale)

### Novel Activity Detector – Restrained Version

We also experimented with a lower throughput NAD that did not engage users as often as the Speculative NAD. Consequently, for the other participants, we used a more restrained NAD mechanism of generating annotation requests. In this version, we not only learned the importance of reducing user participation, but we also utilised the data collected from the previous participants (their sets of annotations) who used the Speculative NAD. The Restrained NAD (a single model for all subsequent participants) was a *population* model constructed using a Nearest Neighbour classifier from the median feature vectors of each class – a training set of just 9 points[3]. Furthermore, a Nearest Neighbour classifier using such a small training set would still be able to deliver very fast online classifications. The second version of the NAD used this activity model to classify newly computed feature vectors. The NAD maintained a list of user-provided labels, but as classified by the population model. Namely, when the population model classifies a new activity $a_i^{pop}$ which has not been estimated before, then an annotation is requested for the current segment. Regardless of what label the

[3]Median values were used here because they are generally insensitive to outliers.

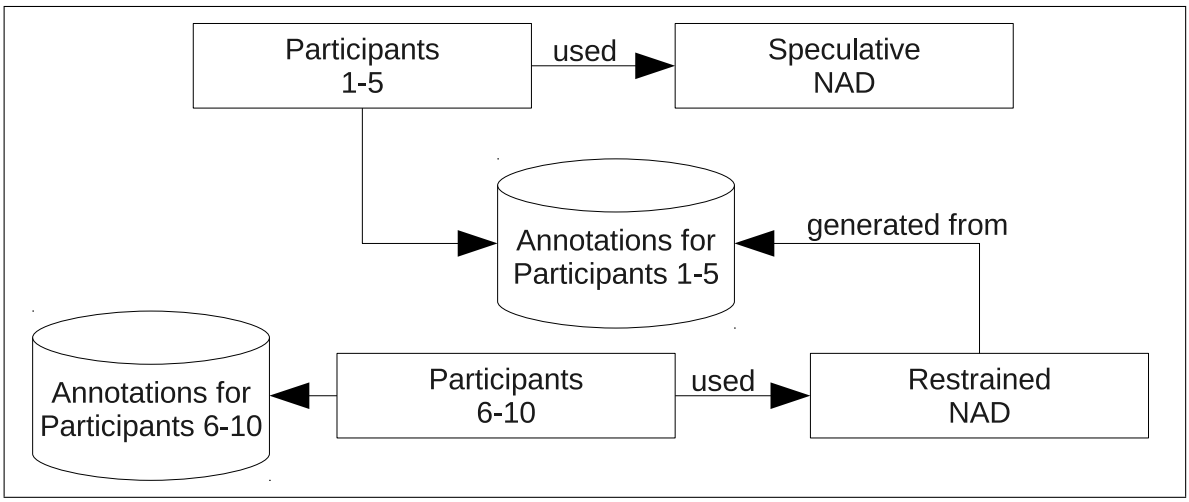



Figure 5.3: NAD Usage Throughout the Experiment.

user provides, say $a_i^{provided}$, the label $a_i^{pop}$ is marked as annotated even if $a_i^{provided} \neq a_i^{pop}$. This ensures that the NAD never requests more than one annotation per activity class. Consequently, most annotation requests come from Online Active Learning. The Restrained version of the NAD, despite using a population model, still supports the bootstrapping of a fully personalised activity model, just as the Speculative NAD does.

### *Updateable Bootstrap Aggregation*

In the mobile app we adapted the learning machinery from Section 4.4 and used a modified ensemble of classifiers. A key difference is that we implemented and used an *updateable* bootstrap aggregator - one that is able to adapt incrementally to new training data instances as they become available.

An ordinary bootstrap aggregator, as was used in Section 4.4, operates on a given training set by sampling with replacement a set of instances which are then used to train one of the base classifiers, as illustrated in Figure 5.4. The sampling step is repeated for every base classifier [120].

The resulting ensemble structure is not updateable – without complete re-training, it cannot adapt incrementally to novel annotations, and this is at odds with how data accumulates in our stream-based scenario. Partial training data must be incorporated incrementally in the model because repeatedly re-training activity models on a mobile platform is computationally expensive.

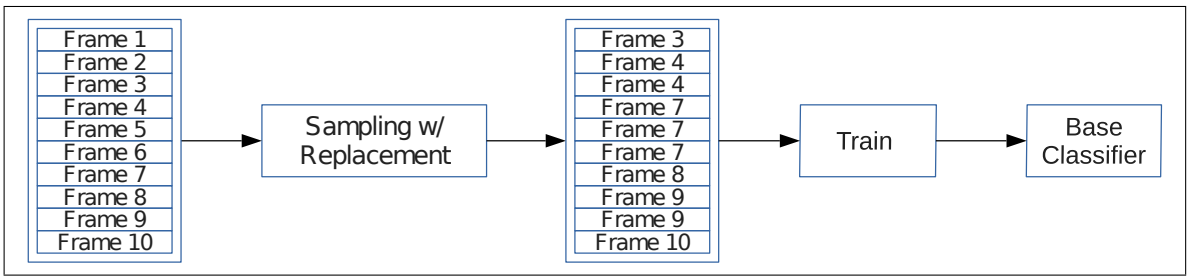



Figure 5.4: Training with boostrap aggregation.

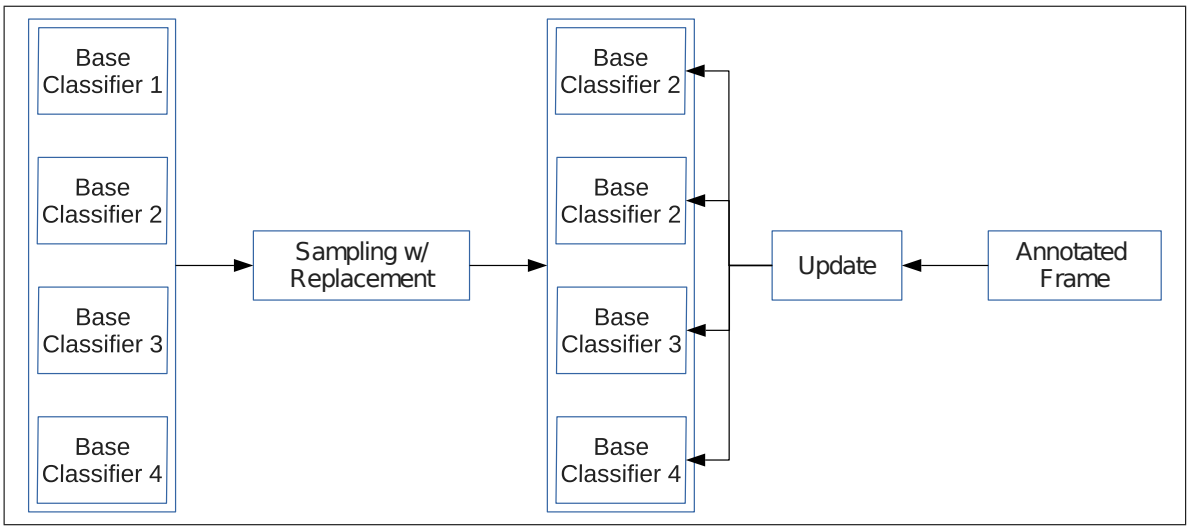



Figure 5.5: Updating with modified boostrap aggregation.

Instead, we use a modified bootstrap aggregation which uses updateable base classifiers (the Naive Bayes classifiers are updateable [120]) and which can itself be updated with new annotations. The resulting classifier can therefore be bootstrapped incrementally and so it can be improved with new annotations, as they become available. The key difference here is that models are no longer *trained* from scratch whenever new annotations appear, but instead they can be *updated* with new data. Updating updateable classifiers is more computationally efficient than training them from scratch.

Given a labelled frame from a new annotation, instead of training each base classifier with a training set sampled from the entire set of annotations, as is typical of bootstrap aggregation, we sample the set of base classifiers from the existing classifiers to be updated with the new data (instead of training them from scratch). This iteration is repeated for every feature vector in a newly annotated segment. An example is illustrated in Figure 5.5, where, as a result of sampling with replacement, for a selected annotated feature vector base Classifiers 3 and 4 are updated once, Classifier 2 is updated twice and Classifier 1 is not updated with this labelled frame.

### *Effect of the Segmentation Procedure*

In Chapters 3 and 4 we simulated sudden transitions between activities: the frames on either side of a true segment boundary belong to a single activity. In a realistic deployment, as is the case in this chapter, and with continuous monitoring, transitions are not so clear cut. Frames capture fixed temporal boundaries which are generally not aligned with activity changes for two reasons. Firstly, the overlap between frames causes at least one frame to contain movement for two activities. Secondly, activity transitions are not instantaneous, but rather there is an intermediary time interval which is characterized by transitional movement which cannot be classified as either activity. These factors create uncertainty over when an activity ends and another one starts.

Although we rely on automatic segment boundary detection to yield a high true positive rate (not to fail to detect an activity change when it occurs), we still assume uncertainty regarding the instantaneity of the boundary. Specifically, we discard a number of frames (two in this particular case) from either side of the detected segmentation boundary from annotation requests. More frames could be discarded either side of the segment boundary to increase the quality of annotated frames as this decreases the prevalence of label noise within a segment. However, this would come at the cost to the quantity of annotated frames because each segment would have fewer frames remaining for annotation.

## Mobile App Architecture

We now detail the complete implementation system of the mobile app. Its core features include:

- Supporting human activity recognition from wearable sensors.
- Parsimoniously interacting with the user to obtain activity annotations.
- Propagating annotations back to the machine learning pipeline and updating the user's personalised activity model.

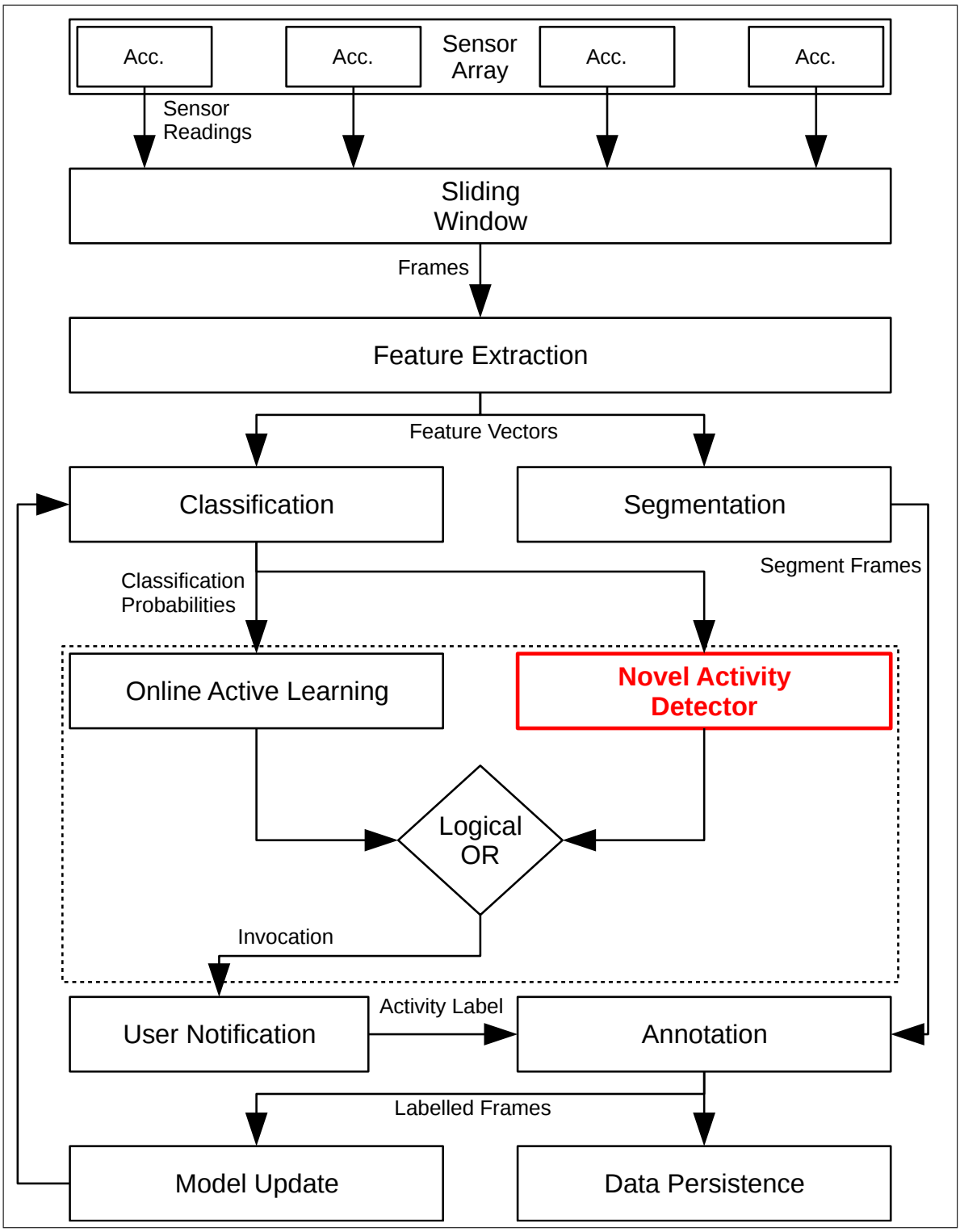



Figure 5.6: App data flow diagram.

The system, whose architecture is illustrated in Fig. 5.6, uses Online Active Learning at its core. The architecture is to some extent similar to that used in Chapter 4 (Fig. 4.3), but the novelty introduced is emphasised with the dashed line rectangle. As explained earlier, the NAD, which is included to increase the probability of identifying novel activities, is combined with Online Active Learning using logical disjunction.

**Sensor Array** We use four accelerometers integrated into the WAX9[4] wireless wearable sensor. Data from the sensors is streamed in real time to a Google Nexus 5 smartphone via Bluetooth.

**Data Preprocessing** We operate a single sliding window across all four three-dimensional acceleration timeseries streamed from the wireless sensors. Similarly to Chapter 4, we use a window length of five seconds. In addition, in order to increase the quantity of data, we add an overlap of 50% between adjacent frames.

**Feature Extraction** We extract features as described in 5.3. Each window from a mote is compressed down to nine features. The four simultaneous windows from all four motes result in a feature vector of dimensionality 36.

**Segmentation** The feature vectors from the previous step are used to detect changes in activity using the algorithm from Sections 4.2.2 and 5.3.3.

**Classification** The classification stage, which consists of an activity model trained using a Bootstrap Aggregation with 30 Naive Bayes classifiers, is used to derive class labels with associated classification probabilities for all newly computed feature vectors.

**Annotation Request Decider** The classification probabilities from the classification stage which correspond to the newly identified segment feature vectors from the Segmentation stage are used to inform the Online Active Learning stage on whether the user should be prompted to provide an annotation.

**Online Active Learning** This stage uses logical disjunction between the annotation decisions taken using Eqs. 4.1, which was the sole OAL mechanism used

[4]http://axivity.com/v2/ Accessed 19.12.2014

in Chapter 4 and a NAD, explained in Section 5.3.1. In the deployment, we experimented with both NAD versions described in Section 5.3.1 – participants 1-5 used the Speculative NAD and participants 6-10 used the Restrained NAD.

**User Notification** If a segment is deemed as worthy of annotation, then the app invokes a means of collecting input from the user. We opted for a tactile, visual and, optionally, audible feedback system to notify the user and we used a one-tap interface for the user to provide a label for the annotation in question.

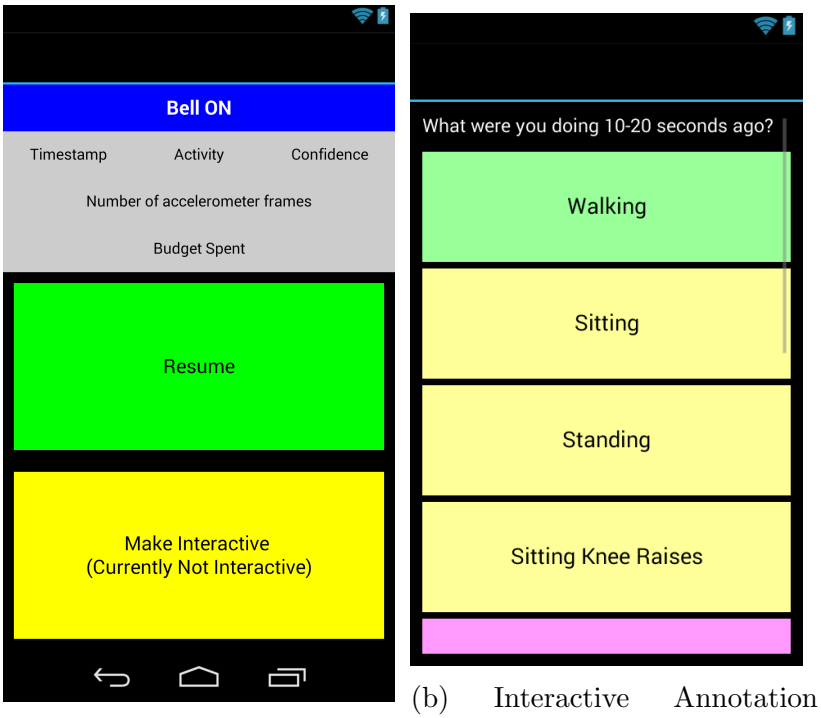
**Annotation** From the previous step, an activity label is collected and associated with the segment frames which triggered the annotation request in the first place.

**Model Update** Newly annotated segment feature vectors are passed to update the existing activity model using the ensemble updating mechanism presented in Section 5.3.2.

**Data Persistence** We recorded the acceleration data and the computed features on-board the secondary storage of the phone in flat files and an SQLite database[5]. While all data processing leading up to and including annotation requests and model updates were performed on the phone using local computation only, at the end of the physical part of the experiment we downloaded the data from the phone to provide the performance analysis in Section 5.5.

Of notable difficulty when implementing the app was enforcing the memory complexity by removing memory leaks. We ensured that references to unnecessary objects were dropped as soon as possible, thus making the associated memory re-allocable. This step included continually removing obsolete data from all data structures used for monitoring or data processing. Maintaining a sleek memory footprint was essential because the app would be expected to run continuously for several hours while processing very large amounts of streaming sensor data; otherwise increasing amounts of memory would never be released and the system runtime would eventually terminate the app.

[5]http://www.sqlite.org/ Accessed 19.12.2014




(a) Main Screen (b) Interactive Annotation Screen

Figure 5.7: App Screens

### *App User Interface*

In terms of the user interface design, we opted for an extremely straightforward visual-tactile interaction pattern. We created an app with two Android activities[6] and tap-only buttons, both illustrated in Figure 5.7.

The main activity pictured in Figure 5.7a allows the user to enable and disable audio feedback (*Bell ON*) and presents simple monitoring notification by displaying the currently identified activity (*Timestamp*, *Activity*, *Confidence*), current segment size (*Number of accelerometer frames*) and number of annotations provided by the user (*Budget Spent*). Users can pause and resume acceleration streaming (*Resume*) or enable or disable just the notification prompts while the acceleration data is still being streamed (*Make Interactive*).

When an annotation is deemed necessary, the Interactive Annotation Screen, shown in Figure 5.7b, is automatically presented to the user, along with tactile feedback. The

[6]In Android terminology, an activity is "*a single, focused thing that the user can do*" [158] and in the Android library represents a container for a user interface. Not to be confused with physical activity.

user can select the label for the newly delineated segment from a predefined menu of activities with a single tap. The Interactive Annotation Screen is presented for a maximum of 15 seconds before the annotation request is discarded. The screen switches back to the main screen and, so, the user is no longer able to provide the annotation. This maximum delay with which an annotation can be provided ensures that the user recall is not required to function past a certain duration.

We initially considered a speech-based interface as a means of obtaining annotations from the user, but we rejected the concept because two problems became apparent. Firstly, there are audio noise implications. In our proposed office environment, where there may be multiple occupants within close proximity, providing spoken feedback is awkward. Secondly, as is shown by Hoque et al. [71], automated speech recognition introduces errors in the provided labels. This, in turn, would reduce the accuracy of the bootstrapped classifier.

### *Online Learning from Annotations*

All application feature implementations revolved around making the entire process of learning from self-reported annotations run in an online fashion, that is in constant time and space complexity with respect to the size of the movement data stream. Our data flow design choices support online execution:

**Sensor Streaming Data Capture** The motes transmit the current accelerometer values in real time, i.e. as soon as possible, according to the specified BLE notification rate.

**Data Preprocessing** The sliding windows capture a accelerometer timeseries fixed history of five seconds before releasing the data and sliding the window to the next five seconds interval.

**Feature Extraction** Our choice of features (means, variances and correlations) are computationally fast and run in linear time with respect to the number of sensor readings in the window[7].

[7]It is known that for all these functions, most of the computation can potentially be done in-

**Segmentation** The segmentation operation is relatively inexpensive as it requires only a fixed number of the most recently computed feature vectors, and hence has constant complexity with respect to the number of generated feature vectors. While this stage was the most pressing in terms of usability due to the relatively long delay it introduced between the actual occurrence and subsequent identification of a change in activity, its computational requirements are nonetheless constant in time.

**Classification** When classifying, a Naive Bayes model has a constant computational cost with respect to the number of training examples and therefore with respect to the number of computed feature vectors. The Bootstrap Aggregation training technique generates a panel of classifiers and the final classification result is given as a majority vote, which has linear complexity in terms of the number of panel classifiers. The classification step, overall, is constant with respect to the number of generated feature vectors.

**Annotation Request Decider** A decision to annotate a segment is the result of an arithmetic and geometric mean of classification confidence levels over the set of frames in the current segment. Because of the online classification step, annotation decisions are taken with linear complexity in terms of segment size, but because segments are relatively short-lived, this is constant complexity in terms of the total number of generated feature vectors. The computation effort does not increase in time as new feature vectors are generated[8].

**User Notification** This step simply augments the decision operation with a single user input, so it simply adds a constant overhead.

**Signal Annotation** When an annotation is provided, the label is propagated to all frames in the segment. Similarly to the annotation decision, the complexity is linear in terms of segment frames, but constant in terms of the number of generated feature vectors.

crementally with each newly arrived value. The Apache *math3* library, for example, has the option of online computation for means and variances: `http://commons.apache.org/proper/commons-math/userguide/stat.html#a1.2_Descriptive_statistics` Accessed 19.12.2014

[8] In order to ensure the app does not consume excessive memory to represent the current frame, we limited the number of frames in any segment to a specified maximum.

**Model Update** Due to the nature of the Naive Bayes classifier, this step is relatively inexpensive computationally because the classifier does not need to be re-trained from scratch, but rather it can be updated incrementally using only the newly annotated segment frames. Updating a Naive Bayes classifier is therefore also linear in the number of segment frames, but constant with respect to the number of generated feature vectors. Bootstrap Aggregation, as documented in [156] is *not* constant, but rather linear in the number of generated feature vectors. For this reason, we use a modified online version of the Bootstrap Aggregation which is updated in time which is linear in the number of base classifiers. This is explained in Section 5.3.2.

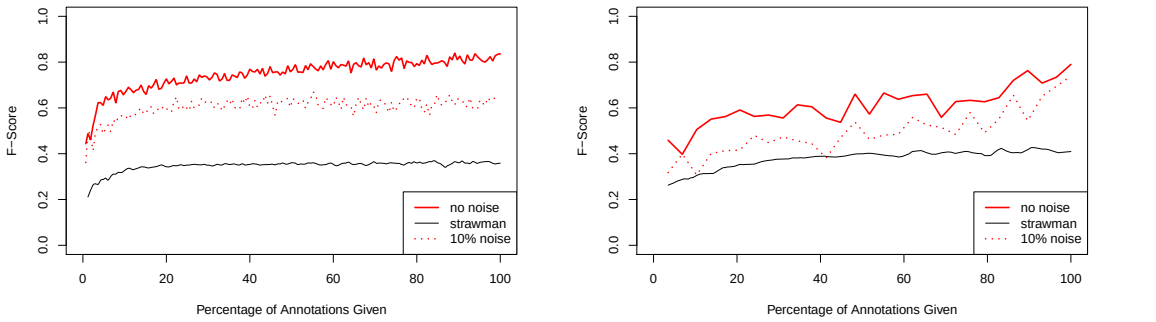
**Data Persistence** Movement data is persisted in constant complexity at the levels of individual accelerometer readings and feature vectors.

## Performance Results

We use the evaluation procedure detailed in Section 4.4.2 for estimating the recognition accuracy of strictly personalised activity recognisers. Using the annotated data collected from the field study, we simulate scenarios whereby activities would be performed in different orders. As in Chapter 4, we focus only on purely individually personalised models. Therefore, we construct training and complementary testing sets only from the pool of annotated data corresponding to one participant at a time.

We divide the recognition performance results in two parts – one for each version of the NAD that was used. The first category corresponds to the initial five participants who provided annotations as directed by the Speculative NAD and by our OAL method. As we pointed out previously, this version of the NAD was overly sensitive and triggered numerous annotation requests, many for already known activities that exhibited small deviations in execution. As a result, the first five per-participant corpora of annotations were relatively large and allowed us to validate the learning strategy over a wide range of training set sizes. Because we obtained several segments for each activity class, this provided us with sufficient testing data for performance evaluation.

The second category of recognition performance results corresponds to the last five



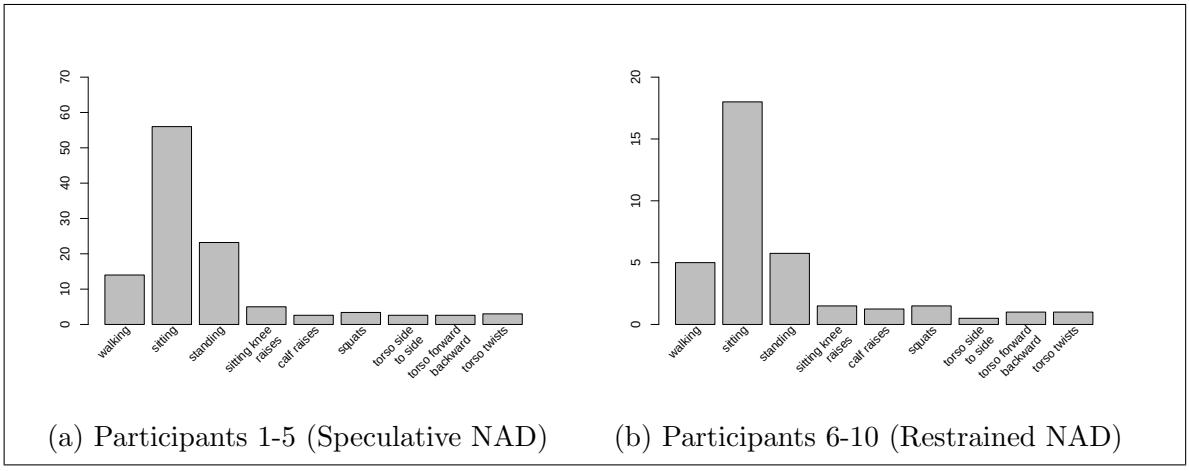

(a) Participants 1-5 (Speculative NAD) (b) Participants 6-10 (Restrained NAD)

Figure 5.8: Average Learning Curves

participants who provided annotations as directed by the Restrained NAD and the OAL method. The second version of the NAD was more lenient in terms of annotation requests because an estimated activity class was considered novel at most once. No other subsequent annotation request for that activity class would be generated by the NAD. While this resulted in less disruption to the participants, sometimes too few annotations were recorded, with some activities being annotated once or even not at all. This negatively impacted our ability to evaluate the activity models. With so little data, models cannot be personalised with high fidelity to the user and performance evaluation was less representative. However, with a longer exposure to the interactive annotation app, given the general trend of model accuracy improvement, we are confident we would have obtained more annotations and we would have noticed recognition performance gains, as for the first five participants.

Fig. 5.8 shows the learning curve of the Bootstrap Aggregator with 30 Naive Bayes base classifiers (solid red line) averaged across all users, for each NAD variant. The model was incrementally built using the unaltered annotations provided by the participants. We compare the performances of this original bootstrapped model with two other models.

Firstly, we calculate the performance of a *strawman* – a simplistic model that makes no informed classification based on the current data instance, but systematically predicts the most prevalent label in the training set (solid black line). The original model substantially outperforms the strawman by 38-47%, which implies that performance




(a) Participants 1-5 (Speculative NAD) (b) Participants 6-10 (Restrained NAD)

Figure 5.9: Distribution of Annotation Requests (Averaged Across Participants)

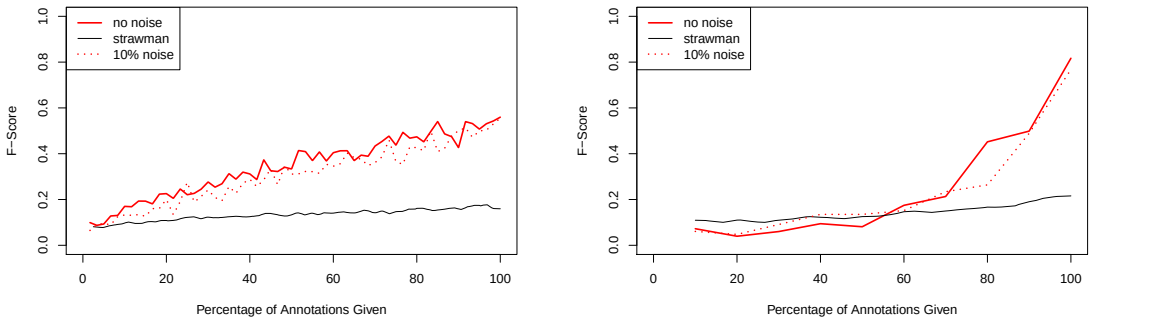
| **NAD** | Speculative | | | | | Restrained | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Participant** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Annotations** | 41% | 47% | 29% | 39% | 40% | 13% | 9% | 14% | 9% | 8% |
| | 39% | | | | | 11% | | | | |

Table 5.1: Volume of annotations as a percentage of total number of detected segments

gains of the original model are not fortuitous, but are the consequence of consolidated learning.

Secondly, we artificially add annotation noise, by randomly altering 10% of the labels in the training set (dotted red line). We observe a substantial decrease in performance when label noise is added. While some label noise may be present in the annotation corpora, due to imperfect segmentation or simply due to user error when reporting an activity, the signal to noise ratio is nonetheless low. Since adding noise hurts performance, we conclude that a significant number of annotations must have been correct and that our annotation method can yield valuable input to personalise an activity model.

The average distributions of activities are shown in Fig. 5.9. As can be seen, the primary activity targeted by annotation requests was *sitting* regardless of the version of the NAD. This was also reported by some of our participants because they reported annoyance as to why the app would insist on this activity. In terms of overall levels of interruption, the NAD plays an important role, which is discernible from Table 5.1.




(a) Participants 1-5 (Speculative NAD) (b) Participants 6-10 (Restrained NAD)

Figure 5.10: Learning Curves Without 'Sitting' Activity

Using the Speculative NAD led to almost four times as many annotations than using the Restrained NAD.

Because a high predominance of a class automatically makes the classification accuracy high, we also analysed whether, by removing this class, the remaining annotations would still construct an accurate activity recogniser. Fig. 5.10 show the F-Scores without the *sitting activity*. In this case, the strawman is outperformed by 40-60% in F-Score. The Restrained version of the NAD limited the number of annotation requests, but was less effective at diversifying the training set because not all activities were discovered for all participants. Consequently, Online Active Learning did not generate many meaningful annotation requests, so the sizes of the annotation corpora were still small by the end of the experiment. While less annotations remain for both sets of participants, the learning tendency of the model improving as annotations accumulate is again evident as there is a noticeable improvement over the strawman. Due to the relative scarcity of labels, adding label noise effects a much smaller change on performance.

In contrast to Chapter 4, where we simulated annotations on public HAR datasets, we did not perform Random Selection in our user deployment. The reason was the very short overall duration of most activities relative to the duration of the *sitting* and *walking* activities as our participants' daily routine is naturally very sedentary. Analysis shows that 43% of the non-sitting and non-walking activities annotations (acquired using the combination of methods described earlier) would have been lost

through Random Selection, if it had been enacted.

## User Feedback

We collected subjective user feedback in order to assess how our design choices, assumptions and limitations of our framework impacted on perceived usability. Firstly, in terms of design choices, we focus on the strategies of generating annotation requests and how these were perceived by the users. Secondly, we implicitly tested how the participants viewed our assumption that the delay in detecting activity changes is small enough to allow for reliable label reporting. Finally, the experimental protocol was limited to a rigid set of activities that could be performed in an office environment. We asked the participants to express their how they felt about performing these activities and whether they would prefer other activities and different contexts.

### *Annotation Requests*

We deployed our user-centric annotation method with a panel of office workers who used the provided smartphone app and also integrated the experimental protocol, which was detailed in Section 5.2.2 in their regular office routine for a day. The most unusual aspect for them was probably dealing with a new type of interruptions which encapsulated annotation requests. Because the replies to interruption were critical to our user study, we gauged how the participants responded and how they think they would respond to such disruptions in different settings.

After presenting the field study details to our participants, but before commencing the physical part of the experiment, we asked them to estimate how they thought they would cope with the volume of annotation requests. Nine out of 10 users said that the volume of requests promised to be manageable or that they did not expect irritability, (i.e. **P3**: *Seems ok, but might be annoying after some time. We will see.*; **P4**: *I think the number and the frequency of input request is fine.*; **P9**: *I think it's ok for me.*). The remaining one out of 10 users said this is acceptable only for a "one-off" experience, but it would be unmanageable on a regular basis (e.g. **P10**: *Sounds reasonable for a one off study participation. Might be too invasive and distracting if*

*updates occurred continuously throughout a normal working period. If the updates are quick (sub 5 seconds) then perhaps it would be fine.*). The aim of this question was to contrast it with an upcoming question in a post-experiment questionnaire so as to gauge the participants' perception on the effort of annotation given progressively increasing amounts of information (before and after doing the experiment). At the end of the physical part of the experiment, this expectation was contrasted with two follow-up questions on their feelings towards the actual volume and, respectively, frequency of the input requests experienced. Eight out of 10 participants pointed out there were some elements of excess in the total number or frequency of requests (e.q. **P7**: *frequency of requests was managable for the first period but annoying and to many for the second. Oddly the second period would have a high frequency for 10 minutes then nothing for ˜30 then a high frequency again.*). Out of these, five of the them initially said they wouldn't mind the level of participation. By contrasting these results, we conclude that the level of interactivity would be too disruptive for long-term use. Some justified their opinion by invoking excessive asking in the second period when there were 20% additional requests or by the requests excessively targeting only a small subset of activities (**P5**: *During the afternoon it was a bit too frequent*; **P7**: *Total number of requests felt like a lot but mainly after the second period, they were more often.*). One notable example of the latter is the *sitting* activity which, when using the Speculative NAD, was deemed as novel on an excessive number of occasions (**P1**: *It was a little annoying as it would often make many requests while I was sitting and not doing anything.*, **P2**: *At times it was a bit much, particularly if continuing the same activity (i.e. standing or sitting, but I didn't mind the total # too much. As mentioned in the pre-study questionnaire, if required all day, everyday, it would be too much.*, **P4**: *Too many request in the sitting or standing position*).

Immediately after the physical part of the experiment, the participants were asked whether they thought the app learned to improve the HAR model. They answered based on information given to them when initially describing the user study (namely that the app would parsimoniously collect annotations in order to learn an activity model and that requests would be more likely when the model is confused between different activities, apart from the element of randomness in the second period) and

on their personal perception of how the system behaved. However, they were shown no objective HAR model performance data, so their response was to a large extent guesswork. This was to test whether the users perceived model improvement as a result of changing patterns in asking for annotations. Seven out of 10 participants said they thought the system learned (i.e. **P2**: *Yes, overall. However, it didn't seem like the torso exercises were picked up as easily by the app.*; **P8**: *I provided the most accurate feedback that I can. I believe that this helped the app to learn.*; **P5**: *Yes because it was "requesting" while I was doing the "specifics"*[9] *exercises. So it knew exactly what I was doing*) while the rest remained neutral or did not assume this happened (e.g. **P7**: *I think for the first period yes, second no.*; **P9**: *I don't know*).

We underline that the answers relating to learning may be biased because of three reasons: First of all, the participants were PhD students and researchers in our school and, even though they were not affiliated with our work, it is expected they have a better understanding than laypeople[10] of what "learning" (as in "machine learning") means. Second of all, although the information in the consent form did not reveal our expectation that learning took place (a hypothesis we were able to test only after the user study was over), the technical background of the participants could have allowed them to intuit that the system learned (even though, at that point, they were not presented with any quantitative evidence). Finally, I (the author of this thesis) was in charge of running the user study the participants were acquaintances of mine. Even though care was taken not to communicate bias (i.e. opinions, guesses as to what the results will be and what they will be indicative of, etc.), the participants may have made statements which they thought could be favourable for our research.

Afterwards, we downloaded the participant's data from the phone and synthesised graphs essentially similar to those in Fig. 5.8 and 5.10 (without adding noise or showing the strawman) which portrayed the personalised recognition performance of each participant. We explained that the graph represented the evolution of accuracy as annotations were provided, but we did not influence the participant with our opinion of whether the app managed to learn or not. Rather, given this new piece of information,

[9] sic

[10] Non-professional within a field.

participants were asked again if they thought the system managed to learn. Seven out of 10 believed that the graphs portrayed improvement(i.e. **P1**: *Yes, according to the plots I saw, the app was able to predict my activity better.*; **P3**: *Definitely yes.*; **P8**: *I think my feedback helped the app learn, since most of my activities were recorded and adequately analysed and represented in the results produced.*), with all five initial participants who gave the most annotations being in this category. Three out of 10 participants, all of whom used the second version of the NDA and provided a much smaller number of annotations, were either negative (**P10**: *Not particularly - there does not appear to be a significant upward trend in the accuracy of the model over time, which indicates that the app is not learning well from my feedback.* ) or neutral (**P9**: *I am not sure*).

### *Annotation Delay*

The app performs stream-based segmentation through the use of the online segmentation method explained in Section 3.4.1. If $L$ is the width of the segmentation window, then a change in activity is detected with a delay of $L/2 + 1$ frames. Given our parametrisation of $L = 6$, in theory, the delay is four frames, with the fifth frame being the last of the finished segment.

If the activity for which an annotation is requested last took place five frames in the past, then, in the preprocessing stage which operates a sliding window of length 5s with 2.5s overlap, this results in a delay of 15 seconds between the instant in time a participant switched activities and the instant in time the change in activity is detected by the app. This figure is still subject to uncertainty, according to Section 5.3.3, so, prior to starting the physical monitoring, we informed our participants that they should consider a delay of 10-20s when responding to annotation requests.

After running the experiment we asked each participant two questions relating to the delay.

Firstly, when asked if they thought they were able to reliably factor in the delay, the majority pointed out that, even though they claimed it didn't create confusion, the pressure on their memory was uncomfortable (e.g. **P1**: *In general, they were timely.*

*I think I would have preferred to arrive the moment I changed activities.*; **P3**: *Maybe 5-10 sec instead of 10-20. Especially after the special activities (squats, etc.)*; **P5**: *Slightly sooner, after the device realised a "change" in the type of activity*; **P7**: *Sooner within 1-2 seconds maximum (although I may just be impatient), however it is hard to judge 10-20 seconds*).

Finally, when asked whether they would have coped with longer delays, the majority expressed their disapproval of the prospect (i.e. **P1**: *No, I think that would have made the study a lot more difficult.*; **P2**: *No! I have poor memory.*; **P8**: *No, it would be quite difficult remembering the activities.*).

In terms of usability, we conclude that a shorter segmentation delay may lead to more productive user involvement. In objective terms, the accuracy of the system improves substantially from user provided annotations, as shown in Section 5.5. This means that the annotations are of high quality and this necessarily shows that the users themselves can provide annotation labels for their own activities using nothing but their short-term memory as a source of ground truth. Together with an altered design that would encourage user involvement, we argue that our proposed method of collecting annotations could enable bootstrapping personalised HAR models on a longer term than the duration of our experiment.

### *Activities*

We gauged the emotional response of participants relating to the activities they performed and relating to the feasibility of performing them in an office environment. After their experience with an interactive activity monitor in a context defined by our experimental protocol, we asked the participants to extrapolate and suggest potential changes to the app and to the context in general.

In terms of activities, we draw four conclusions from the questionnaires:

- In terms of the fixed set of activities specified in the experiment protocol, the participants had mixed feelings of incorporating these into their daily routine. Four of the 10 thought the activities were useful (i.e. **P9**: *Yes, I am very interesting how many activities I did a day*; **P10**: *Yes - as previously mentioned*

*- I have noticed feeling more energic and therefore productive, lpus the impact on my day would be minimal as I already take several short breaks at regular intervals throughout the day (Pomodoro).*), but five of the 10 found some context elements inappropriate (i.e. **P6**: *I might include this type of learning in the mobile from my activities as a privacy issue so I might be a bit cautious where and how much I let it learn.*). For example, certain activities were deemed as not suitable to offices or the duration of exercise was longer than they would normally perform (e.g. **P1**: *No, not really. The activities are kind of out of place in my work environment.*; **P6**: *Yes, although some activities might look not appropriate in the office while others are working around especially squats.*). Out of the first five participants, those who used the Speculative version of the NAD, which, as explained in Section 5.3.1, was unintentionally prone to generating numerous annotation requests, four gave negative feedback, possibly owing to irritation accumulated from dealing with the high volume of requests throughout the day.

- Five of the 10 respondents said they would prefer the system to learn more about the same activities (i.e. **P5**: *Yes it should focus on the same activities in order to predict and learn better*), while the others were opposed to the idea of reinforcing the same activities (e.g. **P8**: *No, I will prefer that it captures other activities in addition to the current activities.* **P10**: *Not necessarily. I think users should be able to pick activities from a larger set.*). Seemingly paradoxically, these were the first five participants, who were more negative towards the activities in the protocol and who previously declared themselves against performing these activities. While our intention was to gauge whether they would provide more input, it may be that the users would expect gains in recognition performance with less input from them. Also, because of the relatively large quantity of annotated data, they were able to notice the greatest improvement, so this may have biased their potential interest for the system to learn more about the same activities. The other five users, who used the Restrained NAD, witnessed a smaller improvement in recognition accuracy. They reported that they would not be interested to invest further in the same activities. Probably, the small improvement was not enough to gratify them, so these participants were not made aware of the

full potential their annotations can have on model performance.

- Participants pointed to a wide range of physical activities (lunges, press-ups, running, jumping, using the stairs, sit-ups, etc.) in terms of what activities they would like to see in the app. Additionally most users preferred the flexibility of specifying what activities they would like monitored. Given the very little consensus on the activities, such an interactive activity monitor would benefit from being able to incrementally discover new activity classes, instead of relying on the users to specify what activities they prefer. However, the ability to reliably discover new activities can be at odds with an acceptably low involvement on behalf of the user. This can be seen in the behaviour of the Speculative NAD, which, in our experiments, carried a high true positive rate, but also a high false positive rate, as can be deduced from Table 5.1. In order to bypass the problem of discovering novel activities, one may rely on the user to *pro-actively* provide initial annotations for novel activities, as discussed in Section 2.2.2. In this case, it is possible to use just Eq. 4.1 to improve upon known activities. This is relevant from a user's tolerance point of view, because, as we show in Chapter 6, it is possible to fine-tune the behaviour of Eq. 4.1 so that it meets a user's propensity towards annotation involvement.

- All participants pointed out that such an application could be used in other contexts. While diverse, almost all of their suggestions can be grouped under the fitness (**P2**: *Yes. I think other exercises like lunges (with sensors on two legs) or pushups (they are called something else here) would be great additions.*), sports (**P1**: *I would also like it to learn about running, cycling, and perhaps some other types of stretches.*; **P5**: *As in answer b) it would be interesting in sport/gyms activities.*) and medical rehabilitation (**P10**: *The flexibility to learn series of activities constituting either a fitness routine or a physio therapy session would be a very good addition.*) categories suggested in our question, as possible examples. Three categories were mentioned in the question text ("specific fitness routines, sports, medical rehabilitation"), so this might be a source of bias in the users' answers. However, the majority users did not dismiss these categories, so these proposed may be targeted by such a monitoring system. While we aimed

our activity monitor at an office-centred context, the participants considered the activity monitor to be applicable to other contexts as well (e.g **P6**: *I might include this type of learning in the mobile from my activities as a privacy issue so I might be a bit cautious where and how much I let it learn.*).

Because the app automated a part of the annotation process by detecting segment boundaries and identifying the most useful segments to be annotated, we were able to use a mechanism to collect annotations involving a single tap and, if needed, a screen scroll. While we did not explicitly ask the participants to express their views towards this method, Participant 2 pointed out that scrolling was a source of some errors and that the participant would have preferred a smaller list of "educated guesses" for activities (**P2**: *I think the scrolling to select the activity could lead to mistakes. Perhaps a pop-up educated guess list of activities instead? Oh, and better sensors (in terms of comfort while wearing them).*).

# Conclusions

In this chapter we discussed the design, implementation and evaluation of a user study aimed at online bootstrapping of personalised activity models from user-provided annotations collected using Online Active Learning.

We decided upon a realistic experimental protocol and a deployment in a naturalistic environment. We primarily aimed to replicate an "in the wild" scenario where participants were not supervised during the experiment, but rather they were left to act naturally. We settled for a set of activities that seemed appropriate for exertion in the environment of the deployment – at the office. In retrospect, the choice of activities was fit for the purpose. Despite the wide range of participants' opinions on what would constitute an ideal set of light activities for an office environment, there were very few instances where participants singled out activities which were out of place. Moreover, some participants found adhering to the experimental protocol physically beneficial for them.

This chapter builds upon the techniques in Chapter 4 by employing applicable analysis methods to the type of activities under scrutiny. We moved from the offline simulations

of online methods in Chapter 4 to an actual deployment using an Android app and worn wireless accelerometers that provisions the collection and annotation of movement data from parsimoniously solicited user-provided input. Given the short timescale of the experiment, we accelerated the Online Active Learning method from Chapter 4 by using it in parallel with a Novel Activity Detection (NAD) method for discovering novel activities. While we had some success with the NAD, it was mixed and we argue next why relying solely on the NAD (instead of using it in conjunction with the OAL) would be detrimental. On the one hand, the Speculative NAD discovered all activity classes, but it was also too sensitive and generated excessive annotation requests (as reported by the participants) for known activities as well. The participants who used this NAD version expressed frustration at the amounts of input solicited. The Speculative NAD could be turned off during the times when a large number of user interruptions would become disruptive. During these times, if a small number of interruptions are permitted, one could rely on OAL to derive only a small number of critical annotations.

On the other hand, the Restrained NAD, by design, limits the number of annotation requests, but it proved to be too insensitive to collect initial annotations for all activity classes. Consequently, the Online Active Learning method had a poor base of annotations from which to improve. Most personalised activity models which benefited from the Restrained NAD registered smaller gains in performance than when using the Speculative NAD.

Contrasting the objective performance results with the feedback provided from the participants, we observe there is a problematic trade-off between activity model performance and user involvement. This conflict has been made even more apparent by the limited duration of our user study because we tried to concentrate a meaningful number of annotations within a single day at the office.

The quantitative results in Section 5.5 show that self-provided annotations are a valid solution to the problem of bootstrapping personalised activity models. The results in Chapters 3 and 4 obtained from very similar online techniques point in this direction, but the results were obtained from simulations operating with offline and carefully curated annotations. In this chapter we applied a very similar data processing pipeline

on user provided annotations without any external curation or intervention. Our results show that bootstrapping from scratch is possible under naturalistic conditions and, if a sufficient number of annotations are provided, clear performance gains of personal models can be observed. However, as opposed to Chapter 4, we do not have a large enough set of annotations that would allow us to replicate the same Active-versus-Random analysis. In fact, as we have shown, performing Random Selection would have been a mis-allocation of research effort as most activity classes would have suffered from under-annotation. Moreover, based strictly on the contributions in this chapter, due to the mechanisms used to trigger annotation requests which were only partially adaptive (20% random annotation requests in the second half of each physical experiment; unexpectedly high sensitivity of the first version of the Novel Activity Detector) we cannot merit solely Online Active Learning for the gains in recognition performance. Our results show that model personalisation from user annotations is achievable regardless of the mechanism that led to the acquisition of each annotation. Nonetheless, the learning curves in this chapter and the previous exhibit clear positive learning profiles, so we are confident that, given a longer term exposure to annotation requests, accurate models can be obtained even from pure Online Active Learning.

The qualitative results in Section 5.6 reveal how participants perceived the design of the experiment (Section 5.6.3) and the implementation details that had a noticeable impact on usability (Sections 5.6.1 and 5.6.2). Subjective feedback can be used not only to understand how the design of the system is affecting usability, but also to highlight existing issues and to act to correct them. We have done just that mid-way in our experiment by changing the Novel Activity Detection component as was explained in Section 5.3.1.

Overall, valuable insight can be drawn from our user study. Firstly, we have shown that it is indeed feasible to deploy a fully online interactive annotation pipeline for bootstrapping personalised activity models. Secondly, noticeable improvements of the bootstrapped model can become apparent in several hours – the duration of our deployment. Thirdly, we have gauged our participants' subjective perceptions and we have shown how the interactive activity monitor can be altered to meet these expectations. These insights can inform future designs such that users themselves can

tune the behaviour of the app.

# 6

# Online Active Learning with Budget Constraints

## Contents

# Introduction

In Chapter 3 we compiled a fixed schedule of times when to ask for segment annotations according to a budget specification. In Chapter 4, annotation decisions were deferred until runtime so that an Online Active Learning heuristic function could inspect individual segments and inform the decisions to annotate the segments. Budget-based annotation is useful for meeting varying user willingness to provide annotations, whereas OAL-based annotation optimises the performance gains from individual annotations. The methods exhibit complementary advantages, but, without modification, they cannot be used in tandem.

In this chapter we propose an overarching method that unifies the previous two annotation methods. We modify the Online Active Learning method used previously and incorporate budget-based restrictions into the annotation decisions. In doing so we maintain, on the one hand, a budget based method's flexibility in coping with user preferences towards the provision of annotation, and, on the other hand, a performance boost due to Online Active Learning (relative to Random Selection). Effectively, a budget spending strategy is adhered to, but some deviations from the budget are allowed. With this degree of limited freedom, we can prioritise certain annotations, according to Online Active Learning, which improve model performance over the annotations that would be chosen if the budget specifications were adhered to more strictly. We therefore still perform informed annotation through Online Active Learning, but we also allow the user to specify a budget configuration to which OAL should adhere to.

We describe and evaluate a budget-oriented annotation method which defers all annotation decisions until runtime. The decisions to annotate are made by a modified Online Active Learning heuristic function, which is similar to the one described in Section 4.2.1. The main distinction is that the method now aims to balance performance gains, like the ones registered in Chapter 4, with close adherence to a user-specified annotation budget configuration, as in Chapter 3. The balance is achieved by continually fine-tuning the OAL heuristic function parameters so as to encourage (make more probable) or dissuade (make less probable) annotation requests based not only on the importance of individual annotations, but also according to the probability that

budget spending is going to be on target, according to a previously established budget specification.

### *Contributions*

The contributions of this chapter are four-fold. Firstly, we improve upon the work from Chapter 3 by formalising the budget and by no longer making annotation decisions according to a fixed and pre-defined schedule. Instead, all annotation decisions, which aim to meet a budget spending specification, are made online, i.e. while the system is running. To this end, we introduce the notion of a *target budget*, which informs the annotation heuristic by altering the degree to which annotations are encouraged or discouraged. The target budget is used as a "moving target" which is judiciously varied so that budget spending is effected according to the specified budget size and strategy. We provide a general closed-form theoretical expression for *setting* the target budget which can be used to match any arbitrary distribution of annotation requests. The mechanism of setting the target budget is generic enough to work with a wide range of heuristic functions that are flexible enough to be able to attain any such budgets.

Secondly, we devise a heuristic annotation function that can *attain* a target budget, using piece-wise linear approximations, and we couple it with the previous target budget setting procedure. Specifically, we adapt our Online Active Learning method to work within the constraints of a set target budget while still securing performance gains over Random Selection. To achieve this, starting from a set target budget (described earlier), the parameters of the annotation heuristic are optimised so that the target budget can be attained. The heuristic's operating parameters are modified so that ideal budget spending is the most probable, but if there are deviations from ideal spending, then the heuristic's parameters are tuned to compensate. Temporary under-spending is met with increasing the probability of annotating segments, while temporary over-spending is addressed by decreasing the probability of annotating. The general behaviour of the heuristic is that the criticality threshold with which segments are deemed worthy of annotation is varied in order to attain the set budget. Under this "*set-attain*" budget spending control mechanism, the result is an accumulation

of annotations whose distribution in time approximately matches the desired budget configuration. In addition, because annotation decisions are informed by an Online Active Learning heuristic, we still expect performance gains greater than those that would be obtained with Random Selection.

Thirdly, we apply the budget-constrained OAL method to the two probabilistic budget strategies discussed in Chapter 3 (Uniform Random and Exponential). This serves not only as a means of evaluating our budget-constrained OAL method, but also as a demonstration of how it could be applied to any budget strategy. We evaluate Online Active Learning with budget constraints on the HAR datasets we used in Chapter 4: Opportunity for non-periodic activities and PAMAP and USC-HAD datasets for periodic activities. Despite the budget constraints, our results still show performance improvements of Online Active Learning over Random Selection. In addition, our results also show that the distribution in time of actual annotation requests closely matches the ideal distribution corresponding to the budget specification. Consequently, we conclude that Online Active Learning can be constrained with the user's inclination towards annotation provision and that specific performance gains are still possible.

Finally, we experimented with an additional step which introduces control over how tightly Online Active Learning can be further constrained to match the expected budget spending strategy. Our results show that a wide range of possibilities of constraint are possible: from near zero additional coercion over the *set-attain* procedure mentioned earlier to gradually increasing coercion up to a virtually deterministic budget spending strategy that adheres almost exactly to the ideal budget distribution.

# Method

As in previous chapters on Online Active Learning, the annotation method in this chapter also operates on a stream of activity data and yields annotation decisions in accordance to the data being examined. Unlike Chapter 3, no annotation decisions are pre-computed before the monitoring starts. It is necessary to defer annotation decisions until runtime for the following reasons:

- At its core, the method operates a pure Online Active Learning annotation

heuristic (Eq. 4.1) and, so, all decisions must be made online.

- The method corrects the deviations in spending made by Online Active Learning (which is budget-agnostic) so that the specified budget is best approximated. We do not schedule annotation requests with the intent of correcting the budget spending without observing the stream of data because doing so would deny Online Active Learning the possibility of improving model performance.

- A part of the mechanics of the system is monitoring the classifier prediction confidence over a recent horizon of predictions. As explained later, these historic confidence values are used to make assumptions about the future of the stream and, consequently, are used to inform the asking probabilities output by the OAL heuristic. These confidence levels, which depend (1) on the data seen and (2) on the annotations made by the current point in time, cannot be predicted without observing the activity data first.

### *Mathematical Considerations*

As in Chapter 3, we consider a budget specification as a triplet (*Horizon*, *BudgetSize*, *BudgetStrategy*) where, intuitively:

**Horizon** is the interval of time the user is willing to reply to *occasional* annotation requests.

**Budget Size** is the total number of annotations the system is going to ask the user until the **Horizon** expires.

**Budget Strategy** is a theoretical distribution of annotations over time which models how the total number of annotations (**Budget Size**) is distributed in time until **Horizon** expires.

More formally, we consider a budget specification as a triplet $(t_{horizon}, B_{total}, f(t))$ where the budget size is $B_{total}$ and the budget spending strategy $f$ is a probability density function (PDF) *of one annotation being asked* over the interval of time $[0, t_{horizon}]$. $f$

is defined as follows:

$$f : [0, t_{horizon}] \rightarrow [0, 1]$$

Since $f(t)$ accounts for only one annotation being requested at time $t$, $B_{total} \cdot f(t)$ accounts for all annotation requests being made at time $t$ (this is done simply by multiplication because the annotation requests are i.i.d.).

Let $F(t) = \int_0^t f(x)dx$ be the cumulative distribution function corresponding to $f$ [159]. Here $F(t)$ is the probability that one particular annotation requests has been made up to and including time $t$. Again, given that all $B_{total}$ annotation requests are i.i.d., the following formula

$$B_{total} \cdot F(t) = B_{total} \cdot \int_0^t f(x)dx$$

models the ideal cumulative distribution in time of all annotation requests corresponding to the budget specification $(t_{horizon}, B_{total}, f(t))$. Intuitively, because the co-domain of $f$ and $F$ is $[0, 1]$, the co-domain of $B_{total} \cdot f$ and $B_{total} \cdot F$ is $[0, B_{total}]$. This means that the budget specifies how all $B_{total}$ annotation requests are distributed by either $f$ (or $F$, by direct implication) until the time horizon.

### *Step 1: Setting a Target Budget*

We also have the reverse relationship [159]:

$$f(t) = \frac{dF}{dt}(t) \tag{6.1}$$

If $F$ is infinitely differentiable, then it can be expressed using a Taylor series [160] expansion around a point in time $\tau$:

$$F(t) = \sum_{n=0}^{\infty} \frac{1}{n!} \cdot \frac{d^n F}{dt^n}(\tau) \cdot (t - \tau)^n$$

We use the first degree[1] ($n \leq 1$) approximation of $F$:

$$F(t) = F(\tau) + \frac{dF}{dt}(\tau) \cdot (t - \tau)$$

[1]Even if $F$ is not infinitely differentiable, its first order derivative is always well defined by construction, according to Eq. 6.1.

which, according to Eq. 6.1, becomes a piece-wise linear approximation:

$$F(t) = F(\tau) + f(\tau) \cdot (t - \tau)$$

Suppose that $t$ is the current timestamp and $\tau$ is the last timestamp when an annotation has been made with the budget spent so far being $B_{spent}$. If the budget strategy curve had been followed exactly, the budget should have been $B_{theoretical} = F(\tau) \cdot B_{total}$. However, if the current budget expenditure does not meet the theoretical expectations ($B_{spent} \neq B_{total} \cdot F(\tau)$), we estimate the target budget $B_{target}$ at $t_{horizon}$. In general, we have the following approximation:

$$B_{target} = B_{spent} + f(\tau) \cdot (t_{horizon} - \tau) \cdot (B_{total} - B_{spent}) \tag{6.2}$$

The right-hand side of Eq. 6.2 can be renamed as follows:

$$B_{target} = B_{spent} + B_{remaining}$$

This means that, in the remaining time, we should aim to spend

$$B_{remaining} = f(\tau) \cdot (t_{horizon} - \tau) \cdot (B_{total} - B_{spent})$$

budget units. $B_{target}$ is therefore a "moving target" whose value is updated after every annotation request according to Eq. 6.2.

The target budget $B_{target}$ does not have to be the same as the budget size $B_{total}$, the latter being part of the budget specification. In fact, as shown in Eq. 6.2, in order to obtain arbitrary distributions of annotation requests, the target budget can generally be equal to or greater than the total budget.

### *Step 2: Attaining a Target Budget*

Having set a target budget, one must now attain the target budget. We now show how to fine-tune the parameters of the annotation heuristic so that the remaining $B_{target} - B_{spent}$ budget units would be spent within the remaining time horizon. The mechanism for attaining the set target budget generally assumes that the remaining annotations are going to be distributed *uniformly* in time, in accordance with our

previous piece-wise linear approximation. This, however, does not result in loss of generality – we argue next (and demonstrate with simulations) that this mechanism can be used to approximate arbitrary distributions, not just linear ones. Because the target budget is not fixed, but instead refined at every timestamp and with each newly made annotation, as explained previously, the timing of the annotations results in a piece-wise linear approximation of the ideal budget distribution.

### Random Selection

For Random Selection, we use a simple heuristic that simply yields a positive annotation decision with probability $f_{RS}(\tau)$, which depends on when the annotation decision must be made so that the budget specification is met, but which is independent of the current segment for which an annotation is requested:

$$p_{ask} = f_{RS}(\tau) \tag{6.3}$$

For Random Selection, in order to make uniform spending the most probable outcome, then the asking frequency is set to

$$f_{RS}(\tau) = \frac{t_{horizon} - \tau}{B_{target} - B_{spent}}$$

so that the outstanding number of annotations is distributed approximately uniformly in the remaining interval of time.

### Online Active Learning

For Online Active Learning we use Eq. 4.1 as in Chapters 4 and 5. We take advantage of the known monotonicity of the heuristic function with respect to the $\gamma$ parameter as follows: As shown in Section 4.2.1, for a fixed level of classification confidence, the heuristic $p_{ask} = exp(-\gamma p_{pred})$ is strictly decreasing with $\gamma$. Meeting the budget constraint therefore entails fine-tuning the $\gamma$ parameter so that $B_{target} - B_{spent}$ would be ultimately spent from the current time onward. To do this, we maintain a short history of the most recent classification confidence levels. We set $\gamma$ so that, under the assumption that the same average confidence will reappear in the future, the

most probable number of annotations will be $B_{target} - B_{spent}$. This is a best-effort approximation which relies on the most recent confidence levels as good estimates.

Even if the levels of classification confidence vary substantially and the actual budget spending begins to deviate from the ideal one, the incentive to make or abstain from annotations will rapidly become increasingly pronounced (as a result of under- or over-spending). This would lead to a re-alignment of the actual spending toward the ideal. Another cause of corrective action is the number of segments remaining to be monitored: after every seen segment (not necessarily annotated), there are fewer segments from which to annotate (the difference $t_{horizon} - t$ becomes smaller), so $\gamma$ will be re-evaluated to account for the reduced number of segments from which to annotate.

The most likely number of annotation requests is strictly monotonic with the $\gamma$ parameter, so we use a binary search method, illustrated in Algorithm 2, to calculate $\gamma$ in order to meet the target budget.

**input :**

*targetBudget* /* the target budget as calculated in Section 6.2.2 */

*pConfHistory* /* history of classifier's recent confidence levels */

**output:**

$\gamma_{Best}$ /* $\gamma$ which closely attains *targetBudget* */

**initialization:**

$\gamma = 0.5$

$\gamma_{min} = 0$

$\gamma_{max} = +\infty$

$diffBest = +\infty$

**for** *fixed number of iterations* **do**

$pAsk \leftarrow mean\{askHeuristic(pConf, \gamma)$ **for** $pConf \in pConfHistory\}$

$numAnnotExpected \leftarrow pAsk \cdot numSegmentsLeft$

$diffExpected \leftarrow |numAnnotExpected - targetBudget|$

**if** $diffExpected < diffBest$ **then**

$diffBest \leftarrow diffExpected$

$\gamma_{Best} \leftarrow \gamma$

**end**

**if** $numAnnotExpected < targetBudget$ **then**

$\gamma_{max} \leftarrow \gamma$

$\gamma \leftarrow (\gamma + \gamma_{min})/2$

**end**

**if** $numAnnotExpected > targetBudget$ **then**

$\gamma_{min} \leftarrow \gamma$

**if** $\gamma_{max} = +\infty$ **then**

$\gamma \leftarrow \gamma \cdot 2$

**else**

$\gamma \leftarrow (\gamma + \gamma_{max})/2$

**end**

**end**

**end**

**Algorithm 2:** Searching for the optimal value for $\gamma$ which attains the target budget.

### *Preliminary Conclusions*

So far, we have proposed a budget-based annotation method which can closely mimic any cumulative distribution function by (1) setting a convenient target budget and (2) attaining the target budget by continuously tuning the parameters of the annotation heuristic. In what follows, we look at two particular distribution functions: the *Uniform Random* and, respectively, *Exponential* strategies presented in Chapter 3. These distributions have specific expressions for $F$ and $f$ which we use in Eq. 6.2. These are the only stochastic budget spending strategies we explored in Chapter 3; therefore that they are the only ones that could potentially be improved upon by using Online Active Learning. The other strategies are deterministic and they are not amendable to Online Active Learning because the annotation decisions cannot be influenced at run-time.

We incorporate our budget-driven annotation method into a similar pipeline to the one in Fig. 4.3 from Chapter 4, but this time with an additional mechanism for meeting budget constraints. The result, as shown in Fig. 6.1, illustrates how the overall framework remains largely unchanged (the novelty – the effect of the budget specification – is delineated by the dashed rectangle), except for how annotation requests are now affected by the budget specification.

### *Uniform Random Budget Spending Strategy*

For the *Uniform Random* strategy, given that, overall, there should be $B_{total}$ annotations to be made when $t_{horizon}$ segments are going to be seen, then the frequency of asking for annotations is

$$f_{unif}(t) = \frac{B_{total}}{t_{horizon}}$$

In this case, the distribution of annotation requests becomes

$$F_{unif}(t) = \frac{t}{t_{horizon}}$$

By applying Eq. 6.2, we obtain $B_{target} = B_{total}$. Fig. 6.2 exemplifies the spending of budget units according to a uniform distribution. The blue line is the ideal spending strategy, while the black continuous line represents a hypothetical example of an actual

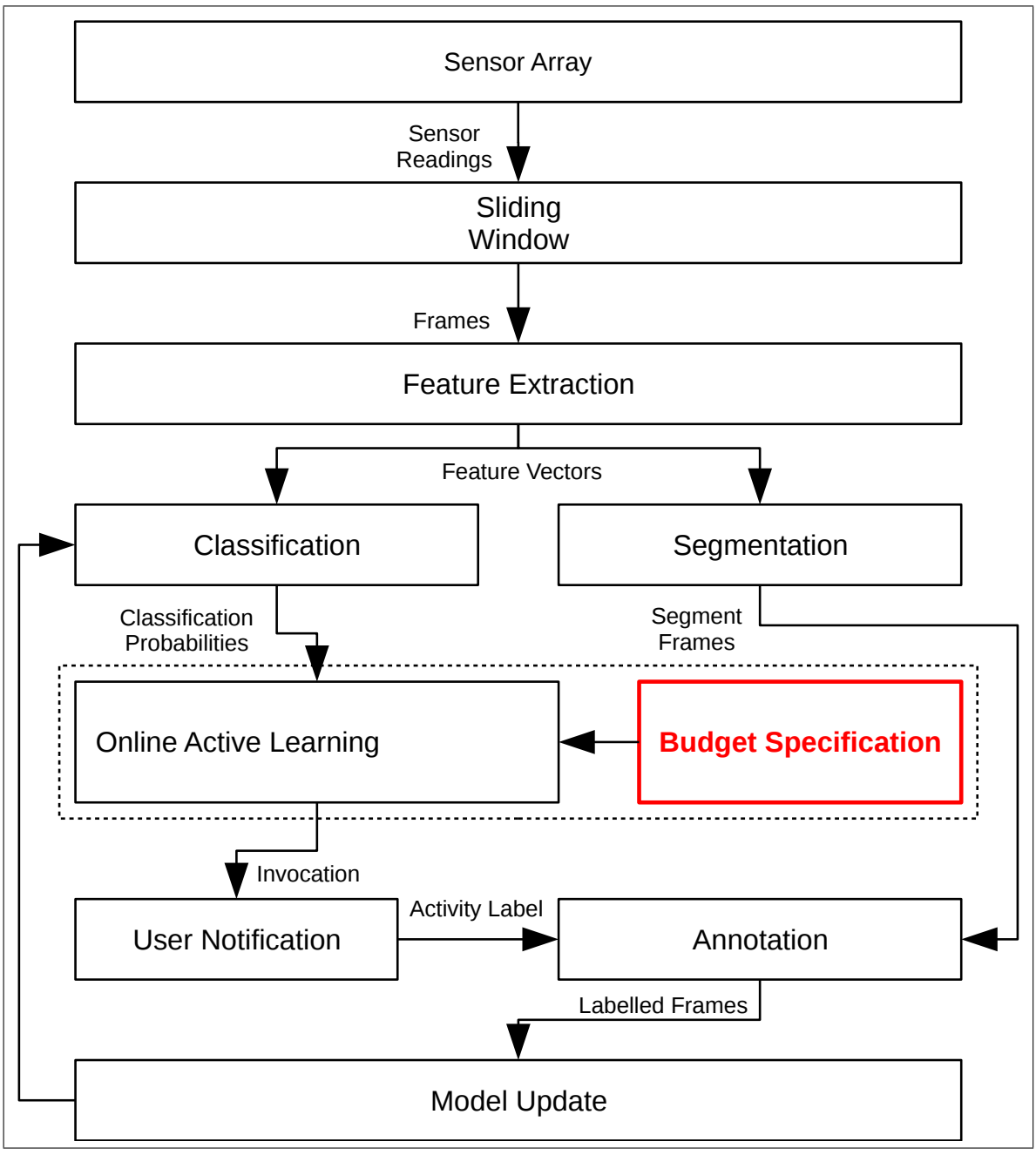



Figure 6.1: Online Active Learning with Budget Constraints; System Schematic

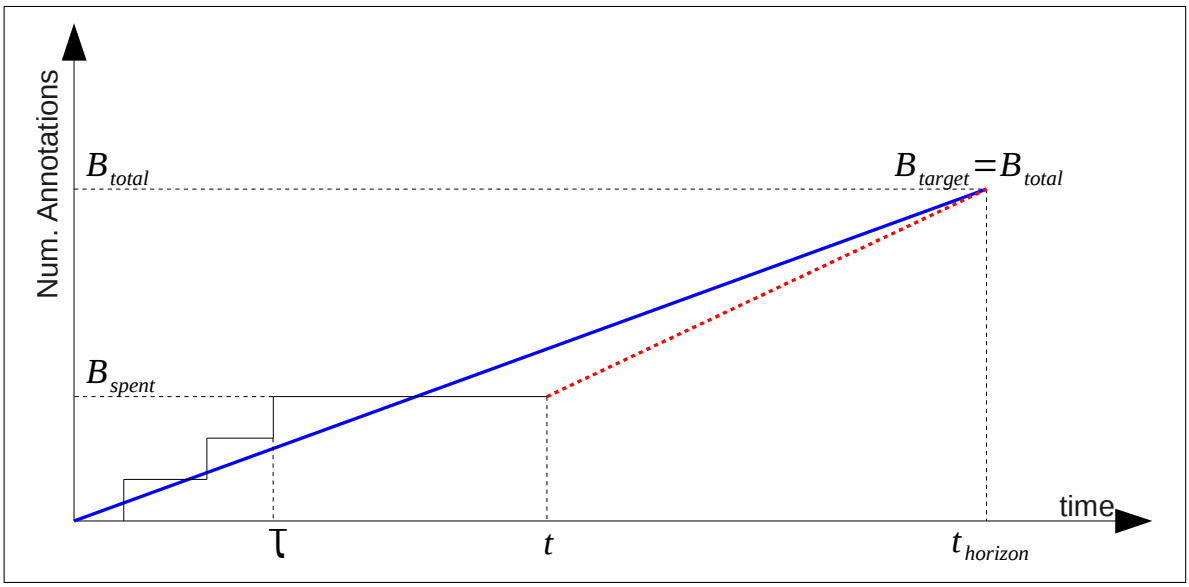



Figure 6.2: Online Learning under Budget Constraints - Uniform Distribution

spending of budget units. The red dotted line illustrates how the remaining budget units should be spent from the current timestamp $\tau$ onwards until the time to spend the budget runs out at $t_{horizon}$. In the case of a uniform distribution of annotations, because the target budget is fixed, one should always aim to request $B_{total}$ annotations in the time remaining.

### *Exponential Budget Spending Strategy*

For the *Exponential* strategy, the PDF of asking for each annotation is

$$f_{exp}(t) = \lambda \cdot e^{-\lambda \cdot t}$$

and the distribution of annotation requests is

$$F_{exp}(t) = (1 - e^{-\lambda \cdot t})$$

and is illustrated in Fig. 6.3 with the blue line. As before, the actual distribution of annotations in this example is represented by the continuous black line. The dotted red line illustrates the expectation at timestamp $\tau$ of how to spend the budget until the end. Regardless of the budget spending strategy, the mechanism to approximate the ideal spending is constant: at every timestamp the heuristic function is tuned so that the target budget is attained. In this case, $B_{target} \geq B_{total}$ which means that in

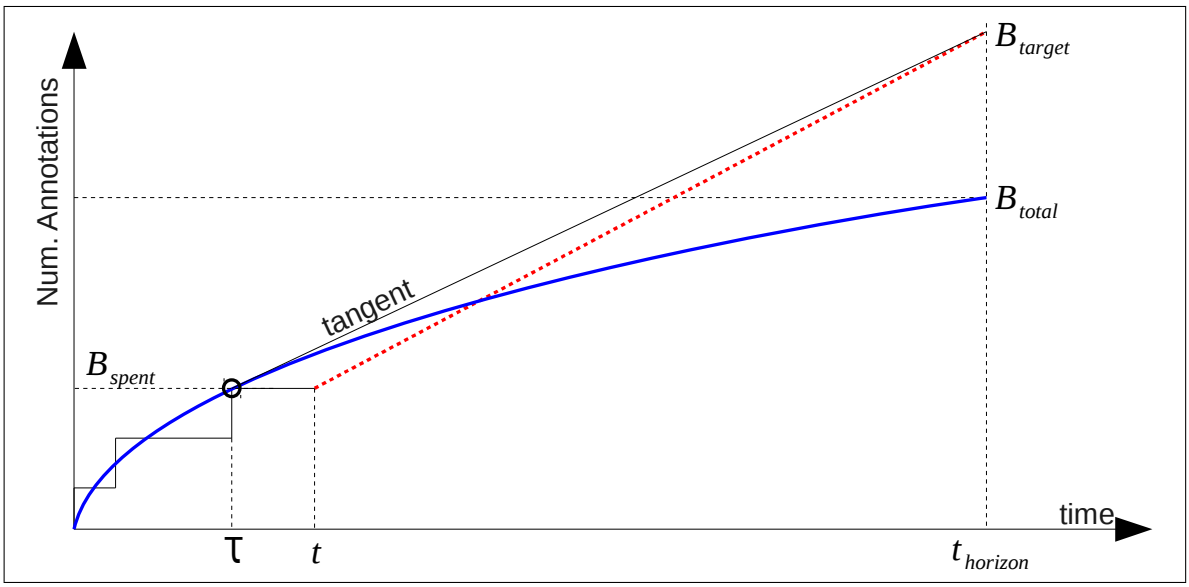



Figure 6.3: Online Learning under Budget Constraints - Exponential Distribution

order to increase the density of annotations at the beginning, one needs to attempt to constantly overshoot the total budget. However, as more annotations are requested, the slope of the tangent (given by $f_{exp}$) decreases as well and the target budget $B_{target}$ will progressively drop towards $B_{total}$.

## Simulations

We now explain how we apply the budget restrictions from Section 6.2 to our proposed interactive annotation framework described in Section 4.2. As in Chapter 4, we use the Opportunity dataset to show that the method benefits the recognition performance of non-periodic activities. We also show how performance can be improved for periodic activities as well, by using the PAMAP dataset. In order to save space, we do not include the results for the USC-HAD dataset of periodic activities, as we did in Chapter 4. As we discuss later, the results from the simulations for the USC-HAD dataset are qualitatively identical to those for the PAMAP dataset.

In order to make the history of confidence levels representative, we start with a training set with one example for every activity class. This ensures that we avoid the work-arounds from Section 5.3.1 to the *Ignorant Classifier* problem. These would complicate estimating the annotation heuristic function parameter and can skew the desired distribution of annotation requests.

The machine learning pipelines remain largely unchanged and we still use Eq. 4.1 to solicit informed annotations or, alternatively, Eq. 6.3 to request annotations at random, but now according to budget spending constraints. In order to follow a given budget strategy, a size of a target budget is computed and continually refined, as shown in Section 6.2.2. These budget targets are approximately met via annotation heuristic parameter regulation, as explained in Section 6.2.3.

We simulate a stream of activities by replaying a fixed number of segments (2000). We maintain a recent history of classification confidence levels, we establish target budgets according to Eq. 6.2 and, using the confidence history, we fine-tune the heuristic function so that the target budget is met. We consider the following simulation cases:

- For the Uniform strategy, we use a budget size of $B = 200$.
- For the Exponential strategy, we use a budget size of $B = 200$ and vary the $\lambda$ parameter, which intuitively controls the steepness of the decay, to $\lambda \in \{2, 3\}$.

### *Results for Non-periodic Activities*

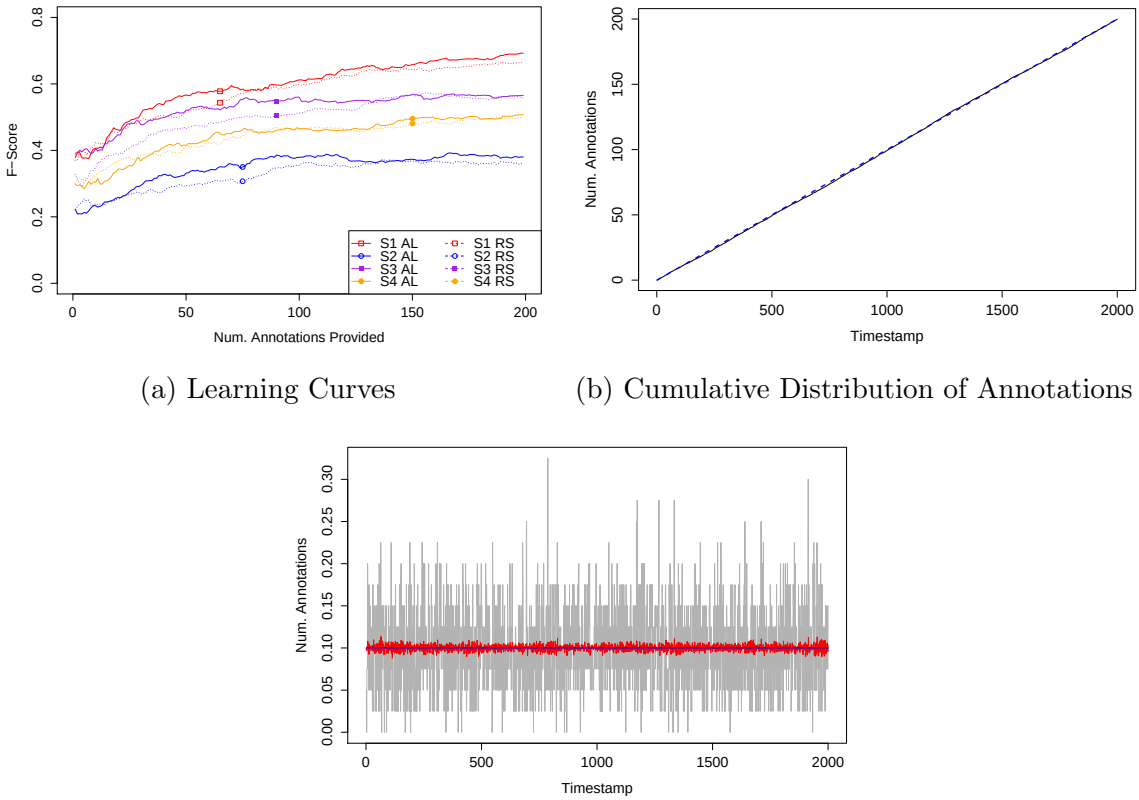
Fig. 6.4a illustrates the learning curves for all participants in the Opportunity dataset for the Uniform strategy with a budget size of $B = 200$ annotations. As in Chapter 5, for the majority of points on the learning curve, the model bootstrapped from informed annotation requests outperforms the model bootstrapped from random annotation requests. As in all previous scenarios, we constructed fully personalised models. For the purposes of model building and model evaluation, we considered every participant's data in isolation. We enacted 10 repetitions of the simulation procedure and, for each participant in turn, we averaged the results from her repetitions.

Figs. 6.4c and 6.4b are illustrative of the user's disruption and show the degree of compliance to the budget strategy. In Fig. 6.4c the light grey curve illustrates the timeseries of frequency of annotations (TFA) – the frequency of asking for an annotation at a point in time during the annotation process. For all participants, we counted all annotation requests that happened at every point in time and then, for every timestamp, we averaged the result across all participants.

The TFA curve is very jagged, so we fitted a timeseries approximation model in order to identify the general trend of the timeseries. For this, we constructed an ARMA (Auto-Regressive Moving Average) model [161], which is an approximation of the original timeseries, but with emphasis on the general trend, rather than spontaneous deviations from the trend. The Auto-Regresive (AR) component seeks to fit a polynomial regression on any $p$ consecutive timeseries values so that the prediction error on the next value is minimised. The Moving-Average (MA) component, on the other hand, simply computes the average of every consecutive $q$ values. We used $p = 5$ and $q = 5$ because these values were the lowest which offered a clear trend, from an optical perspective[2]. Together, the AR and MA components outline the general trend (illustrated by the red line) of the TFA. The trend line (red) is contrasted to the ideal asking behaviour that matches the initial budget specification exactly (blue). It transpires that, while the TFA varies substantially, the general trend (red) follows the ideal distribution (blue) very closely. Therefore, the budget-constrained OAL improves the model accuracy over the random baseline and also closely matches the user's model of tolerance to disruption.

While the frequency of annotation varies considerably around the ideal value (which, in this scenario is $\nu = (200\ \textit{annotations})/(2000\ \textit{segments}) = 0.1$), the cumulative distribution of annotations, illustrated in Fig. 6.4b, is, in fact, much more well behaved. The actual distribution of annotations (black continuous line) matches the ideal/theoretical distribution of annotations (blue dashed line) extremely closely. This means that the method responds very well to budget restrictions in terms of cumulated annotations. The apparent contrast between Figs. 6.4c and 6.4b is explained as follows: If there is a deviation in spending from the ideal budget configuration, the method does not respond instantaneously and this is seen Fig. 6.4c where no individual timestamp is more likely to ask for an annotation than neighbouring ones. The relatively flat trend of the annotation frequency supports this view. This does not mean that the actual spending of the budget is erratic and does not conform to the ideal specification. In fact, Fig. 6.4b shows that when switching the perspective from the frequency of annotations to the cumulation of annotations in time, the actual spending comes very close

[2]Some of the models suffer from singularity issues, in which case we use $p = 5$ and $q = 6$, with very similar optical properties.




(a) Learning Curves

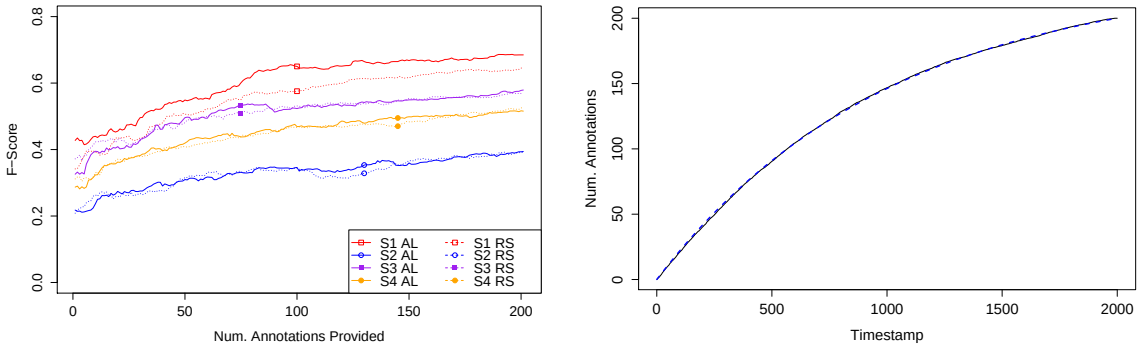
(b) Cumulative Distribution of Annotations

(c) Frequency of Annotations

Figure 6.4: Budget-Based OAL; Opportunity Dataset; Uniform Strategy; 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

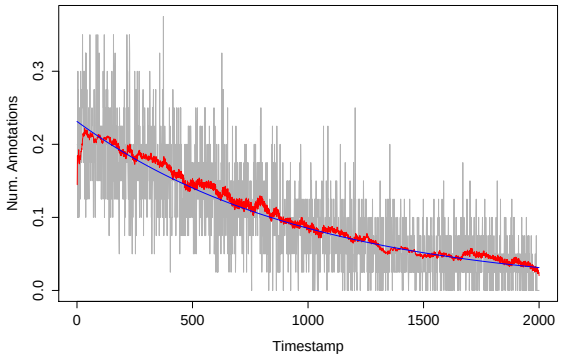
to the ideal cumulative distribution.

The results for the evaluation of the Exponential strategy are illustrated as follows: for $\lambda = 2$ in Fig. 6.5 and for $\lambda = 3$ in Fig. 6.6. As the Uniform strategy illustrated previously, the figures show not only a close approximation of the trend of the actual asking frequencies to the ideal asking probabilities, but also learning improvement over Random Selection, despite the budget-based constraints that are enforced by an Exponential strategy. The actual TFA (grey) in Figs 6.5c and 6.6c is still very jagged and varies substantially around its general trend, but the general trend (red) closely matches the ideal behaviour (blue), when averaged across all participants and repetitions. The high degree to which the ideal budget is approximated also transpires from Figs. 6.5b and 6.6b which illustrate how well the actual distribution of annotation matches the ideal.



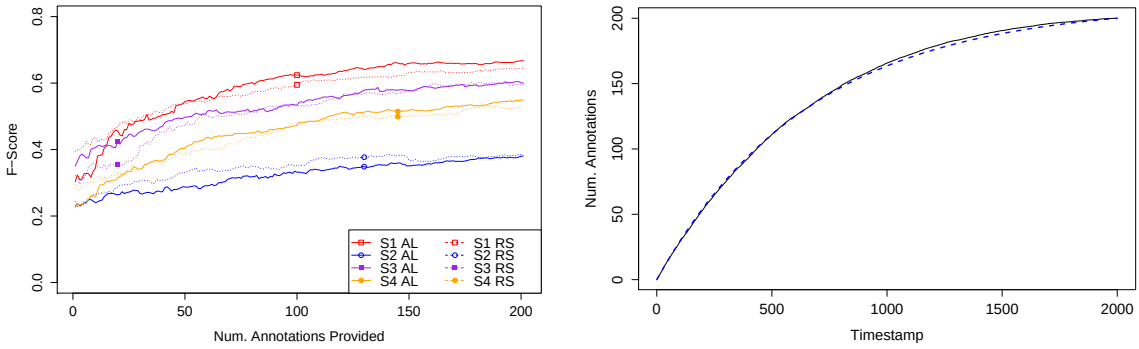

(a) Learning Curves

(b) Cumulative Distribution of Annotations




(c) Frequency of Annotations

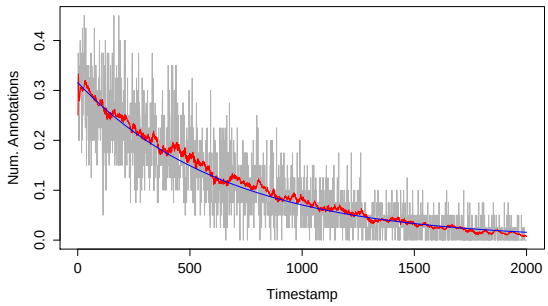
Figure 6.5: Budget-Based OAL; Opportunity Dataset; Exponential Strategy ($\lambda = 2$); 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves




(b) Cumulative Distribution of Annotations




(c) Frequency of Annotations

Figure 6.6: Budget-Based OAL; Opportunity Dataset; Exponential Strategy ($\lambda = 3$); 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

### *Results for Periodic Activities*

Given the limited size of the data and, in order to distribute a large enough number of annotations, we replay only 1-frame segments. We have shown in Chapter 4 that Online Active Learning works very well with longer segments and recognition performance still improves over Random Selection in these circumstances.

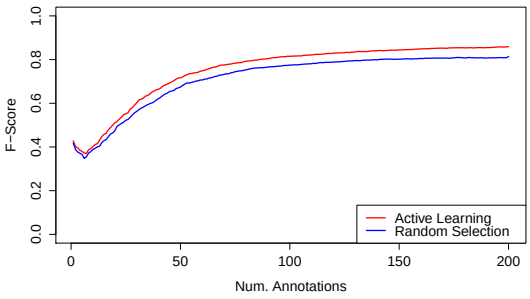
The interpretation of the results for periodic activities is very similar to that for non-periodic activities, discussed in Section 6.3.1. For the PAMAP dataset, Fig. 6.7 characterise the budget-constrained annotation process for the Uniform strategy, whereas Figs. 6.8 and 6.9 do so for the Exponential strategy. Overall, learning improvement is registered for a majority of points on the learning curves and the actual asking behaviours averaged across all dataset participants closely follow the ideal budget specifications.

The results for the USC-HAD dataset are not included in order to save space. Qualitatively, they are identical: (1) OAL still improves over RS, (2) the TFA is still jagged, but its trend comes close to the ideal and (3) the actual cumulative distribution of annotations closely matches the ideal one.

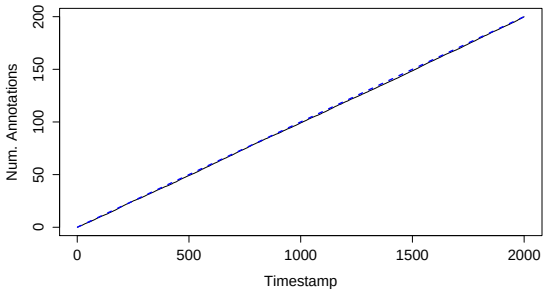
## Additional Constraint

The method outlined in Section 6.2 sets up a two step "set-attain" process of asking for annotations using an Online Active Learning approach while, at the same time, trying to adhere to an ideal spending budget strategy.

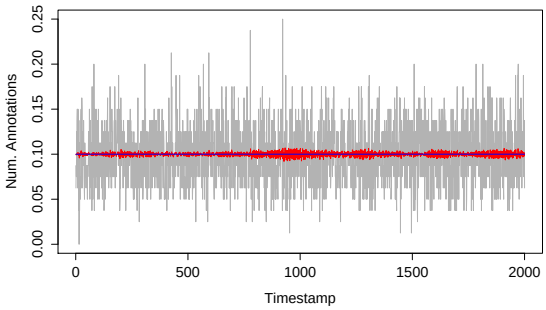
The method was evaluated in Section 6.3 where results show performance gains over Random Selection. Additionally, results show that the method adheres to the ideal spending strategy, especially when evaluating how well the cumulation of annotations approximates the ideal configuration. While previous compilations of results demonstrate that budget adherence is possible, however, the speed with which spending deviations are addressed is not controlled. This is seen in the plots illustrating the frequency of annotations (Figs. 6.4c, 6.5c, 6.6c, 6.7c, 6.8c, 6.9c) – these were discussed earlier in Section 6.3. We mentioned that the lack of urgency with which the sys-




(a) Learning Curves

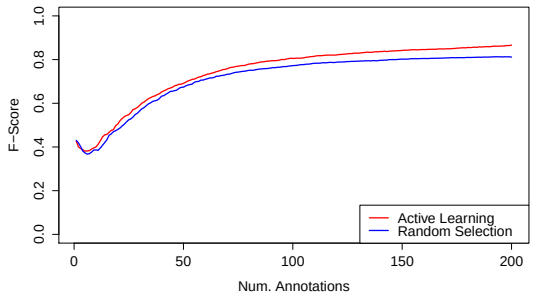



(b) Cumulative Distribution of Annotations

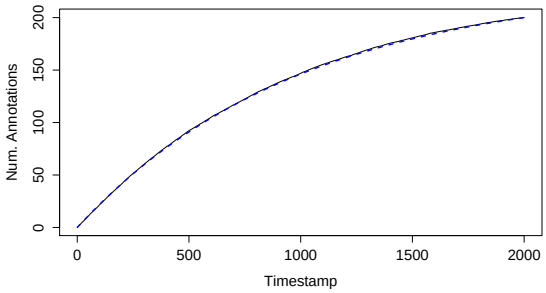



(c) Frequency of Annotations

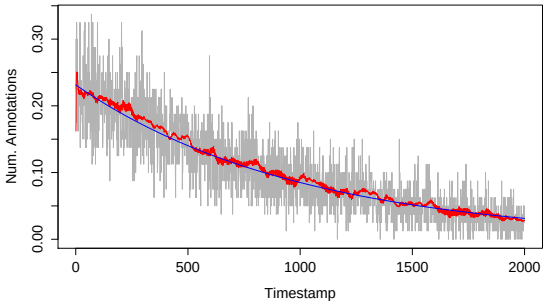
Figure 6.7: Budget-Based OAL; PAMAP Dataset; Uniform Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves

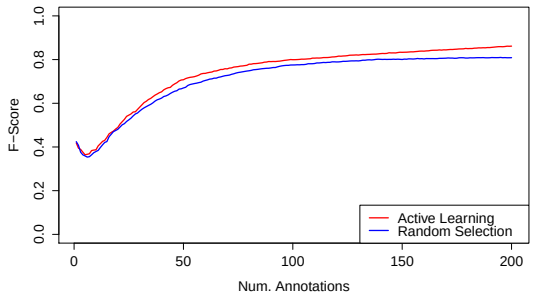



(b) Cumulative Distribution of Annotations

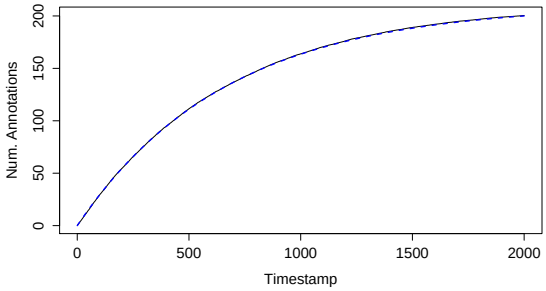



(c) Frequency of Annotations

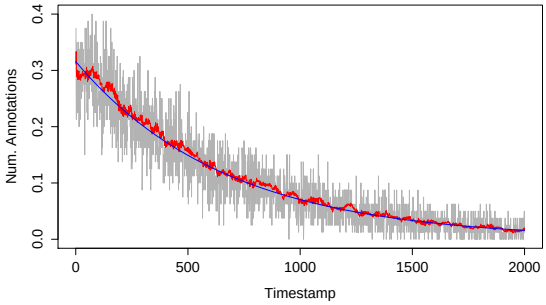
Figure 6.8: Budget-Based OAL; PAMAP Dataset; Exponential Strategy ($\lambda = 2$); 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves




(b) Cumulative Distribution of Annotations




(c) Frequency of Annotations

Figure 6.9: Budget-Based OAL; PAMAP Dataset; Exponential Strategy ($\lambda = 3$); 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

tem reacts to spending deviations is evident in the relatively smooth trend line of the frequency with which annotations are requested.

In this section we add a third step to the annotation method which comes after the "set-attain" steps. This third step tunes the balance between, on one hand, requesting annotations as a result of Online Active Learning identifying critical annotations and, on the other hand, requesting annotations as a result of adhering closely to the ideal spending strategy – i.e. effectively accelerating how quickly budget spending deviations are corrected. The step uses a penalty-based approach which alters the asking probability so that the importance of Online Active Learning decisions are attenuated, in a commensurate manner, by the degree of deviation of the actual spending relative to the ideal spending.

### *Step 3: Coercing the Budget*

At any point in time $t$, $B_{spent}$ budget units have been spent, while, according to the budget spending strategy $F$ which outlines the ideal spending behaviour, $B_{ideal} = B_{total} \cdot F(t)$ budget units should have been spent. We define the budget spending deviation $B_{diff} = B_{ideal} - B_{actual}$ as the signed difference between the ideal budget size that should have been spent at the current time $t$ and the actual size of the budget that has been spent by that point.

In this section, we modify the annotation behaviour so that, while informed Online Active Learning annotation requests are still carried out according to the method outlined in Section 6.2, budget spending deviations, when they arise, would be corrected more quickly. We still calculate $p_{ask}^{OAL}$, the asking probability due to Online Active Learning, using Eq. 4.1 with the $\gamma$ parameter set according to the method in Section 6.2.2. We introduce a new factor $p_{ask}^{budget} \in [0, 1)$, which is a function of the current deviation from the ideal budget. $p_{ask}^{budget}$ moderates $p_{ask}^{OAL}$ according to Eq. 6.4:

$$p_{ask} = \begin{cases} (1 - p_{ask}^{budget}) \cdot p_{ask}^{OAL} + p_{ask}^{budget} & \text{, if } B_{diff} \geq 0 \\ (1 - p_{ask}^{budget}) \cdot p_{ask}^{OAL} & \text{, otherwise} \end{cases} \tag{6.4}$$

Intuitively, the more the actual spending deviates from ideal spending, the less em-

phasis on $p_{ask}^{OAL}$ and the greater the emphasis on corrective action, such as requesting annotations (first branch of Eq. 6.4) or restraining from annotation requests (second branch of Eq. 6.4, depending on the direction of the spending deviation.

If there is under-spending ($B_{diff} \geq 0$), then the following three cases describe the asking behaviour:

- If $p_{ask}^{budget} \to 1$,then $p_{ask} \to 1$, which implies that relatively large under-spending deviations are addressed by requesting, with higher probability, uninformed (immediate) annotation requests so that under-spending is alleviated.
- If $p_{ask}^{budget} = 0$, then $p_{ask} = p_{ask}^{OAL}$, which means that the less severe the under-spending, the more Online Active Learning becomes unconstrained so that the emphasis is placed on obtaining high quality annotations.
- In general, when under-spending, we have as follows:

$$p_{ask} - p_{ask}^{OAL} = (1 - p_{ask}^{budget}) \cdot p_{ask}^{OAL} + p_{ask}^{budget} - p_{ask}^{OAL} = p_{ask}^{budget} \cdot (1 - p_{ask}^{OAL}) \geq 0$$

We conclude that $p_{ask} \geq p_{ask}^{OAL}$ holds true in the case of under-spending. This means that Online Active Learning annotation requests are complemented to a commensurate degree by more urgent but less well informed annotation requests so that the actual budget spending gets back in line with the ideal budget.

If there is over-spending ($B_{diff} \leq 0$), then:

- If $p_{ask}^{budget} \to 1$, then $p_{ask} \to 0$, meaning that in case of severe over-spending, Online Active Learning is effectively suppressed and no annotation requests will be probable until over-spending ameliorates with the passage of time.
- If $p_{ask}^{budget} = 0$, then $p_{ask} = p_{ask}^{OAL}$, which means that Online Active Learning is unconstrained.

- In general, when over-spending, we have as follows:

$$p_{ask} - p_{ask}^{OAL} = (1 - p_{ask}^{budget}) \cdot p_{ask}^{OAL} - \cdot p_{ask}^{OAL} = -p_{ask}^{OAL} \cdot p_{ask}^{budget} \leq 0$$

Therefore, in case of over-spending, $p_{ask} \leq p_{ask}^{OAL}$. This entails that the asking probabilities due to Online Active Learning are decreased so that annotations are discouraged until actual spending gets in line with ideal spending.

We have shown that $p_{ask}^{budget}$ increases the probability of asking for an annotation if there is under-spending ($B_{diff} < 0$) and decreases the probability of requesting an annotation if there is over-spending ($B_{diff} > 0$). Additionally, because we have seen previously that asking according to $p_{ask}^{OAL}$ leads to performance gains over Random Selection, we have also shown that $p_{ask}^{budget}$ does not change this behaviour if there is no deviation from ideal spending.

We define the $p_{ask}^{budget}$ as a probability which is a function of the size of the deviation in budget spending as follows:

$$p_{ask}^{budget}(B_{diff}) = 2 \cdot \left[ \frac{1}{1 + e^{-1/\beta \cdot |B_{diff}|}} - 0.5 \right] \tag{6.5}$$

Eq. 6.5 represents the upper half of a sigmoid function. $p_{ask}^{budget}$ increases with $B_{diff}$, so, effectively, the greater the deviations in budget spending, the greater the value of the $p_{ask}^{budget}$ factor and, consequently, the greater the restriction on Online Active Learning.

Since $p_{ask}^{budget}$ is a moderation factor for $p_{ask}^{OAL}$, we modelled it to be strictly increasing with the budget deviation, as can be seen in Fig. 6.10. The $\beta$ parameter controls the degree to which emphasis is shifted from Online Active Learning to immediate and uninformed annotation requests. Specifically, for a fixed value of the $B$ budget deviation parameter, the factor $p_{ask}^{budget}$ is strictly decreasing with $\beta$. This means that lower values for $\beta$ will make the transition from a factor value of 0 to 1 more sudden and, therefore, the asking behaviour would be more prone to correct small budget deviations immediately than to focus on obtaining highly critical annotations. Higher values for $\beta$ lead to a smoother transition, so the factor would not substantially change focus

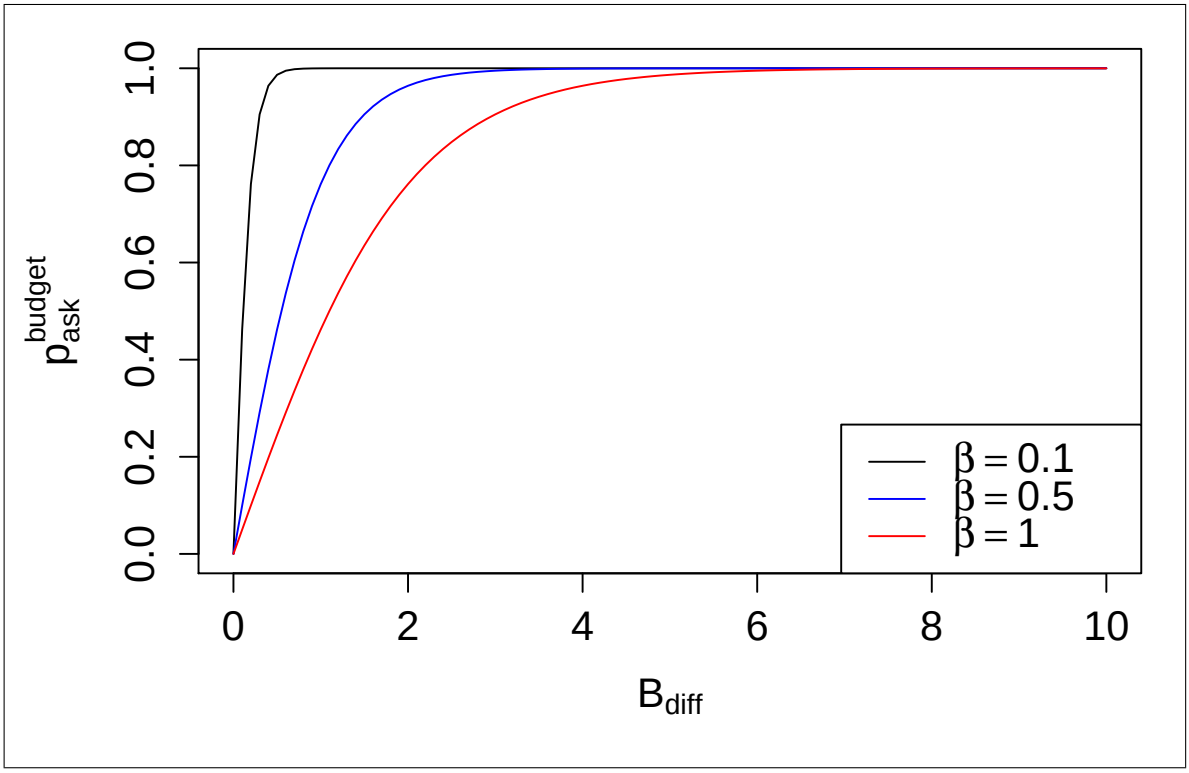



Figure 6.10: The effect of the $\beta$ parameter on $p_{ask}^{budget}$.

from requesting critical annotations to fixing budget deviations unless the deviations become substantially large.

## Results

We now present the evaluation results of the annotation method described in Section 6.4 which uses all three budget-based constraint steps described in Sections 6.2 and 6.4. As in Section 6.3, we again evaluate the effects the annotations had on the performance of the models and the level of compliance of the schedule of annotation requests to the ideal schedule of the budget spending strategy.

As before, we there are two evaluation outcomes. Firstly, we contrast the learning performances of Online Active Learning and Random Selection. Secondly, we present the evaluation results for the degree of compliance to the ideal budget spending strategy.

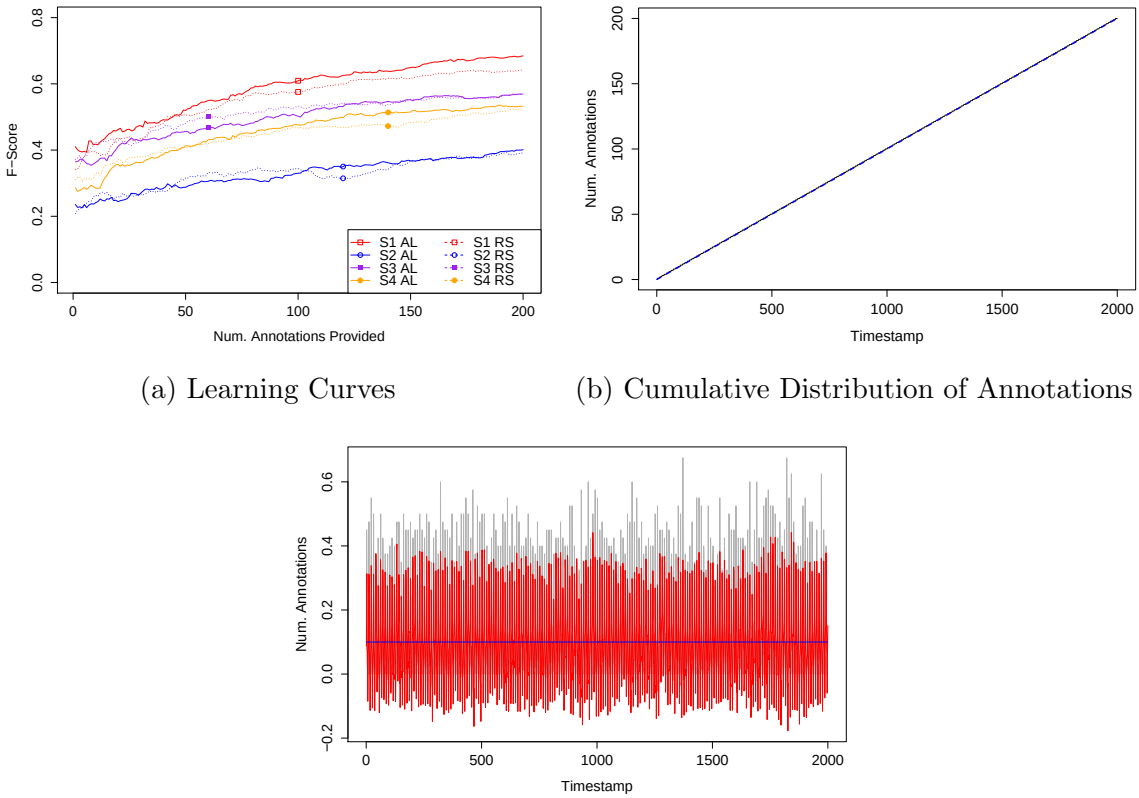
Since the $\beta$ parameter controls the overall behaviour of the present annotation method, in what follows we focus on the link between $\beta$ and the resulting annotations. We investigate the effects of three values of the $\beta$ parameter: *strict* coercion ($\beta = 0.1$)

in the OAL-driven annotation process, *moderate* coercion ($\beta = 0.5$) and, finally, *mild* coercion ($\beta = 1$). We do not consider larger values for the $\beta$ parameter because these only result in insubstantial influence (OAL is practically unconstrained and behaves as in Section 6.3).

### *Results for Non-Periodic Activities*

For the *strict* coercion scenario ($\beta = 0.1$) in Fig. 6.11, the most noticeable result is that the trend curve of the ARMA model (red line), which, previously, clearly illustrated a smooth trend line, is now very jagged. This suggests that the variance in the timeseries is no longer caused by random noise, but it is symptomatic of an underlying pattern. Figs. 6.12a and 6.12b, where we have "zoomed in" and focused on the first 50 timestamps, reveals the pattern: the very strict budget enforcement configuration coerces many annotations to be requested around a fixed schedule, in an almost deterministic manner. In this scenario, given that 200 annotations are to be uniformly requested out of 2000 timestamps, on average 1 out of every 10 timestamps should yield an annotation. The annotations are not spread out evenly, but, instead, with high frequency, annotations are clustered around every 10 timestamps. This is due to the value of the $\beta$ parameter which transforms even slight under-spending into immediate annotation request decisions.

The phenomenon, which we call *fractional under-spending* is illustrated in Fig. 6.13. The ideal spending curve is that of a continuous-valued function, in order to bring it in line with the mathematical construction at the beginning in Section 6.2. However, the actual spending curve is necessarily discrete-valued (because it is a count) and so is the best actual spending – the discrete curve that most closely matches the ideal spending curve. Specifically, an annotation is ideally requested when $B_{diff} = 0.5$, which, in our case, would happen at timestamps $5, 15, 25, ...$ However, at $t = 1$ we have $B_{diff} = 0.1$, but, because of the low value $\beta = 0.1$, this results in $p_{ask}^{budget} = 0.46$ which strongly biases $p_{ask}$ towards issuing an annotation request. If an annotation is not requested at this point, then this phenomenon is further compounded at $t = 2$, when $B_{diff} = 0.2$, which gives $p_{ask}^{budget} = 0.76$, or at $t = 3$, when $B_{diff} = 0.3$, so $p_{ask}^{budget} = 0.90$, etc. This explains how even small fractional spending deviations can trigger almost immediate




(a) Learning Curves

(b) Cumulative Distribution of Annotations

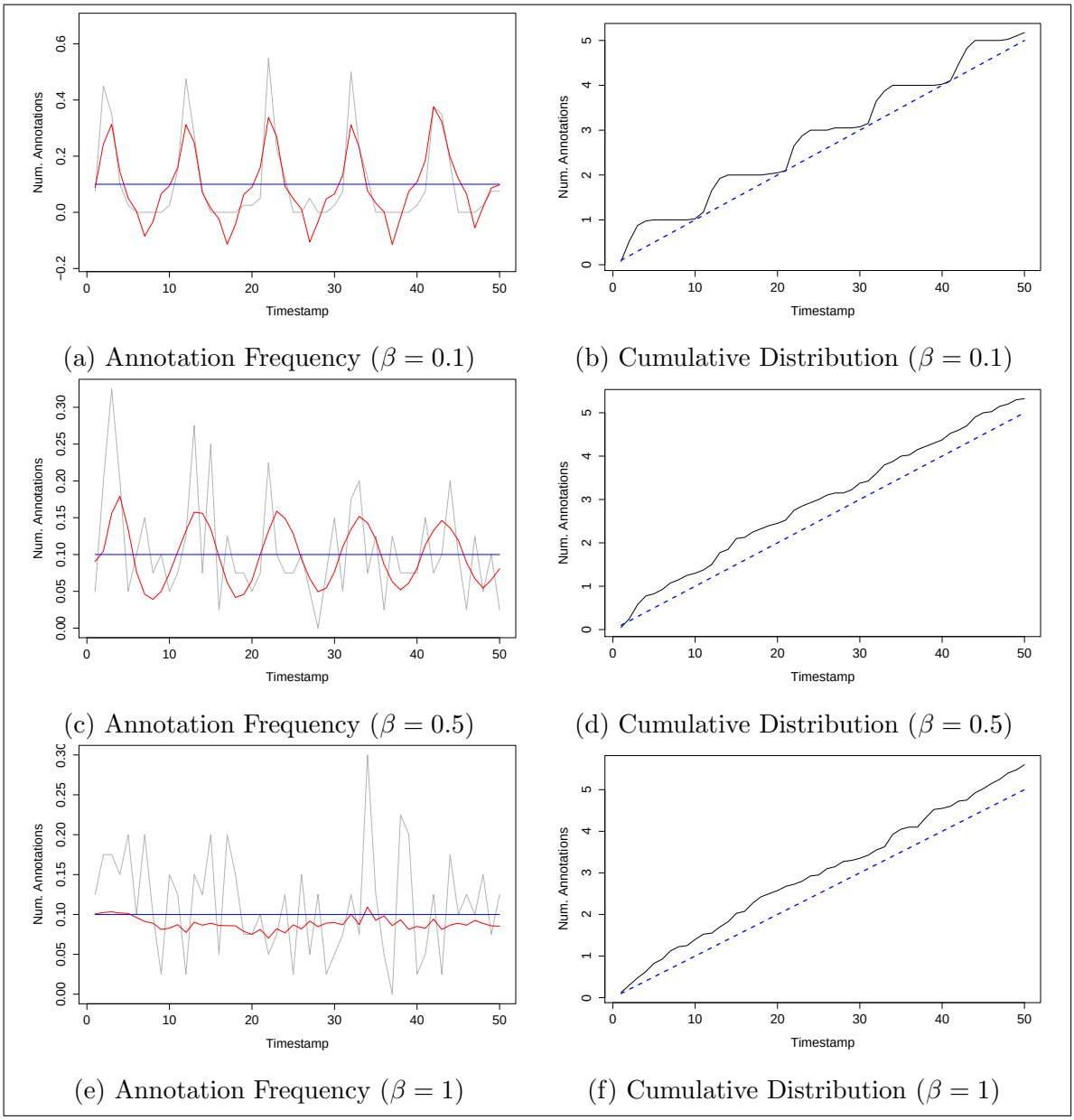
(c) Frequency of Annotations

Figure 6.11: Budget-Based OAL with Additional Constraint ($\beta = 0.1$); Opportunity Dataset; Uniform Strategy; 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

annotation requests if the $\beta$ parameter value is highly coercive. Consequently, with high coercion, one obtains greater control over the frequency of annotation requests by dictating how fast budget spending deviations should be corrected.

From a model performance point of view, however, for the *strict* scenario, as it transpires from Fig. 6.11, Online Active Learning has a severely reduced autonomy. The performance gains of OAL over RS are very low when compared with the results in Section 6.3.1 where OAL was less constrained.

The *moderate* interference scenario ($\beta = 0.5$) is illustrated in Fig. 6.14. The behaviour is again symptomatic of budget-related interference in the annotation process. The ARMA model again yields a periodic trend, but it is less pronounced in amplitude. This time the annotation requests are more evenly spread out (relative to the previous *strict* scenario), as can be seen in Fig. 6.12c. Also, compared to the *strict* scenario,




(a) Annotation Frequency ($\beta = 0.1$)

(b) Cumulative Distribution ($\beta = 0.1$)

(c) Annotation Frequency ($\beta = 0.5$)

(d) Cumulative Distribution ($\beta = 0.5$)

(e) Annotation Frequency ($\beta = 1$)

(f) Cumulative Distribution ($\beta = 1$)

Figure 6.12: Uniform Strategy; Opportunity Dataset; Distribution of Annotations (Zoom-In).

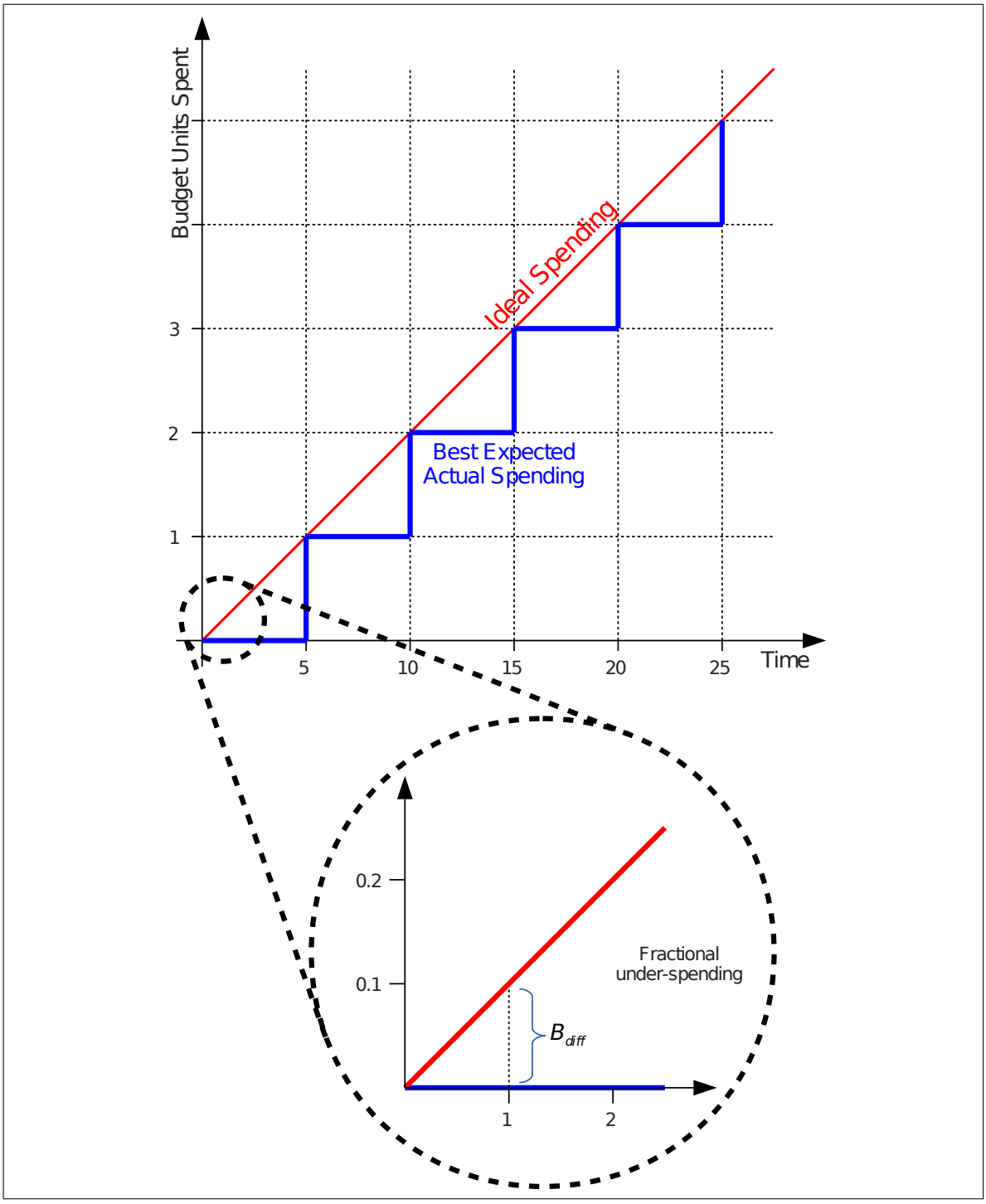
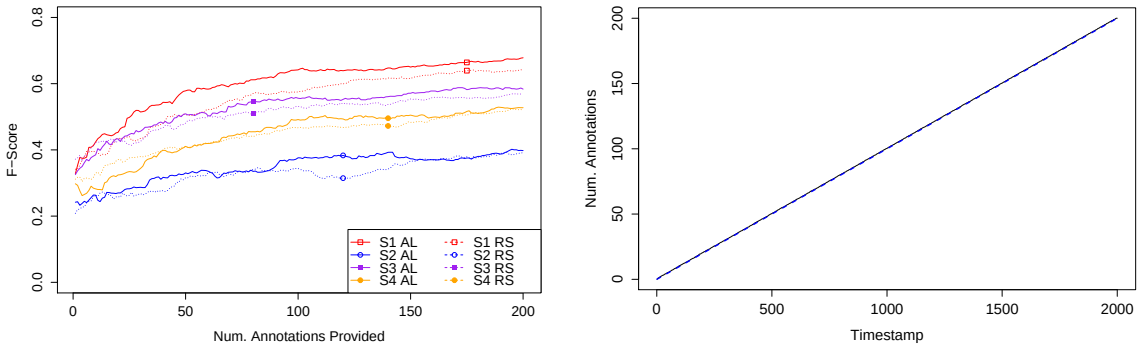


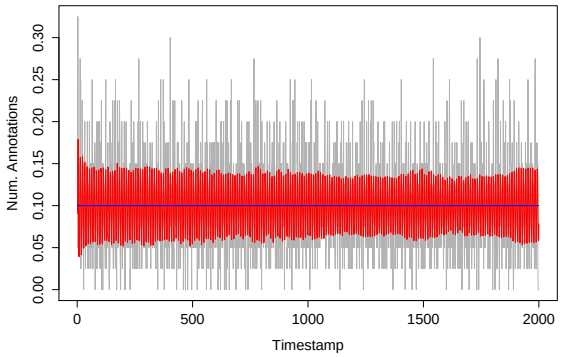

Figure 6.13: The Fractional Under-spending Phenomenon




(a) Learning Curves

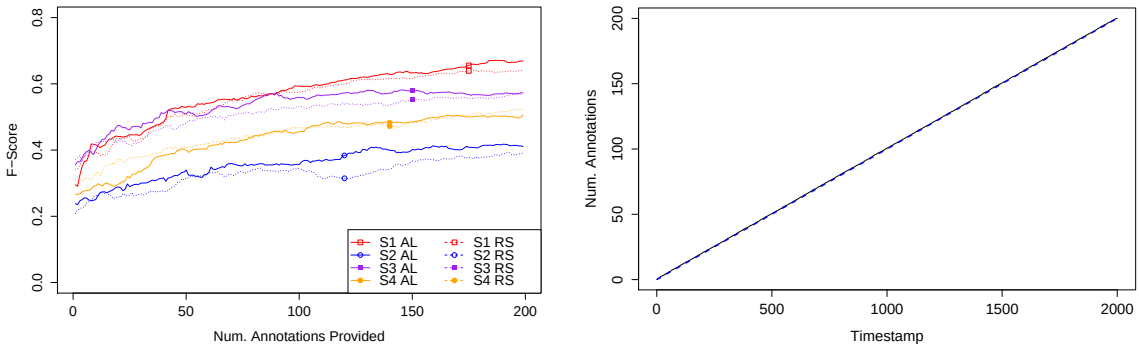



(b) Cumulative Distribution of Annotations



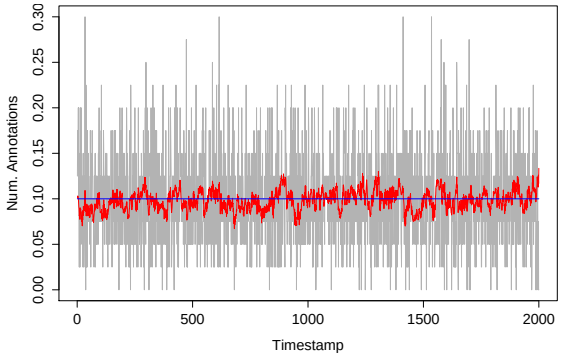

(c) Frequency of Annotations

Figure 6.14: Budget-Based OAL with Additional Constraint ($\beta = 0.5$); Opportunity Dataset; Uniform Strategy; 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves

(b) Cumulative Distribution of Annotations



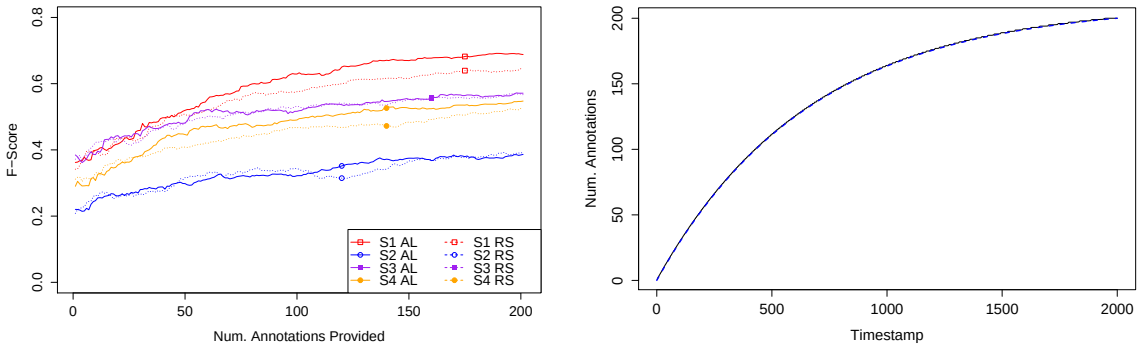

(c) Frequency of Annotations

Figure 6.15: Budget-Based OAL with Additional Constraint ($\beta = 1$); Opportunity Dataset; Uniform Strategy; 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

and because of the smaller constraints due to budget spending, Online Active Learning registers a greater improvement over Random Selection.

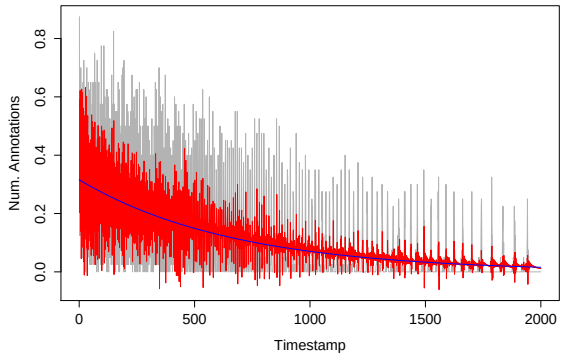
The *mild* interference scenario ($\beta = 1$) is illustrated in Fig. 6.15. Because of the relatively weak effect of the $p_{ask}^{budget}$ factor on $p_{ask}$, the annotation process is largely unaltered by budget constraints. The ARMA model does not fluctuate as much, as can be seen in Fig. 6.12e, indicating that the annotation requests are relatively uniformly distributed in time. Because Online Active Learning is largely unaffected by the $p_{ask}^{budget}$, it continues to yield performance gains over Random Selection comparable to those in Section 6.3.1 when Online Active Learning was largely unconstrained.

Interfering with the Exponential budget spending strategy ($\lambda = 3$) reveals similar effects as with the Uniform strategy examined previously. The *strict* interference scenario ($\beta = 0.1$) is illustrated in Fig. 6.16. The effect of extremely strict adherence



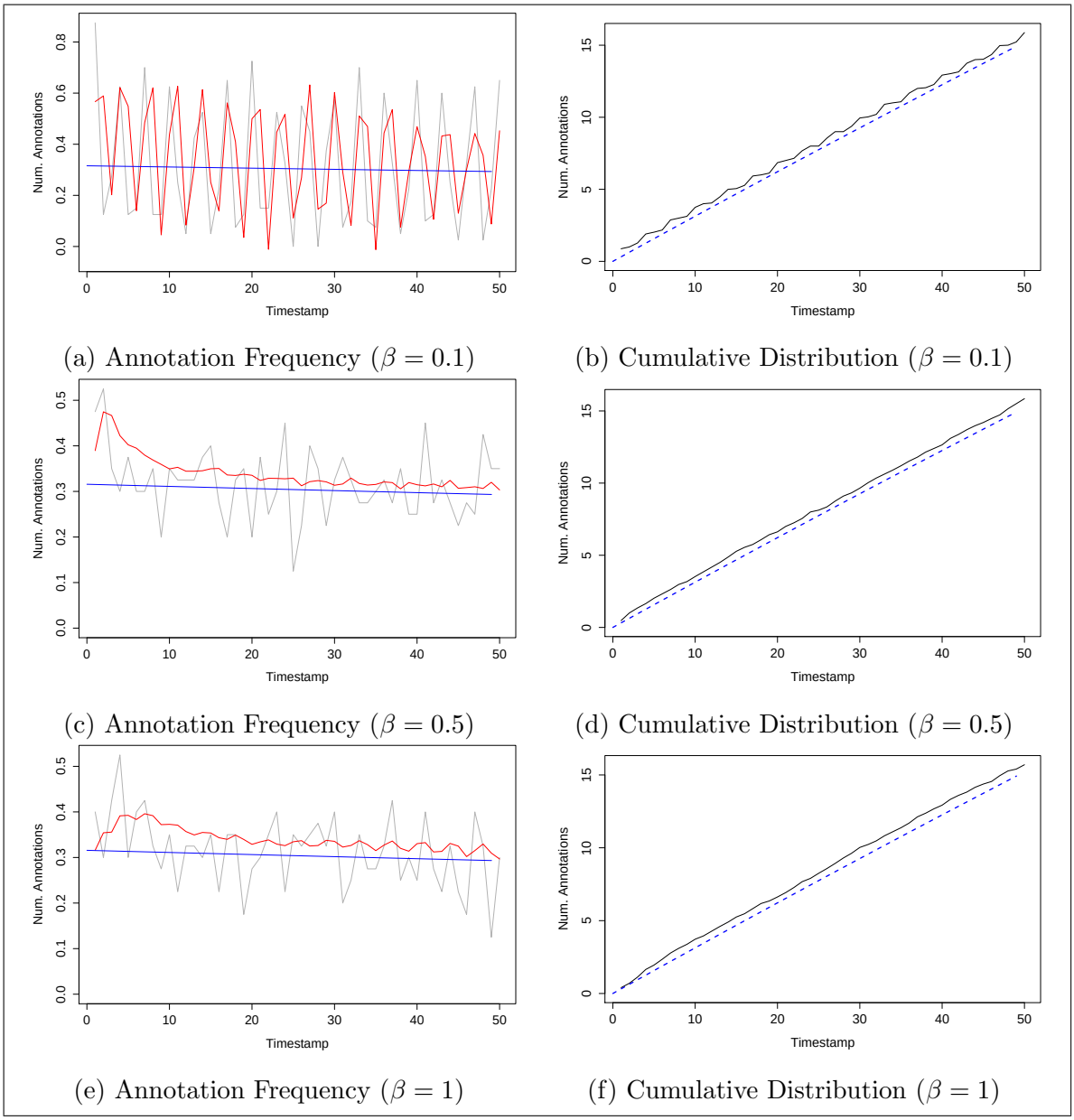

(a) Learning Curves

(b) Cumulative Distribution of Annotations




(c) Frequency of Annotations

Figure 6.16: Budget-Based OAL with Additional Constraint ($\beta = 0.1$); Opportunity Dataset; Exponential Strategy; 200 Budget Units; (S1 – Subject 1; AL – Online Active Learning; RS – Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Annotation Frequency ($\beta = 0.1$)

(b) Cumulative Distribution ($\beta = 0.1$)

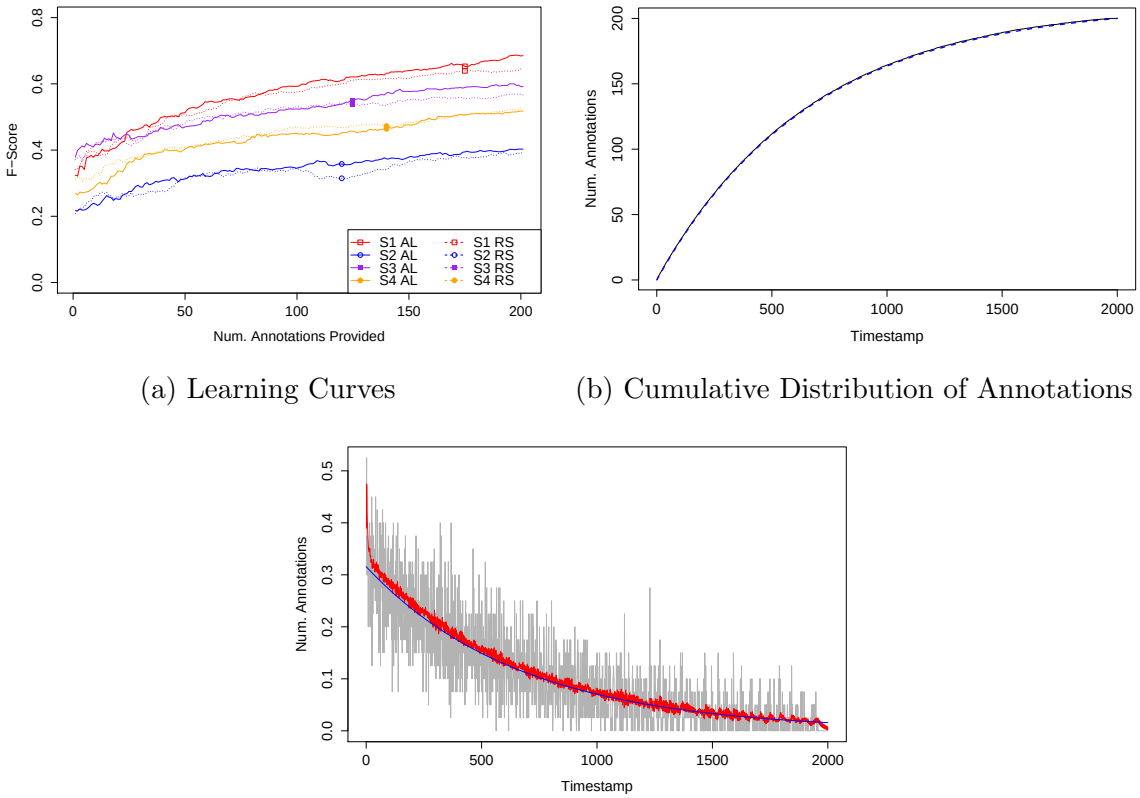
(c) Annotation Frequency ($\beta = 0.5$)

(d) Cumulative Distribution ($\beta = 0.5$)

(e) Annotation Frequency ($\beta = 1$)

(f) Cumulative Distribution ($\beta = 1$)

Figure 6.17: Exponential Strategy; Opportunity Dataset; Distribution of Annotations (Zoom-In).



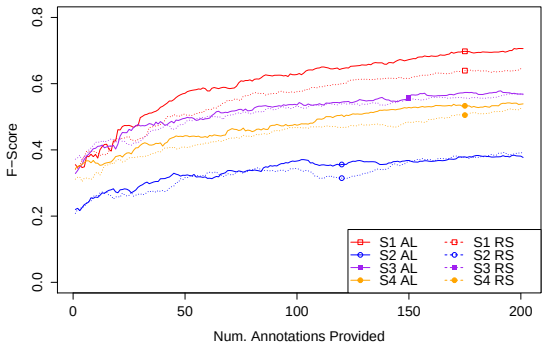

(a) Learning Curves

(b) Cumulative Distribution of Annotations

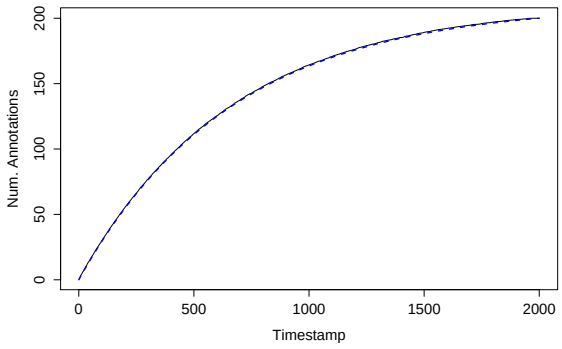
(c) Frequency of Annotations

Figure 6.18: Budget-Based OAL with Additional Constraint ($\beta = 0.5$); Opportunity Dataset; Exponential Strategy; 200 Budget Units; (S1 - Subject 1; AL - Online Active Learning; RS - Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

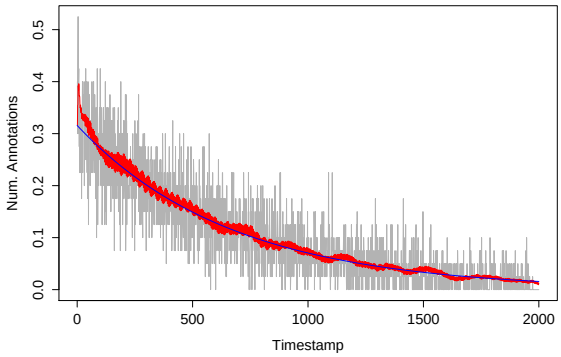
to the annotation schedule is again made evident by a very jagged ARMA trend line (red). "Zooming in" as before, Figs. 6.17a and 6.17b illustrate more how most annotation requests are concentrated around the ideal annotation timestamps. In terms of performance gains, as in the Uniform *strict* scenario, Online Active Learning is heavily constrained and, so, the performance gains over Random Selection are reduced.




(a) Learning Curves

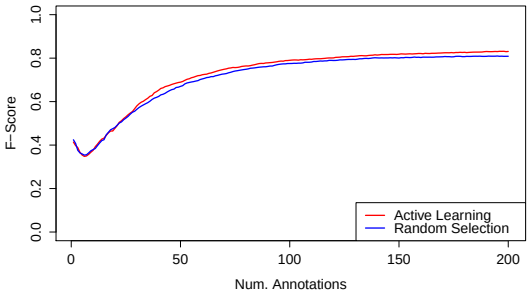



(b) Cumulative Distribution of Annotations

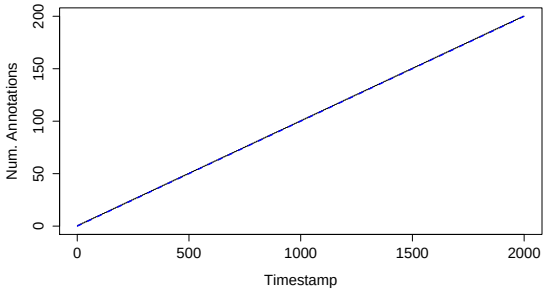



(c) Frequency of Annotations

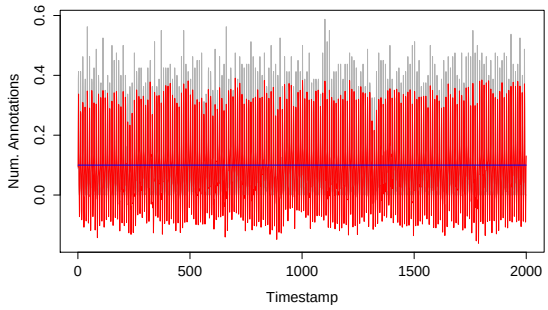
Figure 6.19: Budget-Based OAL with Additional Constraint ($\beta = 1$); Opportunity Dataset; Exponential Strategy; 200 Budget Units; (S1 - Subject 1; AL - Online Active Learning; RS - Random Selection); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves




(b) Cumulative Distribution of Annotations




(c) Frequency of Annotations

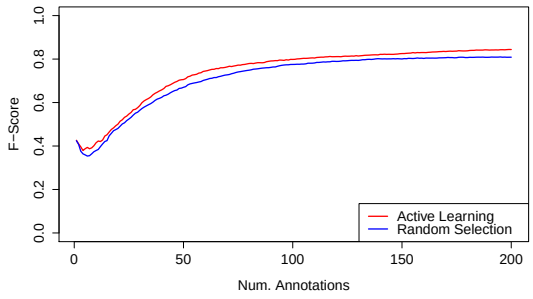
Figure 6.20: Budget-Based OAL with Additional Constraint ($\beta = 0.1$); PAMAP Dataset; Uniform Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

The *moderate* ($\beta = 0.5$; Fig. 6.18) and *mild* ($\beta = 1$; Fig. 6.19) scenarios show that reduced interference smooths the trend line, but does not improve the dispersion of the average number of annotations (grey line), as was desired. With reduced interference from the budget enforcement mechanism, Online Active Learning registers clear improvement over Random Selection.

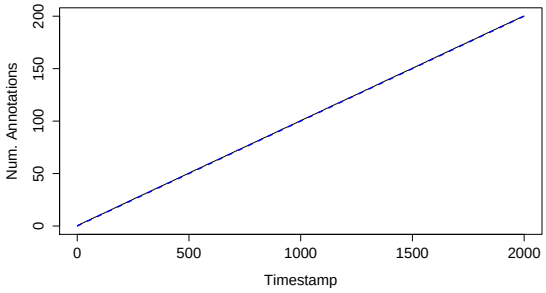
### *Results for Periodic Activities*

For periodic activities, we have evaluated our method, as before, on the PAMAP dataset. The results are essentially similar to the ones for non-periodic activities.

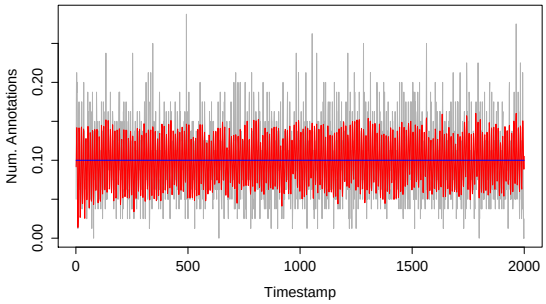
The evaluations for the Uniform strategy are illustrated in Fig. 6.20 for the *strict* scenario, in Fig. 6.21 for the *moderate* scenario and in Fig. 6.22 for the *mild* scenario. Recognition performance again suffers if the value of the $\beta$ parameter is excessively




(a) Learning Curves

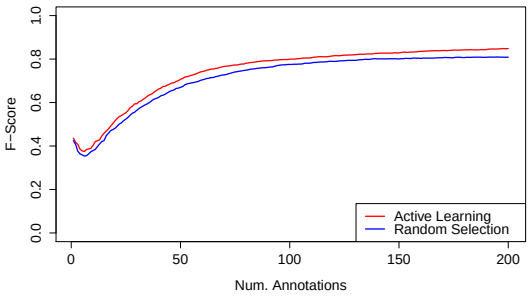



(b) Cumulative Distribution of Annotations

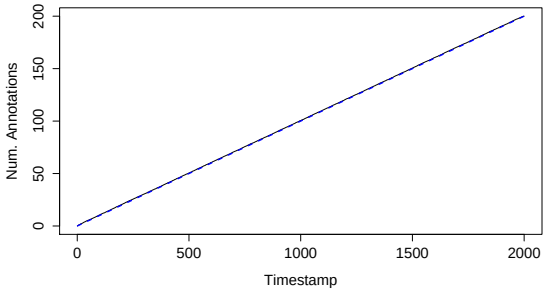



(c) Frequency of Annotations

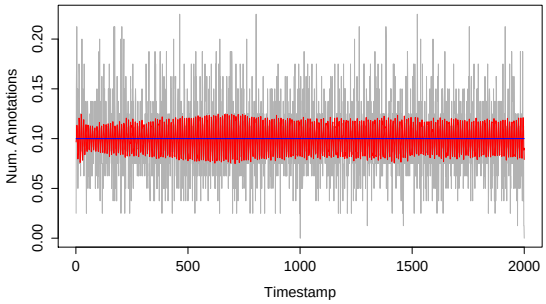
Figure 6.21: Budget-Based OAL with Additional Constraint ($\beta = 0.5$); PAMAP Dataset; Uniform Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves

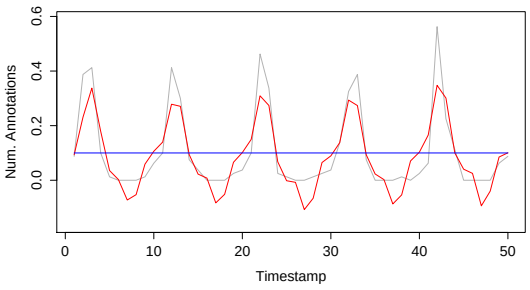



(b) Cumulative Distribution of Annotations

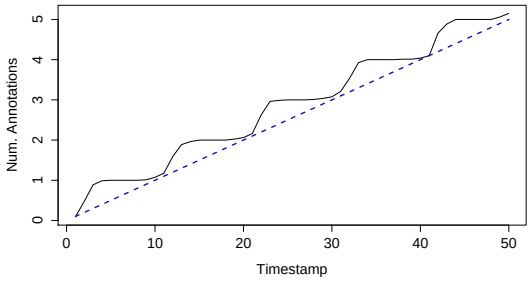



(c) Frequency of Annotations

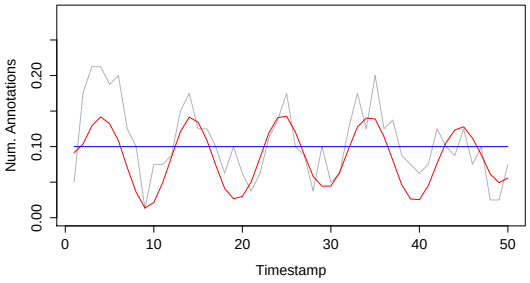
Figure 6.22: Budget-Based OAL with Additional Constraint ($\beta = 1$); PAMAP Dataset; Uniform Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).



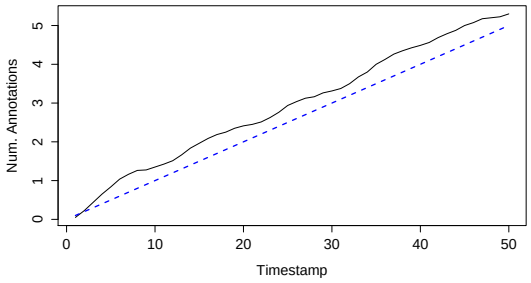

(a) Annotation Frequency ($\beta = 0.1$)



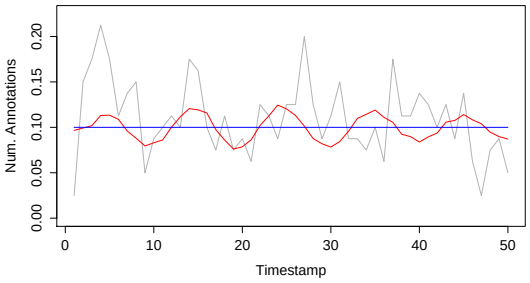

(b) Cumulative Distribution ($\beta = 0.1$)



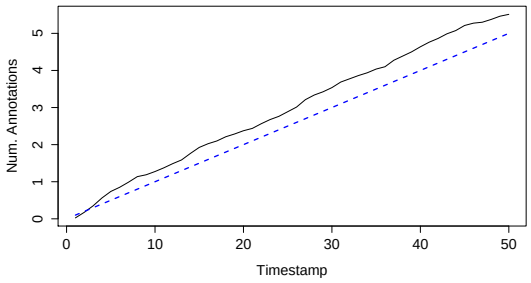

(c) Annotation Frequency ($\beta = 0.5$)




(d) Cumulative Distribution ($\beta = 0.1$)

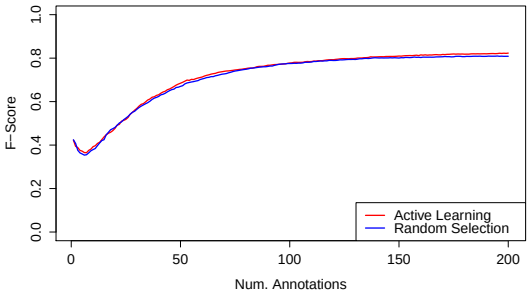



(e) Annotation Frequency ($\beta = 1$)

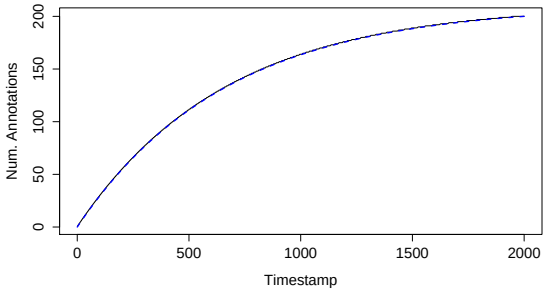



(f) Cumulative Distribution ($\beta = 1$)

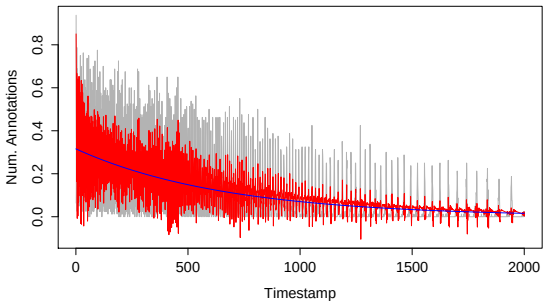
Figure 6.23: Uniform Strategy; PAMAP Dataset; Distribution of Annotations (Zoom-In).




(a) Learning Curves




(b) Cumulative Distribution of Annotations



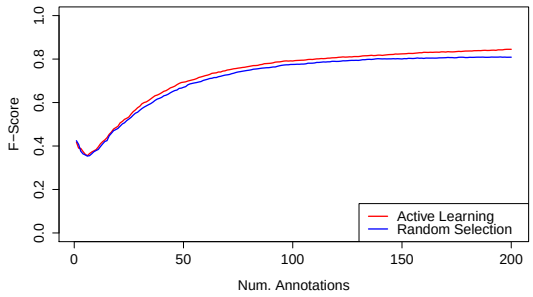

(c) Frequency of Annotations

Figure 6.24: Budget-Based OAL with Additional Constraint ($\beta = 0.1$); PAMAP Dataset; Exponential Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).

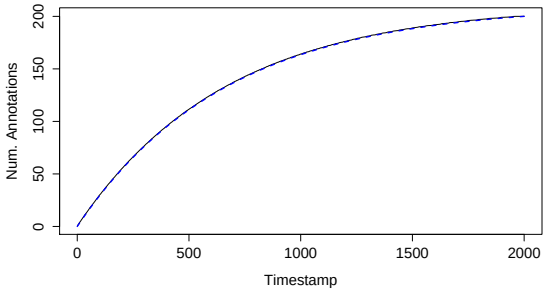
coercive. The *strict* scenario restricts OAL excessively, so the performance gains are minimised. The other two scenarios free up OAL to operate, and the performance gains over RS increase.

The zoom-in on the first 50 timestamps in Fig. 6.23 illustrates the degree of adherence to the ideal budget spending strategy as a function of the $\beta$ parameter.

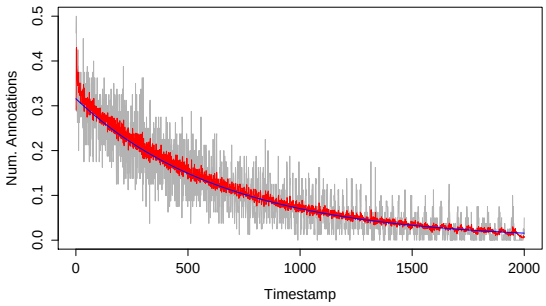
For the Exponential strategy, Fig. 6.24 illustrates the *strict* scenario, Fig. 6.25 illustrates the results for the *moderate* scenario and, finally, Fig. 6.26 illustrates the *mild*. The zoom-in on the first 50 timestamps in Fig. 6.27 illustrates the degree of adherence to the ideal budget spending strategy as a function of the $\beta$ parameter.




(a) Learning Curves

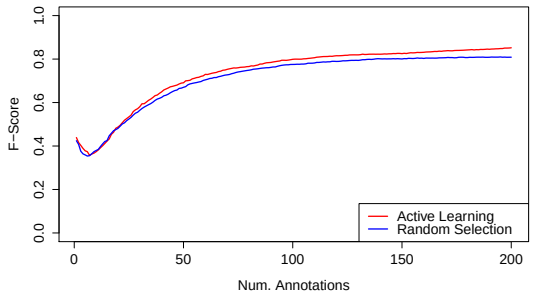



(b) Cumulative Distribution of Annotations

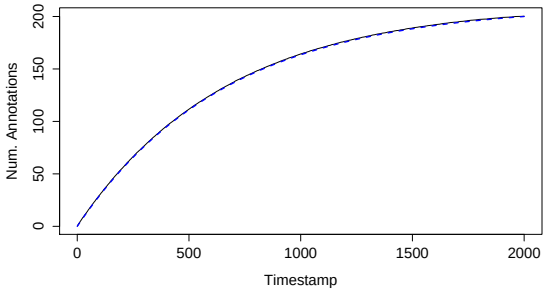



(c) Frequency of Annotations

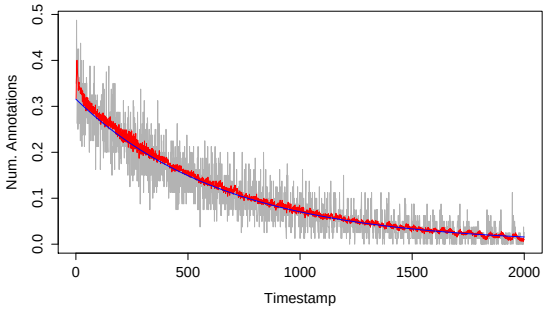
Figure 6.25: Budget-Based OAL with Additional Constraint ($\beta = 0.5$); PAMAP Dataset; Exponential Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).




(a) Learning Curves

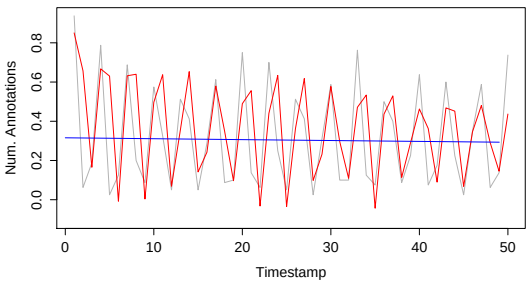



(b) Cumulative Distribution of Annotations

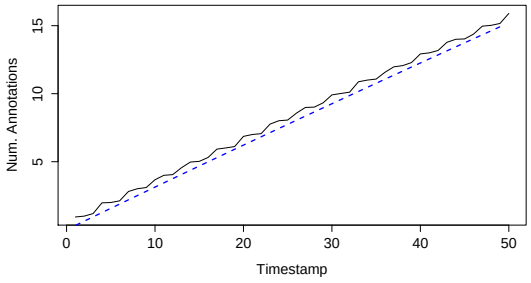



(c) Frequency of Annotations

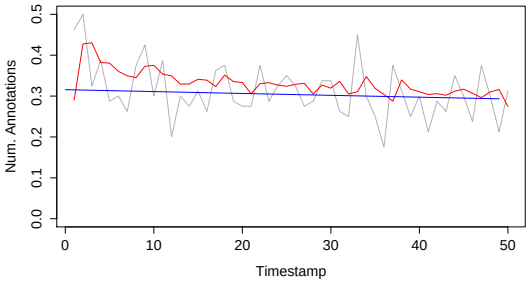
Figure 6.26: Budget-Based OAL with Additional Constraint ($\beta = 1$); PAMAP Dataset; Exponential Strategy; 200 Budget Units; Frequency: Theoretical (blue), Actual (grey) and ARMA (red); Cumulative Distribution: Theoretical (blue dotted) and Actual (black solid).



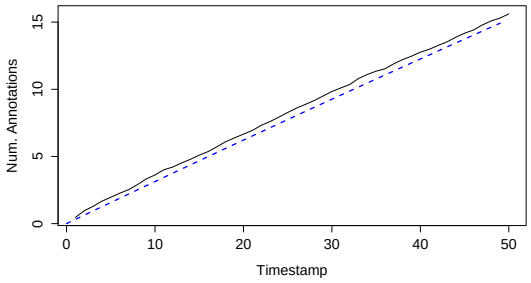

(a) Annotation Frequency ($\beta = 0.1$)



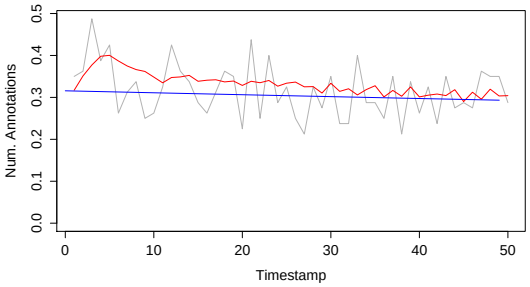

(b) Cumulative Distribution ($\beta = 0.1$)



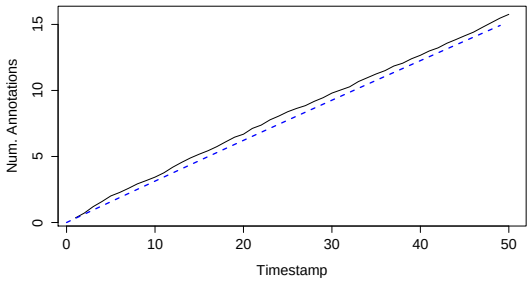

(c) Annotation Frequency ($\beta = 0.5$)




(d) Cumulative Distribution ($\beta = 0.5$)




(e) Annotation Frequency ($\beta = 1$)




(f) Cumulative Distribution ($\beta = 1$)

Figure 6.27: Exponential Strategy; PAMAP Dataset; Distribution of Annotations (Zoom-In).

### *Discussion*

We extended the budget-constrained Online Active Learning method by adding a third step which compensates budget spending deviations with either immediate annotation requests or dissuading from raising annotation requests. Our penalty-based approach ensures that budget spending deviations are minimised while Online Active Learning is in place so that high quality annotations would be revealed. The method was aimed as a further refinement of Steps 1 and 2 from Sections 6.2.2 and 6.2.3, respectively. However, we underline that this final third step is optional: it may or may not be added after the first two steps. The integrity of our proposed annotation method is not affected by this step, but the end result is altered by adding it.

Our theoretical grounding from Section 6.4.1 promised that using just the first two steps from Section 6.2, budget deviations could be partially controlled. In practice, results showed that the cumulative distribution of annotation matches the theoretical desired distribution very closely. However, the speed with which budget spending deviations are corrected is not under control and this is observable by the relatively constant annotation frequencies.

With the addition of the third step, discussed in Section 6.4, the frequency of annotation requests can be controlled so that annotations are clustered around certain timestamps. With this added layer of control over the annotation process, however, the performance gains from Online Active Learning over Random Selection were greatly diminished. Overall, there is a trade-off between, on the one hand, the degree to which the budget is enforced and, on the other hand, the performance gains of OAL over RS. Strong budget enforcement leads to very small OAL improvement over RS, whereas, weaker budget enforcement frees OAL and encourages gains over RS. Within the application of our proposed budget coercion method, very strict budget spending can be enforced for low values of the $\beta$ parameter (even asymptotically deterministic for $\beta \to 0$) while very relaxed configurations, where Online Active Learning is dominant, are possible for high values of the $\beta$ parameter ($\beta \to \infty$).

## Conclusions

In this chapter we have furthered the theoretical concepts presented in Chapters 3 and 4 by proposing an aggregate method that combined the two annotation heuristics, one that operated strictly on budget considerations and another one that focused entirely on optimising the performance gains, into a single method that balanced both approaches.

We have shown how Online Active Learning can be used in conjunction with a budget spending strategy and we provided a simulation-based evaluation. Our results generally reveal three-fold implications. Firstly, the time distributions of annotations closely mimic the theoretical distribution functions of ideal budget specifications. This demonstrates that budget spending strategies can be enforced with online annotation decisions without violating given annotation schedules. Secondly, we have also show that, by making annotation decisions using Online Active Learning, even though the decisions are also influenced by budget strategy conditions, we still register performance gains over Random Selection. Finally, our results show that the degree of adherence to an ideal budget can be controlled by intuitive parameter tuning. However, the strictness of adherence and the extent of performance gains of OAL over RS are at odds: greater strictness results in diminished gains and vice-versa.

In Chapter 5, we applied our Online Active Learning method, but without budget constraints, because this would have further complicated the evaluation. The question, therefore, is whether the budget-constrained version of Online Active Learning could be applied in a real deployment where user sentiment towards interruptions for annotation purposes mattered.

**[[First of all, in this chapter, we made an additional assumption that there is an initial corpus of annotations.]]** This ensures that the confidence levels of the activity model do not fluctuate wildly and are representative of a degree to the underlying domain of personalised annotations. These more stable confidence levels lead to more robust estimations for the purpose of heuristic fine-tuning, which, in turn, is used to meet target budgets. The alternative to a pre-existing corpus of annotation implies starting from no annotations and employing a NAD (discussed previously in Section

5.3.1). However, using a NAD or an alternative mechanism adds to the complexity of reliably estimating the parameters required to meet a certain target budget.

Second of all, in a real deployment, the notion of budget strategy needs a definition which is applicable to that practical context. In this chapter and in Chapter 3 we assumed that time is the index of a segment in a stream of activities. In reality, users are more likely to specify budget strategies in terms of the actual time of day because, arguably, the time of day may be more meaningful to them than segment indices. Given that activities can be of arbitrary duration, meeting these physical time constraints is likely to be practical only with a modified version of Eq. 6.2 that accounts for physical time instead of indices for a stream of segments. Best-effort guesses for activity duration would probably have to be employed to estimate the target budget, but these would introduce additional uncertainty when trying to approximate an ideal budget strategy.

Finally, coercing annotations according to a budget specification does not affect the ability of the system to run in online mode. Calculating the target budget and optimising the heuristic function for best meeting the target budget require constant time and memory complexity, so both are suitable for online calculation (relative to the heuristic in Chapter 4, the additional computations amount only to an additive constant). In Chapter 5, we demonstrated that a budget-free system can run in an online regime. Therefore, so can a budget-constrained Online Active Learning-based system function in online mode.

# 7

# Conclusions

## Contents

In this chapter we summarise the contributions from the thesis and we reflect upon their implications on the current research landscape and potential for future work.

## Motivation and Context

In this thesis we acknowledged the prevalence of wearables and their importance for encouraging healthy routines for individuals. As we argued in Chapters 1 and 2, HAR model personalisation typically improves monitoring accuracy, which, in turn, leads to making better informed decisions, such as inhibiting certain damaging behaviours or integrating beneficial behaviours. Model personalisation, as defined in Section 1.3, is the core problem we addressed in this thesis.

Researchers have recognised the improvements brought by HAR model personalisation and have proposed diverse ways of achieving personalisation. This thesis addressed the personalisation problem by proposing a user-centered solution. Specifically, we involved the user in the personalisation process and we assumed the user was cognisient enough of her own activities that she would be able to provide on-demand personalising feedback.

The feedback was structured in the form of *annotations*, which are descriptions of sensor data that could be used to improve the user's HAR model. The system identified contiguous portions in the sensor timeseries, called *segments*, each of which would ideally correspond to a single activity. The segmentation procedure operated in *online* mode, meaning that it was designed to recognise segment boundaries shortly after an activity ended. The system would then inspect the segment data and would decide whether or not to ask the user to provide an annotation for the segment. If such an annotation was provided by the user, then it would be used to update the user's personal HAR model.

When the user provided her annotations, we assumed no external help, such as other human annotators or a complex infrastructure with video footage collection that would assist the user in providing annotations. These external aids would arguably be perceived as obtrusive and would reduce the degree of realism of our study. Instead, we opted for a highly naturalistic environment where the user would be assisted just by a

lightweight mobile app. In our opinion, this setup introduced virtually no constraints when compared to an analogous naturalistic context.

Because of the lightweight infrastructure which did not support video footage collection, we relied on the user's short-term memory as a source of ground truth annotation. As discussed earlier, we employed an online segmentation that operated over the user's stream of activities. Since the discovered segments correspond to the most recently finished activity, we assumed the users would be able to provide online annotations from their short-term memory.

Finally, we recognised that our mechanism of obtaining annotations can be intrusive as it relies on interrupting the user to provide input on-demand. To alleviate this, we accounted for models of user tolerance levels. We factored in mechanisms that restricted the volume and distribution of annotation requests so that user tolerance boundaries are not crossed.

## Results and Significance

In this section we discuss how our contributions relate to each other. As was shown in Fig. 1.5, the contributions are presented in an evolutionary fashion where new contributions address limitations of previous contributions.

### *Chapter 3 – Online Learning with a Budget*

We investigated the effects of user tolerance on model performance in Chapter 3. Specifically, we modelled user tolerance with a *budget* – a three-tuple specifying (1) the *budget size*, meaning the number of annotation requests the user would be willing to respond to, (2) the *budget horizon*, defining the maximum duration of time over which annotations would be requested and (3) the *budget strategy*, which specifies how the annotation requests are distributed during the budget horizon. Using an evaluation-based approach, we showed the effects of a budget specification on the final model classification accuracy and, respectively, how quickly this accuracy is attained.

The budget size, defined as the number of interactions with the user, is arguably intuitive enough for the user to specify it directly. However, budget spending strategies

are arguably not as intuitive. Using the budget strategy the timings of the annotation requests are generated which, as explained in Chapter 3, are sampled over the interval $[0, 1]$. These timings on $[0, 1]$, which are not specific to any physical measure (including physical time as perceived by the user or segment sequence numbers as counted by the activity monitor), are then linearly scaled over $[0, t_{horinzon}]$. It is therefore $t_{horinzon}$ that defines the physical nature of the timings of the annotation requests. In our simulations, $t_{horinzon}$ is defined in terms of number of segments, meaning that the system expects to have distributed all annotation requests by the time $t_{horinzon}$ have been registered. This definition of the horizon is a natural fit to the data available in our simulations: because the dataset (Opportunity [38]) contains only a limited number of annotated activities/segments, so defining the horizon as the total number of segments ensures that each segment can potentially be annotated at most once.

However, in a real-life deployment, we argue that defining the horizon in terms of the number of monitored segments is not intuitive to the user. Instead, we propose that physical time be used as a reference and not the number of monitored segments. Furthermore, the definition can be enlarged to account the instant of physical time when the interaction with the user begins ($t_{start}$), besides the duration of the interaction (which, under these circumstances, could be more conveniently renamed $\Delta t_{horizon}$). As an illustration, if $t_{start} = 09{:}00$ and $\Delta t_{horizon} = 3\text{h}$, then this can be easily understood by the user (i.e. "*I am willing to respond to annotation requests for 3 hours starting at 9:00.*"). Other ways of defining the *time boundaries* for the interaction with the user are certainly possible.

Regardless of whether the annotation schedule targets references sequence numbers or physical time, a shortcoming of the budget-based annotation method in Chapter 3 is that it is not informed in any way by the monitored data. The annotation schedule is computed before the annotation starts and it remains fixed until the budget horizon is reached. A detrimental consequence to this approach is that the method does not target rare activities or activities that are inaccurately classified. There are possibly numerous work-arounds to this problem. For example, one might use the budget not to adhere to user tolerance, but to concentrate annotation requests in the time intervals rich in activities which are poorly estimated by the activity classifier. As

a practical application, suppose the budget horizon is one day. In previous days the classifier has been trained according to daily annotation schedules, but suppose there is are persistent intervals where the classifier does not perform well. Under the assumption that the distribution of activities during each day is *stationary*, a larger number of annotation requests may be scheduled during those times (similarly to Micallef et al. [162]) with the expectation that the resulting annotations would yield considerable gains to classification accuracy.

Nonetheless, a disadvantage to the previous approach is that there is considerable delay (i.e. at least a day in the previous example) between the time that it is recognised that some activities are possibly misclassified and the time that corrective action can be taken in the form of annotations targeted at those activities. Therefore, in subsequent chapters, we proposed alternative annotation methods, including budget-based ones, which probabilistically request annotations from the user immediately when potential misclassifications occur.

Overall, while it can be argued that physical time is, from the user's perspective, a more intuitive measure of the evolution of her activities than the sequence numbers in the activity stream, our work still adds value to how an annotation schedule can be generated according to a pre-specified distribution.

### *Chapter 4 – Online Active Learning in the Lab*

In Chapter 4, we temporarily departed from the considerations for user tolerance and budget-based annotation in Chapter 3 and instead focused on improving classification accuracy by identifying the most promising potential annotations from a stream of activities. We used an online version of Active Learning – a semi-supervised learning method that attempts to increase classification accuracy by identifying highly critical annotations from activities as the activities occur. Our implementation of Online Active Learning operated on a stream of activities, a necessary assumption due to the users' limited short-term memory, as we reasoned earlier. Our results, based on a simulation-based approach on public HAR datasets, show that the model accuracy improvements are greater than the improvements due to Random Selection, a widely used baseline for active learning. This signifies not only that users can provide feedback

about very recent activities, so they could do so using just their short-term memory, but also that HAR model personalisation can be accelerated using Online Active Learning.

The Online Active Learning heuristic converts the classification confidence of a monitored activity segment into probabilities to interrupt the user to annotate the activity. In particular, low classification confidences (which are typically symptomatic of poor classification accuracy) result into high interruption probabilities. The heuristic is tunable via the $\gamma$ parameter as follows: for a fixed classification confidence, higher values of $\gamma$ result in the probability for interruption decreasing, but to a greater extent for classification confidences. Effectively, progressively higher values for $\gamma$ concentrate annotation requests only towards lower classification confidences. Conversely, progressively lower values for $\gamma$ make annotation requests more probable for high confidences. In Chapter 4 the value of the parameter $\gamma = 6$ was chosen empirically, on the one hand, as high enough to illustrate the contrast of classification accuracy between Online Active Learning and Random Selection and, on the other hand, as low enough to not delay the run-time of the simulations (since very high values of $\gamma$ result in a very high rejection rate of potential annotation requests when the model is increasingly personalised and systematically yields high classification confidences).

Even though this approach of using a fixed value for $\gamma$ makes sense from the strictly objective perspective of classification accuracy since results show that Online Active Learning improves with respect to Random Selection, two issues arise. First of all, this style of annotation is budget-agnostic, so, by itself, it is unable to account for user-preferred times and volume of annotations. Second of all, from the user's perspective, it is not immediately intuitive how to choose $\gamma$. The problem is that the parameter is not only highly non-linear in relation to the resulting number of annotation requests, but $\gamma$ is not the only factor influencing the total number of requests. For example, for the same value of $\gamma$, a user engaging in only a few activities will probably be confronted with fewer annotation requests than a user performing a wider range of activities. Both limitations are addressed in Chapter 6 where the annotation heuristic is extended to account for a budget. This shields the user from non-intuitive $\gamma$ values who can instead choose a budget within which to operate Online Active Learning.

### *Chapter 5 – Online Active Learning in the Wild*

In Chapter 5 we discussed the outcomes of deploying within a user study a complete Online Active Learning system implementation. The user study allowed us, primarily, to collect both sensor together with the genuine annotations the users provided and, secondarily, to obtain subjective user feedback on their experience interacting with the system. The model evaluation results from the deployment data show that our Online Active Learning method can collect high quality annotations that lead to HAR model personalisation and accuracy improvement when contrasted with a simplistic *strawman* model. This implies that, even under realistic conditions and when users operate in their natural environment, it is expected that HAR models can be personalised from user-provided annotations.

Additionally, after compiling the user's feedback on interacting with the system, we discovered two aspects. Firstly, if a sufficiently large number of annotations are provided, the users perceived that the system learned and became better at recognising their activities. This leads us to believe that users would start to see the benefits of their input after some time, therefore justifying their effort. Secondly, our annotation mechanism was perceived as invasive and, overall, the participants in our experiment would have preferred fewer annotation requests. However, the level of interruptions could be turned lower so that, even though model personalisation would take longer, the user would not be discouraged from engaging with the system.

However, as noted in Chapter 5, given the limitations due to sample size and familiarity of some of the participants with the members of the research team, it is possible that the subjective feedback will not generalise in the same way to a larger pool of anonymous users. A high degree of bias may therefore emerge from our compiled feedback. Rather, in order to obtain more representative feedback, one can apply in situ methods of obtaining questionnaire feedback from users, similarly to Wang et al. [163]. We did not resort to remote and anonymous users because of the more complicated initial setup stage of the sensors (including exact placement, sensor orientation).

Nonetheless, our qualitative results are still valuable. While there exists a possibility the feedback is not representative for a larger base of users, the concerns raised by

our panel of participants could be taken into account when designing a similar but larger deployment. First of all, the volume and frequency of annotations should be controlled by the users. In Chapters 3 and 6 we suggest annotation mechanisms which control annotation requests according to a specified total number and strategy of distribution. Second of all, even a relatively small panel of participants pointed out that they would like the monitoring application to account for specific activities. Therefore, future deployments should consider whether the choice of activities should rest with the designers of the monitoring application or with the users themselves.

### *Chapter 6 – Online Active Learning with Budget Constraints*

Finally, in Chapter 6, we combined the budget-based annotation method from Chapter 3 with the apparently incompatible Online Active Learning method from Chapters 4 and 5. The budget-based Online Active Learning annotation method builds upon concepts from Chapters 3 and 4 and extends the technical implementation from Chapter refch:daptive.learning.in.the.lab with novel algorithms. The end result is that this new hybrid annotation method makes it possible to balance meeting a budget specification with requesting highly critical annotations. Our results show that it is possible to run a budget-constrained Online Active Learning method that still improves over Random Selection while, on average, adhering closely to the budget specification. The work in this chapter addresses the weaknesses from previous chapters:

- The improvement over Chapter 3 is that annotations are now informed by Online Active Learning and highly critical segments are given preference, even when closely adhering to a budget specification.

- Conversely, the improvement over Chapter 4 is not only that a budget is introduced, but the user can now specify her preferences when it comes with being confronted with annotation requests. Instead of having to choose an arguably unintuitive value of the $\gamma$ parameter (as in Chapter 4), the user can instead declare a budget specification (the total number of annotation, a horizon over which she is accepting to be interrupted and a distribution of number of annotations over this horizon).

The implications for HAR are promising. Firstly, accelerated model personalisation is still possible under budget-constrained Online Active Learning, so highly critical annotations can be identified and provided for users. Secondly, our method ensures that budget specifications are closely followed with little risk of under- or over- spending, so the system does its best to obtain all the annotations the user has committed to provide, with preference given to highly critical ones, without crossing user tolerance boundaries.

## Our Contributions in the Wider Research Context

Having analysed how the different contributions in our thesis relate to each other, we now re-survey key research literature from Chapter 2 and analyse how our contributions fit in the wider research context. The structure of this section mirrors the major structure of Chapter 2.

### *Obtaining Annotations*

In this thesis, we examined multiple related methods of obtaining annotations from users of HAR systems. We investigated the effects on physical activity classification accuracy and on user tolerance only of *reactive* annotations. We defined this class of annotations as those provided by the user of the HAR system herself *after* an activity has already occurred. This is in contrast to *proactive* annotations, which by definition are supplied in advance of executing the activity (i.e. Berchtold et al. [69] and van Kasteren et al. [72]) or *prospective*, which require considerable infrastructure to collect useful ground truth, such as video footage (for instance, Lester et al. [56] or Chavarriaga et al. [38]).

Our approach is similar to Abdallah et al. [74, 75] where the potential usefulness (for example, in terms of expected classification accuracy gains) can be assessed immediately after an activity has been detected. Cleland et al.[73] also propose reactive annotations, but do not use a heuristic and instead annotate all activities indiscriminately. In this current work, we show using empirical evidence that, for the same number of annotations, a heuristic-based annotation approach can outperform random annota-

tion (Online Active Learning versus Random Selection). This has direct consequences on the diversity of the labels collected (e.g. Section refsubsec:periodic.results) and on the speed (which, intuitively is the inverse of the user's annotation effort) with which HAR monitors can be personalised.

On the one hand, given our context sensing limitations, we are forced to dismiss retrospective annotations on the basis of insufficient infrastructure to collect annotations and on the basis of a limited user's short term memory. On the other hand, proactive annotations are a feasible extension to our system. Berchtold et al. [69] and van Kasteren et al. [72] have shown that HAR classifiers can be constructed from these. With reference to our results, contributing with proactive annotations would be equivalent to Random Selection because the decision to generate the annotation is not based on sensor data, which, at the point of annotation, is not yet recorded. However, proactive annotations are an alternative and complementary solution to the *Ignorant Classifier* problem discussed in Chapter 5. Users can recognise or can be convinced that a completely new activity could be annotated proactively to ensure that the user's personalised classifier will have scope to improve this new activity (rather than risk not discovering it). This could be done in conjunction with a Novel Activity Detector (Chapter 5) so that the recall on annotating new or rare activities is increased further. Additionally, in case a NAD is not used, an initial proactive annotation could be employed to fulfil the assumption of an existing corpus of diverse annotations, as we did in Chapter 6 with the purpose of budget-based Online Active Learning.

A key issue with reactive annotations in a mobile context is the user's limited short-term memory Eisen et al. [50]. To avoid problems with unreliable distant memory recall, we propose that annotations are targeted only at the last identified activity segment. For reasons having to do with our segmentation procedure, as explained in Section 3.4.1, the delay between an activity ending and an annotation decision being made was in the region of $10 - 20$s. Previously, Linnap and Rice [77] discovered that this figure is typical of interaction with annotation devices and Cleland et al.[73] had very similar delays in their physical activity annotation user study.

The advantage of reactive annotations over the other types of annotations is that they provide stronger guarantees about the degree of naturalism of the user's context. The

goal of obtaining user-provided annotations throughout this thesis is to *bootstrap* fully personalised HAR classifiers. However, the same techniques that collect annotations from the user's naturalistic environment can potentially be used to adapt existing models, i.e. Abdallah et al.[74, 75] and Nguyen et al. [65], or to expand the vocabulary of physical activities Hossmann et al. [67], Lu et al. [66].

### *The User's Perspective*

One of our main working assumptions is that, given that interrupting the user in order to provide annotations for her own physical activities is both *useful* and *intrusive*, a trade-off must be struck between the two. It is known that user interruptions tend to be intrusive (i.e. Pejovic et al. [85]) and the subjective feedback we collected during our user-based case study in Chapter 5 reinforces this knowledge. Therefore, our main contribution is aimed at attempting to maximise the utility of the annotation process, given a fixed level of acceptable intrusiveness into the user's lifestyle. We measured utility as classification accuracy and ultimately quantified the merits of all the annotation methods in this thesis against this benchmark. Since the accuracy is a key characteristic of a monitoring system, then we argue that the immediate benefits of accurate classification (accurate day-to-day activity monitoring, more informed lifestyle change decisions, etc.) can be conveyed back to the user. This would be in line with notion of *intelligibility* introduced by Lim and Dey [80] to measure and improve interactive systems because the user could be made aware of the benefits and therefore would possibly be willing to collaborate and supply annotations.

As underlined in Chapter 2, it has already been recognised that the process of annotating data is not effort-free and, consequently, there exists a finite amount of annotations that can be provided by any one user. In response to that, we modelled the user tolerance as not only a number of available annotations, but also as a distribution of annotation requests across a time horizon. A possible avenue of further research is the exploration of what are the users' tolerance levels of interacting with an annotation device over longer periods of time than those in our user study (which were the order of hours per participant). With relation to our work, what are the budget specifications users would opt for in real life? As it surfaces from our participants' subjective

feedback, the frequency and/or volume of annotation requests during the user study were collectively perceived to be intrusive to some extent. This finding mirrors a similar user-based case study by Cleland et al. [73] whose participants similarly indicated they were prompted with annotation requests "too often". Therefore, valid questions include *"What constitute acceptable levels of interruption?"* or *"What would the users decide to be these levels of disruption?"*. We believe that answering such questions would contribute towards understanding the previously mentioned trade-off between usefulness and intrusion.

Despite these unanswered questions, our work is not invalidated. The annotation methods proposed throughout this thesis can work with a large class of user tolerance levels which would not lose significant detail if modelled as a budget (see Chapter 3). Therefore regardless of the user's preferred level of disruption, which can be modelled as a budget specification, our annotation methods can meet it. The trade-off between usefuless and intrusion could be studied under different budget criteria, such as:

- Firstly, if the user has strong temporal demands about potential annotations, then a budget-only method (Chapter 3) or a highly constrained budget-based online active learning method (Section 6.4) would be compatible.
- Secondly, if the user has weaker temporal demands, then a more flexible budget-based online active learning (Section 6.2) is preferred since it has greater possibilities to improve the classification accuracy with respect to Random Selection.
- Finally, if the user has no temporal constraints (i.e. she is willing to provide annotations at any time), but she would prefer to reduce her involvement as much as possible while, at the same time, obtaining the greatest improvement in classification accuracy, then a budget-agnostic online active learning method (Chapter 4) is the best option because the decision to annotate depends only on the potential classification accuracy improvements due to the current segment under consideration.

### *Machine Learning*

A constant characteristic throughout all the analyses is the use of classification algorithms. To this end, we used techniques commonly used in activity recognition as surveyed by Bulling et al. [98]. We set up a data processing pipeline that transformed raw sensor data into classification estimates of monitored physical activities.

Our work uses accelerometer sensor data, but, as existing research shows (Shoaib et al. [95], Lara and Labrador [96]), other modalities can be similarly used. Raw sensor readings are transformed into features – a condensed representation suitable for subsequent machine learning. There is a plethora of features from which to choose from [96], including statistical measures and frequency-domain-derived quantities. We use a combination of such features, which shows that our proposed annotation system is potentially applicable to existing configurations without major modifications of the machine learning pipeline. Nonetheless, features from Deep Belief Networks (Erhan et al. [101]) can be more flexible in picking up underlying patterns in data and are a strong alternative for physical activity recognition, as shown by Plötz et al. [103].

As defined in Section 2.1.1, an integral part of an annotation is the segmented sensor data which is characterised by a label. As shown in Chapter 2, segmentation of physical activities remains a difficult research problem, especially for the non-periodic case where we are not aware of any evidence that automatic segmentation does not require some form of prior knowledge about the user's activities. We nonetheless assume that there exists an accurate segmentation for non-periodic activities so that we can evaluate our annotation method to activity segments for this case and to show that this annotation method is robust enough to cope with this difficult learning scenario (relative to the periodic case). For periodic activities, which do not exhibit strong temporal dependencies as non-periodic ones, we proposed an automatic segmentation procedure adapted from Cooper [119]. Our method has built-in assumptions about the nature of the activities, which have to be periodic so that short sequences of individual frames are highly representative of their activity segment. Nonetheless, unlike Cleland et al. [73] who make very strong assumptions about the order the activities, our segmentation procedure can work with arbitrary sequences of activities,

as shown in Chapter 4.

### *Learning Methodologies*

We explored a range of learning methodologies throughout the thesis from fully supervised to semi-supervised. In Chapter 3 we employed fully supervised machine learning. In contrast, in Chapters 4, 5 and 6 we additionally use semi-supervised machine learning (Online Active Learning) to improve classification accuracy of HAR activities. However, as we underlined in Chapter 2, Online Active Learning is not the only semi-supervised mechanism to improve classification accuracy. For example, performance improvement for HAR models is also possible with Transfer Learning [52, 54, 90, 122, 125, 126], Self-Training [78, 123], Co-Training [78, 123] or other Semi-Supervised Learning methods [55, 124], even though these are not substitutes for Online Active Learning when it comes to identifying valuable annotations. Nonetheless, we argue these methods can potentially complement Online Active Learning by furthering the performance gains. We exemplify with this high level procedure:

1. Obtain highly critical annotation using Online Active Learning.
2. Update model.
3. Apply complementary SSL technique to further improve model.
4. Use model from Step 2. to scan for potential annotations and eventually repeat Step 1.

The procedure is illustrated in Fig. 7.1 where we plotted a schematic of the expected evolution of the model's performance. Perhaps the model that is used to scan for annotations is not necessarily the same model that is used to provide the most accurate monitoring. The monitoring model is probably better constructed from different sources of data, depending on the SSL technique used, whereas the annotation model is focused exclusively on the user's ground truth so it may better detect gaps in the user's training set.

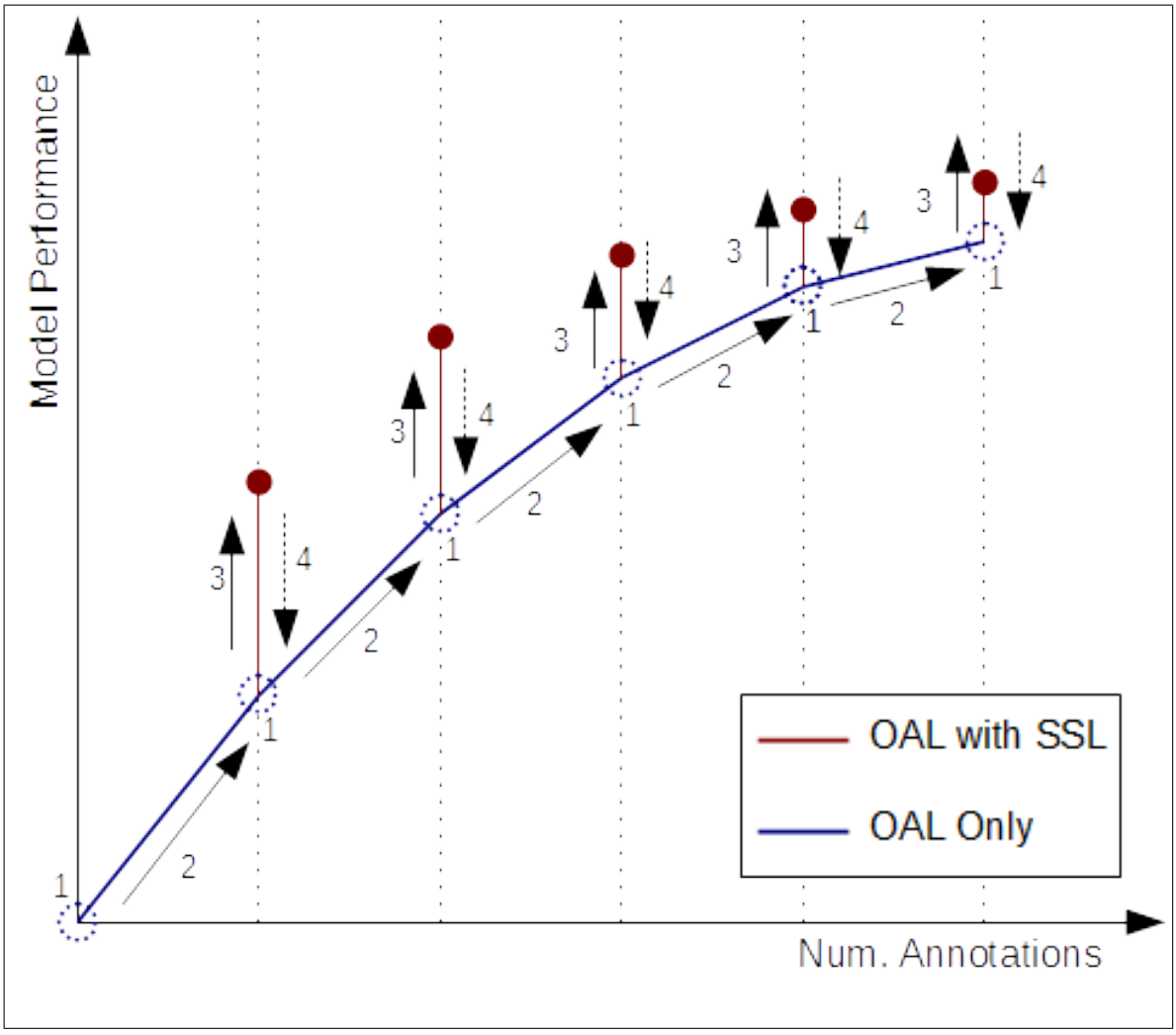



Figure 7.1: Complementing Online Active Learning (OAL) with Other Semi-Supervised Learning (SSL) Methods (Schematic).

In any case, this is a difficult problem, not least given the large amounts of required data for successfully applying a complementary SSL technique and for performance evaluation, and it also presents a complex solution space to navigate. For example, in our solution in Chapter 5, we opted for updateable classifiers that can be bootstrapped progressively in constant time and constant memory complexity on a mobile phone. However, many SSL methods require constructing large sets of throw-away classifiers or processing whole datasets [123, 164] and we argue this is computationally too demanding for a mobile processor to cope with. Substantially more computational power can be harnessed if the data is uploaded to cloud servers which can then construct the required models. This allows intensive computation for learning HAR models [52, 69], but it also presents another problem in the form of cost. Should these servers be centralised and community-shared, they effectively become finite resources because they may serve numerous remote clients only on a limited basis, otherwise they would become over-subscribed. This presents at least two-fold complications. Firstly, should the computational cost of building a user classifier be modelled in some way? Secondly, could the system request more than one annotation before the model was re-constructed on the server? In order to introduce diversity in the set of annotations between model updates, one might investigate batch-mode Active Learning [51] or variants thereof.

### *Machine Learning in HAR Applications*

Our contributions are focused more on the theoretical analysis of variants of online learning, budget-based learning and on analysis of empirical evidence of learning human physical activities from user-provided annotations. As such, we our work does not contribute towards the systems-side analysis of HAR monitoring. Nonetheless, the central piece in our contributions involves the deployment of an actual mobile HAR monitor (in Chapter 5) which had to function within the constraints of the experimental boundaries (user-only input without external interventions; continuous sensing, preprocessing, monitoring and learning updates for the duration of an entire day at the office). Without performing a quantitative analysis on system parameters such as computational power and energy consumption, we can nonetheless report on the

following qualitative system characteristics that ensured that the experiment protocol could be met:

- [**Mobile Computing Power**] The Nexus 5 smartphone used as a base station to collect the sensor data from the wireless sensors had sufficient computational power to perform all the required data reception and processing without back-logging and causing delays to user prompts for annotations.

- [**Battery Life**] Both the smartphone and the WAX9 wireless sensors had enough battery life to last for the entire duration of the experiment. All participants finished in their own time.

  However, we mention that in order to not waste battery power on the smartphone, the screen had to be kept turned off at all times, except when an annotation was requested for the user. After this, the screen had to be turned off again. Still, the phone was prevented from going to sleep and, instead, a CPU lock was used to ensure that the device would not go to sleep when the screen was turned down and that background continuous HAR monitoring could still take place.

# A

# Consent Form

## Consent Form

Participant Identification Number: .....

Gender: Male / Female

Age: .....

***Study description:*** *Human Activity Recognition (HAR) research seeks to construct systems that automatically detect and classify individual movements or overarching behaviours of the user. In order to do build accurate models, sensor data relating to motion is easily collected , but human judgement is still needed to correctly annotate the sensor data.*

*In this user study, we will ask you to wear four bluetooth accelerometer sensors strapped to your body and interact with a smartphone app (which we provide) whenever you are prompted to provide input. You will be asked to perform several light intensity physical activities which will be explained to you. Each activity can be done as many times as you like, preferably 4-5 times each before midday and 4-5 times after midday, for as long as you prefer, and different activities can be interleaved in any order and with breaks in between. As you perform these activities, the smartphone app collects acceleration data from the four sensors and the smartphone's onboard accelerometer and will occasionally ask you to name the activity you were performing 10-20 seconds* ***before*** *the input prompt. The input request is accompanied by audio and vibratory feedback.*

*You will be asked to complete three short paper questionnaires – one immediately before the start of the experiment and two immediately after the end the experiment.*

1. I confirm that I have read and understand the study description for this study. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.

2. I understand that my participation is voluntary and that I am free to withdraw at any time, without giving any reason.

3. I understand that any information given by me will be anonymized and may be used in future reports, articles or presentations by the research team.

4. I confirm that I am fit to perform the physical exertion required during the study, that the level of physical activity is below vigurous for me and that I will perform the physical activities in a safe manner so that I do not endanger myself or anyone else.

5. I agree to take part in the above study.

Researcher .....

Date .....

Signature .....

Name of participant .....

Date .....

Signature .....

# B

# Structure of questionnaires

## Pre-experiment questionnaire

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

2. Do you have any other comments?

# Post-experiment questionnaire 1

1. What do you think of the total number of requests?
2. What do you think of the frequency of requests?
3. Do you think the feedback you provided helped the app learn?
4. When you were asked to name activities, did you have trouble remembering or deciding?
5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?
6. Do you think you would have coped with questions aimed at activities in the more distant past?
7. Do you have any other comments?

# Post-experiment questionnaire 3

1. Do you think the feedback you provided helped the app learn?
2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?
3. If you were to use this app regularly:
   (a) would you like it to learn more about the same activities?
   (b) would you like learn about a modified or another set of activities altogether? Which activities?
   (c) would you like to be able to choose and modify the activities as you use the application?
   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?
4. Would you like to control the frequency or total number of requests from the app?
5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?
6. Which features of the existing system do you think are the most valuable to you?
7. What other features would you like to see?
8. What features you would like to see removed from the system?
9. Do you have any other comments?

# C

# Questionnaire Answers

# Participant 1

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *I feel that giving input of 20% of the time will probably not interfere very much with my day-to-day activities. At this time, I feel that 20% may not be enough to get useful data for the app, but I am not an expert on how the app figures out what activity I am doing.*

2. Do you have any other comments?

   *Not at this time.*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *It was a little annoying as it would often make many requests while I was sitting and not doing anything.*

2. What do you think of the frequency of requests?

   *The frequency seemed to change. As I said before, it seemed like most requests would come while I was sitting. While I was doing the activities, it seemed like it wouldn't ask as much.*

3. Do you think the feedback you provided helped the app learn?

   *I hope so. It was sometimes difficult to tell if the app was asking for what I was just doing a minute ago or if it was asking for what I had started doing 30 seconds ago.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *Yes, this was especially the case when I was changing activities. It was no problem when I was already in the middle of an activity.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *In general, they were timely. I think I would have preferred to arrive the moment I changed activities.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No, I think that would have made the study a lot more difficult.*

7. Do you have any other comments?

   *Nothing about the study itself, but besides the annoyance of being requested for input throughout the day, I continually was asked why I had a phone strapped to*

*my arm. This led to the added annoyance of explaining that I was participating in a study*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Yes, according to the plots I saw, the app was able to predict my activity better.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *No, not really. The activities are kind of out of place in my work environment.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Yes, I would like it to learn more information about how the activities are affecting my health.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *I would also like it to learn about running, cycling, and perhaps some other types of stretches.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes, this would help me tune the routine to my lifestyle.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *I think it would be very useful in other exercise and sport related routines.*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes. Sometimes more are ok, but often times I would prefer fewer requests.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

*Yes, and I wouldn't mind performing some of the activities longer to help boost the accuracy.*

6. Which features of the existing system do you think are the most valuable to you?

   *Having an app that can learn what activities I am performing to tell me how long I was doing each task.*

7. What other features would you like to see?

   *I would like to see integration with some system that could measure calories burned or to remind me to do other things instead of sitting all day.*

8. What features you would like to see removed from the system?

   *The sensors are a bit annoying, but I don't know how to make the system work without them.*

9. Do you have any other comments?

   *Thanks for the interesting study. I hope my participation helps.*

# Participant 2

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *Seems reasonable. I suppose it depends on the robustness of your code! If it were an all day, everyday thing, it might be obnoxious though.*

2. Do you have any other comments?

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *At times it was a bit much, particularly if continuing the same activity (i.e. standing or sitting, but I didn't mind the total # too much. As mentioned in the pre-study questionnaire, if required all day, everyday, it would be too much.*

2. What do you think of the frequency of requests?

   *The timing seemed mostly good, but there were times when I performed an activity and no input was requested. I wasn't sure if this meant it couldn't differentiate my current activity from my previous one.*

3. Do you think the feedback you provided helped the app learn?

   *Yes, overall. However, it didn't seem like the torso exercises were picked up as easily by the app.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *On occasions, especially if I had been shifting through multiple activities (i.e. walking then stopping then walking again like at a crosswalk) that I wasn't sure which activity was being detected. And once I knew which activity I had been doing (forward leans), but accidentally logged it as side leans.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *I think instead of sooner, a smaller range of time would be more useful (i.e. a 5 second difference instead of 10). Because there were times when the activity I was doing 10 seconds before and the one 20 seconds before were different, I had to settle on reporting whatever was being performed an average 15 seconds to stay consistent.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No! I have poor memory.*

7. Do you have any other comments?

   *No.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Yes. The trend shows that the accuracy improved with more annotations with a few exceptions. Since the app/sensors crashed twice early in the experiment and there are two downward spikes early on, this may be correlated.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes and no. I wouldn't be against taking an ergonomic 10 minute stretch/exercise break on occasions but not all the time!*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Yes.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Yes. I think other exercises like lunges (with sensors on two legs) or pushups (they are called something else here) would be great additions.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *I could see the usefulness of adding this feature, but I'm unsure how exactly I'd want to track my activity.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Yes, as previously mentioned in 3b. Especially when performing an exercise routine, it would be useful for performance tracking.*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes, especially when not doing much of anything (like sitting!).*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *Maybe a little, but couldn't the app reach a threshold accurach and then start decreasing requests?*

6. Which features of the existing system do you think are the most valuable to you?

   *Mostly the types of exercises performed (active* vs. total incidences of non-active exercises (i.e. sitting or standing). If someone wanted to improve their fitness regimen, this could be a basic metric that they should exercise more.

7. What other features would you like to see?

   *I think the scrolling to select the activity could lead to mistakes. Perhaps a pop-up educated guess list of activities instead? Oh, and better sensors (in terms of comfort while wearing them).*

8. What features you would like to see removed from the system?

   *I can't think of anything per se.*

9. Do you have any other comments?

   *Nope!*

# Participant 3

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *Seems ok, but might be annoying after some time. We will see.*

2. Do you have any other comments?

   *I like strapping stuff to my body.*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *I think it was appropriate. Some of the requests were missing I think (after exercises) but not many of them.*

2. What do you think of the frequency of requests?

   *OK! During one period of 15-20 min though the frequency got really high for some reason (while seating) and every 30-60 sec request would appear.*

3. Do you think the feedback you provided helped the app learn?

   *I think so.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *Not at all. It was very intuitive (almost subconscious) after I got to know the interface.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *Maybe 5-10 sec instead of 10-20. Especially after the special activities (squats, etc.)*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No, I don't think so.*

7. Do you have any other comments?

   *Nope*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Definitely yes.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes, I would be more tempted. What would really help though is a change in the environment that would facilitate those kind of activities.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Sure.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Walking up the stairs.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes. And maybe input your own activities too.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Running or football.*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes, especially while sitting.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *Not necessarily.*

6. Which features of the existing system do you think are the most valuable to you?

   *N/A*

7. What other features would you like to see?

   *No idea*

8. What features you would like to see removed from the system?

   *So far, none of them were particularly removable. I liked them all.*

9. Do you have any other comments?

   *Nope. Thank you :)*

# Participant 4

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *I think the number and the frequency of input request is fine.*

2. Do you have any other comments?

   *No*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *Too many request in the sitting or standing position*

2. What do you think of the frequency of requests?

   *Quite high frequency of requests in sitting position.*

3. Do you think the feedback you provided helped the app learn?

   *Yes. I interacted with the application quite a lot.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *No*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *Sometimes they were delivered quire late. I would prefer having them sooner (approx. 5 sec.)*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No. I think the question should be asked within 5-10 sec after the activity.*

7. Do you have any other comments?

   *Good recognition of walking.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Yes. I think my feedback helped because the learning curved reached the value of about 0.9*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Yes*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Yes. Learning recognising running or jumping would make it more applicable to sport activities*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes. I would like be able to define my own activities and let the app learn them.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Yes.*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes. Because sometimes it is very irritating*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

*No.*

6. Which features of the existing system do you think are the most valuable to you?

   *The ability to label the activity*

7. What other features would you like to see?

   *Refining your own activity*

8. What features you would like to see removed from the system?

   *None.*

9. Do you have any other comments?

   *Need to work on recognising sitting and standing as these activities are the main sources of the requests*

# Participant 5

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *It sounds fine to me*

2. Do you have any other comments?

   *No*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *It's okay on overall but it might be differently (equally) distributed across the day*

2. What do you think of the frequency of requests?

   *During the afternoon it was a bit too frequent*

3. Do you think the feedback you provided helped the app learn?

   *Yes because it was "requesting" while I was doing the "specifics" exercises. So it knew exactly what I was doing*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *No, just a bit at the beginning*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *Slightly sooner, after the device realised a "change" in the type of activity*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *Probably not*

7. Do you have any other comments?

   *No.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Yes as across the time the accuracy trend increased*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Only if the activities could be short, as 5 mins in total every 4 hours*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Yes it should focus on the same activities in order to predict and learn better*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Another set should use instead of the control one, not together. Sport activities would be interested, maybe applied to football (predict different actions of players)*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *As in answer b) it would be interesting in sport/gyms activities.*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes, in order to set according to how busy I am.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

*No, I prefer a better accuracy.*

6. Which features of the existing system do you think are the most valuable to you?

   *The real-time prediction (type of movement)*

7. What other features would you like to see?

   *A live accuracy trend in order to check how the system is learning. A suggestion on the "worse" predicted activity in order to record it better.*

8. What features you would like to see removed from the system?

   *None*

9. Do you have any other comments?

   *No.*

# Participant 6

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *I think it would be 100 times.*

2. Do you have any other comments?

   *1) I suggest to include more activities and expand the target audience from office workers to more diverse ones. 2) You can consider new capabilities of iOS (Health module) to expand your research.*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *around 40 requests, it was much less than I expected.*

2. What do you think of the frequency of requests?

   *It was not frequent on the first half of the day, but on the second half I was asked nearly every 10 minutes.*

3. Do you think the feedback you provided helped the app learn?

   *Yes*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *Yes, especially when it was repetitive walking and standing combination.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *No, I think they were more activity sensitive. In same cases I wish it was earlier, especially on walking and standing case.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *Definitely yes.*

7. Do you have any other comments?

   *I wish we could have included more types of activities.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Yes, considering the number of requests, I guess the feedback helped the app a little. I am sure with more requests, the results would increase.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes, although some activities might look not appropriate in the office while others are working around especially squats.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *I might include this type of learning in the mobile from my activities as a privacy issue so I might be a bit cautious where and how much I let it learn.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Not at the moment.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Yes*

4. Would you like to control the frequency or total number of requests from the app?

   *I like to have control on where the requests happens. E.g.: I want to give feedback when I am only at work or gym and in that areas, I can respond to as many number of requests as asked.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *In early stages which the learning is not mature enough, I am happy to give annotations but after a while, I might get annoyed if I were asked frequently.*

6. Which features of the existing system do you think are the most valuable to you?

   *The fact that it makes me do some exercise since at work, I am sitting most of the time.*

7. What other features would you like to see?

   *It might be good if you could provide a recommender which can suggest me do some customized specific exercises based on my previous actions.*

8. What features you would like to see removed from the system?

   *Number of sensors can be less.*

9. Do you have any other comments?

   *Do hope to see it coming as a real-world application.*

# Participant 7

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *The random 20% seems like a low number which won't be annoying, the occassionally asking for input to reinforce activities seems as though it may be a little much but it depends how often ocassionally is.*

2. Do you have any other comments?

   *I am interested to see if the random 20% will be more or less frequent than the first period.*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *Total number of requests felt like a lot but mainly after the second period, they were more often.*

2. What do you think of the frequency of requests?

   *frequency of requests was managable for the first period but annoying and to many for the second. Oddly the second period would have a high frequency for 10 minutes then nothing for ˜30 then a high frequency again.*

3. Do you think the feedback you provided helped the app learn?

   *I think for the first period yes, second no.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *I had trouble with the 10/20 second delay as I am not very good at timing. So often from walking -> standing I would be unsure what it was asking me about. 10/20 seconds is too long.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *Sooner within 1-2 seconds maximum (although I may just be impatient), however it is hard to judge 10-20 seconds*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No I wouldn't have remebered unless prompted before hand to remember exactly what I was doing.*

7. Do you have any other comments?

   *I found the sensors a little uncomfortable after a few hours and they got in the way a bit.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *I would say some did however for activities such as torso movement & squats I did not seem to.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *I would definitely add the torso movements in and more frequent walking as it helped to loosen my back up.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *I would maybe incorporate more activities.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *So running, maybe more core activities like situps, but that would be in the thought process of the application being a sports performance app.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *This would be good as you progressed.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Yes so training and keeping a record of what you were doing. But it would be really good for sports physio.*

4. Would you like to control the frequency or total number of requests from the app?

   *Maybe but I liked the first period where it asked you as it was learning.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *No so this is the same as the above answer I liked it learning and I felt it asked a lot less quite quickly in the first period.*

6. Which features of the existing system do you think are the most valuable to you?

   *The graph of annotations was interesting as it made you think more about what you were doing through the day.*

7. What other features would you like to see?

   *Shorter interval between asking what you were doing i.e not 10-20 seconds.*

8. What features you would like to see removed from the system?

   *Random interval were a little annoying.*

9. Do you have any other comments?

   *If it was an application for rehabilitation maybe a prettier UI with encouragement for users - just a thought.*

# Participant 8

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *I am ready to provide any necessary inputs.*

2. Do you have any other comments?

   *None*

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *The total number of request were not too many. They fit well into my schedule.*

2. What do you think of the frequency of requests?

   *The requests were well spaced and not too frequent.*

3. Do you think the feedback you provided helped the app learn?

   *I provided the most accurate feedback that I can. I believe that this helped the app to learn.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *Not really. The 10 to 20 seconds period for activity feedback was short-enough to readily remember the recent-past activity. None-the-less, a real-time notification be appreciated.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *The questions were very timely. I didn't prefer them any sooner.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No, it would be quite difficult remembering the activities.*

7. Do you have any other comments?

   *No.*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *I think my feedback helped the app learn, since most of my activities were recorded and adequately analysed and represented in the results produced.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes, I think these activities provided a source of exercising in my daily routine.*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *No, I will prefer that it captures other activities in addition to the current activities.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *- Arm activities(eg. during stretching)*
   *- Leg activities (eg. running)*
   *NB: People can go running during lunchtime or while climbing the staircase.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes, some days may not register certain activities and it will be good to exclude such activities.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Yes, this can be applied to other routines that I will be involved in.*

4. Would you like to control the frequency or total number of requests from the app?

   *This option will be a good addition, but the current frequency of requests do not bother me at all.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *No, I will rather choose accuracy to get a good representation of my activities.*

6. Which features of the existing system do you think are the most valuable to you?

   *The notification feature seems quite valuable to me.*

7. What other features would you like to see?

   *A feature that presents a final analysis of my day's activities will be a good addition.*

8. What features you would like to see removed from the system?

   *Not applicable.*

9. Do you have any other comments?

   *No.*

# Participant 9

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *I think it's ok for me.*

2. Do you have any other comments?

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *Too much*

2. What do you think of the frequency of requests?

   *Sometime too frequency*

3. Do you think the feedback you provided helped the app learn?

   *I don't know*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *No*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *It's ok.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *I don't know*

7. Do you have any other comments?

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *I am not sure*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes, I am very interesting how many activities I did a day*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *No*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *I have no idea*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *Sports*

4. Would you like to control the frequency or total number of requests from the app?

   *Yes, this is very important.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *Yes*

6. Which features of the existing system do you think are the most valuable to you?

   *except sitting*

7. What other features would you like to see?

   *like sport things*

8. What features you would like to see removed from the system?

   *I don't know*

9. Do you have any other comments?

# Participant 10

## *Pre-experiment questionnaire*

1. The app is designed so that it will likely ask you to annotate a newly performed activity which you hadn't performed before and you will also occasionally be asked for input which reinforces already known activities. Subsequently you will be asked, at random, to provide input for about 20% of all detected activities. What do you think of this number and frequency of input requests that will be directed at you?

   *Sounds reasonable for a one off study participation. Might be too invasive and distracting if updates occurred continuously throughout a normal working period. If the updates are quick (sub 5 seconds) then perhaps it would be fine.*

2. Do you have any other comments?

## *Post-experiment questionnaire 1*

1. What do you think of the total number of requests?

   *The total number of requests was fine for an "interested" audience (people who already log their activity to gain personal psychological insight).*

2. What do you think of the frequency of requests?

   *Had the requests been more consistently "spaced" throughout the time period I think that would have been better. As it was there were periods of very frequent questions which could get intrusive.*

3. Do you think the feedback you provided helped the app learn?

   *It is difficult to judge. Perhaps if the app were to present it's guess of the activity you were recently performing (which you could then confirm or correct) then it would be easier to see whether your feedback was having a +ve impact on accuracy.*

4. When you were asked to name activities, did you have trouble remembering or deciding?

   *Occasionally it was difficult to decide whether you were walking or standing. It would be good to ask the user to observe ~30 seconds of stillness after each activity, so that they are more clearly demarked.*

5. Do you think the questions were delivered in a timely manner? Would you have preferred them sooner? If so, when?

   *The time elapsed between activity end question was generally fine, however it would be good to see an indicator of how much time has passed since the end of the Activity which the app is asking about.*

6. Do you think you would have coped with questions aimed at activities in the more distant past?

   *No - location data and other abstract forms of physical telemetry for the activity in question could increase the time gap without increasing the difficulty in remembering.*

7. Do you have any other comments?

   *I could see the benefit to a regime of regularly performing and tracking physical activity throughout the work day. I felt like I had more energy and was productive for longer. Furthermore, the required breaks from work allows users to benefit from the proven impact of altered work schedules with many short breaks (see Pomodoro technique).*

## *Post-experiment questionnaire 2*

1. Do you think the feedback you provided helped the app learn?

   *Not particularly - there does not appear to be a significant upward trend in the accuracy of the model over time, which indicates that the app is not learning well from my feedback.*

2. Taking into account the activities and the environment in which you performed the activities, would be more tempted you incorporate this sort of physical exertion in your daily routine?

   *Yes - as previously mentioned - I have noticed feeling more energic and therefore productive, lpus the impact on my day would be minimal as I already take several short breaks at regular intervals throughout the day (Pomodoro).*

3. If you were to use this app regularly:

   (a) would you like it to learn more about the same activities?

   *Not necessarily. I think users should be able to pick activities from a larger set.*

   (b) would you like learn about a modified or another set of activities altogether? Which activities?

   *Some activities I would like to see be: traversing stairs, sit ups.*

   (c) would you like to be able to choose and modify the activities as you use the application?

   *Yes.*

   (d) would you like to apply it to another situation, e.g. specific fitness routines, sports, medical rehabilitation or something else?

   *The flexibility to learn series of activities constituting either a fitness routine or a physio therapy session would be a very good addition.*

4. Would you like to control the frequency or total number of requests from the app?

*Not in a fine grained fashion, though a slider which correlated to request frequency would make the app more useful to a wider audience.*

5. Some of the annotation requests were made on the basis of them being expected to lead to increased accuracy gains. Would you sacrifice such gains so that you attain a lower level of interruption?

   *No - if the model is to inaccurate then any interruption was pointless.*

6. Which features of the existing system do you think are the most valuable to you?

   *The ability to discard a question, the answer to which you are unsure of, is valuable*

7. What other features would you like to see?

   *- Summaries of activities visible by a user throughout the day.*
   *- Reminders to perform activity (of some kind) during long periods of relative inactivity*

8. What features you would like to see removed from the system?

   *- N/A*

9. Do you have any other comments?

   *- N/A*

# D

# Code and Data

The code used to analyse data and to compile the results in this thesis and the dataset used in Chapter 5 can be found in the following *git* repository: `https://github.com/miu/phd_thesis_code_and_data`

Each analysis consists of a number of a number of R scripts with function definitions, which include all the simulations and algorithms evaluated in the present thesis, *runner scripts* which invoke the simulations to process the data and associated plotting scripts which were used to compile the presentations of the results.

For each chapter, the scripts are as follows (paths are relative to main folder):

- **Chapter 3:**
  - `code/R/do.opportunity.budget.R`
  - `code/R/plot.opportunity.budget.R`
- **Chapter 4:**

  Periodic activities:
  - `code/R/al.usc.had/run.R` (and using the SimulateOnlineALForSubject function) – for annotating single frames
  - `code/R/al.usc.had/plot.*` – to produce the graphs
  - `code/R/usc.had.seg.al/do.index.sampling.R` – incorporates the segmentation function
  - `code/R/usc.had.seg.al/do.usc.had.ideal.segments.activity.expansion.R` – for analysing a biased distribution of activities

  - `code/R/usc.had.seg.al/plot.*` – to produce the graphs

  Nonperiodic activities (`code/R/opportunity.butterworth.dtw/`):

  - `run.simulation.R`
  - `plot.exploratory.baseline.and.pop.R`

- **Chapter 5:**

  - `code/R/user.study/do.compute.performance.R`
  - `code/R/user.study/do.plot.R`

- **Chapter 6:**

  Periodic activities (`code/R/al.usc.had/`):

  - `run.modulated.R` – without additional constraint
  - `run.modulated.beta.R` – with additional constraint
  - `plot.*`

  Nonperiodic activities (`code/R/opportunity.butterworth.dtw/`):

  - `run.simulation.modulated.beta.R`
  - `plot.modulated.budget.R` – without additional constraint
  - `plot.modulated.budget.beta.R` – with additional constraint

The code for the Android app used in Chapter 5 is located in the following locations:

- **With Speculative NAD:** `code/speculative-nad`
- **With Restrained NAD:** `code/restrained-nad`

The dataset collected as part of Chapter 5 is located in the following location: `dataset/`

# Bibliography


[1] T. Miu, T. Plötz, P. Missier, and D. Roggen, "On strategies for budget-based online annotation in human activity recognition," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, (New York, NY, USA), pp. 767–776, ACM, 2014.

[2] T. Miu, P. Missier, and T. Plötz, "Bootstrapping personalised human activity recognition models using online active learning," in *Proceedings of the 14th IEEE International Conference on Ubiquitous Computing and Communications*, 2015.

[3] M. Weiser, "The computer for the 21st century," *Scientific American*, 1991.

[4] G. D. Abowd, "What next, Ubicomp? Celebrating an intellectual disappearing act.," in *Proc. Int. Conf. Ubiquitous Comp. (UbiComp)*, 2012.

[5] N. Barengo, G. Hu, T. Lakka, H. Pekkarinen, A. Nissinen, and J. Tuomilehto, "Low physical activity as a predictor for total and cardiovascular disease mortality in middle-aged men and women in Finland," *European Heart Journal*, vol. 25, pp. 2204–2211, Dec. 2004.

[6] M. Aadahl, M. Kjaer, and T. Jø rgensen, "Associations between overall physical activity level and cardiovascular risk factors in an adult population.," *European journal of epidemiology*, vol. 22, pp. 369–78, Jan. 2007.

[7] G. Hu, Q. Qiao, K. Silventoinen, J. G. Eriksson, P. Jousilahti, J. Lindström, T. T. Valle, a. Nissinen, and J. Tuomilehto, "Occupational, commuting, and leisure-time physical activity in relation to risk for Type 2 diabetes in middle-aged Finnish men and women.," *Diabetologia*, vol. 46, pp. 322–9, Mar. 2003.

[8] J. E. Manson, E. B. Rimm, M. J. Stampfer, G. a. Colditz, W. C. Willett, a. S. Krolewski, B. Rosner, C. H. Hennekens, and F. E. Speizer, "Physical activity and incidence of non-insulin-dependent diabetes mellitus in women.," *Lancet*, vol. 338, pp. 774–8, Sept. 1991.

[9] F. B. Hu, T. Y. Li, G. a. Colditz, W. C. Willett, and J. E. Manson, "Television watching and other sedentary behaviors in relation to risk of obesity and type 2 diabetes mellitus in women.," *JAMA : the journal of the American Medical Association*, vol. 289, pp. 1785–91, Apr. 2003.

[10] C. C. Gotay, "Behavior and cancer prevention.," *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, vol. 23, pp. 301–10, Jan. 2005.

[11] J. A. Levine, S. K. Mccrady, L. M. Lanningham-foster, P. H. Kane, R. C. Foster, and C. U. Manohar, "The Role of Free-Living Daily Walking in Human Weight Gain and Obesity," *Diabetes*, 2008.

[12] A. K. Yancey, C. M. Wold, W. J. McCarthy, M. D. Weber, B. Lee, P. a. Simon, and J. E. Fielding, "Physical inactivity and overweight among Los Angeles County adults.," *American journal of preventive medicine*, vol. 27, pp. 146–52, Aug. 2004.

[13] W. H. Organization, *Global Recommendations on Physical Activity for Health*. WHO Press, 2010.

[14] N. Hex, C. Bartlett, D. Wright, M. Taylor, and D. Varley, "Estimating the current and future costs of type 1 and type 2 diabetes in the uk, including direct health costs and indirect societal and productivity costs," *Diabetic Medicine*, vol. 29, no. 7, pp. 855–862, 2012.

[15] N. Townsend, K. Wickramasinghe, P. Bhatnagar, K. Smolina, M. Nichols, J. Leal, R. Luengo-Fernandez, and R. Mike, *Coronary heart disease statistics - A compendium of health statistics, 2012 edition*. British Heart Foundation, 2012.

[16] N. B. Oldridge, "Economic burden of physical inactivity: healthcare costs associated with cardiovascular disease," *European Journal of Cardiovascular Prevention & Rehabilitation*, vol. 15, no. 2, pp. 130–139, 2008.

[17] T. Lakka and C. Bouchard, "Physical activity, obesity and cardiovascular diseases," in *Atherosclerosis: Diet and Drugs*, pp. 137–163, Springer, 2005.

[18] K. T. Verkooijen, G. A. Nielsen, and S. P. Kremers, "The association between leisure time physical activity and smoking in adolescence: an examination of potential mediating and moderating factors," *International journal of behavioral medicine*, vol. 15, no. 2, pp. 157–163, 2008.

[19] G. Papathanasiou, M. Papandreou, A. Galanos, E. Kortianou, E. Tsepis, V. Kalfakakou, and A. Evangelou, "Smoking and physical activity interrelations in health science students. is smoking associated with physical inactivity in young adults," *Hellenic J Cardiol*, vol. 53, no. 1, pp. 17–25, 2012.

[20] S. Liangpunsakul, D. W. Crabb, and R. Qi, "Relationship among alcohol intake, body fat, and physical activity: A population-based study," *Annals of Epidemiology*, vol. 20, no. 9, pp. 670 – 675, 2010.

[21] P. Scarborough, P. Bhatnagar, K. K. Wickramasinghe, S. Allender, C. Foster, and M. Rayner, "The economic burden of ill health due to diet, physical inactivity, smoking, alcohol and obesity in the uk: an update to 2006âĂŞ07 nhs costs," *Journal of Public Health*, 2011.

[22] I.-M. Lee, E. J. Shiroma, F. Lobelo, P. Puska, S. N. Blair, and P. T. Katzmarzyk, "Effect of physical inactivity on major non-communicable diseases worldwide: an analysis of burden of disease and life expectancy," *The Lancet*, vol. 380, no. 9838, pp. 219 – 229, 2012.

[23] D. W. Dunstan, B. Howard, G. N. Healy, and N. Owen, "Too much sitting âĂŞ a health hazard," *Diabetes Research and Clinical Practice*, vol. 97, no. 3, pp. 368 – 376, 2012.

[24] N. Owen, "Sedentary behavior: Understanding and influencing adults' prolonged sitting time," *Preventive Medicine*, vol. 55, no. 6, pp. 535 – 539, 2012.

[25] G. A. M. Ariens, W. van Mechelen, P. M. Bongers, L. M. Bouter, and G. van der Wal, "Physical risk factors for neck pain," *Scandinavian Journal of Work, Environment and Health*, vol. 26, no. 1, p. 7, 2000.

[26] P. T. Katzmarzyk, T. S. Church, C. L. Craig, and C. Bouchard, "Sitting time and mortality from all causes, cardiovascular disease, and cancer," *Med Sci Sports Exerc*, vol. 41, no. 5, pp. 998–1005, 2009.

[27] M. Virtanen, *Long Working Hours and Health in Office Workers: A Cohort Study of Coronary Heart Disease, Diabetes, Depression and Sleep Disturbances*. University College London (University of London), 2012.

[28] J. Chau, M. Daley, S. Dunn, A. Srinivasan, A. Do, A. Bauman, and H. van der Ploeg, "The effectiveness of sit-stand workstations for changing office workers' sitting time: results from the stand@work randomized controlled trial pilot," *International Journal of Behavioral Nutrition and Physical Activity*, vol. 11, no. 1, 2014.

[29] G. N. Healy, E. G. Eakin, A. D. LaMontagne, N. Owen, E. A. Winkler, G. Wiesner, L. Gunning, M. Neuhaus, S. Lawler, B. S. Fjeldsoe, and D. W. Dunstan, "Reducing sitting time in office workers: Short-term efficacy of a multicomponent intervention," *Preventive Medicine*, vol. 57, no. 1, pp. 43 – 48, 2013.

[30] A. A. Thorp, B. A. Kingwell, N. Owen, and D. W. Dunstan, "Breaking up workplace sitting time with intermittent standing bouts improves fatigue and musculoskeletal discomfort in overweight/obese office workers," *Occupational and Environmental Medicine*, vol. 71, pp. 765–771, Nov. 2014.

[31] L. L. Andersen, K. B. Christensen, A. Holtermann, O. M. Poulsen, G. SjÃÿgaard, M. T. Pedersen, and E. A. Hansen, "Effect of physical exercise interventions on musculoskeletal pain in all body regions among office workers: A one-year randomized controlled trial," *Manual Therapy*, vol. 15, no. 1, pp. 100 – 104, 2010.

[32] H. J. C. G. Coury, R. F. C. Moreira, and N. B. Dias, "Evaluation of the effectiveness of workplace exercise in controlling neck, shoulder and low back pain: a systematic review," *Brazilian Journal of Physical Therapy*, vol. 13, no. 6, pp. 461–79, 2009.

[33] D. W. Dunstan, B. A. Kingwell, R. Larsen, G. N. Healy, E. Cerin, M. T. Hamilton, J. E. Shaw, D. A. Bertovic, P. Z. Zimmet, J. Salmon, and N. Owen, "Breaking up prolonged sitting reduces postprandial glucose and insulin responses," *Diabetes Care*, vol. 35, no. 5, pp. 976–983, 2012.

[34] S. Crouter, J. Churilla, and J. Bassett, DavidR., "Estimating energy expenditure using accelerometers," *European Journal of Applied Physiology*, vol. 98, no. 6, pp. 601–612, 2006.

[35] A. Helmer, F. MÃijller, O. Lohmann, A. Thiel, F. Kretschmer, M. Eichelberg, and A. Hein, "Integration of smart home health data in the clinical decision making process," in *Biomedical Engineering Systems and Technologies* (M. FernÃąndez-Chimeno, P. L. Fernandes, S. Alvarez, D. Stacey, J. SolÃľ-Casals, A. Fred, and H. Gamboa, eds.), vol. 452 of *Communications in Computer and Information Science*, pp. 354–366, Springer Berlin Heidelberg, 2014.

[36] H. Gjoreski, B. Kaluža, M. Gams, R. Milić, and M. Luštrek, "Ensembles of multiple sensors for human energy expenditure estimation," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, (New York, NY, USA), pp. 359–362, ACM, 2013.

[37] V. T. van Hees, R. C. van Lummel, and K. R. Westerterp, "Estimating activity-related energy expenditure under sedentary conditions using a tri-axial seismic accelerometer," *Obesity*, vol. 17, no. 6, pp. 1287–1292, 2009.

[38] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, Jan. 2013.

[39] N. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing," *Proceedings of the 5th International ICST Conference on Pervasive Computing Technologies for Healthcare*, 2011.

[40] A. Helal, D. J. Cook, and M. Schmalz, "Smart home-based health platform for behavioral monitoring and alteration of diabetes patients.," *Journal of diabetes science and technology*, vol. 3, pp. 141–148, Jan. 2009.

[41] P. Roy, S. Giroux, B. Bouchard, A. Bouzouane, C. Phua, A. Tolstikov, and J. Biswas, "A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimerâĂŹs patients," in *Activity Recognition in Pervasive Intelligent Environments* (L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, eds.), vol. 4 of *Atlantis Ambient and Pervasive Intelligence*, pp. 33–58, Atlantis Press, 2011.

[42] M. Swan, "The quantified self: fundamental disruption in big data science and biological discovery," *Big Data*, vol. 1, no. 2, pp. 85–99, 2013.

[43] E. K. Choe, N. B. Lee, B. Lee, W. Pratt, and J. A. Kientz, "Understanding quantified-selfers' practices in collecting and exploring personal data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), pp. 1143–1152, ACM, 2014.

[44] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay, "Activity sensing in the wild: A field trial of ubifit garden," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, (New York, NY, USA), pp. 1797–1806, ACM, 2008.

[45] E. Velloso, A. Bulling, and H. Gellersen, "Towards qualitative assessment of weight lifting exercises using body-worn sensors," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, (New York, NY, USA), pp. 587–588, ACM, 2011.

[46] Business Insider. http://uk.businessinsider.com/the-wearable-computing-market-report-2014-10?op=1?r=US. Accessed 24.04.2015.

[47] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[48] D. Sculley, "Online Active Learning Methods for Fast Label-Efficient Spam Filtering," in *Conference on Email and AntiSpam*, 2007.

[49] S. S. Intille, L. Bao, E. M. Tapia, and J. Rondoni, "Acquiring in situ training data for context-aware ubiquitous computing applications," *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, vol. 6, no. 1, pp. 1–8, 2004.

[50] M. L. Eisen, J. A. Quas, and G. S. Goodman, *Memory and Suggestibility in the Forensic interview (Personality and Clinical Psychology Series)*. 2001.

[51] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, 2010.

[52] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao, "Enabling large-scale human activity inference on smartphones using community similarity networks (csn)," *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11*, p. 355, 2011.

[53] J. Rebetez, H. F. Satizábal, and A. Perez-Uribe, "Reducing user intervention in incremental activityrecognition for assistive technologies," in *Proceedings of the 2013 International Symposium on Wearable Computers*, ISWC '13, (New York, NY, USA), pp. 29–32, ACM, 2013.

[54] D. Cook, K. Feuz, and N. Krishnan, "Transfer learning for activity recognition: a survey," *Knowledge and Information Systems*, vol. 36, no. 3, pp. 537–556, 2013.

[55] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors.," *IEEE Trans. Pattern Analysis and Machine Intell. (TPAMI)*, vol. 33, pp. 2521–2537, Feb. 2011.

[56] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Pervasive Computing* (K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, eds.), vol. 3968 of *Lecture Notes in Computer Science*, pp. 1–16, Springer Berlin Heidelberg, 2006.

[57] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, "Recofit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), pp. 3225–3234, ACM, 2014.

[58] C. Pham and P. Olivier, "Slice & dice: Recognizing food preparation activities using embedded accelerometers," in *Ambient Intelligence* (M. Tscheligi, B. de Ruyter, P. Markopoulus, R. Wichert, T. Mirlacher, A. Meschterjakov, and W. Reitberger, eds.), vol. 5859 of *Lecture Notes in Computer Science*, pp. 34–43, Springer Berlin Heidelberg, 2009.

[59] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "Elan: a professional framework for multimodality research," in *Proceedings of Language Resources and Evaluation Conference (LREC)*, 2006.

[60] S. Bagaveyev and D. J. Cook, "Designing and evaluating active learning methods for activity recognition," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014.

[61] W. S. Lasecki, Y. C. Song, H. Kautz, and J. P. Bigham, "Real-time crowd labeling for deployable activity recognition," in *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 1203–1212, ACM, 2013.

[62] T. Maekawa, Y. Yanagisawa, Y. Kishino, and K. Ishiguro, "Object-Based Activity Recognition with Heterogeneous Sensors on Wrist," *Pervasive Computing*, pp. 246–264, 2010.

[63] J. M. Smyth and A. A. Stone, "Ecological Momentary Assessment Research In Behavioral Medicine," *Happiness Studies*, pp. 35–52, 2003.

[64] S. Intille, E. Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, L. Bao, and K. Larson, "Tools for studying behavior and technology in natural settings," in *UbiComp 2003: Ubiquitous Computing* (A. Dey, A. Schmidt, and J. McCarthy, eds.), vol. 2864 of *Lecture Notes in Computer Science*, pp. 157–174, Springer Berlin Heidelberg, 2003.

[65] L.-V. Nguyen-Dinh, U. Blanke, and G. Tröster, "Towards scalable activity recognition: Adapting zero-effort crowdsourced acoustic models," in *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, p. 18, ACM, 2013.

[66] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pp. 165–178, ACM, 2009.

[67] T. Hossmann, C. Efstratiou, and C. Mascolo, "Collecting big datasets of human activity one checkin at a time," in *Proceedings of the 4th ACM international workshop on Hot topics in planet-scale measurement*, pp. 15–20, ACM, 2012.

[68] N. Kawaguchi, H. Watanabe, T. Yang, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, H. Hada, S. Inoue, *et al.*, "Hasc2012corpus: Large scale human activity corpus and its application," in *2nd International Workshop on Mobile Sensing*, 2012.

[69] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl, "ActiServ : Activity Recognition Service for Mobile Phones," in *International Symposium on Wearable Computing*, 2010.

[70] S. Harada, J. Lester, K. Patel, T. S. Saponas, J. Fogarty, J. A. Landay, and J. O. Wobbrock, "Voicelabel: using speech to label mobile sensor data," in *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 69–76, ACM, 2008.

[71] E. Hoque, R. F. Dickerson, and J. A. Stankovic, "Vocal-diary: A voice command based ground truth collection system for activity recognition," in *Proceedings of the Wireless Health 2014 on National Institutes of Health*, WH '14, (New York, NY, USA), pp. 9:1–9:6, ACM, 2014.

[72] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, (New York, NY, USA), pp. 1–9, ACM, 2008.

[73] I. Cleland, M. Han, C. Nugent, H. Lee, S. McClean, S. Zhang, and S. Lee, "Evaluation of prompted annotation of activity data recorded from a smart phone," *Sensors*, vol. 14, no. 9, pp. 15861–15879, 2014.

[74] Z. Abdallah, M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Streamar: Incremental and active learning with evolving sensory data for activity recognition," in *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, vol. 1, pp. 1163–1170, Nov 2012.

[75] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, 2015.

[76] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, (New York, NY, USA), pp. 337–350, ACM, 2008.

[77] M. Linnap and A. Rice, "Managed participatory sensing with yousense," *Journal of Urban Technology*, vol. 21, no. 2, pp. 9–26, 2014.

[78] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, pp. 1–7, March 2010.

[79] S. S. Intille, K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik, "A living laboratory for the design and evaluation of ubiquitous computing technologies," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, (New York, NY, USA), pp. 1941–1944, ACM, 2005.

[80] B. Y. Lim and A. K. Dey, "Investigating Intelligibility for Uncertain Context-Aware Applications," *Proc. Int. Conf. on Ubiquitous Computing*, 2011.

[81] A. Meschtscherjakov, "Investigating emotional attachment to mobile devices and services from an hci perspective," tech. rep., University of Salzburg, 2008.

[82] R. H. Thaler and C. R. Sunstein, *Nudge: Improving decisions about health, wealth and happiness*. London: Penguin, Mar. 2008.

[83] B. P. Bailey and J. a. Konstan, "On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state," *Computers in Human Behavior*, vol. 22, pp. 685–708, July 2006.

[84] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), pp. 3055–3064, ACM, 2014.

[85] V. Pejovic and M. Musolesi, "Interruptme: Designing intelligent prompting mechanisms for pervasive applications," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014.

[86] J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM Trans. Comput.-Hum. Interact.*, 2005.

[87] A. Kapoor and E. Horvitz, "Experience Sampling for Building Predictive User Models : A Comparative Study," in *SIGCHI Conference on Human Factors*, 2008.

[88] D. Helmbold and S. Panizza, "Some label efficient learning results," in *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, COLT '97, (New York, NY, USA), pp. 218–230, ACM, 1997.

[89] J. Attenberg and F. Provost, "Online active inference and learning," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, (New York, NY, USA), pp. 186–194, ACM, 2011.

[90] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, G. Tröster, H. Sagha, R. Chavarriaga, J. del R. Millán, D. Bannach, K. Kunze, and P. Lukowicz, "The opportunity framework and data processing ecosystem for opportunistic activity and context recognition," *International Journal of Sensors, Wireless Communications and Control, Special Issue on Autonomic and Opportunistic Communications*, pp. 102–125, Dec. 2011.

[91] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-People Mobile-Phone Based Activity Recognition," in *International Joint Conference on Artificial Intelligence*, pp. 2545–2550, 2006.

[92] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2010.

[93] J. Pärkkä, L. Cluitmans, and M. Ermes, "Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree.," *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 14, pp. 1211–5, Sept. 2010.

[94] Y. Xuel and L. Jinl, "A Naturalistic 3D Acceleration-based Activity Dataset & Benchmark Evaluations," in *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp. 4081–4085, 2010.

[95] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[96] O. Lara and M. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys Tutorials, IEEE*, vol. 15, pp. 1192–1209, Third 2013.

[97] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, Bruges, Belgium 2013.

[98] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, pp. 33:1–33:33, Jan. 2014.

[99] D. Figo, P. C. Diniz, D. R. Ferreira, and J. a. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal Ubiquitous Comput.*, vol. 14, pp. 645–662, Oct. 2010.

[100] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, sOc-EUSAI '05, (New York, NY, USA), pp. 159–163, ACM, 2005.

[101] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Mar. 2010.

[102] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, pp. 1–127, Jan. 2009.

[103] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pp. 1729–1734, AAAI Press, 2011.

[104] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.

[105] E. Keogh and M. Pazzani, "Scaling up dynamic time warping to massive datasets," in *Principles of Data Mining and Knowledge Discovery* (J. Åżytkow and J. Rauch, eds.), vol. 1704 of *Lecture Notes in Computer Science*, pp. 1–11, Springer Berlin Heidelberg, 1999.

[106] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 4930–4933, Aug 2007.

[107] J. O. Laguna, A. G. Olaya, and D. Borrajo, "A dynamic sliding window approach for activity recognition," in *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, UMAP'11, (Berlin, Heidelberg), pp. 219–230, Springer-Verlag, 2011.

[108] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mob. Comput.*, vol. 10, pp. 138–154, Feb. 2014.

[109] S.-L. Chua, S. Marsland, and H. Guesgen, "Behaviour recognition from sensory streams in smart environments," in *AI 2009: Advances in Artificial Intelligence* (A. Nicholson and X. Li, eds.), vol. 5866 of *Lecture Notes in Computer Science*, pp. 666–675, Springer Berlin Heidelberg, 2009.

[110] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive and Mobile Computing*, vol. 10, Part B, no. 0, pp. 155 – 172, 2014.

[111] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, Feb 1989.

[112] J. Deng and H. Tsui, "An hmm-based approach for gesture segmentation and recognition," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3, pp. 679–682 vol.3, 2000.

[113] N. C. Krishnan, P. Lade, and S. Panchanathan, "Activity gesture spotting using a threshold model based on adaptive boosting," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pp. 155–160, July 2010.

[114] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recogn.*, vol. 41, pp. 2010–2024, June 2008.

[115] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Thirteenth International Conference on Machine Learning*, (San Francisco), pp. 148–156, Morgan Kaufmann, 1996.

[116] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, "Online context recognition in multisensor systems using dynamic time warping," in *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pp. 283–288, Dec 2005.

[117] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1 ed., 1957.

[118] T. Stiefmeier, D. Roggen, and G. Tröster, "Gestures are strings: Efficient online gesture spotting and classification using string matching," in *Proceedings of the ICST 2Nd International Conference on Body Area Networks*, BodyNets '07, (ICST, Brussels, Belgium, Belgium), pp. 16:1–16:8, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[119] M. Cooper, "Video segmentation combining similarity analysis and classification," *Proceedings of the 12th annual ACM international conference on Multimedia - MULTIMEDIA '04*, no. 1, p. 252, 2004.

[120] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd ed., 2011.

[121] R. Chavarriaga, H. Bayati, and J. D. Millán, "Unsupervised adaptation for acceleration-based activity recognition: Robustness to sensor displacement and rotation," *Personal Ubiquitous Comput.*, vol. 17, pp. 479–490, Mar. 2013.

[122] T. Maekawa and S. Watanabe, "Training data selection with user's physical characteristics data for acceleration-based activity modeling," *Personal Ubiquitous Comput.*, vol. 17, pp. 451–463, Mar. 2013.

[123] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pp. 81–88, Sept 2008.

[124] M. Stikic and B. Schiele, "Activity recognition from sparsely labeled data using multi-instance learning," in *Location and Context Awareness* (T. Choudhury, A. Quigley, T. Strang, and K. Suginuma, eds.), vol. 5561 of *Lecture Notes in Computer Science*, pp. 156–173, Springer Berlin Heidelberg, 2009.

[125] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, pp. 1345–1359, Oct. 2010.

[126] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, (Berlin, Heidelberg), pp. 342–357, Springer-Verlag, 2008.

[127] H. Alemdar, T. van Kasteren, and C. Ersoy, "Using active learning to allow activity recognition on a large scale," in *Ambient Intelligence* (D. Keyson, M. Maher, N. Streitz, A. Cheok, J. Augusto, R. Wichert, G. Englebienne, H. Aghajan, and B. KrÃűse, eds.), vol. 7040 of *Lecture Notes in Computer Science*, pp. 105–114, Springer Berlin Heidelberg, 2011.

[128] B. M. Abidine, B. Fergani, M. Oussalah, and L. Fergani, "A new classification strategy for human activity recognition using cost sensitive support vector machines for imbalanced data," *Kybernetes*, vol. 43, no. 8, pp. 1150–1164, 2014.

[129] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with evolving streaming data," in *Machine Learning and Knowledge Discovery in Databases* (D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, eds.), vol. 6913 of *Lecture Notes in Computer Science*, pp. 597–612, Springer Berlin Heidelberg, 2011.

[130] E. Hoque and J. Stankovic, "Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, 2012.

[131] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *Proceedings of the 2012 16th Annual International Symposium on Wearable Computers (ISWC)*, ISWC '12, (Washington, DC, USA), pp. 17–24, IEEE Computer Society, 2012.

[132] J. Smith, N. Dulay, M. Toth, O. Amft, and Y. Zhang, "Exploring concept drift using interactive simulations," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pp. 49–54, March 2013.

[133] J. Zhu, H. Wang, E. Hovy, and M. Ma, "Confidence-based stopping criteria for active learning for data annotation," *ACM Trans. Speech Lang. Process.*, vol. 6, pp. 3:1–3:24, Apr. 2010.

[134] A. Vlachos, "A stopping criterion for active learning," *Comput. Speech Lang.*, vol. 22, pp. 295–312, July 2008.

[135] F. Laws and H. Schätze, "Stopping criteria for active learning of named entity recognition," in *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, (Stroudsburg, PA, USA), pp. 465–472, Association for Computational Linguistics, 2008.

[136] K. Shin and P. Ramanathan, "Real-time computing: a new discipline of computer science and engineering," *Proceedings of the IEEE*, vol. 82, pp. 6–24, Jan 1994.

[137] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. 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*, pp. 37–40, Oct 2007.

[138] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, (New York, NY, USA), pp. 71–84, ACM, 2010.

[139] H. Martín, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal Ubiquitous Comput.*, vol. 17, pp. 675–695, Apr. 2013.

[140] F. C. Bull, M. W. Kreuter, and D. P. Scharff, "Effects of tailored, personalized and general health messages on physical activity," *Patient Education and Counseling*, vol. 36, no. 2, pp. 181 – 192, 1999.

[141] A. Brand, "Public health genomics and personalized healthcare: a pipeline from cell to society," *Drug Metabolism and Drug Interactions*, vol. 27, no. 3, 2012.

[142] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proc. Int. Conf. on Ubiquitous Comp. (UbiComp)*, 2008.

[143] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Proc. Int. Conf. Pervasive Computing (Pervasive)*, 2004.

[144] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active Learning with Statistical Models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[145] T. Plötz, P. Moynihan, C. Pham, and P. Olivier, "Activity Recognition and Healthier Food Preparation," in *Activity Recognition in Pervasive Intelligent Environments*, Atlantis Press, 2010.

[146] H. Sagha, S. T. Digumarti, R. Chavarriaga, A. Calatroni, D. Roggen, and G. Troester, "Benchmarking classification techniques using the Opportunity human activity dataset," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[147] S. H. Hirano, R. G. Farrell, C. M. Danis, and W. A. Kellogg, "Walkminder: Encouraging an active lifestyle using mobile phone interruptions," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, (New York, NY, USA), pp. 1431–1436, ACM, 2013.

[148] S. Rosenthal, A. Dey, and M. Veloso, "Using decision-theoretic experience sampling to build personalized mobile phone interruption models," in *Pervasive Computing* (K. Lyons, J. Hightower, and E. Huang, eds.), vol. 6696 of *Lecture Notes in Computer Science*, pp. 170–187, Springer Berlin Heidelberg, 2011.

[149] A. Kapoor, E. Horvitz, and M. Way, "On Discarding , Caching , and Recalling Samples in Active Learning," in *Uncertainty in Artificial Intelligence*, 2007.

[150] M. Zhang and A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, (Pittsburgh, Pennsylvania, USA), September 2012.

[151] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, (New York, NY, USA), pp. 40:1–40:8, ACM, 2012.

[152] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, pp. 37–66, Jan. 1991.

[153] T. Giorgino, "Computing and visualizing dynamic time warping alignments in r: The dtw package," *Journal of Statistical Software*, vol. 31, pp. 1–24, 8 2009.

[154] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematics Software*, vol. 3, pp. 209–226, September 1977.

[155] N. Bicocchi, M. Mamei, and F. Zambonelli, "Detecting activities from body-worn accelerometers via instance-based algorithms," *Pervasive Mob. Comput.*, vol. 6, pp. 482–495, Aug. 2010.

[156] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[157] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Eleventh Conference on Uncertainty in Artificial Intelligence*, (San Mateo), pp. 338–345, Morgan Kaufmann, 1995.

[158] "Android activity javadoc." Accessed 11.12.2014.

[159] G. Schay, *Introduction to Probability with Statistical Applications*. Birkhäuser Boston, 2007.

[160] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

[161] T. C. Mills, *Time series techniques for economists*. Cambridge: Cambridge University Press, 1990.

[162] N. Micallef, M. Just, L. Baillie, and G. Kayacik, "Stop questioning me!: towards optimizing user involvement during data collection on mobile devices," in *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, pp. 588–593, ACM, 2013.

[163] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. 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*, UbiComp '14, (New York, NY, USA), pp. 3–14, ACM, 2014.

[164] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, (New York, NY, USA), pp. 193–200, ACM, 2007.