



TAMPERE UNIVERSITY OF TECHNOLOGY

# Context Awareness for Navigation Applications

Robert E. Guinness

Doctoral thesis

July 27, 2015



## **ABSTRACT**

This thesis examines the topic of context awareness for navigation applications. Context awareness can be defined as a computer's ability to understand the situation or context in which it is operating. In particular, we are interested in how context awareness can be used to understand the navigation needs of people using mobile computers, such as smartphones, but context awareness can also benefit other types of navigation users, such as maritime navigators. There are countless other potential applications of context awareness, but this thesis focuses on applications related to navigation. For example, if a smartphone-based navigation system can understand when a user is walking, driving a car, or riding a train, then it can adapt its navigation algorithms to improve positioning performance.

We argue that the primary set of tools available for generating context awareness is machine learning. Machine learning is, in fact, a collection of many different algorithms and techniques for developing "computer systems that automatically improve their performance through experience" [1]. This thesis examines systematically the ability of existing algorithms from machine learning to endow computing systems with context awareness. Specifically, we apply machine learning techniques to tackle three different tasks related to context awareness and having applications in the field of navigation: (1) to recognize the activity of a smartphone user in an indoor office environment, (2) to recognize the mode of motion that a smartphone user is undergoing outdoors, (3) to determine the optimal path of a ship traveling through ice-covered waters. The diversity of these tasks was chosen intentionally to demonstrate the breadth of problems encompassed by the topic of context awareness.

During the course of studying context awareness, we adopted two conceptual "frameworks," which we find useful for the purpose of solidifying the abstract concepts of context and context awareness. The first such framework is based

strongly on the writings of a rhetorician from Hellenistic Greece, Hermagoras of Temnos, who defined seven elements of “circumstance”. We adopt these seven elements to describe contextual information. The second framework, which we dub the “context pyramid” describes the processing of raw sensor data into contextual information in terms of six different levels. At the top of the pyramid is “rich context”, where the information is expressed in prose, and the goal for the computer is to mimic the way that a human would describe a situation.

We are still a long way off from computers being able to match a human’s ability to understand and describe context, but this thesis improves the state-of-the-art in context awareness for navigation applications. For some particular tasks, machine learning has succeeded in outperforming humans, and in the future there are likely to be tasks in navigation where computers outperform humans. One example might be the route optimization task described above. This is an example of a task where many different types of information must be fused in non-obvious ways, and it may be that computer algorithms can find better routes through ice-covered waters than even well-trained human navigators. This thesis provides only preliminary evidence of this possibility, and future work is needed to further develop the techniques outlined here. The same can be said of the other two navigation-related tasks examined in this thesis.

## **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 now 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. Then, refine and repeat.

There are many I would like to thank for their contributions to this thesis and to

my overall development. First, I'd like to thank my family for their many years of support, beginning with my parents but also including my siblings, Erin and Joe. Nowadays I have my own family, and I'd like to thank especially my wife, Anne-Mari, for her steadfast support, and my children Pyry, Kilian, and Aarre for bringing joy to each day.

Next, I'd like to thank my thesis supervisors, Prof. Jarmo Takala, Prof. Ruizhi Chen, and Prof. Heidi Kuusniemi. Thank you for all the support and guidance, as well as the opportunity to complete this research under your supervision. Lastly, I'd like to thank my colleagues and co-authors, who have made the environment in which this work was completed very fun and interesting. I have learned a lot from all of you.

Kirkkonummi, 24.7.2015

Robert E. Guinness

## TABLE OF CONTENTS

<i>Abstract</i> . . . . .	i
<i>Preface</i> . . . . .	iii
<i>Table of Contents</i> . . . . .	v
<i>List of Figures</i> . . . . .	vii
<i>List of Tables</i> . . . . .	ix
<i>Abbreviations</i> . . . . .	xi
<i>List of Publications</i> . . . . .	xv
<i>1. Introduction</i> . . . . .	1
1.1 Background and Motivation . . . . .	1
1.2 Research Objectives and Scope . . . . .	4
1.3 Main Contributions . . . . .	7
1.4 Thesis Outline . . . . .	9
<i>2. Context Awareness</i> . . . . .	11
2.1 A Framework for Contextual Information . . . . .	12
2.2 History . . . . .	15
2.3 Related Studies . . . . .	16
2.3.1 Context Awareness for Office Environments . . . . .	17
2.3.2 Context Awareness for Mobility . . . . .	19
2.3.3 Ice Awareness for Maritime Navigation . . . . .	22
2.4 Theory . . . . .	23

---

2.5	How to Sense Context . . . . .	27
3.	<i>Machine Learning</i> . . . . .	31
3.1	Computer Chess: An Example Learning Task . . . . .	32
3.1.1	Rule-Based Methods . . . . .	32
3.1.2	A Trivial Chess Learning Program . . . . .	33
3.2	Modern Machine Learning . . . . .	34
3.3	Supervised Learning . . . . .	38
3.4	Unsupervised Learning . . . . .	43
3.5	Concluding Remarks . . . . .	47
4.	<i>Overview of Publications</i> . . . . .	49
4.1	Summary of Publications . . . . .	49
4.2	Mapping of Publications to Research Areas . . . . .	51
4.3	Author's Contributions to the Publications . . . . .	53
5.	<i>Conclusions</i> . . . . .	55
5.1	Summary . . . . .	55
5.2	Main Findings . . . . .	57
5.3	Significance of the Results . . . . .	58
5.4	Future Work . . . . .	59
5.4.1	Future Work in Investigated Applications . . . . .	60
5.4.2	Future Applications . . . . .	61
5.4.3	General Issues and Potential Solutions . . . . .	62
5.5	Concluding Remarks . . . . .	64
	<i>Bibliography</i> . . . . .	67



## LIST OF FIGURES

2.1	The Context Pyramid . . . . .	29
3.1	Training data for supervised learning . . . . .	41
3.2	Input data for unsupervised learning and one clustering result . . . .	44
3.3	Further results from unsupervised learning . . . . .	48
4.1	Mapping of included publications to research areas . . . . .	52



**LIST OF TABLES**

2.1 Publications related to mobility context . . . . . 20

3.1 Example data for supervised learning . . . . . 40



## **ABBREVIATIONS**

AdaBoost	Adaptive Boosting
ADL	Activities of Daily Life
AESS	Aerospace and Electronics Systems Society
AI	Artificial Intelligence
AIS	Automatic Identification System
ANN	Artificial Neural Network
BC	Before Christ
BN	Bayesian Network
CHMM	Couple Hidden Markov Model
CIL	Contextual Intensional Logic
COTS	commercial-off-the-shelf
CPU	Central Processing Unit
DBSCAN	density-based spatial clustering of applications with noise
DT	Decision Tree
DTS	discriminative temporal smoothing
E	East
EM	expectation-maximization
FIDE	Federation Internationale des Echecs

Five Ws	Who, What, Where, When, and Why
FMS	Future Urban Mobility Survey
GIS	Geographic Information Systems
GMM	Guassian mixture model
GPS	Global Positioning System
GPSAR	Google Play Services Activity Recognition
HCI	Human-Computer Interaction
HF-SVM	Hardware-friendly Support Vector Machines
HIER	hierarchical agglomerative clustering
HMM	Hidden Markov Models
IBk	instance-based k-nearest neighbor algorithm
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
ION	Institute of Navigation
IRO-2	Ice Forecast and Route Optimization
jmc	John McCarthy
kMC	k-means clustering
kNN	k-Nearest Neighbor
KStar	K* algorithm
lat	latitude
LDA	Linear Discriminant Analysis
LMS	local models semantics

---

LoMoCo	Location-Motion-Context
long	longitude
LR	logistic regression
LS-SVM	Least Squares-Support Vector Machine
LSA	latent semantic analysis
LWL	Locally weighted learning
MAP	maximum a posteriori
MEMS	microelectromechanical systems
MLE	maximum likelihood estimate
MLP	Multilayer Perceptron
N	North
NB	Naïve Bayes
OPTICS	ordering points to identify the clustering structure
OS	Operating System
PCA	principal component analysis
PDA	Personal Digital Assistant
PLANS	Position Location and Navigation Symposium
PLC	Propositional Logic of Context
RF	RandomForest
RGB	Red, Green, and Blue
SAR	Synthetic Aperture Radar
SQL	Structured Query Language

SVM	Support Vector Machines
UCI	University of California, Irvine
WGS84	World Geodetic System 1984
WLAN	wireless local area network



## LIST OF PUBLICATIONS

This thesis contains a compilation of five previously published papers, referred to as [P#] throughout the text. The following publications are included:

- [P1] R. Chen, **R. E. Guinness**, “Context Awareness” in *Geospatial computing in mobile devices*. Boston: Artech House, ch. 8, pp. 150-170, 2014.
- [P2] R. Chen, **R. E. Guinness**, “Contextual Reasoning” in *Geospatial computing in mobile devices*. Boston: Artech House, ch. 9, pp. 171-197, 2014.
- [P3] L. Pei, **R. E. Guinness**, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, L. Chen, and J. Kaistinen, “Human behavior cognition using smartphone sensors,” *Sensors*, vol. 13, no. 2, pp. 1402–1424, 2013.
- [P4] **R. E. Guinness**, “Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors,” *Sensors*, vol. 15, no. 5, pp. 9962–9985, 2015.
- [P5] **R. E. Guinness**, J. Saarimäki, L. Ruotsalainen, H. Kuusniemi, F. Goerlandt, J. Montewka, R. Berglund, and V. Kotovirta, "A method for ice-aware maritime route optimization,” in *Position, Location and Navigation Symposium–PLANS 2014*, IEEE/ION, pp. 1371–1378, 2014.



# 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<sup>1</sup>. 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” [2, p. 11]. They define digitization as “converting things into bits that can be stored on a computer and sent over a network” [2, 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” [2, p. 9]. It is within this wider context of digital technologies and the second machine age that this thesis is best understood.

Digital technology 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. There are four specific technologies that have particular relevance to this thesis: (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

---

<sup>1</sup> 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.

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

The applications arising from this technological convergence span many different areas, and we have no intention to cover these applications exhaustively in this thesis. Instead we focus on one application area—navigation. This research scope of this thesis will be defined more precisely in Section 1.2.

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. The first commercially-available phone with a Global Positioning System (GPS) receiver, the *Benefon Esc!*, was released in 1999. 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 [3], and by 2008 this number exceeded 139 million [4]. By 2012, there were already more than one billion smartphones in use worldwide [5]. This number is forecast to reach nearly 2.5 billion in 2015 [6]. 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 traveling 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 it is clear that the developments are 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 fulfill 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.

The field of study related to how humans communicate with computers and vice versa is known as Human-Computer Interaction (HCI), and this thesis finds relevance in HCI. Essentially, we aim to reduce the need for explicit human-to-computer communication by increasing the ability of computers to understand humans. 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” [1]. This is our favorite 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 this 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 (AI) has been an elusive goal of computer science researchers ever since the term was coined in 1955<sup>2</sup>. 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 [2], computers 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 can<sup>3</sup>. 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 wants 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

---

<sup>2</sup> 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 [7]. Therefore, it is not entirely clear who first came up with this term, and in an interview even McCarthy himself could not recall [8].

<sup>3</sup> 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.

yet “smart”. They have the “brawn” 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*)<sup>4</sup>. In particular, this thesis will focus on how machine learning can be used to build context or situation awareness, in order to solve problems in navigation. Highly respected navigation researcher Dr. Paul Groves has called context one of the four key challenges for the next generation of navigation technologies [9], and we share this view.

By focusing on navigation, 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 applications described in this thesis are seen only as examples and not as end goals in themselves. ««««there are a wide range of rich mobile applications that can be enabled with the aid of context awareness, and researchers have identified context awareness as one of the key open issues in mobile computing research [10].

In this thesis, we focus on three tasks related to context awareness that are relevant to the field of navigation: (1) to recognize the activity of a smartphone user in an indoor office environment, (2) to recognize the mode of motion that a smartphone user is undergoing outdoors, and (3) to determine the optimal path of a ship traveling 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 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 provides possible enhancements for a navigation or position tracking system that must work also indoors. Reliable and everywhere-available indoor positioning is one of the biggest unsolved problems in navigation. Context awareness is one way to help solve this problem. For example, if the system detects that a user

---

<sup>4</sup> For consistency, in this thesis we primarily use the term context awareness, but it can be considered synonymous with the term situation(al) awareness.

is sitting and 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.

Similarly, the second task is important for navigation because a navigation system designed to work outdoors 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 know 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.

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 capability 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 link between machine learning and context awareness and exploits this link to demonstrate possible applications in the field of navigation. Firstly, the author has developed a generic conceptual framework for the multi-step processing of raw sensor data into contextual information, which had been largely lacking in the literature. Also, many studies on context awareness either adopt a rather narrow view of context or do not provide any clear framework or mechanism of how to encode a situation or context in a systematic way. This thesis proposes and describes a simple but powerful framework for describing a context in terms of seven key questions, covered further in Chapter 2 and [P1]. Together, these two conceptual frameworks benefit the research community by making the abstract and ambiguous concepts “context” and “context awareness” more concrete and clearly defined, as well as providing a methodological skeleton on which to build context-aware systems.

In addition, this thesis examines three separate use case scenarios or applications of context awareness. These relate to the three tasks described in Section 1.2 above. The remainder of this section describes the key contributions related to these use case scenarios.

Firstly, the thesis presents a probabilistic Location-Motion-Context (LoMoCo) model, combining location and motion context, used to detect human behavior (i.e. activities) in an indoor office environment. The sensors used to detect the human behavior include only sensors available in commercial-off-the-shelf (COTS) smartphones, as well as access points in a wireless local area network (WLAN) used for the positioning component. To our knowledge, this is the first study focused on detecting office-environment activities that utilizes only smartphone-based sensors and standard WLAN access points. This is significant because earlier studies mostly relied on the installation of custom-designed sensors in the office environment or wearable sensors that are not in common use in offices. For example, [11] relies on sensors installed in an office chair and multiple cameras installed in an office room to infer activity. As smartphones and WLAN access points are already widely present in office environments around the world, the results of this research have more potential for widespread application. Our method can be used anywhere within an office building where WLAN signals are present, provided the user has a smartphone.

A problem related to the above topic is the determination of whether a smartphone user is indoors or outdoors. This is important contextual information because the optimal positioning system differs depending on whether the user is indoors or outdoors. Another important benefit of this contextual information is that it can be used to conserve smartphone battery usage. Outdoor positioning systems, namely those based on Global Navigation Satellite Systems (GNSS), are power intensive and should be turned off automatically when the user is indoors. The method described in this thesis for indoor-outdoor determination is, according to our knowledge, the first smartphone-based probabilistic indoor-outdoor method described in the literature.

Next, this thesis includes a systematic evaluation of a large number of machine learning algorithms applied to the problem of detecting “mobility contexts”, including consideration of the computational cost of the resulting classifiers, due to their intended use in resource-limited mobile devices. The number of algorithms investigated and applied to this problem is larger than any other previous study, according to our knowledge. Also, most existing studies dealing with mobility context do not consider or evaluate the computational cost of classifiers, so our study is novel in this aspect.

Furthermore, our study is the first research on mobility context to utilize GNSS, accelerometers, and information from Geographic Information Systems (GIS) for the purposes of detecting mobility context<sup>5</sup>. In particular, GIS is an important source of information for detecting mobility context because it can be used to determine proximity to relevant landmarks, such as train stations and bus stops. Earlier studies did not consider this important source of semantic information, and our research provides strong evidence, as a result of feature selection, that such information improves the context recognition result.

Our research also studied the influence of parameter tuning for the RandomForest algorithm for this particular machine learning problem. After parameter tuning, we achieved an average recall rate of >97.5% for our test data. We are not aware of any other study achieving this level of performance for a comparable classification problem.

Lastly, for the purposes of developing an “ice-aware” maritime navigation system,

---

<sup>5</sup> This research was first published as a conference paper in 2013 (see [12]). The publication included in this thesis is an extended version of this earlier study.

we developed a novel method for route optimization. Compared to earlier works in this area, our method is only the second graph-based approach to the problem of route optimization through ice-covered waters. Compared to the earlier graph-based approach, described in [13], our method is more computationally efficient, since it uses the A\* algorithm rather than Dijkstra's algorithm. We note that in [13] only a few tens of nodes were considered in the route optimization examples given, so computational complexity was perhaps not an apparent issue. To find truly optimal routes over large distances, however, it is necessary to consider thousands of nodes or more. [14], published later than [P5], also used the A\* algorithm. Compared to [14], we incorporated into our method an operational constraint related to ice breaker assistance, whereas the cost function used in [14] did not consider this issue. Another advantage of our cost function is that the cost, expressed in the unit of time, is easier to interpret. The cost function employed in [14] is a linear combination of four different variables, and its physical meaning is difficult to interpret.

An earlier study tackling ice-aware route optimization expressed the problem as a differential equation and used numerical methods to solve it, such as Powell's method [15]. Such methods, however, do not guarantee a global optimum. Due to the complex nature of ice fields, local minima can be significantly worse than the global optimum. The benefit of a graph-based approach is that shortest-path algorithms exist that can guarantee an optimal solution. The main novelties in our method is the design of a suitable graph structure that provides a reasonable trade-off between realistic modeling of ship motion and computational complexity and the incorporation of ice breaker assistance into the cost function used in optimization.

## 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 are the *human* context (i.e. the computer *user's* situation) and the *environment* context, but device-specific context can also be important 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 [16] [17]. Others (e.g. [18]) have attempted to formalize the concept mathematically, a topic to which we will turn in Section 2.4.

Following the approach in [19], we first present a “layman's” definition of context, in order to provide a working, notional notion for our initial discussion. In later sections, we will explore more formal definitions of context. Let us start by seeing how context is defined in a dictionary. In the Merriam-Webster Dictionary, the word context has two definitions [20]:

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, 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 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 Section 1.2, *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 [21]).

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 brief historical overview of context awareness, which also serves as a review of the early context awareness literature. Next, Section 2.3 provides a review of more recent context awareness literature, paying particular attention to studies relevant to the three tasks described earlier in Section 1.2. Section 2.4 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. Finally, Section 2.5 describes the sources and methods of context awareness.

### 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 an authoritative 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 [22]. 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” [23].

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 (FGI) in Masala, Finland

**When:** Tuesday, 10 March 2015 at 11:03AM

**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 it 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 Operating System (OS), which includes a GPS receiver, WLAN-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 completely 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).

The following provides an introduction to the context framework proposed in this thesis:

- “What” usually refers to the *activity context*, that is, what is actually happening. In some contexts, there might be little “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 encode 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. 10–20 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.

Further details about this framework are covered in [P1]. As context-aware systems develop, we suspect that our 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 review of the literature, 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.” [24]. 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” [25]. Although not specifically highlighted by Weiser, the necessity of these computers to understand context is clearly evident. Another

article, published in 1992 by Want et al., may be the first implementation of a context-aware device described in the literature, even though the authors did not use the term “context-aware” [26]. 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 HCI, which provides an excellent review of the state-of-the-art in context awareness for that time period [27].

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” [28]. 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 2.4 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 be discussed further in Section 2.4. 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 [18] [29] [30] [21], and [31]. In addition, [32] and [33] provide excellent reviews of context in artificial intelligence.

### 2.3 Related Studies

A comprehensive review of recent context awareness literature would require covering hundreds, if not thousands, of different studies. The review found in [34], which only looks at journal articles published between 2000 and 2007, covered well over 200 different studies. Searches of several major databases of scientific publications (e.g. IEEE Xplore Digital Library, ACM Digital Library, SpringerLink) using the keyword “context-aware” each yielded thousands of results. Clearly, these publications will vary greatly in their relevance to this thesis. To help focus on only

the most relevant literature, we only include publications related to the three context awareness tasks described in Section 1.2, divided into respective sub-sections below.

### *2.3.1 Context Awareness for Office Environments*

In terms of the first task, recognizing the activity of a smartphone user in an indoor office environment, there are only a few studies basing their results on smartphone sensors, and they mainly rely on location awareness. For example, one early study described user tests of a context-aware Personal Digital Assistant (PDA) application, which can be considered a proto-smartphone-like mobile device, designed to be used in an office environment [35]. The main source of contextual information was location, but the application also used contextual information from calendar events that the user had previously recorded in the application. For example, if a user scheduled an event at a particular time and marked it as a meeting, then the system would automatically suggest meeting-related services when the meeting time approached and the user had entered the meeting room. The described system, however, was not capable of automatically inferring office activities independently from user-supplied information.

More recently, [36] described a pilot study related to smartphone-based context awareness of “mobile knowledge work”, which may partly take place in office environments. The article, however, mainly discusses the research problem in general, as well as describing the pilot data collection campaign. The article does not go into any detail on the results of analyzing the collected data, nor the planned analysis methods.

If we include studies using non-smartphone sensors to detect office-related activities, we encounter several other relevant studies. For example, [37] used a microphone and a USB camera to detect different office activities and motion patterns, including “rest,” “moving near door,” “conversation,” “nobody around,” etc. They used two machine learning techniques to detect motion and activity: Incremental Hierarchical Discriminant Regression (IHDR) trees and Hidden Markov Models (HMM). [38] similarly used a set of cameras to detect whether an office worker is working alone or in a meeting, or whether the office is empty. For activity detection, they used adaptive background modeling, which makes use of Gaussian mixture densities [39].

Another example is [40], which used a richer set of sensors, including two USB

cameras, a pressure sensor, and an acceleration sensor (a Wii remote controller) built into an office chair. This study focused on detecting location and posture context, such as leaning back in the chair, leaning on the desk, or upright sitting posture. They primarily used k-means and k-Nearest Neighbor (kNN) clustering to perform context recognition. Similarly, [41] presented a so-called “smart office chair” designed to measure an office worker’s mental and physiological states, such as sleepiness and concentration. The details of the recognition algorithm, however, are not given in the paper.

One study used a combination of mobile phone sensors and room occupancy sensors installed in an office environment [42]. The resulting application, called *WorkSense*, was able to detect when and where meetings and conversations took place. By sensing social interactions, the authors were also able to identify project groups automatically.

Other relevant studies have focused on using wearable sensors to detect different activities, some of which are common in office environments. For example, [43] used accelerometers attached to various parts of a person’s body. Although they did not specifically focus on an office environment, some of the activities they addressed are very relevant. These include sitting, standing, walking, writing on a whiteboard, typing on a keyboard, and shaking hands. The authors used a naïve Bayes classifier for activity recognition. Similarly, [44] used wearable sensors to detect several different office-related activities (typing, cleaning a whiteboard, using an elevator, etc.). The authors evaluated the use of kNN and Multilayer Perceptron (MLP) classifiers for activity recognition<sup>1</sup>.

Finally, [45] studied context recognition in a meeting room environment, using a suite of sensors including: a microphone array, passive infrared sensors, and an illumination (light) sensor. They detected different states of activity in the meeting room, such as: presentation, monologue, discussion, room idle, people entering room, and room in use. They do not describe their context recognition algorithm in any great detail, but it appears to be a heuristically-defined rule-based algorithm.

The relatively small number of studies related to context awareness in an office environment suggests this is a fairly immature research topic. We can conclude from our literature review that the research in this thesis is rather novel, especially with

---

<sup>1</sup> MLP is a type of Artificial Neural Network (ANN).

regards to smartphone-based context awareness for office environments.

### 2.3.2 Context Awareness for Mobility

With regards to the second task, recognizing modes of motion that a smartphone user is undergoing outdoors, there are comparatively many relevant studies. Despite this fact, to our knowledge no systematic review of this research area has been published, but a few related reviews are available. For example, [46] reviews literature on the use of GPS to study health-related physical activity. The focus of this review, however, is mostly on assessing health behavior, rather than on methods to detect different mobility contexts. [47] reviews smartphone-based “opportunistic user context recognition”, which includes some literature on mobility context recognition, and similarly [48] reviews “anticipatory mobile computing”. While useful in our review of the literature, none of these reviews focus specifically on mobility context awareness, and they exhibit significant gaps in this regard.

Due to the large number of relevant studies, we will not describe them individually, but pertinent facts from these studies can be found in Table 2.1. We make no claim that this compilation of publications on the subject is exhaustive, but it is, according to our knowledge, the most comprehensive compared to existing literature. We decided to include also studies that utilized wearable sensor modules, since this research is closely linked to smartphone-based research. All of the listed studies used supervised machine learning techniques, except for [49], which evaluated several different unsupervised learning techniques.

Note also that many of the studies investigate not solely mobility contexts but also other contexts, such as Activities of Daily Life (ADL) or various posture contexts (sitting, lying down, etc.). In Table 2.1, we have only identified the relevant motion-related modes. Some other motion-related contexts, such as using an elevator or walking upstairs/downstairs, were investigated by a few studies, but we listed only those modes common to a significant number of studies. For those studies incorporating additional contexts, we have identified these contexts collectively using the symbol “+”. Lastly, we point out that some of the included studies investigated indoor motion modes, even though our research task was to study outdoor mobility contexts.

**Table 2.1:** Publications related to mobility context. The definitions for the abbreviations used for algorithms/techniques are given in the Abbreviations section provided above. The abbreviations used for the motion modes are as follows: *S* = static (including standing, sitting, etc.), *W* = walking, *R* = running/jogging, *B* = riding a bicycle, *D* = driving a motor vehicle, *MT* = any kind of motorized transport (including car, bus, train, etc.), *RB* = riding a bus, *RT* = riding a train/tram/light-rail, and *RS* = riding a subway/metro. “+” indicates that other motion modes are also covered by the publication.

Year	Author(s) & Citation	Device(s) Used	Algorithm(s) / Technique(s) Evaluated	Motion Modes Studied
2000	Foerster, Smeja & Fahrenberg [50]	accelerometers	rule-based	S, W, B, +
2002	Lee & Mase [51]	sensor module	fuzzy-rule-based	S, W, +
2005	Lester et al. [52]	sensor module	AdaBoost, NB	S, W, R, B, +
2005	Ravi et al. [53]	sensor module	DT, kNN, SVM, NB, meta-classifiers	S, W, R
2006	Pärkkä et al. [54]	various sensors	rule-based, DT, ANN	S, W, R, B, RB, +
2006	Pirttikangas, Fujinami, & Nakajima [44]	sensor modules	ANN, kNN	S, W, R, B, +
2007	Suutala, Pirttikangas & Röning [55]	sensor modules	SVM, HMM, SVM-HMM, DTS	S, W, R, B, +
2008	Kunze & Lukowicz [56]	sensor modules	DT, kNN, BN	S, W, R, B, +
2008	Jin et al. [57]	sensor modules	fuzzy-rule-based	S, W, R, +
2009	Yang [58]	mobile phone	DT, NB, kNN, SVM, HMM	S, W, R, B, D, +
2010	Reddy et al. [59]	mobile phone	DT, kMC, NB, ANN, SVM, CHMM, DT+DHMM	S, W, R, B, MT,
2010	Pei et al. [60]	mobile phone	rule-based	S, W, +

**Table 2.1 – continued from previous page**

<b>Year</b>	<b>Author(s) &amp; Citation</b>	<b>Device(s) Used</b>	<b>Algorithm(s) / Technique(s) Evaluated</b>	<b>Motion Modes Studied</b>
2010	Frank et al. [61]	sensor module	NB, BN, HMM	S, W, R, +
2011	Stenneth et al. [62]	mobile phone	DT, NB, BN, RF, ANN	S, W, B, D, RB, RT
2011	Pei et al. [63]	mobile phone	DT, BN, SVM	S, W, +
2011	Susi, Borio, & Lachapelle [64]	sensor module	DT, NB, kNN	S, W, R, +
2012	Bancroft et al. [65]	sensor modules	NB, rule-based	S, W, R, B, MT, +
2012	Anguita et al. [66]	mobile phone	SVM, HF-SVM	S, W, +
2013	Guinness [12]	mobile phone	20 different algorithms	S, W, R, D, RB, RT, +
2013	Susi, Renaudin, & Lachapelle [67]	sensor modules	DT	S, W, +
2013	Feng & Timmermans [68]	sensor module	BN	W, R, B, D, RB, RT
2013	Hemminki, Nurmi & Tarkoma [69]	mobile phone	HMM, Adaboost, meta classifier	S, W, RB, RT, RS
2014	Stenneth [70]	mobile phone	NB, BN, DT, RF, ANN, meta-classifiers	S, W, B, D, RB, RT
2014	Xia et al. [71]	mobile phone	SVM	S, W, B, D
2014	Elhoushi et al. [72]	mobile phone	DT	W, R, B, MT
2014	Parvainen et al. [73]	mobile phone	DT, SVM, MAP	S, W, R, B, MT
2014	Yu et al. [74]	sensor module + mobile phone	DT, AdaBoost, SVM	S, W, R, B, MT
2014	Kwon, Kang, & Bae [49]	mobile phone	GMM, kMC, HIER, DBSCAN	S, W, R, +
2014	Sankaran et al. [75]	mobile phone	rule-based, GPSAR, FMS	S, W, MT

**Table 2.1 – continued from previous page**

<b>Year</b>	<b>Author(s) &amp; Citation</b>	<b>Device(s) Used</b>	<b>Algorithm(s) / Technique(s) Evaluated</b>	<b>Motion Modes Studied</b>
2014	Chiang, Yang & Tu [76]	mobile phone	DT, kNN, NB, SVM	S, W, R, B, D, +
2015	Yu and Cho [77]	mobile phone	DT, SVM, ANN	S, W, R, D, RB, RT, RS

### 2.3.3 Ice Awareness for Maritime Navigation

Regarding the third task, determining the optimal path of a ship traveling through ice-covered waters, only a few highly relevant studies are available in the literature. The earliest is [15]. As discussed briefly in Section 1.3, this work expressed the route optimization problem as a differential equation and used numerical methods to solve it, namely Powell’s method. As a result, their system cannot guarantee that the computed route corresponds to a global optimum. Similarly, [78] uses a genetic algorithm for route optimization in ice-covered waters. Genetic algorithms have better capabilities to escape from local minima, but they still do not guarantee a global optimum.

The first study to adopt a graph-based approach is [13]<sup>2</sup>. The authors present a method for ice-aware route optimization that uses Dijkstra’s algorithm to find the optimal route. In the examples given in the paper, only a few tens of nodes were shown for the sea area under consideration. For such small graphs, Dijkstra’s algorithm is tractable, but for larger graphs with many edges, Dijkstra’s algorithm does not scale well. To overcome this challenge, [14] uses the A\* algorithm, which uses a heuristic to guide the search process. Our method, published prior to [14], also uses the A\* algorithm. One difference between [P5] and [14] is that the cost function we developed takes into account possible ice breaker assistance. [14] does not consider this case.

<sup>2</sup> As reported in [78], [79] also used a graph-based approach, but it is only available in Korean. [13] appears to be an extension to [79].



Another related study is [80]. While it does not consider route optimization explicitly, the author investigates ship performance in varying ship conditions. The results are thus applicable to route optimization. Other examples of research in this area include [81], [82], and [83].

Lastly, the project Ice Forecast and Route Optimization (IRO-2), discussed in [84], is very relevant to this topic, but scientific results are not yet available publicly. A preliminary publication, reports on tests of the project's prototype ice-aware ship routing system, but the authors only describe performance of the ice forecast model, not the routing system itself [85].

## 2.4 Theory

As mentioned earlier, some computer science researchers have attempted to formalize the concept of context in a mathematical sense, most notably John McCarthy [18] [30]. 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  $\{1, 2, 3, \dots\}$  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.

McCarthy was one of the pioneers in developing a formalism of context. In a famous 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]hen 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 sheet of paper is between the book and the table) [86].

To deal with this issue he 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 a 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  $c2$  [21]. 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 [18]. He changes the notation slightly (adopting the notation from [29]), where formulas are sentences of the form:

$$c' : \quad ist(c, p), \quad (1)$$

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:

$$\text{“Sherlock Holmes stories”} : \quad ist(\text{“A Study in Scarlet”}, \text{“Watson is a doctor”}), \quad (2)$$

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 another term<sup>3</sup>, 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  $t$ ” [18]. Another

---

<sup>3</sup> In formal logic, a *term* is “a variable, constant, or the result of acting on variables and constants by function symbols” [87].

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” [18]. It is possible, however, that in another context  $c2$   $at$  can take on another meaning (e.g. momentarily at a place).

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

$$c0 : \quad ist(specialize-time(t, c), at(jmc, Stanford)) \equiv ist(c, at-time(t, at(jmc, Stanford))), \quad (3)$$

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

An important concept emphasized by McCarthy, as well as others (e.g. [29]), is that of *lifting relations*, also known as *lifting formulas* or *lifting axioms*. 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. [30] gives more precise definitions of these concepts:

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 3 can be considered a lifting relation because it relates the specialized context  $specialize-time(t, c)$  to the more general context  $c$ .

McCarthy never formalizes lifting in terms of a lifting operator, but he does introduce a general relation  $specializes$ , so that  $specializes(c1, c2)$  holds 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 into context  $c2$  [18]. He also provides non-monotonic relations to allow for inheritance of  $ist$  between a sub-context and super-context:

$$specializes(c1, c2) \wedge \neg ab1(p, c1, c2) \wedge ist(c1, p) \supset ist(c2, p), \quad (4)$$

and

$$specializes(c1, c2) \wedge \neg ab2(p, c1, c2) \wedge ist(c2, p) \supset ist(c1, p). \quad (5)$$

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) [31]. 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 local models semantics (LMS) [88], Contextual Intensional Logic (CIL) [89], and Structured contexts with fibred semantics [90]. For a review of formalized logic, see [91] or [92].

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 of 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  $c1$  is defined as  $specialize-time(t, c2)$  in PLC, then it would seem that in  $c2$  no notion of time even exists. Therefore, it is problematic to think of  $c2$  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 achieve 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, [93] argued persuasively that context cannot be defined independently from its use. Also, [94] 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<sup>4</sup>. 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 [95] and [96]). For other purposes, different formalisms may be needed.

## 2.5 *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

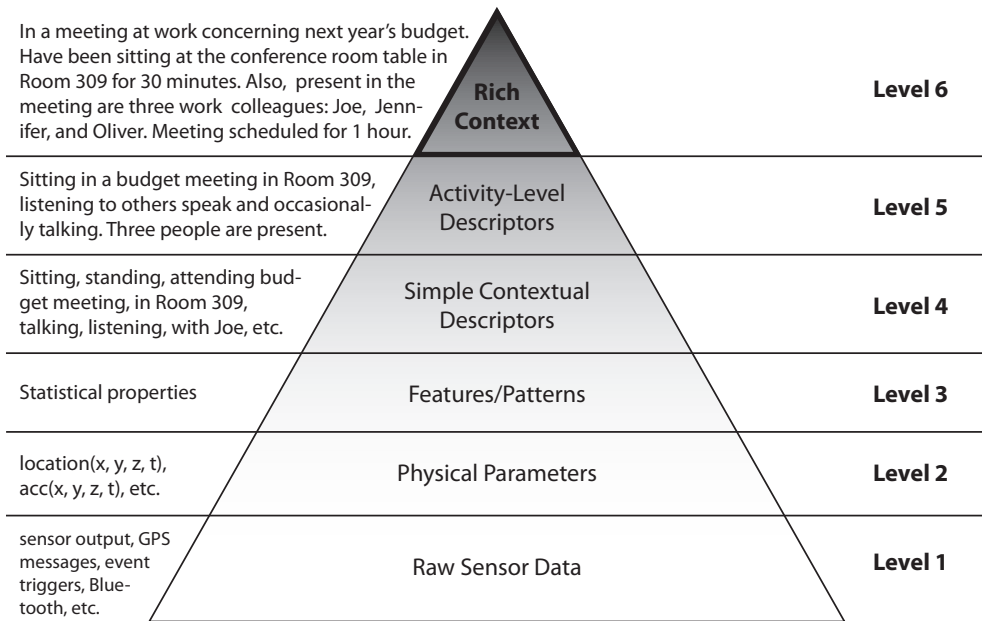
---

<sup>4</sup> It is not only humorous but also true to say that the meaning of context depends heavily on the context.

associated with context awareness vary greatly depending on the application. We have so far given mainly 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 traveling 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 us to the problem of generality in AI, which McCarthy wrote about in 1987, and it seems this problem persists.

Nonetheless, there are a few points that can be made 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 a device's 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 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 in a context-sensing system. We conceptualize this process as a "context pyramid", which was first presented in [P3] and later in [P2]. 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"



**Fig. 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.

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 combines 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.

This is true to a large extent even in the case of an ice-aware maritime route optimization system (i.e. Task 3), which appears at first sight very different from the example contextual information given in Figure 2.1. At the bottom of the pyramid would lie raw data concerning the ice field, e.g. data from Synthetic Aperture Radar (SAR) or other sources. The first step is to extract physical parameters from the raw data, such as a grid of values for ice thickness and ice concentration, or other system parameters such as the location of an ice breaker. Next, one might perform statistical analysis on the ice data, e.g. applying Gaussian process regression for the purposes of interpolation. Then, one uses features from Level 3 and a ship performance model, in order to estimate the ship's theoretical speed at coordinate  $(lat, long)$ . This is analogous to a Level 4 simple contextual descriptor (i.e. the *in what manner* context). Next, the optimal route, computed from the Level 4 contextual information, is roughly analogous to an Activity-Level Descriptor at Level 5 because it is a higher-level context inferred from lower-level contextual information. Finally, combining the optimal route with other navigational information, such as weather information, maritime traffic information, etc., produces a rich contextual description describing the ship's current situation that approaches Level 6 at the top of the pyramid.



### 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 [1].

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”<sup>1</sup>. 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 Section 3.4 describes unsupervised learning. Finally, Section 3.5 provides a few concluding remarks on the topic of machine learning.

---

<sup>1</sup> We have been unable to recover the original source of this quote. Some references cite [97], 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<sup>2</sup>. 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 [98]<sup>3</sup>. 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<sup>®</sup> beating the chess master Garry Kasparov in a six-game match-up in 1997<sup>4</sup>. 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 [100] [101], and [102]. 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<sup>5</sup>. Basically these strategy rules would

---

<sup>2</sup> Some of the earliest works in machine learning and artificial intelligence involved computer-playing games, such as chess and checkers. See [97] [98] [99].

<sup>3</sup> For example, according to colleagues of Alan Turing, Turing began considering computer chess during the 1940s.

<sup>4</sup> We note that Kasparov has accused IBM of cheating by letting human players intervene in one of the matches.

<sup>5</sup> We could have chosen the phrase “contexts in chess” rather than “situations in chess”, thus emphasizing the connection to context awareness, but this seems a bit awkward.

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.

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 exists in 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. If multiple winning records exist containing the same position, *ComputerWhite* should select one of those winning records at random. Otherwise, it would choose any valid move at random. *ComputerBlack* would do exactly the same, and the program would continue until the game’s conclusion. In essence, this program would play many possible chess games and try to avoid repeating any games that result in a loss or draw<sup>6</sup>. Over time it will be less likely to select poor moves because poor moves are more likely to result in a loss.

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 almost completely at random. Nonetheless, it would fit our definition of machine learning because it would

---

<sup>6</sup> Under this algorithm, repetition of non-winning games is possible but rather unlikely. Also, it is not guaranteed to play every possible game, though a somewhat simple modification to the algorithm could ensure every game is eventually played.

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*<sup>7</sup>. [98] calculated that such a database (or dictionary) would require roughly  $10^{120}$  entries to record all possible chess games. Even if we extracted from this database only the set of chess positions and maintained a smaller database of the best move for each chess position, this would still require around  $10^{120}$  records. 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 produce simple/efficient models of the learned task. Learning rate and performance, however, are usually the most important aspects of a machine learning algorithm<sup>8</sup>. 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 [103].

### 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

---

<sup>7</sup> Note this type of learning is similar to the so-called “dictionary attack” used to “learn” a log-in password.

<sup>8</sup> 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.

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:

Machine learning is programming computers to optimize a performance criterion using example data or past experience [104].

This definition appears quite close to that of [1], if we assume that “example data” can be generated automatically. This may be true in some cases, but in most methods described in [104], 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 [1].

Another recent definition is given by [105], which defines machine learning as:

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

This definition includes the “automatic” aspect, similar to [1], although we prefer the definition of [1] due to its simplicity. Also, note that there is no reference to improving performance with experience.

We note that some books on machine learning (e.g. [106]) omit to precisely define machine learning, perhaps because it has come to encompass many diverse methods. One book goes so far as to explicitly refuse to define machine learning in any principled way:

the kind of learning techniques explained in this book...are called machine learning without really presupposing any particular philosophical stance about what learning actually is [107]

Mitchell, on the other hand, 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  $E$  [108].

Continuing with formalisms, 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:

$$f : \mathbf{x} \rightarrow y \tag{6}$$

where  $f$  is a function,  $\mathbf{x}$  is a vector of inputs of arbitrary dimension, and  $y$  is an output with  $y \in Y = \{y_1, y_2, \dots, y_m\}$ , 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 of a 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 problems the form of the function is unknown, as well as its parameters.

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 learning 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 highlights the fact that for some machine learning problems, it may be well accepted that the problem cannot be solved perfectly<sup>9</sup>. Such

---

<sup>9</sup> One can imagine that for some individuals, voice recordings may not be enough to determine gender.

problems may be best represented in probabilistic form.

In many cases, a machine learning algorithm actually estimates 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:

$$y = \arg \max_{y \in Y} p(y|\mathbf{x}) \quad (7)$$

This is known as the maximum a posteriori (MAP) estimate of  $y$ . One of the benefits of estimating  $p(y|\mathbf{x})$ , however, is that it provides a measure of the confidence of the output  $y$ .

Algorithms designed to learn  $p(y|\mathbf{x})$  are known as *discriminative* approaches. An alternative approach is to first learn a model of the joint probability  $p(y, \mathbf{x})$  and then condition on  $\mathbf{x}$  to derive  $p(y|\mathbf{x})$ . These are known as *generative* approaches.

Apart from the distinctions regression vs. classification and discriminative vs. generative, 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 trainer or 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 in 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. Unsupervised learning can be considered automatic, but what can be very challenging in unsupervised learning is to define a notion of performance (recall the definition of machine learning given at the start of the chapter).

From this point forward in this thesis, we will drop the distinction between the modern popular notion of machine learning and the earlier meaning of [1] 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 subset of machine learning where performance improvements take 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 sample 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  $y$ <sup>10</sup>. 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 would like to develop 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. Instead of using the raw accelerometer signal, we define a feature from the accelerometer data, known as *dynamic acceleration*:

$$a_d = \text{var}(\{\sqrt{a_{xi}^2 + a_{yi}^2 + a_{zi}^2}\}_{i=1}^N) \quad (8)$$

where  $\text{var}(\cdot)$  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 researcher, Mary, has painstakingly collected a dataset for developing this context-aware application and labeled whether she was walking, running, or standing

<sup>10</sup> Recall that two points define a straight line.



still. Some sample 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 GPS receiver might have some error) or labeling errors made by Mary. Similarly, the static data contain many points where the speed is non-zero. 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”<sup>11</sup>. In this example, the goal of supervised learning would be to find a function  $f$  that maps the input data  $\mathbf{x}_i = (\text{speed}^i, a_d^i)$  to the correct output class, i.e.  $y_i \in \{\text{‘walking’}, \text{‘running’}, \text{‘static’}\}$ , 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.

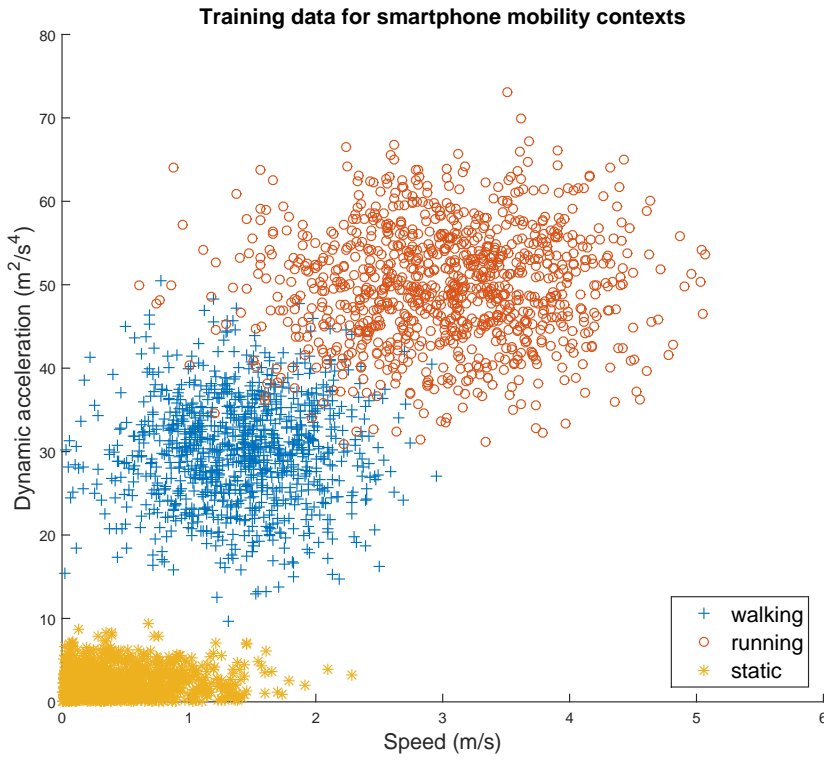
Three important interrelated concepts should now be introduced: *generalization*, *underfitting*, and *overfitting*. Generalization refers to the idea that supervised learning should “generalize” beyond the specific examples given in the training set. In other words, the goal is not simply to map the 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

<sup>11</sup> One technique to avoid such labeling errors is to remove these transition points entirely from the training set.

**Table 3.1:** Example data for supervised learning. The data consist of two-dimensional input data from smartphone sensors and a labeled output class.

ID	Speed (m/s)	Dyn. accel. ( $\text{m}^2/\text{s}^4$ )	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



**Fig. 3.1:** Example training data for supervised learning. The data are similar to those shown in Table 3.1. Note the partial overlap between the “walking” and “running” classes.

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 estimating 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, non-trivial 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, each from a different hypothesis class. That is, for each  $\mathcal{H}_i$ , we have a corresponding fitted function  $h_i$ . Let us denote these as  $\mathcal{H}_{best-in-class} = \{h_i\}_1^N$ , where  $N$  is the number of hypothesis classes.

The next step is to choose the best  $h_i$  from this set. 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 test 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].

So far we have discussed general concepts in supervised learning but not any specific algorithms. Several examples of supervised learning algorithms will be described in [P2].

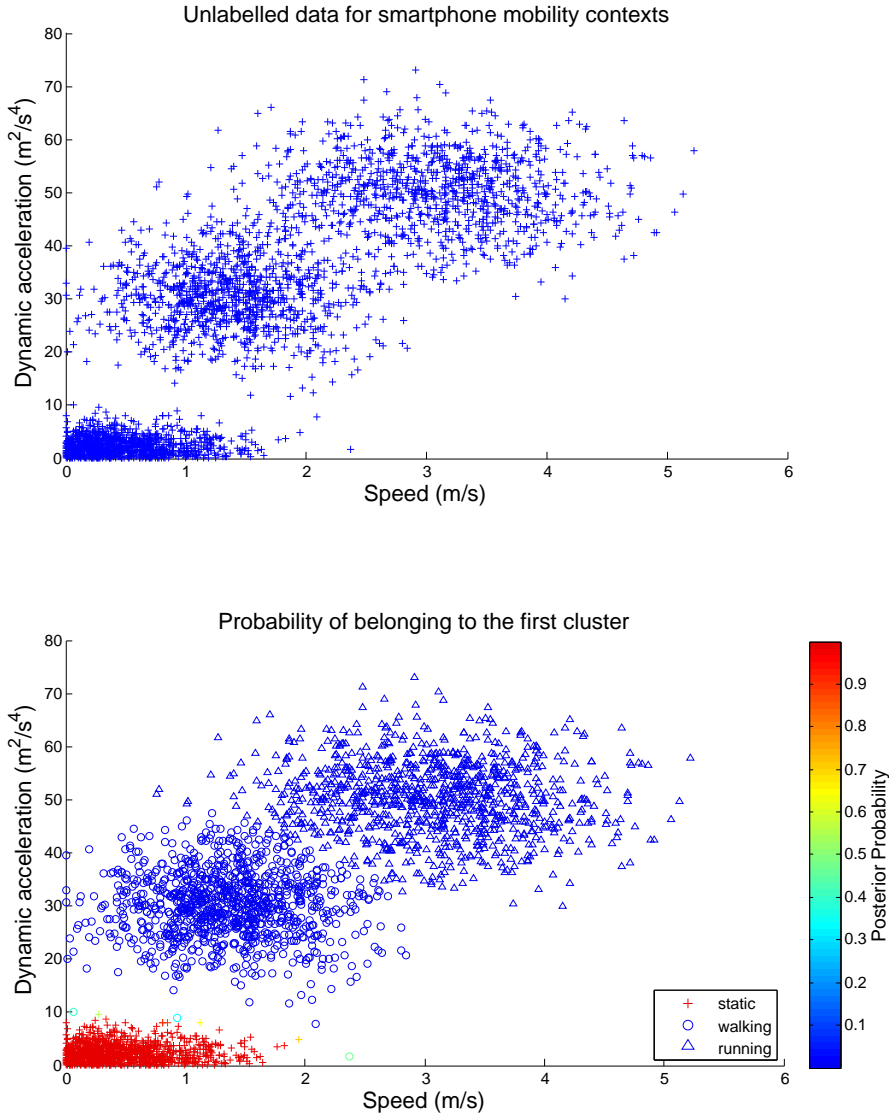
### 3.4 Unsupervised Learning

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<sup>12</sup>.

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

---

<sup>12</sup> This is also true to a certain extent in supervised learning, but the similarity is more striking in unsupervised learning.



**Fig. 3.2:** The top plot shows example input data for unsupervised learning. In contrast to Figure 3.1 no labels are available. The bottom plot shows one result from the EM-based clustering. The color of the points shows the posterior probability that the points belong to the first component in a GMM. Data labels are shown strictly for demonstration purposes. In a real situation, no such label would be available to interpret the unsupervised learning result.

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 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 to 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. There are a plethora of different unsupervised learning algorithms available in the literature that perform clustering. Possibilities include k-means clustering [109], OPTICS [110], and the expectation-maximization (EM) algorithm [111]. In particular, the EM algorithm has its roots in statistics 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:

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^K \pi_k \phi_k(\mathbf{x}; \theta_k) \quad (9)$$

where  $\mathbf{x}$  is a random vector,  $K$  is the number of components in the mixture model,  $\phi_k(\mathbf{x}; \theta_k)$  are normal distributions with parameters  $\theta_k = (\mu_k, \Sigma_k)$ ,  $\pi_k$  are mixing weights satisfying  $\pi_1 + \dots + \pi_K = 1, \pi_k \geq 0$ , and  $\Theta = \{\pi_1, \dots, \pi_K, \theta_1, \dots, \theta_K\}$  is the complete set of model parameters<sup>13</sup>.

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  $\Theta$  as above) for an underlying distribution  $p(\mathbf{x}|\Theta)$  used to model a given dataset, which we denote as  $\mathcal{D} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  [112]. The MLE is obtained by maximizing a function  $Q$  equal to the expected value of the log-likelihood  $\mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})$ , given the observed data  $\mathcal{D}$  and the current parameter estimates  $\Theta^{(i-1)}$ :

$$Q(\Theta, \Theta^{(i-1)}) = E[\log \mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})|\mathcal{D}, \Theta^{(i-1)}] = E[\log p(\mathcal{D}, \mathcal{Y}|\Theta)|\mathcal{D}, \Theta^{(i-1)}] \quad (10)$$

where  $\mathcal{Y} = (y_1, \dots, y_N)$  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

<sup>13</sup> The notation used for the GMM is similar but not identical to that given in [112].

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

Before applying the EM algorithm to find the parameters  $\Theta$  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 is preferred, such as the one described in [113]. In the interest of space, we assume in this example 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  $\Theta$  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  $\Theta$ , 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  $\mu_k$  and use the covariance matrix of  $\mathcal{D}$  for each of the initial  $K$  covariance matrices  $\Sigma_k$  [114].

After initialization, the algorithm then alternates between computing an expectation function (known as the E-step) and finding the parameters  $\Theta$  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:

$$\Theta^{(i)} = \arg \max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \quad (11)$$

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  $\epsilon$ ).



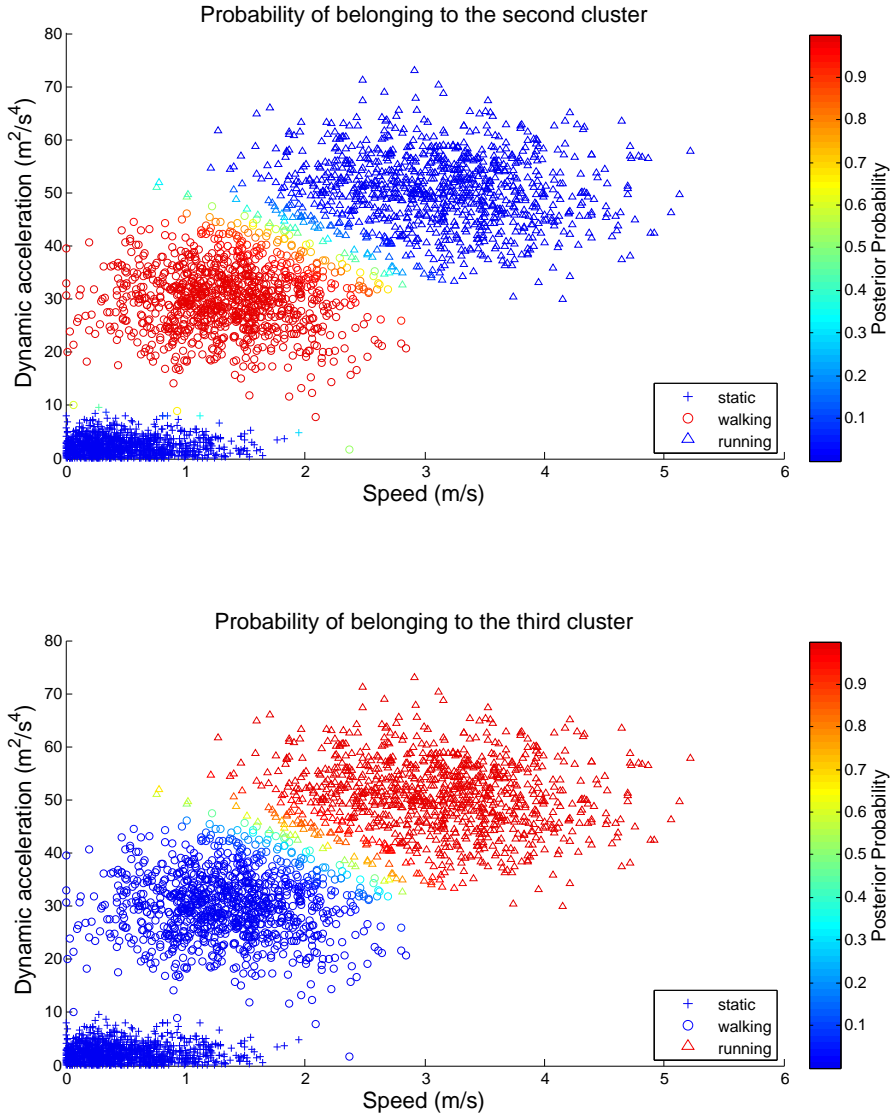
Finally, once the parameters  $\Theta$  are estimated, we can determine the posterior probability that a data sample  $\mathbf{x}_j$  belongs to a particular component  $k$  of the GMM, according to its so-called “membership weight” [114]:

$$w_j^k = p(y_j = k | \mathbf{x}_j, \Theta) = \frac{p_k(\mathbf{x}_j | \theta_k) \pi_k}{\sum_{m=1}^K p_m(\mathbf{x}_j | \theta_m) \pi_m} \quad (12)$$

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 each cluster can be drawn where the posterior probability reaches 0.5.

### 3.5 Concluding Remarks

Machine learning is clearly a wide topic covering many different concepts and techniques. Our purpose in this chapter was to introduce the most important concepts and to elucidate how machine learning can be used to endow computing systems with context awareness. Further examples and details will be given in the included publications. We have emphasized the role of training data in supervised learning. We have also demonstrated how a similar result can be achieved through the use of unsupervised learning, although measuring the performance of the result is problematic.



**Fig. 3.3:** These two plots continue the results presented in Figure 3.2. The coloring used in the top plot shows the posterior probabilities that the points belong to the second component in a GMM, whereas the bottom plot shows the same for the third component. As in Figure 3.2, the data labels are shown strictly for demonstration purposes.

## 4. OVERVIEW OF PUBLICATIONS

This chapter provides an introduction and overview to the five publications included in this compendium thesis. 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 2014 Proceedings of the Position Location and Navigation Symposium (PLANS), 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 briefly summarizes the publications. 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.

### 4.1 Summary of Publications

Next, we briefly summarize the included publications.

[P1] presents the concept of context awareness at a conceptual level, as well as tracing its historical development. It presents a conceptual framework for describing context, adapted from the seven elements of circumstance, first introduced by Hermagoras of Temnos in the 2nd Century BC. Our motivation was that existing literature on context awareness did not provide any comprehensive, simple, yet flexible framework for describing context. The publication also aims to show at a practical level how different aspects of context awareness can be implemented in a mobile device. Examples are given for the Android OS.

[P2] presents the concept of contextual reasoning, which is defined as the process of forming higher level inferences about context from lower level information. From this definition, we elaborate a conceptual model of contextual reasoning, which we call the “context pyramid”. The context pyramid describes contextual reasoning as a series of processing steps at different levels, starting with raw sensor data at the base of the pyramid and working up to the peak of the pyramid, where rich context is realized. We argue that machine learning is the primary technique used for contextual reasoning, and we provide several examples of different machine learning techniques, such as naïve Bayes’ classifiers, Hidden Markov Models (HMM), Bayesian Networks, and Support Vector Machines (SVM).

[P3] combines indoor positioning technologies and smartphone sensors to detect different human activities in an office environment. We provide a real-world implementation of the context pyramid on a smartphone, resulting in a contextual reasoning capability, which we call the “cognitive phone”. The key technologies we utilize include ubiquitous positioning, motion recognition, and human behavior modeling. We combine these technologies into a single probabilistic model, which we call the LoMoCo (Location-Motion-Context) model. In this thesis, we demonstrate the feasibility of the fifth level in the context pyramid—Activity-Level Descriptors. Also, the location accuracy we achieved using WLAN-based positioning is about 2–5 meters, depending on the type of space. For motion recognition, we evaluated several different supervised learning techniques, such as decision trees and Linear Discriminant Analysis (LDA), but the best performance was achieved using a Least Squares-Support Vector Machine (LS-SVM) classifier with an average recall rate of 92.9%.

[P4] investigates the use of smartphone sensors, geospatial information, and machine learning to sense mobility contexts, including walking, running, driving and using a bus or train. Our aim was to evaluate techniques that could be used in real-time or near-real-time (<5 s). We also measured the computational complexity of the resulting classifiers because this impacts smartphone battery usage. We investigate a wide range of supervised learning techniques for classification, including decision trees (DT), support vector machines (SVM), naïve Bayes classifiers (NB), Bayesian networks (BN), logistic regression (LR), artificial neural networks (ANN) and several instance-based classifiers (KStar, LWL and IBk). We performed feature selection to identify the most important features from our dataset for detecting mobility context.

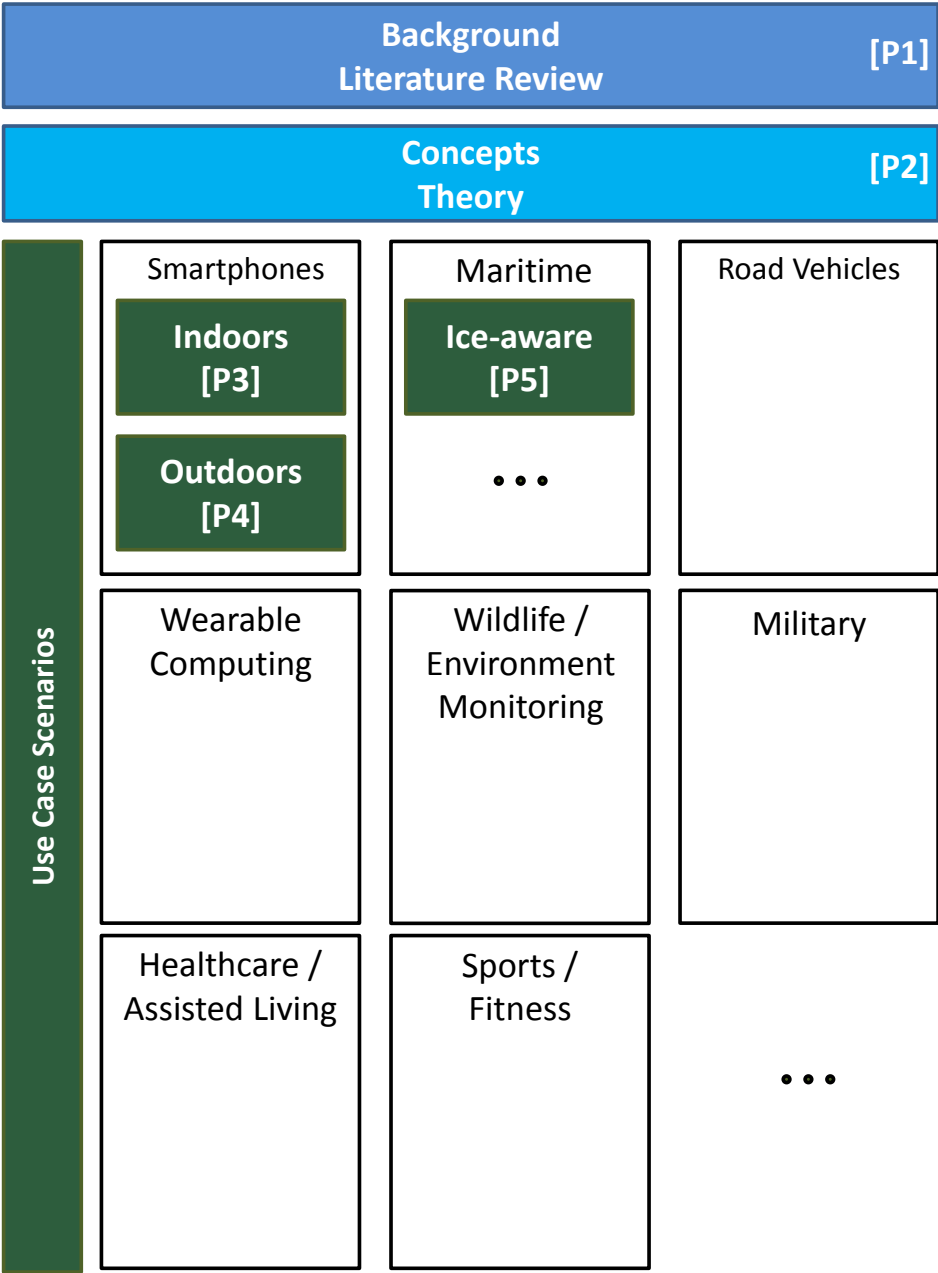
Finally, we focus on the best performing classifier, RandomForest, which is a type of ensemble decision tree algorithm. We tune its parameters to find the optimal performance. This resulted in an average recall rate above 97.5% after tuning. We measure computational complexity in terms of Central Processing Unit (CPU) time needed for classification to provide a relative comparison between the algorithms in terms of battery usage requirements. As a result, we are able to rank the classifiers from lowest to highest complexity as follows: SVM, ANN, LR, BN, DT, NB, IBk, LWL and KStar. The RandomForest algorithm, although it does not generate the simplest classifier in terms of computational cost, provides the best performance with reasonable complexity.

[P5] examines the feasibility of an ice-aware maritime route optimization algorithm and presents a novel method for such purposes. Our aim is to increase the safety and efficiency of maritime transport under icy conditions. Earlier works in this area mainly used numerical methods that could not guarantee global optimum solutions. Our proposed method combines several elements, including (1) a sea spatial model, (2) ship maneuverability model, (3) sea ice model, and (4) ship performance model. Route optimization is performed using the A\* algorithm. The main novelty in this research is the development of an intuitive cost function that takes into account ice conditions and available icebreaker assistance. This research is also the first application of the A\* algorithm to long-range maritime route optimization. We present example results based on the method, using the Baltic Sea as a case study. Generated routes are compared with historical routes under the same ice conditions to provide preliminary validation of the method.

## 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 Figure 4.1 indicates, there are many potential use case scenarios that are not



**Fig. 4.1:** Mapping of included publications to research areas. The additional empty boxes are intended to emphasize that many use case scenarios have yet to be explored. Further discussion on future use case scenarios can be found in Section 5.4.2.

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 one considers [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 detailed 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 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 the case of [P1], the 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 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 first author. The thesis author contributed equally to the design and implementation of the experiments. Implementation of the data collection application

on a smartphone was primarily the responsibility of the thesis author. He also assisted in the analysis of the test results. In addition, 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 on 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 data collection, as well as general guidance from 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 formulated the concept of combining a sea spatial model, ship maneuverability model, sea ice model, and ship performance model. In particular, the ship maneuverability model, which fixes the graph structure and discretizes the ship’s maneuverability, was the thesis author’s own invention. Furthermore, he implemented the optimization algorithm in Matlab, basing the implementation only roughly on an open source implementation. The cost function was developed mainly by the thesis author, although various preliminary ideas were discussed together with the second author. The only parts of the method that were not implemented or provided by the thesis author were the resistive ship speed model, the ice data, and 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.



## 5. CONCLUSIONS

This chapter offers some conclusions based on this thesis, together with the included publications. It is organized as follows: Section 5.1 briefly summarizes the thesis. Section 5.2 outlines our main findings. Section 5.3 explains the significance or potential impact of the results. Section 5.4 describes our future work planned in the areas addressed by this thesis. Finally, Section 5.5 provides a few concluding remarks.

### 5.1 *Summary*

The overall 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 is 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. Three of the included publications ([P3]–[P5]) aim to 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 Hermagoras of Temnos, 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 reasoning.

On the topic of machine learning, this thesis has examined the original goal of machine learning, as envisioned by pioneers such as Arthur Samuel. In the tradition of Samuel and Shannon, we elucidate the concept of *automatic learning* using the toy example of computer chess. We then examine the modern notion of machine learning, including the two major types, supervised learning and unsupervised learning. We provide a tutorial-like example of both types of learning, using an example problem from context awareness. In particular, we have emphasized the importance and benefits of automatic learning. That is, supervised learning usually requires manual labeling of training data, whereas unsupervised learning can largely meet the desire for automated learning, although it often requires some human interpretation of the results.

Finally, the included publications provide further details on machine learning and its application to context awareness, and in particular [P3] and [P4] demonstrate the use of machine learning in practice. Lastly, [P5] provides an example application of context awareness in the field of navigation, i.e. an ice-aware route optimization method.

## 5.2 Main Findings

Below summarizes our main research findings:

- Many aspects of context awareness are feasible to realize using only the sensors in smartphones. For example, we have shown that smartphones can reliably detect different mobility contexts (>97% recall rate) and detect different office-environment activities (>90% recall rate). We have also introduced a method to detect whether a user is indoors or outdoors. Existing algorithms from machine learning, especially supervised learning, provide adequate levels of performance for these use cases, although generalization to large user populations will require the collection of more extensive training data. Nonetheless, context-aware smartphone applications are presently feasible and can be realized using existing machine learning techniques.
- Although 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. After evaluating the performance of 20 different supervised learning algorithms, we determined that the RandomForest algorithm performed the best on our dataset. Our results also showed that performance is optimized only after applying extensive feature selection and parameter tuning. Another general recommendation from our research is that, due to the need to train many different classification models (with various feature sets and parameter settings), a high level of automation for this type of analysis is desirable, and we developed some software tools to improve the automation of this type of analysis.
- 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 human-intensive task of route planning performed by crews of ships sailing in ice covered waters. Our research showed that graph-based approaches are feasible for modeling maritime transportation in ice-covered waters and that

the A\* algorithm can be applied to find optimal paths. In order to realize this implementation of the A\* algorithm, our research presents a simple but novel cost function that takes into account the operational constraints posed by ice breaker assistance. Essentially, this cost function captures contextual information about a ship's theoretical speed through an ice field, taking into account the ship's own ice-breaking performance and possible assistance from an ice breaker.

- Our research suggests not only the feasibility of context-aware applications related to the three use case scenarios investigated, but also that many other applications of context awareness are evident in emerging technologies. We believe context awareness will play an even stronger role in the future, especially in so-called “smart devices” such as smartphones and smartwatches.

### *5.3 Significance of the Results*

This thesis contributes to the overall body of knowledge on context awareness, which as previously discussed, has many potential applications. We have introduced two general and flexible frameworks related to context awareness—one for the systematic encoding of contextual information and the other for the processing of raw sensor data into “rich context”. These frameworks serve as a methodological skeleton on which other researchers and developers can build new context-aware systems.

Much of this thesis focuses on context awareness that can be achieved using only the sensors in smartphones. Because of the widespread prevalence of smartphones in modern society, the results of smartphone-based context awareness have a strong potential for widespread adoption. We can compare this to previous context awareness studies, for example those related to office environments, that rely on custom sensors being installed in the environment; such studies suffer from the limitation that they require installation of new hardware into the environment, increasing the cost of adoption. According to our knowledge, our research is the first to look at office-related context awareness utilizing only smartphone sensors and standard WLAN access points.

Regarding the performance results reported in our mobility context research, it is difficult to determine definitively whether or not our results represent an improvement

over the state-of-the-art. This is due to the fact that we can not compare our results against those obtained with comparable datasets. This issue will be further discussed below in Section 5.4.3. What we can conclude confidently is that extensive analysis, consisting of evaluating different machine learning algorithms, performing feature selection, and systematically tuning parameters of the algorithm, will result in better performance regardless of the dataset. Surprisingly, many research works in context awareness, especially in the navigation community, overlook this fact and report results from evaluating only one or a few machine learning algorithms and without reporting any results of parameter tuning. We hope that the methodology described in this thesis informs other researchers as to the benefits of such detailed analysis.

Lastly, our research demonstrates the feasibility of developing an ice aware maritime navigation system, specifically to provide automatic route optimization. Although further refinement is needed, the proposed methods have the potential to provide significant savings for the maritime transportation system. After presenting these methods to Director-General of the Finnish Ministry of Transportation and Communications, Mr. Pekka Plathan commented that this research has the potential to save billions for the Finnish maritime industry.

Ice-aware route optimization can not only bring economic benefits for the maritime industry, but it also has the potential to provide safety benefits. The methods described in this thesis are flexible in that the cost function can be modified to optimize non-economic factors as well. For example, using models for the risk that ice poses towards damaging ships or getting the ship completely stuck in the ice, one can design an appropriate cost function aimed to minimize these risks.

## 5.4 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. These areas of future work will be discussed in separate sub-sections below.

#### *5.4.1 Future Work in Investigated Applications*

In our research on detecting office-environment contexts, we investigated a small number of different workplace contexts, including working in one's office, having lunch, taking a break, and fetching coffee or water. There are obviously a large number of other workplace contexts that could be investigated, such as having a meeting, giving a presentation, having an impromptu conversation, talking on the phone, etc. In our research, we grouped all "non-defined" contexts into a single category called "undefined context". Also, we conducted this research in only one particular office environment. Extending this research to many diverse office environments and types of work would also improve the robustness of the results.

Another way to improve this line of research would be to expand the sources of raw data to include other sensors. For example, smartphones have microphones and sound could be an important source of contextual information. In fact, we informally explored using audio as a feature, but there are some challenges in this regard. For example, when the phone is in the pocket, the audio signal becomes very muffled and contact with clothing can cause loud undesirable signals. Nonetheless, we believe audio is an important source of contextual information, and we aim to explore this further in the future.

Also, different social context aspects of the workplace environment will be included in the future. In this thesis, we did not address social context at any depth, but especially in an office environment, it should be feasible to recognize different social contexts because in many workplaces the identities of most of the people present is largely known. Using various proximity sensing technologies, a smartphone could apply contextual reasoning about different social contexts, such as "with the boss," "with subordinates," "with colleagues," "with a customer," etc. Such contextual information may not be necessarily needed for navigation applications, but it would certainly have other applications related to mobile computing.

On the subject of mobility contexts, we also plan to expand the range of contexts under investigation, such as cycling, riding trams, riding metros, etc. In addition, we plan to investigate whether we can recognize a number of other mobility-related

leisure activities, such as hiking, berry or mushroom picking<sup>1</sup>, playing golf, dog-walking, etc.

Also, in this thesis, for detecting mobility context we only utilized two smartphone sensor types, namely GPS and accelerometers. In the future, we aim to include other types of sensors, such as gyroscopes, pressure sensors, microphones, and light sensors.

Lastly, we plan to expand the number of test subjects participating in the collection of training data. This is important to ensure that the trained models are robust. In this thesis, we asked the test subjects to keep the smartphone in a particular location (pants pocket), so in future research we will also study the effects of placement of the smartphone in different locations, such as a backpack, handbag, belt “holster,” etc.

On the subject of ice-aware route optimization, further work is needed to validate the proposed method. This should include further analysis of historical data from Automatic Identification System (AIS), as well as simulator-based studies and actual testing of routes at sea. Also, in this thesis the route optimization method focused on minimizing travel times for ships, but in the future other aspects should be investigated, such as fuel usage, operational efficiency, safety, and reliability. Lastly, contextual information regarding the ice conditions should be enriched compared to the model used in this thesis. For example, the current model does not take into account ice compression, which can have a large effect on ice-going ship performance.

#### 5.4.2 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 system” 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

---

<sup>1</sup> These are popular leisure activities in Finland.

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.

#### *5.4.3 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]). While supervised learning has many advantages compared to unsupervised learning,



it can be very costly and time-consuming to generate the required labeled data. 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], and there are a few other examples in the literature (e.g. [74] [115]). Generally, this is not a common practice in context awareness research. 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 [116]).

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. Table 2.1 demonstrates this problem.

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. Many context ontologies have been proposed in the literature (e.g. [117] [118] [119] [120] [121]), and one of these could form the basis of a context ontology standard. 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. The problem is that no forerunners for a

standard have emerged and no good software tools for working with the proposed ontologies have been made available. There are many possible reasons for this, and we would not like to speculate too deeply in this regard. It suffices to say that in our future work we aim to contribute to and advance the notion of standard context ontologies, including open source tools for working with such 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 always realistic.

The solution is straight-forward. As an alternative, context researchers should always 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. It is, however, a common practice in the machine learning community to test techniques against benchmark data. For example, the University of California, Irvine (UCI) maintains a repository of over 300 datasets that can be used for machine learning research [122]. Unfortunately, very few of these datasets relate to context awareness. A few other sources of open data for context awareness research exist, including the Mobile Data Challenge (MDC) Dataset collected as part of the Lausanne Data Collection Campaign [123] and data from the University of Helsinki's "Context project" [124]. We aim to follow open data practices in our future work and also to actively promote this practice, either by promoting the use of UCI's machine learning repository or by setting up a dedicated portal for context awareness research.

### *5.5 Concluding Remarks*

The remainder of this thesis consists of reprints of the five included publications described earlier. The order of the publications has been chosen to go from the most

---

general to more specific and detailed applications. They can be read, however, in any order, depending on the reader's specific interests.



## BIBLIOGRAPHY

- [1] T. Mitchell, B. Buchanan, G. DeJong, T. Dietterich, P. Rosenbloom, and A. Waibel, "Machine learning," *Annual Review of Computer Science*, vol. 4, pp. 417–433, 1990.
- [2] E. Brynjolfsson and A. McAfee, *The second machine age: work, progress, and prosperity in a time of brilliant technologies*. New York: W. W. Norton & Company, 2014.
- [3] Canalys.com, Ltd., "64 million smart phones shipped worldwide in 2006." Available online: <http://bit.ly/1HSNZBZ>, 2007.
- [4] Gartner, "Gartner says worldwide smartphone sales reached its lowest growth rate with 3.7 per cent increase in fourth quarter of 2008," March 2009.
- [5] N. Mawston, "Worldwide smartphone population tops 1 billion in q3 2012." Available online: <http://bit.ly/1DZuyYo>, October 2012. (archived on 26 February 2015).
- [6] Korea Times, "Smartphone users expected to hit 2.5 billion next year." Available online: <http://bit.ly/1HSO5cQ>.
- [7] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon, "A proposal for the Dartmouth summer research project on artificial intelligence," *AI Magazine*, 2006. Reprint of proposal originally written in 1955.
- [8] D. Crevier, *AI: The tumultuous history of the search for artificial intelligence*. Basic Books, Inc., 1993.
- [9] P. D. Groves, L. Wang, D. Walter, H. Martin, K. Voutsis, and Z. Jiang, "The four key challenges of advanced multisensor navigation and positioning," in

- Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, pp. 773–792, IEEE, 2014.
- [10] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, and L. T. Yang, “Rich mobile applications: genesis, taxonomy, and open issues,” *Journal of Network and Computer Applications*, vol. 40, pp. 345–362, 2014.
- [11] Y. Manabe, H. Saito, K. Akiyama, R. Ikeda, S. Kanda, and K. Sugawara, “Perceptual functions for context-awareness of an office worker,” in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pp. 583–589, IEEE, 2010.
- [12] R. E. Guinness, “Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors,” in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, (Nashville, TN), pp. 2868–2879, September 2013.
- [13] J.-H. Nam, I. Park, H. J. Lee, M. O. Kwon, K. Choi, and Y.-K. Seo, “Simulation of optimal arctic routes using a numerical sea ice model based on an ice-coupled ocean circulation method,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 5, no. 2, pp. 210–226, 2013.
- [14] M. Choi, H. Chung, H. Yamaguchi, and K. Nagakawa, “Arctic sea route path planning based on an uncertain ice prediction model,” *Cold Regions Science and Technology*, vol. 109, pp. 61–69, 2015.
- [15] V. Kotovirta, R. Jalonon, L. Axell, K. Riska, and R. Berglund, “A system for route optimization in ice-covered waters,” *Cold Regions Science and Technology*, vol. 55, no. 1, pp. 52–62, 2009.
- [16] P. W. Foltz, W. Kintsch, and T. K. Landauer, “The measurement of textual coherence with latent semantic analysis,” *Discourse Processes*, vol. 25, no. 2, pp. 285–307, 1998.
- [17] M. Bazire and P. Brézillon, “Understanding context before using it,” in *Modeling and Using Context: Lecture Notes in Computer Science* (A. Dey, B. Kokinov, D. Leake, and R. Turner, eds.), Springer Berlin/Heidelberg, 2005.

- 
- [18] J. McCarthy, "Notes on formalizing context," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, (San Mateo, California), pp. 555–562, Morgan Kaufmann, 1993.
- [19] R. Chen and R. E. Guinness, *Geospatial computing in mobile devices*. Boston: Artech House, 2014.
- [20] "Context." *Merriam-Webster Dictionary*, Available online: <http://bit.ly/1LK7K2O>, 2015.
- [21] V. Akman and M. Surav, "Steps toward formalizing context," *AI magazine*, vol. 17, no. 3, p. 55, 1996.
- [22] Wikipedia, "Five ws — wikipedia, the free encyclopedia." Available online: <http://bit.ly/1SIXRkI>, 2015.
- [23] B. S. Bennett, "Hermagoras of temnos," *Classical Rhetorics and Rhetoricians: Critical Studies and Sources*, p. 187, 2005.
- [24] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, no. 5, pp. 22–32, 1994.
- [25] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.
- [26] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.
- [27] T. Moran and P. Dourish, "Introduction to this special issue on context-aware computing," *Human-Computer Interaction*, vol. 16, no. 2-4, pp. 87–95, 2001.
- [28] J. McCarthy, *Programs with common sense*. Defense Technical Information Center, 1963.
- [29] R. V. Guha, *Contexts: a formalization and some applications*, vol. 101. Stanford University Stanford, CA, 1991.
- [30] J. McCarthy and S. Buvac, "Formalizing context (expanded notes)," in *Computing Natural Language, CSLI, Lecture Notes* (A. Aliseda, R. J. van

- Glabbeek, and W. D., eds.), pp. 13–50, Center for the Study of Language and Information, Stanford University, 1998.
- [31] S. Buvač and I. A. Mason, “Propositional logic of context,” in *Proceedings of the eleventh national conference on artificial intelligence*, sn, 1993.
- [32] P. Brézillon, “Context in artificial intelligence: I. a survey of the literature,” *Computers and artificial intelligence*, vol. 18, pp. 321–340, 1999.
- [33] V. Akman, “Context in artificial intelligence: a fleeting overview,” in *La svolta contestuale* (C. Penco and V. Akman, eds.), Milano: McGraw-Hill, 2002.
- [34] J.-y. Hong, E.-h. Suh, and S.-J. Kim, “Context-aware systems: A literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.
- [35] M. Tähti, V.-M. Rautio, and L. Arhippainen, “Utilizing context-awareness in office-type working life,” in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pp. 79–84, ACM, 2004.
- [36] M. Heiskala, E. Palomäki, M. Vartiainen, K. Hakkarainen, and H. Muukkonen, “A research framework for the smartphone-based contextual study of mobile knowledge work,” in *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, pp. 246–257, Springer, 2014.
- [37] X. Huang, J. Weng, and Z. Zhang, “Office presence detection using multimodal context information,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP’04). IEEE International Conference on*, vol. 3, pp. iii–773, IEEE, 2004.
- [38] M. Danninger and R. Stiefelhagen, “A context-aware virtual secretary in a smart office environment,” in *Proceedings of the 16th ACM international conference on Multimedia*, pp. 529–538, ACM, 2008.
- [39] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, IEEE, 1999.



- 
- [40] Y. Manabe, H. Saito, K. Akiyama, R. Ikeda, S. Kanda, and K. Sugawara, "Perceptual functions for context-awareness of an office worker," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pp. 583–589, IEEE, 2010.
- [41] K. Kiyokawa, M. Hatanaka, K. Hosoda, M. Okada, H. Shigeta, Y. Ishihara, F. Ooshita, H. Kakugawa, S. Kurihara, and K. Moriyama, "Owens luisăĂŢa context-aware multi-modal smart office chair in an ambient environment," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pp. 1–4, IEEE, 2012.
- [42] K. K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. J. Rentfrow, "Smartphone sensing offloading for efficiently supporting social sensing applications," *Pervasive and Mobile Computing*, vol. 10, pp. 3–21, 2014.
- [43] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Ambient Intelligence*, pp. 220–232, Springer, 2003.
- [44] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *Ubiquitous Computing Systems*, pp. 516–527, Springer, 2006.
- [45] H. Park, J. Park, H. Kim, J. Jun, S. Hyuk Son, T. Park, and J. Ko, "Relisce: Utilizing resource-limited sensors for office activity context extraction," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, pp. 1151–1164, Aug 2015.
- [46] M. J. Duncan, H. M. Badland, and W. K. Mummery, "Applying gps to enhance understanding of transport-related physical activity," *Journal of Science and Medicine in Sport*, vol. 12, no. 5, pp. 549–556, 2009.
- [47] S. A. Hoseini-Tabatabaei, A. Gluhak, and R. Tafazolli, "A survey on smartphone-based systems for opportunistic user context recognition," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 27, 2013.
- [48] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 47, 2015.

- [49] Y. Kwon, K. Kang, and C. Bae, “Unsupervised learning for human activity recognition using smartphone sensors,” *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.
- [50] F. Foerster and J. Fahrenberg, “Motion pattern and posture: correctly assessed by calibrated accelerometers,” *Behavior research methods, instruments, & computers*, vol. 32, no. 3, pp. 450–457, 2000.
- [51] S.-W. Lee and K. Mase, “Activity and location recognition using wearable sensors,” *IEEE pervasive computing*, vol. 1, no. 3, pp. 24–32, 2002.
- [52] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, “A hybrid discriminative/generative approach for modeling human activities,” in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 766–772, 2005.
- [53] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Proceedings of the national conference on artificial intelligence*, vol. 20, pp. 1541–1546, July 2005.
- [54] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, “Activity classification using realistic data from wearable sensors,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 119–128, 2006.
- [55] J. Suutala, S. Pirttikangas, and J. Rääning, “Discriminative temporal smoothing for activity recognition from wearable sensors,” in *Ubiquitous Computing Systems* (H. Ichikawa, W.-D. Cho, I. Satoh, and H. Youn, eds.), vol. 4836 of *Lecture Notes in Computer Science*, pp. 182–195, Springer Berlin Heidelberg, 2007.
- [56] K. Kunze and P. Lukowicz, “Dealing with sensor displacement in motion-based onbody activity recognition systems,” in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 20–29, ACM, 2008.
- [57] J. Yin, Q. Yang, and J. Pan, “Sensor-based abnormal human-activity detection,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 8, pp. 1082–1090, 2008.

- 
- [58] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–10, ACM, 2009.
- [59] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 13:1–13:23, 2010.
- [60] L. Pei, R. Chen, J. Liu, W. Chen, H. Kuusniemi, T. Tenhunen, T. Kröger, Y. Chen, H. Leppäkoski, and J. Takala, "Motion recognition assisted indoor wireless navigation on a mobile phone," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pp. 3366–3375, 2010.
- [61] K. Frank, M. Nades, P. Robertson, and M. Angermann, "Reliable real-time recognition of motion related human activities using mems inertial sensors," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pp. 2919–2932, 2001.
- [62] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gis information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 54–63, November 2011.
- [63] L. Pei, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, T., and Tenhunen, "Using motion-awareness for the 3d indoor personal navigation on a smartphone," in *Proceedings of the 24th International Technical Meeting of The Institute of Navigation (ION GNSS 2011)*, Portland, Oregon, USA, September 20-23, 2011.
- [64] M. Susi, D. Borio, and G. Lachapelle, "Accelerometer signal features and classification algorithms for positioning applications," in *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, pp. 158–169, 2011.
- [65] J. B. Bancroft, D. Garrett, and G. Lachapelle, "Activity and environment

- classification using foot mounted navigation sensors,” in *Proceedings of the 2012 Indoor Positioning and Indoor Navigation (IPIN) Conference*, 2012.
- [66] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine,” in *Ambient assisted living and home care*, pp. 216–223, Springer, 2012.
- [67] M. Susi, V. Renaudin, and G. Lachapelle, “Motion mode recognition and step detection algorithms for mobile phone users,” *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.
- [68] T. Feng and H. J. Timmermans, “Transportation mode recognition using gps and accelerometer data,” *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 118–130, 2013.
- [69] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 13, ACM, 2013.
- [70] L. Stenneth, *Detecting Human Activities Using Smartphones and Maps*. Ph.D. thesis, University of Illinois at Chicago, February 2014.
- [71] H. Xia, Y. Qiao, J. Jian, and Y. Chang, “Using smart phone sensors to detect transportation modes,” *Sensors*, vol. 14, no. 11, pp. 20843–20865, 2014.
- [72] M. Elhoushi, J. Georgy, M. Korenberg, and A. Noureldin, “Robust motion mode recognition for portable navigation independent on device usage,” in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, pp. 158–163, IEEE, 2014.
- [73] J. Parviainen, J. Bojja, J. Collin, J. Leppänen, and A. Eronen, “Adaptive activity and environment recognition for mobile phones,” *Sensors*, vol. 14, no. 11, pp. 20753–20778, 2014.
- [74] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, “Big data small footprint: The design of a low-power classifier for detecting transportation modes,” *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

- 
- [75] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, "Using mobile phone barometer for low-power transportation context detection," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pp. 191–205, ACM, 2014.
- [76] J.-H. Chiang, P.-C. Yang, and H. Tu, "Pattern analysis in daily physical activity data for personal health management," *Pervasive and Mobile Computing*, vol. 13, pp. 13–25, 2014.
- [77] J.-M. Yu and S.-B. Cho, "A low-power context-aware system for smartphone using hierarchical modular Bayesian networks," in *Hybrid Artificial Intelligent Systems*, pp. 543–554, Springer, 2015.
- [78] M. Choi, H. Chung, H. Yamaguchi, and L. W. A. De Silva, "Application of genetic algorithm to ship route optimization in ice navigation," in *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*, 2013.
- [79] I. H. Park, J. H. Nam, and K. S. Choi, "A graphical approach to determine the optimal sea route of icebreakers in arctic region," in *Proceedings of the Korean Association of ocean science and technology societies join conference*, June 2011. Published in Korean.
- [80] J. Esa, "Fuel and economic efficiency of an ice-going vessel on the northern sea route," Master of Science thesis, Aalto University, June 2015.
- [81] J. Montewka, F. Goerlandt, P. Kujala, and M. Lensu, "Towards probabilistic models for the prediction of a ship performance in dynamic ice," *Cold Regions Science and Technology*, vol. 112, pp. 14–28, 2015.
- [82] D. LaPrairie, M. Wilhelmson, and K. Riska, *A transit simulation model for ships in Baltic ice conditions : Documentation of the calculation routine*. Helsinki University of Technology, 1995.
- [83] P. Valanto, S. J. Jones, E. Enkvist, and K. Izumiyama, "The resistance of ships in level ice," *Transactions of the Society of Naval Architects and Marine Engineers*, vol. 109, pp. 53–83, 2001.

- [84] B. H. Fock, A. Beitsch, D. Broehan, M. Dobrynin, A. Gierisch, L. Kaleschke, T. Pohlmann, and K. H. Schlünzen, “Ice forecast and route optimization,” 2012. Poster presented at the SMOSIce User Workshop, Hamburg, Germany.
- [85] M. Dobrynin, B. H. Fock, A. M. Gierisch, T. Pohlmann, L. Kaleschke, H. Schlünzen, *et al.*, “Prediction of arctic sea ice for ship routing: Forecast experiment and ship cruise,” in *OTC Arctic Technology Conference*, Offshore Technology Conference, 2015.
- [86] J. McCarthy, “Generality in artificial intelligence,” *Communications of the ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.
- [87] E. W. Weisstein, “Term.” *MathWorld—A Wolfram Web Resource*, Available online: <http://mathworld.wolfram.com/Term.html>.
- [88] C. Ghidini and F. Giunchiglia, “Local models semantics, or contextual reasoning= locality+ compatibility,” *Artificial intelligence*, vol. 127, no. 2, pp. 221–259, 2001.
- [89] R. H. Thomason, “Contextual intensional logic: Type-theoretic and dynamic considerations,” *Perspectives on contexts*, 2006.
- [90] D. M. Gabbay and R. T. Nossut, “Structured contexts with fibred semantics,” in *Formal Aspects of Context*, pp. 193–209, Springer, 2000.
- [91] P. Bouquet, C. Ghidini, F. Giunchiglia, and E. Blanzieri, “Theories and uses of context in knowledge representation and reasoning,” *Journal of Pragmatics*, vol. 35, no. 3, pp. 455–484, 2003.
- [92] P. Bouquet and L. Serafini, “Two formalizations of context: A comparison,” in *Modeling and Using Context* (V. Akman, P. Bouquet, R. Thomason, and R. Young, eds.), vol. 2116 of *Lecture Notes in Computer Science*, pp. 87–101, Springer Berlin Heidelberg, 2001.
- [93] G. Hirst, “Context as a spurious concept,” *arXiv preprint cmp-lg/9712003*, 1997.
- [94] L. Wolf and S. Bileschi, “A critical view of context,” *International Journal of Computer Vision*, vol. 69, no. 2, pp. 251–261, 2006.

- 
- [95] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," *Readings in artificial intelligence*, pp. 431–450, 1969.
- [96] J. McCarthy, "Some expert systems need common sense," *Annals of the New York Academy of Sciences*, vol. 426, no. 1, pp. 129–137, 1984.
- [97] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.
- [98] C. E. Shannon, "Xxii. programming a computer for playing chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950.
- [99] A. Turing, *Chess*, ch. The Essential Turing, *Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma*. Oxford University Press, 2004. Reprint of paper originally printed in 1953.
- [100] F.-H. Hsu, *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press, 2002.
- [101] D. Spicer and K. Tashev, "The quest to build a thinking machine: A history of computer chess," *CompuComputer: New exhibit showcases game's past and museum's future*, May 2006. A publication of the Computer History Museum.
- [102] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Prentice Hall, 3rd edition ed., 2010.
- [103] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, no. 7, pp. 1341–1390, 1996.
- [104] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.
- [105] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [106] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [107] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

- [108] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [109] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Applied statistics*, pp. 100–108, 1979.
- [110] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS: Ordering points to identify the clustering structure,” *SIGMOD Rec.*, vol. 28, pp. 49–60, June 1999.
- [111] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [112] J. A. Bilmes *et al.*, “A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models,” *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.
- [113] N. Vlassis and A. Likas, “A greedy EM algorithm for gaussian mixture learning,” *Neural processing letters*, vol. 15, no. 1, pp. 77–87, 2002.
- [114] P. Smyth, “Note set 4: Finite mixture models and the EM algorithm.” Lecture notes for course on Probabilistic Learning at University of California–Irvine, Available online: <http://www.ics.uci.edu/smyth/courses/cs274/notes/notes4.pdf>, 2015.
- [115] F. J. Ordóñez, P. de Toledo, and A. Sanchis, “Activity recognition using hybrid generative/discriminative models on home environments using binary sensors,” *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.
- [116] R. E. Guinness, H. Kuusniemi, J. Vallet, T. Sarjakoski, J. Oksanen, M. Islam, M. Syeed, H.-M. Halkosaari, P. Kettunen, M. Laakso, and M. RåŹnneberg, “MyGeoTrust: A platform for trusted crowdsourced geospatial data,” in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, (Tampa, Florida), 2015.
- [117] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, and A. Kriouile, “An ontology oriented architecture for context aware services adaptation,” *arXiv preprint arXiv:1404.3280*, 2014.



- 
- [118] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 18, no. 03, pp. 197–207, 2003.
- [119] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using owl," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pp. 18–22, Ieee, 2004.
- [120] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," in *Proceedings of communication networks and distributed systems modeling and simulation conference*, vol. 2004, pp. 270–275, 2004.
- [121] T. Strang, C. Linnhoff-Popien, and K. Frank, "Cool: A context ontology language to enable contextual interoperability," in *Distributed applications and interoperable systems*, pp. 236–247, Springer, 2003.
- [122] M. Lichman, "UCI machine learning repository." Available online: <http://archive.ics.uci.edu/ml/>, 2015.
- [123] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, *et al.*, "The mobile data challenge: Big data for mobile computing research," in *Pervasive Computing*, no. EPFL-CONF-192489, 2012.
- [124] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, "Contextphone: a prototyping platform for context-aware mobile applications," *Pervasive Computing, IEEE*, vol. 4, pp. 51–59, Jan 2005.