TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ROBERT E. GUINNESS

CONTEXT AWARENESS FOR NAVIGATION APPLICATIONS

Doctor of Science thesis

Examiner: Prof. TBD
Examiner and topic approved by the
Faculty Council of the
Faculty of Faculty of Computing and
Electrical Engineering
on Day Month 2015

ABSTRACT

ROBERT E. GUINNESS: Context Awareness for Navigation Applications
Tampere University of Technology
Doctor of Science thesis, xx pages, x Appendix pages
xxxxxx 201x
Licentiate Degree Programme in Computing and Electrical Engineering
Major: Pervasive Computing
Examiner: Prof. TBD
Keywords:

The abstract is a concise 1-page description of the work: what was the problem, what was done, and what are the results. Do not include charts or tables in the abstract.

Put the abstract in the primary language of your thesis first and then the translation (when that is needed).

PREFACE

The research work presented in this thesis was carried out between November 2011 and May 2014. I have chosen to complete a "compendium-style" dissertation, in part because I have already had the pleasure of preparing a monograph when co-authoring a book with Prof. Ruizhi Chen, published in July 2014. I have no great desire to repeat such an experience yet. As many who have published such monographs can attest, it takes a lot out of you!

Due to other responsibilities, as well as a bad case of the "it's-not-good-enough-yet" syndrome, it took me more than one year to finalize and publish some of the results of my doctoral research in article format. With the aid of gentle nudging from my colleagues and superiors, I prepared the summary content for this compendium mostly between January and July 2015.

There are two particular experiences I'd like to share that also motivated me for completing this dissertation. The first is when I was asked to be a reviewer for an article submitted to one highly-esteemed journal. When I realized that my work, in my own opinion, was superior to that which I was reviewing, I felt suddenly cured of the above-mentioned syndrome. This is one of the side benefits of peer review. The second was when I was participating in an interview of a beloved colleague (he got the job!). We asked him if he could describe one achievement of which he was most proud. Instead of pointing to one particular academic achievement, such as a highly-cited paper, he pointed out another kind of achievement: the fact that he can look back at his publications and realize that some of the early ones were poor but that there has been a steady improvement in the quality over the years. Since hearing that, this is what I aim for: Not to publish the perfect gem some day but to continually put out my work-in-progress for others to see and hopefully benefit from.

Thank yous....

Kirkkonummi, X.Y.2015

Robert E. Guinness

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

CC license	Creative Commons license
LaTeX	Typesetting system for scientific documentation
SI system	Système international d'unités, International System of Units
TUT	Tampere University of Technology
URL	Uniform Resource Locator

a	acceleration
F	force
m	mass

The abbreviations and symbols used in the thesis are collected into a list in alphabetical order. In addition, they must be explained upon first usage in the text.

1. INTRODUCTION

1.1 Background and Motivation

We are currently witnessing an era of technological convergence that rivals some of the great technological upheavals of modern history[1]. The steam engine, the electric lamp, the transistor, the jetliner, the artificial satellite—it is in this same revered company that we can place the technological revolution we are now undergoing. According to authors Erik Brynjolfsson and Andrew McAfee, we are living in a "second machine age" (where the first machine age began with James Watt's steam engine), which they describe as "an inflection point in the history of our economies and societies because of digitization." (Brynjolfsson and McAfee, 2014; p. 11). They define digitization as "converting things into bits that can be stored on a computer and sent over a network" (*ibid.*, p. 10). The resulting digital information has remarkably different properties from the industrial products of the first machine age, a topic which Brynjolfsson and McAfee explore in detail in their book. They define "digital technologies" as "those that have computer hardware, software, and networks at their core" (*ibid.*, p. 9). It is within this wider context of digital technologies and the second machine age that this thesis is best understood.

Digital technologies is a broad category; therefore, it is useful to narrow the focus to a few key technologies that are driving the development of the second machine age. This thesis focuses on four such technologies and looks at how they have impacted a particular application area, namely navigation. The chosen technologies include: (1) mobile telecommunication devices, (2) the Internet, (3) positioning technologies, and (4) a wide range of inexpensive yet highly capable sensors, namely microelectromechanical systems (MEMS). We note that these four technologies have converged over the course of a few decades, so that the changes are clearly evident within one human generation (i.e. 20-30 years). All of these technologies came to a technological crossroads in the late 20th century and early 21st century, so that a child born and raised in the 21st century will have vastly different technological

[1] By "technological convergence", we mean that a set of technologies has undergone rapid advances simultaneously and thus have become available for technological uptake in combinatorial ways.

possibilities, compared to one born and raised in the 20th century.

The first major manifestation of this technological convergence, especially with respect to consumer markets, is the so-called "smartphone", which incorporates or supports all four of the above-mentioned technologies. Looking at the history of mobile devices, it is difficult to say which mobile phone can be considered the first smartphone. In terms of marketing, the Ericsson R380, released in 2000, was the first mobile phone to be called a smartphone. In terms of the four technologies listed above, the Samsung SCH-S310, introduced in 2005, was probably the first to exhibit all four. The first iPhone was released in 2007, and the first Android phone was released in 2008.

About 64 million smartphones were sold globally in 2006 (Canalys, 2007), and by 2008 this number exceeded 139 million (Gartner, 2009). By 2012, there were already more than one billion smartphones in use worldwide (Strategy Analytics, 2013). This number is forecast to reach nearly 2.5 billion in 2015 (Korea Times, 2014). These devices allow their users to stay "connected" virtually everywhere they go, and consequently anyone can connect to these billion plus users from any networked device, including desktop computers and "land-line" phones—no matter where the user is located or travelling to. Ironically, in many technologically advanced societies, it is now considered a societal and/or behavioral challenge for one to go "off the grid" or "disconnected" for any extended period of time.

It is our view that the smartphone is only the first manifestation of this technological revolution. Many other so-called "smart" devices are soon to follow: "smartwatches" and the use of various wearable sensors may soon become a mainstay consumer habit. In addition, the same technologies that have made smartphones possible and popular are quickly making their way into existing everyday devices, including cars, home appliances, and even toothbrushes. Furthermore, it is not just consumer markets that are being transformed but also many industrial markets, ranging from manufacturing to commercial shipping. It would be naïve to speculate exactly how this revolution will play out in the coming decades, but is clear is that it is already changing the lifestyles, habits, and possibilities of people living in the early 21st century, especially those who can afford these (currently) "high-end" consumer devices.

Aside from being a convergence of new digital technologies, is there any unifying concept or principle that is underlying this technological revolution? Some would argue that it is the increased levels of *mobility* that these technologies provide. Others have rallied under the banner of *ubiquitous computing* or *pervasive computing*, which describes the fact that computing devices can now be found nearly everywhere one looks. Certainly these are two important characteristics giving wind to this revolution, but we argue in this thesis towards another underlying principle that provides a common thread and deep insight into how our relationship to these computing devices is changing.

One common development, of course, is the increasing ability of computing devices to fulfil various user desires, e.g. to download large amounts of data at high speeds, to capture or render various high-quality multimedia content, to store and edit content in various ways, etc. What is not advancing or expanding—at least, not at any considerable rate—is the patience or attention span of the users themselves. Therefore, users are expecting (consciously or not) that their devices will "do more" with essentially the same total quantity and quality of human input. Fortunately, however, these devices are rapidly advancing in their ability to know what their users want or need without the user having to explicitly formulate and express these desires to the computer. This is the goal under which this thesis is motivated and focused—to improve our understanding of how computing devices can better understand us and our needs.

The primary method by which this thesis aims to achieve this goal is through *machine learning*. According to Tom Mitchell and co-authors, "machine learning research seeks to develop computer systems that automatically improve their performance through experience" (Mitchell et al., 1990). This is our favourite definition of machine learning among the many found in the literature, but we note that achieving such a system is incredibly difficult. Most methods that go by the name of "machine learning" fail to meet the definition in terms of *automatically* improving performance. Nonetheless, the discipline of machine learning has grown in recent decades, and the set of techniques going by the name of machine learning are indeed very powerful. In many ways, machine learning has become the preferred framework for building up systems that understand users' needs. Some observers may note that such systems exhibit—or at least attempt to exhibit—*artificial intelligence*.

Artificial intelligence has been an elusive goal of computer science researchers ever since the term was coined in 1955[2]. Although computers have not yet replicated human intelligence in a general sense, there are many tasks of increasing complexity that computers can already perform equally well or even better than the most gifted, well-trained humans. As detailed in (Brynjolfsson and McAfee, 2014), computers

[2]Although McCarthy is usually credited with coining the term artificial intelligence, we note that its first usage in the literature was a paper co-authored by McCarthy, Minsky, Rochester, and Shannon (McCarthy et al., 1955). Therefore, it is not entirely clear who first came up with this term, and in an interview even McCarthy himself could not recall.

have been programmed to beat even the best human players of the game-show *Jeopardy!*, to write corporate earnings previews for *Forbes.com* that are indistinguishable from ones written by humans, and to diagnose breast cancer from images of tissue as good as or even better than pathologists could otherwise do[3]. Such examples demonstrate the increasing practicality of artificial intelligence, but what about understanding users' needs? Is it possible for a computer or computing system (including various sensors) to know what its user needs or wnats before he or she makes any keystroke or swipes any touchscreen? Such a system would be considered by many to exhibit a high level of artificial intelligence.

1.2 Research objectives and Scope

The goal stated above is ambitious and open-ended. It is our view that we are not even close to unleashing the full potential of computing devices to understand their users. In many ways, smartphones and other so-called *smart devices* are not yet "smart". They have the "braun" and not the brains, in the sense that they are powerful and capable but deficient in understanding the user's needs. This thesis aims to improve the state-of-the-art in a computer's ability to understand situations or contexts that humans find themselves in. Mobile computing researchers have adopted the term *context awareness* to refer to this ability. In other domains, such as aviation, maritime, and military domains, the term used is situational awareness (or situation awareness)[4]. In particular, this thesis will focus on how machine learning can be utilized for building context or situation awareness, in order to solve problems in navigation. Thus, we have limited the scope of the research to a reasonably-sized domain. That being said, improvements in the state-of-the-art in context awareness have wide-ranging applications, and it is our hope that the few example applications given in this thesis are seen as merely examples and not as end goals in themselves.

In this thesis, we focus on three tasks related to context awareness that are relevant to the field of navigation: (1) to recognize the mode of motion that a smartphone user is undergoing outdoors and (2) to recognize the activity of a smartphone user in an indoor office environment, and (3) to determine the optimal path of a ship travelling through ice-covered waters. These tasks are very different from one another, especially the third task with respect to the first two, demonstrating the breadth of problems encompassed by the topic of context awareness. They were chosen, in

[3] To be precise, what Brynjolfsson and McAfee describe is a system, known as C-Path, that helped to diagnose breast cancer and also identified new features of breast cancer tissue that were shown to be good features for predicting survival.

[4] For consistency, in this thesis we primarily use the term context awareness, although it can be considered synonymous with the term situation(al) awareness.

part, to show how machine learning can be a powerful tool to tackle a wide range of different problems. They also demonstrate wildly different aspects of "understanding users' needs" for different types of users.

The first task is important for navigation because a navigation system can adapt and improve its performance based on the motion mode in which it is used, but it would be easier if the user did not have to manually change the modes of the navigation system when he or she transitions, e.g. from walking to driving. In other words, a context-aware navigation system would automatically that a pedestrian user needs a pedestrian navigation system and a driving user needs a car navigation system; it would adapt itself automatically according to these different needs.

Similarly, the second task provides possible enhancements for a navigation or position tracking system that must work also indoors. For example, if the system detects that a user is working in a static position (e.g. seated at a desk), then it can apply a positioning filter that assumes little or no changes in user position (and perhaps go into a low power consumption mode), but when it detects that the user has stood up, it can change the filter to one that assumes greater possibilities for movement. If the system later detects that the user has done some routine activity, e.g. fetched a fresh cup of coffee, it can apply a post-processing filter to refine the position tracking history, perhaps removing outliers or some other desired refinement.

The third task is a rather classic problem in maritime navigation, but surprisingly this function has been and continues to be performed in a manual way (i.e. the ship captain or navigator manually choosing the route based on ice charts, local observations, and experience). It is also becoming increasingly important to find efficient paths through ice-covered waters due to the opening up of northern sea routes, as well as increased wintertime maritime transport in general (e.g. in the Baltic Sea). In terms of understanding the users needs, this capabiity means that if maritime conditions change such that the captain or navigator needs to alter its route, based on changing ice conditions or other factors, an "ice-aware" navigation system could automatically inform the ship's crew that a new route is recommended and even suggest the optimal route to the crew.

A plethora of other examples of the utility of context awareness could be given, even within the strict confines of navigation, but due to limitations in time, this thesis will only investigate the above three examples, which have been researched and published in separate publications and republished here for completeness.

1.3 Main Contributions

This research explores an important and previously underexamined link between machine learning and context awareness and exploits this link to demonstrate possible applications in the field of navigation. The author has developed and described in this thesis a conceptual framework for the multi-step processing of raw sensor data into contextual information and also provided a framework for describing contextual information in terms of seven key questions. This theoretical or conceptual development, covered in Chapter 2, benefits the research community by making abstract and ambigious concepts such as "context" and "context awareness" more concrete and clearly defined. During the timeframe which this thesis covers, the author has co-authored a textbook titled *Geospatial Computing in Mobile Devices*, where the mentioned frameworks were described in detail. Two of the chapters from this textbook, where the author of this thesis was the primary author, are included in this thesis, dealing with the topics "context awareness" [P1] and "contextual reasoning" [P2].

In addition, this thesis contains three scientific publications, previously published in scientific journals or scientific conference proceedings. The contributions of the author to these works include:

1. A systematic evaluation of various machine learning algorithms applied to the problem of detecting "mobility contexts", including consideration of the computational cost of the resulting classifiers, due to the intended use in mobile devices [P3].

2. Contributing to the development of a combined probabilistic Location-Motion-Context (LoMoCo) model used to detect human behavior (i.e. activities) in an indoor office environment [P4]. This work includes development of a ubiquitous positioning system, an effort to which the author contributed to over a several year period and documented extensively in several other publications.

3. The development and preliminary evaluation of a novel algorithm for optimizing the routes of ships travelling in ice-covered waters, a so-called "ice-aware" route optimization system [P5]. The continued development of this algorithm is being carried out in several ongoing research projects

1.4 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides a theoretical and historical overview of the topic of context awareness. Chapter 3 provides an overview of machine learning. Chapter 4 summarizes and provides an overview of the included publications. Finally, Chapter 5 offers some conclusions that can be drawn from the author's overall work to date in context awareness and provides some suggestions for future areas of research and development.

2. CONTEXT AWARENESS

As stated in the introduction, *context awareness* is the term adopted by mobile computing researchers to describe a computer's ability to understand (i.e. be aware of) the situation or context in which it is operating. Of particular emphasis is the *human* context (i.e. the computer *user's* situation), but device-specific context can also be of importance to the extent that it can affect the user (e.g. low battery of the device may affect how the user uses the device and even cause him or her to alter plans based on this situation).

Many definitions of context and context awareness have been proposed, usually reflecting different discipline-specific perspectives. The word context figures prominently in diverse fields including linguistics, psychology, neuroscience, law, and computer science. Due to the great number of definitions, some researchers have used techniques such as latent semantic analysis (LSA) and principal component analysis (PCA) to find the relationships between the many definitions of context (Foltz et al., 1998; Bazire and Brezillon, 2005). Others have attempted to formalize the concept mathematically, a topic to which we will turn in Section 2.3 (e.g. McCarthy, 1993).

Following the approach in (Chen and Guinness, 2014), we first present a "layman's" definition, in order to provide a working, notional notion of context for our initial discussion. In later sections, we will define it more precisely. Let us start by seeing how context is defined in a dictionary. In the Merriam Webster Dictionary, the word context has two definitions (Merriam-Webster, 2014):

1. the parts of a discourse that surround a word or passage and can throw light on its meaning
2. the interrelated conditions in which something exists or occurs : ENVIRONMENT, SETTING

In this thesis, as in (Chen and Guinness, 2014), we adopt the second definition. This is because we are not directly concerned with human discourse but rather with conditions of an environment or setting that can be "understood" by computers.

Clearly, these two definitions are interrelated—discourse is the way that humans articulate their understanding of an environment or setting. Put in another way, natural language is how humans encode contextual information. In this thesis, we will focus on techniques that computers can use to represent and process context without human intervention. When we refer to context, we refer directly to the conditions in the environment rather than representations of context, such as discourse. As noted in the introduction chapter, *situation* can be used as a synonym for context. We see no reason to distinguish between the two terms, although we note that some formalisms make a distinction (see Akman and Surav, 1996).

With this working definition of context established, we proceed to the remainder of the chapter, which is organized as follows. First, we provide a simple framework for specifying a context (i.e. the "interrelated conditions") in Section 2.1. Then, in Section 2.2 we present a historical overview of context awareness, which also serves as a brief review of the context awareness literature. Next, Section 2.3 presents context in a more formalized manner, using one of the dominant formalizations found in the literature, the Propositional Logic of Context (PLC). In this section, we also discuss the differences between PLC and the notion of context described above. Section 2.4 describes the sources and methods of context awareness. Finally, Section 2.5 concludes the chapter with some summarizing remarks.

2.1 A Framework for Contextual Information

Because context is such an abstract concept, it is useful to choose some techniques for describing a particular context. These techniques can be used to build a framework for expressing contextual information. The goal of this section is to describe one such technique. We make no claim that this technique or framework is a definitive one, nor that it is complete in the sense of exhaustively covering the concept of context.

In our view, the goal of context-aware systems is essentially to mimic the way that humans understand and describe situations, contexts, conditions, or events (we use all these terms almost interchangeably, although they may emphasize different aspects, such as fixed versus dynamic elements). According to this goal, we might employ the classic technique of journalism (since journalism is an age-old craft for describing conditions and events), known as the Five Ws: Who, What, Where, When, and Why (Wikipedia, 2015). This technique can be traced back to the late 2nd century BC when Hermagoras of Temnos defined seven elements of circumstance, which includes (in addition to the Five Ws) "in what manner" and "by what means" (Bennett, 2005).

Using these questions as a framework (with a slightly different order), the following provides an example of elements of a particular context:

What: A small gathering of colleagues for lunch

Who: Present are Mary, Philip, George, and Anita

Where: 60.1609°N, 24.5460°E (WGS84); inside the lunch-room of the Finnish Geospatial Research Institute in Masala, Finland

When: Thursday, 12 March 2015 at 12:03PM

Why: Because it is lunchtime, and it is the custom for this group of colleagues to eat lunch together.

In What Manner: Mary's smartphone is experiencing small, sporadic movements but is mostly remains in a constant orientation. Mary's smartwatch is experiencing more dramatic but also sporadic movements. Both sources of motion data are consistent with a user who is sitting and having a casual conversation and/or eating lunch. Multiple human voices are engaged in conversation of an informal and lively manner.

By What Means: All of the above information has been sensed or reasoned by the sensors and software existing in a smartphone and a smartwatch, plus some additional sensor data recorded by a networked node installed in the lunch-room. In this case, the smartphone is a Samsung Galaxy S5 with Android 4.4.2 OS, which includes a GPS receiver, Wifi-based positioning engine, Bluetooth connectivity, microphone and audio analyzer, ambient light sensor, accelerometers, gyroscopes, compass, and magnetometers. The smartwatch is an LG G Watch with accelerometers, gyroscopes, Bluetooth connectivity, microphone, and an audio analyzer. It runs Android Wear 5.0.1.

This depiction of the situation is not likely to win a Pulitzer Prize in Journalism, probably because the situation is not particularly interesting. Also, note that it has not been formulated into prose but rather is more like a set of notes that a journalist might jot down for later use (except maybe for the latitude and longitude coordinates and the motion description). The "By What Means" section can also be thought of as notes as to the "source" that the journalist might record along with the other information (especially if the account is second-hand).

Details of this framework for contextual information can be found in (Chen and Guinness, 2014). For the purposes of this thesis, we will note only some essential points:

- "What" usually refers to the *activity context*, that is, what is actually happening. In some contexts, there might be little actual "action" taking place, but it may also be of interest for some purposes that "nothing is happening".

- "Who" refers to the human characters in the context. When speaking about context-aware mobile devices, the user of the mobile device in question is usually the main character, whereas others in the environment can be thought of as supporting characters. The "who" portion can also be summarized as the *user and social context.*

- "Where" refers to the *location context.* The most important point to note is that location can be expressed in many different ways: geographic coordinates, an address, or some semantic representation such as "the Finnish Geospatial Research Institute" or perhaps more personalized, such as "my workplace".

- "When" is the *time* and *date context.* We need only to be careful about specifying things like time zones. In addition, it may be important to have, in addition to the time and date, some common sense or semantic knowledge about meaningful aspects, such as "this is after work hours" or "today is a holiday". Time can be specified either as a specific moment, such as in the example above, or as a time or date segment (e.g. 12-15 March 2015).

- "Why" can be thought of as the *motivational context*, e.g. Why is the user doing that? Why is this event taking place?, etc. It could also be appropriate to encode information about whether the context is normal or unusual, as well as an explanation for the unusual events. For example, if a person normally commutes to work along *routeA*, and in the present he or she is driving along *routeB*, then "why" would be a good place to capture the fact that "there was an accident along *routeA*, so the alternate *routeB* was chosen".

- "In What Manner" is a bit of a "catch all" category. It is used to provide additional details that do not fit nicely into any of the other categories. This is less than ideal for any formal system of context, but rather than attempting to list out all the possible categories of context (which is probably impossible), we believe it is more practical to have an "other" category. One way that we use this category is to capture the *motion context.* This is similar to the activity context, but it is more focused on detailed attributes of the motion. For example, if the current activity of a user is "dancing", then "in what manner" might be used to capture the type of dance and the tempo in which the user is dancing.

- "By What Means", as mentioned above, is to capture the source of the contextual information. It includes information about the devices and sensors used in the context-aware system, as well as the reasoning methods employed.

The above framework is a work-in-progress. As context-aware systems develop, we suspect that the framework may change slightly. It is difficult to anticipate what types of contextual information will become important in the future, but this framework should be broad enough to encompass most types of contextual information. Also, our intention is not to have a rigid framework that constrains all future context-aware systems, but rather it is to provide a rough skeleton upon which to experiment and build more elaborate and detailed context ontologies.

2.2 History

According to our literature review, the first explicit reference to context awareness was in a 1994 paper by Schilit and Theimer, where they use the term *context-aware computing* to describe software that can "adapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time." (Schilit and Theimer, 1994). Earlier, however, we can find strong but implicit references to the concept of context awareness. For example, in the 1991 article, titled "The Computer for the 21st Century," Weiser provides a fictional account of a number of different automated or computer-assisted functions made possible by "ubiquitous computing". Although not specifically highlighted by Weiser, the necessity of these computers to understand context is clearly evident. Another article published in 1991 by Want et al. may be the first implementation of a context-aware device described in the literature. By the mid-1990s, many different implementations of context-aware devices can be found, including the ParcTab, stick-e notes, CyberGuide, and CyberDesk. By 2001, the research field was active enough to support a special issue of the journal Human-Computer Interaction, which provides an excellent review of the state-of-the-art in context awareness for that time period (Moran and Dourish, 2001).

Going further back, however, the concept of context has been studied in computer science research for many years. As early as 1963, John McCarthy, one of the "fathers of AI", began developing *situation calculus* as a "formal system in which facts about situations, goals and actions can be expressed" (McCarthy, 1963). A situation is defined as "the complete state of affairs at some instant of time", thus, it is roughly equivalent to our definition of context. Beginning in 1987, McCarthy began to consider the concept of context explicitly and attempted to formalize it. This work on context and related theory will be presented in Section 2.3 below. Formalisms of context, however, do not appear to have led directly to the realization of any context-aware software or devices, except for perhaps one example, Cyc, which will discussed further in Section 2.3. The context-aware devices and applications of the 1990s mostly consisted of location-aware devices, and in our opinion, they do not require an elaborate formalism of context. Nonetheless, the work of McCarthy and others pioneers who offered formalisms of context are worthy of mention in the history of context awareness. In particular, we refer the interested reader to (McCarthy, 1993), (Guha, 1991), McCarthy and Buvač, 1998), (Akman and Surav, 1996), and (Buvač and Mason, 1993). In addition, (Brezillon, 1999) and (Akman, 2002) provide excellent reviews of context in artificial intelligence.

2.3 Theory

Some computer science researchers have attempted to formalize the concept of context in a mathematical sense, most notably John McCarthy. A formalization of context could be useful because computers are better at handling formal mathematical constructs compared to more loosely defined concepts. For example, a computer is quite capable of working with the set of integers {1, 2 , 3,...} or even the primary colors red, blue, yellow (e.g. defined by RGB values). On the other hand, an abstract concept like "at the store", while easily understood by a human, is not very useful on its own to a computer. This is not to say that it is *not* useful at all, but considerable effort must be made to define what is meant by such a construct and how to distinguish it from, e.g. "at the office", so that this construct can be utilized in a consistent manner. One powerful way to formalize context would be in the language of logic, e.g. predicate logic or propositional logic. Because logic has formed the basis for various programming languages (e.g. SQL, Prolog, etc.) it is reasonable to assume that, if context can be formalized in the mathematical language of logic, then computers programs can be written to process and "understand" context.

As mentioned above, computer scientist John McCarthy was one of the pioneers in developing a formalism of context. In a widely-cited paper published in 1987, McCarthy relates the concept of context to the problem of generality in artificial intelligence, which is to say that artificial intelligence programs suffer from a lack of generality. He notes that "[w]henever we write an axiom, a critic can say the axiom is true only in a certain context." He gives the example of the sentence "the book is on the table." and notes that a critic can "haggle about the precise meaning of 'on'" (e.g. if a paper sheet of paper is between the book and the table).

McCarthy proposes a formalization of context, combined with circumscription, where $holds(p, C)$ is an abbreviation for the sentence p being true in the context C. For example, "Watson is a doctor" is true in the context of Sherlock Holmes stories, but

"Watson is computer" is true in the context of IBM's artificial intelligence research. He incorporates generality through the relation $c1 \leq c2$, meaning that context $c2$ is more general than context $c1$. Alternatively, this can be understood as $c1$ is a specialization of $c2$ (Akman and Surav, 1996). McCarthy points out that there is no such thing as a "most general context".

In a paper published in 1993, McCarthy developed his formalization of context further. He changes the notation slightly (adopting the notation from [Guha, 1991]), where formulas are sentences of the form:

$$c' : \quad ist(c, p), \tag{2.1}$$

which asserts that proposition p is true in context c, which is itself asserted in the outer context c'. Thus, the above formula could be re-written as $ist(c', ist(c, p))$.

Continuing our example using this notation:

$$\text{“Sherlock Holmes stories”} : \quad ist(\text{“A Study in Scarlet”}, \text{“Watson is a doctor”}), \tag{2.2}$$

where "Sherlock Holmes stories" is the outer context and "A Study in Scarlet" (a specific novel) is the inner context.

He also argues that some contexts are *rich* objects, meaning that they can never be described completely but certain facts about them can be asserted, whereas others are *poor* and can be completely described.

McCarthy also introduces a term $value(c, term)$, where $term$ is a term[1], for example, $value(c, time)$ which can be used to represent the time in context c. He also introduces a number of different relations among contexts and also functions that output a context as a value. For example, *specialize-time*(t, c) represents the context related to c where the time "is specialized to have the value t" (McCarthy, 1993). Another way to understand this is *specialize-time*(t, c) contains all of the assumptions of c, plus the additional assumption that the time is t.

Another given example is $at(jmc, Stanford)$, so that $ist(c1, at(jmc, Stanford))$ can be used as an assertion that John McCarthy is at Stanford University, where $c1$ is a context in which jmc stands for John McCarthy and $Stanford$ stands for Stanford University and at is understood as "being regularly at a place, rather than momentarily at a place" (McCarthy, 1993). It is possible, however, that in another

[1]In formal logic, a *term* is "a variable, constant, or the result of acting on variables and constants by function symbols" (Weisstein, 2014).

context $c2$ at can take on another meaning (e.g. momentarily at a place).

Combining the predicates $ist(c,p)$, $specialize\text{-}time(t,c)$, and at, we can form relations such as:

$$\begin{aligned} c0: \quad & ist(specialize\text{-}time(t,c), at(jmc, Stanford)) \equiv \\ & ist(c, at\text{-}time(t, at(jmc, Stanford))), \end{aligned} \tag{2.3}$$

where the predicate $at\text{-}time(t,p)$ represents the assertion that the proposition p is true at time t.

An important concept emphasized by McCarthy, as well as others, [Guha] is that of *lifting relations*, also known as *lifting formulas* or *lifting formulas*. These specify the relation between different propositions and terms in different contexts. In other words, they allow one to “lift” information stated in one context into another context. (McCarthy and Buvač, 1997) give more precise definitions:

> Lifting axioms are axioms which relate the truth in one context to the truth in another context. Lifting is the process of inferring what is true in one context based on what is true in another context by the means of lifting axioms.

Thus, Equation 2.3 can be considered a lifting relation because it relates the specialized context $specialize\text{-}time(t,c)$ to the more general context c.

McCarthy never formalizing lifting in terms of a lifting operator, but he does introduce a general relation *specializes*, so that $specializes(c1, c2)$ when context $c2$ does not make any assumptions other than the ones made in context $c1$ and every meaningful proposition in context $c1$ can be translated. He also provides nonmonotonic relations to allow for inheritance of ist between a subcontext and supercontext:

$$specializes(c1, c2) \wedge \neg ab1(p, c1, c2) \wedge ist(c1, p) \supset ist(c2, p), \tag{2.4}$$

and

$$specializes(c1, c2) \wedge \neg ab2(p, c1, c2) \wedge ist(c2, p) \supset ist(c1, p). \tag{2.5}$$

The formalization of logic developed by McCarthy and others (including Guha, Buvac, and Mason) has come to be known as the Propositional Logic of Context (PLC). A full treatment of PLC, including its strengths and weaknesses, is beyond the scope of this thesis. The above is meant to serve as an introduction to familiarize the reader with PLC, as it is perhaps the most widely cited formalization of context available in the literature. Other formalizations exist, such as X, Y, and Z.

We now discuss some similarities and differences between the notion of context formalized in PLC, versus the notion of context described in the introduction to this chapter and used in the remainder of this thesis. PLC's notion of context is essentially the setting in which a set of propositions is true. Compared to the dictionary definition given above (i.e. "the interrelated conditions in which something exists or occurs"), there is clearly some overlap, however, there is, in our view, a clear difference. If one makes the proposition $COLOR(sky) = blue$, meaning "the color of the sky is blue", then this is true in the context of a cloud-free day in a particular location, which we might represent in PLC as $ist(c1, COLOR(sky) = blue)$, where $c1$ is a term representing the particular context specialized in time and place. In the notion of context given by the dictionary definition (and adopted in this thesis), however, the context *is* the setting itself, i.e. the time, place, color or the sky, and any other true parameter of the setting or environment. In agreement with PLC, this notion of context is usually (if not absolutely) a rich object, but otherwise these two notions of context seem quite distinct and incompatible. For example, if $c2$ is defined as $\textit{specialize-time}(t, c1)$ in PLC, then it would seem in $c1$ no notion of time even exists. Therefore, it is problematic to think of $c1$ as a context in the sense of the dictionary definition because one can always describe a setting or environment in terms of time (whether it be a discrete time instance or a time interval).

Furthermore, our criticism of PLC is that appears overly complex, at least if it were to be adopted for the purpose of making computers and devices "context aware". We can appreciate the motivation for PLC, that is, to achieving generality in representing knowledge, but this is not precisely the goal of a context aware program or application. Perhaps for these reasons, few practical implementations of PLC can be found in the literature. The one exception perhaps is Cyc, which is a knowledge database capable of "common sense reasoning". It is developed commercially by a company called Cycorp, but an open source version exists (OpenCyc) and a more complete version is available for research purposes (ResearchCyc). An evaluation of the suitability of Cyc for building context aware applications is, however, beyond the scope of this thesis.

We are not alone in making these criticisms of PLC. For example, (Hirst, 1997) argued persuasively that context cannot be defined independently from its use. Also, (Wolf and Bileschi, 2006) showed experimentally that for one particular application area (computer vision), contextual features had only a marginal benefit, compared to "appearance information". It may simply be the case that for some application areas, context plays an important role, whereas for others it is less important or even superfluous. Furthermore, different notions of context may be useful in some applications but less useful or even incoherent in other applications[2]. In our view, the formalism found in PLC reflects the authors' original motivation for developing such a formalism, which was to create a database of "common sense" that could be used by many different types of artificial intelligence programs (see McCarthy, 1969, and McCarthy, 1984).

2.4 How to Sense Context

So far we have (1) defined context and context awareness, (2) provided a working framework for contextual information, (3) discussed the history of this field, and (4) provided some theoretical background on the formalization of context. We have said very little so far about how computers actually sense context. Such is the intention of this section.

It is difficult to speak in general about how computers can sense context because, in addition to the points already made above, the different methods and hardware associated with context awareness vary greatly depending on the application. We have so far given examples of smartphone-based context awareness, but as mentioned in the introduction, this thesis will also deal with another very different task, that of determining the optimal path of a ship travelling through ice-covered waters. Obviously the type of contextual information relevant for this task, as well as the systems involved, are very different from that of smartphone-based context awareness. In our opinion, it is not very beneficial to generalize in great detail about systems and methods without keeping the ultimate application in mind. This returns to the problem of generality in AI, which McCarthy wrote about in 1987, and we again emphasize that this thesis makes no attempt to conquer this problem.

Nonetheless, there are a few general points that can be make about the systems and methods of context awareness from a general perspective. The first is that all types of information, regardless of the source, can be considered as input to a context awareness system. This can range from sensor data to data from an external database to an image acquired from a spacecraft hundreds of kilometers away from the target environment. Human input can also be considered as a source of contextual information, although in this thesis we mainly focus on contextual information that can be obtained without direct human involvement. For this reason, sensors will be one of the primary sources of input data, including both *in-situ sensors* (such as smartphone sensors) and *remote sensors*, such as an imaging sensor on an Earth

[2]It is not only humorous but also true to say that context depends heavily on the context.

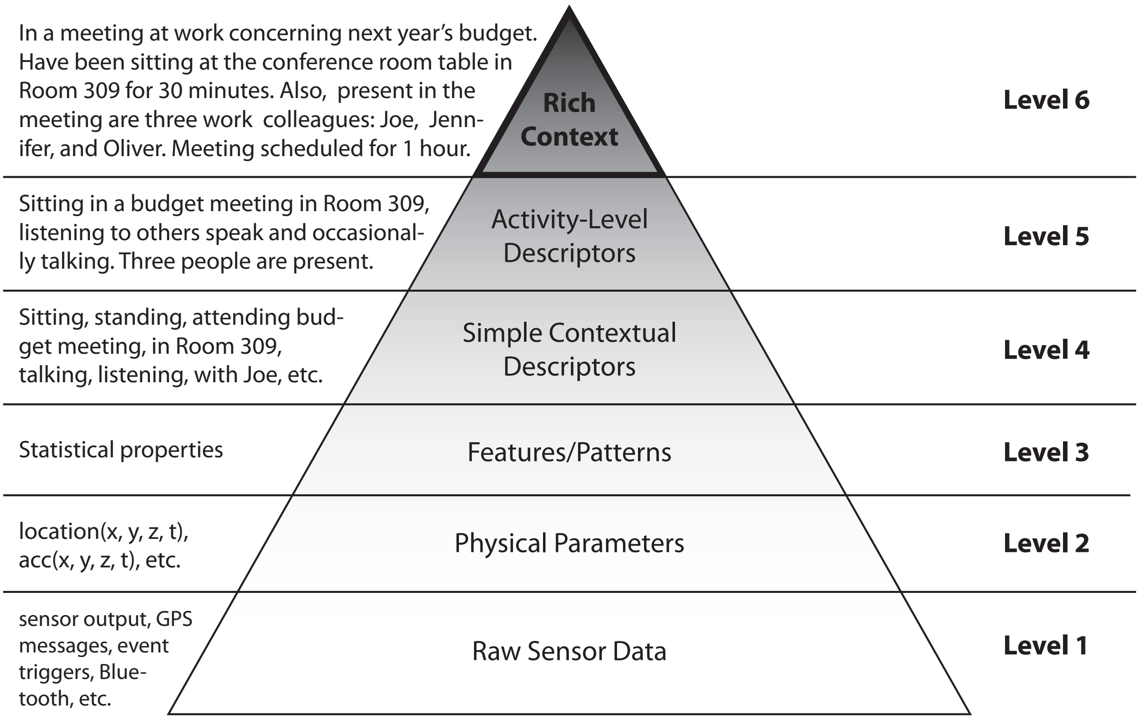

Figure 2.1 *The "context pyramid" shows the different levels of processing to build context-aware systems, starting from raw sensor data at the bottom and working up to "rich context" at the highest level.*

Observation spacecraft. Databases will also be an important source of contextual information, especially with regards to translating "raw data" (such as geographic coordinates) into a more meaningful form (such as a semantically-defined location, e.g. "at the supermarket"). Lastly, we can make some general statements about how to process the input data to a context-sensing system. We conceptualize this process as a "context pyramid", which was first presented in (Pei et al., 2013) and later in (Chen and Guinness, 2014). This context pyramid is also shown in Figure 2.1 below. On the left side is an example of contextual information. At the bottom of the pyramid lies the raw input to the context-sensing system, such as sensor data. The difference between the first and second level in the pyramid is that in Level 2, some "pre-processing" of the data may have been performed, such as reference frame transformation or filtering out noise. In the next level of processing, statistical features are extracted from the data, such as mean values or frequency domain features computed from time series data. The distinction between "pre-processing" and statistical feature extraction can be a bit blurry in some cases, but generally speaking Level 2 data usually have a clear physical meaning, whereas Level 3 data might have only a mathematical or statistical meaning. Next, Level 4 is achieved after the Level 3 data is subjected to a function or algorithm that performs contextual classification or in some cases regression. This topic will be covered in greater detail in Chapter 3. Level 4 data is in the form of *simple contextual descriptors*, which can be thought of as “atomic” elements of context. They each should belong to one of the seven categories described in Section 2.1 above. Then, Level 5 is achieved by combining multiple simple contextual descriptors into an activity-level description of the context, as well as the main pertinent contextual details. Finally, Level 6 combined all available contextual information into a *rich context*. The aim at this level is to approach a description of the context that is indistinguishable from human-written prose. As was the case between Levels 2 and 3, the difference between Levels 5 and 6 can be sometimes blurry. In some context-aware systems there may be fewer processing steps or perhaps more, so even the number of levels should not be taken as dogma. We believe, however, that the general process will always follow the overall trend illustrated in the context pyramid.

2.5 Conclusions

In this chapter we have defined the terms context and context awareness and provided a general introduction to the topic. We have attempted to discuss context awareness in an abstract and general way, but we have also pointed out the limitations of such a discussion. Our aim was to give a rough conceptual outline of this research topic, but our strong preference is that context should be discussed with some particular applications in mind, otherwise the subject is simply too broad to provide a general treatment. That being said, there are some general methods that can be employed towards many problems in context awareness, and such methods will be the focus of the next chapter.

3. MACHINE LEARNING

This chapter will provide background on the topic of machine learning, whose role in this thesis was outlined in the introduction chapter. Our preferred definition of "machine learning research" was also given in that chapter, but it is worth repeating here:

> Machine learning research seeks to develop computer systems that automatically improve their performance through experience (Mitchell et al., 1990).

Stated slightly differently, *machine learning* is concerned with developing and analyzing algorithms used by computer systems that automatically improve their performance through experience. An earlier definition, widely attributed to Arthur Samuel, is that machine learning is "the field of study that gives computers the ability to learn without being explicitly programmed"[1]. This definition also implies *automatic* learning, but it suffers from the problem that the meaning of "learn" is not precisely defined.

As is the case in some fields, the discipline known as "machine learning" has drifted somewhat from its original defining aims. This will become more evident later on in this chapter when we describe the major types of machine learning problems that have developed over the past 30+ years.

The chapter is organized as follows. Section 3.1 presents an intuitive example of machine learning in terms of a programming task. Section 3.2 provides an overview of the modern notion of machine learning. Section 3.3 describes supervised learning, and finally Section 3.4 describes unsupervised learning.

[1] We have been unable to recover the original source of this quote. Some references cite (Samuel, 1959), but the quote is not found in reprints of this article.

3.1 Computer Chess: An Example Learning Task

First, by way of example, we present a classic programming task that could possibly be achieved using machine learning. The task is to create a computer program that can play chess against a human[2]. If the task requirement were only to "play" chess, then the programming task would be rather trivial and would definitely not require machine learning. The programmer would simply have to encode all the rules of chess (namely how the pieces move and what happens as a result of the moves, such as captures) and then implement a random move selection function that obeys these rules. This program, undoubtedly, would be a very terrible chess player, so we add the additional requirement that the program should try to *beat* a human in chess. This is now a formidable programming task, and one that has been considered by computer scientists at least as far back as Claude Shannon's 1950 paper on the subject (Shannon, 1950). Perhaps the most famous chess-playing program was initiated at Carnegie Mellon University in 1985 and later transitioned to IBM, culminating in the computer Deep Blue® beating the chess master Garry Kasparov in a six-game match-up in 1997[3]. Nowadays, similarly advanced chess programs can run on a personal computer or even a smartphone or tablet. A full literature review of computer chess is beyond the scope of this thesis, but we refer the interested reader to (Hsu, 2002), (Spicer and Tashev, 2006), and (Russell and Norwig, 2010). Our intention in this section is to use this chess example as a way to clarify our understanding of the aim of machine learning.

3.1.1 Rule-Based Methods

Let us consider the ways in which we could go about this programming task. One way would be to extend the encoding of the rules of chess to encoding tactics and strategies of great chess masters, attempting to cover as many possible situations in chess, also known as *chess positions*, as we can[4]. Basically these strategy rules would instruct the computer what a chess master would do in each of the possible situations. This is essentially the approach that Deep Blue programmers took. Combined with the ability to evaluate hundreds of millions of chess positions per second, these rules eventually succeeded in beating the best human chess players.

[2] Some of the earliest works in machine learning and artificial intelligence involved computer-playing games, such as chess and checkers. See (Turing, ????, Shannon, 1950; Samuel, 1959)

[3] We note that Kasparov has accused IBM of cheating by letting human players intervene in one of the matches.

[4] We could have chosen the phrase "contexts in chess" rather than "situations in chess", thus emphasizing the conntection to context awareness, but this seems a bit awkward.

This approach, however, hardly fits the above definition of machine learning. In fact, to create Deep Blue required many years of highly "manual" work of programmers refining the set of strategy rules, testing the program against human players, and repeating this process. Thus, it falls short of the aim of *automatically* improving performance.

3.1.2 A Trivial Chess Learning Program

A trivial *machine learning chess program* could work as follows: The program would simply play against itself and then at the end of the game record which side won (or whether there was a draw), as well as the set of moves in the game. Let us call one side *ComputerWhite* and the other *ComputerBlack*, according to convention. As the games are played, *ComputerWhite* would first look-up whether the current position matches any of the records in the database and if the record corresponds to a win for *ComputerWhite*, then it would play the next move listed in the record. Otherwise, it would choose a valid move at random. *ComputerBlack* would do exactly the same, and the program would continue until the game's conclusion. Another very similar version of the program would operate in the same way but allow for the other side to be played by a human. This program would not be very good when the size of the database is small, and furthermore it would learn very slowly because the records in the database would be generated by random sampling (without replacement) from the set of all possible chess games. Nonetheless, it would fit our definition of machine learning, provided that it does not require a human programmer to update the program during or after the learning process.

Such approaches are known as brute force dictionary approaches or *rote learning*[5]. (Shannon, 1950) calculated that such a database (or dictionary) would require roughly 10^{43} entries. Clearly even a very efficient database containing this dictionary would require many "Google-scale" data warehouses, and performing look-ups in the database would be prohibitively expensive.

It is probably fair to say that for most computing tasks, writing a trivial machine learning algorithm is fairly easy, as in the example given above. Therefore, what the discipline of machine learning is mainly concerned about is improving the learning rate and performance rate beyond that of a trivial algorithm, or ideally beyond any state-of-the-art algorithm. There are, of course, other important aspects as well, such as finding algorithms that use minimal amounts of memory or those that can

[5] Note this type of learning is similar to the so-called "dictionary attack" used to "learn" a log-in password.

produce simple/efficient models of the learned task. Learning rate and performance, however, are usually the most important aspects of a machine learning algorithm[6]. Lastly, it is important to note that there is no “one size fits all” machine learning algorithm. Different algorithms perform better or worse relative to their peers on different problems and learning tasks. This phenomenon has been called the “no free lunch theorem”, and it has been shown to have a strong mathematical basis (Wolpert, 1996).

3.2 Modern Machine Learning

In this section, we describe the modern notion of machine learning, which, as we have already alluded to, has developed into something a bit different from what the early pioneers in machine learning had envisioned. That is, today there is a well-established community of machine learning researchers and practitioners whose focus is not entirely the same as what Mitchell, Samuel, or other machine learning pioneers had in mind. Our intention in pointing this out is not to denigrate the discipline of machine learning as a whole but rather to emphasize those aspects of the discipline which fall short of the original goals of machine learning.

Let us first present a few other definitions of machine learning found in recent textbooks on the subject:

Alpaydin: “Machine learning is programming computers to optimize a performance criterion using example data or past experience.” Bishop:

This definition appears quite close to that of (Mitchell et al., 1990), if we assume that “example data” can be generated automatically. This may be true in some cases, but in most methods described in (Alpaydin, 2010) the example data are data that have been manually labeled with the “correct” value relative to the performance criterion that is to be optimized. Although programs using such methods can improve their performance by obtaining more example data, if the example data cannot be generated automatically, then the method would fall short of (Mitchell et al., 1990).

Another recent definition is given by (Murphy, 2012):

Murphy: “a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of

[6] Note that performance can be measured in different ways, a topic which we will return to in Section 3.3. In the chess example given above, a good performance metric might be the percentage of games that the computer wins against a randomly selected human player or perhaps the average Federation Internationale des Echecs (FIDE) rating of the human players it has beaten.

decision making under uncertainty (such as planning how to collect more data!)."

This definition includes the "automatic" aspect, similar to (Mitchell et al., 1990)

This group of methods is known as *supervised learning*, and it will be discussed further in Section 3.3. Similarly, many of the other methods described in (Alpaydin, 2010) are focused on discovering structure in a set of data. The "learning task" is to find the structure in a given set of data, but this structure is inherently dependent on the dataset itself, so this task is not a machine learning task in the sense of (Mitchell et al., 1990). Such methods fall under the category known as *unsupervised learning*, which will be discussed in greater detail in Section 3.4.

Mitchell also provides a precise definition of the concept of learning in the context of machine learning:

> A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Many learning tasks can be expressed in terms of learning a mathematical function between the inputs to the task and the desired outputs. In other words, the learning task is to find some optimal mapping between the inputs and the possible outputs. This can be expressed as follows:

$$f : \mathbf{x} \rightarrow y \tag{3.1}$$

where f is a function, $\mathbf{x}$ is either a vector of inputs of arbitrary dimension, and $y \in Y = \{y_1, y_2, ...y_m\}$, corresponding to the set of all possible outputs (which may or may not be finite). The function f is also called a model in many textbooks on machine learning. A simple physical example function or model would be a spring scale that maps a displacement length to a weight. In this example, we know from Hooke's law that the function is linear and given by the spring constant k, but in many unsolved problems the form of the function is unknown, as well as its parameters. In some cases, the learning task may be framed in probabilistic terms, where the output expresses the conditional probability $p(y|\mathbf{x})$. This distribution, $p(y|\mathbf{x})$, may be intrinsically important to the application at hand, or it may be an intermediate step towards determining the most likely value of y according to:

$$y = \arg\max_{y \in Y} p(y|\mathbf{x}) \tag{3.2}$$

Machine learning techniques differ mainly in how they express and learn this unknown function $f(\mathbf{x})$ and also the form in which y (and therefore the set Y) are expressed. For example, Y may be a continuous range or a finite, discrete set. When the task involves a continuous-valued output value, it is called *regression*. The spring example given above would be a regression problem. In some cases, the desired output may also be a vector of different values, representing different physical quantities (e.g. the height and weight of a person). When the output valuable is discrete, we call it *classification*, since the possible values generally represent different *classes* or categories. One example would be the problem of determining whether a person is male or female based on voice recordings. This example also highlights the fact that for some machine learning problems, it may be well accepted that the problem cannot be solved perfectly. Such problems may be best represented in probabilistic form.

Apart from the distinction between regression and classification, there are two main categories of machine learning techniques, based on how the unknown function f is learned or approximated. The first category is known as *supervised learning*. In supervised learning, a "trainer" supervises the learning process. The goal is essentially then to transfer the knowledge of the supervisor in the form of a mathematical or computerized model. More details on supervised learning will be covered in Section 3.3. The other main category is known as *unsupervised learning*. In unsupervised learning, the learning process is not guided by any significant way. The goal is essentially to uncover patterns that are implicit in the data but unobvious. More details on unsupervised learning will be covered in Section 3.4.

From this point forward in this thesis, we will drop the distinction between the modern popular definition of machine learning and the earlier meaning of (Mitchell et al., 1990) and focus mainly on the mainstream meaning and the main techniques from machine learning. Whenever necessary, we will use the term *automatic learning* to refer to the fact that improved performance takes place without requiring any human intervention.

3.3 Supervised Learning

As stated above, supervised learning uses a "trainer" to supervise the learning process. In most cases, the trainer has encoded his or her knowledge in the form of *labeled data*, also known as *training data* or a *training set*. In terms of the function f expressed above, the training consists of input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is an input of arbitrary dimension, y_i is a "labeled" output, and N is the number of training samples, such that $\mathcal{D}$ provides *examples* of values of the function

$y = f(\mathbf{x})$. In simple terms, the training data provide examples of input data that are *labeled* with the correct or desired output.

It is usually the case that the training data does not exhaustively define the unknown function f. If, however, certain assumptions can be made about the function, then the function might be fully specified by a finite set of training data. In the simplest case, where f is linear and $\mathbf{x}$ is one-dimensional, then only two training samples are needed to specify the relationship between $\mathbf{x}$ and y[7]. Most practical examples of machine learning algorithms are more complicated due to (1) higher dimensionality, (2) non-linearity, and (3) error present in the training data.

Let us consider a simple example from the domain of context awareness. Suppose we have a smartphone application that needs to know whether the user is walking, running, or standing still (i.e. static). We refer to these as *mobility contexts*. The smartphone has a GPS receiver that can record the user's position and speed, and it also has a three-axis accelerometer that can measure acceleration. In order to keep things simple, instead of using the raw accelerometer signal, we define a feature from the accelerometer data, known as *dynamic acceleration*:

$$a_d = var(\{\sqrt{a_{xi}^2 + a_{yi}^2 + a_{zi}^2}\}_{i=1}^{N}) \tag{3.3}$$

where $var(\cdot)$ is an operator that computes the variance over some time series of data (e.g. one second of acceleration data); a_{xi}, a_{yi}, a_{zi} are the accelerations in the x, y, and z directions, respectively, for some given time epoch i; and N is the number of samples in the time series.

A user, Mary, has painstakingly collected a dataset for developing a context-aware application and labeled whether she was walking, running, or standing still. The data are shown in Table 3.1 below, consisting of two dimensions of input data and the labeled output. In order to keep the size reasonable, only 35 data samples are shown in the table. In Figure 3.1, similar data are plotted, but now we include 1000 samples from each class.

Based on this figure, several observations can be made. We clearly see three clusters of data, corresponding to the three mobility contexts. The cluster corresponding to the "static" context is well separated from the other two, but in the case of the "walking" and "running" contexts, there is some overlap. Another important observation is that in the "walking" data, some of the values for speed are very close to 0 m/s. This could be due to errors in the data (i.e. the data from the

[7] Remember that two points define a straight line.

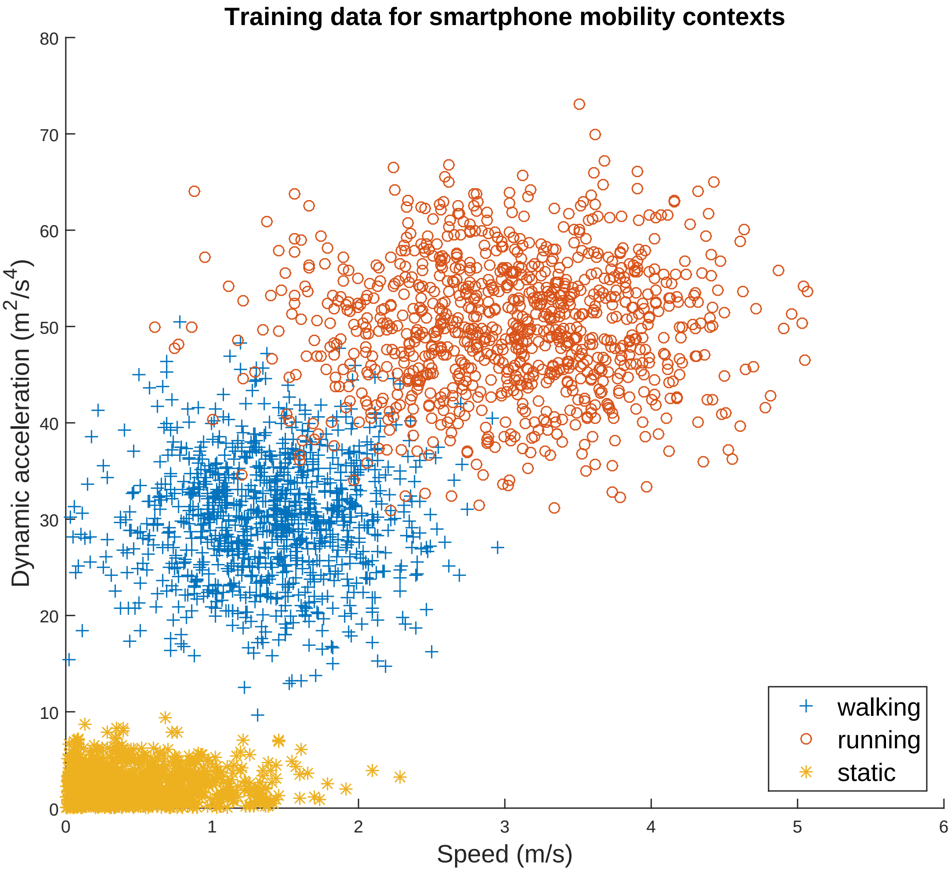

Figure 3.1 *This is a caption.*

GPS receiver might have some error) or labeling errors made by Mary. It is very common with this type of data that some labeling errors are present in the training set. For example, at the transition points between the walking and static contexts, it is difficult to accurately label which data corresponds to "walking" and which corresponds to "static"[8].

In this example, the goal of supervised learning would be to find a function f that maps the input data $(speed^i, a_d^i)$ to the correct output class, i.e. $\{$'*walking*', '*running*', '*static*'$\}$, such that the number of errors are minimized. In this context, errors could be defined as input data that are mapped to the wrong output class, also known as misclassifications.

Three important interrelated concepts should now be introduced: *generalization*, *underfitting*, and *overfitting*. Generalization refers to the idea that supervised learn-

[8]One technique to avoid such labeling errors is to remove these transition points entirely from the training set.

ID	Speed (m/s)	Dyn. accel. (m^2/s^4)	Label
1	2.56	21.10	walking
2	0.94	28.78	walking
3	1.24	31.22	walking
4	2.99	36.66	walking
5	1.24	36.43	walking
6	0.64	29.88	walking
7	0.73	34.13	walking
8	1.68	28.56	walking
9	2.72	32.96	walking
10	1.82	38.57	walking
11	2.10	30.70	walking
12	2.80	49.59	running
13	4.01	47.41	running
14	3.10	61.96	running
15	1.98	54.44	running
16	2.33	53.92	running
17	5.48	44.49	running
18	4.14	52.38	running
19	2.69	52.85	running
20	4.73	44.02	running
21	1.22	48.76	running
22	4.88	47.78	running
23	0.40	2.89	static
24	0.92	0.92	static
25	0.36	1.48	static
26	1.16	3.37	static
27	0.00	5.76	static
28	0.28	3.27	static
29	0.60	0.70	static
30	0.45	2.97	static
31	1.44	1.79	static
32	0.11	1.45	static
33	1.36	1.51	static
34	1.06	0.03	static
35	0.81	1.28	static

Table 3.1 *The caption of the table*

ing should "generalize" beyond the specific examples given in the training set. In other words, the goal is not simply to map the inputs to the outputs for the given training set but rather to find a mapping function that works well on some yet unseen data. If the goal were simply to fit a function to the training set, then it would be trivial to write a function that performs with zero errors (e.g. a simple lookup table would do the job).

Overfitting refers to the situation where the supervised learning algorithm has produced a mapping function that follows the training data in too much detail. Keep in mind that every training set is somehow incomplete and imperfect. If the training results in a function that does not properly take into account the gaps and the noise in the training data, then it will *overfit* the training data and will not generalize well. It is also possible that a mapping function *underfits* the training data. This usually means that the mapping function is overly simple, for example, using a linear model for data that is inherently non-linear. Therefore, good generalization lies in between the two extremes of underfitting and overfitting. The goal of learning is more precisely defined as minimizing the *generalization error*, which is the average error rate that will be produced by any future data, and this means finding a model that neither overfits nor underfits. Of course, it is difficult to estimate the true generalization error. The most common approach is to use an independent *test set*. A test set is simply another labeled dataset that is not used in the learning process but is reserved for measuring the generalization error after learning has already taken place.

A test set provides a way to measure the generalization error and see whether any overfitting or underfitting is occurring, but the question remains: How does one determine the right type of function or model to fit to the training data? This process is known as *model selection*. In model selection, we siphon off yet another portion from the training set, known as the *validation set*. The model selection process then proceeds as follows:

1. Choose a hypothesis set $\mathcal{H}$ containing different hypothesis function types to be used in model selection. This hypothesis set can be of one particular function class, such as the set of all linear functions or can be of several different classes. The goal is to include within the hypothesis set a class of functions that match well with the underlying data under investigation. This is , however, nontrivial and may require some precursory *data exploration*.

2. Given the hypothesis set $\mathcal{H}$, for each hypothesis class $\mathcal{H}_i \in \mathcal{H}$, use the training set $\mathcal{D}$ to find the best function $h_i \in \mathcal{H}_i$. For example, if $\mathcal{H}_i$ is the set of all

linear functions of the form $h(x) = a * x + b$, then this step is equivalent to finding the parameters a and b that best match the training data, according to some linear regression estimator, e.g. the least squares estimator.

3. Now we have a set of fitted functions $h_i \in \mathcal{H}$, and the next step is to choose the best one. For this, we use the validation set to measure the error rate and choose the function with the lowest error, which we denote h_{best}, and its hypothesis class is denoted by $\mathcal{H}_{best}$.

4. Finally, fit a new function $h_i \in \mathcal{H}_{best}$ using the training set plus the validation set, and measure its error using the test set. Since the test set was not used in the learning process, the resulting error rate can be considered an estimate of the generalization error.

Depending on the amount of labeled data available, and the complexity of the underlying structure in the data, it may be necessary to repeat this process with different divisions of the labeled data into the respective training set, validation set, and set. The standard technique for this repetition process is known as *cross-validation*. Due to space limitations, we will not cover cross-validation in detail, but it was employed in [P3] and [P4].

3.4 Unsupervised Learning

Unsupervised learning is, in many ways, quite similar to supervised learning, except that there are no labeled data. In other words, there are only input data, and the goal is to learn something about the structure or patterns in the input data. In this way, unsupervised learning is very similar to traditional statistical methods, where the goal is to infer a statistical model from a set of data. Many unsupervised learning methods, such as density estimation, come straight from statistics. Others differ only in the name or some other superficial characteristics. Especially in recent years, there are large overlaps between statistics research and unsupervised learning research[9].

Consider again the data presented in Table 3.1 and Figure 3.1. Suppose Mary had not gone to the trouble of labeling the data with the actual mobility context associated with each data sample. We would have then only a two-dimensional dataset of input data, and we could make a similar plot as Figure 3.1, except the legend would be missing and we would also not have the information necessary to

[9] This is also true to a certain extent in supervised learning, but the similarity is more striking in unsupervised learning.

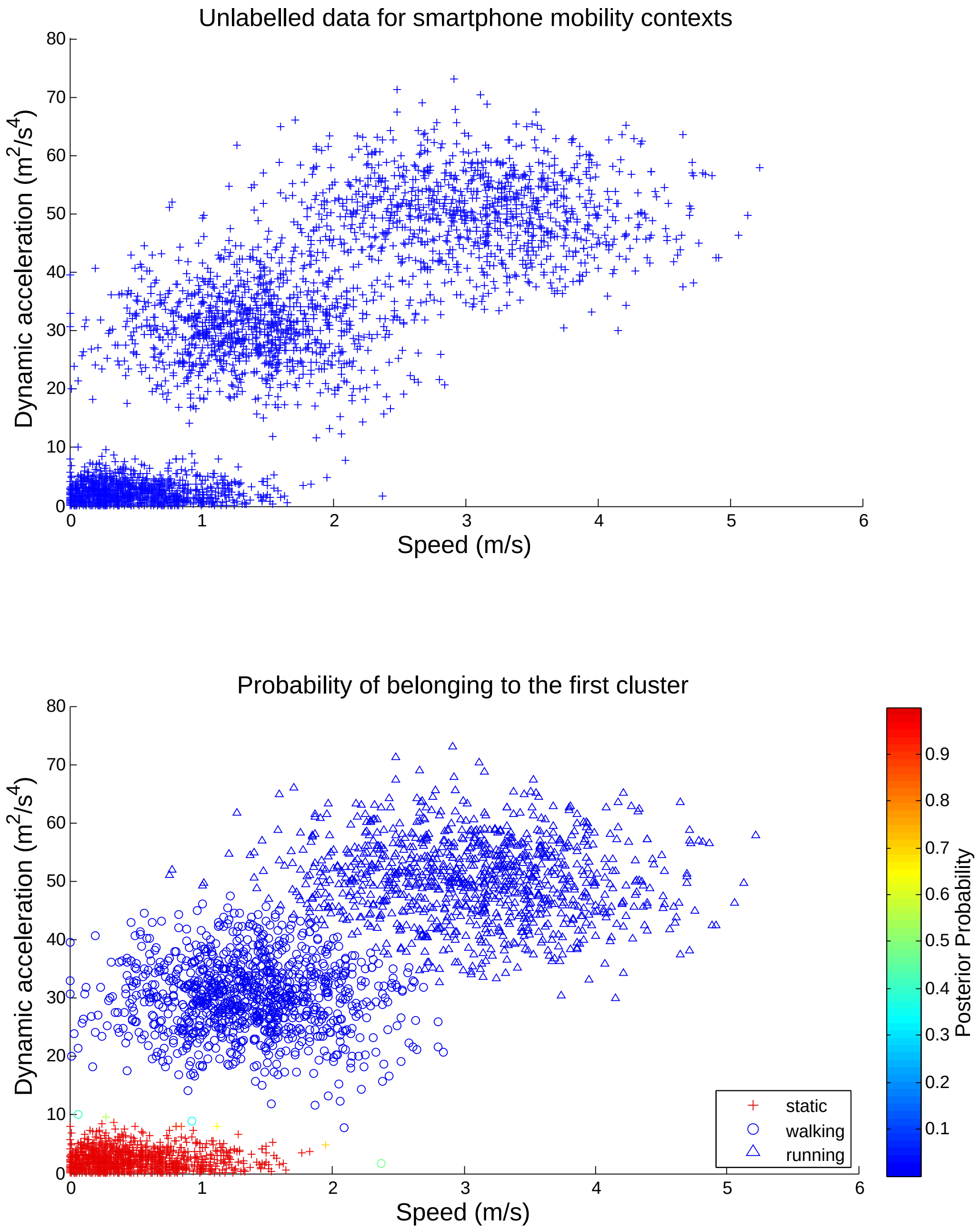

Figure 3.2 *This is another caption.*

label the samples with different colors as in Figure 3.1. The top part of Figure 3.2 shows such a plot.

One unsupervised learning task would be identify different clusters or groups present in the data. Depending on the data and the application, it may or may not be apparent how many clusters are inherently present in the data, so the number of clusters may also be a parameter to determine as part of the unsupervised learning task. As is the case in supervised learning, there are a plethora of different unsupervised learning algorithms available in the literature that perform clustering. Possibilities include k-means clustering (Hartigan and Wong, 1979), OPTICS (Ankherst et al., 1999), and the expectation-maximization (EM) algorithm (Dempster et al., 1977). In particular, the EM algorithm has its roots in statistitcs and can fit observed data to an arbitrary statistical model.

To provide an example of clustering, we used the EM algorithm to fit a Gaussian mixture model (GMM) to the data that we have previously seen in the top half of Figure 3.2. A GMM is of the form:

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{K} \pi_k \phi_k(\mathbf{x}; \boldsymbol{\theta}_k) \tag{3.4}$$

where $\mathbf{x}$ is a random vector, K is the number of components in the mixture model, $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$ are normal distributions with parameters $\boldsymbol{\theta}_k = (\boldsymbol{\mu_k}, \mathbf{\Sigma_k})$, π_k are mixing weights satisfying $\pi_1 + ... + \pi_K = 1, \pi_k \geq 0$, and $\Theta = \{\pi_1, ..., \pi_K, \theta_1, ..., \theta_K\}$ is the complete set of model parameters[10].

The EM algorithm itself is a widely-used iterative algorithm used to find the maximum likelihood estimate (MLE) of the model parameters (which we denote with Θ as above) for an underlying distribution $p(\mathbf{x}|\Theta)$ used to model a given dataset, which we denote as $\mathcal{D} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ (Bilmes, 1998). The MLE is obtained by maximizing a function Q equal to the expected value of the loglikelihood $\mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})$, given the observed data $\mathcal{D}$ and the current parameter estimates $\Theta^{(i-1)}$:

$$Q(\Theta, \Theta^{(i-1)}) = E[\log \mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})|\mathcal{D}, \Theta^{(i-1)})] = E[\log p(\mathcal{D}, \mathcal{Y}|\Theta)|\mathcal{D}, \Theta^{(i-1)})] \tag{3.5}$$

where $\mathcal{Y} = (y_1, ..., y_N)$ is a vector of latent variables that indicate to which component of the GMM a given data sample $\mathbf{x}_j$ belongs. The latent variables can be expressed in various ways, but perhaps the simplest expression is that $y_j = k$ when $\mathbf{x}_j$ belongs to component k. In the above equation i indexes the current iteration interval of the algorithm, so $\Theta^{(i-1)}$ represents the parameter estimate from the pre-

[10] The notation used for the GMM is similar but not identical to that given in (Bilmes, 1998).

vious iteration (or the initial estimate, if $i = 1$).

Before applying the EM algorithm to find the parameters Θ of a GMM, one must decide on the number of components K to incorporate into the GMM. As we shall see, each component k in the model will correspond to a cluster in the final clustering result; thus, this step is, in practice, the same as determining the number of clusters, and we can consider K to be a hyperparameter in the estimation problem.

Various methods can be used to determine the best value for K. For low-dimensional data, a practical method is to simply plot the data (as we did in the top half of Figure 3.2) and try to visualize the inherent number of clusters. For high-dimensional data ($D > 3$), this simple approach is not necessarily adequate, nor does it support the goal of automation described earlier. Therefore, a more sophisticated, systematic approach may be preferred, such as the one described in (Vlassis and Likas, 2002). For this example, we assume in the interest of space that the choice of K is already clear, and for these data $K = 3$ seems to be a reasonable choice.

The next step is simply to apply the EM algorithm to determine the parameters Θ of our three-component GMM. A detailed description of the EM algorithm is beyond the scope of this thesis, but here we provide a brief overview.

First, EM requires an initial estimate of Θ, and various initialization techniques to provide sensible initial estimates can be found in the literature . A simple approach is to use the given dataset $\mathcal{D}$: e.g. select K random samples to initialize $\boldsymbol{\mu_k}$ and use the covariance matrix of $\mathcal{D}$ for each of the initial K covariance matrices Σ_k (Smyth, 2015).

After initialization, the algorithm then alternates between computing an expectation function (known as the E-step) and finding the parameters Θ that maximize this function (known as the M-step). At each E-step, the algorithm calculates a new $Q(\Theta, \Theta^{(i-1)})$. In the M-step, an updated estimate $\Theta^{(i)}$ of the parameter set is obtained by maximizing $Q(\Theta, \Theta^{(i-1)})$, according to:

$$\Theta^{(i)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \tag{3.6}$$

The algorithm terminates when $Q(\Theta, \Theta^{(i-1)})$, evaluated at $\Theta = \Theta^{(i)}$, converges towards a maximum value (i.e. improvement is below some threshold value ϵ).

Finally, once the parameters Θ are estimated, we can determine the posterior probability that a data sample $\mathbf{x}_j$ belongs to a particular component k of the GMM, according to its so-called "membership weight" (Smyth, 2015):

$$w_j^k = p(y_j = k|\mathbf{x}_j, \Theta) = \frac{p_k(\mathbf{x}_j|\theta_k)\pi_k}{\sum_{m=1}^{K} p_m(\mathbf{x}_j|\theta_m)\pi_m} \tag{3.7}$$

Recall that each component of the GMM corresponds to a cluster, and therefore the membership weight for a given k is the posterior probability that the data sample belongs to cluster k. The bottom half of Figure 3.2 and Figure 3.3 show the posterior probabilities for our example data, corresponding to membership in each of the three clusters. Note that a dividing line between membership in each cluster can be drawn where the posterior probability reaches 0.5.

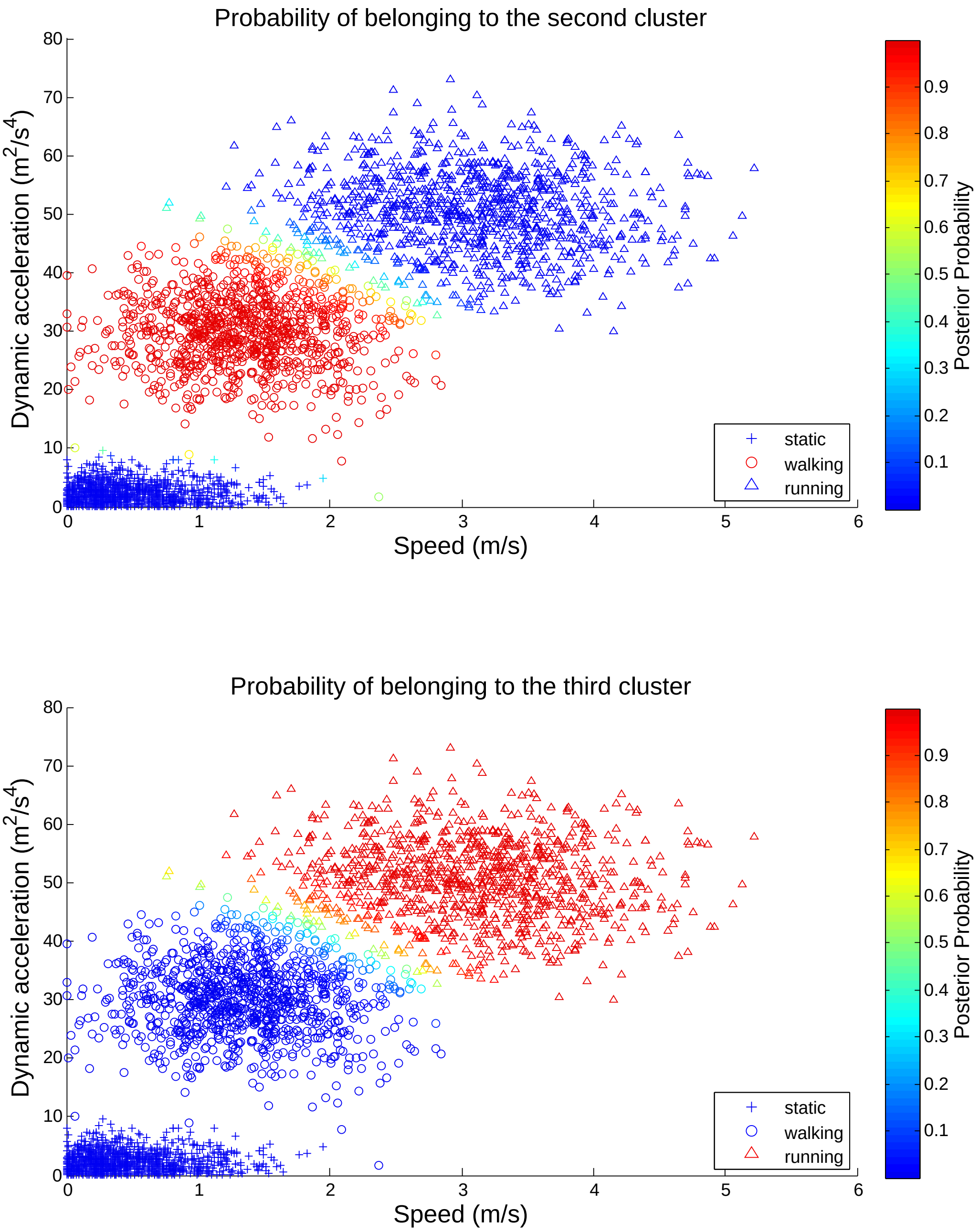

Figure 3.3 *This is another caption.*

4. OVERVIEW OF PUBLICATIONS

This chapter provides an introduction and overview to the five publications included in this compendium thesis. As mentioned above, two of the publications, [P1] and [P2], are excerpted from the book *Geospatial Computing in Mobile Devices*, published by Artech House in 2014. Two others, [P3] and [P4], were published in the peer-reviewed open access journal *Sensors*. The final publication [P5] was published in the Proceedings of the Position Location and Navigation Symposium, jointly organized by the Aerospace and Electronics Systems Society (AESS) and the Institute of Navigation (ION). AESS is an affiliate society of the Institute of Electrical and Electronics Engineers (IEEE).

The remainder of this chapter is organized as follows: Section 4.1 describes the overall research goals and research setting under which the publications were prepared. Section 4.2 maps the included publications to the overall research goals presented in this thesis. Section 4.3 describes the author's contributions to each publication. Finally, Section 4.4 outlines the scientific impact of the included publications.

4.1 Research Context for Publications

The overall research goal of this thesis was stated in Chapter 1: to improve our understanding of how computing devices can better understand us and our needs. Also, the detailed research objectives were summarized in Section 1.2 in terms of three specific tasks. Nonetheless, in this section, we aim to describe the broader research context (no pun intended) in which the five included publications were prepared and written, in order to better frame the publications and understand their interrelationships.

Firstly, a significant focus of our overall research portfolio during the years 2011–2013 was making the transition from growing ubiquitous positioning capabilities in mobile devices to emerging context awareness capabilities. This research focus was especially evident in the project "Indoor Outdoor Seamless Navigation for Sensing Human Behavior" (INOSENSE), funded by the Academy of Finland. The goal of this project was "to carry out research on sensing social context, modeling human behavior and developing a new mobile architecture for social applications" (FGI website). It is primarily within this project that the research described in publications [P3] and [P4] was prepared.

During the INOSENSE project, we decided to initiate an ambitious book project. The book, *Geospatial Computing in Mobile Devices*, written jointly by Prof. Ruizhi Chen and the author of this thesis, was prepared to present our ongoing research work and other efforts in the field of mobile geospatial computing. The intended audience was not only the scientific community but also potential engineers and developers who may be interested in putting the research results into practice. In particular, the two chapters included in this thesis cover the topic of context awareness, a key topic in this thesis. Preparing these book chapters also provided an excellent opportunity to carry out a detailed literature review on the topic of context awareness. Overall, the book was prepared between May 2012 and March 2014 during the author's spare time, alongside INOSENSE and other project responsibilities.

One of overall goals of covering context awareness in "textbook format" was to help raise the profile of context awareness within the mobile computing and geospatial sciences communities. We are aware of no other book in the field of mobile geospatial computing that covers also context awareness, and there are very few books overall that cover context awareness to any significant depth. We believe that the growing capabilities of smart mobile devices, such as the smartphone, present a potentially breakthrough opportunity for context awareness to become an important topic in these research communities.

In August 2013, the author transitioned to a new project with rather different research goals. The name of the project was "Arctic Real-Time Satellite Services for the Public and Commercial End-Use" or ARCSAT for short. Although the project included many different research topics, our research goal in the project was to carry out a feasibility study on the development of an "ice-aware" navigation solution for Arctic seas. It was stated in the project plan that "...awareness of the sea ice and sea weather information will improve the safety of the navigation solution especially in Arctic areas." Although the application area is clearly very different from that of the INOSENSE project, the concept of context awareness was evidently relevant. One other difference between the two projects is that in INOSENSE, the main focus was on *generating* context awareness, whereas applications of context awareness were largely out of scope. By contrast, in ARCSAT the focus was more on utilizing context awareness—in particular ice awareness—for navigation purposes. The main output of the project was publication [P5]. In addition, this project bore fruit in terms of two related research projects which were initiated in 2015.

In about March 2014, the author again transitioned to a new project, called ESA-BALT, which continued our line of research related to maritime navigation. The research results of this project, however, are outside the scope of this thesis and will not be discussed further. These project transitions are the main reason for the gap of more than two years between publications [P3] and [P4], despite the fact that they cover related topics.

4.2 Mapping of Publications to Research Areas

Figure 4.1 presents a mapping between the included publications [P1]-[P5] and different areas of research within the topic of context awareness. These areas can be divided into three broad areas: (1) background and literature review, (2) concepts and theory, and (3) different use case scenarios. Publications [P1] and [P2] fall primarily within the first and second areas, respectively. Publications [P3]-[P5], although containing some elements from the first two areas, mainly deal with different use case scenarios where context awareness can be applied.

As the Figure 4.1 indicates, there are many potential use case scenarios that are not addressed in this thesis. There are, of course, many possible ways to categorize and sub-categorize the use cases, and this chart is not intended to be authoritative in this matter. The list is also not exhaustive. Our original research plan was to cover as many different use case scenarios as possible, but due to time constraints and project limitations, only two separate use case scenarios could be investigated (or three, if you consider [P3] and [P4] to cover separate use case scenarios). Our plans to cover additional use case scenarios will be discussed briefly in Chapter 5.

4.3 Author's Contributions to the Publications

This section outlines the main contributions of the author of this thesis to the included publications.

[P1]: The thesis author was the main author of this chapter, whereas the book co-author provided only editorial comments to a near-final draft. The thesis author conducted all the necessary background literature review and independently decided on the details contents of the chapter. The author also came up with the idea to use Hermagoras's "seven elements of circumstance" to organize and describe the different elements of context. The author also independently identified and collated

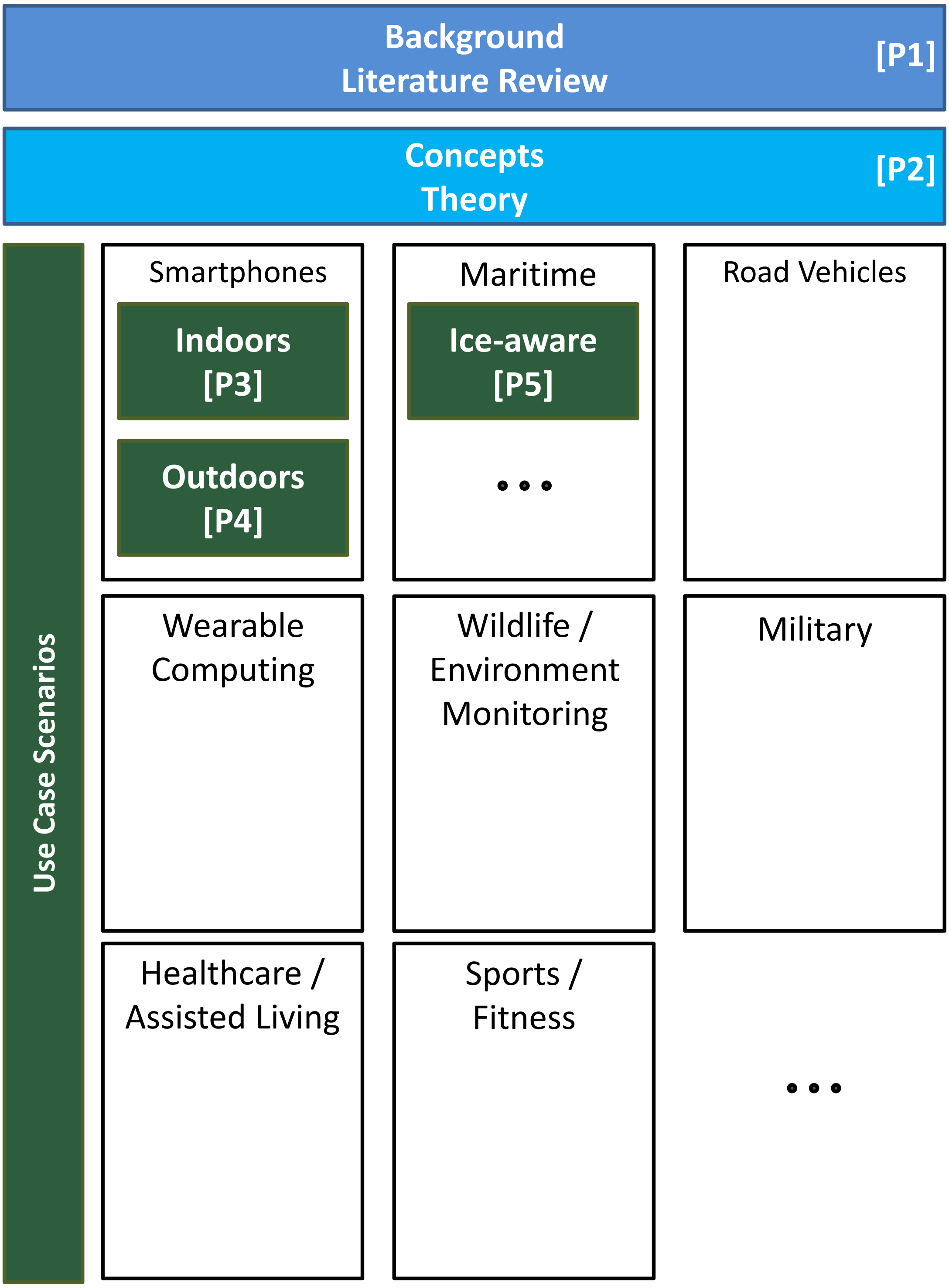

Figure 4.1 This is another caption.

the various Android Application Programming Interfaces (APIs) relevant to context awareness.

[P2]: The thesis author was the main author of this chapter, whereas the book co-author provided only editorial comments to a near-final draft. Similar to [P1] thesis author conducted all the necessary background literature review and independently decided on the detailed contents of the chapter. The author created all of the figures in the chapter and originated the concept of the "context pyramid". Finally, one of the main contributions of this chapter is the formulation and description (including figures) of rather somewhat complex mathematical concepts (e.g. Support Vector Machines) in such a way that they are understandable to anyone with basic knowledge in algebra and probability.

[P3]: The thesis author was the second author of this publication, but the first author has affirmed in writing that the thesis author's contribution was roughly equal to that of the thesis author. The thesis author contributed equally to the design and implementation of the experiments, including implementation of the data collection application on a smartphone. He assisted in the analysis of the test results. Also, he was the originator of the idea used for indoor-outdoor detection based on GPS signal-to-noise ratio and WiFi signal strength and implemented this method in a smartphone. The thesis author contributed to the development of the WiFi fingerprinting indoor positioning system and prepared the test environment by measuring and marking reference points and setting up additional WiFi access points. As stated earlier, he was the originator of the "context pyramid" concept, which is also described and employed in this paper. He was also the originator of the idea of using a graph-based grouping of the reference points. Lastly, he contributed to the preparation of the manuscript, including writing some sections and editing it in its entirety.

[P4]: The thesis author was the sole author of this publication and received assistance only in the data collection part, as well as general guidance form his supervisors. He also implemented all the necessary software used in the experiments, apart from the Weka software platform used in the data analysis. Some extensions to Weka, in terms of automating analysis and integrating Weka with Matlab, were also implemented by the author.

[P5]: The thesis author was the first author of this publication and was the originator of the idea of using a graph-based approach and the A* algorithm for ice-aware route optimization. He implemented the optimization algorithm in Matlab, basing the implementation only roughly on an open source implementation. He also designed the graph structure, in terms of optimizing the discretization of the ship's motion model. The only parts of the implementation that were not done by the thesis author were the implementation of the resistive ship speed model, representation of the ice data, and provision of the historical ship data, all of which were provided by co-authors. Finally, the thesis author led the manuscript preparation, writing most sections, preparing all figures, and editing the manuscript in its entirety.

4.4 Scientific Impact of the Publications

Perhaps the most straightforward approach to measuring the scientific impact of publications is via citations in other scientific works. All of the included publications have received multiple citations, and Table **??** shows these statistics. Data is provided by the Google Scholar service and may not be complete. They have been manually checked for accuracy. In the case of [P4], this statistic includes, in fact, citations from two related publications. [P4] was first published in shorter form in 2013 as a conference paper (Guinness, 2013), and it was only recently (April, 2015) published as a journal article. Thus, it seems fitting that citations from (Guinness, 2013) are included as well. We note also that (Guinness, 2013) received the Student Paper Award of the conference where it was presented.

In addition to citations, the published work has had the result that the author has been invited to participate in the peer review process for several scientific articles on related topics. He has also been invited to serve on a scientific panel at the 2015 ION GNSS+ conference.

Several of the included publications appear in scientific forums that have been rated by the Finnish Publication Forum (in Finnish: Julkaisufoorumi, abbreviated JuFo). At the time of their publication, these forums had the following ratings:

[P1]: JuFo Level 1
[P2]: JuFo Level 1
[P3]: JuFo Level 1
[P4]: JuFo Level 1
[P5]: Not rated by JuFo

Lastly, [P5] was nominated by a peer reviewer for the Walter R. Fried Memorial Award for best paper of the conference.

5. CONCLUSIONS

This chapter offers some conclusions based on this thesis, including the attached publications. It is organized as follows. Section 5.1 briefly summarizes the thesis. Section 5.2 outlines our main findings. Section 5.3 describes our future work planned in the areas addressed by this thesis.

5.1 Summary

The goal of this thesis was "to improve our understanding of how computing devices can better understand us and our needs." As argued in this thesis, such understanding if often embodied, at least partly, in a concept known as context awareness. The primary method used to endow computers with context awareness has been—and we argue it will continue to be—machine learning.

In examining these topics, we have narrowed the focus to application areas related to navigation. Despite this narrowing of application areas, there are still many diverse "needs" in navigation, and this thesis focused on three particular use cases within navigation where context awareness is deemed beneficial: (1) detecting of different human activities inside a typical office environment to improve indoor location tracking, (2) detecting different "mobility contexts" of a smartphone user to improve outdoor location tracking, and (3) enabling "ice aware" route optimization for ships sailing in ice-covered waters to improve and automate the route planning needs of such ships. These use cases demonstrate the breadth of potential application areas of context aware technology. The three related publications included in this thesis improve the state-of-the-art in these application areas by introducing either novel methods, novel combinations of existing methods, or in-depth analysis of the performance of existing methods.

In addition to examining these application areas, this thesis has extensively reviewed the literature concerning context awareness and machine learning. In presenting and summarizing these topics, we have attempted to provide clear, tutorial-like examples, in order to aid readers unfamiliar with these subjects.

We have reviewed the early theoretical work in "context", led by artificial intelligence pioneer John McCarthy and others, while pointing out that generalizations of context have not led to significant breakthroughs in context-aware systems. In presenting the conceptual underpinnings of context awareness, we have introduced two conceptual frameworks for understanding context awareness and contextual reasoning. The first was adapted from the writings of an ancient Greek orator seven Hermagoras, known as the "seven circumstances". The second, which we have dubbed the "context pyramid" presents a division of the various steps in contextual reasoning into six levels ranging from raw data to "rich context". These two frameworks, general in nature, can assist the researcher and developer aiming to build context-aware systems by dividing the problem up into different categories of contextual information and steps in contextual processing.

On the topic of machine learning, this thesis has traced the history of the subject from its early beginnings with Arthur Samuel up to the modern notion. We have presented two major types of machine learning, supervised and unsupervised, using a toy problem, computer chess, as an example. We have emphasized the importance and benefits of automatic learning, despite the fact that supervised learning usually requires manual labeling of training data. Unsupervised learning, on the other hand, can largely meet the desire for automated learning, although it often requires some human interpretation of the results.

5.2 Main Findings

Below summarizes our main research findings:

- Context awareness is a broad and challenging topic, but one can find many beneficial applications of context awareness in the field of navigation.
- Machine learning constitutes a powerful set of methods for endowing computers with context awareness.
- A systematic evaluation of different available machine learning algorithms should be undertaken when applying machine learning to the problem of context awareness, especially if the aim is to maximize performance. The important fact is often overlooked by navigation researchers working on context awareness.
- Context-aware smartphone applications are a present reality. Limited experiments have shown that a smartphone application could detect, e.g. various

activities in an office setting and different outdoor mobility contexts, with high accuracy ($>$90% for the former; $>$97% for the latter).

- As an example of a maritime application, awareness about ice conditions (as a function of space and time) can be exploited to perform automated route optimization. Such capability could augment or even replace the currently manual task of route planning performed by crews of ships sailing in ice covered waters.
- Many other applications of context awareness are evident in emerging technologies, and context awareness will play an even stronger role in the future, especially in so-called "smart devices".

5.3 Future Work

In many ways, this thesis has only scratched the surface in exploring context awareness for navigation applications. In tackling the broader goal "to improve our understanding of how computing devices can better understand us and our needs," we feel even less compelled to declare our work complete. This section outlines some of the planned future work in developing context-aware navigation applications.

Our future work can be divided into three broad categories: (1) future work in the three application areas covered by the included publications, (2) future work in new application areas, and (3) future work that can benefit context awareness broadly. The first category of future work will not be discussed further in this section because it is described in the included publications [P3]-[P5]. The other two categories will be discussed in separate sub-sections below.

5.3.1 Future Applications

In Figure 4.1 we hinted at future application areas or "use case scenarios". Several of the highlighted use case scenarios are part of near future work. For example, in one recently initiated project, we aim to develop a "tactical situation awareness" for soldiers.

Military applications of context awareness are particularly promising because the cost limitations are not as strict as in other application areas and specially-designed sensors can be installed, e.g. attached to various body parts of a soldier (helmet, boots, chest, etc.) or to other military equipment, providing a rich set of raw sensor data from which to generate context awareness. On the other hand, in military applications, reliability requirements are very high and typically there is a strong requirement for real-time functionality. For example, if a system is designed to detect when a soldier is in danger or injured, then false negatives, as well as false positives could prove very costly.

Another application area that has strong potential is healthcare and fitness monitoring. With the growing popularity of "wearable devices," such as smartwatches and small heart-rate monitors, such applications have greater widespread consumer appeal. Many devices already exist that can, e.g. monitor calorie usage by tracking steps, but it remains a challenge to reliably and automatically detect different activities such as walking, running, cycling, hiking, etc. This is, of course, strongly overlapping with the topic of [P4], but we believe healthcare and fitness monitoring can go much beyond mere "mobility context" and incorporate other aspects, such as recognizing social interactions, detecting abnormal health or changes to a person's routine that might affect health and fitness, and warning users of dangerous or unhealthy situations. The concept of a "personal health assistant" is not really a matter of science-fiction but could be realized in the coming years. Context awareness and machine learning are the technologies that are likely to make this concept a reality.

5.3.2 General Issues and Potential Solutions

Lastly, we have noticed in our research several general issues that are relevant to context awareness in a broad sense. These issues are summarized as follows:

1. Supervised learning requires labeled data, and labeled data is expensive.
2. There is a lack of standardization in context awareness research.
3. Many context awareness experiments are not easy to repeat or independently verify.

The first general issue above is related to the use of supervised learning, which is often the adopted approach in many research works (such as in [P3] and [P4]). As mentioned in Section 3.3, it can be very costly and time-consuming to generated the labeled data required to perform supervised learning. Furthermore, it is generally the case that the more data that can be collected, the more performant and reliable the resulting model will be. For example, if we are aiming to develop a context-aware smartphone application that works well across a large population of users, then we will need to collect training data from a large, diverse population of test users. This is very costly, especially in a research setting.

There are two potential solutions to this issue. The first is that researchers and developers would publish and share their training data. This would benefit the overall research community. We have practiced this approach in publication [P4], but generally this is not a common practice. The second approach would be to collect a sizable amount of labeled training data and then to supplement it with unlabeled data (which is less costly to collect). Performing machine learning using a combination of labeled and unlabeled data is known as *semi-supervised learning*. This topic is outside the scope of this thesis but will be explored in our future work. As an example of this approach, a research and development team could collect a limited amount of labeled data using its own staff and volunteers and then supplement it with a large amount of crowdsourced unlabeled data. This is exactly the approach we are taking in a recently initiated project called MyGeoTrust (see Guinness et al., 2015).

The second general issue has to do with standardization. To put it precisely, there is a lack of standardization in context awareness research, and this issue makes it difficult to compare results among different studies. As described in Chapter 2 context is understood in many different ways, and there is no one "correct" way to categorize and organize the context space.To provide an example, in publication [P4] we defined seven mobility contexts: walking, running, static, moving slowly, riding a train, riding bus, and driving. In a related study, (Elhoushi et al., 2014) defined four mobility contexts: walking, running, bicycle, and land-based vehicle (including trains, metros, and cars). In yet another study, (Stenneth, 2103) investigated another set of mobility contexts.

This lack of standardization is understandable, due to the fact that different researchers have different applications in mind and different ideas about how to segregate the context space, but it would be more beneficial for the overall research community if some level of standardization were applied. For example, one could propose one or more standard ontologies of context for various applications. Then, when presenting results researchers could reference these standards, i.e. "the following classification results are according to standard X.Y...". Also, there is no reason to limit results to one particular standard; data could be processed according to several different standards and presented in the same publication. We note that some work on contextual ontologies can be found in the literature, but it is, in our view, an underdeveloped area. In our future work, we aim to contribute to and advance the notion of standard contextual ontologies.

The last general issue we would like to discuss is somewhat related to the second issue, and the solution is ironically similar to the first solution described above. One of the long-standing tenets of scientific research is reproducibility. Experiments should be described in enough detail so that other researchers can independently verify the results. In the case of context awareness research, this means that an independent researcher should be able repeat another researcher's data collection, apply the same algorithms, and achieve similar if not identical results. In reality, there are so many factors related to the environment, devices, and test subjects that collecting comparable data that produces comparable results is not realistic.

The solution is straight-forward. As an alternative, context researchers should publish the data upon which their results are based, along with sufficient documentation so that the data is usable by independent researchers. As already stated, this is rarely done in context awareness research. We aim to follow this practice in our future work and also to actively promote this practice by setting up a dedicated portal for this type of data exchange.

APPENDIX A. SOMETHING EXTRA

Appendices are purely optional. All appendices must be referred to in the body text

APPENDIX B. SOMETHING COMPLETELY DIFFERENT

You can append to your thesis, for example, lengthy mathematical derivations, an important algorithm in a programming language, input and output listings, an extract of a standard relating to your thesis, a user manual, empirical knowledge produced while preparing the thesis, the results of a survey, lists, pictures, drawings, maps, complex charts (conceptual schema, circuit diagrams, structure charts) and so on.