Apprentissage simultané d'une tâche nouvelle et de l'interprétation de signaux sociaux d'un humain en robotique

Jonathan Grizou

INRIA BORDEAUX SUD-OUEST

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE
UNIVERSITÉ DE BORDEAUX

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Bordeaux
Mention : Informatique

Présentée et soutenue par

Jonathan Grizou

Apprentissage simultané d'une tâche nouvelle et de l'interprétation de signaux sociaux d'un humain en robotique

---

Thèse dirigée par Manuel Lopes et Pierre-Yves Oudeyer

préparée à INRIA Bordeaux Sud-Ouest, Équipe Flowers

soutenue le 24 Octobre 2014

Jury :

Rapporteurs :	Mohamed Chetouani	-	Prof.	ISIR, UPMC
	Michèle Sebag	-	DR	CNRS-INRIA
Examinateurs :	Fabien Lotte	-	CR	INRIA
	Luis Montesano	-	Assoc. Prof.	Univ. of Zaragoza
Directeurs :	Manuel Lopes	-	CR	INRIA-ENSTA
	Pierre-Yves Oudeyer	-	DR	INRIA-ENSTA

Résumé substantiel

Cette thèse s'intéresse à un problème logique dont les enjeux théoriques et pratiques sont multiples. De manière simple, il peut être présenté ainsi : imaginez que vous êtes dans un labyrinthe, dont vous connaissez toutes les routes menant à chacune des portes de sortie. Derrière l'une de ces portes se trouve un trésor, mais vous n'avez le droit d'ouvrir qu'une seule porte. Un vieil homme habitant le labyrinthe connaît la bonne sortie et se propose alors de vous aider à l'identifier. Pour cela, il vous indiquera la direction à prendre à chaque intersection. Malheureusement, cet homme ne parle pas votre langue, et les mots qu'il utilise pour dire "droite" ou "gauche" vous sont inconnus. Est-il possible de trouver le trésor et de comprendre l'association entre les mots du vieil homme et leurs significations ?

Ce problème, bien qu'en apparence abstrait, est relié à des problématiques concrètes dans le domaine de l'interaction homme-machine que nous présentons aux chapitres 1 et 2. Remplaçons le vieil homme par un utilisateur souhaitant guider un robot vers une sortie spécifique du labyrinthe. Ce robot ne sait pas en avance quelle est la bonne sortie mais il sait où se trouvent chacune des portes et comment s'y rendre. Imaginons maintenant que ce robot ne comprenne pas a priori le langage de l'humain; en effet, il est très difficile de construire un robot à même de comprendre parfaitement chaque langue, accent et préférence de chacun. Il faudra alors que le robot apprenne l'association entre les mots de l'utilisateur et leur sens, tout en réalisant la tâche que l'humain lui indique (i.e. trouver la bonne porte). Ce problème n'est pas simple, car pour comprendre le sens des signaux il faudrait connaître la tâche, et pour connaître la tâche il faudrait connaître le sens des signaux.

Il s'agit donc, pour un labyrinthe donné, de trouver la suite d'actions permettant de collecter suffisamment d'informations de la part de l'humain pour comprendre à la fois le sens de ses mots et la porte derrière laquelle se cache le trésor. Cela dépend donc de la configuration du labyrinthe et de l'historique complet de l'interaction entre les deux protagonistes.

Dans cette thèse, nous présentons une solution à ce problème. Pour cela, nous faisons d'abord l'hypothèse qu'un nombre fini de tâches est définies et connues de l'homme et de la machine, i.e. qu'un nombre fini de portes existent. Nous supposons également que le robot dispose d'un modèle de la logique de l'utilisateur, et est donc capable de faire le raisonnement suivant : si l'humain veut que j'aille vers la porte 1, alors lorsque je suis à l'intersection I, il devrait logiquement me dire d'aller dans la direction D. A noter que cette phrase commence par une supposition sur la tâche, qui n'est en aucun cas connue à l'avance. Ainsi, le robot étant équipé de plusieurs hypothèses (porte 1, 2, 3,...), lorsqu'il se trouve à l'intersection I, l'utilisateur prononce un mot (par exemple "wadibou"), dont autant d'interprétations sont faites que d'hypothèses sur la tâche.

Notre hypothèse sous-jacente est que l'utilisateur est logique et cohérent tout au

long de l'interaction, utilisant toujours le même mot pour dire la même chose. Il nous faut donc tenir compte de tout l'historique de l'interaction pour analyser quels mots ont été utilisés pour dire quoi, selon chaque hypothèse de tâche. Nous comprenons ainsi que, sous certaines conditions qui sont explicitées au chapitre 4, il est possible d'éliminer toutes les hypothèses générant des interprétations incohérentes du sens des signaux. L'unique hypothèse restante nous informera donc à la fois de la bonne tâche, i.e. la bonne porte à ouvrir, mais aussi de la bonne association entre les mots de l'utilisateur et les sens qui y sont associés, i.e. de son langage.

Une autre façon de décrire ce travail est de parler d'auto-calibration. En effet, en s'adaptant à l'utilisateur pendant l'interaction, notre algorithme ne fait aucun apriori sur le sens des signaux qu'il reçoit. Cela revient bien à créer des interfaces ne nécessitant pas de phase de calibration car la machine peut s'adapter, automatiquement et pendant l'interaction, à différentes personnes qui ne parlent pas la même langue ou qui n'utilisent pas les mêmes mots pour dire la même chose. Cela veut aussi dire qu'il est facile d'étendre notre approche à d'autres modalités d'interaction (par exemple des gestes, des expressions faciales ou des ondes cérébrales).

Remplaçons le problème du labyrinthe par une tâche plus concrète et utile. Prenons l'exemple d'une personne aux capacités de communication réduites avec le monde extérieur, ne pouvant utiliser par exemple que de fragiles clignements des yeux ou ayant recours à l'enregistrement de ses ondes cérébrales (EEG). Il devient alors difficile, voire même impossible de savoir à l'avance les intentions de communication de ces personnes. Il est donc primordial de disposer de machines qui sont à même de s'adapter automatiquement à chaque personne. Il n'est ainsi pas surprenant de voir que c'est la communauté de l'interaction cerveau-machine (BCI) qui s'est intéressée le plus au problème de l'auto-calibration. En effet, à l'opposé des modes d'interaction classiques tels que la parole, les gestes ou les expressions faciales, nous avons très peu d'aprioris sur l'utilisation des signaux du cerveau.

Résultats

Notre approche est donc très générique. Elle permet à un humain de commencer à interagir avec une machine afin de résoudre une tache séquentielle sans que celle-ci ne comprenne à l'avance les signaux de communication de l'utilisateur.

Nous appliquons nos algorithmes d'auto-calibration à deux exemples typiques de l'interaction homme-robot et de l'interaction cerveau-machine : une tâche d'organisation d'une série d'objets selon les préférences de l'utilisateur qui guide le robot par la voix (chapitre 4, voir figure 1 - gauche), et une tâche de déplacement sur une grille guidé par les signaux cérébraux (EEG) de l'utilisateur (chapitre 6, voir figure 1 - droite).

Bien que les expériences du chapitre 4 soient fondatrices, nous préférons nous concentrer pour ce bref résumé sur les expériences BCI. Elles présentent un aspect plus appliqué car testées sur de vrais sujets en temps réel et sur une tâche d'actualité pour les interfaces cerveau-machine.

Au chapitre 6, nous présentons l'application principale de ce travail aux inter-

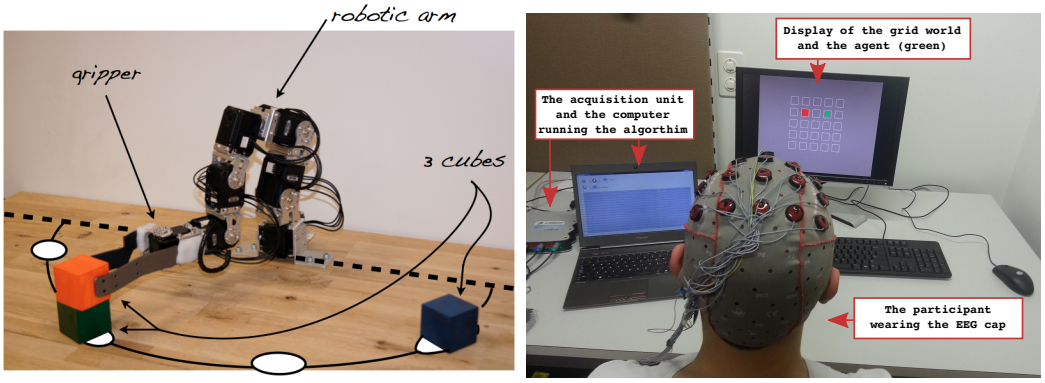


Figure 1: Illustration des deux setups expérimentaux utilisés dans ce travail. A gauche : le bras robotique pour la tâche d'organisation de trois cubes. A droite : l'interface cerveau-machine composée d'un casque avec ses électrodes et d'un écran affichant les informations relatives à la tache.

faces cerveau-machine. Ce genre d'interface permet aux personnes à fort handicap d'interagir avec le monde extérieur par le biais de leur activité cérébrale. Plus précisément, nous pouvons enregistrer des variations de potentiel à la surface de leur cerveau. Ces ondes ont des propriétés différentes en fonction de l'activité mentale du sujet. Il est possible de différencier des activités motrices, ou même des signaux d'erreur de type oui/non. Le problème de ces systèmes est qu'ils ne sont pas universels et doivent être adaptés à chaque utilisateur. Cette adaptation est faite par le biais d'une phase de calibration où l'utilisateur doit répéter plusieurs centaines de fois les mêmes actions mentales. Pendant ce temps, le système est inutilisable et l'intervention d'une personne extérieure est nécessaire. Non seulement cette phase de calibration est ennuyeuse et rébarbative, mais elle doit être effectuée régulièrement car les signaux varient de jour en jour ou car la position du casque change.

L'utilisation d'algorithmes d'auto-calibration permettrait donc une plus grande flexibilité d'utilisation de ces technologies et permettrait de les utiliser chez soi sans la supervision d'un spécialiste.

Dans cette thèse, nous présentons donc des expériences où des sujets humains ont pour tâche de guider un agent dans un labyrinthe en lui indiquant si ses actions sont "correctes" ou "incorrectes" vis-à-vis de l'objectif défini, simplement en pensant à "correct" ou "incorrect". Les "pensées" de l'utilisateur sont mesurées par le biais d'électrodes au contact de son cerveau. Le setup expérimental est celui présenté sur la figure 1 (droite).

La figure 2 présente le résultat principal de cette thèse. Elle compare la différence entre un algorithme nécessitant une phase de calibration et les algorithmes d'auto-calibration développés dans cette thèse. Ce sont des résultats de simulation avec de vraies données EEG. Notre algorithme (figure 2 - haut) permet de résoudre une première tâche en seulement 85 itérations, bien avant que la phase de calibration ne soit complète (400 itérations étant une période typique de calibration pour ce genre

de système). Enfin, notre méthode résout une dizaine de tâches en 400 itérations, soit avant qu'un système traditionnel ne soit opérationnel.

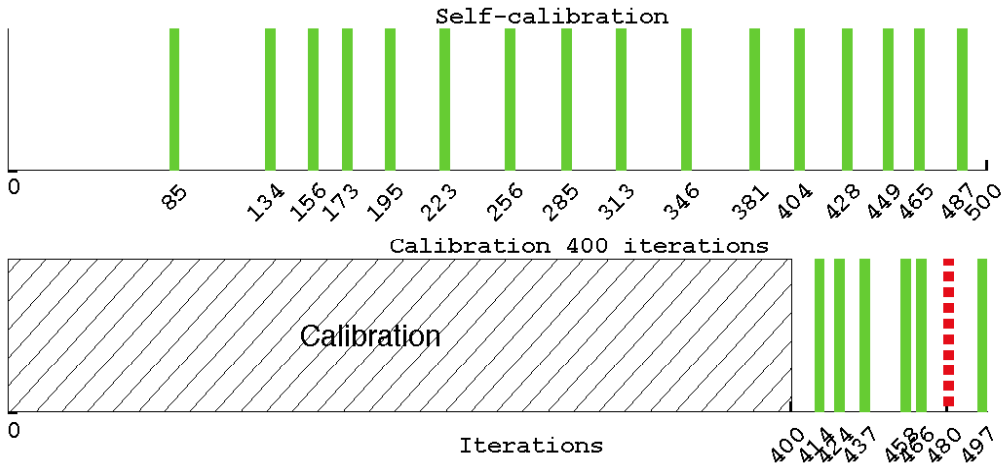


Figure 2: Nombre de tâches résolues sur 500 itérations avec des données EEG. L'algorithme d'auto-calibration (haut) est comparé aux méthodes nécessitant une phase de calibration (bas, ici 400 itérations de calibration). Les barres vertes et rouges représentent respectivement les bonnes et les mauvaises exécutions de la tâche par la machine. La méthode d'auto-calibration proposée dans cette thèse permet de compléter une première tâche plus rapidement, sans pour autant faire d'erreur.

Les mêmes expériences ont été faites avec des utilisateurs réels. Leurs résultats confirment ceux des simulations et sont présentés au chapitre 6 et 7.3. Nos résultats démontrent expérimentalement que notre approche est fonctionnelle et permet une utilisation pratique de l'interface plus rapidement. De plus, notre système ne nécessite pas la présence d'une personne extérieure pour se calibrer. Il est donc un candidat potentiel pour amener l'utilisation des interfaces cerveau-machine dans les foyers.

Planification des actions

Les actions de la machine font partie intégrante de la performance de nos algorithmes. En effet, si la machine ne bouge pas, alors aucun signal ne sera reçu et ni la tâche ni le modèle de langage ne seront jamais appris. Nous avons donc étudié quelle stratégie de sélection des actions devrait suivre la machine afin d'apprendre le plus efficacement possible. Un certain nombre de méthodes sur des problèmes généraux existent. Elles consistent en général à mesurer l'incertitude sur le problème et à trouver les actions ayant la plus grande probabilité de réduire cette incertitude.

Comparé aux algorithmes existants, notre problème inclut une couche d'incertitude supplémentaire : non seulement la tâche est inconnue, mais aussi le sens des signaux. Il faut donc inclure cette double incertitude pour naviguer plus ef-

ficacement dans l'environnement et collecter des informations d'une façon optimisée. Les résultats présentés au chapitre 5 montrent que notre méthode de planification des actions améliore significativement le temps nécessaire à l'identification de la tâche, mais aussi à l'établissement du modèle de langage de l'utilisateur.

Extensions

Au chapitre 7, nous proposons des solutions aux multiples limitations de l'approche présentée dans cette thèse. Nous montrons d'abord qu'il est possible d'utiliser nos algorithmes dans des espaces continus : premièrement pour un état continu du système (chapitre 7.4), mais aussi pour un ensemble infini d'hypothèses sur la tâche (chapitre 7.5). Par la suite, nous montrons que la connaissance a priori du protocole d'interaction n'est pas une limitation forte et que notre système peut détecter le protocole par l'interaction pratique avec l'utilisateur (chapitre 7.6).

Paradoxalement, cette thèse ne traite pas directement du problème simple et symbolique, mais s'intéresse d'abord à une représentation continue des signaux de communication. Ceci est fait dans un but applicatif, auquel de fastidieuses preuves mathématiques dans des domaines trop simplifiés n'auraient guère laissé de temps à l'expérimentation. Ainsi, la formulation simple du labyrinthe présentée au début de ce résumé n'est adressée que dans la toute dernière section de cette thèse (chapitre 7.7) par une preuve de la validité de notre solution, pour le cas de signaux de communication symboliques et sous de fortes contraintes environnementales. Ce dernier développement montre que ce genre de problème peut être modélisé mathématiquement et ouvre la voie à de prochaines explorations plus théoriques. Elles permettront peut-être d'avoir de plus grandes garanties sur la convergence et les performances de nos algorithmes. Il est à noter que ce type de preuve est encore très limité pour l'interaction pratique du fait de l'imprévisibilité du comportement humain.

Expèrience humain-humain

Cette thèse traite également de la mise en place d'un protocole expérimental pour analyser le comportement de deux humains mis dans la situation que doivent résoudre nos algorithmes (chapitre 3). Dans cette expérience, deux personnes doivent collaborer à l'exécution d'une tâche de construction. Elles ne peuvent interagir que par le biais d'une interface dont le sens des signaux transmis est inconnu et indéfini au départ pour les deux parties.

Il sera intéressant de voir la dynamique de construction d'un langage commun entre les deux participants. Ce langage, qui n'était pas prévu au début de l'interaction, s'établit de telle sorte qu'une personne extérieure à l'expérience ne pourra alors pas comprendre ce qui se passe en observant le résultat final de l'interaction.

Conclusion

La vision développée dans cette thèse est qu'il est possible pour une machine d'interagir avec un humain sans comprendre initialement la façon dont l'utilisateur communique. Plus concrètement, notre système n'a pas de préjugé sur le sens des signaux reçus et construit son modèle durant l'interaction pratique avec l'utilisateur sans jamais avoir accès à une source sûre d'information. Nous espérons que cela sera le fruit de nombreux travaux futurs.

Au-delà du défi technique de l'auto-calibration, des questions pratiques d'utilisation et d'acceptabilité apparaissent et sont présentées au chapitre 8. La plus importante à tester en condition réelle est la réaction qu'auront les utilisateurs face au fait que la machine, i.e. le robot, ne soit pas immédiatement réactif à leurs ordres. Le robot doit en effet apprendre le sens des signaux pendant l'interaction. Même si nos algorithmes apportent une plus grande flexibilité d'interaction, ils ne permettent pas à l'utilisateur une fonctionnalité immédiate et parfaite du système. Cette phase d'apprentissage pourrait être perçue comme une déficience et par conséquent impacter l'intérêt et l'utilisabilité réelle de notre système.

Mots-clés : Auto-Calibration, Apprentissage par Interaction, Interaction Humain-Robot, Interface Cerveau-Machine, Interaction Intuitive et Adaptative, Robotique, Acquisition de Symboles, Apprentissage Actif, Calibration.

Ce travail a été financé par INRIA, le Conseil Régional d'Aquitaine et la bourse ERC EXPLORERS 24007.

INRIA BORDEAUX SUD-OUEST

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE
UNIVERSITÉ DE BORDEAUX

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Bordeaux
Specialty : Computer Science

Defended by

Jonathan Grizou

Learning from Unlabeled Interaction Frames

Thesis Advisors: Manuel Lopes and Pierre-Yves Oudeyer

prepared at INRIA Bordeaux Sud-Ouest, Flowers Team

defended on October 24, 2014

Jury :

Reviewers :	Mohamed Chetouani	-	Prof.	ISIR, UPMC
	Michèle Sebag	-	DR	CNRS-INRIA
Examinators :	Fabien Lotte	-	CR	INRIA
	Luis Montesano	-	Assoc. Prof.	Univ. of Zaragoza
Advisors :	Manuel Lopes	-	CR	INRIA-ENSTA
	Pierre-Yves Oudeyer	-	DR	INRIA-ENSTA

Acknowledgments

First of all, I would like to thank Pierre-Yves Oudeyer and Manuel Lopes for supervising me during these three years. Apart from their excellent advice and their open minded and supportive supervision, I particularly enjoyed the opportunity I was given to learn and to simply belong to the lab, listening and taking part in various enlightening scientific and philosophical discussions.

I am grateful to Dr. Charles Capaday, Dr. Ramón Huerta and Prof. Auke Jan Ijspeert for providing me with the early opportunity to discover research in their respective labs. Without them, I would never have pursued doctoral studies.

During these three years, I had the chance to engage in different collaborative works. I thank Iñaki Iturrate and Luis Montesano for their numerous advice and their infinite patience. I thank Anna-Lisa Vollmer and Katharina J. Rohlfing for their invaluable insight on the implication of this work to human behavioral understanding. I thank Peter Stone and Samuel Barrett for their enthusiasm and the opportunity I had to visit the Larg lab.

Many thanks to the members of my jury, Michèle Sebag, Mohamed Chetouani, Luis Montesano and Fabien Lotte, for accepting to review this work.

Numerous thanks to all the members of the Flowers team for contributing in making these three years a memorable experience: Matthieu, Pierre, Fabien, Olivier, Paul, Clément, Thomas, Thomas, Jérôme, Haylee, Didier, Steve, Mai, Thibault, Yoan, Benjamin, and Adrien. I also thank our team assistants Nicolas, Nathalie, Catherine, and the numerous internship students, especially the one I co-supervised: Mathieu, Axel, Julie, Chloé, Brice, and Fabien.

Unlimited thanks to my family for the good start in life they gave me, my friends for they unique ability to make me laugh, and Sara for her support all along these three years.

Abstract

This thesis investigates how a machine can be taught a new task from unlabeled human instructions, which is without knowing beforehand how to associate the human communicative signals with their meanings. The theoretical and empirical work presented in this thesis provides means to create calibration free interactive systems, which allow humans to interact with machines, from scratch, using their own preferred teaching signals. It therefore removes the need for an expert to tune the system for each specific user, which constitutes an important step towards flexible personalized teaching interfaces, a key for the future of personal robotics.

Our approach assumes the robot has access to a limited set of task hypotheses, which include the task the user wants to solve. Our method consists of generating interpretation hypotheses of the teaching signals with respect to each hypothetic task. By building a set of hypothetic interpretation, i.e. a set of signal-label pairs for each task, the task the user wants to solve is the one that explains better the history of interaction.

We consider different scenarios, including a pick and place robotics experiment with speech as the modality of interaction, and a navigation task in a brain computer interaction scenario. In these scenarios, a teacher instructs a robot to perform a new task using initially unclassified signals, whose associated meaning can be a feedback (correct/incorrect) or a guidance (go left, right, up, ...). Our results show that a) it is possible to learn the meaning of unlabeled and noisy teaching signals, as well as a new task at the same time, and b) it is possible to reuse the acquired knowledge about the teaching signals for learning new tasks faster. We further introduce a planning strategy that exploits uncertainty from the task and the signals' meanings to allow more efficient learning sessions. We present a study where several real human subjects control successfully a virtual device using their brain and without relying on a calibration phase. Our system identifies, from scratch, the target intended by the user as well as the decoder of brain signals.

Based on this work, but from another perspective, we introduce a new experimental setup to study how humans behave in asymmetric collaborative tasks. In this setup, two humans have to collaborate to solve a task but the channels of communication they can use are constrained and force them to invent and agree on a shared interaction protocol in order to solve the task. These constraints allow analyzing how a communication protocol is progressively established through the interplay and history of individual actions.

Keywords: Self-calibration, Learning from Interaction, Human-Robot Interaction, Brain-Computer Interfaces, Intuitive and Flexible Interaction, Robotics, Symbol Acquisition, Active Learning, Calibration.

This work has been supported by INRIA, Conseil Régional d'Aquitaine, and the ERC grant EXPLORERS 24007.

Contents

Chapter 1

Introduction

Contents

In the past decades, robotics and autonomous systems have seen tremendous improvements in their motor, perceptual, and computational capabilities. As a good example, we have been able to send and operate rovers for several years on the planet Mars (Spirit, Opportunity, Curiosity), which indicates the technologies are well mastered. However, getting such robots to do what we want them to do remains a skill of few, and bringing robotics systems teachable by everyone and capable of social interaction in our daily life has been identified as the next milestone for the robotic community.

As for bringing computers in our homes required easy and intuitive ways for people to make use of them, bringing robots in our daily life requires easy and intuitive ways for people to instruct robots to do useful things for them. But due to the diversity of skills a robot should be able to execute in our daily environment, including interacting with humans and objects, traditional programming methods hinder the deployment of robotic system at homes and workspaces.

Instead, researchers are trying to endow robotic systems with the ability to learn from social interaction. Several methods have been considered to allow non-technical users to "program" robots, such as *learning by demonstration* where a person demonstrates the skills to the robot, *learning from reinforcement* where a person assesses the actions of the robot with respect to the aimed behavior, or

learning from advice where a person explains the sequence of actions to perform in order to fulfill a task.

Endowing a robot with the ability to learn from interaction with a human requires solving several challenges: the technical challenge of motor, perceptual, and cognitive skills acquisition and generalization, as well as the practical challenge of interacting in a social way with humans. Especially, the robot must be able to understand the communicative signals from the human.

Currently most of these challenges are considered in isolation. For example, when a robot learns a task from human instructions, the robot receives instructions in a symbolic way, e.g. if the human uses speech to communicate his instructions, the robot is assumed to be able convert raw speech into text. Similarly, when a robot learns how to recognize speech utterances, which is how to convert raw speech into a meaningful representation, such as text, the robot is usually fed with many examples of speech utterances associated with their symbolic representation.

In this thesis, we consider the two latter challenges simultaneously, which is learning a new task from raw human instruction signals whose associated meanings are initially unknown. Solving this problem would allow the same robot to be taught by a variety of users using their preferred teaching signals, and without the intervention of an expert to calibrate the system for each users. For example, a robot that accepts speech commands usually accept only one or a limited set of prespecified speech utterances for each command, e.g. using the word "forward" to ask the robot to move forward. With the method described in this thesis, the user could use its preferred word to ask the robot to move forward, e.g. "straight" or "up", but also words whose usual meanings are non-related to the move forward action such as "dog", "backwards", or "blue", or interjection such as "ah", "oh", or even non speech utterances such as a hand clapping. The robot, after some practical interaction with the user, will find out which signal is associated to the action moving forward.

In the following of this introduction, we present in more details the challenges of learning from social interaction with humans and explicit the usual assumptions made when designing such systems. On this basis, we define the specific challenge of *learning from unlabeled interaction frames* and present the contribution of the thesis.

1.1 Social Learning: Robot learning from interaction with humans

It is often easier to acquire a new skill if someone that has already acquired that skill teaches us how to do it. The field of social learning in robotics investigates how knowledge can be transferred from humans to robots through social interaction. Social interaction implies that the human interacts with the machine using similar modalities as when interacting with other human beings, for example using speech, gestures, or by demonstrating some behaviors.

We can identify three main social learning paradigms used in robotics today:

(a) learning from human demonstration, where the robot learns by imitating the human actions, (b) learning from human reinforcement, where the robot learn from assessments on its own actions provided by the user, and (c) learning from human advice, where the robot learns from concrete instructions about what do to next provided by the user.

Each of these paradigms requires to solve two main challenges: (1) the robot must be able to identify which parts of the interaction, and of the environment, are relevant to the acquisition of the new skill, and (2) the robot must be able to infer, from the relevant information extracted from the interaction, the new skill, or task, the human wants the robot to achieve.

As we will see in the following subsections, most of the work in robot social learning considered these two latter challenges separately and most of the efforts focused on the second challenge of learning a new skill from pre-formatted data.

1.1.1 Learning from human demonstrations

Learning from human demonstrations, also called programming by demonstration or learning by imitation, is the process of learning a new skill from practical examples of how to perform that skill [Schaal 1999, Calinon 2008, Argall 2009, Lopes 2010]. More formally, the robot must infer a policy that is a mapping between world states and actions by observing only some, potentially noisy, examples of state to action mapping.

Following the survey of Argall [Argall 2009], we segment our presentation of learning from human demonstration in three parts, first we present the different methods used to collect training data, i.e. to gather the demonstrations, then we present several methods allowing to derive a policy from demonstration, and finally we highlight some limitations of the method.

Collecting demonstrations

Collecting demonstrations is probably the most important part in the learning process. Demonstrations of good quality will result in an easier learning while bad quality demonstrations are likely to impact the quality of the learned behavior.

We group the demonstration recording methods in two categories: (1) by teleoperation, where the human demonstrate a skill by directly controlling the robot, and (2) by external observation, where the robot is observing the human providing demonstration. More formally, learning from data collected by a direct control of the robot by the human is called learning from demonstration, while learning from data collected by observing the human demonstrating the skill is called learning from imitation.

During teleoperation, the robot is directly operated by the teacher and therefore records the demonstration using its own sensors, i.e. the robot directly observes a sequence of state-action pairs in its own referential. It is the most direct and most precise method to provide demonstrations. However this method does not always

apply well to robots. For example, robots with many degrees of freedom cannot be teleoperated efficiently by one person, but also robots that should maintain equilibrium are sometimes impossible to manipulate directly, such as demonstrating a walking behavior by teleoperating the legs of a humanoid robot.

Teleoperation has been used in a variety of robotic applications, including learning of aerobatic helicopter flight [Abbeel 2007], object displacement with environmental constrained (e.g. obstacle avoidance) [Guenter 2007, Calinon 2007b], object stacking [Calinon 2007b], or ball grasping on the Aibo robot [Grollman 2007b].

When learning by imitation, the robot observes a human teacher demonstrating the skill. The fundamental difference with the teleoperation approach is the difference of embodiment between the human and the robot. This issue is referred as the correspondence problem [Nehaniv 2002], which is the problem of mapping between the demonstrator actions (i.e. the human) and the imitator actions (i.e. the robot). For example, when demonstrating a gesture to a humanoid robot, as the human and the robot do not share the same body characteristics, the robot cannot directly transpose the human movement to its own body. If we consider a humanoid robot imitating the posture of a human demonstrator, the problem is better defined as reproducing the human posture as closely as possible while maintaining balance [Hyon 2007, Yamane 2009], where the system designer provides some additional constraints to the robot.

Recording the human demonstration can be done using a variety of sensors, either by adding sensors directly on the users (wearable sensors), either by using only sensors remotely observing the demonstrator body or relevant objects, for example using a motion capture device or using a pair of video cameras.

Learning by imitation has been investigated in a variety of robotic applications, including executing a tennis forehand swing [Ijspeert 2002b], imitating arm movement [Billard 2001] and hand posture [Chella 2004], object grasping [Lopes 2005, Tegin 2009], but also demonstration including a force component such as the fingertip force for grasping and manipulating objects [Lin 2012]. Other works focused on learning by imitation a sequential task, which required combining a sequence of multiple actions to fulfill the task [Pardowitz 2005, Natarajan 2011].

Inferring a policy

Given a dataset of demonstrations collected using one of the methods presented above, the robot should infer what action it should take in any given state to correctly fulfill the task demonstrated by the human. This process can be as straightforward as reproducing the demonstrated behavior exactly. But most often, as the demonstrations may be noisy or incomplete, the robot needs to learn and generalize from these examples. We will differentiate between two approaches: (a) directly deriving a mapping between states and actions, i.e. a policy, from the observed data with the aim or reproducing the teacher policy, and (b) inferring the human objective and reproducing the desired outcome without necessarily using the same actions as the demonstrator. Roughly, the first approach is more suited for imita-

tion, which is the act of reproducing the human demonstration in all details, while the second is more suited for emulation, which is the act of fulfilling the same goal as the human.

The first approach resumes in approximating the policy function observed form the user behavior. Depending on the properties of the problem, the algorithms for learning the policy are either classification or regression techniques.

- **Classification** methods are well suited for mapping discrete or continuous states to discrete actions. An example would be a robot learning to play a video game from demonstration in which, depending on the current state of the agent in the world, the robot should learn to press the appropriate buttons.

  A large variety of classification algorithm has been used in learning from demonstration scenario. Among others, Support Vector Machines were used for a robot learning how to sort balls [Chernova 2008b], Hidden Markov Models have been used for an assembly task [Hovland 1996], Gaussian Mixture Models in a simulated driving domain [Chernova 2009], but also neural networks [Mataric 2000], beta regression [Montesano 2009] or k-Nearest Neighbors [Saunders 2006].

- **Regression** method are well suited for mapping discrete or continuous states to continuous actions. An example would be an autonomous car learning to steer the wheels from demonstration, given information about the surrounding environment the car should turn the driving wheel appropriately.

  A large variety of regression algorithm has been used in learning form demonstration scenarios, they were mainly applied for learning trajectories from noisy demonstrations. Among others, Gaussian Mixture Regression for generalizing trajectories from examples in different applications [Calinon 2007a], Locally Weighted Regression for learning to produce rhythmic movement using central pattern generators [Schaal 1998, Ijspeert 2002a], Neural Networks for learning autonomous driving [Pomerleau 1991], or Incremental Local Online Gaussian Mixture Regression for imitation learning for learning incrementally and online new motor tasks from demonstration [Cederborg 2010].

The second approach consists of inferring the goal of the human from demonstration. By expressing this goal as an optimization problem or as a reward function, the robot can learn to reproduce the human goal by its own means.

Inverse reinforcement learning [Ng 2000, Abbeel 2004, Lopes 2009b] is a popular method that is inferring the hidden reward function the demonstrator is trying to optimize based on the observation of its actions. In addition, the human demonstrations can be used to learn a model of the environment in state unreachable to robot by mere self-exploration. Once the reward function has been evaluated form the demonstrations, and given the dynamic of the environment, the robot can generate a plan to fulfill the task using its own ability. This method is especially interesting when the human and the robot do not have the same abilities. As an example, a

robot may be able to execute a skill faster that a human, but by mere reproduction of the human gestures the robot would not reach the same level of performance than by inferring the underlying goal of the human and solving the problem its own way.

One of the most impressive achievement of the past decade used inverse reinforcement learning methods for the learning of aerobatic helicopter flight [Abbeel 2007]. Demonstration were provided by an expert pilot teleoperating, i.e. flying, the helicopter to help finding its dynamics and the fitness function corresponding to different maneuvers such as flip, roll, tail-in and nose-in funnel.

Finally, in the work of Lopes et al. [Lopes 2009a], the authors propose to combine imitation and emulation in a unified model by considering a continuum space whose three extreme cases are non-social behavior, emulation, and imitation. A demonstration from a teacher is evaluated according to these three baseline, and the agent final policy is a combination, more precisely a weighted mixture, of the three modules. By varying the weight attributed to each module, they were able to reproduce several well-known social learning experimental paradigms.

Limitations and assumptions

The performance of a learning system is obviously linked with the quality of the information provided by the demonstrations. Among others aspects, if some important state-action pairs have not been demonstrated or if the demonstrations were of poor quality, i.e. including a lot of noise or being suboptimal or ambiguous in certain areas, the learner will be unable to generalize properly from the data.

Unfortunately, in many cases, the demonstration are collected beforehand and sent to the learning algorithm in a batch way which do not allow the robot to have access to better demonstration. A potential solution is to ask the teacher for new demonstrations in those states where demonstration are missing or uncertain [Chernova 2008a, Chernova 2009], we will detail more this approaches in the next chapter.

Another problem is that of identifying what the human is really demonstrating. For example, if a human is demonstrating how to fish to a robot, is the human demonstrating the precise movement of the fishing rod or is he demonstrating where to place the float in order to catch more fish. In other words, should the robot imitate the movement of the fishing rod or should it emulate the position of the float. Where imitation is the act of reproducing the human demonstration in all details, and emulation is the act of fulfilling the same goal as the human.

This problem is currently unsolved in the robot social learning literature and in practice the robot is explicitly told whether to imitate or emulate the demonstration. The problem of understanding what to do from the interaction with human is usually solved at design time; where the system designer applies a multitude of constraints to the interaction with the robot such that no uncertainty or ambiguity remains on the demonstrations. For example, the demonstrated movements are provided in isolation and contain only information about the task to be learned. Similarly, the

robot is explicitly "told" that the demonstrations refer to such and such objects and that it is for example a grasping task. Of course saying that the robot is "told" about the interaction is misleading; it is rather the all system that is constrained to optimize only a specific objective.

In the context of learning from human demonstration, four central questions are often predefined at design time: who, when, what, and how to imitate [Nehaniv 2000]:

The **who** question refers to the problem of identifying who to imitate. It may refer to finding that a person is currently providing demonstrations, but also which person is better at providing accurate demonstrations of the task. This question has not been thoroughly investigated in the literature so far. One of the few work tackling this problem consider a finite set of teacher and select the most appropriate one based on the robot current learning rate [Nguyen 2012]. This method allows the robot learner to take advantage of the different levels of skills each teacher provide.

The **when** question refers to the problems of social coordination between the two partners, such as the turn-taking ability. For example, this aspect has been investigated in human-robot drumming activities where turn taking and role switching are important component of a successful interaction [Weinberg 2006, Kose-Bagci 2008]. The when question also applies for cases where the robot should decide whether to try to imitate its human partner or to explore the environment by itself [Chernova 2009, Nguyen 2012].

The **what** question refers to the problem of identifying the important aspect of the demonstrations. It refers for example to the dilemma between imitation at the action level and emulation at the effect level. At the action level, the aim of the robot would be to reproduce the demonstrator action in the same way and in the same order. At the effect level, the robot should understand the underlying purpose associated to the actions of the human.

The latter problem of identifying the effect level of imitation depends on the context in which the interaction takes place. In particular the concept of affordances [Gibson 1986] — which encode the relation between actions, objects and, effects — is of primordial importance for the robot to be able to reproduce demonstrations at the effect level. Several works have consider affordances for human-robot learning, among others they have been used to recognize demonstrations, decompose them in a sequence of subgoals and finally reproduce them [Lopes 2007a]. Montesano et al. presented a method to learn affordances by interacting with several objects [Montesano 2008]. The robot was able to extract relation between its actions, the objects, and the effects they produces using Bayesian inference methods.

While most of the time the interaction protocol is well constrained such that there is no ambiguity about what aspects of the demonstrations should be imitated, some social cues can be used to infer which parts of a demonstration are relevant, such as the temporal differences of demonstration parts. Pauses during interaction have been linked to important key points in a task demonstration. This allows for example to extract subgoals or determine when a demonstration is completed [Theofilis 2013].

The **how** question refers to the problem of determining how the robot will actually perform the behavior so as to conform to the metric identified when answering the what question. When the demonstration is only relevant at the effect level (emulation) the robot can solve the task by its own mean as soon as the objective is identified. However when the imitation is important at the action level (imitation), differences between robot and human morphology and capabilities makes solving the how question not straightforward. This latter issue has been discussed previously and is referred to as the correspondence problem [Nehaniv 2002], which the problem of mapping between the demonstrator and the imitator.

As stated before, the who, when, what, and how questions are usually skipped over in practical application and the data are provided already pre-formatted for the robot.

In the next subsection we present another paradigm for social learning in robotics, the *learning from human reinforcement* approach. In this paradigm, the human never demonstrates the task to the robot but rather observe the behavior of the robot and reinforces or punishes some of its actions in order to shape its final behavior. We also call this approach learning from human feedback, where feedback implies a positive or a negative assessment of the robot's actions.

1.1.2 Learning from human reinforcement

Learning from human reinforcement, also called shaping, is the process of learning a new skill by receiving assessment over recently performed actions. In this paradigm, the human never demonstrates the task to the robot but he rather observes the behavior of the robot and reinforces or punishes some of the robot's actions in order to shape its final behavior. We also call this approach learning from human feedback, where feedback implies a positive or a negative assessment of the robot's actions. Clicker training [Kaplan 2002] is a subclass of this problem that considers the human can only send positive reinforcement.

Pioneer works in this domain include the work of Blumberg et al. [Blumberg 2002] that trained a virtual dog to learn several sequential tasks and associate them with verbal cues using the clicker training method. Kaplan et al. [Kaplan 2002] applied similar methods to train an AIBO robot dog. Another pioneers work considered a software agent, named Cobot, which interacts with human agents in an online chat community called LambdaMOO. Cobot adapts its behavior from various sources of feedback (reward or punishment) provided by human engaged in the chat community [Isbell 2001].

This social learning paradigm shares many aspects with reinforcement learning [Sutton 1998]. In reinforcement the agent goal is to take actions so as to maximize the cumulative reward. We make a difference between reinforcement learning algorithm and learning from human reinforcement in the sense that the nature of the reward information cannot be treated the same way when a human provides it.

For example, reward signals from humans are frequently ambiguous and deviates from the strict mathematical interpretation of a reward used in reinforcement learning [Thomaz 2008, Cakmak 2010]. We will provide more detail about the teaching behaviors of humans in the next chapter but we note that this problem requires developing new algorithms to monitor and handle the teaching style of each user.

Therefore recent works started to investigate how to additionally learn the way humans provide feedback at the same time as the robot learns the skill [Knox 2009b].

However, as for learning from human demonstration, the robot should be able to answer the who, when, what, and how questions. It needs to infer to which actions the human feedback relates to, but it also needs to differentiate between different levels of feedback as some actions may be mandatory to complete the task while others may just be preferences from the users. In addition the user could make mistakes in its assessment or may not perceive the problem as the robot perceives it, therefore making inconsistent feedback. And as for most learning from demonstration systems, most of the works presented above consider predefined and restricted interaction protocols so as to be able to map easily the human reinforcement with the robot's actions. Similarly, if the human is providing feedback using speech commands, there exist system translating speech utterances into meaningful feedback, e.g. mapping the word "good" to a positive reward.

As stated above, the who, when, what, and how questions are also applicable to learning from human reinforcement. This question is usually skipped over in practical applications by providing already pre-formatted data to the robot.

Providing only reinforcement signals to a robot can be limiting, especially when the state space is large increasing the learning time and resulting in a laboring interaction between the human and the robot. In the next subsection we present another paradigm for social learning in robotics, the *learning from human advice* approach. In this paradigm, the human never demonstrates the task to the robot but rather observes the behavior of the robot and provides hints about what action to perform next, which we will call guidance.

1.1.3 Learning from human advice

Learning from human advice, also called learning from instruction, is the process of learning a new skill by receiving explicit instructions about what to do next. In this paradigm, the human never demonstrates the task to the robot. It rather observes the behavior of the robot and provides clues accordingly in order to shape the robot final behavior. We also call this approach learning from human guidance, where guidance implies that the user explicitly indicates to the robot what action to perform next.

In most scenarios, advices are additional pieces of information improving the learning time and efficiency of an agent. It is therefore often combined with rein-

forcement learning algorithms where the advices influence the exploration behavior of the agent or influence directly the value of particular actions.

In [Clouse 1992] and [Maclin 2005] the teacher can influence the action selection of the agent by providing advices about preferred actions. In [Smart 2002] the trainer directly controls the agent's actions at important key states and let the agent learns the fine details. In [Kolter 2007], the authors introduce a hierarchical apprenticeship learning method for teaching a quadruped LittleDog robot to walk on rough terrains. Their method differs from standard inverse reinforcement learning methods. Rather than providing full demonstrations of the skill, they use human advices about low-level actions of the problem. More precisely, the human expert indicates foot placement in situation where the robot made suboptimal footsteps.

It is important for the robot to be able to generalize to unseen situations. In [Lockerd 2004], the robot Leo learns to switch all buttons on or off from human vocal instructions. When a new button is introduced in the environment the robot autonomously generalizes from the instructions and presses all buttons, instead of pressing only the ones it was instructed to in the first place.

As for other learning paradigms, the robot should be able to answer the who, when, what, and how questions. It should infer which part of the environment matters for the advices, if the advices can be generalized or not to other objects in the environment, if the advices are related to what the robot should do next or what it should have done before, or in which referential are the advices given. Ideally the robot should also keep track of other social signals from the human, such as whether the user is really paying attention to the scene. Or whether the user can see the part of the space the robot is in. Most of the time predefining and restricting the interaction protocols solve these problems.

1.1.4 Discussion

Our categorization of social learning paradigms in three categories does not reflect the many subfamilies that exist inside these categories, including those that are shared among categories. It is meant to situate the social learning problem in a more global picture, providing some interesting pointers for the interested readers.

As we noticed, in most of the above presented work, the human had either no direct interaction with the robot, or few highly constrained interactions. For example, the human demonstrations are provided in a batch perspective where data acquisition is done before the learning phase. In the following section, we detail the usual assumptions made in most human robot interaction scenarios. Based on our observations we then define the global challenges addressed in this thesis.

1.2 Usual Assumptions

As seen in the previous section, there is usually a strong decoupling between the process of extracting useful information from the interaction and the process of

learning a new skill from those information. We can already highlight a chicken and egg problem in the social learning literature:

- On the one hand, if the goal of the robot is to learn a new task, the robot will be fed with the relevant data formatted exactly as needed by the learning algorithm. For example, if a user teaches a robot to navigate in a maze, the protocol of interaction will be fixed to match the need of the algorithm. The interaction will be done turn by turn such as it is easy to associate user's instructions to robot's states. But the user will also be asked to comply to the specific signals the robot understands, such as using the word "right" and "left" to mean respectively "right" and "left".

- On the other hand, when we want to learn the user behavior or the protocol, we assume the task the user wants to achieved is known. It allows interpreting the behavior of the user in light with the known objective he is pursuing. This process is usually called a calibration phase. It is for example necessary to provide the robot with the ability to translate human communicative signals, such as speech or gestures, in a symbolic meaningful representation. For example, if we want our robot to learn which words the human uses to mean "right" and "left", we will ask the user to guide the robot in a maze following a specific path. The robot, knowing the path intended by the human, could identify that the human uses the word "right" and "left" or "droite" and "gauche" to mean respectively "right" and "left".

To summarize, in order to teach a robot a new task, the robot must be able to understand the behavior of the human. But to come up with an understanding of the behavior of the human, the robot must know what is the user overall objective. In this thesis, we present methods to overcome this chicken and egg problem in some specific cases. This allows a user to start interacting with a machine using its own preferred signals; removing the need for a calibration procedure. Before entering into more detail, we introduce the concept of interaction frames.

1.2.1 Interaction frames

An interaction frame [Rovatsos 2001] is a structure that defines all the aspects of the interaction that are pre-defined by the system designer. This interaction schema is assumed to be followed by the human and known by the robot.

The concept of interaction frame is a subclass of the more general concept of frame. Frames are a concept that emerge simultaneously in social theory [Goffman 1974] and artificial intelligence [Minsky 1974]. They represent a schema of interpretation given a particular situation or event. It is answering the question: *what is going on here?*, in order to reduce ambiguity of intangible topics by contextualizing the information. It creates a common ground about the purpose of the interaction [Tomasello 2009, Rohlfing 2013] and includes "predictable, recurrent interactive structures" ([Ninio 1996], p. 171).

In [Rovatsos 2001], Rovatsos et al. presented an extended definition of what an interaction frame might be for artificial agents. While his definition can be transposed to human-robot interaction scenarios, its description and formalization of interaction frames is too much detailed for our forthcoming development. To summarize, an interaction frame provides interactants with guidelines about how to behave (a protocol for interaction). It also allows interactants to understand the communicative intentions of their interaction partner. The interaction frame is often implicitly defined and known in robot learning experiments. We exemplify a few interaction frames and then provide our simplified description of an interaction frame.

Naming frame One example of an interaction frame are found in language games [Steels 2002], where a pre-defined sequence of interaction is defined to associate a name to an object. For example, a human presents an object to a robot and pronounce the name of that object. Being aware of this frame, the robot knows that speech of the human corresponds to the name of the object (and not its shape or its color). In addition the interaction is usually well controlled. The human will first hand the object in front of the robot. The robot will then ask always the same question such as *"tell me the name of this object?"*. Once the robot is ready to accept the human speech utterance, it emits a small noise. Finally the human speaks for one second. Following this sequence, the association between objects and names is guaranteed to be unambiguous.

Feedback frame To exemplify the feedback frame, we introduce the navigation task used for our brain computer interaction experiments chapter 4. In this scenario a human assesses the actions of a virtual agent in a grid world (see Figure 1.1). The human wants to guide the agent towards a specific state. To do so he can send to the agent information about the correctness of its last action. For example, if the robot went away from the target state, the human informs the robot that going North was "incorrect" according to the target. After several interactions the robot is able to identify the goal state. We call this specific interaction scenario a feedback frame. A feedback signal is providing information about the optimality of the robot's last action. A feedback signal can only take two values, "correct" or "incorrect". In practice the interaction is turn taking, the robot performs one action and waits until the human provides a feedback signal. That way the association between actions and feedbacks is guaranteed to be unambiguous.

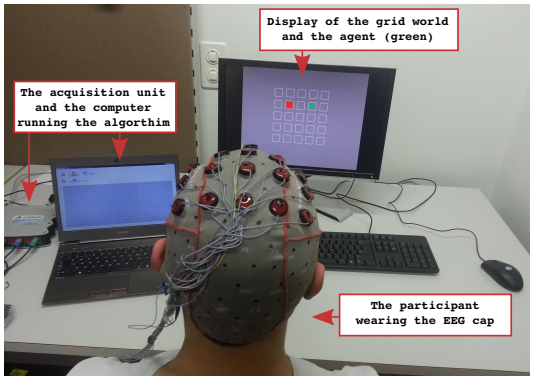


Figure 1.1: The BCI setup for online experiments. On the screen is displayed a grid world with the agent in green.

Guidance frame To exemplify the guidance frame, we introduce the pick and place scenario used in chapter 4. In this scenario a human supervises the work of a robot builder. This robot is able to stack several cubes in order to form different structures (see Figure 1.2). A human wants the robot to build a specific configuration of cubes but cannot directly communicate the high level description of the structure to the robot. The robot only accepts discrete advices about what action to perform next. For example asking the robot to "grasp", "move left", or "release". The robot knows the user's signals correspond to actions it should perform next to fulfill the task. However the robot is not teleoperated and remains the one that selects which action to perform. For example, once the robot understood which cube's configuration the user has in mind, it might build it directly without waiting for further guidance signals. We call this specific interaction scenario a guidance frame. A guidance signal is defined as giving information about what action to perform next. In practice the interaction is turn taking, in some states the robot asks an advice to the human and waits until it receives a guidance signal. That way the association between actions and guidances is guaranteed to be unambiguous.

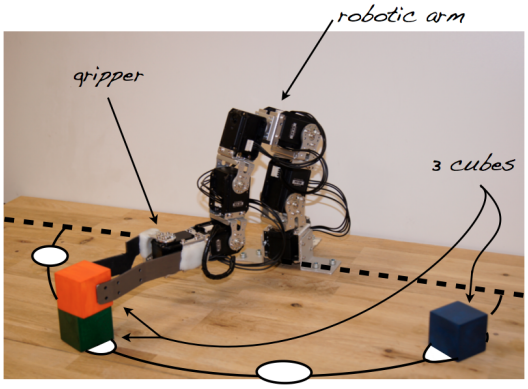


Figure 1.2: A robot builder performing a pick-and-place task with three cubes.

The two latter feedback and guidance frame will be central to the future development of this thesis.

We have seen that interaction frames regulate the interaction between humans and robots. It encodes a way to understand the meanings of the human signals, i.e. their relation with the current context of the interaction. It also includes constraints related to the task, e.g. the human is teaching the robot which room to reach among a finite set of rooms.

In light with our observations, we can list the information provided by the interaction frame:

- **Details and timing of the interaction.** It corresponds to when and how the user will provide instruction signals. For example, the human sends a signal to the robot after every robot's actions. Another example is a human providing a feedback signal between 0.2 and 2 seconds after the robot's action [Knox 2009b].

- **The set of possible meanings the human can refer to.** As depicted before, the set of meaning may include "correct" and "incorrect" for those cases where the user is assessing the robot's actions. It could also be the set of action names when the user provides guidances on what to do next.

- **Constraints on the possible tasks.** The general context of the teaching process is known. For example the robot is aware that the human wants it to reach a specific room in the house, and not to take an object from the fridge. This limits the number of hypotheses the robot can create about what the user has in mind.

Given this information, the interaction frame provides a generic function that, given a context of interaction and a task, returns the meaning intended by the teacher:

$$Meaning = Frame(Context, Task)$$

For example, in a discrete world, if the robot moves from the living room to the kitchen (context), and if the human wants the robot to be in the kitchen (task), then the signal received from the human means "correct" (meaning).

$$\text{“}correct\text{”} = Frame((living\ room \rightarrow kitchen), GoToKitchen)$$

In the following subsection we study how this interaction frame is used in practice. For example, when we want to teach a robot a new task, the task variable is unknown.

1.2.2 Using interaction frames

In the beginning of this section, we identified a chicken and egg problem. In order to teach a robot a new task, the robot must be able to understand the behavior of the human. But to come up with an understanding of the behavior of the human the robot must know what is the user overall objective. In this subsection we explain this problem using our interaction frame formalism.

Calibration: learning the signal to meaning mapping The problem of calibration requires the robot to be able to collect signal-meaning pairs (also called signal-label pairs). Once the robot has access to a dataset of signal-label pairs, it can learn a classifier that given a new signal predicts the meaning associated to this signal. We introduce the decoder function that given a signal, returns a meaning:

$$Meaning = Decoder(Signal)$$

Using our frame definition, to train this decoder the robot must know the task. Following our previous examples, when the robot moves from the living room to the kitchen, it may receive a feedback signal "A". If the task is to go to the kitchen, the robot can infer that the meaning of the signal "A" is "correct". By collecting many of such examples, the robot can learn which meaning correspond to each signal. As a result the robot can build a decoder of user signals. Given a new signal "A" it can deduce:

$$\text{“}correct\text{”} = Decoder(\text{“}A\text{”})$$

Learning: inferring the task The problem of learning the task requires the robot to know how to interpret user's signals. The interaction frame provides the context of the interaction, which includes some constraints about the task. In our example, our robot may know that there are only two rooms in the house.

Given a specific context, e.g. $(living\ room \rightarrow kitchen)$, the robot receives a signal from the user, e.g. "A". And given a decoder trained following the above method the robot knows that the meaning of "A" is "correct".

The robot can compare the meaning received from the user with the one expected from the frame given the two possible tasks:

$$\text{“}correct\text{”} = Frame((living\ room \rightarrow kitchen), GoToKitchen)$$

$$\text{“}incorrect\text{”} = Frame((living\ room \rightarrow kitchen), GoToLivingRoom)$$

Following this method, the robot can infer that the user wants it to go to the kitchen.

Using this simple example highlight the chicken and egg nature of the problem of interacting with machines. To learn the decoder we need to know the task and to learn the task we need to know the decoder. In this work we tackle the problem of learning both the task and the decoder at the same time.

1.2.3 Discussion

By making the interaction frame explicit, we can revise our understanding of the challenges associated to social learning. There is a multitude of challenges that lie between learning human social behaviors and learning new tasks. Identifying and solving these challenges might help us design machines more flexible to loosely defined interaction frames, or even machines that can learn the frames themselves.

Among others, Thomas Cederborg presented an extended reflection on this problem in his PhD thesis [Cederborg 2014a]. He presents a detailed framework to describe robot social learning mechanism [Cederborg 2014b] and propose an extended reflection on how to relax a number of assumptions in robot social learning scenarios.

We present a small sample of questions that arise when we relax some interaction frame assumptions.

A first example is the assumption that human teaching behaviors are consistent. For example, when learning from human reinforcement, humans do not use reinforcement signals as expected by the mathematical formalism of reinforcement learning. For instance, in the work of [Thomaz 2008] the teachers frequently gave a positive reward for exploratory actions or to encourage the robot. In addition, the reward delivered by the human is a moving target, once the human finds the behavior of the robot adequate he will stop delivering rewards. A robot that only seeks to maximize reward could make use of this human bias and generate mistakes on purpose. That way the human will not get use to good performances and will keep generating rewards.

Another usual assumption is that the human has full observability of the robot's actions. An example frequently given in [Cederborg 2014a] is the one of a cleaning robot. The human would like the robot to clean the dust in the apartment during the day. When the human enters the apartment again, if he is happy with the work of the robot he will give it some positive rewards. However, the human user might not be aware that the robot pushed the dust under the carpet or made a lot of noise disturbing the neighbors.

As we described before, in this thesis we remove another type of assumption, that the learner and the teacher share a mutual understanding of the meaning of each other's instructions. In particular the robot is usually assumed to know how to interpret instructions from the user. We define a general scientific challenge: ***Can a robot learn a new task if the task is unknown and the user is providing unlabeled instructions? Which are the constraints and mechanisms that could provide this flexibility in interactive task learning?*** There are two important dimensions in such questions: 1. which are the computational machine learning algorithms and formalisms that are needed for this goal? and 2. how to integrate them within real-world meaningful human-robot interaction such that usability and acceptability can be evaluated in user studies? While we will present experiments with real subjects, given the complexity and novelty of these issues, we focus most of our attention on the first dimension. In the following of this thesis, we call this subclass of problem *learning from unlabeled interaction frames* and we

study how a robot can learn to cope with this lack of information.

1.3 Learning from unlabeled interaction frames

Learning from unlabeled interaction frames corresponds to the problem of learning a task from human instructions signals but where the signal-to-meaning classifier is not given. Nonetheless we maintain important assumptions concerning the interaction protocol and the behavior of the human. We list those assumptions here:

- **The protocol of the interaction.** The human and the robot are able to synchronize together. A signal from the user is easy to map to a state-action pair. In practice this is implemented as a turn taking social interaction. When the robot performs an action in a particular state, it then waits for a signal from the user.
- **The set of possible meanings the human can refer to.** The robot knows the signals from the teacher can take only one meaning out of a finite set of meanings. We will consider only the case of feedback or guidance instruction's signals. When providing feedback signals, the set of meaning includes "correct" and "incorrect". When providing guidance signals, the set of meaning includes the names of the available actions. A meaning will also be called a label to match with the classification algorithm formulation.
- **Constraints on the possible tasks.** The robot knows the context in which the interaction takes place. For example, the robot knows that the user wants it to reach one of the rooms in the house, to create a rhythmic pattern by pressing piano keys, to build a structure by stacking a finite number of cubes, or to grasp an object on the table. This limits the number of hypotheses the robot can create.
- **An interpretation model for each possible task.** The robot has access to a "Frame" function which given a context of interaction and a task, returns the meaning intended by the teacher:

  $$Meaning = Frame(Context, Task)$$

  This function represents a theoretical model of the user teaching behavior given a particular task. It corresponds to the following reasoning: "if the user wants me to perform task T then when I did action A in state S, the user's signal E meant M". However, we remind that the user does not know the task. We further assume this function holds for the full time of the interaction. In other words, the user behavior is consistent throughout the interaction period.
- **The signal to meaning mapping is consistent throughout the interaction period** The user always uses similar signals to mean the same things. Concretely, if the user convey its signals using a two buttons interface to mean

either "correct" or "incorrect", we assume the user is always using one button to mean "correct" and the other to mean "incorrect". But we do not know which button means what in the beginning. We will account for errors in the teaching behavior of the human but we assume that if we ask the user which button means what throughout the game his reply will not change. We note that this assumption is made by all interactive systems.

- **The user's signals are classifiable.** If we had access to a dataset of signal-label pairs from the user, we could compute a decoder that predicts the label of an unobserved signal with more than random accuracy. We note that this assumption is made by all interactive systems.

The fact that the possible set of meanings is known explains the word *"unlabeled"* in the term *"unlabeled interaction frames"*. The robot knows that there is a hidden label — among a finite set of labels — that is associated to each user's instruction signals.

To solve the problem of learning from unlabeled instructions, we will rely on the concept of interpretation hypothesis as introduced in the work of Cederborg et al. [Cederborg 2014b, Cederborg 2014a]. An interpretation hypothesis is the process of interpreting a human signal in light of a hypothetical task and given an interaction frame. As we have seen before, given an interaction frame and a task it is possible to infer the meaning intended by the human in a specific situation. As we have access to constraints about the task we can generate task hypotheses. We can then assign hypothetic labels to every signal received from the human with respect to each possible task. By doing so we create a set of hypothetic datasets of signal-label pairs, one for each task. As the user behavior is assumed to be consistent, the dataset associated to the task the user wants to solve should stand out by having the best coherence between the signals and their hypothetic labels. In others words, the correct task will be the one that explains better the history of interaction.

Solving this problem allows a robot to learn simultaneously what a user wants it to do, as well as the mapping between the human signals and their meanings. As a result, the robot does no have any a priori about which signals it will receive for a particular meaning. As a consequence, people speaking different languages (or using interjections or even hand clapping) could interact with such a system without the need to reprogram it for each particular person.

1.4 Thesis Contributions

The main contribution of the thesis is a method allowing a robot to learn from unlabeled interaction frames. In practice, it allows a user to start teaching a robot a new task using its own preferred teaching signals. For example, let's consider a user providing, using speech, instructions to a robot about what action to perform next. With our method the user is not restricted to a pre-defined set of words and can rather use its preferred words to communicate its advises. The system will learn

simultaneously which words are associated to which meaning, as well as identifying the task the user wants to solve. The user could therefore use words in English, French or Spanish, but also interjections or even hand clapping.

In more detail, we can highlight four important contributions of this thesis:

- We propose a new experimental setup to study the co-construction of interaction protocols in collaborative tasks with humans (conference: [Vollmer 2014a]) (chapter 3). In this setup, an architect and a builder must communicate using a restricted ten buttons channel in order to achieve the joint activity of constructing a structure using simple building blocks. We report experiments with human subjects, which indicates that the kinds of meanings the participants coordinate on is limited to a specific subset. This subset is composed of feedback ("correct", "incorrect"), guidance ("left", "right", "assemble"), feature based ("red", "small"), or global ("end", "reset") instructions. Especially most of the users seem to concentrate on feedback instructions. Finally we report that humans solve the problem by projecting the interaction into different common interaction frames.

- We present an algorithm allowing to simultaneously learn a new task from human instructions as well as the mapping between human instruction signals and their meanings (conferences: [Grizou 2013c, Grizou 2014b, Grizou 2014a], workshops: [Grizou 2013a, Grizou 2014c] (chapter 4).

  Our method consists of generating interpretation hypotheses of the teaching signals with respect to a set of possible tasks. We will see that the correct task is the one that explains better the history of interaction. We demonstrate the efficiency of our method in a pick and place scenario where a teacher uses spoken words to instruct a robot to build a specific structure. We show that our method works if the teacher provides feedback ("correct" or "incorrect"), or guidance ("left", "right", ...) instructions. Finally we show that our system can reuse the knowledge acquired about the signals of the users to learn a second task faster.

- We propose a measure of uncertainty on the joint task-signal space that takes into account both the uncertainty inherent to the task, which is unknown and remains to be estimated, as well as the uncertainty about the signal to meaning mapping, which is also unknown and remains to be estimated. We use this measure of uncertainty to optimize the action selection of our agent, which improves significantly the learning time (conferences: [Grizou 2014b, Grizou 2014a]) (chapter 5).

- We apply our algorithm to brain computer interfaces (BCI) (conference: [Grizou 2014b], workshop: [Grizou 2013b]) (chapter 6). We present experiments where several subjects control an agent from scratch by mentally assessing the agent's actions and without requiring a calibration phase to train a decoder of the user's brain signals. In all experiments, our algorithm was

able to identify a first task in less iteration than a usual calibration procedure requires.

We believe the theoretical and empirical work presented in this thesis can constitute an important first step towards flexible personalized teaching interfaces, a key for the future of personal robotics.

1.5 Thesis Outline

The first aim of this manuscript is to explain the problem of *learning from unlabeled interaction frames* and to provide an intuition on what properties can be exploited to solve this problem. We will introduce the most important aspects of the work by simple visualization of the problem and of the specific properties we exploit. Our objective is therefore to endow the interested readers with sufficient understanding of the problem to implement their own version of the algorithm with the tools they are more familiar with.

In chapter 2, we present an overview of the related work which span from language acquisition to brain computer interfaces.

In chapter 3, we introduce a new experimental setup to study the co-construction of interaction protocols in asymmetric collaborative tasks with humans. By presenting our results based on this setup, we draw interesting lessons for our problem. This work on human experiment is a joint collaboration with Anna-Lisa Vollmer and Katharina J. Rohlfing.

In chapter 4, we introduce in more specific terms the problem and provide a visual intuition on what properties we will exploit. We continue by formalizing the problem in a probabilistic framework, describe how each subcomponent of our algorithm are implemented and present results from a robotic pick and place scenario.

In chapter 5, we introduce the planning specificities related to our problem and provide a visual intuition on what properties we should track. We then define the uncertainty measure used planning the actions of our agents. Finally, we demonstrate on a 2D grid world problem the efficiency of our planning method with respect to other planning strategies.

In chapter 6, we present an application of the algorithm to a BCI scenario where human subjects control a virtual agent on a grid. We report online experiments showing that our algorithm allows untrained subjects to start controlling a device without any calibration procedure by mentally assessing the device's actions. This work on BCI is a joint collaboration with Iñaki Iturrate and Luis Montesano.

In chapter 7, we discuss and provide algorithmic solution to a number of limitations. The limitations include the use of a discrete state space, the need for a finite set of task hypotheses, and the fact that the interaction frame is defined in advance. We further propose a proof for our algorithm in restricted conditions.

Code is available online under the github account `https://github.com/jgrizou/` in the following repositories: lfui, experiments_thesis, and datasets.

Chapter 2

Related Work

Contents

In most robot social learning experiments today, there is a strong decoupling between the process of extracting useful information from the interaction and the process of learning a new skill from these information. For example, the human demonstrations are provided in a batch perspective where data acquisition is done before the learning phase. The properties of teaching interactions with a human in the loop are not yet considered in depth.

In this chapter we highlight the difference between systems learning from well-controlled interactions and systems trying to close the interaction loop allowing more flexibility in the interaction process. These issues have began to be addressed in a subfield called *interactive learning* which combines ideas of social learning with extrinsic and intrinsic motivated learning. With this approach, the robot acquires more autonomy with respect to how to deal with the human in the loop.

After presenting the related work in interactive learning, we broaden the scope of this work by linking with the computational modeling of language, some aspects of unsupervised learning, and specific works on ad-hoc team whose stated challenge is to enable cooperation without prior-coordination in multi-agent scenarios. Finally, we present related works from the brain computer interfaces (BCI) community.

2.1 Interactive Learning

In this section, we present a number of works considering the human component into the learning loop. We call this area of research *interactive learning* [Nicolescu 2003, Breazeal 2004]. It aims at developing machines that can learn by practical interaction with the user.

Interactive learning combines ideas of social learning with extrinsic and intrinsic motivated learning. It differs from the works presented in the introduction as both the human and the robot are simultaneously involved in the learning process [Kaplan 2002, Nicolescu 2003, Breazeal 2004, Thomaz 2008]. Under this approach, the teacher interacts with the robot and provides extra feedback or guidance. In addition, the robot can act to improve its learning efficiency or elicit specific responses from the teacher. Recent developments have considered: extra reinforcement signals [Thomaz 2008], action requests [Lopes 2009b], disambiguation among actions [Chernova 2009], preferences among states [Mason 2011], iterations between practice and user feedback sessions [Judah 2010] and choosing actions that maximize the user feedback [Knox 2009b].

We decided to split this related work in four categories. Firstly, we present works combining multiple sources of information, such as combining demonstration and feedback. Secondly, we present some studies about the behavior of human when teaching robots. Thirdly, we present works that try to model some aspects of the user behavior or of the protocol. Fourthly, we present works considering an active robot, which try to learn faster from or about the interaction. Finally, we discuss and situate our work in this scope.

2.1.1 Combining multiple learning sources

Researchers have considered mixing different learning paradigms in order to improve the quality of the interaction and of the learning process. They considered:

- Mixing environmental rewards with human rewards [Knox 2010, Griffith 2013, Grave 2013]. The main problem is to balance the influence of the environmental reward with the human generated reward.

- Iterations between practice and user feedback sessions [Judah 2010]. The learner first practices the task a few times to learn from environmental reward. Then a user can observe its practice session and classify the policies or actions as good or bad. The learner updates its policy given the reward from the environment and the user critiques, and the process repeats again.

- Giving some demonstrations first, and having the robot practicing the skill under online human supervision (feedback or guidance) [Nicolescu 2003, Pardowitz 2007].

- Mixing concrete instructions and rewards to balance human efforts with communication efficiency [Pilarski 2012].

- Combining learning from demonstration and mixed initiative control [Grollman 2007a]. Mixed initiative control is when the control can transition smoothly from the demonstrator control to the robot control. In [Grollman 2007a] the authors used this method to teach different behaviors to a robot, such as mirroring the head position with the tail position or to seek for a red ball, using the same algorithm.

- Combining transfer learning, learning from demonstration and reinforcement learning [Taylor 2011].

- Demonstrating only parts of trajectories. In [Akgun 2012], the users only demonstrate some keyframe positions along the trajectory. The robot can then autonomously infer a trajectory that match with each keyframe position.

But researchers also created new learning paradigms, such as learning from users' preferences [Mason 2011, Akrour 2011]. In this new paradigm, the system learns the preferences of the human and will pro-actively generalize and apply them autonomously.

In [Mason 2011], the user starts by teleoperating the robot and can mark some states as good or bad. From this data, the robot can create a user profile. Next, the robot can select its own goal without the need for human teleoperation. Once a desirable state of the world has been reached, the human has a possibility to classify the state as good or bad again. The robot can update its user profile, and the process iterates.

In [Akrour 2011, Akrour 2012, Akrour 2014, Wilson 2012], the robot demonstrates some candidate policies and asks the human to rank them by preferences. Based on this ranking the algorithm learns a policy scoring function, which is later used to generate new policies. The user ranks these new policies again, and the process iterates. This method differs from the learning from human reinforcement paradigm as the user evaluates full demonstrations. It differs from inverse reinforcement learning because the robot is it-self generating the demonstrations. But more importantly, demonstrations are ranked between them, which differs from the usual assumptions that all demonstrations given to the learning algorithm are equally correct but noisy.

Most of the methods above consider the users are somehow optimal or at least predictable in their teaching behaviors. However this is not always the case, in next subsection we review studies about the behaviors of humans when teaching robots.

2.1.2 How people teach robots

An important challenge is to deal with non-expert humans whose teaching styles can vary considerably. Users may have various expectations and preferences when interacting with a robot and predefined protocols or instructions may bother the user and dramatically decrease the performance of the learning system [Thomaz 2008, Kaochar 2011, Knox 2012, Rouanet 2013]. These studies show that even when using

well-defined protocols, it is important to consider how different instructions can be used for learning.

People will not always respect predefined conventions. Several studies discuss the different behaviors naive teachers use when instructing robots [Thomaz 2008, Cakmak 2010]. When learning from human reinforcement, an important aspect is that the feedback is frequently ambiguous and deviates from the mathematical interpretation of a reward or a sample from a policy. For instance, in the work of A. L. Thomaz et al. [Thomaz 2008] the teachers frequently gave a positive reward for exploratory actions even if the signal was used by the learner as a standard reward. Also, even if we can define an optimal teaching sequence, humans do not necessarily behave according to those strategies [Cakmak 2010]. This is often because the user and the robot do not share the same representation of the problem.

For the specific case of learning from human reinforcement, several works studied how people actually teach by explicit reward and punishment. In [Thomaz 2006], the authors found that people gave more positive than negative rewards. Also, users tend to use feedback signals to provide guidance to the agent and to encourage the agent in its exploratory actions. In [Knox 2009a], the authors show that humans reinforce almost always state-action pairs and not state only. People perceive intentionality in the robot's actions, and therefore human trainers reinforce given the expected long-term returns of an action, i.e. they do not provide a solely immediate reward as reinforcement learning algorithms rely on. Human teachers reinforce what the robot is about to do (they perceive intentionality) or what the robot just did. Therefore the question of how to divide human feedback between future and past actions is not obvious. In addition, human reinforcement behavior is a moving target and cannot be considered as sampled from an immutable hidden reward function. Finally, in [Loftin 2014], the authors studied the role of non-explicit feedback. Some users do not always give explicit feedback in response to a robot's action. For example, they have shown that some users are more likely to provide positive feedback than negative feedback. Surprisingly, some users might never give positive feedback. This variety of user profiles makes it difficult to create a general algorithm for learning from human reinforcement. However, if the users are consistent in their strategies, it might be possible to model and exploit them individually.

Given these observations, considering people as optimal teaching agents seems flawed. Every user may not experience what is optimal for a robot, in a mathematical sense, as optimal. And more importantly each user might experience it differently. There are a number of design principles that have been derived from such experiments to create better interactive learning systems.

Transparency It is for example important for the user to understand the way the robot "thinks" and what are its "intentions". A learner displaying its current "state of mind" is called a transparent learner [Thomaz 2008]. A simple example would be a robot that displays its current level of understanding of the task using a colored LED. The robot could also directly vocalize its understanding of some part of the

problem, or if it does not understand some words from the teacher [Chao 2010]. An other option for the robot is to demonstrate what it understands so far while asking for confirmation or correction to the user [Cakmak 2012b].

Also it may be useful to characterize the preferences of users in terms of teaching behavior. In [Cakmak 2012b], Cakmak et al. used human-human experiments to find out which types of question were most often used. Based on their observations, queries about features of the problem were identified as the most common questions. They were also perceived as the smartest when used by the robot. Using this method the robot explicitly tests precise aspects of the task and asks to the teacher: "can I do that?".

Controlling the leader/follower balance Asking feedback from the user is more useful when it allows to differentiate ambiguous states. In [Chao 2010] active learning is shown to improve the accuracy and efficiency of the teaching process. However active learning may illicit undesirable effects of acceptability by affecting the leader/follower balance during the interaction. In [Chao 2010], some people felt uncomfortable when the robot asked too many questions and did not feel like they were the teacher, i.e. the one leading the interaction. As a conclusion, the interaction is best accepted when a proper balance is achieved between autonomy, feedback request and human control. A robot asking a question every step is boring for the user, and asking too infrequently is unpredictable. Finally, allowing users to send feedback to the robot whenever they wanted was preferred by the users but was less efficient for the learning process.

Testing the robot As a kind of transparency, it is important for the teacher to be able to ask the learning agent to perform the taught skill to verify and correct it. It allows the user to understand how the agent learns and generalizes from examples. For instance, in [Kaochar 2011] when the participants had the opportunity to test the agent's comprehension, more than half of them preferred testing the student systematically after a new concept or procedure was introduced. They also showed that people tend to test the agents more during the last third of the teaching process.

To summarize, all teachers are different and most of the time they are not optimal. Even if there are a number of design principles allowing reducing the variability of human teaching behaviors, it is almost impossible to design an experiment where human teaching behavior can be fully predictable. Therefore modeling the users seems a natural next step.

2.1.3 User modeling, ambiguous protocols or signals

Modeling the user during the interaction is primordial to adapt to an a priori unknown human. Some works investigate how to learn the user's teaching behavior

online [Knox 2009b], how to learn the meaning of new human signals starting from a set of known signals [Lopes 2011, Loftin 2014], or how to directly learn the meaning of unknown signals but when the agent has access to a direct measure of its performance [Branavan 2011, Kim 2012, Doshi 2008].

In [Knox 2009b], an artificial agent learns from human reinforcement but the human signals are not treated as a reward in a reinforcement learning problem. Instead the agent models the trainer reinforcement function, and considers it as a moving target. The idea is that the human reinforcement already includes the long-term consequences of the agent's actions, whereas in reinforcement learning the reward act just locally. Therefore, by modeling the user reinforcement function, the agent can act greedily on this function to achieve the desired task. Their approach has been extended to continuous states and actions [Vien 2013].

In [Lopes 2011], the learning agent receives signals of both known and unknown meanings. The agent learns a task using the known information and is then able to infer the associated meaning of the a priori unknown signals. Similarly in [Loftin 2014] the agent learns the meaning of non-explicit signals, e.g. when the user does not press any button, but knowing the meaning of all explicit signals. Our problem differs because we do not have access to a subset of signals of known meaning beforehand.

In [Branavan 2011], the learning agent automatically extracts information from a text manual to improve its performance on a task. The agent learns how to play the strategy game Civilization II and it has access to a direct measure of its performance. But the agent also has access to the game manual, which gives some explanation about the game strategy. However the agent does not know how to read and interpret this manual beforehand. The agent then autonomously learns to analyze the text in the manual and to use the information contained in the manual to improve its strategy. In other words, the agent learns the "language" of the game manual. While the agent could learn to play the game alone, their results show that *"a linguistically-informed game-playing agent significantly outperforms its language-unaware counterpart"*. Our problem differs because our agent does not have access to a measure of its performance on the task, and can only rely on the unlabeled signals received from the teacher. However we will process much simpler signals without syntactic structure.

Some other works have focused on learning semantic parsers, either from natural language as text [Branavan 2011, Kim 2012] or real speech [Doshi 2008]. Semantic parsers allow for a more natural human-robot interaction where more advanced set of instructions can be used. In [Kim 2012] the algorithm can produce, with some limitation, previously unseen meaning representation. However these works assume the agent has access to a known and constrained source of information about the task. Either a direct access to its performances [Branavan 2011], to a reward from a teacher [Doshi 2008], or to a tuple (text instruction sentence, state, action sequence) where the instruction describes at a higher level the observed action sequence [Kim 2012].

Modeling parts of the user behavior allows an interactive learning agent to adapt to a variety of teaching behaviors. The work presented in this thesis follows along the same lines. We learn mapping between the user's teaching signals and their meanings. But contrary to the works presented above, we simultaneously estimate the desired task, and do not have access to a measure of our performance on the task or to other known sources of information. It allows a user to teach a machine a new task using signals unspecified in advance. As a consequence, if speech is the modality of interaction, our system should handle different languages or even interjections or hand clapping.

2.1.4 Active learners and teachers

Finally another crucial aspect for an efficient interaction is to have both a learner and a teacher seeking to maximize the learning of the learner. We usually call these types of agent *active learners* and *active teachers*. An active learner will seek for situation in which it feels uncertain about what to do, and ask the teacher for more information about that situation. An active teacher will try to provide the most useful demonstrations or instructions to the learning agent. Ideally an active teacher considers the learning capabilities of the learner to adapt its teaching behavior.

Active learners The interested reader can refer to [Lopes 2014] for a review of active learning for autonomous intelligent agent. In the following paragraphs, we only focus on active learning agents in social interactive learning conditions. The notion of uncertainty is often used in active learning algorithm. Uncertainty refers to situation where the agent does not know how to behave in order to fulfill the task. By collecting more information about that situation, the agent should reduce uncertainty and increase its performance on the task.

A number of previously presented works already includes an active component to their agents. For example, in [Lopes 2011], the agent is more efficient at learning both the task and the meaning of new signals when seeking for uncertain state-action pairs. In [Judah 2012], the authors consider active imitation learning. Instead of passively collecting demonstrations from the user, the learning agent queries the expert about the desired action at specific states.

In [Chernova 2009], the authors propose to balance autonomy and demonstration request using a confidence estimate, measured by the uncertainty of the classifier. The robot asks for demonstration only in states it is unsure about what to do. Otherwise the robot acts autonomously but can still be corrected by the user at any time. A problem with this approach is that the information on the dynamics of the environment is not taken into account when learning the policy. To address this issue, Melo et al. [Melo 2010] includes the information of the environment dynamics. They use the method proposed by Montesano et al. [Montesano 2012] to make queries where there is lower confidence of the estimated policy.

Active learning has been considered inside the inverse reinforcement learning framework [Lopes 2009b]. Once a set of demonstration has been observed, it is possible to compute the posterior distribution of reward that explains the teacher behavior. By taking a query by committee approach, the agent can disambiguate among probable reward functions by asking the teacher the correct action in an uncertain state. An interesting extension of this work is to query the correct action for states whose expected uncertainty reduction of the global uncertainty is maximal [Cohn 2010, Cohn 2011], instead of considering only the local uncertainty [Lopes 2009b]. Also, instead of asking the optimal action for a given state (action queries), the learner could directly ask about the reward value at a given location (reward queries) [Regan 2011]. Finally, reward queries and action queries can also be combined [Melo 2013].

Active teachers An active teacher tries to provide demonstrations or instructions that will make the learning process more efficient for the learning agent.

In [Cakmak 2012a], the authors study how a teacher can optimally provide demonstrations for a sequential problem. Concretely, the teacher should find the smallest sequence of examples that allow the learner to identify the task. Their optimal teaching algorithm allows a much faster convergence in all four presented tasks. Similarly in [Torrey 2013], the teacher has a limited number of advises to give and the authors study how to best use these advises to improve the learning gain of the learning agent. They showed that advices could have greater impact when they are spent on important states, or to correct agent's mistakes.

Active teaching finds applications in several domains, especially in the educational one, where giving individual advises for each student given their individual proficiency may improve the collective learning gain of a classroom. For example, in [Clement 2014] the authors present an *intelligent tutoring systems* which *"adaptively personalizes sequences of learning activities to maximize skills acquired by each student"*. They take into account constraints about the limited time and motivation resources of each student. Their approach seeks at optimizing the learning gain of students, by selecting the exercises that should make the student progress best.

In chapter 5 we will present an active version of our algorithm. As for other works, our active learner will seek at reducing uncertainty by reaching states of maximal uncertainty. However, our uncertainty measure differs from previous works in that both the task and the signal to meaning mapping is unknown at start. Therefore there is uncertainty both at the task and at the signal level, which required developing a new uncertainty measure specific to our problem.

2.1.5 Discussion

In this section we discovered a number of works dealing with the human teacher inside an interaction loop. We have seen that information coming from a human

teacher cannot always be considered as optimal or following simple mathematical rules. Moreover as each user is different, current research are advancing toward modeling the user teaching behavior during the interaction. Yet to model some aspects of the user, the robot is assumed to have access to an explicit known source of information about either the task or the meaning of some signals.

In this thesis, we want to learn from unlabeled interaction frames. It means that the robot will not know the meaning of the signal it receives, neither the particular task it should achieve. However the robot is already equipped with a theoretical model of the human teacher, and is able to deduce the meaning the user should send given a specific context (state-action pair) and a specific task. Moreover the user is assumed to be consistent, i.e. a user behavioral model is provided to the robot.

Our two latter assumptions are conflicting with the observations about the behavior of human teachers presented in this section. To account for variability between users, we will simply introduce a noise parameter in our models. In chapter 7, we soften the assumption that the robot is equipped with a theoretical model of the human teaching behavior.

Finally we will consider an active learning agent and present in chapter 5 a new uncertainty measure that takes into account both the uncertainty about the task and the uncertainty about the signal to meaning mapping.

2.2 Language Acquisition

While this is not the main target of this thesis, this work is also relevant with regards to the computational modeling of language acquisition. The general question of how certain sub-symbolic communication signals can be associated to their meanings through interaction has been largely studied in the literature. But the specific question of how teaching signals (e.g. speech words) can be mapped to teaching meanings, and how they can be used for learning new tasks, has, to our knowledge, not been computationally modeled.

The literature on the computational modeling of language acquisition by machines and robots is large and diverse, and focused on many aspects of language learning [Steels 2012a, Steels 2002, Cangelosi 2010, Kaplan 2008, Steels 2003, Brent 1997, Yu 2007]. An important line of work investigated the Gavagai problem [Quine 1964], i.e. the problem of how to guess the meaning of a new word when many hypothesis can be formed (out of a pointing gesture for example) and it is not possible to read the mind of the language teacher. Various approaches were used, such as constructivist and discriminative approaches based on social alignment [Steels 2007, Steels 2008a], pure statistical approaches through cross-situational learning [Xu 2007, Smith 2008] or more constrained statistical approaches [Roy 2005, Yu 2007]. In all these existing models, meanings were expressed in terms of perceptual categories (e.g. in terms of shape, color, position, ...) [Steels 2007, Steels 2008a, Yu 2007], or in terms of motor actions [Steels 2008b, Massera 2010, Sugita 2005]. This applies to models implemented in

robots, such as in [Heckmann 2009], where the robot ASIMO is taught to associate new spoken signals to visual object properties, both in noisy conditions and without the need for bootstrapping.

2.2.1 Language games

The work of Steels and colleagues [Steels 2012a, Steels 2002] have extensively shown the importance of language games, instantiating various families of pre-programmed interaction frames specifically designed to allow robots to learn speech sounds [De Boer 2000, Oudeyer 2006], lexicons [Steels 2002] or grammatical structures [Steels 2007, Steels 2008a]. Other works used similar interaction protocols to allow a structured interaction between humans and robots so that new elements of language could be identified and learnt by the robot learner [Roy 2002, Lyon 2012, Cangelosi 2006, Yu 2004, Cangelosi 2010, Sugita 2005, Dominey 2005, Cederborg 2011]. In particular, it was shown that these interaction protocols fostered efficient language learning by implementing joint attention and joint intentional understanding between the robot and the human [Kaplan 2006, Yu 2005, Yu 2007], for example leveraging the synchronies and contingencies between the speech and the action flow [Rohlfing 2006, Schillingmann 2011].

Most of the existing models study communicative signals whose meanings were expressed in terms of proper names, color and shape terms, motor actions, or body postures. Only very few models so far have explored how other categories of word meanings could be learned. Cederborg et al. presented a model where word meanings expressed the cognitive operation of attentional focus [Cederborg 2011]. Some models of grammar acquisition dealt with the acquisition of grammatical markers which meaning operates on the disambiguation of other words in a sentence [Steels 2012c]. Spranger et al. studied how a spatial vocabulary and the concepts expressed by it can emerge in a population of embodied agents from scratch. They considered the emergence of various spatial language systems, such as projective, absolute and proximal [Spranger 2012b, Spranger 2013], of spatial relations, such as landmarks [Sprangler 2013], and of basic spatial categories such as left-right, front-back, far-near or north-south [Spranger 2012a]. Finally, the Lingodroid project [Schulz 2010] used robotic rats (called iRats) as embodied agent to study the emergence of geopersonal spatial language and language for time event (such as day-night cycle) in a population of robots. iRats were equipped with shared attention mechanism, they could measure the light level and they were able to build their own map of the environment. Pairs of robots could play a meet-at and meet-when game. By repetitively playing the game, the robots population agreed on specific terms for spatial communication and time of the day, such as the concept of morning or afternoon [Schulz 2011, Heath 2012]. These concepts of morning and afternoon were changing with the season according to the lightning cycle and allowed robot to synchronize their behavior based on relative cyclic time rather than an absolute notion of time or a calendar.

Language games usually consider a direct relation between the communicative

signals and the environment. For example, the agents learn to associate names to objects, colors, spatial relations, or time events. The problem considered in this thesis will consider more abstract relation between the communicative signals and their meaning, such as whether the past action of one agent was "correct" or "incorrect" with respect to a global objective. Or if the agent should have move "left" or "right" to get closer to the goal. While there is no specific limitation from our work to handle typical language game scenarios, most of the methods presented above have not been applied to the more abstract relation considered in this thesis. Finally most of the works presented so far consider a rather rigid interaction protocol between agents, where the communication goal is often defined before hand. For example, when playing a meet-at or a meet-when game, the iRat robots are aware that the communicative signals respectively refer to a location on the map or to a time event as measured by their light sensors.

In the next subsection, we highlight the work of Cederborg et al. [Cederborg 2011] that, to our knowledge, is the closest work in language acquisition considering a setup similar to the problem of *learning from unlabeled interaction frames*.

2.2.2 Work of Thomas Cederborg et al.

In this subsection, we present the work of Thomas Cederborg as published in [Cederborg 2011, Cederborg 2013] and in the chapter 6 of his thesis manuscript [Cederborg 2014a]. This work has been categorized in the language acquisition field by the authors but it has wider application especially in human-machine interaction. As we will discuss in the following paragraphs, this work is strongly related with our problem of *learning from unlabeled interaction frames* and the solution proposed to their problem is closely linked with the algorithm proposed in this thesis.

In [Cederborg 2011], Cederborg et al. *"show that it is possible to simultaneously learn never before encountered communicative signs and never before encountered movements, without using labeled data, and at the same time learn new compositional associations between movements and signs"*. They present an experiment where a robot learns to produce appropriate gestures in response to the communicative signals of one human, called an interactant. To do so, the robot can observe another human, called the demonstrator, which already knows how to interpret the interactant signals and produce the corresponding gestures. The interactant always provides two consecutive symbolic signals, one is associated to a type of gesture (e.g. drawing a triangle or a circle) and the other is associated to a drawing referential (e.g. red, blue or green object). The demonstrator, which knows how to interpret the interactant symbols, can then demonstrate the appropriate task, for example drawing a circle around the blue object. The robot observes both the interactant signals and the demonstrator trajectories and learns both the meaning of the communicative signals of the interactant and how to respond to them.

This setup is closely related with our problem of *learning from unlabeled interaction frames* as both the task and the signal to meaning mapping are unknown

at start. A number of differences can be listed: a) the robot is not active in the learning process and passively observes the interactant and the demonstrator, b) the robot has access to full demonstrations of the task, and c) the association between the task and the signals is direct, whereas in the scenario considered in this thesis the meaning of the signals are more abstract and for example refer to whether the action was "correct" or "incorrect" with respect to the aimed task. However their setup requires to learn the meaning of two symbolic communicative channels (type of gesture or drawing referential), as well as the particular signal to meaning mapping within each channel (triangle/circle and red/blue/green). The problems we tackle in this thesis only consider one channel of communication. In addition their agent can learn the gestures and generalize reproduction in other coordinate systems given previously unseen combination of interactant signals. In this thesis, we will also demonstrate how our agent can reuse their knowledge about the interactant signals to learn new tasks faster.

But the most interesting aspect of their work lies in the introduction of interpretation hypothesis. Even if not explicitly named that way in their early work [Cederborg 2011], the terms of interpretation hypothesis was central to the thesis of Thomas Cederborg [Cederborg 2014a] and it is also a central concept in the present thesis. An interpretation hypothesis is the fact of systematically interpreting or evaluating the observed data with respect to a set of hypotheses. In their work the hypothesis set corresponds to the referential of the demonstrated trajectories, unknown at start but known to belong to a finite set of possible referential (e.g. there is only three objects). By making the hypothesis that each trajectory refer to each of the referential (see Figure 5 of [Cederborg 2011]), they can find out which gesture belong to which referential and which trajectories are of the same type (see Figure 6 of [Cederborg 2011]). Similar ideas are pushed forward in this thesis, however we note that in the work of Cederborg et al. the agent was first grouping the trajectories per type and only then was able to identify the meaning of the communicative signals of the interactant. In our work, the process of learning the task is not differentiable from the process of learning the signal to meaning mapping.

We will summarize the similarities and differences between the work presented in this thesis and several works presented in this chapter in section 2.6.

2.2.3 Semiotic experiments

In this subsection, we briefly introduce the field of experimental semiotics, and briefly introduce our experimental scenario that study how human can deal with the problem of *learning from unlabeled interaction frames*. More details will be provided in chapter 3.

The ability to learn from unlabeled interaction frames might seem to be an artificial and unrealistic scenario made up for practical purposes in human-machine interaction. Yet, this capability is crucial in infant social development and learning, as well as in adult mutual adaptation of social cues. This has been the subject of experiments in experimental semiotics [Galantucci 2009].

The field of experimental semiotics studies the emergence and evolution of communication systems [Galantucci 2009]. Instead of computer simulations as presented in previous subsections [Cangelosi 2002, Steels 2012b], controlled experiments in laboratory settings are designed to observe communication between human participants who perform joint tasks. For instance, Galantucci et al. showed that pairs of participants performing a joint task could coordinate their behaviors by agreeing on a symbol system [Galantucci 2005].

Most experimental semiotics studies developed to study joint action involve symmetric communication (cf. [Galantucci 2011]), where both participants are able to send and receive communicative signals. In this thesis, we study asymmetric communication where only one of the two partners can send signals. To our knowledge two semiotic studies have considered asymmetric communication [De Ruiter 2010, Griffiths 2012].

The work conducted by Griffiths et al. [Griffiths 2012] is more directly related to our problem of *learning from unlabeled interaction frames*. They explore a human-to-human interaction in a categorization task where instructions can only be provided via six unlabeled symbols (thus the meaning of teaching signals are unknown to the learner). The learner has however access to some environmental reward on its performance on the task. This study shows that tutors seem to spontaneously use three main types of instruction in order to help the learner: positive feedback, negative feedback, and concrete instructions (e.g. name of next optimal action).

In chapter 3, we will present our experiment setup which is a variant of the work of Griffiths et al., where teaching signals are unknown at start, sub-symbolic and not from a pre-determined set. However in our experimental scenario it is impossible for the learner to perform the task without understanding the communicative acts of the teacher. By removing access to an environmental reward to the participants, the learner is no more able to improve its understanding of the task independently of the understanding of the teaching signals; which makes our experiment more suited to study how humans deal with the problem of *learning from unlabeled interaction frames*. Astonishingly, even with such unconstrained interaction, we will see that most participants agreed on a communication system and succeeded in solving the task.

2.3 Multi-agent interaction without pre-coordination

As robots are moving into the real world, they will increasingly need to group together for cooperative activities with previously unknown teammates. In such ad hoc team settings, team strategies cannot be developed a priori. Rather, each robot must be prepared to cooperate with many types of teammates, which may not share the same capabilities or communicative means. This challenge of multi-agent interaction without pre-coordination (MIPC), also called the pickup team challenge [Gil Jones 2006] or the ad-hoc team challenge [Stone 2010a], states that agents should learn to collaborate without defining pre-coordination schemes and/or with-

out knowing what the other agents will be capable of [Bowling 2005, Gil Jones 2006, Stone 2010a]. The ad-hoc team challenge is specific to scenarios where one agent is removed from a working and synchronized team, and replaced by a new agent, called the ad-hoc agent, which never interacted with the team before [Stone 2010a].

A prototypical example is the one of a street soccer team. Such team is composed of players coming from different areas of a city, with different soccer skills, different preferences in terms of placement on the field, and even different ways of communicating game strategies. Yet such teams are quickly formed and functional in a matter of minutes. MIPC aims at creating agents solving similar problems. Among others, researchers in the field have considered soccer teams scenarios[Bowling 2005], treasure hunting tasks [Gil Jones 2006], bandit problems [Barrett 2013a], and the pursuit domain [Barrett 2011b].

This area of research is still in its early stages and the full challenge of MIPC is difficult to tackle directly. Researchers have started investigated only certain aspects of the larger problem by making suitable assumptions. The most common assumption is that all agents on the field share a common objective, i.e. that all agents are partners towards achieving the same task [Barrett 2011b]. In [Bowling 2005, Gil Jones 2006] all agents follow complex pre-specified plans where each agent can be attributed a role to which is associated synchronized action sequences. In [Stone 2010b, Stone 2013], the ad-hoc agent knows the behaviors of the other agents and are assumed to be fixed (i.e. other agents do not learn).

There are different roles an ad-hoc agent can play in the team:

- A first scenario is when the new agent knows the environment and the task to achieve. In this case, the ad-hoc agent must influence the other agents to achieve the correct task. For example, in [Stone 2010b, Stone 2013], an ad-hoc agent should influence other agents' behaviors such that the team gets more payoffs or to guide the other agents towards specific states. This ad-hoc agent cannot communicate directly with the other agents. However the other agents' behaviors are known and are influenced by the ad-hoc agent actions. The problem is therefore to find the correct sequence of actions that may lead the other agents towards the correct states, resulting in a higher performance on the task.

- A second scenario considers that all agents share the same goal, but the new ad-hoc agent does not know a priori the behaviors of its partners. To help solving the task, the ad-hoc agent should learn other agents' behaviors and selects its actions accordingly [Barrett 2011a, Barrett 2011b, Barrett 2013b]. For example, in [Barrett 2011b] the ad-hoc agent should help its teammates catch a prey and is more efficient when trying to understand the behavior of the other agents. Often to make this problem feasible, it is assumed that the other agents sample their latent policy (or type) from a finite set. The ad-hoc agent then only has to learn to match each agent with its true model. In [Albrecht 2014], the authors analyzed convergence properties of this kind of

scenario. But sometimes, the other agents are totally unknown to the ad-hoc agent. For example, in [Barrett 2011b] the ad-hoc agent models online and from scratch the behavior of its teammates. Even for cases when students, on which the authors had no control, have designed the other agents, the algorithm of the authors was able to perform even better than the initial student teams.

Finally, it is only recently that explicit, but initially unknown, communication between agents has been considered. Samuel Barrett et al. introduced an abstract arm bandit domain with communication [Barrett 2013a]. This work is, to our knowledge, the first work in MIPC considering communication between agents and where the ad-hoc agent initially does not know how the other agents interpret its messages. However this problem differs from the challenge of *learning from unlabeled interaction frames* as the task the agent should optimize could be inferred without the use of communication through environmental reward only, and communication only intends to speed up the learning process.

Some aspects of MIPC are closely related to our problem of learning from unlabeled interaction frames, such as the challenge of communication between teammates. Considering robots can come from different factories in different countries, they might not use the same protocols of interaction and adapting to such protocols is a central future challenge of MIPC. Yet, the communication aspect has been only little investigated [Barrett 2013a], and we believe the work presented in this thesis can bring interesting perspectives to the MIPC challenge. Especially it can be interesting to investigate domains where communication between agents is mandatory to succeed in the task, but where communication protocols between teammates are a priori unknown.

2.4 Unsupervised learning

Unsupervised learning is the problem of finding hidden structures in unlabeled data. It mostly applies in clustering tasks where a dataset is divided into subgroups of data sharing similar characteristics, such as a close proximity in the feature space. In the following, we present two unsupervised learning problems that share some similarities with our problem of *learning from unlabeled interaction frames*.

Unsupervised multimodal learning In unsupervised multimodal learning, the system has access to synchronized raw information from multiple modalities. A particular instance of multimodal learning is the acquisition of language where the learner has to link perception of an object to the sound of its name, or of a sound to a gesture such as in [Mangin 2013]. The learner receives continuously a visual and an audio stream and should learn to associate parts of the visual information with

their associated audio stimulus. But the visual and audio information are already synchronized such that the relevant information from the visual stream is perceived simultaneously with its associated audio stimuli.

In a robotic application, Yasser Mohammad et al. used multimodal learning to segment and associate gesture commands from a user to actions of a robot [Mohammad 2009b]. The gestures and actions were observed from a continuous stream extracted from a Wizard of Oz experiment (where the robot is secretly controlled by a human). They relied on a motif discovery algorithm to identify recurrent and co-occurrent patterns in the gesture and action flow [Mohammad 2009a]. In [Mohammad 2010] the same authors extended their approach to allow their system to derive controllers for the robot and not just find recurrent patterns, as well as a methods to accumulate the acquired knowledge for long-term operation.

However, while being unsupervised, the stream of data where synchronized and collected using a Wizard of Oz setup, meaning that the association between the gestures and the robot's actions was provided. And importantly, the relation between the gesture commands from the user and the actions of the robot was direct. Contrary to our problem of learning form unlabeled interaction frame, there is no intermediate steps of analysis required to infer the meaning of the human gestures.

Simultaneous localization and mapping Simultaneous localization and mapping (SLAM) [Smith 1990, Dissanayake 2001] is the problem of constructing a map of an unknown environment while simultaneously keeping track of the robot's location in that environment.

SLAM seems to include a chicken and egg problem. To build the map, the robot needs to know its location on the map such as to be able to include its current measurements to the map. And to know its location on the map, the robot needs to know the map such as to infer its position from its measurements. In practice, the answers to the two questions cannot be delivered independently of each other.

However the robot knows that the data received from its sensors refers, for example, to noisy information about distances to obstacles. The robot also often knows the qualities of its sensors and motors, and roughly how it's actions influence its position. For example, by measuring changes in wheels rotary encoders, the robot can approximate its position shift after small control commands. Accessing to an approximation on its position shift, the robot can now update the map given its new sensory information. Using only this source of information is limiting, especially because every error accumulates over time. There are several others sources of information the robot can rely on. For example, the environment is often assumed to be fixed. Hence the robot can track its relative position to some landmarks, and incrementally update its position on the map while detecting some other landmarks and incrementally building the map.

Unsupervised learning also deals with unlabeled data. But contrary to our problem, unsupervised learning only identifies direct relations between observations. In our

problem of *learning from unlabeled interaction frames* the system must also identify a task, unknown at start, from the incoming unlabeled data. This makes the relation between observations non direct. Indeed, the association between the different observations requires an additional abstract piece of knowledge, i.e. the task, that is yet unknown at the beginning of the interaction.

2.5 Brain computer interfaces

EEG-based brain-computer interfaces (BCIs) have been used successfully to control different devices, such as robotic arms and simulated agents, using self-generated (e.g. motor imagery) and event-related potentials signals (see [Millán 2010] for a review). Error-related potentials (ErrPs) are one kind of event-related potential (ERP) appearing when the user's expectation diverges from the actual outcome [Falkenstein 2000, Chavarriaga 2014]. Recently, they have been used as feedback instructions for devices to solve a user's intended task [Chavarriaga 2010, Iturrate 2013a].

As in most BCI applications, ERP-based BCI requires a calibration phase to learn a decoder (e.g. a classifier) that translates raw EEG signals from the brain of each user into meaningful instructions. This calibration is required due to specific characteristics of the EEG signals: non-stationary nature [Vidaurre 2011], large intra- and inter-subject variability [Polich 1997], and variations induced by the task [Iturrate 2013b]. The presence of an explicit calibration phase, whose length and frequency is hard to tune and is often tedious and impractical for users, hinders the deployments of BCI applications out of the lab.

Thus, calibration free methods are an important step to apply this technology in real applications [Millán 2010]. We note that the problem of *learning from unlabeled interaction frames*, which is central to this thesis, is the same problem as removing the calibration procedure for interactive systems, of which BCI is a good example. Despite the importance of calibration-free BCI, there are only few BCI applications that are able to calibrate themselves during operation.

Several works considered online adaption of classifiers. In [Vidaurre 2010] the authors show that it is possible to adapt the decoder online for long-term operation using sensory-motor rhythms. Similarly for BCI based on event-related potentials or steady-state evoked potential (SSEP) many works have studied how to continuously adapt the brain decoder [Fazli 2009, Lu 2009, Fazli 2011, Congedo 2013, Schettini 2014].

However, while the above methods allow a more flexible and online adaptation to each user, they are not strictly calibration-free methods. They require a relatively smart prior on the decoder of brain signals beforehand. Such prior is usually extracted from intersubject information [Fazli 2009, Lu 2009, Vidaurre 2010]. We identified two other works that start the adaptation process from a randomly seeded classifier. While still requiring a prior on the classifier these methods have been shown to be robust to a large range of initialization.

In invasive BCI, Orsborn et al. proposed a method to learn from scratch and in closed loop a decoder for known targets using pre-defined policies to each target [Orsborn 2012]. However, their method requires a warm-up period of around 15 minutes. Using non-invasive technologies (EEG based), to our knowledge only one group of researchers achieved calibration-free interaction [Kindermans 2012a, Kindermans 2014a]. We detail their work in the following subsection.

2.5.1 Work of Pieter-Jan Kindermans et al.

Kindermans et al. considers the problem of P300 spellers. A P300 signal is an event-related potential elicited in the process of decision making [Polich 2003]. It is evoked by the reaction to a visual or auditory stimulus, and it is linked with the process of evaluation or categorization of stimulus by our brain.

A P300 speller exploits the properties of P300 ERPs to build a communication tool allowing users to input texts or commands to a computer by thought. The speller interface consists of letters arranged in rows and columns (see Figure 2.1). The user is asked to focus his sight on the letter he wants to write. Then the rows and columns of the matrix are successively and randomly highlighted. By detecting the P300 signals in the users brain activity, it is possible to decode which row and column are associated to the letter the user wants to write. As each rows an columns are flashed the same number of times, the P300 stimulus has a frequency of $\frac{1}{N}$ (where N is the number of rows or columns of the matrix).

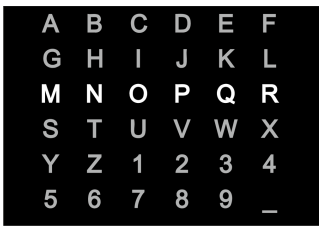


Figure 2.1: A speller interface with the third row highlighted.

Kindermans et al. proposed a method to auto-calibrate the decoder of P300 signals by exploiting multiple source of information [Kindermans 2012b, Kindermans 2014b]. As for most of the work presented above, they consider transfer learning where a model of previous subjects is used to *"regularizes the subject-specific solution towards the general model"*. As it is a spelling task, they also make use of language models as a prior probability on the possible next letter. They also include a dynamic stopping criterion that is a measure of confidence on the next letter allowing the system to stop when it reaches a confidence threshold. Finally, and of more interest for us, they make use of unsupervised learning using an EM algorithm to update the classifier as new data comes in. They exploit the particular fact that among the multiple stimulations only one event out of six encodes a P300 potential in the speller paradigm.

While still requiring to bootstrap the system with several random classifiers as well as a warm-up period, Kindermans et al. have shown their unsupervised learning method coupled with specific properties of the task allows to start interacting with a speller without the need for calibration procedure [Kindermans 2012a, Kindermans 2014a]. This achievement correspond to solving the problem of *learning from unlabeled interaction frame* and is therefore of high interest for our work. We now explain what specific information was used to solve this problem and identify it as being of a very specific nature, which differs from all other approaches.

As detailed earlier, the P300 speller problem offers some guarantee on the repartition of "correct" and "incorrect" P300 events. Only one row and one column should elicit a P300 response. In the case of a 6 rows speller, if each row are systematically scanned the same number of time, only one signal out of 6 will encode a positive P300 signal. And even more informative is the fact that, even if the wrong letter is identified in the end, at least 4 labels out of 6 will be correctly assigned. Indeed, if the wrong letter is identified, two labels will be swapped, resulting in two association errors, but still four "incorrect" labels will be correctly assigned. Obviously, if the correct letter is identified, the "correct" label will be correctly assigned, as well as the five "incorrect" labels. In the end, this is quite a lot of information that can offer good guarantees for their EM algorithm to identify properly the "incorrect" signal cluster; leaving the second cluster for the "correct" signals. As more data are collected, the EM algorithm will be better at identifying the underlying structure of the data and will be able to identify the cluster of "correct" signals from the one of "incorrect" signals given the constraints detailed above. As the process continues, identifying further letter is made easier, and importantly, by going back in the history of interaction, the system can correct letters that were wrongly identified.

As we will discover in next chapters, our method does not require having access to such constraints and guarantees about the task, which makes our work easily generalizable to many types of problems. However, the work of Kindermans et al. already exploits information of a very specific nature to solve the problem of *learning form unlabeled interaction frames*. Contrary to all the other approaches, their information source does not provide a direct knowledge about the task (as a language models do), neither about how to decode the signals themselves (as transfer learning methods do). It rather provides information emerging for the joint combination of a task and of a signal decoder. That is, that for the correct task (i.e. the correct letter), only one signal should be classified as "correct" and all the others as "incorrect".

This type of information, that acts neither on the task, neither on the signal decoder, but rather on the combination of both is at the core of the work we will present in forthcoming chapters. As we have seen in section 2.2.2, Cederborg et al. also make us of a similar source of information but reasoning about the consistency of some

gestures with respect to different geographical references, e.g. object positions. We will summarize those works in next section 2.6 and highlight the differences and improvements of our method.

2.6 Discussion

We reviewed an extensive number of related works ranging from the computational modeling of language to more practical brain computer interaction problems. While releasing some important assumptions on the interaction, in most of those works the communicative signals had a direct relation to one element of the environment or to the task itself, such as being the name of a color, a shape, or a gesture type. In our work the signal to meaning relation will be more abstract such as whether an action was "correct" or "incorrect" with respect to an objective. Also, in most of these existing works the interaction between partners was pre-programmed and most of the time the robot knew how to use or understand communicative signals innately, e.g. how the teacher expresses "correct" or "incorrect" feedback.

We note that in this thesis we will assume teachers are optimal and simply model some percentage of teaching mistakes to account for the variability between users. This might not be an accurate assumption given the work presented in the beginning of this chapter about human teaching behaviors. However our method is not restricted to the use of optimal teacher models, the only requirement is to have access to model of the human teaching behavior, which may include systematic errors or bias.

The work we present in this thesis shows mechanisms allowing a learner to simultaneously learn a new task and acquire the meaning associated to feedback and guidance signals in the context of social interaction. Furthermore, we show mechanisms allowing the learner to leverage learned signals' meanings to acquire novel tasks faster from a human. To our knowledge, only two works are tackling the same problem as the one presented in this thesis. And surprisingly, those two works lies in the computational modeling of language acquisition (work of Cederborg et al. in section 2.2.2) and in the BCI domain (work of Kindermans et al. in section 2.5.1).

Especially, it is in the BCI domain that the idea of adaptive interface seems to be highly developed, with many methods to continuously adapt a brain decoder during operation. This may be explained by the specific nature of brain signals, which are not a natural way for humans to interact with machines. Therefore humans do not share common abilities in their generation and use of brain signals, and at design time we cannot use our daily intuition for creating universal decoders of brain signals. This differs from work on speech or facial expression recognition where many a priori knowledge can be included into the system. This kind of consideration may explain why the problem of adaptive interfaces and our specific problem of *learning form unlabeled interaction frame* has only been considered recently in human-robot interaction scenarios.

In the following of this discussion we summarize the main similarities and dif-

ferences between our work and the work of Cederborg et al. and of Kindermans et al. as respectively discussed in section 2.2.2 and section 2.5.1. For the interested readers, this discussion section may be worth reading again once the reader has been through the remaining of this thesis, especially through chapter 4.

We can list a number of differences between the work presented in this thesis and the related work presented in this chapter:

- First, we explicitly define and provide some solutions to the problem of *learning from unlabeled interaction frames*. This problem is still relatively new in the domain of human-machine interaction. It represents a new step towards creating machines able to flexibly adapt to each particular users by learning the way such users communicate specific meanings to the machine.

- Compared to the work of Cederborg et al. [Cederborg 2011], our robot is already equipped with sufficient skills to perform the task, i.e. if it knew the goal it could fulfill it by its own mean. In most of our experiment, the robot further knows that the task belong to a limited set of task. In [Cederborg 2011], less constraints are applied on the task space, the robot only knows it will have to reproduce a continuous gesture of unknown type which is not restricted to belong to a limited set. However, in their work, one communicative channel directly encodes the "name" of the gesture demonstrated; in our work the relation between the teaching signals and the robot's actions is indirect and depend on the true unknown task.

- Compared to the work of Kindermans et al. [Kindermans 2012a, Kindermans 2014b] our method does not require to bootstrap the system with random classifiers, which are updated step by step but unreliable at start. Our method rather identifies the classifier from scratch. This difference is mainly due to the experimental setup used in our respective work. For example, in the P300 speller of Kindermans et al. a new letter must be identified every 15 flashes. Logically the system requires a warm up period that produces a high number of spelling errors in the beginning of each experiment. Such errors are however detected and corrected later on, after the so called "eureka" moment [Kindermans 2012a], when their EM algorithm had access to enough data to identify the positive and negative clusters. To the contrary, by applying our method to the speller paradigm, the system would only pick a letter once it is confident that the letter is the correct one; therefore reducing dramatically the number of spelling errors but with a longer "blank sheet" period for the user in the beginning. However the computational cost of our method increases with the number of possible tasks (e.g. the number of rows and columns of the speller), which is not the case for the work of Kindermans et al..

- Another difference between our work and the work of Kindermans et al. lies in the properties that their world should hold in order to ensure a proper functioning of their algorithm. In the work of Kindermans et al., the world

should guarantee a specific ratio of "correct" and "incorrect" signals in the received signals. This ratio could be in favor of either one or the other label but is mandatory to be asymmetric, with more signals from one class than from the other. Indeed, their EM algorithm alone can identify two clusters in the feature space of the signal, but cannot attribute labels to each cluster without having access to additional information (the ratio of positive and negative P300 signals in their case). Our method is more generic and can be applied to a majority of sequential problems, even when it is impossible to define a sequence of actions that guarantee a specific ratio of meanings in the received signals.

- Compared to both Cederborg et al. and Kindermans et al. our approach is more generic and can be applied directly to a variety of sequential problems which are common in the human-robot and human-computer interaction domains. In particular we highlight the chicken and egg problem inherent to interacting with machine, and define the general challenge of *learning from unlabeled interaction frames*. However, we note that this thesis focus on a very specific problem and more broad considerations are highlighted in the thesis of Thomas Cederborg [Cederborg 2014a].

- We consider sequential tasks, which are tasks requiring the agent to perform a series of correct actions in order to fulfill the task correctly. Therefore there is a planning aspect involved which was not present in the work of Cederborg et al. where the robot passively observed interactant-demonstrator interactions, neither in the work of Kindermans et al. where the row and column flashes patterns were determined in advance. We note that the problem of P300 spellers used by Kindermans et al. could be represented as a sequential problem, where flashing a particular row or column represents the agent's available actions. However, if the sequence of actions is no more pre-defined, i.e. with the same number of flashes per row or column, the guarantees that only one signal out of N encodes a positive ERPs would not be satisfied and their algorithm would be more likely to converge to a wrong classifier.

- Given the sequential nature of our problems, we consider active learning which is the ability of our agent to actively selects its actions in order to improve its performance. As stated previously, this planning aspect was not considered in the work of Cederborg et al. and Kindermans et al.. We will show in chapter 5 that planning when both the task and the signal to meaning mapping is unknown requires to develop a new measure of uncertainty. Our measure takes into account the uncertainty on both the task and the decoder; and is an important contribution of our work.

- We also provide a number of extensions in chapter 7 to our algorithm, such as to cope for continuous state spaces and continuous task spaces. We further release the assumption that the interaction frame (either feedback or guidance

frame) is known in advance and assume it belongs to a pre-defined set of possible interaction frames.

- Moreover, aside from many empirical demonstrations in both simulated and real experiments, we also present in chapter 7.7 a simple mathematical proof providing some guarantees on our method. To our knowledge, we provide the first proof showing that a system is able to learn simultaneously a task from human instructions as well as the signal to meaning mapping of the user's instruction signals.

- Finally in chapter 6, we will test our algorithm in a BCI application. Our experiment differs from the one of Kindermans et al. [Kindermans 2012a, Kindermans 2014a] because our task is a target reaching task where the agent decides on its own which action to take next. This task is sequential, meaning that several actions must be executed to reach the goal. In addition, we use a different kind of error related potential signals to encode a "correct" or "incorrect" feedback for the agent. Our signal is of similar nature than the P300 signals used by Kindermans et al., i.e. they encode a binary event, however they are slower to elicit and are known to be harder to detect [Chavarriaga 2014].

Despite the differences between our work and the work of Cederborg et al. and Kindermans et al., there is similar fundamental properties of the problem that are exploited by our respective works. Especially the notion of interpretation hypothesis developed in Thomas Cederborg's thesis and the use of an information source that emerges only from a combination of constraints on the task and signal spaces.

In [Cederborg 2011], Cederborg et al. reasoned about the consistency of some gestures with respect to different geographical references, e.g. object position, knowing that the signals of the user could refer to only three possible coordinate systems and therefore relying on interpretation hypothesis. In [Kindermans 2012a, Kindermans 2014a], Kindermans et al. reasoned about the ratio of positive and negative P300 ERP signals that should be observed for the correct letter. In our work, we propose to capture the coherence between the organization of the teaching signals in their feature space and their associated labels. We make use of interpretation hypothesis to create one set of signal-label pairs for each task. The correct task hypothesis is the one from which a more coherent, consistent, signal to meaning model emerges from the hypothetic labeling process. That way both the task and the signal to meaning model can be identified. Hence the assumption of coherence between the user behavior and our user model is a primordial prerequisite for our algorithm to work. Interestingly, this measure is more general than the one used by Kindermans et al. and does not require a specific ratio of "correct" and "incorrect" signals to work.

As we will explore in the following chapters, this type of information, that acts neither on the task, nor on the signal decoder, but rather emerges from the combination of constraints on both task and signal spaces are fundamental properties we will exploit to solve the problem of *learning from unlabeled interaction frames*.

Before presenting the core principles of our algorithm in chapter 4, we present in next chapter (chapter 3) a semiotic experiment where two human partners must handle a similar situation than our problem of *learning from unlabeled interaction frames*.

Chapter 3

Can humans learn from unlabeled interactions?

Contents

In previous chapters, we defined a new challenge of interaction without pre-coordination for human-robot interaction scenario, which we called *learning from unlabeled interaction frames*. But can human solve this problem in a human-human interaction scenario?

In this chapter, we start by introducing the challenges related to such human-human experimental studies and present some related works. Then, we present our experimental setup that investigates how human negotiate a protocol of interaction when they cannot rely on already shared one. We took inspiration from the constraints inherent to human-robot interaction, such as restricted perception and communication abilities. The task is a joint construction task in which participants hold asymmetric roles, and can communicate only by pressing buttons of undefined

The work presented in this chapter has been published in [Vollmer 2014a]. It is the result of a collaboration with Anna-Lisa Vollmer and Katharina J. Rohlfing. The experiments reported in this chapter had been carried out during the internship of Chloé Rozenbaum.

meanings. Our experimental results show that participants manage to successfully interact and understand each other under such restricted interaction. They usually rely on stereotyped situations to synchronize their communication and intended meanings. We identified that some situations types are more recurrent and more easily understood than others. Based on our observations, we take some lessons that can be applied to human-robot interaction scenarios. Among them, we observed that participants generated interpretation hypothesis of the communicative signals and tested their hypothesis on next events. This will be the basis for the development of following chapters.

3.1 Introduction

Studying the Co-Construction of Interaction Protocols in Collaborative Tasks with Humans

We consider the overall goal of developing a robot system that should learn from and interact with non-expert users. Without assuming that the robot understands human feedback (i.e., without programming the information on how and when feedback is given into the system beforehand) how should the system understand what the signals it perceives mean and what they are referring to (cf. Gavagai problem [Quine 1964])?

In interaction, humans align and effortlessly, maybe even automatically, create common ground in communication [Clark 1991, Pickering 2004]. For this, they dispose of an immense amount of shared information. They make use of frames established in the history of interaction. Frames create a common ground about the purpose of the interaction [Tomasello 2009, Rohlfing 2013] and include "predictable, recurrent interactive structures" ([Ninio 1996], p. 171). Frames thus provide interactants with guidelines about how to behave (a protocol for interaction) and also help interactants to understand the communicative intentions of their interaction partner. It further comprises basic behavioral patterns like roles, turns, timing, and exchange mechanisms. We aim at investigating how these interaction protocols emerge, because it would shed light on the basic mechanisms underlying interaction and inform us about what are the main issues in building robots capable of a similar interactional flexibility as the one humans possess. We are for instance interested in what kind of strategies humans use to align and what kind of meanings of social signals they converge to. Therefore we need to conduct research into how interaction protocols are negotiated in human-human interaction, aiming for that the obtained findings could be used as priors for a robotic system interacting with humans.

Unlike humans, who assume an immense amount of shared information, a robot system cannot rely on already established protocols for interacting. This is because, on the one hand, little is known about the universal interaction protocols humans rely on in communication, and on the other hand, human-robot interaction (HRI) is

still very different from human-human interaction, as it is clearly characterized by asymmetry and restrictiveness in the sense that the human and the robot in general do not have the same abilities, modalities, mechanisms, and body for communication, perception and action [Lohse 2010]. For example, a robot that does not have arms cannot gesture, a robot without the respective algorithms or sensors does not perceive gaze direction or understand speech commands, and without any knowledge of internal computational mechanisms it is difficult to assess how a robot perceives its interaction partner and his/her actions. It is thus important that robots are able to negotiate meaning online with their interaction partners.

We designed an experimental setup with which we aim at investigating the processes used by humans to negotiate a protocol of interaction, when they do not already share one. In this chapter, we present and justify the method used and mention the results obtained from a pilot study employing the setup.

Humans and robots view the world differently, so if we want to transfer our results to human-robot interaction, we should not assume that in the interactions we want to investigate, the partners see the world/interaction in the same way. To investigate the process of negotiating an interaction protocol, we thus consider a setup of a joint construction task in which participants assume asymmetric roles: the role of a builder and the role of an architect. With building blocks, the builder should assemble a target structure which is unknown to him/her but which the architect knows. This collaborative construction task with a joint goal renders the communication between participants indispensable and thus the game is not solvable by either one of the participants alone, e.g. with mere exploration. Thus, failing to complete the game successfully is equivalent to failing to communicate successfully. Communication is not face-to-face but channels are restricted, so that it is not possible for participants to communicate via familiar verbal or non-verbal communication channels, as for example speech or gestures. At the same time, the setup does not constrain all aspects of communication and thereby gives participants much freedom with respect to some features, including timing and rhythm or possible meanings (e.g. of button presses). The setup does not impose a predefined sequence of interaction upon participants, as it is often done in HRI scenarios [Akgun 2012], but still benefits from a laboratory setting in which we do not need to take the full complexity of natural social interaction into account. With the aim to simulate the sending of signals to an interaction partner who does not have the same perceptual capabilities – similar as in an interaction with a robot – in our study the architect does not know how exactly his/her signals are perceived by the builder. For the successful completion of the thus highly challenging joint task of the game, both participants have to learn how to interact with each other.

The main contribution of this chapter is the presentation of the novel experimental method of our study. We would like to demonstrate that it allows to study important questions for the understanding of human negotiation of interaction protocols in joint construction tasks and that these questions are very important for HRI in the long-term. We first briefly discuss related work, then present our method, the results of the pilot study, and conclude by highlighting the implications of our

results for human-robot interaction.

3.2 Related work

To our knowledge, there exists little research in the field of linguistics or pragmatics on this topic. To investigate this process of negotiation, we chose to design an experimental semiotics study which enables us to modify communication in the desired way, namely to restrict communication between participants who are assuming asymmetric roles.

The field of experimental semiotics studies the emergence and evolution of communication systems [Galantucci 2009]. Here, instead of computer simulations as conducted by others (see [Cangelosi 2002, Steels 2012b]), controlled experiments in laboratory settings are designed to observe communication between participants who perform joint tasks. For instance, Galantucci et al. showed that pairs of participants performing a joint task could coordinate their behavior by agreeing on a symbol system [Galantucci 2005].

Most experimental semiotics studies developed to study joint action involve symmetric communication (cf. [Galantucci 2011]). Two studies that do consider asymmetric communication are the studies conducted by de Ruiter et al. [De Ruiter 2010] and Griffiths et al. [Griffiths 2012].

In their score- and round-based Tacit Communication Game, de Ruiter et al. investigated the cognitive processes responsible for the development and the recognition of new conventions by looking at reaction time. In a 3-by-3 grid world, two participants each manipulate a shape. For both of the shapes, the "sender" sees a target configuration. He/she first has to communicate the other player's target configuration to the other player, the "receiver", and second has to bring the own shape to his/her own respective target. De Ruiter et al. found that participants succeeded 83% of the time and that the timing of movements is used to indicate a position. When comparing success rates for when the sender saw versus did not see the receiver's moves, the authors found that the game involves bidirectional communication and receiving information about the other player facilitates communication. The harder the communicative problem was, the more planning time was needed by both participants.

The setup of the study conducted by Griffiths et al. [Griffiths 2012] is more directly related to our setup. It is based on the alien world game setup by Morlino et al., in which in a square world shown on a computer screen, positions (left or right) and movements (shake horizontally or shake vertically) of 16 objects have to be explored via a mouse to maximize a score [Morlino 2010]. It investigates the learning of categories, so the objects belonged to four categories that were defined by certain properties of the objects. Each category was associated with a target manipulation, i.e. shape and weight determined where an object should be positioned and how it should be moved. In the work by Griffiths et al. the learner could realize this task with the help of information given by a tutor who had prior knowledge about

the categories the learner should explore. For this alteration, two players played the originally single player game simultaneously in separate rooms over a network connection. The computer screens in this setup additionally showed six buttons underneath the grid world. The tutor's communication to the learner consisted of the pressing and releasing of these six buttons using a keyboard. This was the only action the tutor could perform on the world. The authors found that tutors most commonly send feedback and guidance instructions to the learners. Negative feedback was given least often and its amount correlated with task failure. Learners who ignored fewer signals performed the task better.

The main, very important difference between the two asymmetric setups described above and our setup concerns the very nature of the task. Whereas in Griffiths et al.'s study the task is solvable with mere exploration, in our setup the input of the architect is essential. The latter is also the case in the study by de Ruiter et al., but in our setup no score is displayed to either of the players who in our case are not separate learner and tutor, or receiver and sender, but they solve the task together assuming the roles of a builder and an architect. Correspondingly, in our setup, the game does not include multiple episodes or rounds but it is continuous with the builder deciding when the task is completed and the game ends. The game of the study by de Ruiter et al. is based on fixed turns, which is not the case with our game, where participants can act simultaneously and react directly upon each others conduct. By designing a continuous game without displaying a score, interaction remains natural (i.e., free) to a high degree.

Another important difference that makes our setup novel regards the restriction of communicative channels. In contrast to the other two works, in our setup, the architect is not aware of how his/her actions are presented on the builder side and how they will be perceived. This renders the situation similar to human-robot interaction. This difference should also minimize the use of simple iconic feedback (as for example encoding the manipulation of horizontally shaking the object by alternately pressing one button to the right and one button to the left as reported by Griffiths et al.)

3.3 The Collaborative Construction Game

With the aim that improving our understanding of how humans negotiate protocols of interaction could provide hints on how robots could do it also, we designed a new experimental setup that allows to constraint the communication channels between two partners in asymmetric roles who should collaborate in order to achieve a joint construction task. We consider a joint construction where only one participant is aware of the targeted construction (the architect) while only the other has the ability to achieve it (the builder). The communication between partners is reduced to the use of symbolic events that the architect can send to the builder. Neither the architect nor the builder are given any a priori information on the meaning of the symbolic signals and should agree on the meaning of such signals by the mean of

the construction task.

This section describes the details of the experimental setup, the participants we recruited, and the protocol used for running the study.

3.3.1 Setup

Figure 3.1 gives an overview of the experimental setup which considers an architect and a builder that are each seated at a table in front of a computer screen in two separate rooms and can neither hear nor see each other.

The builder is equipped with a set of building blocks, in our case with 12 primary-colored Mega Bloks® toy blocks differing in shape and color (see Figure 3.2b). There were three red two-pads, two red three-pads, two yellow four-pads, two blue three-pads, two green two-pads, and one green four-pads blocks.

The goal of the game is to assemble a specific construction yet unknown to the builder. As exemplified in Figure 3.3, a construction is a flat combination of several blocks at least linked to one another by one pad. It does not necessarily contain all available blocks.

The architect is given an image of the specific construction to be built and is told to guide the other player building it. A screen displays a live top view of the builder workspace. To communicate with the builder, the architect has access to a rudimentary interface made of 10 buttons, see Figure 3.2a. Pressing a button displays a symbol on the screen located in the builder room. Each button is mapped to one of ten symbols and one of ten positions (two rows of five symbols) on the builder's screen, whereby the spatial organization of buttons differs from the spatial organization of displayed symbols. The mapping is randomized for each subject and fixed for the duration of one game. Figure 3.4 shows the different symbols.

3.3.2 Participants

We recruited 22 participants (19 m, 3 f) among students and staff at INRIA Bordeaux Sud-Ouest. Their age range was between 20 and 35 ($M = 25, SD = 3.91$) years. They played the collaborative game in pairs, where the two players in a pair were assigned randomly to the roles of a builder and an architect. Seven of the eleven pairs played the game together twice, such that each of the 14 participants involved assumed each role once. One second round of a dyad was excluded from the analyses, because the architect neglected the task instructions and altered the target structure during the game. This resulted in a total of 17 rounds.

3.3.3 Procedure

Participants were not given the chance to talk about the game before it began. Architect and builder were instructed about their respective roles separately in their respective rooms. We presented the architects with a set of 20 pictures of different constructions from which they chose one. The builder was informed about the

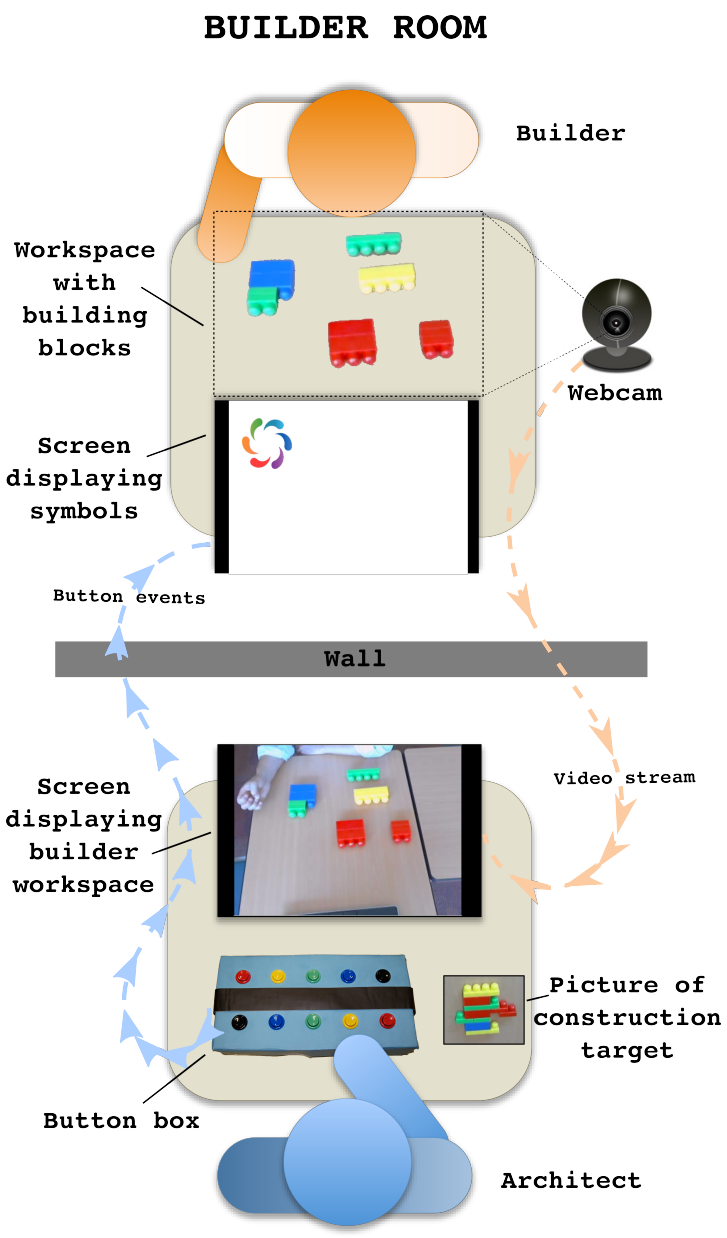
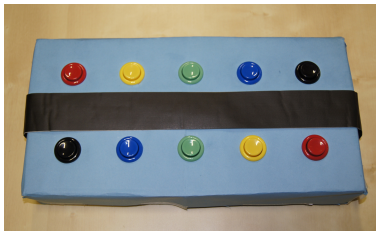
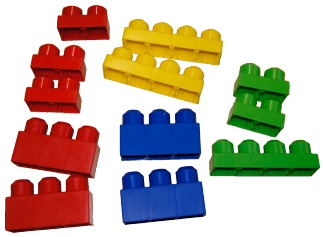


Figure 3.1: Schematic view of our experimental setup. An architect (bottom) and a builder (top) should collaborate in order to build the construction target while located in different rooms. The architect has a picture of the targeted construction, while the builder has access to the construction blocks. The communication between them is restricted. The architect only sees a top view of the builder's workspace and can communicate with the builder only though the use of 10 buttons which, when pressed, display symbols on a screen on the builder side.

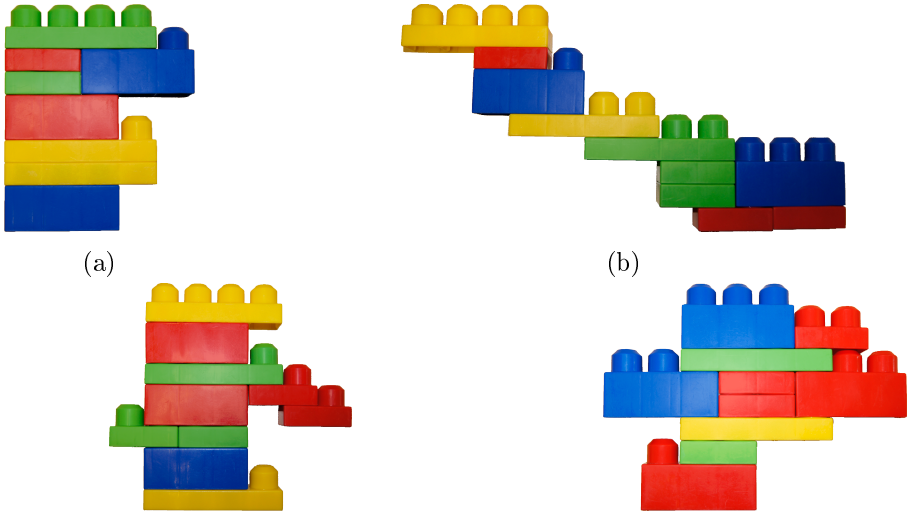


(a) The box and the buttons used as an interface for the architect to communicate with the builder.



(b) All toy blocks used in the collaborative construction task.

Figure 3.2: Elements of the setup.



(a) (b) (c) (d)

Figure 3.3: Four examples of target structures presented to the architect.

Figure 3.4: The ten signs displayed on the builder screen.

constraint that applied on the construction, i.e. flat construction that does not necessarily contain all available blocks. The architect and the builder were specifically told that the button positions did not directly map onto the symbols' positions displayed on the builder's screen, but that the mapping was fixed and arbitrary. Additionally, because the architect could see the hands of the builder during the game (see Figure 3.1), the builder is told to only use his/her hands to move blocks and not to use hand signs. In practice, this was well respected by participants.

The game was **not** preceded by any training sessions. We aimed at reducing the time between the instruction of the participants and the beginning of the game as much as possible, so that they did not have time to elaborate any concrete strategy before the game began.

Once the game started, we observed the behavior of the two players and asked them to speak aloud about the meaning associated to the symbols/buttons. The experimenters took notes on the participants' remarks. The experiment stopped only when the builder decided and told the experimenters that the structure he had build was correct.

3.4 Results

As stated before, the current pilot study serves as a proof of concept. We aimed at designing a setup allowing to study the processes involved in the formation of interaction protocols in asymmetric interaction with the particular constraint that the players could neither solve the task by themselves nor did they have access to any reward function.

Our pilot study revealed a great potential in the use of our experimental method to study many aspects of communication relevant to HRI. With our setup, we will be able to study, among others, questions related to alignment, rhythm, contingency, and feedback, which have been in the focus of HRI research for some time [Kopp 2010, Michalowski 2007, Fischer 2013, Vollmer 2014b, Pitsch 2013, Wrede 2010].

Surprisingly, while the construction task in this setup seems really challenging on paper and participants thought they would never succeed, a majority of the architect-builder pairs succeeded on building the correct construction. We analyzed a total of 17 experiments, of which 13 were successful and 4 failed. The average duration of the runs was 18 minutes ($M = 18\ min, SD = 11\ min$) with a minimum of 7 minutes and a maximum of 45 minutes.

In what follows, we showcase results supporting our claim that our setup can be used to study the co-construction of meaning in restricted, asymmetric interaction. We will first show one run of the game in detail which should give the reader an idea about what happens during an interaction and the richness and aptness of the data to consider a variety of research questions. Then, we will continue with presenting our results on the negotiation of signal meanings and with describing observations of the builder behavior. We will conclude with mentioning interesting additional

considerations that are beyond the scope of this work.

3.4.1 One experiment in detail

Figure 3.5 brings together information about button presses (logs), their intended and interpreted meanings (found by the experimenters from their notes and observations of logs), and the builder's actions (builder video of the construction workspace) and makes clear the bi-directionality of the interaction. On the bottom of the figure, we see that the builder proposes blocks to the architect (blocks not belonging to the target structure in black, blocks belonging to it in gray) (cf. Subsection 3.4.3) and on the top we see how the architect responds to the builder's actions in terms of button presses and meanings. Additionally, we see how the builder interprets these signals of button presses which he/she perceives as symbols on a screen (middle timeline of button presses and meanings) and how these interpretations and believes in turn again influence what the builder does next.

With respect to the meanings of the button presses, we observe changes of button meanings over the course of the interaction. The exact points in time when meaning changes occur have been matched to the button presses by hand and is therefore approximated. While this may be a problem for detailed analyses on a micro level, it is of little importance for the macro analysis presented here. During the first 4 minutes, the architect changes the intended meanings of signals many times and these meanings were not aligned with the builder's interpretation of signals. At 4 minutes, the architect presses all buttons at once, seemingly attempting to ask the builder to clear his/her mind and start over again. Right after this *Reset* signal, the architect changes to one simple *yes/no* strategy using button 1 and 6. On the builder's end, this Reset signal is followed by a pause of actions, which hints at a direct confusion. It is only at 12 minutes into the game that the builder fully understands the intended meaning of the architect's button presses and can start joining two blocks correctly (green graph on the bottom). The experiment continues with the builder suggesting new blocks (bottom - black and gray events) and positions for new blocks (bottom - red and green events) one at a time that are validated or invalidated by the architect. After 19 minutes, the architect presses again all the buttons but this time with the aim of informing the builder that the construction is complete. The builder ended the experiment at that time. The *End* signal was well interpreted by the builder as the interaction was going smoothly until that time and the few remaining blocks were rejected (bottom - black event at 19 min). The final construction was indeed the target one intended by the architect, hence resulting in a successful experiment.

Our setup allows to study the evolution of meanings associated to each button and put it in relation with the current context in the interaction. We find that the constraints inherent to our setup allow to analyze communication, especially the interplay of individual actions and their interactional history, as well as their concrete timing, while lowering interactional complexity and thereby reducing communicative noise.

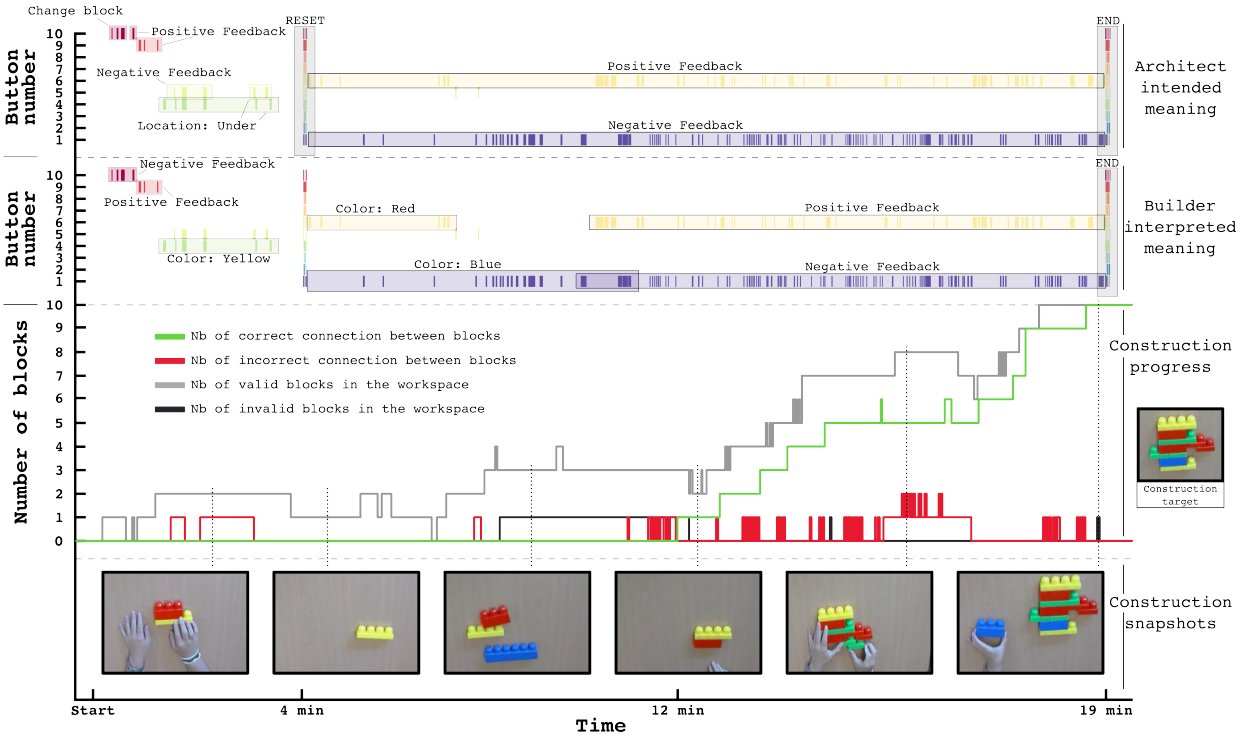


Figure 3.5: Timeline for one experiment of an architect and a builder collaborating towards building the construction target (right hand side). The top and middle part show the timeline of button presses associated with the intended meaning from the architect (top) and the understood meaning from the builder (middle). There were 10 buttons, for which we logged all button presses for each experiment and here display all occurrences as colored dashes. The button events are annotated with the meaning the architect intended or the builder understood as participants reported during the game. Events that are not annotated were not mentioned by the participants. At the bottom, the figure additionally visualizes the progress made by the builder in assembling the target structure and also shows incorrect block propositions, joining of incorrect blocks and mistakes. These events were annotated by hand using the video annotation tool ELAN developed by the Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands [Wittenburg 2006]. A block proposition here started, when the transportation of the block towards the workspace ended and the block lay still on the table. It ended when the block was again picked up and subsequently removed from the workspace. These presentation events were classified into correct and incorrect propositions by determining whether the proposed block was part of the target structure. Equivalently, a joining event started when two blocks were successfully joined at either a correct or incorrect position (again depending on whether the resulting configuration was part of the target structure). It ended right before the two previously joined blocks were again pulled apart.

3.4.2 Meanings

Architects and builders start the game without having agreed on specific meanings the buttons should convey. We start by studying the associated meanings obtained from our notes on signal meanings reported by builder and architect. They seemed to initially consider a large set of possible meanings, but, in the end, were able to agree primarily on only a limited number.

Types of Meanings When analyzing the notes on the participants' explanation of signal meanings (see Subsection 3.3.3), we identified nine different categories of meanings:

1. **Positive Feedback**
2. **Negative Feedback**
3. **End**: The construction is finished.
4. **Reset**: Start over.
5. **Guidance**: Instruction on what to do. It includes *change, invert, revert, new block, continue, stack.*
6. **Color**: Reference to the color of a block. It includes *yellow, blue, red, green.*
7. **Size**: Reference to the size of a block. It includes *small, medium, big.*
8. **Location**: Reference to the location of a block. It includes *under, above, left, right.*
9. **Group**: Reference to a group of blocks. It includes *in, out, group_X.*

Importantly, those categories where **not** suggested to the participants beforehand, but only identified by us in a posteriori analysis.

For each experiment, we determined if the architect or the builder considered each type of meaning (see Figure 3.6). In every single experiment, positive and negative feedback were considered on both architect and builder side. The *End* meaning has been considered on both sides in 14 experiments. More concrete instructions such as *Guidance, Color, Size, or Location* were less often considered, especially by the builder.

This is in line with the findings in [Griffiths 2012], where "correct" and "incorrect" were also identified to be among the most common types of signal meanings.

Matching of meanings between architect and builder Knowing which meaning categories were considered by each of the participants does not tell us if a particular pair of players understood each other. We therefore compared the associated meanings reported by architect and builder for all signals. Similarly to [Griffiths 2012], we then determined the number of signals that were understood,

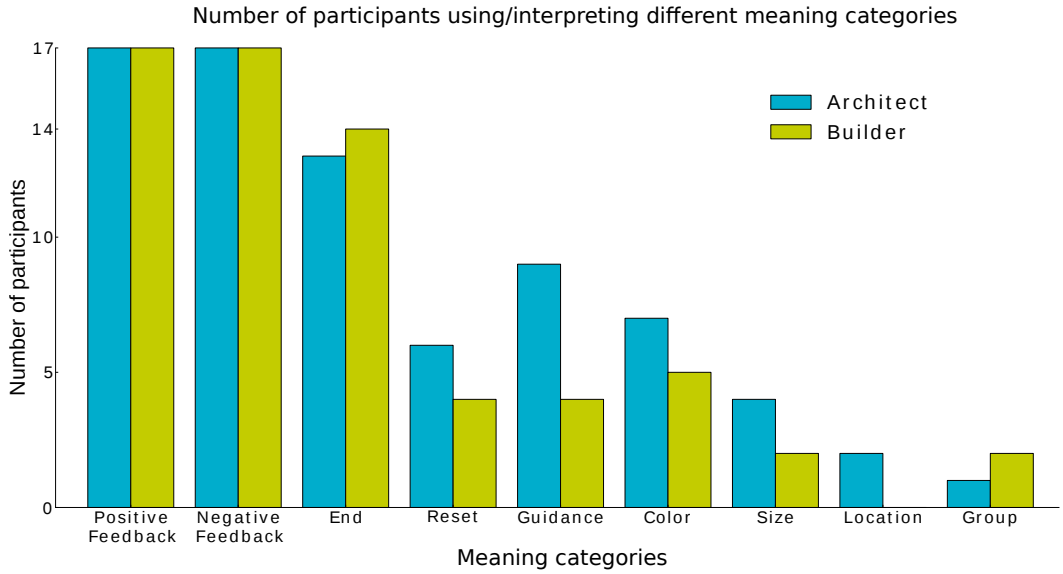


Figure 3.6: Number of participants that used (architect) or interpreted (builder) signals as conveying different types of meaning. All participants considered positive and negative feedback.

misinterpreted, or ignored. A signal is consider understood when both the architect and the builder agree on a common meaning. For signals that were misinterpreted, the builder reported a different associated meaning than the one intended by the architect. The signals that were mentioned by the architect, but not by the builder, were counted as ignored signals. We then averaged the results for successful and failed experiments, see figure 3.7. For successful experiments, the average number of signals understood is $M = 3.6, SD = 0.7$ which mostly corresponds to *Positive feedback, Negative feedback, End*, and occasionally *Reset* when needed (see Figure 3.8). Interestingly for failed experiments, this number drops to $M = 1.3, SD = 1.1$, with a larger amount of signals misinterpreted and ignored.

Even though the architect initially considers many different signal meanings, the players agree only on very few specific ones (positive feedback, negative feedback and End). The question of what are the main factors determining which meanings are considered by participants arises. This leads over to the next subsection in which we will consider the builder behavior to explore its role in which signal meanings are considered and in the ultimate outcome of the game.

3.4.3 Builder Strategies

For the builder, we aimed at identifying common actions across participants in an attempt to quantify the builders' strategies from the video data showing a top-down view of the workspace. What follows is a description of observations on the builders' behaviors.

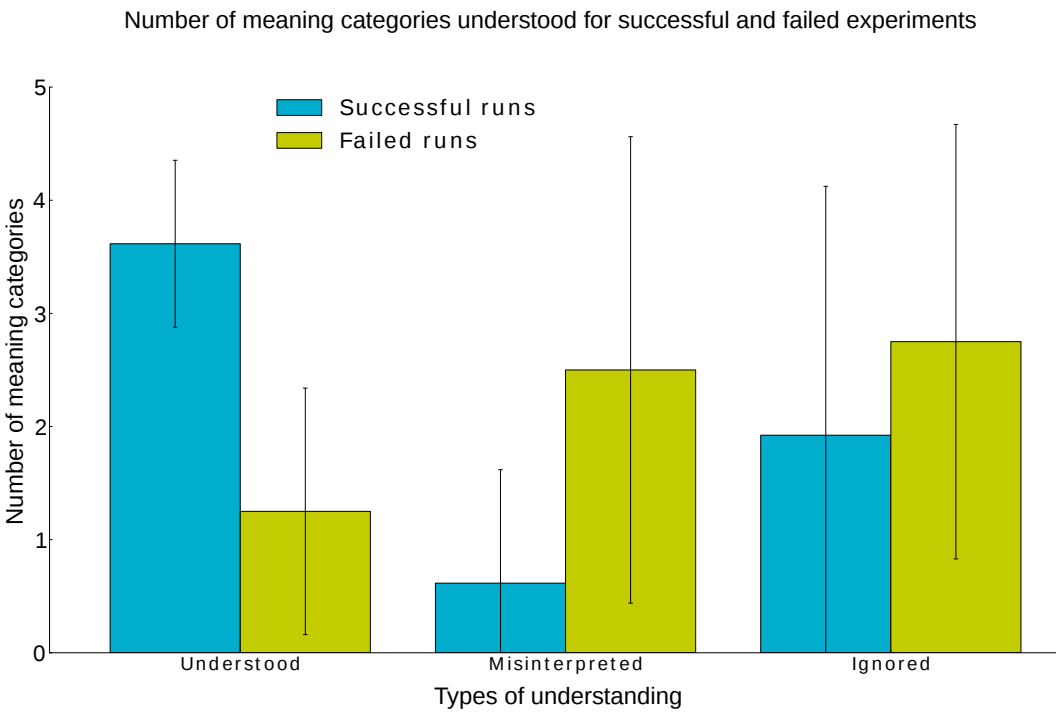


Figure 3.7: Distribution of meaning categories that were understood, misinterpreted, and ignored by the builders. Average across all builders for successful (blue) and failed (yellow) experiments.

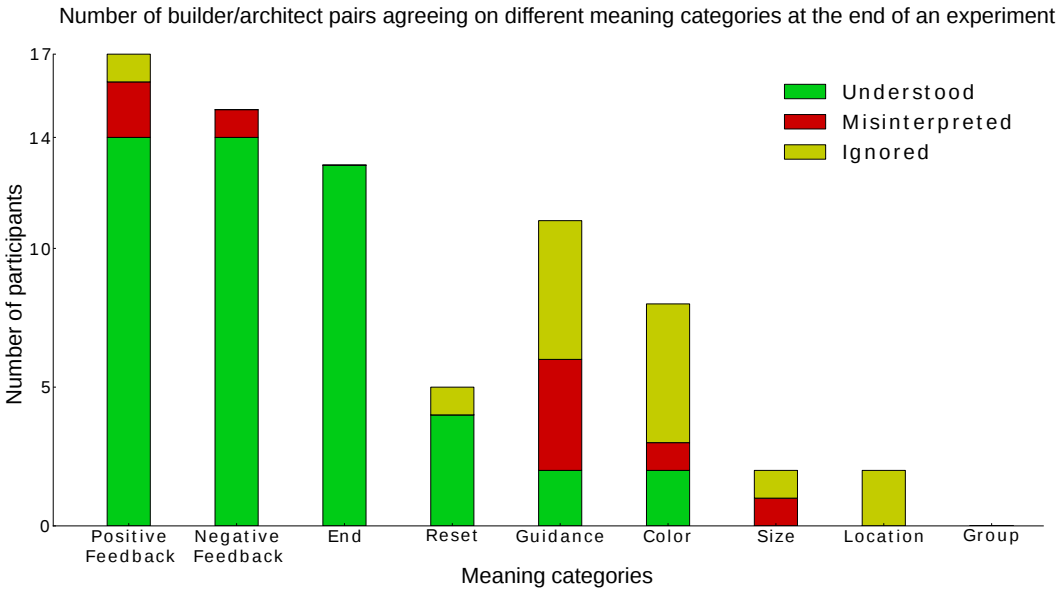


Figure 3.8: Number of builder/architect pairs agreeing or disagreeing on different meaning categories at the end of an experiment.

We identified two main strategies the builders embarked on (for an overview see Table 3.1). For these two strategies, the builders began by presenting only one block at a time. When they presented several blocks at once throughout the game, they did not seem to embark on a successful strategy.

The most common strategy for builders was to determine one correct brick at a time and to subsequently join it with the already assembled structure (see Figure 3.5). Figure 3.5 is a case example of one game/run of the study in which this strategy is used successfully. The builders in 12 (five first rounds and their five respective second rounds, one independent single first round, and one second round) of the 17 runs pursued the same strategy. Only one game (a first round with a successful corresponding second round) of these 12 failed.

The other strategy was to find all blocks belonging to the target structure. Blocks identified as correct were not joined right away, but in a first step all blocks belonging to the target structure were determined and were then subsequently joined one at a time in a second step. This strategy also involved the presentation of only one block at a time and was eventually pursued by two builders who both started out with a different strategy involving the presentation of multiple blocks.

One builder initially tried to find which forms belonged to the target structure. Ultimately, he then identified all blocks belonging to the target structure by one at a time dividing all blocks into two groups. This builder played in a second round, for which in its corresponding first round the builder presented multiple blocks at a time, and the game failed.

Another builder at the beginning tried to elicit a label for either color or form from the architect. In this case, all blocks of one specific color or of one specific shape were presented at a time. This strategy was only pursued by one builder at the beginning of the game, but was not successful and then therefore discontinued in favor of the strategy of finding which blocks belong to the target structure. This builder played in a first round. In the corresponding second round, the builder embarked on the first strategy.

The remaining three builders (in three first rounds) also presented multiple blocks at once but the set of blocks presented did not have any common properties and seemed random. These builders did not have any apparent systematic strategy and their games did not come to a successful end.

Taking a closer look at the four failed experiments, we find that in one of them, where the builder presented one block at a time, in the end the target construction was almost finished. Architect and builder understood each other, but the architect did not signal an early mistake in the position of one block right away. He waited until the rest of the structure was completed and then tried to address the mistake by means of the introduction of a new signal. This new signal was interpreted by the builder as an *End* signal, leading to the end of the game with one block in a position next to the target one. However for the other failed experiments, the structure at the end of the game was far from the target construction and there was no noticeable progress in all three cases.

Whereas, with the current data and analysis, we cannot yet draw any conclu-

Presentation of blocks	Strategy	Number of games	Successful	Failed
Present one block at a time	Find one block and join right away, repeat	12	11	1
	Find all blocks belonging to the structure, then start joining	2	2	0
Present multiple blocks at a time	No strategy	3	0	3

Table 3.1

sions, still this observation suggests that the way the builders propose next steps and ask for information from the architect is important for the success of the game. Builders seem to build frames and create slots for the architect's input. These frames form the context that shapes the interpretation of the signals. This is similar to how in other cases of asymmetric or restricted communication, as for example in interactions with preverbal infants or in interactions with impaired persons, people provide frames to understand what their interaction partners with their different or limited conversational abilities want to communicate [Ochs 1979, Goodwin 1995].

3.4.4 Additional Observations

This subsection briefly indicates interesting, additional observations we made with our pilot study, as well as interesting considerations for future work.

First of all, we would like to state that the history of the interaction is crucial for understanding meanings. A person who has not witnessed the course of the interaction, is not able to fill in and complete the task without special instructions. We observe a phase of confusion and negotiation at the beginning of the interactions and after that a completion phase in which signal meanings have been constituted. The latter seems to be characterized by smooth, consistent patterns. In the initial phase of negotiation, we observed instances where the players adapted to their partners by changing the meaning of a button when they noticed the other player understands it differently (cf. Figure 3.5 in Subsection 3.4.1). There were for example cases in which the meaning of buttons used to convey a positive or negative feedback reversed.

In contrast, we also observed that some players, both architects and builders, insisted on their strategies, even though the interaction with their respective partner did not work, i.e. they did not agree on any meaning and the task did not progress. Thus, there seem to be leaders and followers in terms of strategies, which could be personality-dependent, but could also manifest their ability to employ a theory of mind.

We also note that when builder and architect switched roles after a first round, their behaviors and performances were influenced (e.g., builder strategies were adopted across rounds). If a second round was systematically part of the experimental procedure, it would be interesting to see whether participants succeed faster in the second game they play with reversed roles and if they adopt similar strategies.

Another interesting aspect concerns timing, not only at which points in time the architect gives feedback and instructions, but also the interplay between the builder's and the architect's actions. The rhythm of the interaction partners' actions might be an important low-level feature in determining whether a certain signal means positive or negative feedback.

While the above points are highly relevant and worth investigating, their detailed examination is beyond the scope of this work.

Meaning switches and reset During the experiment, we noticed some participants were misinterpreting a *Positive feedback* as a *Negative feedback* (and reversely), but most of the time they were able to detect and correct this misunderstanding. In few cases, it was the architect that inverted the meaning of the signals but in most cases it was the builder that had to reinterpret the signals, often after a *Reset* instruction from the architect. The data we collected are not detailed enough for a fine-grained temporal analysis but we were able to count the number of feedback interpretation switches per run. In 5 out of 14 successful games (see Figure 3.9) the architect or the builder changed his use or interpretation of signals between positive and negative feedback.

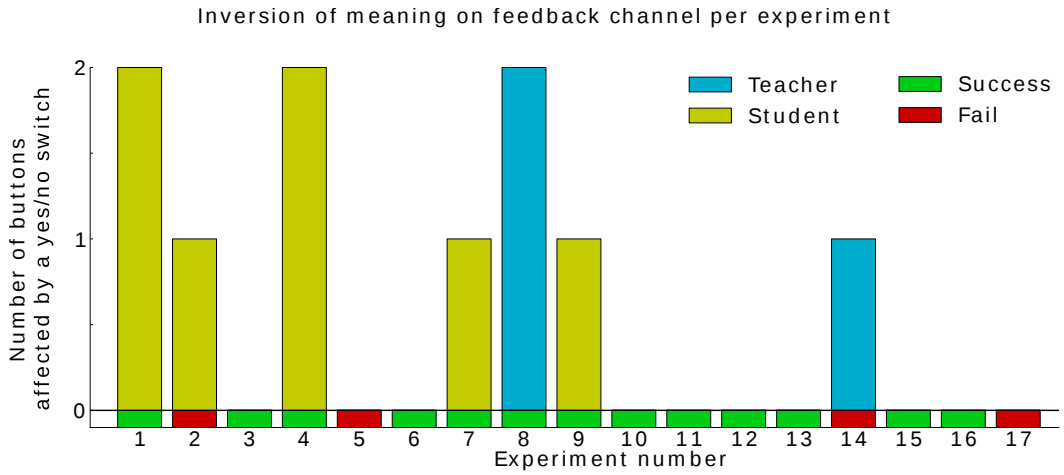


Figure 3.9: Number of signals whose meanings switch between positive and negative feedback during the experiment. In blue, cases where the architect decides to change the meaning of a button from one feedback type to the other. In yellow, cases where the builder changes his/her interpretation of a signal. The colored bar on the bottom indicates if the experiment was successful or not.

Context dependent meaning In several cases the architect pressed all buttons to signify a salient event. This event was either perceived as a *Reset* instruction if the builder felt lost or an *End* instruction if the builder felt confident about his/her understanding of the previous interaction sequences. This is illustrated in figure 3.5, where at $t = 200s$, as players already tried for several iteration with no success, the architect presses all buttons to signify a *Reset*. After this *Reset*, a new set of symbols is used by the architect that is well understood. Finally, to signify that the construction is finished, the architect presses again all buttons simultaneously now with the intended meaning that the task is completed. As the interaction was going well, the builder understood this signal as an *End* signal and the experiment went to a successful end.

As detailed earlier one of the experiments failed even if in the end the target construction was almost finished. Architect and builder understood each other, but an early mistake in the position of one block was not signaled by the architect right away. He waited until the rest of the structure was completed and then tried to address the mistake by means of the introduction of a new signal. Given the context (the interaction was smooth and participants understood each other), the introduction of this new signal was interpreted by the builder as an *End* signal, leading to the end of the game with one block in a position next to the target one.

Timing Figure 3.5 contains information on which signal the architect sends to the builder at which point in time as well at its alignment with the construction progress. Such information allows analyzing individuals' temporal coordination during social interactions, i.e. the timing and interplay of interaction at both a micro and macro scale [Delaherche 2012].

Confirmation bias Some builders were affected by the confirmation bias which is defined as *"the seeking or interpreting of evidence in ways that are partial to existing beliefs, expectations, or a hypothesis in hand"* [Nickerson 1998]. While mistaking negative feedback for positive feedback, participants were progressing far in a wrong direction, even if the signal would seem contradictory for an outside observer. It was difficult for some users to re-assess their belief, they better thought the architect was mistaking or were pursuing in a very improbable direction. Few builders were able to overcome the confirmation bias problem by themselves, leading either to a failed experiment or needed the architect to produce a salient event to reset the experiment. With the recorded data, it is unfortunately not possible to quantify this phenomenon even if the figure 3.9 may provide useful information.

Workspace From our video recording, we observed that some builders (9 out of 17) cleaned their workspace in the beginning of the experiment, such that no block is remains visible. They then tried to maintain a clean workspace during the game, giving them a presentation space, where they could propose blocks in an unambiguous way. Another strategy was pursued by eight builders who from the

beginning kept all blocks on the workspace and therewith enabled the architect to witness the process of choice of block. Of these eight builders, three neatly ordered and aligned their blocks on the workspace and proposed one block at a time by pointing to it. The remaining five builders did not order or align the blocks in any way. These participants opened up a workspace inside the overall workspace (i.e., proposing blocks in-between or next to the rest).

Propositions Essentially the task consisted in two subtasks, finding correct blocks and joining them. For this, participants proposed blocks and positions of blocks in different ways. For proposing blocks in search for a correct one, builders "present" blocks by placing them alone on the workspace or in a separate sub-workspace, they point to the block they wish to receive feedback about, or they lift the respective block to highlight it.

To find at which position a specific block is correctly joined with others, the propositions differ in the level of accuracy and precision of the proposed position. Some builders begin with bumping two blocks together to receive feedback about if they should be joined at all. In some cases the respective block is placed above, below, on the right or on the left of a structure to receive course feedback about the location of the correct position. Another way of presentation is to continuously move the respective block around the structure with expected positive feedback when the correct position is reached. Some builders discretely test or propose positions on the way around the structure by only pausing, joining blocks half way, or fully joining the blocks at each possible position.

3.5 Lessons Learned

We presented a new experimental method that allows studying important aspects of human communication with high relevance to human-robot interaction. We show that two players that never had a chance to interact by the means of a restricted interface before were able to communicate and act upon communicative acts whose meanings were never explicitly negotiated between interaction partners. What can we learn from the experiments? How can it be used for human-robot interaction?

We first link our experiment with the concept of interaction frame defined in introduction (chapter 1). We then describe the main strategy used by our participants. We highlight the active role of the builder in creating slots for the architect to provide information. And further identify the main strategy used to learn the meanings of button presses, which consist of generating interpretation hypothesis and trying to validate or discard them through further interactions.

3.5.1 Use of interaction frames

The experimental setup described above is less constrained than our challenge of *learning from unlabeled interaction frames* defined previously. As a reminder this problem assume the interaction frames associated to the interaction between the

robot and the human is known and only the mapping between teaching signals and their meanings is unknown. Knowing the interaction frame means having access to: (a) the set of possible meanings the teacher can refer to, (b) the details and timing of the interaction, and (c) the constraints that apply on the possible tasks.

In the human-human experiment described in this chapter, the interaction frame is not defined in advance. The meanings associated to the button events are not constrained to belong to a finite set, and the details and timing of the interaction, i.e. the protocol, are also undefined at start. Only the context in which the interaction takes place is provided to both participants, which is to build a flat construction that does not necessarily contain all available blocks.

The first interesting fact is that, while all of participants thought the problem impossible to solve, most of them were able to successfully cooperate under restricted and asymmetric interaction.

The second interesting fact is that users seemed to rely on "usual" interaction frames to make sense of the interaction (cf figure 3.6). Especially, participants came up with strategies involving both the details and timing of the interaction and the possible meanings associated to the button events. In the next two subsections, we will highlight the following observations:

- The timing and alignment of the interaction between both participants quickly converged. Especially the builder seemed to be the leader in the construction of the interaction protocol. With his/her propositions of blocks and positions, the builder provides frames in which he/she creates slots for the architect to provide information.

- The architects and the builders considered only a limited number of meaning types; among which only positive and negative feedback was considered by all participants. Builders seem to rely on the assumption that the signal observed would belong to one of these categories. They then relied on interpretation hypothesis with respect to both the task (i.e. the possible constructions) and the meaning of the signals. By testing several combination of task and signal's meaning, the builder was able to identify the correct signal to meaning mapping, most often leading to a success in the construction task.

3.5.2 Slots of interaction

Signals' meanings are co-constructed by the interaction partners, but the builder's actions seem to play a key role in structuring the interaction. With his/her propositions of blocks and positions, the builder provides frames in which he/she creates slots for the architect to provide information. And thus the builder's created frames constrain the meaning of the architect's input to a large extent.

For example, by cleaning the workspace of all blocks and presenting new blocks one at a time, the builder influences the architect to provide a signal whose meaning can be: "this block belongs/does not belong to the construction" or "this block is blue/red/yellow" for example. This way the builder additionally imposes the timing

of the interaction, e.g. a turn taking social behavior where the builder proposes a new block and waits for a signal from the architect. As a result, the builder is now faced with a similar problem than our problem of *learning from unlabeled interaction frames*, where the meanings are limited to a finite set, known from both partners. The frame created by the builder also defines the association between world's events (e.g. movement of cubes) and instruction signals. However the particular meaning of the button presses inside each frame is still to identify, e.g. whether the observed signal means the block belongs or does not belong to the final structure.

This behavior has also been observed in other asymmetric and restricted interactions involving interaction partners with limited communicational abilities, as for example preverbal infants or impaired persons [Ochs 1979, Goodwin 1995].

Therefore, it might be interesting to consider similar mechanisms of proposition in a learning robot as means to elicit appropriate signals from a human tutor in HRI [Cakmak 2012b, Vollmer 2014b, Cangelosi 2010], especially if the interaction protocol is not explicitly defined in advance. These interesting directions are not the subjects of this thesis, and in our experiments we will assume the human teacher is aware of the interaction frame. It is only in chapter 7.6 that we soften this assumption, assuming a finite set of possible interaction frames is available.

3.5.3 Interpretation hypothesis

Humans are capable of solving the kind of communication problem robots can encounter with humans. We have observed that both builders and architects have preconceptions of what interaction frames the other player is likely to understand, trying to use or interpret signals with respect to those frames. With the "feedback frame" the most commonly thought about and the easiest to understand in the context of our experiment. And with *Reset* and *End* instructions being more frequently considered than guidance, color, or size related instructions.

To solve the restricted asymmetric interaction problem arising from our experimental setup, participants projected the ongoing interaction into those different common interaction frames. They were creating interpretation hypothesis of the signals and behaviors of each other, which were later discarded or validated in light of the next observations. Especially, a hypothesis is retained if its predictions are more coherent with the history of interaction.

For example, let's consider you are the builder and you present only two blocks on the workspace to be visible to the architect. You then test one by one every possible stacking combination with these two cubes. Between each test you wait few seconds to observe the signal from the architect. Given your behavior, you expect to elicit a yes/no type of signal from the architect and will start hypothesizing the signals you receive belong to this category.

Therefore, after having tried all possible stacking combination of blocks, you expect only two possible outcomes. The first possibility is that you received the same, unique, signal all along, and you may assume that the two blocks selected do not stack together in the final construction. In addition you could also assume that

the signal you observed mean your actions were "incorrect". The other possibility is that you observed two different signals, with one being way more frequent than the other. In such case, you may hypothesize that the less frequent signal corresponds to an "incorrect" meaning. Indeed, given the construction task, there should be more incorrect possible stacking than correct stacking. Therefore the other signal should mean "correct", and the associated stacking of block should be part of the current structure. In that case, you can stack the block together and try to introduce a new block, that time knowing what signal means "correct" and what signal mean "incorrect".

But things are not that easy. The architect might not have understood your behavior or the fact you asked for a yes/no type of instruction. It might have send always the same signal but asking you to take a new block. In that case, given your hypothesis, you might believe this signal means "incorrect" instead of meaning "pick a new block". But the architect may also have tried to guide your movement towards the correct position (using "above", "under", "to the left", or "to the right" instructions), in such case you should have noticed that you received more than two different signals and would have to reconsider your hypothesis.

Therefore it might be useful to check if the behavior of the architect could not belong to another interaction frame. You might try to find a situation allowing differentiating between the remaining hypothesis. And after a more of less lengthy procedure, you might end up being sure of the architect intended meanings and succeed in the construction task.

As a final note, on top of all these hypotheses, you cannot assume the architect behavior is constant through time because he also tries to adapt to your behavior. This makes our human-human experiments way more complex than the problem considered in this thesis, where we consider the teaching behaviors of our users are constant through time.

The example we provided is well illustrated by the first four minutes of interaction of the experiment presented in Figure 3.5. We can observe that, during these four minutes, both the builder and the architect change frequently their use and interpretation of the signals. After a *Reset* event is sent by the architect, the interaction starts again on a more structured interaction, which finally led both participants to agree on a communication system and to succeed at the co-construction task.

Based on our observations, we can aim at constructing robots capable of learning a task from human instructions without programming them in advance to understand the human communicative signals. To do so, we should inform the robot about the interaction frame, which indicates: (a) the set of possible meanings conveyed by the teacher (e.g. the teacher use only positive and negative feedback), (b) the details and timing of the interaction (such as to map teaching signals to world events), and (c) some constraints on the possible tasks (such as to limit the search

space of the robot). Given this information, by making hypothesis on the task, the user can generate interpretation hypothesis of every users' communicative signal according to each hypothesized task. As the teacher only follows one of the task, the hypothesis from which emerges a better coherence between the interaction history, the interaction frame, and the task, is likely to be the one the teacher as in mind. We formalize this idea in next chapter and provide simple visual examples of both the problem and the properties we exploit to solve it.

Chapter 4

Learning from Unlabeled Interaction Frames

Contents

We identified a potential mechanism for robots to learn a new task from human instructions without programming them in advance to understand the human instructions signals. This mechanism is based on the generation of interpretation hypothesis of the teaching signals with respect to specific constraints from the task and the interaction frame. It hypothesizes that the correct hypothesis will explain better the history of interaction.

In this chapter, we exemplify the problem in a simple seven discrete states world, remind the underlying assumptions and define the notation used. We illustrate the interpretation hypothesis mechanism on our visual example and, based on our observation, we define the metric our algorithm will rely on. We then apply our algorithm to a pick and place scenario using a six degrees of freedom robot and speech utterances as the modality of interaction. We show that our algorithm is able to identify a task in less than one hundred iterations when the teacher is providing feedback signals whose mapping to their associated meaning is a priori unknown. We further show that the system is robust to some teaching mistakes and that the knowledge learned during a first experiment can be reused for learning a second task faster. Finally, we will show that two different simple action selection methods for our robot lead to differences in learning efficiency. This observation opens the question of how our robot can plan its action to improve its learning performances, which will be investigated in the next chapter.

4.1 Problem formulation

In chapter 1, we defined the problem of *learning from unlabeled interaction frames*. In short, a human instruct a robot to perform a task by providing it instructions through communicative signals. The problem is that the robot does not know the task, neither the mapping between the teacher' signals and their meanings. The robot is not teleoperated but rather decide by itself which actions to perform. The task is sequential which means the robot should perform a sequence of multiple actions to fulfill it. We exemplify with the following example.

4.1.1 Example of the problem

We present a T world example (see Figure 4.1) that will follow us during the remaining of this thesis. In this example, an agent lives in a discrete seven states world

The work presented in this chapter has been published in [Grizou 2013c]. Code is available online under the github account `https://github.com/jgrizou/` in the following repositories: lfui, experiments_thesis, and datasets.

that has a T shape. The agent can perform four different actions (go left, right, up ,and down).

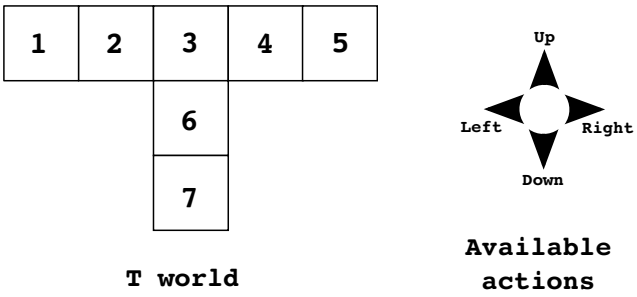


Figure 4.1: The T world and the available actions.

A simulated teacher wants the robot to reach, and stay at, the left edge of the T world (i.e. state 1). To this end, the teacher provides feedback information to the robot. Feedback signals are represented as two dimensional feature vectors and can have two different meanings: "correct" or "incorrect". As depicted in Figure 4.2, we assume these signals are randomly generated by two multivariate normal distributions, one for each meaning. We associate green and red colors respectively to signals of "correct" and "incorrect" meanings. When the teacher wants to send a feedback of meaning "correct", he samples a signal from the right, green, Gaussian. Respectively, a signal of meaning "incorrect" will be generated on the left side of the feature space. These signals are represented in a two dimensional feature space, which could represent any modality used by the teacher to communicate with the robot, such as speech, gestures, facial expression, or even brain signals.

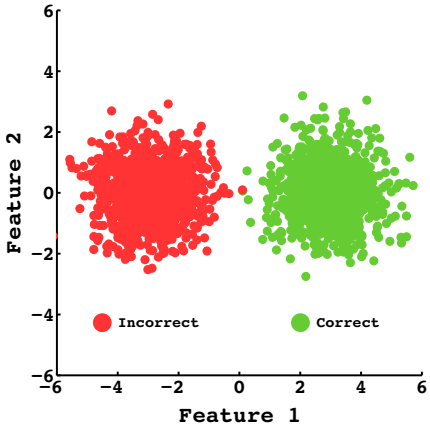


Figure 4.2: The feedback signals used in our visual examples. A signal of meaning "correct" will be generated on the right side of the feature space, and a signal of meaning "incorrect" will be generated on the left side. Importantly, the agent will never have access to the label information, represented by the color of each signal.

The interaction between the agent and the teacher is turn-taking. First, the agent, which is in a particular state, performs one action and transitions to its next

state. The teacher is observing the robot and evaluates the robot's actions with respect to the task he has in mind (i.e. the robot should go and stay in state 1). The teacher then sends the corresponding signal to the robot. However, the robot neither has access to the task the user has in mind, nor it has access to the meaning of the signal sent by the teacher. For the sake of the example, we assume that there are only two possible tasks, reaching G1 or G2.

For example, as depicted in Figure 4.3, the agent starts in state 3, performs action left, and ends-up in state 2. The teacher wants the agent to go to G1, therefore he sends a signal of meaning "correct" (i.e. in the right part of the feature space). Note that the signal shown in Figure 4.3 (left) is neither green nor red, its label is undefined.

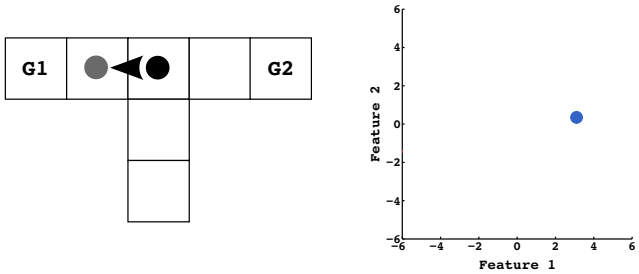


Figure 4.3: The teacher provides a feedback signal after each action of the agent. The agent starts in state 3, performs action left, and ends-up in state 2. The teacher wants the agent to go to G1, therefore he sends a signal meaning that the previous action was "correct" with respect to the goal. The signal is on the right side of the space as described in Figure 4.2. However the agent does not have access to the label associated to this signal, it only observes a point in a two dimensional space.

After performing several actions randomly, the robot ends-up with a lot of observations associating a state, an action and a feedback signal. As depicted in Figure 4.4, we can observe that two clusters have emerged in the feature space. A straightforward assumption is that one cluster is associated to the "correct" meaning, and the other to the "incorrect" meaning. We will see how this assumption of consistency in the signals can be exploited in the coming sections.

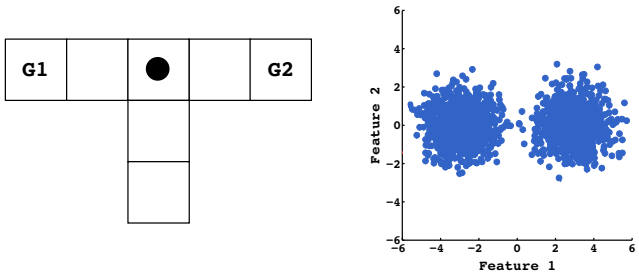


Figure 4.4: After performing several random actions, the robot ends-up with many of observation associating a state, an action and a feedback signal.

4.1.2 What the agent knows

The problem described in this section is impossible to solve without further information. Indeed, even if the agent was able to identify the two clusters, it does not have access to the meaning associated with each cluster. In practice it would be easier if the robot had access to the mapping between teaching signals and their meanings. A typical solution is therefore to rely on a phase of calibration, where the system is given signal-meaning pairs and learns the mapping using a supervised learning algorithm. Given this information, in our example of Figure 4.3, it becomes trivial to identify the task. Starting in state 3, if the robot does action "left", it ends up in state 2, and if it receives a signal of meaning "correct", then the correct task is to reach the left edge of the T marked by G1.

As mentioned before, in this work the robot cannot rely on the phase of calibration. However the robot has access to the interaction frame, which provides theoretical information about the human teaching behavior. The robot knows:

- **Details and timing of the interaction.** After each action, the robot waits for a signal from the teacher. This signal provides information related with the action the robot just performed.
- **The set of possible meanings the human can refer to.** The teacher assesses the last action of the robot with respect to an unknown task. The signals' meanings can be "correct" or "incorrect".
- **Constraints on the possible tasks.** There are only two possible tasks, reaching the left (G1) or the right (G2) edge of the T world.

In addition the robot has access to the $Frame(Context, Task)$ function that, given a context of interaction and a task, returns the meaning intended by the teacher. For example, the robot knows that if it moves from state 3 to state 2, and that the human wants it to go in G1, then the signals received from the human means "correct".

$$\text{“}correct\text{”} = Frame((s3 \rightarrow s2), G1)$$

Respectively, if the robot moves from state 3 to state 2, and that the human wants it to go in G2, then the signals received from the human means "incorrect".

$$\text{“}incorrect\text{”} = Frame((s3 \rightarrow s2), G2)$$

4.2 What do we exploit

Following our T world example, we now present a visual representation of the interpretation hypothesis mechanism. From the observation made in chapter 3.5.3, the robot will generate interpretation hypothesis of the signals with respect to all possible tasks. For a particular task hypothesis, the robot will assign hypothetic meanings, or labels, to the human signals according to its previous actions and

knowing the meanings are either "correct" or "incorrect". The system is "reasoning" as follow: *"If the human wants me to solve task G1, then when I performed action "left" in state* 3*, its feedback signal should mean "correct" "*. For the sake of our example, we only consider two hypothesis, G1 and G2, as depicted in Figure 4.4.

4.2.1 Interpretation hypothesis

For each action, the robot receives raw unlabeled two dimensional signals. Following the above explanation, for a particular hypothesis (G1 or G2), the robot can assign hypothetic meanings to the human signals knowing they are limited to a finite set and according to the interaction history. We assume our teacher is optimal and therefore assume our agent is aware of the optimal policies for each task (see Figure 4.5), which can be used to interpret the human signals.

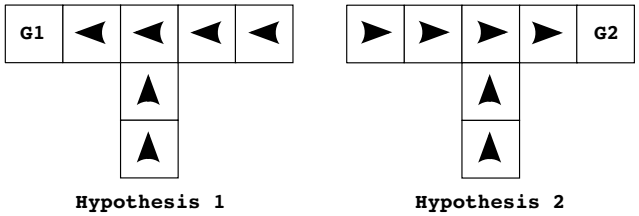
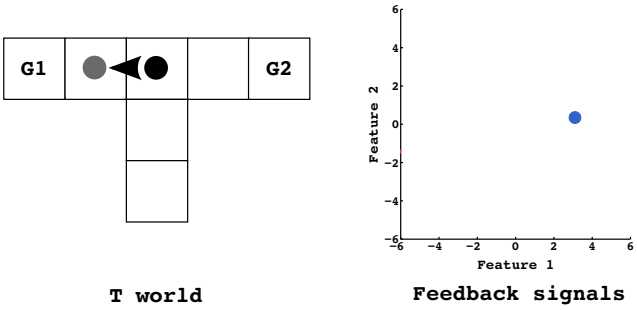


Figure 4.5: Optimal policies associated to the two task hypotheses G1 and G2 in the T world. Such policies are known by both the human and the agent, and allow the agent to interpret a human signal with respect to a given task.

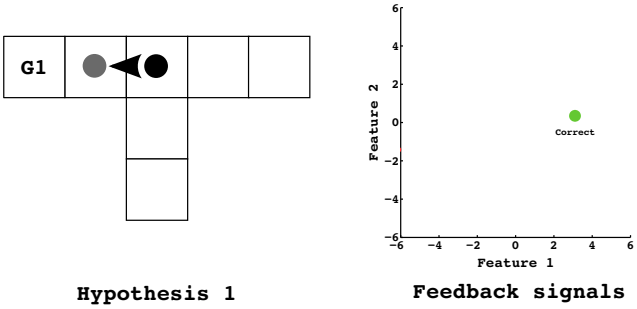
The teacher wants the agent to go to G1. The agent starts in state 3, performs action left, and ends-up in state 2. The teacher sends a signal in the right part of the feature space, meaning that the previous action was "correct". However the agent does not have access to the label associated to this signal and it only observes a point in a two dimensional space (Figure 4.6a). The agent generates interpretation hypothesis according to G1 and G2. With respect to G1, the action was "correct" (Figure 4.6b), while with respect to G2 the action was "incorrect" (Figure 4.6b).

By repeating this process for several iteration steps, with the agent taking random actions, the system end-up with a set of possible interpretation of the human teaching signals (see Figure 4.7). But as the user has only one objective in mind, in our case G1, only the correct interpretation will assign the correct labels to the observed signals. We say that the corresponding hypothesis exhibit a coherence between the signals and their associated meanings.

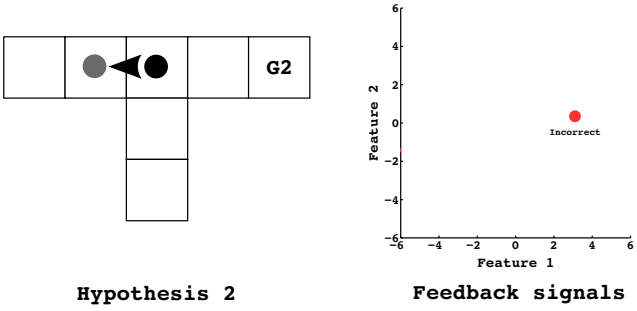
Part of the *learning from unlabeled interaction frames* problem defined in chapter 1.3 is the assumption that the user is coherent and uses always the same kind of signal for the same meaning. By visual inspection, we can infer that hypothesis G1 is the correct one as the resulting mapping between signal and meaning is more coherent. The key challenge is to find out how to identify coherence between the spatial organization of signals in the feature space and their associated labels with the tools available to the robot, i.e. algorithmically.



(a) Feedback signal as received by the agent without label.



(b) Feedback signal labeled according to G1.



(c) Feedback signal labeled according to G2.

Figure 4.6: Interpretation hypothesis made by the agent according to G1 (4.6b) and G2 (4.6c). The agent starts in state 3, performs action left, and ends-up in state 2. The meaning of the signal is different for both hypotheses.

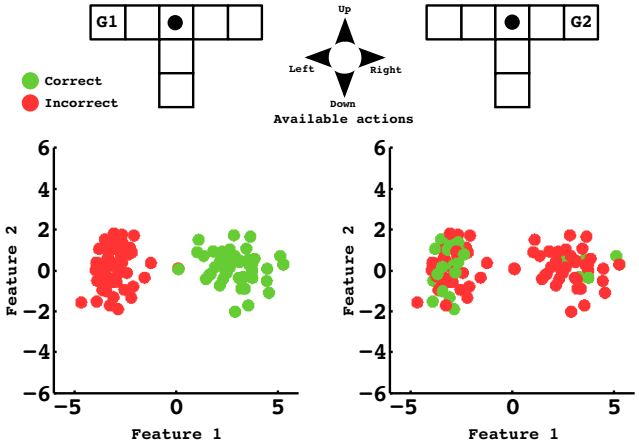


Figure 4.7: Interpretation hypothesis made by the agent according to G1 and G2 after many interaction steps. The teacher's task is to have the agent reach G1. The agent is exploring all the state space randomly. The labels associated to the task G1 are more coherent with the spatial organization of signals in the feature space.

We will formalize this idea in section 4.4. Before that, we add two comments to this section and we summarize all the underlying assumptions of our problem in section 4.3.

4.2.2 Different frames

In our example, we considered only the feedback frames, where the user assesses the robot's actions. In this thesis, we will also consider other interaction frame, such as the guidance frame where the user indicates to the robot which action to perform next. We will provide several visual examples of the guidance frame in the following of this chapter.

4.2.3 Why not a clustering algorithm

When we first look at the unlabeled signals (see Figure 4.4), the first approach that comes to mind is to use a clustering algorithm to identify the two clusters in the feature space. For simple datasets, like the one used in our example, a clustering algorithm will find the two clusters. However, without any additional information, it is impossible to know which one is associated to the meaning "correct" or to the meaning "incorrect".

More importantly, clustering algorithms are prone to local extrema in the optimization process and for datasets in high dimension with overlapping classes it is unlikely to find the correct underlying structure of the data. Our approach has the advantage to generate hypothetic labels allowing fitting a classifier for each task hypothesis.

4.3 Assumptions

As described in the introduction, a number of assumptions are made about the information accessible to the robot and the constraints applied to the interaction. We remind them again briefly here and, now that we have exemplified the mechanism of interpretation hypotheses, we present an additional required property that the world must hold for our problem to be solvable. We will see that, in some cases, it is impossible to discriminate between two hypotheses because they result in symmetric interpretations of the signals. We describe this properties in subsection 4.3.3.

4.3.1 Frames

Our first assumption is that the robot and the human are aware of the frame in which the interaction takes place. This frame regulates the interaction between the two partners, it includes:

- **Details and timing of the interaction.** It corresponds to when and how the user will provide instruction signals. For example, the human sends a signal to the robot after every robot's actions. Another example is a human providing a feedback signal between 0.2 and 2 seconds after the robot's action [Knox 2009b].

- **The set of possible meanings the human can refer to.** As depicted before, the set of meaning may include "correct" and "incorrect" for those cases where the user is assessing the robot's actions. It could also be the set of action names when the user provides guidances on what to do next.

- **Constraints on the possible tasks.** The general context of the teaching process is known. For example the robot is aware that the human wants it to reach a specific room in the house, and not to take an object from the fridge. This limits the number of hypotheses the robot can create about what the user has in mind.

By combining those three aspects of an interaction frame, the robot can create a set of interpretation hypothesis for the received teaching signals. For one possible task, and given a specific context (e.g. state and action performed in the environment), the robot can infer the meaning intended by the human user ($Meaning = Frame(Context, Task)$). By doing so for every possible task, the agent creates a set of interpretation hypothesis, which we rely on to find the task taught by the user, as well as the signal to meaning mapping.

To do so we rely on specific properties of the human teaching signals.

4.3.2 Signals properties

We make two assumptions about the human teaching signals properties:

- If the true intended meaning associated to each user signal was known, it would be possible to train a classifier with better than random accuracy. We will see in chapter 5.6 that the performance of the system are highly impacted by the quality of the training data.

- The teacher is consistent in its use of teaching signals and will always use the same kind of signals to mean the same things. For the case of two buttons, he will always use the same button for the same meaning. It also applies for speech, facial expression, gestures, or brain signals.

Those two properties are typical assumptions in human-robot interaction scenario, we simply assume we can rely on the teacher behavior and that we could, in theory, learn a decoder of the human teaching signals.

However there is one practical constraint that differs from more standard human-robot interaction scenario. Here, in theory, we cannot know in advance if a signal to meaning mapping can be learn. Indeed we do not have access to a database of signal-meaning pairs to train a classifier first, which allow trying different feature extraction processes or different classifiers beforehand. This limitation requires ensuring the representation of the signal and the classifier allow to learn a usable decoder.

We will see from results in chapter 5.6 that our algorithm can cope with highly overlapping data where the classifier produces close to random prediction.

4.3.3 World properties and symmetries

There are some cases where different hypothesis are not distinguishable. As the robot do not have a direct access to the true intended meaning of the teaching signals, it can only rely on the interpretation hypothesis made for each task.

Two problems could appear: (a) two hypothesis may share the same interpretation model and cannot be differentiated as they attribute the same meanings to the signals, and (b) two hypothesis may end up with opposite interpretations that are both as valid. .

For those cases where two hypothesis share the same interpretation model, either the task are the same with respect to the user, either some parts of the problem are hidden to the human, which can not provide appropriate instructions. These questions are the core of the theoretical analysis of Cederborg's thesis [Cederborg 2014a, Cederborg 2014b]. We do not consider this problematic in this thesis and assume the world properties ensure that two hypothesis will never share the same interpretation model. Most of the hypothesis will share parts of the interpretation model but it will always exist one situation, i.e. one state-action pair, where two interpretation models differs.

For those cases where two hypotheses end up with opposite interpretations that are both valid, we illustrate the problem for the case of both feedback and guidance instructions using a visual example.

Symmetries: the feedback case

We present the line word in Figure 4.8 which contains only the top T bar of the T world. This world is well suited to describe the symmetry problem.

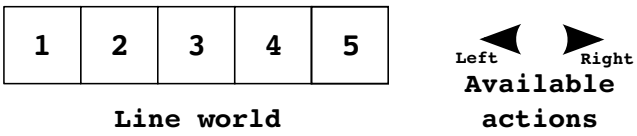


Figure 4.8: The line world and the available actions.

The interaction follows the same protocol as in previous examples. As depicted in Figure 4.9, after several interaction steps, the interpretation hypothesis for G1 and G2 display symmetric properties. Indeed, according to G1, signals on the left side of the feature space mean "incorrect", and signal on the right means "correct". Inversely, according G2, signals on the left side of the feature space mean "correct", and signal on the right means "incorrect". Therefore, even if the interpretation of the signals differs between each hypothesis, the two interpretations are equally coherent. As the optimal policies to reach each of the two goal states are opposite in every state, an action that triggers a "correct" feedback with respect to G1, triggers a "incorrect" feedback for G2 and vice versa. It is therefore impossible to know the true associated meaning of the signals without further information.

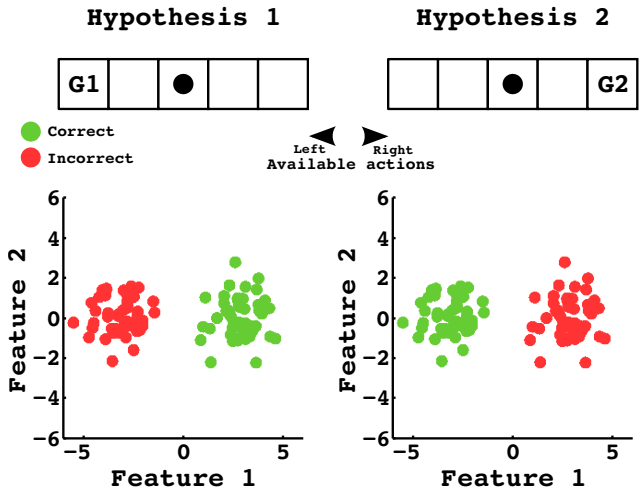


Figure 4.9: Interpretation hypotheses made by an agent receiving feedback on its action in the line word and where the hypothetic tasks are G1 or G2. The agent can only perform right or left actions which results in symmetric interpretation hypothesis of the feedback signals.

It is theoretically impossible to differentiate symmetric task hypotheses, therefore we will not consider environments holding this symmetric property. One way to bypass this problem is to add a "no move" action, as illustrated in Figure 4.10, that is valid only at the goal state

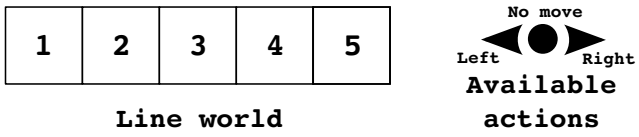


Figure 4.10: The line world and the new available actions, including a "no move" action.

When taking the "no move" action the agent does not change position. This action allow to break the symmetry effects, as its interpretation will be the same for all states that are not in the set of hypothetic goal state, i.e. all state but G1 and G2. In other words, if the agent performs action "no move" in state 3, the user will produce a signal of meaning "incorrect" because the agent is not progressing towards the goal state(G1 here). But the agent did not progress either towards the G2. Therefore the signal will be interpreted as "incorrect" by the two interpretation hypothesis, breaking the symmetry problem. The interpretation results after several iteration steps, and using the new "no move" action, are depicted in Figure 4.11.

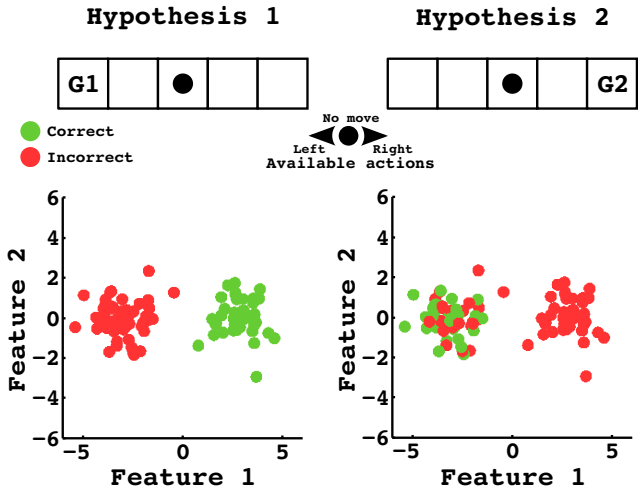


Figure 4.11: Interpretation hypotheses made by an agent receiving feedback on its action in the line word and where the hypothetic tasks are G1 or G2. The agent can perform right, left, or "no move" actions. As opposed to Figure 4.9, the "no move" action allows to break the symmetry of interpretation between G1 and G2.

Symmetries: the guidance case

This problem of symmetries also applies to the guidance frame. Under the guidance frame, the set of possible meaning includes the name of all possible actions. If the agent can only choose between the "right" and "left" actions, the teacher can only advise for "left" and "right" actions. We represent the guidance signals from the teacher in a two dimensional feature space as shown in Figure 4.12.

We can easily understand that if the teacher can only advise for "left" and "right"

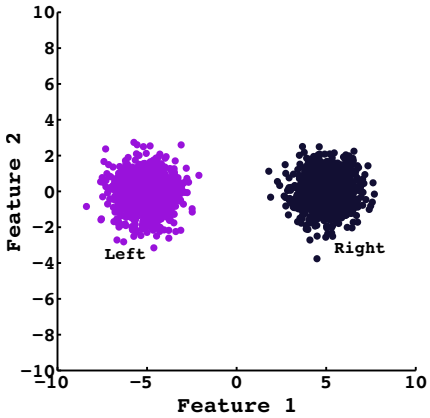


Figure 4.12: The guidance signals used by our simulated teacher in our line world visual examples with two actions.

actions, the interpretation hypothesis for G1 will be symmetric as the one for G2. As our user wants the robot to reach G1, it will only produce "left" guidance signals, i.e. signals in the left part of the feature space. And the "right" commands will never be used. However, these signals will be interpreted as meaning "left" according to G1, and "right" according to G2. Yet the two interpretation models are equally coherent. The resulting interpretation hypothesis are shown in Figure 4.13.

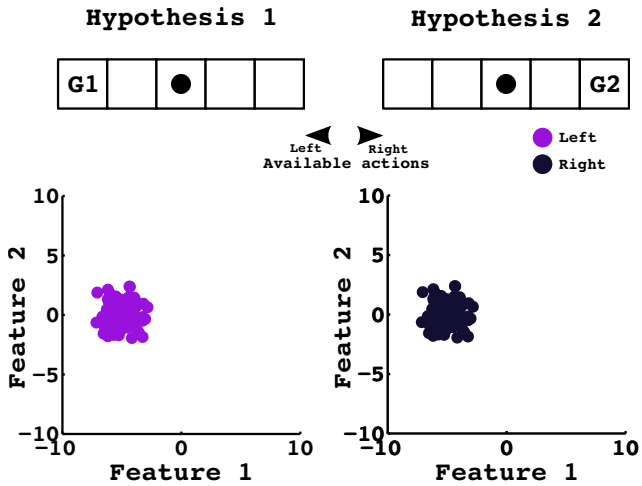


Figure 4.13: Interpretation hypotheses made by an agent receiving guidance on its actions in the line word and where the hypothetic tasks are G1 or G2. The agent can only perform right or left actions which results in symmetric interpretation hypothesis of the guidance signals.

As for the feedback case, introducing a "no move action" allow to break the symmetry. With the "no move" action available, the user can now produce three different kinds of meaning, which is represented by three different clusters of signals in the feature space (see Figure 4.14).

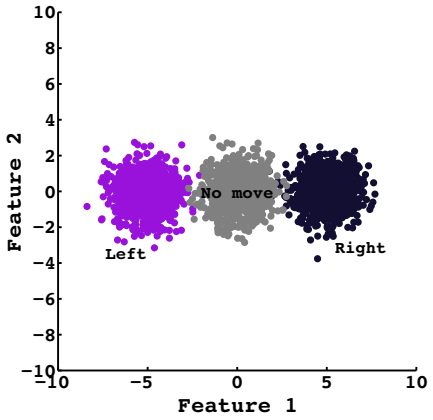


Figure 4.14: The guidance signals used by our simulated teacher in our line world visual examples with three actions.

The "no move" signal will be used only at the goal state. As this state is not the same for each hypothesis, the "no move" signals break the symmetry. The resulting interpretation hypothesis are shown in Figure 4.15.

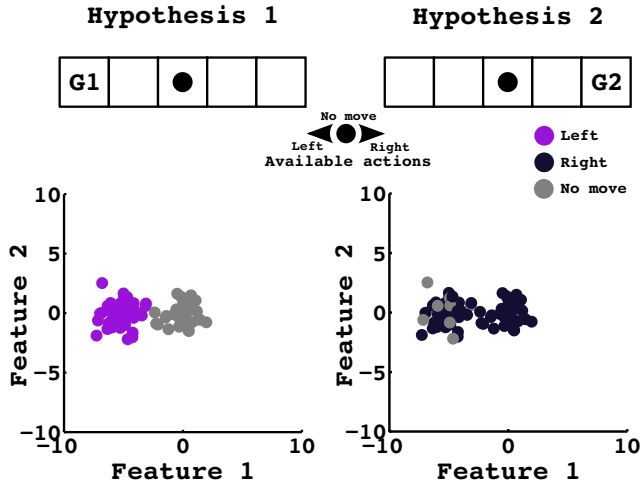


Figure 4.15: Interpretation hypotheses made by an agent receiving guidance on its actions in the line word and where the hypothetic tasks are G1 or G2. The agent can perform right, left, or "no move" actions. As opposed to Figure 4.13, the "no move" action allows to break the symmetry of interpretation between G1 and G2.

As it is theoretically impossible to differentiate symmetric task hypotheses, we will not consider environments holding this symmetric property.

4.3.4 Robot's abilities

We further assume the robot is able to plan its action in order to fulfill a specific task. In other words, if the robot knew what the user wants it to do, it will be able to do it. It implies the robot has full knowledge of the world dynamics and knows how to make a plan. This way the robot can understand the theoretical relation between one action, a specific task and a signal of the user; and therefore create interpretation hypothesis.

The following of this chapter will consider the full set of assumptions defined above.

Most of these constraints are typical from interactive learning experiments. Several aspects are often more constrained. Especially, either the signal to meaning mapping is known in advance, and the agent infer the task based on the known instructions [Kaplan 2002, Chernova 2009, Knox 2009b], either the task itself is known, allowing the robot to assign meanings to the teaching signals such that the signal to meaning mapping can be learned (e.g. the calibration phase for BCI systems). Our method generalizes over these approaches as we neither need to know the task, nor the signal-to-meaning mapping.

Other constraints are not always applied, such as the ability of the robot to plan its action, or the fact that a finite number of tasks are defined in advance.

The ability to plan is linked to the need for the robot to interpret the signals of the user in different situations. To do so the robot needs to be able to project itself in the future to judge of the "long term" effects of its actions.

The finite set of task hypothesis is more of a practical constraint. Considering an infinite set of task hypothesis would add another layer of complexity. Given our interpretation hypothesis mechanism, we would have to sample a finite number of hypotheses. Then given the results of our hypothesis based method on this finite set, we would have to re-sample some new hypothesis and test them again until some stopping criterion. This sampling process is logically less reliable than assuming the correct task belongs to a finite set. Therefore, in our main experiments, we only consider problems where a finite set of task hypothesis can be defined. It is only in chapter 7.5 that this assumption is removed.

4.4 How do we exploit interpretation hypotheses

As exemplified in Figure 4.7, generating interpretation hypothesis for each possible task allows to find out the task the user has in mind. As the user has only one objective in mind, only the correct hypothesis will assign the correct meanings to the observed signals. In our example, we can identify this task visually, by looking at the coherence between the spatial organization of the signals and their associated meanings. But our robots and algorithms cannot use our visual intuition. The key challenge it to find out a measure that can reflect the coherence between the spatial organization of the signals and their associated meanings.

As a measure of coherence we can measure the quality, i.e. the accuracy, of a decoder trained on each hypothetic dataset of signal-label pairs. As for the wrong hypotheses some signals are not associated with their correct meanings, the quality of the resulting classifiers should be worst than the quality of the classifier trained on the correct hypothesis.

For example, if we assume the signals generated by the teacher can be separated by a quadratic curve, and following our T world example (cf. Figure 4.7), for each task, we can use the quadratic discriminant analysis (QDA) [Lachenbruch 1975] approach to fit a classifier on the data. For two classes, this classifier resumes in computing the maximum likelihood for the mean and covariance matrix associated to each labels. The results of this process is illustrated in Figure 4.16.

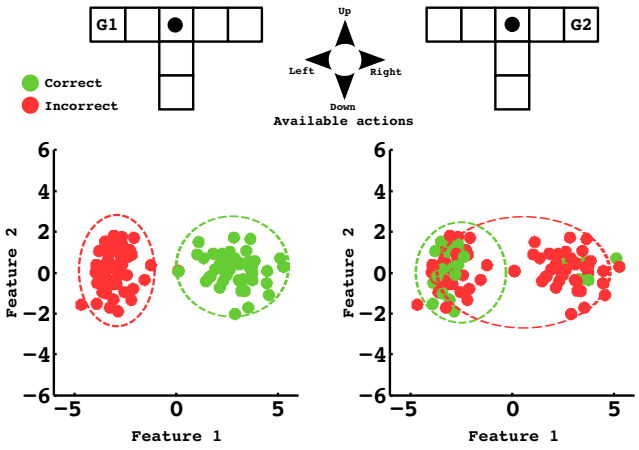


Figure 4.16: Interpretation hypotheses made by the agent according to G1 and G2 after many interaction steps. For each class, we compute a Gaussian distribution shown as a dotted line (approximated by hand). The teacher wants the agent to reach G1. The agent is acting randomly. The labels associated to the task G1 are more coherent with the spatial organization of signals in the feature space. It can be measured by the difference in classification performances made by each Gaussian classifier.

By computing the performance of the resulting classifiers, we can test which hypothesis satisfies better the assumption that the signals can be separated by a quadratic curve. As stated in the previous section 4.3.2, here the choice of the classifier encodes our hypothesis on the underlying structure of the data.

The following of this section formalizes this idea. Next section presents results on a pick and place robotic scenario where the user provides instructions using speech utterances. We use the term label to refer to the meaning associated to user's signals.

4.4.1 Notation

We consider interaction sessions where a machine can perform discrete actions from a set of available actions $a \in \mathcal{A}$ in an either discrete or continuous state space $s \in \mathcal{S}$. The user, that wants to achieve a task $\hat{\xi}$, is providing instructions to the machine using some specific signal e, represented as a feature vector $e \in \mathbf{R}^d$. The task is sequential meaning it is completed by performing a sequence of actions. The machine do not have access to the task the user has in mind, as well as to the actual meaning of each user's signal. Its objective is to simultaneously identify the task and learn to decode user's signals. To achieve this, it has access to a sequence of triplets in the form $D_M = \{(s_i, a_i, e_i),\ i = 1, \ldots, M\}$, where s_i, a_i and e_i represent, respectively, the state, action and instruction signals at time step i. D_M represents the history of interaction up to step M. The behavior of the machine is determined by the actions $a \in \mathcal{A}$ and the corresponding transition model $p(s' \mid s, a)$.

We assume the system has access to a set of task hypothesis $\xi_1, \ldots, \xi_T$ which includes the task $\hat{\xi}$ the user wants to solve. We assume instruction signals e have a finite and discrete number of meanings $l \in \{l_1, l_2, \ldots, l_L\}$ which we call labels. The machine knows the set of possible labels. We further consider that the agent is given a frame function that given a state s, an action a and a task ξ returns a label l. We will formalize our algorithm in terms of probabilities, therefore the frame represents the conditional probability of a label given a state, an action, and a task, written as $p(l|s, a, \xi)$.

Given this frame, the history of interaction D_M, and the set of possible task $\xi_1, \ldots, \xi_T$, we can generate interpretation hypothesis. For a particular task ξ, we can associate a label (or probability of label) to each signal according its associated state and action. For each task, this result is a dataset of signal-label pairs. As there are T task hypotheses, we end up with T hypothetic sets.

We assume that given such one set of signal-label pairs, it is possible to compute one decoder that classifies signals e into labels l, which we also call the signal to meaning mapping. The parameters of such a model will be denoted by θ. We formalize the decoder function as the conditional probability of a label given a signal and a set of parameters, written as $p(l|e, \theta)$.

As both the frame and the decoder refers to probabilities of labels, we will use different notation for the labels given by the frame, which we denote l^f, and predicted by the classifier, which we denote l^c.

Finally, for a given iteration step i, we will subscript our notation with a i referring to the iteration number. i will be the only letter refering to iteration numbers. We will abuse notation for labels and l_i will refer to a label at step i, e.g. l_i^f is the label given by the frame at iteration i. Any other subscripting letter for l will refer to a particular class, e.g. l_k is the k class.

4.4.2 Estimating Tasks Likelihoods

We remind that, to measure the coherence of each interpretation hypothesis, we measure the quality, i.e. the accuracy, of a classifier trained on each hypothetic dataset of signal-label pairs. More precisely, for each interpretation hypothesis, we will compute the probability that every observed signal is correctly classified. We remind that the agent never has access to the true labels of the data, therefore, here, a "correct" classification always refers to the hypothetic labels associated to each task.

The probability that one signal is correctly classified is the sum across all labels of the probabilities that a signal is of a given label times the probability that the model classifies it as being of this same label. Given an interaction tuple (s, a, e), a task ξ, and a classifier θ, we can compute the probability that the signal e is correctly classified according to the frame as:

$$p(l^c = l^f | s, a, e, \theta, \xi) = \sum_{k=1,\dots,L} p(l^c = l_k | e, \theta) p(l^f = l_k | s, a, \xi) \tag{4.1}$$

where we assume independence between l^c and l^f. There exists a dependence between the state-action pair considered (s, a) and the meaning of the signal receive (e), but this relation only exists with respect to the task the user has in mind $\hat{\xi}$, which is unknown to the agent. The role of our algorithm is to identify this task. Therefore when evaluating a signal, our system should not have any a priori about the label of such signal, but only rely on the classifier's prediction.

This equation estimates the joint probability for one iteration step, i.e. given only one interaction tuple (s, a, e), and assuming the classifier is already given. We should now compute the probability that all the labels expected by the frame and all the labels predicted by the classifier match together given the history of interaction and for a hypothesized task. Given the full interaction history D_M up to time step M, and a task ξ_t, we can infer the expected labels $l^f_{1,\dots,M,\xi_t}$ associated to the signals $e_{1,\dots,M}$, and compute the associated classifier represented by the set of parameters θ_{M,ξ_t}.

For clarify, we simplify our notation and remove the ξ_t superscript. It is important for the reader to keep in mind that the robot will never have access to the true intended meaning of the users, therefore, as soon as we infer labels, they are always linked to a hypothesized task. Note that the tuple (s, a, e) are observations independent of the task.

Given the classifier θ_M associated to the task ξ_t at time step M, the probability that every expected and predicted labels match together, which we call the likelihood of the task ξ_t, is given by:

$$\begin{aligned} \mathcal{L}(\xi_t) &= \prod_{i=1,\dots,M} p(l^c_i = l^f_i | D_M, \xi_t) \\ &= \prod_{i=1,\dots,M} \sum_{k=1,\dots,L} p(l^c_i = l_k | e_i, \theta_M) p(l^f_i = l_k | s_i, a_i, \xi_t) \end{aligned} \tag{4.2}$$

This equation computes the odds that all the predictions made by the classifier equals the labels used to train this classifier. However, the classifier θ_M is here computed using all the history of interaction including all the pairs (e_i, l_i^f). This may lead to overfitting problems. For example, if we use a simple nearest neighbor classifier between the provided signal-label pairs, Equation 4.2 will computes a likelihood of 1. Indeed, we train and test on the same dataset. Therefore, we should only estimate the likelihood on signals not used to train the classifier.

What we really want to test is if the system is able to make correct prediction about what the frame will predict for a new, never observed, situation, i.e. a new tuple (s, a, e). One possible option is to incrementally update the likelihood of each task as soon as new data comes in:

$$\begin{aligned}\mathcal{L}_i(\xi_t) &= p(l_i^c = l_i^f | D_i, \xi_t) \;\; \mathcal{L}_{i-1}(\xi_t) \\ &= \left(\sum_{k=1,\dots,L} p(l_i^c = l_k | e_i, \theta_{i-1}) p(l_i^f = l_k | s_i, a_i, \xi_t) \right) \;\; \mathcal{L}_{i-1}(\xi_t) \end{aligned} \quad (4.3)$$

where θ_{i-1} is the classifier trained on all the past observations up to time $i-1$ and according to the label generated from task ξ_t. And with $\mathcal{L}_0(\xi_t)$ being the prior at time 0 (before the experiment start) for the task ξ_t, usually uniform over the task distribution.

While this is a good enough option as it will be demonstrated in the remaining of this thesis, it does not use all available information. Indeed, the update that was made at time $i-10$ is now out of date as, at time i, we now have 9 more observation tuples available that may change our classifier. Therefore, it would be better to reassess the performance of the classifier given the full set of observation. To do so, and in order to avoid the problem of overfitting, the classifier should be trained on all data but the one tested. We denote by θ_{-i} a classifier trained on all data available up to time M but the one of time step i. We can now rewrite the likelihood as:

$$\begin{aligned}\mathcal{L}(\xi_t) &= \prod_{i=1,\dots,M} p(l_i^c = l_i^f | D_M, \xi_t) \\ &= \prod_{i=1,\dots,M} \sum_{k=1,\dots,L} p(l_i^c = l_k | e_i, \theta_{-i}) p(l_i^f = l_k | s_i, a_i, \xi_t) \end{aligned} \quad (4.4)$$

While this equation exhibit minor changes over Equation. 4.2, it avoids problems of overfitting. However, this Equation quickly becomes computational costly and is unlikely to be usable online in practice. For example, after 100 steps, if just 10 task hypotheses were considered, the system would have to compute 1000 classifiers to update the likelihoods of each task. While for the previous equations (Eq.4.2 and Eq. 4.3), if 10 task hypothesis are considered, only 10 classifiers must be computed each step.

Still, this last approach is not taking into account the quality of the classifier itself. The question is of knowing how reliable the predictions of the classifier are.

A common method to evaluate the uncertainty on a classifier's predictions is to use a cross-validation procedure to estimate the confusion matrix associated to the classifier. Such confusion matrix allows to infer the conditional probability of one label given the label predicted from the classifier $p(l^{cc} = l_k | l^c = l_q, \theta)$, for every combination of k and q in $1, \ldots, L$. Where l^{cc} is the corrected, or "temperated", label predicted by the classifier given our estimates on the quality of the classifier's predictions using the cross validation procedure.

For example, a dummy classifier could predict that any given signal e will have a probability 1 of being of class l_1. This classifier is obviously wrong if there more than two labels in the training dataset, and the cross-validation procedure will capture and quantify the classifier bias. If the training dataset were composed of 2 classes with equal number of samples, the confusion matrix will give us the following information: $p(l^{cc} = l_1 | l^c = l_1, \theta) = p(l^{cc} = l_2 | l^c = l_1, \theta) = 0.5$. Meaning that when the classifier predicts label l_1 there is 50 percent of chances that the true label is l_1, and 50 percent that it is actually l_2. In other words, the classifier is useless. On the contrary, a perfect classifier, that never makes classification errors will be represented by the following conditional probabilities: $p(l^{cc} = l_1 | l^c = l_1, \theta) = p(l^{cc} = l_2 | l^c = l_2, \theta) = 1$ and therefore $p(l^{cc} = l_1 | l^c = l_2, \theta) = p(l^{cc} = l_2 | l^c = l_1, \theta) = 0$.

We include the following measure of uncertainty on the classifier's predictions in Equation 4.4:

$$\begin{aligned} \mathcal{L}(\xi_t) &= \prod_{i=1,\ldots,M} p(l_i^{cc} = l_i^f | D_M, \xi_t) \\ &= \prod_{i=1,\ldots,M} \sum_{k=1,\ldots,L} p(l_i^{cc} = l_k | e_i, \theta_{-i}) p(l_i^f = l_k | s_i, a_i, \xi_t) \end{aligned} \tag{4.5}$$

with:

$$p(l_i^{cc} = l_k | e_i, \theta_{-i}) = \sum_{q=1,\ldots,L} p(l_i^{cc} = l_k | l_i^c = l_q, \theta_{-i}) p(l_i^c = l_q | e_i, \theta_{-i}) \tag{4.6}$$

These latter equations capture well the full aspect of the problem. However the computational cost explodes, it would require to train 10000 classifiers after 100 steps, to compute the likelihood of just 10 task hypotheses, and using a 10 fold cross-validation procedure. This is impossible to use in real time and, as for Equation 4.3, we will rely on an iterative process to cope with this problem:

$$\begin{aligned} \mathcal{L}_i(\xi_t) &= p(l_i^{cc} = l_i^f | D_i, \xi_t) \;\; \mathcal{L}_{i-1}(\xi_t) \\ &= \sum_{k=1,\ldots,L} p(l_i^{cc} = l_k | e_i, \theta_{i-1}) p(l_i^f = l_k | s_i, a_i, \xi_t) \;\; \mathcal{L}_{i-1}(\xi_t) \end{aligned} \tag{4.7}$$

with:

$$p(l_i^{cc} = l_k | e_i, \theta_{i-1}) = \sum_{q=1,\ldots,L} p(l_i^{cc} = l_k | l_i^c = l_q, \theta_{i-1}) p(l_i^c = l_q | e_i, \theta_{i-1}) \tag{4.8}$$

where θ_{i-1} is the classifier trained on all the past observation up to time $i-1$ and according to the label generated from task ξ_t. And with $\mathcal{L}_0(\xi_t)$ being the prior at

time 0 (before the experiment start) for the task ξ_t, usually uniform over the task distribution.

Following this latter equation, at each step, for 10 task hypothesis, and using 10 fold cross-validation to estimate $p(l^{cc}|l^c, \theta_{i-1})$ the system would have to compute 100 classifiers to update the likelihoods of each task.

To summarize, we described several measures of quality of classifiers. We incrementally included some uncertainty measurements to avoid making too sharp estimates when the classifiers are known to be of unreliable and to avoid problems of overfitting.

Each term of our pseudo-likelihood is computed from three terms:

- $p(l^f|s, a, \xi)$ is the frame function, it represents the probability distributions of the meanings according to a task, the executed action and the current state, i.e. it represent the interaction frame.
- $p(l^c|e, \theta)$ is the raw prediction of the classifier θ.
- $p(l^{cc}|l^c, \theta)$ encodes which label should be actually recovered by θ. It is the probability that the classifier itself is reliable in its predictions.

In practice our pseudo-likelihood is maximized in two steps. First, the maximum a posteriori estimate θ of the classifier is computed, and the term $p(l^{cc}|l^c, \theta)$ is approximated using the confusion matrix associated to the classifier based on a cross validation procedure on the training data. Then, given the classifier and the confusion matrix, the likelihood of the task is evaluated. Finally, the best task ξ should be the one that maximizes the pseudo-likelihood.

Note that the term $p(l^{cc}|l^c, \theta)$ is a global approximation of the uncertainty of classifier's prediction. It considers that the classifier suffer from the same biases for any given signal, i.e. it does not depend on the signal e to be predicted.

4.4.3 Decision

Using any of the measures described above does not inform us about when our system is confident about which task hypothesis is the correct one. Indeed at every iteration step, the likelihood of one task will be higher that all others. Which criteria should we used to decide when "higher" is enough?

The simplest method is to normalize the likelihood estimates to 1, and considered the resulting value as the probability of each task:

$$p(\xi_t) = \frac{\mathcal{L}(\xi_t)}{\sum_{u=1,\ldots,T} \mathcal{L}(\xi_u)} \tag{4.9}$$

Given this measure, we can define a probability threshold β, and, when it exists a t such that $p(\xi_t) > \beta$, we can consider the task ξ_t is the one taught by the user.

This method suffers from one important drawback, it does not scale well with the number of hypotheses. Indeed, the more tasks, the more the differences in likelihood between the best task and the other tasks should be important to reach the defined threshold. Consider for example two cases: a) for only two tasks whose respective likelihoods are $[0.45, 0.05]$, their normalized likelihood is $[0.9, 0.1]$ b) for four tasks whose respective likelihoods are $[0.45, 0.05, 0.05, 0.05]$, their normalized likelihoods are $[0.75, 0.083, 0.0833, 0.083]$. While the difference of likelihood between the best task and the other tasks is the same in both condition, the normalization decreases the importance of the first task with respect to the others. By scaling this reasoning to one hundred hypotheses, the normalized likelihood method requires immense likelihood differences to reach the same threshold. Therefore, the normalized likelihood method requires to change the threshold for every scenario depending on the number of tasks considered.

Comparing the likelihood by pairs is a more robust estimate. Considering the example described above, the first hypothesis was 9 times more likely than all other hypotheses in all conditions (2 or 4 tasks considered). We therefore define W^{ξ_t} as the minimum of pairwise normalized likelihood between hypothesis ξ_t and each other hypothesis:

$$W^{\xi_t} = \min_{u \in 1,\ldots,T \setminus \{t\}} \frac{\mathcal{L}(\xi_t)}{\mathcal{L}(\xi_t) + \mathcal{L}(\xi_u)} \tag{4.10}$$

When it exists a t such that W^{ξ_t} exceeds a threshold $\beta \in]0.5, 1]$ we consider task ξ_t is the one taught by the user.

Going back to our previous example: a) for only two tasks whose respective likelihoods are $[0.45, 0.05]$, their normalized likelihood is $[0.9, 0.1]$ while their minimum pairwise normalized likelihood is $[0.9, 0.1]$ b) for four tasks whose respective likelihoods are $[0.45, 0.05, 0.05, 0.05]$, their normalized likelihoods are $[0.75, 0.083, 0.0833, 0.083]$ while their minimum pairwise normalized likelihood is $0.9, 0.1, 0.1, 0.1]$. With our latter measure W^{ξ_t}, we can define a threshold that will hold for every scenario independently of the number of hypothesis.

In our various experiments, both measures will be considered.

4.4.4 From task to task

Once a task is identified with confidence, the robot executes that task and prepares to receive new instructions from the user according to a new task. Assuming the user starts teaching a new task using the same kind of signals, we now have much more information about the signal to meaning mapping. Indeed, once we are confident that the user was providing instructions related to a specific task, we can infer the true labels of the all the past signals. Therefore we can propagate these labels to all other task interpretation hypothesis (see Figure 4.17), and, by using the same algorithm, we can start learning the new task faster as all hypothesis now share a common set of signal-label pairs. As described in Figure 4.18, the signal to meaning

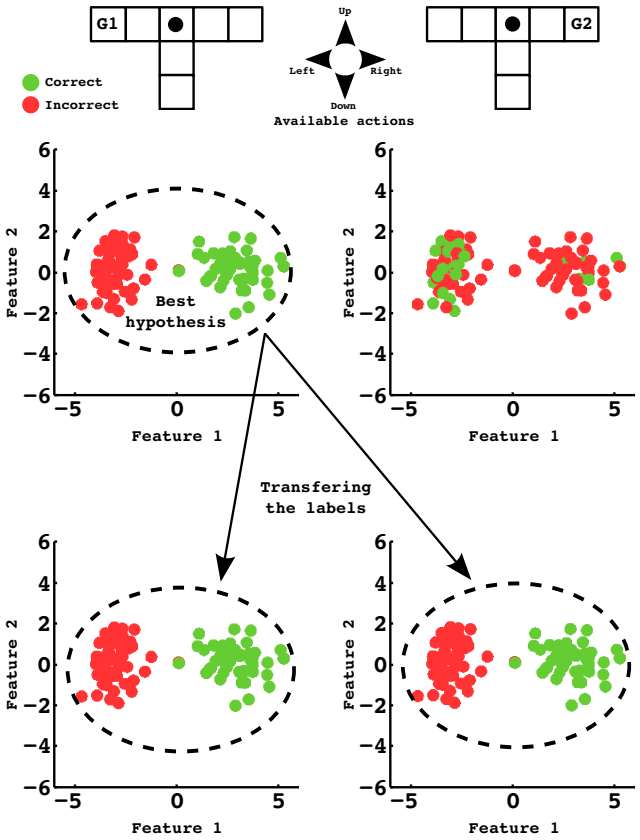


Figure 4.17: Once a task is identified with confidence, we propagate the labels of the best hypothesis to all the other hypotheses.

models for each hypothesis are still updated every step until the new task is identified and labels reassigned.

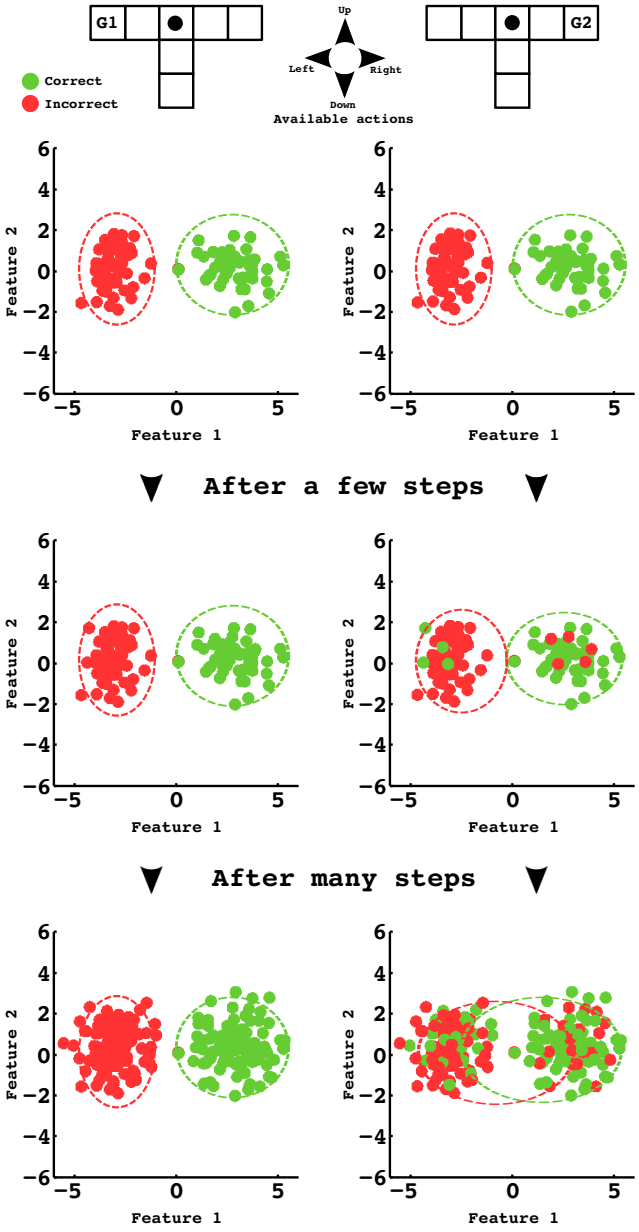


Figure 4.18: When teaching a second task, all hypotheses start with the same signal-label pairs. After a few new interaction steps, some differences in labeling occurs which are easy to detect as they do not conform to the now shared signal model. Therefore allowing to discard quickly the hypothesis G2. We note the interpretation hypothesis process continues to impact the classifiers of each task. The classifier associated to the correct task (here G1) keeps the same quality level, but the one associated to G2 progressively becomes less accurate. This is clearly visible after many steps in the new interaction session.

This phase of reuse of previous information could be assimilated to the results of a calibration procedure. Indeed, after a first run we have access to the true intended labels associated to the human signals. The simplest option is therefore to compute one classifier, common to all tasks, and to use it to classify new signals.

The method described above differs in that we keep assigning hypothetic labels for each task and we keep updating every classifier. This process allows to decrease the quality of the classifier associated to wrong hypotheses. It helps identifying the correct task faster and more robustly. This effect is more important for the first few task identified. But as the number of signal-label pairs shared between hypothesis increases, the number of new observations needed to sensibly modify the classifiers increases. Therefore our method progressively converges to the use of a common classifier shared among all hypothesis. This process will be tested in chapter 6 where we will compare our method with a standard calibration procedure approach using EEG signals.

4.4.5 Using known signals

In some cases, the robot may already understand some of the communicative signals from the human. For example, the user could have access to two colored buttons, one green to mean "correct" and one red to mean "incorrect". But the user may prefer using speech to interact with the robot. To allow for flexible interaction, such speech command should not be preprogrammed as each user may speak a different language, or may prefer using the word "yes" instead of "correct" for example. Considering the mapping between buttons and meanings is known and the mapping between speech utterances and meanings is unknown, we can add a terms to our likelihood equations that includes the information provided by the known signals.

Knowing the meaning of a signal is knowing the parameters θ_{button} corresponding to the mapping between button presses and their meanings. We can therefore define a separate likelihood update for the known signals, but we simply use the same classifier for each task:

$$\mathcal{L}_{button}(\xi_t) \quad = \quad \prod_{i=1,\dots,M} p(l_i^{cc} = l_i^f | s_i, a_i, e_i, \xi_t, \theta_{button}) \tag{4.11}$$

The likelihood from the speech can be computed using the equations described in subsection 4.4.2, which we rename $\mathcal{L}_{speech}(\xi_t)$ for convenience.

Finally we can compute the final likelihood as the product of both estimates:

$$\mathcal{L}(\xi_t) \quad = \quad \mathcal{L}_{button}(\xi_t) \;\; \mathcal{L}_{speech}(\xi_t) \tag{4.12}$$

It is important to understand the difference between our approach and a method learning from signals of known meanings. With our approach, we estimate one classifier per task hypothesis. However, if we have access to the true signal to meaning mapping, we must use the corresponding classifier for all hypothesis. Therefore all the equations remain the same, only replacing a global classifier for known signals by hypothetic ones for unknown signals.

4.4.6 Two operating modes

Our algorithm is divided into a classification algorithm, estimating one classifier for each hypothesis based on past interaction, and a filtering algorithm that uses the predictions and properties of this classifier to update a belief over all tasks hypothesis. The key point is that each hypothesis is considered as if it was the true one. We model the signal to meaning mapping of the user with respect to each task. We then simply test if each classifier can make accurate predictions. As the user is acting according to only one hypothesis, only that hypothesis will be able to predict correctly future interactions. Once a task is identified, we have access to the true intended labels of the user. Which we transfer to all the other hypotheses and start learning a new task using the same equation and by continuing the interpretation hypothesis process. As all hypothesis now share a common set of signal-label pairs, we should be able to learn the new task faster.

We highlight the different processes acting during a full experiment when learning multiple tasks. We will refer to two operating modes: a) mode 1 is learning the first task from unlabeled instructions, and b) mode 2 is learning a task when most of the labels are shared between hypothesis. Our update equation is the same for the two operating modes but different properties are more or less active during mode 1 or mode 2.

Mode 1 is the main contribution of this work. During mode 1 our measure of uncertainty on classifiers' predictions has more impact than the raw predictions of each classifier. Indeed, with very few data available, the classifiers are unable to predict correctly unseen data. Therefore all classifiers are considered as unreliable, and our update equation makes only small updates each step. It is only once one classifier stands apart as being more reliable than the others that differences between likelihoods will emerges.

Mode 2 is almost the contrary. Once many tasks have been identified, all hypothesis share a similar classifier because of the transfer of labels. Therefore they all have similar confusion matrix and make similar predictions. Mode 2 is therefore similar to learning from a known source of information, where all tasks share the same classifier. And it is only by comparing the label prediction of new signals to their expected label for each task that we differentiate hypotheses. This process is logically faster than mode 1 because strong updates are made for each received signal.

Between mode 1 and mode 2 is a period of transition where the effects of both modes are active. When only few signal-label pairs are shared between hypotheses, each classifier evolves quickly as new observations come in.

To sum up, the same processes are active in both modes and are captured by the same equation (see Equation 4.5). In mode 1, it is the classifier intrinsic quality that has the most impact. In mode 2, it is the classification of each individual signal that has the most impact.

This dynamics may be hard to visualize yet. It will be reminded and illustrated in chapter 6, where we display the evolution of classification rate of all classifier

during an experiment where an agent learn several tasks in a raw. Mode 1 will be observed on Figure 6.4 (top), where, during the learning of the first task, all classifier have accuracy close to random (50%). It is only at step 83 that the correct hypothesis stands apart by being consistently more reliable than the other. Mode 2 will be observed on Figure 6.4 (top), where, after the step 200, the difference between classifier qualities is very small. Indeed, 5 tasks have already been identified and all hypotheses share most of their signal-label pairs, therefore all classifiers make similar predictions. The transition between mode 1 and mode 2 will be observed on Figure 6.4 (top) between step 83 and 200.

In next sections, we present results from our algorithm considering a pick and place robotic scenario where a human wants a robot to build a specific structure with cubes and provides instructions to the robot using vocal commands, whose meaning are unknown to the robot at start. We present results both in simulation and with a real robotic system where we test different aspects: (a) how our algorithm scale to a robotic scenario considering a feedback frame, (b) how it behaves for the case of guidance words, (c) the combining of unlabeled signals with signals of known meanings (buttons), (d) the reuse of a learned signal-to-meaning mapping for the learning of a new task.

4.5 Method

We construct a small size pick-and-place task with a real robot. This robot is going to be programmed using a natural speech interface whose words have an unknown meaning and are **not** transformed into symbols via a voice recognizer. The robot has a prior knowledge about the distribution of possible tasks.

The interaction between the robot and the human is a turn taking, the robot performs an action and waits for a feedback, or guidance, instruction to continue. This allows to synchronize a speech wave with its corresponding pair of state and action. The experimental protocol is summarized in figure 4.19.

4.5.1 Robotic System

We consider a six d.o.f. robotic arm and gripper that is able to grasp, transport and release cubes in four positions. We used a total of three cubes that can form towers of at most two cubes. The robot has 4 actions available: *rotate left*, *rotate right*, *grasp cube* and *release cube*. The state space is discrete and defined as the location of each object, including being on top of another object or in the robot's gripper. For a set of 3 objects we have 624 different states. Figure 4.20 shows the robot grasping the orange cube.

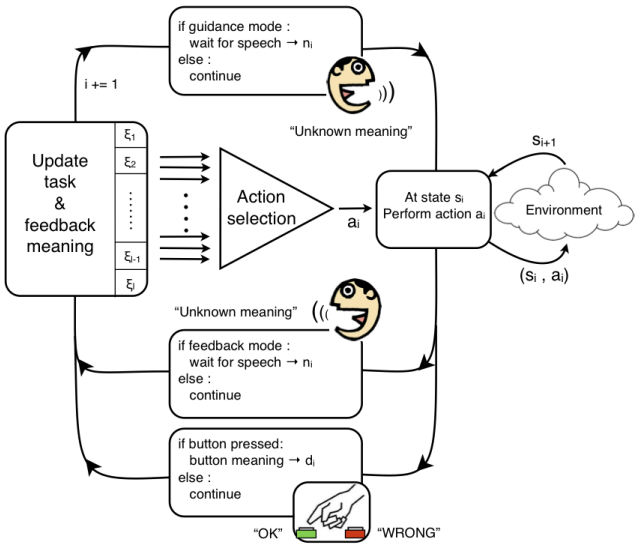


Figure 4.19: Experimental protocol showing the interaction between the teacher and the learning agent. The agent has to learn a task and the meaning of the instructions signals provided by the user, here recorded speech. The teacher can use guidance or feedback signals, and may also have access to buttons of known meanings for the robot.

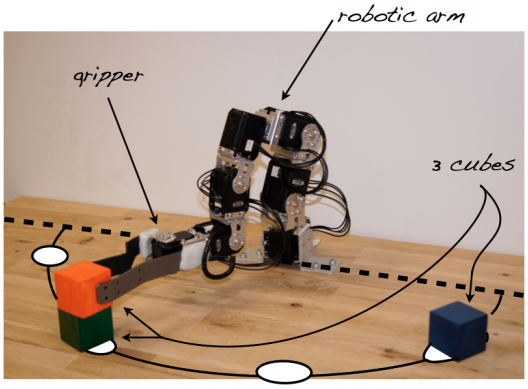


Figure 4.20: The six d.o.f robotic arm and gripper learning to performing a pick-and-place task with three cubes.

4.5.2 Task Representation

We assume that for a particular task ξ we are able to compute a policy π representing the optimal actions to perform in every state. One possibility is to use *Markov Decision Processes* (MDP) to represent the problem [Sutton 1998]. From a given task ξ represented as a reward function we can compute the corresponding policy using, for instance, Value Iteration [Sutton 1998]. In any case, our algorithm does not make any assumption about how tasks are represented.

For this particular representation, we assume that the reward function representing the task taught by the human teacher is sparse. In other words, the task is to reach one, yet unknown, of the 624 states of the MDP. Therefore we can generate possible tasks by sampling sparse reward functions consisting of a unitary reward in one state and no reward in all the other.

4.5.3 Feedback and Guidance Model

From a given task ξ, we can compute the corresponding policy π_ξ. This policy allows to interpret the teaching signals with respect to the interaction protocol defined. In this experiment, we will consider the user is providing either feedback or guidance on the agent's actions.

For the feedback case, we define $p(l^f|s,a,\xi)$ as:

$$p(l^f = correct|s,a,\xi) = \begin{cases} 1-\alpha & if\ a = \arg\max_a \pi_\xi(s,a) \\ \alpha & \text{otherwise} \end{cases} \tag{4.13}$$

with α modeling the expected error rate of the user and $p(l^f = wrong|s,a,\xi) = 1 - p(l^f = correct|s,a,\xi)$.

For the guidance case, the user instructions represent the next action the robot should perform, therefore it only depends on the current state of the robot and the task considered. In our scenario, there are 4 different possible actions ($nA = 4$) in each state. We define $p(l^f|s,\xi)$ for each action as:

$$p(l^f = a|s,\xi) = \begin{cases} 1-\alpha & if\ a = \arg\max_a \pi_\xi(s,a) \\ \frac{\alpha}{nA-1} & \text{otherwise} \end{cases} \tag{4.14}$$

with α modeling the expected error rate and assuming only one action is optimal. As the agent can perform 4 different actions, we used the constant $\frac{\alpha}{3}$ for non-optimal actions in order to conserve the property that $\sum_a p(l^f = a|s,\xi) = 1$. For those cases where there is more than one optimal action, the probability is uniformly splitted among them. If all actions are optimal, they all share the same probability of $\frac{1}{nA}$.

It is important to remember that these frames, while capturing a realistic interaction protocol, are arbitrary and we explicitly ask the users to conform to them. Here we assume that the teacher is aware of the optimal policies to fulfill the task, and additionally shares the same representation of the problem than the robot. Especially, for the scenario considered, the user should be aware that the robot cannot move from position 1 to position 4 in one action. The robot should rather pass thought all intermediate positions ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$). However as we know the user will sometime make teaching mistakes, we added a noise term α that account for unpredictable teaching mistakes. For all following experiments α was set to 0.1.

4.5.4 Speech Processing

We consider speech as the modality for interacting with the robot. After each action we record the teaching word pronounced by the user. This data is mapped into a 20 dimensional feature space using the methodology described next.

A classical method for representing sounds is the *Mel-Frequency Cepstral Coefficients* (MFCC) [Zheng 2001]. It represents a sound as a time sequence of MFCC vectors of dimension 12. Comparing sounds is done via *Dynamic Time Warping* (DTW) between two sequences of feature vectors [Sakoe 1978]. This distance is a measure of similarity that takes into account possible insertions and deletions in the feature sequence and is adapted for comparing sounds of different lengths. Each recorded vocal signal is represented as its DTW distance to a base of 20 pre-defined spoken words, which are **not** part of words used by the teacher.

By empirical tests on recorded speech samples, we estimate that a number of 20 bases words were sufficient and yet a relatively high number of dimensions to deal with a variety of people and speech. This base of 20 words has been randomly selected and is composed of the words: *Error, Acquisition, Difficulties, Semantic, Track, Computer, Explored, Distribution, Century, Reinforcement, Almost, Language, Alone, Kinds, Humans, Axons, Primitives, Vision, Nature, Building.*

4.5.5 Classifiers

Any machine learning algorithm working for classification problems can be used in our system. This classifier should be able to generalize from the data and should have appropriate underlying assumptions on the structure of those data. In other words, if the labels were known, the classifier should be able to find a good mapping between the signals and their meanings. The only required characteristic is the ability to output a probability on the class prediction, i.e. $p(l|e, \theta)$.

In this study we decided to compare three classifiers:

- Gaussian Bayesian Classifier (also called quadratic discriminant analysis (QDA)): Computing the weighted mean μ and covariance matrix Σ, the usual equations for Gaussian mixture hold.

- Support Vector Machine (SVM): Using a RBF kernel with $\sigma = 1000$ and $C = 0.1$. The parameter values have been estimated via a swap of parameters and

by estimating performances via a cross validation procedure on the dataset. For SVM probabilistic prediction refer to [Platt 1999].

- Linear Logistic Regression: The predictive output value ($[0, 1]$) is used as a probability measure.

Our classifiers are tested on our labeled speech dataset in order to verify their adequacy to model the signal to meaning mapping. All three classifiers obtain accuracy close to 100 percent. This is a rather optimal scenario, we will see in next chapter 5.6 how our algorithm behaves with data of poorer quality.

4.5.6 Action selection methods

The selection of the robot's actions at runtime can be done in different ways. We will compare two different methods: random and ε-greedy. When following random action selection the robot does not use its current knowledge of the task and randomly selects actions. Whereas with ε-greedy method the robot performs actions according to its current belief on the tasks, i.e. it follows the policy corresponding to the most likely task hypothesis. The corresponding optimal action is chosen with $1-\varepsilon$ probability, otherwise, a random action is selected. In our experiment, we only consider results with $\varepsilon = 0.1$.

It is only in the next chapter that we will investigate how the robot can actively selects its future actions in order to improve its performances.

Before presenting the results of our experiments, we illustrate in next section the pick and place scenario as well as the results of the labeling process for the feedback and guidance case.

4.6 Illustration of the pick and place scenario

We illustrate in Figure 4.21 the pick and place world (where we used balls instead of cubes). There are three objects that can be moved in four different positions and stacked on two levels maximum. The robot's gripper can only grasp the object on top of the stack. An object is always released on top of a stack, except if the stack is full, in which case the release action produces no effect.

In order to complete a task, i.e. to reach a specific configuration of cubes, the robot must perform an ordered sequence of actions. For illustration purpose, we only consider 3 out of the 624 possible hypotheses. Figure 4.22 shows a sequence of actions starting in our hypothesis 1 configuration and going to our hypothesis 3 configuration using the shortest possible number of actions. Hypothesis 2 is a state on this path. While hypothesis 1 and hypothesis 3 seems "close" in terms of position of the cubes, they are actually "far" one from the other in terms of the action sequence.

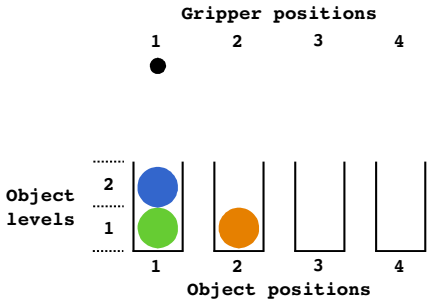


Figure 4.21: A schematic view of the pick and place problem. There is three objects that can be moved in four different positions and stacked on two levels maximum. The robot's gripper can only grasp the object on top of the stack. An object is always released on top of a stack, except if the stack is full, in which case the release action produces no effect.

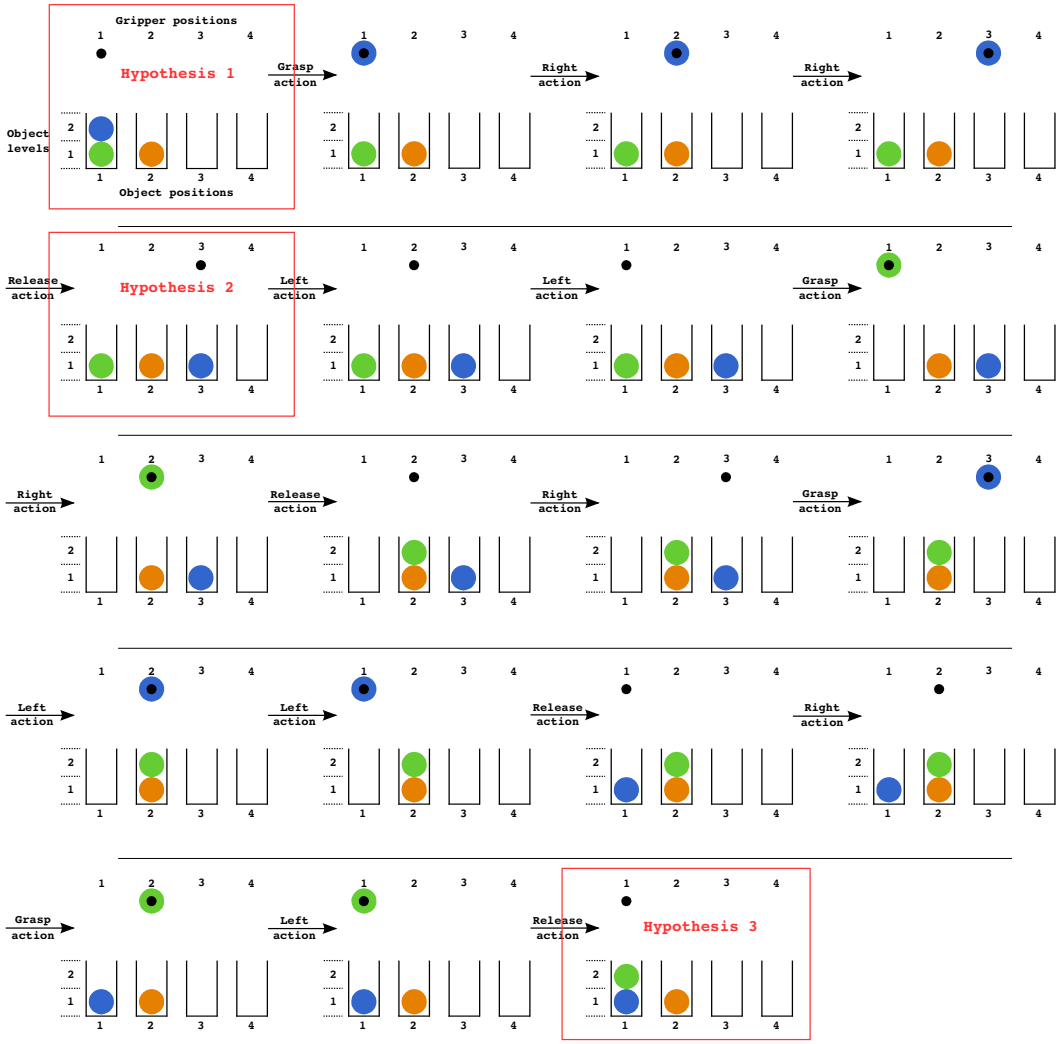


Figure 4.22: A pick and place sequence showing three hypotheses and the sequence of actions from hypothesis 1 to hypothesis 3 through hypothesis 2. While hypothesis 1 and hypothesis 3 seems "close" in terms of position of the cubes, they are actually "far" one from the other in terms of the action sequence.

If the user is delivering feedback signals, the labeling process is presented in Figure 4.23 for a robot acting randomly in the environment. Note that hypothesis 1 and 2 are the most difficult to discriminate by acting randomly as they share most of their optimal policies.

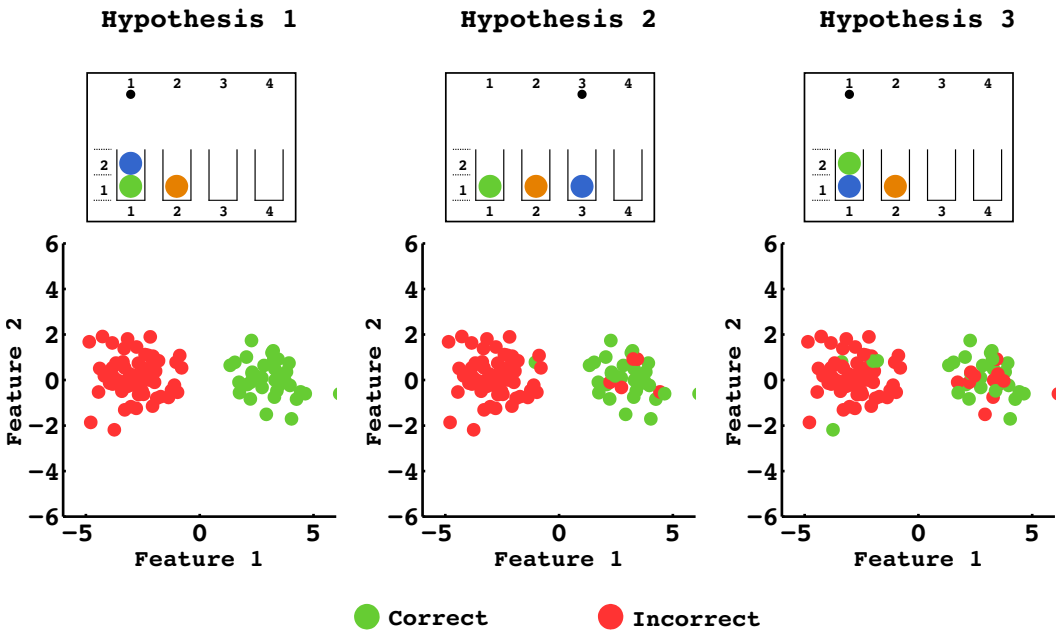


Figure 4.23: Results of the labeling process for our three hypotheses and considering the feedback frame. The robot explores randomly the state space. The teacher provides feedback with respect to hypothesis 1. Only a few state-action pairs allowed to differentiate between hypothesis 1 and 2.

For the guidance case, the teacher uses the signals presented in Figure 4.24 and the labeling process is presented in Figure 4.25 for a robot acting randomly in the environment. Note that for some states there may be two optimal actions. For example, in Figure 4.22, for inverting two stacked cubes there is two different optimal policies, either the one presented in Figure 4.22, or putting the blue ball in position 2 and the green in position 3 during the exchange of position. These equally optimal options make the learning process more difficult, we can still visually find out that for hypothesis 1 all points in each cluster share one color, which is not the case for the other two hypotheses.

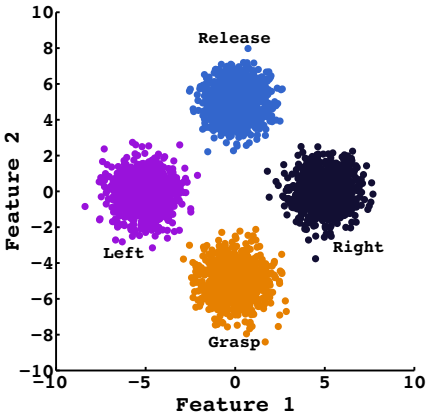


Figure 4.24: The guidance signals used for our visual example.

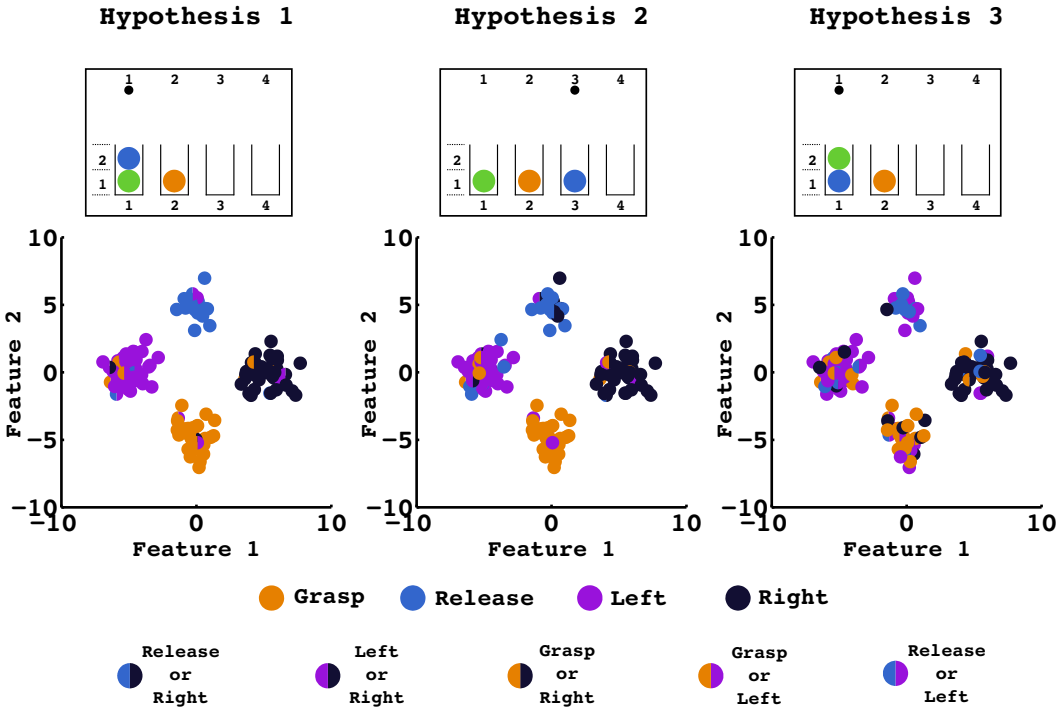


Figure 4.25: Results of the labeling process for our three hypotheses considering the guidance frame. The robot explores randomly the state space. The teacher provides guidance with respect to hypothesis 1. The labeled signals that contain two colors represent situations where the user could have given two different guidance signals, i.e. where two actions were optimal. It is only for hypothesis 1 that all signals in each cluster share one color. The case of guidance with multiple optimal actions in some states makes the learning process more ambiguous and may require some additional time compared to the feedback case.

We have provided an example of the pick and place world with two dimensional signals and considering only three hypotheses. In next section, we consider real spoken words mapped to a 20 dimensional space and the full space of hypothesis consisting of 624 possible object configurations.

4.7 Results

The experiments presented in this section follow the protocol described in figure 4.19, where each turn the agent performs one action and waits for the teaching signals from the teacher. We first present a set of simulated experiments using the same MDP as for the real world experiment. We start by assuming that the teacher provides feedback instructions without any mistakes. We compare first the different classifiers, and then the performances of ε-greedy versus random action selection methods both for the feedback and guidance cases. Later, we present an empirical analysis of robustness to teaching mistakes. The last simulated experiment considers a teacher having also access to buttons of known meaning. Finally, we show a result using the real robot and a human user, where we study how signals knowledge learned in a first run can be used in a second one to learn more efficiently.

In order to be able to compute statistically significant results for the learning algorithm, we created a database of speech signals that can be used in simulated experiments. All results report averages of 20 executions of the algorithm with different start and goal states.

As there are 624 hypotheses, we must update 624 likelihoods at each step. Depending on the likelihood equation considered this may not be feasible in real time. As our aim is to run our system in real time, and as we know that the speech signals in our dataset are well separated in their feature space, we use the simplest version of our likelihood estimation methods described in Equation 4.2. To estimate the probability of each task we normalize the likelihood estimates $\mathcal{L}_(\xi_1), \ldots, \mathcal{L}_(\xi_T)$ to 1.

4.7.1 Learning feedback signals

In this experiment, the teacher is providing spoken signals whose meanings are either "correct" or "incorrect". The robot should simultaneously learn the task and the mapping between the spoken words and the binary meanings. The action selection of the robot is done using the ε-greedy method. The user uses only one word per meaning.

The results comparing the different classification methods are shown in Figure 4.26. It shows the evolution of the probability associated to the task the teacher has in mind. We can track this information because we know, as experimenters, the true task taught by the teacher. Note that after 200 iterations all three classification methods identified the correct task as the most likely, i.e. the normalized likelihood values of the correct task are greater than 0.5, meaning that the sum of all the others is inferior to 0.5. Logistic regression provides the worse results in terms of convergence rate and variance.

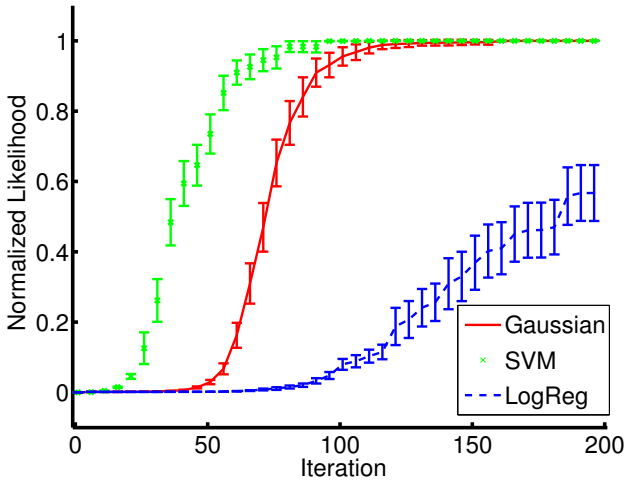


Figure 4.26: Taught hypothesis normalized likelihood evolution (mean + standard error) thought iterations using different kinds of classifiers. The teacher is providing feedback signals using one word per meaning and the agent is performing actions according to the ε-greedy strategy.

The user is not restricted to the use of only one word per meaning. Table 4.1 compares the taught task normalized likelihood value after 100 iterations for feedback signals composed of one, three and six spoken words per meaning. SVMs have better performances when using one word per meaning but the Gaussian classifier has overall better results with less variance, see Table 4.1.

	One word	**Three words**	**Six words**
Gaussian	1.0 (0.1)	1.0 (0.1)	0.7 (0.1)
SVM	1.0 (0.0)	0.5 (0.4)	0.3 (0.4)
LogReg	0.1 (0.1)	0.2 (0.3)	0.2 (0.3)

Table 4.1: Taught hypothesis normalized likelihood values after 100 iterations (mean and standard deviation). Comparison for different classifiers and number of words per meaning. The Gaussian classifier has overall better performances.

Interestingly the Gaussian classifier learns better after 100 iterations than the other classifiers with many words per meaning. This counter intuitive result can be explains by the high dimensionality of the space where even one Gaussian can differentiate several groups of clusters. Linear logistic regression has lower performance presumably due to the linear decision boundary. For the SVM classifier, which is kernalized, as only 100 data points are distributed between each cluster, the more the number of clusters increases the less data points belong to each cluster. The fitting process of the SVM is therefore more likely to consider some data as noise, omitting some clusters. For the following experiments, we will only consider the Gaussian classifier, first because it has overall better performance, but also because

it is the faster to train and thus is the only one usable for online experiments.

4.7.2 Learning guidance signals

In Figure 4.27, we compare the performance between using feedback or guidance signals. From feedback to guidance the number of meanings is increased from two (correct/incorrect) to four (left/right/grasp/release). As depicted in Figure 4.27, the robot is able to identify the task based on unlabeled guidance signals. However it requires more iterations to reach the same level of confidence. This may look counter intuitive because guidance signals are more informative. However the robot now needs to classify instructions in four different meanings, i.e. to identify four clusters of signals, which requires more samples.

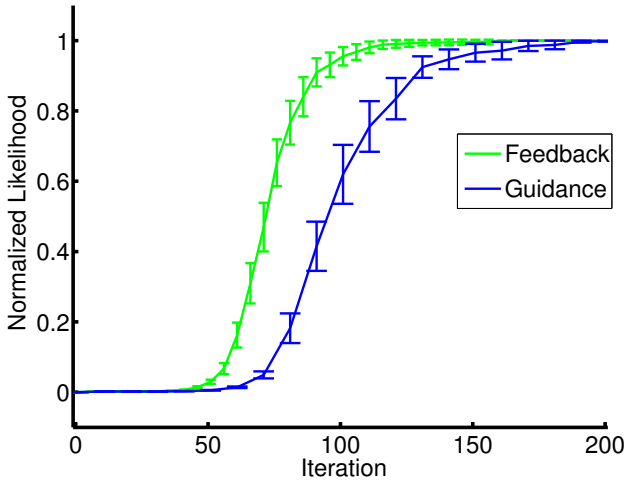


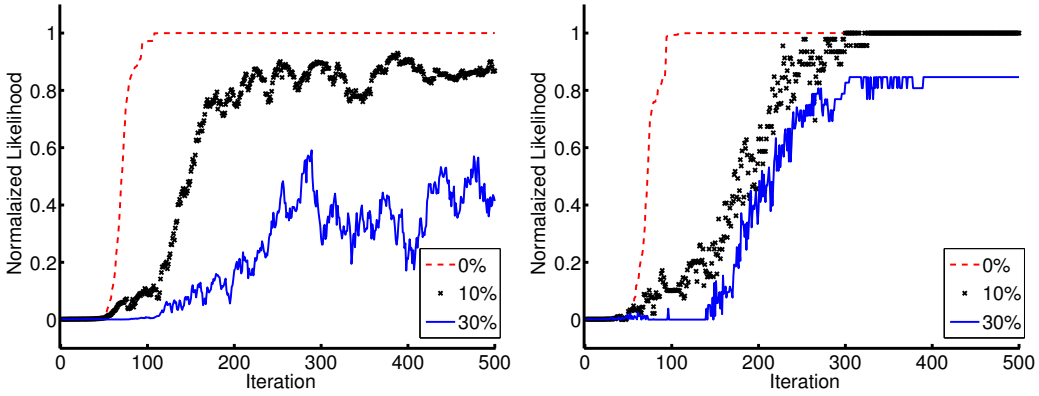
Figure 4.27: Taught hypothesis normalized likelihood evolution (mean + standard error) thought iterations using Gaussian classifier. Comparison of feedback (green) and guidance (blue) instructions using one word per meaning. The robot is able to learn the task based on both feedback and guidance signals but needs more iterations for the guidance case.

4.7.3 Robustness to teaching mistakes

Until now, we made the assumption that the teacher is providing feedback or guidance signals without any mistake. But in real world scenario, people can fail in providing optimal feedback. This is why we initially included the *alpha* constant in our frame equations (see section 4.5.3). An empirical analysis of robustness is shown in figure 4.28 using feedback signals, Gaussian classifier, and one word per meaning. We compares two ways of training the Gaussian classifiers: (1) estimating the maximum likelihood (ML) of the Gaussian for each class, namely the mean and covariance, and (2) using the expectation maximization (EM) algorithm

[Dempster 1977] to iteratively update the mean and covariance of each class in order to find the underlying structure of the data.

We show that the EM approach is improving robustness to teaching mistakes. Referring to our previous discussion in section 4.2.3, note that we initialized the EM algorithm with the ML estimates for each Gaussian, and we kept track which Gaussian belongs to which meaning. In addition, the representation used for the spoken words is of high quality and separates well the signals in the feature space. Therefore it is unlikely for the EM algorithm to fail at finding the two clusters given the data properties.



(a) ML (b) EM

Figure 4.28: Taught hypothesis normalized likelihood evolution thought iterations using Gaussian classifier. Comparison of the ML estimates (left) versus EM estimates (right). The teacher is providing feedback using one word per meaning with different percentage of mistakes. Actions are selected following the ε-greedy method. Standard error has been omitted for readability reason.

4.7.4 Including prior information

Learning purely from unknown teaching signals is challenging for the researcher but could be restrictive for the teacher. Therefore additional sources of known feedback are consider, such as a green and a red button, where the green button has a predefined association with a "correct" meaning, as red button with a "incorrect" meaning.

In this study, the teacher still provides unlabeled spoken words but can also use the red and green button as described in figure 4.19. However, and in order to avoid the possibility of direct button to signal association, the user can never use both modalities at the same time and uses them alternatively with equal probability. Therefore, in average, after 250 iterations the robot has received 125 labeled button presses and 125 unlabeled speech signals. In most systems, the speech signals would be ignored but our method enables learning from the unlabeled signals. We compare three learning methods: (1) the robot is learning only via the labeled button

presses, (2) it uses only the unlabeled speech signals, and (3) it uses both labeled and unlabeled signals. Figure 4.29 shows results from this setting.

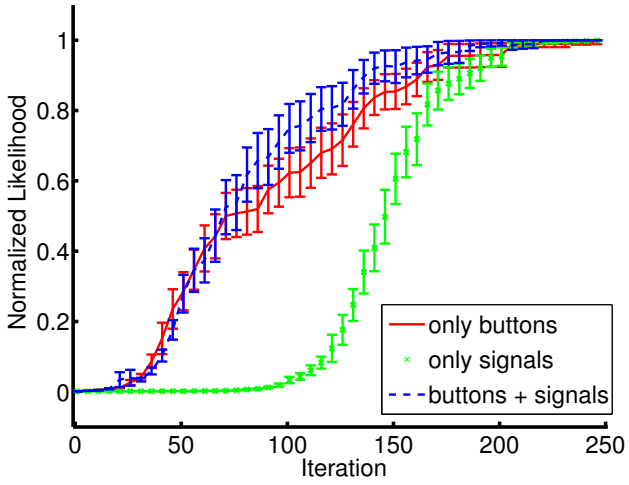


Figure 4.29: Taught hypothesis normalized likelihood evolution (mean + standard error) thought iterations using Gaussian classifier. Comparison of using known button presses, unknown spoken signals, and both.

As expected, learning from labeled feedback is faster than with unlabeled signals. However taking advantage of different sources of information, even a priori unknown, can led to slightly better performances than using only known information. Importantly, the signals to meaning mapping of speech signals learned during a first task, could later be reused in further interactions.

4.7.4.1 Reuse using a real robot

Statistical simulations have shown that our algorithm allows an agent to learn a task from unlabeled feedback in a limited amount of interactions. To bridge the gap of simulation we tested our algorithm in real interaction conditions with our robotic arm. In this experiment, the teacher is facing the robot and chooses a specific goal to reach (i.e. a specific arrangement of cubes he wants the robot to build). He then decides one word to use as positive feedback and one as negative feedback, and starts teaching the robot. For this experiment the word *"yes"* and *"no"* were respectively used for the meaning "correct" and "incorrect".

Once the robot has identified the first task, we keep the corresponding classifier and start a new experiment where the human teacher is going to use the same feedback signals to teach a new task. However the second time, the spoken words are first classified as "correct" or "incorrect" meaning according to the previously learned classifier. We study here two things, first does our system bridges the reality gap and can we reuse information about the signal to meaning mapping learned from a previous interaction session?

Figure 4.30 shows the result from this setting. In the first run it took about

100 iterations for the robot to identify the task. Whereas in the second run, when reusing knowledge from the first one, the robot is able to learn a new task faster, in about 30 iterations. The second run being faster that the first one indicates that our algorithm identified correctly the mapping between the user's speech signals and their corresponding meanings.

The author of this thesis was the user for this study and was therefore aware of the task representation used by the robot, i.e. a MDP with four discrete actions. As explained in subsection 2.1.2, an important challenge is to deal with non-expert humans whose teaching styles can vary considerably. While this challenge is not part of the current study, we observed that non-informed users teaching our robot had various understanding of the robot behaviors. It most often led to unsuccessful interactions due to an important amount of teaching mistakes from non-informed users.

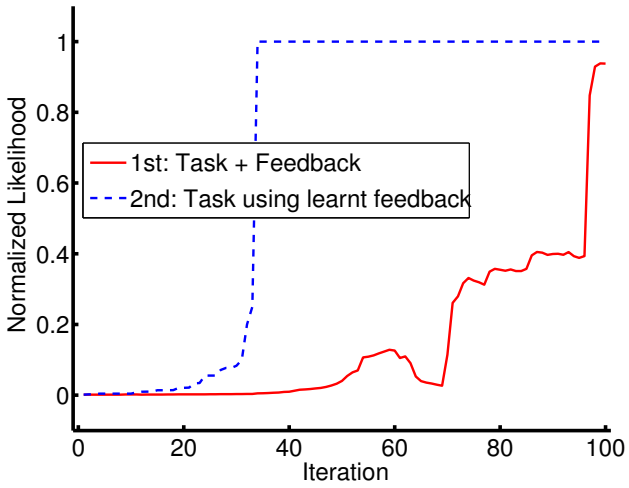
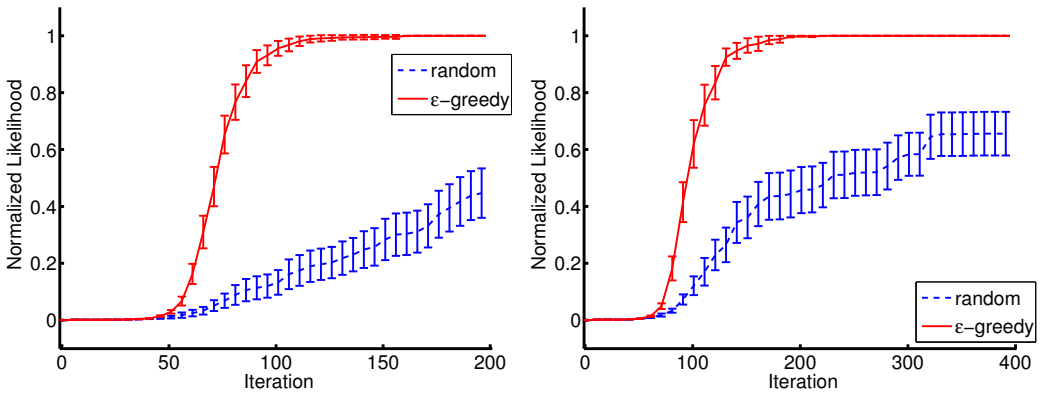


Figure 4.30: Taught hypothesis normalized likelihood evolution thought iterations using Gaussian classifier. A real teacher delivers spoken feedback signals using one word per meaning. The robot uses the ε-greedy action selection method. A first run of 100 iterations is performed where the robot learns a task from unknown feedback. Then, by freezing the classifier corresponding to the most likely task, the user teaches the robot a new task.

4.7.5 Action selection methods

Finally, we compare the impact of using different action selection methods, we consider the ε-greedy and the random action selection methods.

Figure 4.31 left and right compares respectively the action selection method for the case of feedback and guidance interaction frames. The ε-greedy method results in a faster learning with less variance. The ε-greedy method leads the robot in the direction of the most probable goal. In this way, the robot will receive more diverse feedback and will visit more relevant states than what a random exploration does.



(a) Feedback (b) Guidance

Figure 4.31: Taught hypothesis normalized likelihood evolution (mean + standard error) thought iterations using Gaussian classifier. The teacher is providing feedback (left) or guidance (right) signals using one word per meaning. The ε-greedy action selection method allows a faster learning than the random method. Note that the x-axis, showing the number of iterations, does not considered the same range for the feedback (left) and guidance (right) cases.

4.8 Discussion

We showed that learning simultaneously a task and the meaning of an a priori unknown human instruction is possible and that the action selection method impacts the learning performances. Can we do better than random or *epsilon*-greedy action selection methods?

Using an active learning approach [Settles 2010], it might be more efficient to choose the actions that are expected to reduce the uncertainty as fast as possible. In next chapter, we investigate and detail the specific properties of the uncertainty in our problem, where not only the task is unknown but also the signal to meaning mapping. We will provide measures of uncertainty that can be used as exploration bonuses for exploration strategies. We will finally present results from artificial dataset of different qualities in a two-dimensional grid world scenario.

Chapter 5

Planning upon Uncertainty

Contents

In the previous chapter, we presented our algorithm allowing to solve a task from unlabeled human instruction signals. We have seen that the performance of our system is affected by the action selection method used by our robot. In this section, we investigate how the agent should plan its action to improve its learning efficiency. To do so, our agent will look for actions that disambiguate between hypothesis, i.e. which reduce the uncertainty about which hypothesis is the correct one.

The work presented in this chapter has been published in [Grizou 2014a] in collaboration with Iñaki Iturrate and Luis Montesano. Code is available online under the github account `https://github.com/jgrizou/` in the following repositories: lfui, experiments_thesis, and datasets.

We start by explaining what are the methods and measures of uncertainty used by a system that has access to the meanings of the teaching signals. We then provide an intuitive explanation of what are the additional sources of uncertainty inherent to our problem. We will see that this problem is linked to the symmetries properties described in chapter 4.3.3. We then propose two ways of estimating the uncertainty, one on the signal space and one projected on the meaning space. We finally present simulated experiments showing that our measure of uncertainty allows the robot to plan its actions in order to disambiguate faster between hypotheses. These results considered datasets of different qualities and dimensionality, we will see that the performance of the system is affected by the quality of the data more than their dimensionality.

On this basis we will transition to chapter 6 which presents an application to brain computer interaction scenarios, where human users teach an agent to perform a reaching task by assessing the agent's actions using their brain, and without having to calibrate the brain decoder before hand.

5.1 Uncertainty for known signal to meaning mapping

If the mapping between instruction signals and their meanings is provided to the machine, the learning process is rather trivial. The robot should only compare, for each task, whether the meaning received from the human matches with the meaning predicted by the frame. If the meanings match, the probability of the task is increased, if they do not match the probability is decreased.

To accelerate its learning progress, the robot must therefore seek for state-action pairs that maximally disambiguate between hypotheses. For example, if for one given state-action pair, half of the hypotheses expect a signal of meaning "correct" while the other half expect one signal of meaning "incorrect", there is high uncertainty on that action. By performing this action in that state, once the user provides its feedback, the system can rule out half of the hypotheses.

This process must be weighted by the current probability associated to each task hypothesis. Indeed, once half of the hypotheses are discarded, the robot should focus on differentiating the remaining hypotheses. To do so, the robot must seek for a state-action pair where only the remaining hypotheses disagree about the expected teacher feedback.

In the real world, the robot cannot query any state-action pair (it cannot teleport), and rather must navigate through the environment to reach a specific state-action pair. And on its way there, it continuously receives new feedback signals from the user, which may change its belief on the hypotheses probabilities.

A solution is to consider exploration-bonuses, where, for each state-action pair, we associate a reward proportional to the uncertainty of this state-action pair. The agent can then plan its next actions considering the full map of uncertainty. There are several efficient model-based reinforcement learning exploration methods that add an exploration bonus for states that might provide more learning gains. Sev-

eral theoretical results show that these approaches allow to learn tasks efficiently [Brafman 2003, Kolter 2009].

Measuring uncertainty on the task is the basic principle of active learning for inverse reinforcement learning problems [Lopes 2009b]. The idea is to take a query-by-committee approach, where each member of the committee, i.e. each task hypothesis ξ_k, votes according to its weight in the committee, to its respective probability $p(\xi_k)$.

We can define a vector that accumulates the weighted optimal actions of each hypothesis:

$$c(s,a) = \sum_{t=1,\ldots,T} p(\xi_t)\delta(\pi_{\xi_t}(s,a) > 0)$$

where δ is a Dirac function that is 1 if the argument is true and zero otherwise. For each state, the vector entropy of the $c(s,a)$ measures the disagreement between hypotheses.

$$U(s,a) = \mathcal{H}(c(s,a))$$

We can define a reward function that, for each state-action pair, returns an uncertainty value. By computing the policy that maximizes the cumulative reward, i.e. the uncertainty, the agent will visit uncertain states that disambiguate between hypotheses. After several steps, the task probabilities and the uncertainty map are updated. The process is repeated again until the task is identified.

This method works well if the machine has access to the true intended meanings of the user. We will now investigate what makes the uncertainty in our problem different.

5.2 Where is the uncertainty?

In order to exemplify the specificity of the uncertainty for our problem, we rely again on our T world scenario and compare the effects of different action selection strategies. We remind that the teacher wants the robot to reach the left edge of the T (G1).

If the agent knew how to interpret the teaching signals, i.e. which signal corresponds to "correct" or "incorrect" feedback, the optimal action to discriminate G1 and G2 is to move from right to left in the top part of the T. However, as the classifier is not given, we build a different model for each hypothesis (see Figure 5.1). As a result, we end-up with symmetric interpretation of the signals, which are both as valid and do not allow to differentiate between hypothesis.

Considering that the agent does not know the signal to meaning mapping, a sensitive option is to select actions unequivocally identifying the signal model. In the T world, taking only up and down actions in the trunk of the T leads to identical interpretation for each hypothesis (see Figure 5.2). However this action selection method alone does not allow disambiguating between hypotheses as both model are the same, therefore as valid. Moreover, in most settings, such as the grid world we consider later, there is no state-action pair leading to unequivocal interpretation of the signals.

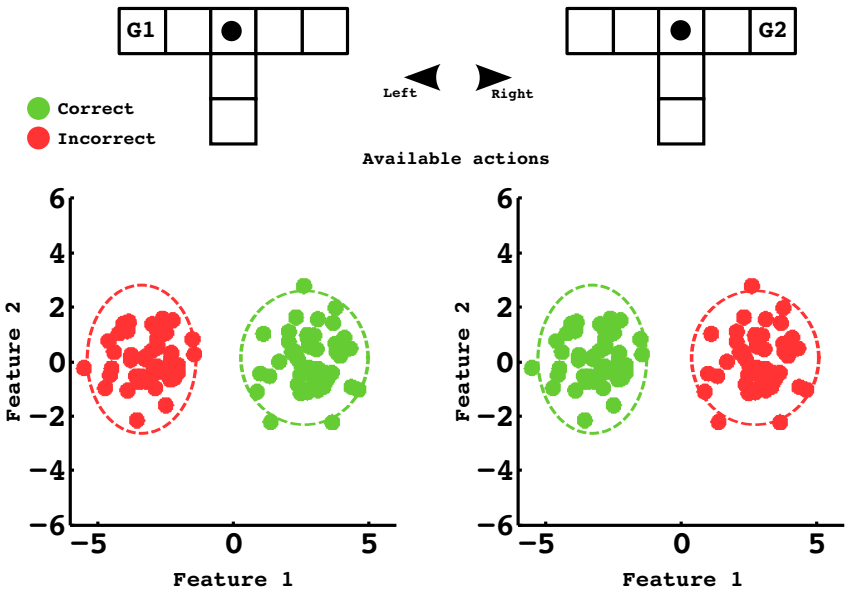


Figure 5.1: Result of the labeling process if the agent only performs right and left actions in the top of the T world. This is the symmetry problem encountered in previous chapter 4.3.3. The resulting signal-label pairs for G1 and G2, while symmetric, are both as likely to explain the data.

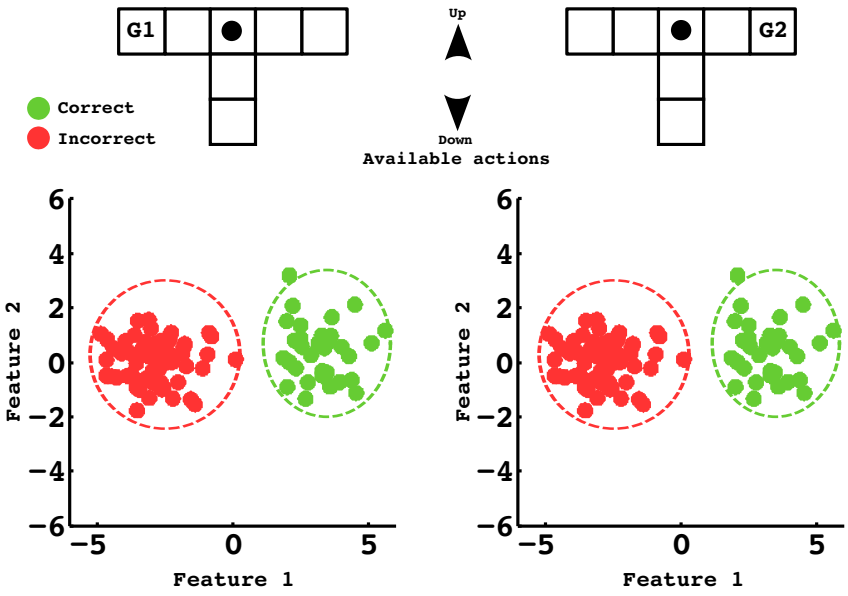


Figure 5.2: Result of the labeling process if the agent only performs up and down actions in the trunk of the T. The interpretation of the signals is the same for both hypotheses, therefore as likely to explain the data.

Those two examples exemplify the specificity of the uncertainty in our problem. The agent cannot just try to differentiate hypothesis by finding state-action pairs where expected feedback differs (left-right actions in Figure 5.1) but should also collect data to build a good model (up-down actions in Figure 5.2). What is the optimal next action to take in the previous condition?

- In the situation of Figure 5.1, the agent ends-up with a symmetric interpretation for G1 and G2 and it should therefore perform an action breaking this symmetry. It must collect one signal whose label is identical for both hypotheses. Performing a "down" action in the T trunk, both hypotheses will associate an "incorrect" label to the received signal, which will break the symmetry.

- In the situation of Figure 5.2, the agent ends-up with an identical interpretation for G1 and G2 and it should therefore collects one signal whose label is different for each hypothesis. By performing a "left" action in the top of the T, hypothesis G1 will associate the label "correct", while hypothesis G2 will associate the label "incorrect", which will break the similarity between models.

Can we find an uncertainty measure that account for both cases? The measure defined in the previous section would not work because it was independent of the signal-to-meaning mapping. Indeed, this mapping was the same for every hypothesis, but in this work, each hypothesis has a different signal-to-meaning mapping. In other words, there is an additional layer of uncertainty on the signal-to-meaning mapping.

We must therefore measure uncertainty taking into account the uncertainty related to both the different tasks and different classifiers. This process will be exemplified in the next section using our T world example. We will present two ways of measuring the uncertainty. The first method measures the uncertainty on the expected signals between each hypothesis. The second method measures uncertainty on the meaning space by making hypothesis on future observed signals.

5.3 How can we measure the uncertainty

Before providing visual examples and equations of our uncertainty measures, we remind the importance of weighting the votes of each hypothesis proportionally to their current probability. We then present our two methods. The first method measures the uncertainty on the expected signals between each hypothesis. The second method measures uncertainty on the meaning space by making hypothesis on future observed signals.

5.3.1 The importance of weighting

We want to measure the uncertainty about both the tasks and signal models in order to collect information allowing reducing this uncertainty. The uncertainty is

therefore not constant, it depends on our current belief about each hypothesis and must be updated when new teaching signals are observed.

As we want to find which task is the correct one among the set of task hypothesis, our aim is to pull apart the hypotheses that are currently the more probable. Once we have ruled out half of the hypotheses, we should only focus on differentiating the remaining hypotheses. Therefore, when estimating the uncertainty, we should weight each vote according to each hypothesis probability. In practice, if one hypothesis has a probability of 1 (i.e. all other hypotheses have a probability of zero) there should be no uncertainty for all state-action pairs.

5.3.2 A measure on the signal space

As explained in section 5.2, our uncertainty measure should take into account both the uncertainty due to the task and the signal model. We illustrate how this uncertainty can be measured by comparing the expected signals from each task hypothesis.

Symmetric signal models

We start by considering the situation of Figure 5.1 where G1 and G2 have symmetric signal models. As depicted in Figure 5.3, when selecting action left, both hypothesis agree that they should receive a signal in the right part of the feature space, even if they disagree on its meaning. Therefore taking action left in state 3 has low uncertainty on the expected signal.

However for action down, both hypothesis agree they should receive a signal of meaning "incorrect", but disagree on the expected location of such signal in the feature space (see Figure 5.4). Therefore taking action down in state 3 has high uncertainty on the expected signal.

Identical signal models

We now consider the situation of Figure 5.2 where G1 and G2 have the same signal model. As depicted in Figure 5.5, when going down both hypothesis agree that they should receive a signal in the left part of the feature space, and agree on its meaning. Therefore taking action down in state 3 has low uncertainty on the expected signal.

However for action left, both hypothesis disagree about the meaning of the signal they should receive, and, as both share the same signal model, they expect a signal in different locations of the feature space (see Figure 5.6). Therefore taking action left in state 3 has high uncertainty on the expected signal.

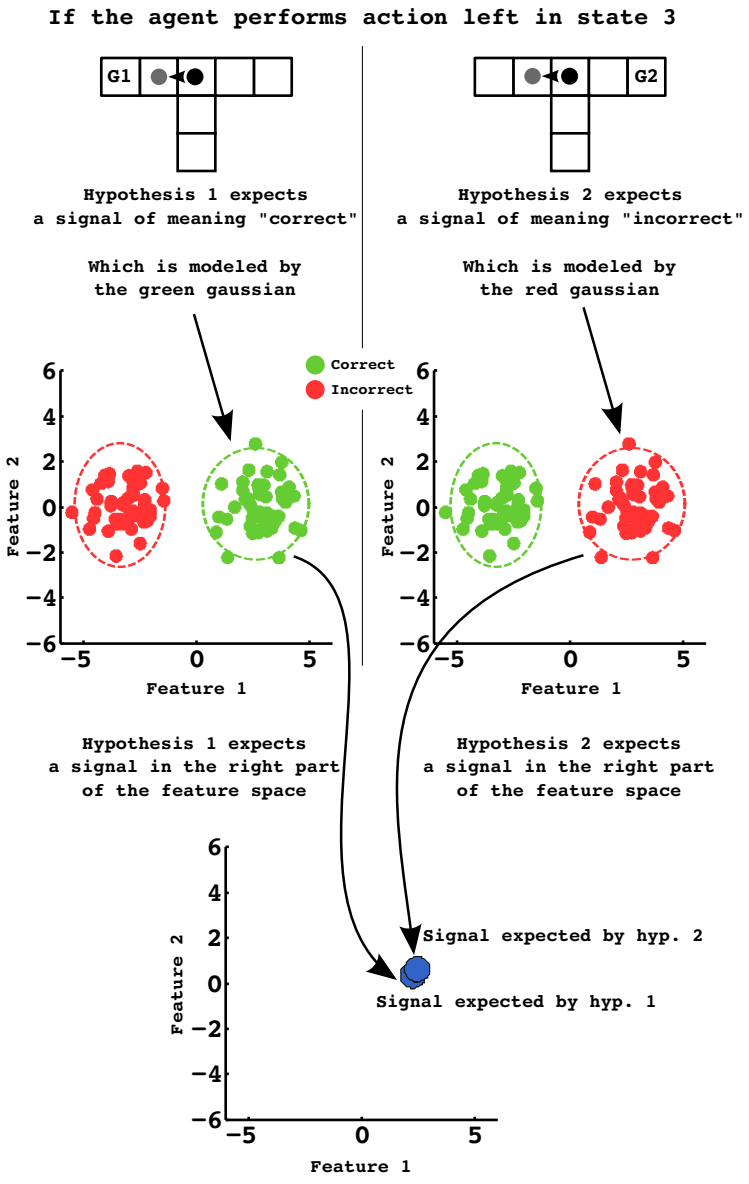


Figure 5.3: Expected signal for both hypothesis if agent performs action left in state 3 and given they currently have a symmetric interpretation of the signals (see Figure 5.1). Both hypotheses expect the same signal, therefore there is no uncertainty associated to this state-action pair, and the agent should not select this action.

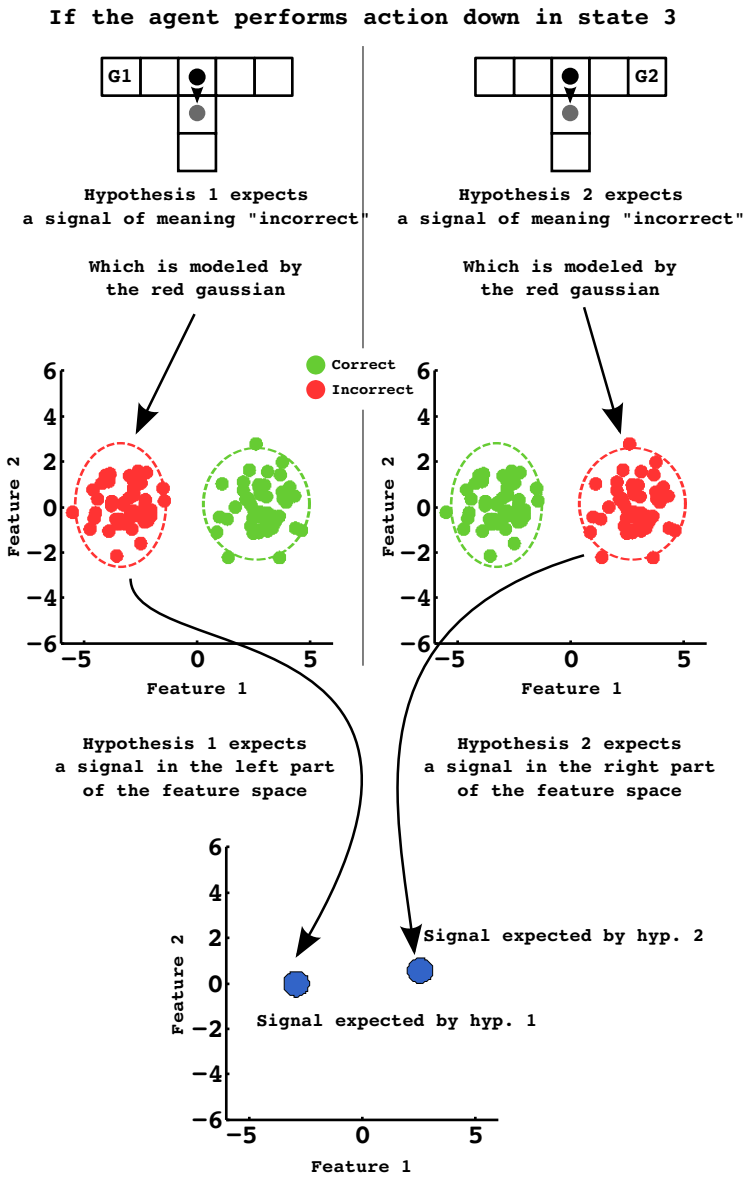


Figure 5.4: Expected signal for both hypothesis if agent performs action down in state 3 and given they currently have a symmetric interpretation of the signals (see Figure 5.1). The two hypotheses expect two different signals, therefore there is high uncertainty associated to this state-action pair, and the agent should better perform this action.

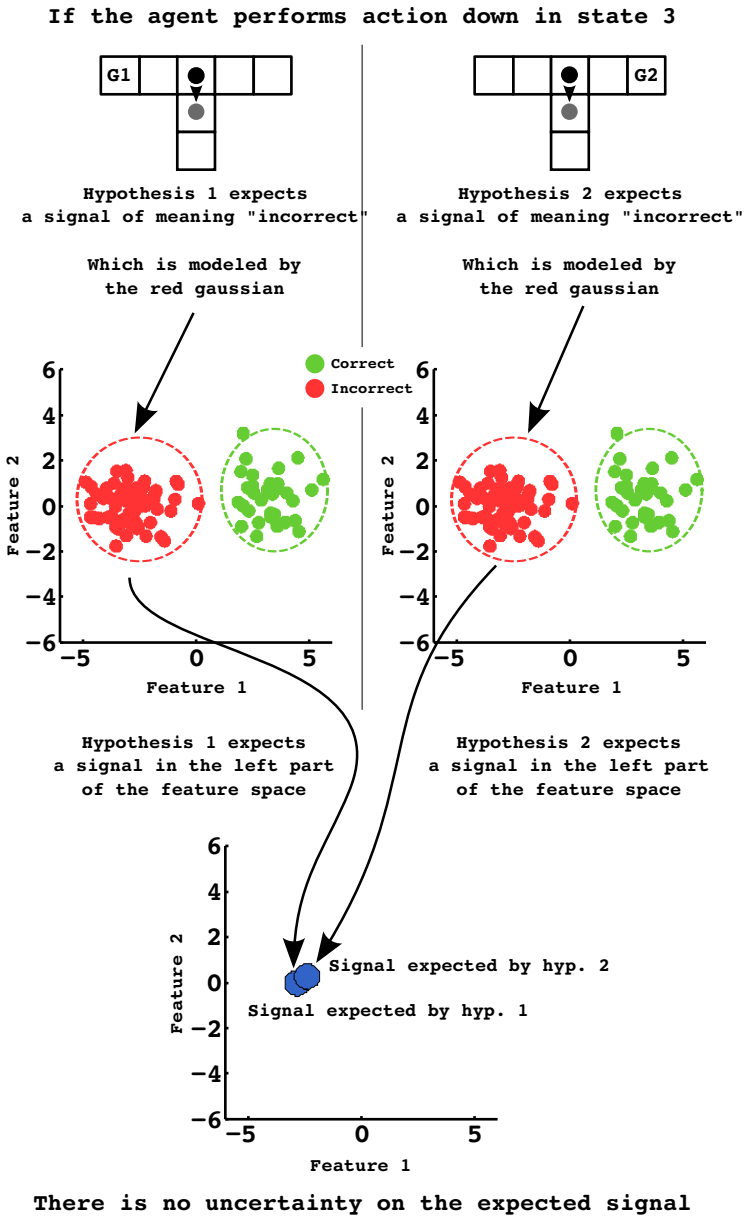


Figure 5.5: Expected signal for both hypothesis if agent performs action down in state 3 and given they currently have a similar interpretation of the signals (see Figure 5.2). Both hypothesis expect the same signal, therefore there is no uncertainty associated to this state-action pair, and the agent should not select this action.

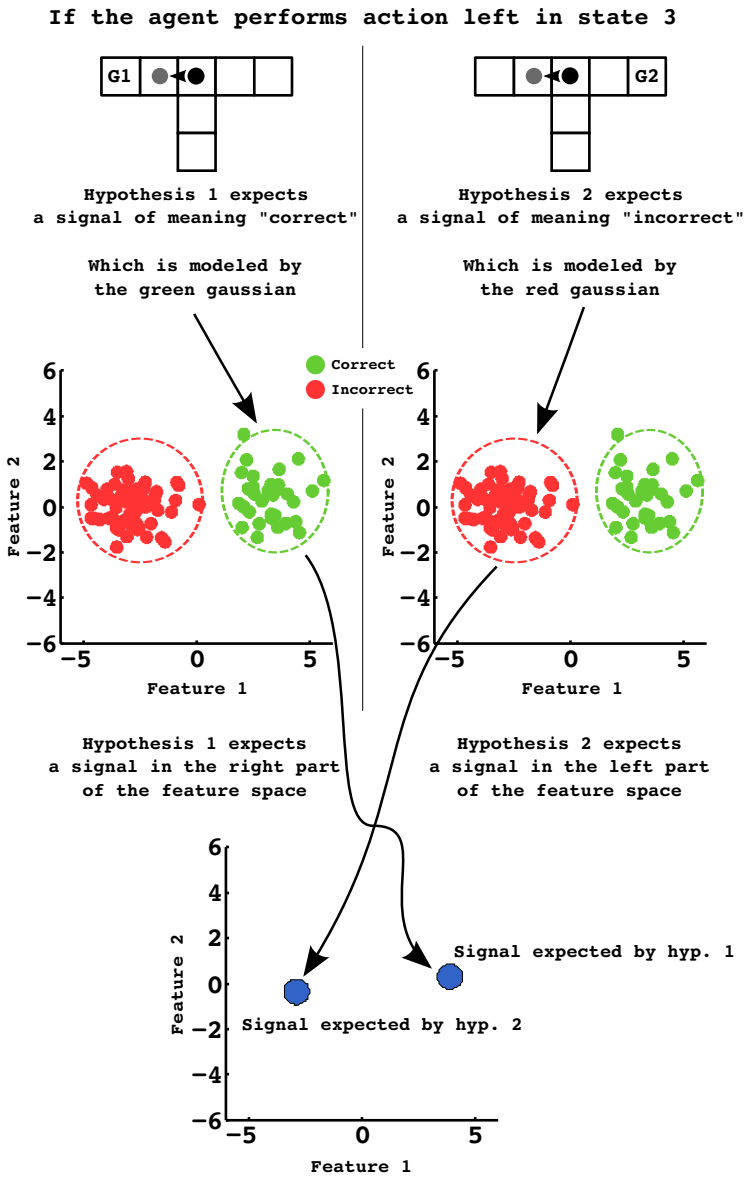


Figure 5.6: Expected signal for both hypothesis if agent performs action left in state 3 and given they currently have a similar interpretation of the signals (see Figure 5.2). The two hypothesis expect two different signals, therefore there is high uncertainty associated to this state-action pair, and the agent should better perform this action.

Equations

To sum up, to compute the uncertainty associated to a state-action pair, we can compute the similarity between the expected signals from each task. The more the expected signals are similar the less there is uncertainty. And we remind that the vote of a hypothesis should be weighted by the current probability of this hypothesis.

Our visual examples represent the expected signal for each hypothesis as the mean of the signal distribution corresponding to the expected label. This is a very rough approximation, indeed, the signals are modeled by a Gaussian distribution. Therefore comparing the similarity between the respective distribution would be more suited than comparing only their mean. Moreover, the frame function is not always deterministic. For example, we take into account possible teaching mistakes by assigning some probability of receiving an "incorrect" feedback while the action was optimal. Ideally, this should also be taken into account when computing the similarity between signal distributions. In addition, our examples consider only two hypotheses, and as soon as the number of hypotheses increases, we should compute the similarity between multitudes of distributions.

Measuring similarity between expected signals can be complex. In practice, it will be easier and more efficient to compute the uncertainty based on the mean of the distribution only. Whatever the method chosen, we define a similarity matrix S where each element $S_{ij}(s,a)$ corresponds to the similarity between the expected signals from tasks i and task j if action a is performed in state s.

The final uncertainty value $U(s,a)$ is computed as the opposite of the weighted sum of the similarity matrix elements:

$$U(s,a) \quad = \quad -\sum_{i=1}^{T}\sum_{j=1}^{T} \; S_{ij}(s,a)\; p(\xi_i)\; p(\xi_j) \tag{5.1}$$

Computed for every state and action, this measure is then used as an exploration bonus to guide the agent towards states that better disambiguate task hypotheses. We provide an example of planning using this method in chapter 7.3, where we measure similarity between Gaussian distribution using their means only.

5.3.3 A measure projected in the meaning space

Estimating uncertainty in the signal space can be very costly. It requires computing, for every state-action pair, the overlap between many continuous probability distributions weighted by their respective expected contributions. In this subsection we present another metric for computing the uncertainty which relies on our pseudo-likelihood measure defined in chapter 4.4. This method relies on sampling some teaching signals and asking every hypothesis whether the sampled signals are expected or not given a state-action pair.

Symmetric signal models

We start by considering the situation of Figure 5.1 where the two hypothesis have symmetric signal models. As depicted in Figure 5.7, when selecting action left in state 3 and if the user sends a signal in the right part of the feature space, both hypothesis agree that this particular signal is expected given this state-action pair. Hypothesis 1 expects a signal of meaning "correct", and the teacher signal is classified as being of class "correct". Hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "incorrect". Therefore receiving this particular signal after taking action left in state 3 has low uncertainty.

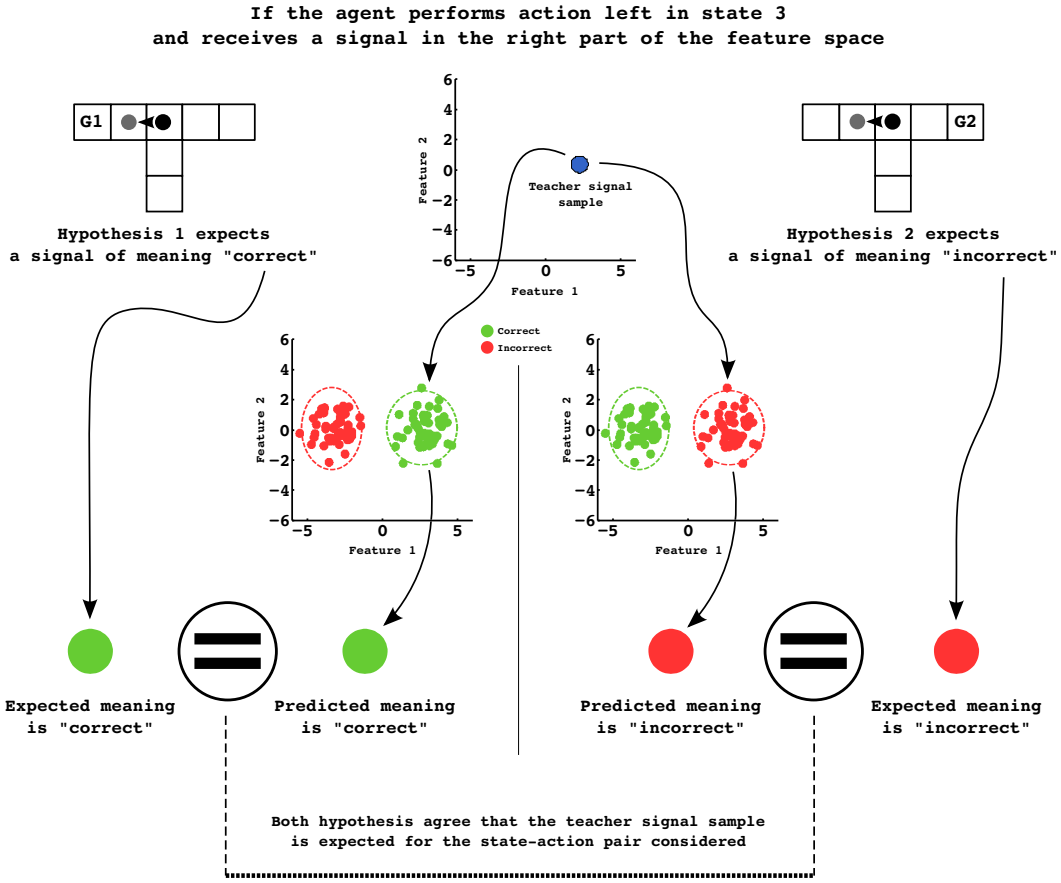


Figure 5.7: Matching between expected labels and the prediction of a teaching signal sampled on the right side of the feature space and if the agent performs action left in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.1). Both hypotheses agree that the label associated to a signal on the right side of the feature space matches with the label predicted given the frame and the state-action pair considered. Therefore, there is no uncertainty associated to this state-action pair and the agent should not select action left.

This same process can be executed for any teaching signal. For example, as depicted in Figure 5.8, considering a teaching signal on the left side of the feature space, if the agent performs action left in state 3, both hypothesis agree that this particular signal is not expected. Hypothesis 1 expects a signal of meaning "correct", and the teacher signal is classified as being of class "incorrect". Hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "correct". Therefore receiving this particular signal after taking action left in state 3 has low uncertainty.

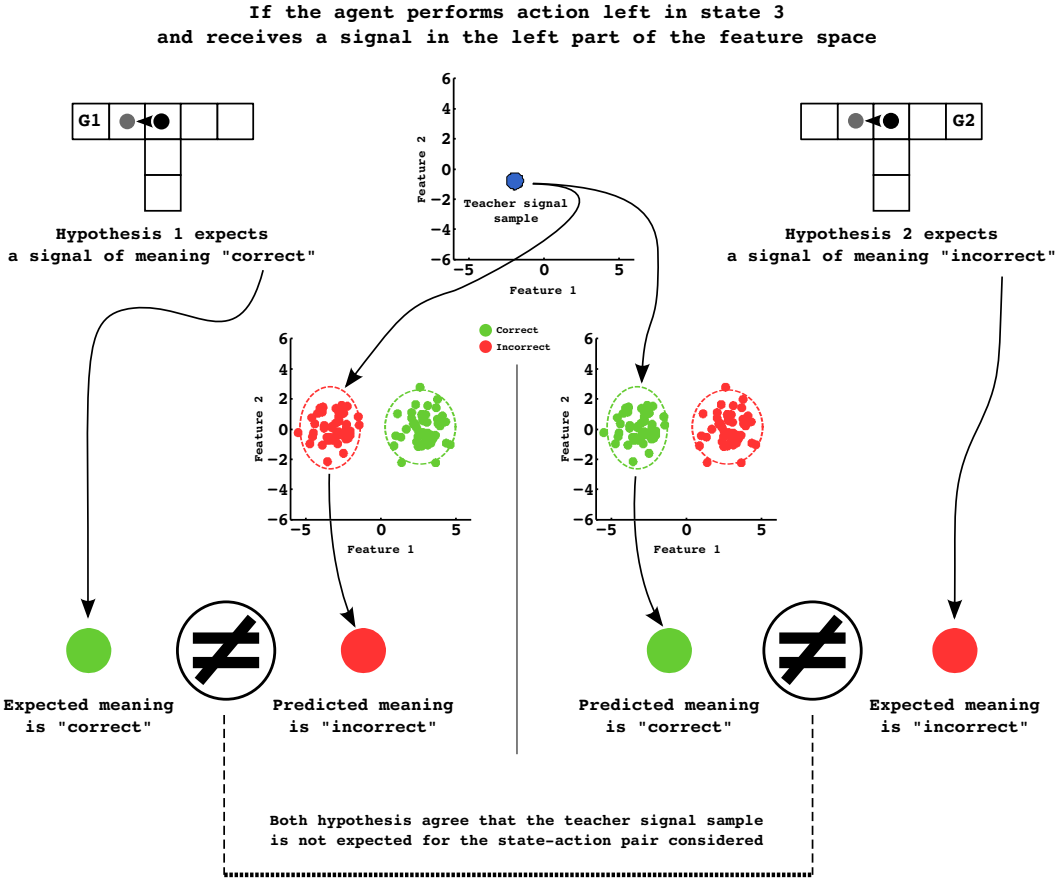


Figure 5.8: Matching between expected labels and the prediction of a teaching signal sampled on the left side of the feature space for the two hypothesis if the agent performs action left in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.1). Both hypotheses agree that the label associated to a signal on the left side of the feature space does not match with the label predicted given the frame and the state-action pair considered. Therefore, there is no uncertainty associated to this state-action pair and the agent should not select action left.

However for action down, the two hypotheses disagree on whether such signals are expected or not given the state-action pair considered. As depicted in Figure 5.9, when selecting action down in state 3 and if the user sends a signal in the right part of the feature space, hypothesis 1 expects a signal of meaning "incorrect", and the teacher signal is classified as being of class "correct". And hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "incorrect". Therefore receiving this particular signal after taking action down in state 3 is not expected for hypothesis 1 but expected for hypothesis 2, there is high uncertainty.

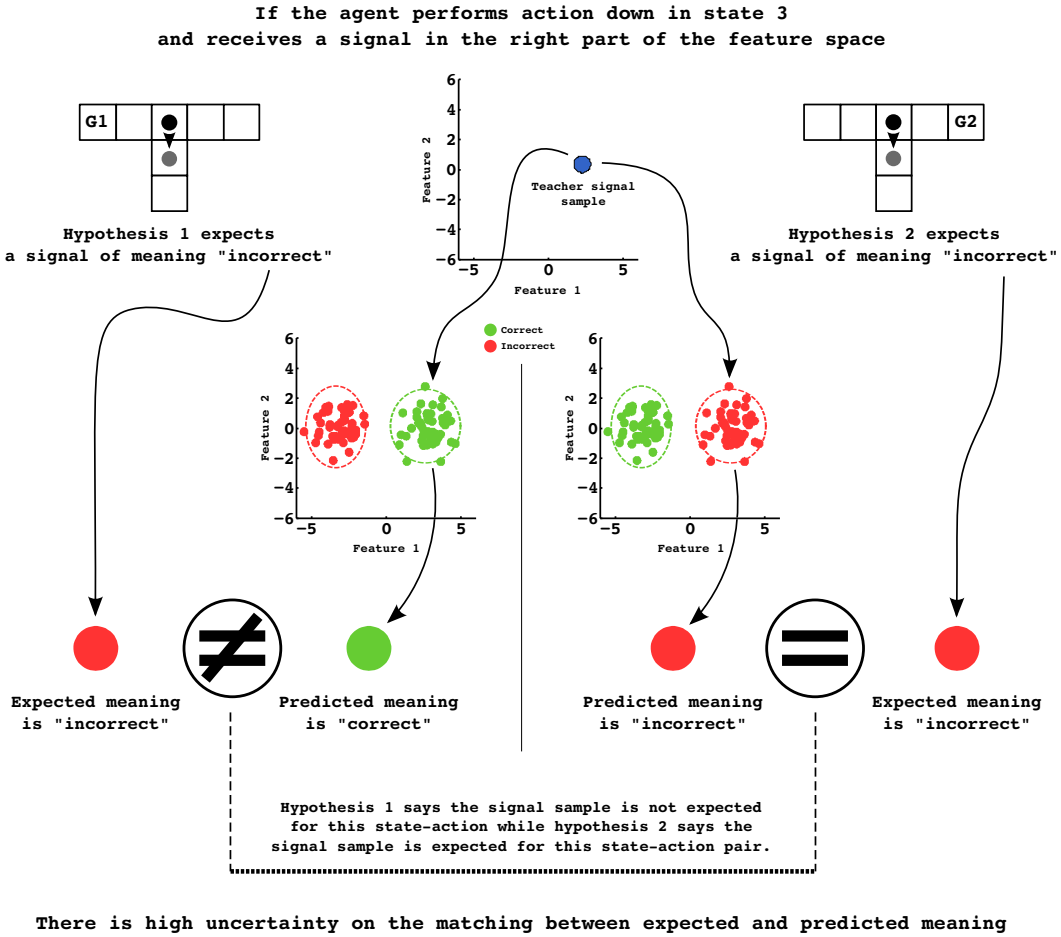


Figure 5.9: Matching between expected labels and the prediction of a teaching signal sampled on the right side of the feature space if the agent performs action down in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.1). Hypothesis 1 identify the signal as meaning "correct" which was not expected, while hypothesis 2 expected a signal meaning "incorrect" and classify the signal as "incorrect" which is what was expected. Therefore there is high uncertainty associated to this state-action pair and the agent should better perform action down in order to disambiguate between hypotheses.

Similarly, as depicted in Figure 5.10, considering a teaching signal on the left side of the feature space, if the agent performs action down in state 3, hypothesis 1 expects a signal of meaning "incorrect", and the teacher signal is classified as being of class "incorrect". And hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "correct". Therefore receiving this particular signal after taking action down in state 3 is expected for hypothesis 1 but not expected for hypothesis 2, there is high uncertainty.

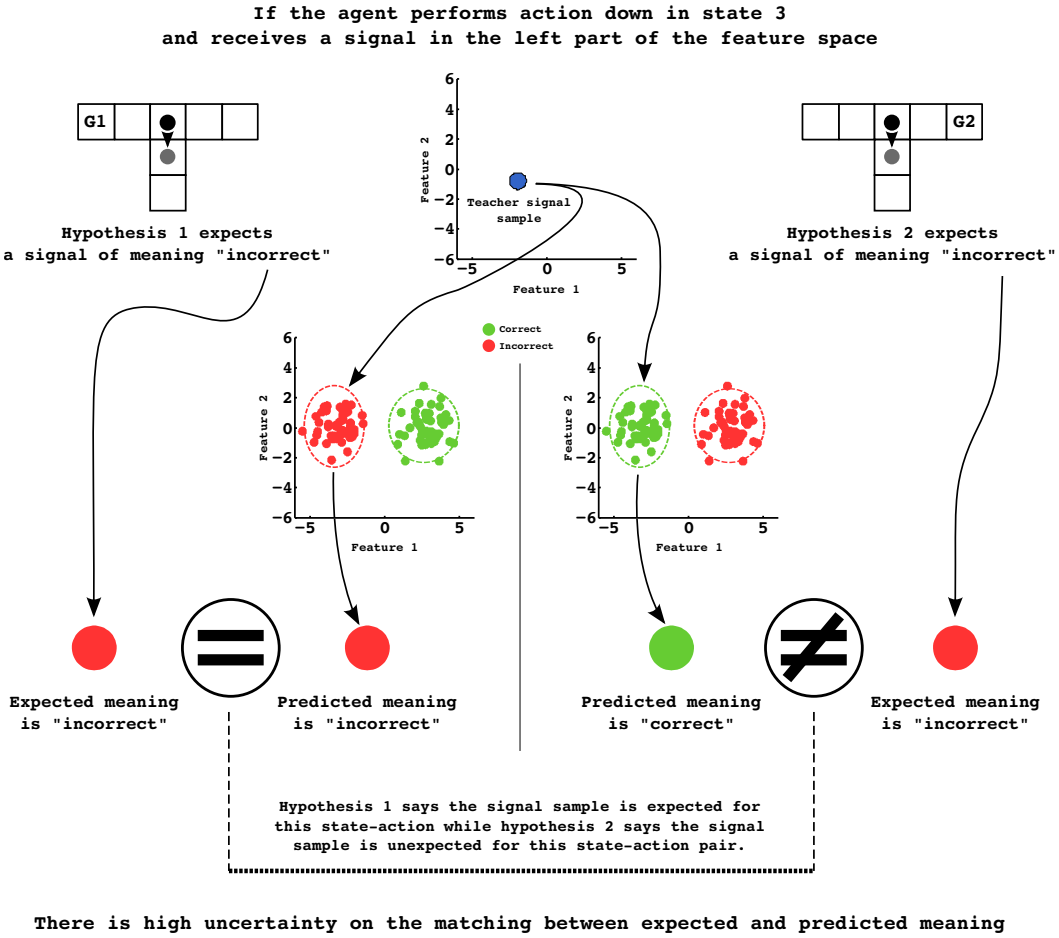


Figure 5.10: Matching between expected labels and the prediction of a teaching signal sampled on the left side of the feature space for the two hypothesis if the agent performs action down in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.1). Hypothesis 1 says a signal on the left side of the feature space means "incorrect" which was expected given the interaction frame, while hypothesis 2 expected a signal meaning "incorrect" but classify the signal as "correct" which was not expected. Therefore there is high uncertainty associated to this state-action pair and the agent should better perform action down in order to disambiguate between hypothesis..

Identical signal models

We now consider the situation of Figure 5.2 when the same model is shared between hypothesis. As depicted in Figure 5.11, when selecting action down in state 3 and if the user sends a signal in the right part of the feature space, both hypothesis agree that this particular signal is unexpected given this state-action pair. Hypothesis 1 expects a signal of meaning "incorrect", and the teacher signal is classified as being of class "correct". Hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "correct". Therefore receiving this particular signal after taking action down in state 3 has low uncertainty.

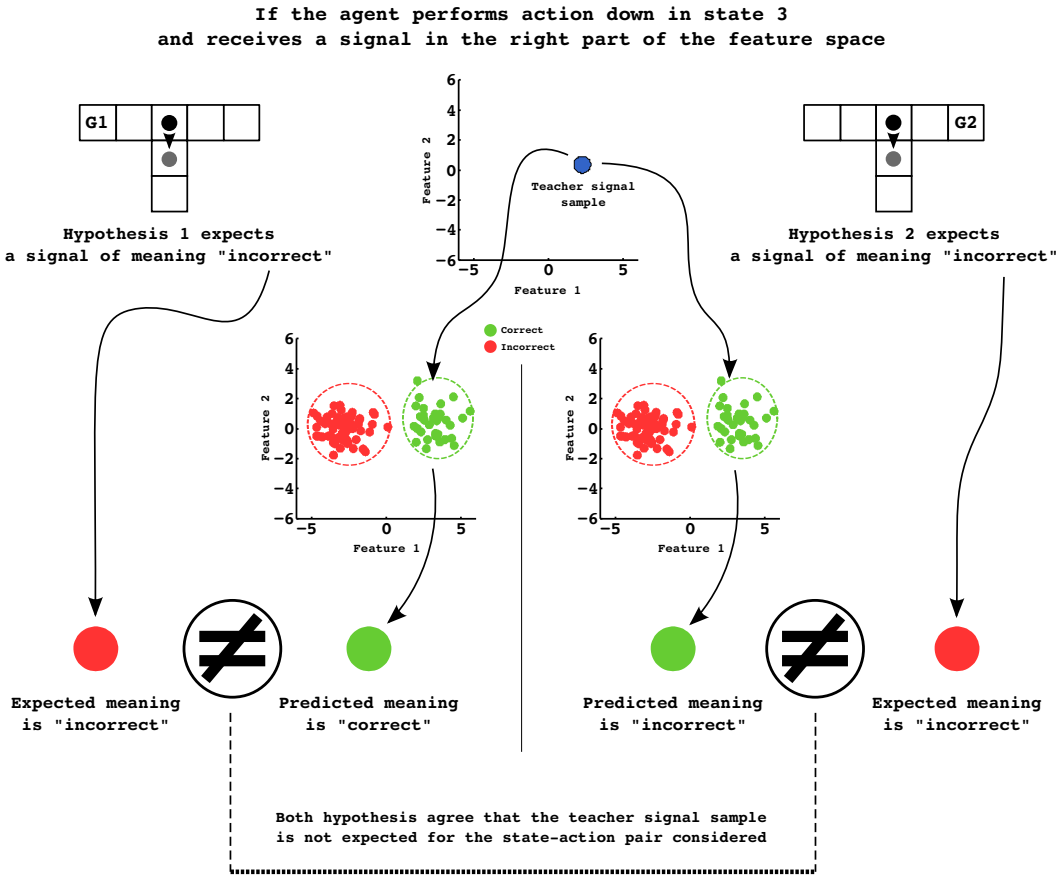


Figure 5.11: Matching between expected labels and the prediction of a teaching signal sampled on the right side of the feature space for the two hypothesis if the agent performs action down in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.2). Both hypotheses agree that the label associated to a signal on the right side of the feature space does not match with the label predicted given the frame and the state-action pair considered. Therefore there is no uncertainty associated to this state-action pair and the agent should not select action down.

This same process can be executed for any teaching signal. For example, as depicted in Figure 5.12, considering a teaching signal on the left side of the feature space, if the agent performs action down in state 3, both hypothesis agree that this particular signal is expected. Hypothesis 1 expects a signal of meaning "incorrect", and the teacher signal is classified as being of class "incorrect". Hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "incorrect". Therefore receiving this particular signal after taking action down in state 3 has low uncertainty.

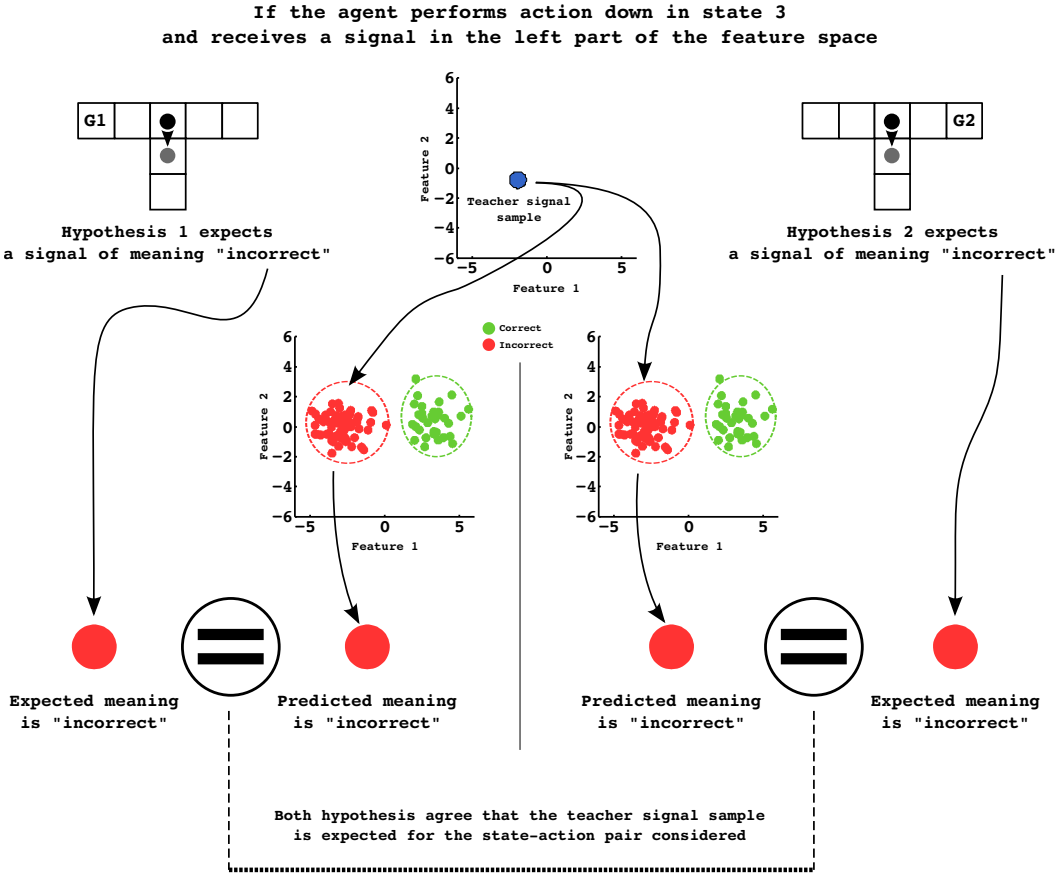


Figure 5.12: Matching between expected labels and the prediction of a teaching signal sampled on the left side of the feature space for the two hypothesis if the agent performs action down in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.2). Both hypotheses agree that the label associated to a signal on the left side of the feature space match with the label predicted given the frame and the state-action pair considered. Therefore there is no uncertainty associated to this state-action pair and the agent should not select action down.

However for action left, the two hypotheses disagree on whether such signals are expected or not given the state-action pair considered. As depicted in Figure 5.13, when selecting action left in state 3 and if the user sends a signal in the right part of the feature space, hypothesis 1 expects a signal of meaning "correct", and the teacher signal is classified as being of class "correct". And hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "correct". Therefore receiving this particular signal after taking action down in state 3 is expected for hypothesis 1 but not expected for hypothesis 2, there is high uncertainty.

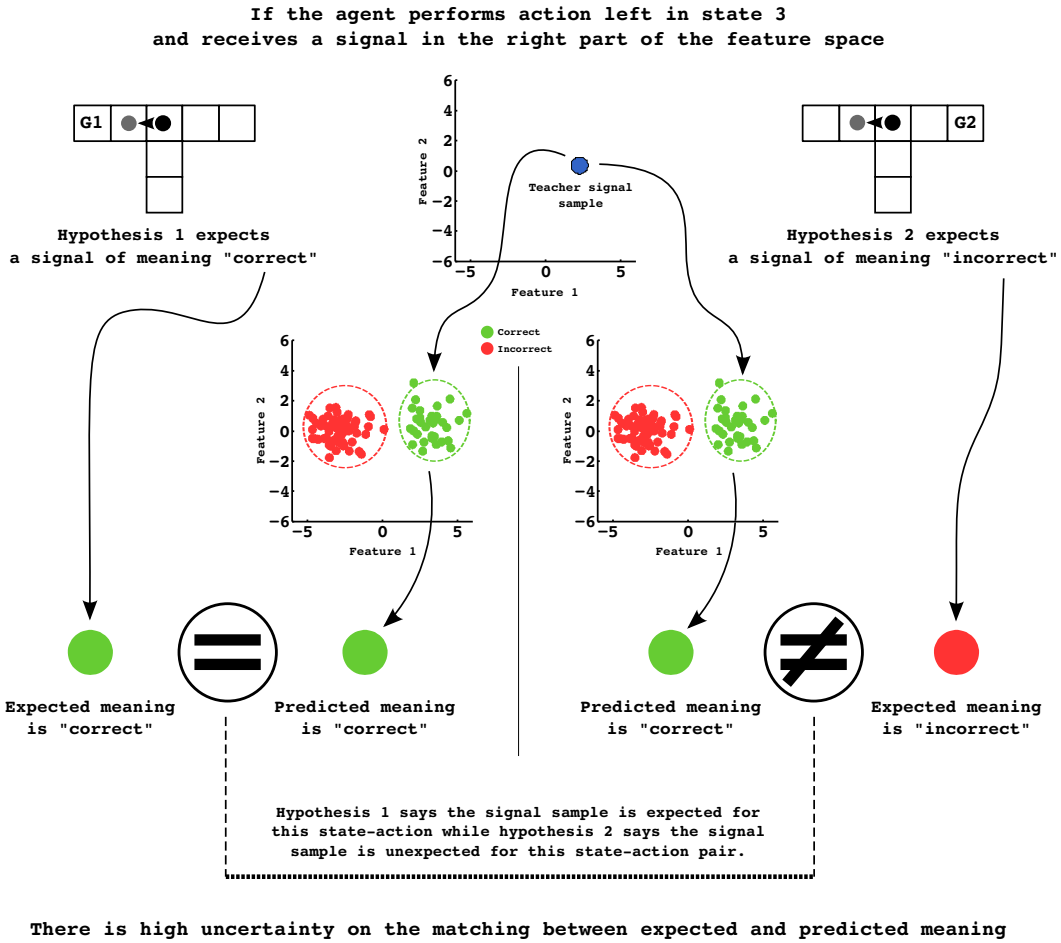


Figure 5.13: Matching between expected labels and the prediction of a teaching signal sampled on the right side of the feature space for the two hypothesis if the agent performs action left in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.2). Hypothesis 1 identify the signal as meaning "correct" which was not expected, while hypothesis 2 expected a signal meaning "incorrect" but classify the signal as "correct" which was not expected. Therefore there is high uncertainty associated to this state-action pair and the agent should better perform action left in order to disambiguate between hypotheses.

Similarly, as depicted in Figure 5.14, considering a teaching signal on the left side of the feature space, if the agent performs action left in state 3, hypothesis 1 expects a signal of meaning "incorrect", and the teacher signal is classified as being of class "incorrect". And hypothesis 2 expects a signal of meaning "incorrect" and the teacher signal is classified as being of class "correct". Therefore receiving this particular signal after taking action down in state 3 is not expected for hypothesis 1 but expected for hypothesis 2, there is high uncertainty.

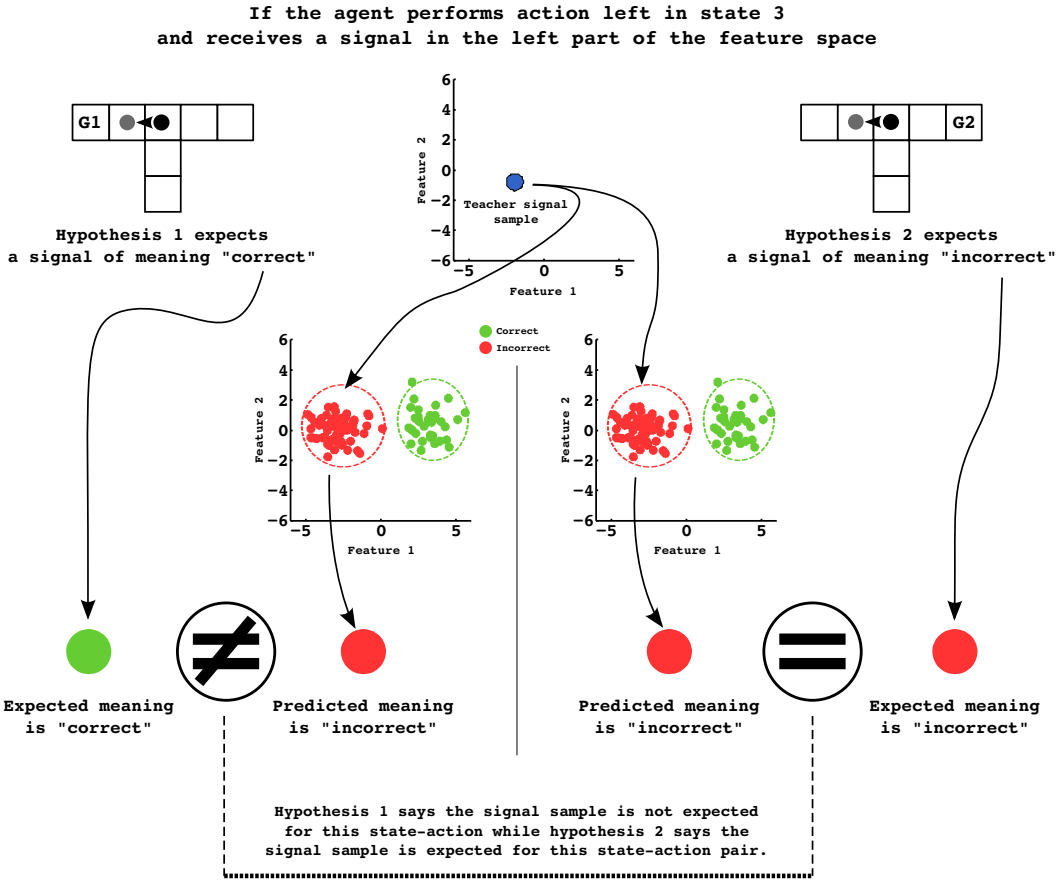


Figure 5.14: Matching between expected labels and the prediction of a teaching signal sampled on the left side of the feature space for the two hypothesis if the agent performs action left in state 3 and the two hypothesis currently have a symmetric interpretation of the signals (see Figure 5.2). Hypothesis 1 says a signal on the left side of the feature space means "incorrect" which was not expected given the interaction frame, while hypothesis 2 expected a signal meaning "incorrect" and classify the signal as "incorrect" which is what was expected. Therefore there is high uncertainty associated to this state-action pair and the agent should better perform action left in order to disambiguate between hypotheses.

Equations

To summarize, in order to estimate uncertainty between hypothesis for a given state-action pair, we can ask the system to classify some teaching signals e and compute the probability that the predicted labels l^c equals the expected labels l^f. By comparing the resulting joint probability between each hypothesis, if there is low variance there is low uncertainty. Respectively, if there is high variance, there is high uncertainty.

This measure has the important advantage of using the same equations as the one used for computing the likelihood of each task (chapter 4.4.2). Additionally, we do not have to compute the similarity between continuous distributions, and only rely on the classifiers, that are already computed. We only need to compute the predicted labels (l^c) associated to the sampled signals (e) once per hypothesis. Then, to compute the full uncertainty map for each state and action pair, we have to compare these predicted labels with the expected labels (l^f) from each state-action pair and each hypothesis.

We note $J^{\xi_t}(s,a,e) = p(l^c = l^f | s, a, e, \theta_{xi_t}, \xi_t)$, which is Equation 4.2 given the classifier θ_{ξ_t} associated to task ξ_t and a particular state, action, and signal. We note $J^{\xi}(s,a,e)$ the vector $[J^{\xi_1}(s,a,e), \ldots, J^{\xi_T}(s,a,e)]$. And $W_i^{\xi} = [W^{\xi_1}, \ldots, W^{\xi_T}]$ the weights associated to each hypothesis. Such weights can be the one defined in Equation 4.10 (i.e. the minimum of pairwise normalized likelihoods) or the probabilities from Equation 4.9 (i.e. the normalized likelihoods).

The uncertainty of one state-action pair ((s,a)) given a signal e is computed as the weighted variance of the joint probabilities:

$$U(s,a|e) = weightedVariance(J^{\xi}(s,a,e), W^{\xi}) \tag{5.2}$$

The uncertainty for a state-action pair is given by:

$$U(s,a) \quad = \quad \int_e U(s,a|e)p(e)de \tag{5.3}$$

which we approximate by summing values of $U(s,a|e)$ for different signals e:

$$U(s,a) \quad \approx \quad \sum_e U(s,a|e)p(e) \tag{5.4}$$

with $p(e)$ assumed uniform.

Signal samples (e) could be sampled randomly in all the feature space. However, there is a high risk of taking non-relevant samples, as well as likely practical computational problem for some classifiers. In practice, it is better to sample some signals from our past history of interaction, which may lead to overfitting problems that can be solved by using a cross validation procedure.

Our measure of uncertainty $U(s,a)$ will be higher when, for a given state-action there is a high incongruity of expectation between each hypothesis and according to the probability of each hypothesis. This measure is then used as a classical

exploration bonus method. We provide an example of planning using this method in the following of this chapter.

Interestingly the two approaches proposed generalize over other active planning methods [Lopes 2009b], if the signal to meaning classifier is known, i.e. the same for all hypothesis, our equations reduces to the one presented in [Lopes 2011]. For example, our first method, which relies on measuring the uncertainty on expected signals, will be equivalent to a measure on the expected meanings because all hypotheses will have identical signal's models. For our second method, all classifiers will be identical, therefore the resulting equations will no longer be dependent on signal e. As our uncertainty function combines uncertainty on both signal and task space, when former is known, the latter becomes the sole source of ambiguity.

5.3.4 Why not building model first

A usual question concerning Figure 5.2, is why don't we first select state-action pairs which lead to unequivocal interpretation of the signals? Indeed, it allows to first build a database of known signal-label pairs. The resulting classifiers could then be used to classify further teaching signals, as in a calibration procedure.

Obviously this is not always possible, for example if we add a third hypothesis G3, that is at the bottom of the T trunk, it is no more possible to find actions leading to an unequivocal interpretation of the received signal. Neither the left and right actions (Figure 5.15), nor the up and down actions (Figure 5.16) alone allow to have an unequivocal interpretation of the teaching signals. However taking all the actions and exploring all the state space still highlight hypothesis 1 (G1) as being the goal state the user as in mind (Figure 5.17).

In all the experiments presented in this thesis, there are no state-action pairs allowing for an unequivocal interpretation of the teaching signal.

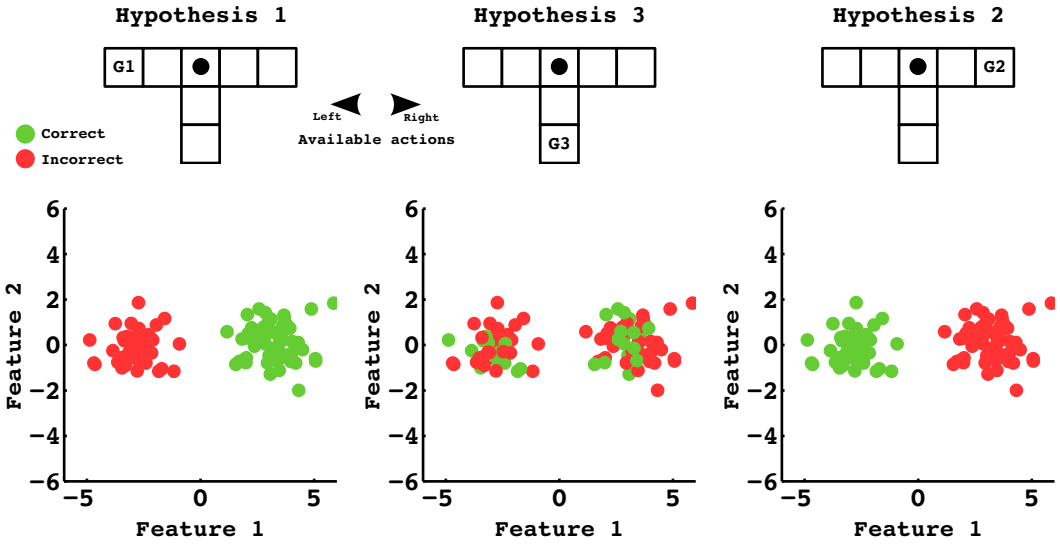


Figure 5.15: Interpretation hypothesis made by the agent according to G1 (left), G2 (right), and G3 (middle).

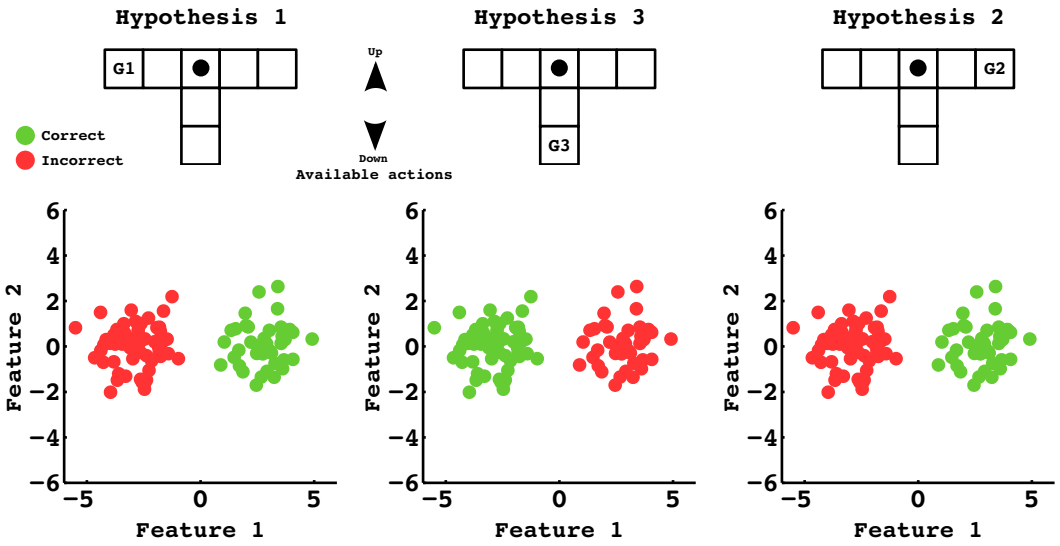


Figure 5.16: Interpretation hypothesis made by the agent according to G1 (left), G2 (right), and G3 (middle). The agent performs only up and down actions. The labels associated to G1 and G2 are similar but the labels associated to G3 are symmetric. Up and down actions do not create an unequivocal interpretation of signal considering these three hypotheses. Moreover u and down actions do not allow to discard any of the hypothesis.

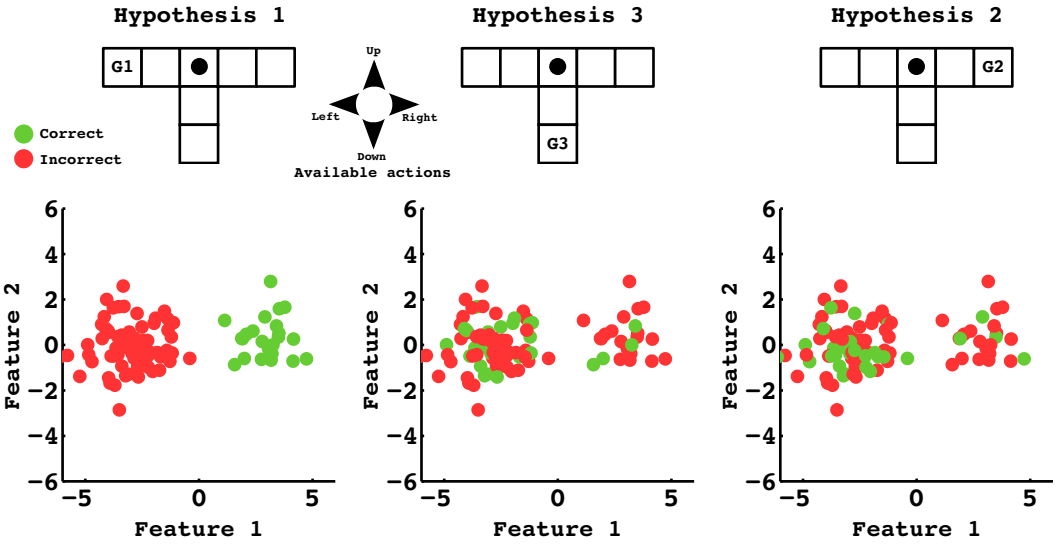


Figure 5.17: Interpretation hypothesis made by the agent according to G1 (left), G2 (right), and G3 (middle). The agent performs all possible actions. The labels associated to G1 are more coherent than with the spatial organization of the data than the labels associated to G2 and G3, which tells us G1 is the task the user has in mind.

5.4 Method

In the subsequent analysis, we considerer a reaching task where an agent lives in a grid world and should learn to which square the teacher wants it to go. We considered the teacher provides feedback for the actions taken by the agent. We will use artificial dataset of different qualities and dimension to evaluate our algorithm.

5.4.1 World and Task

We consider a 5x5 grid world, where an agent can perform five different discrete actions: move up, down, left, right, or a "no move" action. The user goal is to teach the agent to reach one (unknown to the agent) of the 25 discrete positions that represent the set of possible tasks. We thus consider that the agent has access to 25 different task hypotheses (one with goal location at each of the cells). We use *Markov Decision Processes* (MDP) to represent the problem [Sutton 1998]. From a given task ξ, represented as a reward function, we can compute the corresponding policy π_ξ using, for instance, Value Iteration [Sutton 1998]. We consider the user is providing feedback on the agent's actions, and use the feedback frame function as previously defined in chapter 4 Equation 4.13.

5.4.2 Simulated teaching signals

We analyze our algorithm using artificial datasets. The goal of this evaluation is to analyze the feasibility of learning a task from scratch in a 5x5 grid world. The artificial dataset are composed of two classes, with 1000 examples per class. Each signal was generated by sampling from a normal distribution with a covariance matrix of diagonal 1 and mean selected randomly. The datasets were generated while varying two factors: (i) the dimensionality of the data, where 2, 5, 10 and 30 features were tested; and (ii) the quality of the dataset, measured in terms of the ten-fold accuracy the classifier would obtain. We exemplify datasets of different qualities in Figure 5.18.

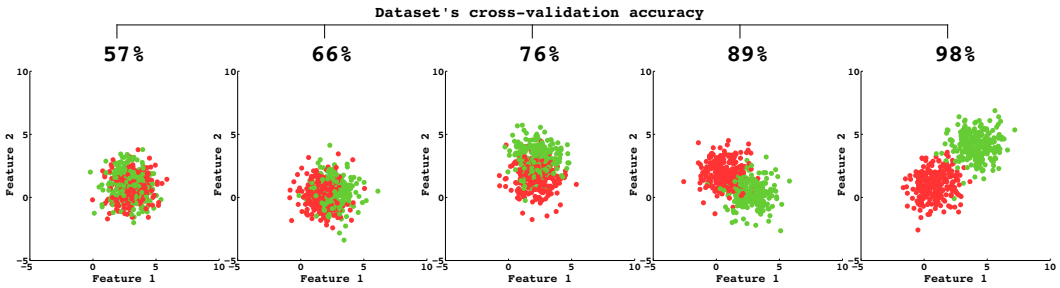


Figure 5.18: Artificial datasets generated by sampling from normal distributions with a covariance matrix of diagonal 1 and means selected randomly. From left to right, we show datasets of increasing quality as measured by a 10 fold cross-validation train-test procedure using a Gaussian classifier.

5.4.3 Signal properties and classifier

We rely on Gaussian classifiers and model the signals using independent multivariate normal distributions for each class, $\mathcal{N}(\mu_c, \Sigma_c), \mathcal{N}(\mu_w, \Sigma_w)$. With θ the set of parameters $\{\mu_c, \Sigma_c, \mu_w, \Sigma_w\}$. Given the high dimensionality of some datasets we also need to regularize. For this we apply shrinkage to the covariance matrix ($\lambda = 0.5$) and compute the value of the marginal pdf function using a noninformative (Jeffrey's) prior [[Gelman 2003], p88]:

$$p(e|l,\theta) \quad = \quad t_{n-d}(e|\mu_l, \frac{\Sigma_l(n+1)}{n(n-d)}) \tag{5.5}$$

where θ represents the ML estimates (mean μ_l and covariance Σ_l for each class l) required to estimate the marginal under the Jeffreys prior, n is the number of signals, and d is the dimensionality of a signal feature vector.

Finally to compute the probability of a label given a signal, we use the Bayes

rules as follows:

$$\begin{aligned} p(l = l_i|e, \theta) &= \frac{p(e|l = l_i, \theta)p(l = l_i)}{\sum_{k=1,\dots,L} p(e|l = l_k, \theta)p(l = l_k)} \\ &= \frac{p(e|l = l_i, \theta)}{\sum_{k=1,\dots,L} p(e|l = l_k)} \end{aligned}$$

As we do not have a priori knowledge on the user intended meaning, we assume all labels are equiprobables, i.e. $\forall k,\ p(l = l_k) = \frac{1}{L}$.

5.4.4 Task Achievement

We use Equation 4.7 to compute the likelihood of each task using a 10 fold cross-validation to compute the confusion matrix. It implies we train 250 classifiers at each iteration. To compute the probability of each task, we will rely on the minimum of pairwise normalized likelihood measure as defined in Equation 4.10.

A task is considered completed when the confidence level β has been reached for this task and the agent is located at the task associated goal state. If the corresponding state is the one intended by the user, it is a success. Whatever the success or failure of the first task, the user selects a new task, i.e. a new goal state, randomly. The agent resets the task likelihoods, propagates the previous task labels to all hypothesis, and the teaching process starts again. At no point the agent has access to a measure of its performance, it can only refer to the unlabeled feedback signals from the user.

5.4.5 Evaluation scenarios

Using our artificial datasets, three different evaluations are performed: (i) the performance of our proposed planning strategy versus a) random action selection, b) greedy action selection, and c) the task-only uncertainty based method; (ii) the time required by the agent to complete the first task (i.e. to reach the first target with confidence), and (iii) the number of tasks that can be completed in 500 iterations.

5.4.6 Settings

We used $\alpha = 0.1$, $\beta = 0.9$. For dataset of dimension d, we started computing likelihoods after $d + 10$ steps as equation 5.5 requires at least $d + 1$ samples and to allow for cross validation. For the planning (Eq. 5.4) we sample randomly 20 signals from D_M.

5.5 Illustration of the grid world scenario

We illustrate in Figure 5.19, a smaller 3x3 grid world example and the results of the hypothetic labeling process. The teacher is providing feedback with respect to hypothesis 1. The labeling process for hypothesis 1 is the more coherent. We note

that hypothesis 9 has symmetric properties with hypothesis 1 but the use of the "no move" action allows breaking that symmetry.

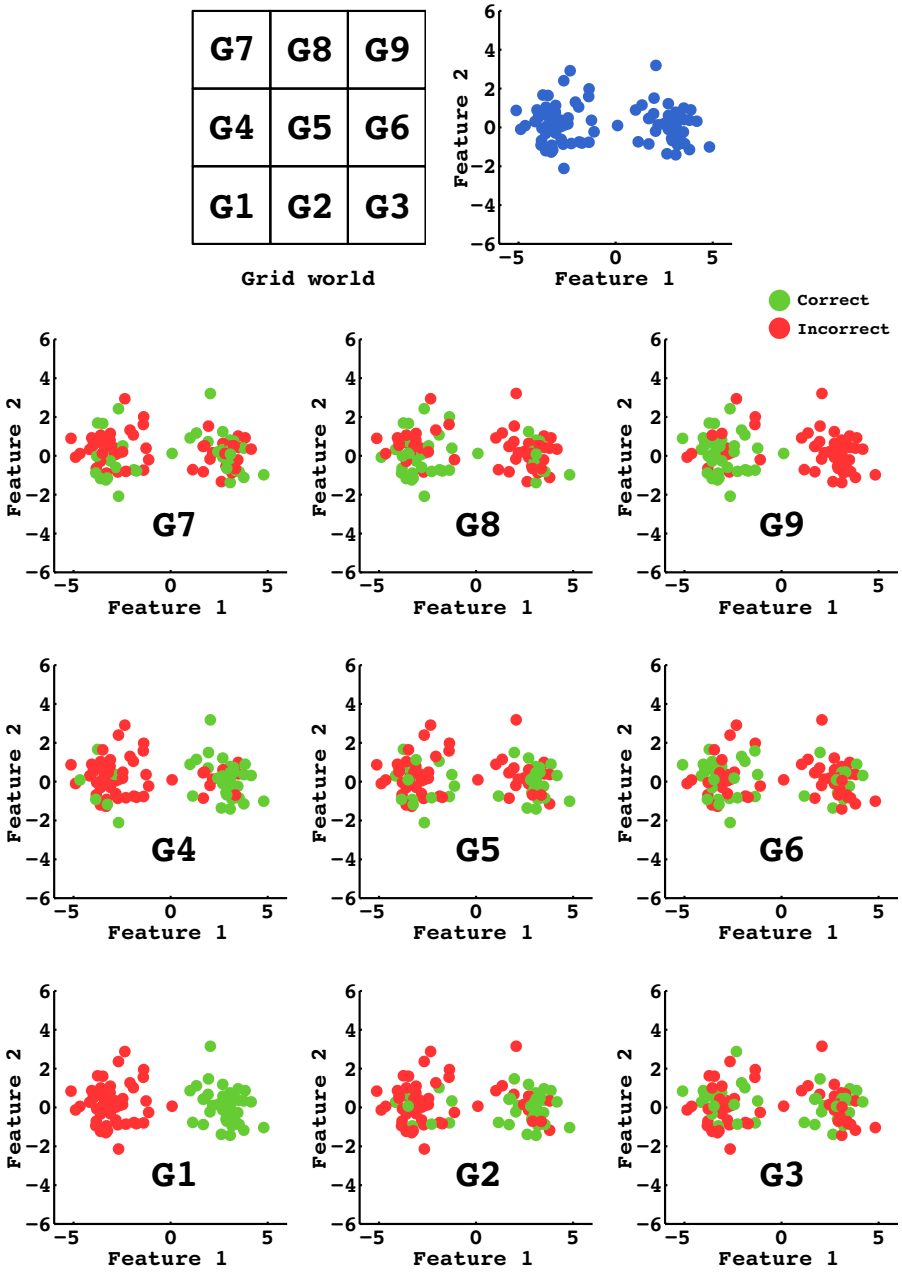


Figure 5.19: A schematic view of a 3x3 grid world scenario. There are nine possible hypotheses and the agent is acting randomly for this example. We show the results of the labeling process considering the feedback frame. The teacher is providing feedback with respect to hypothesis 1. The labeling process for hypothesis 1 is more coherent with the spatial organization of the data, which indicates it is the one taught by the user. Hypothesis 9 has symmetric properties with hypothesis 1 but the use of the "no move" action allows breaking that symmetry.

5.6 Results

In the following, we present most of the results in terms of the quality of the dataset, measured as the ten-fold classification accuracy that a calibrated signal classifier would obtain. Each simulation was run 100 times using different sampled datasets, and their associated box plots were computed. For each boxplot, colored bars show the interquartile range (between 25th and 75th percentile). The median and the mean are marked as a horizontal line and a colored dot respectively. The two whiskers show the 5th and 95th percentiles, black crosses are outliers.

We first study the impact of the uncertainty based exploration approach proposed in this chapter. We then evaluate the performance and robustness of our algorithm with respect to the dimension and the quality of the datasets.

5.6.1 Planning methods

Figure 5.20 compares the number of steps (with maximum values of 500 steps) needed to reach the first task with confidence using different planning methods. Following the most probable task (i.e. going greedy) does not allow the system to explore sufficiently. The planning method proposed in this chapter leads the system towards state-action pairs that discriminates hypotheses faster. Furthermore, our planning method performs better than assessing uncertainty on the meaning space only. Given these results, the remaining of this section will only consider our planning method.

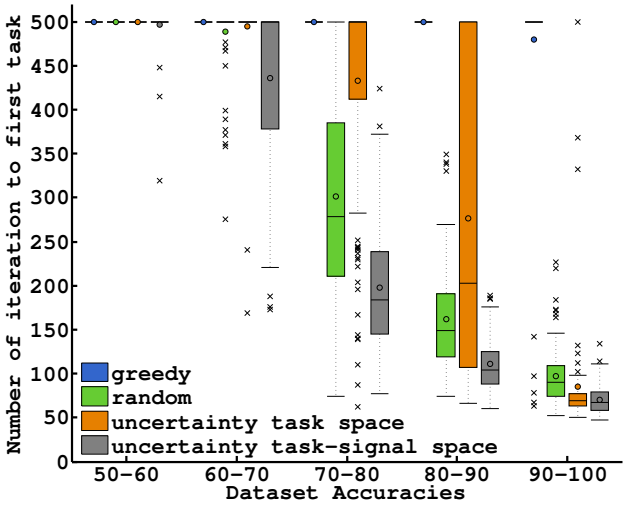


Figure 5.20: Number of steps to complete the first task, comparison of different exploration methods with 30 dimensional artificial data. When learning from scratch, planning upon both the task and the signal to meaning mapping uncertainty performs better than relying only on the uncertainty about the task. Greedy action selection rarely disambiguates between hypotheses.

As explained in section 5.3, the machine needs to collect two types of information,

some about the true underlying model (Fig. 5.2) and some to differentiate between hypotheses (Fig. 5.1). The properties of the grid world make the random strategy quite efficient at collecting those two types of information. The differences between our active planning method and a random exploration should be sharper when navigating a complex maze.

We present a small study on how different world properties affect the learning efficiency in chapter 7.2.

Finally, we note that all planning methods were switched to pure exploitation (greedy) once the confidence level was reached. Therefore the performance in Figure 5.20 compares the ability of the different methods to discriminate between different task hypotheses, not their ability to solve the task itself.

5.6.2 Dimensionality

Figure 5.21 compares the number of steps (with maximum values of 500 steps) needed to reach the first task with confidence when learning from scratch considering different dimensionality of datasets. The convergence speed is only slightly affected by the features dimensionality. On the other hand, the dataset quality (measured in terms of its associated ten-fold accuracy) is the main cause of performances decay. Furthermore, for these datasets with accuracies between 50% and 60%, the system is not able to identify a task with confidence after 500 steps. This is the expected behavior as for such dataset (see Figure 5.18 left), none of the hypothesis is able to find a classifier of good enough accuracy and should therefore not take any decision.

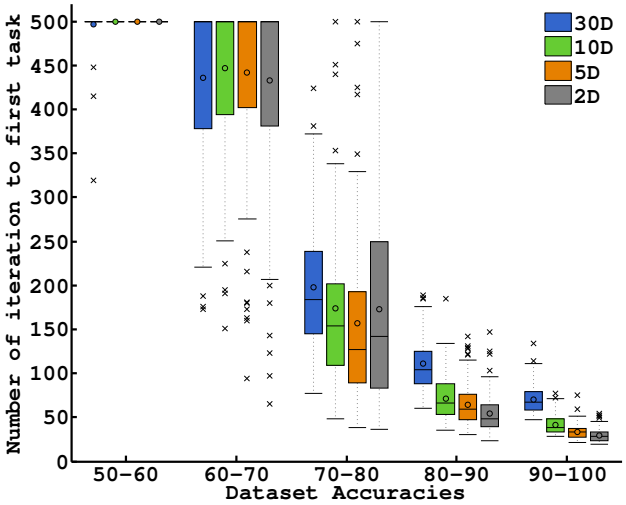


Figure 5.21: Number of steps to complete the first task using artificial data. For datasets of low quality, i.e. under 60 percent accuracy, the confidence threshold cannot be reached in 500 steps. The datasets' quality, more than their dimensionality, impacts the learning time.

5.6.3 Reuse

Once the first task is completed, a new one is selected randomly. Figure 5.22 compares the number of tasks that can be achieved in 500 steps. As expected, the lower the quality of the data, the less number of tasks can be completed. With dataset of accuracies higher than 90% we can achieve more than 30 tasks on average.

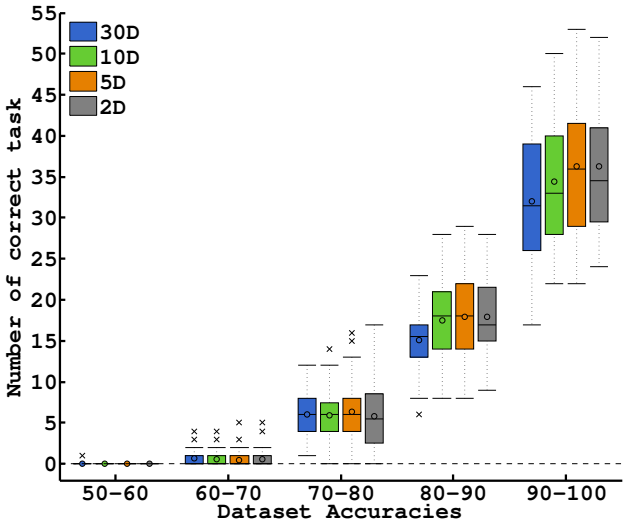


Figure 5.22: Number of tasks correctly achieved in 500 steps using artificial data. Quality of datasets impacts the number of tasks identified in 500 steps because more evidence should be collected to reach the confidence threshold.

Importantly, our algorithm makes very few mistakes when identifying the first task. We reported only 9 erroneous estimations across all simulated experiments (5 in the 70-80 group and 4 in the 80-90 group).

5.7 Discussion

In this chapter, we presented a planning method allowing reducing the number of iterations needed to identify the correct task from unlabeled teaching signals. This method was based on assigning an uncertainty value to each state-action pair. By asking the agent to look for the most uncertain state-action pair, it can collect more useful data to disambiguate faster between the hypotheses. We identified two sources of uncertainty, one coming from the task and the other coming from the signal model associated to each task hypothesis. We presented two methods to measure this uncertainty. The first method measures the uncertainty on the expected signals between each hypothesis. The second method measures uncertainty on the meaning space by making hypothesis on future observed signals.

We want to apply this algorithm to a more concrete scenario with real users. In next chapter, we present a brain computer interaction scenario following the reaching task presented in this section. But instead of using artificial data, we will

investigate how our algorithm scales to the use of brain signals, first in simulation and then during online experiments with real subjects.

Chapter 6

Application to Brain Computer Interaction

Contents

We presented an algorithm that exploits task constraints to solve simultaneously a task under human feedback and learn the associated meanings of the feedback signals. We detailed an uncertainty measure than allow our agent to solve this problem more efficiently and shown that our algorithm can transition from task to task in a smooth way. Our algorithm has important practical application since the user can start controlling a device from scratch, without the need of an expert defining the meaning of signals or carrying out a calibration phase.

The work presented in this chapter is the result of a collaboration with Iñaki Iturrate and Luis Montesano. Code is available online under the github account `https://github.com/jgrizou/` in the following repositories: `lfui`, `experiments_thesis`, and `datasets`.

In this chapter, we explore the use of our algorithm in brain computer interaction scenario. We consider the grid world reaching task scenario as presented in chapter 5.4. After briefly presenting the related work, we introduce the experimental setup and the Error-related potential EEG signals we will use. Then, we first test our algorithm with a database of EEG signals and compare the performance of our method with a calibration procedure method (that first collects signal-label pairs during a calibration period and trains a unique classifier). We will highlight one run of our experiments in detail.

However, we point out a main difference between calibration procedure and our self-calibration method in that the EEG signals properties can be affected by the action selection of the agent. As our planning method cannot guarantee the same agent behavior than during a typical calibration phase, the quality of the signals generated by the users can be impacted. To address this problem, we introduce a prior information of the Error-related potential EEG signals used, namely that the signal corresponding to an "incorrect" meaning are more "powerful" than the one associated to meaning "correct". We will exploit this property, in addition to our interpretation hypothesis method, and show that we can achieve better performances.

Finally, we present results where real users teach our artificial agent by assessing agent's actions in their mind, and without calibrating the system beforehand.

The results with real EEG signals allow us to envision that such algorithm could have practical applications in the real word. By removing the need of an expert to collect and calibrate the system, the use of brain computer interfaces may become more practical allowing their users to go out of the labs.

6.1 Experimental setup and EEG signals

In this section, we introduce the BCI visual navigation experimental setup as well as the brain signals encoding "correct" and "incorrect" feedback we record from the subjects' brain.

6.1.1 The visual navigation task

The setup of our online experiment is shown in Figure 6.1. A human subject is equipped with an EEG cap and is facing a screen displaying a two dimensional grid. The grid is composed of 25 discrete states, 5 rows and 5 columns. In green is displayed an agent that is able to move in the four cardinal direction (N/S/E/W). In red is the target the user selected. The user will mentally assess each agent's action a being "correct" or "incorrect" with respect to the target location.

6.1.2 The brain signals

We are interested by error-related potentials (ErrPs) in the subject's brain activity. These potentials are a specific kind of event-related potential (ERP) generated in the user's brain after s/he assesses actions performed by an external agent

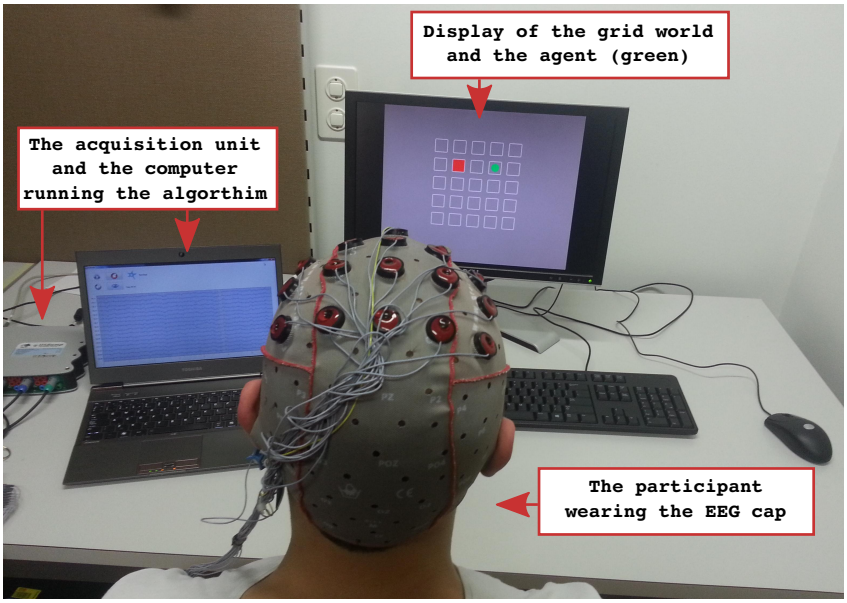


Figure 6.1: The BCI setup for online experiments. On the screen is displayed the grid world with the agent in green. We displayed the intended target in red, which was selected randomly. The purpose of this red square is to help the user remembering the target and our algorithm is at no point aware of the position of this red square.

[Chavarriaga 2010]. Correct and erroneous assessments will elicit different brain signals. As shown in Figure 6.2, the EEG signals associated to "incorrect" labels have slightly higher amplitude than the one associated to "correct" labels, especially at around 350ms, but lower amplitude elsewhere, around 600ms.

Past approaches have already demonstrated that these signals can be classified online with accuracies of around 80% and translated into binary feedback, thanks to a prior calibration session that lasts for 30-40 minutes [Chavarriaga 2010, Iturrate 2013b].

Difference with P300 speller In the related work chapter, we presented the work of Kindermans et al. which also achieve calibration free BCI but consider the speller paradigm using P300 EEG signals (section 2.5.1). We identified an important difference between our respective work in that the speller task ensure that one signal out of 6 is encodes a P300 signal. This information allows their EM algorithm to attribute the class of each identified cluster. Following our approach we do not need do guarantee such ratio, which makes our approach applicable to a broader variety of task.

In addition the nature of the brain signals considered in our respective work differs, they use P300 signals and we use ErrP signals. The P300 and the ErrP both come from the same family of EEG signal, called event-related potentials (ERP) [Chavarriaga 2014]. Both are generated as a reaction to internal or external events. The main difference is that P300 can be generated as many times as needed, i.e. each

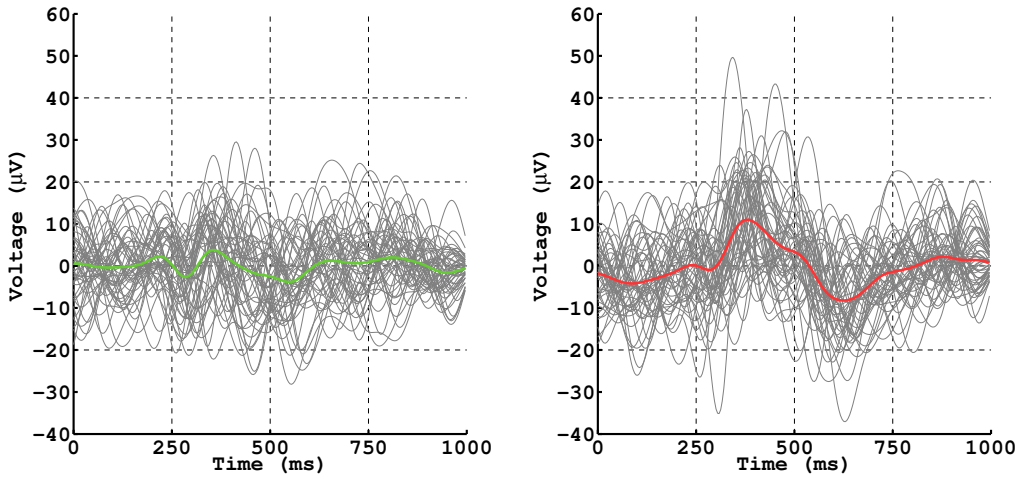


Figure 6.2: Low-pass filtered EEG signals associated to "correct" labels (left) and to "incorrect" labels (right). The signals for each class have slightly different amplitudes, especially at around 300ms.

time the correct row or column is flashed a P300 is triggered in the subject's brain. Therefore it is possible to average, increasing the signal to noise ratio. Whereas the ErrP cannot be generated on demand, once the agent performed an unexpected, erroneous, action the corresponding potential must be detected when it appears, it is a single trial detection. Hence the ErrP are harder to use in online interactive scenarios.

Building the feature vector After every agent's action, the brain signals from the user are recorded via a computer using a gtec system. To build our feature vector we consider two fronto-central channels (FCz and Cz) in a time window of $[200, 700]$ ms (0 ms being the action onset of the agent) and downsampled the signal to 32 Hz. Each element of the feature vector is the value in microvolts of the signal at the corresponding time.

6.1.3 The signal model

Following the literature [Lotte 2007, Blankertz 2010], we rely on Gaussian classifiers and model the signals using independent multivariate normal distributions for each class, $\mathcal{N}(\mu_c, \Sigma_c), \mathcal{N}(\mu_w, \Sigma_w)$. With θ the set of parameters $\{\mu_c, \Sigma_c, \mu_w, \Sigma_w\}$. Given the high dimensionality of our datasets we also need to regularize. For this we apply shrinkage to the covariance matrix ($\lambda = 0.5$) and compute the value of the marginal pdf function using a noninformative (Jeffrey's) prior [[Gelman 2003], p88]:

$$p(e|l,\theta) \quad = \quad t_{n-d}(e|\mu_l, \frac{\Sigma_l(n+1)}{n(n-d)}) \tag{6.1}$$

where θ represents the ML estimates (mean μ_l and covariance Σ_l for each class l) required to estimate the marginal under the Jeffreys prior, n is the number of signals, and d is the dimensionality of a signal feature vector.

Finally to compute the probability of a label given a signal, we use the bayes rules as follows:

$$\begin{aligned} p(l = l_i|e, \theta) &= \frac{p(e|l = l_i, \theta)p(l = l_i)}{\sum_{k=1,\dots,L} p(e|l = l_k, \theta)p(l = l_k)} \\ &= \frac{p(e|l = l_i, \theta)}{\sum_{k=1,\dots,L} p(e|l = l_k)} \end{aligned}$$

As we do not have a priori knowledge on the user intended meaning, we assume all labels are equiprobables, i.e. $\forall k,\ p(l = l_k) = \frac{1}{L}$.

6.2 Using pre-recorded EEG signals

Before trying out our algorithm with real subjects, we test the feasibility of our method using pre-recorded ErrP datasets. The objective of this analysis is to study the scalability of our method to EEG data, which may have different properties than our artificial dataset. We will see that our algorithm maintains good properties with EEG signals.

6.2.1 Datasets and scenario

EEG datasets The ErrPs EEG data were recorded in a previous study [Iturrate 2013b] where participants monitored on a screen the execution of a task where a virtual device had to reach a given goal. The motion of the device could be correct (towards the goal) or erroneous (away from the goal). The subjects were asked to mentally assess the device's movements as erroneous or non-erroneous. The EEG signals were recorded with a gtec system with 32 electrodes distributed according to an extended 10/20 international system with the ground on FPz and the reference on the left earlobe. The ErrP features were extracted from two fronto-central channels (FCz and Cz) within a time window of $[200, 700]$ ms (0 ms being the action onset of the agent) and downsampled to 32 Hz. This leaded to a vector of 34 features.

Comparison with calibration methods In order to show the benefit of learning without explicit calibration, we compare our method with a typical supervised BCI calibration procedure. Such calibration procedure requires an experimenter to record enough labeled data from the user. Following the literature on ErrPs [Chavarriaga 2010, Iturrate 2013b] our training data will consist of 80 percent of positive examples (associated to a correct feedback) and 20 percent of negative examples (associated to an incorrect feedback). ErrPs signals are generated by a user when he observes unexpected agent's behaviors, which explains why, during the calibration phase of their system, researchers uses 80 percent of the time a correct

action (i.e. moving towards the goal), and only 20 percent of the time an incorrect action, which is therefore unexpected. Our proposed algorithm is compared with different (but standard) sizes of calibration datasets: 200, 300 and 400 examples.

6.2.2 One example detailed

We use Equation 4.7 to compute the likelihood of each task using a 10 fold cross-validation to compute the confusion matrix.

Figure 6.3 shows one particular run of 500 steps comparing our self-calibration method with a calibration procedure of 400 steps. The two independent runs use a real EEG dataset with 80% ten-fold cross-validation classification accuracy. As our algorithm is operational from the first step, it can estimate the real task when sufficient evidences have been collected. On the other hand, a calibration approach collects signal-label pairs for a fixed number of steps, and use the resulting classifier without updating it. This provokes that, during the calibration phase, no tasks can be learned, substantially delaying the user's online operation.

Of important interest is the ability of the algorithm to evaluate when sufficient evidence has been collected. The dataset considered is of relatively good quality, and we do not need 400 steps to identify the first task. When doing a calibration procedure, the experimenter cannot know in advance the quality of each particular subject. Therefore, he must run a calibration for long enough so as to have enough examples to adapt to differences in signals' quality.

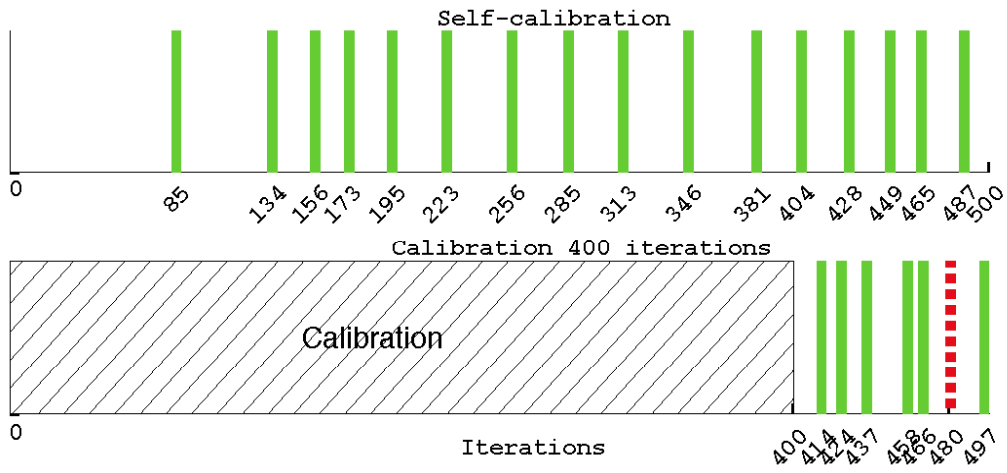


Figure 6.3: Timeline of one run using an EEG dataset of 80 percent ten-fold cross-validation classification accuracy. Self-calibration (top) versus 400 steps calibration (bottom). Green (filled) and red (dashed) bars represent respectively correct and incorrect task achievements. The proposed self-calibration method allows reaching a first task faster than it takes to run a calibration procedure.

Figure 6.4 shows the evolution of classification rate between our self-calibration method and a calibration procedure of 400 steps. As our method assigns different la-

bels to each new teaching signal, the resulting classifiers have different performances, which helps identifying the correct task. Once a task is identified (e.g. step 85 and 134), as explained in chapter 4.4.4, the corresponding labels are taken as ground truth, and all classifiers will be the same for one iteration. As a consequence, all classifiers have the same accuracies each time a task is completed (e.g. step 85 and 134). As the agent starts exploring again to estimate the new tasks, all the classifiers except the true one will start to have worse accuracies again.

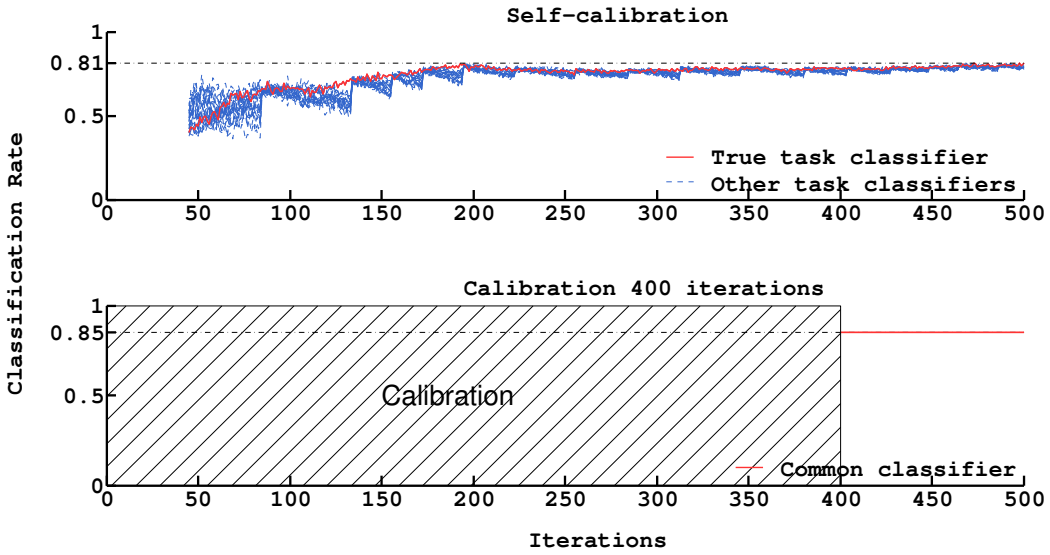


Figure 6.4: Evolution of the classification rates of all classifiers for one run using EEG data. Self-calibration (top) versus 400 steps calibration (bottom). On top, the red line represents the classifier corresponding to the successive task taught by the user, the dashed blue lines represent the classifiers of all other tasks. Our method updates 25 classifiers every steps.

Before the step 200, we observe a strong evolution of every classifier (see Figure 6.4 top), during this phase the algorithm does not have enough data to create a good classifier of the data and rely mainly on the hypothetic labeling process to differentiate between hypotheses. For example at step 130, the classifier corresponding to the true task is of better quality than all the other ones. Therefore, via the estimation of its confusion matrix, its predictions are more trusted than the predictions from the other hypothesis.

However after step 200, the difference between classifier qualities is very small. Indeed, 5 tasks have already been identified and they now share most of their signal-label pairs (due to the propagation of label after each task identified seen in chapter 4.4.4). From iteration 200, the algorithm behaves similarly if a calibrated classifier common to all hypotheses was provided. Indeed, all classifiers are similar and make similar predictions.

Interestingly, these two modes are captured by the same equation (see Equation 4.5), which compares predicted and expected labels while taking into account

the confidence in the predictions of the classifiers using their respective estimated confusion matrix.

6.2.3 Planning

Figure 6.5 compares the number of steps (with maximum values of 500 steps) needed to identify the first task when learning from scratch with different planning methods. Our proposed planning method guide the agent towards states that maximize disambiguation among hypotheses, which outperforms the other action selection methods. Given these results, the remainder of this section will only consider our planning method.

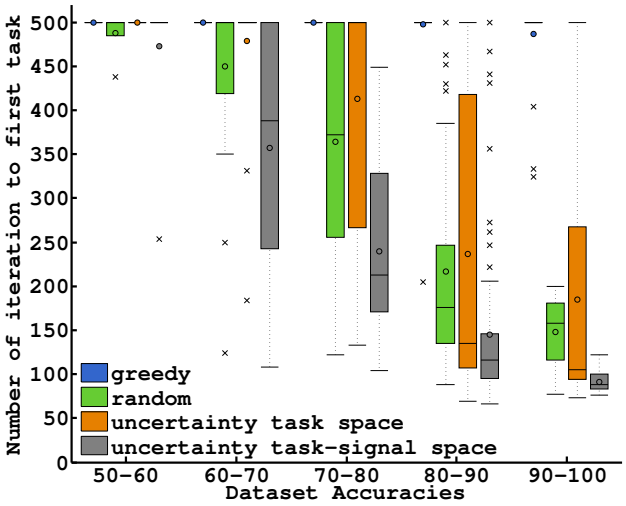


Figure 6.5: Number of steps to complete first task using EEG data of different quality. The EEG data have similar properties than our 30 dimensional simulated data in Figure 5.20. Our planning method, based on both the task and the signal to meaning mapping uncertainty, is more efficient that choosing action randomly, greedily, or only based on the uncertainty on the task.

6.2.4 Time to first task

Figure 6.6 shows the number of iterations needed to identify the first task and compares the results between our self-calibration method and calibration periods of 200, 300, and 400 iterations. The percentage of time the first task was correctly identified is shown on top of each box plot. For our self-calibration method, the learning time is strongly correlated with the dataset quality. This is an important properties, it means our method is able to adapt online to the quality of the data it receives. For datasets of more than 80 percent classification accuracy, we can complete the first task in less than 150 steps on average and without mistake.

Compared to calibration based methods, our algorithm allows completing the first task without errors. However, calibration based methods, which do not update

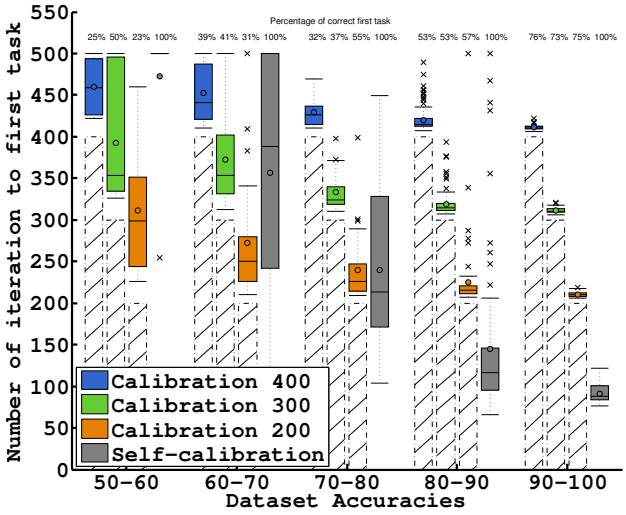


Figure 6.6: Number of steps to complete the first task using EEG data. The agent plans its action based on the uncertainty measure. The percentage of time the first task was correctly identified is shown on top of each box plot. For our self-calibration method, the learning time is strongly correlated with the dataset quality. Contrary to the calibration approaches, we do not make mistakes with low quality datasets.

their classifier once calibrated, identify more tasks incorrectly. In addition, the time to complete the first task is less correlated with the datasets quality for the calibration based methods than for our self-calibration procedure. Training one classifier per task makes our algorithm more robust. We will propose several explanations of the bad performances of calibration based methods in section 6.3.

6.2.5 Cumulative performances

Figure 6.7 compares the number of tasks achieved in 500 steps. With datasets of more than 90% classification accuracy, we achieve more than 20 tasks on average.

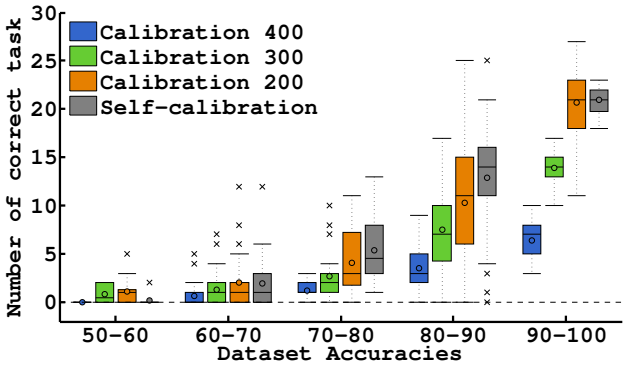


Figure 6.7: Number of tasks correctly achieved in 500 steps with EEG data. Calibration methods reach fewer targets because most of the time is spent for calibrating.

The calibration based methods cannot complete a significant number of tasks because most of the experimental time is spent on calibrating the system. A calibration of 200 steps allow to reach as many target correctly than the self-calibration method, but it also produces many wrong estimation, see Figure 6.8. For calibration based methods, the less time spent on calibration the poorer the classifier, which implies more mistakes.

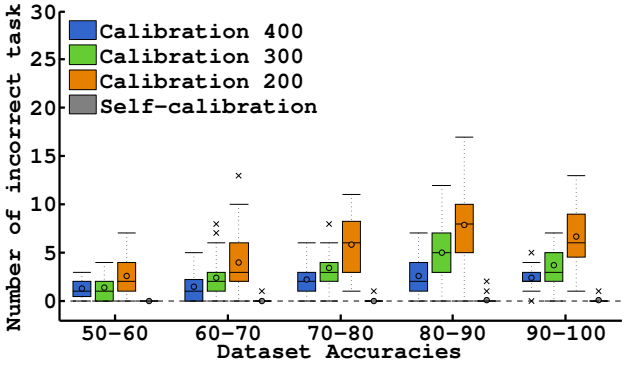


Figure 6.8: Number of tasks incorrectly achieved in 500 steps with EEG data. Calibration based methods, which do not update their models, make more errors.

6.2.6 Last 100 iterations performances

Figure 6.9 compares the number of tasks achieved during the last 100 steps with EEG data. During the last 100 steps, all methods are active at their full potential because no time is lost in calibrating the system. With dataset in the range of 80-90%, all methods achieve an average success rate of one task every 20 steps. However calibration based methods, which do not update their classifiers once calibrated, make more mistakes (see figure 6.10). While our method achieve slightly less task during the last 100 steps, it makes less mistakes, which seems to indicate our method is more conservative. We will discuss that point in section 6.3.

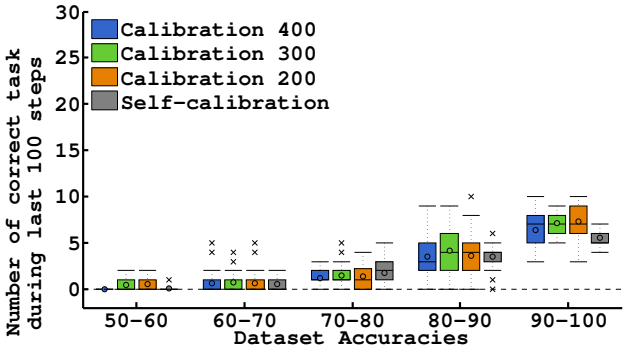


Figure 6.9: Number of tasks correctly achieved during the last 100 steps using EEG data. All methods allow the agent to complete an equivalent of tasks.

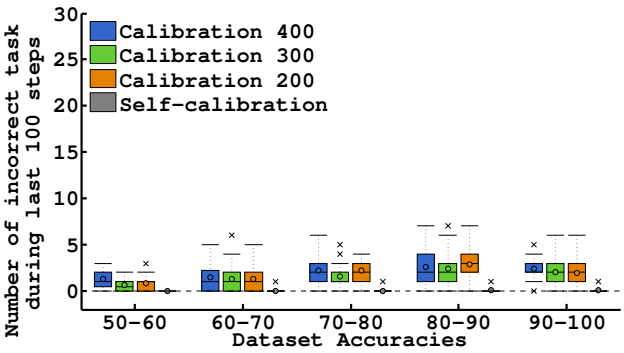


Figure 6.10: Number of tasks incorrectly achieved during the last 100 steps using EEG data. Calibration based methods, which do not update their classifiers once calibrated, make more errors.

6.3 Why are we cheating with pre-recorder EEG samples?

For this BCI scenario, we can identify two main differences between our self-calibration method and the usual calibration based approaches:

1. **Online update of multiple classifiers.** Our method integrates new data at each new step, and classifiers can differ between task hypotheses. For incorrect task hypothesis, the signal-label pair added to the training datasets can be incorrect and decrease the performance of the associated classifier. This dynamic can be observed in figure 6.4, where classifiers associated to incorrect tasks (in blue) have lower estimated accuracy than the classifiers associated to the correct task (in red). As a result, our algorithm makes different predictions and updates for each hypothesis.

2. **Positive/Negative percent ratio of training examples**. Following the literature [Chavarriaga 2010, Iturrate 2013b], the training dataset for calibration based methods was composed of 80 percent of the signals of meaning "correct", and only 20 percent of "incorrect". The ratio obtained during the self-calibration experiments is more balanced (around 50/50, see Table 6.1), resulting in classifiers with better properties. But, during online real world experiments, a 50/50 percent ratio may lead to practical problems and should be studied in more details.

This latter aspect concerning the positive/negative ratio of training example is usually required due to the properties of the signal we seek for in the subjects' brain. Indeed ErrP signals are more powerful when triggered by non-expected movement of the agent, rather than being a voluntary erroneous assessment. In other words, for the ErrP signal to be observable and of good intensity, the user should not expect

Dataset Accuracies	Self-calibration	Calibration
50-60	0.48 (0.02)	0.8 (0)
60-70	0.50 (0.03)	0.8 (0)
70-80	0.53 (0.03)	0.8 (0)
80-90	0.57 (0.03)	0.8 (0)
90-100	0.59 (0.01)	0.8 (0)

Table 6.1: Mean ratio of positive examples in the training datasets (standard deviation shown in parenthesis). Calibration procedures for creating a usable dataset of ErrP signals usually account for an 80 percent ratio of positive examples. However, when the task is unknown, it is impossible to guarantee such ratio. In practice, using our self-calibration method, an agent will collect as many positive than negative examples. This is likely to impact the quality of the ErrP signals received from the human brain during online experiments.

the agent to make a mistake. This explains why, during the calibration phase of their system, researchers uses 80 percent of the time a correct action (i.e. moving towards the goal), and only 20 percent of the time an incorrect action, which is therefore unexpected [Chavarriaga 2014]. This is possible during a calibration period because both the experiment informs both the user the agent of the task to consider. As the agent knows the task, it can plan its action to maintain an 80/20 percent meaning ratio.

However, in our learning scenario, the agent is not aware of the task the user has in mind. Therefore it is impossible to guarantee an 80/20 percent ratio of positive/negative examples. In practice, using our approach, the agent acquires as many signals of meaning "correct" as of meaning "incorrect" according to the true intended task, see Table 6.1.

At a glance, our observation of the ratio of positive/negative signals can explain the results of Figure 6.6, Figure 6.8, and Figure 6.10, where the calibration based methods, while using the same update equation, make more mistakes than our self-calibration method. Apart from the fact that our method trains one classifier per class, calibrating using 400 examples should produce similar results than our self-calibration approach. But after 400 steps, the calibration based method only observed 80 signals corresponding to the "incorrect" class, while the self-calibration method observed 200 signals. As the signals are represented in a 34 dimensional space, 80 samples may not be enough to build a good model, especially for low quality datasets.

Figure 6.11 shows the difference between the perceived accuracy of the classifiers (i.e. when estimating their quality on their training data) versus the actual accuracy of the classifiers (i.e. when estimating their quality on the remaining data in our bigger dataset). For dataset of good quality, our method generates classifiers that are under-confident while the calibration method tends to produce over-confident classifiers. This over-confidence is likely to explain the higher number of estimation

errors when relying on a calibration procedure, versus the very low error rate observed with our self-calibration method which tend to under-estimate the quality of its classifiers.

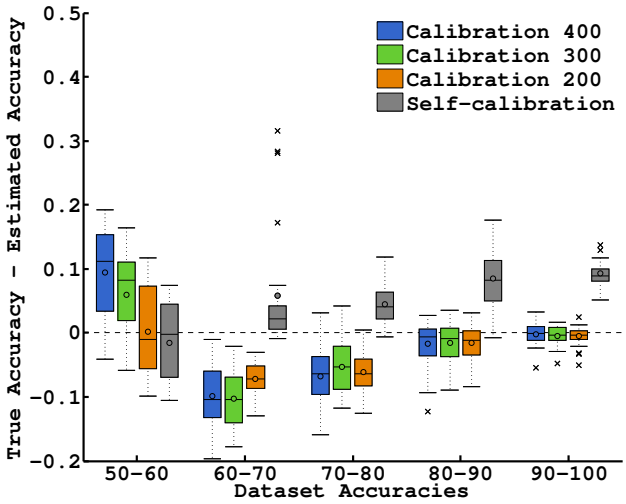


Figure 6.11: Difference between "true accuracy" and estimated accuracy. "True accuracy" is the performance of the classifier on the unused data. Estimated accuracy is the 10 fold cross validation performance of the classifier on the training data. A negative(positive) value indicates the classifier is over(under)-estimating its performance. Calibration methods tend to produce over-confident classifiers, certainly due to the biased positive to negative training example ratio, see table 6.1.

While this conclusion seems satisfying and is likely to explain the observation made in the previous section, we remind here that the data used in our simulated experiments were collected using an 80/20 percent ratio between the correct and incorrect signal samples. Will the brain signals conserve similar properties when using our self-calibration method online? This is unlikely due to the 50/50 percent ratio of signals' meaning observed using our method.

The work of Chavarriaga et al. and Iturrate et al. shows that high variability in the teaching signals properties are observed when varying the task and the teaching protocol [Chavarriaga 2010, Iturrate 2013b]. Indeed ErrP signals are elicited more from non-expected movements of the agent than they are voluntary erroneous assessment. In other words, for the ErrP signal to be observable and of good intensity, the user should not expect the agent to make a mistake. With a 50/50 percent ratio, the subject is unlikely to be surprised and may produce signal of lower quality.

During our first experiment with real subjects, we observed that the quality of the received data were very poor, even for subjects that were highly trained to the brain assessment task. The main cause was that the behavior of the agent was very confusing with respect to the goal state. The agent seems to act randomly, without trying to move toward the target. Therefore subjects had a lot of difficulty to generated ErrP signals of good quality, which makes the process longer, do not

improve the behavior of the agent and further reduces the engagement of the users in the teaching task. Studying in more details the impact of the agent behavior on the ErrP signals is not an objective of this thesis. We acknowledge that a thorough analysis is required to provide firmer conclusion.

Consequently, while some subjects succeeded in the teaching task, we observed a relatively high number of errors and long teaching time. Therefore, in order to improve the learning time and the behavior of the agent, we decided to include an a priori information in the system. This information relates to the difference in power (sum of the EEG feature squared) between EEG signals of meaning "correct" and "incorrect". The signals related to the unexpected, erroneous action, are, on average, more powerful. But this property alone is not enough to identify "correct" and "incorrect" signals. In next section, we study in more detail the power component of ErrP signals and present how it can be exploited in our system.

6.4 Including Prior Information

In this section, we detail how we can exploit the difference of power between positive and negative ErrP signals. Positive ErrP signals are slightly more powerful that negative ErrPs. Hence, measuring the power of a signal provides an absolute information about the meaning of a given signal. While this property is not enough to classify with good accuracy "correct" and "incorrect" signals, we will see that it allows to differentiate between symmetric hypothesis, which improves the performances of our algorithm as well as the perceived behavior of the agent at run time.

6.4.1 Difference of power between correct and incorrect signals

As shown in Figure 6.2, the EEG signals associated to "incorrect" labels have slightly more amplitude than the one associated to "correct" labels, especially around 300ms. To compute the power information contained in our signal we simply compute the sum of the square of each feature representing our signal. This simple approximation allows to capture the slight difference in power between "incorrect" and "correct" signals (see Figure 6.12). However this is not enough to classify a single signal with more than 60 percent accuracy. But considering a group of point we can observe that the mean of power of the "incorrect" class is higher that the mean power of the "correct" class. We will exploit this property as an a priori information of which group of point should means "correct" or "incorrect".

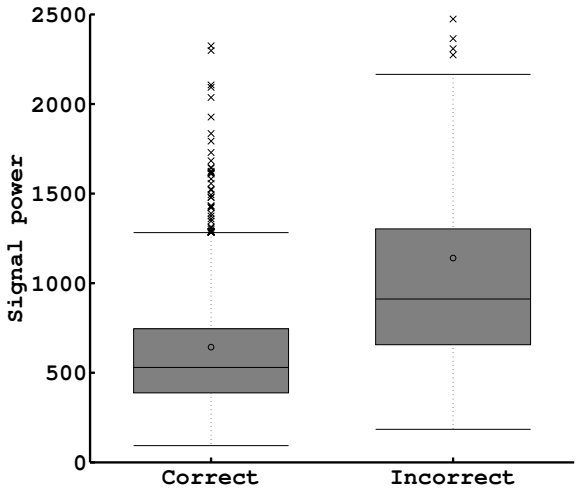


Figure 6.12: Box plot of the power of ErrP signals from one of our EEG datasets. A classifier trained on this dataset reaches a classification rate of 83 percent. The mean power information from the "incorrect" signals is higher than for the "correct" ones.

To compute the average power information from the signals having "correct" labels, we simply compute the weighted mean of the signals' power, with weights the probability that each signal being is of label "correct".

$$powerCorrect(\xi_t) = \frac{\sum_{i=1}^{M} p(l^c = \text{“}correct\text{”}|e_i, \theta) \;\; e_i^T e_i}{\sum_{i=1}^{M} p(l^c = \text{“}correct\text{”}|e_i, \theta)} \tag{6.2}$$

with θ representing the classifier trained on the available signal-label pairs associated to the task ξ_t.

Similarly, for the "incorrect" labels, we simply compute the weighted mean of the signals' power, with weights the probability that each signal is of label "incorrect".

$$powerIncorrect(\xi_t) = \frac{\sum_{i=1}^{M} p(l^c = \text{“}incorrect\text{”}|e_i, \theta) \;\; e_i^T e_i}{\sum_{i=1}^{M} p(l^c = \text{“}incorrect\text{”}|e_i, \theta)} \tag{6.3}$$

with θ representing the classifier trained on the available signal-label pairs associated to the task ξ_t.

For the dataset shown in Figure 6.12, $powerCorrect = 670$ and $powerIncorrect = 1031$. Note that this is different from the value shown by the gray circle in Figure 6.12. For our estimate we use the probability of each signal of being of one label as predicted by our classifier and not the probability from the training data.

Finally, we note that it is impossible to define an absolute threshold that differentiates between "correct" and "incorrect" signals (see Figure 6.12).

6.4.2 How to use the power information?

As for the case of known signals described in chapter 4.4.5, we define a specific likelihood function for the information provided by the power information, and combine it with the information from our initial algorithm. We define this likelihood as the ratio of the power associated to the "incorrect" class over the power of the "correct" class:

$$\mathcal{L}_{power}(\xi_t) = \frac{powerIncorrect(\xi_t)}{powerCorrect(\xi_t)} \tag{6.4}$$

For our previous example of Figure 6.12, this ratio is equal to 1.54. A ratio above 1 indicates that the labels are more likely to be correctly associated to the signals. Considering our algorithm that assigns different labels per task hypothesis and the specific case of symmetry as discussed in chapter 4.3.3. In such cases, two hypotheses have a symmetric labeling of the data, a cluster of signal considered as meaning "correct" for one hypothesis will be considered as being of meaning "incorrect" by the other hypothesis. And vice et versa. The power information breaks this symmetry. Indeed, the "correct" cluster should be more "powerful" that the "incorrect" cluster. In our above example, the correct hypothesis will have a power ratio of 1.54 as the label for "incorrect" would actually be associated to the "incorrect" signals. But for the symmetric case, while our non-informed algorithm could not make the difference, our new measure results in a power ratio of 0.65. Indeed, as the labels are switched, the power of class "correct" will be higher than the one from class "incorrect". Finally, considering a hypothesis whose labels are mixed, the power ratio will be around 1 because signals of low and high power will be equally distributed between "correct" and "incorrect" classes.

We note that, disambiguating between symmetric cases is likely to improve the perceived behavior of the agent, therefore likely to improve the quality of the signals receive from the subjects.

To include the task likelihoods computed as the ratio between the power component of each class, we simply multiply them with their respective likelihoods computed using our initial algorithm. The method is the same as described in chapter 4.4.5 when buttons of known meaning where available to the users.

It is of crucial importance to understand the use of the power information is only possible thanks to the specific nature of the ErrP EEG signals. It would be of not use, even misleading, for the previously considered artificial 2D datasets. However other signals may have similar properties, for example, when using speech, the tone of voice may differ between "correct" and "incorrect" feedback.

Before running online EEG experiments with human subjects, we verify how the power information method behaves using our pre-recorded datasets. In next section, we compare the performance between using only the power information, only our initial algorithm, or the combination of both.

6.4.3 Comparison with and without the power information

We consider the same grid world setting presented in previous sections (e.g. section 6.2) using pre-recorder EEG signals and considering a feedback frame.

Time to first task Figure 6.13 shows the number of iterations needed to reach the first target with confidence between our general method (matching), using the power information (power), or the combination of both methods (power matching). The use of the power information affects the performance for the low quality datasets (under 60 percent of accuracy). For datasets of low quality, while the time to first target seems more advantageous when using only the power information, most of the task estimations are erroneous (see Table 6.2), which makes the use of the power information critical for low quality data. However low quality datasets are not the main target of our algorithm. Indeed, for such data it would be better to change the representation of the brain signals or the classifier used. For datasets of higher quality (above 60 percent), the power information allows to speed up the learning compared to our initial algorithm (matching), which does not rely on known information.

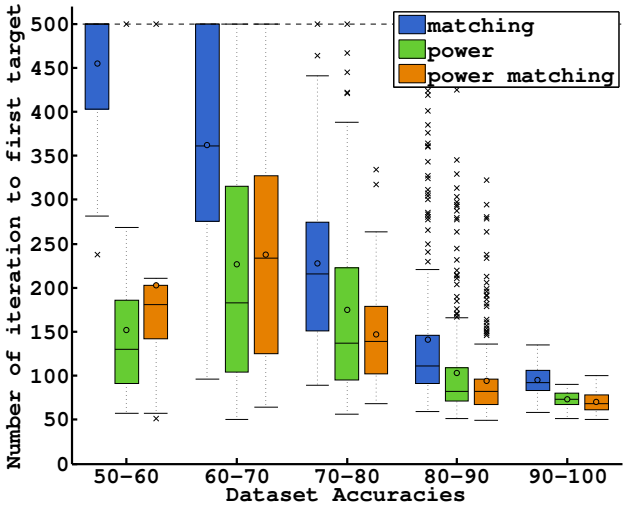


Figure 6.13: Number of steps to complete the first task using EEG data. Comparison between our general method (matching), or using the information that "incorrect" signals are more powerful than the "correct" signals (power), or both methods combined (power matching). The use of the power information affects the performance for the low quality dataset (under 60 percent of accuracy). For datasets of low quality, while the time to fist target seems more advantageous when using only the power information, most of the task estimations are erroneous (see Table 6.2), which makes the use of the power information critical for low quality data. For datasets of higher quality (above 60 percent), the power information allows to speed up the learning compared to our initial algorithm (matching), which do not rely on known information.

Dataset Accuracies	Matching	Power	Power-Matching
50-60	0	0.83	0.62
60-70	0	0.10	0.02
70-80	0	0.03	0.03
80-90	0	0.03	0.02
90-100	0	0	0

Table 6.2: Percentage of erroneous estimation of the first task using EEG data. Comparison between our general method (matching), or using the information that "incorrect" signals are more powerful than the "correct" signals (power), or both methods combined (power matching). For very low quality datasets (under 60 percent of accuracy), the power information increases the number of erroneous estimation.

Number of tasks achieved in 500 steps We compare the number of tasks correctly (Figure 6.14) and incorrectly (Figure 6.15) completed in 500 steps between our general method (matching), using the power information (power), or both methods combined (power matching). The power information makes more mistakes for low quality dataset, which also impacts the power matching method. However these errors occur for very low quality datasets only, which are not the main target of our algorithm. For signals above 60 percent of classification rate, the power information improves the number of tasks we can reach.

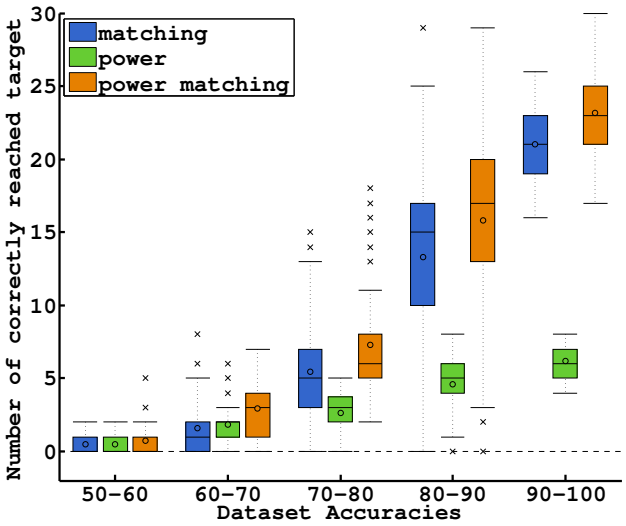


Figure 6.14: Number of tasks correctly achieved in 500 steps with EEG data. Comparison between our general method (matching), or using the information that "incorrect" signals are more powerful than the "correct" signals (power), or both method combined (power matching). The power information alone is sufficient to solve our problem but is less efficient than the other methods.

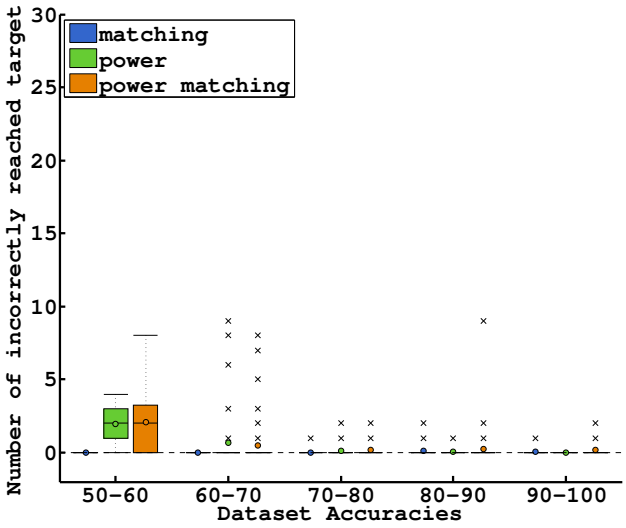


Figure 6.15: Number of tasks incorrectly achieved in 500 steps with EEG data. Comparison between our general method (matching), or using the information that "incorrect" signals are more powerful than the "correct" signals (power), or both method combined (power matching). The power information makes more mistakes for low quality dataset, which also impacts the power matching method. However these errors occur for very low quality datasets only, which are not the main target of our algorithm.

The power information alone is not enough to identify a high number of tasks, even if the number of steps to reach the first target is similar. The difference lies in the reallocation of labels, we performed after a task is identified. As described in chapter 4.4.4, once one task is identified with confidence, we propagate its labels to all other hypotheses. As a consequence, the number of new signals with different labels needed to pull apart two hypothesis in terms of power ratio increases. This problem arises because the power information is a global measure, which depends on averaged values over all collected observations. Our non-informed method (matching) classifies each new signal individually, which speeds up the learning process, especially when all hypotheses share a similar classifier (cf discussion of Figure 6.4).

The results presented in this section confirm that the use of the power information improves the performance and robustness of our algorithm. In addition, by disambiguating faster the task with symmetric properties, the perceived behavior of our agent should improve. We can therefore expect to receive ErrP signals of better quality during our online experiments. At the time of writing, our study was not terminated and this particular point requires a more a detailed analysis to quantify this difference if it exists.

6.5 Experiments with real users

We ran online experiments with 3 subjects. Each subject controls an agent in a virtual world. The setup of our online experiment is shown in Figure 6.1. Each subject was asked to mentally assess the agent's actions with respect to a given target. The system was not calibrated to decode the user EEG signals beforehand. Once the agent identified a task, and whatever the success or failure of the task identification, the user selected a new goal state randomly, the agent reseted the task likelihoods, propagates the believed labels, and teaching started again. At no point the agent has access to a measure of its performance, it could only refer to the unlabeled feedback signals from the user. There was an action every three seconds. Each experiment lasted 500 actions minimum, after 500 steps we kept running the system until a next task was reached.

Results

As depicted in Figure 6.16, our system was able to identify several tasks correctly. As for our simulated experiments, there are strong variations among subjects but we note that our system always identified the first task correctly (see Table 6.3). Importantly, the first task was always identified in less iterations than a normal calibration procedure requires (between 300 and 600 examples depending on the user performance [Chavarriaga 2010, Iturrate 2010]) (see Figure 6.17).

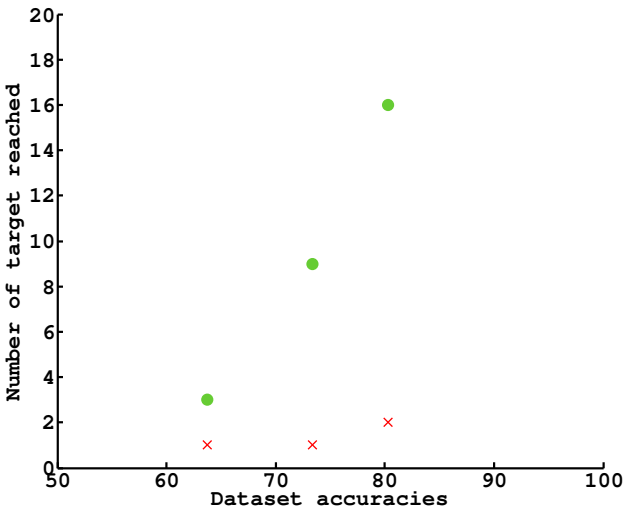


Figure 6.16: Number of tasks correctly (green dot) and incorrectly (red crosses) achieved in 500+ steps during our online experiments with real subjects. We kept running the experiments after 500 steps until the systems identified the next task. The results are plotted against the a posteriori computed 10-fold accuracy of our classifier on each subject EEG signals. The performance of the system is correlated with the quality of the EEG signals. These results matches well with the results from simulated experiments.

Subject	Class. rate	Steps to first task	First correct	N. correct	N. error
S1	80	101	Yes	16	2
S2	73	131	Yes	9	1
S3	64	265	Yes	3	1

Table 6.3: Results from our online experiments. For each subject, we provide the a posteriori computed classification rate of classifier on subject's brain signals (Class. rate), the number of steps needed to identify the first task, and whether or not the task identified was the correct one. Finally, we give the number of task that were correctly and incorrectly identified in 500 steps.

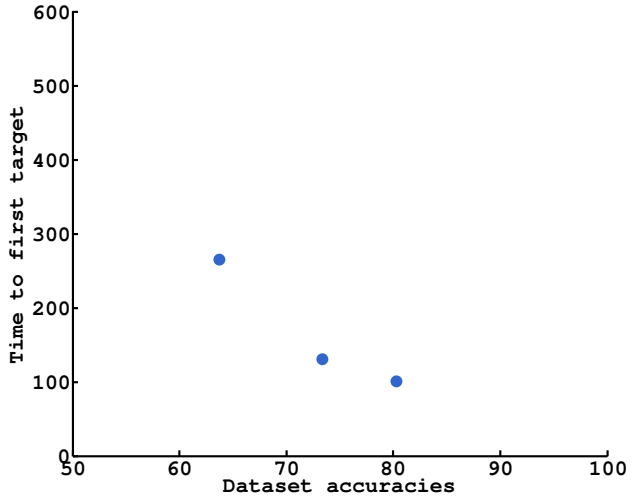


Figure 6.17: Number of steps to complete the first task for all subjects in our online experiments. The results are plotted against the a posteriori computed 10-fold accuracy of our classifier on each subject EEG signals. The relation between data quality and the time to first task is in line with our simulated results shown in Figure 6.13. Note that the first target was evaluated correctly for every subject.

6.6 Discussion

Results presented in this chapter with real EEG signals allow us to envision that the algorithm presented in this thesis could have practical applications in the real word. By removing the need of an expert to collect and calibrate the system, the use of brain computer interfaces may become more practical allowing their users to go out of the labs.

While this work offers a good solution to start interacting with machines without defining in advance the particular signals that will be used by the users, we have only demonstrated its performances on relatively simple scenarios. Especially, we considered discrete states and actions, synchronous protocol, and a finite set of task hypothesis. While these constraints have no impact on most BCI scenarios, they

are a more limiting factor for robotics experiments. In the next chapter, we address some of these limitations in simple experiments, which may provide ideas for the future developments of this work.

Chapter 7

Limitations, Extensions and Derivatives

Contents

In the previous chapters, we described an algorithm allowing a robot to learn the task desired by a user without defining in advance how the signals of the user maps with their meanings. We tested this algorithm on two domains, a pick and place scenario using speech commands, and a virtual cursor navigation task using EEG signals. We demonstrated the use of our system in real time and with real subjects using their brain to assess agent's actions with respect to a final desired position.

A number of assumptions and constraints were used. In this chapter we will detail important limitations, discuss the possibility to release them and provide small experiments to demonstrate our ideas.

In section 7.1 we compare the performance of Equation 4.3 and Equation 4.7 defined in chapter 4. We show that correcting classifiers' predictions given our knowledge about their confusion matrix (Equation 4.7) makes our algorithm more robust than relying on the raw classifiers' outputs (Equation 4.3).

In section 7.2, we present a small study on how different properties of the world impacts the efficiency of several planning methods. We specifically study the impact of the size and the maze like properties of our environment. This study will highlight the fact that our uncertainty based planning method allows to identify the correct task with best performance in several types of problems. However we will see that, by not considering the performance on the task itself during the exploration, for some problems our method lacks of efficiency with respect to solving the task as fast as possible.

In section 7.3 we introduce another method to identify the first task based on the overlap of signal models. We present online results with real subjects in a BCI scenario, and show the limitation of this new method to identifying a sequence of multiple tasks[1].

In section 7.4, we address the problem of continuous state space, and show that our method is not impacted by the continuous aspect of the problem. Indeed, as

Code for most experiments presented in this chapter is available online under the github account `https://github.com/jgrizou/` in the following repositories: lfui, experiments_thesis, and datasets.

[1] The work presented in section 7.3 has been published in [Grizou 2014b]. It is the result of a collaboration with Iñaki Iturrate and Luis Montesano.

our method only requires to known the optimal policy for each task, we can rely on any algorithm that computes a policy for continuous states given a pre-defined task.

In section 7.5, we release the assumption that a finite set of tasks is available and rely on a particle filter based method to dynamically update a finite subset of hypothesis. We show that sampling actively new tasks, as well as selecting actively the next visited states, significantly improves the final performance of our method.

In section 7.6, we release the assumption that the interaction frame is known and consider the agent has access to a finite number of hypothetic interaction frames. We illustrated this problem in a simple line world scenario. We present results from simulated experiments that demonstrate the ability of our method to not only learn the task and the signal to meaning mapping, but also the interaction protocol used by the teacher.

In section 7.7, we propose a minimalist proof for our algorithm. We spotlight the importance of understanding the properties of our algorithm, and to be able to have some certitude about its convergence and accuracy properties.

7.1 Why should we temperate classifiers' predictions

We compare the performance of Equation 4.3 and Equation 4.7 defined in chapter 4. The main difference between these two equations is that the second (Equation 4.7) is adding another layer of verification, we temperate the prediction of the classifiers given our knowledge about their quality, which we measure by computing the confusion matrix via a cross-validation procedure.

7.1.1 Artificial data

We consider the same setting as for the experiments described in chapter 6.2 and used our two dimensional datasets of different qualities as presented in chapter 5.4.2. We ran 500 simulations for each method. We consider only the planning method described in chapter 5.

Time to first task Figure 7.1 compares the number of iterations needed to reach the first task with confidence. We call the method using equation Equation 4.3 "simple matching" and we call "matching" the method using Equation 4.7 which corrects the classifiers' predictions. There are strong differences between our methods especially for low quality datasets. For extremely overlapping data (50/60% accuracy), the "matching" method is never confident about a task while the "simple matching" method show huge variability and sometime outputs confidence after very few time steps.

This over confidence of the "simple matching" method reflects in the number of first tasks that were erroneously identified. As shown in Table 7.1, the lower the quality of the data, the higher is the percentage of erroneously identified first task. For extremely overlapping data (50/60% accuracy), this percentage goes up to 20 percent. While the "matching" method may seems too conservative, it is particularly

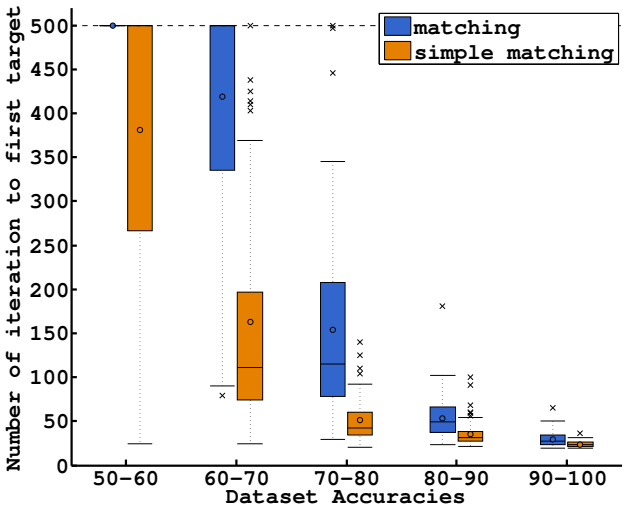


Figure 7.1: Number of steps to complete the first task using 2D artificial datasets. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix.

important to not make mistakes when estimating the first task. Indeed once a first task is identified, its associated labels are taken as ground truth. A false estimation of the first task will falsify the signal-label pairs for the remaining of the interaction.

Dataset Accuracies	Simple Matching	Matching
50-60	0.21	0
60-70	0.16	0
70-80	0.03	0
80-90	0.02	0
90-100	0.01	0

Table 7.1: Percentage of time the estimation of the first task was erroneous using 2D artificial datasets. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. Only the "matching" method, which temperates the predictions of the classifiers, does not make mistakes when estimating the first task.

Number of tasks achieved in 500 steps We compare the number of tasks correctly (Figure 7.2) and incorrectly (Figure 7.3) achieved in 500 steps between our two methods. While the "simple matching" method allows to reach more targets correctly, it also makes more mistakes for low quality datasets. The "matching" method is more conservative and does not make mistakes for all classifier quality, at the cost of reaching fewer targets.

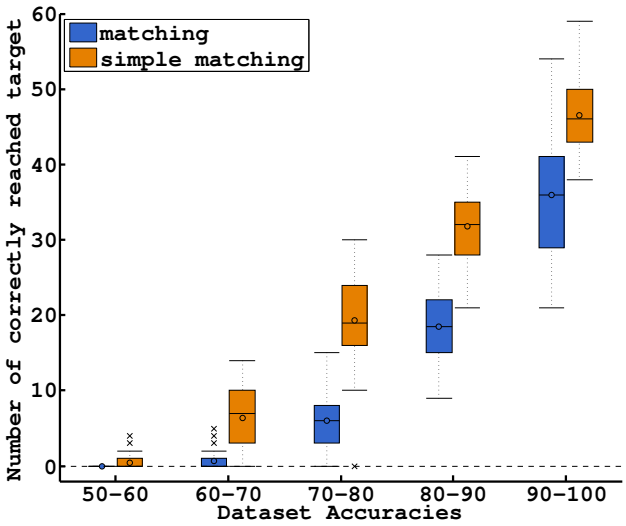


Figure 7.2: Number of tasks correctly achieved in 500 steps using 2 dimensional artificial data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. The "simple matching" method allows to reach more tasks correctly in 500 steps for all dataset quality.

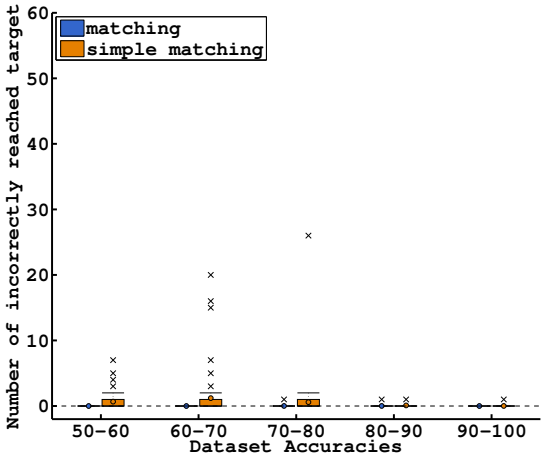


Figure 7.3: Number of tasks incorrectly achieved in 500 steps using 2 dimensional artificial data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. The "simple matching" method starts making errors for dataset with accuracies lower than 80 percent. However, the "matching" method is more conservative and does not make mistakes.

These results considered only low dimensional dataset (2D), which were generated from Gaussian distribution matching perfectly with the assumption made by our classifiers. We now investigate how the performances are affected by more complex signals, such as the EEG datasets used in chapter 6, which are 34 dimensional with data distributions that do not necessarily follow the Gaussian assumption.

7.1.2 EEG data

We consider the same setting as for the previous subsection and use the EEG datasets described in chapter 6. We ran 500 simulations for each method. We consider only the active planning method used in chapter 5.

Time to first task Figure 7.4 compares the number of iterations needed to reach the first task with confidence. There are strong differences between methods especially for low quality datasets. The "simple matching" method performances are not correlated with the classifiers' quality, which reflects the overconfidence of this method.

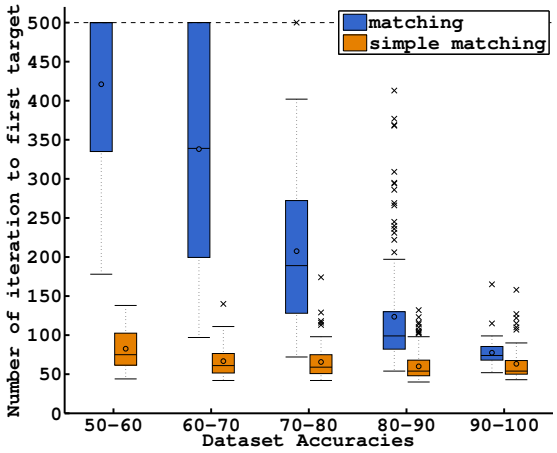


Figure 7.4: Number of steps to complete first task with our pre-recorded EEG data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. The "simple matching" method performances are not correlated with the classifiers' quality, which reflects the overconfidence of this method.

The over confidence of the "simple matching" method is reflected by the number of first tasks that were erroneously identified. As shown in Table 7.2, the lower the quality of the data, the higher the percentage of erroneously identified first tasks. In all cases, this percentage was above 50 percent which makes the use of the

"simple matching" method impossible for practical experiments. On the contrary, the "matching" method does not make any mistake when estimating the first task.

Dataset Accuracies	Simple Matching	Matching
50-60	0.81	0
60-70	0.80	0
70-80	0.66	0
80-90	0.53	0
90-100	0.60	0

Table 7.2: Percentage of time the first task estimation was erroneous using our pre-recorded EEG data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. Only the "matching" method, that temperates the predictions of the classifiers does not make mistakes when estimating the first task.

Number of tasks achieved in 500 steps We compare the number of task correctly (Figure 7.5) and incorrectly (Figure 7.6) reached in 500 steps. While the two methods allow to reach a similar number of targets correctly. The "simple matching" method also makes a many mistakes for all datasets. The "matching" method makes only few mistakes for all classifiers' quality.

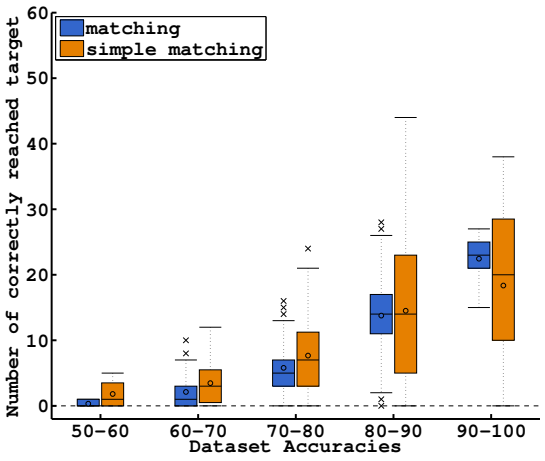


Figure 7.5: Number of tasks correctly achieved in 500 steps using our pre-recorded EEG data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. Both methods reach a similar number of targets correctly when using EEG datasets. The "simple matching" method shows more variability.

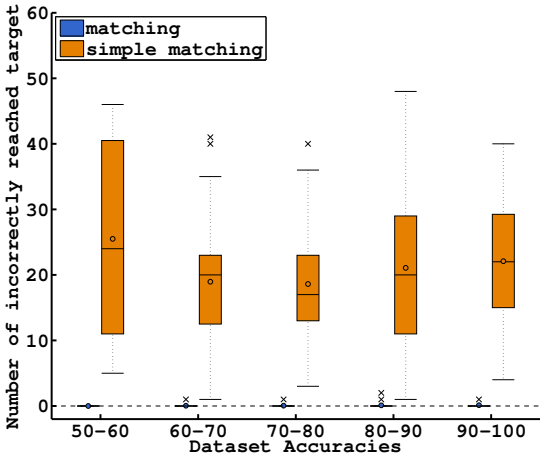


Figure 7.6: Number of tasks incorrectly achieved in 500 steps with our pre-recorded EEG data. Comparison between Equation 4.3 (simple matching) and Equation 4.7 (matching), where the latter corrects the predictions of the classifiers given the estimation of their confusion matrix. The "simple matching" is not reliable for EEG data.

7.1.3 Discussion

The results presented in this section confirm that taking into account the uncertainty about the predictions of the classifiers makes the algorithm more robust. However, if we knew the data will be of good enough quality, it is not necessary to correct the classifiers' outputs (as we did for the speech dataset used in chapter 4), which divides the computational cost by a factor of 10 (for a 10 fold cross-validation). However, as soon as we have to deal with signals of various qualities and with different properties (like for BCI data in chapter 6), it is better to include a measure of classification uncertainty in our likelihood update rule.

7.2 World properties

How the world properties (symmetries, size, ...) affect the learning properties?

As discussed in section 4.3.3, the properties of the world can affect the learning performances. For example some worlds have symmetric properties, which makes some tasks impossible to differentiate.

In this section, we compare how various planning methods perform on two different worlds, namely the pick and place scenario and the grid world. We investigate the performance of planning using a random strategy, several ε-greedy methods, a strategy based on the task uncertainty (where we do not take the signal to meaning mapping uncertainty in to account), and our uncertainty based method described in chapter 5. We will see that the size of the worlds and the properties of optimal policies impact the performance of these planning methods.

7.2.1 Hypothesis and world properties

We hypothesized that differences in the properties of each world will impact the performances of several planning methods, especially the random method and the e-greedy methods that are blind to the problem properties.

In the coming analysis, we consider three different world instances, a 5 by 5 grid world, a 25 by 25 grid world and the pick and place world of chapter 4. In the following we present the main differences between these worlds.

First testing our planning method on a 5x5 and 25x25 allows to test how the size of the world influences the performances and to verify that our uncertainty measure is robust to such change. The main hypothesis is that the random action selection method will not scale well to this change in dimensionality. Indeed, in a 5x5 grid, taking random actions allows to explore the state space quite uniformly in a small number of steps, however in a 25x25 grid (625 states) the robot is unlikely to visit useful states given a limited number of iterations.

We choose to use a 25x25 grid because the resulting number of states (625) is almost equal to the number of states of the pick and place scenario (624), which allows removing the size effects when comparing those two scenarios. By comparing the grid world and the pick and place scenario, we aim at investigating how the maze like properties of the pick and place world compares with the more simple structure of the grid world. For the pick and place scenario, to reach the correct cubes' configuration the robot must achieve a very specific sequence of action in the correct order. As for a maze, only one correct path can be followed, however for the grid world a multitude of paths can be chosen.

7.2.2 Method

We used the same conditions as used in chapter 5, where the teacher is providing instructions following the feedback frame but we use only two dimensional signals of very good quality (i.e. between 90 and 100 percent of classification rate).

We simulated 50 runs of 100 iterations for each planning methods and each world considered. There were 10 steps of initialization before the agent starts computing the first likelihood. During the first 10 steps, the agent was acting randomly for all methods.

7.2.3 Results

In this subsection, we analyze the Figure 7.7 which displays the number of iterations needed to reach the first task with confidence. We first comment the difference between the 5x5 grid world and the 25x25 grid world, and then compare the grid world and the pick and place scenario.

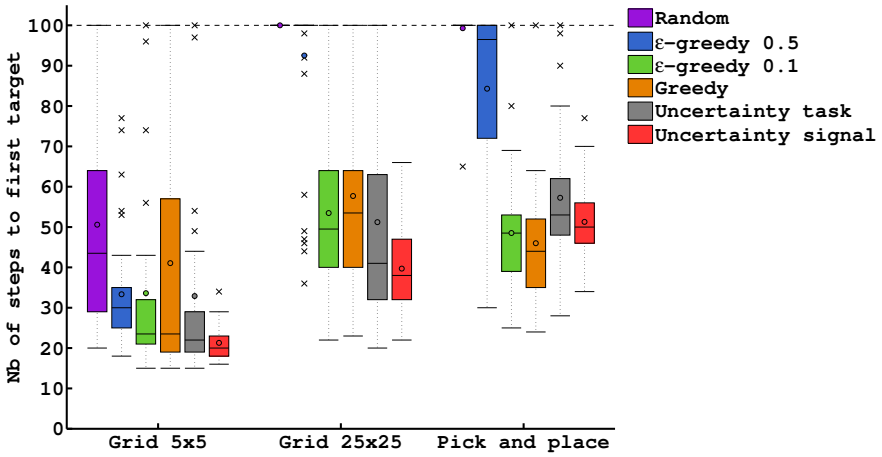


Figure 7.7: Number of steps needed to reach the first target state with confidence. When the dimensionality of the world increase, selecting actions randomly does not allow to identify any task in 100 iterations. Our uncertainty based method (uncertainty signal), is the most efficient at reaching the first task in the grid world scenarios but seems outperformed by a simple greedy approach in the pick and place scenario.

There are several aspects to keep in mind when analyzing Figure 7.7. First, it displays the number of steps needed to reach the target state while being confident this state is the correct one. But the agent can become confident one task is the correct one while being in a state "far away" from the target state. This fact will play an important role in the following discussion.

Also, when a method was not able to reach a task with confidence in 100 steps we considered a value of 100. This is very optimistic, for example the random method

is likely to need more than 100 steps for worlds with many states. We report the number of runs than reached a first target in less than 100 iterations in Table 7.3. These results indicate that only our uncertainty based method was able to always identify a task in less than 100 steps.

Planning methods	Gridworld 5x5	Gridworld 25x25	Pick and place
Random	47	0	1
ε-greedy 0.5	50	13	27
ε-greedy 0.1	46	48	48
Greedy	41	43	47
Uncertainty task	45	42	48
Uncertainty signal	50	50	50

Table 7.3: Number of experiments where the agent reached at least one target with confidence in 100 steps.

Finally, our plots include correctly and wrongly identified first targets, but only a handful of tasks were incorrectly identified. We report only 12 erroneous first task estimations across all 900 runs of our experiments and conditions. For the 5x5 grid world, 1 error for the random method, 1 for "uncertainty task" and 1 for "uncertainty signal". For the 25x25 grid world, 1 error for the greedy method. For the pick and place scenario, 1 for ε-greedy 0.5, 2 for ε-greedy 0.1, 2 for greedy, 1 for "uncertainty task" and 2 for "uncertainty signal".

World size effects As expected selecting actions randomly fails at identifying a task when the state space grows. The first obvious observation is that all methods require more iterations when the size of the world increased. In a 5x5 grid world, a random strategy allows to visit a good percentage of the states that makes it probable that the agent collected useful evidences. However, in a bigger world, it is important to target useful states.

We note that in our results of chapter 5 Figure 5.20, the greedy method performed worst than random. The only difference lies in the dimensionality of the dataset. In the experiment of this section, the signal are 2 dimensional and of good quality, in addition, the agent starts by 10 random movements before starting updating likelihoods. Therefore, after 10 steps, the agent has already enough data to build a good model. In the experiments of chapter 5 Figure 5.20, the agent used 30 dimensional data and performed 42 steps of random initialization, which may explains the difference observed. The effect of the dimensionality and quality of the datasets remains to be investigated in more details.

Maze properties effects When comparing the performance on the grid world versus the pick and place world on Figure 7.7, we observe that our uncertainty based planning method is not the most efficient method in the pick place word, and a very simple method such as acting greedily performs better. This result is in line

with the results from chapter 4 Figure 4.31 (left), where after 100 steps most of the experiments identified the correct task after 100 steps using a greedy planning method.

Potential users of our system will be interested by the time the agent takes to understand their instruction and fulfill the task. However none of the planning methods considered are taking this objective into account. Obviously the random or greedy methods are not following any specific goal, while the uncertainty based methods only try to differentiate hypothesis, not to reach the goal state. This is why we switch to a pure exploitation of the task once the confidence level is reached.

Therefore it may be more relevant to look at the time needed to reach the confidence level for the first task, which is displayed in Figure 7.8. Interestingly, our uncertainty method is faster at identifying the task than the greedy method.

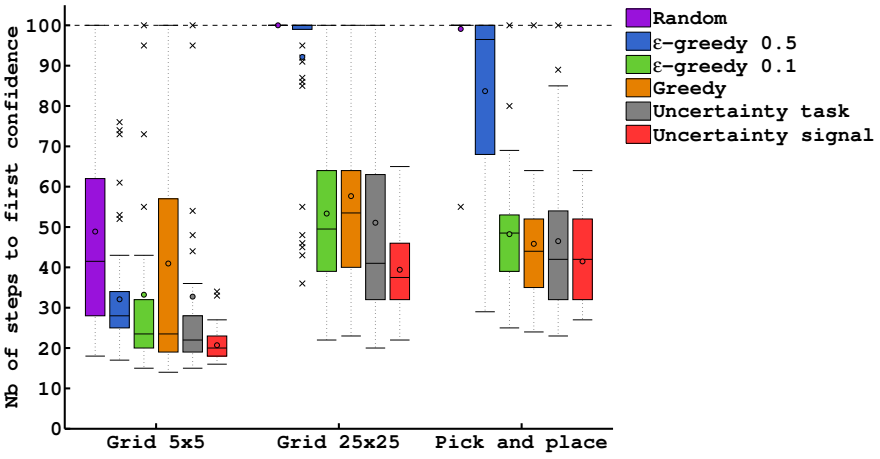


Figure 7.8: Number of steps to reach confidence level for the first target.

Figure 7.9 shows the number of actions needed for the agent to reach the goal state once the task is identified with confidence. This plot only considers the runs where a target was reached (see Table 7.3). For the grid world, all planning methods identify the task less than 5 steps away from its associated goal state. However for the pick and place problem, by following our uncertainty based planning method the agent is on average 10 steps away from the goal state when it identifies a task with confidence.

We hypothesized that, given the maze like properties of the pick and place problem, our agent would need to go toward the best hypothesized target states to differentiate between them faster; and therefore that our uncertainty planning method may be more efficient in such case. This hypothesis is not confirmed and requires more investigation on what properties are actually influencing the efficiency of our algorithm and what additional metrics should be considered to improve our strategies. Indeed none of the method presented are considering their performance on the task (yet unknown but estimated) in the action selection process.

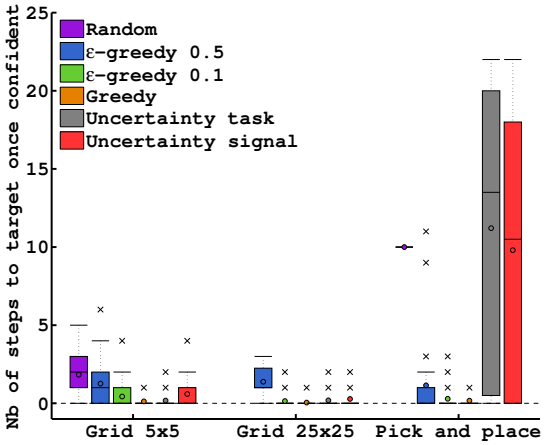


Figure 7.9: Number of actions needed to reach the first target once the agent reached confidence level for this target. This plot only considers the runs where a target was reached in less than 100 steps (see Table 7.3). Random action selection for the 25x25 grid world is not represented as it never reached any, and random for the pick and place only considers one run.

7.2.4 Discussion

The main conclusion of this study is that we do not understand well the impact of worlds and datasets properties on the final performance of the system. Many of these properties are tightly linked together and the additional layer of uncertainty inherent to our problem makes the dependencies hard to identify.

However one important aspect highlighted by the study is that our uncertainty measure should be combined with other metrics to optimize additional criteria on the task. Our measure was developed to discriminate faster the correct hypothesis from the set of possible tasks and not to also execute that task as fast as possible.

On this basis, we propose two different types of scenario to investigate:

- **Target based scenarios:** In these scenarios, the goal of the agent is to execute one specific action in a particular state, but in situation where failing the task have bad consequences. Lets consider a robot that should identify one object among a finite set and put it to the bin for a human. The robot can navigate freely around the objects in order to collect feedbacks from the human. However, the robot should only grasp and throw an object once it is confident that it is the object intended by the human. This problem is an instantiation of the visual navigation task used in our BCI experiments. In chapter 7.2, we have seen that our uncertainty method can be outperformed by a simpler method (greedy) when the goal is to identify and perform the task as fast as possible. It is likely that a pure greedy method can be outperformed. The problem with our uncertainty measure was that the robot could disam-

biguate between tasks "far away" from their respective goal states. Requiring additional steps to reach the correct goal state once identified. A potential avenue is to merge our measure of uncertainty with information about the optimal policy of each task, such that, for two states of equal uncertainty, the state closer to the potential targets is preferred. The resulting problem lies in weighting between seeking for uncertainty reduction and optimizing the position of the agent with respect to the, yet unknown, goal state.

- **Reward maximization scenarios:** In these scenarios, the goal of the robot is to maximize the cumulative reward associated to the correct task. The problem is that many tasks may have similar reward functions. Therefore it is not always necessary to identify the correct task with confidence to collect maximal rewards. For example, in our puddle word scenario of section 7.4, two tasks may share the same goal area but have different areas to avoid. If the robot can reach the shared goal area by avoiding the negative areas of both hypotheses, then the agent will have maximized the collected reward without ever knowing what specific task the user had in mind. In such cases, the agent must known whether merging two reward functions is more optimal than trying to differentiate between them.

7.3 Exploiting overlap between distributions

In this section, we propose a different approach to exploit the interpretation hypothesis process. We consider the same scenario as for our BCI experiments of chapter 6. This new method exploits the overlap between the signal models for each class to identify the correct task hypothesis. We present simulated experiments using pre-recorded EEG signals, and show that we achieve similar performances than calibration based systems. Finally, we report online experiments where four users control, by means of a BCI, an agent on a virtual world to reach a target without any previous calibration process.

7.3.1 Using the Bhattacharyya coefficient

Following [Lotte 2007, Blankertz 2010], we model the EEG signals using independent multivariate normal distributions for each class ($\mathcal{N}(\mu_c, \Sigma_c)$ and $\mathcal{N}(\mu_w, \Sigma_w)$). We will denote by θ this set of parameters $\{\mu_c, \Sigma_c, \mu_w, \Sigma_w\}$.

We propose to exploit the fact that when labels are mixed, the Gaussian corresponding to each classes should overlap more than for the correct label association (see Figure 4.16). The Bhattacharyya coefficient measures this overlap, it has been related to the classification error of Gaussian models [Kailath 1967] and is inversely proportional to the classification rate. Although there is no analytical relation between the coefficient and the classification rate, it is possible to derive bounds and good empirical approximations [Lee 2000].

The Bhattacharyya coefficient $\rho \in [0, 1]$ between the Gaussian distributions associated to label "correct" ($\mathcal{N}(\mu_c, \Sigma_c)$) and "incorrect" ($\mathcal{N}(\mu_w, \Sigma_w)$) is:

$$\rho = e^{-D_B(\theta)} \tag{7.1}$$

where D_B is the Bhattacharyya distance:

$$D_B(\theta) = \frac{1}{8}(\mu_c - \mu_w)^T(\frac{\Sigma_c + \Sigma_w}{2})^{-1}(\mu_c - \mu_w) + \frac{1}{2}ln\left(\frac{det(\frac{\Sigma_c+\Sigma_w}{2})}{\sqrt{det\Sigma_c det\Sigma_w}}\right) \tag{7.2}$$

Finally, we approximate the expected classification rate as:

$$Ecr \propto 1 - \rho \tag{7.3}$$

Now that we have an estimation of the expected classification rate, which is proportional to the overlap between the model of each class, we need to take a decision with respect to which task is the one intended by the user. To do so we compare the expected classification rate of every task hypothesis ξ_t with $t \in \{1, \ldots, T\}$.

The hypothesis whose associated model overlaps the less, i.e. which has the highest expected classification rate, i.e. the lowest value of ρ, is expected to be

the one intended by the user. However it is meaningless to define an absolute threshold on the value of the expected classification rate itself. Indeed, different people generate different signals, which results in classifiers of different qualities. Also, even for the correct signal-label pairs, the model may overlap by quite some amount, as illustrated in our 2 dimensional examples in Figure 5.18. To bypass this problem we rely on a voting system where we attribute to each hypothesis ξ_t a weight that is updated at every iteration.

We rely on a pseudo-likelihood metric that for each hypothesis ξ_t accumulates the expected classification rate over time:

$$\mathcal{L}(\xi_t) = \prod_{i=1}^{M} 1 - \rho_i^{\xi_t} \tag{7.4}$$

with M the current number of iteration and $\rho_i^{\xi_t}$ the Bhattacharyya coefficient associated to task ξ_t using all data up to time i. By normalizing the pseudo-likelihood values between every hypothesis, we obtain what can be viewed as the probability of each target:

$$p(\xi_t) = \frac{\mathcal{L}(\xi_t)}{\sum_{u \in \{1,\dots,T\}} \mathcal{L}(\xi_u)} \tag{7.5}$$

Once a target reaches a probability threshold β we consider it is the correct one, i.e. the one intended by the user. We used $\beta = 0.99$.

Finally, once we identified the first target, we will switch back to a classification based algorithm as described in chapter 4.4.4 and as used in the previous chapters of this thesis. We will see in section 7.3.3 that this switch is necessary to maintain good performances since the classifier makes a much harder decision for each new EEG signal.

7.3.2 Planning

As we are using a model based method, we rely on our uncertainty measure that directly acts in the signal space. This method was described in chapter 5.3.2. To summarize it is based on computing, for every state-action pairs, the similarity between the expected signals for each task. The more the expected signals are similar the less there is uncertainty.

For computing the similarity between two Gaussian distributions we could rely again on the Bhattacharyya coefficient describe above. However computing this coefficient between all models and for all state-action pairs was not feasible in real time. In order to improve computation efficiency we do not rely on a precise metric between Gaussian distributions and only consider the similarity between their means. The closest the means are, the more similar they are.

7.3.3 Offline analysis

The objective of the offline analysis is to study the impact of our uncertainty based planning method and to evaluate if the classifier learned from scratch with our algorithm can be reused for learning new tasks. To ensure we have sufficient data to achieve statistically significant results, we rely on a large dataset of real EEG data. We used the same dataset as described in chapter 6.2 from [Iturrate 2013b], which covers ten subjects that performed two different control problems.

For each subject, we simulated 20 runs of 400 iterations following the control task. Each time the device performed an action, we sampled the dataset using the ground truth labels corresponding to the correct task and then removed the chosen signal from it. After a first task was identified we continued running the system to identify new tasks.

We present most of the results in terms of the quality of the dataset, measured by the classification accuracy that a calibrated brain signal classifier would obtain.

Planning Methods We compared the average number of steps (with maximum values of 400 steps) needed to identify the first task when learning from scratch with different planning methods.

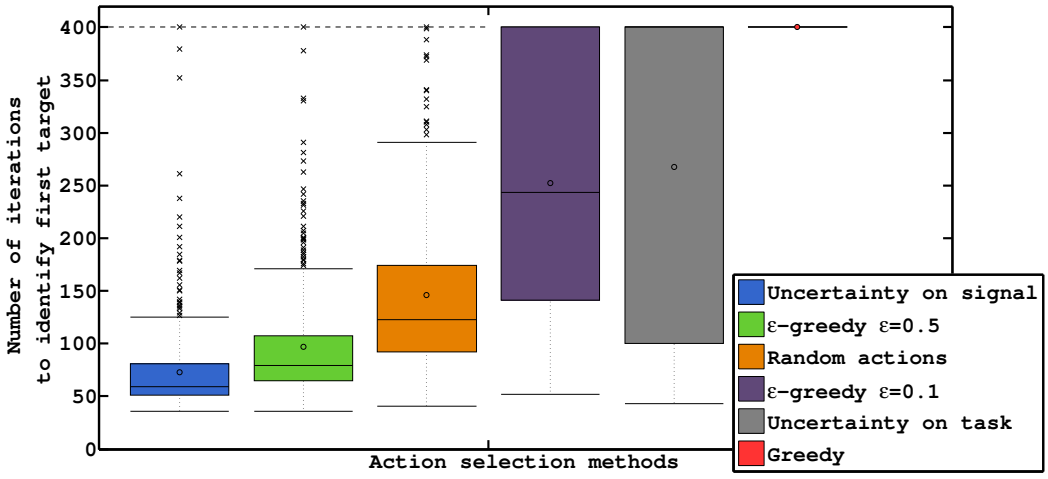


Figure 7.10: Comparison of different exploration methods. Our proposed method, based on the uncertainty on the expected signal, allows leading the system to regions that improve disambiguation among hypotheses in a faster way. For the greedy method, all values were 400 which indicates it never allowed to identify any task.

Figure 7.10 shows the results averaged across subjects, runs and datasets. A value of 400 means the confidence threshold was not reached after 400 iterations. Our proposed method, based on the uncertainty on the expected signal, allows leading the system to regions that improve disambiguation among hypotheses in a faster way. Trying to follow the most probable task does not allow the system to explore sufficiently (greedy), and at least some random exploration is necessary to

allow a correct identification of the task (ε-greedy). Assessing uncertainty only on the task performs poorly as it does not take into account the signal interpretation ambiguity inherent to our problem. The large variability in the results is mainly due to the large variations in classification accuracy across subjects and datasets. Given these results, the remainder of this section will only consider our proposed planning method.

Using the Bhattacharyya coefficient in the long run After identifying the first task, and following our approach, we continued running the system and measured how many tasks were identified after 400 steps. Figure 7.11 demonstrates the advantage of switching to a classification based method after identification of a first target instead of keeping the estimation given by the Bhattacharyya coefficient. On the one hand, Bhattacharyya coefficient works very well for small amounts of data because it directly compares model parameters. On the other hand, after identifying many tasks, all models share most of their signal-label pairs and it requires much more data to modify the models and detect overlaps. Therefore using a classifier allows for a faster identification since the classifier makes a much harder decision for each new EEG signal. This discussion is in line with the observation on the use of the power information made in chapter 6.4.

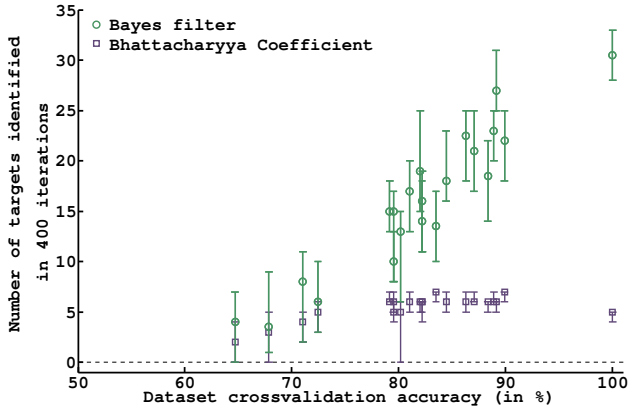


Figure 7.11: Number of targets correctly identified in 400 iterations (the markers show the median values and the error bars the 2.5th and 97.5th percentiles). Comparison between switching to a Bayes filter method after identification of a first target instead of keeping the estimation given by the Bhattacharyya coefficient. The classification based method allows for a faster identification.

Given these results, in the remaining of this section we only consider switching to a classification based method once the first task has been identified.

After 400 steps Figure 7.12 shows the number of tasks correctly and incorrectly identified in 400 iterations. For datasets of good qualities, we are able to identify more than 20 tasks in 400 iterations without the need for a calibration procedure

(recap that previous works needed between 300 and 600 examples for the calibration phase [Chavarriaga 2010, Iturrate 2010]). The number of correctly identified tasks is strongly correlated to the quality of the dataset.

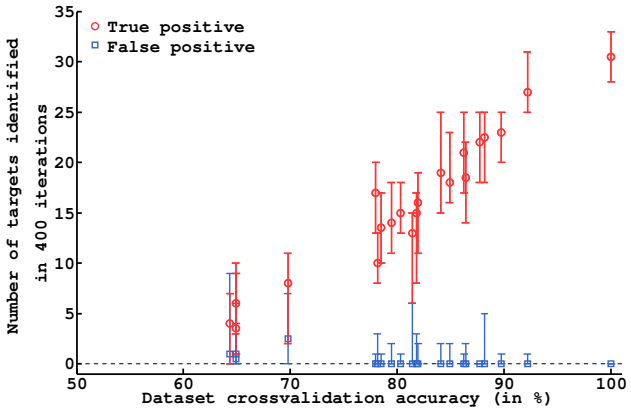


Figure 7.12: Number of targets correctly and incorrectly identified in 400 iterations (the markers show the median values and the error bars the 2.5th and 97.5th percentiles). For datasets of good qualities, we are able to identify more than 20 tasks in 400 iterations without the need for a calibration procedure.

The quality of our unsupervised method can be measured according to the percentage of labels correctly assigned (according to the ground truth label), see Figure 7.13. In general, having dataset with classification accuracies higher than 75% guaranteed that more than 90% of the labels were correctly assigned. This result shows that our algorithm can also be used to collect training data for calibrating any other state-of-the-art error-related potentials classifier, but has the important advantage of controlling the device at the same time.

7.3.4 Online control

The experiments were conducted with four subjects (aged between 25 and 28). Each subject was asked to mentally assess the agent's actions with respect to a given target. The system was not calibrated to decode the user EEG signals beforehand. Each subject performed 5 runs, for each run a new target was randomly selected and provided to the user. There was an action every three seconds. Each run lasted 200 actions, and the time between runs was around one minute.

The algorithm was able to identify the correct target for all runs of all the subjects, see Figure 7.14. There are strong variations among subjects but we note that our system identified each task in less iterations than a normal calibration phase requires (between 300 and 600 examples depending on the user performance [Chavarriaga 2010, Iturrate 2010]).

Table 7.4 shows for each subject and run the number of iterations needed to reach the confidence threshold for the subject selected target. On average, the number of iterations needed to identify the target was of 85 ± 32.

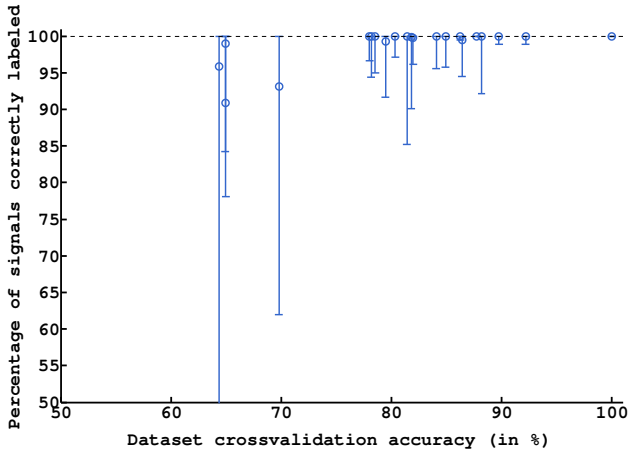


Figure 7.13: Percentage of labels correctly assigned according to the ground truth label (the markers show the median values and the error bars the 2.5th and 97.5th percentiles). In general, having dataset with classification accuracies higher than 75% guaranteed that more than 90% of the labels were correctly assigned.

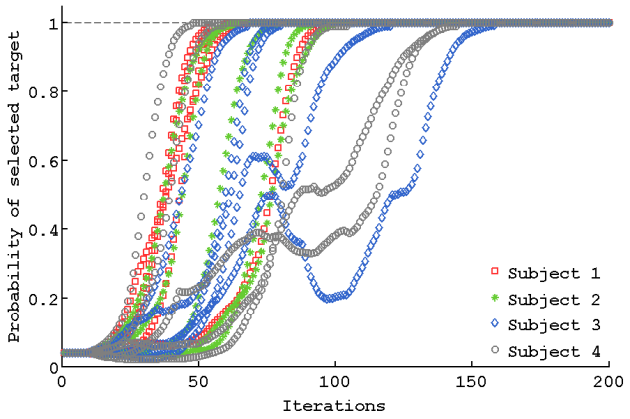


Figure 7.14: Results from the online experiments: Evolution of the probability of the correct task for each subject and run. The algorithm was able to identify the correct target for each subject and run in less than 200 iterations.

	Run1	**Run2**	**Run3**	**Run4**	**Run5**	**mean±std**
S1	95	62	56	60	64	67 ± 16
S2	89	77	98	60	62	77 ± 17
S3	68	80	118	76	157	100 ± 37
S4	98	142	57	142	47	97 ± 45

Table 7.4: Results from the online experiments: Number of iterations needed to identify the correct target for each subject and run. On average, the number of iterations needed to identify the target was of 85 ± 32.

7.3.5 Discussion

We introduced a new method to exploit the facts that, when associating hypothetic labels to all task hypotheses, only the correct task assigns the correct labels to the correct hypothesis. This method compares directly the overlap between the distributions modeling the generation of such signals. As for wrong hypothesis, the labels tend to be mixed with respect to the underlying structure of the data, the overlap between distributions is a good and stable measure.

However, we have seen that once all hypotheses share the same signal-label pairs, this method requires collecting more and more data to detect a change in the overlap of the wrong hypotheses. As a consequence the system should make use of two different sets of equations, one specific to the first target and one for the forthcoming targets.

This latter aspect shows the important advantage of the method we presented in the body of this thesis (chapter 4, 5, and 6), which uses the same equation from the first to the last iteration. This equation captures both phases of the interaction, where during a first phase the classifier qualities are playing a major role, and in a second phase the classifier predictions are taking the lead by taking more hard decisions.

7.4 Continuous state space

How to deal with continuous states?

As for now, our algorithm assumes the world can be represented by a limited number of discrete states. In this section we extend our algorithm to a continuous world, but still consider discrete actions. In addition, we present a new interaction frame that combines the feedback and guidance frames. We investigate how our algorithm scales to such problem and how different exploration strategies perform.

7.4.1 Experimental System

We consider a puddle world, in which an agent must reach a goal region while avoiding a penalty region. We consider a 2 dimensional puddle world with each dimension ranging between 0 and 1. The state of the agent can be any coordinate in the 2D world. Agent's actions are discrete and represent steps in the North, South, East, or West direction. One step length is sampled from a normal distribution of mean 0.1 and standard deviation 0.01.

As in the experiment of chapter 4, we consider speech as the modality for interacting with the robot and we reuse the dataset presented in section 4.5.4. The interaction between the agent and the teacher follows a turn taking social behavior, where the agent is performing an action and waits for a feedback or guidance signal to continue. We only consider a Gaussian classifier.

Task Representation To define the set of possible tasks we project a 5x5 regular grid on top of the continuous world. One task is represented by a +1 reward in one of the 25 projected squares and a -100 reward in three consecutive (vertically or horizontally) squares. +1 and -100 areas cannot overlap (see figure 7.16e for an example). The set of possible tasks is defined as all possible combinations of such reward function, for a total of 660 hypotheses.

Our algorithm only needs to have access to the optimal policies to be able to interpret a signal with respect to the feedback or guidance frame. We use the MDP framework to compute the corresponding policies. The world being continuous we use the tile coding function approximation [Sutton 1998], with 10 overlapping 50x50 regular grids. A Q-Learning algorithm [Watkins 1992] is used to compute the Q-Values, with a discount rate of 0.99 and a learning rate of 0.01. The optimal policies are then defined as greedy according to the Q-Values.

Mixed feedback and guidance frame In previous chapters, we considered only the feedback or the guidance frame separately. Such limitation can be restrictive for the user, we now consider the case where the teacher can use both feedback and guidance signal. We define as F as the set of meanings associated to the feedback

meanings (i.e. "correct" and "incorrect"), and G the set of meanings associated to the guidances meanings (i.e. "action 1", "action 2", ...). Extending our algorithm to cases where possible meanings include both feedback and guidance (i.e. $l^f \in \{F \cup G\}$) requires a probabilistic model of how the teacher distributes feedback and guidance signals. This model must hold the following property $\sum_{l \in \{F \cup G\}} p(l^f = l|s, a, \xi) = 1$. We define a variable ϕ that represents the probability of the user providing a feedback signal at each step, i.e. $p(l^f \in F) = \phi$, which implies $p(l^f \in G) = 1 - \phi$.

Under this new definition we can change our frame definition to:

$$p(l^f = l|s, a, \xi) = \begin{cases} \phi \; p(l^f = l|s, a, \xi) & \text{for } l \in F \\ (1 - \phi) \; p(l^f = l|s, \xi) & \text{for } l \in G \end{cases} \tag{7.6}$$

where Equation 4.13 holds for the feedback component (for $l \in F$) and Equation 4.14 holds for the guidance component (for $l \in G$).

We assume the mixing parameter ϕ is known in advance. We set ϕ to 0.5 meaning the user is providing feedback half of the time and guidance the other half of the time.

Exploration strategies We investigate four different agent behaviors: a) random, b) ε-greedy, c) myopic uncertainty based exploration, which aims at selecting the action that is the most uncertain in the current state, and d) full uncertainty based exploration which requires an uncertainty map to decide what to explore next.

As we are in a continuous domain, we cannot compute the full uncertainty for each state as presented in chapter 5, we therefore approximate this process. Extensions of the general problem already exist for the continuous state problem [Nouri 2010, Hester 2013] and we will rely on a sampling based method. One hundred random states are generated and evaluated in terms of their uncertainty. Each sampled state is associated to a reward value proportional to its uncertainty. This value is propagated to neighborhood states by using a fixed Gaussian kernel. We use as amplitude the uncertainty value and a diagonal covariance matrix of value 0.01 for each component. The resulting approximated uncertainty map is then used as a reward function. By solving the corresponding MDP, using for instance Q-Learning, the agent plans its actions to visit the most uncertain regions. We decided to use an ε-greedy policy on the Q-values. In the following experiment, the agent will use an exploration ratio ε equal to 0.1.

7.4.2 Results

We present results from 75 runs of our experiment, where for each run we randomly choose a task to teach from the set of hypotheses, as well as the initial state of the agent. The simulated teacher makes 10 percent of teaching mistakes, i.e. sending an erroneous signal 10 percent of the time. For each experiment, we compute the likelihoods every 15 steps and performs a total of 35 updates, for a total of 525

iterations. Figure 7.15 shows the average evolution of the taught task hypothesis likelihood.

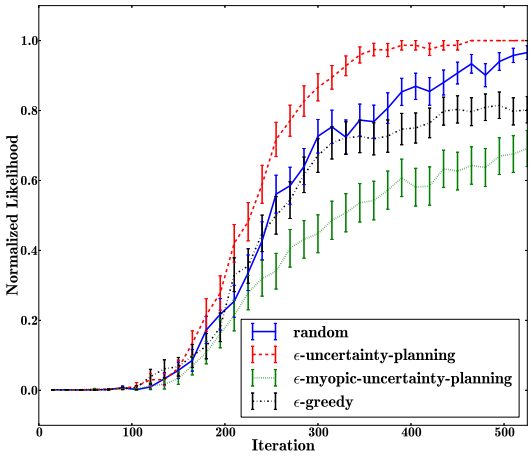
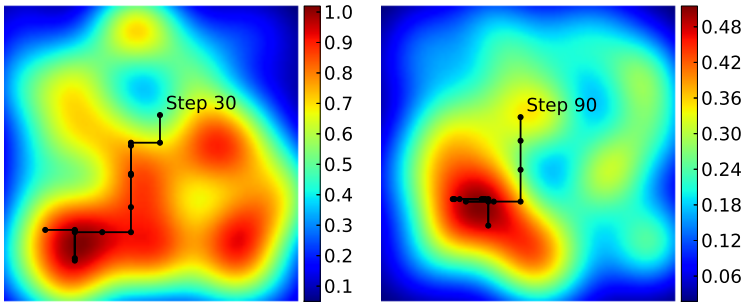


Figure 7.15: Taught hypothesis normalized likelihood evolution (mean + standard error) thought iteration using a Gaussian classifier. Comparison of different exploration strategies. Uncertainty based exploration method, which plan on the long term, performs significantly better on average.

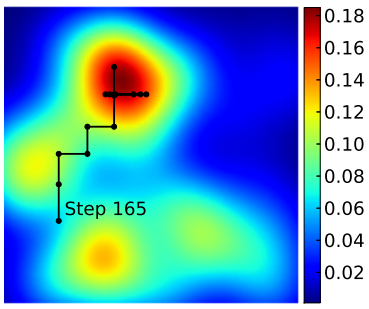
These results show that our algorithm can learn a task in a continuous world from unlabeled and noisy instructions whose possible meanings are both feedback and guidance and 10 percent of the instructions were teaching mistakes. The uncertainty based planning strategy outperforms random action selection. Interestingly, myopic uncertainty based strategy, which is also based on our uncertainty measure, is not efficient. This result illustrates that, when considering the agent as not being able to teleport, a long term planning approach is more suited to explore efficiently the state space than a short-term vision by selecting the next action with higher immediate reward, i.e. higher uncertainty.

Figure 7.16 shows the evolution of the estimated uncertainty map for one run of the experiment. For each uncertainty map, the agent plans its actions to reach a maximal uncertainty region. The maximum uncertainty value decreases as the agent is correctly estimating the task.

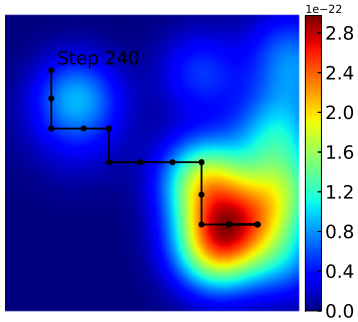


(a) After 30 iterations.

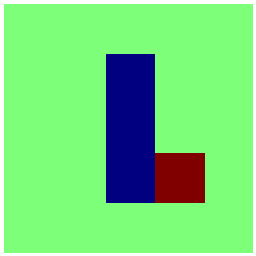
(b) After 90 iterations.



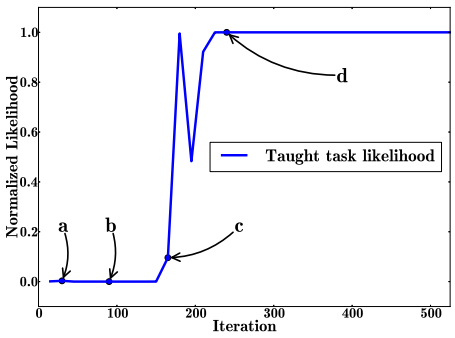
(c) After 165 iterations.



(d) After 240 iterations.



(e) Puddle world used by the teacher.



(f) Taught hypothesis normalized likelihood evolution.

Figure 7.16: Log Uncertainty maps after a) 30, b) 90, c) 165 and d) 240 iterations. e) shows the puddle world chosen by the teacher and f) shows the learning progress and the frame associated to each of the uncertainty map. In order to display the differences between log values, we bounded the color map between -5 and 0, which correspond to uncertainty values between 0.0067 and 1. Some log values, especially for d), are lower than -5 and are displayed in the same color as -5. Best shown in color.

7.4.3 Discussion

We have shown how our algorithm could be applied to continuous state domains and seen that, given the interaction frame considered, our algorithm only needs to have access to the optimal policies associated to each task to be able to interpret a signal. Therefore any method that allow to compute a policy for continuous domains could be used. The only problem is then related to the computational cost of those methods than to the formalism of this work. We will see in next section that, for some specific frames and worlds, it is not always needed for the robot to know the optimal policies to interpret the teaching signals from the human, which can considerably reduce the computational cost of running our algorithm.

7.5 Continuous set of hypothesis

How to relax the assumption that
the correct task belong to a known finite set of hypothesis?

In order to make the learning problem tractable, we assumed that the robot learner knows that the task to be learnt can be approximated by one task among a pre-defined set of tasks. Indeed, without constraining the space of possible tasks, an infinite number of tasks may explain the particular teaching data received. In practice, the number of pre-defined tasks in the experiment was still relatively large, allowing a certain level of flexibility. Yet, it would be highly desirable to extend the possibility to deal with continuous task representation, allowing potentially infinite task spaces.

A potential avenue to address this is to constrain search through a combination of regularization and particle filter approaches. In the following of this section, we present a simple particle filter based algorithm that allow an agent to identify a task from unlabeled instruction and considering an infinite set of hypothesis. The agent lives in a 2 dimensional continuous state space and should identify which coordinate it should reach, among the infinite number of possible coordinates.

7.5.1 World and task

We consider an agent living in a 2 dimensional continuous space bounded between 0 and 1 in both dimensions. A teacher is providing indication about the orientation of the goal state compared to the robot state by drawing some patterns on a tablet. Those directions can only be selected among of the four cardinal directions that are the directions of north, east, south, and west. The teacher wants the robot to reach a particular state that can be any position in the continuous 2 dimensional space. The robot is able to teleport itself to any location of the space to receive a new indication.

We still consider a strong a priori knowledge on the space of task, which is that there is only one goal state. This is a very strong a priori regularization on the complexity of the problem. Considering there could be several goal positions, depending for example on the current position of the agent, would increase dramatically the search space; it would then be likely that many hypothesis of different complexity would explain well the observed data. In such case a rule for regularizing the hypothesize task solutions would be needed.

7.5.2 Interaction frame

We define the cardinal direction frame. In this frame, the user provides information about the cardinal direction of the goal state with respect to the current agent position. The agent does not need to know the optimal policy to interpret a signal

but only its current state. The teacher provides indication on the absolute direction of the goal state with respect to the agent position. As an example, we consider a teacher that indicates the cardinal direction of the object, i.e. the message to the robot is: *"the object is North (South, West or East) with respect to your position"*.

We illustrate this frame in Figure 7.17. The choice of the cardinal direction to send to the agent is modeled by a probabilistic model, where the probability of one cardinal direction is proportional to the angle between the target-agent direction and the cardinal direction considered.

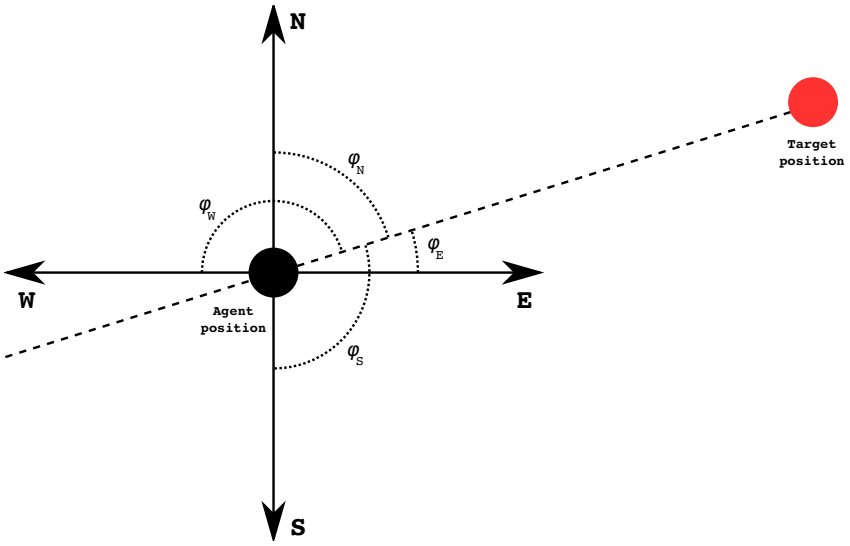


Figure 7.17: Example of the cardinal frame. The signal from the teacher indicates in which cardinal direction (N,S,W,E) is the target position. There is a probabilistic model that describes the user behavior, such that the probability of generating a signal of meaning "West" is proportional to the angle between the agent position and the target position. This frame does not requires the agent to know how to reach the target position, but only its own position with respect to that goal.

We defined as φ_N the angle between the target-agent direction and the North cardinal direction, and respectively φ_S, φ_W, and φ_E the angles with respect to the South, West, and East directions. The probability that the user refers to the North cardinal direction is defined as follows:

$$p(l^f = north \mid \varphi_N) = \begin{cases} (1 - \frac{2\varphi_N}{\pi})(1-\alpha) & if\ \varphi_N < \frac{\pi}{2} \\ \frac{\alpha}{K} & \text{otherwise} \end{cases} \tag{7.7}$$

with K the number of cardinal direction that do not satisfies the condition $\varphi_N < \frac{\pi}{2}$, which means K can take value of 2 or 3 only. α is the error rate of the user. Finally, we consider unsigned angles only within the $[0, \pi]$ intervals, meaning that angles of $\frac{-\pi}{2}$ or $\frac{3\pi}{2}$ are taken as $\frac{\pi}{2}$. The same equation applies for all cardinal direction and should maintain the following properties $\sum_{c \in \{N,S,E,W\}} p(l^f = c|\varphi_c) = 1$.

In practice, if we consider our visual representation of Figure 7.17, we obtain the following angle measurements: $\varphi_N = \frac{9\pi}{22}$, $\varphi_S = \frac{13\pi}{22}$, $\varphi_W = \frac{20\pi}{22}$, and $\varphi_E = \frac{\pi}{11}$. In that case $K = 2$. If we consider $\alpha = 0$, we obtain the following probabilities values: $p(l^f = north \mid \varphi_N) = 0.18$, $p(l^f = south \mid \varphi_S) = 0$, $p(l^f = west \mid \varphi_W) = 0$, and $p(l^f = east \mid \varphi_E) = 0.82$, which we represent as a vector $[0.18, 0, 0, 0.82]$. If we account of some probability of errors from the teacher, taking for example $\alpha = 0.05$, we obtain the following vector of probability: $[0.17, 0.025, 0.025, 0.78]$.

We will use this frame in our experiment with $\alpha = 0.01$. Note that the same frame can be used with different referential, instead of the cardinal direction, one could refer to the direction with respect the robot orientation, or with respect to the position of the human teacher in the room.

7.5.3 Finger movement's datasets

We will present results using two different datasets made of finger movements performed on a tablet.

Our first dataset shown in Figure 7.18 is build from a user generating directional trajectories starting from the center of the tablet and going toward the edges of the tablet. We considered four different movements, one toward each edge, representing the four cardinal directions that are the directions of north, east, south, and west.

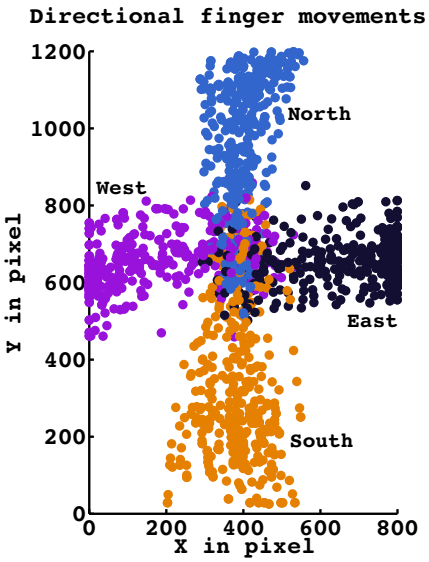


Figure 7.18: Dataset of finger movements for North/South/East/West commands. The user is sliding his finger from the middle of the screen to the corresponding edge of the screen.

Our second dataset shown in Figure 7.19 is build from a user drawing the cardinal letters (N, S, W, and E) in the middle of the tablet.

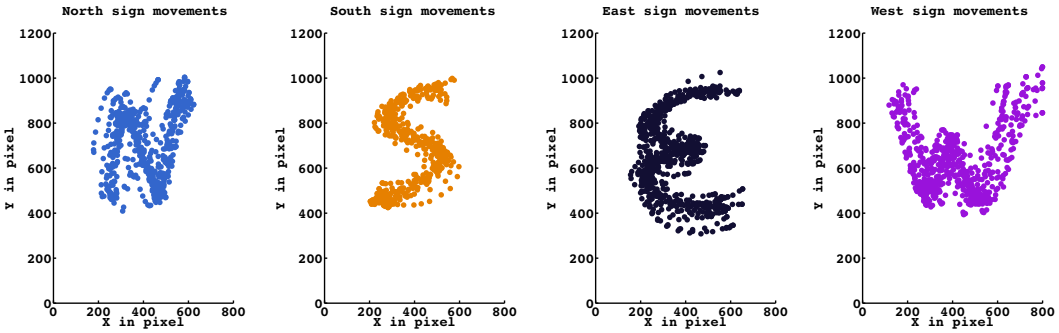


Figure 7.19: Dataset of finger movements for North/South/East/West commands. The user is drawing the first letter of the cardinal on the screen.

To represent those trajectories, our feature vector is composed of 11 dimensions, encoding:

- The start X and Y positions (2 features)
- The end X and Y positions (2 features)
- The delta position between start and end position for X and Y coordinate (2 features)
- The median X and Y positions (2 features)
- The distance between start and end position (1 feature)
- The total distance traveled by the finger (1 feature)
- The average speed of the finger (1 feature)

Using this representation we achieve 100 percent accuracy on the directional movements dataset and 99 percent accuracy on the cardinal signs dataset, using a simple Gaussian classifier with one Gaussian per class.

We remind that each movement has no a priori meaning for the robot. For example, in our simulation we may use the "W" sign signals to mean the goal state is north to the agent position.

7.5.4 Evaluating task likelihood

As there is an infinity of possible goal states, the agent cannot estimate the probability of all possible tasks in parallel. We will rely on a particle filter based approach [Gordon 1993, Doucet 2009, Thrun 2002]. The main idea consists of sampling a finite number of tasks and computing a confidence measure for each of those tasks. Given the ranking between them, we will apply a resample step that consists of

keeping some of the best tasks and sample new ones. More details are provided in next subsection 7.5.5, we present in this subsection how we estimate the probability of each sampled task.

Our algorithm, as presented so far, was cumulatively accumulating evidence for each task and was updating the likelihood of each task on a step by step basis. However, for this experiment, as the task hypotheses are changing every step, we cannot update the likelihood of each task on a step by step basis, as described in Equation 4.7. This approach allowed us to reduce the computational cost of our algorithm so as to be able to run our experiments in a reasonable amount of time. A possible option would be to use Equation 4.5, but it would requires to train a 100000 classifiers at iteration 200, which was not feasible in reasonable time.

We selected another method that relies on sampling different classifiers from a meta-classifier. It allows generating classifiers at a low computational cost. Then, given many classifiers for each task, we can compare the likelihoods predicted by these classifiers and rank the task by a statistical test. We describe each step of this process in the following paragraphs.

The first step is to compute a "meta" model that encodes a distribution of probability on the classifier parameters, i.e. which encodes a probability distribution over the mean and covariance of each class. To do so, and given that we are using multivariate normal distributions, we use a noninformative (Jeffrey's) prior [Gelman 2003] to estimate the probability distribution over the means and covariances:

$$p(\mu_l|D) = t_{n-d}(\mu|\bar{x}_l, \frac{S_l}{n(n-d)}) \tag{7.8}$$

$$p(\Sigma_l|D) = IW_{n-1}(\Sigma_l|S_l) \tag{7.9}$$

where $\bar{x}_l$ and S_l respectively represents the ML estimates of the mean and covariance for each class l based on the dataset D, n is the number of signals, and d is the dimensionality of a signal feature vector. μ_l and Σ_l are the posterior estimates of the mean and covariance given the noninformative prior. IW denotes an Inverse Wishart function which is the multidimensional generalization of the inverse Gamma, it represents a probability distribution on covariance matrix.

This "meta" model encodes the distribution of probability on the classifier parameters. Given this model we can sample, for very low computational cost, a multitude of possible Gaussian classifiers by sampling a mean and covariance for each class. The more we have data to fit our model, the less uncertainty remains and the less variability will be observed in the generated classifiers.

In our experiment, we will sample 20 classifiers per task. For each sampled classifiers, we compute the probability value (i.e. the normalized likelihood) of each task. As a result we have 20 estimations of the probability of each task. We will consider one task as the best one once one of the tasks has a significantly better probability than all the others.

To do so we model our 20 probability estimates for each task by a normal distribution, and denote as μ_{ξ_t} and σ_{ξ_t} the associated maximum likelihood estimates of the mean and variance. To compare two distributions, we compute the probability that one sample from the Gaussian associated to the first task has higher value than one sample from the Gaussian associated to the other task. To do so we compute the normal difference distribution between the two models of each task probabilities. The resulting model is also a normal distribution with mean and variance as follow:

$$\mu_{\xi_t - \xi_u} = \mu_{\xi_t} - \mu_{\xi_u} \tag{7.10}$$

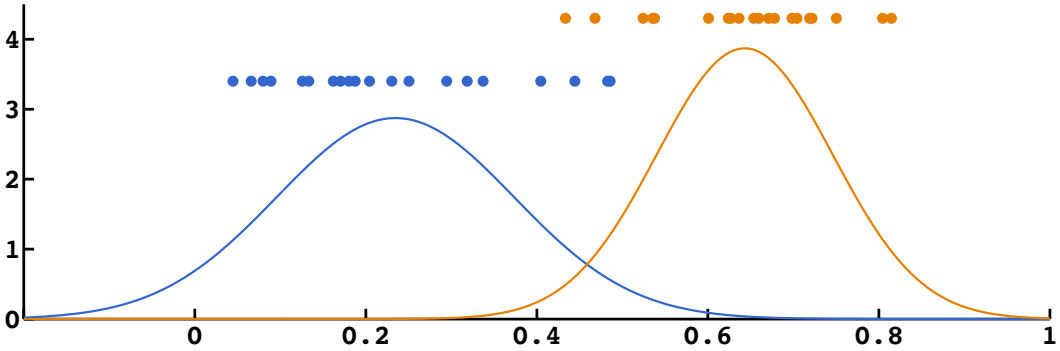
$$\sigma^2_{\xi_t - \xi_u} = \sigma^2_{\xi_t} + \sigma^2_{\xi_u} \tag{7.11}$$

Finally we compute the probability that one sample from that class has a value above zero. This is simply $1 - \Phi(\mu_{\xi_t - \xi_u}, \sigma^2_{\xi_t - \xi_u})$, with $\Phi(\mu_{\xi_t - \xi_u}, \sigma^2_{\xi_t - \xi_u})$ the cumulative normal distribution associated to the normal distribution of mean $\mu_{\xi_t - \xi_u}$ and variance $\sigma^2_{\xi_t - \xi_u}$. Then, as for equation 4.10 we take as probability for the task ξ_t the minimum of the pairwise comparison with all other tasks ξ_u with $u \in \{1, \ldots, T\} \setminus \{t\}$.

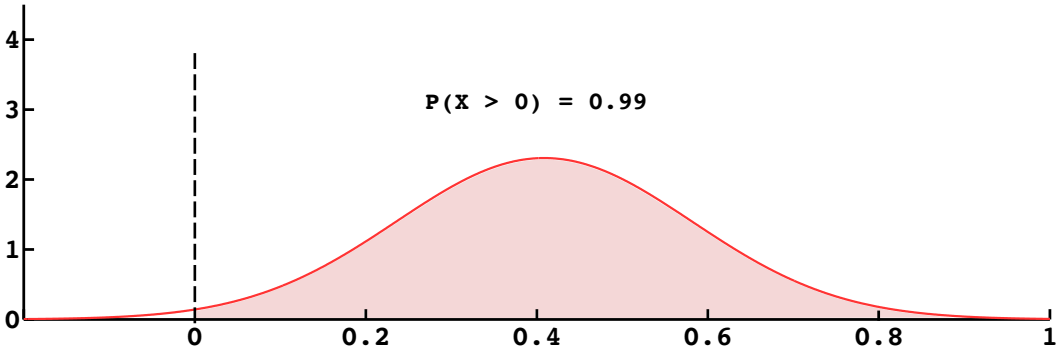
We illustrate this process in Figure 7.20. 20 samples were generated randomly to simulate some estimates for two task hypothesis and model their respective distribution using normal distribution (see Figure 7.20a). Finally we compute the probability that a sample from the distribution with highest mean has higher value than a sample from the distribution with lowest mean. We use Equations 7.10 and 7.11 to compute the mean and variance of the normal difference distribution between the two distributions; from which the area under curve from 0 to +Inf is our probability measure.

There are several weaknesses in this approach and we note that modeling a distribution on the interval [0, 1] using a normal distribution is not appropriate. Using a beta distribution would have been more suitable but we could not find an analytical solution to the difference between two beta distributions. However, we tried to use more standard statistical test, such as the one tailed Student's t-test or the Welch's t-test, but the results were not satisfying as these tests only check whether or not the means of the distributions are equals.

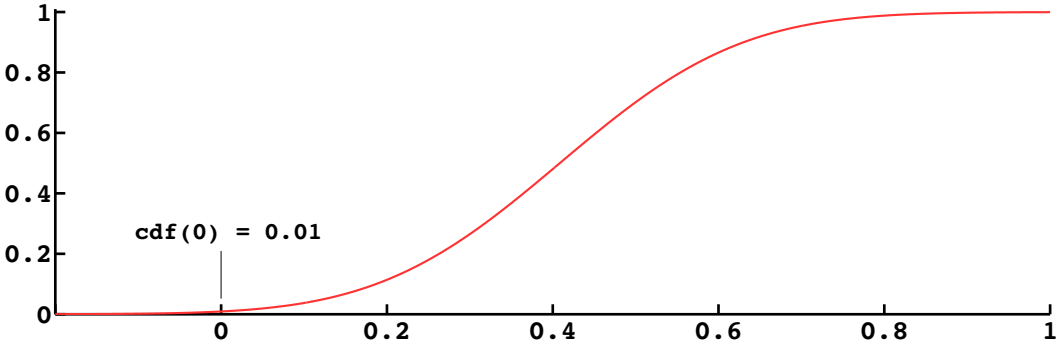
Note that it may seem more straightforward to directly compute the marginal probability distribution of Equation 5.5, which integrates over the full distribution of parameters. Here we tried to get a measure of confidence on top of our likelihood estimates. This is why we generate several classifiers, test their performances and measure the probability that one set of classifiers is on average better that another set of classifiers. To do so we model the distribution of performances of a set of classifiers by a normal distribution; and compute the probability that a sample drawn from the distribution associated to one set of classifiers has higher value than one drawn from the distribution associated the an other set of classifiers.



(a) Normal distributions fitted from the estimated values of two hypotheses. On top are the 20 samples associated to each hypothesis. The orange distribution has a mean of 0.71 and a standard deviation of 0.17. The blue distribution has a mean of 0.23 and a standard deviation of 0.16.



(b) Normal difference distribution between the two distribution of Figure 7.20a (the orange one minus the blue one). Mean is 0.48 and standard deviation is 0.23. From this distribution we estimate the probability that a sample has a value above zero, in this example it would be 0.99.



(c) Cumulative normal distribution of the Gaussian in Figure 7.20b.

Figure 7.20: The procedure used to estimate the probability that one hypothesis generates better classifiers than an other.

7.5.5 Selection and generation of task hypotheses

As there is an infinity of possible goal states, the agent cannot estimate the probability of all possible tasks in parallel. We rely on a particle filter based approach [Gordon 1993, Doucet 2009, Thrun 2002]. The main idea consist of sampling a finite number of tasks and compute a confidence measure for each of these tasks. Given the ranking between them, we will apply a resample step that consist of keeping some of the best task and sample new ones.

There are many parameters that will influence the performance of such an algorithm. We can change the number of tasks sampled, the criteria for selecting the tasks that stay in the pool from one step to another, and we can change the method used to sample new tasks.

As this is an exploratory experiment, we will restrict our analysis to the influence of the method used to resample the pool of task hypothesis, and consider either a random or an active strategy. We consider a pool of 50 hypotheses. Each step, we will keep only the best hypothesis from the pool and replace the 49 others using one of the sampling strategies define next.

The random generation of task simply keeps the best hypothesis and generates 49 new tasks hypothesis randomly.

Our active task generation method simply selects new tasks around the current best task hypothesis. To do so, we create a mixture of Gaussians that define the probability distribution used to sample the new tasks. This mixture model is composed of:

- one fixed Gaussian at the center of the state space (i.e. $[0.5, 0.5]$), with a diagonal covariance matrix, where each value on the diagonal is equal to 0.1, and have an associated weight of 0.2. This Gaussian, which has a large covariance matrix relative to the state space, maintains a level of exploration in the task generation process.

- a multitude of Gaussians, one at each location of the previous hypothesis positions (i.e. hypothesized task), whose associated weights are proportional to the probability associated to each of these tasks. The sum of the weights of these Gaussians is 0.8, such as the sum of the weights all mixture components is 1. All these Gaussians have a diagonal covariance matrix, where each value on the diagonal is equal to 0.01. For computational purpose, each Gaussian had a minimal weight of $1e^{-6}$.

Note that the resulting distribution will be truncated as all the points generated outside of the boundaries of the space (i.e. between 0 and 1 for each dimension) will be shifted to the closest position in the state space.

7.5.6 Uncertainty based state sampling

The agent can also control the next state to teleport to. As seen in chapter 5, actively controlling agent states can lead to better performances. Indeed the state

of the agent influences the signal sent by the teacher.

We will compare two kinds of sampling, random and an uncertainty based method. The random method simply teleport the agent to a random position in the world. The active method relies again on a sampling method. At each step, we generate 1000 states randomly and compute the uncertainty associated to these states using the method described in chapter 5 by Equation 5.4 and using up to 20 sampled signals from our history of interaction. To choose the next state, we select, among the 1000 points, the state that has higher uncertainty, and teleport the agent to that state in order to collect the next teaching signal.

7.5.7 Results

We compare all four combinations of the methods described above a) random state and task selection (which we call "random random"), b) random selection of next state and active task sampling (which we call "random active"), c) uncertainty based selection of next state and random task selection (which we call "uncertainty random"), and d) uncertainty based selection of next state and active task sampling (which we call "uncertainty active").

We ran 100 simulated experiments for each method and each dataset. Each experiment lasted 200 iterations and started by 12 random steps such as to collect enough signals to use Equation 7.8 and 7.9 with our 11 dimensional signals.

Distance to goal state We first analyze the results using the directional finger movement dataset shown in Figure 7.18. For each method, we compare the evolution of the distance between the best task hypothesis through iteration (the more probable according to our estimate) and the goal state (see Figure 7.21).

Only the combination of actively sampling new tasks and actively selecting new states based on their uncertainty has overall better performance than any other combination of methods. These two methods are complementary, our active task sampling method allows to explore close to our previous best estimates, and our uncertainty based state selection allows to sample states in very precise location to be able to differentiate between close hypothesis. It also explains why using one of the active methods alone does not reach the same performances.

In Figure 7.21 (right), we compare the distribution of final distance between our best hypothesis and the true goal position. First note that the important difference between displaying our results in terms of mean and standard error or in terms of a box plot, which shows the median and the 25th and 75th percentile (you can see the mean value as a colored dot). Especially for the "uncertainty random" method, the visual impression of the performance of the methods differs. This is due to the outliers, where even a few values far away from the main group of point can "push" the mean away. The normal distribution assumption does not hold for presenting our results.

Therefore, in order to statistically compare the efficiency of our methods, we use the Mann-Whitney U-test [Mann 1947] that is a nonparametric test for equality of

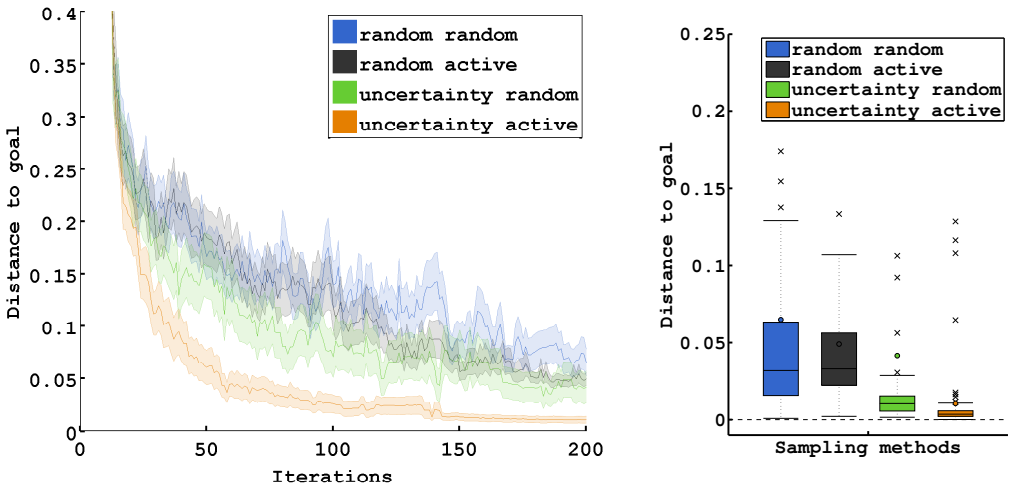


Figure 7.21: Evolution of the distance to the target using the directional finger movement dataset shown in Figure 7.18. On the left is the evolution of the distance of the best position hypothesis to the goal position (mean and standard error shown as shaded area). On the right is a box plot of the distance of the best position hypothesis to the goal position at the end of the 200 iterations. Actively sampling new task hypothesis as well as selecting new state based on our uncertainty estimation outperform allow to identify the target position with very high accuracy and low variance. We note that some distant outliers are not shown on the box plots for readability reasons.

population medians of two independent samples. We use the one tailed version to specifically test whether one population has greater performances than the other. There is no measurable statistical difference between the "random random" and "random active" methods ($p = 0.68$). The "uncertainty random" performances over "random random" ($p < 1e^{-10}$) and "random active" ($p < 1e^{-10}$) are highly significant. As well as the difference between the "uncertainty active" and "uncertainty random" difference in performance ($p < 1e^{-10}$).

The results presented above were obtained using the directional finger movement dataset shown in Figure 7.18. We now demonstrated how the same algorithm could handle different user finger gestures. We repeat the experiment with the cardinal sign dataset of Figure 7.19.

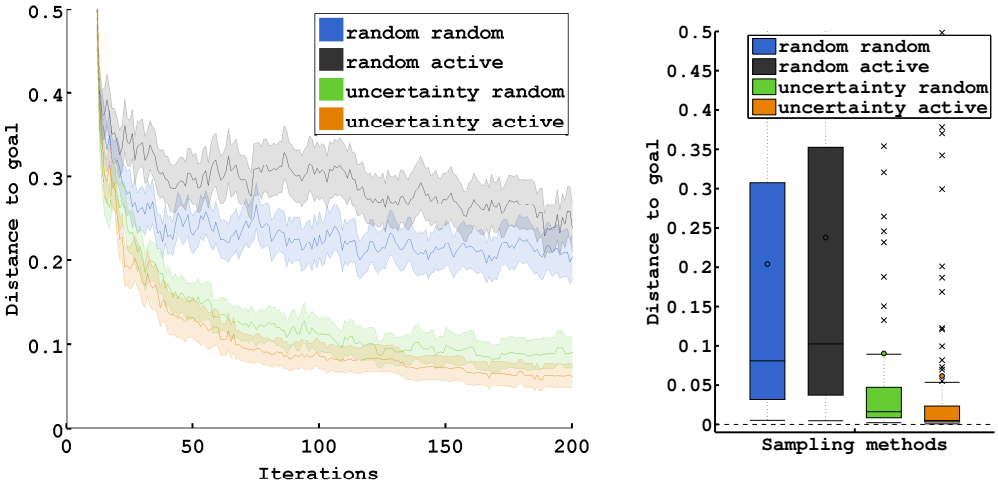


Figure 7.22: Evolution of the distance to target using the cardinal signs finger movement dataset shown in Figure 7.19. On the left is the evolution of the distance of the best position hypothesis to the goal position (mean and standard error shown as shaded area). On the right is a box plot of the distance of the best position hypothesis to the goal position at the end of the 200 iterations. Actively sampling new task hypothesis as well as selecting new state based on our uncertainty estimation outperform allow to identify the target position with very high accuracy and low variance. We note that some distant outliers are not shown on the box plots for readability reasons.

Figure 7.22 (left) shows the evolution of the distance between the best task estimates and the goal task. We observe a larger difference between random and uncertainty based selection of next state. This could be explained by the properties of the cardinal sign dataset, where the system must consider all features to differentiate between classes, therefore requiring more signals to be collected. For the directional finger movement dataset (Figure 7.18), only two features (end position on X and Y axis) were enough to differentiate between all classes.

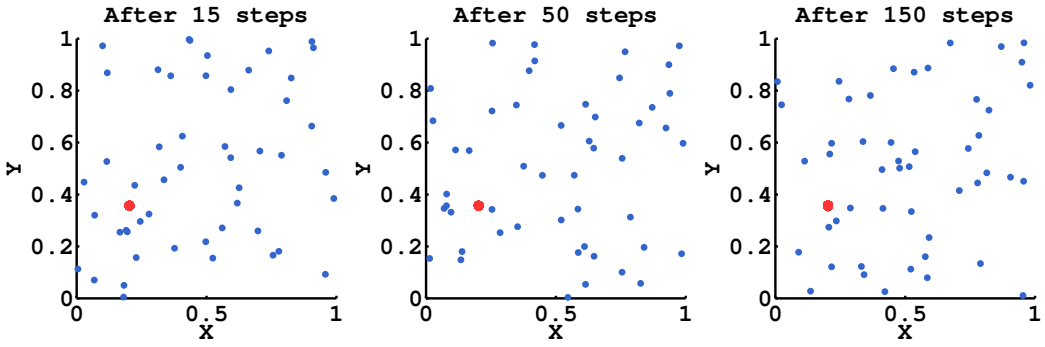
In Figure 7.22 (right), we compare the distribution of final distance between

our best hypothesis and the true goal position. There is no measurable statistical difference between the "random random" and "random active" methods ($p = 0.8687$). The "uncertainty random" performances over "random random" ($p < 1e^{-10}$) and "random active" ($p < 1e^{-10}$) are highly significant. As well as the difference between the "uncertainty active" and "uncertainty random" difference in performance ($p < 1e^{-5}$).

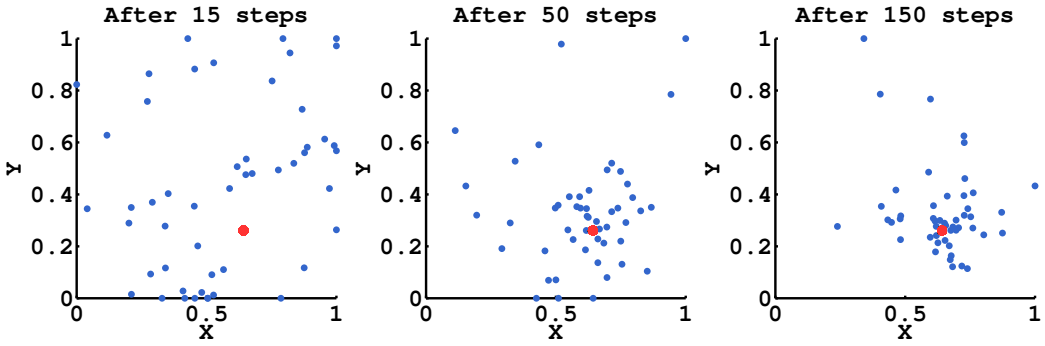
Finally, we note that the final median distance between the best position estimation and the goal position are 0.0036 and 0.0051 for the directional movement and cardinal sign movement respectively. It is important to project these results in the real word, it means that given a one meter square area, our agent is able to find the position the user has in mind with less than 5 millimeters (half of the time and given 200 requests), but without knowing the signal to meaning mapping beforehand. Moreover, as we have seen with our two datasets (Figure 7.18 and Figure 7.19), given our simple representation of the finger movements, a great variety of possible signals can be considered.

Task sampling comparison We briefly illustrate the difference between our two task sampling methods, namely "random" and "active".

Figure 7.23 shows the task sampled (in blue) at steps 15, 50, and 150 following a random (Fig. 7.23a) or an active (Fig. 7.23b) resampling step. The active resampling allows focusing the set of task around the goal state (in red), which increases the changes of finding a better estimate of the task.



(a) Example of task hypothesis sampled using a random method.

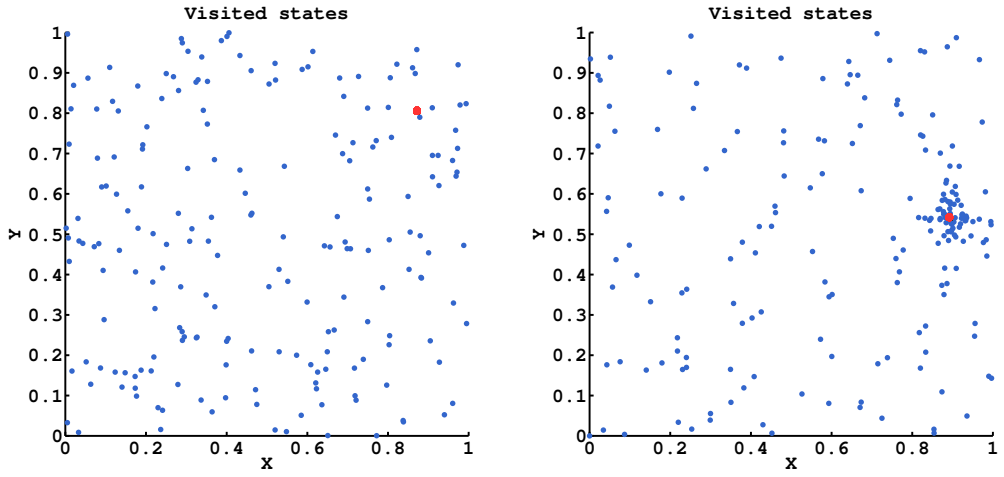


(b) Example of task hypothesis sampled using our active method.

Figure 7.23: Examples of task hypothesis sampling strategies. In red is the goal task. In blue are the sampled task hypothesis at iteration 15, 50 and 150. The active sampling method progressively focus his sampling around the goal state.

State sampling comparison We briefly illustrate the difference between our two state sampling methods, namely "random" and "uncertainty".

Figure 7.24 shows the state visited (in blue) at after 200 steps following a random (Fig. 7.24a) or an uncertainty based (Fig. 7.24b) next state selection method. The uncertainty based method allows visiting more states around the goal state (in red).

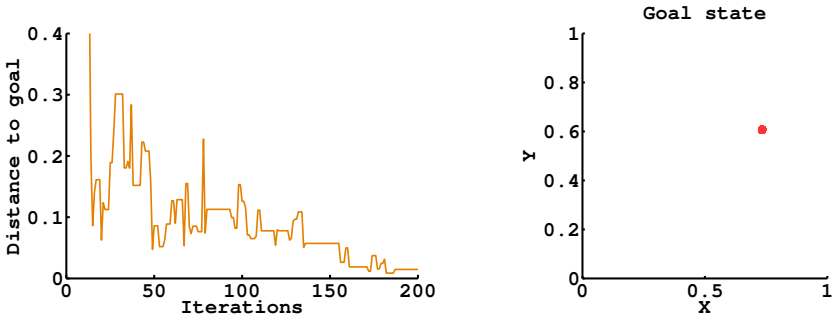


(a) Visited states after 200 steps with random state selection.

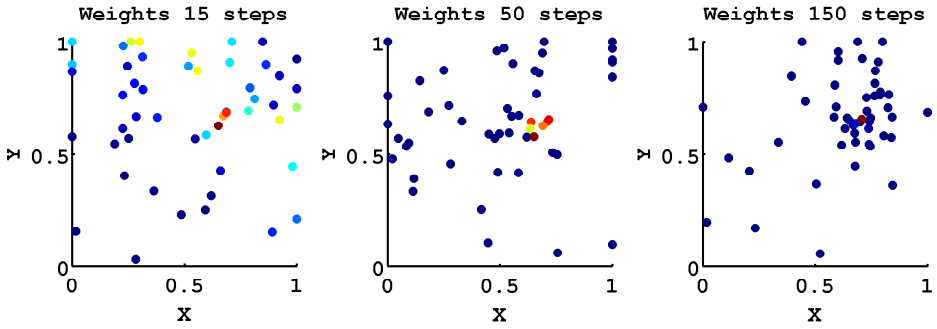
(b) Visited states after 200 steps with uncertainty based state selection.

Figure 7.24: The state visited by the agent, i.e. in which it received instruction signals, after 200 steps. In red is the goal task. Comparison of random state selection (left) and uncertainty based state selection (right). Selecting state according to their uncertainty allow to collect more information around the goal state, which allow to identify more precisely the target location.

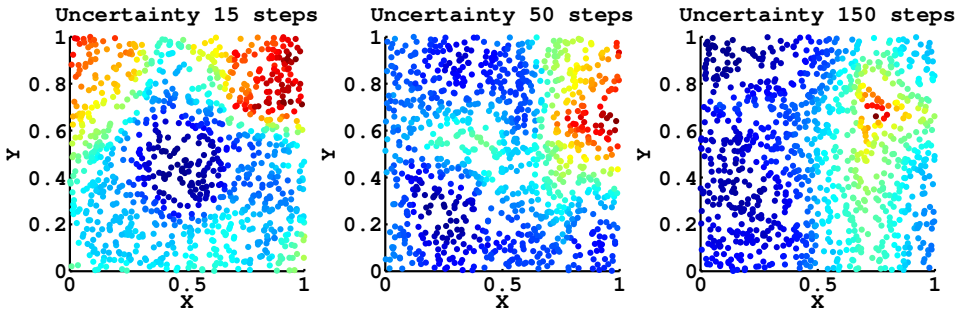
Figure 7.25 details the uncertainty based method for selecting the next visited states. Figure 7.25a present the goal state for this specific run (right) and the evolution of the distance to the goal of the best estimate (left). Figure 7.25b shows the set of task hypothesis at steps 15, 50, and 150 (as for Figure 7.23) where the colors associated to each point represent the estimated probability of each task (red is high, blue is low). To sample the following agent's state, we sample 1000 states randomly and estimates the uncertainty of each of those point using the method described in chapter 5 by Equation 5.4. Computing this uncertainty requires to have a set of hypothesis and their associated weights (shown in Figure 7.25b), access to the interaction frame, and to a classifier for each hypothesis. The resulting maps for step 15, 50, and 150 are displayed in Figure 7.25c where the colors represent the uncertainty values (red is high, blue is low). The less the distribution of probably on task is flat, the more the uncertainty map is narrow and focused around the best estimate. The active task resampling step is beneficial to the uncertainty state selection step as it allows to specify the uncertainty in key areas.



(a) Evolution of distance between the best estimate and the goal position (left). The goal position being the red dot on the right plot.



(b) Set of hypothesis after 15, 50, and 150 steps, with their associated probability show as colors. The associated values to each color are different for each time step. Red is associated to the most probable task, and blue for the less probable. The colors linearly maps according to their associated weights, red (blue) for the most (less) probable task.



(c) Uncertainty associated to each of the 1000 sampled states after 15, 50, and 150 steps. The most uncertain state will be selected as the next state of the agent. The uncertainty maps evolve through time as the set of task hypothesis and their respective probabilities are updated. After 150 steps, the uncertainty is located around the goal position. The colors linearly maps according to their associated weights, red (blue) for the most (less) probable task.

Figure 7.25: Illustration of the uncertainty based state sampling process after 15, 50, and 150 steps. The uncertainty is evaluated for 1000 randomly generated states according to the current set of task hypothesis and their associated probabilities.

7.6 Interaction frame hypothesis

How to relax the assumption that interaction frame is pre-defined and unique?

Until now we have assumed that the interaction frame, which specifies the details of the interaction between the human and the machine was known. In this section, we considered the case where multiple interaction frames are defined, but only one of them accounts for the interaction between the human and the machine.

7.6.1 Illustrations

We will use a very simple example to illustrate the problem and show computational results. We consider that the agent lives in the line word as defined in chapter 4.3.3, where the agent has access to the "no move" action in order to remove the symmetry problem. The agent knows it should reach either of the two edges of the world, G1 or G2. And the agent knows that the teacher is providing either feedback or guidance instructions. To handle this new hidden information we will rely again on our interpretation hypothesis process. This time, one hypothesis will be the combination of one task hypothesis and one frame hypothesis. For our simple example, it results in having four hypotheses.

The result of the labeling process is shown in Figure 7.26 for a teacher providing feedback instructions according the task G1. The hypothesis that labels the signals according to the task G1 and the feedback frame is the one whose signal-label pairs match better with the underlying structure of the data. Indeed, for the guidance case, the labeling process for hypothesis G1 is always giving a "left" label whether or not the agent is moving away or closer to the target, which allows to differentiate between feedback and guidance cases. To differentiate between G1 and G2, the same principle than the one described in chapter 4.3.3 applies.

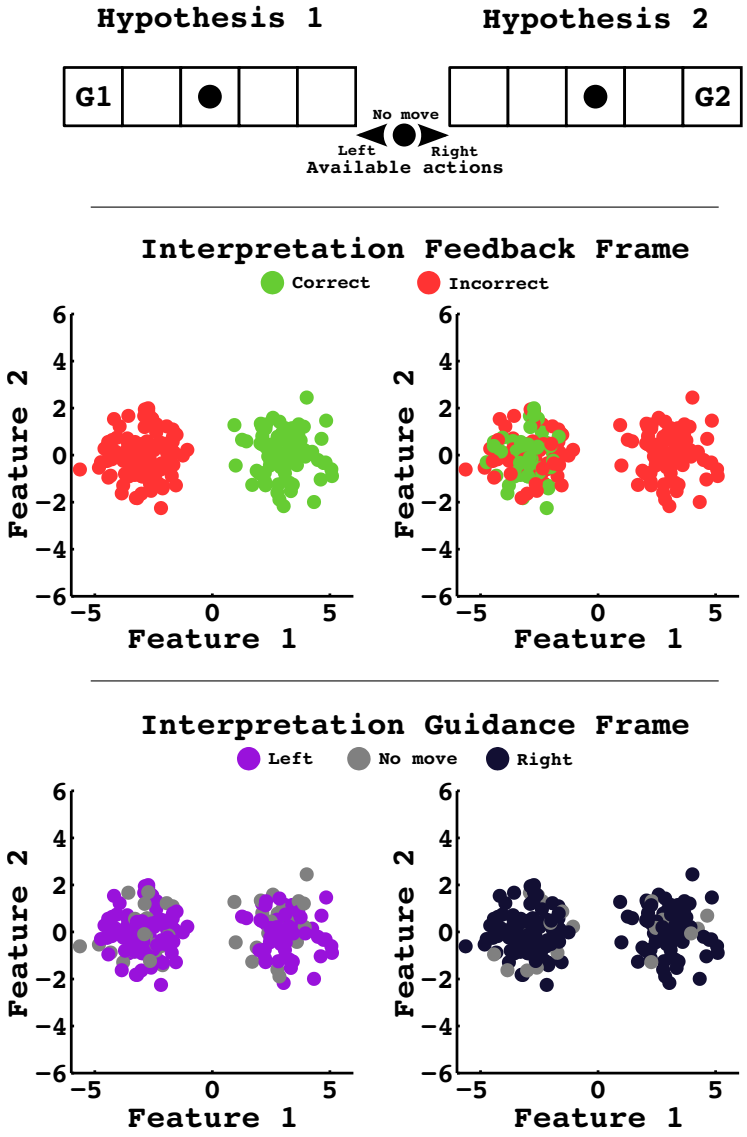


Figure 7.26: Illustration of the labeling process on both task and interaction frame hypothesis. The agent can perform right, left, or a "no move" action. The agent receives feedback on its action in the line word according to G1 . The agent does not know which task (G1 or G2) neither which interaction scheme the teacher is following (feedback or guidance). The result of the labeling process allows to identify the hypothesis on task G1 and feedback frame as the more likely.

Considering now that the teacher is providing guidance instructions according the task G1, the results of the labeling process can be seen in Figure 7.27. The same explanation than for Figure 7.26 applies.

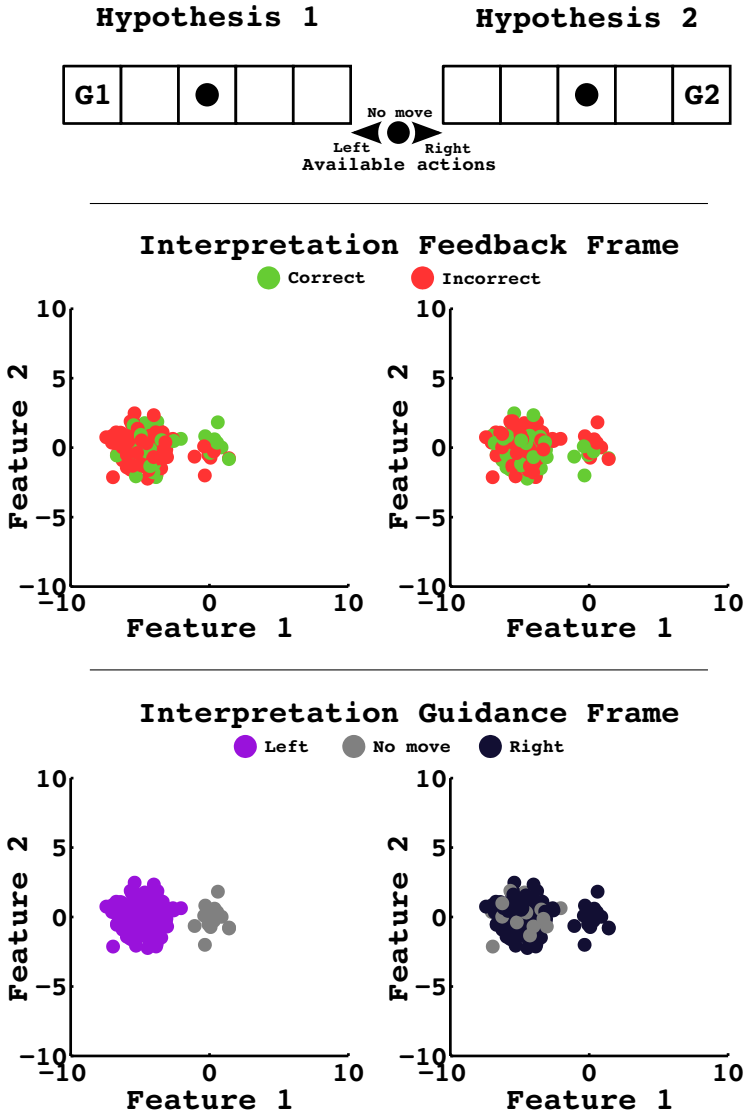


Figure 7.27: Illustration of the labeling process on both task and interaction frame hypothesis. The agent can perform right, left, or a "no move" action. The agent receives guidance instruction on its action in the line word according to G1 . The agent does not know which task (G1 or G2) neither which interaction scheme the teacher is following (feedback or guidance). The result of the labeling process allows to identify the hypothesis on task G1 and guidance frame as the more likely.

7.6.2 Simple experiments

We now verify that the algorithm works in practice. We consider the same line world scenario as described above. For our experiments, the simulated teacher selects randomly a target (G1 or G2) and an interaction frame (feedback or guidance). The agent is using our uncertainty based planning method. We ran 100 simulations. All other settings were set as for the experiments of chapter 5.4.

Figure 7.28 shows the evolution of the probability associated to the correct combination of task and interaction frame (we use the minimum of pairwise normalized likelihood from Equation 4.10 in chapter 4.4.3). After 200 steps, all our experiments identified with probability 1 the correct combination of task and interaction frame.

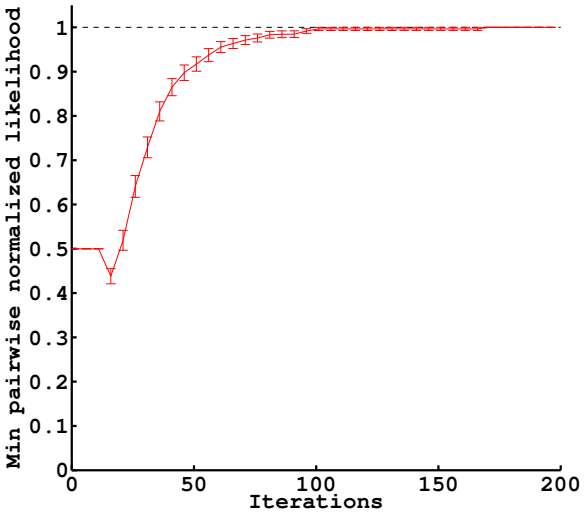


Figure 7.28: Evolution of the minimum of pairwise normalized likelihood for the correct hypothesis. After 200 steps, all our experiments identified with probability 1 the correct combination of task and interaction frame.

We plot in Figure 7.29 the cases where the teacher was using the feedback frame (left) or the guidance frame (right). The performances are similar in both cases.

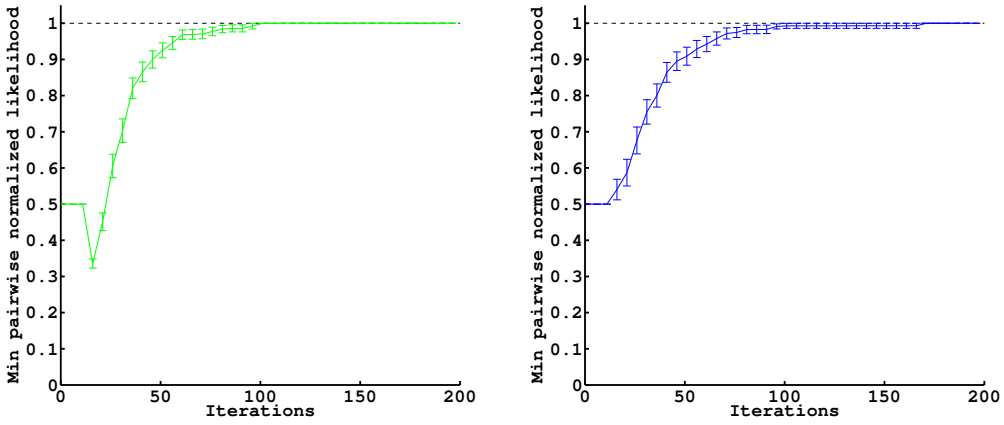


Figure 7.29: Evolution of the minimum of pairwise normalized likelihood for the correct hypothesis if the teacher provided feedback (left) or guidance (right) instruction. After 200 steps, all our experiments identified with probability 1 the correct combination of task and interaction frame. Most of the experiments would have identified the task in slightly more than 50 steps with a confidence threshold of 0.9.

7.6.3 Discussion

Following the interpretation hypothesis method on a combination of task and interaction frame, we can start learning a task from unlabeled instructions and undefined interaction frames. In other words, such system cannot only learn the task and the signal to meaning mapping, but also the interaction protocol used by the teacher.

Considering our example in section 7.5, an application this method can be to consider different coordinate system for the cardinal frame. For example, the signals from the teacher can be relative to the true North magnetic pole, to the current position of the user relative to the agent, or relative to the current orientation of the robot. This experiment performed with a real robot, real users, considering a tablet, and different interaction frames has great potential to demonstrate the potential application of this work.

Finally, we note that a particle filter based method (as used in section 7.5 for dealing with continuous task) could be considered for dealing with a continuous set of interaction frames. For example, in our example of section 7.4, we used a parameterized frame that merged feedback and guidance frame (see Equation 7.6), and introduced a feedback to guidance ratio α. By generating, testing, and resampling a set of α, i.e. a set of interaction frames, we may be able to learn, not only the task and the signal to meaning mapping, but also the details of the interaction protocol used by the teacher.

7.7 A minimalist proof

It is of paramount importance to understand the properties of our algorithm, and to be able to have some certitude about its convergence and accuracy properties. The work presented in this thesis neglected this aspect and relied only on empirical evaluation. In the following of this section, we present a proof about the principle of our algorithm under restricted condition.

7.7.1 Problem and assumptions

We consider a robot in a discrete state and action world. A teacher is providing feedback instructions to the robot through the use of a simple interface with two buttons, one button for "correct" and one button for "incorrect". But the mapping between the buttons and the meanings is unknown to the robot at start. This simplified setting allows studying specific details of the algorithm in more details, without the problem of dealing with continuous signals.

We assume that the user is coherent and uses one button for one meaning, and always the same button for the same meaning. Therefore, as exemplified in Figure 7.30 the mapping between symbolic signals and their meaning can only be of two forms.

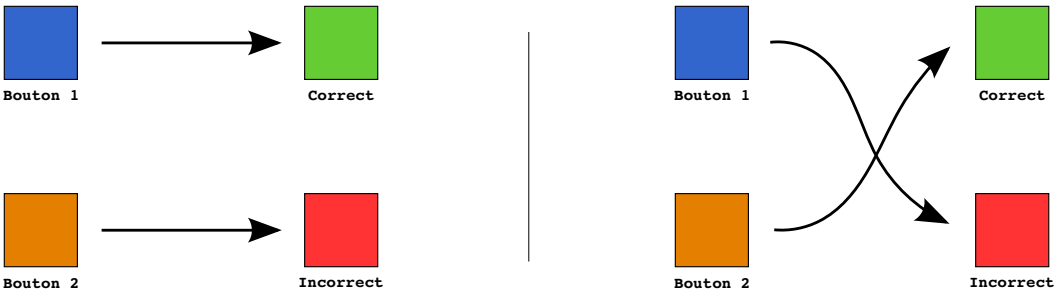


Figure 7.30: The two possible button to meaning mapping.

We further assume the robot is provided with a set of task hypothesis $(\xi_1, \ldots, \xi_T)$, represented by their associated policies $(\pi_1, \ldots, \pi_T)$. This set includes the task $\hat{\xi}$, the teacher as in mind, and when the robot performs an optimal action according to the optimal policy $\hat{\pi}$, the teacher presses the button associated to the "correct" meaning. He respectively presses the button associated to the "incorrect" meaning for a non-optimal action. We assumed that the teacher never makes teaching mistakes.

We define a number of terms that will simplify the notation in further subsections. nS is the number of states in the environment, nA is the number of actions available to the robot, and nSA is the number of state-action pairs an agent can visit, which is simply $nS * nA$. We note as $diff(\pi_t, \pi_u)$ the number of optimal state-action pairs that differs between the optimal policies π_t and π_u respectively associated to the task ξ_t and ξ_u. Therefore the ratio of optimal state-action pairs that differs between two task hypothesis is denoted as $\frac{diff(\pi_t,\pi_u)}{nSA}$.

The $diff()$ function logically outputs 0 when comparing one task to itself, i.e. $diff(\pi_t, \pi_t) = 0$. And a ratio of 1 means the two tasks are symmetric (see discussion about symmetry in section 4.3.3), which means whatever the action the robot will choose, the meaning inferred according to the first task will be the opposite of the meaning inferred according to the second task. This property, as will be seen in our minimalist proof, does not allow differentiating between two symmetric tasks.

7.7.2 Illustration

Before describing our simple proof, it is important to have an intuition on the relation between the buttons and the meanings in different conditions. We consider again our T world scenario as an illustration (see chapter 4.1.1).

Figure 7.31 shows all possible button presses sequences expected from the teacher in different conditions. On top are the state-action pairs considered (a). (b) and (c) lines represent the expected meanings for each of the state-action pairs and according to the hypothesis G1 (b) or G2 (c). (d) and (e) lines represent the possible button presses sequence of the teacher when teaching hypothesis G1 and considering the two possible mappings. Respectively (f) and (g) for hypothesis G2.

First, before entering into more detail, given the extensive number of assumptions defined, for the simple example of Figure 7.31 it would be easy to find the correct hypothesis by visiting only two state-action pairs. Indeed as taking an action in the trunk of the T will be interpreted similarly by both hypotheses, and given the user is not making teaching mistakes and is coherent in its use of the buttons, we could instantaneously know the meaning of the button pressed and therefore the meaning of the other button. Then taking an action in the top bar of the T would allow us to differentiate between G1 and G2. However we will not exploit this type of properties for our proof and we remind that in all the experimental scenarios presented in this thesis, there were no state-action pairs that allowed for an unequivocal interpretation of a signal.

For the purpose of our demonstration, we should read this figure by comparing lines (d), (e), (f), and (g) with the expectation from lines (b) and (c). We will denote B the blue button, O the orange button, C the "correct" meaning (the green patch), and W the "incorrect" meaning (the red patch, W for wrong). For example, let's imagine we receive the sequence of presses of line (d) that we note $[B, O, O, B]$. For hypothesis 1 (G1) we expected $[C, W, W, C]$, and for hypothesis 2 (G2) we expected $[W, C, W, C]$.

Given these two possible interpretations we can build a statistical model for the signal to meaning mapping. For G1, we obtain the following model $p(C|B, G1) = \frac{2}{2} = 1$, $p(C|O, G1) = \frac{0}{2} = 0$, and $p(W|B, G1) = \frac{0}{2} = 0$, $p(W|O, G1) = \frac{2}{2} = 1$. To simplify notation we note $[1, 0]_{B,G1}$ and $[1, 0]_{O,G1}$ the model for each button where the first element of the vector is the probability associated to the "correct" meaning and the second is the one associated to the "incorrect" meaning. And the underscript details the button and the task considered. Using the same reasoning, again for line (d) but for hypothesis G2, the classifier is: $[0.5, 0.5]_{B,G2}$ and $[0.5, 0.5]_{O,G2}$.

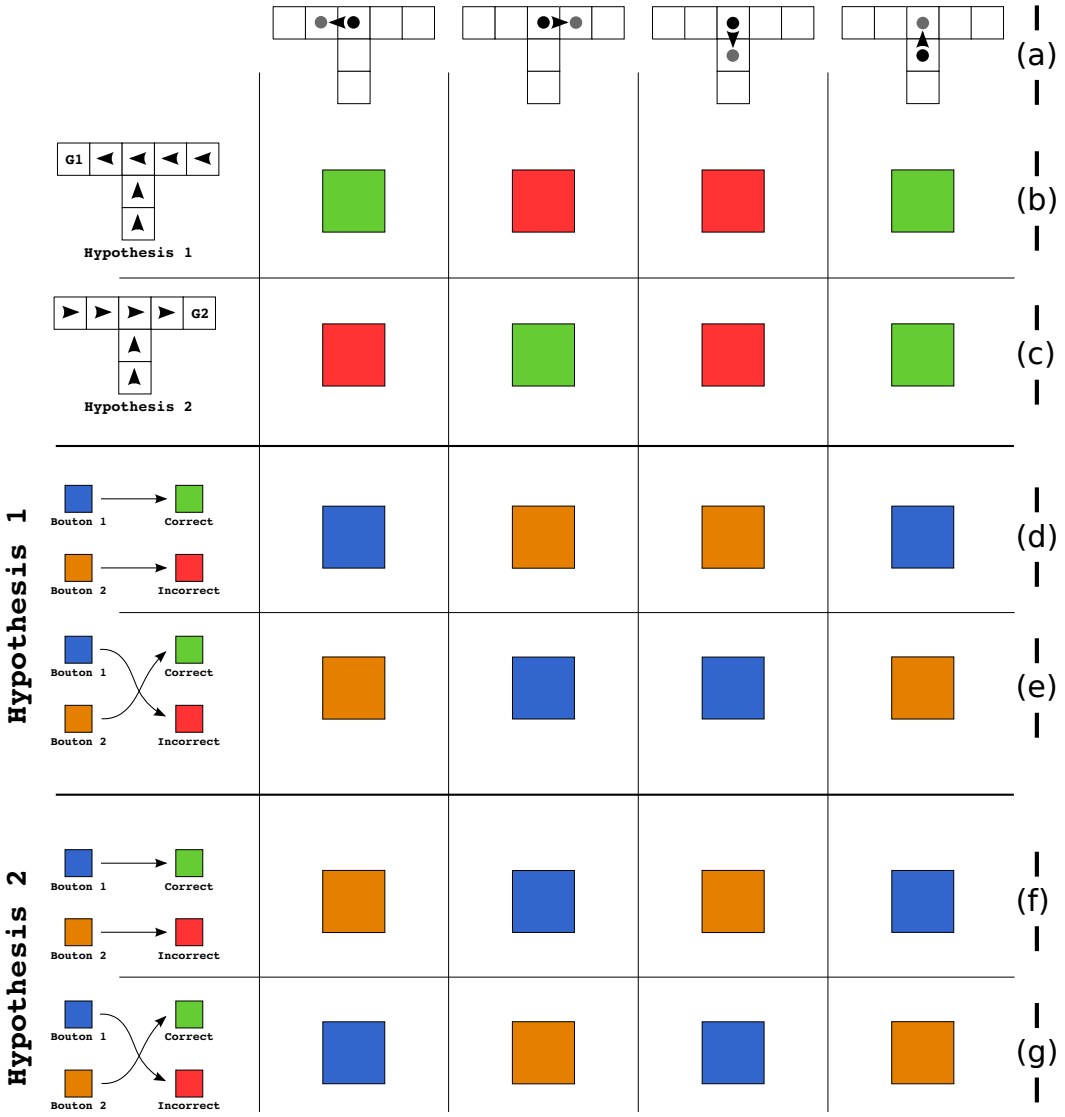


Figure 7.31: Illustration of the teacher's button presses for several state-action pairs. On top are the state-action pair considered (a). (b) and (c) lines represent the expected meanings for each of the state-action pairs and according to the hypothesis G1 (b) or G2 (c). (d) and (e) lines represent the possible button presses sequence of the teacher when teaching hypothesis G1 and considering the two possible mappings. Respectively (f) and (g) for hypothesis G2.

In this thesis, as we were not considering symbolic signals, and used a metric that compares the expectation from the frame (i.e. lines (b) and (c)) with prediction from the classifier associated to each task. Let's use Equation 4.2 to compute the likelihood for each task. We remind the likelihood equation:

$$\begin{aligned}\mathcal{L}(\xi_t) &= \prod_{i=1,\dots,M} p(l_i^c = l_i^f | D_M, \xi_t) \\ &= \prod_{i=1,\dots,M} \sum_{k=1,\dots,L} p(l_i^c = l_k | e_i, \theta_M) p(l_i^f = l_k | s_i, a_i, \xi_t)\end{aligned}$$

where $p(l_i^c = l_k | e_i, \theta_M)$ is the classification of the signal e_i from our classification model θ_M, and $p(l_i^f = l_k | s_i, a_i, \xi_t)$ is the expected label given by the frame.

For G1 we obtain:

$$\begin{aligned}\mathcal{L}(G1) &= ((1 \times 1) + (0 \times 0))\,((0 \times 0) + (1 \times 1))\,((0 \times 0) + (1 \times 1)) \;\dots \\ &\quad \dots\; ((1 \times 1) + (0 \times 0)) \\ &= 1\end{aligned}$$

And for G2 we obtain:

$$\begin{aligned}\mathcal{L}(G2) &= ((0.5 \times 0) + (0.5 \times 1))\,((0.5 \times 1) + (0.5 \times 0)) \;\dots \\ &\quad \dots\; ((0.5 \times 0) + (0.5 \times 1))\,((0.5 \times 1) + (0.5 \times 0)) \\ &= 0.0625\end{aligned}$$

By normalizing the likelihoods, we obtain the probability of each task: $p(G1) \approx 0.94$ and $p(G2) \approx 0.06$. We see that our measure of likelihood is able to identify the correct task. The same process can be repeated for each case (i.e. (e), (f), and (g)) and will always identify the correct hypothesis.

Given this explanation, we can start drafting the proof, but, to simplify further the proof, we will add an additional assumption.

7.7.3 The proof

In order to make the proof simple and illustrative, we assume each hypothetic policy has an equal number of optimal state-action pairs than of non-optimal state-action pairs. Therefore when the agent has visited once all the state-action pairs, it has collected the same amount of signals with label "correct" than with label "incorrect". It ensures that, when the agent has visited once all the state-action pairs, the user will have pressed as many time the blue button than the orange button, exactly $\frac{nSA}{2}$ times. But it also ensures that all interpretation hypotheses estimated that half of the labels are of meaning "correct" and half are of meaning "incorrect". Therefore, if the agent visits once all the state-action pairs, the signal to meaning model for each class (i.e. for C and W) will be symmetric for every task hypothesis considered.

Therefore, we can evaluate the possible signal to meaning mappings based on the difference in policies between the optimal task and any hypothetic task using

the $diff()$ function defined earlier. For a given task ξ_t we can compute the ratio of optimal state-action pairs that are the same as for the true task $\hat{\xi}$, which we denote $\Upsilon_{\xi_t} = \frac{nSA - diff(\pi_t, \hat{\pi})}{nSA}$. The agent will never have access to this information and we only use this measure for our proof.

Given our previously defined assumption, and assuming the agent visited all state-action pairs once, if the user uses the blue (B) button to mean "correct", then the blue signal model will be $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}]_{B,\xi_t}$. Which implies the orange button mapping is $[1 - \Upsilon_{\xi_t}, \Upsilon_{\xi_t}]_{O,\xi_t}$. Respectively, if the user uses the blue button to mean "incorrect", then the blue signal model will be $[1 - \Upsilon_{\xi_t}, \Upsilon_{\xi_t}]_{B,\xi_t}$. Which implies the orange button mapping is $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}]_{O,\xi_t}$.

Deriving the likelihood with respect to Υ_{ξ_t}

Using this notation, we can write the likelihood equation as a product of vector's products. Where for example, $\sum_{k=1,\dots,L} p(l_i^c = l_k | e_i, \theta_M) p(l_i^f = l_k | s_i, a_i, \xi_t)$ can be written as $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}]_{B,\xi_t} . [1, 0]^T_{s_i,a_i,\xi_t}$ for those cases where the user pressed the blue button (i.e. $e_i = B$) after an optimal state-action pair (i.e. the expected meanings is "correct", i.e. $[1, 0]_{s_i,a_i,\xi_t}$), and given that the blue button was the one used by the teacher to mean "correct", resulting in $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}]_{B,\xi_t}$ as the button to meaning model.

We can now list all the possible cases. We can split the state-action pairs in half, the one that are optimal according to the teacher intended task $\hat{\xi}$ (there is $\frac{nSA}{2}$ of them) and the one that are non-optimal according to the teacher intended task $\hat{\xi}$ (there is $\frac{nSA}{2}$ of them). For the state-action pairs that are optimal, the user will press the button he uses to mean "correct" (i.e. the blue or the orange one), respectively for the non-optimal, he will press the other button (i.e. the orange or the blue one).

But the agent evaluates those button presses with respect to the task hypothesis currently considered ξ_t, which might not be the one the teacher as in mind. Therefore, only a fraction of the time the button presses match with what is expected by the task considered ξ_t. This number can be exactly identified as $\frac{nSA}{2} . \Upsilon_{\xi_t}$. Therefore for $\frac{nSA}{2} . \Upsilon_{\xi_t}$ state-action pairs, the "correct" button was pressed for the "correct" meaning. For one state-action pair, this represent an update of the likelihood function by $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}] . [1, 0]^T$, which is simply Υ_{ξ_t}. As there is $\frac{nSA}{2} . \Upsilon_{\xi_t}$ similar situations the update is $\Upsilon_{\xi_t}^{\frac{nSA}{2} . \Upsilon_{\xi_t}}$.

Similarly there is $\frac{nSA}{2} . (1 - \Upsilon_{\xi_t})$, where the "incorrect" button was pressed for the "correct" meaning. Which represents an update of $[\Upsilon_{\xi_t}, 1 - \Upsilon_{\xi_t}] . [0, 1]^T$, which is simply $1 - \Upsilon_{\xi_t}$. As there is $\frac{nSA}{2} . (1 - \Upsilon_{\xi_t})$ similar situations the update is $(1 - \Upsilon_{\xi_t})^{\frac{nSA}{2} . (1 - \Upsilon_{\xi_t})}$.

And, as the situation is symmetric for the non-optimal state-action pairs of the teacher intended task $\hat{\xi}$, the likelihood equation can be rewritten as:

$$\begin{aligned}\mathcal{L}(\xi_t) &= \Upsilon_{\xi_t}^{\frac{nSA}{2}.\Upsilon_{\xi_t}} \times (1-\Upsilon_{\xi_t})^{\frac{nSA}{2}.(1-\Upsilon_{\xi_t})} \times \Upsilon_{\xi_t}^{\frac{nSA}{2}.\Upsilon_{\xi_t}} \times (1-\Upsilon_{\xi_t})^{\frac{nSA}{2}.(1-\Upsilon_{\xi_t})} \\ &= \Upsilon_{\xi_t}^{nSA.\Upsilon_{\xi_t}} \times (1-\Upsilon_{\xi_t})^{nSA.(1-\Upsilon_{\xi_t})} \end{aligned} \tag{7.12}$$

Note that this equation is the same whatever the button chosen by the user to mean "correct". We can check our previous likelihood estimate in our simple example of Figure 7.31 considering the 4 state-action pairs visited are the only one available in the world, and considering we receive button presses as in line (d). We obtain the same likelihoods as the one derived in the first subsection, i.e. $\mathcal{L}(G1) = 1^{1\times4} \times 0^{0\times4} = 1$ and $\mathcal{L}(G1) = 0.5^{0.5\times4} \times 0.5^{0.5\times4} = 0.5^4 = 0.0625$.

Analyzing the likelihood function

We can plot the likelihood function with respect to the full range of value that Υ_{ξ_t} can take, i.e. between 0 and 1. We consider that $nSA = 1$ for now. Obviously such a value of nSA is impossible in practice given our assumptions, we need as many optimal and non-optimal state-action pair, which means that nSA must be an even number. However, now that we have our theoretical estimate of the likelihood function we shall study its properties in a theoretical way. Additionally, our equation is only valid if the agent visited all state-action pairs but for the sake of our analysis, we consider that the value of nSA represents the number of state-action pair visited by the agent. Moreover, there exist a relation between the number of state-action pairs and the discrete set of value that Υ_{ξ_t} can take given our assumptions, but for the sake of the analysis we consider the full range of value between 0 and 1.

Figure 7.32 shows the likelihood function for $nSA = 1$. As expected the likelihood value is higher for the correct task, i.e. when $\Upsilon_{\xi_t} = 1$, and decrease as the number of state-action pairs that differ increases, i.e. when Υ_{ξ_t} decrease. However this function holds an interesting property, which is that once more than half of the optimal state-action pair differs with the true task, the function increases again. Until is reaches a point where none of the optimal state-action pairs of the task are the same as for the true task. This specific case is what we called a symmetric task hypothesis, where the symmetric interpretation of the feedback signals is as likely as the correct interpretation of the signals. Indeed none of the state-action pairs allow breaking this symmetry for the feedback frame.

Therefore, given all the assumptions considered, if the agent is provided with a set of task hypothesis, that included the correct task $\hat{\xi}$ but does not include any symmetric hypothesis (which means all tasks hold the following property $\Upsilon_{\xi_t} > 0$), we can guarantee that if the teacher visits all state-action pairs once, the user intended task $\hat{\xi}$ (which hold the property $\Upsilon_{\xi_t} = 1$) will have the greater likelihood. In other words: $\mathcal{L}(\hat{\xi}) > \mathcal{L}(\xi_t)$ if $\Upsilon_{\xi_t} \in\ ]0, 1[$.

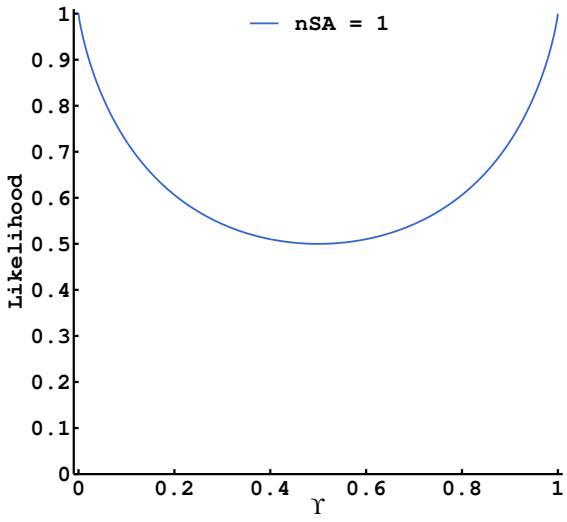


Figure 7.32: The likelihood function of Equation 7.12 for $nSA = 1$.

Building confidence

We now discuss the problem of estimating the confidence that one task is a better candidate than another one. Of course, in this setting, given the strong assumption that the user is never making mistakes, such confidence mechanism is not needed. However, we have seen in previous chapter that deciding when to stop is a critical part of the algorithm. The simplest method consists of normalizing the likelihood for each task defining a probability threshold above which a task is considered as the correct one. If we consider for example 10 task hypothesis with different values of Υ, normalizing the likelihood when $nSA = 1$ won't produce a very sharp probability distribution on task. All tasks will roughly share the same probability. However, when visiting more and more states, the likelihood function becomes more and more sharp and only normalizing likelihoods will split the hypothesis apart. In the limit, when $nSA \to +Inf$, only the hypothesis with a better value of Υ (i.e. closer to 0 or 1) will reach a probability of 1.

This model allows understanding in more conceptual terms some properties of our algorithm. And in practice very few of the assumption considered are applicable in our experimental setups.

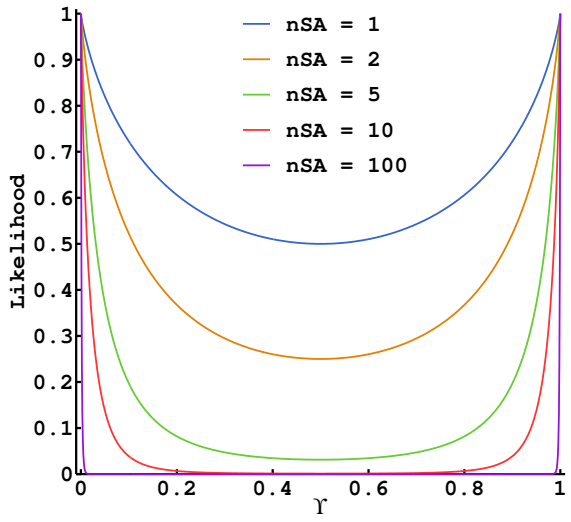


Figure 7.33: The likelihood function of Equation 7.12 for $nSA = 1, 2, 5, 10, 100$. The more we have collected evidence, the more the difference is sharp between task hypotheses.

A simple scenario

Finally, for illustration purpose, we present a world holding all our assumption properties, we named it the clock world (see Figure 7.34). This world has 12 states, which we represent as the hours on a clock. The agent has two actions available: turning clockwise or counter-clockwise. The user wants the agent to reach one of 12 states. This world is the extension of the line word but where the line loops on itself.

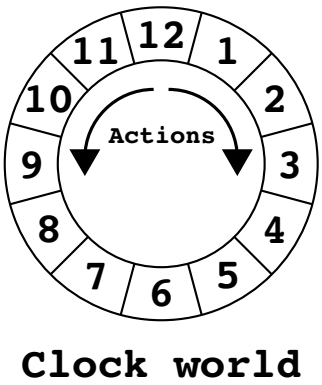


Figure 7.34: Illustration of the clock world. The agent has two actions available: turning clockwise or counter-clockwise. The user wants the agent to reach one of 12 states.

7.7.4 Why not using the entropy of the signal models?

We continue here the discussion about the difference between the method detailed in chapter 4 of this thesis, where we differentiate between hypothesis by computing the probability that all signals are correctly classified, and the method presented in section 7.3, where we consider only the signal models and differentiate between tasks by looking at the overlap of their Gaussian model associated to each class.

We can also apply a similar method than used in section 7.3 in our simple example. As we assumed that the user is coherent in his button presses, the task hypothesis whose associated button presses model is the less uncertain is the more likely to be the correct one. Given the history of interaction, we can model the button presses of the user as Bernoulli variables, which can take only two values "correct" or "incorrect", such as a coin-flipping problem. Following the previous development, we can compute the probability that the user will press the blue button to mean "correct", which is Υ_{ξ_t} (or $1-\Upsilon_{\xi_t}$ if the orange button is used for "correct").

The binary entropy function, denoted $H_{\mathrm{b}}(p)$, is the entropy of a Bernoulli process with probability of success p. It is a measure of uncertainty about the outcome of sampling from a Bernoulli process and can be computed as follows:

$$H_{\mathrm{b}}(p) = -p\log_2(p) - (1-p)\log_2(1-p) \tag{7.13}$$

The binary entropy function is shown in Figure 7.35. As one would expect the shape of the entropy function holds similar properties than our likelihood Equation 7.12.

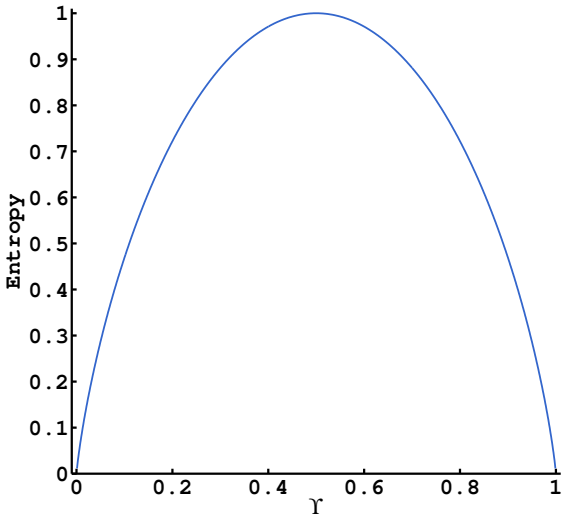


Figure 7.35: The binary entropy function.

This method allows us to rank correctly our task hypothesis with respect to the uncertainty of their estimated models. However, this function will not "sharpen"

when the agent visit more and more states. The only change will result in a better approximation of the Bernoulli process modeling the button presses.

Therefore, this method alone is not enough to estimate which task is the correct one, and we should also measure the uncertainty of this measure of uncertainty. To do so, we propose to use beta distribution, which is the conjugate prior probability distribution for the Bernoulli distributions. A beta distribution encodes a probability distribution over the parameter of the Bernoulli signal model given the amount of evidence available. By comparing between the beta distribution associated to each task, we could expect to find a suitable measure of task confidence. The interested readers may refer to [Montesano 2012] for a practical robotics example using this process.

7.7.5 Discussion

While it is always interesting and useful to formulate proof of algorithm, the restricted assumption used in this section makes it impossible to use this result in practical scenarios. But we note that our experimental results shows that our algorithm can work with fairly good performance on different scenarios using continuous signals as noisy as EEG signals.

Nonetheless, it is important to pursue the theoretical analysis by progressively relaxing some assumptions. Sensible progresses can be achieved quickly, a first step is to consider a non-optimal user making uniform teaching mistakes and worlds with more realistic properties (e.g. with different ratios of optimal state-action pairs). Reaching that level of proof would already offer some guarantees for simple scenarios using discrete states, discrete actions, and symbolic signals, but under more realistic teaching conditions. The next step will be to consider non-symbolic signals, assuming they are sampled from latent distributions of known type.

7.8 Discussion

We reviewed an extensive number of limitations and proposed a number of possible extensions addressing them. The main extensions address the problems of continuous state space, continuous task hypothesis space and unspecified interaction frames. Our results make us envision the use of our algorithm in more complex scenarios more suited to real world robotics applications.

Chapter 8

Discussion and Perspectives

In this chapter, we summarize our contributions and we explicit a number of possible directions for future research in this domain. We particularly advocate for the importance of testing this algorithm with a multitude of users in a variety of tasks. As a final note, we highlight the challenge of learning new meanings and identifying new interaction protocols through practical interaction with humans.

Contributions

Our main contribution is a method allowing a user to start teaching a robot a new task using its own preferred teaching signals. The machine will learn simultaneously which signals are associated to which meaning, as well as identify the task the user wants to solve. Our method consists of generating interpretation hypotheses of the teaching signals with respect to a set of possible tasks. We then assume that the correct task is the one that explains better the history of interaction.

We highlight four important contributions of this thesis: (1) we proposed a new experimental setup to study the co-construction of interaction protocols in collaborative tasks with humans (chapter 3); (2) we presented an algorithm allowing to simultaneously learn a new task from human instructions as well as the mapping between human instruction signals and their meanings (chapter 4); (3) we described a measure of uncertainty on the joint task-signal space that takes into account both the uncertainty inherent to the task, as well as the uncertainty about the signal to meaning mapping (chapter 5); and (4) we showed the applicability of the approach to brain-machine interfaces based on error potentials which could work out of the box without calibration, a long-desired property of this type of systems (chapter 6).

We also proposed a number of possible extensions releasing several assumptions made by our initial algorithm. We address the problems of continuous state space (chapter 7.4), continuous task hypothesis space (chapter 7.5) and unspecified interaction frames (chapter 7.6).

A frame is a generic function

An interaction frame is not limited to the straightforward meaning correspondence we assumed (feedback and guidance), it can include various aspects of timing (e.g. teaching delays, asynchronous signals), social cues (e.g. gaze of the user), and do not always requires the robot to know how to perform a task. We provide below some examples of what a frame might includes.

A task is not always a fixed target We only considered tasks represented as a sparse reward function in a discrete state and action MDP. There is no reason to be limited to this representation of a task, especially to concept of a reaching task. Our algorithm only need to have access to a frame function interpreting each teacher signal given a context and a hypothesized task. Considering the feedback and guidance frame, as soon as the policies associated to each task can be provided to the robot, our algorithm can be applied.

No need for planning skills Although in most of the problem described in this work, the agent needed to know the optimal policy for each task, it is only a specificity of the feedback and guidance frame we considered. For example, in section 7.5 we considered a frame where a teacher provides indication about the absolute direction of objects. Therefore, interpreting a signal with respect to various objects only requires knowing the positions of these objects, without the need to know how to reach each object.

Asynchronous instructions The interaction between the user and the machine would be easier if the robot could act continuously and the human could provide instructions when he deemed necessary. Our pick and place scenario of chapter 4 has been experienced as boring by the users, which had to provide a feedback after each movement of the robot. In some domains, the frequency of actions is too high to afford waiting for a feedback signal between each action. Either the action would be so small that the user would not be able to evaluate it, either the interaction flow and execution time would be dramatically affected by the many pauses in the task execution.

To allow for continuous operation of the robot, asynchronous delivery of signals should be accepted. A potential avenue is to consider a temporal function that distributes a signal event across a subset of previous robot's actions [Hockley 1984, Knox 2009b].

Including social clues Information known to be true for most interaction scenarios can be included in the frame definition. For example, if the user is looking away from the scene, he is less likely to provide correct feedback. Such information can be included to the frame function by decreasing the probability that the user will provide an appropriate signal if the user is looking away. Other potential sources of teaching mistakes include the presence of other persons in the room, or the fact that some objects are hidden to the teacher's eyes.

Studying humans in the loop

It is only by demonstrating that this work can be applied and allows to improve over existing interaction methods that the idea of adaptive and flexible systems will be considered by a larger audience.

Finding application Yet, only the BCI scenario can convincingly be conceived as a potential short term practical application of our method. We believe other applications are yet to identify and is an important direction for the future work in the domain. A good application will allow advertising the potential benefits of adaptive interactive systems, which is to learn from, and interact with, many different users who use different type of signals given their own limitations and preferences.

User studies In the following paragraphs, we highlight the importance of conducting various user studies to evaluate the scalability, efficiency, and acceptability of our method to real world applications.

We mostly used prerecorded datasets. Bu when we performed real time experiments with real subjects, such as the BCI experiments in chapter 6, we noticed that brain signals are sensitive to the protocol of interaction, the duration of the experiment, and to the percentage of errors made by the agent. In addition, people attribute mental states to the agent according to its actions and sometime try to adapt their teaching behavior accordingly. Therefore a fist question to investigate is: *To which extend the behavior of our agent changes the properties of the teaching signals?*

Also, in most real-world applications, the users are told how to interact with the machines. Our algorithm allow a free choice in terms of signal, and having such a choice on some details of the interaction may finally become a disadvantage. An adaptive interface designed on the basis of our work would not be fully operational during the first few interactions (except if other known sources of information are available). Our algorithm needs a few initial steps to adapt to the teaching signals, which may discourage some users. They may rather prefer a more rigid but more intuitive interface. Therefore a second set of questions to investigate is: *Do people want to have an open-ended choice about what signal to use? Would they be more efficient? When is it better to use a calibration procedure?*

Finally, an interesting direction is to consider the same experimental semiotic experiment as described in chapter 3 to build various human-robot interaction scenario. The setup allows to seamlessly use a human or a machine on either of the side of the interaction. A natural extension is therefore to replace the human builder by an agent using our algorithm. But one could also study active teaching algorithm [Cakmak 2012a], by replacing the teacher side by an artificial agent. In addition, the setup also allows biasing, and controlling, some specific aspects of the interaction. For example, we could study how specific agent's behaviors affect the teaching behavior of humans. But also study how an unobservable bias in the interaction, such as one button having no effect, not being delayed, or being displayed at random locations, could affect a human-human interaction. We may finally test our assumption that, in order to succeed in such asymmetric interaction games, participants must be able to use theory of mind and project themselves in different common interaction frames; for example by asking people with specific neurodevelopmental

disorder, such as autism, to participate in similar experiments.

Creating meanings and interaction protocols

We focused on the problem of adaptation to the specificities and limitations of each user's communicative signals. To do so we considered the interaction frame is known by the robot and used by the human. This latter assumption is easily opposable, not two humans will socially behave in the exact same way. Learning the interaction frame seems to be the natural next step, and raises the question of creating novel meanings [Steels 2002], as well as the problem of detecting and understanding new interaction protocols [Mohammad 2010, Lopes 2011]. Advances in this domain may allow a user to progressively provide higher-level instructions throughout the life of a robot. Therefore, creating dynamic and hierarchical learning architectures will play a key role to enable life long learning of interactive skills, and following developmental learning approaches may be the way to go [Lungarella 2003, Demiris 2005, Lopes 2007b].

Bibliography

[Abbeel 2004] Pieter Abbeel and Andrew Y. Ng. *Apprenticeship learning via inverse reinforcement learning.* In Proceedings of the 21st International Conference on Machine Learning (ICML'04), pages 1–8, 2004. (Cited on page 5.)

[Abbeel 2007] Pieter Abbeel, Adam Coates, Morgan Quigley and Andrew Y Ng. *An application of reinforcement learning to aerobatic helicopter flight.* Advances in neural information processing systems, vol. 19, page 1, 2007. (Cited on pages 4 and 6.)

[Akgun 2012] B. Akgun, M. Cakmak, J. W. Yoo and A.L. Thomaz. *Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective.* In Proceedings of the International Conference on Human-Robot Interaction (HRI), 2012. (Cited on pages 23 and 47.)

[Akrour 2011] Riad Akrour, Marc Schoenauer and Michele Sebag. *Preference-based policy learning.* In Machine Learning and Knowledge Discovery in Databases, pages 12–27. Springer, 2011. (Cited on page 23.)

[Akrour 2012] Riad Akrour, Marc Schoenauer and Michèle Sebag. *APRIL: Active preference learning-based reinforcement learning.* In Machine Learning and Knowledge Discovery in Databases, pages 116–131. Springer, 2012. (Cited on page 23.)

[Akrour 2014] Riad Akrour, Marc Schoenauer, Michèle Sebag, Jean-Christophe Souplet*et al.* *Programming by Feedback.* In International Conference on Machine Learning, 2014. (Cited on page 23.)

[Albrecht 2014] S.V. Albrecht and S. Ramamoorthy. *On Convergence and Optimality of Best-Response Learning with Policy Types in Multiagent Systems.* In Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI), Quebec City, Canada, July 2014. (Cited on page 34.)

[Argall 2009] Brenna Argall, Sonia Chernova and Manuela Veloso. *A Survey of Robot Learning from Demonstration.* Robotics and Autonomous Systems, vol. 57, no. 5, pages 469–483, 2009. (Cited on page 3.)

[Barrett 2011a] Samuel Barrett and Peter Stone. *Ad Hoc Teamwork Modeled with Multi-armed Bandits: An Extension to Discounted Infinite Rewards.* In Tenth International Conference on Autonomous Agents and Multiagent Systems - Adaptive Learning Agents Workshop (ALA), May 2011. (Cited on page 34.)

[Barrett 2011b] Samuel Barrett, Peter Stone and Sarit Kraus. *Empirical evaluation of ad hoc teamwork in the pursuit domain.* In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pages

567–574. International Foundation for Autonomous Agents and Multiagent Systems, 2011. (Cited on pages 34 and 35.)

[Barrett 2013a] Samuel Barrett, Noa Agmon, Noam Hazon, Sarit Kraus and Peter Stone. *Communicating with Unknown Teammates*. In AAMAS Adaptive Learning Agents (ALA) Workshop, May 2013. (Cited on pages 34 and 35.)

[Barrett 2013b] Samuel Barrett, Peter Stone, Sarit Kraus and Avi Rosenfeld. *Teamwork with Limited Knowledge of Teammates*. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 2013. (Cited on page 34.)

[Billard 2001] Aude Billard and Maja J Matarić. *Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture*. Robotics and Autonomous Systems, vol. 37, no. 2, pages 145–160, 2001. (Cited on page 4.)

[Blankertz 2010] B Blankertz, S Lemm, M Treder, Haufe S. and Klaus-Robert Müller. *Single-trial analysis and classification of ERP components: A tutorial*. Neuroimage, 2010. (Cited on pages 144 and 178.)

[Blumberg 2002] Bruce Blumberg, Marc Downie, Yuri Ivanov, Matt Berlin, Michael Patrick Johnson and Bill Tomlinson. *Integrated learning for interactive synthetic characters*. In ACM Transactions on Graphics (TOG), volume 21, pages 417–426. ACM, 2002. (Cited on page 8.)

[Bowling 2005] Michael Bowling and Peter McCracken. *Coordination and adaptation in impromptu teams*. In AAAI, volume 5, pages 53–58, 2005. (Cited on page 34.)

[Brafman 2003] R.I. Brafman and M. Tennenholtz. *R-max a general polynomial time algorithm for near-optimal reinforcement learning*. Journal of Machine Learning Research, vol. 3, 2003. (Cited on page 113.)

[Branavan 2011] SRK Branavan, David Silver and Regina Barzilay. *Learning to win by reading manuals in a monte-carlo framework*. In Proceedings of ACL, pages 268–277, 2011. (Cited on page 26.)

[Breazeal 2004] Cynthia Breazeal, Andrew Brooks, Jesse Gray, Guy Hoffman, Cory Kidd, Hans Lee, Jeff Lieberman, Andrea Lockerd and David Chilongo. *Tutelage and collaboration for humanoid robots*. International Journal of Humanoid Robotics, vol. 1, no. 02, pages 315–348, 2004. (Cited on page 22.)

[Brent 1997] M.R. Brent. Computational approaches to language acquisition. MIT Press, 1997. (Cited on page 29.)

[Cakmak 2010] M. Cakmak and A.L. Thomaz. *Optimality of Human Teachers for Robot Learners*. In Proceedings of the International Conference on Development and Learning (ICDL), 2010. (Cited on pages 9 and 24.)

[Cakmak 2012a] Maya Cakmak and Manuel Lopes. *Algorithmic and Human Teaching of Sequential Decision Tasks.* In AAAI, 2012. (Cited on pages 28 and 223.)

[Cakmak 2012b] Maya Cakmak and Andrea L Thomaz. *Designing robot learners that ask good questions.* In Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction, pages 17–24. ACM, 2012. (Cited on pages 25 and 65.)

[Calinon 2007a] S. Calinon, F. Guenter and A. Billard. *On Learning, Representing and Generalizing a Task in a Humanoid Robot.* IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation, vol. 37, no. 2, pages 286–298, 2007. (Cited on page 5.)

[Calinon 2007b] Sylvain Calinon and Aude G Billard. *What is the teacher's role in robot programming by demonstration?: Toward benchmarks for improved learning.* Interaction Studies, vol. 8, no. 3, pages 441–464, 2007. (Cited on page 4.)

[Calinon 2008] Sylvain Calinon. *Robot programming by demonstration.* In Springer handbook of robotics, pages 1371–1394. Springer, 2008. (Cited on page 3.)

[Cangelosi 2002] Angelo Cangelosi and Domenico Parisi. Simulating the evolution of language. Springer London, 2002. (Cited on pages 33 and 48.)

[Cangelosi 2006] A. Cangelosi and T. Riga. *An Embodied Model for Sensorimotor Grounding and Grounding Transfer: Experiments with Epigenetic Robots.* Cognitive Science, vol. 30, no. 4, pages 673–689, 2006. (Cited on page 30.)

[Cangelosi 2010] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori*et al.* *Integration of action and language knowledge: A roadmap for developmental robotics.* Autonomous Mental Development, IEEE Transactions on, vol. 2, no. 3, pages 167–195, 2010. (Cited on pages 29, 30 and 65.)

[Cederborg 2010] Thomas Cederborg, Ming Li, Adrien Baranes and P-Y Oudeyer. *Incremental local online gaussian mixture regression for imitation learning of multiple tasks.* In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 267–274. IEEE, 2010. (Cited on page 5.)

[Cederborg 2011] T. Cederborg and P.Y. Oudeyer. *Imitating operations on internal cognitive structures for language aquisition.* In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 650–657. IEEE, 2011. (Cited on pages 30, 31, 32, 41 and 43.)

[Cederborg 2013] Thomas Cederborg and P-Y Oudeyer. *From Language to Motor Gavagai: Unified Imitation Learning of Multiple Linguistic and Nonlinguistic*

Sensorimotor Skills. Autonomous Mental Development, IEEE Transactions on, vol. 5, no. 3, pages 222–239, 2013. (Cited on page 31.)

[Cederborg 2014a] Thomas Cederborg. *A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation*. PhD thesis, University of Bordeaux, March 2014. (Cited on pages 16, 18, 31, 32, 42 and 78.)

[Cederborg 2014b] Thomas Cederborg and P-Y Oudeyer. *A Social Learning Formalism for Learners Trying to Figure Out What a Teacher Wants Them to Do*. PALADYN, Journal of Behavioral Robotics, 2014. (Cited on pages 16, 18 and 78.)

[Chao 2010] Crystal Chao, Maya Cakmak and Andrea Lockerd Thomaz. *Transparent active learning for robots*. In Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on, pages 317–324. IEEE, 2010. (Cited on page 25.)

[Chavarriaga 2010] R Chavarriaga and JdR Millán. *Learning from EEG error-related potentials in noninvasive brain-computer interfaces*. IEEE Trans Neural Syst Rehabil Eng, vol. 18, no. 4, 2010. (Cited on pages 37, 143, 145, 151, 153, 160 and 182.)

[Chavarriaga 2014] Ricardo Chavarriaga, Aleksander Sobolewski and José del R Millán. *Errare machinale est: The use of error-related potentials in brain-machine interfaces*. Frontiers in Neuroscience, vol. 8, no. EPFL-ARTICLE-200119, 2014. (Cited on pages 37, 43, 143 and 152.)

[Chella 2004] Antonio Chella, Haris Džindo, Ignazio Infantino and Irene Macaluso. *A posture sequence learning system for an anthropomorphic robotic hand*. Robotics and Autonomous Systems, vol. 47, no. 2, pages 143–152, 2004. (Cited on page 4.)

[Chernova 2008a] Sonia Chernova and Manuela Veloso. *Multi-thresholded approach to demonstration selection for interactive robot learning*. In Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on, pages 225–232. IEEE, 2008. (Cited on page 6.)

[Chernova 2008b] Sonia Chernova and Manuela Veloso. *Teaching multi-robot coordination using demonstration of communication and state sharing*. In Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3, pages 1183–1186. International Foundation for Autonomous Agents and Multiagent Systems, 2008. (Cited on page 5.)

[Chernova 2009] S. Chernova and M. Veloso. *Interactive policy learning through confidence-based autonomy*. J. Artificial Intelligence Research, vol. 34, pages 1–25, 2009. (Cited on pages 5, 6, 7, 22, 27 and 83.)

[Clark 1991] Herbert H Clark and Susan E Brennan. *Grounding in communication.* Perspectives on socially shared cognition, vol. 13, no. 1991, pages 127–149, 1991. (Cited on page 46.)

[Clement 2014] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, Manuel Lopes*et al.* *Online Optimization of Teaching Sequences with Multi-Armed Bandits.* In 7th International Conference on Educational Data Mining, 2014. (Cited on page 28.)

[Clouse 1992] Jeffery A Clouse and Paul E Utgoff. *A teaching method for reinforcement learning.* In ML, pages 92–110, 1992. (Cited on page 10.)

[Cohn 2010] Robert Cohn, Michael Maxim, Edmund Durfee and Satinder Singh. *Selecting operator queries using expected myopic gain.* In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, volume 2, pages 40–47. IEEE, 2010. (Cited on page 28.)

[Cohn 2011] Robert Cohn, Edmund Durfee and Satinder Singh. *Comparing action-query strategies in semi-autonomous agents.* In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, pages 1287–1288. International Foundation for Autonomous Agents and Multiagent Systems, 2011. (Cited on page 28.)

[Congedo 2013] Marco Congedo, Alexandre Barachant and Anton Andreev. *A New Generation of Brain-Computer Interface Based on Riemannian Geometry.* arXiv preprint arXiv:1310.8115, 2013. (Cited on page 37.)

[De Boer 2000] B. De Boer. *Self-organization in vowel systems.* Journal of Phonetics, vol. 28, no. 4, pages 441–465, 2000. (Cited on page 30.)

[De Ruiter 2010] Jan Peter De Ruiter, Matthijs L Noordzij, Sarah Newman-Norlund, Roger Newman-Norlund, Peter Hagoort, Stephen C Levinson and Ivan Toni. *Exploring the cognitive infrastructure of communication.* Interaction Studies, vol. 11, no. 1, pages 51–77, 2010. (Cited on pages 33 and 48.)

[Delaherche 2012] Emilie Delaherche, Mohamed Chetouani, Ammar Mahdhaoui, Catherine Saint-Georges, Sylvie Viaux and David Cohen. *Interpersonal Synchrony: A Survey Of Evaluation Methods Across Disciplines.* IEEE Transactions on Affective Computing, vol. 3, no. 3, pages 349–365, 2012. (Cited on page 62.)

[Demiris 2005] Yiannis Demiris and Anthony Dearden. *From motor babbling to hierarchical learning by imitation: a robot developmental pathway.* 2005. (Cited on page 224.)

[Dempster 1977] Arthur P Dempster, Nan M Laird, Donald B Rubin*et al.* *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal statistical Society, vol. 39, no. 1, pages 1–38, 1977. (Cited on page 106.)

[Dissanayake 2001] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte and Michael Csorba. *A solution to the simultaneous localization and map building (SLAM) problem.* Robotics and Automation, IEEE Transactions on, vol. 17, no. 3, pages 229–241, 2001. (Cited on page 36.)

[Dominey 2005] P.F. Dominey and J.D. Boucher. *Learning to talk about events from narrated video in a construction grammar framework.* Artificial Intelligence, vol. 167, no. 1, pages 31–61, 2005. (Cited on page 30.)

[Doshi 2008] F. Doshi and N. Roy. *Spoken language interaction with model uncertainty: an adaptive human–robot interaction system.* Connection Science, vol. 20, no. 4, pages 299–318, 2008. (Cited on page 26.)

[Doucet 2009] Arnaud Doucet and Adam M Johansen. *A tutorial on particle filtering and smoothing: Fifteen years later.* Handbook of Nonlinear Filtering, vol. 12, pages 656–704, 2009. (Cited on pages 193 and 197.)

[Falkenstein 2000] M. Falkenstein, J. Hoormann, S. Christ and J. Hohnsbein. *ERP components on reaction errors and their functional significance: A tutorial.* Biological Psychology, vol. 51, pages 87–107, 2000. (Cited on page 37.)

[Fazli 2009] Siamac Fazli, Florin Popescu, Márton Danóczy, Benjamin Blankertz, Klaus-Robert Müller and Cristian Grozea. *Subject-independent mental state classification in single trials.* Neural networks, vol. 22, no. 9, pages 1305–1312, 2009. (Cited on page 37.)

[Fazli 2011] Siamac Fazli, Márton Danóczy, Jürg Schelldorfer and Klaus-Robert Müller. *l1-Penalized Linear Mixed-Effects Models for BCI.* In Artificial Neural Networks and Machine Learning–ICANN 2011, pages 26–35. Springer, 2011. (Cited on page 37.)

[Fischer 2013] Kerstin Fischer, Katrin Lohan, Joe Saunders, Chrystopher Nehaniv, Britta Wrede and Katharina Rohlfing. *The impact of the contingency of robot feedback on HRI.* In Collaboration Technologies and Systems (CTS), 2013 International Conference on, pages 210–217. IEEE, 2013. (Cited on page 53.)

[Galantucci 2005] Bruno Galantucci. *An experimental study of the emergence of human communication systems.* Cognitive science, vol. 29, no. 5, pages 737–767, 2005. (Cited on pages 33 and 48.)

[Galantucci 2009] B. Galantucci. *Experimental semiotics: A new approach for studying communication as a form of joint action.* Topics in Cognitive Science, vol. 1, no. 2, pages 393–410, 2009. (Cited on pages 32, 33 and 48.)

[Galantucci 2011] Bruno Galantucci and Simon Garrod. *Experimental semiotics: a review.* Frontiers in human neuroscience, vol. 5, 2011. (Cited on pages 33 and 48.)

[Gelman 2003] Andrew Gelman, John B Carlin, Hal S Stern and Donald B Rubin. Bayesian data analysis. CRC press, 2003. (Cited on pages 134, 144 and 194.)

[Gibson 1986] James Jerome Gibson. The ecological approach to visual perception. Psychology Press, 1986. (Cited on page 7.)

[Gil Jones 2006] E Gil Jones, Brett Browning, M Bernardine Dias, Brenna Argall, Manuela Veloso and Anthony Stentz. *Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks.* In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 570–575. IEEE, 2006. (Cited on pages 33 and 34.)

[Goffman 1974] Erving Goffman. Frame analysis: An essay on the organization of experience. Harvard University Press, 1974. (Cited on page 11.)

[Goodwin 1995] Charles Goodwin. *Co-constructing meaning in conversations with an aphasic man.* Research on language and social interaction, vol. 28, no. 3, pages 233–260, 1995. (Cited on pages 60 and 65.)

[Gordon 1993] Neil J Gordon, David J Salmond and Adrian FM Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* In IEE Proceedings F (Radar and Signal Processing), volume 140, pages 107–113. IET, 1993. (Cited on pages 193 and 197.)

[Grave 2013] Kathrin Grave and Sven Behnke. *Learning sequential tasks interactively from demonstrations and own experience.* In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 3237–3243. IEEE, 2013. (Cited on page 22.)

[Griffith 2013] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles Isbell and Andrea L Thomaz. *Policy Shaping: Integrating Human Feedback with Reinforcement Learning.* In Advances in Neural Information Processing Systems, pages 2625–2633, 2013. (Cited on page 22.)

[Griffiths 2012] S. Griffiths, S. Nolfi, G. Morlino, L. Schillingmann, S. Kuehnel, K. Rohlfing and B. Wrede. *Bottom-up learning of feedback in a categorization task.* In Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on, pages 1–6. IEEE, 2012. (Cited on pages 33, 48 and 56.)

[Grizou 2013a] Jonathan Grizou, Iñaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer*et al.* *Interactive Task Estimation From Unlabelled Teaching Signals.* In International Workshop on Human-Machine Systems, Cyborgs and Enhancing Devices, 2013. (Cited on page 19.)

[Grizou 2013b] Jonathan Grizou, Inaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer*et al.* *Zero-calibration BMIs for sequential tasks using error-related potentials.* In IROS 2013 Workshop on Neuroscience and Robotics, 2013. (Cited on page 19.)

[Grizou 2013c] Jonathan Grizou, Manuel Lopes and Pierre-Yves Oudeyer. *Robot learning simultaneously a task and how to interpret human instructions.* In Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on, pages 1–8. IEEE, 2013. (Cited on pages 19 and 70.)

[Grizou 2014a] Jonathan Grizou, Iñaki Iturrate, Luis Montesano, Pierre-Yves Oudeyer and Manuel Lopes. *Interactive Learning from Unlabeled Instructions.* In Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014. (Cited on pages 19 and 111.)

[Grizou 2014b] Jonathan Grizou, Inaki Iturrate, Luis Montesano, Pierre-Yves Oudeyer, Manuel Lopes*et al. Calibration-Free BCI Based Control.* International AAAI Conference on Artificial Intelligence, pages 1–8, 2014. (Cited on pages 19 and 164.)

[Grizou 2014c] Jonathan Grizou, Manuel Lopes, Pierre-Yves Oudeyer*et al. Robot Learning from Unlabelled Teaching Signals.* In HRI 2014 Pioneers Workshop, 2014. (Cited on page 19.)

[Grollman 2007a] Daniel H Grollman and Odest Chadwicke Jenkins. *Dogged Learning for Robots.* In ICRA, pages 2483–2488, 2007. (Cited on page 23.)

[Grollman 2007b] Daniel H Grollman and Odest Chadwicke Jenkins. *Learning robot soccer skills from demonstration.* In Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on, pages 276–281. IEEE, 2007. (Cited on page 4.)

[Guenter 2007] Florent Guenter, Micha Hersch, Sylvain Calinon and Aude Billard. *Reinforcement learning for imitating constrained reaching movements.* Advanced Robotics, vol. 21, no. 13, pages 1521–1544, 2007. (Cited on page 4.)

[Heath 2012] Scott Heath, Ruth Schulz, David Ball and Janet Wiles. *Long summer days: grounded learning of words for the uneven cycles of real world events.* Autonomous Mental Development, IEEE Transactions on, vol. 4, no. 3, pages 192–203, 2012. (Cited on page 30.)

[Heckmann 2009] M. Heckmann, H. Brandl, J. Schmuedderich, X. Domont, B. Bolder, I. Mikhailova, H. Janssen, M. Gienger, A. Bendig, T. Rodemann*et al. Teaching a humanoid robot: Headset-free speech interaction for audio-visual association learning.* In Robot and Human Interactive Communication, RO-MAN., pages 422–427. IEEE, 2009. (Cited on page 30.)

[Hester 2013] Todd Hester, Manuel Lopes and Peter Stone. *Learning exploration strategies in model-based reinforcement learning.* In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, pages 1069–1076. International Foundation for Autonomous Agents and Multiagent Systems, 2013. (Cited on page 186.)

[Hockley 1984] William E Hockley. *Analysis of response time distributions in the study of cognitive processes.* Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 10, no. 4, page 598, 1984. (Cited on page 222.)

[Hovland 1996] Geir E Hovland, Pavan Sikka and Brenan J McCarragher. *Skill acquisition from human demonstration using a hidden markov model.* In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, volume 3, pages 2706–2711. Ieee, 1996. (Cited on page 5.)

[Hyon 2007] S Hyon, Joshua G Hale and Gordon Cheng. *Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces.* Robotics, IEEE Transactions on, vol. 23, no. 5, pages 884–898, 2007. (Cited on page 4.)

[Ijspeert 2002a] Auke Jan Ijspeert, Jun Nakanishi and Stefan Schaal. *Learning rhythmic movements by demonstration using nonlinear oscillators.* In Proceedings of the ieee/rsj int. conference on intelligent robots and systems (iros2002), numéro BIOROB-CONF-2002-003, pages 958–963, 2002. (Cited on page 5.)

[Ijspeert 2002b] Auke Jan Ijspeert, Jun Nakanishi and Stefan Schaal. *Movement imitation with nonlinear dynamical systems in humanoid robots.* In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, volume 2, pages 1398–1403. IEEE, 2002. (Cited on page 4.)

[Isbell 2001] Charles Isbell, Christian R Shelton, Michael Kearns, Satinder Singh and Peter Stone. *A social reinforcement learning agent.* In Proceedings of the fifth international conference on Autonomous agents, pages 377–384. ACM, 2001. (Cited on page 8.)

[Iturrate 2010] Inaki Iturrate, Luis Montesano and Javier Minguez. *Single trial recognition of error-related potentials during observation of robot operation.* In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pages 4181–4184. IEEE, 2010. (Cited on pages 160 and 182.)

[Iturrate 2013a] I. Iturrate, L. Montesano and J. Minguez. *Shared-control brain-computer interface for a two dimensional reaching task using EEG error-related potentials.* In Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2013. (Cited on page 37.)

[Iturrate 2013b] I. Iturrate, L. Montesano and J. Minguez. *Task-dependent signal variations in EEG error-related potentials for brain-computer interfaces.* Journal of Neural Engineering, vol. 10, no. 2, 2013. (Cited on pages 37, 143, 145, 151, 153 and 180.)

[Judah 2010] K. Judah, S. Roy, A. Fern and T.G. Dietterich. *Reinforcement Learning Via Practice and Critique Advice.* In Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10), 2010. (Cited on page 22.)

[Judah 2012] Kshitij Judah, Alan Fern and Thomas G Dietterich. *Active Imitation Learning via Reduction to IID Active Learning.* In AAAI Fall Symposium: Robots Learning Interactively from Human Teachers, 2012. (Cited on page 27.)

[Kailath 1967] T. Kailath. *The divergence and Bhattacharyya distance measures in signal selection.* IEEE Trans. Commun. Technol., vol. 15, no. 3, pages 52–60, 1967. (Cited on page 178.)

[Kaochar 2011] T. Kaochar, R. Peralta, C. Morrison, I. Fasel, T. Walsh and P. Cohen. *Towards Understanding How Humans Teach Robots.* User Modeling, Adaption and Personalization, pages 347–352, 2011. (Cited on pages 23 and 25.)

[Kaplan 2002] Frédéric Kaplan, Pierre-Yves Oudeyer, Enikö Kubinyi and Adám Miklósi. *Robotic clicker training.* Robotics and Autonomous Systems, vol. 38, no. 3, pages 197–206, 2002. (Cited on pages 8, 22 and 83.)

[Kaplan 2006] F. Kaplan and V.V. Hafner. *The challenges of joint attention.* Interaction Studies, vol. 7, no. 2, pages 135–169, 2006. (Cited on page 30.)

[Kaplan 2008] F. Kaplan, P.Y. Oudeyer and B. Bergen. *Computational models in the debate over language learnability.* infant and child development, vol. 17, no. 1, pages 55–80, 2008. (Cited on page 29.)

[Kim 2012] J. Kim and R.J. Mooney. *Unsupervised PCFG induction for grounded language learning with highly ambiguous supervision.* In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning, EMNLP-CoNLL, volume 12, 2012. (Cited on page 26.)

[Kindermans 2012a] Pieter-Jan Kindermans, David Verstraeten and Benjamin Schrauwen. *A bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI.* PloS one, vol. 7, no. 4, page e33758, January 2012. (Cited on pages 38, 39, 41 and 43.)

[Kindermans 2012b] PJ Kindermans and Hannes Verschore. *A P300 BCI for the Masses: Prior Information Enables Instant Unsupervised Spelling.* In NIPS, pages 1–9, 2012. (Cited on page 38.)

[Kindermans 2014a] Pieter-Jan Kindermans, Martijn Schreuder, Benjamin Schrauwen, Klaus-Robert Müller and Michael Tangermann. *True Zero-Training Brain-Computer Interfacing–An Online Study.* PloS one, vol. 9, no. 7, page e102504, 2014. (Cited on pages 38, 39 and 43.)

[Kindermans 2014b] Pieter-Jan Kindermans, Michael Tangermann, Klaus-Robert Müller and Benjamin Schrauwen. *Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller*. Journal of Neural Engineering, vol. 11, no. 3, page 035005, 2014. (Cited on pages 38 and 41.)

[Knox 2009a] W Bradley Knox, Ian Fasel and Peter Stone. *Design Principles for Creating Human-Shapable Agents*. In AAAI Spring Symposium: Agents that Learn from Human Teachers, pages 79–86, 2009. (Cited on page 24.)

[Knox 2009b] W.B. Knox and P. Stone. *Interactively shaping agents via human reinforcement: The TAMER framework*. In Proceedings of the fifth international conference on Knowledge capture, pages 9–16. ACM, 2009. (Cited on pages 9, 14, 22, 26, 77, 83 and 222.)

[Knox 2010] W Bradley Knox and Peter Stone. *Combining manual feedback with subsequent MDP reward signals for reinforcement learning*. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, pages 5–12. International Foundation for Autonomous Agents and Multiagent Systems, 2010. (Cited on page 22.)

[Knox 2012] W. Knox, B. Glass, B. Love, W. Maddox and P. Stone. *How humans teach agents: A new experimental perspective*. International Journal of Social Robotics, Special Issue on Robot Learning from Demonstration, 2012. (Cited on page 23.)

[Kolter 2007] J Zico Kolter, Pieter Abbeel and Andrew Y Ng. *Hierarchical apprenticeship learning with application to quadruped locomotion*. In Advances in Neural Information Processing Systems, pages 769–776, 2007. (Cited on page 10.)

[Kolter 2009] J Zico Kolter and Andrew Y Ng. *Near-Bayesian exploration in polynomial time*. In International Conference on Machine Learning. ACM, 2009. (Cited on page 113.)

[Kopp 2010] Stefan Kopp. *Social resonance and embodied coordination in face-to-face conversation with artificial interlocutors*. Speech Communication, vol. 52, no. 6, pages 587–597, 2010. (Cited on page 53.)

[Kose-Bagci 2008] Hatice Kose-Bagci, Kerstin Dautenhahn and Chrystopher L Nehaniv. *Emergent dynamics of turn-taking interaction in drumming games with a humanoid robot*. In Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on, pages 346–353. IEEE, 2008. (Cited on page 7.)

[Lachenbruch 1975] Peter A Lachenbruch. Discriminant analysis. Wiley Online Library, 1975. (Cited on page 84.)

[Lee 2000] Chulhee Lee and Euisun Choi. *Bayes error evaluation of the Gaussian ML classifier*. Geoscience and Remote Sensing, IEEE Transactions on, vol. 38, no. 3, pages 1471–1475, 2000. (Cited on page 178.)

[Lin 2012] Yun Lin, Shaogang Ren, Matthew Clevenger and Yu Sun. *Learning grasping force from demonstration*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1526–1531. IEEE, 2012. (Cited on page 4.)

[Lockerd 2004] Andrea Lockerd and Cynthia Breazeal. *Tutelage and socially guided robot learning*. In Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, volume 4, pages 3475–3480. IEEE, 2004. (Cited on page 10.)

[Loftin 2014] Robert Loftin, Bei Peng, James MacGlashan, Michael L Littman, Matthew E Taylor, Jeff Huang and David L Roberts. *Learning Something from Nothing: Leveraging Implicit Human Feedback Strategies*. 2014. (Cited on pages 24 and 26.)

[Lohse 2010] Manja Lohse. *Investigating the influence of situations and expectations on user behavior: empirical analyses in human-robot interaction*. 2010. (Cited on page 47.)

[Lopes 2005] Manuel Lopes and José Santos-Victor. *Visual learning by imitation with motor representations*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 35, no. 3, pages 438–449, 2005. (Cited on page 4.)

[Lopes 2007a] Manuel Lopes, Francisco S. Melo and Luis Montesano. *Affordance-based imitation learning in robots*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07), pages 1015–1021, USA, Nov 2007. (Cited on page 7.)

[Lopes 2007b] Manuel Lopes and José Santos-Victor. *A developmental roadmap for learning by imitation in robots*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, no. 2, pages 308–321, 2007. (Cited on page 224.)

[Lopes 2009a] Manuel Lopes, Francisco S Melo, Ben Kenward and José Santos-Victor. *A computational model of social-learning mechanisms*. Adaptive Behavior, vol. 17, no. 6, pages 467–483, 2009. (Cited on page 6.)

[Lopes 2009b] Manuel Lopes, Francisco S. Melo and Luis Montesano. *Active Learning for Reward Estimation in Inverse Reinforcement Learning*. In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, ECML PKDD '09, pages 31–46, 2009. (Cited on pages 5, 22, 28, 113 and 131.)

[Lopes 2010] Manuel Lopes, Francisco Melo, Luis Montesano and Jos'e Santos-Victor. *Abstraction Levels for Robotic Imitation: Overview and Computational Approaches.* In Olivier Sigaud and Jan Peters, editeurs, From Motor to Interaction Learning in Robots, volume 264 of *Studies in Computational Intelligence*, pages 313–355. Springer, 2010. (Cited on page 3.)

[Lopes 2011] M. Lopes, T. Cederborg and P.-Y. Oudeyer. *Simultaneous acquisition of task and feedback models.* In Development and Learning (ICDL), 2011 IEEE International Conference on, volume 2, pages 1 –7, aug. 2011. (Cited on pages 26, 27, 131 and 224.)

[Lopes 2014] Manuel Lopes and Luis Montesano. *Active Learning for Autonomous Intelligent Agents: Exploration, Curiosity, and Interaction.* arXiv preprint arXiv:1403.1497, 2014. (Cited on page 27.)

[Lotte 2007] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, Bruno Arnaldi*et al.* *A review of classification algorithms for EEG-based brain–computer interfaces.* Journal of neural engineering, vol. 4, 2007. (Cited on pages 144 and 178.)

[Lu 2009] Shijian Lu, Cuntai Guan and Haihong Zhang. *Unsupervised brain computer interface based on intersubject information and online adaptation.* Neural Systems and Rehabilitation Engineering, IEEE Transactions on, vol. 17, no. 2, pages 135–145, 2009. (Cited on page 37.)

[Lungarella 2003] Max Lungarella, Giorgio Metta, Rolf Pfeifer and Giulio Sandini. *Developmental robotics: a survey.* Connection Science, vol. 15, no. 4, pages 151–190, 2003. (Cited on page 224.)

[Lyon 2012] C. Lyon, C.L. Nehaniv and J. Saunders. *Interactive Language Learning by Robots: The Transition from Babbling to Word Forms.* PloS one, vol. 7, no. 6, page e38236, 2012. (Cited on page 30.)

[Maclin 2005] Richard Maclin, Jude Shavlik, Lisa Torrey, Trevor Walker and Edward Wild. *Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression.* In Proceedings of the National Conference on Artificial intelligence, volume 20, page 819. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005. (Cited on page 10.)

[Mangin 2013] Olivier Mangin and Pierre-Yves Oudeyer. *Learning semantic components from subsymbolic multimodal perception.* In Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on, pages 1–7. IEEE, 2013. (Cited on page 35.)

[Mann 1947] Henry B Mann and Donald R Whitney. *On a test of whether one of two random variables is stochastically larger than the other.* The annals of mathematical statistics, pages 50–60, 1947. (Cited on page 198.)

[Mason 2011] Martin Mason and Manuel Lopes. *Robot Self-Initiative and Personalization by Learning through Repeated Interactions.* In 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI'11), 2011. (Cited on pages 22 and 23.)

[Massera 2010] Gianluca Massera, Elio Tuci, Tomassino Ferrauto and Stefano Nolfi. *The Facilitatory Role of Linguistic Instructions on Developing Manipulation Skills.* IEEE Computational Intelligence Magazine, vol. 5, no. 3, pages 33–42, 2010. (Cited on page 29.)

[Mataric 2000] Maja J Mataric. *Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics.* In Imitation in animals and artifacts. Citeseer, 2000. (Cited on page 5.)

[Melo 2010] Francisco S Melo and Manuel Lopes. *Learning from demonstration using mdp induced metrics.* In Machine Learning and Knowledge Discovery in Databases, pages 385–401. Springer, 2010. (Cited on page 27.)

[Melo 2013] Francisco Melo and Manuel Lopes. *Multi-class Generalized Binary Search for Active Inverse Reinforcement Learning.* arXiv preprint arXiv:1301.5488, 2013. (Cited on page 28.)

[Michalowski 2007] Marek P Michalowski, Selma Sabanovic and Hideki Kozima. *A dancing robot for rhythmic social interaction.* In Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on, pages 89–96. IEEE, 2007. (Cited on page 53.)

[Millán 2010] JdR. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, C. Neuper, K.-R. Müller and D. Mattia. *Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges.* Front Neurosci, vol. 4, 2010. (Cited on page 37.)

[Minsky 1974] Marvin Minsky. *A framework for representing knowledge.* 1974. (Cited on page 11.)

[Mohammad 2009a] Yasser Mohammad and Toyoaki Nishida. *Constrained motif discovery in time series.* New Generation Computing, vol. 27, no. 4, pages 319–346, 2009. (Cited on page 36.)

[Mohammad 2009b] Yasser Mohammad, Toyoaki Nishida and Shogo Okada. *Unsupervised simultaneous learning of gestures, actions and their associations for human-robot interaction.* In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 2537–2544. IEEE, 2009. (Cited on page 36.)

[Mohammad 2010] Yasser Mohammad and Toyoaki Nishida. *Learning interaction protocols using augmented baysian networks applied to guided navigation.* In

Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 4119–4126. IEEE, 2010. (Cited on pages 36 and 224.)

[Montesano 2008] Luis Montesano, Manuel Lopes, Alexandre Bernardino and José Santos-Victor. *Learning object affordances: From sensory–motor coordination to imitation.* Robotics, IEEE Transactions on, vol. 24, no. 1, pages 15–26, 2008. (Cited on page 7.)

[Montesano 2009] Luis Montesano and Manuel Lopes. *Learning grasping affordances from local visual descriptors.* In Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on, pages 1–6. IEEE, 2009. (Cited on page 5.)

[Montesano 2012] Luis Montesano and Manuel Lopes. *Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions.* Robotics and Autonomous Systems, vol. 60, no. 3, pages 452–462, 2012. (Cited on pages 27 and 219.)

[Morlino 2010] Giuseppe Morlino, Claudia Gianelli, Anna M Borghi and Stefano Nolfi. *Developing the Ability to Manipulate Objects: A Comparative Study with Human and Artificial Agents.* In Processings of the Tenth International Conference on Epigenetic Robotics, pages 169–170, 2010. (Cited on page 48.)

[Natarajan 2011] Sriraam Natarajan, Saket Joshi, Prasad Tadepalli, Kristian Kersting and Jude Shavlik. *Imitation learning in relational domains: A functional-gradient boosting approach.* In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two, pages 1414–1420. AAAI Press, 2011. (Cited on page 4.)

[Nehaniv 2000] Chrystopher L Nehaniv and Kerstin Dautenhahn. *OF HUMMINGBIRDS AND HELICOPTERS: AN ALGEBRAIC.* Interdisciplinary Approaches to Robot Learning, vol. 24, page 136, 2000. (Cited on page 7.)

[Nehaniv 2002] Chrystopher L Nehaniv and Kerstin Dautenhahn. *The Correspondence Problem.* Imitation in animals and artifacts, page 41, 2002. (Cited on pages 4 and 8.)

[Ng 2000] Andrew Y Ng, Stuart J Russell*et al.* *Algorithms for inverse reinforcement learning.* In Icml, pages 663–670, 2000. (Cited on page 5.)

[Nguyen 2012] Sao Mai Nguyen and Pierre-Yves Oudeyer. *Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner.* Paladyn Journal of Behavioural Robotics, vol. 3, no. 3, pages 136–146, 2012. (Cited on page 7.)

[Nickerson 1998] Raymond S Nickerson. *Confirmation bias: A ubiquitous phenomenon in many guises.* Review of general psychology, vol. 2, no. 2, page 175, 1998. (Cited on page 62.)

[Nicolescu 2003] M.N. Nicolescu and M.J. Mataric. *Natural methods for robot task learning: Instructive demonstrations, generalization and practice.* In Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pages 241–248. ACM, 2003. (Cited on page 22.)

[Ninio 1996] Anat Ninio and Catherine E Snow. Pragmatic development. Westview Press, 1996. (Cited on pages 11 and 46.)

[Nouri 2010] A. Nouri and M.L. Littman. *Dimension reduction and its application to model-based exploration in continuous spaces.* Machine learning, vol. 81, no. 1, pages 85–98, 2010. (Cited on page 186.)

[Ochs 1979] Elinor Ochs, Bambi B Schieffelin and Martha Platt. *Propositions across utterances and speakers.* Developmental pragmatics, pages 251–268, 1979. (Cited on pages 60 and 65.)

[Orsborn 2012] A.L. Orsborn, S Dangi, H. G. Moorman and J. M. Carmena. *Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions.* IEEE Trans. on neural systems and rehabilitation engineering, vol. 20, no. 4, 2012. (Cited on page 38.)

[Oudeyer 2006] P.Y. Oudeyer and J.R. Hurford. Self-organization in the evolution of speech. Oxford University Press Oxford, 2006. (Cited on page 30.)

[Pardowitz 2005] Michael Pardowitz, Raoul Zollner and Rüdiger Dillmann. *Learning sequential constraints of tasks from user demonstrations.* In Humanoid Robots, 2005 5th IEEE-RAS International Conference on, pages 424–429. IEEE, 2005. (Cited on page 4.)

[Pardowitz 2007] Michael Pardowitz, Steffen Knoop, Ruediger Dillmann and RD Zollner. *Incremental learning of tasks from user demonstrations, past experiences, and vocal comments.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, no. 2, pages 322–332, 2007. (Cited on page 22.)

[Pickering 2004] Martin J Pickering and Simon Garrod. *Toward a mechanistic psychology of dialogue.* Behavioral and brain sciences, vol. 27, no. 2, pages 169–189, 2004. (Cited on page 46.)

[Pilarski 2012] Patrick M Pilarski and Richard S Sutton. *Between Instruction and Reward: Human-Prompted Switching.* In AAAI Fall Symposium: Robots Learning Interactively from Human Teachers, 2012. (Cited on page 22.)

[Pitsch 2013] Karola Pitsch, Anna-Lisa Vollmer and Manuel Muhlig. *Robot feedback shapes the tutor's presentation How a robot's online gaze strategies lead to micro-adaptation of the human's conduct.* Interaction Studies, vol. 14, no. 2, 2013. (Cited on page 53.)

[Platt 1999] J. Platt*et al.* *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.* Advances in large margin classifiers, vol. 10, no. 3, pages 61–74, 1999. (Cited on page 99.)

[Polich 1997] J Polich. *On the relationship between EEG and P300 : individual differences, aging, and ultradian rhythms.* International Journal of Psychophysiology, vol. 26, no. 1-3, 1997. (Cited on page 37.)

[Polich 2003] John Polich. Theoretical overview of p3a and p3b. Springer, 2003. (Cited on page 38.)

[Pomerleau 1991] Dean A Pomerleau. *Efficient training of artificial neural networks for autonomous navigation.* Neural Computation, vol. 3, no. 1, pages 88–97, 1991. (Cited on page 5.)

[Quine 1964] W.V.O. Quine. Word and object, volume 4. MIT press, 1964. (Cited on pages 29 and 46.)

[Regan 2011] Kevin Regan and Craig Boutilier. *Eliciting additive reward functions for Markov decision processes.* In IJCAI Proceedings-International Joint Conference on Artificial Intelligence, volume 22, page 2159, 2011. (Cited on page 28.)

[Rohlfing 2006] K.J. Rohlfing, J. Fritsch, B. Wrede and T. Jungmann. *How can multimodal cues from child-directed interaction reduce learning complexity in robots?* Advanced Robotics, vol. 20, no. 10, pages 1183–1199, 2006. (Cited on page 30.)

[Rohlfing 2013] Katharina Rohlfing, Juana Salas Poblete and Joublin Frank. *Learning new words in unfamiliar frames from direct and indirect teaching.* 2013. (Cited on pages 11 and 46.)

[Rouanet 2013] Pierre Rouanet, Pierre-Yves Oudeyer, Fabien Danieau and David Filliat. *The impact of human–robot interfaces on the learning of visual objects.* Robotics, IEEE Transactions on, vol. 29, no. 2, pages 525–541, 2013. (Cited on page 23.)

[Rovatsos 2001] Michael Rovatsos. *Interaction frames for artificial agents.* 2001. (Cited on pages 11 and 12.)

[Roy 2002] D. Roy and A. Pentland. *Learning words from sights and sounds: A computational model.* Cognitive science, vol. 26, pages 113–146, 2002. (Cited on page 30.)

[Roy 2005] D. Roy. *Semiotic schemas: A framework for grounding language in action and perception.* Artificial Intelligence, vol. 167, no. 1, pages 170–205, 2005. (Cited on page 29.)

[Sakoe 1978] H. Sakoe and S. Chiba. *Dynamic programming algorithm optimization for spoken word recognition.* Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 26, no. 1, pages 43–49, 1978. (Cited on page 98.)

[Saunders 2006] Joe Saunders, Chrystopher L Nehaniv and Kerstin Dautenhahn. *Teaching robots by moulding behavior and scaffolding the environment.* In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, pages 118–125. ACM, 2006. (Cited on page 5.)

[Schaal 1998] Stefan Schaal and Dagmar Sternad. *Programmable pattern generators.* In 3rd International Conference on Computational Intelligence in Neuroscience, pages 48–51. Citeseer, 1998. (Cited on page 5.)

[Schaal 1999] Stefan Schaal. *Is imitation learning the route to humanoid robots?* Trends in cognitive sciences, vol. 3, no. 6, pages 233–242, 1999. (Cited on page 3.)

[Schettini 2014] F Schettini, F Aloise, P Aricò, S Salinari, D Mattia and F Cincotti. *Self-calibration algorithm in an asynchronous P300-based brain–computer interface.* Journal of Neural Engineering, vol. 11, no. 3, page 035004, 2014. (Cited on page 37.)

[Schillingmann 2011] L. Schillingmann, P. Wagner, C. Munier, B. Wrede and K. Rohlfing. *Acoustic Packaging and the Learning of Words.* Frontiers in Computational Neuroscience, vol. 5, page 20, 2011. (Cited on page 30.)

[Schulz 2010] Ruth Schulz, Arren Glover, Gordon Wyeth and Janet Wiles. *Robots, communication, and language: An overview of the Lingodroid project.* In Australasian Conference on Robotics and Automation (ACRA), Brisbane, Australia. Citeseer, 2010. (Cited on page 30.)

[Schulz 2011] Ruth Schulz, Gordon Wyeth and Janet Wiles. *Lingodroids: socially grounding place names in privately grounded cognitive maps.* Adaptive Behavior, page 1059712311421437, 2011. (Cited on page 30.)

[Settles 2010] Burr Settles. *Active learning literature survey.* University of Wisconsin, Madison, 2010. (Cited on page 109.)

[Smart 2002] William D Smart and Leslie Pack Kaelbling. *Effective reinforcement learning for mobile robots.* In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, volume 4, pages 3404–3410. IEEE, 2002. (Cited on page 10.)

[Smith 1990] Randall Smith, Matthew Self and Peter Cheeseman. *Estimating uncertain spatial relationships in robotics.* In Autonomous robot vehicles, pages 167–193. Springer, 1990. (Cited on page 36.)

[Smith 2008] L. Smith and C. Yu. *Infants rapidly learn word-referent mappings via cross-situational statistics.* Cognition, vol. 106, no. 3, pages 1558–1568, 2008. (Cited on page 29.)

[Spranger 2012a] Michael Spranger. *The co-evolution of basic spatial terms and categories.* Experiments in cultural language evolution. Benjamins, Amsterdam, pages 111–141, 2012. (Cited on page 30.)

[Spranger 2012b] Michael Spranger and Luc Steels. *Emergent functional grammar for space.* Experiments in Cultural Language Evolution. John Benjamins, Amsterdam, 2012. (Cited on page 30.)

[Spranger 2013] Michael Spranger. *Grounded lexicon acquisition - Case studies in spatial language.* In Development and Learning and Epigenetic Robotics (ICDL, 2013 IEEE Thrid Joint International Conference on), pages 1–6. IEEE, 2013. (Cited on page 30.)

[Sprangler 2013] Michael Sprangler. In Advances in Artificial Life, ECAL, volume 12, pages 1999–1205, 2013. (Cited on page 30.)

[Steels 2002] L. Steels and F. Kaplan. *Aibos first words: The social learning of language and meaning.* Evolution of communication, vol. 4, no. 1, pages 3–32, 2002. (Cited on pages 12, 29, 30 and 224.)

[Steels 2003] L. Steels. *Evolving grounded communication for robots.* Trends in cognitive sciences, vol. 7, no. 7, pages 308–312, 2003. (Cited on page 29.)

[Steels 2007] Luc Steels and Martin Loetzsch. *Perspective Alignment in Spatial Language.* In Kenny R. Coventry, Thora Tenbrink and John. A Bateman, editeurs, Spatial Language and Dialogue. Oxford University Press, 2007. to appear. (Cited on pages 29 and 30.)

[Steels 2008a] L. Steels and M. Spranger. *Can body language shape body image.* Artificial Life XI, vol. 11, pages 577–584, 2008. (Cited on pages 29 and 30.)

[Steels 2008b] L. Steels and M. Spranger. *The robot in the mirror.* Connection Science, vol. 20, no. 4, pages 337–358, 2008. (Cited on page 29.)

[Steels 2012a] L. Steels. *Grounding Language through Evolutionary Language Games.* Language Grounding in Robots, pages 1–22, 2012. (Cited on pages 29 and 30.)

[Steels 2012b] Luc Steels. Experiments in cultural language evolution, volume 3. John Benjamins Publishing, 2012. (Cited on pages 33 and 48.)

[Steels 2012c] Luc Steels, Joachim Beule and Pieter Wellens. *Fluid Construction Grammar on Real Robots.* Language Grounding in Robots, pages 195–213, 2012. (Cited on page 30.)

[Stone 2010a] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein*et al.* *Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination.* In AAAI, 2010. (Cited on pages 33 and 34.)

[Stone 2010b] Peter Stone and Sarit Kraus. *To teach or not to teach?: decision making under uncertainty in ad hoc teams.* In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, pages 117–124. International Foundation for Autonomous Agents and Multiagent Systems, 2010. (Cited on page 34.)

[Stone 2013] Peter Stone, Gal A. Kaminka, Sarit Kraus, Jeffrey R. Rosenschein and Noa Agmon. *Teaching and leading an ad hoc teammate: Collaboration without pre-coordination.* Artificial Intelligence, vol. 203, pages 35–65, October 2013. (Cited on page 34.)

[Sugita 2005] Y. Sugita and J. Tani. *Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes.* Adaptive Behavior, vol. 13, no. 1–2, pages 33–52, 2005. (Cited on pages 29 and 30.)

[Sutton 1998] R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction, volume 28. Cambridge Univ Press, 1998. (Cited on pages 8, 97, 133 and 185.)

[Taylor 2011] Matthew E Taylor, Halit Bener Suay and Sonia Chernova. *Integrating reinforcement learning with human demonstrations of varying ability.* In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems, 2011. (Cited on page 23.)

[Tegin 2009] Johan Tegin, Staffan Ekvall, Danica Kragic, Jan Wikander and Boyko Iliev. *Demonstration-based learning and control for automatic grasping.* Intelligent Service Robotics, vol. 2, no. 1, pages 23–30, 2009. (Cited on page 4.)

[Theofilis 2013] Konstantinos Theofilis, Katrin Solveig Lohan, Chrystopher L Nehaniv, Kerstin Dautenhahn and Britta Werde. *Temporal emphasis for goal extraction in task demonstration to a humanoid robot by naive users.* In Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on, pages 1–6. IEEE, 2013. (Cited on page 7.)

[Thomaz 2006] Andrea Lockerd Thomaz and Cynthia Breazeal. *Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance.* In AAAI, volume 6, pages 1000–1005, 2006. (Cited on page 24.)

[Thomaz 2008] A.L. Thomaz and C. Breazeal. *Teachable robots: Understanding human teaching behavior to build more effective robot learners.* Artificial Intelligence, vol. 172, no. 6-7, pages 716–737, 2008. (Cited on pages 9, 16, 22, 23 and 24.)

[Thrun 2002] Sebastian Thrun. *Particle filters in robotics.* In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pages 511–518. Morgan Kaufmann Publishers Inc., 2002. (Cited on pages 193 and 197.)

[Tomasello 2009] Michael Tomasello. The cultural origins of human cognition. Harvard University Press, 2009. (Cited on pages 11 and 46.)

[Torrey 2013] Lisa Torrey and Matthew Taylor. *Teaching on a budget: Agents advising agents in reinforcement learning.* In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, pages 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems, 2013. (Cited on page 28.)

[Vidaurre 2010] C. Vidaurre and B. Blankertz. *Towards a cure for BCI illiteracy.* Brain Topogr, vol. 23, no. 2, pages 194–198, 2010. (Cited on page 37.)

[Vidaurre 2011] C Vidaurre, M Kawanabe, P von Bünau, B Blankertz and KR Müller. *Toward unsupervised adaptation of LDA for brain-computer interfaces.* IEEE Trans Biomed Eng, vol. 58(3), 2011. (Cited on page 37.)

[Vien 2013] Ngo Anh Vien, Wolfgang Ertel and Tae Choong Chung. *Learning via human feedback in continuous state and action spaces.* Applied intelligence, vol. 39, no. 2, pages 267–278, 2013. (Cited on page 26.)

[Vollmer 2014a] Anna-Lisa Vollmer, Jonathan Grizou, Manuel Lopes, Katharina Rohlfing and Pierre-Yves Oudeyer. *Studying the Co-Construction of Interaction Protocols in Collaborative Tasks with Humans.* In Development and Learning and Epigenetic Robotics (ICDL), 2014 IEEE Fourth Joint International Conference on, 2014. (Cited on pages 19 and 45.)

[Vollmer 2014b] Anna-Lisa Vollmer, Manuel Mühlig, Jochen J Steil, Karola Pitsch, Jannik Fritsch, Katharina J Rohlfing and Britta Wrede. *Robots Show Us How to Teach Them: Feedback from Robots Shapes Tutoring Behavior during Action Learning.* PloS one, vol. 9, no. 3, page e91349, 2014. (Cited on pages 53 and 65.)

[Watkins 1992] Christopher JCH Watkins and Peter Dayan. *Q-learning.* Machine learning, vol. 8, no. 3, pages 279–292, 1992. (Cited on page 185.)

[Weinberg 2006] Gil Weinberg and Scott Driscoll. *Robot-human interaction with an anthropomorphic percussionist.* In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1229–1232. ACM, 2006. (Cited on page 7.)

[Wilson 2012] Aaron Wilson, Alan Fern and Prasad Tadepalli. *A bayesian approach for policy learning from trajectory preference queries.* In Advances in Neural Information Processing Systems, pages 1133–1141, 2012. (Cited on page 23.)

[Wittenburg 2006] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann and Han Sloetjes. *Elan: a professional framework for multimodality research.* In Proceedings of LREC, volume 2006, 2006. (Cited on page 55.)

[Wrede 2010] Britta Wrede, Stefan Kopp, Katharina Rohlfing, Manja Lohse and Claudia Muhl. *Appropriate feedback in asymmetric interactions.* Journal of Pragmatics, vol. 42, no. 9, pages 2369–2384, 2010. (Cited on page 53.)

[Xu 2007] F. Xu and J.B. Tenenbaum. *Word learning as Bayesian inference.* Psychological review, vol. 114, no. 2, page 245, 2007. (Cited on page 29.)

[Yamane 2009] Katsu Yamane and Jessica Hodgins. *Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data.* In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 2510–2517. IEEE, 2009. (Cited on page 4.)

[Yu 2004] C. Yu and D.H. Ballard. *A multimodal learning interface for grounding spoken language in sensory perceptions.* ACM Transactions on Applied Perception (TAP), vol. 1, no. 1, pages 57–80, 2004. (Cited on page 30.)

[Yu 2005] C. Yu, D.H. Ballard and R.N. Aslin. *The role of embodied intention in early lexical acquisition.* Cognitive Science, vol. 29, no. 6, pages 961–1005, 2005. (Cited on page 30.)

[Yu 2007] C. Yu and D.H. Ballard. *A unified model of early word learning: Integrating statistical and social cues.* Neurocomputing, vol. 70, no. 13, pages 2149–2165, 2007. (Cited on pages 29 and 30.)

[Zheng 2001] F. Zheng, G. Zhang and Z. Song. *Comparison of different implementations of MFCC.* Journal of Computer Science and Technology, vol. 16, no. 6, pages 582–589, 2001. (Cited on page 98.)