UNIVERSIDADE Ð COIMBRA

Gonçalo dos Santos Martins

TOWARDS AUTONOMOUS INTERACTION FOR USER-ADAPTIVE SOCIAL ROBOTS

PhD Thesis on the scientific field of Electrical and Computer Engineering, Specialization in Automation and Robotics, supervised by Professor Jorge Manuel Miranda Dias and presented to the department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Coimbra.

August 2018

Gonçalo dos Santos Martins
TOWARDS AUTONOMOUS INTERACTION FOR USER-ADAPTIVE SOCIAL ROBOTS
UNIVERSIDADE Ð COIMBRA

Towards Autonomous Interaction for User-Adaptive Social Robots

Gonçalo dos Santos Martins

PhD Dissertation on the scientific field of Electrical and Computer Engineering, Specialization in Automation and Robotics, presented to the department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Coimbra.

Supervisor
Jorge Manuel Miranda Dias, Hab., Ph.D.

Coimbra, August 2018

Acknowledgements

While reflecting my own creative, investigative and experimental work, this document portrays also my growth as a person, as a scientist and as a professional, which would certainly not have been possible without the crucial intervention of some people. I will now long-windedly try to honour them all, with my deepest apologies to those I have inevitably forgotten. This work is dedicated to all of them.

The interventions of Prof. Jorge Dias and Dr. Luís Santos, my supervisors, allowed me to enjoy the most intense and fulfilling professional period of my career. I would like to thank Prof. Jorge Dias for his support, rigour, clear-headedness, creativity and absolutely unbounded ambition; I only hope to have absorbed enough of these qualities during these brief years. To Luís, who first convinced me to pursue this madness, who was unrelenting in his support for my work and whose thoroughness, availability and quick thinking shaped most of the ideas and arguments developed throughout the work. He also taught me, unwittingly I believe, that there is no problem without a solution; there is always a solution, even if only a terrible one to be used as a stepping stone.

To my mother, whose ambition, honesty, kindness and determination I can only hope to fall short of. To my girlfriend Filipa her support, and for putting up with how incredibly boring I can be for all these years. To my sister Beatriz, grandma Teté, my father Augusto, and Zé for their unconditional love and support throughout this journey. To my (step)father Mi, who inspired me to become the man and scientist I am today.

To the AP4ISR team for the warmth, companionship and mentoring I received from its many staff, researchers and students for the past five years. I truly wish the best for the team, and may it continue to foster personal and professional growth in an open-minded, free, creative and nurturing environment.

To the GrowMeUp project[1], who supported me financially, and also to the GrowMeUp team itself. To my colleagues at the various partners, with a special note to the kind people of Cáritas Diocesana de Coimbra, in the persons of Carina Dantas, Ana Luísa Jegundo, Flávia Rodrigues, all of the Elderly users who tested the various systems, and so many others. To Profs. Ana Paula Matos, José Joaquim Costa, Manuela Vilar and Mário Zenha-Rela for their insight and support in the Psychbot sub-project. Their cooperation was irreplaceable in the elaboration of some crucial parts of this work, and their kindness and friendliness will not be forgotten.

Lastly, to my friends, luckily too many to name individually. To Tiago, César, Jorge, Fábio, Daniela, Elísio, Fernando and so many, many others who, in one way or another, supported and influenced me in the course of this work. To Raquel, for her irreplaceable cooperation in the

[1]GrowMeUp was funded by European Union's Horizon 2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement Nº 643647

Psychbot sub-project, for the stimulating discussions and for keeping me on my toes with her eternally-sceptical opinions about robots. In a world where robots are generally being accepted without much question, it is important to find someone who with the bravery of facing apparent certainty with a critical mind.

My (short) experience so far has taught me that, as much as one may try, there is no truly individual work. In one form or another, positively or negatively, with a deeper or shallower impact, other people's influences creep into our very being. By our nature we enrich each other, inspiring love, pain, happiness and regret that unwittingly drips into our every thought and, therefore, action. To all of you who have cared to influence me, you have my thanks.

Abstract

This Thesis studies the methodologies and effects of social robots that automatically adapt their actions to the user they are interacting with. We aim to survey, study and improve the state of the art in user-adaptive interaction achieved by current social robots, with the ultimate goal of enhancing their autonomy in interaction.

The work is split into four main parts. Part I aims to provide insight into the problem at hand, providing the motivation and theoretical background needed. The state of the art in the field is surveyed through the analysis of recent works, which are categorized into a novel taxonomy and systematically compared, taking into account aspects such as their usage of user models and their experimental maturity. This survey culminates in the determination of a number of key research and technological gaps, which motivate the remainder of the work.

In Part II, we develop mechanisms that allow for the correct characterization of the user. We start by presenting a technique, BUM, which allows for the learning and storage of information on a population of users using information gathered from heterogeneous, distributed sources. In order to establish the relationship between evidence signals and true user characteristics, the Psychbot study is then presented, wherein a social robot is used to extract the psychological characteristics of its users through adapted interaction. This study involved a sample of elderly participants, allowing us to demonstrate with statistical significance that the answers provided to psychological scales by these users to a robot and to a human evaluator are correlated. Ending this part, we present a surprisal-based dataset reduction technique, which is aimed at reducing the amount of data that techniques such as BUM need to process in order to achieve acceptable results. This technique is tested on benchmark datasets, achieving competitive performance against state-of-the-art approaches.

We start Part III by presenting a set of preliminary experiments performed with a preliminary approach to user-adaptive decision making. The technique presented is based on the concepts of satisfaction and expectation, and aims to learn which action the user expects in which context. A number of flaws with this technique are found, which serve as motivation for a refined version of it. Then we present αPOMDP and βPOMDP, POMDP-based decision-making techniques able to gauge their own impact on the user through a novel State Value Reward (SVR), and learn from it using a novel operation loop.

We then proceed, in Part IV, with an evaluation of the success of the work, framing it into the original research gaps. Each individual contribution is matched to the research gaps it closes, and also to the tangible outcomes it has led to. This analysis is used to derive a set of opportunities for future work, with which we end the monograph.

Keywords: User-Adaptive Robots, User Modelling, Decision Making, Automated Planning, Machine Learning, Dataset Reduction

Resumo

Esta Tese estuda as metodologias usadas e os efeitos provocados por robots sociais que adaptam automaticamente as suas acções ao utilizador com quem estão a interagir. Procuramos examinar, estudar e melhorar o estado da arte em interacção adaptativa ao utilizador actualmente conseguida por robots sociais, almejando melhorar a sua autonomia na interacção.

O trabalho divide-se em quarto partes. A Parte I introduz o problema, providenciando a motivação e bases teóricas necessárias. O estado da arte na área é exposto através da análise de trabalhos recentes, que são categorizados numa nova taxonomia e comparados de acordo com a utilização de modelos de utilizador, maturidade experimental, *etc.* Este trabalho resulta num conjunto de lacunas tecnológicas e científicas que motivam o restante trabalho.

A Parte II foca-se nos mecanismos que permitem a caracterização automática do utilizador. Começamos por apresentar o BUM, que permite a aprendizagem e armazenamento de características de uma população de utilizadores, utilizando informação recolhida por fontes heterogéneas e distribuídas. De modo a estabelecer uma relação entre os sinais de entrada e as características reais do utilizador, apresentamos o estudo Psychbot, no qual um robot social é usado para extrair as características psicológicas dos seus utilizadores. Este estudo envolveu uma amostra de participantes idosos, permitindo-nos demonstrar que existe correlação entre as respostas dadas ao robot e a um avaliador humano no contexto de uma avaliação psicológica. Terminando esta Parte, apresentamos uma nova técnica de redução de dados baseada na auto-informação, reduzindo a quantidade de dados de que técnicas como o BUM necessitam para obter resultados aceitáveis. Esta técnica é testada em dados de *benchmark*, conseguindo um desempenho competitivo quando comparada com técnicas do estado da arte.

Iniciamos a Parte III com um conjunto de experiências envolvendo uma primeira abordagem à tomada de decisão. A técnica baseia-se nos conceitos de expectativa e satisfação, e procura aprender qual a acção que o utilizador espera em cada contexto. Esta técnica sofre de um conjunto de falhas, que são identificadas e servem de motivação para o desenvolvimento de uma versão refinada. Apresentamos de seguida o αPOMDP e βPOMDP, técnicas de tomada de decisão baseada em POMDPs capazes de avaliar o impacto das suas acções no utilizador através de um novo mecanismo de recompensa, e de aprender a partir desta informação.

A Parte IV apresenta uma avaliação do sucesso do trabalho, enquadrando-o nas lacunas encontradas inicialmente. É feita a correspondência entre cada contribuição e as lacunas para cuja colmatagem contribuem, assim como os resultados tangíveis a que levou. Esta análise é usada para derivar um conjunto de oportunidades para trabalho futuro, que fecham o documento.

Palavras-Chave: Robots Adaptativos ao Utilizador, Modelação de Utilizadores, Tomada de Decisão, Planeamento Automático, Aprendizagem Automática, Redução de Dados

Contents

List of Acronyms

AAL	Ambient Assisted Living
API	Application Programming Interface
APP	Automatic Personality Perception
APR	Automatic Personality Recognition
APS	Automatic Personality Synthesis
BI	Bayesian Inference
BUM	Bayesian User Model
GCD	Global Characteristic Definition
HAL	Hardware Abstraction Layer
HCI	Human-Computer Interaction
HRI	Human-Robot Interaction
IJCNN	International Joint Conference on Neural Networks
IR	Interaction Regulation
I/O	Input/Output
ISVR	State-Value Reward with Information Term
LED	Light-Emitting Diode
LRF	Laser Range Finder
MBRL	Model-Based Reinforcement Learning
MDP	Markov Decision Process
MAE	Mean Absolute Error
MSVR	Multiple State-Value Reward

NN	Neural Network
PDDL	Planning Domain Definition Language
PIR	Passive Infra-Red (Sensor)
POMDP	Partially-Observable Markov Decision Process
RGB	Red Green Blue
RGBD	Red Green Blue Depth
ROS	Robot Operating System
SARSOP	Successive Approximations of the Reachable Space under Optimal Policies
SVM	Support Vector Machine
SVR	State-Value Reward
TRL	Technology Readiness Level
UAB	User-Adaptive Behaviour
UC	University of Coimbra
USPS	United States Postal Service
YAML	YAML Ain't Markup Language

Notation

Symbol	Description
CHAPTER 3: BUM: BAYESIAN USER MODELLING	
$P(x)$	The probability of proposition x.
$\mathbf{C}$	A vector of the user's characteristics. Usually, $\mathbf{C} \in \mathbb{R}^n$, *i.e.* $\mathbf{C}$ is a vector of real numbers.
C_i	An element of the $\mathbf{C}$ vector.
$\mathbf{E}$	A vector of evidence for user characteristics estimation. Usually, $\mathbf{E} \in \mathbb{R}^m$, *i.e.* $\mathbf{E}$ is a vector of real numbers.
E_i	An element of the $\mathbf{E}$ vector
Id	The user's ID, a unique number that identifies each user. $Id \in \mathbb{N}$, *i.e.* Id is an integer.
$\mathbf{U}_u$	A vector containing the the user u's characteristics obtained through *maximum a posteriori* (MAP), defining a point in characteristics space. In essence, an instantiation of the $\mathbf{C}$ vector for a particular user.
T_i	A tuple containing the necessary information for fusion: L_i, $\mathbf{E}$, and h_i
$\mathbf{T}$	A vector of T_i tuples.
L_i	The label obtained for characteristic i via MAP.
h_i	The entropy of distribution $P(\mathbf{C}_i))$, *i.e.* the posterior distribution over all users for characteristic i.
D	A manually-tuned learning factor, $D \in \mathbb{R}$.
ψ	A normalization factor ensuring the validity of probability distributions.
μ	The mean of a Gaussian distribution.
σ	The standard deviation of a Gaussian distribution.
M	The means of a Gaussian mixture.
Σ	The covariance matrix of a Gaussian mixture.

Continued on next page

Continued from previous page

Symbol	Description
d_i	The Euclidean norm in characteristics space from a given point to cluster i.
D_{KL}	The Kullback–Leibler divergence.
ϵ	Estimation error.
Chapter 4: Psychbot: Psychological Evaluation using a Social Robot	
A	Bag-of-words obtained from the user's answer.
B_i	Bag-of-words corresponding to the an item's possible answer i.
D_i	The distance between A and B_i.
mae	The mean absolute error of the robot's estimation.
$a_{i,human}$, $a_{i,robot}$	Normalized answers ($a_i \in [0, 1]$) provided by the user to the human and robot evaluators, respectively.
r_p	The Pearson correlation coefficient, denoting how strong the relationship between two random variables is.
r_s	The Spearman correlation coefficient, denoting how strong the relationship between two random variables is.
Chapter 5: Surprisal-Based Dataset Reduction	
$I(X = x)$	Surprisal resulting from drawing example x from a random variable X.
$\mathbf{S}$	A sequence of labelled examples for training, the "dataset".
$s_k = \{x_k, y_k\}$	The k-th element of $\mathbf{S}$, containing all features x_k and the resulting label y_k.
τ	A surprisal threshold. Training examples below this threshold are discarded for not being informative enough.
$\mathbf{N}$	The set of samples deemed necessary for training. $\mathbf{N}_k$ denotes the set of necessary examples when observing the sequence $\mathbf{S}$ up to element k, and $\mathbf{N}_n$ the final reduced dataset obtained after observing the full n elements of $\mathbf{S}$.
μ	(unchanged from previous chapters)
σ	(unchanged from previous chapters)
A	Classification accuracy obtained when operating on a given reduced dataset.

Continued on next page

Continued from previous page

Symbol	**Description**
T	Time, in seconds, necessary to train the system with a given reduced dataset.
D	The fraction of examples classified as needed.
v	Input features of the UC-3D dataset.
c	Output labels of the UC-3D dataset, denoting the action descriptors.
Chapter 6: Preliminary Experiments on User-Adaptive Decision Making	
χ	A set of possible actions, the robot's action space.
x_i	An individual possible action.
$\mathbf{C}$	The context in which the interaction occurs, composed of information about the user and information about the world.
$\mathbf{U}$	The available information about the user, a vector of integers.
u_i	An individual variable containing information about the user.
$\mathbf{W}$	The available information about the world, also a vector of integers.
w_i	An individual variable containing information about the world.
ξ	The user's satisfaction level, $\xi \in [0, 1]$.
$\mathbf{R}$	The user's reaction to the robot's action, a vector in integer variables.
r_i	An individual variable pertaining to the user's reaction to the robot's action.
Chapter 7: αPOMDP: POMDP-based User-Adaptive Decision Making	
S	The state space, all possible combinations state variables, denoting all of the possible states that the system can be in.
s	The current state of the system, an element of S.
s'	The resulting state after an action is taken.
A	The action space; all of the possible actions that the robot can take.
a	An individual action that the robot can take, $a \in \mathbb{N}$.
T	The transition matrix, composed of distributions of the form $P(s'\|s, a)$, describing how each action in a given state results in a new state.
$R(s, a)$	A reward function, attributing a numeric (usually real) reward to an action taken in a given state.
γ	The POMDP discount factor.

Continued on next page

Continued from previous page

Symbol	**Description**
Ω	The observation space, denoting all of the possible observations the robot can make.
O	The observation matrix, composed of distributions of the form $P(o\|s', a)$.
$b(s)$	A probability distribution over the whole state space that encodes the robot's belief as to which is the current state.
$\Pi(b(s))$	A policy, which given the belief over the state space, returns the optimal action to take, maximizing the cumulative reward over a given horizon.
$\Gamma(s, a)$	The hidden transition function of the user, that maps how the user transitions between states when exposed to the robot's actions.
$V(s)$	The value of state s, used in the definition of the state-value rewards.
$I(s', s)$	The impact on the user caused by a transition from state s to s', *i.e.* the difference of $V(s)$ and $V(s')$.
$H = h(T(s, s', a))$	The entropy of the particular transition probability distribution $T(s, s', a)$.
$L = \{s', s, a\}$	A transition sample, essentially a supervised training example for the system, allowing it to learn the impact of its actions.
$s_{a,i}$	The state variables that model agent i.
$s_{u,j}$	The state variables that model user j.
b_f	The fused belief common to all agents.
F_b	A function capable of fusing beliefs.
b_i	The belief of agent i.
T_f	The fused transition matrix common to all agents.
F_t	A function capable of fusing transition matrices.
T_i	The transition matrix of agent i.
$a_{s,i}$	The action to be taken by agent i in state s.
s_i	The current state of agent i.
n	The number of iterations that the system was allowed to run in our experiments.
t_c	The policy calculation period: the policy was recalculated every t_c iterations.
R_c	Cumulative reward obtained by the system. At iteration i, the cumulative reward is the sum of all rewards received up until that iteration.

Continued on next page

Continued from previous page

Symbol	Description
t_3	Number of iterations that the system spent in the three most valuable states.
t	Total execution time of a particular execution run.

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

> "He was determined to discover the underlying logic behind the universe. Which was going to be hard, because there wasn't one."
>
> — Terry Pratchett, Mort

Autonomy has been a central issue in the development of robots since the field's inception. Through the decades, roboticists strove to increase these system's ability to self-govern, aiming to create true agency and achieve the levels of autonomy seen in works of fiction. Early robots were able to autonomously perform elementary movements without supervision, enabling them to perform tasks in industrial environments, such as assembly lines. These robots have been improved over the years, resulting in intelligent production systems which are currently leading the fourth industrial revolution.

More recently, mobile robots have become able to move autonomously in their environment. The development of navigation techniques, such as motion planning and localization, mapping and exploration, allow these systems to autonomously navigate and explore even unknown environments. Step by step, these systems have become more and more autonomous, being nowadays able to navigate even hazardous, harsh environments such as nuclear spill sites and areas affected by natural disasters.

Reasoning is the natural next step; how can a robot decide *which* action to perform, *when* and *how* to perform it? In order to answer these questions, the field of Automated Planning [52] has been applied to problems of robotics, resulting in the development of robots able to autonomously decide the optimal action policy to follow, given their operational constraints, state and goals. Thus, robots became able to evaluate their surroundings, to elaborate plans dealing with uncertain conditions, to gain knowledge and achieve their goals.

Social robots are now looking to share the same environment as human users, carving themselves a place on the human social space. Unlike industrial workers, who operate within strict protocols, easily accommodating the restrictions of industrial robots, domestic users do not follow pre-established routines in their daily life. In fact, domestic users can be very

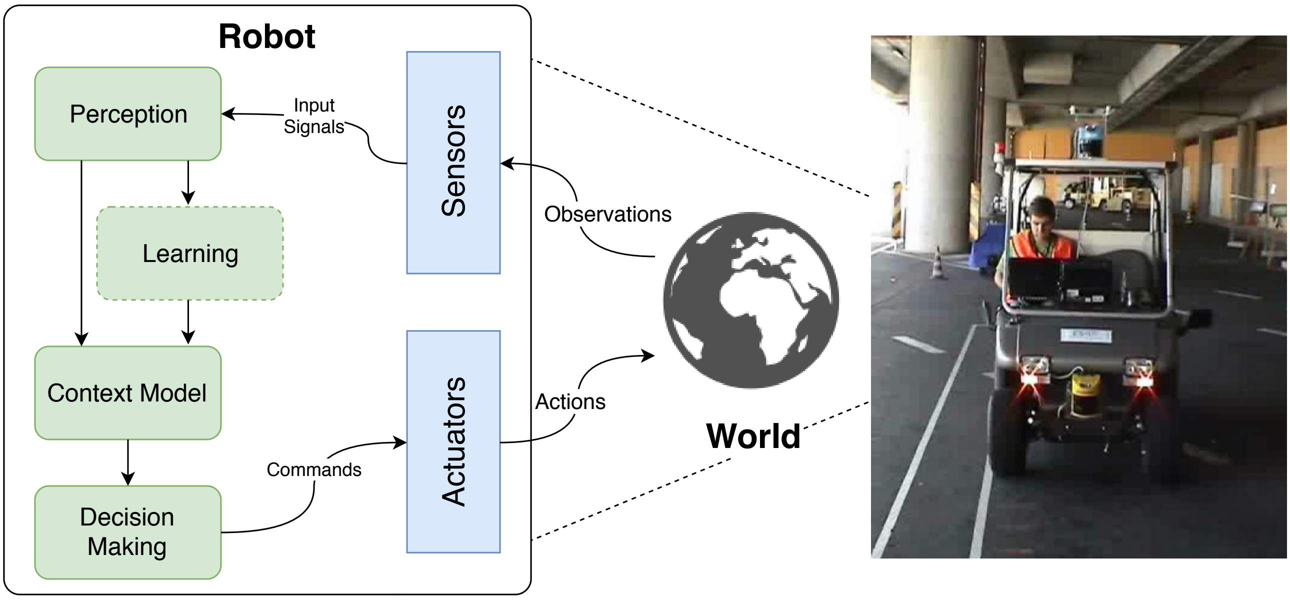


Figure 1.1: An illustration of a context-aware system. The system gathers knowledge on the environment, and uses that knowledge as a modulator for shaping behaviour, while still striving to complete its goal.

unpredictable, constantly inventing new nuanced forms of interaction[1]. Social Robots must, then, gain the **autonomy in interaction** necessary to, in addition to navigating the physical space and dealing with environmental and contextual issues, interact satisfactorily with their users. **Robots of the past have been accommodated by the environment and its users; robots of the future must accommodate the environment and its occupants.**

1.1 Autonomy, Adaptiveness and User-Adaptiveness

Autonomous systems use knowledge to achieve their autonomy. These systems are able to gain information on the world they are acting on, and operate on that knowledge to change their behaviour to accomplish their goal in a manner that is adapted to the environment they are operating in, as illustrated in Fig. 1.1. In this case, the *robot* has *knowledge* about the environment, and uses it to *adapt* its actions to it, achieving autonomy: it does not need to be steered or externally controlled to achieve its goal. Through this process, robots become context-aware and *adaptive* to contextual conditions.

Similarly, "educated" users, such as those in charge of industrial robots, have a very clear and well-bounded understanding of what the robot does, what they can expect from it and what it expects from them. These users know what functions the robot is limited to, and will not try to misuse it by trying to have it perform other functions. Instead, they will operate on the knowledge they have of the system to change their behaviour and correctly work with it. Thus, the *user* has *knowledge* about the robot, which they use to *adapt* their actions to the

[1] In `https://youtu.be/tLt5rBfNucc` we can observe a common kind of unexpected robotic interaction; the Roomba was not designed to carry passengers, and thus continues its operation unperturbed by the user.

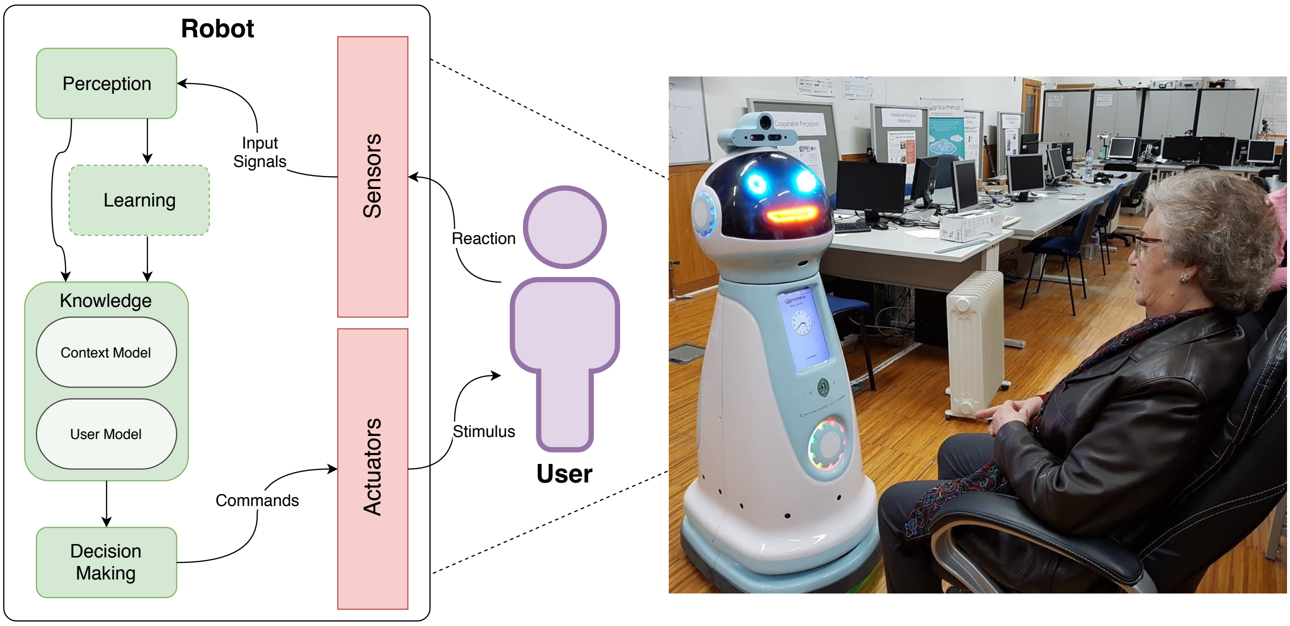


Figure 1.2: An illustration of a user-adaptive system. The system gathers knowledge on the user, and makes use of that knowledge to change its actions according to the user it is interacting with.

robot's characteristics.

Domestic and social environments are populated by non-expert users, who do not possess the knowledge needed to accommodate the limitations of the robotic system. When presented with a system that displays a face, or arms, or any humanoid features, users *expect* that the system be able to display human-like behaviour. If the robot has hands, why should it not be able to hold hands with the user? If it has eyes, how does it not know where the user left their keys, or when their family members last visited? If it has legs, how is it not able to run along with children? There is, thus, a disparity in what the users expect the system to do, and what it does in fact.

Many of these issues, such as physical interaction, object localization or bipedal locomotion, are out of the scope of this work. However, there is a central issue to all of them: in order to hold hands with the user, or to find objects, or to run, the autonomous robot needs to *know* that that is the correct action to take. Furthermore, it needs to know how to apply each action, taking into account the user at hand, and gauge impact of these actions on them. It needs to know the user, its own actions, and how to apply them.

Thus, social robots need to be able to *gather knowledge on their users*, and to *adapt their actions* to their characteristics, limitations, preferences, *etc.* In this sense, social robots can gain **autonomy in interaction**, becoming able to navigate social environments autonomously, much like mobile robots are able to navigate the physical environment; social robots need to become **user-adaptive** in order to gain higher levels of autonomy. In order to become user-adaptive, these systems need to transition from the context-awareness of Fig 1.1, wherein a context model is used to mould the robot's actions, to the human-aware, user-adaptive architecture of Fig. 1.2. In this extended architecture, a user-adaptive robot maintains not only knowledge on relevant context, but also on the user themselves, using this information to shape its decision-making routines, thus producing user-adapted actions.

User-adaptive systems in themselves are not a complete novelty; the issue has been thoroughly studied in the context of Human-Computer Interaction (HCI). In this context, it has long been established that user-adaptive interfaces lead to significant improvements in acceptance when compared to non-adaptive ones [106]. As a consequence, modern HCI systems are already increasingly tailored to the needs and idiosyncrasies of their users, an effect generally achieved by the gathering and usage of data on the user, such as their behaviour patterns or their preferences, which can be gleaned from their usage of social media, their search history, *etc*[2]. Some systems can even make use of the user's psychological characteristics, such as their personality [54][151], to achieve higher levels of adaptation.

1.2 Goals and Main Contributions

Throughout this monograph, we will explore and contribute to the various components of a typical user-adaptive system as presented in the loop of Fig. 1.2. The main goal of this thesis is to contribute each of these sub-problems, enhancing the robot's ability to adapt to the user, an ability we dub *user-adaptiveness*. We aim to contribute to the state of the art by endowing the robot with the ability to *get to know* its user on a deeper level and to *adapt* its actions to them.

We define our main research question as follows:

How can a social robot gain knowledge on and represent its user's characteristics, and use this knowledge to better adapt its actions to them?

We contribute specifically to the four different aspects of user-adaptive systems in Fig. 1.2: user modelling, perception, learning and decision making. We tackle the issue of **user modelling**, aiming to analyse input data and abstract a representation of the user. In this regard we contribute with:

- A novel Bayesian User Model (BUM) which can be used seamlessly by a social robot, a team of social robots, distributed sensor networks or combinations thereof;

We contribute to the issue of **perception**, aiming to provide a connection between the signals observed by the robot and the abstract constructs (such as BUM) used to represent them. On this issue, we perform an exploratory study which demonstrates that:

- It is possible to use a Social Robot to perform psychological screening of elderly users, obtaining answers to questions that correlate with the answers obtained by a professional evaluator, demonstrating it can accurately extract the psychological state of the user.

Then, we contribute to the **learning** block, in which machine learning techniques can be used to adjust a user model to the user based on input data. We show that:

[2]This unbounded data collection has attracted a significant amount of user outrage and controversy, which we consider out of the scope of this work. This problem has been competently discussed, for instance, in this documentary: `https://vimeo.com/nothingtohide`.

- It is possible to employ Bayesian techniques to reduce a dataset into a minimum necessary set, which can be used to learn and classify data with acceptable loss in performance, through the application of a novel surprisal-based technique.

Lastly, we contribute to the **decision making** problem, which aims to take the information gathered and produce a plan for the agent to execute. Namely, we contribute with:

- Preliminary experiments on using Bayesian estimators to adapt a robot's actions to its user;

- A novel POMDP-based technique which, based on a state-based reward and a learning loop, is able to gradually learn the impact of its actions on the user, thereby adapting to them.

At the time of writing, this work has developed into an international cooperation between researchers at the University of Coimbra and the Khalifa University in Abu Dhabi, UAE, resulting in the ongoing development of the βPOMDP approach (Chapter 7).

These contributions resulted in a number of scientific publications, which serve as the basis for this report. The survey of Chapter 2 has been largely adapted from

- Gonçalo S. Martins, Luís Santos, and Jorge Dias. User-Adaptive Interaction in Social Robots: A Survey Focusing on Non-Physical Interaction. *International Journal of Social Robotics*, 2018. ISSN 1875-4805. doi: 10.1007/s12369-018-0485-4. URL `https://rdcu.be/Y9JJ`.

The chapters on user modelling are mainly adapted from

- Gonçalo S. Martins, Luís Santos, and Jorge Dias. BUM: Bayesian User Model for Distributed Social Robots. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*. IEEE, 2017. doi: 10.1109/ROMAN.2017.8172469;

- João Quintas, Gonçalo S Martins, Luis Santos, Paulo Menezes, and Jorge Dias. Toward a Context-Aware Human-Robot Interaction Framework Based on Cognitive Development. *IEEE Transactions On Systems, Man, And Cybernetics*, pages 1–11, 2018. doi: 10.1109/TSMC.2018.2833384 (section on user modelling).

The chapters on user-adaptive decision making are adapted from

- Gonçalo S. Martins, Paulo Ferreira, Luís Santos, and Jorge Dias. A Context-Aware Adaptability Model for Service Robots. In *IJCAI-2016 Workshop on Autonomous Mobile Service Robots*, New York, 2016;

- Gonçalo S. Martins, Hend Al Tair, Luís Santos, and Jorge Dias. αPOMDP: POMDP-based user-adaptive decision-making for social robots. *Pattern Recognition Letters*, 0: 1–10, 2018. ISSN 01678655. doi: 10.1016/j.patrec.2018.03.011. URL `https://www.sciencedirect.com/science/article/pii/S0167865518300825`.

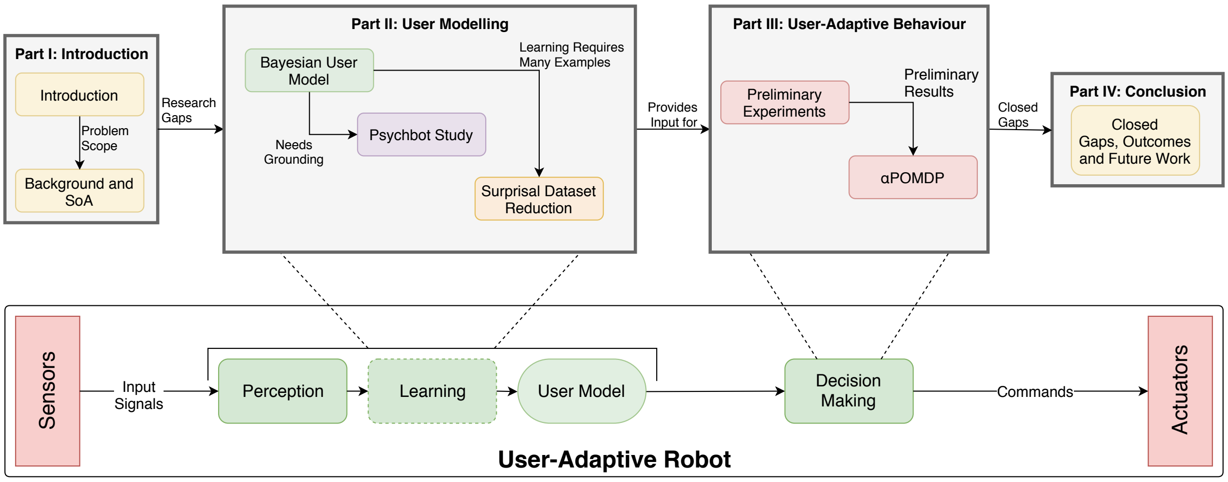


Figure 1.3: An illustration of the structure of the document superimposed on the architecture of a user-adaptive robot (Fig. 1.2). Part I introduces the work, Part II deals with user modelling, Part III deals with user-adaptive decision making, and Part IV presents our conclusions.

All of the adapted passages are the author's original work. The relationship of these publications as outcomes of the work described herein, as well as their relationship to the research gaps uncovered in Chapter 2, are further discussed in Chapter 8.

The contributions of this work extend beyond scientific reporting; every main proof-of-concept is accompanied by an implementation, which is made publicly available within the limits and regulations of the University of Coimbra, and described briefly in the text. These prototypes are published in the hopes that they can serve as a stepping stone for the further testing of the techniques. Their dissemination and reuse both by us and by third parties can boost the technology's readiness level [85] and the impact on society of the work presented in this monograph.

1.3 Document Overview

This document is structured according to the rationale illustrated in Fig. 1.3. Part I presents the necessary motivation (Chapter 1), as well as the necessary theoretical background and state of the art, revealing the research gaps to be explored in our approaches (Chapter 2). These research gaps motivate a stream of novel research, with the general goal of contributing to the state of the art in those specific niches.

In Parts II and III, the research gaps are treated as sub-problems, each solved separately and producing isolated contributions, as described in Section 1.2, presenting our contributions to the fields of User Modelling (Chapters 3 and 4), Dataset Reduction (Chapter 5) and User-Adaptive Decision Making (Chapters 6 and 7), also corresponding to the components of Fig. 1.2.

Regarding user modelling, Chapter 3 presents BUM, a Bayesian User Model able to model a user population from heterogeneous sources of data. Chapter 4 presents a study in which we demonstrate that it is possible to obtain a psychological profile of the user using a social robot and an adaptation of the standard evaluation scales used in Psychology. Seeing as both these techniques support themselves in Machine Learning, potentially requiring large amounts of data, Chapter 5 presents our novel technique for dataset reduction based on surprisal.

Regarding user-adaptive decision making, Chapter 6 presents our preliminary approach, based on Bayesian programming, for adapting a robot's decisions to its user. Chapter 7 presents an improved and extended version of this paradigm, αPOMDP, which extends the POMDP formulation to adapt to the user based on the impact of its actions, as well as its own extension to multi-agent planning, βPOMDP.

Lastly, Part IV analyses the success of the work as a whole, presenting the research gaps which we have tackled, and discussing how our work contributes to them. It also presents an aggregate of the scientific and technological outcomes of the work, relating them to the corresponding research gaps. Chapter 8 concludes the monograph by presenting a discussion on the lines of work that, while mainly inspired by the state of the art, must be left as lines of future work.

Chapter 2

Background and State of the Art

> "Whosoever seeks the truth will not proceed by studying the writings of his predecessors and by simply accepting his own good opinion of them. Rather, the truth-seeker will mistrust his established opinion. He will rely solely on his understanding of the texts by following the criteria of logic rather than the statements of authors who are, after all, human, with the errors and faults which this naturally involves. Whosoever studies works of science must, if he wants to find the truth, transform himself into a critic of everything he reads. He must examine texts and explanations with the greatest precision and question them from all angles and aspects. But he must also observe himself with with a critical eye in this process, so that his judgement is neither too strict nor too lax. If he follows this path, the truths will reveal themselves to him and the possible inadequacies and uncertainties in the works of his predecessors will come to the fore."
>
> — Ibn al-Haitham, circa 1041 AD (quoted in "Muslim Journeys.", Bridging Cultures Bookshelf)

The work developed in this Thesis is set on a number of concepts, theoretical principles, related work and technological tools which must be first explored. In this chapter we cover two main topics: the background that constitutes the basis of both scientific and technological development, as well as the state of the art in user-adaptive HRI, which serves as motivation for the remainder of the work.

We start by defining a number of terms that will be used throughout the work. Then we explore the theoretical principles that served as basis for the developed approaches, such as Bayesian Programming, decision-making and automated planning techniques, and Machine Learning. This is followed by a description of a number of adaptive devices found in everyday life, as well as the social robots used during our experiments. We then survey the state of the art, enumerating a number of research gaps which are used as motivation for the remainder of the work.

To provide a comprehensive overview of the field, spanning from its inception in the twentieth century to the latest developments in the past few years, we have collected a number of recent scientific works, narrowed down from an initial sample of approximately 400 works. To focus the effort of studying and analysing the surveyed works, we employed two key inclusion criteria:

- **Autonomous Systems**: This work analyses systems that adapt to the user in an autonomous manner, as opposed to solutions which are designed with the user's needs in mind [72][100][112][98] or that have to be manually configured [42] in order to act in an adapted manner;

- **Human-Robot Interaction (HRI)**: The focus of this survey are works that deal with social robots (embodied artificial agents) as opposed to other frameworks and pure computational solutions, even if possibly applicable to Robotics.

The works are discussed in a bisected way, first from the scientific perspective, and then from the technological maturity perspective.

The surveyed works are analysed considering the following aspects:

- **Taxonomy Based on the Usage of User Models**: does the work use an explicit user model? What sort of data and representation does it use?

- **Adaptive Parameters and Decision Making**: what parameters of the system are adaptive? How are they adapted to the user?

- **Input-Output (I/O) Interface**: what type of input and output modalities does the system take advantage of?

- **Experimental Maturity**: What is the technique's Technology Readiness Level (TRL) [85]? How is it tested, by whom and where?

These four analysis dimensions allow establishing an overview over the current state of the art and to carve out potential future research lines. Using these dimensions, we enumerate the main trends present in the work, identifying the research gaps that form the foundation for the remainder of the work. We also analyse the technological readiness of the field, based on the experimental procedures of the surveyed works, outlining an additional set of gaps to explore in this work and beyond. These gaps serve as motivation for the remainder of the work presented in this Thesis, and are revisited in Chapter 8, where they are re-connected to all of the work and outcomes achieved.

This chapter is structured as follows:

- Section 2.1 presents the definition of some key terms used throughout the monograph;

- Section 2.2 presents the main scenario used throughout the work, as well as the assumptions derived from it;

- Section 2.3 details the main theoretical background that sustains the remainder of the work;

- Section 2.4 presents a number of everyday applications of user-adaptive systems, as well as the main social robots used throughout the work;
- Section 2.5 presents the taxonomy into which each of the works will be classified, according to their usage of user models;
- Section 2.6 presents a summary of each of the works under analysis, according to their placement in the taxonomy;
- Section 2.7 discusses the surveyed works, exposing a number of research gaps;
- Section 2.8 discusses the experimental maturity of the surveyed works, exposing an additional number of technological gaps;
- Section 2.9 presents a summary of the chapter, including of all the main gaps found throughout the survey.

2.1 Definitions

Robot The meaning of "robot" has been a debated issue, and did not yet attain a unified, consensual definition. For the purposes of this work, we consider that a robot is an embodied machine, commonly able to make its own decisions, to perceive the world around it by processing raw sensor data, and able to actuate in the environment, even if not mechanically. These systems are distinguished from common computers by being embodied, *i.e.* existing in a body which contains the robot's apparent existence, granting it a lifelike self-contained nature. This does not mean that robots cannot be enhanced by external devices and networks; solely that they require a main body to be recognized as robots by the user.

Social Robot A social robot is a robot, as defined above, which aims to introduce itself into the human social space. The main distinction between robots and social robots is their means of communication: whilst a common robot may communicate only via encoded signals, code, or not communicate with humans at all, a social robot communicates primarily with its human users via modalities that can be commonly understood. Furthermore, social robots aims to be a part of the human social space, forming relationships with its user, accompanying them, learning from them; occupying a space in their users' psyche which would otherwise not be filled by a common machine.

User The main application scenario of the techniques presented in this monograph is the usage of a social robot in a domestic environment. Thus, for the purposes of this work, we define a user as any person that is interacting with the a robot, regardless of the duration of that interaction. If applicable, the concept of user can also be expanded to persons that, while not directly interacting with the system, are observed by it and included in its model of the population, as in Chapter 3.

Adaptivity We define adaptivity as a system's ability to perform its function in different scenarios by automatically changing its operational parameters accordingly. These parameters can be any controllable aspects that affect the performance of the system, *e.g.* an air conditioning machine can *adapt* to the outside temperature by changing the velocity of its cooling fan. Similarly, a robot can adapt to an unknown environment by changing its internal parameters (such as the map it uses to navigate) to fit the current environment.

User-Adaptiveness User-adaptiveness is defined as the system's ability to adapt to its user's characteristics. This definition falls in line with previous work [106][97]. User-adaptiveness can be observed in systems that deal with differing scenarios emerging from a switch in user-related conditions, such as the user's identity, preferences, expertise, *etc.*

User Model As seen in [97], a user model is an explicit repository of knowledge on the user, which can be used by an adaptive system to retrieve the information needed for adaptation. User models can be represented in many different ways, ranging from a single attribute representing some relevant characteristic of the user, to probabilistic models that combine the representation of the model with its inference, as in Chapter 3. In our view, user adaptivity does not require a user model, as a system can adapt to its user simply by changing its operational parameters on the fly to suit the user. However, model-based systems explicitly maintain a cache of data on the user that can be, at every step, used to fine-tune the system's mode of operation.

Context In this work, we define *context* as any information, not pertaining to the user, that a system may use to improve its adaptation to the environment. For instance, mobile devices often employ light sensors to adapt the brightness of their displays to the luminosity level they observe. For a survey that deals with the issue of context-aware systems, the authors would like to refer the interested reader to [110].

2.2 Scenario and Assumptions

Throughout this work, we will be operating on the following scenario:

> A social robot, integrated into a network of smart devices which may include other robots, is living with an elderly user at their home. The robot is able to take a number of actions, such as speaking to the user, moving around and bringing them objects. These actions can be combined into complex services, such as accompanying the user to a location (which is composed of several "move around" actions). Whenever interacting with the user, the robot and user take turns performing actions.
>
> The user, being a human being, has a number of characteristics that differentiate them from the remainder of the population. These characteristics can be of many natures. Naturally, they can include information that is simple to obtain from the user, such as their age or gender or be of a more personal nature including, for instance, psychological characteristics.

> The robot is completely autonomous, and should be able to find ways to gain the information that it needs on the user and the environment, to integrate that knowledge through learning mechanisms, and to make decisions based on the accumulated knowledge it has. It is expected that it be able to learn the user's preferences, habits and characteristics and to make decisions that take these aspects into account, thus achieving user-adaptiveness.

From this scenario, we derive the following assumptions:

1. The user and robot interact in a turn-taking manner, in discrete time steps for action and reaction;
2. Unless otherwise noted, the robot is the only influence on the user's state at the time of interaction;
3. The robot is endowed with software components that allow it to execute the actions selected by our techniques, *i.e.* the problem of **low-level control** of robot platforms falls out of the scope of the work;

The approaches presented henceforth are not expected to be limited in application to this scenario in particular; in fact, all of them intend to be easily generalizable to different problems. However, this scenario has guided the development of the approaches, motivated the reasoning behind many of the design decisions, and thus serves as a general guide for the work.

2.3 Theoretical Background

2.3.1 Bayesian Inference and Programming

Throughout this work, all of the techniques introduced employ Bayesian inference [46] in some of its forms. Bayesian inference provides a mathematical framework for reasoning under uncertainty, allowing an autonomous robot to operate in unpredictable and uncertain conditions. In essence, it assumes that conventional models are, somehow, incomplete: it is impossible to represent all possible variables that influence a given phenomenon, giving rise to the existence of *latent variables* which are unaccounted for, thus resulting in uncertainty. This is a result of the fact that models result from an abstraction of the world which, by nature, is incomplete.

In order to minimize this problem, probabilistic reasoning can be used, thus accounting for uncertainty in the models themselves. Bayesian inference is based on Bayes' theorem (or Bayes' rule):

$$\begin{aligned} P(A|B) &= \frac{P(A)P(B|A)}{P(B)} \\ &= \mu P(A)P(B|A) \\ &\propto P(A)P(B|A) \end{aligned} \tag{2.1}$$

which states that the probability of proposition A occurring given that proposition B occurred is proportional to the probability of A occurring, multiplied by the probability of B occurring given

Example Bayesian Program

program
- *description*
  - *specification*
    - ***Variables:***
      $X_1, X_2, ...$
    - ***Decomposition:***
      $P(X_1 \wedge X_2 \wedge ...) = ...$
    - ***Formulation:***
      Specification of the parametric forms of each distribution.
  - ***Identification:***
    Method for the estimation of all parameters and/or distributions.
- ***Questions:***
  $P(searched|known)$, where *searched* are the variables we aim to gain knowledge on, and *known* are the known variables

Figure 2.1: An example of a Bayesian program.

that A occurred. This is often simplified to the middle form of Eq. 2.1, where a normalization term μ is employed in place of $P(B)$, ensuring that the resulting distribution is still valid.

In Bayesian terms, $P(A)$ is known as the *prior distribution*, and represents *a priori* knowledge on the probability of A occurring by itself. $P(B|A)$ is known as the *likelihood distribution*, and represents the probability of B occurring given that A occurred. Given that A is normally the *searched* variable, *i.e.* the proposition whose veracity we are looking to ascertain, this distribution can be thought of as the probability of a given dataset fitting the hypothesis A. Lastly, $P(A|B)$ is known as the *posterior distribution*, the result of the inference, indicating the probability of A being true given the occurrence of B.

Several mechanisms exist for the formulation of Bayesian models, such as Kalman Filters, Markov Decision Processes or Bayesian Networks [46]. In order to achieve a general representation of these models, and a notation and formalism that could generally encompass and extend them all, Bayesian programming was developed [16].

The Bayesian Program, as illustrated in Fig. 2.1, is split into two main parts: a *description* and a *question*. The description of the program details how a joint distribution is obtained, while the question presents the distribution that we are looking to obtain, in terms of variables that are *searched* for and *known*. The description itself is split into two parts: a *specification* and an *identification*. The specification details which variables are involved in the problem, and which distributions establish their relationships. The identification specifies how the free parameters of these distributions are obtained.

Lastly, the specification itself is split into three parts: *variables*, which enumerates and defines all of the variables involved in the problem; *decomposition*, which details the independence relationships that are assumed in the problem, and how those result in partial distributions; and *parametric forms*, which defines the distribution types for each of the elementary distributions obtained during decomposition.

The Bayesian programming paradigm allows, thus, for a formal definition of probabilistic problems, including all of the elements involved. Its expressive power allows it to represent a very wide array of different problems and algorithms. However, it does not, by default, provide any insight into the tractability of a given problem: whereas other Bayesian solutions, such as POMDPs, have well-known tractability and feasibility limits, Bayesian programs do not enjoy

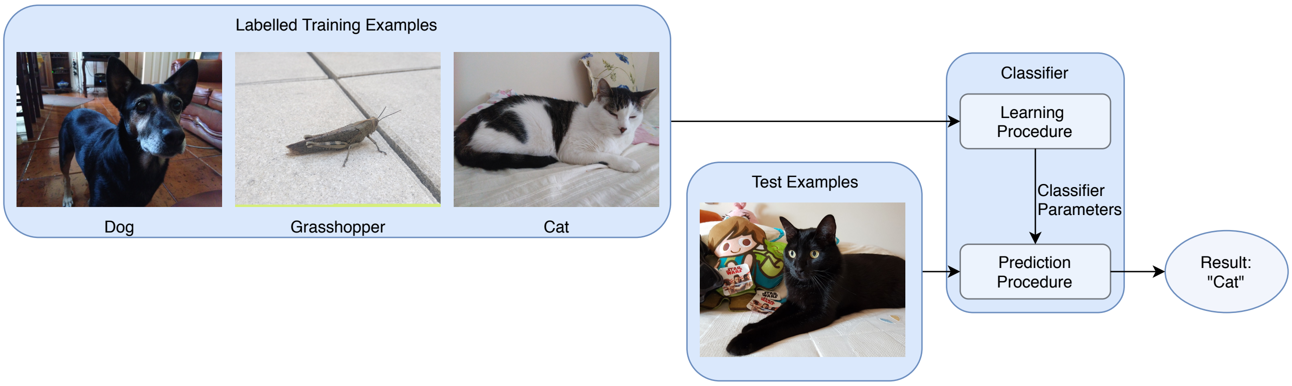


Figure 2.2: An illustration of the functionality achieved by a classifier employing supervised learning [17]. The classifier is fed examples of the various classes, which then enables it to attribute classes to new, unlabelled examples.

such limitations, and can thus easily be made infeasible.

2.3.2 Machine Learning

Machine learning techniques [17] aim to enable an autonomous machine to refine its operation based on data, improving its performance on a given task without being specifically programmed to do so. Specifically, the act of learning consists of the incorporation of information into a model, which is then used to influence the way the task is performed[1]. This paradigm differs from the classical approach, wherein a technique is algorithmically developed for a certain task, and iteratively tested and fine-tuned to improve its performance in the task. In turn, learning techniques are able to automatically adjust themselves to the data made available for learning, requiring little to no manual tuning.

Machine learning techniques operate on *training data*. They are given a set of n examples, x_i with $i \in \{1, 2, ..., n\}$, and must incorporate these into their own knowledge to improve their performance on task T according to a metric M. For testing purposes, it is common to split the dataset into *training* and *testing* sets, wherein the technique is allowed to train on the training set and is then evaluated on the testing set. Cross validation is achieved by rotating the examples in each of the training and testing sets, employing different subsets of the dataset as training and testing examples, thus avoiding any biases introduced by training the technique when applied to a particular training set.

Machine learning techniques enjoy wide application in varied fields such as automated planning [53], computer vision [113] or personality computing [152], and is widely used by technological companies such as Google or Facebook to deliver and improve their services.

Supervised and Semi-Supervised Learning Supervised learning techniques operate by incorporating labelled examples into their model. These techniques receive labelled data, *i.e.* for each example x_i, the technique is also given a label y_i which encodes the "correct" answer

[1]The resemblance of this definition to that of adaptiveness and user adaptiveness is no coincidence; by incorporating data, the learning mechanism is adapting to it.

for example x_i. The supervised learning paradigm can be conceptualized as the system having access to a teacher or oracle, who is able to provide the correct answers to input combinations.

In the example of Fig. 2.2, we can observe the training and testing procedure of a classifier, a technique which takes its input variables and projects them in a discrete space. In this case, the technique is fed labelled images of several animals, and uses this information to fit its model. Then, once given a test image, the technique is able to produce a prediction of what the image may contain.

Semi-supervised techniques operate similarly to their supervised counterparts, but are only given part of the y labels, leaving a set of unlabelled examples. The goal of semi-supervised techniques is thus to surpass the performance obtained by supervised techniques, which discard the unlabelled data, and of unsupervised techniques, which disregard the labels, by combining both datasets.

Examples of supervised learning algorithms include Support Vector Machines, Naïve Bayes or Artificial Neural Networks [17].

Reinforcement Learning Reinforcement learning techniques operate on a dynamic environment, receiving feedback for each action taken. These techniques improve their performance by iteratively performing their function and receiving negative or positive *reinforcement* from an external source, which indicates whether the action was successful or not.

Reinforcement learning techniques are fed a set of examples, x_i, as in the supervised case, which are not labelled. Instead, for each prediction or action, the technique receives a *reward*, conceptually similar to the rewards employed in MDP and POMDP techniques. The technique then optimizes its actions with the goal of maximizing the reward received, fitting itself to the reward function employed.

Unsupervised Learning Unsupervised learning techniques are not fed any labelled examples at all, nor are they fed any feedback on their actions or predictions. Instead, unsupervised techniques are expected to autonomously detect patterns in the data, and to incorporate them in their models.

A common application of unsupervised learning techniques is on the problem of clustering, wherein the technique is provided a number of examples x_i and tries to split them into a number of groups c_j, attributing to each example a corresponding group. This effort results in a grouping of examples, which may reveal important relationships among them. For instance, in [111] and [149], clustering techniques are used to group users of web applications with one another, which allows for the transference of preference profiles to different users. Recommender systems [75] also make wide use of clustering techniques, using them to predict the preferences of unknown users by matching them to previous groups.

Another common application area is Anomaly Detection [29], wherein the goal of the technique is to find anomalous examples in the data, without being fed any reference. Anomaly detection allows for the discovery of examples that fall out of the pattern followed by the remainder of the dataset, making them useful in applications such as fraud detection.

2.3.3 Decision-Making Processes and (Partially-Observable) Markov Decision Processes

The aim of decision making (or automated planning) algorithms is to select an agent's actions such that it achieves a given goal, ideally in a manner that optimizes some performance metric. Generally, the agent receives a set of percepts, such as processed sensory input, and outputs commands to an underlying actuation mechanism [53].

Decision-making techniques are domain-independent, in the sense that, given the correct assumptions, they can be applied to artificial agents in different domains with small changes. The key to the successful application of these techniques to different domains lies in the *variable grounding problem*, *i.e.* in the relationship that is established between the theoretical problem that the technique is tackling and the real problem at hand. In other words, it is necessary to bind each of the variables used in the modelling of the system to an observable, physical quantity with a real-world meaning, thus establishing the technique's relationship with the concrete problem it is solving.

Variable grounding can be achieved by languages such as Planning Domain Description Language (PDDL) [52], which bridge the real-world problem and the abstract problem that a decision-making technique solves. Languages such as PDDL aim to model the application domain of a decision-making technique, thus allowing for the grounding of abstract decision making into realistic domains.

Alternatively, the problem at hand can be grounded manually, specifying the real-world problem in a way that can be directly tackled by a decision-making technique. We have opted for this approach, since our simplification of the user model allowed for such a direct grounding, as will be seen in Part III.

Markov Decision Processes Markov Decision Processes (MDPs) model fully-observable, sequential stochastic decision processes. Within this framework, at each iteration an agent selects an action to perform based on its policy, which maps its current state to the action it should take. Depending on the current state and action taken, a reward is attributed to the agent, which is used as a basis for optimizing the agent's policy. The overall objective of the agent is to maximize the cumulative reward over a given temporal horizon. An MDP is defined as a 4-tuple $< S, A, T, R >$, where S is a finite set of states, A is a finite set of actions, $T(S', S, A) = P(S'|S, A)$ is the probability that a certain action a leads from state s to state s' and $R(S, A)$ is the reward obtained from taking action a in state s.

Partially-Observable MDPs POMDPs ([136]) are an automated planning technique which differs from its counterparts by taking a probabilistic approach as to the state that the agent is in, *i.e.* it does not assume that the agent knows its current state, expressing its knowledge as probability distributions which are refined as the agent gains information. To compensate for this, the system assumes that the world is static, *i.e.* the world is only assumed to change as a result of the agent's actions. It is also assumed that taking an action is an atomic procedure, which falls in line with the scenario of Section 2.2.

Partially Observable Markov Decision Processes (POMDPs) are an extension to MDPs in which the current state s is unknown. They are represented as a 6-tuple $< S, A, T, R, \gamma, \Omega, O >$ ([139, 63]) where S, A , T and R are defined as previously, Ω is a finite set of observations, $O(s', a, o) = P(o|s', a)$ is the observation function, and γ is the discount factor ($\gamma \in [0, 1]$) per time step.

Not being able to observe its state, the agent maintains a belief $b(s)$, defining the probability of being in state s. b is updated after taking action a and receiving observation o:

$$b^{a,o}(s') = \frac{O(s', a, o) \sum_{s \in s} T(s, a, s') b(s)}{P(o|a, b)}, \tag{2.2}$$

Previous approaches have attempted to encourage agents to improve their knowledge of the current state via a belief-based reward function ([11]), reducing the uncertainty in the agent's belief through the use of the entropy [134] of the belief state distribution b as a measure of uncertainty.

Normally, a POMDP rewarding scheme consists of defining a function $r(s, a) : \mathbb{R}^n \to \mathbb{R}$ that maps each combination of state and action with a scalar reward. This rewarding scheme allows for the modelling of single-objective systems operating under uncertainty. Multi-objective reward functions $R(s, a) : \mathbb{R}^n \to \mathbb{R}^n$ ([138]) can also be used, encoding each objective as a discrete function, which can be optimized separately or through a scalarization scheme:

$$r(s, a) = \sum_k w_k R(s, a)_k \tag{2.3}$$

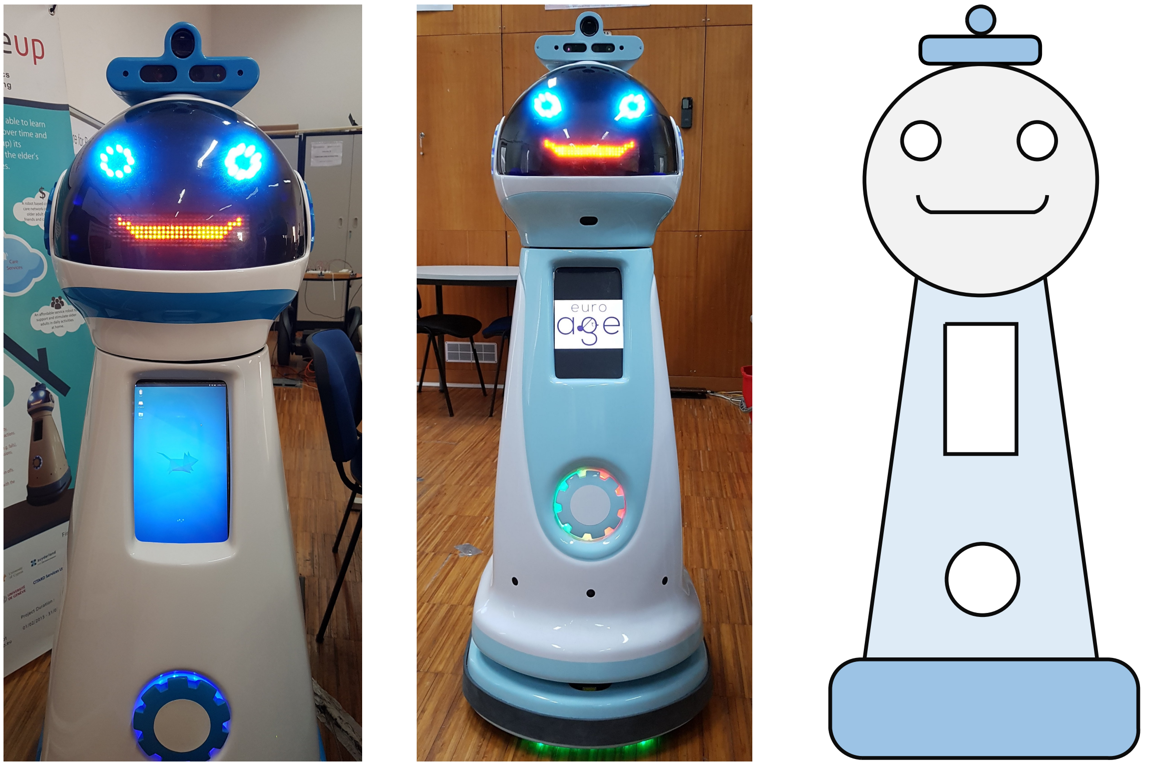
resulting in a single reward function that is the weighted sum of the multiple rewards, according to the w_k weights. Multi-objective rewards can also be solved by prioritizing reward functions ([160]) using a set of reward preferences.

Solving an MDP or a POMDP consists of finding a policy which provides the optimal action to perform in each state, *i.e.* a function $\Pi(b(s)) : b(s) \to a$ that, given a current state (in the MDP case) or a belief function (in the POMDP case), returns the optimal action for the system to take. Several mechanisms exist for solving POMDPs, such as Dynamic Programming [28], value iteration ([128, 27]), policy iteration ([139]), accelerated value iteration ([158]), structured representation ([19]), and approximation ([164]).

Several software packages have been developed that implement POMDP solvers, such as QMDP[2] and SARSOP[3], which are employed in this work. QMDP [26] is an approach to find Q functions for POMDPs by making use of Q values of the underlying MDP $Q(s, a) = R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V(s')$, and linearising across $Q-$values to obtain the value at a belief: $V(b) = max_{a \in A} \sum_{s \in S} b(s) Q(s, a)$ (). By choosing the action that has the largest Q value for a given belief state, an agent can behave optimally.

Successive Approximations of the Reachable Space under Optimal Policies (SARSOP) [76] is a point-based algorithm which samples a set of points from the belief space. The sampled points form a tree ($\mathcal{T}_\mathcal{R}$) where the root is the initial belief b_0 which leads to actions and each action leads to observations. Each node in $\mathcal{T}_\mathcal{R}$ represents a sampled point b. If all possible sequences of actions and observations are applied than the set of nodes in $\mathcal{T}_\mathcal{R}$ will be exactly the reachable space $\mathcal{R}$. SARSOP avoids this by focusing on finding the approximate cover, maintaining both a lower bound and upper bound on the optimal value function $V*$. It then gradually reduces the gap between the upper and the lower bounds on the value function at b_0, until a stopping condition is reached.

[2]`https://github.com/JuliaPOMDP/QMDP.jl`
[3]`https://github.com/JuliaPOMDP/SARSOP.jl`



(a) The first version of the GrowMu robot.

(b) The second version of the GrowMu robot.

(c) An illustration used throughout the work.

Figure 2.3: An illustration of the social robots used throughout this work.

2.4 Technological Background

2.4.1 Hardware

The techniques developed in this Thesis require testing with human users for validation, data gathering, *etc.* To this end, a social robot must be used as a tool that embodies the behaviour encoded in each technique.

Several kinds of social robots exist on the market nowadays. The Paro robot [108], a seal-like therapeutic robot aiming at helping depressed patients, has been widely used in research [153]. It communicates solely via its movements and chirping sounds, reacting to the user's touch and voice. The Jibo [119] robot is a personal assistant robot, aiming at aiding users in their day-to-day activities, such as ordering food or taking pictures. It is not mobile, but does employ moving joints for expressiveness. Information on Jibo's inner workings is scarce, but it is able to learn the user's preferences regarding the its actions, as well as their habits. Similarly, Buddy [62] intends to be a personal assistant for the home. Unlike Jibo, it is mobile and able to roam around the house. In terms of user-adaptiveness, they both seem able to get to know their users in much the same way, learning their schedules, names and habits, and we postulate that it operates on a static user profile that it builds in the first or first few interactions with the user.

Pepper [4] is a humanoid domestic robot, which aims at interacting with its users emotionally. It is able to recognize the user's emotional state from their voice and facial expression, and adapts to the state the user is in. In terms of user-adaptivity, Pepper uses immediate information to reactively adapt its actions to the user's state and, to the best of our knowledge, this is the only user-adaptive ability of this robot. Naturally, the robot is also *adaptable*, which means it can be configured by the user to act as they want, namely via the installation of *apps*. This represents a relatively limited form of user-adaptiveness, when compared to some of the systems analysed before. However, this simplicity allows the system to be robust and, as such, more suitable to the target environment and users.

To assist in testing and gathering data for the novel techniques in this work, two versions of the GrowMu social robot were employed, illustrated in Fig. 2.3. The GrowMu robotic platform that be used was designed during the social robot[4] [123] [6] project, and was then adopted for use in the context of the GrowMeUp project [86]. The platform was designed with the intent of striking a balance between three factors: aesthetics, functionality and cost.

The robot's aesthetics aim to present it as a socially-recognizable presence, but still a peaceful, non-imposing one. As such, it features a relatively short stature (125cm), as well as an armless humanoid shape, in an attempt to convey its ability to interact naturally.

Functionally, the robot is split into two main parts, the top and bottom sections. The bottom section houses the drive system, power supply, processing unit and navigational sensors, including a laser range finder and sonar sensors. The top section contains the user-interface components, such as a 10-inch touch screen, LED-based facial expressions, RGBD and RGB cameras. Aside from the powered wheels, the robot contains no mechanical actuators, instead focusing on verbal interaction. The robot's systems are mostly comprised of consumer-grade components, thus achieving a cost-effective design.

2.4.2 Software

In order to interface with this hardware, the GrowMeUp [86] system was used, a collection of ROS packages for interfacing with the hardware and achieving basic functionality, such as navigation, speech synthesis and recognition, and facial expression through the LED panels. This system was then extended using three main technological tools: ProBT, Scikit-Learn and JuliaPOMDP.

ROS *ROS* stands for Robot Operating System. ROS is a set of libraries, packages and guidelines that allow for the streamlined development of software targeting robots. Its standards allow for the decoupling of packages to the point where software can be developed regardless of the robotic hardware it will run on, thus allowing for great flexibility and code reusability.

Executable code is contained in *nodes*, processes that run on a computer and which are linked against (or import) the ROS libraries. ROS nodes have access to the ROS API, allowing them to benefit from the ROS framework but are normal processes in every other respect. The execution of nodes can be configured using *parameters*, which are stored globally and can be accessed by any node.

[4]More information on the SocialRobot project can be found at `http://ap.isr.uc.pt/?w=project_information&ID=55`.

ROS nodes communicate with each other using *messages*, which contain *fields*, much like the `structs` as found in many programming languages. Message delivery is *asynchronous* and happens through callback functions that execute in parallel with the main loop. Messages can carry various types of information, including sensor readings, percepts such as maps or information on the status of the user of a social robot.

Messages travel in *topics*. All of the nodes that subscribe to a certain topic receive the messages that are published in that topic and, conversely, any node can publish on any topic. Topics can only support one message type, and all nodes must agree on the message type that flows in each particular topic.

In this work, the ROS library is accessed mainly through its Python[5] interface, `rospy`[6]. This allowed for a shorter prototyping times, higher flexibility in integration with other technologies, and access to the very large Python package repository.

ProBT, Scikit-Learn and JuliaPOMDP The ProBT package, developed by ProbaYes[7], was used for modelling and computing the Bayesian inference techniques used in our models and systems. ProBT allows for the definition and solving of Bayesian Programs, and was used throughout the works of Chapters 3, 5 and 6.

Scikit-Learn[8] is an open-source Machine Learning library which enjoys widespread use, development and maturity. It can be used for common Machine Learning-related tasks, including supervised and unsupervised learning, regression, classification and dimensionality reduction. Its library contains implementations of many state-of-the-art algorithms which can be freely reused. These were used throughout the work for benchmarking, namely in Chapter 5.

JuliaPOMDP[9] is a collection of packages that allow for the specification, solving and testing of MDP and POMDP-based techniques. It relies on the Julia[10] programming language, and contains implementations of common algorithms for MDP and POMDP solving, such as QMDP and SARSOP, which are used extensively in Chapter 7.

2.5 Taxonomy of User-Adaptive Systems

As seen in [106], in order to be a user-adaptive system, having *information about the user* is key. This information is kept in the form of a *user model*, describing the aspects pertaining to the user that are of importance towards the operation and adaptation of the system. This has led to the birth of the field of *user modelling*, of which we can find an early survey in [97], a field concerned with the organization, representation and acquisition of information on a system's user.

In this section we present a taxonomy, inspired by early surveys [106][97][34], which is used to categorize the works under review. These categories will be used throughout the remainder of the text for describing, analysing and discussing the systems under survey as groups. The

[5] More information on Python can be found at `python.org`.

[6] Rospy is described at `http://wiki.ros.org/rospy`.

[7] ProbaYes can be contacted through `http://www.probayes.com/en/`.

[8] More information on Scikit-Learn can be found at `http://scikit-learn.org/`.

[9] More information on JuliaPOMDP can be found at `https://github.com/JuliaPOMDP`.

[10] More information on the Julia programming language can be found at `https://julialang.org/`.

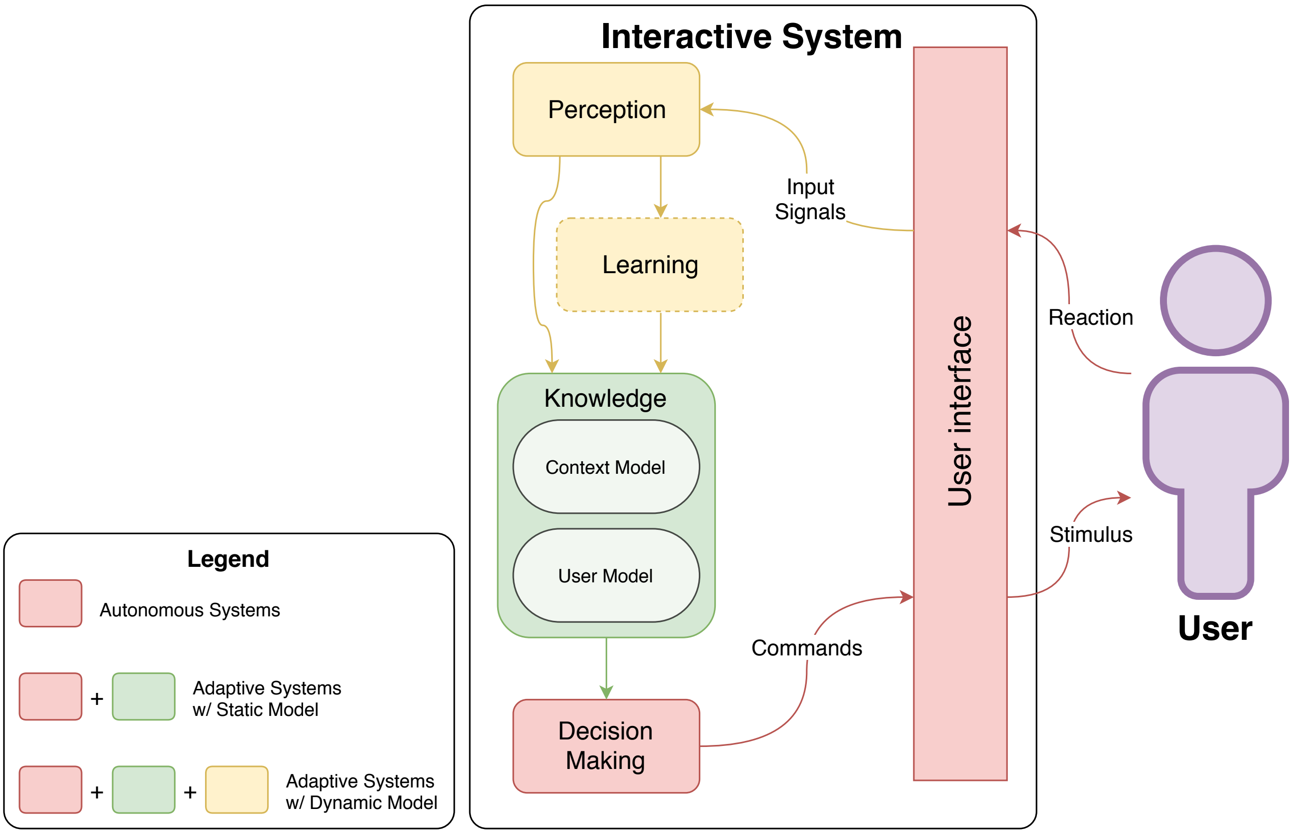


Figure 2.4: The general structure of an interactive system with adaptive abilities, a generalization of Fig. 1.2. Systems that contain only the red modules operate with no user model. The systems that make use of the red and green modules operate with static user models. Systems that combine all coloured models operate on dynamic user models.

categories are presented considering two key factors: the existence of an explicit user model, and its persistence in time. They are enumerated as follows:

1. **Adaptive systems with no user model**: these systems do not keep an explicit cache of information on the user. They are characterized by a reactive behaviour with respect to the user's immediate feedback;

2. **Systems based on static user models**: these systems use predefined information about the user's relevant attributes, or information acquired in a preliminary interaction, making use of this explicit knowledge for adaptation;

3. **Systems based on dynamic user models**: like the previous, these systems maintain an explicit model of the user, which is updated as they operate with basis on information gathered during operation.

We aim to advance beyond the methodology of recent surveys, such as [94][121], by providing a more in-depth analysis of the works under survey, as well as an additional discussion on technological maturity.

Fig. 2.4 illustrates the generic framework for a user-adaptive system. The framework is generically composed of two main components.

- **User Interface**: the layer where the information exchange between the system and the user occurs. It is constituted by sensors and actuators that deliver and perceive stimuli to and from the user.

- **Decision Making Module**: the layer where decision algorithms [53] take as input the perceived information, and generate a future response action that will be synthesized by the interface.

The user's model can be implicit in the design of the adaptive system itself [97]. This results in a system that is able to achieve adaptive behaviour without persistent information on its user, as illustrated in red in Fig. 2.4. These techniques perform *reactive adaptation* by adapting their behaviour to immediate information on their users, such as their intention or walking speed. Changes in the user's behaviour that the system is monitoring trigger immediate changes in the system's operational mode, and are not stored or updated in a model of the user.

A system can be user-adaptive using a static, immutable view of the user during the whole interaction period, as illustrated in red and green in Fig. 2.4. This static user model can be obtained by the system using two different mechanisms: 1) the model is created by the system itself during a preliminary of the interaction or 2) it can be provided *a priori* by an external agent, for instance, using a questionnaire or a form. These category of systems is unable to dynamically learn the user's characteristics.

Systems endowed with Dynamic User-Models have the ability to change their view of the user through a feedback mechanism, adapting to new circumstances, as illustrated in red, green and yellow in Fig. 2.4. In this case, the user's feedback is used to trigger updates to the existing user model, thus allowing the system to evolve with the user. Systems with these characteristics have been suggested as the best (yet most complex) solutions for user adaptiveness [97][106], since they are able to iteratively improve their knowledge on the user, and to evolve as they do.

2.6 User-Adaptive Robots

2.6.1 Adaptive Systems with no User Model

The authors of [141] present a system for aiding the user in their mobility. The system, consisting of an intelligent wheelchair or walker, is able to determine the user's intended goal on a map, and their satisfaction with the current path that the system is taking, encoding these variables as hidden states in a POMDP. None of this information is kept in a user model *per se*, but is instead used to adapt the system's actions according to the user, thus achieving user-adaptiveness. The users do not evaluate the system with respect to its adaptiveness, instead demonstrating only that the system does indeed work.

In [92], the authors present a system based on a robotic wheelchair, able to carry its user to their intended destination. The system employs Bayesian techniques to estimate the user's intended goal, which it then uses to guide its navigational efforts. The user's intention is not kept in a user model, and is instead represented solely by its belief. The system was not tested in realistic conditions, but the authors show that it is indeed able to infer a user's goal from the user's input.

The authors of [31] present a system based on an intelligent walker, able to adapt to a user's walking speed. The system constantly monitors the user's walking speed, through odometry force sensors located in the system's handlebars, which it uses to modulate its own speed, thus adapting to the user. The system was tested in demonstrative trials, and shown to be able to accomplish its goal.

Similarly, the authors of [101] and [107] present a system, integrated in the MOBOT project [43] and making use of its robotic walker platform, aiming a aiding the user in walking. The system infers the user's intention using their inputs and their movements, as measured by an LRF. The system is able not only to infer the user's intended goal when going through crossroads and intersections, but is also able to be teleoperated hands-free with the user walking behind the device. User tests were carried out with 35 participants, demonstrating the system's operation.

In [48], the authors present a system, integrated in the SPENCER project [144], intended to adaptively guide a user to a location. The system monitors the user's movements, and adjusts to their walking speed and engagement level, proactively engaging the user if needed. The system makes use of a hierarchy of Mixed Observability Markov Decision Process (MOMDP) that subdivide the decision-making process into smaller sub-processes, thus making it computationally tractable. In a demonstrative trial, the authors show that the system was able to improve the user's engagement and reduce both the mean and variance of the distance to the user, indicating successful adaptation.

The authors of [71] present a robotic vacuum cleaner that is able to adapt to its user's preferences. The robot does not keep an explicit model of the user, but identifies the user's commands and any obstacles it finds on its map, and determines the times of the day where these areas are best accessible, thus adapting to its users' occupancy of the environment. The authors do not present any experiments with users, but validate their mathematical solution in demonstrative trials.

In [60], the authors present a system that aims at helping a user gather the ingredients for a recipe. The user selects items by pointing at a board with drawings of the items, and the robot adapts to the user by estimating their intention, from their gaze and speech, speeding up the delivery of the item. The authors employ Bayesian techniques, and show that this proactiveness on the part of the robot significantly speeds up the process, in a set of tests involving 26 participants.

The work presented in [82] exploits the *entrainment* effect, wherein two or more people have a tendency to adjust their prosodic characteristics are they become closer. The system aims at teaching basic mathematics to users, and adjusts its pitch as the interactions with the user take place, progressively matching that of the user. The system was evaluated with 48 participants, who indicated that they experienced a much higher social presence when interacting with the adaptive system.

The authors of [22] present a study on the impact of the inclusion of user intention and explicit time dependency as hidden variables in a POMDP framework. The authors argue, without explicitly discussing user-adaptiveness, that the inclusion of the user's intention can improve the quality of the interaction. Indeed, a study with 35 participants, where participants were interacting with a simulated robot in a driving experience, shows that the users do indeed prefer the adaptive system, and it is able to achieve significantly higher rewards over time.

Similarly, the authors of [78] present an assistive driving system which is able to determine when the user is distracted and to compensate by taking control of the vehicle for a short amount of time. The system learns models of non-distracted drivers in an off-line step, which it then applies to each user to determine their state at each moment. The system uses Bayesian techniques to maintain a set of beliefs over the state of the driver, which it uses to estimate when the driver needs help. The system was tested with an undisclosed number of participants, and the authors show that the system was able to prevent a number of accidents in a driving simulator.

The authors of [137] present a study on the transition of the CADENCE turn-taking system to a user-adaptive version. The system monitors the interaction with the user's status and adapts to their cadence of active/inactive status. A study with 15 participants shows that the system was able to elicit the same social response as the non-adaptive version could, with the crucial difference that the adaptive version was able to automatically achieve results within a single interaction, whereas the original system have to be manually tuned between interactions.

In [135], a study on the impact of user-adaptiveness on users, namely on the impression of rapport, is presented. The authors implemented a humanoid robot that mimics the user's gestures while speaking, via an estimation of synchronism between the user and robot. A study with 23 participants shows that most users preferred interacting with the adaptive version of the system.

2.6.2 Systems Based on Static User Models

In [41], a system that makes use of Personas for adaptation is presented. Personas, also used in [84] in the context of HCI, consist of a set of manually-built user profiles that, combined, aim to represent a large portion of the potential user base. Each persona represents a number of users, and new users can be quickly matched to a known persona, with adaptation taking place in accordance to the matched persona, a mechanism akin to that used commonly in recommender systems. In this work, the authors have defined the Personas with basis on tests with 28 users, through the administration of questionnaires. This technique eschews the usage of large learning dataset, and favours the usage of experts in building the Personas. The authors show that the system is able to adapt to new users in demonstrative tests.

The authors of [8] explore the adaptation of a robot's synthesized personality to the personality of its user. The user model is constituted by the user's personality, which is estimated at the beginning of the interaction and remains static throughout the interaction. The robot communicates through gestures and speech, and the authors have determined that users interacting with the adaptive robot have found it to be more expressive.

The work presented in [73] aims at assisting a user in dressing themselves. The system maintains a list of poses that the user cannot reach, thus adapting to their limitations. This model does not evolve during normal execution, but is learned by the robot in a specific interaction, during which the user is asked to position themselves in a variety of poses. The system adapts by compensating for the positions the user cannot reach, and the authors have found that the system, in its adaptive operation, is faster at accomplishing its goal.

Similarly, [51] presents a system for dressing assistance. The system maintains a model of the user's mobility, namely the positions achievable by their joints, which is learned in dedicated tests. The system adapts to the user's limitations by compensating for the lack of mobility of the user. The authors do not provide a comparison with a non-adaptive system, but have demonstrated that the system is able to achieve its function.

The authors of [120] present a study on the usage of an adaptive robot for teaching dance lessons for children. The system interacts with children one-on-one, and maintains a static model of their personal information, as well as of the history of interactions they have shared before, which it uses to adapt its speech and gestures to the child it is interacting with. The authors perform a thorough study of the effects of this system on the children, and note that the system was able to teach the lessons, and be perceived by the children as a peer or a sibling, instead of a tutor or teacher.

The HOBBIT system, presented in [49], is able to provide several services to elderly users. Its adaptivity relies on an initialization phase, during which the user provides the robot with their preferences, such as voice type and volume, to which the robot then adheres in future interactions. The authors do not compare their system with a non-adaptive version, but demonstrate its functionality in a number of trials involving elderly users.

The work presented in [2] exploits crowd-sourced information to determine its user model. The system focuses on organizing shelves according to user preferences, and these preferences are learned, via collaborative filtering, from data gathered from a number of participants. The robot is then able to organize the shelves by representing the user's preferences as constraints, and using an optimization process to violate as little constraints as possible when placing objects on containers. The system is not compared with a non-adaptive version, but is able to organize the shelves successfully.

2.6.3 Systems Based on Dynamic User Models

A *proactive* system is presented in [57], integrated in the ACCOMPANY project [9]. The system maintains a state of the user, and a set of rules that cause that state to evolve. The goal of the robot is to keep the user in a "good" state. By observing the environment and the user's choices, the robot identifies opportunities for action that can divert the user from reaching an undesirable state. For instance, in the example presented in the work, the system detects that the user has not taken their medication, despite the robot's warning and, knowing that this can lead to an undesirable state, the robot takes action and fetches the user's medication, thus compensating for their attitude. In one demonstrative trial, the authors show that the system can indeed identify opportunities and act proactively.

The authors of [55] present a robotic Intelligent Tutoring System, first presented in [156], that aims at aiding a child in learning how to read. The system maintains knowledge on the user's reading level, which it periodically evaluates and updates using an Active Learning technique. This information is then used to adapt the serious game that the child and system are playing, with the goal of enhancing their learning performance. The authors show that the system is able to interact with children of varying ages, and that children interacting with the adaptive system were able to learn more effectively than with regular systems.

Joint tasks, tasks performed cooperatively between user and robot, are explored in [105]. The system, applied to the problem of moving a table out of the room, monitors the user's level of adaptability to the robot's optimal plan, and adjust its actions accordingly. As the user complies, or not, with the robot's suggested change of plans, the robot adjusts its model of the user and, thus, its actions. The system employs a Multi-Agent Markov Decision Process for decision making, and the authors have found that user preferred interacting with the adaptive version of the system, and found it more trustworthy.

Similarly, the authors of [39] present a system that aims at performing a joint task with the user. The system maintains a Theory of Mind representation of the user, namely of their current task and beliefs. This representation is updated as the interaction takes place, with the robot aiming to minimize explicit instructions between it and the user. The authors evaluate the system in a table cleaning scenario, in tests with an undisclosed number of users, and conclude that the adaptive system can perform the task significantly faster than the non-adaptive alternative.

The authors of [132] present a system that aims at alleviating the workload of an operator behind a Wizard of Oz. The system maintains a model of the Wizard's action policy, *i.e.* when and how the user acts, which is updated as the user uses the system. Gradually, the system refines its representation of the user, to the point where it is able to replace them. The authors have used a two-robot setup, simulating an assisted learning scenario, to validating their technique, and have shown that the system does indeed alleviate the workload on the user while maintaining the same results in terms of child-robot performance.

In [13] and [14], the authors present a system that aims at adapting the coloured lights in a robot to the tastes of the user. The system relies on three basic preference profiles, which are adapted to each user via a technique akin to Reinforcement Learning. The authors did not test their approach with users, but have demonstrated its functionality in simulated scenarios.

User-adaptivity is explored, as a primary task, in the work presented in [70], which is the culmination of a sequence of works [66, 67, 68, 69]. The system learns the user's preferences, which are updated using interaction traces obtained as the robot repeatedly interacts with the user. The system makes use of an MDP formulation to recalculate its policy according to the user's preferences, with the authors noting that other decision-making approaches could be used. Tests with 17 participants have shown that the users believe that the system can indeed adapt to its user, and that it progressively adapts to their needs.

The authors of [127] present a system that aims a cooperatively performing music with the user. The system makes use of Context-Free Stochastic Grammars, and is taught a baseline user profile in a dedicated interaction. During interaction with the user, they can inform the system that they dislike the robot's musical decisions, which triggers a change in their preferences profile, and thus in the system's actions. The authors performed tests with users, who reported that they found decreasing difficulty in producing music with the system, indicating the successful adaptation of the system to the users.

The Dialogue Manager of the SERROGA system is presented in [102][103]. This manager implements turn-based dialogue, which is made adaptive by the incorporation of the user's feedback on the Bayesian-like Dynamic Factor Graph of the system, adapting it to the user's preferences. The system was tested with real users in a 10-day test, and the users noted that the system was indeed able to change in accordance with their preferences.

A robotic recommender system [75] is presented in [80], aiming at aiding users in learning English. The system operates on the principles of classic recommender systems: it builds and maintains a preferences profile of the user, which is maintained in an ontology, regarding the serious games used for learning. This profile is updated during interaction via n-gram analysis of the events. The system relates this data with both data from the same and from other users to provide better suggestions to the user. A study with 12 participants has shown that the usage of this system improves the users' performance when learning.

An evaluation of various interactions is presented in [131], relying on a system that aims at learning from the user, eventually being able to carry out commands without explicit orders. The system maintains a model of the user's preferences, which evolves at every interaction. The system was tested with 25 non-expert users, and the authors conclude that the users prefer the adaptive system over the non-adaptive one.

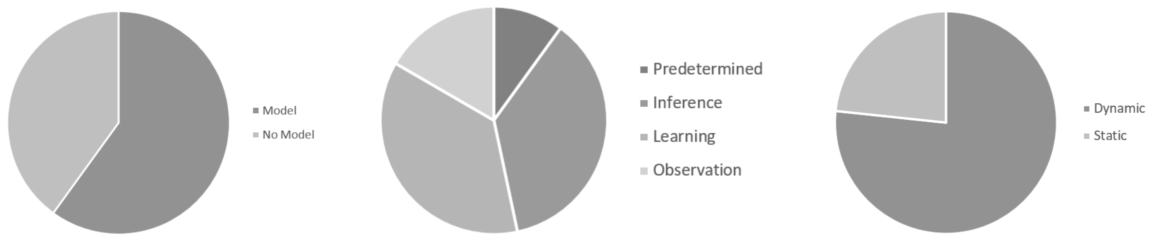
2.7 Analysis and Research Gaps

In this section we discuss the works presented in Section 2.6, uncovering research gaps to support the remainder of the work. We perform our analysis according to the key characteristics identified in the beginning of the chapter, and present research gaps in each of the relevant dimensions.

2.7.1 Taxonomy Trends in User-Adaptive Systems

Fig. 2.5 and Table 2.1 present an overview of the taxonomy trends in User-Adaptive HRI systems, complemented by an overview of the usage of psychological information on the user. It is visible that most works apply user models for adapting to the user. Furthermore, systems tend to gain their own information on the user, with the portion of works that are given the user model beforehand being relatively small. As seen in Fig. 2.5c, most of the systems update their user model during execution. Thus, in general terms, all of the works surveyed fit into one of the architectures of Fig. 2.4.

We have observed, in Section 2.6.1, that techniques that do not require a user model to operate tend to focus on a single attribute for adaptation. Furthermore, these techniques tend to adapt to the user in a reactive manner, continuously measuring the characteristic of relevance



(a) Usage of user models. (b) Source of user data. (c) User Data Persistence.

Figure 2.5: Pictorial illustration of the usage of user models in the works under survey.

Table 2.1: User model types, source, persistence and usage of emotions and personality in user-adaptive HRI. Observed models are those that are measured directly, with little perception taking place; inferred models employ perception to the raw signals; learned models are obtained via learning techniques; and predetermined models are pre-built and given to the system.

Reference	Data Source	Persistence	Emotions	Personality
(a) Techniques with no user model				
[48]	Inference process	Dynamic	No	No
[22]	Inference process	Dynamic	No	No
[31]	Direct observations	Dynamic	No	No
[60]	Inference process	Dynamic	No	No
[71]	Inference process	Dynamic	No	No
[78]	Inference process	Dynamic	No	No
[82]	Direct observations	Dynamic	No	No
[92]	Inference process	Dynamic	No	No
[101]	Inference process	Dynamic	No	No
[135]	Direct observations	Dynamic	No	No
[137]	Direct observations	Dynamic	No	No
[141]	Inference process	Dynamic	No	No
[129]	Direct observations	Dynamic	No	No
(b) Techniques with static user models				
[8]	Learning	Static	No	Yes
[73]	Learning	Static	No	No
[41]	Predetermined	Static	No	Yes
[49]	Predetermined	Static	No	No
[51]	Learning	Static	No	No
[120]	Predetermined	Static	No	No
[2]	Learning	Static	No	No
(c) Techniques with dynamic user models				
[132]	Learning	Dynamic	No	No
[13]	Learning	Dynamic	No	No
[39]	Inference process	Dynamic	No	No
[56]	Learning	Dynamic	No	No
[57]	Inference process	Dynamic	No	No
[70]	Learning	Dynamic	No	No
[80]	Learning	Dynamic	Yes	No
[105]	Inference process	Dynamic	No	No
[127]	Direct observations	Dynamic	No	No
[102]	Learning	Dynamic	No	No
[131]	Learning	Dynamic	No	No
[12]	Inference process	Dynamic	Yes	No

and updating their decision-making routines to compensate. Thus, these techniques tend to make use of decision-making techniques into which the user's attributes can be seamlessly integrated, examples of which include the POMDP, where the status of the user can be encoded as a hidden variable of the model or used as input to the model's state (as in Chapter 7).

These techniques lack the ability to distinguish users from one another, operating with complete ignorance as to the identity of the user. This results in, effectively, a generalization of user characteristics, according to the reactions they display to the stimuli generated by the system. In other words, if different users display the same reactions to the stimuli offered by the system, the system will operate in the same manner, regardless of user identity.

Systems based on static user models, seen in Section 2.6.2, overcome this shortcoming by employing a model of the user which is given a priori or inferred during a preliminary interaction. This model can be built in a personalized manner for each individual user, endowing these systems with the ability to adapt to different users and, if necessary, to adapt to different users according to their identity.

We can observe, in Fig. 2.5 and Table 2.1, that many of the systems surveyed are given part or the whole of the information used for adaptivity, and that this information remains static throughout execution. These techniques based on static models are unable to adapt to the changes of their users, or to evolve as they do. Information on the user is gained only once during the experiment, and cannot be changed to accommodate for the changes that the user may undertake. Therefore, these techniques appear to be better capable of dealing with larger numbers of differentiated users than non-model techniques but are, at the same time, unable to deal with long-term interactions with single users.

This fact does not necessarily represent a drawback: some information on users changes very slowly or unnoticeably throughout the duration of the interaction, even if it lasts for long periods of time. For instance, systems based on personality, such as [143], can argue that this particular aspect of the user is subject to little change with age [148]. However, systems based on more transient aspects of the user, such as their preferences and habits, need to be able to readjust these characteristics to ensure long-term viability.

Lastly, systems based on dynamic user models, presented in Section 2.6.3, gradually learn and adjust the relevant characteristics of their users. These techniques represent the combination of the best qualities of both the other categories of systems: they are both able to adapt to several users, and to keep adapting as their users change. These systems have the additional advantage of not requiring a setup phase for profile determination or definition, nor manual gathering and introduction of user information.

Furthermore, systems based on dynamic user models are able to continuously improve their perspective of the user as time changes. This represents increased autonomy and robustness for the system: even if the user does not change, the system's initial perception of them may be partially incorrect, and these systems have the possibility of improving upon those errors. Thus, these systems are, in our view, the most appropriate for long-term interaction: as users grow and age, these systems have the potential to grow and age with them.

Table 2.2: Adaptive parameters, I/O modalities, Decision Making and Evaluation of the social robots under survey.

Ref.	Adaptive Parameters	Input Modality	Output Modality	Decision Making	Evaluation Metrics	Evaluation Process
(a) Social robots with no user model						
[48]	Robot Speed	User's pose and speed	Motor control	MOMDP	Distance to user and speed difference	Measurements
[22]	Decisions (take left turn)	Physical controls	Image, Sound	POMDP	POMDP Rewards, driving performance, perceived control, naturalness, similarity to real world, social appropriateness	Measurements, Questionnaires
[31]	Robot Speed	Odometry, Physical controls	Motor controls	Fuzzy control	n/a	n/a
[60]	Decisions (object to move)	Speech, Gaze	Robot arm movement	Rule-based	Projection accuracy, prediction accuracy, response time, perceived awareness and intentionality	Measurements, Questionnaires
[71]	Decisions (room to clean)	User locations, task success	Motor control	Rule-based	n/a	n/a
[78]	Decisions (Warn driver or intervene)	Physical controls	Image, Sound	Hidden Mode Stochastic Hybrid System	Time in safe and unsafe conditions	Measurements
[82]	Voice pitch	User speech	Robot speech	Rule-based	Perceived social presence, rapport, persistence and learning gain	Questionnaires
[92]	Robot's Navigation Goal	Physical Controls	Robot Commands	Rule-based	Recognition accuracy	Measurements
[101]	Robot speed and path	Physical Controls	Robot Commands	Rule-based	n/a	n/a
[135]	Robot's gestures	Vision, Speech	Robot Commands	Rule-based	Information distance, perceived gesture recognition, perceived behaviour performance, perceived social interaction, enjoyment	Measurements, Questionnaires
[137]	Decisions (when to speak, what objects to manipulate)	Speech, Vision, Depth	Speech, Robot Commands	Rule-based	User's speech time	Measurements
[141]	Decisions (navigation goal)	Physical Controls	Robot Commands	POMDP	Robot path, state variables, destination probabilities	Measurements
[129]	Decisions (difficulty of items to present)	Speech	Speech	GOAL	Intrinsic motivation	Questionnaires

Continued on next page

Table 2.2 – *Continued from previous page*

Ref.	Adaptive Parameters	Input Modality	Output Modality	Decision Making	Evaluation Metrics	Evaluation Process
(b) Social robots with static user models						
[2]	Decisions (placement of objects)	Crowd-sourced data	Robot controls	Rule-based	F-scores	Measurements
[8]	Robot's speech and gestures	Speech	Speech, Gestures	Rule-based	Preference towards a type of adaptation	Questionnaires
[73]	Decisions (how to dress the user)	User's pose, speech	Robot Commands	Rule-based	Task completion speed	Measurements
[41]	Font size, interface complexity, warning levels, robot location	n/a	n/a	Rule-based	n/a	n/a
[49]	Sound volume, robot speed, speech output gender, robot's name	Speech, touch	Robot commands, speech	Rule-based	Perceived usability, acceptance	Questionnaires
[51]	Decisions (how to dress the user)	User's pose, speech	Robot Commands	Rule-based	Classification accuracy	Measurements
[120]	Sequence of dance movements	User's pose	Robot commands	Rule-based	Gaze position, facial emotion, body language, perceived bond, satisfaction, amusement, anxiety, enjoyment, observed leadership and expectancy	Manual classification, Questionnaires
(c) Social robots with dynamic user models						
[132]	Decisions (what interactions to perform with the user)	Physical Controls	Robot Commands	Rule-based	Human intervention ratio, child learning rate	Measurements, Questionnaires
[13]	LED Colours	Physical Controls	LED Colours	Rule-based	Cumulative reward from users, estimation error	Measurements
[39]	Decisions (Adaptation to user's choice of subtask)	Vision, Speech	Robot Commands, Speech	Rule-based	Number of communications needed to instruct the user	Measurements
[56]	Reading difficulty level	Speech, Touch	Speech, Images	Active Learning	Number of words learned	Measurements, Questionnaires
[57]	Decisions (when to render services)	Speech, Touch	Speech, Images, Robot commands	Equilibrium Maintenance	Relevant opportunities for services found	Measurements

Continued on next page

Table 2.2 – *Continued from previous page*

Ref.	Adaptive Parameters	Input Modality	Output Modality	Decision Making	Evaluation Metrics	Evaluation Process
[70]	Decisions (when to take actions, including services and parameter adjustment)	Gestures	Sound, Projected images	MDP	User satisfaction, perceived coherence, ease of use, originality, perceived helpfulness, perceived adaptivity	Questionnaires
[80]	Decisions (what content type to learn)	Speech, Physical Controls	LEDs, Robot commands	Rule-based	n/a	n/a
[102]	Decisions (dialogues to execute)	Tactile sensors, sound, touch	Image, speech, robot commands	Dynamic Factor Graph	User opinion	Questionnaires
[105]	Decisions (where to move a shared object)	Vision, Physical controls	Robot commands	MAMDP	Ration of participants who changes strategies, perceived trustworthiness	Measurements, Questionnaires
[127]	Decisions (sounds to make)	Physical Controls	Sound (music)	Context-Free Stochastic Grammars	Number of user interventions, perceived difficulty, engagement, conformity, progression, speed	Measurements, Questionnaires
[131]	Decisions (where to guide the user)	Vision, odometry, robot position, user Attention, speech	Robot Commands, navigation	Rule-based	User opinion (score)	Questionnaires
[12]	Decisions (positive, neutral or negative output)	Facial expressions, electro-dermal data, RGBD, touch screen	Images, speech, gestures	Rule-based	Perceived enjoyment, understanding, trust	Questionnaires

Research Gap: Learning New Users As the user-adaptive system interacts with multiple users, it continuously models their characteristics. This process can potentially take large amounts of time and data. If the robot interacts with an unknown user, it will have no information on this user, and will have to adjust their model from the beginning. A solution for this problem is the matching of a new user to an existing model that exhibits the same initial characteristics as the new user. In practice, the system would use an existing model as a starting point, enabling the system to quickly adapt to an approximate view of the new user.

An important aspect of user-adaptive systems in HCI, such as recommender systems, is the manner in which these systems make use of inter-user information. In recommender systems, users are, for instance, clustered in representative groups[77] which can then be used for extrapolating the characteristics of users on which there is relatively little information. Other systems, such as [165], even explore the latent social connections between users to increase the level of adaptivity of the system. At first, this will result in a user model that suffers from an approximation error. However, as the system interacts with the user, it should be able to continuously adapt and, thus, correct the initial error.

Furthermore, optimized learning strategies can be used for efficiently learning new users. Techniques such as Active Learning [23] and techniques based on Information Theory could be useful to optimize the information gathering procedure.

Research Gap: Psychological Trait Modelling As seen in Table 2.1, there is very little attention dedicated to adapting systems to a user beyond the general usage of personal and behavioural data. Characterizing users on a deeper, psychological level, can yield unprecedented satisfaction and acceptance levels[7]. Psychological measures on the user can include, for instance, their personality [152] or their emotional state [116].

The usage of Personality in Affective Computing, and thus HCI, is becoming a popular trend [151], but its presence in user-adaptive social robots seems to be relatively restricted. Very few of the works surveyed take these aspects into account, but those that do achieve positive results. This exposes research gap in the refinement of personality and emotional information to achieve higher levels of fidelity and adaptation.

No technique that we have found combines the knowledge of the user's personality with the knowledge of other aspects, such as routines and preferences. Thus, psychological and behavioural analysis of the user seem disjointed in the literature, constituting another research gap. The combination of behavioural and psychological information can result in a *holistic* profile of the user, which could be the basis for unprecedented adaptivity levels.

Research Gap: Big Data Crowd-sourcing approaches are used by some techniques to improve their user-adaptive abilities. These works can be seen as a step forward, with respect to systems that learn solely from a reduced number of users, as they are able to leverage larger amounts of data for adapting to the user. A possible research line would be to explore the influence of Big Data and data mining techniques in the improvement of the adaptive abilities of embodied systems. Shedding the purpose-built model paradigm, a more generalized and extensible user model [74] could benefit from big data and data mining techniques by incorporating larger and more varied pieces of information on the user, potentially adapting to increasingly finer points of their characteristics. The Internet of Things may also be an instrumental addition to this paradigm, by providing continuous streams of additional multimodal data into the systems, which can then analyse it and extract patterns that inform the system on the user.

This can lead to the problem of *over-modelling* the user, in which too much data on the user is kept and never used in any of the system's functions. A possibly interesting line of future work is the study of this trade-off: the determination of *how much* data on different aspects on the user is relevant for adaptation, and if a saturation effect is achieved after a certain number of aspects or volume of data is incorporated into the model.

2.7.2 Adaptive Parameters and Decision Making

As seen in Table 2.2, the majority of the works under review uses as single adaptive parameter the decisions made by the system. Generally, these works adapt to the user in *what* they do. This allows systems to achieve a manner of *functional adaptation*, wherein their choice of actions is influenced by their information on the user. This leads to systems that are able to, for instance, navigate autonomously to where they believe they can best interact with the user. In the case of these systems, the adaptive process is intertwined with the system's own function, and its goal can only be achieved through adaptation.

On the other hand, a number of works adapts in *how* they interact with the user. This manner of *non-functional adaptation* allows systems to adapt the parameters of their actions, which translates into changes in their movement speed or prosody, for example. These systems can achieve higher levels of adaptation without affecting their main function, decoupling the adaptive process from their main goal, which can be achieved with or without adaptation.

A smaller number of works adapts in both of these perspectives, changing both *what* actions they take and *how* they are taken. In these cases, the robot is, for instance, able to adapt both the way it conveys information, and what information better suits the user at that particular time. These systems take the adaptive process to a higher level, truly adapting both to the problem and user at hand.

Research Gap: Continuous Adaptation An important requirement for a companion social robot is long-term viability, which requires that a long-term companion be able to live and cooperate with its users for extended periods of time with no intervention from technical personnel. As such, in addition to being able to function normally, it must be able to learn from its users and continuously adapt to the changes it observes on its users.

An important number of surveyed works do indeed continuously adapt to their users, iteratively re-evaluating their users' characteristics. However, these systems are the least developed, often relying on single measurements of their users, or not storing this information in a re-usable user model. This line of research is, thus, ripe with opportunity for future work, and may be the key for enabling social robots to live with their users in the long term.

Additionally, long-term viability enables the system to build true relationships with its user, as seen, for instance, in the Paro [153] and in other experiments [64][65]. The further exploration of these long-term adaptive interactions, with more complex and complete adaptive mechanisms, could also constitute an interesting line of research.

Research Gap: User-Adaptive Robotic Perception As noted for instance in [106], user-adaptive behaviour can stretch beyond the synthesis of behaviour proper. Knowing the user more deeply can enable a system to better understand their actions and states, resulting in the application of user-adaptiveness not only to behaviour synthesis but also to *perception*.

Artificial perception is already a very active field of research [46], and it requires intricate systems to achieve interesting results. The addition of a user-adaptive layer to perceptive systems would likely increase their complexity, but would also likely significantly improve their performance, as seen in [163].

Some works on user-adaptive perception exist, such as [126] for colour description systems, but this line of research is, to the best of our knowledge, unexplored in interactive social robotic systems. Thus, endowing an adaptive robot with the ability to adapt its perceptive abilities as well constitutes a research gap.

Research Gap: Interaction with Multiple Users Interactions with groups of users is becoming an important trend in research. This type of interaction is an important factor in the integration of social robots as members of society, since group interactions among humans are very frequent. However, none of the surveyed works are able to interact with multiple users, which constitutes an important research gap.

Research Gap: User Adaptivity as a Layer The techniques surveyed in this work cover a wide number of application areas, showing the usefulness of user-adaptiveness in many applications. In these applications, being user-adaptive, as mentioned before, is seldom the main task of the system. In our view, user-adaptiveness can thus be seen as transversal to all areas in which HRI is involved.

As a future line of research, it could be interesting to explore the modelling of user-adaptive characteristics as a layer that can be applied to any HRI or HCI systems. This layer would be responsible for modulating the signals received from the user, and the actions decided by the decision-making modules, in a manner that was best adapted to the user. This comes as a natural consequence of the architecture presented in Section 2.5: user-adaptiveness can be seen as a layer between the user interface and decision-making blocks. This paradigm would allow a larger number of systems and applications to benefit from the advantages of user-adaptiveness, and can be seen as a generalization of user-adaptivity in both perception and action.

2.7.3 I/O Interface

Regarding the user interface used by these systems, Table 2.2 presents the interaction modes used by the surveyed works. We can observe that, unlike systems dedicated to HCI, which focus on traditional interaction modalities such as keyboards and mice, social robots employ a myriad of different interaction modalities. Social robots can make use of natural communication channels to express themselves, resulting in the increased expressiveness noted, for instance, in [8].

We can observe that speech is, by far, the most popular choice of interaction modality. This trend is to be expected, since speech is one of the most natural communication channels for humans, and speech analysis and synthesis have been the focus of significant attention in recent years. Furthermore, we observe that a significant number of HRI systems make use of more than one interaction modality, thus employing multi-modality of speech and other channels.

Humans also communicate through two additional channels, aside from verbal: nonverbal, usually consisting of gestures, and paraverbal, namely via prosodic changes. The paraverbal channel, as far as we know, has been seldom explored in user-adaptive robots, with only a few works making use of it. However, the nonverbal channel, namely gestures and physical interactions such as moving robots and objects on a table, seems to be gaining popularity among researchers, with a number of techniques adopting these modalities and exploring their possibilities.

Research Gap: Physical and Haptic Systems Physical interaction is an important aspect of human relationships, and is a hallmark of intimate interpersonal relationships. However, with the exception of the robotic walker systems of Section 2.6.1, we could not find any systems able to interact physically with the user *in an adaptive manner*, *i.e.* in which the physical interaction itself was adapted to the user's behaviour or characteristics, despite there being an important body of physical HRI work, such as [79]. While some systems do employ physical controls, as seen in Table 2.2, none of them employ touch as an adaptive parameter or adaptive output modality.

With the advent of haptic systems, it could be interesting to explore the impact of user-adaptive, touch-based physical interaction on the objective and subjective measurements employed in the evaluation of these systems.

2.8 Maturity of User-Adaptive Systems

We now aim at determining the overall readiness of user-adaptive technologies, and in identifying the main obstacles impeding further progress. One of the main indicators of technological maturity is the demonstration of its functionality in the operational environment – the user's home, in the case of domestic systems. This poses a number of scientific and technological problems, namely the study of the long-term impact of social robots in human environments and the development of solutions able to operate for extended periods of time. Some of the systems under review already make use of relevant environments, such as purpose-built rooms and so-called living labs. However, in order for technological progress to be achieved, these solutions need to operate autonomously with no supervision at the users' homes which, to the best of our knowledge, none of the surveyed systems has.

Similarly, it is important to transition from non-expert users to end-users, *i.e.* from analogous user samples to those who the technology is actually being developed for. Instead of employing non-expert users and corridor sampling techniques in their development, user-adaptive systems need to be tested with end-users. These tests allow for the gathering of crucial feedback that can be used to improve the technology towards the end-user, not necessarily towards scientific developments.

User Skill Level One important aspect of mature systems is the ability to deal with their end-users, as opposed to educated users or technical personnel, as noted in Chapter 1. We

Table 2.3: Readiness metrics of the surveyed user-adaptive HRI techniques.

Reference	Environment	Participants	Participant Expertise
(a) Social robots with no user model			
[48]	Laboratory	1	Undisclosed
[22]	Laboratory	35	Non-expert
[31]	Laboratory	Undisclosed	n/a
[60]	Laboratory	26	Non-expert
[71]	n/a	n/a	n/a
[78]	Simulation	Undisclosed	Undisclosed
[82]	Laboratory	48	Non-expert
[92]	Simulation	n/a	n/a
[101]	Laboratory	35	End-users
[135]	Laboratory	23	Non-expert
[137]	Laboratory	15	Non-expert
[141]	Laboratory	1	Undisclosed
[12]	Relevant Environment	51	End-users
[129]	Relevant Environment	22	End-users
(b) Social robots based on static user models			
[8]	Laboratory	21	Non-expert
[73]	Laboratory	2	Undisclosed
[41]	n/a	n/a	n/a
[49]	Relevant Environment	49	End-users
[51]	Laboratory	3	Undisclosed
[120]	Relevant Environment	12	End-users
[2]	Laboratory	15	Non-expert
(c) Social robots based on dynamic user models			
[132]	Laboratory	10	End-users
[13]	Simulation	n/a	n/a
[39]	Laboratory	Undisclosed	Undisclosed
[56]	Laboratory	49	End-users
[57]	Laboratory	1	Undisclosed
[70]	Laboratory	17	Undisclosed
[80]	Laboratory	12	End-users
[105]	Simulation	69	Non-expert
[127]	Laboratory	8	End-users
[102]	Laboratory	16	Expert Users
[131]	Laboratory	25	Non-experts

can observe that solutions with no explicit model of the user tend to be tested with more end-users. This indicates a higher maturity of no-model systems when compared to the remaining classes of systems, which can be attributed to the fact that these tackle simpler problems, and constitute simpler solutions.

As seen in Table 2.3, the most popular test subject of these works is the non-expert user (*e.g.* students), *i.e.* users that, while not proper end-users, are also not part of the system's development. This points at a lack of readiness in the field: the prevalent use of students as test subjects indicates that the systems are not mature enough to be presented to the end-users. This constitutes a technological challenge.

This is largely not a problem of research *per se*, as the scientific principles in question can still be demonstrated on non-end-users. However, in order to progress technologically, it is important that the end-users be involved in the final stages of development, thus providing important insight into whether the systems under development *actually* fit their needs.

Relevant Environments Another important aspect of a technology's overall maturity is its ability to be tested outside of the highly-controlled environment of a laboratory. We can observe that the vast majority of works has not yet left the laboratory. This fact reveals another technological challenge: seeing as the impact of user-adaptiveness has been demonstrated, these systems could benefit from technological transference into mature, commercial systems.

Long-term Scenarios Long-term test scenarios are an unavoidable obstacle in the development of these systems. Passing a long-term test indicates maturity in the system, and is necessary, in our view, to classify a system as over TRL4. However, the very definition of "long-term" is of an ambiguous nature. For our purposes, we define "long-term" as a trial that takes place for over 5 or more consecutive days. Only one of the adaptive systems under review [102] has successfully performed long-term tests, albeit of only 10 days. In fact, the trend points to very short test sessions with the users, of only a few minutes, which last only long enough to provide insight into the principles at work, which points to a low technological maturity. These short sessions tend to be sufficient to demonstrate the intended research, and are thus the most popular method.

However, in technological terms, long-term interaction is key for the maturity of HRI systems, namely domestic social robots. Robots should be able to interact continuously or intermittently with their users for months or years of use, as is the case with current consumer electronics. This indicates another technological challenge in the field: long-term tests are demanding, from a technological standpoint, leading to a tendency to produce proof-of-concept systems with little impact on society.

Metrics and Standardization Adaptivity is seldom the main task of the described systems. Indeed, this is to be expected: user-adaptiveness in and of itself offers little utility to the user. However, this results in a well-observable disparity in the measurements used for evaluating the performance of the adaptive effort. The performance measurements used by the surveyed works (Table 2.2) can be split into three basic types:

- **Introspective measurements**, such as POMDP rewards or classification accuracy;

- **Interaction measurements**, such as speech time, automatic measurements that relate to the user's experience with the system;
- **Subjective measurements**, such as ease of use, assessing the user's experience with the system through questionnaires.

This results in a lack of standardization, and thus maturity, in the field.

Introspective measurements provide little to no information on the user's experience with the system. Mostly, these measurements show that the system was able to achieve some self-motivated goal, such as achieving a high POMDP reward, or a high confidence as to the user's characteristics. They typically demonstrate that the system's mathematical intricacies work as designed, and approximate reality as closely as the authors intended. However, these cannot be trivially related to the user's experience with the system, providing little insight as to the actual impact of the adaptive process on the user.

On the other hand, subjective measurements, such as user acceptance [37][131] and user satisfaction [87], are able to provide deep insight into the user's experience in the system, and provide objective and empirical information on the impact of the adaptive system. Many of the works under survey employ questionnaires in some way or another, demonstrating that their adaptive processes do indeed produce the intended impact on the users, be it ease of use, perceived bond, among others. However, if a system's goal is to be as autonomous as possible while having measurements as to its own performance on user adaptiveness, these measurements suffer from a major flaw: they are not automatic, and require extensive human intervention, not only in their administration, but also in their interpretation.

On the middle ground, interaction measurements, such as user intervention time, provide limited insight into the system's impact on the user and allow some level of further personalization based on those metrics. These measurements are able to close the interaction loop, providing the system with on-line information on how its action are influencing the user. Automatic performance metrics are a desirable trait of an autonomous system, since they enable the robot to evaluate its own performance, and apply, for instance, techniques for self-rewarding [133] and self-motivated reinforcement learning. However, they tend to be extremely domain-specific, and not generalizable to other types of interaction.

There is a lack of a unified metric, or standard set of metrics, that can objectively measure the performance of user-adaptive systems[11]. Automatic interaction measurements can possibly constitute a viable first step towards a solution to this problem. Autonomy in this measurement is, thus, an indispensable requirement for the robot's overall autonomy. An interesting avenue of future work would be to automatize subjective metrics or, from the other perspective, devise interaction measurements that can be empirically validated. This gap is already being partially tackled by initiatives such as RobotCup@Home, which stimulate development by promoting a competitive, standardized environment.

Open Databases An important characteristic of mature research areas is the ability for different works to compare their results against one another. The standardization of metrics,

[11]For these reasons, we have opted for not including the results obtained by each individual technique in Table 2.3, as their comparison would be meaningless.

discussed above, is an important factor in this comparison. Common datasets are also an important factor, as they provide the common basis upon which each system will work.

The surveyed works seem to each operate on their own data, which is not made widely available. This constitutes a problem when it comes to comparing different techniques, as it removes the important common ground for comparison. Thus, it may become important in the future to build a database of domestic usage of robots, such as those found commonly for Computer Vision [50] and Action Recognition [125].

Interface Reduction *vs* User-Adaptiveness The technologically-mature systems we can observe on the market today tend to exhibit the following characteristics:

- Ease of use;
- Low versatility: focus on a single function;
- Robustness and fault tolerance.

These characteristics stem from a simple design pattern that can be found across these systems: their interface is maximally simplified, allowing a large majority of users to use them successfully. By simplifying the interface design to a point where any person, regardless of expertise, is immediately able to understand how they can reap the benefits of the use of the robot. For instance, the user interface of the Roomba robot is reduced to a large button in the centre of the device labelled "CLEAN". This enables any user, from any demographic, to make use of the robot: they simply press the largest button on the device, and it works.

The Paro robot, one of the most successful among our examples, also features a reduced interface. It communicates only via the non-verbal channel and emulates, as closely as possible, the behaviour of an immobile pet. Since users are accustomed to interacting with animals, interaction with Paro becomes natural, despite the simplistic adaptive facilities of Paro. However, this simplicity comes at a price: Paro is not a versatile solution, although it is very successful at its single intended function.

This form of *interface reduction* can be seen as one of the ways to design a system for the majority of users, and it is currently one of the most successful strategies for ensuring wide acceptance and usage of user interfaces. However, it is impossible, except in very concrete cases, to lower the difficulty of the interface of a complex system to such a level where everyone can use it flawlessly. Furthermore, it is a relatively straightforward process to simplify the interface of a single-function device, but becomes increasingly harder as devices become more complex and versatile, as is the case for many robots which, at this point, expose only APIs as interfaces.

We propose user adaptiveness as an alternative solution for designing systems for every user. By detecting (or learning) that a user is experienced in the usage of a device, a user-adaptive system can stimulate the user into learning more about the device itself and making use of more advanced functions. Thus the system becomes almost a "tutor of itself", potentially lowering the knowledge entry barrier of these systems to even lower levels than those that can be achieved by extremely simplified interfaces by exploiting the natural processes already in place in the human mind.

Ethical Considerations A clear technological hurdle is the necessity of these systems to make use of personal data for adaptation. Indeed, some of the most intricate and successful user-adaptive robots make use of extremely personal, identifiable and sensitive information, such as the user's personality, emotional patterns or medical data.

Users are naturally reluctant to supply this information to systems they do not know, and with no knowledge of how this information can be used in the future. This issue is tackled, in lab tests, by anonymizing data and employing transparent procedures in data collection and manipulation, thus complying with legal requirements. However commercial systems are naturally opaque, leading to a lack of clarity in the treatment of personal user data, akin to the phenomenon observed in services such as Google accounts[12].

The Paro robot has effectively side-stepped this issue. It has become a successful system while not employing identifying or personal information on its user. However, complex systems cannot take the same route, and will inevitably need to manipulate the personal data of their users.

On the other hand, the GrowMeUp Project [86] has opted for embracing its use of personal data, instead defining a protocol that adheres to the regulations of the countries involved in the studies conducted. This solution is suitable when there are regulations in place that account for the novelty of interactive robots. However, obsolete and restrictive regulation can hinder the performance of studies that could be of vital importance to the development of these technologies. This issue stretches beyond the domain of user-adaptive systems, and the solution to this problem will have to be found for personal and Social Robotics as a whole; the problem of privacy and data regulation pose yet another technological and societal hurdle that user-adaptive systems will need to overcome.

2.9 Summary

2.9.1 Research Gaps in User-Adaptive Systems

Table 2.4 presents a summary of the research gaps uncovered on this survey. These gaps constitute the main outcome of this analysis: the potential lines of future research that were uncovered by this work, and that motivate the remainder of the work in this Thesis.

A number of research gaps still remain to be explored: in general terms, user-adaptive robots need to evolve to match the developments that were observed in the field of HCI over the last few decades. Learning new users efficiently is still a relatively unexplored research avenue, with promising techniques already presented in HCI systems such as recommender systems.- Modelling aspects such as modelling multiple users and using big data are also mostly unexplored, despite solutions existing in other fields.

There needs to be an agreement among researchers as to the proper evaluation techniques of user-adaptive techniques, which consequently would lead to the creation of open databases for benchmarking. Furthermore, user-adaptiveness can be taken to the next level by the employment of user-specific psychological information, such as personality traits, mood or even

[12] An issue which is currently being tackled by the new General Data Protection Regulation by the European Commission.

psychological disorders, which would potentially extend the application range of these systems from the domestic to the clinical environment. Lastly, all of the surveyed works focus on user-adaptive *actuation* in some form. This leaves open the field of user-adaptive *perception*, which would allow a system to adapt its analysis of incoming data to the user it is currently interacting with, potentially improving the performance of state-of-the-art behaviour analysis systems.

2.9.2 Overall Readiness of User-Adaptive Systems

Table 2.5 presents a summary of the technological gaps found in this survey. These gaps contain important aspects that, if taken into account, can contribute to the widespread success of user-adaptive systems.

In order to determine the overall status of the field, in technological terms, we have performed a general TRL analysis of the systems under analysis. The categorization of mature systems is hindered by the opaqueness of the commercial systems under study. Since none of the mature systems have published, to the best of our knowledge, any details on how they handle their data on the user, it is impossible to categorize them unequivocally. For instance, the Buddy robot seems able to know its users' name and age, but *how* that information is gained, which constituted the tipping point of our analysis, is not disclosed.

Systems that make use of no explicit user model for adaptivity are very close to mass public availability. Adaptive systems of this nature have been tested in relevant scenarios both in scientific and non-scientific scenarios, and have shown their ability to operate in a variety of scenarios. Systems of this nature are well-established in the realm of HCI, and are becoming so also in HRI. Taking into account the success of the Pepper robot, and the underlying scientific research on this category of systems, they can be classified, overall, as TRL 8.

Systems that make use of static models have been, as illustrated in Table 2.3, tested in relevant environments, such as model homes or controlled home-like environments. The absence of these techniques in technologically mature robots dictates the insertion of these techniques in TRL 5. Similarly, techniques relying on dynamic user models seem to never have left the laboratory and can thus be classified, at best, as TRL 3. This analysis is illustrated in Fig. 2.6.

By observing Table 2.3 and Fig. 2.6, we can conclude that all of the model-based systems analysed in the previous sections inhabit TRL levels ranging from 1 through 5. Conversely, user-adaptive systems in HCI can be easily categorized as TRL 9, as there are already user-adaptive solutions in broad use, *e.g.* recommender systems. Thus, there is a clear technological gap between HRI and HCI in this matter, which can provide an interesting platform for future developments.

2.9.3 Closing Remarks

In this chapter we have provided an overview of the state of the art on user-adaptive HRI.

This chapter has introduced the basic theoretical concepts that support the remainder of the monograph. It has covered topics such as Machine Learning and Bayesian Programming, Markov Decision Processes, and the multitude of technological tools used throughout the work, both software and hardware.

A twofold exploration of the state of the art was also presented. It covers the scientific

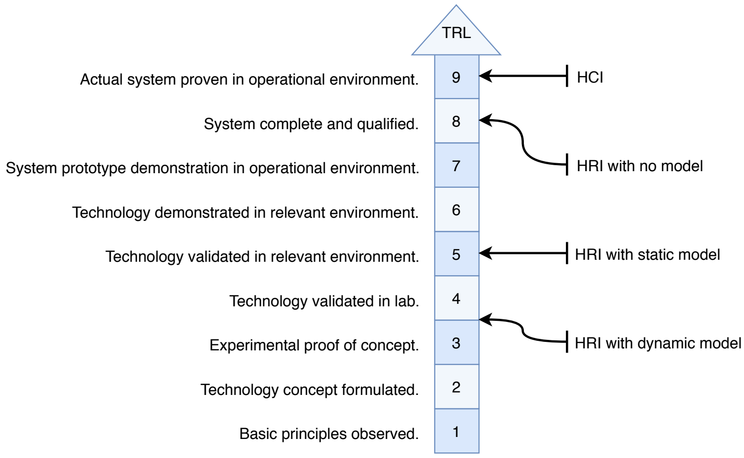


Figure 2.6: Illustration of the general TRLs of the surveyed techniques, according the TRL definition by the European Commission [44]. Sub-fields are ranked in this scale according to the most advanced technique found in our survey.

Table 2.4: A summary of the research gaps uncovered in this analysis.

ID	Gap	Description
1	Learning New Users	There is a need for improved learning mechanisms for efficiently learning new users.
2	Interaction with Multiple Users	None of the works surveyed are able to interact with multiple users simultaneously.
3	Psychological Trait Modelling	There is little work based on psychological constructs.
4	Big Data	There is a gap in the application of Big Data techniques in this context.
5	Continuous Adaptation	No works have ensured the continuous adaptation of user-adaptive robots.
6	User Adaptiveness as a Layer	Can user-adaptiveness be applied generically to all robotic tasks with users?
7	User-Adaptive Perception	Little work is devoted to adaptive perception on robots.
8	Physical and Haptic Systems	Little work is devoted to user-adaptive physical interaction.

Table 2.5: A summary of the technological gaps uncovered in this analysis.

ID	Gap	Description
9	User Skill Level	Most works are not tested with their intended end-users.
10	Long-Term Scenarios	It is still rare to find long-term trials of adaptive systems.
11	Relevant Environments	Most works are not tested in their intended final operation scenario.
12	Metrics and Standardization	There is a lack of automated measurements and standardization to evaluate the user's state.
13	Open Databases	Open HRI databases for testing are very rare.

aspects of the field, enumerating and analysed a number of currently-published systems. This resulted in a number of research and experimental gaps into which the remainder of the work is framed.

In general terms, we can conclude that user-adaptive systems are harnessing the attention of researchers from several fields, in an apparent renaissance of the field since its inception in HCI. We believe that user-adaptiveness in itself constitutes an interesting and rich field of research, as demonstrated by the works surveyed.

We will now address, in Parts II and III, the various stages of a user-adaptive system, as discussed in Chapter 1, starting by presenting our user modelling technique. The research and experimental gaps revealed in this chapter will then be re-linked with the work performed in Chapter 8.

Part II

User Modelling

Chapter 3

BUM: Bayesian User Modelling

> "But what can I tell you? I have known Rodion for a year and a half; he is moody, melancholy, proud, and haughty; recently (and perhaps for much longer than I know) he has been morbidly depressed and over-anxious about his health. He is kind and generous. He doesn't like to display his feelings, and would rather seem heartless than talk about them. Sometimes, however, he is not hypochondria-cal at all, but simply inhumanly cold and unfeeling. Really, it is as if he had two separate personalities, each dominating him alternately."
>
> — Fyodor Dostoevsky, Crime and Punishment

The quote above illustrates an important phenomenon of human life: Razumikhin, the speaking character, describes the main characteristics of his good friend Rodion Raskolnikov, whom he has known for a length of time, has observed and has interacted with often. His description illustrates a process common in social life, through which we *get to know* those that surround us, capturing their main characteristics, representing them mentally as a set of concepts that we can understand and, as in Razumikhin's case, narrate back. Having knowledge on someone, even if only superficialities such as their name, leads to the possibility of adapted behaviour, wherein we model our actions, consciously or not, according to information we have on our interlocutors. Thus, if we are looking to achieve user-adaptive behaviour, gathering knowledge about the user is a critical function to perform.

As seen in the previous chapters, this function of collecting and storing information is fulfilled by a user modelling technique, through which user-adaptive systems are able to learn and represent their users' preferences [70], skill levels [56] and impairments [73][97]. These systems have been widely explored in the field of Human-Computer Interaction (HCI), outperforming non-adaptive systems in multiple metrics and scenarios [35], enjoy widespread attention in this community [106][97][1][35][47][109], and have reached a state of maturity and commercial use *e.g.* in systems such as Netflix or Google Search. This model can be as simple as a single attribute, *e.g.* the user's proficiency in using the system, or as complex as the user's personality [143].

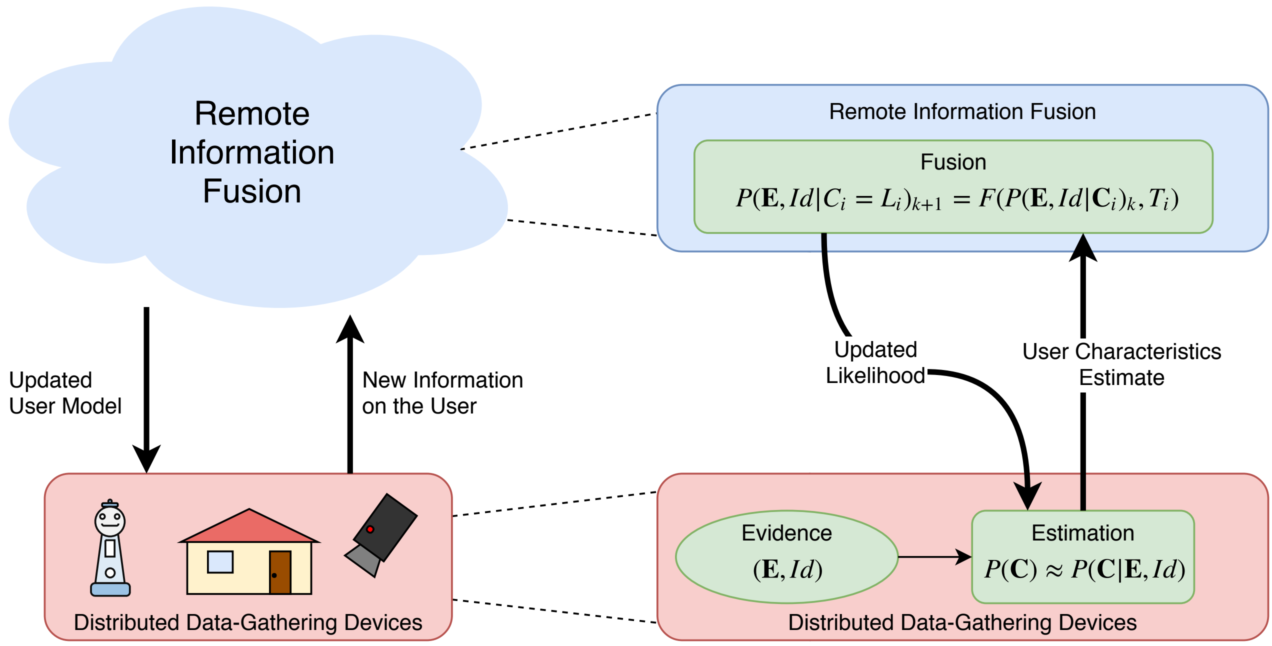


Figure 3.1: An overview of the system. Distributed devices gather new information on the users of the system, generating new estimates, which are remotely fused and re-propagated, allowing the distributed system to gain information on the user.

In this chapter we present a Bayesian User Model (BUM) for determining different characteristics of users, learning them by fusing information from heterogeneous sources. BUM makes use of a Bayesian model to estimate the characteristics of the user from arbitrary input evidence, similar to the Bayesian Programs (Chapter 2) used throughout this monograph. This estimation results in a probability distribution over the characteristics of the user in all dimensions which, for the sake of simplicity, we approximate as a sum of Gaussian distributions. If we apply maximum a posteriori to this distribution, we obtain a set of points that represent the user in an n-dimensional characteristics space, where each individual user is a point.

On-line learning of user characteristics is performed by employing individual data-gathering devices as weak learners and estimators. Each estimation by individual devices is used to improve the knowledge of the system via a centralized entropy-based information fusion system, which is able to add information to the system.

The user characteristics resulting from the estimation process can be clustered, yielding user profiles that represent a portion of the population in each cluster. These clusters isolate the common characteristics of a number of users into a stereotype, a construct widely used in user modelling [97]. These clusters can be useful in two main situations. If one of the system's modules fails, and a number of characteristics can no longer be estimated, the user can be matched to nearby clusters, yielding and plausible approximation of the missing characteristics. Secondly, when the system contacts a new user for the first time, it can use the information from the clusters it has already obtained to serve as preliminary information on this new user until further data has been gathered.

Fig. 3.1 presents a possible implementation of BUM. By splitting the fusion and estimation processes into separate machines, it is possible to have a centralized cloud infrastructure iteratively estimating the new parameters of the user model, which are then fed into the data gathering devices. In turn, these distributed, heterogeneous devices use this model to produce new estimations, which are used to further refine the model itself.

This chapter is structured as follows:

- Section 3.1 presents the goals and key contributions of the chapter;
- Section 3.2 presents some related work relative to user modelling;
- Section 3.3 presents the BUM model, split into the estimation and fusion sub-processes;
- Section 3.4 presents the user profile component of the system;
- Section 3.5 presents the experimental goals and methodology;
- Section 3.6 presents the results their discussion, comparing them to our goals and validating our claims;
- Lastly, Section 3.7 presents a summary of the chapter.

3.1 Chapter Goals and Contributions

We aim to advance the state of the art by proposing a novel user modelling system able to estimate the user's characteristics and fuse these estimations into a new user model. Furthermore, since User Modelling techniques have not fully transitioned into HRI, we also advance the state of the art by proposing a technique that can be seamlessly used with robots. Specifically, we propose a system that:

1. Is able to perform on-line learning of user characteristics, fusing heterogeneous data from distributed sources;
2. Is able to correctly estimate and group them into profiles while operating;
3. Is tolerant and robust to failures in evidence generation and characteristic estimation.

These system features constitute the claims we aim to support with our experiments and discussion.

3.2 Related Work on User Modelling

User Modelling systems have been used to support both Human-Robot Interaction (HRI) and Human-Computer Interaction (HCI). These systems have been under development in the HCI community since, at least, the 80s [97][106], having reached a mature state. The application of these techniques to robotic systems is a relatively infant field, as seen in Chapter 2. As detailed in the aforementioned chapter, user models can either be provided to the modelling technique or learned while the system operates. We could not find any systems in the state of the art that are able to seamlessly combine information gathered from heterogeneous sources, and able to be implemented in multiple topologies, including combinations of Social Robots and external sensor networks.

In the latter category, information on the user can be obtained in two main ways: by asking the user for input (explicit) or without user intervention (implicit) [35]. Explicit data gathering can be seen, for instance, in [41], wherein the static Personas are built from questionnaire data. Implicit data gathering is very common among works that make use of dynamic user models, for instance in [70] where interaction traces are used to infer user preferences, and in [105] where the user's adaptability to the robot's plan is inferred through their actions. In this chapter, we present a system that aims to combine these two approaches, making it possible to learn the user's characteristics both by implicit and explicit means.

Three concepts developed in User Modelling for HCI serve as the basis for this work: model interoperability, shell systems and user model servers, which we describe below.

Model Interoperability consists of the degree to which different user modelling systems are able to integrate their information for building more comprehensive models, aiming for a holistic representation of the user [25]. Interoperable systems become very interesting when the adaptive system is decentralized or composed of heterogeneous devices, since high interoperability allows for a more efficient use of information.

Shell Systems are "empty" user modelling systems, that have to be populated with domain knowledge, such as the model's structure and update rules, in order to function [74]. These systems are very versatile, with the same system being able to be used in a variety of applications.

User Modelling Servers consist of services operating on the client-server model, wherein a user model is constructed by a server that receives information from one or more clients [47][25]. This model can then be requested by clients, which can use them to adapt in the context of their specific application. User modelling servers avoid the issue of interoperability by centralizing the construction of the user model. This implies the usual disadvantages of centralized systems, such as the increased local workload and the possibility of catastrophic centralized failure.

BUM draws from each of these concepts to form a unified user modelling approach. The concept of interoperability can be seen in the way BUM combines information from heterogeneous sources: by making the observations produced from each source compatible with one another, we achieve interoperability. Shell systems form the conceptual basis of the fusion process: it is as generic as possible, and is populated with information from the distributed devices. Similarly, user modelling servers served as inspiration for the example implementation of Fig. 3.1, where a remote server is used to fuse the information on the population of users.

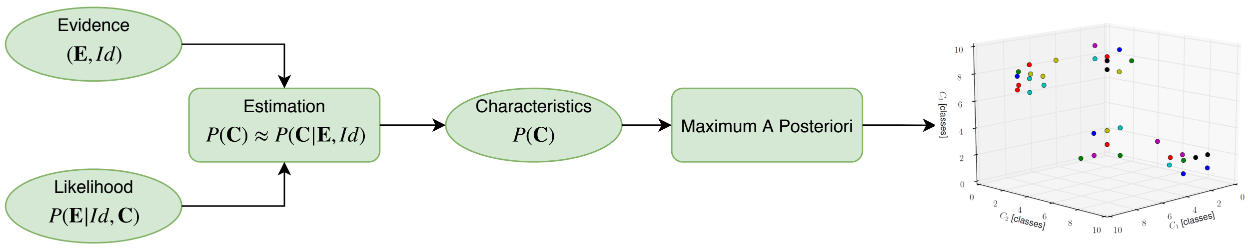


Figure 3.2: An illustration of the estimation process. User characteristics are estimated via Bayesian programming, yielding the main output of the system. Via Maximum-a-Posteriori, a characteristics-space representation of the population can be extracted, where each point represents a user.

3.3 Estimation and Learning of User Characteristics

3.3.1 Estimation of User Characteristics

The main goal of the model is to infer a vector of the user's characteristics, $\mathbf{C} \in \mathbb{R}^n$, where n is the number of user characteristics under study. This process is dubbed Estimation, as illustrated in Fig. 3.2. This process takes as input a vector of evidence $\mathbf{E} \in \mathbb{R}^m$, where m is the number of evidence variables, and the user's identity $Id \in \mathbb{N}$. The main output of the system is the distribution

$$P(\mathbf{C}|\mathbf{E}, Id) \propto P(\mathbf{C})P(\mathbf{E}, Id|\mathbf{C}), \tag{3.1}$$

which encodes the user Id's' characteristics revealed by the evidence. By using the user's identity as an additional evidence variable, the model is able to represent a population of users while still allowing for individual personalization.

This estimation process is split into modules, with each module inferring one of the characteristics in the $\mathbf{C}$ vector. This modularity is set on the assumption that all characteristics are statistically independent from one another. For each model, a Bayesian Program [46] is used to infer the characteristic, solving Eq. 3.2. Once estimated, the inferred characteristics can be used by the underlying system, for instance, for user-adaptive interaction or to generate new labels for learning. Furthermore, for the purposes of implementation, we also assume that all evidence variables are independent, allowing us to decompose Eq. 3.2 into

$$P(\mathbf{C}|\mathbf{E}, Id) \propto P(\mathbf{C}) \prod_{E_i \in \mathbf{E}} P(E_i, Id|\mathbf{C}). \tag{3.2}$$

While not strictly necessary, this assumption allows for a much simpler implementation of the model.

A characteristics-space representation of all users is then obtained, as illustrated in Fig. 3.2. This representation is obtained by performing *maximum a posteriori* estimation for all characteristics of a user, producing a point in characteristics space for each user. Each point in defined as

$$\mathbf{U}_u = [C_1, C_2, ..., C_n], \tag{3.3}$$

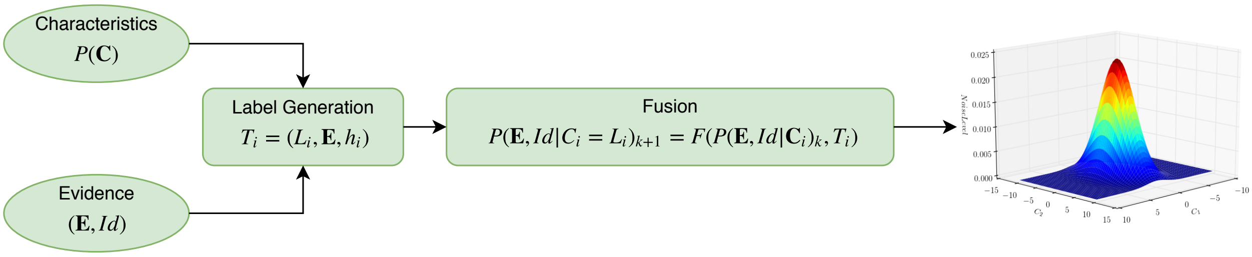


Figure 3.3: An illustration of the information fusion process. Evidence and estimations are received by the system, which are packed into an independent tuple. Each tuple is fused into the unified likelihood to be propagated among the estimation modules, thus increasing the global knowledge of the system.

where n is the number of characteristics being inferred, u is the identity of the user, and $\mathbf{U}$ is the vector containing all users. In this formulation, different users can show the same characteristic for different inputs and, conversely, different users can show different characteristics for the same inputs. Furthermore, there is no limitation in how many users can display exactly the same characteristics, which can lead to superimposed users in the characteristics-space representation.

3.3.2 Information Fusion and Learning

As illustrated in Fig. 3.3, the systems uses the result of the estimation step to update the common representation of the users in the system, and is able to combine both soft and hard labels, *i.e.* information obtained from the system's self-learning and ground-truth information.

For each estimation of the system, a tuple is generated:

$$T_i = (L_i, \mathbf{E}, h_i), \tag{3.4}$$

where $L_i \in \mathbb{N}$ is the label obtained for characteristic C_i, via Maximum a Posteriori estimation:

$$L_i = \underset{x}{\operatorname{argmax}} \; P(C_i|\mathbf{E}, Id). \tag{3.5}$$

h_i is the entropy of the distribution $P(\mathbf{C}_i))$, which we approximate by the posterior obtained from the estimation step, $P(\mathbf{C_i}|\mathbf{E}, Id)$:

$$h_i = H(P(\mathbf{C}_i)) \approx H(P(\mathbf{C_i}|\mathbf{E}, Id)). \tag{3.6}$$

H is the entropy function, as defined in [134]:

$$H(X) = -\int_{\mathbb{X}} P(x) \log P(x)\, dx \tag{3.7}$$

The system's common information is stored in a likelihood of the form $P(\mathbf{E}, Id|\mathbf{C})$. This represents the aggregate knowledge of the whole system, and is used for determining the user's characteristics, as seen in Section 3.3.1, through the estimation process.

This distribution is iteratively constructed as a Gaussian kernel by performing

$$P(\mathbf{E}, Id|C_i = L_i)_{k+1} = \frac{1}{\psi}(P(\mathbf{E}, Id|C_i = L_i)_k + D) \tag{3.8}$$

where D is a learning factor, function of T_i, defined according to the label received and ψ is a normalization factor, ensuring that the resulting probability distribution is valid. This sub-process is performed every time a new tuple T_i is received by the system.

By employing this fusion mechanism, and in addition to the benefits presented in Section 3.1, the system becomes able to learn faster through the parallelisation of the learning procedure. Additionally, by employing individual devices as soft learners, the system can operate without the express need for supervision. This is achieved by the fact that the distributed system can integrate information from various devices simultaneously, thus allowing for the integration of various pieces of information on each user without the limitation of one-on-one interaction. There is no limit to how many devices estimate the same characteristic of the user; many devices can estimate the same characteristic from different sets of evidence, thus contributing to the system's global knowledge of the population.

Soft Labels

Soft labels, *i.e.* labels generated by the system's distributed classifiers, are fused according to the entropy they contain:

$$D = P(\mathbf{E}, Id|C_i = c)_{observed} = \mathcal{N}(\mu, \Sigma) \tag{3.9}$$

where μ is set to the value of the evidence received and Σ is a covariance matrix where each diagonal element is defined by entropy:

$$\Sigma_{i,i} = F(h_i), \tag{3.10}$$

which is calculated in Eq. 3.6, and the remaining elements set to zero.

Hard Labels

Hard labels, *i.e.* labels received from an external labeller, which encode ground-truth data, are also fused through the Gaussian kernel principle:

$$D = \mathcal{N}(\mu, \Sigma) \tag{3.11}$$

μ is defined as before. Σ is defined according to Eq. 3.10, where h_i is set to a very low value, conveying the high certainty of information received from the external labeller.

These can be obtained, for instance, by directly asking the user a question, in an approach resembling active learning [23]. By relying on this mechanism, the system is unable to deal with users who purposefully mislead it, a problem which we consider out of the scope of this work.

3.4 User Profiling through Clustering

User profiles are obtained from the characteristics-space representation of the users, as illustrated in Fig. 3.2. The Expectation-Maximization [46] algorithm is used for clustering the users, producing a Gaussian mixture on characteristics space. This results in a set of n-dimensional Gaussian distributions, which are the main output of this sub-process:

$$(M, \Sigma) = EM(\mathbf{U}), \tag{3.12}$$

where M contains the means of the clusters, and Σ the respective covariance matrices. EM denotes the Expectation-Maximization algorithm.

The algorithm consists of alternating the expectation step:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right] \tag{3.13}$$

and the maximization step:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \tag{3.14}$$

until a convergence condition, defined by the difference of the model obtained in subsequent iterations, is reached.

Each of the resulting clusters can be interpreted as a *user profile*. A profile represents a common "type" of user found by the system when combining its knowledge on their characteristics. Thus, the system operates on the whole population, revealing inter-identity relationships among users which can later be used in adaptive processes.

As with any unsupervised learning technique, these profiles present two key challenges: the determination of the number of clusters to optimize and the attribution of semantic value to the clusters. Information-theoretic techniques can be used to determine the number of clusters that best fits the data [46]. In our case, we opted for the exploitation of *a priori* knowledge for the definition of the number of profiles, as discussed in Section 3.5. Attributing semantic value to the profiles can be achieved by superimposing a semantic sectioning of the characteristics space. For instance, if 3D space is split in 8 octants, each octant can be attributed to a certain semantic value, and each cluster "named" according to the octant they fall in.

3.4.1 User Profile Classification

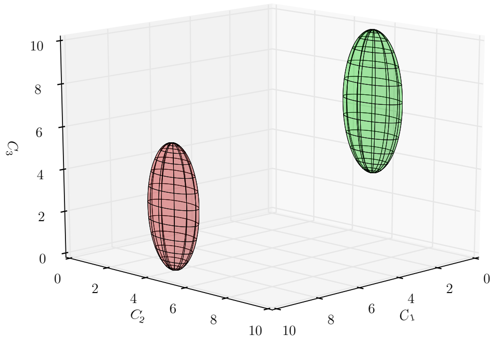
User profiles can be used, for instance, to endow the system with a degree of fault-tolerance in case one or more of the user's characteristics can no longer be estimated due to a system failure, or two quickly estimate the missing characteristics of a new user. Users are matched to the cluster that minimizes a distance metric. Assuming a user given by a vector with incomplete characteristics:

$$\mathbf{U}_{inc} = [C_0, ..., C_{k-1}, C_{k+1}, ..., C_n], \tag{3.15}$$

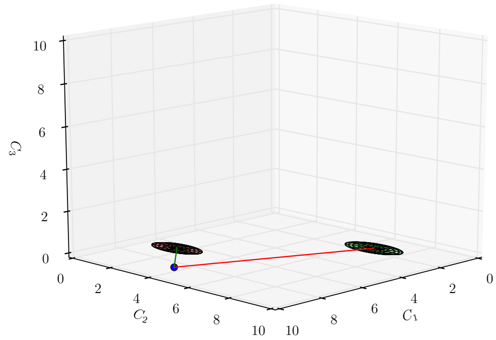
i.e. as defined in Eq. 3.3 except for the missing k-th component, C_k.

The distance from this user to each of the clusters is obtained through

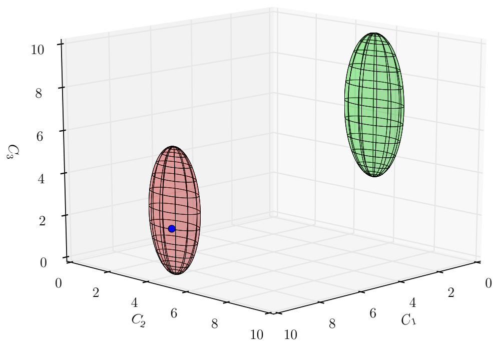
$$d_i(\mathbf{C}, \boldsymbol{\Sigma}) = \sqrt{\sum_{C_j \in \mathbf{U}_{inc}} (C_j - \Sigma_j)^2}, \tag{3.16}$$



(a) Clusters are pre-calculated by the system during normal operation, representing the whole user population.



(b) Lacking information on C_3, clusters are projected and the user is matched to the nearest cluster.



(c) Remaining characteristic is estimated from the matched cluster, re-projecting the user into the full characteristics space.

Figure 3.4: An illustration of the user matching procedure. The pre-existing clusters are used to estimate any missing characteristics of the user by projecting the clusters into the remaining dimensions and matching the estimated user to the nearest cluster. The remaining missing characteristics are then extracted from the matched cluster.

where Σ_j is the element of cluster i corresponding to the characteristic C_j of the user. d_i corresponds to the Euclidean distance over the projections of clusters in $n-1$-dimensional space, *i.e.* excluding the dimension that could not be estimated due to a fault. The closest cluster to the user is selected by minimizing the distance:

$$\Sigma^* = \underset{\Sigma}{\operatorname{argmin}} \ d_i(\mathbf{C}, \mathbf{\Sigma}) \tag{3.17}$$

Once the user is matched to a cluster, each of the missing characteristics would then be obtained from the means of the clusters by performing

$$C_j = \Sigma_j^* \tag{3.18}$$

for every missing characteristic C_i. This process is illustrated in Fig. 3.4, and could be easily expanded to a larger number of missing characteristics.

3.5 Experiments

3.5.1 Goals and Metrics

The goal of our experiments is to evaluate the system's performance and support the claims presented in Section 3.1. In order to evaluate the system's performance, we employ the following performance metrics:

- Estimation error, ϵ;
- K-L Divergence, D_{KL};

The estimation error ϵ is defined as:

$$\epsilon = |x_{est} - x_{real}| \tag{3.19}$$

where x_{est} and x_{real} represent the characteristic as it was estimated and it ground-truth value, respectively.

The Kullback–Leibler divergence is defined for continuous distributions as

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x, \tag{3.20}$$

which is equal to

$$D_{\mathrm{KL}} = E_p[log(p(x)/q(x))], \tag{3.21}$$

which we compute numerically with a Monte Carlo [46] approximation based on 10^5 samples of the Gaussian mixture.

Estimation Process for One Characteristic

program

description

specification

Variables:
$\mathbf{C} \in \mathbb{R}^n$, $\mathbf{E} \in \mathbb{R}^m$, $Id \in \mathbb{N}$
Decomposition:
$P(C_i, \mathbf{E}, Id) \propto P(C_i)P(\mathbf{E}, Id|C_i) = P(C_i)\prod_{j=1}^{m} P(E_j, Id|C_i)$
Formulation:
$P(C_i)$: Uniform distribution.
$P(E_i, Id|C_1)$: Gaussian kernel.

Identification:
$P(C_i)$: Uniform distribution.
$P(E_j, Id|C_i)$: Fusion Process.

Question:
$P(C_i|E_1, ..., E_n, Id)$

Figure 3.5: Bayesian Program for estimating one of the user's characteristics. We make use of Gaussian kernels to represent the model's likelihoods, which start as uniform distributions.

3.5.2 Model Materialization

We have implemented our model to estimate three main characteristics of the user:

- User's preferred speech volume (C_1).
- User's preferred distance to the robot (C_2);
- User's talkativeness (C_3)

We consider that a talkative user speaks at maximum 5 words per second, corresponding to 300 words per minute. This approaches the maximum speaking speed of a normal person, with the recommended value for narration being of 180 words per minute [159]. This information allowed us to create a seed model for the estimation of C_3, which is thus learned from soft evidence only, *i.e.* the user is never asked about their talkativeness during interaction.

Each characteristic had a source information (evidence):

- C_1: The robot periodically asks the user if it is talking too loudly;
- C_2: The robot periodically asks the user if it is too far away from them;
- C_3: The average speaking speed of the user for each reply for the past five replies.

These characteristics were estimated, according to the formulation of Section 3.3.1, using a Bayesian program, illustrated in Fig. 3.5.

We have opted to determine the number of clusters to use *a priori*. For the simulated dataset, the number of clusters to determine was the same as the profiles injected in the simulated population. Regarding the real dataset, the correct number of clusters was determined through multiple attempts.

input : Simulated $[\mathbf{C}, \mathbf{E}, Id]$ for all population
output: $P(\mathbf{C}|\mathbf{E}, Id)$, performance metrics
visited_combinations = {};
for *i = 0; i ¡ total_iterations; i++* **do**
    $[\mathbf{C}, \mathbf{E}, Id]$ = random sample from population;
    $[\mathbf{L}, h]$ = estimate($[\mathbf{E}, Id]$);
    if $[\mathbf{E}, Id] \notin$ *visited_combinations* **then**
        T_i = generate_label($\mathbf{L}_i = \mathbf{C}, \mathbf{E}, h = 0.001$);
        visited_combinations $\leftarrow [\mathbf{E}, Id]$;
    else
        T_i = generate_label($\mathbf{L}_i = \mathbf{L}_i, \mathbf{E}, h_i = h$);
    end
    fuse(T_i);
end

Algorithm 1: An algorithm illustrating the operation of our experimental set-up. The system runs iteratively, and hard evidence is provided for unknown evidence combinations.

3.5.3 Synthetic Dataset Generation

As defined in the previous sections, the model accepts a great variety of implementations. We have opted for an implementation involving a simulated team of social robots and cloud infrastructure. The fusion sub-process takes place on the cloud, with the remaining sub-processes taking place on each robot. Label vectors ($\mathbf{T}$) flow from each data-gathering robot to the cloud, where they are fused with previous information, resulting in new likelihoods which are propagated to the team of robots, as described in Section 3.3.

A test-bench involving a simulated team of robots and user population was developed. The user population was randomly generated according to a fixed set of profiles given to the simulator. Uniformly-distributed random noise was added to the fixed profiles to generate a richer population.

The system was operated iteratively, as illustrated in Algorithm 1. Randomly selected evidence was sampled from the population and used to simulate sensory input. Evidence was fed to the Estimation process, generating soft labels for fusion. For the first iteration of each combination of input variables, the label generator was given the ground truth classification for the evidence at hand, as illustrated by `visited_combinations` variable in Alg. 1. Thus we could allow the system to start with uniform likelihoods, and to converge on its own based on the seed model formed by the initial hard evidence provided. The system was operated until convergence or for a fixed number of cycles. A fault-tolerance scenario was also developed, in which one of the modules of the system is deactivated after a number of iterations, and later a number of new users are added to the population. This experiment allows us to gain insight into the system's ability to tolerate faults, and also to integrate new users into its population.

```
Input  : Q: Normal questions to be asked,
         V: Questions about volume,
         D: Questions about distance.
Output: T: Tuples used for training the BUM system.
asked_vol = False;
asked_dist = False;
talk_evidence = [];
while Q ≠ {} do
    // Remove random question from Q
    q = Q.pop();
    // Ask the question via robot.
    reply, time = ask(q);
    // Generate evidence from response.
    e = reply.n_words() / time;
    talk_evidence.append(e);
    send_evidence(talk_evidence);
    // Ask about volume
    if asked_vol == False and rand() ¡ 0.25 then
        asked_vol = True;
        q_v = V.pop();
        reply, time = ask(q);
        adjust_volume(reply);
        send_tuple(reply);
    end
    // Ask about distance
    if asked_dist == False and rand() ¡ 0.25 then
        asked_dist = True;
        q_v = D.pop();
        reply, time = ask(q);
        adjust_distance(reply);
        send_tuple(reply);
    end
end
```

Algorithm 2: An algorithm illustrating the operation of the user-adaptive decision-making module used for data gathering. The system gets random questions from a pool, which are prompted to the user. Randomly, the system asks about the distance or volume that it is using to communicate, and the replies used to generate hard evidence.

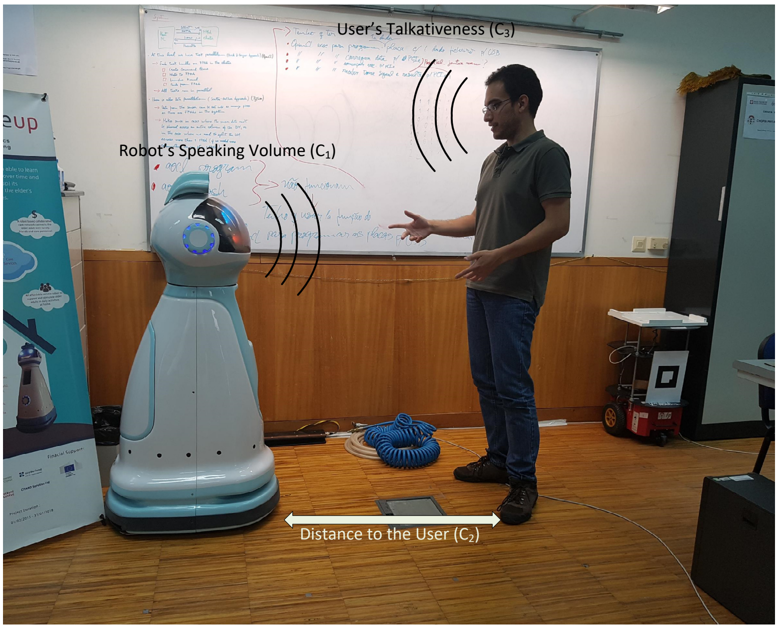


Figure 3.6: A picture of the experimental setup used for data gathering. The user faced the robot, speaking naturally while answering the robot's questions. This allowed for the collection of all necessary measurements, as represented.

3.5.4 Real Dataset Collection

The GrowMu robot (Chapter 2), was employed as the main interactive device and data source for these tests. A simple user-adaptive decision-making technique was developed for the purposes of the data gathering, described in Alg. 2. 8 participants participated in the study by interacting with the robot. All test subjects were male, with ages ranging between 20 and 35 years.

The specific scenario, from the perspective of the user, consists of the following:

> The robot engages the user in casual conversation on topics such as their education, their research experience and the weather. While taking turns speaking, the robot sometimes asks the user if they are happy with the current volume and speaking distance. The user's answers are used to adapt the robot's operation to the user.

This scenario follows the principles of the main scenario of Section 2.2.

Each session with a user proceeded as follows:

1. Users were introduced to the research being performed and to what data would be gathered;

2. Users were asked about each individual characteristic, establishing a baseline "self-assessment";

3. The users interacted freely with the system;

4. The session was concluded.

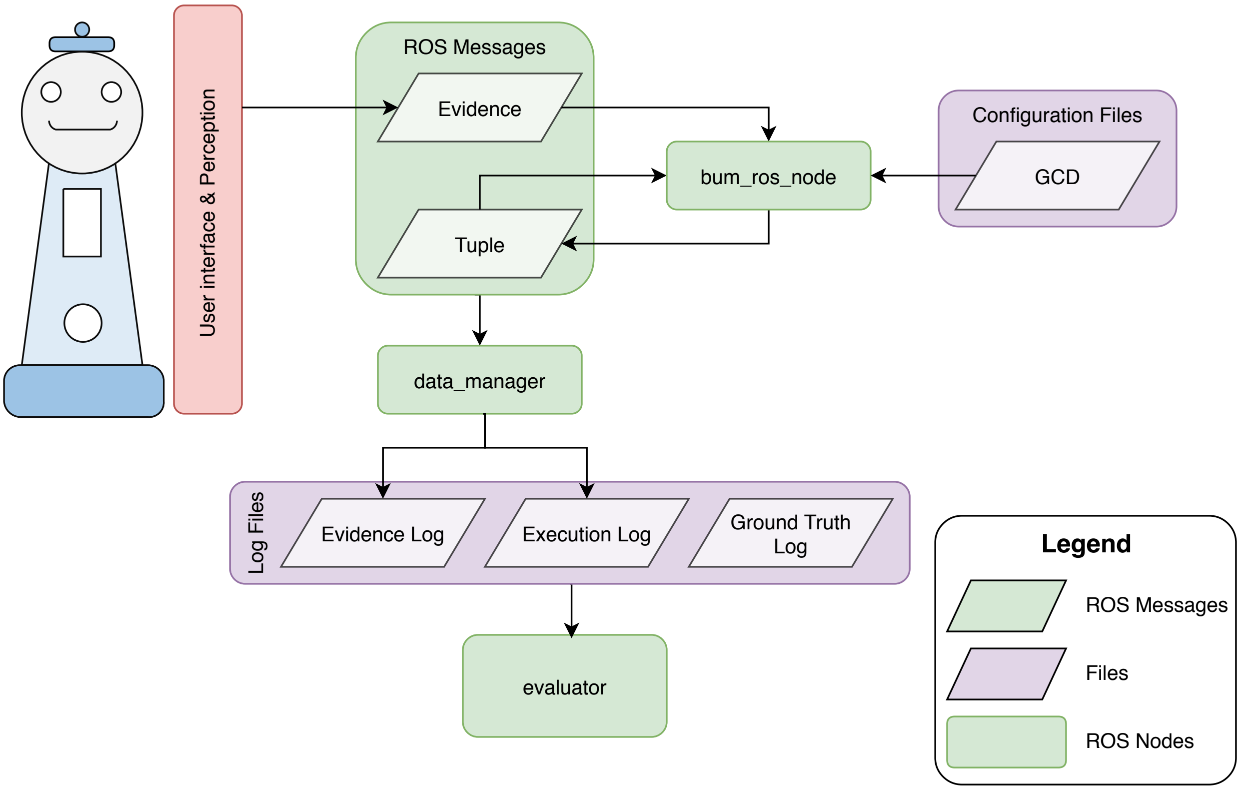


Figure 3.7: An overview of the ROS implementation of BUM. Evidence and Tuples are propagated as custom ROS messages, and are processed by `bum_ros` nodes, which use them to estimate and learn the users' characteristics. This node is configured by a Global Characteristic Definition (GCD) file. All data processed and produced by the system is recorded by the `data_manager`, and can later be evaluated by the `evaluator`.

The sessions took place in a laboratory setting, illustrated in Fig 3.6. Each interactive session with the robot lasted for approximately 10 minutes of continuous, turn-based conversation.

The sessions with users resulted in a dataset that, given the flexibility of the system, is analysed and processed in several ways, allowing for various tests to be performed. Namely, the dataset was used for performing tests with the data from a single user, as well as for the full population, by selectively playing back and interleaving the information obtained during the tests. The baseline measurements performed allowed for the comparison between the robot's results and the user's self-evaluation, thus re-defining the estimation error (ϵ) measurement to

$$\epsilon = |x_{est} - x_{self\_assessment}| \tag{3.22}$$

in the context of this analysis.

3.5.5 Implementation

In order to test the system with multiple topologies, and in both simulation and real trials, a modular `bum_ros` system was developed. It is depicted in Fig. 3.7, and is composed of three main ROS nodes:

1. The `bum_ros_node` node, responsible for all of the main functions described in this Chapter, including estimation and fusion;

2. The `data_manager` node, responsible for the management of all data produced and received by the system, including generating and playing back log files;

3. The `evaluator` node, mainly used for visualization and system evaluation purposes.

The source code for this implementation is freely available under the GPLv3 open source license[1].

The `bum_ros` system communicates through two main types of information: tuples containing estimation results, and evidence collected from the user. This information is contained in two custom ROS messages:

1. The `Tuple` message, containing the results of estimation as well as hard evidence received by the system. By default, these flow in the `bum_ros/tuple` topic;

2. The `Evidence` message, containing regular evidence received by the system, used for estimation. By default, these flow in the `bum_ros/evidence` topic.

In order to make our experiments easily repeatable, the system needed to maintain logs of its knowledge and of the information received. The system's logs are stored in three files, according to the data they keep:

1. The Ground Truth Log (`gt_log`), which keeps ground truth data created by a human annotator to be used for evaluation purposes;

2. The Evidence Log (`ev_log`), which keeps all of the evidence received by the system, including any hard evidence received from an annotator in `Tuple` messages;

3. The Execution Log (`exec_log`), which maintains all of the estimation results produced by the system.

A main configuration file, the Global Characteristic Definition (GCD) is also necessary for regular operation, and each individual instance of the `bum_ros` system expects to find the path to a GCD in its parameters. This configuration file describes the structure of the inference problem, including which variables are involved in the process, and their interdependence assumptions, allowing for the inference mechanism to be instantiated dynamically as needed.

Lastly, the system is supported by a hardware abstraction layer (HAL). This layer must be custom-built for each data-gathering device. The only requirement that `bum_ros` imposes on this layer is that it conform to the GCD when producing `Evidence` messages. No other synchronization methods are employed, and the system is able to operate in a completely asynchronous manner. In fact, the `Evidence` messages produced do not even need to correspond to the evidence needed by any particular characteristic. The HAL can also produce `Tuple` messages containing hard evidence, if the interaction they implement allows for the gathering of information that is somehow confirmed by the user, such as by answering the system's questions.

In the simplest operation mode, with a single device, the operation flows as follows:

[1] `https://github.com/gondsm/bum`

1. The HAL produces an `Evidence` message;
2. This message is simultaneously logged to the log file and processed by the `bum_ros_node`, potentially producing new estimations (if enough evidence variables are received);
3. If estimation occurs, this produces a `Tuple` message, which is simultaneously logged to the Execution Log, and re-used by the `bum_ros_node` for fusion.

This modular implementation has allowed for the testing of the technique in various conditions, including the simulated and real tests reported in this chapter. It is also described in more detail in [91].

3.6 Results and Discussion

3.6.1 Results

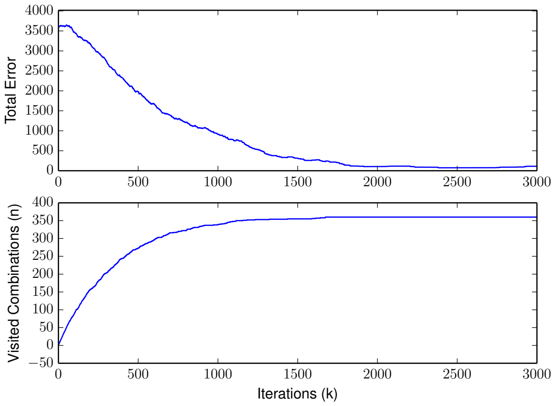
Fig 3.8 shows the results obtained as the system interacts with both the simulated and real population. Fig. 3.8a shows the evolution of the system's accuracy as new combinations of users and evidence are visited for the simulated population. We can observe that the global estimation error decreases monotonically as the number of iterations (and interactions) increases, demonstrating the fusion of new data into the system's representation of the users.

Fig. 3.8b shows the evolution of the estimation error and user characteristics as the system is fed data from the interaction. We can observe that, as the system receives tuples, its estimation on the user's characteristics improve, achieving zero estimation error after a few iterations.

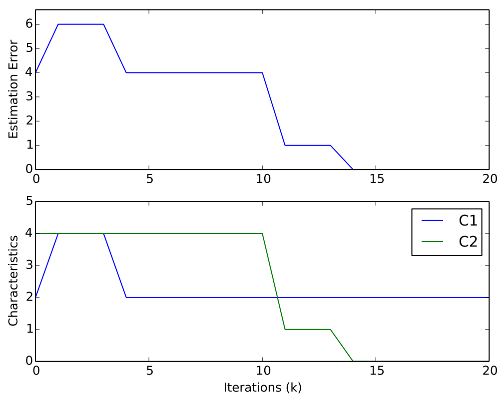
Fig. 3.8c shows the evolution of the estimation error when the system is run on the full dataset, *i.e.* with all the users which interacted with the system. Error rate tends to lower as the system gains information on the users, despite the progressive addition of more users to the system. The error rate lowers to a shallow boundary, unable to reach a null error. This is due to the discrepancies observed between the characteristics reported by the users and the hard evidence gathered by the robot during interaction, illustrated in Fig. 3.8d. Thus, the user's self-assessment, while an important tool for guiding the experimental effort, cannot be used as ground-truth for the system to operate on.

Fig. 3.9 illustrates the evolution of the population user profiles determined, as well as of the K-L divergence of the associated Gaussian mixture models when compared to the reference user profiles extracted from the reference population. We can observe that, as the number of interactions increases and the system incorporates new data, the characteristics of all users converge to the ground-truth. The clusters also converge, thus better capturing the inter-identity relationships present in the data.

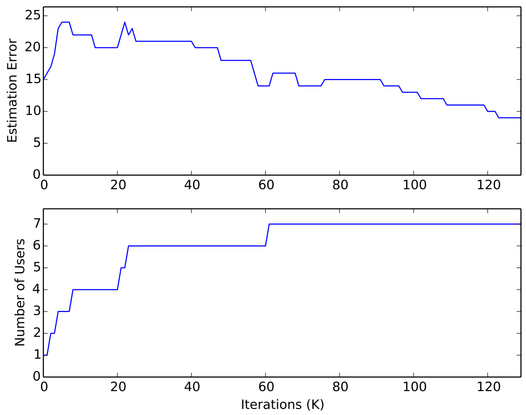
Fig. 3.10 illustrates the results obtained for the fault-tolerance scenario. Fig. 3.10a corresponds to the fault-tolerance scenario running on the simulated population. We can observe, on iteration 1000, that the estimation error stagnates, a sign of the fault in the system, but that by making use of the clusters, the system is able to maintain a roughly constant error. We can also observe a large error spike on iteration 2000, where a new population of users is added. While still operating with a failed module, the system is able to learn these new users, achieving a decreasing trend in estimation error comparable to that observed before system



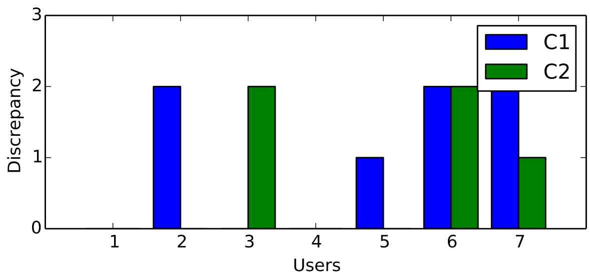
(a) An illustration of the evolution of the error of the system as more combinations of evidence are learned when operating on synthetic data.



(b) An illustration of the evolution of the error rate and characteristics estimate for a run with a single real user. In this case, the system successfully determined all user characteristics.



(c) An illustration of the evolution of the error rate and number of users when testing the system with the full real dataset. We can observe that the error plateaus, despite a general decreasing trend.



(d) An illustration of the discrepancies observed between the real users' self-assessment and the hard evidence retrieved by the system. Each bar represents the deviation between the self-assessment provided by the respective user and the hard evidence gathered by the robot, indicating the magnitude of estimation error expected for that user.

Figure 3.8: Results obtained for our proof-of-concept, with the system operating both on the synthetic and the real dataset.

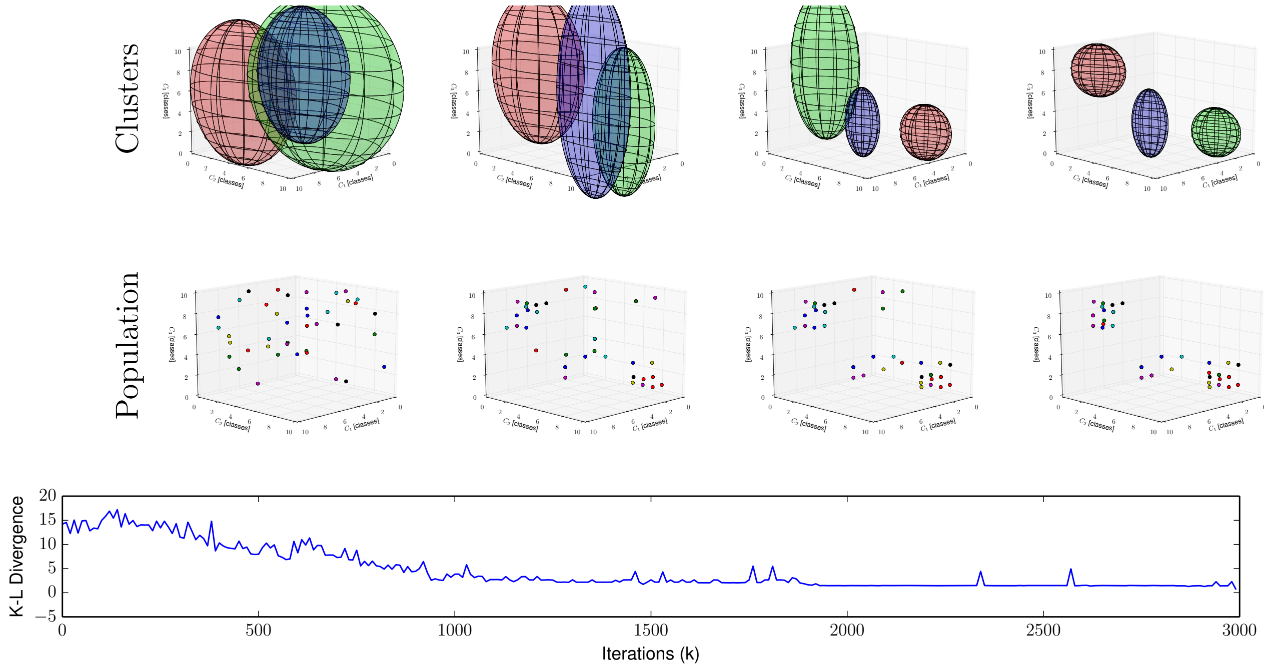


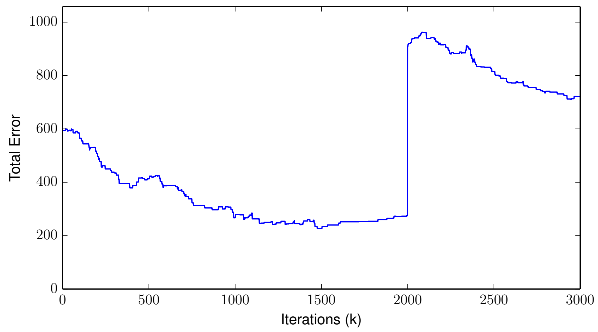
Figure 3.9: An illustration of the evolution of the clustered user population, profiles and the K-L divergence measurement. As the system gains information on the population, the characteristics-space representation and clusters become more accurate. This is further illustrated by the decreasing K-L divergence between the reference population and the clusters at each iteration. Some spikes can be observed in the K-L divergence due to the inherent randomness of the expectation-maximization process.

failure. Fig. 3.10b illustrates the same behaviour of the system when one of the modules fails when operating on the real dataset.

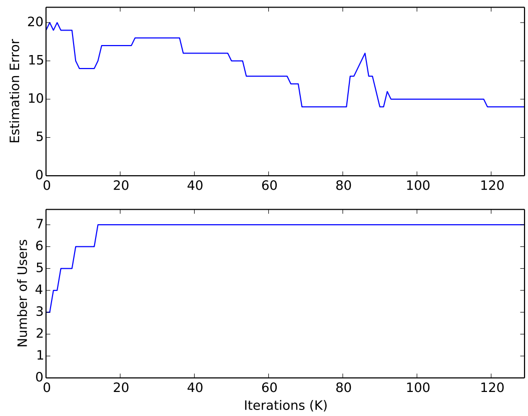
3.6.2 Discussion

The results of a single run of the system, illustrated in Fig. 3.8, demonstrate that the system is able to achieve low estimation error, and to do so when starting from no information on the user population. This figure shows that the system is able to learn new characteristics from the heterogeneous information provided, and consequently improve its estimation error while operating.

Figs. 3.8b and 3.8c show that, even with the relatively lower amount of real data obtained, when compared to the synthetic dataset, the system was still able to estimate the user's characteristics, as well as to incorporate new information as it was received. In both tests, convergence to a low error rate was achieved, demonstrating that the model can learn and correctly estimate the user's characteristics with comparable performance for a single or multiple users. The estimated characteristics tend to converge and stabilize as new information is received, achieving a stable representation of the user. This representation is generally aligned with the user's self-assessment, as demonstrated by the lowering error rates observed. Thus, our results



(a) Results of the fault-tolerance scenario on synthetic data. One of the modules was deactivated on iteration 1000, and new users were added to the system on iteration 2000.



(b) Evolution of the error rate when testing the system's fault-tolerance on real data. A module was disabled on iteration 60.

Figure 3.10: The results obtained for the fault-tolerance scenario. One of the system's modules was disabled at iteration 1000, and a new population of users was added at iteration 2000.

support claim 1.

It is interesting to note that the user's self-assessment differs significantly from the measurements gathered by the robot during direct interaction, as illustrated in Figs 3.8b and 3.8c. In these figures, we observe a remainder of estimation error, *i.e.* a lower boundary of error below which the system does not improve. This boundary corresponds to the deviation between the user's self-assessment and the characteristics estimated by the robot during direct interaction, illustrated in Fig. 3.8d. This deviation occurs despite the fact that the users indicated to the robot that they felt comfortable with its actions and, as thus, ensured that that was their preference. These discrepancies can be attributed to several sources, such as small inconsistencies in the operation of the robot while gathering ground truth, or the user changing their mind when they start interacting with the system.

By observing the evolution of the population and clusters presented in Fig. 3.9, we can see that the system is able to cluster users into user profiles. As indicated by the low K-L divergence achieved upon convergence, the user profiles obtained match the profiles obtained from the reference population. The decreasing nature of the K-L divergence observed can also be taken as evidence that the system is learning from interactions, thus further supporting claim 2.

The results of the fault-tolerance scenario, presented in Fig.3.10, illustrate how the system is able to continue operation despite a failure in one of the modules, both in the synthetic and real datasets. We can observe that the system is able to maintain performance and learn a new population of users, albeit in a slower manner than before. The system was also able to achieve descriptive user profiles for the population with real data, as illustrated by the recovery observed after the failure point in both cases. The obtained clusters are used, as described in Section 3.4, for the estimation of characteristics when the corresponding modules fail, achieving a lowering error rate that matches what was observed before the failure. Thus, our results support claim 3.

3.6.3 Edge Cases

Despite supporting our claims, the system presents a number of theoretical particularities that warrant discussion. The system will have a tendency to reinforce its seed model while it is executing. As such, if the system receives as input the same evidence repeatedly, its likelihood may deteriorate to the point where neighbouring evidence values will be mis-estimated. In this case, the system displays behaviour observed in Fig. 3.8c, although for different reasons, *i.e.* reaching a performance plateau.

The current fusion mechanism takes into account all evidences equally. If there is a mismatch in the discriminatory power of each evidence value, the system may be integrating unnecessary information into the model. This allows the learning of tuples even if the label is not consonant with all evidence, potentially learning erroneous information.

3.7 Summary

In this Chapter we have presented BUM, a Bayesian User Model able to learn the characteristics of a user population by gathering information from distributed sources. We have presented the formulation of the model, its materialization into a test-able model, and its validation using synthetic and real datasets.

We have shown that the system is able to learn and accurately classify the users' characteristics, and to generalize users into profiles consisting of a Gaussian Mixture via the Expectation-Maximization clustering technique, supporting our initial claims. Additionally, we have shown that the system is fault-tolerant, being able to compensate for the loss of an estimator, and also able to cope with the addition of new users.

BUM is able to learn the characteristics of a population of users, condensing them into a representation that can be used to inform overlying systems. However, the relationship between the evidence vector and tangible, measurable characteristics of the user was left out of the scope of the work; this relationship is assumed to be true. Chapter 4 presents a study that aims to establish a social robot's ability to determine the psychological characteristics of its user, with basis on the classical scales employed in Psychology, demonstrating a means to ground this relationship in concrete signals.

Chapter 4

Psychbot: Psychological Evaluation using a Social Robot

> "At a particular time, my senses inform me of a shape, colour, hardness, taste that belongs to the wax. But at another time my senses inform me of a different shape etc. belonging to the wax. My senses show me *nothing but* these diverse qualities (which we can call "sensory qualities", since our senses take them in). I nevertheless make a judgement of identity: it is the same piece of wax on the earlier and the later occasion. So, it is the nature of the ball of wax that it can possess different sensory qualities at different times. So, to understand what the wax *is* I must use my understanding, not my senses."
>
> — Simon Blackburn, Think

As Blackburn put, true knowledge (in the rationalist perspective of epistemology) lies in the bridge between sensory input and reasoning. An intermediate step needs to take place between sensation and knowledge, wherein the agent evaluates the signals they are receiving against their history and the knowledge it possesses (*e.g.* wax melts, and therefore the ball of wax and the pool of wax can bear the same identity). Thus, in order to endow a robotic system with the ability to properly model its user, it has to be able to operate, to some degree, on the sensory data it is receiving, and obtain from it true knowledge, or approximately so. This problem is commonly known as *perception* [46].

In this chapter, we are interesting in endowing a robot which is equipped, for instance, with the BUM system of Chapter 3, with the ability to perceive the user's psychological state. This state can include aspects such as their personality, their mood, signs of depression, resilience, among others. In classical Psychology [93], this is achieved using validated scales, usually under the form of self-administered questionnaires, which we aim to translate into HRI. This process carries the natural advantage of having been extremely well-studied and perfected throughout the years, with the field of Psychometrics having achieved a very high level of maturation; these techniques are employed every day by professionals throughout the world for study and diagnosis of the human psyche, resulting in countless successful treatments being administered. However, it carries a few important drawbacks, namely this process:

- is **unwieldy** and **unnatural**, with some questionnaires consisting of hundreds of items which cause its administration to last very long periods of time;
- demands **high-level expertise** in the administration of the scales, particularly to the elderly and children;
- is **invasive**, as its application always demands the transmission of sensitive information to another person (even if a health professional);
- must be performed in **discrete**, **formal** moments in time, and is therefore naturally not continuous and unable to provide continuous insight into the user's status.

Automating this process, distilling it into a natural interaction between a user and a robot, could greatly benefit the user and potentially mitigate all of these drawbacks.

This chapter presents a study conducted with a sample of 25 elderly users, which were exposed both to a an interaction with a robot and to the classical collection method, both aiming to determine the same psychological characteristics of the user. In essence, the goal is to show that there is potential in the automation of this process, even if only for preliminary screenings, by showing that the answers obtained by the classical and new methods share enough similarity. This demonstration could pave the way for autonomous psychological, by establishing a clear relationship between gathered signals and the user's true psychological status.

4.1 Chapter Goals and Contributions

An important body of work in Personality Computing [152][151], which in part tackles this problem by devising novel methodologies for perceiving, recognizing and synthesizing the personality of human users. However, these works are limited in scope, being only applied to Personality, and in application, generally not making use of social robots.

Some works have employed personality in Social Robotics, as analysed in Chapter 2, such as [143]. These works share the same limitations in scope as the works in Personality Computing, applying only personality to model the user's psyche. Furthermore, these works do not focus on the determination of the user's psychological characteristics themselves, nor on exploring the connections between the robot's perception of the user and the actual results found in formal testing.

In general terms, this chapter aims to bridge the gap between the classical psychological evaluation techniques and the natural, casual interaction that can be provided by a social robot. We aim to partially transpose the classical method in a way that a robot can apply it, and determine whether the robot's results are comparable with the ones achieved by the classical methodology. We implemented a natural dialogue in a social robot, which presents to the user an adapted version of the classical scales, and matches their answer to a predefined Likert scale. Specifically, we aim to disprove the following null hypothesis:

H_0: There is no correlation between the answers given to the robot and those given to the human evaluator applying the classical method;

Disproving this hypothesis will demonstrate that our system can be used to gain information on the targeted psychological dimensions, even if only for preliminary screenings.

4.2 Assessment of Psychological Characteristics

The determination of the psychological characteristics of users, the object of the field of Psychometrics, is usually performed through two main complementary methods: self-assessment scales, under the form of questionnaires, and interviews. A very large body of work has been dedicated, in this field, to the study of how the responses to these scales relate to the behaviours and characteristics externalized by the subjects under study. The general consensus indicates that well-validated scales exhibit a strong relationship between the answers they elicit and the user's characteristics, enabling them to be used as evaluation techniques. Furthermore, validated scales exhibit interesting characteristics such as internal consistency, the ability to elicit consonant responses throughout the questionnaire, and inter-evaluator consistency, the ability to elicit consonant responses regardless of the human evaluator who applies them.

In general terms, evaluation scales consist of a number of questions, or items, which are graded in a Likert scale, usually ranging from two to seven points. Items are normally applied to the subject according to a fixed protocol, which has been validated in large-scale studies. In the case of elderly subjects, the scales are applied with the aid of a Psychologist, who aids in the reading, comprehension and filling in of the scale, taking care to disturb the results as little as possible, despite some level of evaluator-induced bias being unavoidable.

In order to account for this, scales are evaluated according to the inter-evaluator consistency they present, *i.e.* the level of variability in the results when related to who is applying the scale to the user. This metric is part of the evaluation of the scale itself, with better scales being more resistant to the influence introduced by the evaluator.

In this chapter we are interested in the psychological constructs and scales detailed in Table 4.1. These scales were selected by Psychology experts for providing an overview of

Table 4.1: Psychological constructs and scales taken into account in this study.

Construct	Scale	Ref. (Original Scale)	Ref. (Portuguese Version)
Personality	NEO-FFI	[96]	[81]
Resilience	Resilience Scale	[154]	n/a
Memory	Subjective Memory Complaints Scale	[130]	n/a
Health Status	SF-8	[146]	[117]
Quality of Life	WHOQOL-OLD	[114]	[150]
Depression	GDS-30	[162]	n/a
Social Desirability	EDS-20	n/a	[5]
Mood	PANAS	[33]	n/a

the psychological state of the subjects, and for being of particular interest in the case of the elderly. Validated scales in the Portuguese language were used, as indicated in Table 4.1, as to match the language spoken by our population sample. These scales aim to measure 9 different psychological constructs of each subject, namely:

Personality Personality is one of the most widely studied psychological constructs. Several models have been developed, such as the Meyer-Briggs Type Indicator [95] or Eysenck's Big Three [45]. These models aim to analyse common patterns in human behaviour and, through statistical analysis, determine a number of independent characteristics which, once combined, can inform on the user's most common behaviours and thought patterns.

Currently, the most popular model of personality is the Five-Factor model, or the "Big Five" model [54][93]. This model splits human personality info five basic traits, the combination of which should ideally constitute a complete definition of a subject's personality, and which are used as a tool when analysing human behaviour. The five factors are as follows:

- **Openness** is correlated with how easily a person decides to subject themselves to new experiences. This trait describes the level of intellectual curiosity of the subject, and is also correlated to their creativeness and independence.
- **Conscientiousness** indicates how careful, organized and mindful a person is, for instance, of their duties. Conscientious subjects tend to be self-disciplined and dutiful, and is also correlated with obsessive behaviour.
- **Extroversion** describes a person's tendency to be outgoing, energetic, talkative and sociable. Extroverted subjects tend to show dominance in social situations.
- **Agreeableness** describes the subject's tendency to be compassionate, friendly or forgiving. Low-agreeableness subjects tend to be described as stubborn, argumentative and suspicious of others.
- **Neuroticism** is related to how prone a subject is to experience emotional stress and variability. A neurotic subject will tend to experience higher emotional variability as a result of relatively less significantly stimuli.

This construct was deemed necessary in this study given the importance of Personality in the general behaviour of the subjects, providing a strong, general and persistent explanation of their behaviour. Personality is also one of the most popular psychological factors explored in HRI, as mentioned before in Chapter 2.

Resilience Psychological resilience describes a subject's ability to emotionally cope with adverse situations. Resilient individuals are able to deal with stressful situations, such as those arising from their familiar context, workplace, health or finances, maintaining a positive outlook through unfavourable conditions. Resilience is an ability, can be successfully trained in most individuals, and is particularly effective if employed in a social group, such as a family. Resilience was selected for this study as it provides insight into how the user is able to deal with adverse conditions, which can be of particular importance when evaluating elderly subjects who may be less equipped, in terms of social support, to deal with these situations.

Memory Memory is the faculty of the human mind responsible for storing and retrieving information. It is of extreme importance to all individuals, as it maintains all of the elements used for learning, adapting and forming an individual identity. Mild memory loss is a natural result of the process of ageing, and is particularly aggravated in the presence of certain conditions, such as Alzheimer's disease. Due to its strong effect on the subject's daily life, memory loss is in interesting candidate for preliminary evaluation via self-report scales, often serving as a first step towards the diagnosis of memory-associated conditions.

Health Status The general health status scale allows for an overview of the the subject's self-perceived health status. It can be an important measurement when combined, for instance, with depression scales, allowing for the formulation of hypotheses as to the causes of some of the subject's psychological aspects.

Quality of Life Quality of life describes the general well-being of the subject's, and how generally satisfied they are with life, their daily living, social relationships, independence, *etc*, and the quality of life scale allows for an overview of the subject's self-perception of this aspect of their lives. Seeing as the elderly tend to experience a particular decline in quality of life, it becomes important to obtain insight into this particular aspect of the subjects at hand.

Depression Depression is a mental disorder commonly associated with consistently negative emotional states, low self-esteem, loss of interest in previously-enjoyed activities, low energy and, in more serious cases, substance abuse and suicidal thoughts. It is one of the most common factors in deaths by suicide, and is becoming alarmingly prevalent in some segments of the population, namely the elderly. Given its prevalence and strong impact on the subject's daily life, depression is a natural candidate for scale-based pre-diagnosis, which is reinforced by the availability of relatively short validated scales.

Social Desirability Social desirability is a response bias that leads subjects to respond to questions in ways that they expect will project their best image, namely in the evaluator's perception. Thus, when sensitive matters are approached, subjects seeking social desirability will tend to respond in the way that they believe is "right" or "good", according to whatever moral compass or social norms they believe the evaluator follows, and not necessarily truthfully. For instance, subjects may have a tendency to skew responses describing their personality, over-inflating the aspects that they believe reflect better on themselves, denying important aspects such as depressive or unhappy states, substance abuse, *etc*. Employing social desirability scales allows the researcher to account for this bias: for instance if a subject responds to both a depression and social desirability scale, scoring average in the first and high in the second, the first scale's results should be interpreted with less certainty on the evaluator's part. This ability to provide meta-insight into the subject and the other results simultaneously makes this construct of particular importance in our evaluation.

Mood Mood is an unfocused transient emotional state which shapes the subject's expectations of incoming external stimuli, such as pleasure or pain, generally influencing how the subject experiences daily life. Unlike instant reactions to stimuli, moods can last for days or weeks, and being diffuse in nature are harder to cope with. This construct gains importance in this test because of its high temporal variability, allowing us to detect inconsistencies in the robot's evaluation.

4.3 Implementation of the Classical Scales on a Social Robot

A reduced version of the combination of the scales of Table 4.1 was devised by Psychology experts for implementation in the robot. This scale included slightly rephrased adaptations of the most statistically significant subset of items from the original scales. Both the phrasing of the items themselves and of the possible answers were adapted to ensure that the questions generated during the interaction sounded natural to the user. The Likert scales employed in the scales were also adapted, *e.g.* compressing 7-point items into 5 points and standardizing the response scale, ensuring that they could be applied by the robot.

Care was taken not to pollute the semantics of each item, maintaining their phrasing as close to the original as possible. Changes were, thus, contained as much as possible to the syntactical changes needed to have the items be spoken in the first person and as direct questions. This effort resulted in a final 55-item questionnaire, which was applied to each user in full.

This questionnaire was used by the robot as a basis to communicate with the user, and was the main stimulus applied to the user. The interaction system performed three main functions:

- **Item selection**, in which the next item to be evaluated was selected from a database.
- **Utterance generation**, in which the selected item was combined with connective text and a temporal reference to produce an understandable utterance.
- **Answer parsing and rephrasing**, responsible for determining which of the item's possible answers the user was referring to, narrowing down the possible answers if needed.

As in the scenario of Chapter 1, the interaction took place as an iterative process, where each iteration consisted of the robot posing a question to the user and receiving their response, as illustrated in Fig 4.1.

4.3.1 Utterance Selection and Generation

The adapted questionnaire had to be synthesized by the robot, thus stimulating the user. To achieve this effect, each item was paired with a temporal reference, which together resulted in an understandable and cohesive sentence.

At each iteration, the system selected a new item at random, without replacement, to present to the user. Item selection was kept uniformly random in order to avoid introducing an additional bias based on the item order; every different user was exposed to a different sequence of questions, thus avoiding the bias.

After selection, each item is combined with a temporal reference that matches the specific item's intended time window, such as "in the past two weeks" or "in general terms". Connective text is also used in order to link the item text with the temporal reference, *e.g.* "do you feel

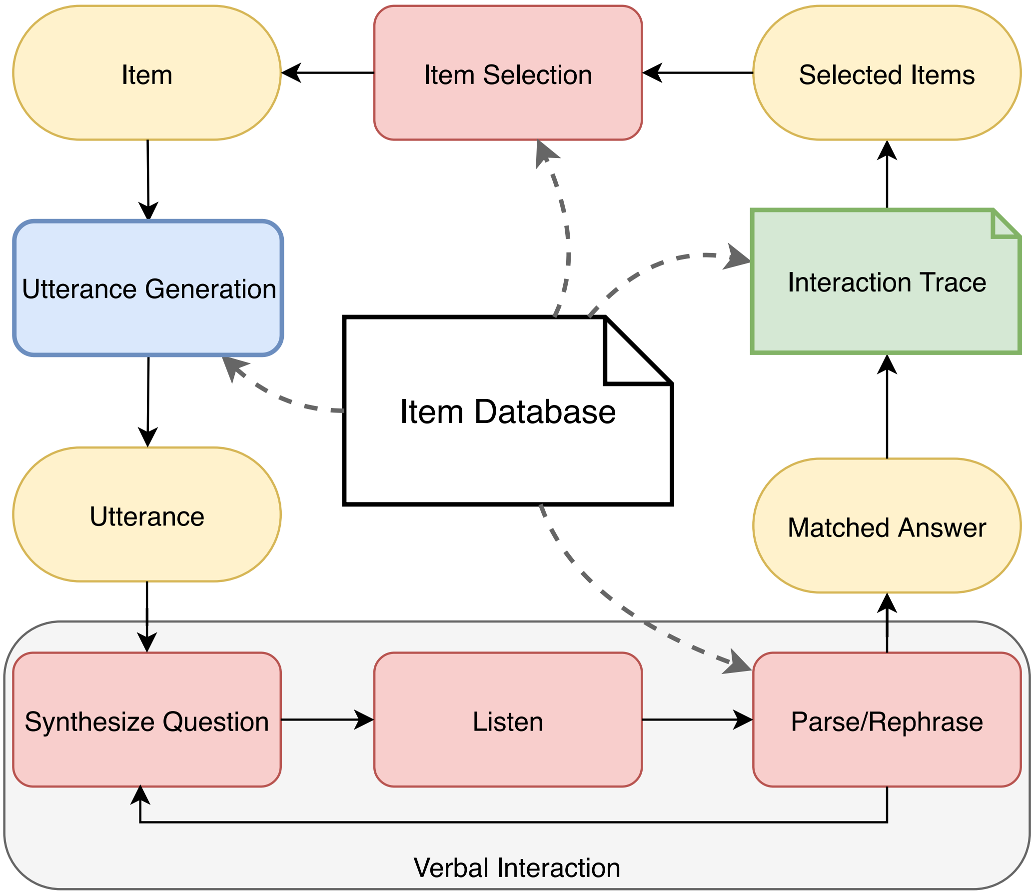


Figure 4.1: The interaction loop that was implemented on the robot to perform the automatic screening. Items are randomly selected from a database, and synthesize-able utterances are generated from them. Then the verbal interaction takes place, the result of which is used to fill in an interaction trace, which is then used to inform the item selection procedure.

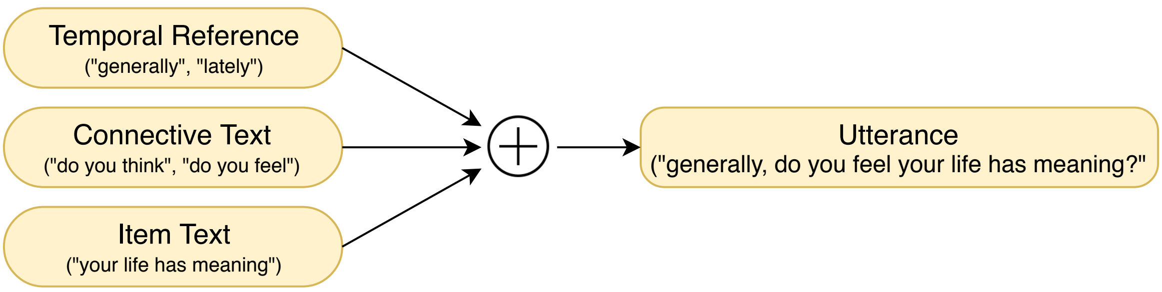


Figure 4.2: An illustration of how synthesize-able utterances are generated. A temporal reference, connective text and the item's adapted text are combined into a cohesive, grammatically correct and understandable utterance.

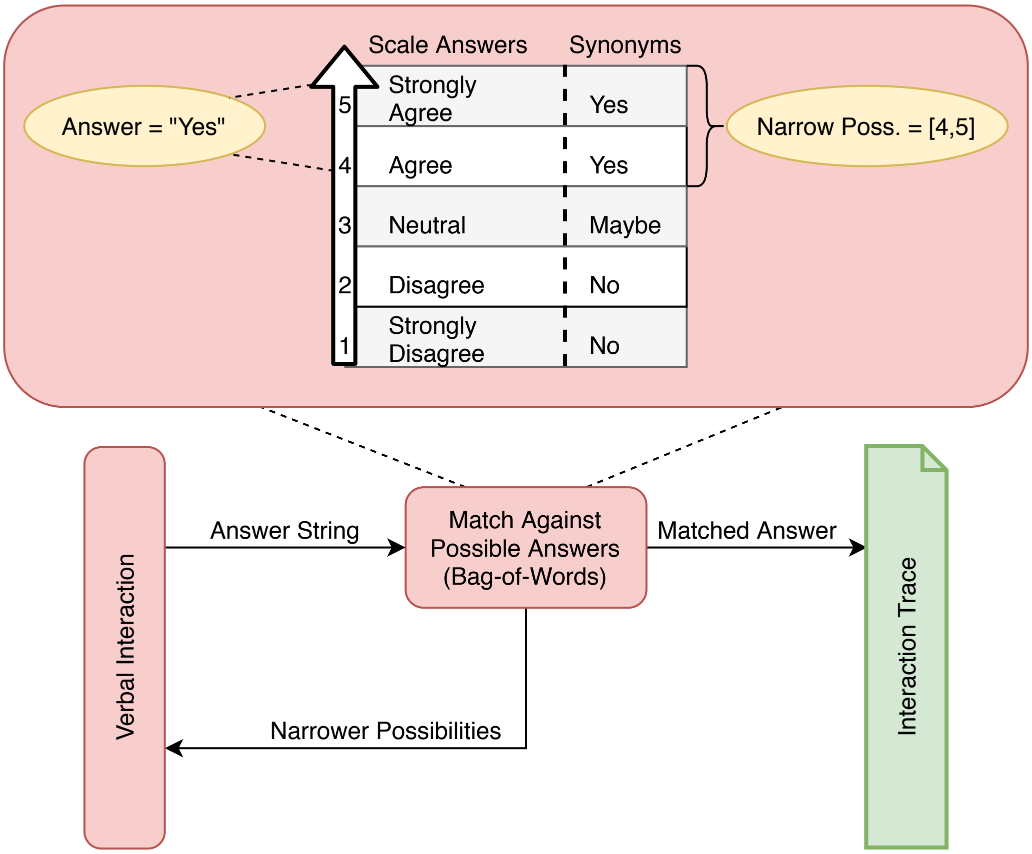


Figure 4.3: An illustration of the parsing and rephrasing mechanism. The answers are matched against the original answers and a set of "synonyms". If enough ambiguity occurs, as in the example, a narrower set of possible answers is sent back to verbal interaction. This process iteratively narrows down the set of available answers until only one of them remains, which is accepted as the user's intended answer.

that", "do you think that". This results in a synthesize-able string which is sent for speech synthesis, as illustrated in Fig 4.2.

4.3.2 Parsing and Rephrasing

After the question is synthesized, the user is allowed to reply to the system as they see fit. Their answer results in a string returned by the speech recognition module, which is automatically matched to the possible answers to the item via a bag-of-words approach, as illustrated in 4.3.

Let A be a bag-of-words obtained from the user's reply. For each possible answer to the item, i, let B_i be the bag-of-words obtained from the answer. Using set arithmetic, we can define a distance:

$$D_i = \frac{\mathbf{card}(A - B_i) + \mathbf{card}(B_i - A)}{\mathbf{card}(A) + \mathbf{card}(B_i)}, \tag{4.1}$$

resulting in $D_i \in [0, 1]$ which measures how close the user's answer is to the possible answer i. Thus, we can match the user's answer to the possible answer which minimizes the D metric, *i.e.* that better matches the possible answer.

If an ambiguous match is found, *i.e.* if the user's answer does not match any of the possible answer, the system discards the lowest matches and presents the remainder to the user. This mechanism mimics a technique widely used by Psychology professionals when delivering these scales, where when given an ambiguous answer, the evaluator will bisect the scale in order to narrow down the user's true opinion. Thus, when there is ambiguity in the user's answer, the system presents to the user a number of possible answers, contiguous on the scale, so that the user can narrow down their answer.

Following the results of the pre-tests detailed in Section 4.5.1, a number of synonyms were added to the possible answers. These enlarged the array of possible answers to each item, doubling or tripling them in most cases, which allowed for a more natural interaction with the robot. Following this addition, the user could answer most questions with common phrases (such as "yes" or "no"), and the system would automatically narrow down their answer to a smaller number of possible answers. This procedure is liable to introduce additional error in the estimation, but generally reduced the amount of time needed to reach a definite answer, and visibly reduced the levels of user frustration during tests, as observed informally by researchers and caregivers.

4.4 Experiments

In order to validate the approach, and ascertain the validity of the hypothesis of Section 4.1, tests took place with an elderly population. The experiments were split into three main phases: a focus group validation of the selected items and constructs, a pre-tests phase used to hone the system, and the main tests, in which the bulk of the data was gathered.

The focus group session was used, essentially, to ensure that the constructs selected for evaluation were indeed important, from the perspective of the users. The session consisted of approximately two hours of open, informal conversation with a sample of potential users, who were selected for being perceived by their caregivers as particularly open-minded and willing to discuss psychological issues openly. The session was recorded and transcribed for analysis. Most importantly, all of the concepts under evaluation were spontaneously proposed by the users as important during the session, which demonstrated their pertinence.

For both the pre-trial and main trial phases, each subject was exposed to two interactions:

- An interaction with the robot, where the robot autonomously administered its adapted scale;
- A session with a Psychology professional, where the classical method was employed to administer the full scales of Table 4.1.

The goals of the pre-trial phase were two-fold:

- Hone the interaction system to a point where it was usable;
- Collect preliminary results that could indicate whether the system was promising enough for a large-scale dataset collection and evaluation.

Figure 4.4: An elderly user interacting with the Psychbot system implemented on the GrowMu robot.

Three pre-trial sessions took place, with development cycles taking place between them. No major changes to the system were made during this phase, as mostly quality-of-life improvements were needed: synonyms were introduced, the robot's pronunciation was tuned, and the matching algorithm was set as as aggressive as possible to minimize the repetitiveness of the interaction, as mentioned before.

Once the pre-tests had demonstrated that the system was promising enough, a large-scale dataset collection phase took place. In this phase, each session with the system consisted of the robot administering the same adapted 55-item scale. This was complemented by the human evaluator, who administered the original full scales, for a total of 194 questions per session with the human evaluator.

Every session took place in a private office, safeguarding the subject's privacy. The temporal relationship between the session with the robot and with the human evaluator was not fixed, to avoid introducing unnecessary bias. Each session with the robot followed a rigid structure:

1. Introduction: the subject was introduced to the study, informed of their right to privacy, anonymity and to leave the study at any time with no consequence whatsoever.

2. Interaction: the interacted with the robot/human evaluator until all of the questions were answered.

3. Conclusion: the user was shortly debriefed on the interaction, with the experimenter asking informal questions on how the system could be improved or on what their thoughts were on the interaction with the human evaluator.

The sessions with the human evaluator took place in a one-on-one setting, while during the sessions with the robot a total of three people were present in the room: the subject, a caregiver and an experimenter. The caregiver was tasked with keeping the user comfortable, and was responsible for their well-being. When absolutely necessary to keep the user from becoming frustrated, the caregiver helped the user understand the robot's question by repeating it. The experimenter in the room was tasked with making sure the system was running was operating correctly. The system operated in a completely autonomous manner, making its own decisions while interacting with the user, but was not autonomously recognizing their speech. This was due to the fact that, early in the pre-trial phase, it became clear that the speech recognition system being employed[1] was not able to accurately recognize the elderly user's speech in a manner that avoided frustration (*i.e.* avoiding excess repetition). As such, the experimenter was also responsible for inputting the user's speech into the system, which then processed it exactly as if it had been received from the speech recognition module.

In total, 25 elderly users took part in the study, all aged 60+ years old and deemed mentally fit for this trial by their caregivers. 6 of those users participated in the focus group session, 3 in the pre-trial phase and the remainder in the main trial phase. The pre-trial phase resulted in 165 data points. Interaction with the robot typically lasted for 30 minutes, while the professional sessions normally lasted between 30 and 90 minutes. In total, the robot asked 1029 questions, and the Psychology professional 3285, resulting in 1029 final data points from the intersection of the two evaluations.

4.4.1 Metrics and Evaluation

With the goal of evaluating the system's performance and disproving the null hypothesis of Section 4.1, three metrics were used: the mean absolute error (MAE), the Pearson correlation [40] and the Spearman correlation [40]. The mean absolute error is defined as

$$mae = \frac{1}{N}\sum_{i=0}^{N}|a_{i,human} - a_{i,robot}| \tag{4.2}$$

where N is the total number of answers analysed, $a_{i,human}$ and $a_{i,robot}$ are the normalized answer given to the human evaluator and robot, respectively. Normalized answers are defined as

$$a_n = \frac{a}{N_a} \tag{4.3}$$

where N_a is the number of possible answers to the specific question. Thus, all of the answers under analysis are $a \in [0, 1]$, regardless of any adaptations made to the Likert scales in use during the implementation of the scales on the robotic system.

The Pearson correlation is defined as

$$r_p = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.4}$$

where r_p is the Pearson correlation coefficient, and x and y are the two samples under analysis, which correspond to the set of answers given to the human and robot evaluators.

The Spearman correlation is defined as

$$r_s = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X}\sigma_{\mathrm{rg}_Y}} \tag{4.5}$$

where r_s is the Spearman correlation coefficient, rg is the ranking operator, cov is the covariance operator, and σ is the standard deviation operator.

[1]The speech recognition system used, developed within GrowMeUp, is available at `http://wiki.ros.org/speech_recog_uc`.

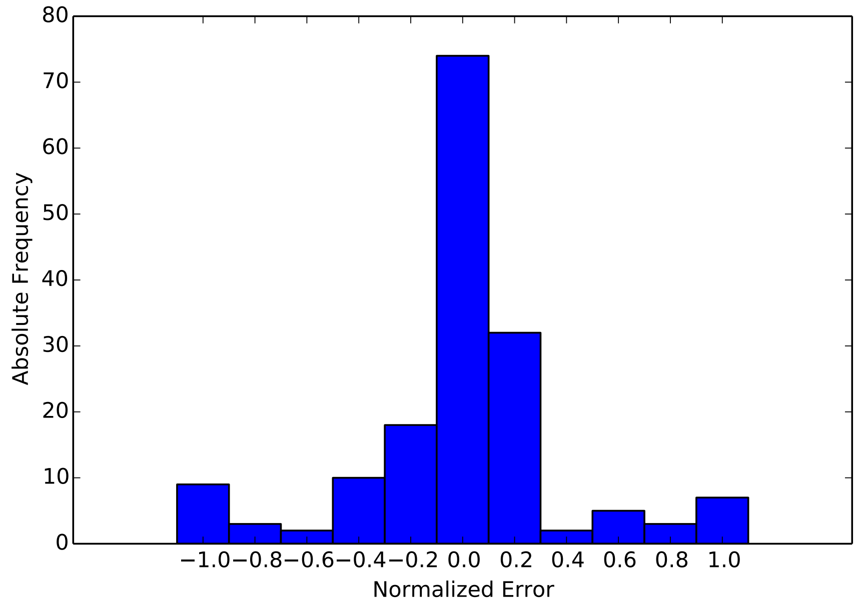


Figure 4.5: A histogram illustrating the distribution of the normalized error for all of the answers collected during the pre-tests.

4.5 Results and Discussion

4.5.1 Pre-Tests

Fig. 4.5 presents the distribution of the normalized error committed by the automated system with respect to the answers provided to the human evaluator. We can observe that the error is well-centred on zero, meaning that in the majority of cases the system makes relatively small mistakes, with 70% of the answers falling within 30% of normalized error, as illustrated in Fig. 4.6. These results are promising, and seem to disprove the null hypotheses of Section 4.1: in order to obtain these results, the answers given to the robot must apparently bear a relationship with those obtained by the human evaluator. The error roughly follows a Gaussian distribution, which could correspond to the inter-evaluator error which can be observed when validating the classical scales themselves. In other words, the robot's results differ from those of the human evaluator in the way that would be expected by a mere change in the human evaluator themselves, further pointing to the validity of the system. Thus, the distribution of the error committed by the system when performing its evaluation shows promise: it tends to obtain answers that, while not complete matches with the human evaluator, are generally consonant with them.

Fig. 4.7 illustrates the global distribution of answers collected by the human evaluator and the robot. We can observe that the users tended to provide comparable distributions of answers to the human and robot evaluators with the exception fourth bin which, in the case of five-point scales, corresponds to the boundary between the fourth and fifth higher-scored answers. In these cases, the phrasing of the robot version of the questionnaire were quite close

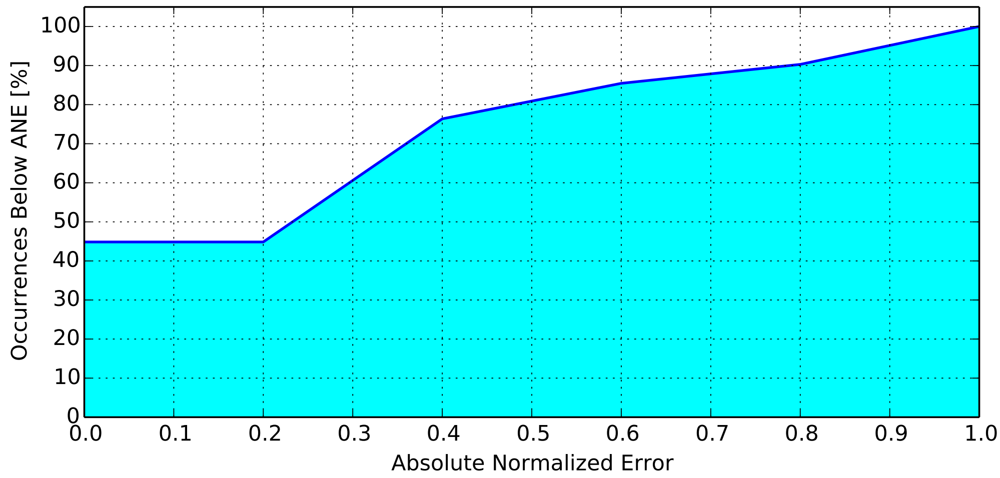
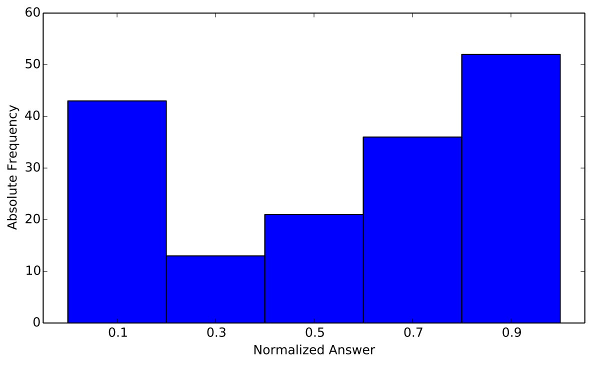
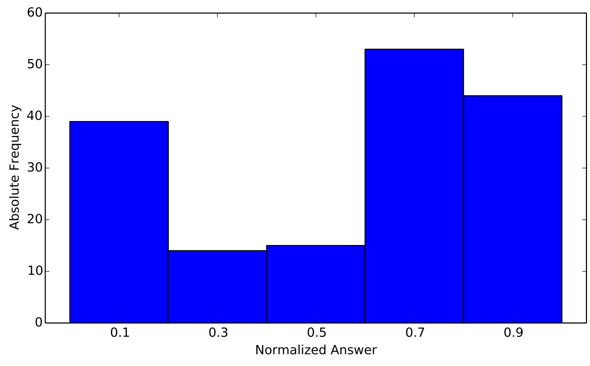


Figure 4.6: The cumulative error for all answers collected during the pre-tests. At each point in X, Y% of the answers resulted in X or less error, *e.g.* about 40% of the answers incurred in 85% or less error.



(a) Distribution of answers collected by the human evaluator.



(b) Distribution of answers collected by the robot.

Figure 4.7: Histograms illustrating the distribution of normalized answers collected during the pre-tests.

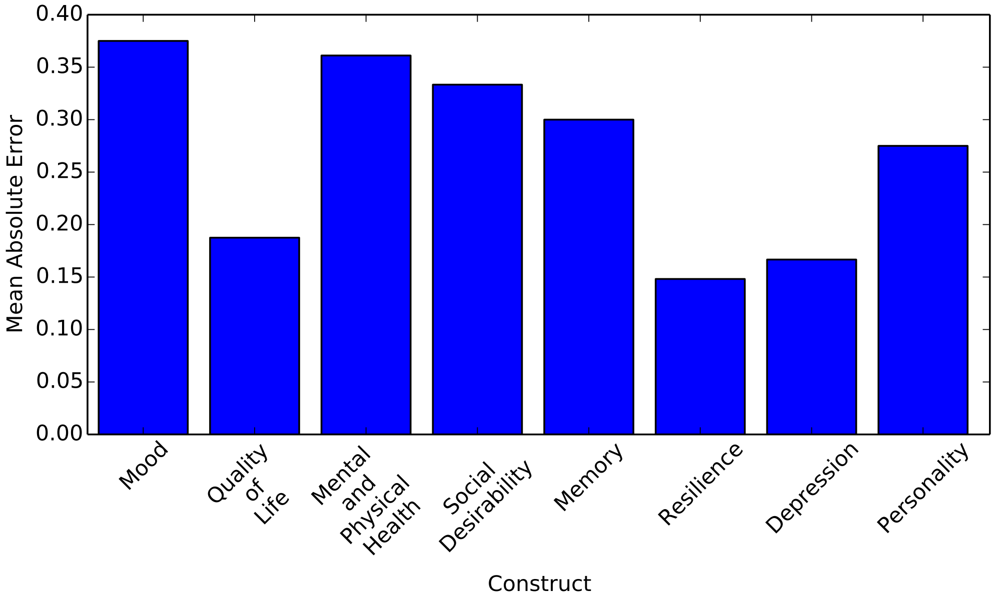
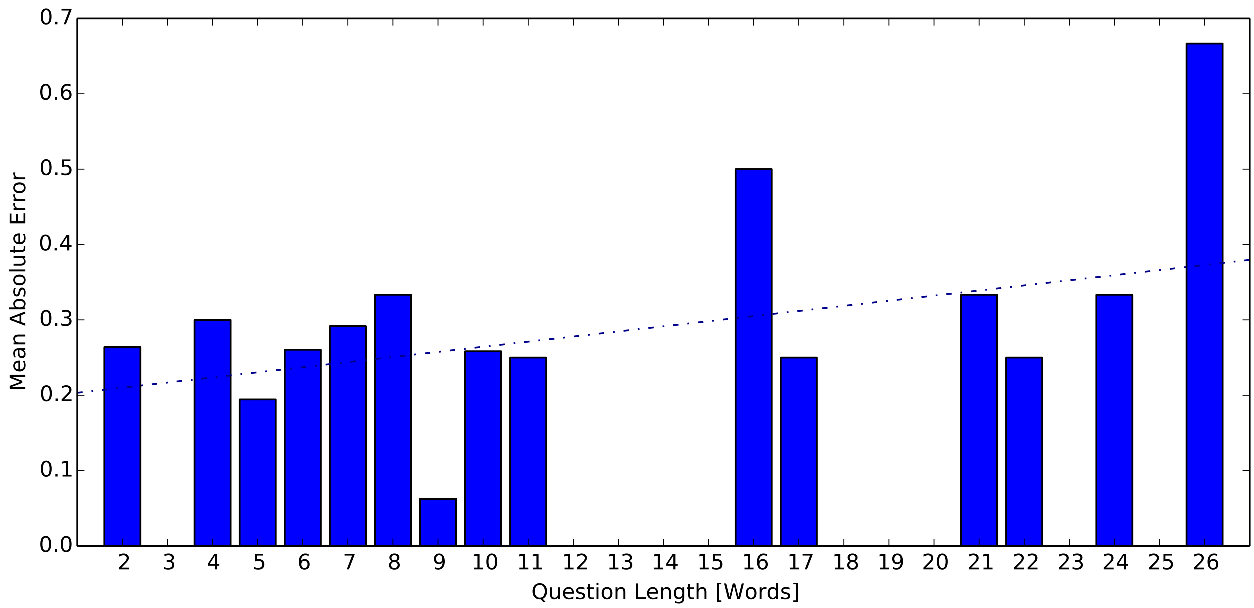


Figure 4.8: The distribution of error as a function of the construct being evaluated.

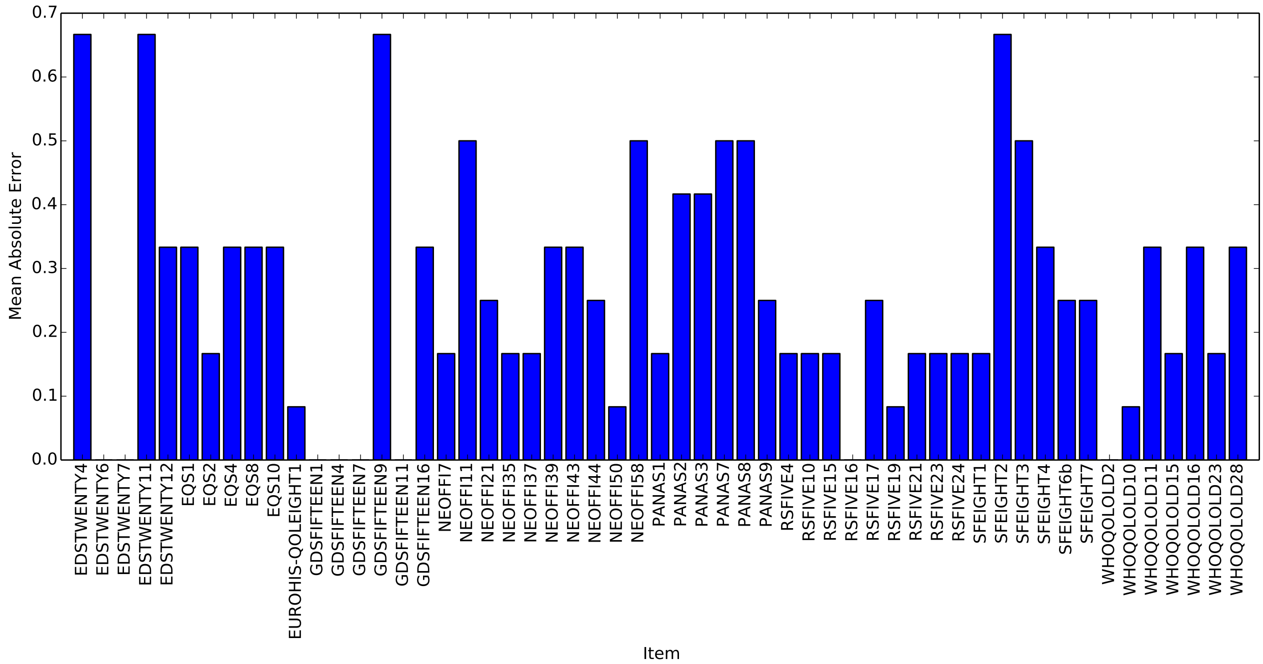
for both possible answers, which may be in part the source of this disparity: the robot may have mismatched these answers on occasion, leading to the observed disparity. Furthermore, the robot mispronounced part of the higher-scoring answer ("strongly agree") on a significant portion of the items, which may have also skewed the results and led to this disparity. However, the remainder of the distributions are perfectly comparable: the users avoided non-absolute negative answers, and favoured absolute negative answers and provided positive answers that were more subtly distributed.

Fig. 4.8 illustrates the error obtained as a function of the construct under evaluation. We can observe that some constructs, such as Quality of Life, Resilience and Depression, display a lower error than other constructs. This can be attributed to the fact that these constructs tend to be stable through time, and thus be more resistant to the error introduced by the time elapsed between the interaction with the robot and the with the human evaluator. However, the low sample size does not allow for statistically significant conclusions to be drawn.

Fig. 4.9 illustrates two measurements performed on individual items: the mean error as a function of each individual item, and as a function of the length of the final question used by the robot to present the item to the user. We can observe that, as the length of the question increases, the disparity between the robot and human evaluation tends to increase, as illustrated by the superimposed linear regression of Fig. 4.9a. This can be explained by a combination of two factors. Firstly, the robot's pronunciation, as mentioned above, was not perfect. Thus, as the robot spoke long sentences, the successive mispronunciation of certain words could lead the user to misunderstand the robot and, thus, skew their answer in spite of the repetition mechanism that was available to the user. Secondly, the robot was not able to decode the question that it was presenting. Thus, when faced with a user that could not fully understand a question, the robot's best solution was to repeat which, given its sub-optimal pronunciation,



(a) The distribution of error as a function of the length of the question posed by the robot, measured as the number of words. The superimposed line corresponds to a linear regression.



(b) The distribution of error as a function of the individual item being evaluated.

Figure 4.9: Illustration of the item-specific measurements. The mean absolute error is plotted as a function of the item it corresponds to.

was not always successful. These compounding effects could lead to the disparity observed.

We can observe, in Fig. 4.9b, that items administered by the robot generally tended to accrue uniform error. However, four items in particular accumulated the most error:

- `EDSTWENTY4`, a binary item, asks the user if they normally act according to their own words.
- `EDSTWENTY11`, a binary item, asks the user if they are prone to procrastinating necessary tasks to the day after.
- `GDSFIFTEEN9`, a binary item, asks the user if they have felt happy for most of the time in the last few weeks.
- `SFEIGHT2`, a threefold item, asks the user if their health limits them in their daily tasks. Due to its phrasing in Portuguese, this item is the longest in the scale, measuring 26 words in length.

Three of these four items share a common characteristic: they are binary items, meaning that they only accept "yes or no" answers. Thus, whenever the robot and human evaluator disagree, the absolute error according to the formulation of Eq. 4.2 will be maximum, which may result in an exacerbated error value for these items. Furthermore, half of these correspond to the social desirability construct, which is one of the constructs with the highest global error. `GDSFIFTEEN9`, despite belonging to one of the constructs with the lowest global error, depression, accumulated a disproportionate amount of disparity. This may be due to the differences in phrasing between the robot and paper scales, and to the ability of the human evaluator to better decode the question, indicating its general nature. Lastly, the error accumulated by `SFEIGHT2` can be explained by the length of the item itself, as detailed above: longer items tend to accrue higher error, and this item is the longest of the scale.

Table 4.2 presents the Pearson and Spearman correlation coefficients and p-values for a number of tests performed on the data. We can observe that, globally, the answers given to the robot and the human evaluator are positively correlated in a statistically-significant manner, resulting in a p-value of 0.0049, well under the 5% threshold that is usually considered for significance. Regarding each individual construct, it was only possible obtain to significant results for the Depression and Quality of Life constructs, with Pearson p-values of 0.0049 and 0.0065, respectively. However, the remaining constructs also consistently achieved positive correlations, despite not achieving statistical significance. Regarding each individual test subject, it was possible to achieve statistical significance for each individual full test. We could not definitely correlate the robot's error with the length of the question, as is further illustrated in Fig. 4.9a, despite obtaining a positive correlation. We can observe that, except in the case of Subject 9, the Pearson and Spearman coefficients and p-values tend to agree, indicating that there is a linear relationship between the normalized answers.

In general, the results presented in this section strongly indicate that the null hypothesis of Section 4.1 can be disproved with statistical significance. However, being based on only three subjects and 165 data points, these results are insufficient for a final conclusion. Furthermore, a number of important improvements and questions for future work were also raised. Thus, the results of the pre-tests were found promising enough to warrant a wider dataset collection and analysis and, as such, the wider data collection was performed.

Table 4.2: Pearson and Spearman correlation coefficients and p-values for various comparisons between the answers given to the robot and to the human evaluator during the pre-tests. A star (*) indicates that the result is statistically significant (p-value < 0.05).

Test	Pearson Coefficient	p-value	Spearman Coefficient	p-value
Global	0.4537	9.3436e-10 *	0.4083	5.2267e-8 *
Results Per Construct				
Depression	0.6325	4.8549e-3 *	0.6325	4.8549e-3 *
Memory	0.2448	3.7926e-1	0.3638	1.8248e-1
Mental and Physical Health	0.07560	7.6558e-1	0.04696	8.5321e-1
Mood	0.2258	3.6767e-1	0.2236	3.7243e-1
Personality	0.2860	1.2545e-1	0.2067	2.7305e-1
Quality of Life	0.5398	6.4726e-3 *	0.4559	2.5174e-2 *
Resilience	0.2692	1.7455e-1	0.1899	3.4265e-1
Social Desirability	0.3273	2.3369e-1	0.3273	2.3369e-1
Results Per Subject				
Subject 7	0.6084	8.3357e-7 *	0.5483	1.4696e-5 *
Subject 8	0.4533	5.1007e-4 *	0.4109	1.8326e-3 *
Subject 9	0.2816	3.7305e-2 *	0.2565	5.8671e-2
Error w/ Question Length	0.3474	1.8740e-1	0.1953	4.6861e-1

4.5.2 Main Tests

Fig. 4.10 presents the global error distribution obtained during the main trial phase for all 16 users. We can observe that the error distribution is similar to that obtained during the pre-trial phase (Fig. 4.5), in that the error seems to follow a very narrow Gaussian distribution. In this case, the distribution is even narrower than during the pre-tests, with the majority of examples accumulating low error.

Fig. 4.11 presents the cumulative error obtained during the main tests phase. This figure is also similar to its pre-trial counterpart, demonstrating, for instance, that 80% of the items incurred in 4% or less error. In the main tests, the curve displays a higher initial slope when compared to that of Fig 4.6, indicating that the global error is lower in this portion of the data.

Fig. 4.12 presents the distribution of answers provided to the human and robot evaluators, respectively. Unlike the results observed in the pre-trial phase (Fig. 4.7), the figures present very similar answer distributions. This indicates that the development that took place in-between pre-trial sessions was successful, and also that the pre-trial phase lacked data volume to be conclusive, as expected; the problems that could result in the disparate results found before were either solved in between the pre-trial and trial phases, or were simply a manifestation of a lack of data.

Fig. 4.13 presents the mean absolute error as a function of the construct under evaluation.

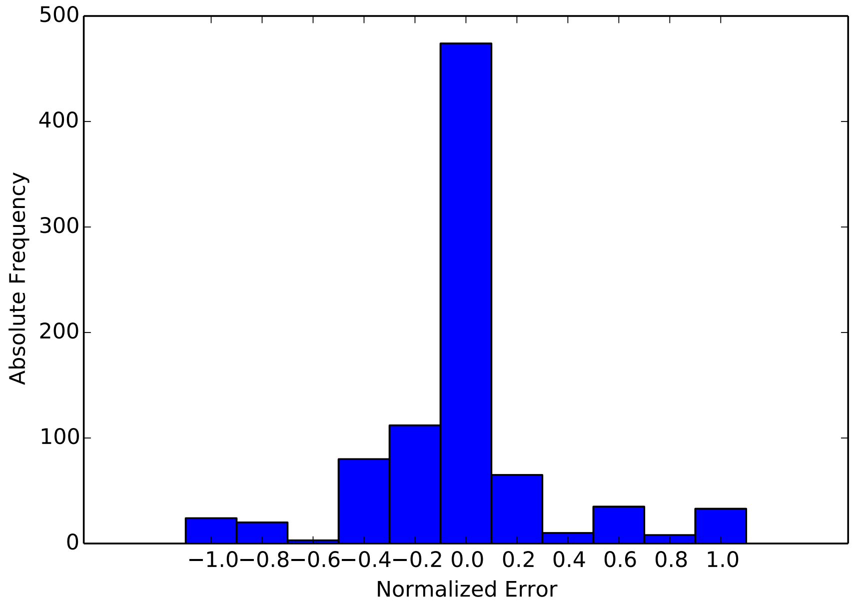


Figure 4.10: A histogram illustrating the distribution of the normalized error for all of the answers collected during the main tests.

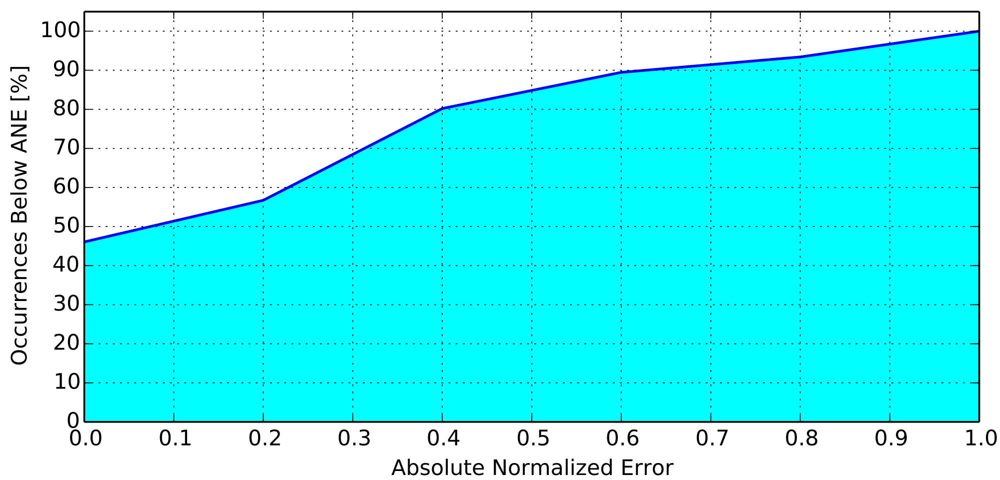
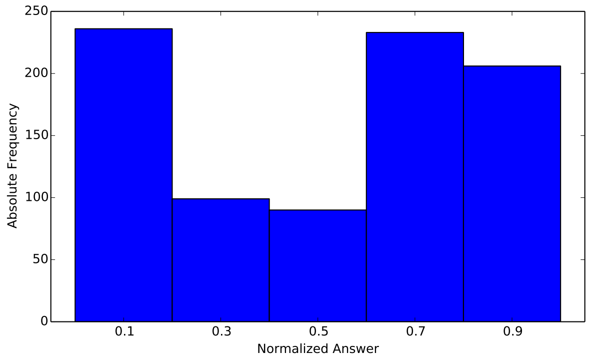
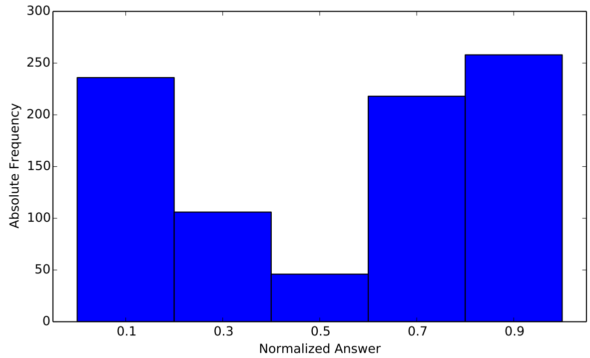


Figure 4.11: The cumulative error for all answers collected during the main tests. At each point in X, Y% of the answers resulted in X or less error, *e.g.* about 40% of the answers incurred in 80% or less error.



(a) Distribution of answers collected by the human evaluator.



(b) Distribution of answers collected by the robot.

Figure 4.12: Histograms illustrating the distribution of normalized answers collected during the main tests.

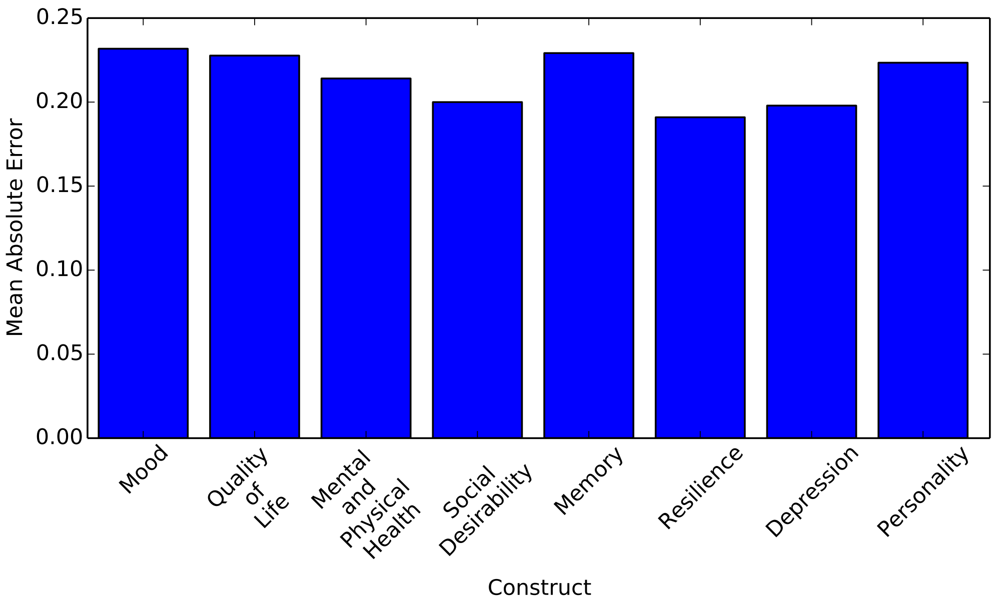


Figure 4.13: The distribution of error as a function of the construct being evaluated.

Unlike the results obtained in the pre-tests, there is now no significant discrepancy in the accumulated error obtained for each construct, resulting in a nearly-flat distribution of average error over constructs. This indicates that the disparate results found in the pre-trial phase were, again, either a result of lack of data or of interaction issues that were since solved. Thus, we can conclude that the robotic evaluator is able to evaluate all constructs with similar accuracy.

Fig. 4.14 illustrates the item-specific measurements taken. Similarly to the results obtained in the pre-tests phase, there seems to be a positive relationship between the length of the questions asked by the robot and the error incurred in the corresponding items. Unlike the pre-trial results, the error seems evenly distributed among the items, with only a few exceptions, none as notable as in the pre-trial phase.

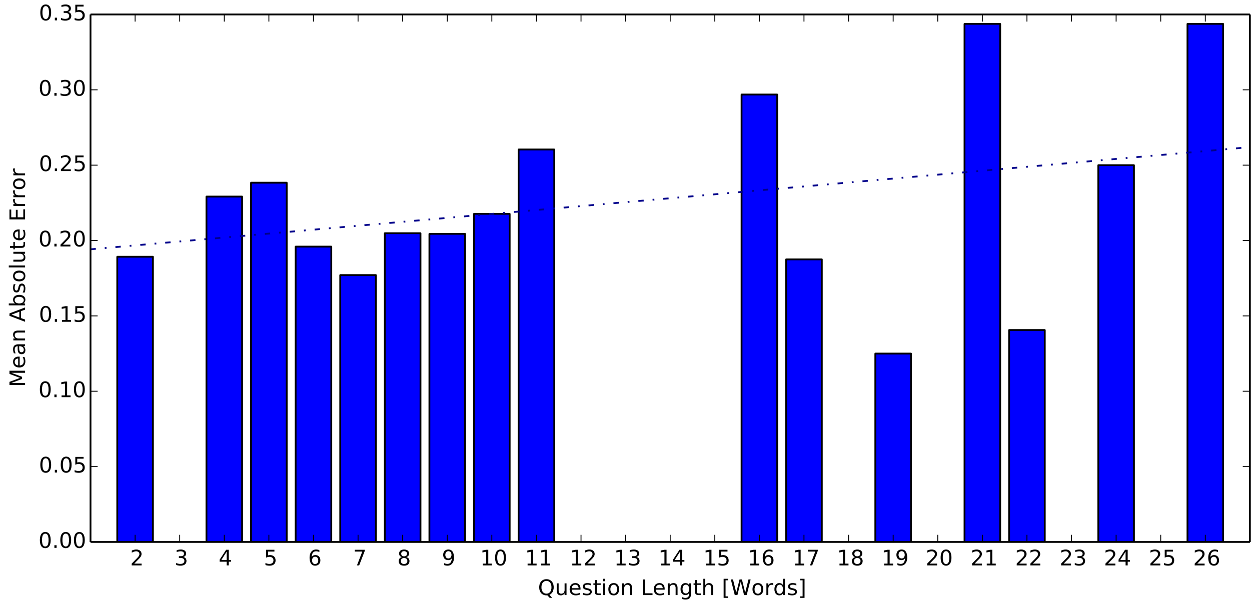
Table 4.3 presents the Spearman and Pearson correlations between the normalized scores obtained by the robot and the human evaluator. Unlike the pre-trial results of Table 4.2, the system managed to achieve statistical significance in all of the tests, including globally, per construct and per subject. Thus, based on the 864 data points obtained during this phase of the experimentation, the null hypothesis of Section 4.1 can be safely rejected, with the robot and human evaluators reaching a best-case correlation coefficient of 0.58 with a p-value of 2.13e-81. This allows us to conclude that the robot can be used to obtain approximate information on these psychological constructs when operating with elderly users, as originally intended.

These results demonstrate a much stronger relationship between the results obtained from the robot and human evaluators than was observed in the pre-trial phase, achieving both higher correlation coefficients and lower p-values. There do not seem to be any constructs where the system performs particularly badly, with the best-case correlation coefficient reaching 0.61, at a p-value of 3.45e-11, indicating a positive linear relationship which may not be very strong, but is extremely significant. A higher degree of variation can be observed in the per-subject evaluation, in which the system achieved a best-case 0.84 correlation coefficient, at a p-value of 3.64e-15, still achieving statistical significance in the worst-case of Subject 11. It was again not possible to significantly correlate the error found with the length of the question posed by the robot, despite the apparently positive relationship illustrated in Fig. 4.14a.

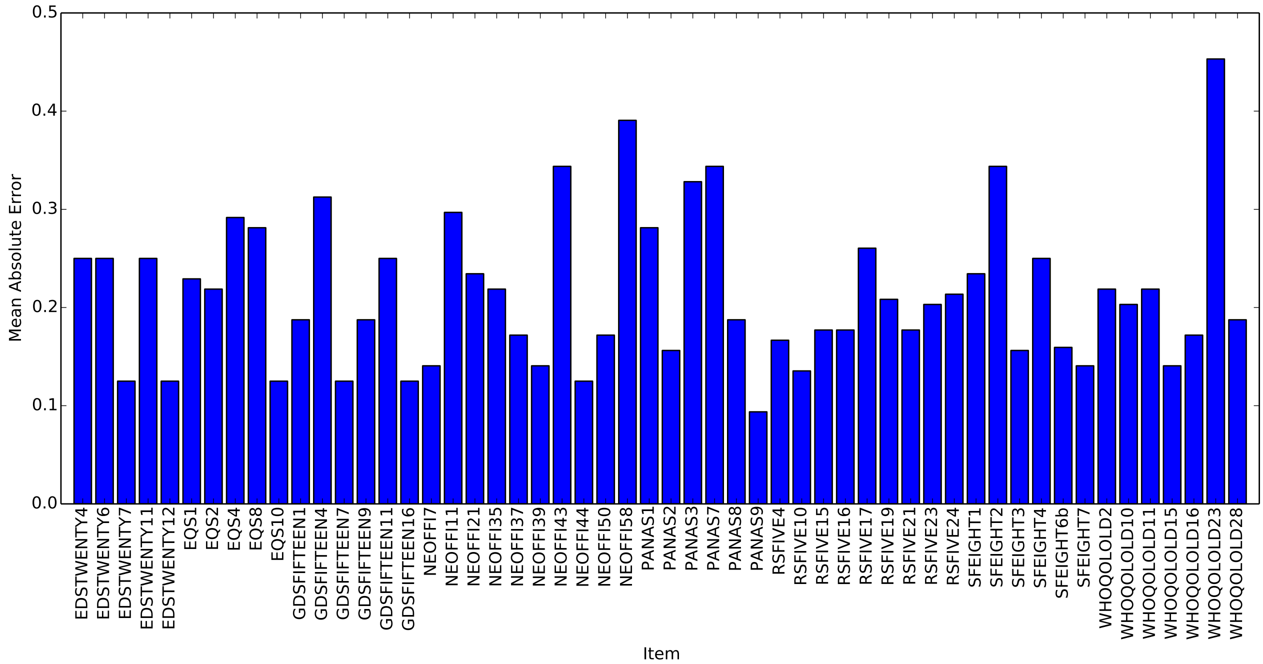
Despite correlating significantly with the answers provided to the human evaluator, the robotic system introduces error in the measurement of the constructs, as discussed above, where we can observe extremely significant relationships which, due to the errors introduced by the system, are not accompanied by equally extreme positive relationships between the evaluations. Despite not having performed formal research into the sources of these errors, we can postulate a number of plausible causes that stem from the functioning of the robot itself. Possible error sources include:

- **Verbal interaction errors** the robot's speech synthesis system is not perfect, and has been observed on numerous occasions to induce the user in error. As such, it is plausible that, if the user could not understand the robot as well as they believe, their response would be biased;

- **Matching algorithm** the matching algorithm of Section 4.3 aims for a trade-off between accuracy and time efficiency. For instance, instead of repeatedly asking the user the same question in order to find an exact answer, it will gradually increase its sensitivity, enabling it to match to answers which are farther from any of the acceptable ones. In this context,



(a) The distribution of error as a function of the length of the question posed by the robot, measured as the number of words. The superimposed line corresponds to a linear regression.



(b) The distribution of error as a function of the individual item being evaluated.

Figure 4.14: Illustration of the item-specific measurements.

Table 4.3: Pearson and Spearman correlation coefficients and p-values for various comparisons between the answers given to the robot and to the human evaluator during the main tests. A star (*) indicates that the result is statistically significant.

Test	Pearson Coefficient	p-value	Spearman Coefficient	p-value
Global	0.5878	2.1270e-81 *	0.5682	5.2014e-75 *
Results Per Construct				
Depression	0.5808	5.5079e-10 *	0.5808	5.5079e-10 *
Memory	0.4016	2.2248e-4 *	0.4086	1.6806e-4 *
Mental and Physical Health	0.6121	3.4517e-11 *	0.5942	1.7547e-10 *
Mood	0.5853	3.7943e-10 *	0.5820	4.9832e-10 *
Personality	0.5051	9.6728e-12 *	0.5057	9.0357e-12 *
Quality of Life	0.5590	1.5035e-10 *	0.5481	3.9813e-10 *
Resilience	0.3314	4.9521e-5 *	0.2529	2.2296e-3 *
Social Desirability	0.5490	1.3419e-7 *	0.5490	1.3419e-7 *
Results Per Subject				
Subject 10	0.4956	1.3863e-4 *	0.4954	1.3986e-4 *
Subject 11	0.2910	3.2797e-2 *	0.3254	1.6355e-2 *
Subject 12	0.4910	1.6354e-4 *	0.4748	2.8631e-4 *
Subject 13	0.5950	2.0871e-6 *	0.5891	2.7790e-6 *
Subject 14	0.4202	1.5598e-3 *	0.4119	1.9682e-3 *
Subject 15	0.8777	3.0535e-18 *	0.8481	5.8395e-16 *
Subject 16	0.4548	5.5063e-4 *	0.4414	8.3435e-4 *
Subject 17	0.5796	4.3762e-6 *	0.5733	5.8769e-6 *
Subject 18	0.4880	1.8148e-4 *	0.4404	8.5914e-4 *
Subject 19	0.4985	1.2495e-4 *	0.4870	1.8819e-4 *
Subject 20	0.6719	2.6615e-8 *	0.6009	1.5540e-6 *
Subject 21	0.7761	5.4212e-12 *	0.7551	4.1843e-11 *
Subject 22	0.8360	3.6495e-15 *	0.7966	5.8632e-13 *
Subject 24	0.5649	8.5789e-6 *	0.5566	1.2373e-5 *
Subject 25	0.4244	1.3822e-3 *	0.3597	7.5560e-3 *
Subject 26	0.6409	1.7879e-7 *	0.5784	4.6433e-6 *
Error w/ Question Length	0.3215	2.2471e-1	0.2369	3.7694e-1

it is likely that this system introduces some degree of error whenever it cannot match the user's answer with enough certainty;

- **Item phrasing** the phrasing of the items had to be adapted in order for the robot to be able to synthesize them in the first person, as if asking the questions itself. Furthermore, as seen in Section 4.3, all items are combined with a temporal reference and connective text, further altering their final phrasing. Since the phrasing of the items themselves has been shown to be of crucial importance in these tests, it is possible that this re-phrasing also introduced some error in the measurements;

- **Temporal disparity** the time period between the automated and manual tests was not controlled in this experiment, and varied between a few hours to two weeks. Despite falling within the margins prescribed by the scales themselves, this disparity could have introduced some error in the measurements, seeing as some of the constructs are liable to change with great frequency;

- **Deliberate misinformation** a great number of the items included in both the automated and manual evaluation are of a very personal and intimate nature, towards which some of the users displayed discomfort. Despite being assured of the anonymity and confidentiality of their participation in the experiment, some users may have deliberately misinformed the evaluators during the experiment; if disparity exists in the way misinformation is passed to each evaluator, the user introduces error in the measurement.

4.6 Summary

This chapter has presented the Psychbot study, aiming to demonstrate that the answers given to a robot administering a modified version of the classical scales from the field of Psychology can be used as indication of the results obtained by human evaluators. Experiments were split into three phases, including pre-tests and main tests, wherein 1029 total data points were gathered, interacting with 25 subjects. A total of 9 psychological constructs, including Personality and Depression, were evaluated, aiming to obtain a general view of the population. Results show that the answers given to the robot correlate positively and statistically significantly with those given to the human evaluator, thus demonstrating that the robot can be used as a screening tool to obtain preliminary evaluations from users. This system can now be built upon to further automate the screening process, enabling true perception of psychological characteristics, as described in Chapter 8.

Having demonstrated how a user modelling system can establish the relationship between the signals it observes and the user's characteristics, it is now time to tackle how this information can be integrated into the model itself. This process may involve a learning step, which traditionally requires large amounts of data. The dataset reduction technique of the next chapter strives to reduce the amount of data needed to achieve this effect, filtering the data examples one-by-one to achieve a minimal dataset that allows a classifier to achieve comparable results.

Chapter 5

Surprisal-Based Dataset Reduction

> "The programmer at wit's end for lack of space can often do best by disentangling himself from his code, rearing back, and contemplating his data. Representation is the essence of programming."
>
> — Fred Brooks, The Mythical Man-Month

Techniques that employ learning mechanisms, such as the BUM technique of Chapter 3, need to be fit with *training data*, with the general trend indicating that larger datasets correlate with better system performance. This led to the development of large datasets, such as ImageNet [122] and WordNet [99], which are used to train state-of-the-art techniques. These datasets contain very large numbers of labelled examples, which are used to train supervised and semi-supervised learning techniques. These datasets can contain high levels of redundancy, far exceeding the minimum amount and variety of samples needed to achieve state-of-the-art performance. Thus, as Brooks put, we have a problem of representation: if we can better represent the dataset, then perhaps we can perform the same function with only a subset of it.

To tackle this issue, dataset reduction techniques have been proposed, which filter these large datasets, striving to achieve a more compact representation of the information therein. We propose such a dataset filtering algorithm, allowing training data to be selected as relevant or not relevant to be learned, constituting the dataset deduction module illustrated in Fig. 5.1. We employ the surprisal [134] measurement as an estimator of the informative value of each training example with respect to an existing model.

Surprisal is defined for a random variable X, from which a random example x is drawn, as

$$I(x) = -\log(P(X = x)). \tag{5.1}$$

This measurement provides insight into how well a trained model can explain a new example, with higher surprisal encoding higher potential information gain, and vice-versa.

We hypothesize that this measurement can be used to filter training examples, expediting the training process while maintaining state-of-the-art levels of performance. This chapter

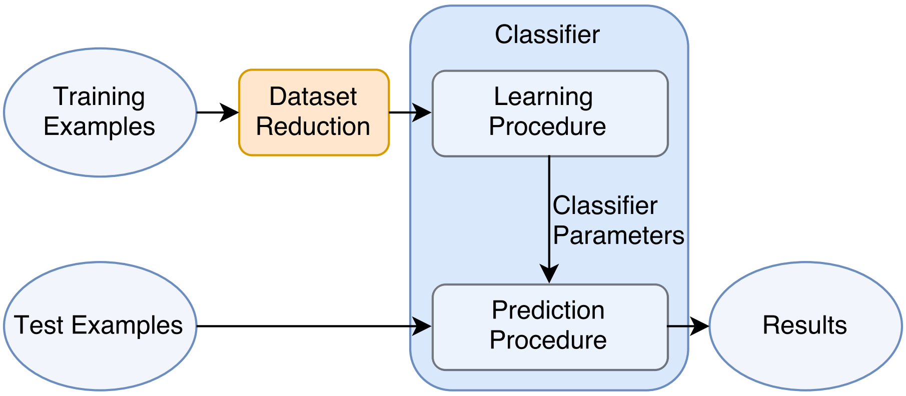


Figure 5.1: A collapsed view of Fig. 2.2, illustrating how dataset reduction techniques fit into the normal operation of a classification system that uses supervised learning.

focuses on the exploration of this hypothesis, and on the demonstration of its validity using various artificial and benchmarking datasets.

This chapter is organized as follows:

- Section 5.1 presents the main goals and contributions of the chapter;
- Section 5.2 presents some related works;
- Section 5.3 details our technique;
- Section 5.4 presents our experimental set-up, metrics and dataset generation procedures;
- Section 5.5 presents our results and discussion on these results;
- Lastly, Section 5.6 presents a summary of the work described in the chapter.

5.1 Chapter Goals and Contributions

The main goal of this chapter is to present and experimentally demonstrate a technique for dataset reduction based on the surprisal measurement. It introduces the following key innovations:

- A surprisal-based methodology for reducing a dataset while maintaining acceptable classification performance;
- A proof-of-concept of the functionality on synthetic data and with benchmark datasets, achieving significant reductions;
- A comparison of the results with those achieved by state-of-the-art techniques.

We aim to show that:

1. Our technique can be used to filter out unneeded samples from synthetic and real datasets while maintaining state-of-the-art performance;

5.2 Related Work and Adjacent Fields

Active Learning (AL) techniques operate on large amounts of unlabelled training data, and seek to select the most informative examples. These are labelled by an oracle, providing ground truth information, and used for training a machine learning technique that can operate on the remaining examples. Techniques include uncertainty sampling [166] and query by disagreement or committee [32]. For a thorough survey, the authors refer the reader to [23]. Active learning techniques are similar to dataset reduction in that both select the most informative examples from the dataset. However, the techniques tackle different problems: dataset reduction focuses on discarding redundant examples, while AL focuses on minimizing the labelling cost. As such, AL operates on unlabelled examples, while dataset reduction techniques operate normally on labelled examples.

Other works, such as [118] and [155], have also tackled the issue of dataset reduction. These works perform dataset reduction for improving the performance of a nearest-neighbour classifier, and by using an ensemble of SVMs and information values to discard data. For a survey on data reduction techniques, the authors would like to refer the reader to [147].

Concepts such as principal component analysis [59], eigenjoints [161] and eigenfaces [145] can be employed to enhance the representation of the data in a transformed feature space, thus improving classifier performance. For an extensive survey on dimensionality reduction techniques, the authors would like to refer the reader to [36].

Feature Selection is also a widely used technique to reduce the amount of data used to train predictors. Instead of reducing the dimensionality of the data by transforming it in some way, these techniques focus on determining which features are the most discriminant among the data, eliminating those that are not. For an extensive survey on the matter, the authors would like to refer the reader to [30].

Several works have successfully used the surprisal measurement, or Surprisal Analysis, in several fields. The authors of [38] employ it in the scheduling of Pan-Tilt-Zoom cameras with the goal of pointing the devices at the locations where there is the most information to be gained, finding that their entropy driven scheduler outperforms other scheduling methodologies. The measurement, in a slightly different formulation, has been employed in the analysis of natural language, namely in the prediction of reading complexity [18], in the comparison of language models [20], and in the definition of grammars [58]. Entropy, the expected value of surprisal, is used by the authors of [125] as input for a methodology for adapting a sliding window to the data, enhancing the performance of their classifier, and by the authors of [83] in the design of experiments with co-evolution systems.

5.3 Dataset Reduction Using Surprisal

We assume that the learning mechanism for a classifier is fed with a sequence $\mathbf{S}$ of training examples and we aim to classify each s_k as needed or not. These training examples are a tuple constituted by a materialization of the input features, x, and a ground-truth label, y, corresponding to one of the classes that the system is prepared to classify into.

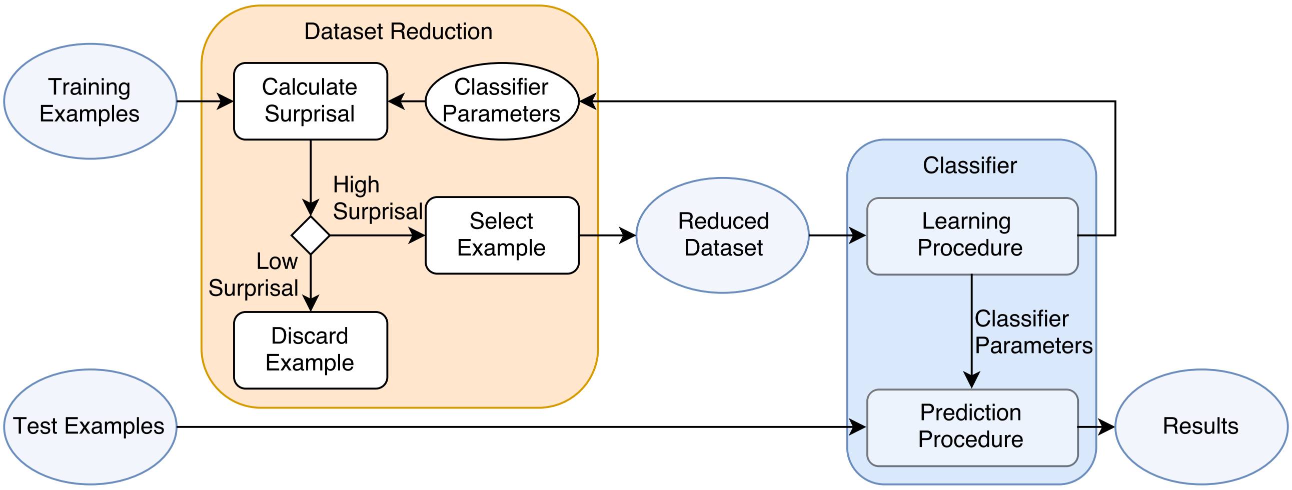


Figure 5.2: An overview of the dataset reduction system. Examples are fed to the reduction module, which decides whether they should be used for learning or not. Iteratively, the classifier is re-trained, providing different insight into the remaining examples.

Formally, the sequence of instances, $\mathbf{S}$ is defined as

$$\mathbf{S} = [s_1, s_2, s_3, ..., s_n] \tag{5.2}$$

where

$$s_k = \{x_k, y_k\} \tag{5.3}$$

and we wish to make a decision as to whether $s_k \in \mathbf{N}$, $\mathbf{N}$ being the set of needed examples, for all s_k.

Bayesian systems are widely used in various applications involving learning, and the general Bayesian Programming paradigm can be used as a generalized formulation of other techniques, such as Bayesian Networks and Hidden Markov Models. Bayesian classifiers were, thus, employed as the main test subjects in our experiments. A Bayesian classifier builds a probability distribution of the form

$$P(Y|X) \propto P(Y)P(X|Y) \tag{5.4}$$

where X is the set of features used for classification, and Y the resulting class, corresponding to the application of Bayes' Rule to the problem of classification. We employ the surprisal measurement, presented in Eq. 5.1, applied to the basic Bayesian classifier, resulting in the following formulation of the metric:

$$I(s) = -log(P(X = x|Y = y)), \tag{5.5}$$

measured in units that depend on the logarithm employed.

For each new instance s_k fed into the system, the value of $I(s_k)$ is calculated. If it is higher than a certain threshold τ, that example is added to $\mathbf{N}$, and used for training. So, for each example s_k in S:

$$\mathbf{N}_k = \begin{cases} \mathbf{N}_{k-1} \cup \{s_k\} & I(s_k) > \tau \\ \mathbf{N}_{k-1} & I(s_k) \leq \tau \end{cases} \tag{5.6}$$

$\mathbf{N}_n$, the reduced dataset, is the last set obtained by through Eq. 5.6 after processing all available examples. This process is illustrated in Fig. 5.2.

5.4 Experiments

5.4.1 Goal and Metrics

We aim to demonstrate that our technique can be used to significantly reduce the amount of training data fed to a classifier while maintaining or causing negligible impact on its performance. In order to test the technique, we employ Bayesian classifiers with Gaussian likelihoods. The Gaussian likelihood of these systems are trained using examples of the same form as those in Eq. 5.3. As examples are received, the system is *trained* by fitting the $P(X|Y)$ distribution by performing

$$P(X|Y) = \mathcal{N}(\mu, \sigma), \tag{5.7}$$

where μ is the average of the x_i values of examples where $y_i = j$, and σ is the standard deviation of the x_i values of those same examples. Likelihoods are initialized as uniform distributions. Surprisal is always determined as described in Eq. 5.5.

We employ three performance metrics:

- **Classification Accuracy** (A), the main measure of the classifier's performance, defined as

$$A = \frac{n_{correct}}{n_{total}} \tag{5.8}$$

  where $n_{correct}$ represents the number of correct classifications made by the system, and n_{total} represents the total number of examples used on the respective test.

- **Training Time** (T), measured in seconds, defined as:

$$T = \frac{1}{1000}\sum_{i=1}^{1000} T_i, \tag{5.9}$$

  where T_i is the result of each individual run of the training procedure. T is averaged over 1000 runs of the training procedure to rule out any random effects during execution.

- **Dataset Size** (D), the fraction of examples that were classified as needed, the fraction of examples remaining in the dataset:

$$D = \frac{N_r}{N_t}, \tag{5.10}$$

  where N_r and N_t represent the number of remaining and total training examples, respectively.

5.4.2 Experiments using Artificial Data

We have generated several synthetic datasets to train our classifiers, producing a proof-of-concept of our solution. Examples were generated randomly by drawing x_i from a uniform random distribution and defining y_i as:

$$y_i = \begin{cases} 0 & x_i \in [0, 0.5] \\ 1 & x_i \in]0.5, 1] \end{cases} \tag{5.11}$$

To generate a noisy dataset, the same formulation is used with 33% of the training data being purposefully mislabelled, thus compensating for the simplicity of the initial dataset. For demonstrating the applicability of the system to multiclass problems, multiclass datasets were also generated using the same procedure, but in this case $X \in [0, 1]$ and $Y \in \{1, 2, ..., n\}$, where n is the desired number of output classes.

We have tested dataset sizes ranging from 2000 to 10000 examples, observing little difference in our results, and have thus elected to thoroughly discuss the tests based on 2000 examples.

To demonstrate applicability to multi-output cases, we make use of the model and dataset described in [124]. This model is meant to study of the routine of a user, taking a vector of input variables, and classifying each instance into multiple classes. It generalizes the multi-output classification problem to a multi-class problem with a single output variable, $Y \in \{0, 1, ..., q-1\}^r$, where r is the number of system outputs and q is the number of states of each output. For the purposes of these tests, we have set $q = 2$ and $r = 4$, and have used a base dataset of 5000 examples, resulting in 2^4 possible states.

5.4.3 Experiments using Real Data

Three benchmarking datasets were employed in these tests: the UC-3D dataset [1], the IJCNN1 dataset [2] and the USPS dataset [3].

The UC-3D dataset was gathered by observing a number of persons performing different actions, and contains RGB, Depth and motion tracking data, of which we use the latter. MVN data was captured using a motion capture suit at 120Hz. We have manually labelled each sequence with respect to the action descriptor elements they reveal, and have combined that data with the features extracted from the trajectories, as described in [125], to obtain our training examples. Being manually labelled, it is to be expected that the data be noisy.

Overall, we have used:

1. 130 sequences from UC-3D, divided into 5 actions and 9 subjects, with 23 discrete frequency features as input v, $v = (v_1, v_2, ..., v_{23}), v_i \in \mathbb{N}$ and an output vector of five binary variables $c_i \in (c_1, c_2, c_3, c_4, c_5)$, where each c_i corresponds to an action descriptor.

2. The IJCNN dataset, containing 22 input features $x = (x_1, ..., x_{22})$, $x_i \in [-1, 1]$ discretised into 1000 values, and a single binary output variable.

[1] The UC-3D dataset can be found at `http://mrl.isr.uc.pt/experimentaldata/public/uc-3d/`.

[2] The IJCNN1 dataset can be found at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`

[3] The USPS dataset can be found at `https://www-i6.informatik.rwth-aachen.de/~keysers/usps.html`

3. The USPS dataset, containing 128 input features in the same range and discretisation and the previous dataset, with a discrete output variable $y \in [1, 10]$.

A naïve Bayes classifier was developed for each of these real datasets, allowing us to test the efficacy of our technique on each dataset. The system for the UC-3D dataset, despite the independence of the action descriptors, was implemented as a single multi-output classifier, and not as five independent classifiers.

5.5 Results and Discussion

5.5.1 Results with Artificial Data

In Fig 5.3a, we can observe that the accuracy of the classifier tends to stabilize at 100%, and that the surprisal measurement stabilizes in relatively low values at roughly the same time.

Fig. 5.3b illustrates the application of our technique. We can observe that, as we increase τ, *i.e.* the surprisal threshold above which a instance is selected as needed, we are able to maintain near-constant accuracy with a significant reduction of both the number of training examples used and the training time. As the threshold is increased, we can identify three different regions:

1. $\tau \in [0, 10]$: Increases in τ have a strong influence on dataset size and training time. Significant portions of the dataset are discarded at each step;

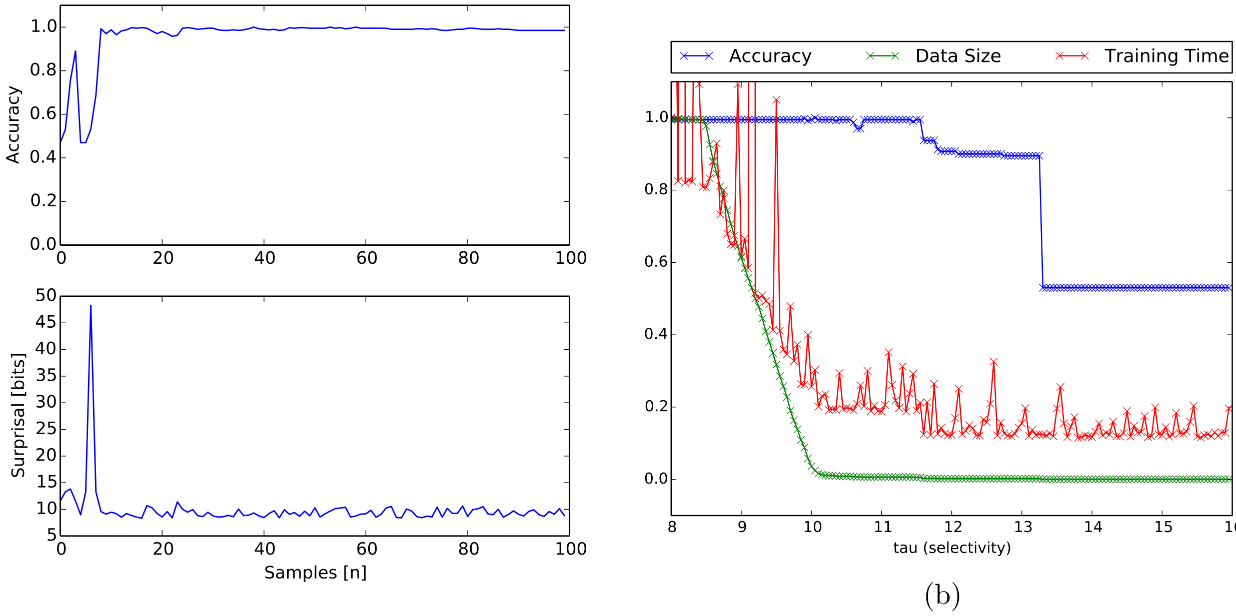
2. $\tau \in]10, 13]$: Maximum reduction with minimal accuracy impact is reached. Portions of the dataset are discarded at each step have a larger effect on accuracy. This interval contains the point of equilibrium of the learning curve;

3. $\tau > 13$: The system is saturated. No more examples can be discarded, since all examples cause high surprisal. Accuracy decays and training time remain constant.

Thus, there is an equilibrium point to be exploited between the second and third region, where dataset reduction is maximal and performance loss minimal.

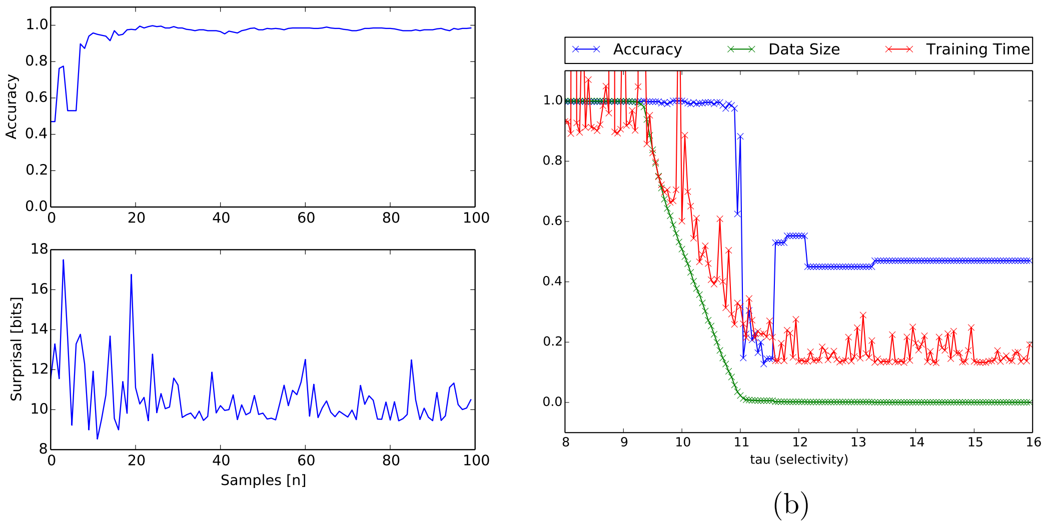
Fig. 5.4 presents the results obtained from the application of our technique on noisy data. The learning curve is no longer monotonic, and the surprisal levels no longer stabilize in low values. The same general regions as in Fig 5.3b can be observed, with significant reduction being observed only for slightly higher levels of surprisal, and stabilization happening at roughly the same values. Accuracy does not decrease monotonically, and is instead affected by the noisiness of the data.

Fig. 5.5 shows the results obtained using the multiclass technique, for different numbers of output classes. The minimum number of examples and the maximum reduction were registered for a maximum performance decay of 5% of the performance obtained when training with the full dataset. We can observe that significant reductions can be achieved even for large numbers of output classes. We note that the minimum number of examples needed tends to follow a rough linear trend. Regarding the τ values obtained for these minimum datasets, we can observe that they remain generally constant.



(a) (b)

Figure 5.3: Results of the tests using artificial data. (a): An illustration of the classifier's learning curve. Surprisal is calculated with respect to the next instance to be added. (b): Results obtained by measuring the technique's accuracy, training time and dataset reduction for varying τ.



(a) (b)

Figure 5.4: Results of the tests using noisy artificial data. (a): An illustration of the classifier's learning curve. Surprisal is calculated with respect to the next instance to be added. (b): Results obtained by measuring the technique's accuracy, training time and dataset reduction for varying τ.

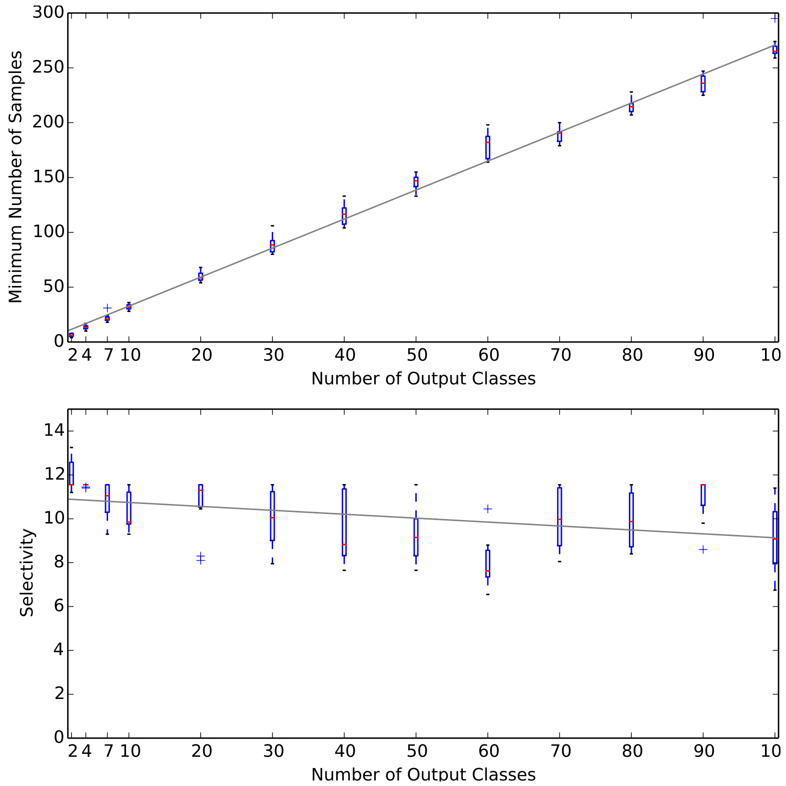


Figure 5.5: Top: an illustration of the evolution of the relationship between the number of outputs of the classifier and the minimum number of examples needed to achieve 95% of initial performance, with a linear fit of the averages in grey. Bottom: an illustration of the evolution of the value of τ obtained on each run, with a linear fit in grey. 100 tests were performed for each number of classifier outputs.

Table 5.1: A comparison of the results obtained for the various artificial datasets. Final training time is expressed as a fraction of the initial training time.

Technique	Base A	Final A	Final D	Final T	τ
Binary Data (clean)	100%	100%	$<$ 1%	20%	13.3
Binary Data (noisy)	100%	100%	10%	18%	10.6
Multiclass Data	98%	70%	8%	20%	10.2
Multilabel Data	98%	70%	20%	50%	1.8

Table 5.2: Recommended Values of τ for different problem structures. The recommended τ is obtained when a 5% decrease in performance is observed.

Inputs	Input States	Outputs	Output States	Recommended τ
1	100	1	2	6.6
1	100	1	4	6.6
1	100	1	10	6.6
2	100	1	2	6.6
3	100	1	2	6.6
4	100	1	2	6.6
1	200	1	2	7.6
1	500	1	2	8.9
1	1000	1	2	9.9
1	2000	1	2	10.9
1	3000	1	2	11.5

Table 5.1 shows a summary of the results obtained regarding the three main metrics we are taking into account. We can observe that very significant reductions can be obtained for all datasets whilst maintaining acceptable, or at least comparable, classifier performance. We can also observe that the τ levels which proved effective in each case vary significantly between tests.

Table 5.2 presents the results obtained when applying our technique to problems with different structures. We can observe that the most influential factor in the choice of τ for a given problem is the number of input states involved.

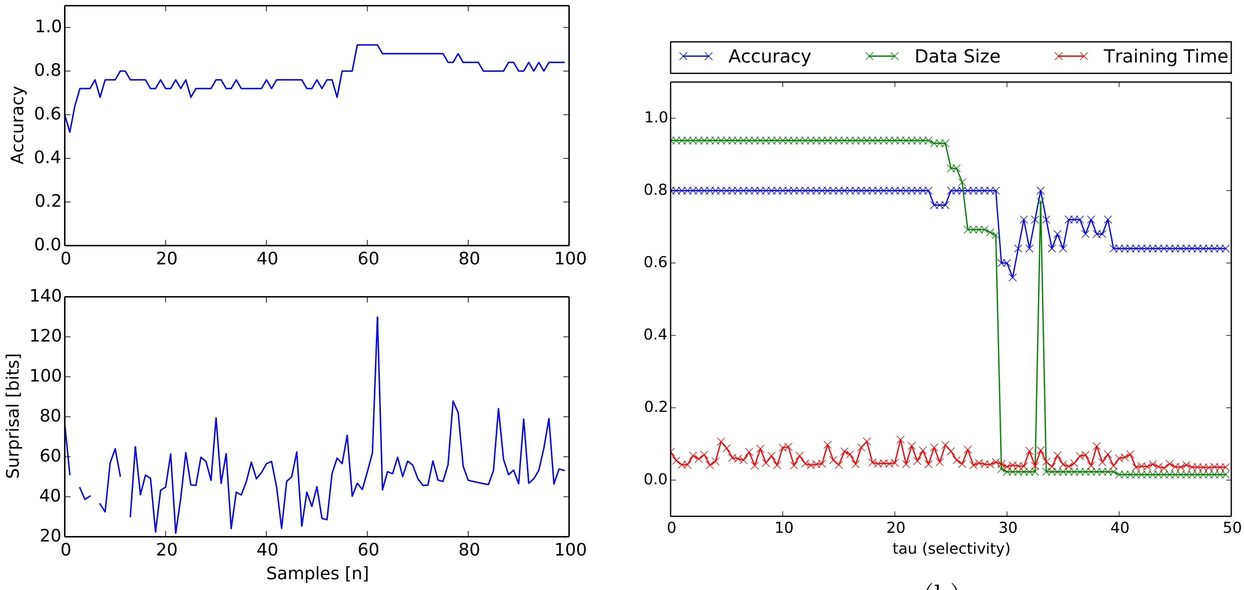
5.5.2 Results with Benchmarking Datasets

Fig. 5.6a shows the evolution of the accuracy and surprisal measurements for UC-3D data. We can observe that, as in the previous cases, accuracy increases with the addition of training examples, a trend which is not locally constant but generally observable. Unlike the previous case, there is no clear stabilization of surprisal or accuracy, which we attribute to the noisy nature of this dataset.

Fig. 5.6b shows the results of the application of our technique to the action descriptor classifier. We can observe that the system is able to drastically reduce the number of training examples employed while maintaining acceptable levels of accuracy. Similarly to Fig. 5.3a, three different regions emerge:

1. $\tau \in [0, 25]$: Increments in τ do not seem to carry much influence on any of the three metrics;

2. $\tau \in]25, 35]$: A drastic drop in dataset size can be observed, maintaining accuracy;

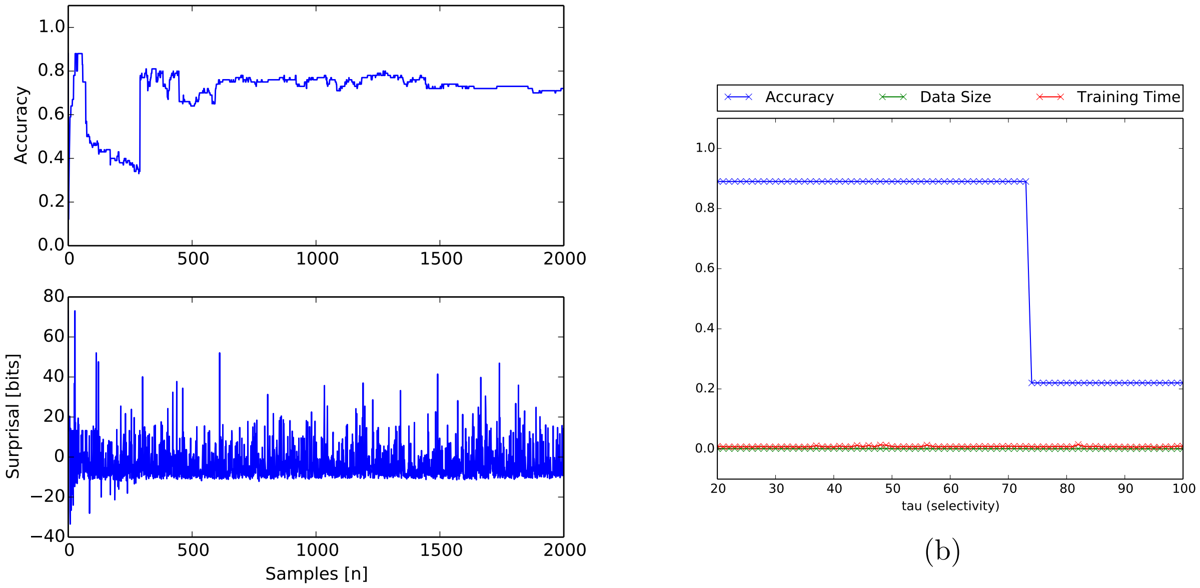


(a) (b)

Figure 5.6: Results of the tests using the technique presented in [125]. (a): An illustration of the classifier's learning curve. Surprisal is calculated with respect to the next instance to be added. (b): Results obtained using the action descriptor technique by measuring the technique's accuracy, training time and dataset reduction for varying τ.

Table 5.3: Results obtained for the various benchmarking datasets. Final training time is expressed as a fraction of the initial training time. The highlighted values correspond to the cases where our approach performed best.

Dataset	A	D	T	τ
UC-3D (no surprisal)	80%	100%	100%	(n/a)
UC-3D	**70%**	< **1%**	**100%**	**32**
UC-3D	60%	< 1%	100%	40
IJCNN (no surprisal)	92%	100%	100%	(n/a)
IJCNN	**82%**	< **1%**	**1.5%**	**72**
IJCNN	20%	< 1%	1.5%	80
IJCNN ([155])	99.8	8%	(n/a)	(n/a)
USPS (no surprisal)	75%	100%	100%	(n/a)
USPS	**75%**	**67%**	**89%**	**13.2**
USPS	2%	< 1%	45%	14
USPS ([118])	78%	1.2%	(n/a)	(n/a)



(a)

(b)

Figure 5.7: Results of the tests using the IJCNN dataset. (a): An illustration of the classifier's learning curve. Surprisal is calculated with respect to the next example to be added. (b): Results obtained using the IJCNN dataset by measuring the technique's accuracy, training time and dataset reduction for varying τ.

3. $\tau > 25$: The system is saturated; few examples are discarded at each step, as all available examples cause high surprisal. Accuracy shows great variation, while training time remains approximately constant.

Additional tests were performed with the IJCNN and USPS datasets. A similar behaviour was observed for these datasets, as illustrated in Fig. 5.7, where a significant reduction in dataset size can be observed. The latter also features a very abrupt drop in accuracy, as observed in some of the artificial datasets. This can be attributed to the fact that, at this τ level, the reduction achieved is already very high; when this level reaches a very high threshold, one or two crucial examples for the binary classifier are removed, drastically lowering its accuracy. Table 5.3 shows a highlight of the results obtained regarding the metrics we are considering.

5.5.3 Discussion

Results with non-noisy artificial data, for binary, multiclass and multilabel output, illustrated in Figs 5.3 through 5.5 and Table 5.1, show that the system's concept is sound. Major dataset reductions can be observed, with minimal impact to classification performance. The learning curves illustrated show clear a relationship between the stabilization of surprisal accuracy, supporting the idea that surprisal predicts the classifier's knowledge state. We can also observe a linear trend between the maximum data reduction achievable and the number of output classes, which can be explained by the fact that, on a non-noisy dataset, the minimum number of examples needed will correlate with the number of output classes. The system also achieves high reduction levels when operating on noisy artificial data, despite the fact that surprisal no longer stabilizes, and that the learning curve is no longer monotonous.

The system has also achieved significant reductions for the benchmarking datasets, as illustrated in Figs. 5.6, 5.7 and Table 5.3. The initial accuracy of the naïve Bayes classifiers employed was not excellent, but the system was able to maintain it while achieving large dataset reductions. Accuracy in these systems drops suddenly, e.g. in Fig. 5.6b, which is due to the influence of each example discarded on the subsequent examples, causing the amassing of information value in a small number of samples which, when discarded, produce a sudden impact on performance. Regarding in particular the UC-3D dataset, implementing the classifiers independently would yield significantly better results, which we have refrained from doing to so maintain fairness, analysing each dataset as a whole in our benchmark. In summary, we have achieved a 70% reduction on the UC-3D dataset, a 99% reduction on the IJCNN dataset and a 33% reduction on the USPS dataset, thus demonstrating Claim 1 of Section 5.1.

Dataset reduction has yielded a very significant reduction in the training time in almost all datasets, as illustrated in Figs. 5.3 through 5.7. These benefits were not as expressive in the UC-3D and USPS datasets, which we attribute to the fact that these datasets exhibit a low sample-to-feature ratio. In fact, while the IJCNN dataset uses 50000 data points for 22 features (a ratio of 2200), the UC-3D yielded only 130 data points for 23 features (a ratio of 5.7), and the USPS dataset yielded 5000 data points for 128 features (a ratio of 39). Since our training procedure exhibits $\mathcal{O}(n)$ complexity in number of output variables, features and examples, these low ratios cause the gains in training time to be overpowered by the complexity caused by the number of features and output classes in the data.

We compare our work to the dataset reduction techniques of [118] and [155], as seen in Table 5.3. Regarding the USPS dataset, these works outperform our technique in all our metrics. However, our technique outperforms related work on the IJCNN dataset, with [155] reducing the dataset to 8% total data, and our technique managing to reduce it to less than 1% with no impact on classifier performance when compared to our baseline. These results seem to indicate that our technique is effective for binary datasets, but becomes less effective when dealing with multiclass outputs. This can be overcome, for instance, by splitting the multiclass problem into a one-vs-all problem, using an ensemble of binary classifiers, taking advantage of our system's strong performance when operating on binary datasets.

We can observe, in Figs. 5.3 through 5.6, a strong disparity in the τ values employed. Table 5.2 shows that the optimal τ value varies with problem structure, mainly with variable discretisation, *i.e.* the number of output classes in a multiclass problem. In order to select the τ value for a specific problem, it is important to compare its structure to those found in Table 5.2, using the recommended value as a first approximation.

5.6 Summary

In this chapter, we set out to determine whether it is possible to apply the surprisal measurement to reduce the amount of data used to train a classifier while maintaining acceptable levels of performance. We have introduced a novel methodology for dataset reduction based on surprisal, and presented results showing that our system can significantly reduce the number of samples used by training procedures using synthetic data and the UC-3D, IJCNN and USPS benchmarking datasets.

This technique can now be applied in any problems where Bayesian classifiers are used, such as action recognition [125], optical character recognition [61] or intrusion detection [10]. In fact, the technique should be applicable to any problem in which knowledge is probabilistically represented, by extracting surprisal being from any associated distributions.

By applying this technique, we take a step towards being able to learn the user's characteristics in an efficient manner, striving to a complete yet efficient representation of the user. However, we are yet to apply this knowledge in making useful decisions, having only thus far discussed ways of extracting it from the user. Part III of the thesis will now introduce our techniques for using this information on the user and processing it to achieve successful decision-making and positive impact on the user.

Part III

User-Adaptive Behaviour

Chapter 6

Preliminary Experiments on User-Adaptive Decision Making

"A robot may not injure a human being or, through inaction, allow a human being to come to harm.
A robot must obey orders given it by human beings except where such orders would conflict with the First Law.
A robot must protect its own existence as long as such protection does not conflict with the First or Second Law."
— Isaac Asimov, Runaround (I, Robot)

Having the means to gain information on its user, a basic ability of an autonomous user-adaptive robot is then to be able to analyse its environment, and to use that information to make decisions that guide it towards its goal. The chapters of Part II have dealt with the development of such user modelling systems able to efficiently capture the characteristics of the user. This information should now be used to synthesize behaviour that is automatically adapted to the user in question, taking steps towards fulfilling the main goals of this work, as introduced in Chapter 1.

The model presented in this chapter, dubbed the "adaptive services" model, tackles the problem as one of estimation which, in essence, aims to derive a set of rules (much like those in the quote above) that establish a relationship between a set of contextual and user-related variables and the correct actions to take. It is based on two key concepts, *expectation* and *satisfaction*, that aim to establish a causal relationship: if the robot performs the action that the user expects the robot to perform in a given context, then the user will be satisfied and the robot will have been successful. In order to tackle the problem, two Bayesian Programs (Chapter 2) are presented for estimating the user's expectation from the current context and previous satisfaction, and estimating satisfaction from the data gathered from sensors. With these two mathematical tools, an execution loop was built such that the robot would iteratively learn the correspondence between each combination of context variables and desired actions meaning that, in time, it would be fully adapted to the user. In this case, the user model was assumed to provide the system with long-term characteristics in the form of user information, which would inform the expectation estimator.

This chapter presents the adaptive services technique, a preliminary user-adaptive decision making technique, as well as the experiments, results and discussion that were built on it. The remainder of this chapter is organized as follows:

- Section 6.1 presents the main goals, contributions and claims of the chapter;
- Section 6.2 presents the preliminary adaptive services model built on Bayesian Programming;
- Section 6.3 presents the experiments performed on this model, the corresponding results and discussion;
- Section 6.4 presents a summary of the chapter.

6.1 Chapter Goals and Contributions

The main goal of this chapter is to present and experimentally demonstrate a novel decision-making technique for social robots. This model introduces the following key innovations:

- A novel decision-making technique for user-adaptive operation based on the concepts of expectation and satisfaction;
- Two Bayesian Programs to estimate the user's expectation and satisfaction.

We aim to show that:

1. The system is able to learn the user's expectations in different contexts;
2. The system is able to act according to the user's expectation;
3. The system, once trained, is able to maintain the user in a high satisfaction state;
4. The system is able to, to some degree, become resilient to errors in the estimation of satisfaction.

6.2 The Adaptive Services Model

Let us assume, as in Chapter 1 that a service robot is interacting with a person, on one-on-one interaction. How can the robot provide a service that is *pertinent, appropriate* and *adjusted to the context* that the robot and the user are immersed in? More formally, given a discrete set of actions $x_i \in \chi = \{x_1\ x_2\ ...\ x_n\}$, where χ corresponds to the space of the actions that the robot can perform, how can the robot determine which action should be performed in a given situation? We assume that the user is consistent in their preferences, *i.e.* that they always prefer the same service in a given context.

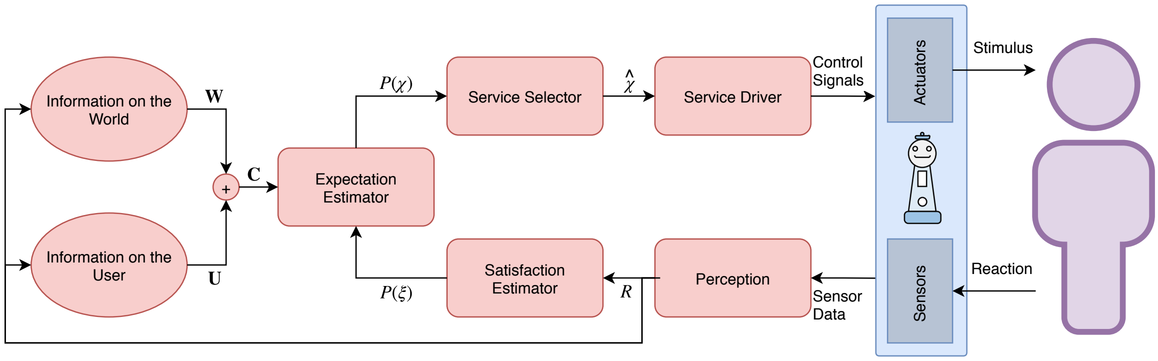


Figure 6.1: An overview of the composition and functionality of the envisioned system. The rounded rectangles represent data processing modules, the ellipses represent data sources and memory, and the dark rectangle represents all of the robot's hardware.

We propose that the interaction between a robot and a human agent can be made adaptive through the application and exploitation of two concepts: *expectation* and *satisfaction*. The user *expects* the robot to perform a task, and when the robot performs this task, the user is *satisfied* to a certain degree. The estimation of these measures and the associated decision making model are the main focus of this first work and are formally defined in the following sections.

6.2.1 Functional Description

Let $\mathbf{C}$ be the context of the interaction as the robot perceives it. Contextual information is divided into two types: information on the user $\mathbf{U} = [u_1 \; u_2 \; ... \; u_m]$ and on the world $\mathbf{W} = [w_1 \; w_2 \; ... \; w_l]$. Variable u_i pertains to the user (*e.g.* their preferences output from the user model) while w_i is a contextual variable. Thus, context $\mathbf{C}$ is given by:

$$\mathbf{C} = [\mathbf{U} \; \mathbf{W}] \tag{6.1}$$

This information is used to estimate the user's *expectation*. We aim to determine the *probability distribution* of user expectation, $\hat{\chi}$, as a function of the current context and user *satisfaction level* ξ obtained in previous interactions, such that:

$$\hat{\chi} = f(\xi, \mathbf{C}) \tag{6.2}$$

Knowing this distribution will allow for the selection of an action which is most likely to match the user's expectation and achieve satisfaction.

Every time a an action takes place, the user reacts to it, as illustrated in Fig. 6.1. This reaction is measured by the robot's sensors, generating new signals which are processed by the *Perception* module (as in Fig. 2.4, generating the *user's reaction* variable, $\mathbf{R}$. $\mathbf{R}$ is a result of the processing of the raw signals that are captured by the robot's sensors and is also a vector of the form $[r_1, r_2, ..., r_n]$, where r_k is a component of the user's reaction as perceived by the robot. The user's reaction is then used as a means to estimate their satisfaction level, ξ, which aims to describe how well the user accepts the robot's actions. It is mapped into the $[0, 1]$ range, where $\xi = 1$ indicates that the user is maximally satisfied.

6.2.2 Model Specification

Decomposition

We employ Bayesian estimators for estimating ξ and χ, which are implemented using Bayesian Programming [46]. We propose to estimate the user's satisfaction level as $\hat{\xi} = P(\xi|\mathbf{R})$, yielding:

$$\hat{\xi} = P(\xi|\mathbf{R}) \propto P(\xi)P(\mathbf{R}|\xi). \tag{6.3}$$

Since we have no prior information on user satisfaction, $P(\xi)$ is defined as a uniform distribution.

The *Expectation Estimator*'s objective is to determine which of the actions known by the robot is the most likely to correspond to the user's current expectation. By applying Bayes' Rule, we obtain:

$$P(\chi|\xi, \mathbf{C}) \propto P(\chi)P(\xi, \mathbf{C}|\chi). \tag{6.4}$$

Which, applying the Chain Rule to the term $P(\xi, \mathbf{C}|\chi)$, yields:

$$P(\chi|\xi, \mathbf{C}) \propto P(\chi)P(\mathbf{C}|\chi)P(\xi|\mathbf{C}, \chi). \tag{6.5}$$

The action is selected by the *Service Selector* through the expression

$$\hat{\chi} = \underset{x}{\mathrm{argmax}} \ \ P(\chi|\xi = 1, \mathbf{C}), \tag{6.6}$$

i.e. by finding the action that maximizes the probability of user satisfaction given the current context and user status.

Inference

Regarding the estimation of satisfaction, the $P(\mathbf{R}|\xi)$ factor in Eq. 6.3 encodes our learning process and is described using a histogram. In order to learn this distribution, we can perform a number of observations, where each observation produces a record of the form $\{Reaction, Satisfaction\}$ where ξ is known. Then $P(r_k|\xi)$ can be determined by operating over the records:

$$b_{i,j} = P(r_k = i|\xi = j) = \frac{1}{N_t}N_{r_k} \tag{6.7}$$

where N_t is the total number of available records, and N_{r_k} is the number of records where $r_k = j$, j can be either 0 or 1, and i represents the i-th possible state of r_k. An example of these matrices can be seen in Fig. 6.2.

Regarding the estimation of expectation, all of the right-hand side terms in Eq. 6.5 can also be described using histograms. These are recalculated on every interaction based on the record of the form $\{Expectation, Context, Satisfaction\}$ that is produced. The recalculation of these distributions, and the way they influence the decisions the system takes, are the key to this model's adaptivity. This process is described as a Bayesian Program in Fig. 6.3. This decomposition, as opposed to the different ones that could arise from different applications of the Chain Rule, is particularly advantageous, since it exposes the $P(C|\chi)$ distribution, which is key to the success of the learning procedure.

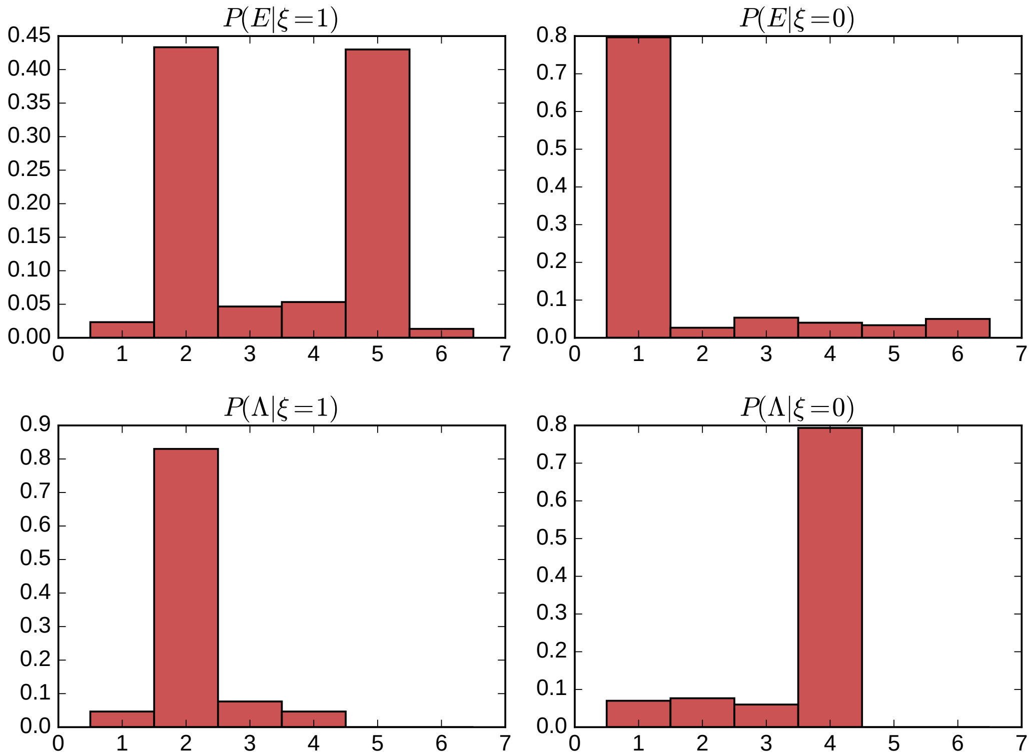


Figure 6.2: An example of the probability distributions produced during the training of the Satisfaction Estimator, and that can be used to generate the simulated user's reactions. The classes in the bottom axes refer to the classes of **U** as instantiated in Eq. 6.9.

Expectation and Satisfaction Estimators

program

description

specification

Variables:

$\mathbf{C}$, χ, ξ, $\mathbf{R}$

Decomposition:

$P(\chi, \xi, \mathbf{C}) \propto P(\chi)P(\mathbf{C}|\chi)P(\xi|\mathbf{C}, \chi)$

$P(\xi, \mathbf{R}) \propto P(\xi)P(\mathbf{R}|\xi)$

Formulation:

$P(\mathbf{C}|\chi), P(\xi|\mathbf{C}, \chi), P(\mathbf{R}|\xi)$: Learned Histograms.

$P(\chi), P(\xi)$: Uniform Distribution.

Identification: Training from records obtained from interaction.

Questions:

Expectation: $P(\chi|\xi, \mathbf{C})$

Satisfaction: $P(\xi|\mathbf{R})$

Figure 6.3: Bayesian Program for performing satisfaction and expectation estimation.

Since the user is assumed to maintain their preferences under equal context information, the Markov assumption is employed. We argue that this is a reasonable assumption if we assume that we can increase the dimensionality of the model in order to increase its descriptive power and support this assumption. For example, we can add a contextual variable describing the user's personality, allowing the robot to adapt not only to the context at hand but to that aspect of the user as well. Therefore, we assume that the previously-estimated expectation distribution, at each time step, can be used as the prior distribution for the subsequent steps, taking care to penalize erroneous decisions by reducing the probability of the erroneous action in the prior distribution injected in the next step. *i.e.* for a time instant k:

$$P(\chi_k|\xi_{k-1}, \mathbf{C}_k) \propto P(\chi_{k-1})P(\xi_{k-1}, \mathbf{C}_{k-1}|\chi_{k-1}), \tag{6.8}$$

a detail have been omitted from previous equations for the sake of readability. This corresponds to a computational implementation of the inference described by Eq. 6.4. χ_k is computed synchronously every time an action takes place and a reaction is observed.

6.3 Experiments

6.3.1 Model Instantiation and Experimental Methodology

For these trials, we have instantiated the model as follows. $\mathbf{U}$ is constituted by both the user's current action and displayed emotion (see the following equation).

$$\begin{aligned} \mathbf{U} = [u_1\ u_2], &\text{with} \\ u_1 \in \mathbf{E} : \mathbf{E} \equiv \{&sad, happy, angry, \\ &scared, joyful, neutral\} \\ u_2 \in \Lambda : \Lambda \equiv \{&wave, nod, walk, \\ &shake\ head\}, \end{aligned} \tag{6.9}$$

and $\mathbf{W}$ is composed by the user and robot's location and the time of day

$$\mathbf{W} = [w_1\ w_2\ w_3] \tag{6.10}$$

where w_1 and w_2 contain the user's and robot's location, respectively, and w_3 contains the current time of day. In these experiments, the user's reaction is conveyed by a different materialization of the same set of variables as $\mathbf{U}$, although there is no strict need for this to be the case:

$$\mathbf{R} \equiv \mathbf{U} \tag{6.11}$$

Additionally, we assume that the reaction variables are statistically independent, yielding

$$\hat{\xi} = P(\xi|\mathbf{R}) \propto P(\xi)P(r_1|\xi)P(r_2|\xi). \tag{6.12}$$

This is a strong independence assumption, which was particularly useful for this formulation, and may not hold for other instantiations of the model in which these variables are clearly not independent.

Figure 6.4: A user interacting with the robot.

We strove to design simulations that were as realistic as possible, namely by the injection of uncertainty in certain steps of the process. We have implemented a module which emulated the "human" component in the system by probabilistically generating an appropriate reaction to the stimuli it receives, depending on whether it is satisfied. This module implemented a *person profile*, essentially defined by the distribution $P(\mathbf{R}|\xi)$ and a set of constant, coherent rules that allow it to establish a connection between $\mathbf{C}$ and χ. For example, a profile may be constituted by the distribution illustrated in Fig. 6.2 and the rule "when the robot and the user are in the same location, the user wants to be entertained".

Two types of short-term tests were conducted:

- using constant $\mathbf{W}$;
- using using randomly-varying $\mathbf{W}$.

The first test had the goal of determining if the system was able to converge to the correct solution, and how quickly; the second type of test aimed to study how the system could adapt to constant changes in the environment. We have also measured the average number of cycles needed to converge to the correct solution when starting with no knowledge. Long-term simulation tests were also conducted, in order to observe the system's ability to converge to correct solutions given enough time to study the user. In these tests, $\mathbf{W}$ was varied as before, but the system was allowed to run for a much longer time, ranging from 1000 to 10000 trials. The Satisfaction Estimator was trained with 300 examples drawn from the aforementioned distribution, and no training was given to the Expectation Estimator.

Regarding the tests with users, we have implemented a number of services on the GrowMu robot [86], instantiating the χ vector as

$$\chi = [follow \;\; go\_to \;\; entertain \;\; do\_nothing]. \tag{6.13}$$

As a general way of measuring the level of adaptability of the system, we have calculated the error rate, *i.e.* the ratio of the number unsatisfactory trials over the number of satisfactory ones. This was done both cumulatively and within a sliding window that progresses through the results of the tests.

The tests with users, pictured in Fig. 6.4, were conducted over two "days", in simulated time by manipulating the robot's timekeeping primitives, allowing for more testing in less real time. The two users that participated in the test were instructed to act according to a "user profile", which guaranteed that there was consistency in their expectations. The satisfaction of these users was not estimated from their reaction; they were instead asked to state their satisfaction explicitly. This allowed us to gauge the system's effectiveness in the absence of satisfaction estimation error, and, more practically, to avoid having the robot perform actions that the user was not interested in.

6.3.2 Results and Discussion

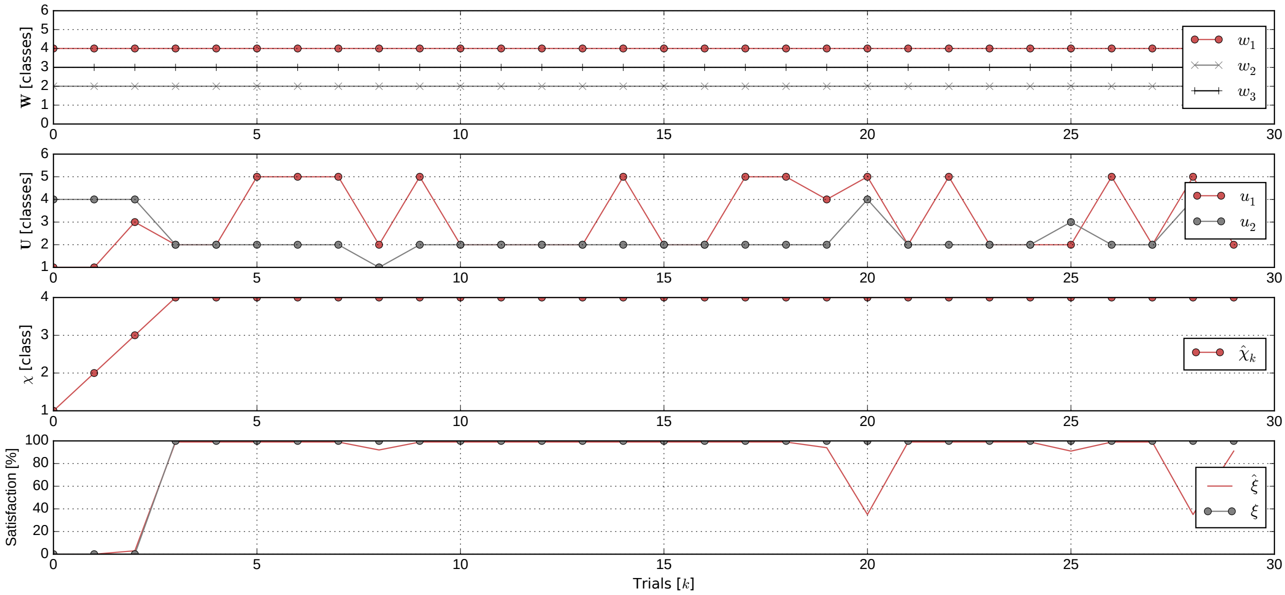
Fig. 6.5 illustrates the simulation results obtained, both for constant (6.5a) and varying (6.5b) **W**. The model took, on average, 3.85 iterations to converge on a "typical" trial such as the one presented in Fig. 6.5.

Fig. 6.6 shows the results of the long-term tests. We can observe, in Fig. 6.6a that the error rate that the system experiences tends to stabilize around 15%, but that despite its initial general decreasing trend, does not reach a negligible value. We can also observe, in Fig. 6.6b, that in the later stages of the experiment the error rate is rather unstable.

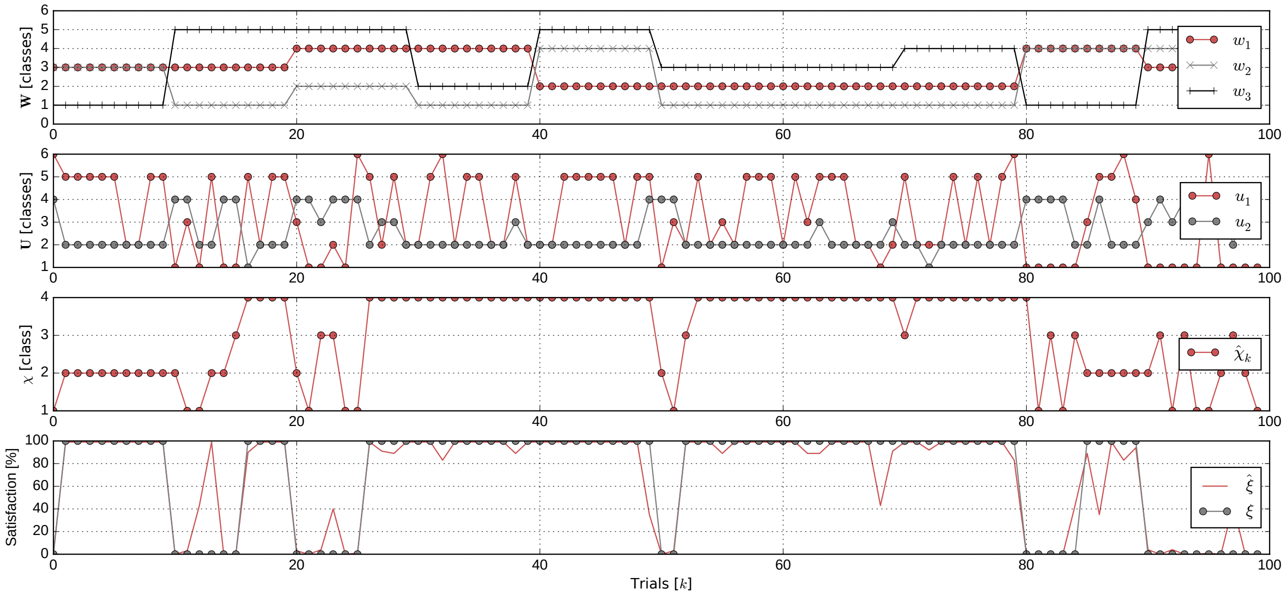
Lastly, the tests with users yielded the following results. When training the system, it displayed a behaviour similar to that of the first few trials in Fig. 6.6a, with relatively unstable error rates. Once the system was trained, and in the absence of the Satisfaction Estimator, it exhibited null error rates, inferring the user's preferences from context perfectly every time it was asked to do so. In other words, the error rate of the system tends to zero once enough iterations have taken place.

The system is able to effectively explore the solution space for the correct one, as demonstrated in Fig. 6.5, thus validating claims 2 and 3. We can also observe that, when a combination of context variables is repeated, the system already knows the correct action to take, thus validating claim 1. When found, it tends to maintain that decision dealing, to some degree, with possible errors in the estimation of the user's satisfaction, thus validating claim 4.

In the long-term tests we can observe that the error rate does not stabilize when calculated within a sliding window, although it does show a clear tendency in the cumulative case. We attribute this observation to the fact that the user's satisfaction is generated and estimated by probabilistic models; although not explicitly measured in this work, the satisfaction estimator suffers from a significant error rate itself. Furthermore, we can observe that the system converges to a low error rate early in the test, showing that it does not need an exceedingly large amount of information to be able to operate well.

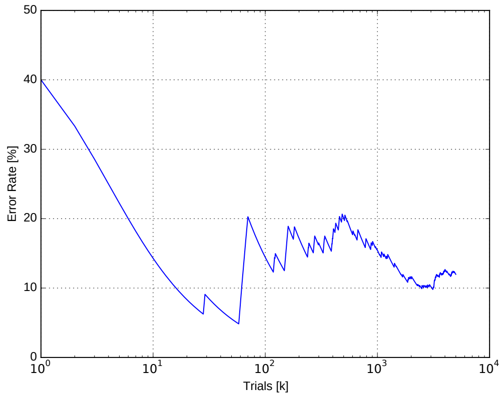


(a) Results obtained during simulated system testing with constant **W**.

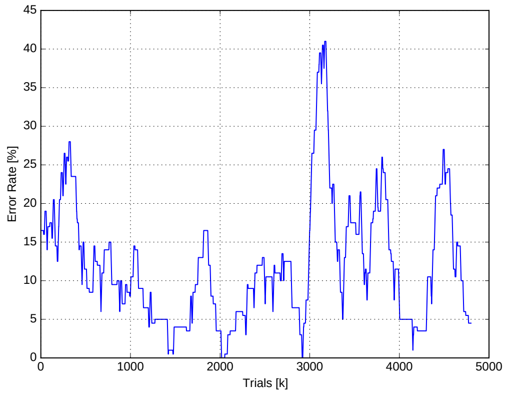


(b) Results obtained during simulated system testing with randomly-varying **W**.

Figure 6.5: The results obtained from the short-term simulation trials. **W** is varied in these tests, aiming to determine the system's ability to learn the user's preferences in different contexts.



(a) Error rate calculated cumulatively for the whole experiment window.



(b) Error rate within a sliding window with a width of 100 trials.

Figure 6.6: Error rate progression observed during the experiments. Please note the logarithmic scale on the x-axis of (a), used to evidence the high initial error rate observed during the training process.

During testing, we have observed that the system is able, to a small degree, to deal with changes in the user's preferences, if the changes happen early enough in the experiment. If they happen too late, on the other hand, the system takes a large number of iterations to adjust, up to the number of successful iterations that occurred under the former preference.

In the tests with real users, where the Satisfaction Estimator was replaced by the user's explicit answer, and where we observed that the system can converge to a null error rate, we can conclude that the system is able to achieve a full knowledge of the user's preferences in the contexts under inspection.

6.4 Summary

In this work we have presented and experimentally demonstrated a preliminary adaptive services system. We have performed tests in a simulated benchmark, demonstrating our technique's abilities and mathematical validity. Our results demonstrate the claims from Section 6.1, showing that the system is able to make positive decisions, keeping the user satisfied, and also to explore and learn from the user.

Despite proving our initial claims and providing insight into how decision-making can be applied to user-adaptiveness, this work is found to be limited in several ways. It is unable to deal with any user feedback aside from the expression of satisfaction, and is unable to make decisions that involve planning beyond the next iteration in the interaction. Thus, despite the system's shown ability to adapt to the user, it is limited to a myopic planning horizon of just one iteration, operating in a greedy manner that can lead to sub-optimal results. Furthermore, the system is limited to a one-dimensional rewarding scheme, *viz.* the satisfaction variable. Lastly, the satisfaction variable is also a part of the user's state, resulting in an imbalanced partition of the user's state and in mathematical ambiguity as to which variables should be considered the user's state and which should not.

POMDPs allow for planning over a broad horizon, as well as to incorporate multiple rewards into the decision-making procedure. POMDPs also provide a standardized framework for the definition of state variables, as well as a system of assumptions that better circumscribes the underlying mathematical approach. As such, an extension to this technique was developed, with basis on this paradigm, and is presented in the next chapter.

Chapter 7

αPOMDP: POMDP-based User-Adaptive Decision Making

> "What does it mean to say that agents are self-interested? It does not necessarily mean that they want to cause harm to each other, or even that they care only about themselves. Instead, it means that each agent has his own description of which states of the world he likes—which can include good things happening to other agents—and that he acts in an attempt to bring about these states of the world."
> — Kevin Leyton-Brown and Yoav Shoham, Essentials of Game Theory

As Leyton-Brown and Shoham put, "each agent has his own description of which states of the world he likes", meaning that different states in state space are more valuable, in an agent's perspective, than others. This chapter presents αPOMDP, a POMDP-based decision-making mechanism able to learn and adapt to a user, refining the previous formulation by extending the concept of satisfaction to that of *state value*, which encode the aforementioned differences in subjective value attributed to states. Thus, instead of striving to maximize a single user-related variable, the concept of state value allows us to optimize *combinations of state variables* which are deemed more important than others. αPOMDP's formulation is based on the classical POMDP, as described in Chapter 2, and its development was governed by two main tenets:

1. **The system should be able to learn the information it needs to interact properly**: by receiving as little information *a priori* as possible, the system is able to adapt to any user, regardless of their individual characteristics. Thus, by enabling the system to learn by itself, it can achieve true adaptiveness;

2. **The system's actions should take into account the impact they produce on the user**: following the POMDP formulation, the robot is assumed to be the only influence on the user's state (as in Chapter 2). As such, the robot should be able to learn the impact of its actions on the user, *i.e.* the state transitions its actions cause and their effect on the user, as described by the state value.

By employing these principles, the system remains generic enough to adapt to any user, and also autonomous enough to gradually learn and improve its actions.

Instead of describing the user based only on the information from the user model or their satisfaction status, αPOMDP extends the representation of the user with a *joint state space* for both agents (robot and user) and context. αPOMDP models the impact of the robot's actions as a transition in this state space and, through the POMDP formulation, can plan for as wide a horizon as desired. Thus, the previous learning loop is extended to include not only the positive and negative reinforcement provided by the satisfaction variable, but also the full change in state space; the robot becomes able to learn and gauge the impact of its actions on the full state of the user, not only a single arbitrary value.

This chapter is organized as follows:

- Section 7.1 presents the goals and contributions of the chapter;
- Section 7.2 presents the main techniques that served as theoretical basis for αPOMDP;
- Section 7.3 presents the αPOMDP model, including the novel reward formulations and learning mechanism;
- Section 7.4 presents the experimental design employed;
- Section 7.5 presents and discusses the results obtained;
- Section 7.6 presents a summary of the chapter and concluding remarks.

7.1 Chapter Goals and Contributions

The main goal of this work is to present and demonstrate αPOMDP, a novel decision-making technique for social robots. It introduces the following innovative factors:

- A novel state-based reward formulation;
- A novel learning mechanism and execution loop.

We present experimental results obtained with the proposed system, showing that:

1. The system is able to make decisions that correlate positively with the value functions that encode the impact on the user, keeping them in the most valuable states for significant portions of the experiment;
2. The system is able to learn the impact of its actions on the user, exploring the user's state space to gain information that leads it to improve its performance;
3. The system can achieve these goals in realistic simulations and is transferable to experiments with real users.

As in previous chapters, Claims 1 through 3 will be discussed and validated in the discussion on results of Section 7.5, and represent the main goals of the experiments performed in this chapter.

7.2 Related Work

POMDPs are able to model a number of different scenarios, many of which applicable in user-adaptive robots. Their ability to deal with stochastic observations, costs and rewards make them especially suited for decision-making in the uncertain environment of human-robot interaction. This formulation has seen use in some of the surveyed works (Chapter 2), for instance in [142] or [70].

Two classes of techniques in the field of automated planning are the closest to αPOMDPin terms of goals and theoretical formulation: reward shaping [104], the basis for the novel state-based reward formulation, and model-based reinforcement learning [140], the basis for the novel learning mechanism and execution loop.

Reward shaping consists of manipulating the reward given to reward-based systems, such as POMDPs, to aid the learning system to achieve the goal. It generally constitutes an informed deviation from the parameters of the problem at hand, heuristically providing a reward that better steers the system towards its goal. We employ an entropy-based reward shaping mechanism that encourages the system to explore potentially rewarding actions in uncertain states, as seen in the following section.

Model-based reinforcement learning (MBRL) consists of applying reinforcement learning [15] to a previously-built model. MBRL allows for the fitting of a previous model, such as the transition model $P(s'|s,a)$, to the characteristics of the specific problem at hand. We employ a simplified version of MBRL based on Bayesian Learning in order to adjust our system's transition model as it executes.

7.3 αPOMDP

Let us recover the relevant assumptions of Chapter 2:

- The robot is the only influence on the user's state;
- The interaction takes place in a turn-taking manner, modelled in discrete time.

αPOMDP, illustrated in Fig. 7.1, extends the regular operation loop of POMDP-based systems with a knowledge integration and policy recalculation steps, which allow the system to gradually adapt to the user. At each iteration, the system updates its transition matrix T, which encodes the probabilities of each state-action combination resulting in each specific state. This allows it to, iteratively, learn the impact of its actions on the joint state space. Rewards are formulated as a function of *state values*: different states carry a different utility or value to the user. These values allow for the formulation of a single-valued reward function, able to be solved by common POMDP solvers, while taking into account the full state space.

7.3.1 Reward Function

The user's state is modelled as a combination of discrete variables, $s \in \mathbb{N}^n$, where n is the number of user characteristics under analysis, with each variable encoding one of the user's characteristics, the robot's state or a relevant contextual variable. These characteristics can be

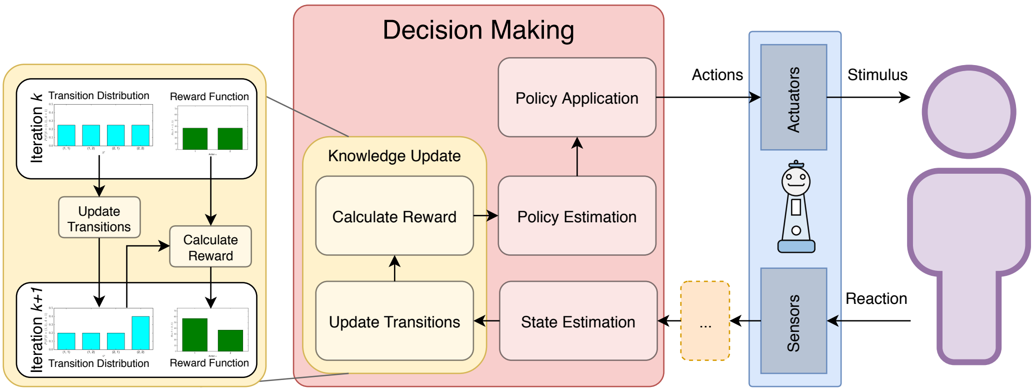


Figure 7.1: An expanded view of the "Decision Making" block of Fig. 1.2, providing an overview of the execution loop of αPOMDP. The transition and reward functions can be re-calculated each time new information is obtained by the system, resulting in a new policy which is better suited to the user.

inferred through the system's observations, as in the original POMDP formulation, or through the BUM approach of Chapter 3.

Whenever the system takes an action, it produces an impact on the user under the form of a state transition:

$$s' = \Gamma(s, a), \tag{7.1}$$

where s' is the user's final state after action a, s is the initial state and Γ is a hidden function through which the user transitions from state to state.

From the user's perspective, s' may be *more valuable*, *less valuable* or *equally valuable* when compared to s. This can be represented as a state value function $V(s) : s \to \mathbb{R}$, mapping each possible state to a scalar. This function can encode the *semantic* value of each state, such as the user's health in each state, or their immediate happiness. Thus, it allows us to quantify the importance of each state. With this basis, we can define the impact on the user produced by a robot's action as they transition from s to s':

$$I(s', s) = V(s') - V(s) = V(\Gamma(s, a)) - V(s) \tag{7.2}$$

quantifying the value of each state transition and, by extension, each possible action.

However, the $\Gamma(s, a)$ function is not known *a priori*, and may even depend on hidden variables, which are not accounted for in the formulation and, thus, introduce uncertainty in the process. The POMDP formulation already contains a solution to this problem in the form of the $T(s', s, a) = P(s'|s, a)$ function, the transition matrix, which encodes state transitions as probability distributions. Thus, in order to make its decisions while taking into account the likely impact on the user, the system can employ T: if an action is unlikely to produce positive impact, then it should be penalized, and vice-versa. As such, the basic State Value Reward (SVR) function is formulated as follows:

$$\begin{aligned} R(a, s) &= \sum_{s' \in S} T(s, s', a) * I \\ &= \sum_{s' \in S} P(s'|s, a) * (V(s') - V(s)) \end{aligned} \tag{7.3}$$

resulting in a scalar reward for each possible action in a given state, according to its probable impact on the user. This differs from the classical POMDP formulation in the fundamental sense that, through the $V(s)$ function, the value of the robot's action is now dependent on the user's states and not on actions in a given state. Actions are, thus, valued only by their influence on the user, and by their likelihood of transitioning the user to a state that is considered more valuable than the one they are currently in.

In order to encourage the system to gain information on the user, avoiding being stuck in the same state-loop indefinitely, we have devised an information-based term:

$$H = h(T(s, s', a)) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i), \tag{7.4}$$

where h is the entropy function [134]. The information term H will increase with the uncertainty in the $P(s'|s, a)$ distribution, reading its maximum when the distribution is uniform, *i.e.* when no information on the respective potential transition is known. Thus we formulate the State Value Reward with Information Term (ISVR):

$$R(a, s) = \sum_{s' \in S} T(s, s', a) \cdot I + H \tag{7.5}$$

By using the information term H, we increase the reward given to an action that leads to unknown transitions, thus encouraging the system to investigate the impact on the user of new actions.

Several $V(s)$ functions can be used to encode multiple rewards, such as when the robot needs to maximize several dimensions of the state, *e.g.* user health and happiness. This results in multiple formulations of the SVR of Eq. (7.3). By scalarising these multiple objectives we obtain the Multiple State Value Reward (MSVR):

$$R(a, s) = \sum_{k} \sum_{s'} w_k \cdot P(s'|s, a) \cdot (V_k(s') - V_k(s)) + H \tag{7.6}$$

which allows for the encoding of different semantic information, such as health status *vs* immediate happiness, into the weighted reward function through the w_k weights.

7.3.2 Transition and Reward Learning

In order to predict the impact of its actions on the user, the robot must approximate the Γ function (Eq. (7.1)). Through the application of the POMDP formulation, Γ can be approximated by $P(s'|s, a)$, which must be learned for each possible transition as the system executes.

Each interaction with the user, as seen in Fig. 7.1, yields transition information in the form of a sample:

$$L = \{s', s, a\}, \tag{7.7}$$

a tuple encoding the initial and final states, as well as the action employed by the robot. This information is used to learn $P(s'|s, a)$ by constructing a histogram, as usually seen in the naïve Bayes Classifier formalism, and as in our previous techniques:

$$P(s'|s, a) = \frac{1}{N} N(s', s, a) \tag{7.8}$$

where N is the number of available samples, and $N(s', s, a)$ is the number of samples where $S' = s', S = s, A = a$. A practical example of this mechanism in action is illustrated in Fig. 7.1.

These tuples are added to the distribution on every interaction, thus enriching the system's knowledge of Γ. By re-calculating the reward functions that depend on this function, the system's information is fully updated. This updated information can then be used to re-calculate the policy, resulting in a policy that is potentially better adapted to the user.

7.3.3 βPOMDP: Extension to Robotic Teams

αPOMDP is meant to guide a single robot in its interaction with the user, gradually learning the impact of its actions on the user and guiding them to high-value states. However, with some modification, it could also be applied to the management of a team of robots. The multiple agents interact with unknown users, as assumed before, and the Γ_i functions (of user i) of these users are unknown. The robots aim to approximate them in order to be able to predict the impact of their actions. If individual robots perform this task with individual users in an isolated manner, they will not be able to cooperate or replace one another if the need arises, for instance if one of the robots fail. They will also be unable to coordinate globally, failing to achieve the coordination level needed for a social robot team operating in a community setting, as seen for instance in project Robot-Era [57]. Despite not being tested in this work, this section presents the theoretical extensions needed for αPOMDP to operate with a team of robots.

βPOMDP, illustrated in Fig. 7.2, is an extension to αPOMDP for multiple agents. The agents operate in a shared state space S that models all of them and all of the users. For instance, if $s_{a,i}$ are the variables that model agent i, and $s_{u,j}$ the variables that model user j, then the state space is of the form

$$S = [s_{a,1}, s_{u,1}, s_{a,2}, s_{u,2}, ...] \tag{7.9}$$

The basic execution loop of Fig. 7.1 is largely unchanged, *i.e.* the robots act, observe the impact and update their knowledge of the user from their observations. The result of this knowledge update, namely the updated beliefs and transition matrices, are shared among agents. From this process, a unified belief is obtained:

$$b_f = F_b(b_1, b_2, ...), \tag{7.10}$$

where b_f is the fused belief, F_b a function that fuses beliefs and b_i the individual belief of each agent. Similarly, a unified transition matrix is obtained:

$$T_f = F_t(T_1, T_2, ...), \tag{7.11}$$

where T_f is the fused transition matrix, F_t is a function that fuses transition matrices and T_i is the transition matrix obtained by agent i.

This information fusion step allows each agent to plan their actions with the same knowledge as the others. Thus, they will all obtain a policy that is equal to that of the remaining agents, or at least equivalent. These common policies, when associated with the state that each individual agent is in, will lead to different actions for each agent:

$$a_{s,i} = \Pi(s_i), \tag{7.12}$$

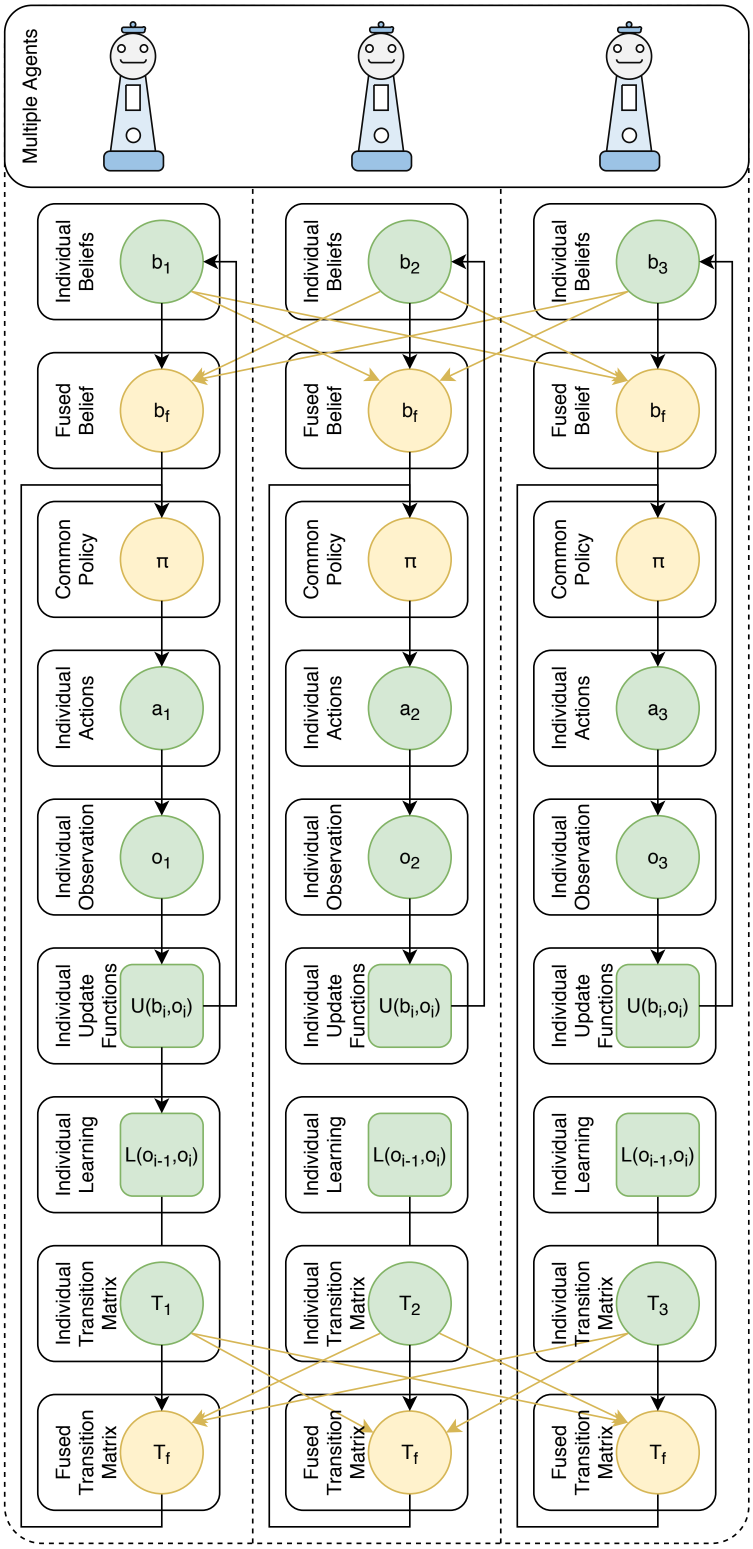


Figure 7.2: βPOMDP, an extension of αPOMDP for multiple agents. Agents share the results of the knowledge update steps to generate common policies which lead to coordinated actions.

where $a_{s,i}$ is the action to be performed by agent i in state s, Π is the common policy and s_i is the current state of agent i. However, since all agents share the same $V(s)$ functions, these actions, while different for each agent, will be in accordance with the common goals of the system, *i.e.* acting with an impact that transitions the global state to a better one.

7.4 Experiments

To validate αPOMDP, we applied it in the context of a realist use-case scenario, aiming to demonstrate the claims of Section 7.1. The scenario consists on a user interaction with a social robot, which models the user's state and performs actions to influence it, guided by the αPOMDP technique. We have performed both simulated and tests with real users, with both types of experiments being set in the same general scenario. We have released our code as an open-source package[1].

7.4.1 Use Case Scenario and Model Instantiation

Interaction takes place iteratively, following the loop of Fig. 1.2. Every iteration, the robot should decide about the action to execute, which is dependent on the current state of the user. For instance, giving the user chocolate could make them happier but potentially harm their health, and performing exercise may lessen their happiness but contribute to better health.

We define the state space as $S = \{S_1, ..., S_5\}$, $S_i \in \{1, 2, 3\}$ where:

S_1: (User) satisfaction, $S_1 = 1$ means the user is unsatisfied, and $S_1 = 3$ means they are fully satisfied;

S_2: (Robot) robot's current speaking volume, with $S_2 = 3$ meaning that the robot is at full volume;

S_3: (Robot) the robot's current distance to the user, with $S_3 = 3$ meaning that the robot is as close as possible.

S_4: (User) health, with higher levels indicating better health.;

S_5: (World) Time of Day.

With respect to the robot's action space, we define it as $A = \{A_1, ..., A_8\}$ with each action corresponding to:

A_1: Ask the user a question;

A_2: Move the robot 15cm forward;

A_3: Move the robot 15cm back;

A_4: Increase speaking volume by one interval;

[1] `https://github.com/gondsm/apomdp`

A_5: Decrease speaking volume by one interval;

A_6: Give the user food;

A_7: Give the user candy;

A_8: Do nothing.

Simulation Scenario

Each simulated *trial* consists on the execution of the loop described in Fig. 7.1 for a set number of n iterations. For each trial, a number of parameters are variable:

1. The POMDP solver in use by the system;
2. The number of iterations that the system will run (n);
3. The reward function to use;
4. The $V(s)$ function(s);
5. The simulated user's profile;
6. The policy calculation period t_c;
7. The state and action spaces.

In order to determine if different solvers have an impact in the system's performance, we have used two different POMDP solvers, QMDP and SARSOP, discussed in Chapter 2. Each experimental condition was repeated 1000 times for statistical significance.

The system interacted with a simulated user characterized by a $V(s)$ function and a user profile composed of a deterministic Γ function (Eq. (7.1)) that maps each a and s pair to the resulting s'. The $V(s)$ function was re-generated randomly for each trial, attributing a random discrete value to each state. Similarly, the user profile was re-defined between trials, with each state-action pair being attributed a random destination state. This stochastic generation of the user profile can lead to a number of pitfalls, for instance when the profile is generated in a way that high-value states are unreachable, or that all states receive extremely low values, potentially hindering the system's efforts.

The policy calculation period t_c is the periodicity in which the system is allowed to recalculate its policy, with a $t_c = 1$ meaning that the system re-calculates the policy every iteration. This parameter also varied, deterministically, between tests, namely to determine the system's robustness to failures in the integration of new data, and to allow determining whether the policy needs to be re-calculated at each step for the system to execute successfully.

We have tested all of our three proposals (SVR, ISVR and MSVR). For the MSVR reward, the function was randomized at each trial, with new weights being generated and three simultaneous $V(s)$ functions being used.

Simulation results were obtained using both the full and a reduced version of the scenario of Section 7.4.1, which limited it to $S = \{S_1, S_2\}$ and three actions. The reduced tests allowed us to select the optimal experimental parameters, which were then applied to the full problem. Interested readers are strongly encouraged to analyse our code and replicate these experiments.

Real Scenario

The real scenario is a transposition of the scenario described in Section 7.4.1 to a real setting, implemented on the GrowMu social robot. The state and action spaces were trimmed to variables S_1 through S_3 and A_1 through A_5, limiting our tests to a duration of about 10 minutes. The user's state was estimated at the end of each iteration via verbal interaction. The human-robot interaction occurred as described in Section 7.4.1, where each iteration consists of the robot performing an action and estimating the resulting user status. Based on the result analysis from the simulated scenario variant trials, we have used the QMDP solver, the ISVR reward, a t_c of 1 and formulated $V(s) = 10 * s_1$, reinforcing only the user's satisfaction.

7.4.2 Evaluation Metrics

We employ the following evaluation metrics:

R_c: Cumulative Reward;

t_3: Iterations that the system spent in top 3 states;

$\bar{H}(T)$: Average entropy on the $T(s', s, a)$ function;

t: Execution Time.

R_c is defined as

$$R_c = \sum_k R_k \tag{7.13}$$

where R_k is the reward received by the system up to iteration k. The metric represented by t_3 is defined as

$$t_3 = \sum_{i=1}^{3} N_i \tag{7.14}$$

where N_i is the number of iterations spent in the i-th most valuable states, as defined by the $V(s)$ function. The average entropy of the T function, $\bar{H}(T)$ is defined as

$$\bar{H}(T) = \frac{1}{N} \sum_{s' \in S} \sum_{s \in S} \sum_{a \in A} H(T(s', s, a)) \tag{7.15}$$

with $H(T(s', s, a))$ defined as in Eq. (7.4), and N as the number of combinations of s', s and a. Execution time is measured in seconds per trial, and is treated statistically.

7.4.3 Implementation

Like the other techniques developed within this Thesis, such as BUM, the implementation of αPOMDP led to a novel ROS package, `apomdp`. The source code for this implementation is freely available under the GPLv3 open source license, as mentioned before.

`apomdp` can be used in two main modes: ROS-based and non-ROS operation. The first is used to run tests with real users, and the second to run simulated experiments. Both operation modes rely on the novel `apomdp.jl` library, which implements the POMDP definition and solving, policy extraction, *etc.*

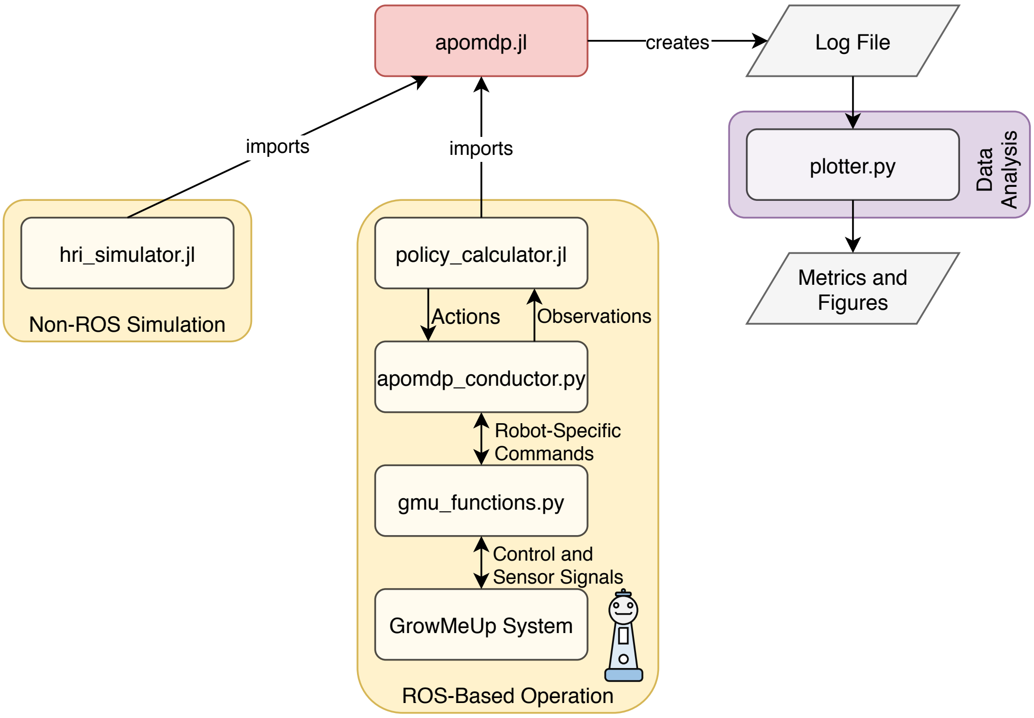
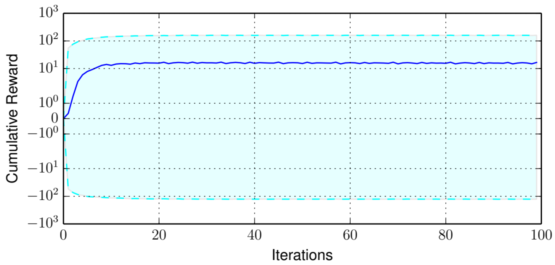


Figure 7.3: An overview of the `apomdp` package. A single definition of the POMDP problem and solver are present in the `apomdp.jl` script. This is then imported by both ROS and non-ROS simulators, which stimulate the system in different ways to produce log files, which contain information on the system's performance.

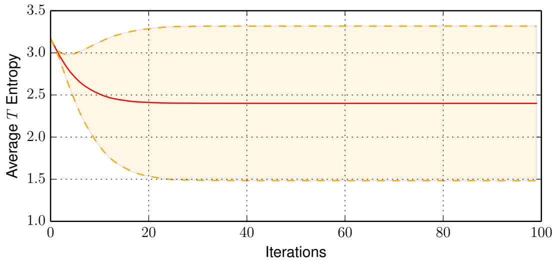
7.5 Results and Discussion

Figures 7.4, 7.5, 7.6 and 7.7 represent the evolution of the cumulative reward, average T entropy, as well as the number of iterations spent in each state according to its rank, for our simulated trials. The top graph of each figure represents the evolution of cumulative reward for the number of iterations used. The dark blue line represents the average value, while the cyan background represents the $\mu \pm 2\sigma$ area. The middle graph represents the evolution of the average T entropy, $\bar{H}(T)$, metric, with the lighter background representing the $\mu \pm 2\sigma$ area. The bottom graph represents the average number of iterations that the simulated user spent on each state, according to the rank of the state. States are ranked according to their value (as specified by the $V(s)$ function), with state 1 corresponding to the highest-value state, 2 to the second-highest, and so on. This allows us to visualize the system's ability to keep the user in a valuable state since, as mentioned before, each trial uses a different $V(s)$ function and user profile. Fig. 7.9 presents the results of seven trials that took place in the human scenario, using the same measurements.

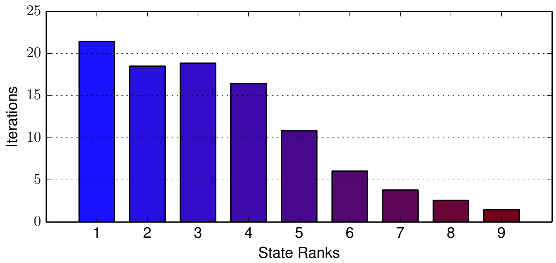
Table 7.1 presents the aggregate results of our simulated trials for varying conditions. Each row represents a single condition, *i.e.* one combination of the possible input parameters, which was run 1000 times. We can observe that, in general, the system is able to achieve high cumulative rewards with all reward functions. Furthermore, the system is able to maintain the user in the most valuable states, achieving t_3 values of 70% in the best cases. We can



(a) Evolution of cumulative reward.

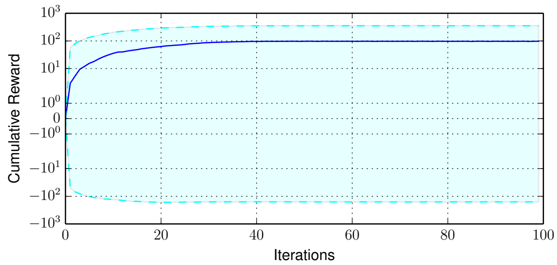


(b) Evolution of T entropy.

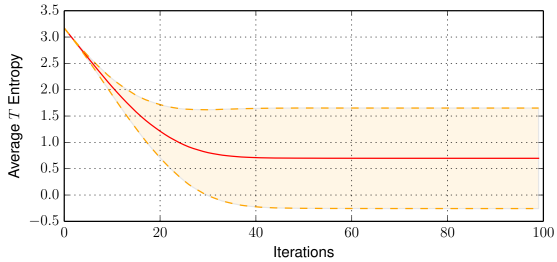


(c) Average iterations spent in each state according to rank.

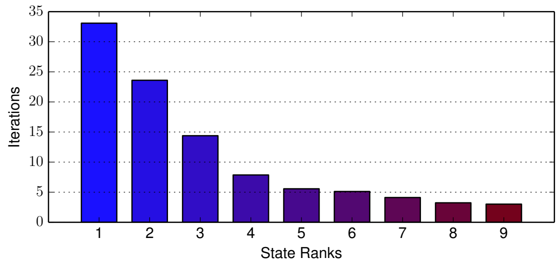
Figure 7.4: Results obtained using the QMDP solver and the **SVR** reward for 1000 trials of 100 iterations, re-calculating the policy every iteration. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.



(a) Evolution of cumulative reward.

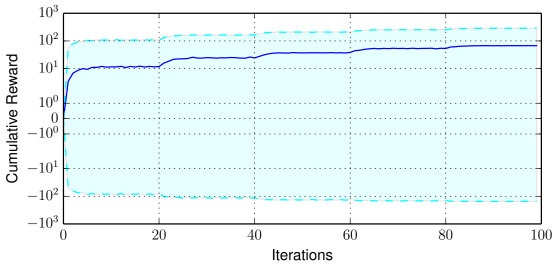


(b) Evolution of T entropy.

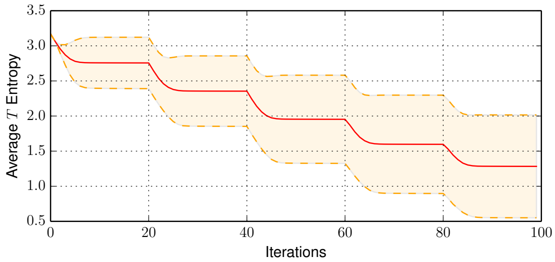


(c) Average iterations spent in each state according to rank.

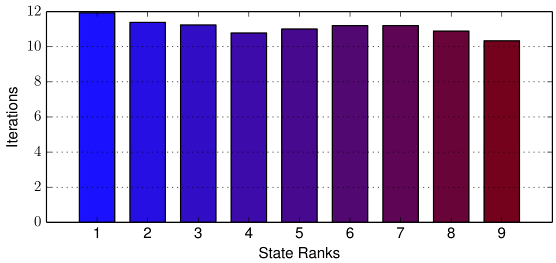
Figure 7.5: Results obtained using the QMDP solver and the **ISVR** reward for 1000 trials of 100 iterations, re-calculating the policy every iteration. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.



(a) Evolution of cumulative reward.

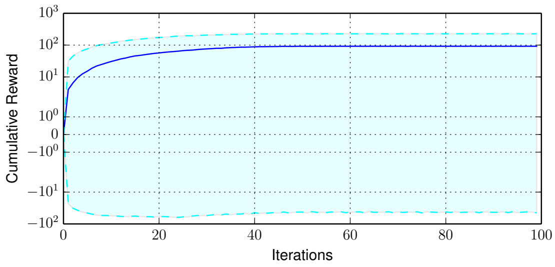


(b) Evolution of T entropy.

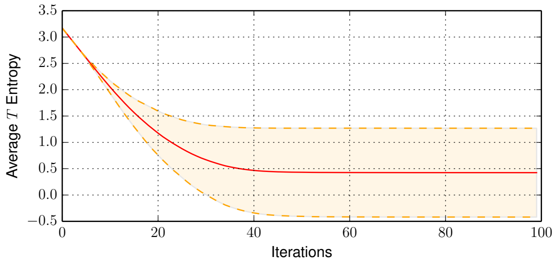


(c) Average iterations spent in each state according to rank.

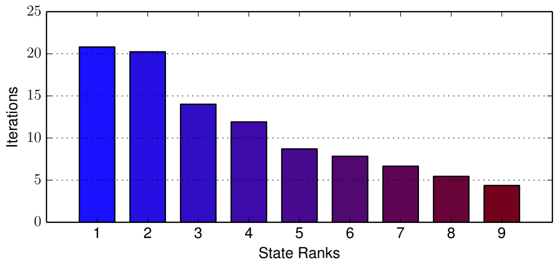
Figure 7.6: Results obtained using the SARSOP solver and the **ISVR** reward for 1000 trials of 100 iterations, re-calculating the policy every 20 iterations. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.



(a) Evolution of cumulative reward.

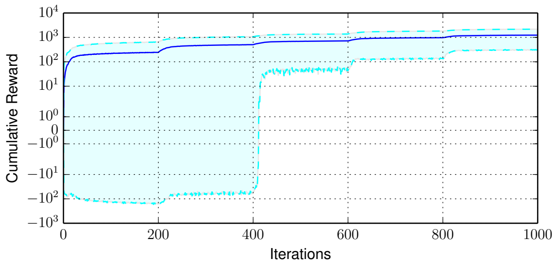


(b) Evolution of T entropy.

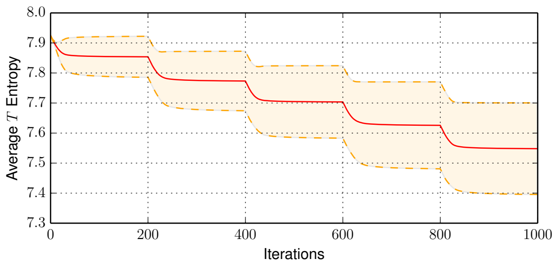


(c) Average iterations spent in each state according to rank.

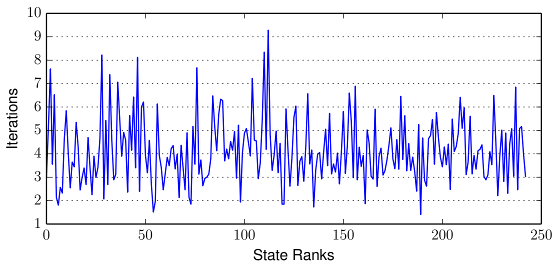
Figure 7.7: Results obtained using the SARSOP solver and the **MSVR** reward for 1000 trials of 100 iterations, re-calculating the policy every iteration. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.



(a) Evolution of cumulative reward.

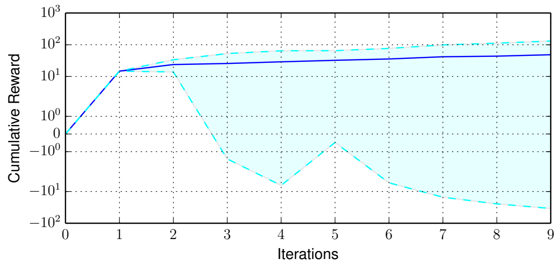


(b) Evolution of T entropy.

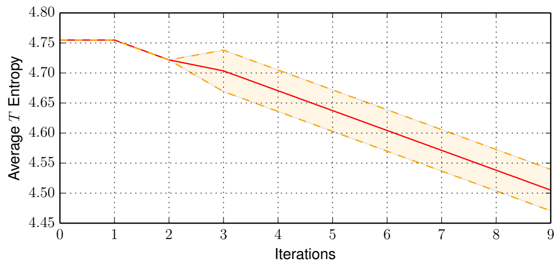


(c) Average iterations spent in each state according to rank.

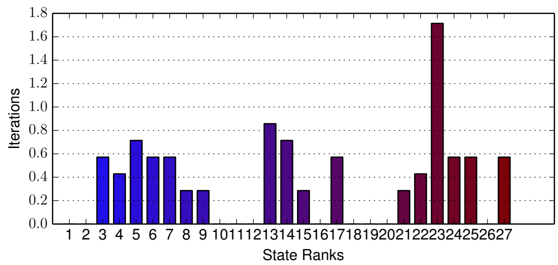
Figure 7.8: Results obtained for the complete scenario using the QMDP solver and the **ISVR** reward for 100 trials of 1000 iterations, re-calculating the policy at every 200 iterations. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.



(a) Evolution of cumulative reward.



(b) Evolution of T entropy.



(c) Average iterations spent in each state according to rank.

Figure 7.9: Results obtained using the QMDP solver and the **ISVR** reward for trials with human users. The top graph represents the cumulative reward, with the coloured background representing $\pm 2\sigma$. Similarly for the middle graph, representing the average entropy in the $P(s'|s,a)$ distributions. The bottom graph represents the average number of iterations spent in each state, from the most (left) to least (right) valuable.

also observe that the system is able to obtain low values for the final average entropy of the transition function, reaching values as low as 0.42 bits in the best cases. Thus, in general terms, we can conclude that claim 1 is validated.

Figs 7.4 and 7.5 illustrate the performance of the SVR and ISVR rewards, respectively. We can observe that the ISVR reward obtains, on average, better performance than SVR for all metrics, achieving increases of as much as a 6.3x increase on cumulative reward, 13% increase in t_3 and a 71% decrease in final entropy of the transition function, according to Table 7.1. This indicates that the ISVR formulation results in a system that is much more capable of maintaining the user in valuable states, and also in gaining information about them. In fact, the ISVR formulation leads the agent to properly explore the user's transitions, thus gaining information that the SVR-enabled agent most likely did not gain. The MSVR formulation, illustrated in Fig 7.7, performs similarly to ISVR, since it also incorporates the information term that exists in ISVR. Thus, in using the ISVR formulation, our results support claim 2.

Regarding the performance of the solver used, QMDP and SARSOP, we could not find a significant difference. For the R_c, t_3 and entropy metrics, both solvers score similarly to within a 1% difference, meaning that their performance is extremely similar. In terms of average execution time t, however, some larger deviations can be observed, with no solver coming definitely ahead; for instance, QMDP is faster when using SVR, and SARSOP is faster when using ISVR.

We can observe, in both Table 7.1 and by comparing Fig 7.6 to any of the others, that t_c has a strong impact on all metrics. Firstly, the execution time drops very significantly, to roughly 10% of its original value. However, the remaining performance metrics are mostly impacted negatively, as seen in Table 7.1: cumulative reward generally drops to 60% of its original value, as does t_3, with entropy roughly doubling. This can be attributed to the fact that by limiting the re-calculation of the policy, we are effectively limiting the system's ability to integrate new information. However, seeing as the execution time drops very significantly with increases in policy re-calculation, an advantageous trade-off may be found for different device configurations.

Using this information, we performed the trial of Fig. 7.8, where the complete scenario of Section 7.4.1 was tested. We can observe that, while testing with a larger scenario, the techniques performance is largely maintained: its ability to incorporate data is maintained, as observed in Fig. 7.8(b), as is its ability to converge to a high reward. The system's performance in the t_3 metric suffers with the larger state space, which is explained by the fact that the system was only run for 1000 iterations, which did not allow for a complete convergence to the user profile.

We can observe in Fig. 7.9 that, similarly to the results obtained in simulation, the system is able to systematically gain information on the user, and gradually increase its cumulative reward. We can also observe that the system was unable to fully maintain the user in the most beneficial states, due to the larger state space used in these experiments, combined with the lower number of iterations allowed; it was not enough to achieve full convergence of the learning mechanism and thus the results do not fully correlate with the $V(s)$ function. However, these trials demonstrate that the technique is transferable to real scenarios, thus supporting claim 3, and its potential usefulness in long-term scenarios.

Table 7.1: Results obtained with varying solvers, reward types, number of iterations and re-calculation periods. n stands for the number of iterations per trial, t_c stands for the policy re-calculation period, $t_{V(s)}$ stands for the $V(s)$ change period, R_c stands for cumulative reward, t stands for execution time, t_3 stands for the number of iterations spent in the three most valuable states, $H(T)$ is the entropy of the $P(s'|s,a)$ distribution. Each condition (row) was repeated 1000 times. Results are presented in the $\mu \pm \sigma$ format, indicating the average and standard deviation for the 1000 trials that took place for each condition.

Reward	n	t_c	Final R_c	t	t_3	Final $H(T)$
(a) Results with the QMDP Solver						
SVR	100	1	15.164 ± 71.247	1160.456 ± 585.176	$58.814\% \pm 32.752$	2.4 ± 0.459
SVR	100	20	18.016 ± 73.132	105.912 ± 29.719	$51.438\% \pm 31.714$	2.403 ± 0.445
ISVR	100	1	96.448 ± 127.955	1477.554 ± 410.262	$71.069\% \pm 19.727$	0.697 ± 0.477
ISVR	100	20	56.281 ± 111.467	145.815 ± 9.093	$35.009\% \pm 17.185$	1.301 ± 0.377
MSVR	100	1	92.642 ± 70.935	1579.666 ± 364.688	$56.858\% \pm 24.222$	0.439 ± 0.444
MSVR	100	20	53.172 ± 55.49	180.573 ± 29.628	$35.034\% \pm 16.29$	1.282 ± 0.379
(b) Results with the SARSOP Solver						
SVR	100	1	17.054 ± 72.459	929.113 ± 181.598	$59.318\% \pm 32.256$	2.419 ± 0.459
SVR	100	20	14.797 ± 69.72	86.376 ± 12.59	$49.324\% \pm 30.746$	2.379 ± 0.448
ISVR	100	1	100.709 ± 126.883	2246.031 ± 35082.49	$70.589\% \pm 19.313$	0.669 ± 0.467
ISVR	100	20	67.367 ± 109.899	88.36 ± 7.483	$34.561\% \pm 16.506$	1.283 ± 0.366
MSVR	100	1	91.859 ± 66.424	1183.665 ± 84.448	$55.056\% \pm 25.558$	0.425 ± 0.422
MSVR	100	20	55.047 ± 55.485	113.89 ± 47.13	$33.687\% \pm 16.248$	1.288 ± 0.372

7.6 Summary

In this chapter we have presented and experimentally demonstrated αPOMDP, a User-Adaptive Decision-Making framework based on the POMDP formulation. We have performed tests in a simulated benchmark, demonstrating our technique's abilities while operating on several POMDP solvers, and also with human users, demonstrating its ability to produce impact on the user. Our results demonstrate the claims from Section 7.1, showing that the system is able to make positive decisions, maintaining the user in valuable states, and also to explore and learn from the user, both in simulated and real trials.

This Chapter has demonstrated a possible means of using information on the user to tailor a robot's actions to their needs and specificities. Thus, we have closed the loop of Fig 1.2, and now have a full, user-adaptive system. We will now conclude the manuscript with a summary of the work performed, as well as a reflection on possible lines of future work.

Part IV

Conclusion

Chapter 8

Closed Research Gaps, Outcomes and Future Work

"'Mistakes' is the word you're too embarrassed to use. You ought not to be. You're a product of a trillion of them. Evolution forged the entirety of sentient life on this planet using only one tool: the mistake."
— Robert Ford (Anthony Hopkins), "The Original", Westworld

The overarching goal of this work was to study and improve the state of the art on user-adaptive social robots, under the premise that this advancement would aid these systems in becoming more autonomous in their interaction with users. To this end, we have surveyed the state of the art, proposed a general architecture for user-adaptive systems, and uncovered a number of research gaps to be tackled, in Part I. This overall architecture and research gaps have served as motivation for the work developed in Parts II and III, splitting the architecture in two essential parts: user modelling, responsible for gaining information on the user, and decision making, responsible for deciding how to act in the face of this information.

Part II, focusing on user modelling, presented three new techniques. We started with BUM (Chapter 3), our user modelling technique, which is able to integrate and represent a user population, iteratively learning their characteristics and matching new users to existing profiles, all based on heterogeneous data from distributed sources. We have demonstrated BUM's ability to capture, cluster and retrieve the characteristics of a population of users with acceptable performance, allowing us to characterize them.

BUM operates on the assumption that there is an evidence vector extracted from raw data via perception, linking the values in this vector to reality. The Psychbot (Chapter 4) study demonstrates that a social robot can be used to determine the psychological state of a user via verbal interaction, based on the questionnaires used in Psychology, thus providing a means to establish this link. The combination of BUM and Psychbot could be used to inform a robot on the true psychological characteristics of its user.

BUM is a learning technique and, as such, could potentially need a very large amount of examples to achieve its desired performance levels. This issue is tackled by the dataset reduction technique of Chapter 5, which aims to reduce the amount of data necessary for Bayesian techniques such as BUM to learn the representation of the user population. We have shown that, in our case, it can be used to significantly reduce a training dataset for these techniques without compromising the classification abilities of these techniques. Thus, this technique provides yet another optimization that could improve BUM's performance.

Part III, focusing on decision making, presented one preliminary study and a new technique. The technique of Chapter 6 demonstrates that it is possible to use Bayesian programs to enable a social robot to learn the preferences of its users in what regards its actions in context. The αPOMDP technique (Chapter 7) then presents a POMDP-based technique, which features a refinement to the previous technique, including the introduction of a novel rewarding mechanism and learning execution loop. By using BUM to determine the current state of the user as the necessary vector of discrete variables, αPOMDP can be used to close the loop, and achieve the autonomous user-adaptive functionality described in Chapter 1.

Thus, we have contributed to the state of the art in the four main areas described in Chapter 1: perception, user modelling, learning and decision making. We have also contributed to answering our research question, providing partial answers in each sub-field, which can now be combined to provide a unified possible answer.

The remainder of this chapter focuses on reflecting upon the achievements of this work, demonstrating in detail how it closes or tackles several of the research gaps uncovered, and proposing a number of future research lines. It is structured as follows:

- Section 8.1 details which of the research gaps of Chapter 2 were tackled, and how;
- Section 8.2 summarizes the main outcomes of this work, including publications and open-source software;
- Section 8.3 presents a a number of open lines of future work.

8.1 Closed Research Gaps

Chapter 2 has presented the state of the art in user-adaptive robots. It has culminated in Tables 2.4 and 2.5, which presents the main research gaps found during the survey. Of these, a subset were tackled during this work, advancing scientific knowledge in their sub-domains. In this section we re-discuss each of these research gaps, and demonstrate how we have advanced the state of the art in what concerns them.

Research Gap 1: Learning New Users Our survey has determined that, while user-adaptive robots do exist, they do not seem to employ techniques to improve the process of learning new users. Indeed, the techniques under survey were generally unable to exploit similarities between users to achieve higher levels of adaptation quicker, instead relying on learning each individual user from the start.

BUM (Chapter 3) suggests an approach to this research gap. Through the clustering and matching algorithm of Section 3.4, BUM is able not only to deal with fault cases, but also to efficiently estimate the characteristics of new users. This can be achieved through the exploitation of the matching mechanism described: once a large enough population of users is known, each new user is likely to share some similarity with previous users. When that is the case, the distance measurement D can be used to match the user to previous clusters, thus estimating any missing long-term characteristics, which can later be refined through interaction. Thus, the application of BUM carries the potential to lessen the impact of new users on the system's actions, as seen in for instance Fig. 3.10. Furthermore, the dataset reduction technique presented in Chapter 5 has the potential to further aid in this process, by reducing the amount of data needed on each new user to successfully learn them.

Research Gap 2: Interaction with Multiple Users None of the systems surveyed are able to interact and model multiple users, and are seemingly limited to one-on-one interaction. Despite not resulting in a one-to-many interaction mode, BUM can partially close this gap through its ability to model multiple users. Indeed, the model created by BUM contains, in the same characteristics space, all of the users of the system, which then allows it to perform the aforementioned clustering and matching operations. Thus, BUM can be coupled with a sophisticated decision-making system to interact with, model and adapt to multiple users. Despite not being tested, βPOMDP, when combined with BUM, could also constitute a step forward in this regard, enabling multiple robots to model and accommodate multiple users.

Research Gap 3: Psychological Trait Modelling In our survey of the state of the art we noticed that some systems, such as [143], tended to make use of psychological information on the user, such as their personality traits. In our view, this could lead to higher levels of adaptation to the user, and potentially new levels of user satisfaction, acceptance, trust, *etc.*

In the Psychbot experiment of Chapter 4, we have established that it is possible for a social robot to faithfully and accurately determine the psychological status of its user, including both long-term personality traits and short-term constructs such as mood. This could allow the social robots of the future to make use of deeper knowledge on the user to achieve more successful interactions. Furthermore, the technique described paves the way for a deeper integration of Active Learning [23] techniques, wherein the robot can use the user themselves as the oracle whilst learning about their psychological characteristics.

Research Gap 4: Big Data Big Data is a recent trend in Science, and has seen some use in Robotics of late in techniques such as OpenPose [24]. However, we could find no works that tackled this issue directly in the realm of user-adaptive robots.

The dataset reduction technique of Chapter 5 constitutes a preliminary approach for this problem, being able to reduce datasets used by Bayesian by close to 90%. Thus, it can allow for the reduction of datasets used to train Bayesian systems, such as BUM, and potentially ease the problem of Big Data.

Research Gap 5: Continuous Adaptation The problem of long-term, dynamic adaptation to the user remains, as seen in Chapter 2, largely unsolved. Although some of the surveyed techniques are indeed able to adapt to the user even when given no knowledge about them, they are generally unable to accompany the user's change as they evolve.

We propose that this gap can be tackled by the combination of BUM and αPOMDP, since both of these techniques are able to adapt to the user as they change. Indeed, the Bayesian likelihood constructed by BUM always retains a level of uncertainty as to the user's characteristics, which allows it to "revert" an estimation if contrary evidence is received consistently. Similarly, the T matrix constructed by αPOMDP always retains a level of uncertainty, as demonstrated for instance by the entropy levels of Fig. 7.5. Thus, by receiving consistent contrary evidence, *i.e.* if suddenly a certain action in a certain state causes the user to transition to different states than it did before, the entropy of the T matrix will increase. This will lead the system to deeply explore this new source of information (if it is operating on ISVR) in order to establish the true new impact of its action.

Thus, the combination of these approaches can be used to build a system that is truly able to accompany the user in their evolution through life, and achieve continuous adaptation to a changing user.

Technological Gap 9: User Skill Level In Chapter 2, we found that a significant number of works did not test their approaches on their end-users, instead relying on "corridor sampling" or proxy user samples to demonstrate their principles. This technological gap was not tackled in this work *per se*, but was taken as a strong guideline in the design of experiments. Namely, when designing and performing the Psychbot study of Chapter 4, all users included in both phases of testing were elderly users, the exact potential end-users of the system.

Technological Gap 12: Metrics and Standardization The survey of Chapter 2 found a lack of a unified way to measure the impact of the interaction on the user, as well as well-established and proven ways for the robot itself to evaluate the status of its interaction with the user. We aimed to take another step in this direction with the Psychbot study of Chapter 4. In fact, the results obtained demonstrate that the robot can make use of the modified questionnaire items therein to evaluate the state of the user accurately and, thus, better gauge its own impact on them. Thus, the results of this study could provide an opportunity for the development of standardized ways for robots to autonomously evaluate their impact on their users, as well as the status of the interaction, through the usage of standardized testing techniques applied in the field of Psychology.

8.2 Outcomes

As desired, the outcomes of this work comprise both scientific results, under the form of research papers, and also engineering works, composed mainly of open-source software packages. For each of the research sub-areas contemplated in this work, we have proposed and validated a novel approach and have disseminated it as both scientific literature and open-source software. In summary, this work resulted in around 12 scientific publications, of which 3 were published in journals, and in 4 open-source software packages. The main outcomes of the work, both technological and scientific, are highlighted in Table 8.1.

Table 8.1: Summary of the research gaps closed and of the main outcomes of this work.

Research Gaps	Software Packages	Main Publications
(a) Outcomes of the SoA Survey		
n/a	n/a	• Gonçalo S. Martins, Luís Santos, and Jorge Dias. User-Adaptive Interaction in Social Robots: A Survey Focusing on Non-Physical Interaction. *International Journal of Social Robotics*, 2018. ISSN 1875-4805. doi: 10.1007/s12369-018-0485-4. URL `https://rdcu.be/Y9JJ`
(b) Outcomes of the work on User Modelling		
1, 2, 3, 5, 4	`bum`[1], `psychbot`[2], `surprisal_experiments`[3]	• Gonçalo S. Martins, Luís Santos, and Jorge Dias. BUM: Bayesian User Model for Distributed Social Robots. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*. IEEE, 2017. doi: 10.1109/ROMAN.2017.8172469 • Gonçalo S. Martins, Luís Santos, and Jorge Dias. bum_ros: Distributed User Modelling for Social Robots using ROS. In *Robot Operating System: The Complete Reference (Volume 3)*, pages 531–567. Springer, 2018. ISBN 978-3-319-91590-6. doi: 10.1007/978-3-319-91590-6_15 • João Quintas, Gonçalo S Martins, Luis Santos, Paulo Menezes, and Jorge Dias. Toward a Context-Aware Human-Robot Interaction Framework Based on Cognitive Development. *IEEE Transactions On Systems, Man, And Cybernetics*, pages 1–11, 2018. doi: 10.1109/TSMC.2018.2833384 • Additional journal paper entitled "BUM: Bayesian User Model for Distributed Learning of User Characteristics from Heterogeneous Information" submitted to IEEE Transactions on Cognitive and Developmental Systems (under minor revision). • Additional journal paper on the Psychbot study to be submitted to the International Journal of Social Robotics.
(d) Outcomes of the work on Adaptive Behaviour		
5	`apomdp`[4]	• Gonçalo S. Martins, Hend Al Tair, Luís Santos, and Jorge Dias. αPOMDP: POMDP-based user-adaptive decision-making for social robots. *Pattern Recognition Letters*, 0: 1–10, 2018. ISSN 01678655. doi: 10.1016/j.patrec.2018.03.011. URL `https://www.sciencedirect.com/science/article/pii/S0167865518300825` • Hend Al Tair, Gonçalo S. Martins, Luís Santos, and Jorge Dias. αPOMDP: State-Based Decision Making for Personalized Assistive Robots. In *Thirty-Second AAAI Conference on Artificial Intelligence, Workshop 3: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2018 • Additional journal paper on βPOMDP under development.

8.3 Future Work

Integration and Long-Term Testing This work has presented a number of components which can be integrated into a single, user-adaptive system for controlling a social robot. By gradually learning and adapting to the user, this integrated system could improve a social robot's abilities to interact with users in the long-term.

Thus, the most significant portion of future work to be performed in the context is twofold: to materialize the unified system into an integrated platform, and to test it in long-term experiments with end-users. Despite the mathematical compatibilities of the individual methods (*i.e.* the dataset reduction technique operates on supervised data, which can be fed to BUM, which produces an estimate of the user's state for αPOMDP), the integration of these systems would be a task of important technical difficulty.

Long-term testing could then provide a better answer to the question of Chapter 1, determining objectively how the user perceives a user-adaptive robot when compared to a non-adaptive one. Standardized scales, such as the System Usability Scale [21] or the Almere Model [157], can be used to evaluate these tests together with the data gathered by the robot itself, allowing us to gain further insight into the impact of this system on the user's daily life.

Personality Sensors One of the primordial ideas that led to the development of the BUM system (Chapter 3) and Psychbot study (Chapter 4) was the development of *personality sensors*: virtual devices which, individually, were able to determine partial estimates of the user's personality profile, which would then be fused into a unified, more accurate, representation of the user. This initial idea was split twofold, resulting in the distributed sensing and fusion mechanism of the BUM approach, and the attempt to link signals and psychological constructs of the Psychbot study.

It would now be interesting to return to the original idea: to implement isolated software and hardware modules able to provided partial, empirically-validated views of the user's psychological state, which could then be fused to obtain an accurate representation of the user's psyche. This information can then be used not only to further adapt automated systems to the user, but also to pre-diagnose potential psychological conditions. Due to the constant monitoring that can be achieved by automated approaches, this could lead to the extremely precocious detection of certain signs of disease, which could potentially greatly improve the user's chances of successful treatment.

Person Identification Through User Profile Given BUM's ability to build user profiles from distributed data, it would be interesting to explore its ability to identify an individual user from the profile gathered during interaction. This line of research would be interesting to explore not only as an exploitation of BUM's discriminative abilities, but also as a solution to a specific use case, wherein an autonomous system, such as a simplistic embedded sensor, is unable to identify a user clearly but is able to infer one or more of their characteristics. For instance, a network of PIR sensors in a home could not, trivially, identify its user. However, by employing BUM, it can be used to extract their characteristics, which could in turn be used to identify the user themselves.

Integration of Dataset Reduction in User Modelling The surprisal-based dataset reduction technique presented in Chapter 5 can be used to reduce large datasets by exploiting their redundancy. However, we have not applied it in BUM as a means of, somehow, discriminating which data should or should not be allowed to influence the global population model, despite BUM employing entropy as a weight in its fusion step. Thus, an interesting line of future work would be to integrate these two techniques, and to make use of surprisal to properly gauge the value of information being added to the model, and perhaps improve its performance.

Adaptive Psychological Evaluation The Psychbot study of Chapter 4 demonstrated that a social robot is able to gather the psychological characteristics of a user, with some degree of accuracy, via natural voice-based interaction with the user. However, the scales were presented to the user in a random way: each question asked was selected randomly from those remaining.

It would be interesting to study the possibility of building the user model, using BUM, as the questions were asked. Thus, a level of certainty, via entropy, would be available with respect to each of the psychological characteristics under evaluation. This information could be used to inform the item selection procedure, which would focus on gaining as much information with as fewer questions as possible. This process could further be informed by the factor analyses already performed in the field of Psychology, which determined which items, within each construct, are the most predictive of the final result of the evaluation. Thus, we could potentially shorten the interaction time needed to determine the user's psychological state, potentially shortening it to only a handful of questions, which would be a vast improvement over the current 55-question protocol and would also contribute to research gap 7.

Autonomous, User-Adaptive Psychological Evaluation through Active Learning Voice-based interaction is the social robot's preferred modality when it needs concrete, unambiguous information from the user. However, other perception techniques can and should be used, *i.e.* the user should not have to verbally inform the robot of all of the information the robot is looking to gather. As such, it would be interesting for the system to be able to infer the user's model, including psychological characteristics, based on non-verbal, non-intrusive stimulation and observation.

Empowered with the questions that, as we demonstrated, lead to positive results with respect to the determination of the user's psychological characteristics, a social robot could now autonomously learn how to determine their user's psychological state from indirect interaction. In fact, these questions would allow the system to perform an Active Learning process, formulating its own hypotheses as to the user's state and, when deemed necessary, employing the scale questions to obtain ground truth as to the user's state. Thus, the robot could autonomously develop user-specific models able to correlate the user's behaviour, other perceived characteristics, features such as facial expressions, prosody, *etc*, with their current state of mind. After an initial learning period, the robot would be able to predict its user's psychological state from its own sensor readings, and thus adapt seamlessly to their current state of mind without the need for verbal interaction.

Furthermore, assuming that there is a large number of robots interacting with a large number of users, these individual models could gradually be generalized. This would lead to new perception models able to generically (with the natural loss in accuracy) predict the psychological states of users based on their behaviour, which would be applicable to any user in any situation. This advancement could, naturally, grow beyond the scope of Social Robotics, and be employed in other fields such as automated video surveillance and clinical Psychology itself.

αPOMDP, βPOMDP and Domain Independence The αPOMDP formalism of Chapter 7 was only applied, and extended in βPOMDP, to the field of Social Robotics. However, like most automated planning techniques [53], αPOMDP is domain-independent, and could be applied to any domain given that its basic requirements are met, namely that the state space of the new application can be reduced to a vector of discrete variables. Thus, it would be interesting to study the behaviour and potential of these techniques when applied to different action domains that require operation in partially or completely unknown environments, such as search-and-rescue or agricultural robotics.

Robot-Robot Interaction It is common, in the HRI community, to employ human users as testers for new HRI systems, as seen in throughout the Thesis. In turn, when multiple robots are employed, they communicate through specialized channels, such as the cloud mechanism used by BUM (Chapter 3). This generates a discrepancy in communication channels: robots can understand each other and humans, but humans can only understand each other, and cannot understand (or even perceive) the interaction that is going on between robots. Steps are being taken in the field of explainable AI to bridge this gap, with a few successful techniques having already been developed.

Thus, a relevant line of future work would be to extend the adaptive techniques presented in this Thesis to decide when to communicate among robots via hidden channels, and when to communicate via human channels, and what level of abstraction to employ. Thus, in a social setting, multiple robot and humans could share the same social space, and understand each other's intentions seamlessly.

Bibliography

[1] Ahmad Abdel-Hafez and Yue Xu. A Survey of User Modelling in Social Media Websites. *Computer and Information Science*, 6(4):59–71, 2013. ISSN 1913-8997. doi: 10.5539/cis.v6n4p59.

[2] Nichola Abdo, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robot, organize my shelves! Tidying up objects by predicting user preferences. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1557–1564, 2015. doi: 10.1109/ICRA.2015.7139396.

[3] Hend Al Tair, Gonçalo S. Martins, Luís Santos, and Jorge Dias. αPOMDP: State-Based Decision Making for Personalized Assistive Robots. In *Thirty-Second AAAI Conference on Artificial Intelligence, Workshop 3: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2018.

[4] Aldebaran. Pepper, the humanoid robot from Aldebaran, a genuine companion. URL `https://www.ald.softbankrobotics.com/en/cool-robots/pepper`.

[5] Pedro Almiro, Daniela Almeida, Ana Margarida Ferraz, Rute Ferreira, Maria João Silvestre, Carlos Perdiz, Isabel Dias, Sonia Goncalves, Liliana Sousa, and Mário Simões. Escala de Desejabilidade Social de 20 itens (EDS-20). In Mário R. Simões, Leandro S. Almeida, and Miguel M. Gonçalves, editors, *Psicologia forense: Instrumentos de avaliação*, chapter 20, pages 335–352. Pactor, Lisboa, 2017. ISBN 978-989-693-076-9.

[6] Paulo Alvito, Carlos Marques, Paulo Carriço, and João Freire. A Robotic Platform for the Social Robot Project. In *Proc. of the 23rd IEEE International Symposium on Robot And Human Interactive Communication (RO-MAN 2014), Workshop on Interactive Robots for aging and/or impaired people, Edinburgh*, 2014.

[7] Amir Aly. *Towards an Interactive Human-Robot Relationship: Developing a Customized Robot's Behavior to Human's Profile*. Phd thesis, ENSTA ParisTech, 2014. URL `https://pastel.archives-ouvertes.fr/tel-01128923/`.

[8] Amir Aly and Adriana Tapus. A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 325–332, 2013. ISSN 21672148. doi: 10.1109/HRI.2013.6483606.

[9] Farshid Amirabdollahian, Rieks Op Den Akker, Sandra Bedaf, Richard Bormann, Heather Draper, Vanessa Evers, Gert Jan Gelderblom, Carolina Gutierrez Ruiz, David Hewson, Ninghang Hu, Iolanda Iacono, Kheng Lee Koay, Ben Krose, Patrizia Marti, Herve Michel, Helene Prevot-Huille, Ulrich Reiser, Joe Saunders, Tom Sorell, and Kerstin Dautenhahn. Accompany: Acceptable robotiCs COMPanions for AgeiNG Years - Multidimensional aspects of human-system interactions. *2013 6th International Conference on Human System Interactions, HSI 2013*, pages 570–577, 2013. ISSN 2158-2246. doi: 10.1109/HSI.2013.6577882.

[10] Nahla Ben Amor, Salem Benferhat, and Zied Elouedi. Naive bayes vs decision trees in intrusion detection systems. *ACM Symposium on Applied Computing*, pages 420–424, 2004. doi: 10.1145/967900.967989. URL `http://dl.acm.org/citation.cfm?id=967989`.

[11] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A POMDP extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems*, pages 64–72, 2010.

[12] Ruth Aylett, Arvid Kappas, Ginevra Castellano, Susan Bull, Wolmet Barendregt, Ana Paiva, and Lynne Hall. I know how that feels — An empathic robot tutor. *eChallenges e-2015 Conference*, pages 1–9, 2015. doi: 10.1109/eCHALLENGES.2015.7441088. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7441088`.

[13] Kim Baraka and Manuela Veloso. Adaptive Interaction of Persistent Robots to User Temporal Preferences. In *International Conference on Social Robotics*, pages 61–71. Springer International Publishing, 2015. ISBN 978-3-642-25503-8. doi: 10.1007/978-3-642-25504-5. URL `http://link.springer.com/10.1007/978-3-319-02675-6`.

[14] Kim Baraka, Ana Paiva, and Manuela Veloso. Expressive lights for revealing mobile service robot state. *Advances in Intelligent Systems and Computing*, 417:107–119, 2016. ISSN 21945357. doi: 10.1007/978-3-319-27146-0_9.

[15] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012. ISBN 9780521518147.

[16] Pierre Bessière, Emmanuel Mazer, Juan-Manuel Ahuactzin, and Kamel Mekhnacha. *Bayesian Programming*. 2014. ISBN 9781439880333.

[17] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

[18] Marisa F. Boston, John T. Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2(1):1–12, 2008. ISSN 1995-8692. doi: 10.16910/jemr.2.1.1.

[19] Craig Boutilier and David Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1168–1175, 1996.

[20] Philemon Brakel. *Testing surprisal theory using connectionist networks and dimensionality reduction methods*. Msc thesis, University of Amsterdam, 2009.

[21] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189 (194):4–7, 1996. ISSN 1097-0193. doi: 10.1002/hbm.20701. URL `http://hell.meiert.org/core/pdf/sus.pdf`.

[22] Frank Broz, Illah Nourbakhsh, and Reid Simmons. Planning for Human-Robot Interaction in Socially Situated Tasks: The Impact of Representing Time and Intention. *International Journal of Social Robotics*, 5(2):193–214, 2013. ISSN 18754791. doi: 10.1007/s12369-013-0185-z.

[23] Burr Settles. *Active Learning*. Morgan \& Claypool Publishers, 2012. ISBN 9781608457250. URL `https://dl.acm.org/citation.cfm?id=3019233`.

[24] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.143.

[25] Francesca Carmagnola, Federica Cena, and Cristina Gena. User model interoperability: A survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331, 2011. ISSN 09241868. doi: 10.1007/s11257-011-9097-5.

[26] Anthony R. Cassandra and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9-12 1995*, page 362. Morgan Kaufmann, 2016.

[27] Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. Acting optimally in partially observable stochastic domains. In *AAAI*, volume 94, pages 1023–1028, 1994.

[28] Anthony R. Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 54–61. Morgan Kaufmann Publishers Inc., 1997.

[29] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):1–58, 2009. ISSN 03600300. doi: 10.1145/1541880.1541882.

[30] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. ISSN 00457906. doi: 10.1016/j.compeleceng.2013.11.024. URL `http://www.sciencedirect.com/science/article/pii/S0045790613003066`.

[31] Hsin Han Chiang, Yen Lin Chen, and Chun Tse Lin. Human-robot interactive assistance of a robotic walking support system in a home environment. *Proceedings of the International Symposium on Consumer Electronics, ISCE*, pages 263–264, 2013. doi: 10.1109/ISCE.2013.6570218.

[32] Jens K Christensen, Kasper Lamberth, Morten Nielsen, Claus Lundegaard, Peder Worning, Sanne L Lauemoller, Soren Buus, Soren Brunak, and Ole Lund. Selecting informative data for developing peptide-MHC binding predictors using a query by committee approach. *Neural Comput.*, 15(12):2931–2942, 2003. ISSN 08997667. doi: 10.1162/089976603322518803.

[33] John R. Crawford and Julie D. Henry. The Positive and Negative Affect Schedule (PANAS): Construct validity, measurement properties and normative data in a large non-clinical sample. *British Journal of Clinical Psychology*, 43(3):245–265, 2004. ISSN 01446657. doi: 10.1348/0144665031752934.

[34] Nigel Cross. *Computers and people*, volume 8. 1987. ISBN 0121377555. doi: 10.1016/0142-694X(87)90013-5.

[35] Ayse Cufoglu. User Profiling-A Short Review. *International Journal of Computer Applications*, 108(3):1–9, 2014.

[36] John P. Cunningham and Zoubin Ghahramani. Linear Dimensionality Reduction: Survey, Insights, and Generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015. ISSN 15337928.

[37] Maartje Margaretha Allegonda de Graaf. *Living with Robots: Investigating the User Acceptance of Social Robots in Domestic Environments*. Phd thesis, University of Twente, 2015.

[38] Henry Detmold, Anton Van Den Hengel, Anthony Dick, Christopher Madden, Alex Cichowski, and Rhys Hill. Surprisal-aware scheduling of PTZ cameras. *2009 3rd ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2009*, 2009. doi: 10.1109/ICDSC.2009.5289368.

[39] Sandra Devin and Rachid Alami. An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution. *International Conference on Human-Robot Interaction*, pages 319–326, 2016. ISSN 21672148. doi: 10.1109/HRI.2016.7451768.

[40] Shirley Dowdy, Stanley Weardon, and Daniel Chilko. *Statistics for Research.* 2005. ISBN 047126735X (print)\n0471477435 (online). doi: 10.1198/tech.2005.s832.

[41] Ismael Duque, Kerstin Dautenhahn, Kheng Lee Koay, Lan Willcock, and Bruce Christianson. A different approach of using Personas in human-robot interaction: Integrating Personas as computational models to modify robot companions' behaviour. In *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pages 424–429, 2013. ISBN 9781479905072. doi: 10.1109/ROMAN.2013.6628516.

[42] Martina Eckert, Marcos López, Carlos Lázaro, Juan Meneses, and José F Martínez Ortega. MoKey – A Motion based Keyboard Interpreter. pages 4–5, 2015. doi: 10.1109/ISCE.2015.7177799.

[43] Eleni Efthimiou, Stavroula-evita Fotinea, Theodore Goulas, Athanasia-lida Dimou, Maria Koutsombogera, Vassilis Pitsikalis, Petros Maragos, and Costas Tzafestas. The MOBOT Platform – Showcasing Multimodality in Human-Assistive Robot Interaction. 1:382–391. doi: 10.1007/978-3-319-40244-4.

[44] European Commission. Technology Readiness Levels (TRL). *HORIZON 2020 – WORK PROGRAMME 2014-2015 General Annexes, Extract from Part 19 - Commission Decision C*, (2014):4995, 2015.

[45] Hans J. Eysenck. Dimensions of personality: 16, 5 or 3?-Criteria for a taxonomic paradigm. *Personality and Individual Differences*, 12(8):773–790, 1991. ISSN 01918869. doi: 10.1016/0191-8869(91)90144-Z.

[46] João Filipe Ferreira and Jorge Dias. *Probabilistic Approaches for Robotic Perception*. Springer International Publishing, 2014. ISBN 9783319020051. doi: 10.1007/978-3-319-02006-8. URL `https://www.springer.com/us/book/9783319020051`.

[47] Josef Fink and Alfred Kobsa. A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. *User Modeling and User-Adapted Interaction*, 10:209–249, 2000. ISSN 0924-1868. doi: 10.1023/A:1026597308943.

[48] Michelangelo Fiore and Harmish Khambhaita. An Adaptive and Proactive Human-Aware Robot Guide. 2015. doi: 10.1007/978-3-642-25504-5. URL `http://link.springer.com/10.1007/978-3-319-02675-6`.

[49] David Fischinger, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann, Tobias Koertner, Astrid Weiss, Antonis Argyros, and Markus Vincze. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, 75:60–78, 2014. ISSN 09218890. doi: 10.1016/j.robot.2014.09.029. URL `http://linkinghub.elsevier.com/retrieve/pii/S0921889014002140`.

[50] Robert Fisher. CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision. URL `http://homepages.inf.ed.ac.uk/rbf/CVonline/CVentry.htm`.

[51] Yixing Gao, Hyung Jin Chang, and Yiannis Demiris. User Modelling for Personalised Dressing Assistance by Humanoid Robots. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1840–1845, 2015. ISSN 21530866. doi: 10.1109/IROS.2015.7353617.

[52] Malik Ghallab, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL - The Planning Domain Definition Language. 1997. doi: 10.1.1.51.9941.

[53] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. ISBN 9781107037274.

[54] L R Goldberg. An alternative "description of personality": the Big-Five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216–1229, 1990. ISSN 0022-3514. doi: 10.1037/0022-3514.59.6.1216.

[55] G Gordon, S Spaulding, Jacqueline Kory Westlund, J J Lee, L Plummer, M Martinez, M Das, and C Breazeal. Affective personalization of a social robot tutor for children's second language skills. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, (2011):3951–3957, 2016.

[56] Goren Gordon and Cynthia Breazeal. Bayesian Active Learning-based Robot Tutor for Children ' s Word-Reading Skills. *Aaai*, pages 1343–1349, 2015.

[57] Jasmin Grosinger, Federico Pecora, and Alessandro Saffiotti. Making Robots Proactive through Equilibrium Maintenance. In *IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3375–3381, 2016.

[58] John Hale. A probabilistic earley parser as a psycholinguistic model. *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies (NAACL)*, 2:159–166, 2001. doi: 10.3115/1073336.1073357.

[59] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933. ISSN 0022-0663. doi: 10.1037/h0071325.

[60] Chien-ming Huang and Bilge Mutlu. Anticipatory Robot Control for Efficient Human-Robot Collaboration. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016. ISBN 9781467383707. doi: 10.1109/HRI.2016.7451737.

[61] S. Impedovo, L. Otaviano, and S. Occhinegro. Optical Character Recognition: A Survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1), 1991. doi: 10.1142/S0218001491000041.

[62] Blue Frog Robotics Inc. Buddy: Your Companion Robot, 2016. URL `http://www.bluefrogrobotics.com/en/buddy-your-companion-robot/`.

[63] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998.

[64] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro. Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. *Human-computer Interaction*, 19:61–84, 2004. ISSN 0737-0024. doi: 10.1207/s15327051hci1901&2_4.

[65] Takayuki Kanda, Rumi Sato, Naoki Saiwaki, and Hiroshi Ishiguro. A Two-Month Field Trial in an Elementary School for Long-Term Human-Robot Interaction. *IEEE Transactions on Robotics*, 23(5):962–971, 2007. ISSN 1552-3098. doi: 10.1109/TRO.2007.904904.

[66] Abir B. Karami and Abdel-illah Mouaddib. A Decision Model of Adaptive Interaction Selection for a Robot Companion. *proceedings of the 5th European Conference on Mobile Robots, ECMR'11*, pages 83–88, 2011.

[67] Abir B. Karami, Karim Sehaba, and Benoit Encelle. Adaptive and personalised robots - Learning from users' feedback. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, pages 626–632, 2013. ISSN 10823409. doi: 10.1109/ICTAI.2013.98. URL `https://www.engineeringvillage.com/blog/document.url?mid=cpx{_}M34eddd49145902ed14eM529d10178163125{&}database=cpx`.

[68] Abir B. Karami, Karim Sehaba, and Benoit Encelle. Towards adaptive robots based on interaction traces: A user study. *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–6, 2013. doi: 10.1109/ICAR.2013.6766484. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6766484`.

[69] Abir B. Karami, Karim Sehaba, and Benoit Encelle. Learn to adapt based on users' feedback. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2014. ISBN 978-1-4799-6765-0. doi: 10.1109/ROMAN.2014.6926322.

[70] Abir B. Karami, Karim Sehaba, and Benoit Encelle. Adaptive artificial companions learning from users feedback. *Adaptive Behavior*, 24(2):69–86, 2016. ISSN 1059-7123. doi: 10.1177/1059712316634062. URL `http://adb.sagepub.com/cgi/doi/10.1177/1059712316634062`.

[71] Han Gyeol Kim, Jeong Yean Yang, and Dong Soo Kwon. Experience based domestic environment and user adaptive cleaning algorithm of a robot cleaner. *11th International Conference on Ubiquitous Robots and Ambient Intelligence, (URAI)*, pages 176–178, 2015. doi: 10.1109/URAI.2014.7057525.

[72] Jieun Kim, Ahreum Lee, and Hokyoung Ryu. Personality and its effects on learning performance: Design guidelines for an adaptive e-learning system based on a user model. *International Journal of Industrial Ergonomics*, 43(5):450–461, 2013. ISSN 01698141. doi: 10.1016/j.ergon.2013.03.001. URL `http://dx.doi.org/10.1016/j.ergon.2013.03.001`.

[73] Steven D. Klee, Beatriz Quintino Ferreira, Rui Silva, Paulo Costeira, Francisco S. Melo, and Manuela Veloso. Personalized Assistance for Dressing Users. In *7th International Conference on Social Robotics (ICSR)*. Springer International Publishing, 2015. ISBN 978-3-642-25503-8. doi: 10.1007/978-3-642-25504-5. URL `http://link.springer.com/10.1007/978-3-319-02675-6`.

[74] Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2):49–63, 2001. ISSN 09241868. doi: 10.1023/A:1011187500863.

[75] Joseph A. Konstan and John Riedl. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, 2012. ISSN 09241868. doi: 10.1007/s11257-011-9112-x.

[76] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland., 2008.

[77] Kleanthi Lakiotaki, Nikolaos F. Matsatsinis, and Alexis Tsoukiàs. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76, 2011. ISSN 15411672. doi: 10.1109/MIS.2011.33.

[78] Chi-pang Lam, Allen Y Yang, Katherine Driggs-campbell, Ruzena Bajcsy, and S Shankar Sastry. Improving Human-In-The-Loop Decision Making In Multi-Mode Driver Assistance Systems Using Hidden Mode Stochastic Hybrid Systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. doi: 10.1109/IROS.2015.7354197.

[79] Yanan Li, Keng Peng Tee, Rui Yan, Wei Liang Chan, Yan Wu, and Dilip Kumar Limbu. Adaptive Optimal Control for Coordination in Physical Human-Robot Interaction. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015. ISBN 9781479999934.

[80] Gi Hyun Lim, Seung Woo Hong, Inhee Lee, Il Hong Suh, and Michael Beetz. Robot recommender system using affection-based episode ontology for personalization. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pages 155–160, 2013. doi: 10.1109/ROMAN.2013.6628437.

[81] Margarida P Lima and António Simões. O Inventário da Personalidade NEO-PI-R: Resultados da aferição portuguesa. *Psychologica*, 18:24–46, 1997.

[82] Nichola Lubold, Erin Walker, and Heather Pon-Barry. Effects of Voice-Adaptation and Social Dialogue on Perceptions of a Robotic Learning Companion. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 255–262, 2016. ISBN 9781467383707. doi: 10.1109/HRI.2016.7451760.

[83] Daniel Le Ly and Hod Lipson. Optimal experiment design for coevolutionary active learning. *IEEE Transactions on Evolutionary Computation*, 18(3):394–404, 2014. ISSN 1089778X. doi: 10.1109/TEVC.2013.2281529.

[84] A. Madureira, B. Cunha, J. P. Pereira, S. Gomes, I. Pereira, J. M. Santos, and A. Abraham. Using Personas for Supporting User Modeling on Scheduling Systems. In *14th International Conference on Hybrid Intelligent Systems (HIS)*, pages 279–284. IEEE, 2014. ISBN 9781479976331.

[85] John C Mankins. Technology Readiness Levels. *White Paper April*, 6(2):5, 1995. ISSN 08956308. doi: 10.1080/08956308.2010.11657640.

[86] Gonçalo S. Martins, Luís Santos, and Jorge Dias. The GrowMeUp Project and the Applicability of Action Recognition Techniques. In Jorge Dias, Francisco Escolano, George Ezzopardi, and Rebbeca Marfil, editors, *Third Workshop on Recognition and Action for Scene Understanding (REACTS)*. Ruiz de Aloza, 2015.

[87] Gonçalo S. Martins, Paulo Ferreira, Luís Santos, and Jorge Dias. A Context-Aware Adaptability Model for Service Robots. In *IJCAI-2016 Workshop on Autonomous Mobile Service Robots*, New York, 2016.

[88] Gonçalo S. Martins, Luís Santos, and Jorge Dias. BUM: Bayesian User Model for Distributed Social Robots. In *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*. IEEE, 2017. doi: 10.1109/ROMAN.2017.8172469.

[89] Gonçalo S. Martins, Hend Al Tair, Luís Santos, and Jorge Dias. αPOMDP: POMDP-based user-adaptive decision-making for social robots. *Pattern Recognition Letters*, 0: 1–10, 2018. ISSN 01678655. doi: 10.1016/j.patrec.2018.03.011. URL `https://www.sciencedirect.com/science/article/pii/S0167865518300825`.

[90] Gonçalo S. Martins, Luís Santos, and Jorge Dias. User-Adaptive Interaction in Social Robots: A Survey Focusing on Non-Physical Interaction. *International Journal of Social Robotics*, 2018. ISSN 1875-4805. doi: 10.1007/s12369-018-0485-4. URL `https://rdcu.be/Y9JJ`.

[91] Gonçalo S. Martins, Luís Santos, and Jorge Dias. bum_ros: Distributed User Modelling for Social Robots using ROS. In *Robot Operating System: The Complete Reference (Volume 3)*, pages 531–567. Springer, 2018. ISBN 978-3-319-91590-6. doi: 10.1007/978-3-319-91590-6_15.

[92] Takamitsu Matsubara, Jaime Valls Miro, Daisuke Tanaka, James Poon, and Kenji Sugimoto. Sequential Intention Estimation of a Mobility Aid User for Intelligent Navigational Assistance. In *24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 444–449. IEEE, 2015. doi: 10.1109/ROMAN.2015.7333580.

[93] Gerald Matthews, Ian J. Deary, and Martha C. Whiteman. *Personality Traits*. Cambridge University Press, 2003. ISBN 9780521887786.

[94] Nikolaos Mavridis. A review of verbal and non-verbal human-robot interactive communication. *Robotics and Autonomous Systems*, 63(P1):22–35, 2015. ISSN 09218890. doi: 10.1016/j.robot.2014.09.031. URL `http://dx.doi.org/10.1016/j.robot.2014.09.031`.

[95] Robert R. McCrae and Paul T. Costa. Reinterpreting the Myers-Briggs Type Indicator from the perspective of the five-factor model of personality. *Journal of Personality*, 57 (1):17–40, 1989. ISSN 0022-3506. doi: 10.1111/j.1467-6494.1989.tb00759.x.

[96] Robert R. McCrae and Paul T. Costa. A contemplated revision of the NEO Five-Factor Inventory. *Personality and Individual Differences*, 36(3):587–596, 2004. ISSN 01918869. doi: 10.1016/S0191-8869(03)00118-1.

[97] Michael F. McTear. User modelling for adaptive computer systems: a survey of recent developments. *Artificial Intelligence Review*, 7(3-4):157–184, 1993. ISSN 02692821. doi: 10.1007/BF00849553.

[98] Ross Mead and Maja Matarić. Toward Robot Adaptation of Human Speech and Gesture Parameters in a Unified Framework of Proxemics and Multimodal Communication. In *2015 IEEE International Conference on Robotics and Automation Workshop on Machine Learning for Social Robots*, 2015. ISBN 978-4-254-10692-3 C3340.

[99] George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. ISSN 00010782. doi: 10.1145/219717.219748.

[100] Raúl Miñón, Lourdes Moreno, Paloma Martínez, and Julio Abascal. An approach to the integration of accessibility requirements into a user interface development method. *Science of Computer Programming*, 86:58–73, 2014. ISSN 01676423. doi: 10.1016/j.scico.2013.04.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0167642313001032.

[101] George Moustris, Milad Geravand, Costas Tzafestas, and Angelika Peer. User-adaptive shared control in a mobility assistance robot based on human-centered intention reading and decision making schemes. In *IEEE International Conference on Robotics and Automation*, pages 1–4, 2016. doi: 10.13140/RG.2.2.21072.61449.

[102] Steffen Muller, Sina Sprenger, and Horst Michael Gross. Online adaptation of dialog strategies based on probabilistic planning. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, (October):692–697, 2014. doi: 10.1109/ROMAN.2014.6926333.

[103] Steffen Müller, Sina Sprenger, and Horst-Michael Gross. OfficeMate: A Study of an Online Learning Dialog System for Mobile Assistive Robots. *ADAPTIVE 2014, The Sixth International Conference on Adaptive and Self-Adaptive Systems and Applications*, (Adaptive):104–110, 2014.

[104] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations : Theory and application to reward shaping. *Sixteenth International Conference on Machine Learning*, 3:278–287, 1999. ISSN 1098-6596. doi: 10.1.1.48.345.

[105] Stefanos Nikolaidis, Anton Kuznetsov, David Hsu, and Siddhartha Srinivasa. Formalizing Human-Robot Mutual Adaptation: A Bounded Memory Model. In *Human-Robot Interaction*, pages 75–82, 2016. ISBN 9781467383707. doi: 10.1109/HRI.2016.7451736.

[106] Anthony F. Norcio and Jaki Stanley. Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. *IEEE Transactions on Systems, Man and Cybernetics*, 19(2): 399–408, 1989. ISSN 21682909. doi: 10.1109/21.31042.

[107] Xanthi S Papageorgiou, George P Moustris, Vassilis Pitsikalis, Georgia Chalvatzaki, Athanasios Dometios, Nikolaos Kardaris, Costas S Tzafestas, and Petros Maragos. User-Oriented Cognitive Interaction and Control for an Intelligent Robotic Walker. In *Workshop of the 2015 7th International Conference on Social Robotics, "Improving the quality of*

life in the elderly using robotic assistive technology: benefits, limitations, and challenges, 2015.

[108] ParoRobots. The Paro Robot. URL `http://www.parorobots.com/`.

[109] Jian Peng, Kim Kwang Raymond Choo, and Helen Ashman. User profiling in intrusion detection: A review. *Journal of Network and Computer Applications*, 72:14–27, 2016. ISSN 10958592. doi: 10.1016/j.jnca.2016.06.012.

[110] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1):414–454, 2014. ISSN 1553877X. doi: 10.1109/SURV.2013.042313.00197.

[111] D S Phatak and R Mulvaney. Clustering for personalized mobile Web usage. *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems FUZZ-IEEE'02.*, 1:705–710, 2002. ISSN 10987584. doi: 10.1109/FUZZ.2002.1005079.

[112] Frank Piller, Christoph Ihl, and Frank Steiner. Embedded toolkits for user co-design: A technology acceptance study of product adaptability in the usage stage. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–10, 2010. ISSN 15301605. doi: 10.1109/HICSS.2010.178.

[113] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010. ISSN 02628856. doi: 10.1016/j.imavis.2009.11.014.

[114] Mick Power, Kathryn Quinn, and Silke Schmidt. Development of the WHOQOL-Old module. *Quality of Life Research*, 14(10):2197–2214, 2005. ISSN 09629343. doi: 10.1007/s11136-005-7380-9.

[115] João Quintas, Gonçalo S Martins, Luis Santos, Paulo Menezes, and Jorge Dias. Toward a Context-Aware Human-Robot Interaction Framework Based on Cognitive Development. *IEEE Transactions On Systems, Man, And Cybernetics*, pages 1–11, 2018. doi: 10.1109/TSMC.2018.2833384.

[116] William Revelle and Klaus R Scherer. Personality and emotion. *Handbook of personality and affective science*, pages 304–305, 2005. doi: 10.1007/0-387-29907-6_6.

[117] José P. Ribeiro. *O importante é a saúde: estudo de adaptação de uma técnica de avaliação do estado de saúde - SF-36*. Lisboa, 2005.

[118] Juan Ramón Rico-Juan and José M. Iñesta. Adaptive training set reduction for nearest neighbor classification. *Neurocomputing*, 138:316–324, 2014. ISSN 18728286. doi: 10.1016/j.neucom.2014.01.033.

[119] Roboticstrends. The Jibo Robot. URL `http://www.roboticstrends.com/company/jibo`.

[120] Raquel Ros, Ilaria Baroni, and Yiannis Demiris. Adaptive human-robot interaction in sensorimotor task instruction: From human to robot dance tutors. *Robotics and Autonomous Systems*, 62(6):707–720, 2014. ISSN 09218890. doi: 10.1016/j.robot.2014.03.005.

[121] Silvia Rossi, François Ferland, and Adriana Tapus. User profiling and behavioral adaptation for HRI: A survey. *Pattern Recognition Letters*, 99:3–12, 2017. ISSN 01678655. doi: 10.1016/j.patrec.2017.06.002.

[122] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 15731405. doi: 10.1007/s11263-015-0816-y.

[123] L. Santos, D. Portugal, E. Christodolou, G. Samaras, P. Alvito, and J. Dias. Personalizable ICT-based Service Provision : The SocialRobot Solution. In *AmI 2014 International Workshop on Intelligent Environments Supporting Healthcare and Well-being (WISHWell'14)*.

[124] Luís Santos and Jorge Dias. User Routine Model using a Cloud-Connected Social Robot. In *5th IEEE International Conference on Cloud Networking*, 2016. doi: 10.1109/CloudNet.2016.34.

[125] Luís Santos, Kamrad Khoshhal, Jorge Dias, Kamrad Khoshhala, and Jorge Dias. Trajectory-based human action segmentation. *Pattern Recognition*, 48(2):568–579, 2015. ISSN 0031-3203. doi: 10.1016/j.patcog.2014.08.015.

[126] Ismael Sanz, Lledó Museros, Zoe Falomir, and Luis Gonzalez-Abril. Customizing a Qualitative Colour Description for Adaptability and Usability. *Pattern Recognition Letters*, 67:2–10, 2015. ISSN 01678655. doi: 10.1016/j.patrec.2015.06.014. URL `http://www.sciencedirect.com/science/article/pii/S0167865515001786`.

[127] Miguel Sarabia, Kyuhwa Lee, and Yiannis Demiris. Towards a Synchronised Grammars Framework for Adaptive Musical Human-Robot Collaboration. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, 2015. doi: 10.1109/ROMAN.2015.7333649.

[128] Katsushige Sawaki and Akira Ichikawa. Optimal control for partially observable markov decision processes over an infinite horizon. *Journal of the Operations Research Society of Japan*, 21(1):1–14, 1978.

[129] Bob R. Schadenberg, Mark A. Neerincx, Fokie Cnossen, and Rosemarijn Looije. Personalising game difficulty to keep children motivated to play with a social robot: A Bayesian approach. *Cognitive Systems Research*, 43:222–231, 2017. ISSN 13890417. doi: 10.1016/j.cogsys.2016.08.003.

[130] Ben Schmand, Cees Jonker, Chris Hooijer, and Jaap Lindeboom. Subjective memory complaints may announce dementia. *Neurology*, 46(1):121–125, 1996. doi: 10.1212/WNL.46.1.121.

[131] Ali Sekmen and Prathima Challa. Assessment of adaptive human-robot interactions. *Knowledge-Based Systems*, 42:49–59, 2013. ISSN 09507051. doi: 10.1016/j.knosys.2013.01.003.

[132] Emmanuel Senft, Paul Baxter, James Kennedy, and Tony Belpaeme. SPARC: Supervised Progressively Autonomous Robot Competencies. In *International Conference on Social Robotics*. Springer International Publishing, 2015. ISBN 978-3-642-25503-8. doi: 10.1007/978-3-319-25554-5_60. URL `http://link.springer.com/10.1007/978-3-319-02675-6`.

[133] Pedro Sequeira, Francisco S Melo, and Ana Paiva. Learning by appraising: an emotion-based approach to intrinsic reward design. *Adaptive Behavior*, 22(5):330–349, 2014. ISSN 1059-7123. doi: 10.1177/1059712314543837. URL `http://adb.sagepub.com/content/22/5/330.abstract`.

[134] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(1):379–423, 1948.

[135] Qiming Shen, Kerstin Dautenhahn, Joe Saunders, and Hatice Kose. Can real-time, adaptive human-robot motor coordination improve humans' overall perception of a robot? *IEEE Transactions on Autonomous Mental Development*, 7(1):52–64, 2015. ISSN 19430604. doi: 10.1109/TAMD.2015.2398451.

[136] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[137] Justin S Smith, Crystal Chao, and Andrea L Thomaz. Real-Time Changes to Social Dynamics in Human-Robot Turn-Taking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3024–3029, 2015. doi: 10.1109/IROS.2015.7353794.

[138] Harold Soh and Yiannis Demiris. Evolving policies for multi-reward partially observable markov decision processes (mr-pomdps). In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 713–720. ACM, 2011.

[139] Edward J Sondik. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations research*, 26(2):282–304, 1978.

[140] Istvàn Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. *on Machine Learning (ICML-10)*, pages 1031–1038, 2010. URL `http://www.icml2010.org/papers/546.pdf`.

[141] Tarek Taha and Jaime Valls Mir. A POMDP Framework for Modelling Human Interaction with Assistive Robots. 2011.

[142] Tarek Taha, Jaime Valls Miró, and Gamini Dissanayake. A pomdp framework for modelling human interaction with assistive robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 544–549. IEEE, 2011.

[143] Adriana Tapus, Cristian Tapus, and Maja Mataric. User-Robot Personality Matching and Robot Behavior Adaptation for Post-Stroke Rehabilitation Therapy. *Intelligent Service Robotics*, 1(2):169–183, 2008.

[144] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michiel Joosse, Harmish Khambhaita, Tomasz Kucner, Bastian Leibe, Achim J Lilienthal, Timm Linder, Manja Lohse, Martin Magnusson, Billy Okal, Luigi Palmieri, Umer Rafi, Marieke Van Rooij, and Lu Zhang. SPENCER : A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports. pages 1–14.

[145] M. A. Turk and Alex Pentland. Face Recognition Using Eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR'91)*, volume 3, pages 72 – 86. IEEE, 1991. ISBN 0-8186-2148-6. doi: 10.1109/CVPR.1991.139758.

[146] Diane M. Turner-Bowker, Martha S. Bayliss, John E. Ware Jr., and Mark Kosinski. Usefulness of the SF-8TM Health Survey for comparing the impact of migraine and other conditions. *Quality of Life Research: An International Journal of Quality of Life Aspects of Treatment, Care & Rehabilitation*, 12(8):1003–1012, 2003. ISSN 0962-9343. doi: 10.1023/A:1026179517081.

[147] Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U. Khan. Big Data Reduction Methods: A Survey. *Data Science and Engineering*, 1(4):265–284, 2016. ISSN 2364-1185. doi: 10.1007/s41019-016-0022-0. URL `http://link.springer.com/10.1007/s41019-016-0022-0`.

[148] Joke Van Den Broeck, Gina Rossi, Eva Dierckx, and Barbara De Clercq. Age-neutrality of the NEO-PI-R: Potential Differential Item Functioning in older versus younger adults. *Journal of Psychopathology and Behavioral Assessment*, 34:361–369, 2012. ISSN 08822689. doi: 10.1007/s10862-012-9287-4.

[149] Devara Vasumathi, A. Govardhan, and Kumar Suresh. Effective web personalization using clustering. *2009 International Conference on Intelligent Agent & Multi-Agent Systems*, pages 1–7, 2009. doi: 10.1109/IAMA.2009.5228056. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5228056`.

[150] Manuela Vilar, Liliana B. Sousa, and Mário R. Simões. The European Portuguese WHOQOL-OLD module and the new facet Family/Family life: reliability and validity studies. *Quality of Life Research*, 25(9):2367–2372, 2016. ISSN 15732649. doi: 10.1007/s11136-016-1275-9.

[151] Alessandro Vinciarelli. More Personality in Personality Computing. *IEEE Transactions on Affective Computing*, 5(3):297–300, 2014. ISSN 1949-3045. doi: 10.1109/TAFFC.2014.2341252.

[152] Alessandro Vinciarelli and Gelareh Mohammadi. A Survey of Personality Computing. *IEEE Transactions on Affective Computing*, 5(3):273–291, 2014. doi: 10.1109/TAFFC.2014.2330816.

[153] Kazuyoshi Wada and Takanori Shibata. Robot therapy in a care house-Its sociopsychological and physiological effects on the residents. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, (May):3966–3971, 2006. doi: 10.1109/ROBOT.2006.1642310.

[154] Gail M. Wagnild and Heather M. Young. Development and psychometric evaluation of the Resilience Scale. *Journal of Nursing Measurement*, 1(2):165–178, 1993.

[155] Senzhang Wang, Zhoujun Li, Chunyang Liu, Xiaoming Zhang, and Haijun Zhang. Training data reduction to speed up SVM training. *Applied Intelligence*, 41(2):405–420, 2014. ISSN 0924669X. doi: 10.1007/s10489-014-0524-2.

[156] J Kory Westlund, Goren Gordon, Samuel Spaulding, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. Learning A Second Language with a Socially Assistive Robot. *Almere, The Netherlands*, 2015.

[157] Sally Whelan, Kathy Murphy, Eva Barrett, Cheryl Krusche, Adam Santorelli, and Dympna Casey. Factors Affecting the Acceptability of Social Robots by Older Adults Including People with Dementia or Cognitive Impairment: A Literature Review. *International Journal of Social Robotics*, 2018. ISSN 1875-4791. doi: 10.1007/s12369-018-0471-x. URL `http://link.springer.com/10.1007/s12369-018-0471-x`.

[158] Chelsea C White III and William T Scherer. Solution procedures for partially observed markov decision processes. *Operations Research*, 37(5):791–797, 1989.

[159] James R Williams. Guidelines for the use of Multimedia in Instruction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 1998.

[160] Kyle Hollins Wray and Shlomo Zilberstein. Multi-objective pomdps with lexicographic reward preferences. In *IJCAI*, pages 1719–1725, 2015.

[161] Xiaodong Yang and Yingli Tian. Effective 3D action recognition using EigenJoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014. ISSN 10473203. doi: 10.1016/j.jvcir.2013.03.001.

[162] Jerome A Yesavage, T.L. Brink, Terence L Rose, Owen Lum, Virginia Huang, Michael Adey, and Von Otto Leirer. Development and validation of a geriatric depression screening scale: A preliminary report. *Journal of Psychiatric Research*, 17(1):37–49, 1982. ISSN 00223956. doi: 10.1016/0022-3956(82)90033-4.

[163] Hong Zhang, Ping Wu, Aryel Beck, Zhijun Zhang, and Xingyu Gao. Adaptive incremental learning of image semantics with application to social robot. *Neurocomputing*, 173:93–101, 2016. ISSN 18728286. doi: 10.1016/j.neucom.2015.07.104.

[164] Nevin L Zhang and Wenju Liu. Planning in stochastic domains: Problem characteristics and approximation. Technical report, Technical Report HKUST-CS96-31, Hong Kong University of Science and Technology, 1996.

[165] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. Taxonomy discovery for personalized recommendation. *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 243–252, 2014. doi: 10.1145/2556195.2556236. URL `http://dl.acm.org/citation.cfm?doid=2556195.2556236`.

[166] Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Matthew Ma. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1323–1331, 2010. ISSN 15587916. doi: 10.1109/TASL.2009.2033421.