HAL open science

A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation

Thomas Cederborg

To cite this version:

Thomas Cederborg. A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation. Interface homme-machine [cs.HC]. Université Sciences et Technologies - Bordeaux I, 2013. Français. NNT : . tel-00937615v2

HAL Id: tel-00937615

https://theses.hal.science/tel-00937615v2

Submitted on 29 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Bordeaux 1

Department of Computer Science

Date: December 10, 2013

Thomas Cederborg

A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

Thesis Advisor Dr. Pierre Yves Oudeyer

Thesis Reader Dr. Katharina J. Rohlfing

Thesis Reader Dr. Chrystopher L. Nehaniv

Thesis Reader Dr.Peter F. Dominey

A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation

By

Thomas Cederborg

Dissertation

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Computer Science Department at
Université Bordeaux 1, 2013

Bordeaux, France

Quid quid latine dictum sit, altum videtur
Roughly:
”That which is said in Latin sounds profound”.
This is a demonstration of how a text can have multiple; layers of communication, interpretations, referents, etc.

Acknowledgments

First I would like to thank my PhD supervisor Pierre-Yves Oudeyer for his advice and insights on fundamental research questions as well as design of experiments experiments, presentation of the results, etc. I would also like to thank all the colleagues that during my PhD listened to my often way to long explanations, gave valuable insights, helped with graphical presentation of results and helping me solve a never ending flood of various technical problems. And perhaps most importantly they told me when my explanations of my ideas were incomprehensible (this often allowed me to notice actual gaps in my understanding of important issues). A special thanks to Manuel Lopes for doing the majority of the work on the paper that chapter 4 is based on. And I thank all the administrative personnel that have helped me during my PhD.

The thesis is based on published conference papers, a journal article and a book chapter. Some of the reviewers of these works actually took the time to really read the text and then offered useful feedback, and for that I thank them. I would also like to thank the rapporteurs for pointing out problems in the text and suggesting modifications.

I would also like to thank everyone that thought me valuable things before I started my PhD. First those teachers who suggested that maybe I should turn my math and physics interest into a career. Then I would like to thank a large number of students at Chalmers Institute of Technology who demonstrated to me how it is possible to get a project done both right and at the same time very quickly. It was completely unintentional, they were just fighting for their lives under a very heavy course load, but through demonstration they taught me everything I know about how to be efficient. I

would also like to thank everyone at the AI-Lab at the VUB for teaching me a lot of science and a lot about doing science.

I would finally like to thank everyone who somehow managed to not just put up with me when I was being stressed and overworked, but actually offered support.

I am also grateful for the funding, which has come from Conseil Regional d Aquitaine and the ERC grant EXPLORERS 24007.

A Formal Approach to Social Learning: Exploring Language Acquisition Through Imitation

Thomas Cederborg

Department of Computer Science
Université Bordeaux 1
Bordeaux, France
2013

ABSTRACT

The topic of this thesis is learning through social interaction, consisting of experiments that focus on word acquisition through imitation, and a formalism aiming to provide stronger theoretical foundations. The formalism is designed to encompass essentially any situation where a learner tries to figure out what a teacher wants it to do by interaction or observation. It groups learners that are interpreting a broad range of information sources under the same theoretical framework. A teachers demonstration, it's eye gaze during a reproduction attempt and a teacher speech comment are all treated as the same type of information source. They can all tell the imitator what the demonstrator wants it to do, and they need to be interpreted in some way. By including them all under the same framework, the formalism can describe any agent that is trying to figure out what a human wants it to do. This allows us to see parallels between existing research, and it provides a framing that makes new avenues of research visible. The concept of informed preferences is introduced to deal with cases such as "the teacher would like the learner to perform an action, but if it knew the consequences of that action, would prefer another action" or "the teacher is very happy with the end result after the learner has cleaned the apartment, but if it knew that the cleaning produced a lot of noise that disturbed the neighbors, it would not like the cleaning strategy". The success of a learner is judged

according to the informed teachers opinion of what would be best for the uninformed version. A series of simplified setups are also introduced showing how a toy world setup can be reduced to a crisply defined inference problem with a mathematically defined success criteria (any learner architecture-setup pair has a numerical success value).

An example experiment is presented where a learner is concurrently estimating the task and what the evaluative comments of a teacher means. This experiment shows how the ideas of learning to interpret information sources can be used in practice.

The first of the learning from demonstration experiments presented investigates a learner, specifically an imitator, that can learn an unknown number of tasks from unlabeled demonstrations. The imitator has access to a set of demonstrations, but it must infer the number of tasks and determine what demonstration is of what task (there are no symbols or labels attached to the demonstrations). The demonstrator is attempting to teach the imitator a rule where the task to perform is dependent on the 2D position of an object. The objects 2D position is set at a random location within four different, well separated, rectangles, each location indicating that a specific task should be performed. Three different coordinate systems were available, and each task was defined in one of them (for example ”move the hand to the object and then draw a circle around it”). To deal with this setup, a local version of Gaussian Mixture Regression (GMR) was used called Incremental Local Online Gaussian Mixture Regression (ILO-GMR). A small and fixed number of gaussians are fitted to local data, informs policy, and then new local points are gathered.

Three other experiments extends the types of contexts to include the actions of another human, making the investigation of language learning possible (a word is learnt by imitating how the demonstrator responds to someone uttering the word). The robot is presented with a setup containing two humans, one demonstrator (who performs hand movements), and an interactant (who might perform some form of communicative

act). The interactants behavior is treated as part of the context and the demonstrators behavior is assumed to be an appropriate response to this extended context. Two experiments explore the simultaneous learning of linguistic and non linguistic tasks (one demonstration could show the appropriate response to an interactant speech utterance and another demonstration could show the appropriate response to an object position). The imitator is not given access to any symbolic information about what word or hand sign was spoken, and must infer how many words where spoken, how many times linguistic information was present, and what demonstrations where responses to what word. Another experiment explores more advanced types of linguistic conventions and demonstrator actions (simple word order grammar in interactant communicative acts, and the imitation of internal cognitive operations performed by the demonstrator as a response). Since a single general imitation learning mechanism can deal with the acquisition of all the different types of tasks, it opens up the possibility that there might not be a need for a separate language acquisition system. Being able to learn a language is certainly very useful when growing up in a linguistic community, but this selection pressure can not be used to explain how the linguistic community arose in the first place. It will be argued that a general imitation learning mechanism is both useful in the absence of language, and will result in language given certain conditions such as shared intentionality and the ability to infer the intentions and mental states of others (all of which can be useful to develop in the absence of language). It will be argued that the general tendency to adopt normative rules is a central ingredient for language (not sufficient, and not necessary while adopting an already established language, but certainly very conducive for a community establishing linguistic conventions).

Table of Contents

1 Introduction **1**

2 Related Work **7**

2.1 Existing research on various types of learners 8

2.1.1 Imitation Learning/Learning from Demonstration 9

2.1.2 Learning from feedback 14

2.1.3 Combinations and studies of biological systems 15

2.2 Related formalisms 15

2.2.1 Formalisms of imitation learning 16

2.2.2 Formalisms of feedback learning 17

2.3 Related work in language learning 18

2.3.1 In natural systems 18

2.3.2 The role of shared intentionality in language acquisition 19

2.3.3 In artificial cognitive systems 22

2.3.4 Generalizing these computational approaches so that they can deal with language as a special case 25

3 A formalism for the field of social learning **27**

3.1 A formalism for step one: finding $u^\star$ 29

3.2 A formalism for step two: finding Ω 32

3.3 A description of the unsimplified setup 42

3.3.1 What situation is the formalism designed to deal with? 42

3.3.2 Informed preferences and a formal success criteria for the learner . 43

3.3.3 What is the purpose of the simplified setups 48

3.4 Removing further simplifications 50

3.4.1 Step three: allowing the learner to actively gather valuable data. . 50

3.4.2 Step four: dealing with a world that is not perfectly visible to the teacher . . . 52
3.4.3 Step five: dealing with a world that is not perfectly visible to the learner . . . 54
3.4.4 Step six: finding Ω without a known generating distribution . . . 54
3.5 Step seven: viewing existing learning algorithms, operating in the unsimplified setup, as testable interpretation hypotheses . . . 55
3.5.1 Graphical representation . . . 59

4 Experiments with concurrent updating of a task model and an interpretation hypothesis . . . 63
4.1 Inverse Reinforcement Learning with Ambiguous Feedback . . . 65
4.1.1 Bayesian Inverse Reinforcement Learning . . . 65
4.1.2 Feedback Model . . . 66
4.1.3 Sign-Meaning Model . . . 68
4.1.4 Algorithm . . . 68
4.1.5 Active Sampling . . . 71
4.2 Results . . . 71
4.2.1 Navigation Task . . . 71
4.2.2 Collecting Objects . . . 74
4.3 Conclusions . . . 74

5 Unlabeled demonstrations of an unknown number of tasks . . . 80
5.1 Algorithm . . . 81
5.1.1 Gaussian Mixture Regression . . . 82
5.1.2 Incremental Local Online Gaussian Mixture Regression (ILO-GMR) 83
5.1.3 How do we pick local points . . . 85
5.2 Experiment . . . 87
5.2.1 Reproduction while introducing additional tasks . . . 90
5.2.2 Reproductions outside normal starting positions . . . 90
5.2.3 Do the framings make a difference? . . . 91
5.2.4 How many demonstrations are needed? . . . 91

5.2.5 Comparison with GMR 94
5.3 Discussion 97

6 Bootstrapping language acquisition as imitation learning of sensorimotor tasks in multimodal contexts 98
6.1 Introduction 99
6.2 The motor gavagai problem 101
6.3 Learning situation 104
6.3.1 Data capture and representation 107
6.4 Algorithms 108
6.4.1 Representation of the tasks 108
6.4.2 Learning algorithm 108
6.4.3 Reproduction algorithm 112
6.5 Experiments 115
6.5.1 Experiment 1: Extending the context to include speech 115
6.5.2 Experiment 2: Relaxing the assumption of a single channel of communication 120
6.5.3 Experiment 3: Extending the types of word meanings that can be understood and learning simple word order syntax 128
6.5.4 Further investigation of the grouping algorithm 133
6.6 Overall discussion 137

7 Conclusion 138

Table of Contents

List of Figures

3.1 In this setup, a policy $\pi_{j,o;m,s}$ is modified by three interpretation hypotheses; Π_a^1, Π_a^2 and $\Pi_{a,e}$. Inputs to policies are denoted Φ and inputs to interpretation hypotheses are denoted Ψ. The black arrows mark inputs and the grey arrows mark modifications. The policy $\pi_{j,o;m,s}$ is used to modify the transform $g_{b;c}$. If the policy $\pi_{j,o;m,s}$ can be reasonably well learnt using only information in Ψ_a (for example a demonstration of a task), then it can later be used to check if the state in another space Ψ_c contains any useful information (for example, if one state is more frequent in the case where the demonstration was a failure, then this can be detected using $\pi_{j,o;m,s}$). The informedness of states in a space Ψ_c can be judged using $\pi_{j,o;m,s}$ (for example how useful states in Ψ_c is for separating failed demonstrations from successful ones) . It is now possible to use $\pi_{j,o;m,s}$ to choose from two different parameter setting of $g_{b;c}$ (two different parameter sets results in two different Ψ_c spaces, which $\pi_{j,o;m,s}$ can be used to choose between). 58

3.2 A learning from demonstration setup where $\Pi_{e,d}$ modifies the policy $\pi_{j,o;m,s}$. The inputs to $\pi_{j,o;m,s}$ is in joint space Φ_j and estimated object position space Φ_o, and the policy is able to set the states in the action spaces α_m (motor outputs) and α_s (speech outputs). The policy is being modified by $\Pi_{e,d}$ based on an estimated teacher evaluation of its own demonstration Ψ_e and a representation of the demonstration in Ψ_d. The evaluation estimate Ψ_e is obtained by $g_{f,t;e}$ based on facial expression Ψ_f (obtained by $g_{c;f}$ from camera input Ψ_c) and tone of voice (obtained by $g_{a;t}$ from audio input Ψ_a). 60

3.3 This figure shows a learning from feedback setup where Π_{v,e^2,a^2} estimates how highly its actions (represented in Ψ_a) were valued (estimated in Ψ_{e^2}) and how informed the teacher was Ψ_v, and uses this to choose whether or not to add the action (along with the context it was performed in) to a list of such actions contained in the policy $\pi_{j,o;m,s}$. 61

4.1 Relation between feedback signs and intended feedback meaning. There are only $Na + 3$ feedback meanings, one corresponding to each available action and the meanings of CORRECT and WRONG. They are fixed and known from the beginning. We assume that there is at least one feedback signal with a known correspondence to a feedback signal, there is the possibility of unknown feedback signs to exist and their relation to the feedback must be learned. For instance the teacher might say good instead of ok. The table shows an example when the agent has 4 available actions (up, down, left and right). 69

4.2 Comparison of learning of the task model with or without learning the feedback model. Number of states is 400. The figure shows the likelihood of the best estimate of the reward function. 72

4.3 Simultaneous acquisition of the task and the feedback models with three different exploration methods for a problem with 225 states. The figures show the likelihood of the best model for the reward and the feedback The top figure shows the results for random exploration and the middle one for active exploration. The bottom one compares both, and also one using ϵ-greedy exploration. Results are for 10 runs, the mean are variance bars are shown. The active exploration method learns faster with smaller variance and bias. 73

4.4 Learning with 400 states (top) comparison between sampling methods, (bottom) mean and variance for the active method. 75

4.5 Histogram of observed feedback signs. We can see that some signs are very rarely used thus making it impossible to estimate their meanings. . 76

4.6 Mean and variance for the active learning method in the "Object Collecting" Task. The system is able to learn the task, the feedback system and new feedback signs. Top - policy loss; Middle - likelihood of correct feedback model; Bottom - number of correctly assigned signs. 77

4.7 Comparison between active and randomly sampling in the "Object Collecting" Task. The system is able to learn the task, the feedback system and new feedback signs. Top - policy loss; Middle - likelihood of correct feedback model; Bottom - number of correctly assigned signs. 78

5.1 Comparison of ILO-GMR with state-of-the-art regression algorithms for the SARCOS dataset encoding the inverse dynamics of an arm with 21 input dimensions. Here, only the nMSE for the regression of the torque of the first joint is displayed. 85

5.2 This shows five demonstrations of one of the tasks learned in the experiments presented below (specifically task 2, which is to draw an S shape starting from the hands starting position). In the left plot we see the vectors in the framing of hand positions relative to the robot. And in the right plot relative to the starting position of the robot imitators hand. Points that are close to each other in the figure to the left are sometimes from different parts of the task, because the position relative to the robot is not relevant to the task. 87

5.3 Here we see the 3 coordinate systems of the experiment. In the text we will write x_{start} as x_s, x_{robot} as x_r etc, for short. The 8 dimensional state space consists of the hand position in all 3 coordinate systems plus the position of the object in the coordinate system of the robot. 89

5.4 This figure shows 3 individual demonstrations of the 4 different tasks. The top task will be referred to as task 1 in the text, the second highest as task 2, etc. The red cross is the position of the object and the blue cross is the starting position of the hand. 89

5.5 This figure shows 5 different demonstrations for each of the 4 different tasks. The blue figures to the left shows the demonstrations in the framing of hand position relative to the robot, the red figures in the middle show hand positions relative to the object and the green figures to the right show hand positions relative to the starting position. We can see that the demonstrations are more similar if viewed in the correct framing for that particular task. 90

5.6 This figure shows that the tasks are successfully reproduced after demonstrated and that adding demonstrations of additional tasks does not destroy performance (one additional task in the second column, 2 in the third and all the other 3 in the fourth). 91

5.7 This figure shows demonstrations of task 1 (move hand to object) in the framing relative to the object to the top right and the reproductions in the framing relative to the starting position to the top right, where we can see that the reproductions look very different due to the different starting positions. To the bottom left we see that despite starting at different locations the hand moves to the top left corner (the object is always somewhere in the top left corner). Finally we can see to the bottom right the reproductions in the correct hand-object framing. There are some odd behavior at times but in general the task is achieved even when starting outside the area that the demonstrations started within. . 92

5.8 This figure shows demonstrations (top left) and reproductions of task 2 (draw an S shape). We can see that, as we should expect, the reproductions look similar to the demonstrations in the framing of hand relative to starting position (top right). A few of the reproductions does not completely replicate the task in the last turn but overall the reproductions are similar to the demonstrations 92

5.9 This figure shows the demonstrations and reproductions of task 3 (move hand in circle around object). 93

5.10 This figure shows demonstrations (top left) and reproductions of task 4 (make a big cyclic square movement). The task is largely reproduced as demonstrated. 93

5.11 This figure shows at the top two rows in black 10 reproductions made while picking only one set of local points using distance in the full state space. The bottom two rows shows the results using framings. we can see that on average the S shape is better in the bottom two rows, especially the later half of the movement (the second "turn"). 94

5.12 This figure shows at the top row demonstrations nr 1 and 5 in first framing 2 (blue), then framing 3 (red), then framing 1 (green) and finally 5 reproduction attempts in framing 1 (the correct one for task 2) of an agent able to see all demonstrations of the other tasks and demonstrations 1 and 5 of task 2. The bottom row is exactly the same but with demonstrations 3 and 5 instead. The demonstrations at the top look the same in two different framings and as a result the agent will not know what framing is the correct one and the reproduction attempts suffer as a result. The demonstrations at the bottom are different in the two incorrect framings but similar in the correct one giving the agent a chance to find the correct framing and as a result these reproductions (bottom right) is superior to the other reproductions (top right). 95

5.13 Displaying results in the same way as in figure 5.6, we can see that tasks 1 and 3 are well reproduced, but that tasks 2 and 4 are not. 95

5.14 Displaying results in the same way as in figure 5.6, we can see that task 2 is now identical to task 4, meaning a degradation in performance (since the reproduction still held some resemblance to the demonstrations with 7 gaussians as seen in figure 5.13). 96

5.15 Displaying results in the same way as in figure 5.6, we can see that Using 30 gaussians does not lead to any new behavior, compared with that exhibited in figure 5.13. 96

6.1 The learning situation investigated. A human interactant (to the right) speaks or makes a gestures, and the demonstrator (to the left) moves the hand of the robot to show him which movement to do. This movement can depend on either/both the current object position, the interactant speech and/or the interactant gestures. Some tasks are non-linguistic, such as "touch the object when it is on the top left of the table". Some tasks are motor responses to a linguistic signal from the interactant, such as "draw a square around the object when the interactant produces the acoustic wave square". Some tasks are quasi-linguistic responses to linguistic signals, such as "draw a "w" when the interactant gestures a specific sign". The demonstrator and interactant provide no symbols to the robot learner, which perceives each demonstration and context through continuous low-level sensorimotor values. Moreover, different tasks are movements defined in various coordinate systems (e.g. absolute versus object centric vs hand centric coordinate system) called framings. The robot has to learn how many tasks there are, which is the control policy for each task and in what appropriate framing it is defined, as well as to learn which tasks should be triggered depending on the current context. 106

6.2 Blue boxes show algorithms and green boxes show data structures. The demonstrated hand trajectories are analyzed in batch mode off line. The membership estimates created in step 2 are used later by the on line reproduction algorithm. Step 3 always creates estimates relating to framing but the details vary between experiments. In experiments 1 and 2, each movement type has one correct framing, and in those experiments, this is what is estimated. In experiment 3, the framing is requested by the interactant, and what is estimated in step 3 of experiment 3 is the syntax of the interactants sign language. This syntax will help the imitator determine what framing to adopt during reproduction (specifically, the estimated syntax tells the imitator which hand sign to look at for the purposes of determining framing). 109

6.3 The results of the learning algorithm is used during the reproduction. In step 1, the contexts of the groups of the membership estimates is compared to the current context. In step 2 the framing estimates and the group selected in step 1 is used to get the data needed by the ILO-GMR algorithm (reevant trajectories in the currently relevant framing). . 114

6.4 This shows the result of the grouping algorithm applied to the data in experiment 1. The five task groups that where found are shown in framing 1 to the left and in framing 2 to the right. We can see that each group found does correspond to one of the tasks described in the task descriptions, and we can also see that the correct framings were found (also notice that the demonstrations look more coherent when viewed in the correct framing). The ordering of the tasks are random and will be different each time (this time it is e,b,a,d,c) but each time the same set of region-framing-data tuples are found. In order to avoid duplication, this figure also serves to show what was demonstrated (since each of the task groups found consists of the demonstrations of one task, the only difference of showing the task demonstrations separately would be in the ordering). 118

6.5 This shows 4 reproductions each for the five reproduced tasks of experiment 1 in the absolute reference frame (the f_1 or f_2 indicates what framing the imitator estimated during the reproductions; each time the framing found is correct). Comparing to the demonstrations and the task descriptions, we can see that the reproduced trajectories correspond reasonably well with what the imitator was supposed to do. We see that the triangle task (task b) has a tendency to sometimes go into the middle of the triangle and circulate in a deformed trajectory after having completed a few correct laps. This problem is not due to the grouping algorithm and demonstrates a shortcoming of the ILO-GMR algorithm that is presented with only relevant data in the correct framing (and it is presented with all the relevant data). 119

6.6 The 7 tasks demonstrated in experiment 2. In the column to the left, in blue, the data is presented in framing 1 (relative to the robot). In the middle, in red, the data is presented in framing 2 (relative to the object). And finally in the column to the right, in green, the data is presented in framing 3 (relative to the starting position). The demonstrations of a task will look like several instances of a consistent policy in the correct framing but might look incoherent in the other framings. Task 1 and 2 ("L" and "R") is to be executed as a responce to the object being to the left (task 1) and right (task 2). Task 3 and 4 is to be executed as a response to specific speech acts ("dubleve" and "circle" respectively). Task 5 and 6 is to be executed as a response to hand signs (and "S" and a "P" respectively) and task 7 is to be executed in case of a starting position far away from the robot (roughly "when the arm is extended; move close to body"). 123

6.7 The memberships found in experiment 2. The height of a pillar show the membership value of demonstration nr indicated on the left axis of the task indicated on the right axis. The 6 groups of 4 high pillars show tasks 1 to 6. There are several values on the right task axis that have no high values and those correspond to empty groups. For values 25 to 28 on the left demonstration axis we can see that the demonstrations of task 7 is not grouped together. The demonstrations of task number 7 has not been correctly grouped and when utilizing a cutoff value of 50% there are 6 groups formed (all of them with the correct data associated to them) but the demonstrations of task 7 are discarded (meaning that reproduction attempts of task 7 results in some other task being selected). In some runs demonstrations 1, 2 and 3 or demonstrations 1 and 2 of task 7 are grouped together as a 7th task (which also represent a failure since while reproducing task 7, the algorithm does not have access to all the relevant information). 124

6.8 Here we can see why task 7 of experiment 2 has not been correctly grouped. The similarity measure will simply not classify the red and the green demonstration as similar in any of the three framing available (they never move in the same direction, no matter how you pick points from the demonstrations). A framing that considers positions in a coordinate frame where one axis goes between the point $H_{x_r} = 0$, $H_{y_r} = 0$ and the starting point would work since the demonstrations would look the same plotted in this framing. 125

6.9 Here we can see the reproduction of the 6 tasks that was correctly found by the grouping algorithm in experiment 2 (Task 7 was not found by the grouping algorithm, and due to this, no reproduction attempts of this task where made). Each task is reproduced four times with different starting conditions (the reproductions of each task is viewed in the inferred framing indicated to the top right of each subfigure). Tasks 1,2 and 3 are defined in the framing relative to the starting position of the imitators hand, meaning that in this framing the starting position is always at 0,0 (the imitator always start at 0,0 relative to the starting position), resulting in more homogenous reproductions. Comparing the reproduced trajectories with the task descriptions and the demonstrations we can see that they match fairly well and comparing the inferred framings with the framings of the task descriptions we can see that all framings where inferred successfully. 127

6.10 Here we see the 12 demonstrations relative to the three objects of experiment 3. The demonstrator observes the first sign, the second sign and then performs the hand movement presented under. Each of the trajectories are shown relative to the three objects. The rows marked "red", "blue" and "green" respectively show the same trajectory, but each row show the trajectory in the a coordinate system centered on the respective object. 131

6.11 Here we can see the 36 reproduction attempts. The rows indicate movement and the columns indicate coordinate system. In each case the signs given to the imitator led to correctly finding the correct data and the correct coordinate system. Comparing the reproduced trajectories with the task descriptions and the demonstrations, we can see that the corners of the triangles are a bit to round and that in some of the circle task reproductions there is some odd behavior before settling into the correct circular movement, but that overall the reproductions where reasonably accurate. There is also no apparent degradation in performance in the three combinations that were not demonstrated, showing the ability to generalize. 132

6.12 Here we can see what level of similarity between trajectory pairs of the same task is needed for the grouping algorithm to succeed. The explanation for the low variance can be seen in figures 6.13 and 6.14 (correctly grouping one task makes it more difficult to correctly group the remaining tasks, which leads to the oddly low variance). 134

6.13 Here we can see the results of each individual run. The success is color coded from low success at dark blue to high success at red. There are only 7 possible values corresponding to 0 correct groupings, 1 correct, etc. We can see that there are no outliers. For example at $k = 1.35$ there are 95 runs with 5 correct groupings and 5 runs with 4 correct groupings, and not a single run with 3 correct, or 6 correct groupings. At $k = 1.3$ all are either 3,4 or 5 correct groupings, with no run outside this band (and for each instance, all runs are within a narrow band, with the highest variation being at $k = 1.15$ where each run is between 0 and 3). This strongly suggest that the success of two task groups are not independent of each other. In figure 6.14 we can see why the difficulty of grouping a task is increased every time another task is correctly grouped. 135

6.14 Here we can see the memberships of one run with $k = 1.25$. The y axis indicate trajectory number and the x axis indicate group number. For visual convenience, task 1 consists of the first 4 trajectories (at the top), task 2 of the following 4, etc, and the non task trajectories are the last 4 (at the bottom). Red indicates a membership value of 1 and dark blue indicate a membership value of 0. For example, the light blue square in the bottom right corner indicates that trajectory 28 have fairly high membership in group 28, and the dark blue square to the left of this indicate that it has no membership in group 27. And from the red rectangle at the top middle of the figure, we can see that trajectories 1 to 4 all have membership value 1 in task group number 15. In general we can see that trajectories of the same task tend to either be correctly classified, or form several identical groups (trajectories 5 to 8 of task 2, trajectories 9 to 12 of task 3 and trajectories 25 to 28 of task 6). 136

Chapter 1

Introduction

This thesis investigates social learning, where a human or artificial learner is modifying its behavior based on interactions with a teacher. This type of investigation can lead to the building of useful artificial intellects, and can help us better understand social learning in humans by building computational models. One example of social learning is when a teacher performs an action, such as trying to shoot a basketball in a hoop, and a learner watches what the teacher does, and then attempts to do the same thing. Another example would be a learner modifying its behavior based on evaluative comments given by a teacher. This type of teacher behavior will be referred to as instruction signals or teaching signals. In this thesis they are considered as instances of the same type of information source. Both the demonstration and the comments give information about what the learner should do. And in both cases the learner must interpret the information. If the teacher uses words that the learner have never heard before, the amount of learning that can take place is severely limited[1]. If the teacher passes the ball to another basketball player, the learner needs to interpret this situation in order to properly imitate the teacher. Unless the learner finds itself in a situation that is exactly identical, it will need to interpret the situation to figure out what which part of the scene is important to the decision whether to pass the ball or not (whether to pass to a player could depend on things such as the skill of that player, whether or not there are any opponents between the learner and the player, whether the player is shouting "here", etc, etc). If the teacher often fails (perhaps missing the hoop on some occasions), then the learner could learn much more if it could interpret the demonstrations and make a guess about which ones were failures (for example by analyzing facial expressions, body language or what the teacher says after the demonstration). One of the experiments presented includes a learner that is estimating what the evaluative comments of a teacher means, at the same time as estimating what the teacher wants it to do. This experiment shows how learning of how to interpret the teacher concurrently with the task can improve performance. It also demonstrates how the learner can improve performance by actively acquiring more useful information.

[1] If the learner has several hypotheses of what the teacher wants it to achieve, some of those hypotheses might predict that action A and B should be evaluated the same, while other hypotheses predict that actions A and B should be evaluated very differently. Thus it is possible to make updates simply by observing that actions A and B are evaluated the same way, but this is clearly not as informative as it would be if the teacher used words that the learner is familiar with (and even this update is dependent on interpreting the speech as an evaluation of the learners action).

Social interaction is the medium of learning, but it can also be the task that is learnt. The rules: "in certain situations, when a team member shouts "here" the ball should be thrown to that player", and "in certain situations (such as a the learner having a good position, and a team player having the ball, etc) the learner should shout "here"" are perfectly good examples of what a learner can acquire through social interactions. It can for example be learnt by observing a teacher performing these actions, or by the learner acting and listening to verbal feedback from a teacher, or from a learner interpreting an evaluative scalar value produced by the teacher, or any number of other social learning situations. Social learning can be done verbally or non verbally, and the tasks learnt can be verbal or non verbal.

The term "it" will be used for both learner and teacher throughout the thesis due to the generality of the term. As we make points that can refer to for example robots, humans, animals, etc there is no reason to reduce generality by talking in terms of "he" or "she" (there is in principle nothing that prevents us from dealing with a learner that is in no sense a person, especially in the case of narrow range applications such as an autopilot that learns from demonstrations, feedback, etc).

The thesis will present several experiments where the learner acquires multiple skills from unlabeled demonstrations. Some of the skills will have a linguistic component similar to the example above where the speech act of a team member should result in a specific learner action, and other skills will have no linguistic component at all. The same learning mechanism is used for both types of tasks, illustrating that language acquisition might not require any new learning strategies. If a general type of imitation learning is useful for learning group norms even in the absence of language, and is able to learn language, then there is no need to posit any separate "language acquisition strategy" beyond that already used to learn non linguistic tasks. If following group norms is adaptive also in non linguistic groups, and the resulting norm adoption strategy results in language, this could explain the evolutionary origins of language (the possibility of secondary adaptations resulting from reliably being born into a linguistic community is outside the scope of this thesis). New algorithms are introduced and experimentally evaluated in a first step towards establishing the ability of the generalized norm adopting mechanism to learn language.

The thesis also tries to give a solid theoretical foundation gathering all types of social learning under the same formalism. Each information source is treated as the same kind of thing, and a learner is seen as someone that interprets or learns to interpret these information sources. It can often be useful for a scientific project to more clearly define what constitutes success in various situations. If the learner finds a new way of getting the ball into a hoop that the teacher is unable to do or never considered that the learner might do, is it then a success (for example getting a chair to stand on, or jumping up to the hoop)? The formalism attempts to turn this into an empirical question by defining success in terms of what an informed teacher would think.

It also allows us to see what new types of learning situations could be explored. One unexplored avenue is figuring out how to tell a failed demonstration from a successful

demonstration, for example by watching facial expressions and body language. Even without this skill, it can still be possible to confidently learn a task from observing many demonstrations. When the task is known, the learner knows which demonstrations where failures, and can find correlations between failures and facial expressions or body language. Being able to interpret these self evaluation signals could be useful when learning other tasks.

The formalism starts from imitation learning and generalizes it to include a general class of information sources that the learner can interpret. If a teacher performs a demonstration, looks at a certain object while the learner is performing a reproduction attempt, makes a speech comment during the reproduction and finally pushes an evaluative plus or minus button, then we call them information sources. The presented formalism allows us to treat them all in a unified manner as it avoids making assumptions regarding the semantics of a source. Each can tell the learner what the teacher wants it to do, to the extent that the information source can be interpreted. Looking at an object could indicate that it was important, or that it was while interacting with this object that the learner failed, etc. The plus and minus buttons could be pushed based on incremental progress, or absolute performance, or sometimes it might be pushed by a teacher that did not attend the reproduction attempt properly and might have missed something (determining if the teacher rewards incremental progress or absolute performance is difficult to do at programming time as users are different, and in case of a small number of feedback instances, the two hypotheses could lead to different policies). It is easy to see that speech comments are more useful if they can be interpreted correctly, but less obviously this also includes things such as determining whether the word "good" rewards incremental or absolute progress (perhaps "good", "nooo!" and "great" is well modeled as evaluating incremental progress, while "perfect!!" is well modeled as evaluating absolute performance), and, as with the buttons, it might be useful to estimate how informed the teacher was when giving the feedback. The terms teacher and learner will be used regardless of the types of behavior the agent is interpreting (for example demonstrations, speech comments, facial expressions, the pushing of a reward button, tone of voice, EEG readings, eye gaze, etc). The formalism provides a general framework for describing any agent that is trying to figure out what a human wants it to do, without making assumptions regarding the type of behaviors that is interpreted.

After introducing the formalism, an experiment is presented investigating how a learner can learn an unknown number of tasks from unlabeled demonstrations. By extending the traditional context of imitation learning to include speech, experiments become possible where the learner acquires words through imitation using the same framework as non linguistic imitation. Three related experiments are presented which investigates a generalized imitation learning strategy that is able to deal with both linguistic and non linguistic situations. This gives us three main sections:

A formalization of social learning: To say verbally that a robot should be able to interpret various sources of information is ambiguous and should be made concrete and formalized (human demonstrations, facial expressions, eye gaze, etc does not contain

a crisply specified meaning that the learner can decode). To formalize the problem, a success criterion must be introduced. First a success criterion is introduced in a series of simplified worlds (presented in stages where some simplifications are removed at each stage). Then learning algorithms operating in the real world are reinterpreted and a success criterion is introduced in terms of the teachers informed preferences. Informed preferences can be roughly described as what the teacher would want the learner to do if it knew and understood everything it considers relevant to the learners action choice (for example long term consequences the actual teacher is unable to predict, or side effects it does not see).

In the simplified setup, the complexities of the world are reduced to the point where it becomes an inference problem of a very well known form (so that a large amount of existing ideas, insights, solutions and approximate algorithms can be used). Then the simplifications are removed step by step in order to gradually approach the desired setup containing an actual human teacher in an unstructured environment. At one stage of removing assumptions/simplifications, the problem stops being a well defined inference problem with a mathematically defined success criteria (there will be no way of obtaining a number representing the level of success of a learner in a specific situation), and we must fall back to the informed preferences of the teacher defined in the next section.

The formalism for the real world setup starts by re describing existing learning algorithms as a certain way to interpret an information source. The idea is introduced where the way of interpreting an information source can be modified based on observations, which means that any specific interpretation of a source is an hypothesis (for example that an observable scalar value indicates absolute success, or that it indicates incremental progress). A learning algorithm is thus referred to as an interpretation hypothesis. The concept of informed preferences of the teacher is defined and a learning algorithm is now viewed as an hypothesis of how inputs (for example EEG signals, demonstrations, eye gaze, facial expressions, etc) relate to the informed preferences of the teacher. The problem is unfortunately no longer an inference problem of a well known form with a mathematically defined numerical success value. This means that it is no possible to define an optimal solution and thus no longer possible to describe an algorithm as searching for an approximation to a well defined optimal solution[2]. The learner has some rule for selecting actions, which we will refer to as a policy. In addition to suggesting policy updates, the interpretation hypotheses make predictions regarding what will be observed. Let's take the example where a human teacher is given one button with a plus

[2]The initial set of interpretation hypotheses along with update algorithms operating on them can be interpreted as forming a prior distribution over possible informed preferences of the teacher and over ways in which those are connected to the learners inputs. If this is taken as an axiom, then the problem becomes an inference problem again. Success is now however possibly separated from the optimal solution to this problem since the initial assumptions built into system might be inaccurate. This leaves us with an inference problem that can be approximated, but we still do not have access to a number that is guaranteed to represent actual success (the learner can however in principle be guaranteed to perform optimally given the information it has).

sign on it, and one button with a minus sign on it (both including a “volume” control), and is told that it can use these to give feedback to the robot. A learner can now have several competing hypotheses regarding how to interpret this teacher signal. For example that the buttons are pushed comparing the learners most recently performed action to: (i) the best action the teacher has seen (ii) the optimal action (iii) the previous seven actions performed by the learner (iv) the previous two actions performed, etc, etc. For a limited data set, the suggested policy updates can be quite different. They also make different predictions regarding what interaction histories will be observed (if the learner performs the same action in the same context for example, then the different hypotheses will predict different changes in evaluation, in general resulting in different probabilities for a given interaction history). Thus, given an observed interaction history, it is possible to update the set of interpretation hypotheses (discarding, changing probabilities or modifying parameters). If the learner has access to a few reliable task policies that are learnt by interpreting some other information source, and the interaction history of the time these tasks were learnt includes button push data, then these task models can be very useful when the learner chooses between the different ways of interpreting the reward button (both the correct action and the button pushing behavior is then known). It is in general possible to use the interpretation of one information source to learn how to interpret another source (if the well understood source can be used to learn a task policy, then this task policy can be used when learning to interpret another information source).

The concept of informed preferences is designed to deal with cases such as ”the teacher would like the learner to perform an action, but if it knew the consequences of that action, would prefer another action” or ”the teacher is very happy with the end result after the learner has cleaned the apartment, but if it knew that the cleaning produced a lot of noise that disturbed the neighbors, it would not like the cleaning strategy”. After formally introducing the concept of an informed version of the teacher, a learner action choice is judged according to the informed teachers opinion of what would be best for the uninformed version.

This means that even if the teacher has a plus and a minus button, or gives feedback in some other way (such as saying ”stop doing that”, ”be careful with the vase, it’s expensive” or ”good robot”), the goal is not to accumulate as many plus button pushes or as many ”good robot” utterances as possible (but instead to do what the teacher would want it to do if it where properly informed). These things are instead imperfect indicators that the learner did good. One big difference is when hiding a mistake will lead to higher reward. Another is when consistently performing well will result in the teacher deciding that the learner knows the task (and thus no longer see any point in pushing the reward button in order to teach the learner what to do).

Learning an unknown number of tasks from unlabeled human demonstrations: An experiment is presented where the robot has access to a set of demonstrations, and knows that each demonstration is of some task that the human wants it to perform. It does however not know the number of tasks, what demonstration is of what

task, and it is not given access to any symbols or labels (there is no additional information provided besides that which is contained in a set of demonstrated continuous hand trajectories and the rest of the continuous context). The teacher is in this case a demonstrator and is attempting to teach the learner (which is an imitator) a rule where the task to perform is dependent on a feature of the context, specifically the 2D position of an object. The objects 2D position is set at a random location within four different, well separated, rectangles (one for each of the four tasks) during both demonstrations and reproductions. The learner observed the position of its hand in three different coordinate systems. A task can be defined in the robots hands position in a coordinate system centered on the robots body, or centered on the starting position of the robots hand, or centered on the object (for example "move the hand to the object and then draw a circle around it", or "make an "S" shape where the top of the "S" shape is at the position of the imitators hand at the start of the reproduction attempt"). To deal with this setup, a local version of Gaussian Mixture Regression (GMR) was used called Incremental Local Online Gaussian Mixture Regression (ILO-GMR), which was introduced in [102] and [104]. A set of data points is selected based on similarity with the current context, and then a small GMM is built using only this data. The local GMM is used to generate output, which leads to a new position, a new set of local points and a new local GMM. What context is the closest depends on what coordinate system one is measuring distances in (in two different contexts the hand of the imitator could be at the same place relative to the object but not relative to the robot for example). An algorithm was proposed to deal with this by comparing different local GMMs created with the data obtained using different framing assumptions.

Learning linguistic and non linguistic tasks using a generalized imitation learning strategy: The teacher is again a demonstrator and the learner is again an imitator. The imitator is presented with a setup containing two humans, one demonstrator (who performs hand movements), and an interactant (who might perform some form of communicative act). The interactants behavior is treated as part of the context and the demonstrators behavior is assumed to be an appropriate response to this extended context. Two experiments explore the simultaneous learning of linguistic and non linguistic tasks (one demonstration could show the appropriate response to an interactant speech utterance and another demonstration could show the appropriate response to an object position). The imitator is not given access to any symbolic information about what word or hand sign was spoken (these are instead represented as a point in a low dimensional continuous space computed from the raw sensor data), and must infer how many words where spoken, how many times linguistic information was present, and what demonstrations where responses to what word. Another experiment explores more advanced types of linguistic conventions and demonstrator actions (simple word order grammar in interactant communicative acts, and the imitation of internal cognitive operations performed by the demonstrator as a response).

Chapter 2

Related Work

Contents

2.1 Existing research on various types of learners 8

2.1.1 Imitation Learning/Learning from Demonstration 9

2.1.2 Learning from feedback 14

2.1.3 Combinations and studies of biological systems 15

2.2 Related formalisms 15

2.2.1 Formalisms of imitation learning 16

2.2.2 Formalisms of feedback learning 17

2.3 Related work in language learning 18

2.3.1 In natural systems 18

2.3.2 The role of shared intentionality in language acquisition . . . 19

2.3.3 In artificial cognitive systems 22

2.3.4 Generalizing these computational approaches so that they can deal with language as a special case 25

The topic of this thesis is agents trying to figure out what a human wants it to do. A formalism of this setup is proposed that attempts to cover all agents trying to figure out what a human wants it to do (referred to as learners), and it tries to avoid making assumptions about the semantics of the information source that the learner uses to figure this out. Experiments are also presented that focus on an agent that learns several different types of tasks from unlabeled demonstrations, mixing linguistic and non linguistic tasks (and blurring the line between the two by suggesting that they are instances of the same category and that the difference between linguistic and non linguistic lies in the eye of the beholder). This means that related work includes previously proposed formalisms of related areas, all forms of social interaction where the agent investigated can be described as trying to do what a teacher wants it to do, as well as literature on language learning.

2.1 Existing research on various types of learners

There are many ways to learn from social interactions, both in terms of what is learnt, and in terms of what type of information is used to learn. It is for example possible to learn by observing a teacher interacting with another human, and then imitate the teacher as it reacts to the environment and to what another other human is doing or saying. In this situation it is for example possible to learn normative rules for how one should respond to various behaviors of other people, such as speech acts. Another way to learn something is by the teacher taking the learners hand and moving it in a trajectory, for example in a dance move, or to demonstrate how to handle an object. The dance move and the tool use could for example also be learnt from observing a teacher performing the same action, allowing a wider range of tasks but requiring the learner to solve the correspondence problem [137]. Other ways of learning from social interactions would be learning from speech comments, necessitating learning to understand those comments (see [88] and [90]). If social learning is broadly construed, it could also include the type of reinforcement learning where a teacher pushes an evaluative button. This broad view of social learning comprises several research disciplines that can all be viewed as a teacher performing some sort of behavior that can be referred to as instruction or teaching signals, and a learner interpreting these signals and learning from them.

An attempt is made in this thesis to formalize all these types of learning in a unified manner, which means that related work will include all the types of research that is being formalized, as well as research related to the experiments presented. By its unified representation of social learning, the formalism opens up many new possible avenues of research, since viewing a research field in a new light can will often lead to new ideas of how specific projects can be extended or combined. Perhaps the most prominent example is a learner that is learning how various information sources should be interpreted concurrently with learning to perform multiple tasks. These new research avenues are a significant contribution of the thesis and will be discussed while covering the relevant previous work. Learning from demonstration has focused mostly on learning single sensorimotor skills, where what to do is determined by things such as the position of an object. Linguistic learners have treated language and action learning as two different, but highly interconnected, processes. This thesis presents experiments where language and action learning are seen as two special instances of a single imitation learning system. Research on the type of reinforcement learning where a human provides the reinforcement signal can be re interpreted as a special case of social learning. The learner that is trying to figure out what a human providing the reinforcement signal wants it to do might have to re interpret the meaning of this signal, just as evaluative speech comments needs to be learnt / interpreted. A learner might need to determine whether the reinforcement signal indicates incremental progress or absolute performance (the two models will in general make different predictions of what interaction histories are more probable, and imply different policies, making it possible and useful to estimate which one is correct). In a similar way it would be possible to estimate how informed the

teacher was when the signal was provided (different hypotheses of when the teacher is informed will in general predict different interaction histories, and for a limited amount of observations two different hypotheses of teacher informedness can result in very different policies). The learner is no longer maximizing the reinforcement signal, but instead trying to interpret the signal. There is still however a strong connection between the reinforcement learner and the learner presented in the proposed formalism.

The use of the word "teaching signal" should not be taken to imply that the teacher is necessarily transmitting some aspect of an established interaction protocol, and indeed all models can be applied even if the teacher is unaware of the learners existence [1].

2.1.1 Imitation Learning/Learning from Demonstration

Research setups where one agent is trying to replicate the actions taken by another agent is both related to the experiments presented and is at the core of what is covered by the social learning formalism. This research is usually referred to as imitation learning, programming by demonstration or learning from demonstration. In [95], an agent based view of imitation learning is presented and five central questions are put forward. The imitator must decide who, when, what and how to imitate, and someone must address the question of what constitutes success (for example formalized by an experimenter or the creator of an artificial imitator). The question of what constitutes success is a central theme of this theses, and in the formalism presented, success of imitation is evaluated according to the goal of the learning agent, roughly to try to figure out how an informed version of a teacher would like it to act (specified further in section 3). If the learner has access to several types of interaction, the question of when to imitate is thus cast as a question of which type of interaction is most efficient for figuring out what the teacher wants it to do (an alternative to requesting a demonstration could be to perform an action and then try to interpret the teachers response to that action). The thesis will also present experiments that mainly deal with the "what to imitate" question. In [95], the question of what the imitator is trying to achieve is left open (and depends on what type of animal the imitator is or what type of robot it is, etc). If the imitator has a specific goal whose progress it can measure and that is unrelated to the teacher/demonstrator, then the who and when questions can be answered with respect to that goal (the formalism presented is different since the goal is dependent on a teacher whose mind is not necessarily easy to read). We can take the example of a robot that is trying to maximize the amount of states it can reach in a measurable outcome space, and an animal that is trying to maximize food (for example given as rewards for imitation by a human). Who to imitate is then a question of who gives the

[1] There is for example nothing problematic with having a child inferring normative rules or learning how to use a tool by observing adult humans interacting with each other, even if they are unaware that the child is watching. In the same way, there is nothing strange about having an artificial learner acquiring linguistic conventions by looking at two humans interacting in a situation where they are not even aware that they are being observed.

most informative demonstrations relative to current skill levels (producing reproducible action with outcomes it could not previously achieve) or a question of who gives it the most food. When to imitate depends on how effective this is for learning relative to other learning strategies or a question of whether imitation is the most effective way of getting food. The how and what questions also becomes well formalized (imitation of which parts of the demonstration results in food), meaning that the simultaneous exploration of all four questions is a well formed problem. In [40] all four questions are simultaneously addressed by an artificial imitator that is trying to expand the amount of outcomes it can reach through strategic and active life long learning. But in general, the setup where an imitator faces these four questions simultaneously is not well explored.

The field has mostly focused on the what and the how questions where two large technical issues that algorithms must take into consideration has been to maximize at the same time *genericity* and *accurate generalization*. The goal has been to develop techniques that may allow a robot to learn various context-dependant skills without the need for tuning of parameters for each new skill: this implies that the dimensions/variables that the robot can measure in demonstrations should be higher than the number of dimensions that are relevant for determining the action to be done during a (sub-part of) a skill. Indeed, different skills might be determined by different variables/dimensions/constraints. If a robot can to learn the "essence" of a skill, this will allow it to reproduce demonstrations successfully in contexts which are not exactly the same as those of the demonstration. If all details of demonstrations are considered then the learner will never find itself in a situation that is similar to a demonstration, since there is always something that is different, and if the wrong details or the wrong abstractions are used, demonstrations might seem to have been made in similar situations even if they are not. If the learner frames the demonstrations correctly (that is, attends to the relevant details or abstracts the demonstration in the right way) it can find those demonstrations that really are relevant to the current context. In [96] several learning from demonstration projects are covered, classified according to how demonstrations are gathered, and according to the type of algorithm used, as well as discussing some ways of dealing with imperfect demonstrators. It also classifies inverse reinforcement learning as an instance of learning from demonstration. In [64] research into robot programming by demonstration is summarized and different types of algorithms and ways of encoding tasks are presented.

In [137] the correspondence problem is discussed. When an imitator is observing a demonstrator perform an action, it must decide how to map the actions of the demonstrator into actions in its own action space. This problem becomes more difficult the more the embodiments of the demonstrator and imitator differ. Imagine a child observing a much bigger adult demonstrator clap its hands. Should the hands of the imitator be at the same height above the floor, or should the angles of the arms be the same? Should the maximum distance between hands be the same, or should the angles be the same? If the maximum distance should be smaller (so that arm angles can be mimicked), it is not physically possible to simultaneously mimic the speed of clapping, the speed the hands hit each other and the relative speed curve of the hands (unless the

angles are much larger, then the hands of the imitator can only impact each other at the same speed as the adult, if it either increase the pace of clapping or take a pause at some point during the movement). This simple, "clap your hands" example shows that there are difficult questions to answer about what the essence of a movement is, even if embodiment only differ in size, and the movement is only relative to the agent itself. In [121] a framework is presented, mapping one or several parts of a demonstrators body to one or several parts of an imitators body for example mapping the position of the two hands of a human demonstrator to the nose of a dolphin imitator (so that imitation means that the dolphin does the same thing with its nose as the human demonstrator did with its two hands, for example "punch" a ball). In [122] a focus is instead on effects on objects, and various examples are presented showing how the imitation differs as different mappings are used.

One method that has been used to learn this mapping is presented in [116] where a robot observes a human that is mimicking its motions. The robot learns the intended mapping of the human since the human performs the movement that it considers to be best corresponding to the movement of the robot. One common method of demonstration that avoids dealing with the correspondence problem is to tele operate the robot, for example using a joystick to remote control a robot helicopter [67], and the same effect can be achieved by physically directing a robots body, as in for example [127].

One approach is to define a set of primitive behaviors or actions and segment the problem into several different sub problems. The robot needs to learn how to reproduce the individual behaviors, how to classify parts of a demonstration as a series of behaviors and find an algorithm that can learn from a demonstration when described as a list of behaviors (including how to generalize to new situations using this list). The task can be encoded using possibly hierarchical, graph based models, for example a Hidden Markov Model (HMM), with parameters set using machine learning algorithms. Examples of this approach include [117] working with a wheeled robot and [118] and [119] which proposes to use a hierarchical model to encode household tasks such as setting the table. [120] encodes a demonstration using a set of pre defined postures, extracting task rules in the space of these postures. [120] also explores the question of granularity as it is necessary to decide if a given part of the task should be a primitive or composed of more finely grained primitives.

Demonstrations can be encoded at the trajectory level. Instead of building a model that operates on discrete primitive actions models are built that operates in continuous spaces. These spaces could for example be in the joint space of the robot or in the operational/task space, such as the position, speed or torque space of its hands, mapping sensory inputs to motor outputs or desired hand velocities (which can be seen as different levels of granularity). In the early work of [126] the set of acceptable trajectories is spanned by the trajectories seen during the demonstrations, and [128] introduces a non parametric regression technique based on natural vector splines to build a representation of a trajectory either in cartesian (sometimes referred to as task space) or joint space, from several demonstrations. In [129] a method inspired by dynamical systems and

attractors is presented using recurrent neural networks to learn different motions and switch between them. The mimesis model proposes to encode a trajectory as a HMM. Reproduction is achieved using a stochastic algorithm on the transition probabilities of the HMM.

A number of authors have proposed to encode sensorimotor policies as sub-symbolic dynamical systems whose dynamical output is determined by both internal parameters and an input context [64]. These dynamical systems can be implemented/modelled as complex recurrent neural networks [48][7][49] or as more traditional statistical regression techniques [77][87]. These are interesting as they in general do not make assumptions regarding the type of outputs and inputs they are dealing with, making them suitable for use when the context has been extended to include the communicative acts of an interactant. In practice, output has typically been motor commands and input context has been a compact encoding of the past sensorimotor flow [69][64][67] including potentially internal variables that may encode simple things such as motivational states but also predictions of the future or hypotheses about properties of the environment that cannot be observed directly [80].

Calinon et al. showed, through a series of advanced robotic experiments [108, 101, 125, 69], that the Gaussian Mixture Regression technique, introduced in [130], could be very successfully and easily used for encoding demonstrations through a GMM tuned with an expectation-maximization algorithm [109], as well as for extracting their underlying constraints and reproducing smooth generalized motor trajectories. This approach alleviates much of the work from the programmers, who only need to find the number of gaussians for each task. Basically the same algorithm can be used for learning a wide variety of tasks and it robustly reproduces smooth movements. Once the model is built, and following for example the time-independant approach presented in [66], it can be queried quickly with the current state giving the desired action (minimizing the needed computation time once the model has been learnt). This is a very powerful method but it does have some limitations, that prevent it from being directly used when one wants a robot to learn incrementally and online new tasks (using a large number of gaussian is computationally expensive, and specifying the number of gaussians must either be done by hand or recomputed every time a new task might have been demonstrated). This is especially true when the programmer does not know in advance the number and the complexity of tasks, and when the demonstrator is not allowed to provide categorical information about the demonstrations (such as "demonstration number 7 is of task number 2").

Recent and sophisticated statistical inference methods have been devised to learn these context-dependant skills based on the direct modelling of the skill with such dynamical systems. The learning is divided into several interacting parts: 1) given that the robot typically observes much more variables than those that are relevant for the task, dimensionality reduction techniques have to be used for finding the right projections/framing of the full sensorimotor space history onto the dimensions which compactly describes the relevant parts of the context [70][71] 2) the internal parameters of the model of

the dynamical system (neural network or regression method), which given the context representation determine the dynamical output, must also be learnt [65][7][49]. These recent methods include techniques such as Locally Weighted Projection Regression [73] Gaussian Process Regression [74][75][76] or Gaussian Mixture Regression [65] used in conjounction with dimensionality reduction techniques such as principal component analysis [70], independant component analysis [71] or other techniques like ISOMAP [72].

In the case of a single task (so that the number of gaussians does not have to be changed and there is no question of which task is demonstrated) a more incremental approach has been proposed in [101], but it still requires re computation of the model and it is not obvious how to extend it to the context of incorporating a demonstration of a new task within this framework. Indeed, in this approach to GMR when a new task is introduced, the number of gaussians should typically be increased manually by the programmer. Even if this is done automatically one would have to somehow inform the robot that the demonstration is of a new task and then wait for the entire model to be rebuilt after automatically discovering the number of gaussians, which becomes computationally exponentially more difficult as EM needs to tune the parameters of more and more gaussians in the mixture. What we would like is to have a demonstrator teach the robot a task and and that at any point he can start teaching the robot a completely different task (perhaps because the robot does the current task well or if the demonstrator does not think he is able to learn the current task), but avoiding any additional programmer intervention or global/heavy recomputation of a model. In this framework the robot needs to infer what task he is to perform based on the environment only. If the demonstrator performs a number of tasks several times (not necessarily one task several times followed by another task, etc), the imitator should be able to estimate which demonstration is of what task without any programmer intervention or symbol passing.

In [107] we can see another approach to studying imitation learning, in this case using the parrot Alex. Alex is shown to be able to learn a large number of quite complex tasks when trained in a very specific type of setup. Alex is motivated by food, but the computational problems he must solve are similar to that of our artificial agents. Specifically Alex can not use the simulation theory of mind to figure out what a human wants it to do (he can not think "what would I have meant if I was doing that"), since his cognitive architecture is so different from the humans it is learning from. This is interesting for it‘s similarity with artificial learners that do not share the human cognitive architecture (either due to the obvious technical difficulties in implementing that, or because it is desirable to build a robot with a different type of mind). The experiments with Alex thus show that it is possible to learn quite a bit of interesting skills without using the simulation theory of mind.

This thesis will present experiments that moves beyond the usual setup with either a single task or labeled demonstrations. Previous work that explores the same issue includes [65], where two different tasks are learnt from unlabeled demonstrations (the

starting position determines which of two different ping pong swings to perform). The imitator is not told the number of tasks or what demonstrations is of whet task. The problem of multiple tasks is also dealt with in[83] and [84] (this research is discussed at greater length in the following section about related work in language learning).

It would also be nice if a new demonstration could immediately be incorporated online and incrementally during the teaching process. To achieve these goals we use an Incremental, Local and Online formulation of Gaussian Mixture Regression (ILO-GMR), which was introduced in [102] and [104]. The central idea of this technique is to build online and on-demand local Gaussian Mixture Models of the task(s).

2.1.2 Learning from feedback

The idea to let the task model be complemented by feedback from the demonstrator upon reproductions as well as by self-exploration actions done by the robot is presented in [76][77], giving an example of how to combine multiple sources of information from social interaction. In [67] an imitator learns to perform helicopter acrobatics and its skill surpasses that of the teacher (this research exemplifies how important it is for a formalization of social learning to allow a learner to become better than the teacher).

It has also been proposed that learning such context-dependant skills could be achieved through inverse reinforcement learning (see [78] for some early work and [79] [91] for recent overviews): instead of directly modelling the skill at the trajectory level with a dynamical system, a first inference step is performed that consists in trying to infer the reward/cost function, i.e. the constraints, that the observed demonstrations are supposed to optimize. Such a function can for example be a numerical assessment of how much food gets into the mouth of the doll in the case of a set of doll feeding demonstrations, or how close the stone of Steve arrived to the rabbit if the inferred intention of Steve was actually just to hit the rabbit with the stone. Thus, this is a technical approach to learn directly the goal/intention of the demonstrator. When an hypothesis of reward function has been generated, then the learning agent can search for the adequately encoded dynamical system (including encoding of the context) that allows it to maximize the corresponding rewards. The drawback of this approach is that it is difficult to design a hypothesis space of reward functions which is at the same time flexible enough to learn a variety of tasks and allows for efficient statistical search and inference. The advantages of this approach is first that it allows potentially better generalization by letting the robot self-explore alternative strategies to achieve the goal that may be more efficient or more robust than those used by the demonstrator (e.g. see [67]), and second it naturally makes it possible to take advantage of reinforcement feedback from the demonstrator during the reproduction attempts of the robot. In [92] three different drives is contrasted, following non social baseline preferences, emulation and imitation. Imitation as defined in [92] refers to replication of the intentions of a teacher, while emulation means to replicate the effects that the teachers actions had on the world. Emulation is used to describe a sort of non shared intentionality type

of imitation that takes place when the learner notices that doing the same thing as a teacher will result in an effect that is interesting on its own.

2.1.3 Combinations and studies of biological systems

It has to be noted that these various approaches to learning by imitation can also naturally be augmented by more complex interactions involving things such as the demonstrator explicitly drawing the attention of the learner towards the relevant aspects of the context using social cues (e.g. see [80]). Thus, this family of approaches adopt a non-restrictive broad view of learning by imitation, by contrast to more restrictive definitions sometimes used and which have lead some researchers to argue that "learning by imitation is limited because the observed action does not always reveals its meaning [...] In order to understand an action, a learner will typically need to be provided with additional observation given by a teacher who demonstrates what is crucial: the goal, the means and - most importantly - the constraints of a task" [5] (for the same line of thinking, see also [82] and [81]).

Combinations of evaluative feedback and demonstrations have been explored in several different settings. In [98] the learner is provided with demonstrations, and the teacher is able to provide evaluative feedback by indicating parts of a reproduction where the learner did good or bad. While [99] does not explicitly say that it combines combining different kinds of social learning feedback, if we view the motor primitives as a set of demonstrations (or a set of demonstrated skills), then the reinforcement signal acts as a second source of information.

Several studies have also investigated the actual behavior of non expert teachers, for example in [124], where different interaction protocols is tested on non experts to see how they perform in actual situations. In [89], it happens that teachers give rewards in order to encourage a learner. Such a teacher would present an obvious problem for any learner built on the assumption that rewards perfectly measure performance. In general, the problem of learning how to interpret teachers is not very well explored. In [88] and [90], the issue of how to learn the meaning of the teachers feedback is explored. The learner must figure out what the task is, and at the same time refine its interpretation of the teachers guidance and evaluation of its performance (a learner might for example refine its interpretation of comments such as: "go right" or "bad robot").

2.2 Related formalisms

Since a big part of this thesis consists in providing a formalism of social learning, we offer a survey formalisms of sub sections of social learning.

2.2.1 Formalisms of imitation learning

Formalisms of imitation learning have taken two main forms: classification of tasks and mathematical formalisms.

In the classical work [123] an algebraic framework is specified and a success criteria is defined for imitation learning. The set of imitator and environment states is referred to as X, consisting of the states at each instance of a time series (where each time instance contain for example: states internal to the imitator, the fact that it is holding an apple, states in the environment, etc). The set of demonstrator and environment states is referred to as Y. Both X and Y is contained in the state set Z. Finally success is defined as minimizing a distance metric $d : Z \times Z \rightarrow \Re$ (where 0 is optimal imitation). There are three ways in which this is not suitable for the purposes of this thesis; (i) it is not clear how such a distance metric should be obtained as demonstrator evaluation is problematic, for example when a human demonstrator is not aware of everything that happened, (ii) it does not include the possibility of other types of information sources besides demonstrations (which is an important part of the presented formalism), and (iii) it can not formalize the situation of non optimal demonstrations (even given a correct framing and a perfect distance metric between imitator behavior and a demonstration, the situation where a demonstrator simply failed at achieving the task perfectly can not be handled properly). Imagine for example a demonstrator trying to shoot a basketball in a hoop and failing most of the time (a situation an imitator can infer a goal from, especially given complementary information sources). Even if the imitator has identical embodiment, and is in an identical situation to a demonstration, it should not miss on purpose if it knows what the goal was and is able to achieve it, even if the demonstrator did miss in the same situation. But according to the formalism in [123], missing the shot in this situation is always an optimal action, no matter what the demonstrator thinks about this (as long as the shot is missed in the exact same way, it is optimal per definition). [120] describes in a comprehensive way in which one could represent correspondences where embodiments of demonstrator and imitator differ significantly. A human demonstrator might punch a ball with both hands, and a dolphin imitator could map the movement of the two hands to the movement of its nose or its tail, allowing it to transform some demonstrations into desired movements of its own body.

The summary provided in [96] also offers a formalism for learning from demonstration. The demonstrations are seen as generated from a function mapping inputs to outputs, and the goal of the learner is defined as approximating that function. This leaves the question on how to do better than the demonstrator (see for example [67] for an imitator that outperforms the demonstrator). The question is discussed in [96] and one solution offered are to either filter out bad demonstrations or smooth them over with regression techniques. This diverges from the stated definition of success where the learner is to approximate the function that generates the demonstrations, and it does not deal with the case where the teacher is never able to achieve optimal performance (for example attempting to teach a robot how to throw a ball as far as possible, where the robot could in principle throw the ball much further than the teacher). The other approach

presented is to seek feedback. This is strongly in line with the approach taken in this thesis where multiple sources of information is used by a learner to figure out what a teacher wants it to do, but it falls outside of the [96] formalism. In [97] for instance, where reinforcement learning is used to improve on suboptimal demonstrations, the success criteria of the reinforcement learning framework is used and the demonstrations are useful to the extent that they speed up learning (a learner that is only interested in maximizing a reward function could tackle the questions of who and when to imitate in a way similar to [40].

In [138] an attempt to categorize imitation learning is made. The focus of this work is to classify various imitation learning tasks in terms of what type of goal the demonstrator would like the imitator to perform, for example replicating the exact movement, or replicating the end state of an object manipulated. This focus is not the same as what is attempted here, where defining success is an important aspect.

In [139], a formalism is presented for learning from demonstration. It deals with tele operated robots in cases where the demonstrator has a clear understanding of its goal. Besides the fact that it is not restricted to tele operated robots, the presented formalism is also more general in that it covers other types of information sources (eye gaze, a reward button, facial expressions, speech coments, EEG readings, etc) as well as demonstrations. The information spaces and related ideas are however quite similar, and the presented formalism can be seen as building on the ideas of [139].

2.2.2 Formalisms of feedback learning

Reward maximization [140] is a well formalized and well explored research area with significant overlap. A special case of reward maximization is the setup where a teacher is giving rewards to a reward maximizer, and the reward maximizer is trying to figure out how to get the maximum amount of reward. If the best way of getting reward is to do what the teacher wants it to do, the problem faced by the reward maximizer is identical to the problem faced by the learner as specified in the formalism proposed in this thesis. The learner wants to figure out what the teacher wants it to do since doing what the teacher wants it to do is its primary goal, and in this case the reward maximizer wants to figure out the same thing since it is the best way to get a reward. One example of where the optimal behavior of the two formalisms diverge is when it is possible to hide a probable mistake from the reward giver, and doing so will result in higher expected reward. If making sure a teacher sees the probable mistake will result in more informative feedback, this could very well be the optimal solution for a learner (while the reward maximizer will always maximize the reward given).

2.3 Related work in language learning

The adoption of linguistic convention can be seen as the adoption of normative rules and therefor covered by the presented formalism. This research is also relevant due to the link with presented experiments. Some of the experiments presented in this thesis will involve an artificial learner that is observing one human teacher and one other human, referred to as an interactant. The teacher responds to the current situation, which will be referred to as the context, consisting of objects as well as actions of the interactant (speech utterances and hand gestures). The learner will perceive the context as the position of one or several objects, as well as the continuous, non symbolic, representation of the interactants behavior. An interactant speech utterance or hand gesture is transformed from the raw sensor readings into a point in a three dimensional space, making them similar to the object positions from the learners point of view. The part of the context that makes the teacher act could be an object position, a speech utterance or hand gesture of the interactant. The learning of tasks that look linguistic to an outside observer is here viewed as not different in kind from the learning of tasks that depend only on object positions. Many of the methods for imitation learning described above does not make assumptions regarding the type of inputs the imitator is learning from, making them suitable for learning of skills involving speech or gestures. The fact that we are now dealing with an imitator that is learning how to respond to speech means that it is useful to contrast these experiments with various approaches in language learning research, especially the approaches that involve both language and action. The interaction of language and action has been studied both by examining natural systems and by building artificial systems. The following survey of language learning research is taken to a large degree from the article [103].

2.3.1 In natural systems

Even though language is no longer treated as an abstract symbolic system, autonomous from its users and its use in a physical and social reality, it is still seen in the litterature as a system separate from the action system. Even when the language system is grounded in the action system, they are seen as separate. There is now an extensive literature exploring the links between action and perception, founded on the central theoretical hypothesis that sensorimotor skills/action, social interaction skills, and linguistic skills develop in parallel and have a strong impact on each other [5][1]. As argued in [5], a recent review paper establishing a research roadmap for this domain in the future, a central challenge is the understanding of how language and action-perception learning and representations are integrated, both functionally and in their brain and social substrates.

This theoretical approach is actively pursued in several cross-fertilizing domains [6][2][3][1]. First of all, neuroscience has highlighted the strong interactions and interdependances of brain areas related to language and action-perception [4][6]. These brain interactions

support corresponding theories in both developmental psychology and cognitive linguistics about how language is grounded in action and perception [8] [9][10][100][14][15][12][13]. A central idea in this litterature is the argument that meanings are fundamentally rooted in/represented in terms of sensorimotor affordances, i.e. the potential actions that are associated to a given referent, even for abstract linguistic concepts [12][13][10][14][15]. This idea has been explored in particular to try to address the "symbol grounding problem" [16]. The difference with the approach proposed in this thesis is that there are still symbols that needs to be grounded (in this case by being represented in terms of sensorimotor affordances), instead of using a more general strategy that adopts linguistic conventions in the same way as non linguistic conventions. The learner has a strategy for adopting rules regarding how to respond to non linguistic contexts. It uses the same strategy when it is adopting rules regarding how to respond to communicative contexts by doing physical actions, performing communicative acts or performing internal operations. This dissolves rather than solves the symbol grounding problem.

In addition to the strong influences and constraints that action and embodiment impose on language, the Whorf-Sapir hypothesis at the beginning of last century [17] has proposed that perhaps language itself in turn can influence the way humans categorize the sensorimotor world. Recent experimental results [18] has added to the long running controversy related to these ideas.

Further than showing that language and action systems show strong interactions and develop in parallel, some research results even show that there are strong similarities in the very structure of the language and motor systems. Indeed, motor systems have been shown to be highly modular and compositional, in addition to the obvious capability of self-extension and learning [11][19][20][21]. These structural functional homologies are also consistent with the homology between the F5 region in the monkey and Broca's region in humans, pointing towards evolutionary linkages between the motor and language systems [1].

There are also other important ways in which language and action interact, especially as language acquisition happens mainly through social inter-*action*, where action and embodiment, through for example pointing and gazing gestures for achieving joint attention [24][25][26], are essential for helping language learners to guess the meaning of new words and constructions through the establishment of joint intentional understanding [22][23][24][25].

2.3.2 The role of shared intentionality in language acquisition

Shared intentionality is postulated by Tomasello to be important for language acquisition [23]. Two or more individuals are engaged in a shared intentionality activity if the goal of each individual is that the group succeeds, and this is common knowledge to each individual (the goal, and the fact that everyone share this goal, is part of the common ground). A simple example is two people jointly lifting a sofa and where both are certain

that they are working towards the same goal (this situation is importantly different from to two people, each holding an end of a sofa, and each having the same goal position of the sofa, but neither one having any idea why the other is holding the sofa, or what the other is trying to do). Shared intentionality facilitates language by reducing the set of possible meanings of communication and as a motivator for helpfully informing others. It is also interesting to note that some acts can be seen as somewhere in between communicative and non communicative, for example one of the people starting to very lightly tilt the sofa. In the case of shared intentionality this could be interpreted as "we should hold the sofa in another way".

The experimental setup of the language learning experiments presented in this thesis adds a complimentary way in which shared intentionality facilitates language learning. The proposed imitator is attempting to figure out what it should be doing from the actions of the demonstrator, which is fundamentally different from watching the demonstrator for the purpose of building a world model. It needs to notice "the demonstrator follows rule x", as opposed to discovering "the tribe has a set of established norms and acts in a way such that following rule x is beneficial". It is easy to imagine how this could be true in a human tribe when language is already established (for example since non linguistic individuals may have trouble forming social relationships and or have practical difficulties in cooperating). If it is beneficial for an individual to learn language, this can be seen as an enforcement mechanism (the tribe will treat those that adopt normative linguistic conventions better than those that do not). If an agent adopts normative rules without needing to see a benefit, it needs "only" figure out what rule the others are following. An agent that needs to see a benefit for it to adopt a normative rule has an extra inference to make before seeing the point in adopting normative linguistic conventions, and it needs to be born into a group that is already linguistic enough that there is a benefit to adopting these rules. For this reason, the latter type of learning is less conducive to language, both due to the extra inferences required, and due to the fact that language can only be learnt when there is an enforcement mechanism already established. An agent that has no predisposition to adopt normative rules could in principle discover that it is beneficial to it to adopt the local linguistic conventions by observation (if it is born into a group that is already linguistic) but it undoubtedly seems more likely that an agent that unquestioningly adopts normative rules will do so (and it would do so even if only its parents and a small number of others have these conventions). Let's examine how an imitator might come to adopt the rule "when I find berries I should say "berries" and point them out to people that do not know about them" using either one of two types of imitation learning. If the learner is only using the observations of the demonstrator to build a predictive model of the world, it would need to observe not only other people doing this, but it would also need to observe instances where this is not done, and the person not doing it was punished by others (or missed out on a reward given to those that follow the rule). This requires not only that they know that it was the berries that triggered the action (which is a common problem for all types of imitators), but also that they figure out that the reward/punishment was due to following/not following the rule (as opposed to all other things the punished individual

did and did not do). It also requires that the rule is already so well entrenced that it has a system of enforcement. An imitator that is using observations of the demonstrator to figure out what it should do, "only" needs to know that it was the berries that triggered the utterance. The behavior might be adopted without already being well established and without any established enforcement mechanism (and there is no need to observe any negative examples and figure out what triggered the punishment).

In the account of Tomasello, chimpanzees do not participate in activities of shared intentionality, and this is proposed as the main factor explaining their lack of language.

Chimpanzees are capable of estimating others perception, knowledge and goals. For example, when a chimpanzee wants an object that is controlled by a human they can differentiate between the human not trying to give the object and the human trying but failing [27] (they also understand that others make inferences [28] but not that others can have false beliefs; see [29] for a comprehensive overview of how they model other minds). To illustrate how the lack of shared intentionality hinders ape communicative abilities we can look at [32] where a chimpanzee that is looking for food and knows that the food is in one of three locations, will not favor the location pointed to by a human. The chimpanzee follows the pointing to the location but does not assume that the human is trying to help it, and thus the location the human is indicating is not assumed to be more likely to contain the food than other locations (if the human appears to be looking for food and tries but fails to reach the location, then the chimpanzee understands the behavior and favors this location). To illustrate how motivation to communicate helpfully can facilitate language learning, experiments have shown that an orangutang do not point at a searched for tool unless the tool will be used by a human to get something for the orangutang [33] (this significantly reduces the number of instances of attempted communication compared to someone with a motivation for helpful communication).

Since chimpanzees does "action x gets me y" type imitation learning, their gestures are instead learnt in the form "Chimp1 notices that when Chimp2 raises its arm, then Chimp2 will initiate play (raising the arm is a preparation to play-hit), then Chimp2 notice that raising its arm induces Chimp1 to start playing" (this example is modified from [23]). Now Chimp2 knows how to initiate play with Chimp1 using a gesture. This type of learning obviously enables two chimpanzees to establish a linguistic convention, but it does not seem to result in the propagation of a large set of diverse linguistic conventions/normative rules within a population or across generations (learning to follow a rule to avoid punishments will not work here since there is no established enforcement mechanism, such that not starting to play when observing a raised hand results in punishments, and no clear path to the establishment of such an enforcement mechanism). In contrast, a human child might observe and imitate the normative rules of the "raised arm" convention (and as it passes through generations its meaning could change and/or become more general, simply due to imperfections in the rule adoption strategies used).

In [30], the strongly related concepts of "we-mode" versus "I-mode" and collective intentionality are discussed (see also [31] for an early and highly related discussion of different ways in which the meaning of a sentence is understood, for example due to an extensive

common ground).

The account of Tomasello is focused on the speaker, while most of the experiments presented here deal with responding to requests. However, the second language learning experiment shows how the proposed setup can be used to learn to produce communicative acts as a response to the environment. The same algorithm is used to concurrently learn to: (i) respond to speech, (ii) respond to hand signs, (iii) produce hand signs as a response to the physical environment, and (iv) produce hand signs as a response to speech. The speech of an interactant is not treated differently from the position of an object, therefore these tasks can be learnt from a set of unlabeled demonstrations of an initially unknown number of tasks. The linguistic/non linguistic distinction lies in the eye of the beholder, and we shall argue that the experiments show that the distinction is not actually a fundamental one.

2.3.3 In artificial cognitive systems

The multi-discplinary approach in neuroscience, psychology and cognitive linguistics is strongly complemented by, and inspiring for, a flourishing landscape of computational modelling research projects [51][18][52][54][55][58][48][7][49][60]. Aiming at building computational and robotic models of the evolution and acquisition of language, most of these projects have tried to address centrally the grounding of language into action and perception, i.e. the symbol grounding problem (see [16] for Harnads original statement of the problem and [38] for a more recent attempt to dissolve some of the confusion that has arisen in the long debate about symbol grounding).

A central idea of research is now that: 1) symbols should be grounded in the sensorimotor flow, and in particular: the meaning of symbols should be expressed in terms of sub-symbolic affordances; 2) these associations and sensorimotor representations should be learnt by an embodied and situated robot rather than pre-programmed by an external human engineer [38][55].

Following this general approach, computational and robotic models presented in the litterature focus on various aspects. For example, some models have primarily investigated the question of how acoustic primitives in the flow of speech, i.e. phonemes, syllables and words, can be discovered with little initial phonetic knowledge and associated with simple - often symbolic - meanings [39][41][42][43][44]. [46] explores both how to extract meaningful words as well as how word order syntax can be found. The syntax is found by examining in what order various dimensions are more likely to be described. For example the shape dimension would be seen as related more to nouns, while color would be related to adjectives. The learner then deduce from data in what order words used to describe color and words used to describe shape have (in english, ”red ball” is for example more likely to be heard than “ball red”). In [47], the issue of how to detect words from streams of phonemes are investigated by putting a robot in front of talking humans, and have the robot produce first random babbling, then sounds biased towards

the syllables it has found.

Some other models have assumed the existence of quasi-symbolic word representations, and focused on understanding how neural networks could learn to associate these linguistic labels with meanings expressed in terms of simple action sequences also encoded by neural networks [48][7][49]. Yet another family of models investigated the Gavagai problem [22] mentioned above 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 [51][18][52][54][55][58], pure statistical approaches through cross-situational learning [45][59] or more constrained statistical approaches [60]. It is interesting to note that in most of these models focusing on the Gavagai problem, meanings were mostly expressed in terms of perceptual categories (e.g. in terms of shape, color, position, etc), and the exploration of complex action learning has so far not been well explored (but see [52][7][49]). Still another family of models has focused on trying to understand how basic learning of coupled linguistic and action compositionality can emerge out of general neural networks using simple models of syntax [48][7][49]. In [53], the idea is proposed that both linguistic conventions and other, more action centered, normative rules might be adopted by first forming a highly simplified model based on holistically analyzing observations. Then using this model to perform actions that are probably far from perfect, but that may be close enough to good actions that they elicit useful corrective feedback. Finally, some models have been assuming these capabilities to handle basic compositionality and have explored how more complex grammatical constructions and categories could be formed and still be grounded in sensorimotor representations [58][45].

As mentioned before, all these approaches to links between language and action make the assumption (in a more or less explicit manner) that language and sensorimotor processes are two interacting but *separate* processes.

Indeed, this is what is implicitly implied from the start when a theory makes the hypothesis that language and action developp "in parallel", and sets its own target as understanding how they are "integrated". The very definition of the "symbol grounding problem" also makes this assumption: there are on the one hand symbols of language, and on the other hand sub-symbolic sensorimotor processes, and the question is how to link these two apparently quite different spaces and associated processes. Similarly, asking questions such as in [5]: "Why do language and action share such hierarchical and compositional structure and properties? Is there a univocal relationship between them [...] or do they affect each other in a reciprocal way?" implicitly states that there are two separate processes whose linking is to be understood. This theoretical assumption is directly reflected in the whole landscape of computational models of language evolution and acquisition, where models typically start from cognitive architectures with two big modules, a sensorimotor module that encodes sensorimotor experiences and a linguistic module that encodes linguistic "symbols" (either directly with symbols [58] or through

nearly equivalent numerical encodings [48][7][49]), and mechanisms are introduced for allowing a robot/agent to learn the right associations between the (compositional) meanings and their (compositional) linguistic symbols through adequate coordination of these separate modules (even if they are sometimes implemented with nearly identically structured neural networks such as in [49]). Also, implementing such an assumption makes it difficult to improve the understanding of how an organism can discover the use of acoustic waves produced by the vocal tract, i.e. speech, (or some other form of input like gestures, body language, facial expressions, eye gaze, written symbols, etc, etc, etc) as a tool that can be used to direct the attention and action of others (and oneself), hence how an organism can discover the very concept of language rather than simply the associations between certain meanings and certain linguistic constructions. Remember how chimpanzees can discover the "raise your arm to initiate play" convention. This rule can not be said to have originated as a symbol whose meaning was grounded. Even if this would happen in a group of humans, and the rule would be imitated, generalized and transmitted through generations, it does not seem like the rule or the learning mechanism would need to involve symbols at any stage (and this does not change even if the convention changes significantly over time, or if the medium shifts to speech instead of hand signs).

It is surprising that the idea of one single system has been overlooked so far, especially given the strong structural similarities of the language and action acquisition and representation systems. Additionally, more and more theories have suggested, and experimental results have indicated, that language acquisition in human children may happen with very little (or even no) help from language specific innate neural circuitry or language specific capabilities [14][23][61][62], which makes the idea of a single system quite natural.

Thus, it seems like the central thesis of the language learning part of this thesis is original in putting forward the idea that bootstrapping the acquisition of fundamental elements of language might be a particular result of a more general mechanism for learning complex context-dependant sensorimotor skills. For spoken language, the context would simply include the acoustic waves produced by some agent in a particular situation and the rule would dictate that this context should trigger a particular sensorimotor or cognitive response (which can be a standard body movement, but also the production of a replying relevant acoustic wave, manipulating an internal cognitive structure, changing attention, etc, etc).

There are probably multiple reasons why such an approach has been overlooked so far. First, researchers focusing on language acquisition have often considered only a simplified view of sensorimotor learning and reversely researchers focusing on complex sensorimotor/action learning usually do not consider language as a central object of study. Also, the heritage of cognitivism and artificial intelligence, which considered only symbols as their object of study, still imposes the concept of symbol as central to language in most of research projects aiming at understanding how these symbols can be grounded.

As seen above, it is possible to formulate qualitatively the Gavagai problem of language acquisition as a particular case of what we may call the sensorimotor Gavagai learning problem. We have seen how this can be can be done more technically and formally, starting with the methods discussed earlier in the context of learning multiple tasks from unlabeled demonstrations.

2.3.4 Generalizing these computational approaches so that they can deal with language as a special case

As mentioned, the research community that elaborated the techniques described in the previous section have mostly been focusing on teaching "traditional" motor skills to a robot, typically body movements contextually depending on the potentially dynamical and absolute/relative properties of the body or objects around, and did not consider language and linguistic skills as a central issue to be addressed.

For an exception to this rule, see [83] and [84] which deal with both the communicative acts of another human as well as with multiple tasks and the problems of finding the number of gestures of an interactant. It is perhaps the work that is most related to the presented language learning experiments, and it further investigates how one could segment a continuous stream of demonstrator behavior into discrete demonstrations (something not handled by the experiments presented here). The three main technical novelties of the work presented here as compared to the research in [83] and [84] is the concurrent learning of linguistic and non linguistic tasks, the imitation of communicative acts (as a response to the position of an object or as a response to other communicative acts), and finally the imitation of unseen internal cognitive operations (minor novelties are the introduction of speech in addition to gestures and the various new algorithms). The conceptual novelty of the work presented in this thesis is the description of the link between those techniques and the scientific debate around the relation between language and action learning.

An agent using a single strategy for rule adoption can learn skills such as "look at the rabbit when you hear the acoustic wave "rabbit"", but also skills where speech can be generated as a response to another speech context, hence a form of primitive dialog, such as "if you hear the acoustic wave "Where is the X?", pronounce the acousitic wave "X is at Y" where Y is the location of X". Thus, in this view linguistic words are no longer considered as symbols to be necessarily associated with meaning, but are just acoustic waves which can modulate in a context-dependant manner (and potentially compositionally) the dynamical system that drives the learning agent. When observing and imitating such a linguistic exchange between a demonstrator and an interactant, the imitator will need to infer how internal cognitive structures are changed by word order etc, and then how those internal structures modify behavior in order to succeed in novel situations. This dynamical system approach where linguistic symbols disappear is close to the one proposed in [49], which has argued that at the same time this allowed the symbol grounding problem to also disappear. Yet, our work goes further than

[49] in several respects: 1) we do not consider a cognitive module/network for language processing that is separate from the rest of sensorimotor processing; 2) we do not assume that words are already encoded into crisp perceptual categories, but process low-level uncategorized speech streams which are then considered as equal contextual dimensions, just as any other sensorimotor dimension. In the first two experiments that we will present, we also do not assume that the modality that the words are expressed in are known (it can be acoustic waves or hand gestures). In the second experiment, the robot learns several tasks at the same time where some are triggered by sign language and some are triggered by speech input from an interactant. 3) we address the Gavagai problem whereas [49] already manually encoded the dimensions that were relevant for the task to be learnt. Our learner finds out whether or not the speech and gesture channels are relevant for what task to be performed and finding out what should influence the policy of the different tasks. However, as opposed to [49] who focused on the learning of compositionality, we will only investigate a simple form of word order syntax (in the third experiment).

Actually, while the connection between context-dependant motor learning and language learning is largely missing in this litterature, intermediary steps in this direction have already been taken. Indeed, a few researchers have began to use these technical approaches to learn interactive skills by imitation/demonstration. [50] investigates communication and collaboration between a human and a robot with symbol like motor primitive representation of both communicative and non communicative actions and using neural networks. See also [85] and [86] which used the approach initially describred in [87] and based on Gaussian Mixture regression for modelling the context-dependant sensorimotor policy to teach a robot how to jointly manipulate a large object with a human. In this work, the context includes the properties of the behaviour of another human interacting with the demonstrator. Further than encoding the external behaviour of an interactant, [93] have presented a related approach (but based on discrete representations of policies and states) where the Leo robot watches a human and builds a model of that humans world model. When the human is not attending to a box with chocolate in it, the chocolate is removed and later when the human attempts to open the box, Leo hands him some chocolate since this is what he was most likely to try to accomplish. See also [94] for a related experiment involving buttons (some of them not visible to the human). As will be illustrated with computational experiments, this setup can be readily extended with a demonstrator that shows a skill consisting in interacting with an interactant where the inferred mental state of the interactant determines the context of the skill.

Chapter 3

A formalism for the field of social learning

Contents

3.1 A formalism for step one: finding $u^\star$ 29

3.2 A formalism for step two: finding Ω 32

3.3 A description of the unsimplified setup 42

3.3.1 What situation is the formalism designed to deal with? . . . 42

3.3.2 Informed preferences and a formal success criteria for the learner 43

3.3.3 What is the purpose of the simplified setups 48

3.4 Removing further simplifications 50

3.4.1 Step three: allowing the learner to actively gather valuable data. 50

3.4.2 Step four: dealing with a world that is not perfectly visible to the teacher 52

3.4.3 Step five: dealing with a world that is not perfectly visible to the learner 54

3.4.4 Step six: finding Ω without a known generating distribution . 54

3.5 Step seven: viewing existing learning algorithms, operating in the unsimplified setup, as testable interpretation hypotheses 55

3.5.1 Graphical representation 59

By describing research in several different fields within a single unified framework, the formalism attempts to provide new ways of understanding existing research hopefully leading to new ways of extending and combining them. The existing research that the formalism attempts to cover includes any situation where a learner is trying to figure out what a teacher wants it to do. The types of information sources the learner learns from could for example include a teacher performing a demonstration, looking disappointed with its own demonstration, looking at a certain object while the learner is performing a reproduction attempt, making a speech comment during the reproduction or pushes an evaluative plus or minus button. All these information sources are here seen as being

of a similar type. Each can provide information to the learner about what the teacher wants it to do, to the extent that the information source can be interpreted.

The typical learner that the formalism is aimed at is analyzing multiple sources of information at the same time and is refining its understanding of some of them, based on what it learnt from others. Let's take the example of a learner that is able to see, but not fully understand, demonstrations, speech comments, facial expressions, a numerical value provided by the teacher, tone of voice, EEG readings of the teacher and eye gaze. If at least one of these are reasonably well understood, in at least some situations, it is possible to learn some types of tasks. If the learner is able to interpret some types of demonstrations, it can learn a task by looking at only this modality. Then it might be able to see from its history that some type of speech inputs are related to performance, and that if the teacher is observing the learner, then some facial expressions are often made right after the learner made a mistake during reproduction. If there is not enough data to decide how to interpret some speech comments, it might form multiple hypotheses. The ability to interpret facial expressions and speech can now be tested and refined in a new task and if validated, they can be used to learn new tasks. These new tasks might allow it to figure out that eye gaze at an object is correlated with it being important. It could also learn that when the teacher has observed all relevant aspects of the reproduction, the numerical value it provides is correlated with the performance of the action that the learner has just taken (relative to the average performance of a few of its most recent actions). New tasks allows new hypotheses of how to interpret information sources to be formulated, validated and refined. This in turn allows the learning of new tasks and perhaps the reinterpretation of old data. The learner could for example re examine a large amount of old data in light of everything it has learned, and discover that for some object it is very important to not bump into them, and that when the learner does so, or is close to doing so, there will be a certain type of EEG reading. It could also figure out that, for this particular teacher, the numerical value is actually more related to policy similarity with good actions than with absolute performance. The learner could now use this when learning new tasks, and it could also actively test these new ideas by performing actions that it expects to generate observations that will allow it to validate or invalidate them. The class of agents described above is the archetype that the formalism is designed to deal with, and it provides a general framework that is able to describe any agent that is trying to figure out what a human wants it to do. The formalism also tries to make as few assumptions as possible regarding the type of behaviors that is interpreted.

The formalism will be presented in a series of seven progressively more complex situations or setups. Some issues are easier to explain when other complexities are removed, and hopefully the first steps will convey some fundamental insights that will make it easier to describe the fully unsimplified setup. First, a mathematical definition is provided below for two simplified setups as well as a discussion of how they can be solved. In the first one, the learner is assumed to know how to interpret the teachers behavior, and the learner only needs to learn tasks while in the second setup this assumption is relaxed. Then before continuing to relax simplifications, the unsimplified setup that we

are aiming for is described in words to help describe what it is that is being simplified. This is followed by five more setups where more simplifications are dropped step by step. At each step, the formalism is modified to deal with the new complexity and new solutions discussed. In the final seventh step, the learner is dealing with an unstructured real world situation.

3.1 A formalism for step one: finding $u^\star$

In the most simple setup the learner is required to output a policy based on a given data set of interactions. A learner is located in a perfectly visible world with exactly one correct representation, known to the learner (we can say that the world has a single correct ontology, known to the learner), and is perfectly observing the teaching signal. The teacher is also known to perfectly observe the world, and it has direct access to a utility function over world-action pairs, that takes everything into account (including all future consequences). This utility function maps states in the learners action space and the world states to a real valued number (the situation can be roughly described as "the teacher knows what actions it wants the learner to take").

The learner also knows the mapping from what the teacher wants and what the teacher observes to states in a teaching signal space. A simple teacher giving demonstrations could for example be of the form "the teacher gives demonstrations which are perfect with probability 0.8 and otherwise perform random actions" or "the teacher rewards incremental progress and gives a scalar value feedback equal to the utility of the current action minus the average utility of the learners 6 previous actions". Knowing this mapping allows inference even if it is stochastic as each possible utility function results in a probability or a probability density for the actually observed feedback. To be more specific, it is necessary to first introduce some notation:

- **World state** $s = (x_1, x_2, ..., x_{N_S}) \in C^s$. An N_S dimensional vector in (in $\Re^{N_S}$ if it is continuous and unbounded), describing the state of the world. C^s is the space of possible world states. The learner has direct access to the world state in this step.
- **Action** $\alpha = (y_1, y_2, ..., y_{N_\alpha}) \in C^\alpha$. An N_α dimensional vector describing an imitator action.
- **Policy** $\pi \in C^\pi : C^s \rightarrow C^\alpha$. Since the world is fully visible, the imitators policy is definable as a transform from world states to actions.
- **Situation** $\Xi = \{s, \alpha\} \in C^\Xi$. A world state and a learner action. This is what the teacher will respond to by giving a teaching signal.
- **Teaching signal** $f = (z_1, z_2, ..., z_{N_F}) \in C^f$. An N_f dimensional vector describing the teaching signal response to a setup Ξ, for example a demonstration of what

action should have been performed, a speech comment on the imitators action, a scalar value evaluation of the action, the eye gaze towards an important object, etc.

- **Interaction** $I = \{\Xi, f\} \in C^I$. A setup, and the feedback that was given in that setup (a world state s, an imitator action α, and the feedback f that was produced by the demonstrator as a response).

- **Interaction history** $h = \{I_1, I_2, ...\} \in C^h$ is a set of interactions. To refer to an element E in h of interaction number t, we use E_t, for example: Ξ_t and f_t (the setup and response at interaction t). Note that the interaction history consists of states in spaces that are observable to the learner.

- **Learning algorithm** $\Upsilon \in C^\Upsilon : C^h \rightarrow C^\pi$. An interaction history to policy transform. Since the learners job in this step is only to output a policy as a response to data, a learner is defined by an Υ, a history h, and (in case Υ is stochastic) a policy π (there is no possibility to choose actions in order to get informative feedback). A learning algorithm/learner can be defined using an iterative update rule, modifying a policy based on one interaction at a time (since this recursively implies a unique Υ).

- **Utility function** $u \in C^u : C^s \times C^\alpha \rightarrow \Re$. Mapping world state-action pairs to a real number (expressing preferences over the action space, conditioned on the current world state).

- **Teachers utility function** $u^\star \in C^u : C^s \times C^\alpha \rightarrow \Re$. The teacher is assumed to have access to a utility function $u^\star$ that represents exactly what the teacher would have wanted if it where fully informed (for example regarding future consequences of the action). The sole evaluation criteria of the success of the learner is: $E[u^\star(\hat{\pi}(s_R))]$, where s_R is a randomly generated world state (the expected utility of its policy $\hat{\pi}$ when the state of the world is not known). Finally, the learner is assumed to know this fact (although it does not know $u^\star$).

- $u^\star$ **generating distribution:** $\mathcal{D}^{u^\star} : \Theta^{u^\star} \rightarrow \Re$. The utility function $u^\star$ is drawn from a distribution known to the learner. A known function class has a parameter space $\Theta^{u^\star}$, and each possible state is assigned a probability, or probability density, by the known distribution $\mathcal{D}^{u^\star}$. This distribution could for example be over discrete outcomes, or a density function over the continuous parameter space of a function class, or a probability distribution consisting of a density function over a continuous space as well as a set of dirac deltas for certain values in that space, etc. We denote the utility function that parameter $\theta^{u^\star}$ generate as $u(\theta^{u^\star})$.

- **Teacher signal generating transform** $\Omega \in C^\Omega : C^\Xi \times C^h \times C^u \rightarrow C^f$. A stochastic transform[1] from the current situation Ξ, the interaction history h, and

[1]This could denoted as $\Omega \in C^\Omega : C^\Xi \times C^h \times C^u \times C^f \rightarrow \Re$, but to emphasize that the thing

a utility function u into feedback f (the current world state, the current learner action, a utility function and the interaction history determines what distribution a state in the teaching signal space is drawn from). In this step the learner is assumed to have access to this transform (the next section formalizes a setup where this assumption is relaxed).

As the generating distribution $\mathcal{D}^{u^\star}$ over possible $u^\star$s is given, what is needed to get the posterior probabilities of possible $u^\star$s is the probability or the probability density of the observed feedback conditioned on all the different $u^\star$ hypotheses. Since $\Omega : C^\Xi \times C^h \times C^u \rightarrow C^f$ is known and the states in all the other input spaces are known, the probability of the observed feedback is only dependent on $u^\star$. If the probability (or probability density) of observing the feedback f_t at interaction t is denoted p^{f_t}, it is possible to write the same equations for the two cases of (i) a density function over a continuous $\Theta^{u^\star}$ space, and (ii) a probability distribution over a discrete $\Theta^{u^\star}$ space. If $\mathcal{D}^{u^\star}$ is a density function over a continuous $\Theta^{u^\star}$ space, then:

$$p^{f_t} = \int_{\theta^{u^\star} \in \Theta^{u^\star}} D^{u^\star}(\theta^{u^\star}) D(f_t | h(t), \Xi_t, u(\theta^{u^\star})) d\theta^{u^\star} \tag{3.1}$$

If $\mathcal{D}^{u^\star}$ is a probability distribution over a discrete space $\Theta^{u^\star}$ with $N_{u^\star}$ number of hypotheses u_i, and the prior probability that $u_i = u^\star$ is denoted $p_i^{u^\star}$, then:

$$p^{f_t} = \sum_{i=1}^{N_{u^\star}} p_i^{u^\star} p(f_t | h(t), \Xi_t, u_i) \tag{3.2}$$

Now, if the probability or probability density for observing f_t is denoted p, the posterior probability p_{px} of $u^\star$ hypothesis number x being correct is simply $p_{px} = p_{pa} p / p^{f_t}$, where p_{pa} is the a priori probability of $u^\star$ hypothesis number x being correct. The update factor p/p^{f_t} basically measures how good the hypothesis was at predicting the observed feedback compared to other plausible hypotheses.

Since Ω is a known mapping and the probability distribution over possible $u^\star$s is given, finding the posterior distribution over possible $u^\star$s given a history h has thus been cast as a textbook inference problem. Finding an optimal policy π given a finite history h is now a matter of maximizing the expected utility function (the weighted sum of all $u^\star$ hypotheses, or the integral over $u^\star$ space). See for example particle swarm optimization [132] or various methods for approximate bayesian inference [131]. $u^\star$ is defined in the action space given the observable world state, so the exact expected utility (given the known prior distribution over $u^\star$, and the fully observable history) of each action is known to the learner. It could be that even if Ω is known, finding the optimal solution is intractable, necessitating the need for approximate solutions. As simplifications are

generated is a teaching signal, the notation of a stochastic transform is used where $a : b$ means that a stochastically generates states in c according to a distribution that is dependent on states in b.

dropped in later sections, intractability will become an increasing problem, and much effort will be put into discussing approximate solutions.

Since the problem has been formalized in this way, standard ideas and principles of approximate optimization can be used to find approximate solutions. If $\Theta^{u^\star}$ is continuous and high dimensional, and there is a large history, then one possibility is creating a number of discrete $u^\star$ hypotheses, each with its own set of parameters. Then updating the probability of these hypotheses, and their parameters iteratively on one interaction at a time. The probability that an hypothesis is correct is modified in each iteration based on how well it predicted the actual teaching signal, and the parameters is modified so that it better predicts the observed history (and hypotheses that are very unsuccessful at predicting teaching signals from unobserved interactions can be rejected, and new ones constructed by doing alternate parameter modifications on good hypotheses, and the parameters of each hypothesis takes the others into account to avoid crowding in small areas and/or move towards regions that are good/highly populated).

The point is not that all problems within the framework can be solved, but instead that they can be cast as an instance of a well studied type of problems, and that standard ideas can be used to solve them (in the example above using the basic ideas of particle swarm optimization).

3.2 A formalism for step two: finding Ω

In this step the learner must learn to interpret the feedback of the teacher. Specifically, the Ω transform is no longer known, but it is drawn from a known distribution, and must be learnt in a way that is similar to how $u^\star$ was learnt in the previous step. Two different Ω hypotheses will in general give different probabilities, or densities for an observed interaction history, again reducing the problem to an inference problem of a well studied form (so that ideas from proposed approximate solutions can be used). New practical difficulties that arise, and new approximate strategies to deal with them will be discussed below.

The parameters of a known stochastic function class is drawn from a known distribution. Any parameter set results in a static (but not necessarily deterministic) mapping from an interaction history, the utility function and a current world-action pair to an output in a teaching signal space (for example "if I demonstrate something, and then the learner reproduces it wrong, I demonstrate again", "if the learner is doing better than usual, I will press a plus button" or "if the learner fails a lot and look like it needs encouragement, I will push a plus button[2]").

[2]Real humans do this, and the behavior is not caused by failure to observe the world or lack of knowledge about what the correct action is. Therefore this possibility is still relevant to the setup presented. A human learner is capable of noticing this type of teaching signal (for example tone of voice in combination with a partial understanding of the task) and is able to take this into account when making policy updates. Thus, an artificial learner should in principle be able to do the same

As in the previous step, the learners job is to output a policy based on a given data set (an interaction history of known length), meaning that it still does not have to deal with the problem of choosing actions in a way that trades of the maximization of information with actually performing the task. First some additional notation is needed:

- Ω **generating distribution:** $\hat{D}_{\hat{\Theta}^\Omega} : \hat{\Theta}^\Omega \rightarrow \Re$. Ω is drawn from a distribution known to the learner. A known function class has a parameter space $\hat{\Theta}^\Omega$, and each possible state $\hat{\theta}^\Omega \in \hat{\Theta}^\Omega$ is assigned a probability, or probability density by the known distribution $\hat{D}_{\hat{\Theta}^\Omega}$. This distribution could for example be over discrete outcomes, or a density function over the continuous parameter space of a function class, or a probability distribution consisting of an a density function over a continuous space as well as a set of dirac deltas for certain values in that space, etc.

- Ω **estimate distribution** D^Ω. If the learner builds a model of Ω with parameters, then distribution over this space is denoted D^Ω (due to tractability issues, this does not have to be the same as $\hat{D}_{\hat{\Theta}^\Omega}$). We denote the resulting Ω of parameter θ_k^Ω as Ω_k.

Just as in the previous step, the problem is to define an Υ, and success is measured in how well the resulting policy optimizes $u^\star$. Since the prior probability of each possible feedback generating transform is known, and the prior probability of each possible utility function is known, for each interaction history $h \in C^h$ there is at least one policy π such that the expected utility is maximized. That is: there is at least one optimal policy that, given the known information, will give maximum expected utility, and finding it is a textbook inference problem. Below, two examples are presented, and then generalized approximate solution strategies are discussed since intractability is a likely practical problem. The problem of finding a $u^\star$ from a partially known Ω is similar to finding $u^\star$ from a known stochastic Ω, as a set of stochastic Ω hypotheses (weighted by probability) reduces to a single stochastic function. The practical difference is that in an approximate solution, it is possible to update Ω hypotheses concurrently with $u^\star$ hypotheses in an EM inspired way.

An example with a discrete set of possible teachers

$\hat{\Theta}^\Omega$ consists of N discrete possibilities, denoted $\theta_1^\Omega, \theta_2^\Omega, ..., \theta_N^\Omega$, where each θ_n^Ω results in a unique transform Ω_n. We denote the probability that $\Omega_n = \Omega$ as p_n^Ω. Before

(by for example: (i) first failing at a task where the goal is known, (ii) then noticing that there is a statistical pattern in tone of voice space correlated with "failures getting positive feedback", (iii) then confirming the theory in a unrelated setting, (iv) building a detailed model of when this happens, and with what probability, (v) and finally using this during learning in novel settings by keeping track of the probability that a particular teaching signal instance was generated like this (and take that into account during policy updates).

observing the interaction history the learners estimate of Ω is in this case identical to the generating distribution $\hat{\Theta}^{\Omega}$ (if the generating distribution had been to difficult to handle computationally, the learners initial estimate could have been something simpler). The learner observes a single interaction $I^1 = \{\Xi^1, F^1\}$.

The learner also has M hypotheses of what $u^\star$ looks like (also initialized with the generating distribution), $\theta_1^{u^\star}, \theta_2^{u^\star}, ..., \theta_M^{u^\star}$, where each hypothesis is denoted $u_m^\star$. Since there is a discrete set of $u^\star$ and Ω hypotheses as well as a single single interaction $I^1 = \{\Xi^1, F^1\}$, the probabilities of the $u^\star$ hypotheses can be updated with a simple equation. The probability of observing the feedback F^1 given the history, setup, hypothesized $u^\star$ and hypothesized θ_n^Ω is denoted $p(F^1|\Xi^1, h, u_m^\star, \Omega_n)$. We have $M \times N$ possibilities, each corresponding to a utility function-transform hypothesis pair. Each pair has a prior and each assigns a probability to observing the actually observed feedback. This means that each pair can be assigned a posterior probability. For transform hypothesis n and utility function hypothesis m the posterior pair probability is $p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)$

Thus the probability (after updating on the new observation F^1) of each utility function hypothesis and transform hypothesis is given by simply summing the posterior pair probabilities. We denote the probability at time step t as ${}^t p_m^{u^\star}$ so that the probability ${}^2 p_m^{u^\star}$ is the probability that utility function hypothesis number m is correct, after updating on observing F^1. ${}^2 p_m^{u^\star}$ is thus simply the sum:

$$ {}^2 p_m^{u^\star} = p_m^{u^\star} \frac{\sum_{n=1}^N p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)}{\sum_{m=1}^M (p_m^{u^\star} \sum_{n=1}^N p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n))} \tag{3.3} $$

And in just the same way we have the new probability ${}^2 p_n^\Omega$ (the new probability that transform hypothesis number n is correct, after updating on observing F^1) in the sum:

$$ {}^2 p_n^\Omega = p_n^\Omega \frac{\sum_{m=1}^M p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)}{\sum_{n=1}^N (p_n^\Omega = \sum_{m=1}^M p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n))} \tag{3.4} $$

An example with a continuous space of Ω parameters

Now we take the exact same setup, but we have a continuous parameter space of possible Ω transforms. The exact same reasoning applies when it comes to updating the discrete set of ${}^2 p_m^{u^\star}$, with the only difference being that integrals replaces sums, so that we get:

$$ {}^2 p_m^{u^\star} = p_m^{u^\star} \frac{\int_{\Theta^\Omega} D_{\theta^\Omega} p_m^{u^\star} D(F^1|\Xi^1, h, u_m^\star, \theta^\Omega) d\theta^\Omega}{\sum_{m=1}^M p_m^{u^\star} \int_{\Theta^\Omega} D_{\theta^\Omega} p_m^{u^\star} D(F^1|\Xi^1, h, u_m^\star, \theta^\Omega) d\theta^\Omega} \tag{3.5} $$

If the parameter space of possible transforms is high dimensional (that is, there are many ways in which the demonstrators feedback behavior could vary), this integral might be completely intractable. But the problem has at least been reduced to a much

more standard form, and it is clear what exactly an approximate solution is trying to approximate.

One approximate solution is to create a set of hypotheses for how feedback is generated, test them against data, and continuously modify them, discard them or create new ones. We need a hypothesis generating algorithm, an algorithm that tests and modifies or discards hypotheses and an iterative procedure for concurrently updating the $u^\star$ hypotheses and the Ω hypotheses. Let's call such an Ω hypothesis an interpretation hypothesis (as it is an hypothesis regarding how feedback should be interpreted) and denote interpretation hypothesis number i as Π_i.

To test the quality of an hypothesis in this example (with a single data point), we can not do better than check how well the transform model predicts the observed behavior (and of course look at the known density function of transform generators). Any test will be strongly dependent on the current estimate of $u^\star$, since the feedback is dependent on both $u^\star$ and Ω.

Given a test that rates a Π_i conditioned on the current best guess of the $u^\star$, we can update our set of $p^{u^\star}$ probabilities based on the current set of Πs, concurrently with updating our set of Πs based on the current set of $p^{u^\star}$. This shows how the old ideas behind various Expectation Maximization (EM) algorithms can be used when the imitation learning problem has been reduced to this form. The basic idea that can be taken from these algorithms is that when there are two unknowns, and knowing one helps finding the other, updating both concurrently can lead to a functioning and tractable algorithm. An example is when a set of points is known to be generated by a known number of gaussian distributions of with unknown parameters. Knowing what generator generated what points help when estimating the parameters of the generators, and knowing the parameters of the generators helps when estimating which points where generated by what generator. See for example Dempster and Lairds 1977 paper [109] presenting an EM algorithm.

An example with continuous parameter spaces and a large number of interactions

To illustrate the problem faced by an learner in this step, a more specific setup and solution strategy is introduced (this is just re using standard textbook ideas on the formalized problem, but being specific could still help illustrate the basic concepts).

Consider a problem where the teacher is known to have been drawn from the large dimensional, independent distributions D_Ω and $D_{u^\star}$, and where there is a large history h to learn from. It is in principle possible to find the posterior $D_{u^\star}$ conditioned on the a priori D_Ω and the history, but let's look at a class of tractable approximate solutions. We need a bit of notation:

- $\hat{U} = \{\hat{u}_1, \hat{u}_2, ...\}$: The set of discrete hypotheses regarding $u^\star$.

- $\hat{\Omega} = \{\Pi_1, \Pi_2, ...\}$: The set of interpretation hypotheses.
- $h(t) = \{I_1, I_2, ..., I_t\}$: The history up until time t.
- $D^{u^\star}$: The a priori density function over the possible utility functions that the teacher might have.
- $GenerateNew - \Omega - Hypotheses(h(t), \hat{\Omega}, \hat{U}, D^{\Omega})$. An algorithm that generates Ω hypotheses. If the set of hypotheses $\hat{\Omega}$ has any empty slots (either due to not being initialized or due to some Πs having been discarded), this function needs to create hypotheses Π that makes a tradeoff between being probable according to the prior probability D^{Ω}, being consistent with the data $h(t)$ (where the accuracy of the consistency estimate is dependent on the current estimate $\hat{U}$ of the teachers utility function $u^\star$) and being well distributed in the space (these Πs will be modified as a response to data, so several in the same small region could be wasteful as they might converge to the same point).
- $GenerateNew - u^\star - Hypotheses(h(t), \hat{U}, \hat{\Omega}, D^{u^\star})$. The tradeoffs are similar with the situation detailed above, and in this case the accuracy of the consistency estimate is dependent on the current estimate $\hat{\Omega}$ instead of $\hat{U}$.
- $Discard - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$. The Πs have been modified without any access to the interaction I_t, so it is suitable to test them. If an hypothesis does bad enough at predicting new observations compared to the others, it can be eliminated. Another reason to eliminate an hypothesis is that the modification process has made it to similar to another hypothesis. As before, the accuracy of the consistency estimate is dependent on the quality of the current $\hat{U}$ estimate.
- $Discard - u^\star - Hypotheses(I_t, \hat{U}, \hat{\Omega})$. The same types of concerns as above apply in this step.
- $Modify - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$. This algorithm updates the set $\hat{\Omega}$ of interpretation hypotheses based on the new interaction I_t and the current estimate $\hat{U}$ of the utility function. As the quality of the update is dependent on the accuracy of the current $\hat{U}$ estimate, it makes sense to update both estimates concurrently a few times in an EM inspired way.
- $Modify - u^\star - Hypotheses(I_t, \hat{U}, \hat{\Omega})$. This is the other half of the above mentioned EM pair.

With these functions we can build an iterative algorithm 1 that could hopefully approximate the integrals, while remaining tractable.

Algorithm 1 Approximate solution to example 3

Input: $D^{u^\star}$, D^{Ω}, T, $h(T)$

• $D^{u^\star}$: *The probability distribution that the teachers utility function* $u^\star$ *is known to be drawn from.*

• D^{Ω}: *The distribution that the teachers feedback generating transform is known to be drawn from.*

• T: *The number of interactions in the history.*

• $h(T)$: *The history of interactions.*

$\hat{U} \leftarrow GenerateNew - u^\star - Hypotheses(h(0), \hat{U}, \hat{\Omega}, D^{u^\star})$

$\hat{\Omega} \leftarrow GenerateNew - \Omega - Hypotheses(h(0), \hat{\Omega}, \hat{U}, D^{\Omega})$

for $t = 1$ **to** T **do**

  $\hat{\Omega} \leftarrow Discard - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$

  $\hat{U} \leftarrow Discard - u^\star - Hypotheses(I_t, \hat{U}, \hat{\Omega})$

  $\hat{\Omega} \leftarrow GenerateNew - \Omega - Hypotheses(h(t), \hat{\Omega}, \hat{U}, D^{\Omega})$

  $\hat{U} \leftarrow GenerateNew - u^\star - Hypotheses(h(t), \hat{U}, \hat{\Omega}, D^{u^\star})$

  while Stopping criteria not met **do**

    $\hat{U} \leftarrow Modify - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$

    $\hat{\Omega} \leftarrow Modify - u^\star - Hypotheses(I_t, \hat{U}, \hat{\Omega})$

  end while

end for

An example solved by a multiple generator based algorithm

The teacher is approximated as having a number of different teaching signal generators or interaction protocols denoted Γ. In each situation the teacher selects one based on the interaction history and the current setup.

Ω is approximated as a combination of generators $\Gamma : C^{\Xi} \times C^{h} \times C^{u} \rightarrow C^{f}$. Roughly speaking the teacher is approximated as having several ways in which it can interact, and as choosing which way of interacting (choosing which Γ will generate the teaching signal). More precisely, each Γ has a probability to be activated that is state space dependent, and is otherwise a stochastic transform of the same class as Ω, mapping the same input spaces to the feedback space C^{f}. Γs are not hypotheses in the sense of the Πs mentioned above since Ω is not hypothesized to be equal to any one Γ, Ω is instead modeled as built up by a set of Γ transforms. The proposed algorithm concurrently estimates how each generator will produce teaching signals, in what type of situations a generator is used (encoded as a triggering region in situation space for each generator), and what data was generated by what Γ. This is done in a way that is very similar to old and well known Expectation Maximization (EM) methods (just as in the examples discussed above). First some additional notation:

- **Feedback generator number** n**.** $\Gamma^{n} :\in C^{\Gamma} : C^{\Xi} \times C^{h} \times C^{u} \rightarrow C^{f}$. Generator number n that Π is built from.

- Θ^n_Γ is the parameter space of Γ number n.
- $D^t_{\Theta^n_\Gamma}$ is the probability density function over Θ^n_Γ at time t (the current estimate of the parameters of Γ^n).
- $\mathcal{D}^t_\Gamma = (D^t_{\Theta^1_\Gamma}, D^t_{\Theta^2_\Gamma}, ..., D^t_{\Theta^N_\Gamma})$ is a set of Γ parameter estimates.
- **Generating tendency** $G^n : C^\Xi \times C^h \times C^u \rightarrow \Re$. The tendency of basis function number n to generate the feedback. The probability that Γ^n will generate feedback is $\frac{NG^n(\Xi,h)}{\sum_{m=1}^N G^m(\Xi,h)}$ (the probability that a specific observed feedback was generated by Γ^n depends on the type of feedback that Γ^n tends to generate and what the generating tendency in the current state is).
- ${}^G\Theta^n$ is the parameter space of G^n.
- $D^t_{{}^G\Theta^n}$ is the probability density function over ${}^G\Theta^n$ at time t (the current estimate of the parameters of the generating tendency G^n).
- $\mathcal{D}^t_G = \{D^t_{{}^G\Theta^1}, D^t_{{}^G\Theta^2}, ..., D^t_{{}^G\Theta^N}\}$ is the estimate at time t of the generating tendencies.
- $\Theta^{u^\star}$ is the parameter space of $u^\star$.
- $D^t_{\Theta^{u^\star}}$ is the probability density function over $\Theta^{u^\star}$ at time t (the current estimate of the parameters of $u^\star$).
- **Generating probability** p^t_n**:** The estimated probability that the feedback f^t, observed at time t, was generated by Γ^n.
- **Generated feedback** γ^n**:** The current estimate $\gamma^n = \{p^1_n, p^2_n, p^3_n, ...\}$ of what feedback was generated by Γ^n.
- $\mathcal{P}^t_\gamma = \{\gamma^1, \gamma^2, ..., \gamma^N\}$**:** the estimate at time t of what feedback was generated by what Γ.

It is now possible to concurrently re-estimate: (i) The feedback behavior of the basis functions, (ii) Their generating tendency (iii) The set of feedback instances that was generated by each Γ, and (iv) The utility function $u^\star$. We can see this in algorithm 2, which is in turn based on the sub algorithms:

- ${}^{s+1}\mathcal{P} \leftarrow estimateGen({}^s\mathcal{P}, {}^s D^{u^\star}, h, {}^s\mathcal{D}^\Gamma, {}^s\mathcal{D}^G)$**:** Given the model at step s of $u^\star$ and the Γs, the estimate of which Γs generated which feedback is updated. The previous step updated the feedback behavior ${}^s\mathcal{D}^\Gamma$, the generating tendencies ${}^s\mathcal{D}^G$ and the estimate $D^t_{\Theta^{u^\star}}$ of $u^\star$. Given these new estimates, *estimateGen* must approximate the probability that a given basis function was the one that generated the feedback, under the new $u^\star$ estimate. Which Γ generated the feedback is

straightforwardly dependent on where that Γ is expected to generate feedback, and on how probable a Γ is to generate the observed type of feedback (conditioned on the new $u^\star$ estimate).

- ${}^{s+1}D^{u^\star} \leftarrow update - u^\star - estimates({}^{s}D^{u^\star}, {}^{s+1}\mathcal{P}, {}^{s}\mathcal{D}^{\Gamma})$**:** The estimate of what Γ generated the data and the feedback behavior of those Γs has been updated, which means that the data can be re interpreted and used to update $u^\star$. Given a fixed current model of Ω (consisting of a set of Γs), this reduces to a textbook supervised learning problem with data that has a known noise structure (the uncertainty of the Ω estimate and the stochastic nature of the various Γs). Again part of the problem is reduced to a well studied type of subproblem.

- ${}^{s+1}\mathcal{D}^{\Gamma} \leftarrow update - \Gamma - estimates({}^{s}\mathcal{D}_{\Gamma}, {}^{s+1}\mathcal{P}, {}^{s+1}D^{u^\star})$**:** The estimates of what data a given Γ has generated has been updated, but also in what situation it was generated since $u^\star$ has been updated. For example: a learner has observed a "good robot" comment when shooting a basketball close to a hoop. If the learner manages to figure out that the teacher only cares about whether or not a basketball lands inside or outside a hoop, then it can re interpret the feedback generating function. Specifically it can figure out that a failed attempt gets a "good robot" comment if the outcome is closer to good outputs than previous attempts, instead of for example rewarding incremental increase in performance (which would have been more likely if the teacher had instead wanted something like "shoot as close as possible to the hoop"). Given the known world state, and taking the current estimate of $u^\star$ and the current estimate of what points where generated by what Γ for granted, this reduces to a function approximation problem of a known form. Each Γ can have its own type of parameter space but the basic idea is still that of "freezing" all the other estimates and using them to update the generators (it's just not necessary to do the same type of update for each generator).

- ${}^{s+1}\mathcal{D}^{G} \leftarrow estGenTend({}^{s}\mathcal{D}^{G}, {}^{s+1}\mathcal{P}, h, {}^{s+1}D^{u^\star})$**:** Given the current best estimate of which generator actually generated which point, this is a textbook supervised learning problem with a labeled data set.

The algorithm rests on the same principle as building a Gaussian Mixture Model (GMM) with an Expectation Maximization (EM) algorithm that concurrently estimates what data points was generated by what gaussian (based on the current estimated properties of the gaussians), and estimating the properties of that gaussian (based on the current estimate of which points they generated). The algorithm illustrates how a vague problem to "do what the teacher intended the learner to do" has been formalized to the point where the exact solution integrals can be set up, so that tractable approximation can be found using standard techniques, see for example Dempster and Lairds paper [109] from 1977, explaining an EM method based on a very similar idea[3], and solving a very

[3]In both cases there is a data set which has been generated by a set of generators. The properties of the generators are not known, but if they where known, it would be possible to estimate which

Algorithm 2 A multiple generator algorithm to solve the problem in example 4

Input: $\mathcal{D}_0^G$, $\mathcal{D}_0^\Gamma$, $D_0^{u^\star}$, h, S

• $\mathcal{D}_0^G = \{D_0^{G1}, D_0^{G2}, ..., D_0^{GN}\}$ *is the initial estimate of the generating tendencies (at time 0).*

• $\mathcal{D}_0^\Gamma = \{D_0^{\Gamma 1}, D_0^{\Gamma 2}, ..., D_0^{\Gamma N}\}$ *is the initial estimate of the feedback behaviors.*

• $D_0^{u^\star}$ *is the initial estimate of* $u^\star$ *(a probability distribution)*

• h *is the interaction history*

• S *is the number of update steps*

for $s = 1$ **to** S **do**

$^{s+1}\mathcal{P} \leftarrow estimateGen(^s\mathcal{P}, ^s D^{u^\star}, h, ^s\mathcal{D}^\Gamma, ^s\mathcal{D}^G)$

$^{s+1}D^{u^\star} \leftarrow update - u^\star - estimates(^sD^{u^\star}, ^{s+1}\mathcal{P}, ^s\mathcal{D}^\Gamma)$

$^{s+1}\mathcal{D}^\Gamma \leftarrow update - \Gamma - estimates(^s\mathcal{D}_\Gamma, ^{s+1}\mathcal{P}, ^{s+1}D^{u^\star})$

$^{s+1}\mathcal{D}^G \leftarrow estGenTend(^s\mathcal{D}^G, ^{s+1}\mathcal{P}, h, ^{s+1}D^{u^\star})$

end for

similar problem. It is doing essentially the same thing as all the classical EM algorithms (even though the Γs can have different parameter spaces). The purpose of presenting this algorithm is not to present a general solution strategy to any imitation learning problem, but simply to demonstrate that the problem has now been formalized to the point where old standard ideas can be used. The question of solvability is now dependent on factors similar to those that determine whether or not the classical EM algorithms would find a solution (dimensionality, size and quality of the data set, etc).

Let's give a few examples of generators that actual humans might be modeled as being made up of:

- **Demonstrating as a response to failed reproduction attempt** Γ^1**:**. The triggering region G^1 would be where a demonstration was followed by a failed reproduction attempt, and could have parameters relating to how badly the demonstration has to fail, or if the relevant distance is in policy space or outcome space (a close basketball trow can be very close to optimal policy, but still have an outcome that is no better than any other failed attempt, so that if a reproduction needs to be far from optimal in policy space to elicit another demonstration, this would not be within the triggering region). G^1 could also contain an arbitrary amount of other parameters, such as the behavior being more likely when the task is easy for the teacher to perform, or when the teacher looks irritated directly after a learner reproduction attempt, etc. The irritated facial expression could be either a single value attached to the binary output of a fixed "irritated facial expression detector" (multiplying the triggering tendency with the parameter value

generator generated what data point. What generator generated what data point is also not known, but if it where known, it would allow us to determine the properties of the generators. Both things are given initial estimates, and then the estimates are updated concurrently (conditioned on the best current other estimates).

for instance) or it could include parameters regulating what counts as "irritated facial expression" (or more technically "facial expression that is correlated with triggering Γ^1"). Θ^1_Γ describes the feedback generating behavior, and could include parameters of how many mistakes the teacher makes (it could for instance be that the teacher does this only when irritated, which is correlated with tasks that are simple for the teacher, which correlates with good performance (lower noise than other demonstrations)).

- **Verbally evaluating progress Γ^2:** Saying things like "good robot","No!", or "great!" based on how good performance the learner achieved relative to its recent interaction history. Parameters could include the length and weighing of the recent history, the strength of the different words (does "great!" indicate better performance than "good robot", and if so, how much better?), the parameters of how to map speech input to a set of pre defined categories (with pre defined interpretation), the parameters of a transform from speech space to evaluation space, etc, etc. Γ^2 could include parameters regarding how much more likely this feedback behavior is in the case of eye contact, or in the case of a long interaction history consisting of the same types of actions, etc, etc (speech in the case of eye contact could for example be more likely to be relevant).

- **Pushing a reward/punish button based on absolute and relative performance Γ^3:** The reward button is pushed with a value based on: (i) how good the outcome is in an absolute sense, (ii) how good the outcome is compared to recent history, (iii) how close the action was in action space to good actions compared to recent history. The triggering of this behavior could be dependent on anything from the number of demonstrations made to the attitude (angry, happy, etc) of the teacher, leading to a large number of possible parameters of G^3. Θ^3_Γ could include a value defining what constitutes "recent history", the relative weighting of the different considerations, etc.

- **Pushing a reward/punish button to punish the robot for breaking something Γ^4:** Maximal punishment and a surprised and angry facial expression indicate that something was broken, which can help with credit assignment (the problem was not that the basketball was far from the hoop, it was that the basketball went through the window).

- **Pushing a reward button to encourage a robot that has failed a lot and who looks sad Γ^5:** The generating tendency G^5 can have parameters related to teacher facial expressions and eye contact (for example a distribution encoding something like: "Γ^5 was not the generator if the immediate teacher response after looking at the outcome of the learner action was a triumphant smile and a "great!" speech utterance"). This type of feedback is actively harmful to the learners ability to figure out what the teacher wants it to do, but it is still important for the learner to understand this behavior so that it can classify feedback as having been generated by Γ^5 (If the feedback was likely to have been generated by Γ^5, it can

for example be ignored, which is already a big improvement compared to updating policy as if it indicated success).

- **Looking at an object that the learner interacted with badly Γ^6:** When the learner fails, and one particular object is important for that failure, the teacher will tend to look at that object.

The idea behind the algorithm is also similar to Simultaneous Localization And Mapping (SLAM) in that knowing what position a robot had at each time step will allow the building of a good map, and knowing the map makes finding the positions much easier. The analogy with a robot moving around and trying to build a map and at the same time figuring out where it is within that map is useful as it makes the idea of active information gathering obvious (the robot can move to different places, or just direct its sensors as a way of testing competing hypotheses regarding both what the area looks like, and where it is within that area). This solution strategy will be discussed in simplification step three, where the data set is not fixed, and the learner must take actions for the purpose of obtaining as useful data as possible. This will lead us to another old field known as optimal experiment design. And as the resulting "expected information gain integrals" will normally be intractable, we will be making contact with various approximate methods for finding actions that result in good information, for example using biological systems for inspiration, and described in terms such as artificial curiosity. Intuitively, finding good strategies for gathering informative data seems to be a central question at least as important as analyzing that data. Even though this research area is very active and making progress, it is not close to finding neat solutions applicable to any problem, meaning that one might have to dig into all the messy details and integrate a specially designed solution to the active information gathering problem into the full learner architecture from the beginning (as a neat, of the shelf and fully general solution that can be plugged in as a separate module will probably not be available).

3.3 A description of the unsimplified setup

As we remove simplifications step by step, it might be useful to stop and give a rough verbal description of the situation being simplified and where we are trying to get to in the end (a text based description of what the math based formalism is trying to capture). This section describes the unsimplified situation before we move on to describing step 3 in the next section.

3.3.1 What situation is the formalism designed to deal with?

First a short and informal description of the unsimplified setup that the formalism should deal with is presented. A robotic learner and a human teacher are situated

in some unstructured environment. The human might have an idea of what it wants the robot to do, for example "remove dust", but this idea could be very vague. The teacher might also be uninformed in many different ways, for example about future consequences of possible learner actions. The task of the robot is to do what an informed version of the human would consider best for the uninformed version of the human. Informed includes the understanding of concepts (such as what types of actions are possible) and knowing about specific facts (such as long term consequences of actions, or the contents of a box[4]). The robot is **not** trying to perform the actions that the teacher would have performed, or the actions that would make the teacher say "good robot", or the actions that would make the teacher push a reward button, or anything similar to this. Finally, the learner does **not** have access to a sensor that tells it how successful it was. And it does **not** have access to a function over its inputs that specify how successful it was. This setup is interesting as real robots placed in unstructured environments, with non expert humans will have to operate under these conditions. Non expert humans in unstructured environments are not always well approximated as flawless feedback givers whose feedback has an easily encoded meaning. How to interpret the eye gaze or facial expression of a specific teacher will have to be learned just as how to interpret a failed demonstration or a reward button pushed because the teacher failed to notice something or in order to encourage the learner. A set of simplifications is introduced in section 3.3.3 which reduces this setup into an inference problem with a mathematically well defined success criteria. The section below deals with learning algorithms operating in the above situation without simplifications and introduces a new principled way of modifying and combining any set of learning algorithms that modifies a policy based on the behavior of some human, trying to build a policy that the human would approve of (for example learning from demonstration algorithms, or reinforcement learning algorithms that maximize the value of a reward button pushed by a human).

3.3.2 Informed preferences and a formal success criteria for the learner

The concept of informed preferences is designed to deal with cases such as "the teacher would like the learner to perform an action, but if it knew the consequences of that action, would prefer another action" or "the teacher is very happy with the end result after the learner has cleaned the apartment, but if it knew that the cleaning produced a lot of noise that disturbed the neighbors, it would not like the cleaning strategy". These preferences are specified over the learners action choices, and the goal of the learner is to execute preferred actions.

A teacher might lack knowledge about the world, fail to understand certain concepts, not

[4]The example of a box whose contents the teacher is misinformed about, and several other parts of the formalism, are inspired by the work done by Cynthia Breazeal and Andrea Thomaz, especially with the Leo robot [93]

have imagined all possible strategies, be unaware of all consequences of an action, want things due to a misunderstanding, etc. If the teacher would consider the knowledge relevant to the learners action choice if it were made aware of it, then that piece of knowledge is considered relevant. For the purposes of evaluating the relative desirability of the actions that are available to a learner in a specific context, a subset of all knowledge that the teacher is unaware of will be relevant. This subset will be denoted Σ (a set of pieces of knowledge that the teacher would consider relevant to a specific learner action choice, if it were made aware of it). If a version of the teacher that knew about a fact would consider the fact relevant, it would be included in Σ, if a version of the teacher that understood a concept would consider it relevant, the concept would be included in Σ (and similarly with knowing about a possible strategy, or resolving a misunderstanding that made it want something, etc). There are many ways to segment actions. The very same learners actions can be evaluated by a preference ordering over states in its motor output space, or they can be evaluated according to a preference ordering over sequences of motor primitives. The correct segmentation is that would be preferred by the teacher if it were informed. Anything considered relevant to its opinion about the segmentation is also considered relevant to the relative desirability of actions, and thus in Σ. If all things in Σ were acquired by the teacher (facts known, concepts understood, etc), then the resulting person is referred to as the informed version of the teacher. If the informed version of the teacher has an opinion about what would be best for the actually existing uninformed version of the teacher, then this is defined as the informed preferences of the teacher (a preference ordering over the learner actions that are available in the current situation). If the learner faces the decision of whether or not to show the teacher what is in a box, and the informed version of the teacher already knows what is in there, then it might want different things for itself and the uninformed version of itself (since the decision can be different, it matters that the decision is about what is best for the uninformed version). The learner is now defined as a set of interpretation hypotheses and success is judged according to the informed preferences of the teacher.

Let's explore the case of a robotic learner providing security for a building using a camera, a microphone and an alarm. It is also able to move around the building, send video and microphone recordings over the internet to its teacher Steve, and receive commands and feedback from Steve. Steve is expecting that Bill will break into the building and has bought the robot as a way to get revenge on Bill (by showing video recordings to the police of Bill committing a crime), and has provided the learner with pictures of Bill, examples of what type of video would hold up in court, etc. The learner further knows that it is very expensive and that if it is detected, it might get stolen. The learner sees a truck drive through a wall driven by a single masked person. The person gets out of the car and start taking things and putting them in the truck, and this person is very clearly much taller than Bill. The learner hides, triggers the alarm and starts sending a video feed to Steve. Steve is at home when he is alerted by the alarm and immediately sends the command "get a picture of Bills face", a huge negative scalar feedback when he sees that the learner is hiding, and then the command "move forward" (which would result in the learner and the robber seeing each other). In this

case it is of course impossible to now for certain what will be in Σ or what Steve would want if acquiring everything in Σ, but it is in principle an empirical question. It is however possible to make a better than random educated guess even if the number of things that (from the learners perspective) might potentially be in Σ is huge. If it is someone other than Bill that is breaking in then they would take the expensive robot if they saw it, and further video would be useless. If Steve would consider this relevant in his decision of how the learner should respond to his commands, then these facts are part of Σ. If there is nothing else that Steve would consider relevant to his decision, and an informed version of Steve would think that the best thing for Steve would be that the robot stay hidden despite his commands, then this is Steve's informed preference. That the best possible action can not be found with absolute certainty is abundantly clear in this case since the set of facts about the world that might be true and might change Steve‘s mind if he knew them is very big and some of them are very complex.

This is however not different in principle from a robot that maximizes the plus button pushes and that operates in an unstructured environment where an action can result in very bad rewards due to some impossible to predict effect (for example, some actions might make a reward button pusher think that the robot actually knows what to do but refuses to do it, and that it will start cooperating if it is punished enough with the minus button). And the problem can be dealt with in the same way, by making the best guess possible given the available information. It is easy to think of scenarios where impossible to know things impacts Steve's decision in impossible to predict ways, but the problem is not fundamentally different from trying to fulfill any other success criteria in an intractable and unstructured world. The basic strategy of building the best probabilistic models possible given current ability, information and resources, continuously expanding them, continuously re-estimating what situation can be understood and always attempting to stay in situations it can handle, is still viable. It is possible that (i) the robber is Bills accomplice (ii) that Bill just walked in unmasked (and so, moving forward would result in Bill being convicted of breaking into the building) (iii) that if Steve understood some complex concepts of cognitive science regarding how his brain works and why he wants revenge on Bill, then he would conclude that he should not seek revenge after all (iv) that if Steve understood some complicated concepts regarding long term societal consequences of overcrowding in prisons, he would not want to send Bill to prison (v) that if the robot moved forward, it would crush a butterfly under its wheel, that (if the learner does not move) will distract Steve while driving his car the next day, causing an accident that would kill him (and all versions of Steve agree that this outcome would be bad). In case (v) the learner should move forward, but fully understanding the situation is completely hopeless. It is however interesting to note that the effect of the butterfly poses the exact same problem to any robot, regardless of formalism (assuming the formalism is good enough that it classifies a dead Steve as a bad thing). The enormous set of things that might influence a decision is expanded to include a new category (consisting of things like how a teacher would modify what it wants as a response to understanding complex concepts), some of which can be hypothesized and be useful in a probabilistic model, but most of which will be just as unusable

as hypotheses regarding the effects of crushing the butterfly (one can form as many such hypotheses as one likes, in favor of any decision one likes, but they can not be tested and doing this is not a useful strategy when searching for good decisions).

In a slightly different scenario where video of the learner is not routinely recorded for some reason, then Steve might never discover that it was not Bill that broke into the building unless the learner moves (Steve can record images sent to him after an alarm has been triggered, so the basics of the scenario is the same). In this case the most important thing for Steve might be that he learns that it was not Bill that broke into the building. What is best for Steve is now different from what would have been best for an informed version of Steve in the same situation since he already knows that it is not Bill that is breaking in (and then the price of the robot would dominate the decision). That is why the formal success criteria can not be to do what would have been best for an informed version of Steve in the same situation (even correctly answering the question "what action would Steve have preferred me to do if he were informed" will sometimes result in incorrect actions since uninformed versions sometimes have different needs, for example a need to know certain things that the informed version already knows). Doing what would be best for the informed version of the demonstrator if it existed does not seem to make any sense (the informed version is not present, and the informed version has for example different informational needs than the uninformed version).

If a robot sweeps dust under a rug and a teacher that is unaware of this considers its performance good, then the knowledge about the dust might be part of Σ. If the teacher considers the task to be "make the apartment clean", and would consider the learners actions bad if it knew about the dust, then it is part of Σ. But if the teacher considers the task to be "make the apartment look clean before the guests arrive", the information could be completely irrelevant, and thus not part of Σ. If the learner spent a large amount of energy cleaning the apartment, and there exists other cleaning strategies that would consume less energy, then this fact might be part of Σ. If the less energy consuming strategies had unacceptable side effects, it might not be part of Σ. If there are both unfamiliar concepts and unknown facts relating to societal effects of limited resources, then he might prioritize energy efficiency differently. Again, these possibilities are not different in principle from the possibility that a meteor will strike, causing a blackout so that the learner can not re charge, and that the learners removable batteries will actually be extremely important for some complicated reason. A robot operating in unstructured environments will face these types of hypotheses regardless of formalism, and they can be handled in a similar way. The difference is that there might be a few hypotheses that can be tested and that does advocate different actions. In this case the learner can wait until the teacher is watching to sweep the dust under the rug. It acts different from a robot maximizing the additive output of a reward button (which would wait until the teacher is not watching to sweep the dust, so as to avoid risking negative reward) because in this situation it can actually test the two competing hypotheses that (i) the teacher wants a clean apartment and (ii) the teacher wants a presentable apartment (both of which seem like something a human might want and they could both be viable given available demonstrations, feedback, etc).

In the example where the cleaning robot is sweeping dust under the rug when Steve is not looking, success is not very visible, even if the learner receives positive feedback, since it does not know if Steve has an informed preference for this type of behavior. This is basically always the case to some extent since for most possible teachers it is not possible to know for certain what their fully informed preferences would be. Observability of success is thus a matter of degree, and potential experimental setups can be evaluated based on how observable the success is expected to be. And learners can choose their actions partly based on how observable success will be (for example waiting until Steve is looking before sweeping the dust under the rug).

A messy success criteria is needed because the real world is messy and intractable, which means that all non messy success criteria are inaccurate, and thus only moves the messy bit to deciding when the success criteria is useful[5].

One strategy for dealing with an intractable problem in an uncontrolled environment is to autonomously extend the situations it can handle reasonably well and the types of teacher behavior it can interpret reasonably well, and constantly re-estimate the boundaries of what it can handle and what it can interpret. This combines the nice feature of a success criteria where a strategy that is successful in the formalism is actually successful, with the possibility of a robot that can actually do things.

Extending the situations it knows how to act in can be done concurrently with extending the types of teacher behaviors it can understand. For example, if it starts with an interpretation hypothesis Π_d that is able to learn from demonstrations reasonably well (at least in some situations), then it can extend the types of teacher behaviors it can understand by building a feedback interpretation hypothesis Π_f (after learning a task, it goes through the history of demonstrations and reproductions, and notice that what the teacher said was actually related to how good it was performing). When learning a new task it can check if Π_f is accurate in this task as well, and later use Π_f to extend the types of tasks it can learn. Another example would be a learner that, given redundant demonstrations, and redundant speech comments, can discover that demonstrations followed by a disappointed facial expression (or a certain tone of voice, or the speech utterance "Nooo!!") is more likely to be failures. Similarly it can be discovered that the speech utterance "good" is a much worse predictor of good performance than "yes"[6].

[5]In the case of a non messy success criteria without any complicated or unobservable parts, it is instead the suitability of the success criteria that is difficult to observe. It is sometimes obvious that the success criteria was bad, such as when a dust minimizing robot burns down the building and thereby clearly fails and simultaneously performs perfectly according to its non messy success criteria. But at other times the suitability might be difficult to observe. The difference is that there is no formal way to determine the suitability of a success criteria, and an agent that is optimizing a non suitable criteria does not care that it is unsuitable, and will therefore not even try to fix the situation.

[6]Perhaps because the "good" and "bad" speech utterances are sometimes mixed up by a speech to symbol transform, or because "yes" is uttered when the action is clearly successful, while "good" is uttered when this is less clear. The statistical correlation can be discovered even if the underlying reason is not clear to the learner, or even if the programmers did not consider the particular underlying mechanism

An analogy with this concurrent learning of tasks and interpretation hypotheses can be made with trying to build a model of objects at the same time as trying to understand a set of languages that is describing the object. The tasks are analogous to a set of unobservable objects, the interaction history is analogous to a set of descriptions of objects, and the interpretation hypotheses are analogous to the models the languages that the descriptions are written in. A flawed understanding of a language can be used to build a good model of an object if there is enough redundant information about it (for example a large number of separate detailed descriptions from many people, using different vocabulary and describing it at different levels of abstraction, different level of detail and from different complementary perspectives (for example its function, its shape, its component materials, its durability, how it is manufactured, etc, etc)). If enough redundant information is available to build a model that is known to be accurate with respect to some aspects of an object, it is then possible to update the model of any language describing the object. In practice, it might be convenient to concurrently update the model of the object and the model of each language. It is not necessary to directly observe the object being described, or have access to any description in a perfectly understood language. The objects can be modeled and the languages can be learned by concurrently updating interconnected hypotheses. According to the same principle, it is possible to refine an interpretation hypothesis without being able to directly observe the informed preferences of the teacher, or having any flawless interpretation hypothesis. In some sense, interpretation hypotheses are very similar to the different possible world models of an agent with a specified utility function in the ontology of those world models; if they suggest different actions it is useful to distinguish between them, and if they predict different observations, it is possible to distinguish between them (the "actions" being analogous to policy updates, and "what the world actually looks like" to "what the teacher behavior actually means").

3.3.3 What is the purpose of the simplified setups

The simplified setups are introduced so that some problems can be examined without distraction and in order to make it possible to use more beautiful and crisp math. It can also serve as a pedagogical tool since the formalism in the simplified setups are easier to explain, and if the reader understands them it will be easier to explain the formalism of the unsimplified setup.

Let's take the example where the learner only has access to noisy sensor readings of the world, and does not perfectly hear the speech comments that the teacher uses to evaluate its performance, and needs to interpret two inconsistent evaluations (different evaluations of the same action in the same world state). It is now natural to investigate for example the possibilities that: (i) the evaluation was misheard, or (ii) the world model was wrong (so that it was the same action in two different world states that was evaluated), or (iii) the action was not the same in the dimensions that actually matters (which can happen in the case of incorrect assumptions regarding what aspects of an

action is relevant), (iv) the world is viewed in the wrong framing (i.e. the world model is correct both times, but there is some relevant aspect of the world that is not captured by the model), etc, etc. In a noisy world, one of these could very well be the problem, and it makes a lot of sense to investigate all of these possibilities. But the danger is that one overlooks other types of potential problems. Let's say that the world state is observable and given in a known ontology (the world is neatly divided into world states that is shared by the learner and teacher), and that the teacher has access to a flawless policy, only cares about things represented in the world state, and finally that the teacher is giving a fully observable scalar value as feedback (instead of a noisy speech comment). What can the learner do if it observes inconsistent behavior in such a setup? It is now forced to investigate an entirely new class of possibilities, for example: (i) the teacher can not see all relevant objects, or (ii) the teacher is giving rewards for incremental progress, or (iii) the teacher is giving high rewards as encouragement since the robot has failed a lot and looks sad (real humans do this), or (iv) the teacher did not observe the entire action that it was evaluating[7], or any number of similar possibilities.

A simplified setup makes it possible to investigate these types of problems rigorously and without distractions. Inference problems can of course be intractable, and some are impossible to solve perfectly, even in principle. For the intractable inference problems, this formalism aims to provide a clear description of what it is that solutions are an approximation of. In some setups the best course of action can be impossible to find even in principle, and these are cast as an inference problem that contain a set of hypotheses such that each one: (i) has non negligible probability, (ii) imply a different optimal policy, (iii) make the same identical prediction in all observable spaces[8].

Since these problems exists in a simple setup, it seems obvious that they are much worse in more complex setups (at the very least they must be equally bad). As in most problems, the types of solutions that are appropriate in a simple world are not guaranteed to be appropriate in complex worlds. Thus the simplifications are removed gradually so that more realistic setups can be investigated, leading to modifications of both descriptions and solutions. A learner always contain a stochastic transform from an interaction history space $h \in C^h$ to a policy space $\pi \in C^\pi$ at each simplification step. In the first steps, the interaction history is over observable world states and a well separated feedback space, and later this is replaced by inputs. C^π also at first takes inputs in observable world states, but is later changed to have sensor reading type

[7]For example observing the full action sequence of a cleaning behavior in one instance, but only the end result in the other instance. The evaluations could be different if it missed that the learner swept the dust under the rug, or made a lot of noise while moving the furniture (which annoys the neighbors), or damaged the floor under the sofa, etc, etc

[8]That is, there are a set of hypothesis pairs (a teacher informed preference hypothesis and an interpretation hypothesis) that makes identical predictions regarding what feedback will be observed. If the informed preferences are modeled by a utility function, the best that can be done is to collapse them into a weighted sum, according to prior probabilities. This reduces the problem to the same type of inference problem as before. Unless the learner is using very simple models, it is unlikely that it will be able to divide its hypothesis space into groups of such sets, and collapse them into separate utility functions.

inputs. Thus a learner always contain the same type of stochastic transform : $C^h \rightarrow C^\pi$, even though the relevant spaces are given a different interpretation in later steps, as simplifications are removed.

In the first steps, the learner simply analyzes a fixed data set and outputs a policy, so that this transform is a complete specification of a learner. Any update rule that modifies a policy based on a single interaction and then forgets the information, is recursively defining a stochastic transform, so this is also a way of fully specifying a learner (even if this transform is unknown to the programmers and very difficult to find or interpret, any iterative learning rule is still identical to a unique stochastic transform $\mathrm{C}^h \rightarrow C^\pi$). In later steps, the learner needs to perform information gathering actions (meaning that an aditional element is needed to fully specify a learner).

3.4 Removing further simplifications

Step three to six removes simplifications and introduces new problems and solutions, but leaves the formalism and the notation relatively intact, even though step three leads to a new type of solutions.

3.4.1 Step three: allowing the learner to actively gather valuable data.

Let's allow the learner to choose information gathering actions, allowing it to actively gather the data that will allow the learner to distinguish between competing interpretation hypotheses. For example, if one hypothesis is that the teacher is giving rewards corresponding to performance, and another hypothesis is that it is giving rewards in response to incremental improvements (similar to how one does when training a dog for example), then repeating an action can help the learner distinguish between these hypotheses (as they make different predictions in the observable reward space). Actions can be chosen in order to understand the way feedback is generated, and/or to understand what the teacher wants the learner to do, just as a SLAM robot can take actions designed to build a map and/or find out where the SLAM robot is within that map. This can hopefully make an intractable inference problem tractable by actively gathering the information that will allow it to understand the world well enough to make reasonably accurate simplifications. The teachers utility function $u^\star$ is still defined in the same way, and the success criteria is still judged only based on $u^\star$ (choosing actions so that the learner can best estimate the teacher signal generating transform Ω is however probably a good strategy since $u^\star$ is easier to find with a better Ω estimate).

We denote an interaction protocol as the stochastic transform $\wp : C^h \times C^s \rightarrow C^\alpha$. A $\wp$ is a strategy for generating an action based on the interaction history and the current world state. A protocol can for example be defined by a rule for how to modify some

data structure (for example a policy) at each interaction, and then select the next type of interaction based only on the current state of this data structure. The data structure update rule, and the rule for selecting interactions based on its current state together implies a unique $\wp$.

In order to not complicate things, we keep the same success criteria, meaning that the actions only serve to gather information that can be used to build a policy. This sidesteps for the moment the issue of making tradeoffs between learning what should be done, and actually doing what should be done (the learner simply tries to act in the way that will lead to the best possible policy, not needing to worry about how good it is performing tasks during the learning phase). We do not make any strong assumptions regarding the interaction behavior of the teacher, meaning that it could stop giving feedback at unknown times, possibly dependent on how the learner act (it could for example stop interacting or start interacting in a less engaged way because the learner is "not learning", or "done learning", or "boringly repeating the same actions", etc).

Since the problem of building a policy based on a given data set was treated in simplification step two, the only thing left to deal with from a theoretical point of view is "how to select actions with the highest expected usefulness of information". This is a relatively easy step from the point of view of a formalism, but leads us to an active, but basically open, research field when we look for tractable solutions. From a theoretical point of view, we need to find the action that will result in the highest expected amount of useful information. The teacher signal generating transform Ω includes all feedback behavior, so there is no need to introduce any additional transform for the "stop interacting in certain situations" or "start giving less informed feedback if bored" situations. The usefulness of an action is thus dependent on how useful feedback it will generate immediately as well as how it will impact the future interaction behavior of the teacher. The problem of determining the expected amount of immediate information from an action is related to the field of optimal experiment design (tractability issues are different, but the theoretical framework is the same).

The expected usefulness of a single action is simply the weighted sum of the expected usefulness of the action according to all $u^\star$ - Ω pairs (weighted by probability), or the corresponding integrals in the case of continuous parameter spaces. For the hypothesized stochastic Ω^j, and a hypothesized $u^{\star i}$, the usefulness of a single action is the weighted sum of the usefulness of the change in policy from the possible feedback responses. In the continuous case, the sums turns into integrals. Determining the expected usefulness of even a single discrete action, even given absolute knowledge of the world, is thus a completely intractable triple integral. We therefore need to start looking at approximate solutions. For example, one could try to optimize the expected information gain regarding what $u^\star$ looks like. This is an approximation as discriminating between some possible $u^\star$s can be completely useless, even when they are very different (they could result in identical policies, or policies that have identical utility according to $u^\star$). It would also be possible to optimize the information gained about Ω, under the very reasonably sounding approximation that learning to understand the teachers feedback will allow the

learner to do what the teacher wants it to do. It is also possible to make some even more radical approximations and just maximize surprise, but then a TV showing static noise (or any other situation providing completely unpredictable sensory information) would become an attention trap. Luckily there is an entire field of research that is actively exploring what types of approximations can be made, and when some of them leads to traps of the type mentioned above. See [135] for early work and for example [134] and [133] for more recent experiments. In the previously mentioned [40], these methods are used for determining when and who to imitate. See also [136] for the optimal experiment design framework from which some of the basic principles comes from. One common setting is building a forwards model for robot control by selecting exploratory actions to be as informative as possible, but the findings can be used without much modification. Below we can see algorithm 3 that builds on the multiple generator algorithm discussed previously. It is built on the extremely old and very basic idea that if an hypothesis is formed based on a history, and then found to be good at predicting newly observed data, it is probably good. Actions are selected so as to discriminate between competing hypotheses. Hypotheses are discarded or modified based on the new observations, and new hypotheses are created based on history. Just like in the previous multiple generator algorithm, it concurrently estimates $u^\star$, Γs, what points where generated by what Γ, and what the generating regions are.

3.4.2 Step four: dealing with a world that is not perfectly visible to the teacher

Let's remove the perfect visibility of the teacher, and allow some world dynamics. $u^\star$ is now a mapping with the inputs expanded to include the teacher's world model. The teacher can now care about both the real world, and its own world model. The teacher could for example want dust to be removed from an apartment, and/or want to believe that the dust has been removed (a teacher could dislike a dusty apartment, and/or dislike that the apartment looks dusty, leaving the learner to deal with the old "to sweep dust under the rug, or to not sweep dust under the rug" question discussed earlier). The output of $u^\star$ is no longer directly visible to the teacher since the actual world state is not directly visible (and must instead be modeled based on inputs). Success is still defined in exactly the same way however. This changes little for the learner from a theoretical point of view as it never had access to the outputs of $u^\star$ anyway. From a practical point of view it changes everything. Solution strategies such as "wait to sweep the dust under the rug until the teacher is looking in order to improve usefulness of the feedback" becomes central.

What the teacher sees, and what types of worlds/actions are easy for it to see/understand will now have to be monitored, and actions will also have to be chosen so that future world states are informative. The learner also has access to sensor readings of the teacher, that might be informative regarding what is visible to it (for example a camera image of the teacher, from which it can be estimated what it is looking at, and which

Algorithm 3 Active multiple generator algorithm

Input: $\mathcal{D}_0^G$, $\mathcal{D}_0^\Gamma$, $D_0^{u^\star}$, T,$h(T)$,S

• $\mathcal{D}_0^G = \{D_0^{G1}, D_0^{G2}, ..., D_0^{GN}\}$ *is the initial estimate of the generating tendencies (at time 0).*

• $\mathcal{D}_0^\Gamma = \{D_0^{\Gamma 1}, D_0^{\Gamma 2}, ..., D_0^{\Gamma N}\}$ *is the initial estimate of the feedback behaviors.*

• D_0^Ω *is the initial estimate of* $u^\star$

• S *is the number of update steps done as a response to each interaction*

while teacher still giving feedback **do**

  $\alpha_t \leftarrow determineAction(\mathcal{P}_t, D_t^{u^\star}, \mathcal{D}_t^\Gamma, \mathcal{D}_t^G)$

  $I_t \leftarrow \{\Xi, observeDemAction(\alpha_t)\}$

  $\mathcal{D}_t^\Gamma discard\Gamma Hyptheses(I_t, h(t-1), \mathcal{P}_t, D_t^{u^\star}, \mathcal{D}_t^\Gamma, \mathcal{D}_t^G)$

  $h(t) \leftarrow \{I_1, I_2, ..., I_t\}$

  for $s = 1$ **to** S **do**

    ${}^{s+1}\mathcal{P}_t \leftarrow estimateGen({}^s\mathcal{P}_t, {}^s D_t^{u^\star}, h(t), {}^s \mathcal{D}_t^\Gamma, {}^s \mathcal{D}_t^G)$

    ${}^{s+1}D_t^{u^\star} \leftarrow update - u^\star - estimates({}^s D_t^{u^\star}, {}^{s+1} \mathcal{P}^t, {}^s \mathcal{D}_\Gamma^t)$

    ${}^{s+1}\mathcal{D}_t^\Gamma \leftarrow update - \Gamma - estimates({}^s\mathcal{D}_\Gamma^t, {}^{s+1} \mathcal{P}_t, {}^{s+1} D_t^{u^\star})$

    ${}^{s+1}\mathcal{D}_t^G \leftarrow estGenTend({}^s\mathcal{D}_t^G, {}^{s+1} \mathcal{P}_t, h(t), {}^{s+1} D_t^{u^\star})$

  end for

  $\mathcal{P}_{t+1} \leftarrow^S \mathcal{P}_t$

  $D_{t+1}^{u^\star} \leftarrow^S D_t^{u^\star}$

  $\mathcal{D}_{t+1}^\Gamma \leftarrow^S \mathcal{D}_t^\Gamma$

  $\mathcal{D}_{t+1}^G \leftarrow^S \mathcal{D}_t^G$

  $t \leftarrow t + 1$

end while

objects are in its line of sight). Ω now maps the teachers world model to feedback. Given a data set, the learner can in principle build a composite transform consisting of one transform from the actual world to the world model of the teacher, and then simply use that world model instead of the world state as input to Ω. The teacher still shares a given ontology with the learner in this step, meaning that its world model is a point in the same space as the actual world state (but of course not necessarily the same point). Again, it is easy to extend the formalism to include the analysis of a given data set by simply giving the full transform including a learner created transform from world state to teacher world model and Ω, and give this full transform the same place in the equations as Ω had earlier. Let's introduce some notation:

- **teacher sensor readings:** $z \in Z$. For example a camera whose images of the teacher is observable to the learner, which can be used for example to indicate which objects are visible to the teacher.

- **teacher world model:** $w \in \mathcal{W}$: The best guess of the teacher concerning the world state and the learner action.

- **Estimated teacher world building apparatus:** $\mathcal{V} : Z \times S \rightarrow \mathcal{W}$. Since the

teachers feedback behavior is now based on its world model (not the actual world state), it would be useful for the learner to estimate this transform.

- **Changed inputs for Ω:** $\Omega : \mathcal{W} \times C^h \times C^u \rightarrow C^f$. The learner must now estimate the teachers world model in order to interpret the feedback generated from Ω.
- **Changed inputs for the utility function:** $u^\star : \mathcal{W} \times C^s \times C^\alpha \rightarrow \Re$: Preferences can now be defined in both actual world states and estimated states (the teacher can want the apartment to be clean and/or want to avoid the sight of dirt).

Practical difficulty is increased, but it is still a well defined inference problem with an analytical optimal solution (in terms of expected utility) for any finite data set and any generating distributions (if the teachers world building apparatus is also drawn from a known distribution). The problem was intractable even before, so not much has actually changed in terms of needing approximate solutions to evaluating data. But new types of information gathering strategies might be needed.

An obvious solution strategy would be to create the types of situations that is visible to the teacher in order to get more informative feedback. For example: "make sure to sweep the dust under the rug while it is looking, so that feedback will be more informative". The analogy for a map building room would be to find that estimates based on camera images is less reliable in dark rooms so that it can take light conditions into account when updating on data, and of course turn the light on whenever possible.

3.4.3 Step five: dealing with a world that is not perfectly visible to the learner

Let's remove the perfect visibility of the learner, so that sensor reading and internal states take the place of world states. The interpretation hypotheses become transforms from teacher preferences to sensor readings, or states that are obtained by transforming sensor readings. The inputs to interpretation hypotheses and policies are now either input spaces or the results of transforms from input spaces. Very little actually changes from a theoretical point of view. The world states were not visible to the teacher in the previous step, so this space already functioned pretty much like sensor readings that are used to estimate what the teacher was perceiving.

3.4.4 Step six: finding Ω without a known generating distribution

Let's remove the aspect of the learner knowing how likely each possible way to generate the teacher signal is. In other words, the teacher signal generating transform Ω is no

longer drawn from a known distribution. If the learner has a set of hypotheses regarding the distribution from which the teachers Ω transform is drawn, this collapses (from the learners point of view) into an equivalent problem. If the learner is able to investigate interaction histories from multiple teachers, it could in principle revise its estimate of each individual teachers Ω transform concurrently with estimating the distribution from which Ω is drawn.

As the interpretation hypotheses are already defined over sensor readings, and outputs policy changes, nothing changes from the point of view of the learner. The initial set of hypotheses will imply a prior distribution, but finding it would be completely intractable (and of no practical value to the learner). The success criteria has not changed at all. We are now almost at the situation of the previous section, with the only remaining difference being the existence of the teachers utility function $u^\star$. $u^\star$ is however not visible to anyone at this point, and it does not actually influence anything, so removing it would not change anything dramatically from the point of view of the learner (the task of interacting with a teacher in step six, is indistinguishable from interacting with an actual human in an actual unstructured environment from the point of view of the learner).

3.5 Step seven: viewing existing learning algorithms, operating in the unsimplified setup, as testable interpretation hypotheses

We finally remove the assumption of a $u^\star$ in the head of the teacher. As pointed out above, this will change nothing from the point of view of the learner, as $u^\star$ was not actually affecting anything. An interpretation hypothesis Π now maps sensor readings, or states in more abstract spaces ultimately obtained from sensor readings, to policy updates. This means that a Π is now identical to a learning algorithm, and existing learning algorithms can be seen as interpretation hypotheses. Existing learning algorithms will often only be an hypothesis of how some limited set of the input space should be interpreted, making a set of learning algorithms with different types of inputs very suitable for concurrent modifications. By viewing a learning algorithm in this way, we see when and how one learning algorithm can be used to modify another learning algorithm. This section will focus on how to denote this type of concurrent modification of existing learning algorithms as concurrent updates of a set of hypotheses. The focus will be on a graphical representation, and on sketching numerous examples.

A learning algorithm is now re interpreted as encoding assumptions about some information source, such as "demonstrated actions are more likely than random to be good actions", or "a reward button is pushed by a teacher, that very accurately compares the end result of an action with the 7 previous actions". If these assumptions are made explicit as well as modifiable and/or falsifiable, a more suitable name is interpretation

hypothesis. It is for example possible to turn the number of previous actions that the current action is compared to into a variable, that can then be updated based on observations (evaluating the current action compared to the previous 3 actions will lead to different expected interaction histories than if it is compared to the 7 previous actions, allowing us to update the respective probabilities/update the parameter). Perhaps the most basic application would be to take two learning algorithms using the same set of inputs. Then make their assumptions explicit so that it is possible to calculate what types of interaction histories they predict, and finally use observations to determine which one to use. Let's first look closer at a two step algorithm that first estimates the teachers goal by looking at how highly the teacher evaluates its actions, and then learns how to achieve this goal. The hidden assumption is that the teacher evaluates the performance of the learner. Then we can take another learning algorithm that changes the policy to be closer to policies that get a higher evaluation. The hidden assumption being that the teacher evaluates how close the policy is to good policies. The learner uses one of these (or some other method, like interpreting demonstrations) to learn a simple task. It can now go through its history and see how the various types of actions were evaluated, and choose one interpretation hypothesis over the other (for example checking how actions that are close to good action in policy space, but very bad in outcome space, are evaluated).

Another example would be when there is a probability estimate of how often demonstrated actions are better than random, and how far they are from optimal. With a limited data set, improving these estimates will likely improve the policy updates. One idea entailed by this way of viewing learning algorithms is to use multiple interpretation hypotheses, each interpreting a different type of input, and each with a set of parameters (such as history length of a reward button), to learn a task concurrently with changing the parameters of the hypotheses, and estimating their usefulness. If the teacher pushing the reward button does not compare the current action to history, but instead to the optimal action, then it should be possible to discard the interpretation hypothesis as no parameter value will result in an accurate model. The ability to assess the usefulness of an interpretation hypotheses becomes important if there are competing hypotheses on how to interpret the same type of information, or if there are other information sources that can be used instead. The ability to discard some sources of information means that a learner can in theory learn how to deal with a real human in an unstructured environment that provides a diverse and redundant amount of information, and where some of the information is much harder to interpret than others, and where it is not known at programing time what information sources will be most useful (which could be heavily dependent on the type of teacher and the type of task). If a teacher is encountered that often uses a reward button to encourage a learner that looks sad after failing, and the learner is unable to differentiate the two different types of reward button uses, then it might be best to simply learn from other information sources when dealing with this teacher. Similarly, if demonstrations often fail, and the learner is unable to separate a failure from a success (for example by using facial expressions), then it might be best to avoid requesting demonstrations, or to avoid wasting computational resources

trying to learn something useful from them. The next step is to try to better describe what exactly these hypotheses are models of. They are supposed to model "what the information means", but this has to be stated more exactly (see section 3.3.2 where the concept of informed preferences are introduced).

Let's look closer at one algorithm that translates demonstrations into policy changes, and a second algorithm that translates a scalar value following a learner action into policy changes. To make use of these information sources it is necessary to make some sort of assumptions about them, at the very least it is necessary to assume something along the lines of: "demonstrations are more likely than random actions to be good", or that: "good learner actions are more likely to be followed by high scalar values than bad actions". If a learner knows one of these facts, it is possible to infer the other one from observations. If the learner correctly assumes that demonstrations are good actions, it can learn what to do in some restricted circumstance and then notice that good actions are more likely to be followed by high scalar values. And if it correctly assumes that high scalar values indicate good actions, it can learn what to do in some situations and then notice that demonstrated actions are more likely to be examples of good actions. When the correlation has been noticed, it can be used when learning how to act in new situations (learning how to learn by learning how to interpret the various types of feedback given by the teacher). Active learning in this setting means seeking situations and performing interactions that will result in the type of information that will allow it to disambiguate between different hypotheses of how to interpret teacher behavior.

Knowing either the policy or the parameters of one interpretation hypothesis allows us to find the other two (the policy allows us to find the parameters of both interpretation hypotheses, and either interpretation hypothesis allows learning of the policy). The idea is that learning the interpretation hypotheses will be good for learning other tasks, but updating all three things concurrently can be useful even when only learning a single task.

The setup now consists of a learner that can be represented by transforms and input/output spaces, and an unstructured environment containing a human teacher. We have removed the simplification that there exists a utility function somewhere in the head of the teacher. The mathematical notation describing the teacher now only exists as a model that exists fully inside the learner architecture. The setup of step seven thus contain a real world human and an unstructured world, but all the notation is now describing a computational system (the actual physical embodiment of the learner is outside this computational system, even if obviously its model of its own embodiment is part of the system). From the formalism point of view we will focus on this learner architecture and we choose a graphical representation of it in order to get a better overview. Using this representation, we will describe several different concrete architectures, learning from various information sources.

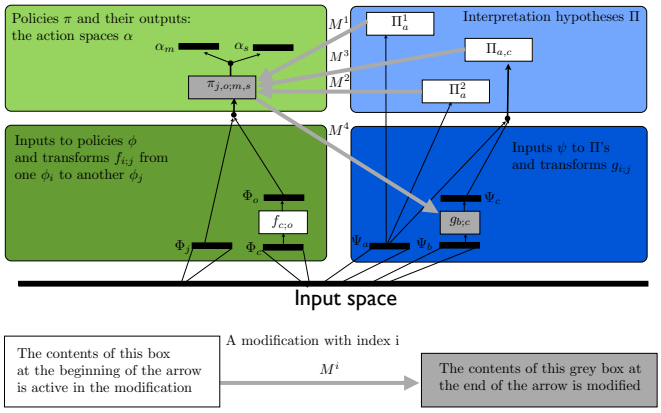

Fig. 3.1: In this setup, a policy $\pi_{j,o;m,s}$ is modified by three interpretation hypotheses; Π_a^1, Π_a^2 and $\Pi_{a,e}$. Inputs to policies are denoted Φ and inputs to interpretation hypotheses are denoted Ψ. The black arrows mark inputs and the grey arrows mark modifications. The policy $\pi_{j,o;m,s}$ is used to modify the transform $g_{b;c}$. If the policy $\pi_{j,o;m,s}$ can be reasonably well learnt using only information in Ψ_a (for example a demonstration of a task), then it can later be used to check if the state in another space Ψ_c contains any useful information (for example, if one state is more frequent in the case where the demonstration was a failure, then this can be detected using $\pi_{j,o;m,s}$). The informedness of states in a space Ψ_c can be judged using $\pi_{j,o;m,s}$ (for example how useful states in Ψ_c is for separating failed demonstrations from successful ones) . It is now possible to use $\pi_{j,o;m,s}$ to choose from two different parameter setting of $g_{b;c}$ (two different parameter sets results in two different Ψ_c spaces, which $\pi_{j,o;m,s}$ can be used to choose between).

3.5.1 Graphical representation

We introduce a way of depicting a system/learner architecture graphically in figure 3.1. Hopefully this representation will make it easier to see new extensions to existing research as well as enable us to describe proposed setups more clearly and more quickly. The top left rectangle contains the policies and the lower left contains the steps that lead to the inputs of the policy (which can be described as feature selection, finding the task space, finding the framing, etc). The top right rectangle is the interpretation hypotheses, and the lower right rectangle contains the transforms that generate those inputs. The black arrows depicts inputs or outputs and the grey arrows depict modifications.

Inputs can for example be: current sensor readings (internal sensors like battery or external sensors like a camera), past sensor readings, predicted future sensor readings, internal states (for example an estimated urgency of the current task), the estimated position of an object at some previous time (where object position is calculated, not present in sensor readings), the output of some opaque pre-processing step that the learner has no access to, the estimated current common ground in a conversation[9], etc, etc.

Modifications of the M^4 kind are uses a task in order to find a new input space for an interpretation hypothesis. For example learning which teacher facial expressions correspond to failed demonstrations (by using a known policy and the recorded history of demonstrations and facial expressions). The reasonably well learnt policy $\pi_{j,o;m,s}$ can be used to determine how good individual demonstrations were. When we have a set of demonstrations with estimated quality, we can search for a way to predict this quality. This enables us to evaluate a space Ψ_c in terms of how well states in Ψ_c enables us to predict the quality of a new demonstration. The ability to evaluate a possible space Ψ_c enables us to modify the transform $g_{b;c}$ that results in Ψ_c (we can choose between two different parameter values of $g_{b;c}$ since we can choose between the two different resulting spaces Ψ_c). In short: $\pi_{j,o;m,s}$ modifies $g_{b;c}$ (which is denoted by a grey arrow from $\pi_{j,o;m,s}$ to $g_{b;c}$, and given the M^4 identifier for easy reference). From a technical point of view this type of modification are not different in principle from the other types of modifications, but the result is that the learner can be said to "learn how to learn".

The states in Ψ_c could for example correspond to facial expressions of the teacher where some facial expressions indicates failure and other facial expressions indicate success. $\pi_{j,o;m,s}$ can help determine if states in Ψ_c is informative and so the "facial expression classifier" $g_{b;c}$ can be modified. If the states in Ψ_c are informative in other situations, this could speed up learning in many other tasks (the teacher might make similar types of facial expressions no matter what task the learner is failing/succeeding at).

[9] What is appropriate to say and do is often dependent on the common ground, since an interlocutor will interpret actions based on this. A learner that can change how the current common ground the current common ground is updated will be denoted by a modifiable policy with an appropriate action space (consisting of manipulations to its model of the current common ground).

A learning from demonstration setup

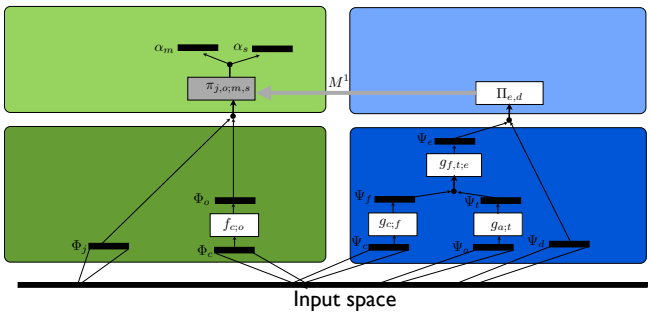

Fig. 3.2: A learning from demonstration setup where $\Pi_{e,d}$ modifies the policy $\pi_{j,o;m,s}$. The inputs to $\pi_{j,o;m,s}$ is in joint space Φ_j and estimated object position space Φ_o, and the policy is able to set the states in the action spaces α_m (motor outputs) and α_s (speech outputs). The policy is being modified by $\Pi_{e,d}$ based on an estimated teacher evaluation of its own demonstration Ψ_e and a representation of the demonstration in Ψ_d. The evaluation estimate Ψ_e is obtained by $g_{f,t;e}$ based on facial expression Ψ_f (obtained by $g_{c;f}$ from camera input Ψ_c) and tone of voice (obtained by $g_{a;t}$ from audio input Ψ_a).

In this setup the learner learns from demonstrations, as well as an estimate of how happy the teacher was with the demonstration it just performed (the teacher is not always successful at performing the task and the learner is trying to predict if a new demonstration was a failure, and use that during learning). A graphical overview can be seen in figure 3.2. The input to $\Pi_{e,d}$ is teacher actions represented in Ψ_d (a set of low dimensional context-action pairs) and an evaluation represented in Ψ_e, obtained by a transform $g_{f,t;e}$ with inputs in facial expression space Ψ_f and tone of voice space Ψ_t. Ψ_f is obtained from a camera input Ψ_c using $g_{c;f}$ and Ψ_t is obtained from an audio input Ψ_a using $g_{a;t}$. Ψ_d is given to the learner directly and the learner can not modify how it is obtained (it is not a sensor reading but, since it can not be modified, it is an input to the learner). $\Pi_{e,d}$ updates a policy $\pi_{j,o;m,s}$ with inputs in joint and estimated object position space, and performing actions in speech and motor α spaces.

$\Pi_{e,d}$ is an exact implementation of an interpretation hypothesis that could be verbally approximated as; "actions in Ψ_d are probably good for certain states in Ψ_e and probably bad for other states in Ψ_e", or even more crudely; "imitate the actions that the teacher seems pleased with". The update is denoted M^1 and is dependent on the details of the hypothesis, for example the assumed noise level of a favorable evaluation of some specific type of demonstration (for learning from demonstration algorithms that assume normally distributed noise, see the GMR based algorithms of [108, 101, 125, 69]). If the update mechanism is static and ad hoc, only implicitly encoding assumptions about noise levels, it is still referred to as an hypothesis (it is just an hypothesis whose details are not easy to see and that is not updated based on observations). If the details of this hypothesis is made explicit there are several ways in which it could be updated: (i)

demonstrations that involve heavy objects can be given a higher expected noise rate[10]. (ii) Adding a word recognizer that detects only the word "Nooo!" (giving a binary input to $\Pi_{e,d}$) and uses this instead of the state in Ψ_e when it is present but ignores it when not present (the noise is of course dependent on the word recognizer and the usefulness is dependent on how often the word is used after failed demonstrations). (iii) A "triadic joint attention[11] detector" could be added based on the finding that the noise level is much lower when this is happening. The learner does not have to understand why the noise is lower in some states of the "triadic joint attention detector". The correlation could be detected for example if: the teacher is putting some real effort into trying to do a good demonstration, or if the facial expression estimator $g_{c;f}$ works better when it has this type of input, or if the type of verbalizations made in this type of interaction is easier to interpret by $g_{a;t}$, or if the types of tasks that are demonstrated with triadic joint attention is easier to learn, or if the types of behavior the teacher does in this type of interaction is the types of behavior that the teacher would like the learner to adopt, etc. The learner can benefit from this "triadic joint attention detector" without fully understanding why it works.

A feedback learning setup

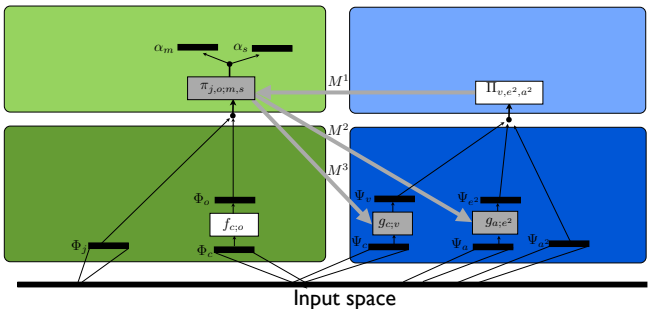

Fig. 3.3: This figure shows a learning from feedback setup where Π_{v,e^2,a^2} estimates how highly its actions (represented in Ψ_a) were valued (estimated in Ψ_{e^2}) and how informed the teacher was Ψ_v, and uses this to choose whether or not to add the action (along with the context it was performed in) to a list of such actions contained in the policy $\pi_{j,o;m,s}$.

We can see this setup in figure 3.3, where the policy $\pi_{j,o;m,s}$ has the same inputs and outputs as before. $\pi_{j,o;m,s}$ contains of a list of previously performed actions (context and

[10]If the teacher is not very proficient at manipulating heavy objects, and furthermore states in Ψ_e mainly captures how pleased it is with its own performance relative to the difficulty level of a task, then it is perfectly possible that the same state in Ψ_e indicate a higher expected noise level in the case of a type of Ψ_d that involves heavy objects. This can be utilized by a learner that simply notices that certain types of states in Ψ_d correspond to higher noise levels (even given the state in Ψ_e).

[11]This is when the teacher is looking at the learner, as well as an object, and is following the learners gaze to make sure they are both attending to the same object.

output) where each action has two scores, one estimated evaluation and one estimate of how informed the teacher was at the time of evaluation. It also consists of a rule for how to select a subset from the list based on the current context, and finally a regression algorithm that gives an output based on the subset of actions selected. If there are no previous actions with contexts close to the current context, a fixed default $\pi^2_{j,o;m,s}$ is executed (not shown in the figure).

$\pi_{j,o;m,s}$ is modified by Π_{v,e^2,a^2} based on the estimated visibility of the teacher Ψ_v (how much of the scene does the teacher see when it gives the feedback), the estimated evaluation Ψ_{e^2} and the action performed Ψ_{a^2} (the action that is assumed to be evaluated, and the parts of the context that is assumed to be relevant for what action is to be performed). There are three input spaces to Π_{v,e^2,a^2} that could be improved. Ψ_{a^2} is the representation of actions and contexts, so any improvement of this would center around re-estimating which part of the context was is relevant, or what part of the action is relevant. In this example the focus is on re-estimating what evaluation the teacher wanted to give, represented in Ψ_{e^2}, by modifying the transform $g_{a;e^2}$. And also on modifying the transform $g_{c;v}$ calculating how visible the scene was to the teacher at the time of the evaluation (represented in Ψ_v). For the modification M^1 to work, the initial way of calculating Ψ_{e^2} must be at least approximately accurate. Even if the data is noisy, it can still lead to an accurate estimate of $\pi_{j,o;m,s}$ (using a larger data set than would have been needed if noise free data was available). A somewhat accurate $\pi_{j,o;m,s}$ can then be used to create a data set that can be used to modify $g_{a;e^2}$ (denoted M^2) consisting of how correct the action was according to $\pi_{j,o;m,s}$ and the input in Ψ_a (the noise level will be dependent on the accuracy of $\pi_{j,o;m,s}$). One way of modifying $g_{a;e^2}$ would be to find a transform that, besides mapping audio input to the known word categories, there is an alternate transform $g_{a;e^2}$ that also maps some audio input to a category that correlate strongly with very large performance improvement (for example corresponding to the teacher loudly saying "Great!").

The modification M^3 is done in a similar way. What is sought after is a space whose states can be used to predict if the evaluations in Ψ_{e^2} is accurate. $\pi_{j,o;m,s}$ tells us the accuracy of individual evaluations, which gives us a data set consisting of pairs of (inputs in Ψ_c and this accuracy) and what we want is a transform $g_{c;v}$ such that states in Ψ_v predict this accuracy. This problem is of a very well explored format (obtaining a function based on input-output pairs) and its solvability is strongly influenced by the accuracy of $\pi_{j,o;m,s}$ due to its influence on the noise level in the data set. One example where this can succeed is if 90% evaluations are correct, and in the other 10%, the teacher is unable to see an object. An accurate $\pi_{j,o;m,s}$ can be learnt (M^1) with data of this noise level, leading to a data set where the accuracy of the individual evaluations are accurately estimated. Now this data set (that does not need to be noisy since $\pi_{j,o;m,s}$ is accurate) can be used to find a $g_{c;v}$ that maximally well separates the 90% accurate evaluations from the 10% inaccurate ones.

The interesting part of this setup is to improve $g_{c;v}$ and $g_{a;e^2}$, which can hopefully be used to learn similar tasks.

Chapter 4

Experiments with concurrent updating of a task model and an interpretation hypothesis

Contents

4.1 Inverse Reinforcement Learning with Ambiguous Feedback **65**

4.1.1 Bayesian Inverse Reinforcement Learning 65

4.1.2 Feedback Model 66

4.1.3 Sign-Meaning Model 68

4.1.4 Algorithm 68

4.1.5 Active Sampling 71

4.2 Results **71**

4.2.1 Navigation Task 71

4.2.2 Collecting Objects 74

4.3 Conclusions **74**

This chapter is based on [88]. The text is adapted for the thesis, but the text and the figures are mostly done by Manuel Lopes. The programing of the experiments were also performed by Manuel Lopes. The thesis author is the second author of [88] and contributed to the design of the experimental setups and algorithms. A system to learn task representations from ambiguous feedback. We consider an inverse reinforcement learner that receives feedback from a teacher with an unknown and noisy protocol. The system needs to estimate simultaneously what the task is (i.e. how to find a compact representation of the task goal), and how the teacher is providing the feedback. This is a good example of a learner that modifies an interpretation hypothesis based on observations. A teacher uses words such as “good”, “bad” and “go left” and at the start, the learner has a partial understanding of what this means, an imperfect interpretation hypothesis. As the learner learns the task, it is able to update this interpretation hypothesis. If for example has learnt that a particular action it performed was good, it can update its hypotheses regarding what the word uttered means. The learner actively

chooses actions and learn the task concurrently with re-estimating the interpretation hypothesis.

The system to starts with a set of known signs and learn the meaning of new ones. We present computational results that show that it is possible to learn the task under a noisy and ambiguous feedback. The active learning is shown to reduce the length of the training period.

Several studies discuss the different behaviors naive teachers use when instructing robots [89, 141]. An important aspect is that, many times, the feedback is ambiguous and deviates from the mathematical interpretation of a reward or a sample from a policy. For instance, in the work of [89] the teachers frequently gave a reward to exploratory actions even if the signal was used as a standard reward. Also, in some problems we can define an optimal teaching sequence but humans do not behave according to those strategies [141]. The system in [83] automatically learns different interaction protocols for navigation tasks where the robot learns the actions it should make and which gestures correspond to those actions.

In this work we consider a setting where the robot must learn a task description (in the form of a reward function) from interacting with a teacher that provides feedback signals. We extend previous approaches by learning simultaneously how the feedback is being provided and what is the meaning of the teacher's feedback signs. Note that we will call what the teacher says/writes *sign or feedback sign* and the meaning of the sign *feedback*. In a human-robot interaction setting we consider the case where the robot tries an action and then receives a feedback signal from the teacher. Such feedback is not restricted to a pre-defined protocol, with a pre-defined set of signs or words, but should allow for new interaction types and instruction commands. The teachers will also provide signals not expected by the robot. A simple case is when the teacher gives synonyms of feedback words.

Our contributions are: a) a learning by demonstration system that learns a task description based on noisy feedback, b) an interactive learning system with a loosely defined protocol in terms of accepted words and their use, and c) an online learning system that estimates simultaneously the task, the feedback protocol and the sign-meaning relations. We assume that the robot is initially equipped with a set of sensory-motor skills and knowledge of some feedback signs. The state space is assumed to be continuous, the set of actions and feedback meanings are finite and the feedback signs can grow infinitely.

The experimental protocol we used is the following. The robot samples a state and tries an action on that state. The teacher has the possibility of providing the robot with a feedback signal. Those signal can refer to the name of the correct action to be used or by explicitly saying if an action is correct or wrong. Our framework is generic and the signal provided by the teacher can refer to the uttered words, gestures, facial expression or even the prosody of speech. By iteratively following this process, the system will learn the task representation. This system is different from typical learning by demonstration systems because the data is acquired in an interactive, and online, setting and not in

batch. It is different from previous learning by interaction systems in that the feedback signals received have a much looser protocol and might make use of unknown signs.

In the next Section we provide the details of the algorithm, including a summary of Bayesian inverse reinforcement learning and an active learning extension. Finally we present simulations of our system and conclusions.

4.1 Inverse Reinforcement Learning with Ambiguous Feedback

In this section we present our learning algorithm. Our problem can be divided in three smaller ones: a) learn the task representation; b) learn how the teacher provides the feedback on the executed actions; and c) learn the meaning of novel feedback signals. We remember that the *feedback* is what the teacher means and the *sign or feedback sign* is what it "says/writes/gestures".

4.1.1 Bayesian Inverse Reinforcement Learning

We consider a standard *markov decision process* (MDP) and follow the notation of [140]. An MDP is defined by a state and action space X and A respectively, a reward function R and a state transition model P. A policy, $\pi(x, a)$, is a function that attributes a probability of selecting an action in each state and the function $r(x, a)$ gives the reward the agent receives when choosing the action a in state x. The goal of reinforcement learning is to find the optimal policy π^*, that is defined as the ones that maximizes the total discounted reward, i.e. $R = \sum_{t=0}^{\infty} \gamma^t r_t$, with γ a discount factor and r_t the reward received at time t. We define the Q^π-function as the value of taking an action at a given state when following policy π, i.e. $Q^\pi(x, a) = E_\pi \left(r(x, a) + \gamma \sum_y P^a_{xy} \max_b Q^\pi(y, b) \right)$, where $P^a_{xy} = p(x_{t+1} = y | x_t = x, a_t = a)$ is the probability of reaching state y when the current state is x and the chosen action is a.

In our case we are not interested in learning a task by self-exploration but will use data from a teacher to learn the representation of the task the teacher wants the learner to acquire. In this situation we do not have a reward function from which we can get samples but have instead samples from the policy, i.e. we do not have a reward but have actions. This formalism is called the *inverse reinforcement learning* (IRL) problem [142]. The goal is to find the reward function that the teacher is trying to maximize and later on use it to select the best actions.

Using a Bayesian perspective, we follow the *Bayesian IRL* approach (BIRL)[143]. In that setting we consider that, if the teacher is performing the task described by the

reward function r, the samples of the demonstration are generated by:

$$p(x, a|r) = \frac{e^{\eta Q(x,a)}}{\sum_b e^{\eta Q(x,b)}}$$

where η is a confidence parameter where high values correspond to the optimal policy and lower values allow samples of non-optimal actions. We assume a uniform state sampling. For numerical purposes it is convenient to rewrite that expression by considering the summed probability of all the optimal actions (A^*) as:

$$p(x, a|r) = \begin{cases} \frac{\sum_{a \in A^*} e^{\eta Q(x,a)}}{\sum_b e^{\eta Q(x,b)}} & if\ a \in A^* \\ \frac{e^{\eta Q(x,a)}}{\sum_b e^{\eta Q(x,b)}} & if\ a \notin A^* \end{cases}$$

To have a normalized probability distribution we have to consider all optimal actions as a single one. To learn the reward we compute the posterior distribution of the reward function after observing a given data vector $D_t = \{A_{0:t}, X_{0:t}\}$:

$$p(R_{t+1}|A_{0:t}, X_{0:t}) \quad \propto \quad p(A_t|R_t, X_t)p(R_t) \tag{4.1}$$

for a suitable choice of prior distribution on R, see [143]. The process of computing this posterior distribution is computationally intensive. We implement it with a filtering perspective [144]. We consider that the reward function is a linear combination of basis functions $\phi(x)$ in the following way $R = w^t\phi(x)$. Then, we estimate not the posterior of the parameter w of the mixture, but the posterior of the activation of each feature vector. An intuitive way to see this is to assume that each sample point is generated from a policy corresponding to a single feature vector. Under this perspective the mean of the feature distribution is the best estimation for the reward function.

4.1.2 Feedback Model

Now, the learner must infer what the task representation is and how the feedback is being provided. In this section we consider that the signs provided by the teacher have a known relation with the feedback meaning, next subsection will relax this assumption. The difference compared to the standard setting is that the demonstration is not given as a sequence of state-action pairs but as feedback on those pairs. For a given state action pair (x, a) we consider the probability of receiving a given feedback signal f.

If the robot performs the correct action, the teacher might say nothing, might verbalize the correct action to reinforce it or acknowledge that it was the correct action. If the learner performs the wrong action the teacher might say "error", just verbalize the correct action, or say nothing. In all circumstances the learner perceives the feedback with noise and so it can even hear the wrong feedback. Table 4.1 shows all the possible

feedback protocols that can range from a pure learning from demonstration behavior (protocol 1) to a pure binary reinforcement one (protocol 8). Each protocol is defined with the feedback that the teacher provides the learner when it does the correct action and when it does the wrong action. The teacher might choose to say the correct action (A), say nothing (∅), give a confirmation (O) or inform the learner that the selected action is wrong (W). This protocol is ambiguous and the same feedback (∅) can either mean correct or incorrect. If more than one correct action is available in a state then the teacher provides, randomly, one of them. To model perceptual errors there is a probability of receiving a random sign instead of the correct one. The only restriction we have in the protocol is that a (W) message after a correct action is made or an acknowledge (O) when a wrong action is executed are only given with low probability. These assumptions model the perceptual noise of the learner and give a small bias that improves the convergence of the algorithm by disambiguating the different protocols. More general protocols could be considered, but for computational efficiency we reduced to a small set that allows the implementation of an efficient filter.

Table 4.1: The 8 feedback protocols considered. Possible feedback instructions given by the teacher when the learner does the **correct** or **wrong** action are: the action name (A), nothing (∅), correct (O) or wrong (W).

ActionFeedback	1	2	3	4	5	6	7	8
Correct	A	A	A	∅	∅	O	O	O
Wrong	A	∅	W	A	W	A	∅	W

Each different teacher that will be modeled as a convex combination of these protocols. For the teacher model we will consider a set of parameters M that describe the mixture of protocols in Table 4.1. We do this to be able to explain more teacher behaviors than just the predefined models, this is specially important when we do not know the level of noise on each protocol. As an example, consider $M = [0\ 0.8\ 0\ 0\ 0\ 0.2\ 0\ 0]$, the statistical model for the feedback is as follows:

$$\text{if A is optimal} \quad \begin{cases} p(F = A|A, M) & = 0.8 \\ p(F = O|A, M) & = 0.2 \end{cases}$$

$$\text{if A is non-optimal} \quad \begin{cases} p(F = \emptyset|A, M) & = 0.8 \\ p(F = A|A, M) & = 0.2 \end{cases}$$

This combines 80% of the time a teacher that reinforces the behavior of the learner when it is correct by providing the correct action and says nothing when the action is wrong, and 20% of the time a teacher that confirms that the chosen action was correct or provides it when the learner chooses it wrong.

We have to extend the model in Eq. 4.1 to include the ambiguous feedback. Our posterior now depends not only on the demonstration but also on the feedback model. By independence we can get the following factored model:

$$
\begin{aligned}
& p(R_{t+1}, M_{t+1}|A_{0:t}, F_{0:t}) \\
& \quad \propto \quad p(F_t|A_t, R_t, M_t)p(R_t, M_t|A_t) \\
& \quad \propto \quad p(F_t|A_t, R_t, M_t)p(A_t|M_t, R_t)p(R_t, M_t) \\
& \quad = \quad p(F_t|A_t, R_t, M_t)p(A_t|R_t)p(R_t, M_t) \qquad (4.2)
\end{aligned}
$$

4.1.3 Sign-Meaning Model

Another aspect of human-robot interaction systems is that the feedback is often given using a natural interface such as gestures or speech. Most of the times there is an implicit assumption that the vocal signs are assumed to have a known semantics for the learner. Now, we will relax this assumption and allow the teacher to provide instructions to the learner that are unknown. We will define the feedback as the instruction the teacher wants to provide to the learner, as defined in Table 4.1, and the signs as the words actually provided by the teacher. In this way it is possible for the learner to accept new words and learn their meanings. As an example, the teacher might say "good", or "ok", or "correct" and the learner should always understand it as a confirmation, i.e. the different signs all correspond to the same feedback.

We have to extend the previous feedback model, in Equation 4.2, to include the uncertainty in the signs received. We will consider a new relation that gives the probability of having a feedback sign g when the teacher wants to provide a given feedback f, $p(g|f, .)$. As the feedback is no longer observed, we have to integrate it out from the observation of the feedback. Finally, we get the following expression:

$$
p(G_{t+1}|D_t) = \sum_g p(G_t|F_t)p(g|D_t) \qquad (4.3)
$$

This posterior distribution on the sign-meaning vector can also be implemented as a particle filter.

4.1.4 Algorithm

The algorithm involves the estimation of three entities from data: the reward, the feedback model and the meanings of the feedback signs. We will use a particle filter to estimate all the variables of interest. To reduce the number of particles we will not represent the full joint distribution but only an approximate of each marginal. We update the weight of each particle taking into account the *maximum a-posteriori* estimate of the other variables. Table 4.2 summarizes the algorithm.

Feedback		
Signs		Meanings
7* Known	up	$\uparrow$
	down	$\downarrow$
	left	$\leftarrow$
	right	$\rightarrow$
	$\emptyset$	CORRECT/WRONG
	ok	CORRECT
	error	WRONG
3* Unknown	good	?
	bad	?
	$\vdots$	?

Fig. 4.1: Relation between feedback signs and intended feedback meaning. There are only $Na + 3$ feedback meanings, one corresponding to each available action and the meanings of CORRECT and WRONG. They are fixed and known from the beginning. We assume that there is at least one feedback signal with a known correspondence to a feedback signal, there is the possibility of unknown feedback signs to exist and their relation to the feedback must be learned. For instance the teacher might say good instead of ok. The table shows an example when the agent has 4 available actions (up, down, left and right).

Table 4.2: Algorithm for the joint estimation of the task representation, feedback and sign-meaning models. It combines three particle filters to approximate the posterior distribution of the three variables.

- Select number of samples n_r, n_g and n_m
- Sample n_r reward vectors
- Sample n_g sign-meaning parameters
- Sample n_m protocol parameters

  1. Sample state x
  2. Choose and execute action a
  3. Observe feedback sign g_t
  4. Sample feedback from $f_t \sim p(f|g_t)$
  5. Find best feedback parameters $M = argmax_i w_f^{(i)}$
  6. $w_r^{(i)} \leftarrow p(f_t|A_t, R_t^i, M)p(A_t|R_t)w_r^{(i)}$
  7. Resample reward particles
  8. Find best reward parameters $r^* = argmax_i w_r^{(i)}$
  9. $w_f^{(i)} \leftarrow p(f_t|A_t, r^*, M_t)p(A_t|r^*)w_f^{(i)}$
  10. Resample feedback model
  11. $w_g^{(i)} \leftarrow \sum_i p(g_t|f_t)w_g^{(i)}$
  12. Resample sign-meaning model
  13. goto 1

4.1.5 Active Sampling

The previous algorithm keeps an approximation of the posterior distribution of the reward function. We can use this information to allow the learner to ask the teacher for more informative samples. We do not consider any intrinsic motivation on the system [145] besides that of reducing uncertainty. From the reward distribution it is difficult to decide what state, or action, provides more information. We can follow the active learning extension for IRL as presented in [146], or alternatively [147], to allow the learner to request the most informative samples. In that approach the policy distribution is inferred from the distribution on the rewards. Then, for each state, a measure of the uncertainty is made to select the state where the policy posterior has higher variance. Intuitively, this state is the state where the rewards agree least.

The criteria used is, for each state, the variance of the weighted sum of all policies.

$$I(x) = variance\left(\sum_i w_r^i \pi^i(x,a)\right)$$

The most informative state will be the one where the previous criteria is smaller, meaning that the policy distribution is flat. The action is selected randomly. We note that this exploration strategy just takes into account the uncertainty on the reward. Creating an exploration strategy based on the uncertainty of the sign-meaning estimation does not provide a gain due to the probabilistic model we used. Other sampling criteria that we are going to test is a counter based random sampling of states and actions, and random states with actions sampled with the usual $\epsilon - greedy$ strategy.

4.2 Results

In this Section we present the results from our algorithm in a set of simulated environments.

4.2.1 Navigation Task

We consider a simulated environment with 5 different actions, the number of states in the discretization grid varies in each problem. All results report averages of 20 executions of the algorithm with different parameters. The true reward function to be found by the learner is randomly generated at each experiment, the same occurs for the meaning and feedback models. The reward in this abstract problem can be seen as corresponding to a navigation task and so the reward is the goal location.

Figure 4.2 compares two situations, the first where the learner estimates the feedback model of the teacher and the second where it does not estimate the feedback model.

But, in both cases, considering that *all feedback signs are known*. We can see that learning is faster and with a better quality if a model is estimated and so it shows the importance of our approach. Some protocols are equivalent in terms of speed of learning even without any particular assumption about it. But consider, for instance, what the teacher means when it does not say anything. In some protocols that is equivalent to say it is correct and in some others it means that the action is wrong. Only after knowing this relation can the learner make use of that data to improve its estimation of the task representation.

Fig. 4.2: Comparison of learning of the task model with or without learning the feedback model. Number of states is 400. The figure shows the likelihood of the best estimate of the reward function.

From Figure 4.3 we can see that the task can be learned even under a noisy feedback signal, and that we can learn simultaneously the model of the feedback behavior. Around 10% of the feedback signals were noisy. The same figure also compares the different sampling methods. We can see that the active exploration is able to learn faster, with less variance and with a better asymptotic convergence. This situation happens even if the active criteria was developed without taking the noise in the feedback into account.

We now present our full system where there are *some feedback signs with unknown*

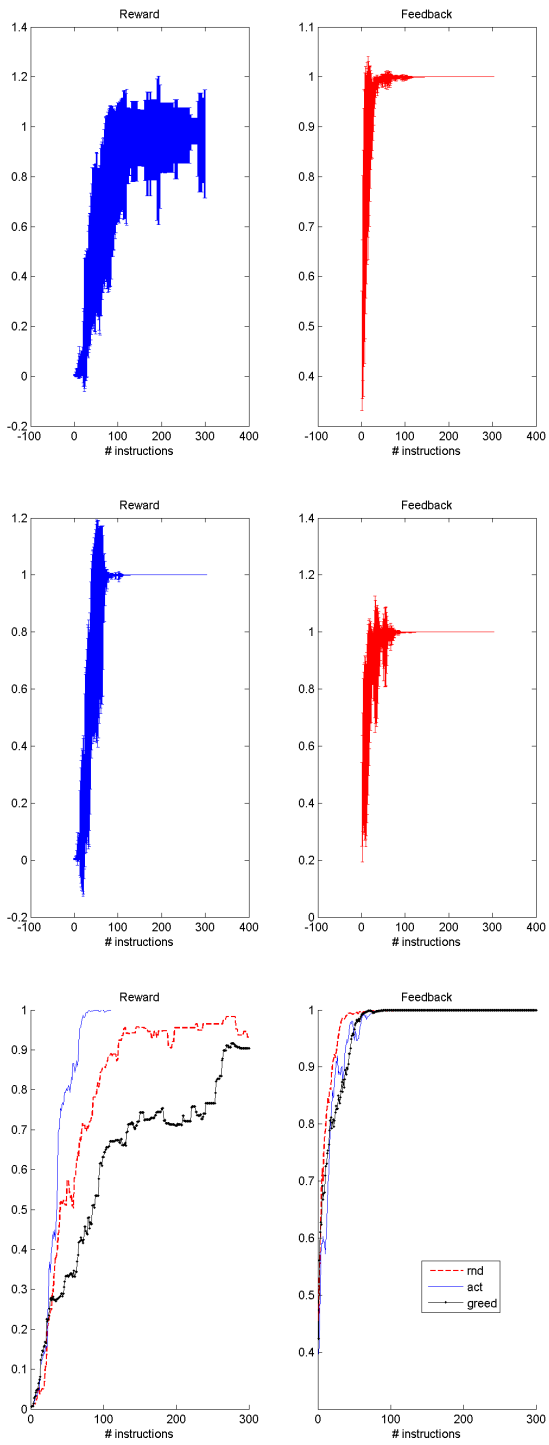

Fig. 4.3: Simultaneous acquisition of the task and the feedback models with three different exploration methods for a problem with 225 states. The figures show the likelihood of the best model for the reward and the feedback The top figure shows the results for random exploration and the middle one for active exploration. The bottom one compares both, and also one using ϵ-greedy exploration. Results are for 10 runs, the mean are variance bars are shown. The active exploration method learns faster with smaller variance and bias.

meanings. The system needs to learn the task, the feedback model and the map of new signs to their meanings. We consider 7 feedback signs whose meanings are known, i.e. the five actions plus O and W, and 7 new signs that can map to any of those meanings. Figure 4.4 shows the results using 100 particles for the estimation of the sign-meaning relations. The first conclusion is that the system can learn all the important variables and, again, the active exploration method learns faster and with less variance than the other methods. Not all the signs meanings are successfully estimated and this situation is caused by a very asymmetrical sampling of the feedback signs. We can observe this in Figure 4.5. For instance, a teacher that always gives the correct action will never use the signs for correct and wrong and so their meanings will not be learned.

4.2.2 Collecting Objects

We now consider an environment where the learner can navigate and where there is a probability of finding three different objects. The learner has to learn which objects it should collect, or not, and for each of the object classes learn where they must be delivered. The number of actions is now 7, the 5 navigation ones plus collect and release. The number of feedback signs is now 10, again we assume that we have an initial known set of signs and the teacher will provide 10 new synonyms.

Figures 4.6 and 4.7 give the results for a problem with three objects and 64 possible locations. In each execution of the problem the system randomly selects the objects that should be collected and their delivery locations. Results are qualitatively equal to the previous problem.

4.3 Conclusions

This chapter showed how a learning system can learn a task description when the feedback it gets from the teacher does not follow a *rigid protocol* and is *very noisy* (10% error in correctly recognizing the feedback signs). We showed that a learner can estimate simultaneously the feedback protocol and the task representation in a reasonable amount of time and computational complexity. This exemplified the idea of learning how to learn that is central to the formalism. An imperfect learning algorithm (for example one that does not have access to a complete understanding of the feedback words of the teacher) can be used to learn a task or part of a task. A better task model can be used to update the interpretation hypothesis, and a better interpretation hypothesis can be used to update the task model. This is a perfect setup for concurrent updating, and the chapter showed that this can work in practice.

By bootstrapping the systems with some known sign-meaning correspondences, the system could successfully estimate the correspondences of new feedback signs. To further improve the efficiency of the system we presented an active learning approach where the

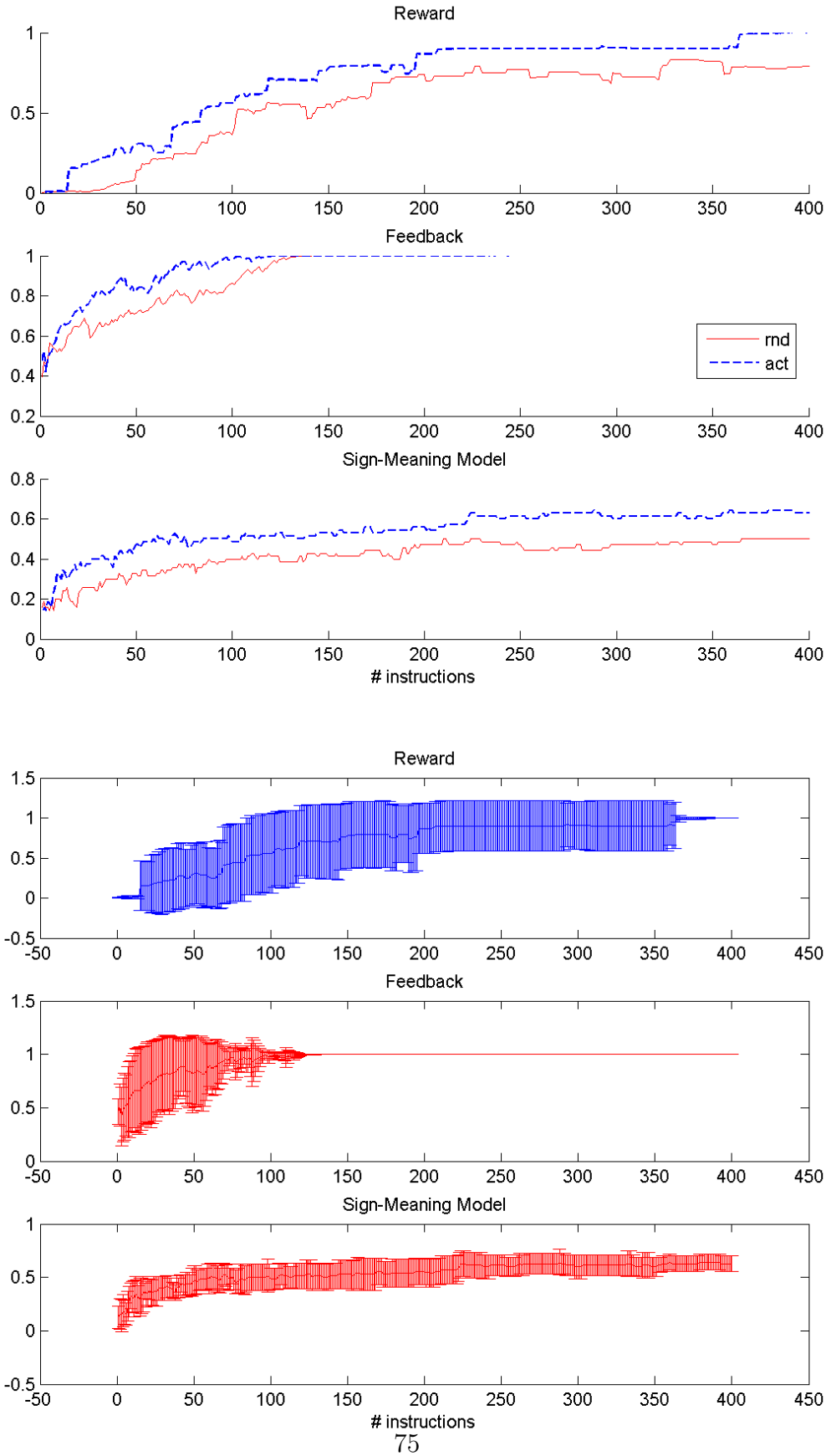

Fig. 4.4: Learning with 400 states (top) comparison between sampling methods, (bottom) mean and variance for the active method.

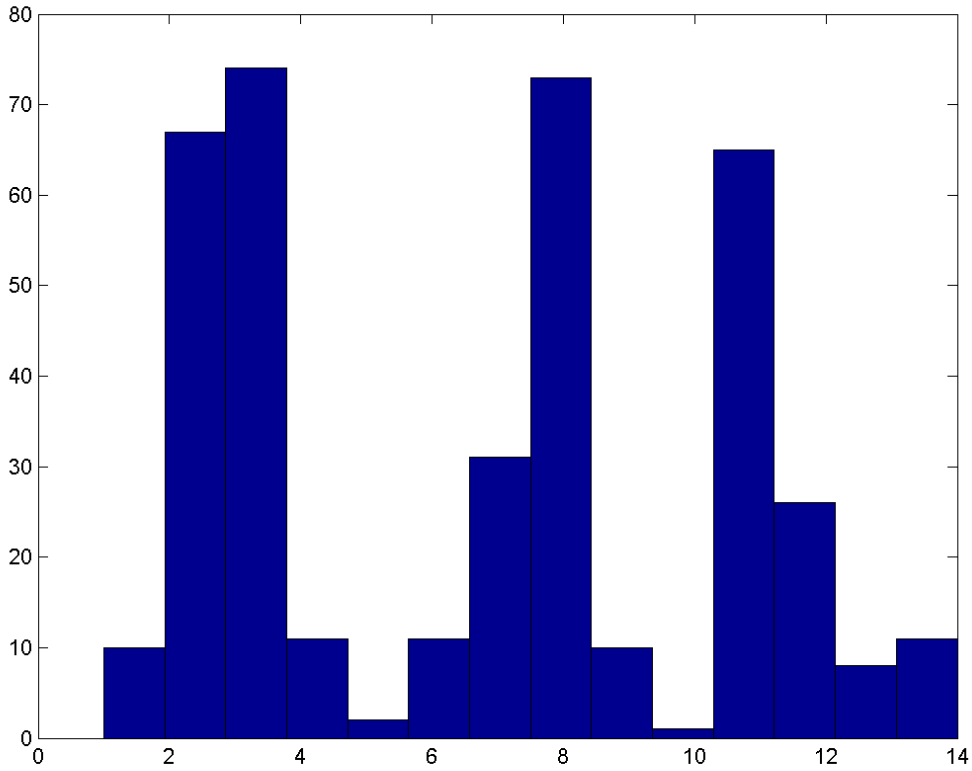

Fig. 4.5: Histogram of observed feedback signs. We can see that some signs are very rarely used thus making it impossible to estimate their meanings.

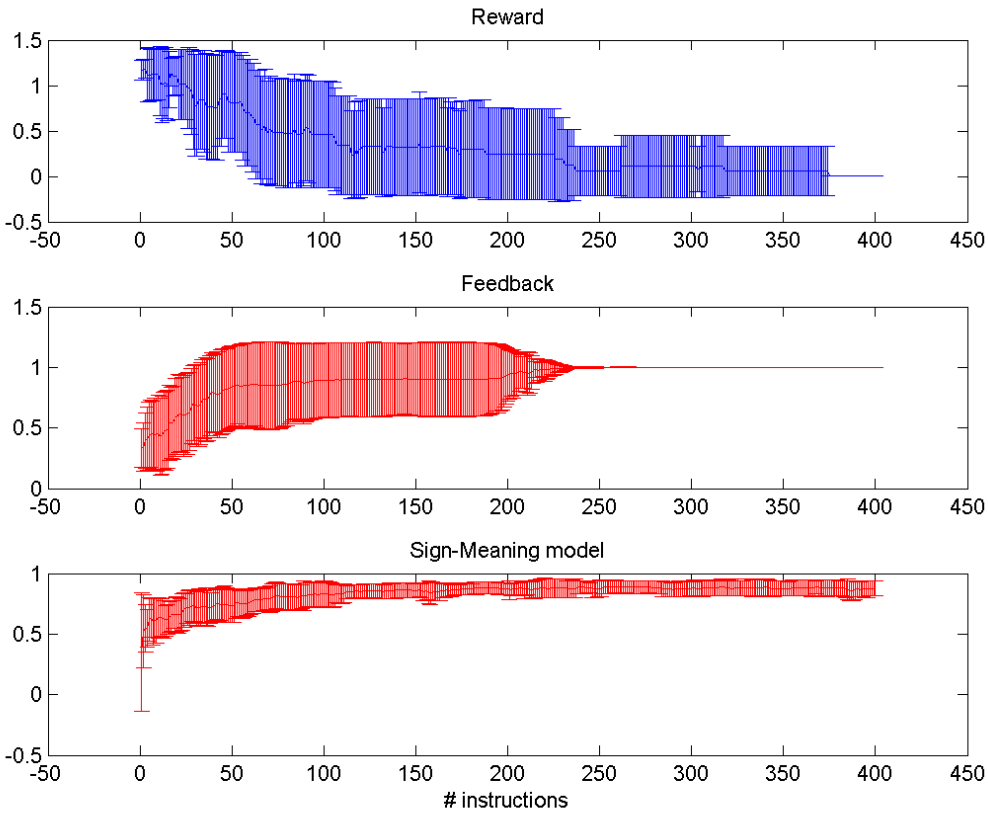

Fig. 4.6: Mean and variance for the active learning method in the "Object Collecting" Task. The system is able to learn the task, the feedback system and new feedback signs. Top - policy loss; Middle - likelihood of correct feedback model; Bottom - number of correctly assigned signs.

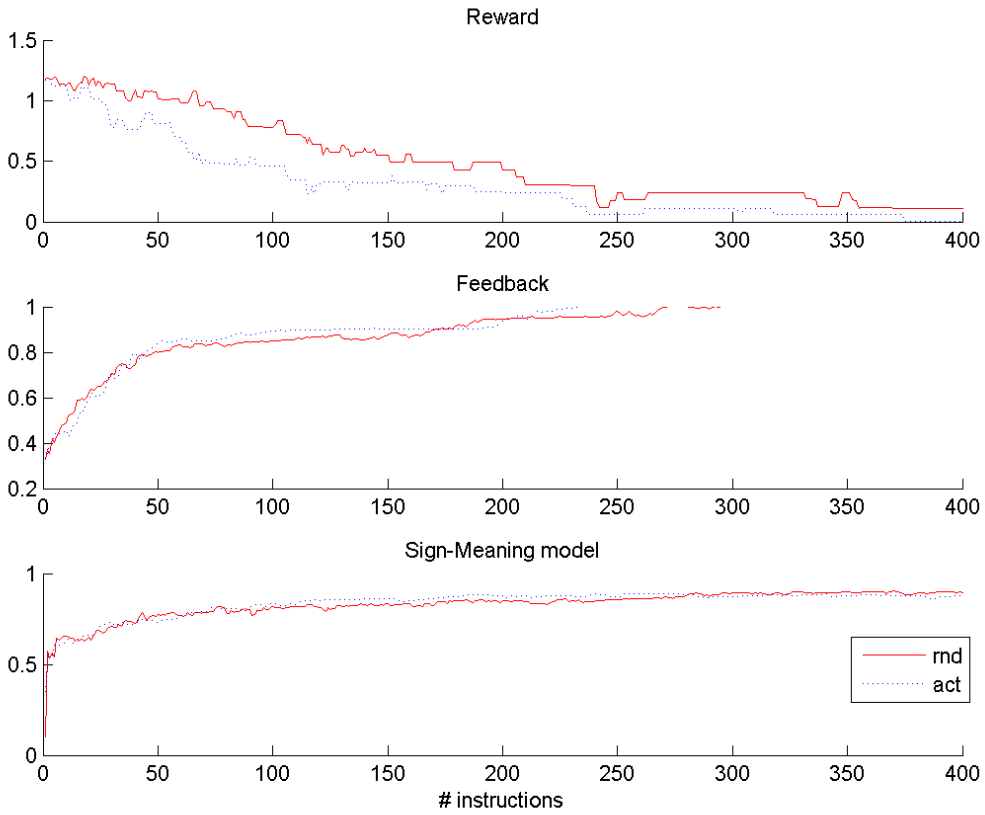

Fig. 4.7: Comparison between active and randomly sampling in the "Object Collecting" Task. The system is able to learn the task, the feedback system and new feedback signs. Top - policy loss; Middle - likelihood of correct feedback model; Bottom - number of correctly assigned signs.

learner asked the teacher for specific information in states where it is more uncertain. This results in faster learning than with a simple random strategy with a smaller variance and bias.

We tested our system in different problems and the qualitative results are consistent among different domains and the system is able to learn all the entities. The results degraded when the noise level increases. The improvement we get from active sampling is dependent on the quality of the posterior estimation and thus it is important to have features that can correctly describe the possible tasks. In terms of the number of signs, in our system we had to assume that some correspondences are known to improve the convergence.

This research can also be compared to language learning research, where we learn synonyms for words labeling actions from a teacher already proficient with the language. The information used to learn these synonyms is not traditionally used in language learning research. It is often the case that a label is used to describe an object or property of an object (and a difficulty often encountered is that it is unknown what property the label refers to), see for example [55]. In our research a label is associated with a desired action that is in fact never seen (the person that knows the language does not perform demonstration of the actions but instead gives the label of the action). The meaning of the label is instead found by building a model of what task the teacher is trying to teach the learner, where a good model of the feedback helps learning a good task model and vice versa (under a set of observations some feedback model task model pairs typically becomes much more probable).

Chapter 5

Unlabeled demonstrations of an unknown number of tasks

Contents

5.1 Algorithm **81**

5.1.1 Gaussian Mixture Regression 82

5.1.2 Incremental Local Online Gaussian Mixture Regression (ILO-GMR) 83

5.1.3 How do we pick local points 85

5.2 Experiment **87**

5.2.1 Reproduction while introducing additional tasks 90

5.2.2 Reproductions outside normal starting positions 90

5.2.3 Do the framings make a difference? 91

5.2.4 How many demonstrations are needed? 91

5.2.5 Comparison with GMR 94

5.3 Discussion **97**

In this chapter and the next, several experiments are introduced which exemplify specific learners as specified by the formalism. The focus will be on exploring new types of setups rather than offering solutions to already explored setups. This is done by starting with a learner that observes demonstrations. The new element in this chapter will be to explore how a learner can solve the problem if a critical assumption of many proposed interpretation hypotheses used in imitation learning is removed. Specifically the assumption that the learner knows what task is being demonstrated. When this is not the case, the learner have a whole new set of problems to solve. The research of this chapter and the next chapter was performed while the formalism was being developed, and while the experiments are certainly instances of the proposed formalism, they are not the prototypical cases or the cases most suitable for exemplifying the formalism. In this chapter the position of an object determines which task is being demonstrated by the teacher/demonstrator or which task should be performed by the learner/imitator. We can view this object as a teaching signal that is being interpreted. The learner does

not start with the knowledge that it is the object position that determines what task should be produced.

When the teacher, which in this case is a demonstrator, provides demonstrations of an unknown number of tasks and the learner is not provided with symbolic labels indicating what task is being performed several novel problems arise. The learner, who in this case is an imitator, needs to figure out the number of separate tasks that has been demonstrated (a set of demonstrations could constitute many demonstrations each of two tasks, or a few demonstrations each of several tasks), which demonstrations where of what task and finally when it should perform what task. The demonstrator may alternate demonstrations corresponding to different tasks in an uncontrolled or even random order. Yet, we assume that elements of the sensorimotor context can allow the robot to statistically infer information that may allow it to accurately reproduce the right task in a given sensorimotor context.

A mechanism is detailed and an experiment is presented showing that it is possible to learn and reproduce several tasks even when the demonstrations are unlabeled. Each tasks should be solved by a policy that takes inputs in a specific subset of the input space, and we will call each such subset a framing. In this case there is no explicit calculation of the number of tasks, and the framing is recalculated at each time step.

Gaussian Mixture Regression has been shown to be a powerful and easy-to-tune regression technique for imitation learning of constrained motor tasks in robots[108, 101, 125, 69]. Yet, current formulations are not suited when one wants a robot to learn incrementally and online a variety of new context-dependant tasks whose number and complexity is not known at programming time, and when the demonstrator is not allowed to tell the system when he introduces a new task (but rather the system should infer this from the continuous sensorimotor context). As will be demonstrated with an experiment, this limitation can be addressed by introducing an Incremental, Local and Online variation of Gaussian Mixture Regression (ILO-GMR) which successfully allows a simulated robot to learn incrementally and online new motor tasks through modelling them locally as dynamical systems, and able to use the sensorimotor context to cope with the absence of categorical information both during demonstrations and when a reproduction is asked to the system. Moreover, we integrate a complementary statistical technique which allows the system to incrementally learn various tasks which can be intrinsically defined in different frames of reference, which we call framings, without the need to tell the system which particular framing should be used for each task: this is inferred automatically by the system.

5.1 Algorithm

The algorithm used, Incremental, Local and Online formulation of Gaussian Mixture Regression (ILO-GMR), does not rely on an explicit segmentation or clustering of demon-

strations. This algorithm was introduced in [102] and its use for learning multiple tasks from unlabeled demonstrations was introduced in [104].

ILO-GMR incrementally and locally models the set of all tasks as a single dynamical system. Different tasks just correspond to different regions of the state space and the thing that changes between different tasks is the part of the state that continuously (as opposed to symbolically or categorically) models the sensorimotor context. The central idea of this technique is to build online and on-demand local Gaussian Mixture Regression regression models of the task(s).

During training in this approach, data points are stored incrementally in a data structure which allows for very fast approximate nearest neighbors retrieval (e.g. such as in [115]).

Then, at prediction/reproduction time, the method looks at the points in the database that are close to the current state of the system, including sensorimotor context information, and a local GMR model is built online. As the model is very local, only a few gaussians are needed (typically 2 or 3), and experimental results show that an EM and GMR with 2 or 3 gaussians and around 100 points can be run in a few milliseconds on a standard computer, which is enough for many real-time robot control setups. Thus, new data points from a demonstration, either of a task already seen or of a new task, can be exploited immediately without heavy recomputations and without any programmer intervention (standard parameters of ILO-GMR experimentally work for large number of tasks of varying complexity) giving us the advantages of a truly incremental and online algorithm.

5.1.1 Gaussian Mixture Regression

The GMR approach [108] first builds a model (typically in task space, but models in the joint space can also be used) using a Gaussian Mixture Model encoding the covariance relations between different variables. If the correlations vary significantly between regions then each local region of state space visited during the demonstrations will need a few gaussians to encode this local dynamics. Given the number of gaussians, the use of an Expectation Maximization (EM) algorithm finds the parameters of the model.

A Gaussian probability density function consists of a mean μ and a covariance matrix Σ. The probability density ρ of observing the output v from a gaussian with parameters μ and Σ is:

$$\rho(v) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\{-\frac{1}{2}(v-\mu)^T\Sigma^{-1}(v-\mu)\} \tag{5.1}$$

To get the best guess of the desired output (e.g. speed in cartesian space of the hand in the robot experiments below) $\hat{v}$ given only the current state x_q (e.g. position and speed

of the hand in various referentials and position of an object construing the context, as in the experiments below) we have:

$$\hat{v}(x_q) = E[v|x = x_q] = \mu^v + \Sigma^{vx}(\Sigma^{xx})^{-1}(x_q - \mu^x) \tag{5.2}$$

Where Σ^{vx} is the covariance matrix describing the covariance relations between x and v.

A single such density function can not encode non linear correlations between the different variables. To do this we can use more than one gaussian to form a Gaussian Mixture Model defined by a parameter list $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_M\}$, where $\lambda_i = (\mu_i, \Sigma_i, \alpha_i)$ and α_i is the weight of gaussian i. To get the best guess $\hat{v}$ conditioned on an observed value x_q we first need to know the probability $h_i(x_q)$ that gaussian i produced x_q. This is simply the density of the gaussian i at x_q divided by the sum of the other densities at x_q, $h_i(x_q) = \frac{\rho_i(x_q)}{\sum_{j=1}^{M} \rho_j(x_q)}$ (where each density $\rho_i(v)$ is calculated just as in (5.1), with Σ replaced by Σ_i^{xx}, v with x_q, etc). Writing out the whole computation we have:

$$h_i(x_q) = \frac{\frac{\alpha_i}{\sqrt{|\Sigma_i^{xx}|}} \exp\{-\frac{1}{2}(x_q - \mu_i^x)^T(\Sigma_i^{xx})^{-1}(x_q - \mu_i^x)\}}{\sum_{j=1}^{M} \frac{\alpha_j}{\sqrt{|\Sigma_j^{xx}|}} \exp\{-\frac{1}{2}(x_q - \mu_j^x)^T(\Sigma_j^{xx})^{-1}(x_q - \mu_j^x)\}}. \tag{5.3}$$

Given the best guesses $\hat{v}_i(x_q)$ from (5.2), and the probabilities $h_i(x_q)$ that gaussian i generated the output, the best guess $\hat{v}(x_q)$ is given by:

$$\hat{v}(x_q) = \sum_{i=1}^{M} h_i(x_q)\hat{v}_i(x_q) \tag{5.4}$$

The parameter list is found using an Expectation Maximization algorithm (EM) [109] that takes as input the number of gaussians and a database.

5.1.2 Incremental Local Online Gaussian Mixture Regression (ILO-GMR)

With ILO-GMR, the datapoints of all demonstrations, possibly including demonstrations of different tasks, are stored in a full data structure D which allows later on for very fast approximate nearest neighbors queries. The datastructure we use is a kd-tree-like incremental variant of approximate nearest neighbors algorithm presented in [115] and already shown to be very efficient in high-dimensional computer vision applications. Then, during each iteration of the reproduction of a task the robot looks at his current state x_q and extracts a local database $D(x_q)$ consisting of the N points closest to x_q using the fast query algorithm. These points are now used as input to GMR as described

above along with a number M of gaussians to use (typically equal to 2 or 3). N is the first parameter of ILO-GMR and is typically slighlty superior to the second parameter M multiplied by the dimensionality of the sensorimotor space. The EM algorithm builds a GMM and then we get the best guess of the current desired speed $\hat{v}(x_q, D(x_q), N, M)$ as described above. The local points are found online during reproduction and therefore it allows the system to take advantage of information it has just acquired as easily as old information available before task execution began.

The number of gaussians M and the number of local points N does not need to be changed when a demonstration of a new task is introduced and the modeling is done online. Thus we can add new demonstrations of old tasks or demonstrations of new tasks incrementally to the system without tampering with model parameters or recomputing a model. The pseudo-code of the algorithm is provided below.

Computational complexity and ease-of-tuning

An important difference between the standard GMR approach and ILO-GMR is that in ILO-GMR an important part of processing is shifted from the training period to online computation. Hopefully, since models in ILO-GMR are "very local", using only 2 or 3 gaussians will be enough for reaching high accuracy while allowing for very fast EM and local GMR steps. As far as the incremental training is concerned, [115] has shown that the datastructure for fast nearest neighbours retrieval could also be updated very fast. In order to make an initial experiment to evaluate ILO-GMR both in terms of accuracy for difficult robot-related regression tasks, as well as computational speed, we have compared the performance of ILO-GMR with other state-of-the-art regression methods, including GMR, on the hard regression task defined in the SARCOS dataset which has been used several times in the literature as a benchmark for regression techniques in robotics. This dataset encodes the inverse dynamics of the arm of the SARCOS robot, with 21 input dimensions (position, velocity and acceleration of 7 DOFs) and 7 output dimensions (corresponding torques). It contains 44484 examplars in the training database and 4449 test examplars. It is available at: http://www.gaussianprocess.org/gpml/data/). The regression methods to which we compared performances on this dataset are: Gaussian Mixture Regression (GMR, [108] and [130]), Gaussian Process Regression (GPR, [111]), Local Gaussian Process Regression (LGP, [113]), support vector regression (v-SVR, [114]) and Locally Weighted Projection Regression (LWPR, [112]). All those algorithms were tuned with reasonable effort to obtain the best generalization results. For ILO-GMR, optimal tuning was done with N=200 and m=2, but results degrade very slowly when moving away from these parameters. Figure 5.1 shows the comparison of the performances of those algorithms for predicting the torques of the first joint in the SARCOS database. We observe that the performance of ILO-GMR matches nearly the best performance (GPR), is slightly better than v-SVR, LGP and GMR, and clearly better than LWPR while being also incremental but much easier to tune. Furthermore, in spite of the fact that our current

implementation of ILO-GMR was done in Matlab and is not optimized, it is already able to make a single prediction and incorporate a new learning examplar in around 10 milliseconds on a standard laptop computer and when 44484 SARCOS data examples are already in memory. Furthermore, we have measured experimentally the evolution of training and prediction time per new examplar: it increases approximately linearly with a small slope in the range 0-44484 learning examplars. Finally, it can be noted that the parameters N and M can be chosen the same for the SARCOS database as for the various motor tasks demonstrated in the experiments presented below, in spite of the fact that they are defined in very different sensorimotor spaces: this illustrates the ease-of-tune of ILO-GMR.

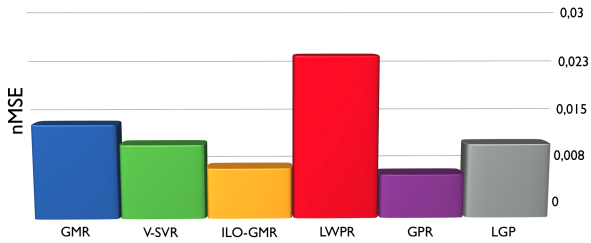

Fig. 5.1: Comparison of ILO-GMR with state-of-the-art regression algorithms for the SARCOS dataset encoding the inverse dynamics of an arm with 21 input dimensions. Here, only the nMSE for the regression of the torque of the first joint is displayed.

5.1.3 How do we pick local points

A potential problem with the use of ILO-GMR in a number of robot learning by demonstration applications is that for given tasks there may be irrelevant or redundant or badly scaled variables defining the state x_q that may cause problems for generalization through the nearest neighbours search. We address these issues in this paragraph.

First, we do not want the importance of a dimension to be encoded in the size of its values so we rescale all the data so that every dimension has the same variance and mean zero. This rescaling uses the entire data set so adding new data means that the rescaling constants should ideally be updated: this can however be done quickly and incrementally. A global model is able to capture the relevant dimensions for every task in every region of the state space given that the number of gaussians are appropriate and that it has enough training data. We must make sure that this very important property of the global GMR algorithm is not lost in ILO-GMR. The way in which local points are selected correspond to an assumption about what dimensions are relevant to the task. To pick the local points from a dataset D with K number of dimensions $d_1, d_2, ..., d_K$; we must decide how we measure distance between two points $p_1 = (x_{11}, x_{12}, ..., x_{1K})$ and $p_2 = (x_{21}, x_{22}, ..., x_{2K})$. If we define a subset of n dimensions $dim_1 = (d_i, ..., d_j)$ as the relevant dimensions the distance in this subset is the distance between the p_1 and p_2

in this subspace $distance(dim_1, p_1, p_2) = \sqrt{(x_{1i} - x_{2i})^2 + ... + (x_{1j} - x_{2j})^2}$. Each subset of dimensions define its own distance measure and thus each subset defines its own set of local points. The local set of points is now uniquely defined by the full database D, the current state x_q and the set of dimensions dim_l. Let's say that the task is to move the robots hand in a 2-D "S" shape and that the task is demonstrated by making "S" shapes in different locations, stored in D (this is one of the tasks that will be used in the experiments). The relevant set of dimensions correspond to the position of the hand *relative* to the starting position of the hand. If we pick local points from the demonstrations based on hand positions in the body referential of the robot (i.e. absolute referential) we will get local points from different parts of the task and maybe none of these points will be from the correct part of the task. We call a subset of dimensions one possible way of framing a task and the resulting local dataset $D(x_q)$ is the current situation viewed in this framing. We can also use the set of dimensions of a framing to view a demonstration, and will do so throughout the chapter.

For example, we look at demonstrations of the first part of the task to draw an S shape in figure 5.2. Too the left the vectors are plotted in a space where the imitators hand position is measured relative to the robots body, and to right they are plotted relative to the starting position of the imitators hand (i.e. where the hand happened to be initially when it was asked to observe/reproduce the task). The points are shown in a 4 D space (2 dimensions determining the position of the vectors and 2 dimensions determining the shape of the speed vectors). We can see that picking a set of points close to each other will result in points from the same part of the task only in the framing of hand position relative to starting position, and we say that this is the correct way to frame the task. If instead the task is to move the hand to and then around an object the relevant set of dimensions is the hand position relative to the object.

Since the robot does not have access to the relevant dimensions of the different tasks it is to perform it needs a way to measure the quality of a framing by just looking at the raw demonstration data. To determine the quality of the set of dimensions dim_f we find the subset D_f consisting of the N points of the full data set that are closest to the current state x_q when measuring distance in the dimensions dim_f. We use this database to build a GMM, using the EM algorithm to set the parameters λ_f. For each point P_{fn}, with n=1,2,...,N in D_f we have a state x_{fn} and a desired velocity y_{fn}. We now do GMR, as described above, on each of the states x_{fn} and get N number of predictions $\hat{y}_{fn}(x_{fn})$. We now determine the relative weights of the recommendations $\hat{y}_f(x_q)$ by comparing training error of angles. The GMR is presented the full dataset in all the dimensions, the framing only affects which points is used as input. The full algorithm is presented in pseudo code in algorithm 4 (the time variable t is not visible to the robot). We are forced to use training error since a true validation set would have to consist of points from an entire demonstration not seen before (the demonstrations are not labeled so this data is not available).

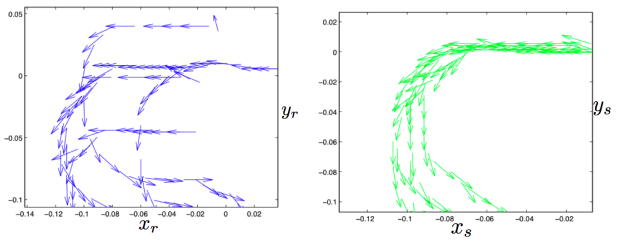

Fig. 5.2: This shows five demonstrations of one of the tasks learned in the experiments presented below (specifically task 2, which is to draw an S shape starting from the hands starting position). In the left plot we see the vectors in the framing of hand positions relative to the robot. And in the right plot relative to the starting position of the robot imitators hand. Points that are close to each other in the figure to the left are sometimes from different parts of the task, because the position relative to the robot is not relevant to the task.

5.2 Experiment

In this section, we present an experiment in a simulated robotic setup which shows how ILO-GMR can be used for learning incrementally four context-dependant motor tasks, defined in different frames of reference, without specifying in advance the number of tasks, without programmer intervention at demonstration time, and without providing categorical labels for the different demonstrations of the tasks.

The world consists of a 2D simulated robot hand and one object (we assume we have a 2D multi-link arm with a precise inverse kinematics model which allows to directly work in the operational space of the hand). During demonstrations a demonstrator takes the simulated hand of the robot and moves it (using the mouse), hence we set ourselves in the kinesthetics demonstration framework, as in [108]. The state space of the system is 8 dimensional and includes the 2-D position of the robot hand in 3 different frames of reference, as can be seen if figure 5.3. One coordinate system is centered on the starting position (x_s,y_s), one is centered on the robot (x_r,y_r) and one is centered on the object (x_o,y_o). We write the position of the hand in these three coordinate systems as (H_{x_s},H_{y_s}), (H_{x_r},H_{y_r}) and (H_{x_o},H_{y_o}). We write the position of the object in the coordinate system of the robot as (O_{x_r},O_{y_r}). From this set of 8 dimensions we define 3 different subsets:

- Framing 1: H_{x_s}, H_{y_s}, O_{x_r} and O_{y_r}
- Framing 2: H_{x_r}, H_{y_r}, O_{x_r} and O_{y_r}
- Framing 3: H_{x_o}, H_{y_o}, O_{x_r} and O_{y_r}

Finally, the action space of the robot is 2 dimensional and consists in setting the speed vector of its hand. The speed is the same in all the 3 coordinate systems since they all have the same orientation.

Algorithm 4 Outline of the pseudo-code for reproducing context-dependant motor tasks with ILO-GMR

Input: D, M, N, x_{q0}

• *D is the full database encoded in an incremental kd-tree like structure for fast approximate nearest neighbours search;*

• *x_{q0} is the initial current state;*

• *N is the number of local points;*

• *M is the number of gaussians in the GMM*

• *$\lambda = (\lambda_1, \cdots, \lambda_M)$ is the GMM parameter list;*

• *$D_f(x_{qt})$ is the local database consisting of N points retrieved given the current state x_{qt} and using framing f, for f=1,2,3*

repeat

  for $f = 1$ **to** 3 **do**

    i) Given the current state x_{qt} at iteration nr t; find the local database $D_f(x_{qt}, N)$ for framing f with fast approximate nearest neighbours search.

    ii) Initialize a GMM parameter list $\lambda_{0f} \leftarrow$ k-mean$(D_f(x_{qt}), M)$.

    iii) Compute the GMM parameter list using EM, $\lambda_{x_{qt}f} \leftarrow$ EM$(D_f(x_{qt}), \lambda_{0f})$

    for $i = 1$ **to** M **do**

      iv) Compute $h_i(x_{qt})$ using (5.3)

    end for

    v) Predict the desired vector $\hat{v}_f(x_{qt})$ using (5.4)

    vi) Get the total training angle error E_f of $D_f(x_{qt})$ and the weight of framing f as $w_f = 1/(0.001 + E_f)$

  end for

  vii) Now we have $\hat{v} = \sum(\hat{v}_f(x_{qt}) * w_f)/\sum w_f$

  viii) Use $\hat{v}$ to update the position and get the new state $x_{q(t+1)} = x_{q(t)} + \hat{v} * \tau$, where τ is a time constant

until Reproduction done

There are 4 different tasks with 5 demonstrations for each task, and the element of context which characterizes each task is the location of the object. The first task is "move hand to object", which is demonstrated/should be reproduced when the object is in the upper left corner (but with varying precise positions). The second task is "draw an S from your starting position", which is associated to a position of the object in the upper right corner (again at varying precise positions). The third task is "encircle the object", associated to the object in the lower left corner, and the fourth task is "move your hand in a big cyclic rounded edges squared shape relative to you" and is associated to the object in the lower right corner. The relevant dimensions (not told to the robot) are hand relative to object in the first and third task, hand relative to starting position in the second task and hand relative to robot in the fourth task.

The demonstrations are made using a mouse movement capturing function in Matlab. The starting position and object position is assigned randomly within a square of the

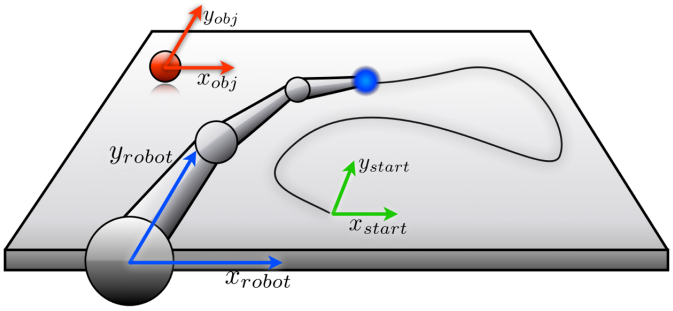

Fig. 5.3: Here we see the 3 coordinate systems of the experiment. In the text we will write x_{start} as x_s, x_{robot} as x_r etc, for short. The 8 dimensional state space consists of the hand position in all 3 coordinate systems plus the position of the object in the coordinate system of the robot.

figure (the starting position in a square centered on the middle and the object is placed somewhere in a square centered in a corner) and plotted in a figure. The demonstrator then clicks a mouse button once and drags the mouse performing the demonstration and clicks again when the demonstration is completed. The positions of the mouse is registered and used to calculate the speed and the relative positions. We can see the tasks demonstrated in figure 5.4. We can see 5 demonstrations of each of the 4 different tasks in figure 5.5.

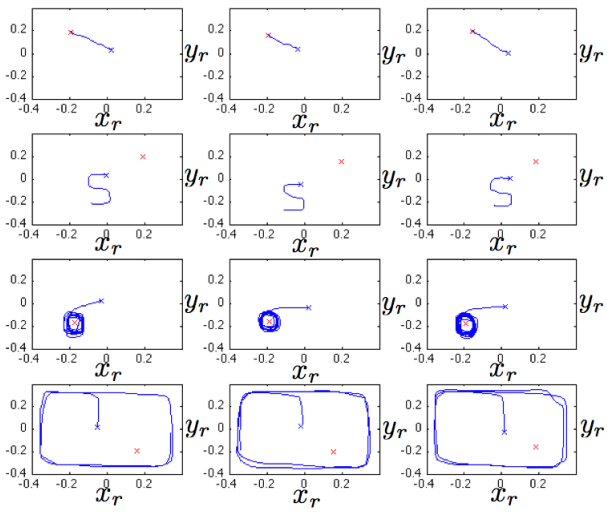

Fig. 5.4: This figure shows 3 individual demonstrations of the 4 different tasks. The top task will be referred to as task 1 in the text, the second highest as task 2, etc. The red cross is the position of the object and the blue cross is the starting position of the hand.

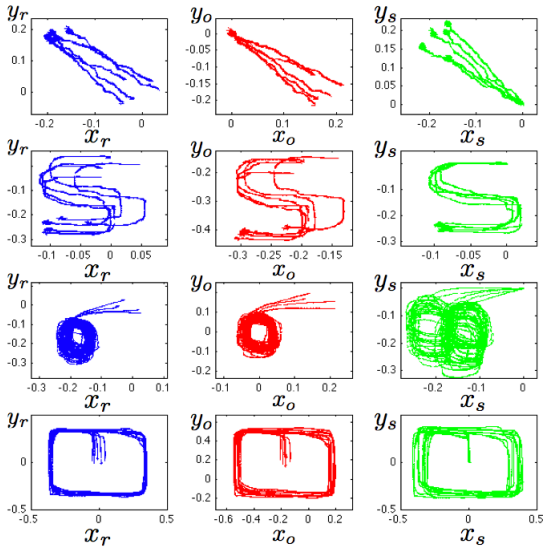

Fig. 5.5: This figure shows 5 different demonstrations for each of the 4 different tasks. The blue figures to the left shows the demonstrations in the framing of hand position relative to the robot, the red figures in the middle show hand positions relative to the object and the green figures to the right show hand positions relative to the starting position. We can see that the demonstrations are more similar if viewed in the correct framing for that particular task.

5.2.1 Reproduction while introducing additional tasks

To make sure that our algorithm can handle new tasks being added we show reproductions of the 4 different tasks in figure 5.6, to the left we see the tasks reproduced with only data of demonstrations of that particular task. The second column shows reproductions after the demonstrations of one more task has been added, the third column shows reproductions after demonstrations of 2 additional tasks have been added and the 4th column furthest to the right show reproductions when all the demonstrations of all the tasks have been made available to the agent. We can see no general trend of task degradation as demonstrations of more tasks are added. In the remainder of this chapter all reproductions shown are of agents that have been shown demonstrations of all tasks (but again, without categorical information).

5.2.2 Reproductions outside normal starting positions

We test the reproduction ability when starting positions are chosen outside the area where the demonstrations started from, and see the results in figures 5.7 to 5.10, for tasks 1 to 4 respectively. In all figures the demonstrations are shown in the correct framing of the task to the top left and then the reproductions are shown in the 3 different framings, all with hand-starting position to the top right, hand-robot to the

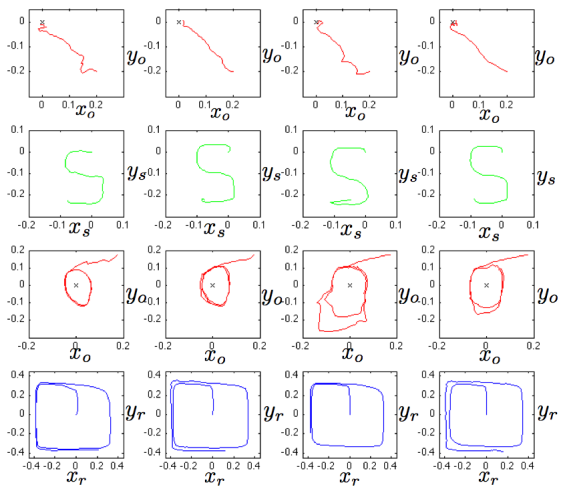

Fig. 5.6: This figure shows that the tasks are successfully reproduced after demonstrated and that adding demonstrations of additional tasks does not destroy performance (one additional task in the second column, 2 in the third and all the other 3 in the fourth).

bottom left and hand-object to the bottom right. The full data of all tasks is available to the agent.

5.2.3 Do the framings make a difference?

In order to determine if the framings are necessary we compare the performance for motor task 2 with the algorithm presented above with the performance of an algorithm that picks only one set of local point where distance is measured in all available dimensions (other than the way the local points are picked the algorithm is identical). In order to make a better comparison we start all demonstrations at the starting position $(0.03, 0.03)$ (relative to the robot). The difficulty of this task is dependent on the starting position so assigning different random starting positions to the two algorithms would just add noise. In figure 5.11 we see 10 reproductions using the algorithm without framing in the top two rows and 10 reproductions using the proposed algorithm in the bottom two rows. We see that the algorithm presented have some problems but in general outperforms the algorithm not using framings. We also see that it is the second part of the task that is the most problematic for both versions.

5.2.4 How many demonstrations are needed?

In order to find the relevant dimensions the demonstrations has to be similar in the relevant dimensions and different in the irrelevant dimensions. We can see in figure 5.12 that indeed the important aspect of the demonstrations is if they contain enough information for the robot to determine what the relevant dimensions are. This means

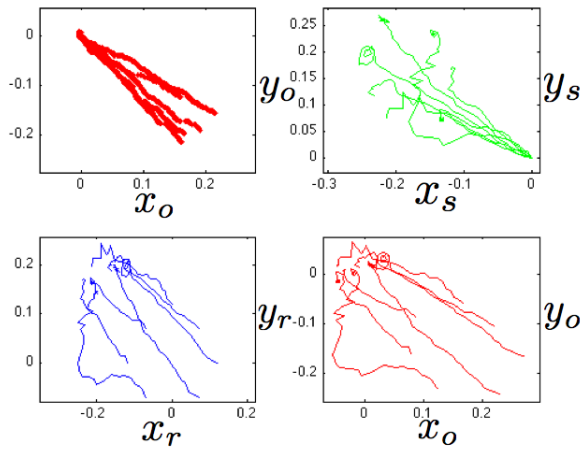

Fig. 5.7: This figure shows demonstrations of task 1 (move hand to object) in the framing relative to the object to the top right and the reproductions in the framing relative to the starting position to the top right, where we can see that the reproductions look very different due to the different starting positions. To the bottom left we see that despite starting at different locations the hand moves to the top left corner (the object is always somewhere in the top left corner). Finally we can see to the bottom right the reproductions in the correct hand-object framing. There are some odd behavior at times but in general the task is achieved even when starting outside the area that the demonstrations started within.

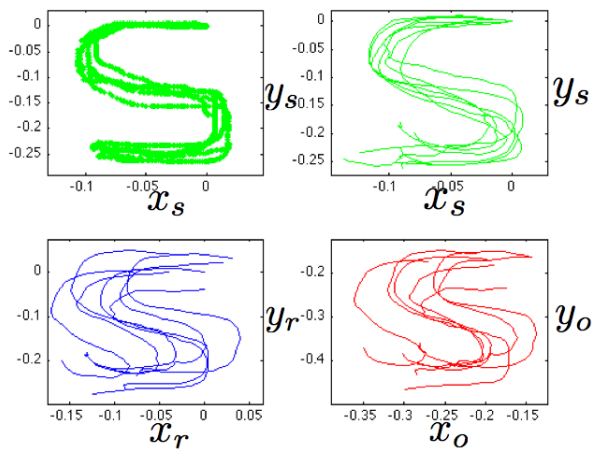

Fig. 5.8: This figure shows demonstrations (top left) and reproductions of task 2 (draw an S shape). We can see that, as we should expect, the reproductions look similar to the demonstrations in the framing of hand relative to starting position (top right). A few of the reproductions does not completely replicate the task in the last turn but overall the reproductions are similar to the demonstrations

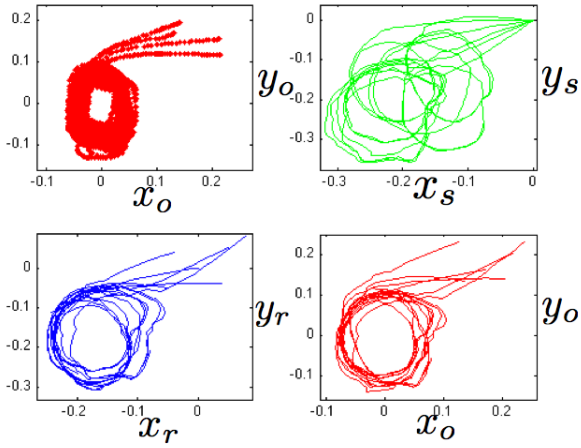

Fig. 5.9: This figure shows the demonstrations and reproductions of task 3 (move hand in circle around object).

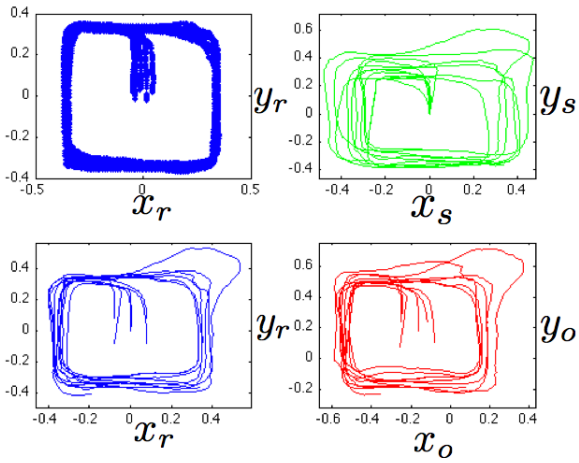

Fig. 5.10: This figure shows demonstrations (top left) and reproductions of task 4 (make a big cyclic square movement). The task is largely reproduced as demonstrated.

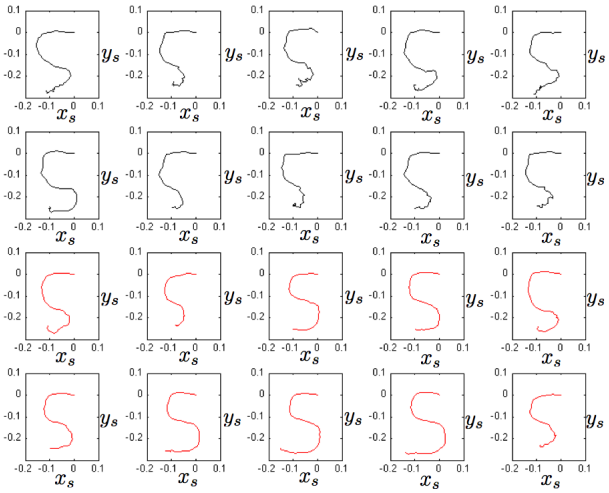

Fig. 5.11: This figure shows at the top two rows in black 10 reproductions made while picking only one set of local points using distance in the full state space. The bottom two rows shows the results using framings. we can see that on average the S shape is better in the bottom two rows, especially the later half of the movement (the second "turn").

that the amount of demonstration needed is such that for each incorrect framing f there is at least one pair of demonstrations which are different from each other when viewed in f. This requirement can be alleviated if the agent is given the correct framing or have well calibrated prior probabilities from learning other tasks. If a human is demonstrated a task where the demonstrator moves his hand around a coffee mug, the subject is unlikely to assume that the task consists of moving the hand in a circle 50 centimeters to the right of the computer (if the robot is to be able to learn this autonomously it will need to extract relevant rules from previously learned tasks; unless this is done the coffee mug will have to be moved and another demonstration made so that the agent sees that the demonstrations look the same if focusing on the relative position of the hand and the coffee mug but not if focusing on the relative positions of the hand and the computer)

5.2.5 Comparison with GMR

The ILO-GMR was used instead of the standard GMR algorithm as it was argued that GMR is not suited for the multi task, multi framing setup explored. It makes sense to complement these arguments with experiments using GMR on the same data as the ILO-GMR experiments were conducted on. The experiment is performed by simply fitting a GMM to the full data set of all demonstrations, and then generating reproductions in the same manner as in the experiments above (with the only difference that, at each iteration, the pre computed GMM is consulted instead of the local GMMs). The issue of finding the appropriate number of gaussians was solved by gradually increasing the

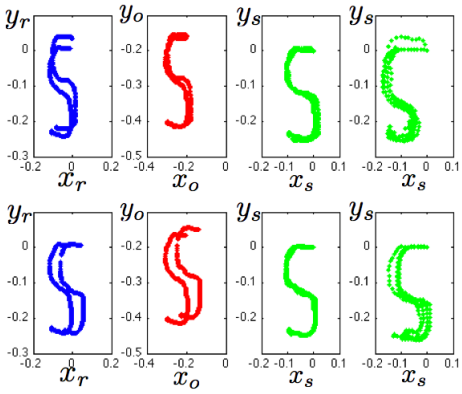

Fig. 5.12: This figure shows at the top row demonstrations nr 1 and 5 in first framing 2 (blue), then framing 3 (red), then framing 1 (green) and finally 5 reproduction attempts in framing 1 (the correct one for task 2) of an agent able to see all demonstrations of the other tasks and demonstrations 1 and 5 of task 2. The bottom row is exactly the same but with demonstrations 3 and 5 instead. The demonstrations at the top look the same in two different framings and as a result the agent will not know what framing is the correct one and the reproduction attempts suffer as a result. The demonstrations at the bottom are different in the two incorrect framings but similar in the correct one giving the agent a chance to find the correct framing and as a result these reproductions (bottom right) is superior to the other reproductions (top right).

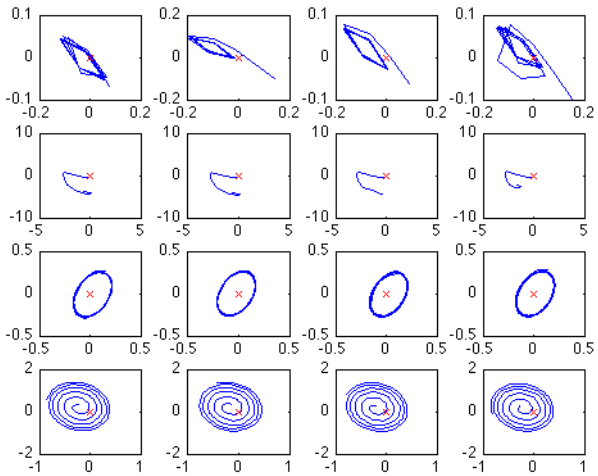

Fig. 5.13: Displaying results in the same way as in figure 5.6, we can see that tasks 1 and 3 are well reproduced, but that tasks 2 and 4 are not.

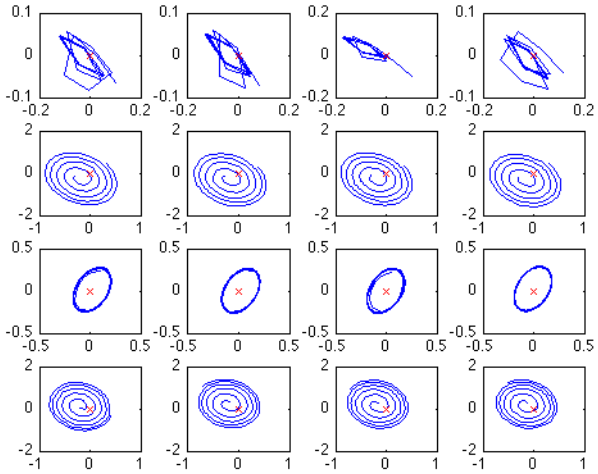

Fig. 5.14: Displaying results in the same way as in figure 5.6, we can see that task 2 is now identical to task 4, meaning a degradation in performance (since the reproduction still held some resemblance to the demonstrations with 7 gaussians as seen in figure 5.13).

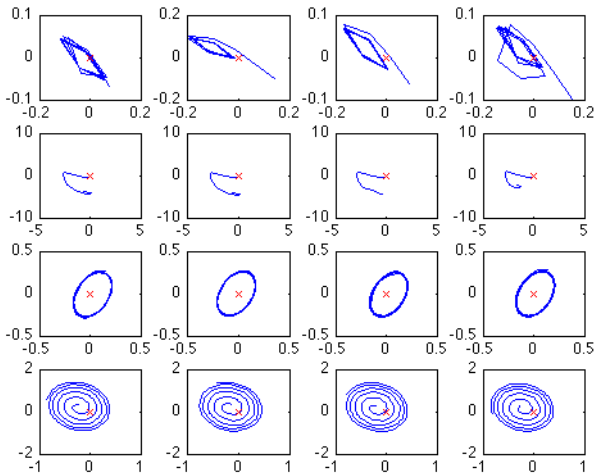

Fig. 5.15: Displaying results in the same way as in figure 5.6, we can see that Using 30 gaussians does not lead to any new behavior, compared with that exhibited in figure 5.13.

number until reproduction behavior stopped changing.

In figure 5.13 we can see the performance of GMR (using 7 gaussians) on the same data as the ILO-GMR algorithm was evaluated on above. Tasks 1 and 3 are reproduced well, but tasks 2 and 4 are not. In figure 5.14 we can see an additional error from the reduction in gaussians, and in 5.15 we can see that the behavior remains identical with 30 gaussians (the algorithm returns several gaussians with priors 0, and several others with very low priors). The two tasks that are reproduced properly are both defined in the coordinate system relative to the object. The starting position varies greatly between tasks in this coordinate system (since the object position is greatly varied), and the location of the demonstrations of tasks 1 and 3 does not overlap in this coordinate system. Task 2 does resemble the "S" shape to some degree. It starts by moving to the left, then down and then to the right. The dimensions are however completely wrong as can be seen if inspecting the axis (the GMR imitator moves many times further to the left than the demonstrator did during demonstrations), and instead of turning down, and then left again, it turns up and stops. The spiral of task 4 also resemble the demonstrations to some degree. It is moving counter clockwise in an expanding circle instead of moving counter clockwise in a fixed size rectangle.

5.3 Discussion

We have shown that the ILO-GMR approach allows a robot to learn to reproduce several different context-dependant tasks at the same time and incrementally without the need to change any parameters as new tasks are added, and without categorical information associated to each demonstration.

Chapter 6

Bootstrapping language acquisition as imitation learning of sensorimotor tasks in multimodal contexts

Contents

6.1 Introduction **99**
6.2 The motor gavagai problem **101**
6.3 Learning situation **104**
6.3.1 Data capture and representation 107
6.4 Algorithms **108**
6.4.1 Representation of the tasks 108
6.4.2 Learning algorithm 108
6.4.3 Reproduction algorithm 112
6.5 Experiments **115**
6.5.1 Experiment 1: Extending the context to include speech . . . 115
6.5.2 Experiment 2: Relaxing the assumption of a single channel of communication 120
6.5.3 Experiment 3: Extending the types of word meanings that can be understood and learning simple word order syntax . . 128
6.5.4 Further investigation of the grouping algorithm 133
6.6 Overall discussion **137**

In this chapter which is largely based on the article [103], we explore how simple forms of language can be learnt from unlabeled data, concurrently with learning non linguistic tasks. The learner/imitator observes several interactions between two humans. There is one interactant that might perform some form of communicative behavior, and one demonstrator that shows what the appropriate response is to the current context (where the behavior of the interactant is treated as part of the context). The inputs and outputs

are all continuous and the interactions are not labeled, meaning that the imitator does not know how many words are expressed in a given set of interactions (it could be two words spoken many times each, or several words spoken a few times each, and some interactions might contain no relevant linguistic information at all). A single imitation learning strategy is proposed for dealing with all tasks in this context, that does not need to be told which demonstrations are of linguistic tasks. Using a single imitation learning strategy for both non linguistic and linguistic tasks, that does not rely on labels or symbolic input, allows us to avoid/disolve the "symbol grounding problem" (there is no symbolic language learning system whose symbols need to be grounded in a separate action learning system). To give an evolutionary account for language, it would now only be necessary to show that this general imitation learning strategy was useful for an individual *even in a non linguistic environment*, and finally that this strategy can learn language (the latter being the goal of this chapter). The group benefit that a tribe of linguistic individuals derive from an already established set of linguistic conventions is not needed as an evolutionary force (the group selection hypothesis is not disproved, but an alternate account does show that we have no need for it).

6.1 Introduction

As shown by a growing number of experimental results and theories, language acquisition is a process that is strongly interacting with and grounded in action and perception [1][2][3][4]. This point has also been explored within the robotics and cognitive modeling communities, see for example [5] and [7] for an overview. Many researchers have put forward the idea that language and action develop in parallel, feeding each other through exploration and interaction with the physical and social environment. In this chapter, through the presentation and experimental validation of computational models, we propose to go one step further by considering that language and general action-perception learning shall not be regarded as two interacting but separate processes, but rather that there could be a single general process.

The proposal is put forward to model language acquisition as a single instantiation of a general imitation learning system, using the same strategy for handling linguistic skills and non linguistic sensorimotor skills. The system starts from standard principles of imitation learning systems [67][87][64] that observe a demonstrator performing non discrete actions as a response to a non discrete context. These classical approaches to action learning by imitation are then extended in two directions. First, the context is extended to include the communicative acts of another human, called an interactant, who is part of the context in the same way as a physical object or any other aspect of the environment. Second, instead of the traditional setting where only one task is to be learnt[67][87], the imitator learns several tasks without any external and/or symbolic information provided about how many tasks there are or about how the demonstrations should be segmented into groups. Some of the tasks are best described as verb learning (the interactant requests a particular type of hand movement as a response to an acoustic

wave or a gesture he produces), but the types of actions that can be learnt also include communicative responses (when an object is to the imitators left/right, it draws an "L"/"R" shape), and yet others are best described as internal cognitive operations.

The behavior of the demonstrator can be influenced by communicative acts of the interactant in the same way as it can be influenced by for example the position of an object. In other words, there is no need for one action system and another separate linguistic system, and therefore also no need to find out how they can interact with each other. Since the thesis is that a single sub-symbolic system or strategy can learn how to respond to both non linguistic and to linguistic contexts, the problem of grounding the symbols of a linguistic system simply does not arise[1]. How to appropriately respond to a physical situation, such as the property or position of an object, is learnt in the same way as how to appropriately respond to communicative acts.

In the experiments presented in this chapter, only single words, or simple two-word word order syntax is learnt, which is not as advanced as in many other models of language learning, but on the other hand it does have the advantage of not suffering from a symbol grounding problem. We shall argue that the context could, in principle, include very complex internal cognitive structures and that the action space could in principle include communicative responses and/or complex operations on the internal structures. This indicates that the type of language that can in principle be learnt is much broader than what is presented here. The actions that are appropriate responses to a linguistic context could also themselves be linguistic. There are of course practical issues, and it is of course possible that the approach will not be the most practical way of achieving certain types of advanced linguistic behavior, but the symbol grounding problem have proven difficult to deal with, suggesting that an approach that does not need to deal with it is valuable. Instead of using discrete (or symbolic) inputs and outputs, the context (including the communicative acts and behavior of the interactant), the demonstrators actions and the imitators actions are all continuous. This means that the imitator does not know what part of the multi modal context might be "words", or how many different words it is observing, or how many different types of demonstrated behaviors it has seen. Indeed, each instance of the same behavior or communicative act is different: the imitator will have to infer from the data which specific communicative acts - speech waves or hand gestures - are instances of the same word and which specific demonstrator actions are instances of the same behavior. In spite of these ambiguities, we will show that some general imitation learning strategies can allow the system to learn the invariants and the appropriate skills.

To exemplify the approach, three experiments are presented where a robotic imitator observes interactions between a human demonstrator and a human interactant (some of them being linguistic), builds a model of how to act based on how the demonstrator

[1]Instead of proposing a new way of achieving this grounding, the proposed system never has the need to ground discrete symbols of a separate system since there are no symbols and no separate systems. Avoiding this need is a property stemming from the fundamental aspects of the architecture in combination with the learning setup.

acts in response to the interactant, and finally itself responds directly to the interactant (and the rest of the context), attempting to imitate the demonstrator:

- The first experiment illustrates the concept with simulatenous imitation learning of skills consisting in achieving an action in response to a speech act and of skills where no form of communication is involved (learning how to act in a traditional context). The context is extended to include a continuous speech input of an interactant, and the demonstrator's behavior is sometimes influenced by a word (i.e. an acoustic wave) that the interactant pronounced.
- The second experiment introduces hand signs in addition to speech, and exemplifies how the system can deal with two possible channels of communication. Now the demonstrator might be responding to a hand sign or a speech utterance from the interactant or to some aspect of the traditional context (e.g. position of an object). The demonstrator also teaches the imitator to perform something that might look like a communicative act to an outside observer (as opposed to only respond to communicative acts of an interactant), specifically: describing the position of an object with a gesture.
- The third experiment extends the type of words that can be learnt to include more than simple action commands. Some words of a simple sign language requests a specific attentional focus on an object (i e. an internal cognitive operation on the attentional system) and other words request the imitator to perform a certain type of action. The imitator learns the words requesting the internal operation of object focus by inferring what unseen cognitive operation the demonstrator performed (which is possible as the state of the internal attention state has a consistent effect on behavior as actions are performed relative to the object focused on).

6.2 The motor gavagai problem

To illustrate the similarity of the problems faced by an imitator adopting normative rules for how to respond to some non communicative context on the one hand and the problem faced by a language learner on the other hand, we will look at the well known gavagai problem [22] from linguistics. Quine gave an example with a man pointing at a rabbit and saying "gavagai" to someone that does not speak the language. The "gavagai problem" is to find out what the word means from this type of interactions (it could mean rabbit but might also mean "dinner" or "look, my pet has escaped"). A similar type of ambiguity can be said to exist when interpreting physical actions. In the words of [63]: "the exact same physical movement may be seen as giving an object, sharing it, loaning it, moving it, getting rid of it, returning it, trading it, selling it, and on and on depending on the goals and intentions of the actor."

Imagine a demonstrator, that we will call Steve, trying to teach you something new. Steve and you are both looking at a rabbit just in front of the trees of a forest. Steve

takes a stone just on his right, and throws it towards the rabbit with a parabolic-shaped trajectory. The stone arrives one meter to the left of the rabbit, just below a tree with blue flowers. Now Steve asks you to try to reproduce what he did. In the meantime, the rabbit moved 10 meters away, a cat has arrived next to the tree, and there are no more stones on your right or on the right of Steve, but a stone on your left and a knife on your right. What should you do? Would you take the stone or the knife? Would you throw it in any direction, trying to reproduce a parabolic trajectory? Or would you throw it towards the left of the rabbit? Or onto the rabbit? Or to the (left of the) cat? Or try to throw it just below a tree with blue flowers? Did Steve intend to throw something at the rabbit but missed it? Or at the closest animal? Did he use the stone just as a way to bring your attention to this beautiful rabbit, using the stone as a pointing gesture? Did he intend to kill the animal? In that case, maybe instead of throwing something, it is more efficient to reproduce what Steve tried to do by first catching the rabbit by hand? Did Steve want to just frighten the rabbit? In this case, maybe you can reproduce what he did by shouting very loudly? Or maybe Steve wanted to show you that the action "throwing a stone with a parabolic trajectory" is an arbitrary action that should be associated/triggered with/by the concept/observation of "rabbit" (and maybe the next demonstration will show you that the concept of a "cat" should be associated with the action "throwing a stone with a straight trajectory[2]"), and you should reproduce it whenever you see a rabbit and want to convey this information to someone else?

Through this very simple situation, we can easily recognize a variation of the above mentioned gavagai problem used by Quine [22] as an illustration of one of the fundamental issues posed by language acquisition. The main difference with the original scenario is that here Steve throws a stone instead of pointing towards the rabbit and pronoucing the "Gavagai" word. But it is not difficult to see that this difference between what we may call the "motor Gavagai problem" and the "language Gavagai problem" is not a fundamental difference, since pronouncing an acoustic wave with peculiar properties and pointing in a certain way at the same time is not very different from launching a stone in a particular way towards a particular direction (even though grown up linguistic humans will have different intuitions about these scenarios). Actually, several of the potential interpretations of the "stone parabolic throwing action in the context of a rabbit being present" demonstration are exactly the same as those of the "pronouncing the word Gavagai in the context of a rabbit being present": the "stone parabolic throwing" could here be described by an external observer as an arbitrary linguistic (visual/gestural) "word" that is associated to the meaning "rabbit". As in the case of the more classical language Gavagai problem, the only way to learn the right interpretation/reproduction/generalization of the demonstration/naming is to use constraints

[2]If this is what Steve intended it also relates to the question of finding an appropriate framing. Is the task to throw the stone at the same spot but with a faster starting speed and a lower angle than before, or is it to throw it in a more straight trajectory (Steve may not have considered this explicitly, and his ability to judge a failure from a success may not give a definite answer under non exotic circumstances).

(which do not need to be specific to language, as we will illustrate in the experiments below) putting assymetric priors on the probabilities of different hypotheses, and of course to engage in further interactions for refining statistical inference, collecting new demonstrations and obtaining various forms of feedback from reproductions. It does not seem like there is a need for two different systems for dealing with these two problems.

As we pointed out, most of the techniques discussed above make no assumptions regarding the semantics of the output and contextual input. They can very well be used to address the motor Gavagai example with Steve throwing a stone as described above, and are theoretically powerful enough to be able to learn non-trivial interpretations of Steve's demonstration such as "The skill consists in making the rabbit run away whatever the means", "The skill consists in hitting the closest animal", or even "The skill consists in throwing a stone with a parabolic trajectory in whatever direction when you see a rabbit", as long as the hypothesis space provided to the robot includes the corresponding relevant dimensions/variables and powerful enough statistical inference methods are used to identify the right framing/compact projection of the full context space. These techniques can be used if Steve produces the "Gavagai" acoustic wave and possibly points at the same time towards the rabbit, without adding any language specific additional assumptions (since they do not make assumptions regarding inputs).

From this, it can be seen that the link to language learning bootstraping may be achieved by using these architectures to teach a robot skills where arbitrary gestures produced by an interactant should trigger particular context-dependant actions (thus we would have here something like a signed language), where the context can be the combination of the gesture with some other aspects of the scene (including the inferred mental state of the interactant). In such a system the meaning of gestures/words consists in a dynamic internal processing, itself generating context-dependant behavior, and being a sub-symbolic equivalent to more symbolic approaches of "meaning as dynamical programs" [38][58][55]. Then, to get to spoken language, it appears that one only needs to include sensorimotor dimensions related to the perception and production of acoustic waves (in a non specific way that does not make them a priori qualitatively different from other dimensions). In [56] and [57] experiments are presented modeling how the hypothesis space for a word like "gavagai" is reduced due to the way an adult structures an interaction with an infant. Experiments show that infant directed speech does indeed provide redundant information and added structure via synchrony (compared to adult directed interaction).

Through three experiments, a technical proof-of-concept is provided of this general view regarding the integration of action and language learning. This is achieved by showing experimentally that the general techniques for context-dependant sensorimotor learning can be used to learn seamlessly, and simultaneously, traditional motor skills as well as (what an external observer might call) "socially interactive linguistic skills" in a system that has no components such that we need to use the word "symbol" to describe the system.

The first experiment will extend the traditional context of imitation learning exper-

iments to include speech input from an interactant. To solve this problem we will start from the approaches which try to model the context-dependant skill directly as a context-dependant dynamical system, more precisely the work done with Gaussian Mixture Regression in [68][87][101]. The problem of finding which demonstrator trajectories are instances of the same movement is first solved and then a local version of the GMR regression method, called ILO-GMR, is used to reproduce the movement during reproduction [104, 102].

In the second experiment the imitator is presented with a context that includes speech, hand signs and an object position. Each task is triggered when an initially unknown "command" is expressed using one of these channels.

Finally, in the third experiment, the imitator will learn more advanced types of words, specifically where the meaning consists of requests to perform a "focus on object" internal cognitive operation. To imitate this internal operation the imitator must infer what internal operations was performed by the demonstrator.

Then we discuss how the method also naturally extends to involve the rules of social interaction themselves as "when an interactant looks at something; focus attention on it", something that is very little explored (but see [84] and [83]). Other such examples are learning how to respond to facial expressions, body language and tone of voice (and anything else that is not easily transformed by a pre-processing step into symbolic input[3]). It also naturally extends to compositional tasks; a demonstration could be described as executing other tasks in sequence or the context of one task could be a combination of: the task(s) recently executed by the imitator, by an interactant, the linguistic input, and other more conventional things such as the position of an object.

6.3 Learning situation

In the following, we will study an experimental setup involving a simulated robot, a human demonstrator, and a human interactant (see figure 6.1). The robot's goal is to learn a repertoire of tasks by observing task demonstrations from the human demonstrator. The robot consists here of an arm, and tasks consists in producing movements of the hand which dynamically depend on the context, and are represented as closed loop policies. The context consists here of the position of a physical object on a table (represented in several -absolute and egocentric- coordinate systems called framings) as well as speech sounds and movements of the hand produced by the human interactant. The human demonstrator demonstrates a task to the robot by kinesthetically moving its hand: it takes the hand of the robot, and shows it what movement should be achieved

[3]Researchers that are determined to handle such things using a pre processor into a symbolic representation can take the work presented here as suggesting one way of building this pre processor. The reaction of the demonstrator to events in a continuous space can be used to determine how this space should be segmented into discrete regions (i.e it can be used to determine the boundaries between regions, where each region in the continuous facial expression space results in some specific symbol).

depending on the current object position as well as on the speech or gestures produced by the human interactant. Examples of tasks will be of the type: draw a square around the object when the interactant produces the acoustic wave "square"; touch the object when it is on the top left of the table; draw a "R" when the object is on the right; draw a circle in the centre of the table when the interactant produces a gesture of a certain shape; draw a square around the green object when the interactant produces the acoustic wave "square green", etc. The division of teacher behavior into separate distinct tasks is an approximation, that corresponds to a learner modeling the teacher as doing "several different things" or "achieving several different goals". This is a standard approximation in learning from demonstration, but it is usually not made explicit due to the learner only learning a single task at a time. The approximation is accurate to the degree by which the teacher considers itself to be performing distinct tasks, and it is useful to the degree that it allows the learner to find a policy that replicates the teachers behavior in novel situations.

The number of tasks to be learnt, as well as the association of each demonstration with a task category, is not provided to the robot learner. Indeed, neither the human demonstrator or the human interactant provide symbolic labels: which tasks should be achieved depends on the continuous context, and the robot has to learn these invariances. What the robot perceives in a demonstration is only continuous stimuli: hand movements are sequences of two-dimensional position on the table (in several framings), the direction the hand was moved in from that position, speech words produced by the interactant are raw trajectories of MFCC features projected in a low-dimensional continuous space, gestures of the interactant are raw trajectories of its hand also projected in a low-dimensional continuous space. Thus, from the point of view of the learner, the object position and the interactant speech and gestures are equally considered as features of the multimodal context. As we will see, speech and gestures of the interactant will sometimes appear linguistic from the point of view of an external observer, and while the learner does not have a specific treatment for "linguistic signals" (he does not even know which channel are for what the external observer would call "communication"), it will learn to respond to the interactant linguistic signal in a sensible way.

The imitator robot is here simulated and is able to move its hand on a 2D plane (for easy visualization) in any direction it wants which, if a physical robot is to be used, would require an inverse kinematics model that translates the current state and desired hand directions to motor outputs. The simulated imitator was more flexible to perform experiments with and since the focus of the presented experiment is about learning what should be done rather than how to do things, it was used in place of a physical robot (in the language of [95], the "what to imitate" instead of the "how to imitate" question is the focus of the presented experiment). There are obviously limits to what types of behaviors a robot can learn to do in simulation before this starts to become a serious simplification, and if more advanced physical manipulations are to be investigated, a physical robot will have to be used. Yet, here the demonstrations are provided by a real human demonstrator (using a mouse to drive the simulated robot hand), and real speech waves and hand gestures are also recorded from a human.

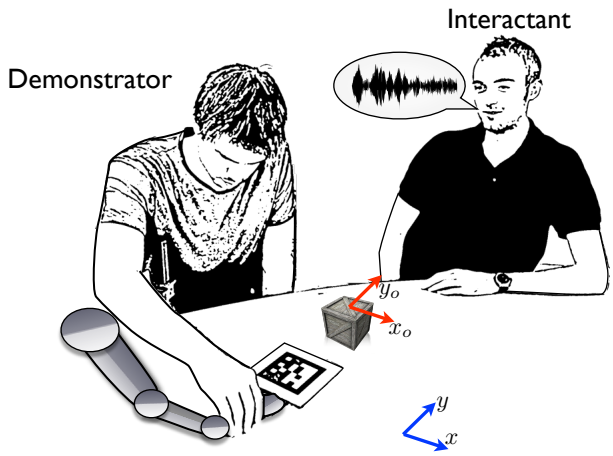

Fig. 6.1: The learning situation investigated. A human interactant (to the right) speaks or makes a gestures, and the demonstrator (to the left) moves the hand of the robot to show him which movement to do. This movement can depend on either/both the current object position, the interactant speech and/or the interactant gestures. Some tasks are non-linguistic, such as "touch the object when it is on the top left of the table". Some tasks are motor responses to a linguistic signal from the interactant, such as "draw a square around the object when the interactant produces the acoustic wave square". Some tasks are quasi-linguistic responses to linguistic signals, such as "draw a "w" when the interactant gestures a specific sign". The demonstrator and interactant provide no symbols to the robot learner, which perceives each demonstration and context through continuous low-level sensorimotor values. Moreover, different tasks are movements defined in various coordinate systems (e.g. absolute versus object centric vs hand centric coordinate system) called framings. The robot has to learn how many tasks there are, which is the control policy for each task and in what appropriate framing it is defined, as well as to learn which tasks should be triggered depending on the current context.

6.3.1 Data capture and representation

Physical objects are represented by its absolute 2D position on the table. Hand movements of the demonstrated tasks are represented as sequences of 2D positions and directions in several coordinates systems (for example: absolute, object relative, relative to starting position). The acoustic speech waves produced by the interactant, as well has his hand gestures, are projected onto a lower-dimensional (and fixed dimensional) space so as to facilitate the statistical inference process described below.

Hand trajectories

The demonstrator uses a mouse to generate the hand trajectories of the demonstrations. At each time step the position (one for each framing/coordinate system) and the hand direction is stored. All of these pairs constitute a hand trajectory. The direction angle of the hand at a given point of the observed or reproduced trajectory is encoded using two dimensions. The directional angle is measured in its rescaled x and y components (under the constraint that $x^2 + y^2 = 1$) which resolves difficulties with linear regression over a periodic variable (the fact that the angles $\theta = -\pi$ and $\theta = \pi$ are identical outputs makes the raw θ unsuitable for the ILO-GMR linear regression method). Then the amplitude of local displacement are computed as averaged over the 7 nearest (in time) data points since the raw captured data is not of very good quality (sometimes smooth mouse movements will results in strange angles of the type; $p_{t=1} = \pi/2, p_{t=2} = 0, p_{t=3} = \pi/2, p_{t=4} = \pi/2, p_{t=5} = 0, ...$ and this type of data causes problems, for example in the policy similarity comparisons of the grouping algorithm).

Interactant speech and sign language

Actual speech recording of a single speaker pronouncing the relevant word was used (recorded in an ordinary office environment without anyone talking in the background). To transform speech into a low-dimensional vector, here with 3 dimensions, an utterance was first encoded using Mel-Frequency Cepstral Coefficients (MFCCs). Each such high-dimensional MFCC trajectory is then compared, using a dynamic time warping similarity measure, to three fixed acoustic wave prototypes which are different enough for the vector of these similarity measures to result in a 3D projection which keeps enough differentiation between the different words produced by the interactant. A way to choose these prototypes is by conducting a K-means clustering of ambient utterances (and the dimensionality K=3 was found to be efficient enough for the proof-of-concept experiments we study below).

The hand sign trajectories was also compared to three prototypes to generate a 3D point (also here using a dynamic time warping similarity measure).

For each demonstration and each reproduction of a task where speech was irrelevant, a

unique random point in the 3D speech space was generated. The same was done with hand sign space.

6.4 Algorithms

The algorithm is divided into a learning algorithm doing off line analysis of the data, and a reproduction algorithm that uses the results of this analysis for data driven on line regression. During reproduction, the imitator is alone with the interactant and the on line reproduction algorithm uses the estimates produced by the learning algorithm to generate behaviour.

6.4.1 Representation of the tasks

The tasks to be learnt by the robot are closed-loop control policies generating a movement that depends dynamically on the context (as opposed to open-loop movement). Hence, learnt tasks are represented by a mapping $(posH_f, posO, S, G) \rightarrow speedH$ where pos_f is the current position of the hand in framing f, $posO$ is the current position of the object, S is the low-dimensional representation of the acoustic wave perceived by the robot, G is the low-dimensional representation of potential gestures perceived by the robot, and $speedH$ is the direction that the hand should currently have. In what follows, this mapping will be computed dynamically and online based on adequately selected examplars of demonstrations and based on the current context.

6.4.2 Learning algorithm

The learning algorithm takes the demonstrated hand trajectories and the contexts (linguistic and non linguistic context) as input, and create estimates of what demonstrations are instances of the same task, and what the correct framing is for each such task, that are later used by the reproduction algorithm. Figure 6.2 shows an overview where we can see the inputs and outputs of the different algorithm components. Each of the three algorithmic steps are explained below. Further details of the grouping algorithm used in experiment 3 are given in appendix **??**.

Similarity estimation

The similarity estimate uses the assumption that each demonstration is of a single task and there is one correct framing, valid during the entire demonstration. This assumption, which we think is reasonable, allows us to considerably improve the system if leveraged appropriately. Of course, several demonstrations might correspond to the same task and the same framing, and different tasks might have the same framing

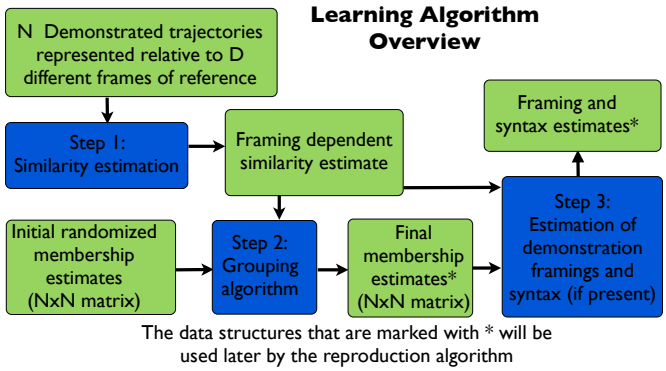

Fig. 6.2: Blue boxes show algorithms and green boxes show data structures. The demonstrated hand trajectories are analyzed in batch mode off line. The membership estimates created in step 2 are used later by the on line reproduction algorithm. Step 3 always creates estimates relating to framing but the details vary between experiments. In experiments 1 and 2, each movement type has one correct framing, and in those experiments, this is what is estimated. In experiment 3, the framing is requested by the interactant, and what is estimated in step 3 of experiment 3 is the syntax of the interactants sign language. This syntax will help the imitator determine what framing to adopt during reproduction (specifically, the estimated syntax tells the imitator which hand sign to look at for the purposes of determining framing).

and vice-versa, but we exclude here demonstrations which are for example unsegmented sequences of motor primitives/tasks. A second assumption that we make in the following (but this is less crucial and might be possible to remove by some modifications of the algorithm not presented here) is that there is a finite number of pre-given available framings.

The goal of the similarity estimation is to assign high similarity to two trajectories where the same hand states (hand position) result in the same hand movements. Since the closeness of inputs is dependent on framing, the similarity is also dependent on framing.

The similarity is dependent on framing assumptions which are treated different in the three experiments. In experiments 1 and 2, each movement has exactly one correct framing, but in experiment 3, the framing is requested by the interactant.

In each case, a number of points are selected from demonstration m and for each of these points the closest point in demonstration n is selected (the distance is measured relative to an hypothesized framing, making the set of closest points dependent on framing assumptions). Then we have $\Delta_{mn} = \sum \delta_i^2$, where δ_i is the difference in output between point i of demonstration m and the point of demonstration n closest to it. The similarity is then the inverse of the distance. Each framing assumption thus give a different similarity between two trajectories. Given D possible framings, experiment 1 and 2 have D number of framing assumptions, while experiment 3 has DxD number of framing assumptions.

Trajectory distance $\Delta_{t;k;i;j}$

To determine which trajectories are instances of the same movement it is necessary to define some measure of distance between two trajectories. In each demonstration the three objects and the starting position is different. For each movement type, the policy of the demonstrator is determined by the hand position in the coordinate system centered on the object focused on. What object is focused on is not observable to the imitator, but two trajectories that are instances of the "circle" movement will only look similar if each is viewed in the coordinate system of the object that is focused on (the object encircled). For this reason the distance between two trajectories is defined relative to two coordinate systems; one used for trajectory 1 and the other one used for trajectory 2. Thus $\Delta_{A;B;1;3}$ is the distance between trajectory A seen in coordinate system 1 and trajectory B seen in coordinate system 3. If trajectory A is a circle around $0, 0$ in coordinate system 1 and trajectory B is a circle around $0, 0$ in coordinate system 3 they will probably look similar (trajectory A viewed in another coordinate system would still be a circle, but not around $0, 0$, and the two trajectories will not look similar).

For each of the N points in trajectory number t the closest points in trajectory k is selected (with distance measured using the respective coordinate systems). For each point p of trajectory number t, the closest point of trajectory number k is found, using the positions in the coordinate systems i and j respectively[4]. δ_p is defined as the angular difference in output of the two points. Then we have $D_{t;k;i;j} = \sum_{p=1}^{N} \delta_p^2/N$. Finally we have $\Delta_{t;k;i;j} = min(D_{t;k;i;j}, D_{t;k;i;j})$.

There are many possible ways of measuring similarity between two trajectories, given the coordinate systems to view them in and no claims of the optimality of the specific similarity measure introduced is made. Like many other parts of the algorithm the important part is not how the specific part is implemented but instead how it is combined with the rest of the algorithm, with the details included only for completeness.

Grouping algorithm

The grouping algorithm takes the estimated similarities as inputs and outputs an estimated set of groups (for N trajectories, this is an NxN matrix P where p_{it} is the probability that trajectory number i is an instance of task group number t).

There is a lot of information available and many reasonable biases that can inform the sorting of demonstrations into groups that are supposed to contain demonstrations of the same task. The main assumption used in all grouping algorithms is that trajectories with high similarity is more likely to be instances of the same movement. The details of the grouping algorithm used in experiment 3 can bee seen in appendix **??**, with experiments 1 and 2 using similar, and earlier versions of the same algorithm.

[4]So that if i=1, j=3, and point number p's position in coordinate system 1 is (0,0.4) then the point of trajectory k that is closest to (0,0.4) in coordinate system 3 is chosen (so that a point above the red object in trajectory t is compared to a point above the blue object in trajectory k).

The grouping algorithm is not a major contribution, no claims of optimality is made, and it is chosen in place of standard clustering algorithms due to its suitability for task specific extensions. The details of the algorithm is presented for the sake of completeness.

The grouping algorithm is a batch computation and, if not modified, would have to be re run for every new demonstration observed. It is however well suited for an incremental version (with current data it takes only a few seconds on a modern laptop but with larger number of tasks, demonstrations and number of possible framings, time could become a problem). When the algorithm has grouped all the observed demonstrations and found the corresponding framings, it can use this information when new demonstrations are added. If a new demonstration is similar to one of the established groups, when viewed in that groups preferred framing, then it can simply be added. Otherwise the membership values already found can be re used so that what is left to determine how to group the new demonstrations.

Technical details:

The current estimate of the probability that trajectory number t is an instance of movement number m is denoted $m_{m;t}$. The suitable value of $m_{m;t}$ is completely determined by what movements the other trajectories are estimated to be instances of. The only thing that matters is that trajectories that are instances of the same movement are grouped together. Since the number of movements is unknown there are as many movements as trajectories (so that M is a NxN matrix for N demonstrations).

Given the similarity between trajectories there are many possible ways to divide them into subgroups and the iterative algorithm proposed is not claimed to be optimal (the reader that is not interested in exactly how similarities between trajectories is used to form groups whose members have high similarity can skip this section). The basic principle of the grouping algorithm is that if two trajectories A and C are more similar to each other than other trajectories likely to be instances of movement x, then $m_{x;A}$ and $m_{x;C}$ will increase. If A and C are less similar than average, then $m_{x;A}$ and $m_{x;C}$ will decrease, and the magnitude of the change depends oh how much the similarity deviates from the other likely members.

The algorithm is described using pseudocode in 5. In order to save space, several variables (either used in the pseudocode or used to define other variables that are used in the in the pseudocode) are defined and explained below rather than in the pseudocode, such as: maximum trajectory similarity $\gamma_{t;k}$, joint memberships: $\omega_{t;k}$, weighted mean similarity ϖ_t and push strength $\xi_{t;k}$.
Maximum trajectory similarity $\gamma_{t;k}$. $\gamma_{t;k;i;j}$ is the inverse of the distance $\Delta_{t;k;i;j}$ and $\gamma_{t;k}$ is the maximum similarity between trajectories t and k, $\gamma_{t;k} = max_{i;j}(\gamma_{t;k;i;j})$ (for example, if trajectories A and C have the highest similarity when A is in coordinate system 1 and C is in coordinate system 2, $\gamma_{A;C} = \gamma_{A;C;1;2}$, which is likely to be the case if trajectory A is a circle around the red object and trajectory C is a circle around the green object).
Joint memberships $\omega_{t;k}$ is a measure of how probable it is that trajectories t and k are

instances of the same movement according to the current state of the membership matrix M. It is calculated as: $\omega_{t;k} = (max_m(m_{m;t} * m_{m;k}))/(\sum_{\tau=1}^{N} max_{m;\tau}(m_{m;t} * m_{m;\tau}))$.
Weighted mean similarity ϖ_t is a measure of the weighted average similarity to trajectory t of trajectories that are likely to be instances of the same movement. $\varpi_t = \sum_{k=1}^{N} \omega_{t;k} * \gamma_{t;k}$.
Push strength $\xi_{t;k}$ is the strength with which trajectory t will affect the memberships of trajectory k in the movement groups that they are both probable members of. If it is positive the presence of trajectory k in a movement group will increase the membership of trajectory t and decrease it if it is negative. It is calculated as: $\xi_{t;k} = e^{((\gamma_{t;k}/\varpi_t)-1)}$, and we can for example see that $\xi_{t;k} = 1$ if the similarity between t and k is exactly the same as the average weighted similarity between t and the other trajectories that has high joint memberships with t. If the similarity $\gamma_{t;k}$ is bigger than the weighted average ϖ_t, the we will get a push strength $\xi_{t;k} > 1$ (and if the similarity $\gamma_{t;k}$ is smaller than the weighted average ϖ_t, we will get $\xi_{t;k} < 1$).

Estimation of framings and syntax

In experiment 1 and 2, each group created is assigned a task framing. Δ_{mnf} is the distance when inputs distances are measured in framing number f. For task t, the framing f for which the sum $\sum_{m=1}^{N} \sum_{n \neq 1} \Delta_{mnf} p_{mt} p_{nt}$ is minimized is selected (where p_{mt} is the probability that trajectory m is an instance of task t). This selects the framing for task t in which demonstrated trajectories of that task look the most similar (i.e where the sum of distances between trajectories is the smallest, weighted according to probability that both trajectories are instances of task t). In experiment 3, a sign language word order syntax, which is used to find framings, is estimated in this step (detailed in the algorithm section of experiment 3).

6.4.3 Reproduction algorithm

After the learning algorithm is done, the imitator interacts with the interactant and performs a reproduction. In each reproduction, the imitator must decide what to do based on a current context consisting of low-dimensional representations of the interactants speech utterance and/or hand sign(s) (or the low-dimensional representation of noise in case of a non linguistic task), the 2D position of object(s), and its current hand position (in the different framings/coordinate systems). The object position(s) and the starting position of the imitators hand is randomized at the start of the reproduction.

Figure 6.3 shows an overview of the on line reproduction algorithm where we see how the estimates that was created by the learning algorithm are used. Specifically they are used to decide which demonstrations in memory are relevant to the current context and what coordinate system to use in the current situation (so that the regression algorithm is able to work with only relevant data, represented in the correct framing). It is a data

Algorithm 5 Overview of the iterative grouping algorithm

Input: M_1, S, N
- M_1 *is the initial membership probabilities*
- S *is the number of steps (S=50 is used in the experiment presented below)*
- N *is the number of demonstrations*

for $s = 1$ **to** S **do**
  $M_{mod} \leftarrow M_s$ ($m_{m;t}$ refers to M_{mod})
  $M_{old} \leftarrow M_s$ ($m_{m;t;old}$ refers to M_{old})
  for $m = 1$ **to** N **do**
    for $t = 1$ **to** N **do**
      for $k = 1$ **to** N, $k \neq t$ **do**
        $m_{m;t} \leftarrow m_{m;k;old}\xi_{k;t} + (1 - m_{m;k;old})m_{m;t}$
      end for
    end for
  end for
  Rescale
  Preferring hypotheses with few movement types:
  $\forall : 1 < m < N, 1 < t < N$**:**
  $m_{m;t} \leftarrow m_{m;t} \times (\sum_{\tau=1}^{N} m_{m;\tau})^{1/4}$
  Rescale
  $m_{m;t} \leftarrow m_{m;t} + 0.0001$
  Rescale
  $M_{s+1} \leftarrow M_{mod}$
end for

note that if the push factor $\xi_{t;k}$ is positive $m_{m;t}$ will increase and if it is negative it will decrease in the central update step. Remember that a positive $\xi_{t;k}$ indicates that the policy similarity between t and k is higher than the weighted average. The rescaling makes the memberships of a single demonstration sum to 1

driven on line algorithm where the estimates of the learning algorithm helps select what data to attend to, and what framing to use.

Group selection

The membership estimates calculated by the learning algorithm gives us groups of contexts (the contexts of those demonstrations whose trajectories were estimated to be instances of the same movement). These groups of contexts are to be compared to the current context.

Since speech utterances and sign language gestures are represented in the same way as the physical context (in this case the position of an object) the distance between contexts can be computed. Therefore a single algorithm can deal with tasks where the linguistic input is important for what to do as well as tasks where the linguistic input

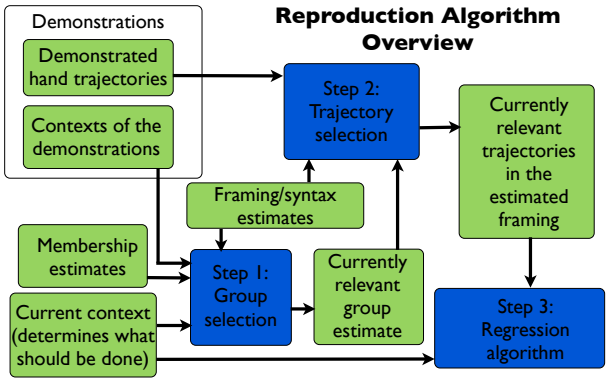

Fig. 6.3: The results of the learning algorithm is used during the reproduction. In step 1, the contexts of the groups of the membership estimates is compared to the current context. In step 2 the framing estimates and the group selected in step 1 is used to get the data needed by the ILO-GMR algorithm (reevant trajectories in the currently relevant framing).

is completely irrelevant (both for execution and triggering).

Experiment 1 and 2 compares the full contexts, while experiment 3 uses an additional assumption and the estimated syntax to decide how to compare contexts.

Trajectory selection

What the regression algorithm needs is examplar trajectories of the task to be performed, represented in the currently relevant framing. In all experiments, the trajectories of the selected group is used and in experiments 1 and 2, the framing of that group is used. In experiment 3, the estimated syntax determines which hand sign to use for finding the framing.

Regression algorithm

Incremental Local Online Gaussian Mixture Regression (ILO-GMR), introduced in [104, 102], was used in experiments 1 and 2. ILO-GMR is a variation of the Gaussian Mixture Regression method (GMR) [101] [65] which allows for fast incremental learning and online reproduction without meta parameters that needs to be tuned by hand (the two meta parameters that needs to be set is kept the same for all tasks). Experiment 3 uses a simpler regression algorithm (detailed later).

6.5 Experiments

Each of the three experiments are described separately, followed by a general discussion of the combined results. They all share the basic properties of what is described above, but the individual differences in setup and algorithms need to be detailed. The performance of the imitator is evaluated by (i) comparing the estimated task groupings of the learning algorithm with the actual task identities of the demonstrations (ii) comparing the estimated framings or the estimated syntax with the actual framings/syntax (iii) comparing the task group selected with the intended task when confronted with the interactant during the reproduction phase (iv) comparing the estimated framing with the actual framing, and finally (v) by comparing the reproduced hand movements with the task description and the corresponding demonstrations.

6.5.1 Experiment 1: Extending the context to include speech

The first experiment exemplifies that the context of an imitator can be extended to include the speech of an interactant, without introducing any discrete representations (where sometimes the speech is a communicative act, while other times it is completely irrelevant to what the imitator should do). The scientific novelty of this experiment consists in making an utterance of a second human part of the context, and learning an unknown number of tasks from unlabeled demonstrations.

Modeling the utterance as symbols would only leave the problem of finding the link between the utterance symbols and the behavior symbols, as well as finding the appropriate motor commands, given the correct behavior symbol. As described earlier fully continuous communicative acts and behaviors are investigated, meaning that the robot does not know how many different words it has observed or how many different tasks there are (or how many of the tasks are linguistic).

Algorithm

The robot can encode sensorimotor policies using one of three framings. Framing 1 encodes the position and speed of its hand in an absolute fixed reference frame (in addition to the absolute position of the object and the speech sound). Framing 2 encodes the position and speed of its hand in the object centered referential (all other dimensions being equal). Framing 3 includes both the absolute and relative position and speeds of the hand (this is never the correct framing, and during the reproductions, it is also never the estimated framing).

For the 3 linguistic tasks (tasks a, b and c, see below) the same object position distribution was used (uniformly distributed over the intervals: $-1 < x < 1, 1 < y < 2$) and for the 2 non linguistic tasks the object y positions were drawn from the uniform distribution $-1.25 < y < -0.5$ and the x positions were drawn from $-1 < x < -0.25$

for task d and $.25 < x < 2$ for task e. The starting hand position (demonstration and reproduction) is always drawn from $-0.25 < x < 0.25, -1.5 < y < -1.25$

An earlier, but very similar, version of the grouping algorithm described below is used.

During reproduction, it is necessary to know what task should be performed. Given state S in the extended context (in this case: the speech utterance of the interactant and the object position) the next step is to find the group of demonstrations that was performed in this situation. For each group of demonstrations the mean μ_{dt} and variance σ^2_{dt} of the data in dimension d is calculated for each dimension (this always consists of 3 speech dimensions and 2 object position dimensions). To determine what task is to be executed in the current state S, each task grouping gets a relevance score $R_t = p_{1t} \times p_{2t} \times ... \times p_{Dt}$, where p_{dt} is the probability density of a gaussian distribution with mean μ_{dt} and variance σ^2_{dt} in the current state S. The task with the highest relevance score R_t is selected and the data of that group (seen in the framing of that group) is used to build local models during the entire reproduction. The relevance score of a task is designed to be higher if the current state is similar to the data of that task.

The regression step uses the ILO-GMR algorithm detailed previously.

Tasks

Five different tasks are learnt at the same time. Three tasks should respectively be triggered by three keywords, and two tasks should be triggered by the objects position being in a certain region. What looks like communicative behavior to an outside observer is treated exactly the same as any other part of the context by the imitator. The fact that a single algorithm can learn both how to respond to a traditional context and how to respond to a communicative act (without being told which is which and not even knowing if there is just communicative tasks, just non communicative tasks or a mix) illustrates the point that a single imitation strategy can be used for language and other sensorimotor learning. We do not explore the situation where a keyword for one task is spoken at the same time as a the objects position is in a region that should trigger another task. The imitator has no way of knowing how to resolve the conflict and would need to see the demonstrator respond in such a situation in order to know what to do. The algorithm would pick one task based on which context is most ”task typical” and execute that task as usual (the algorithm contains the relative match for the different tasks, so the information that the imitator is not certain of what to do is available, but it is not currently used).

- Task a) When the word ”flower” is spoken: encircle the object counter clockwise (task defined in framing 2).
- Task b) When the word ”triangle” is spoken: draw a triangle clockwise to the left of the robot (task defined in framing 1).

- Task c) When the word "point" is spoken: draw a big square clockwise (task defined in framing 1).
- Task d) When the object is close to the robot and to the right: draw a small square counter clockwise with the bottom right corner at the object (no matter what the speech input is) (task defined in framing 2).
- Task e) When the object is close to the robot and to the left: Encircle counter clockwise the point (0,0) in the fixed reference frame no matter what the speech input is (task defined in framing 1). The policy in this task is identical to the one in task a) in that it is to encircle the point (0,0), with the only difference that the reference frame is different (besides different relative starting positions the demonstrations of task a in framing 2 looks just like the demonstrations of task e in framing 1).

Four demonstrations of each task were provided and presented to the robot unlabeled.

Results

In figure 6.4 we can see the results of the grouping algorithm. The demonstrations have been sorted into 5 groups with 4 demonstrations each. We can also see what framing was estimated for each group (marked with a * in the figure). We can see that each group contains hand trajectories that correspond the one of the tasks descriptions (which task the trajectories correspond to is indicated to the left in the figure). As figure 6.4 contains all the demonstrations, we do not include a separate data input figure. Each of the estimated task groups has four speech points and four object positions shown in the two columns to the right.

In figure 6.5 we can see each of the 20 reproductions individually, with the imitators estimate of the currently appropriate framing seen in the top left of each reproduction. In two of the reproductions of task c, the imitator completes a few correct laps around the triangle, but then starts drifting into the middle. Otherwise we can see that the tasks are reproduced adequately if we compare the reproductions with the task descriptions and the demonstrations shown in figure 6.4.

Discussion

We have demonstrated that it is possible for a robotic imitator to autonomously group unlabelled demonstrations into separate tasks and to find the correct framing for these task even if the number of tasks is not provided. We have also shown that language can be included as the context in a task and that the imitator can determine for what tasks the linguistic input is relevant. This means that an imitation learning system evolved for non linguistic reasons can generate language (any potential subsequent adaptations

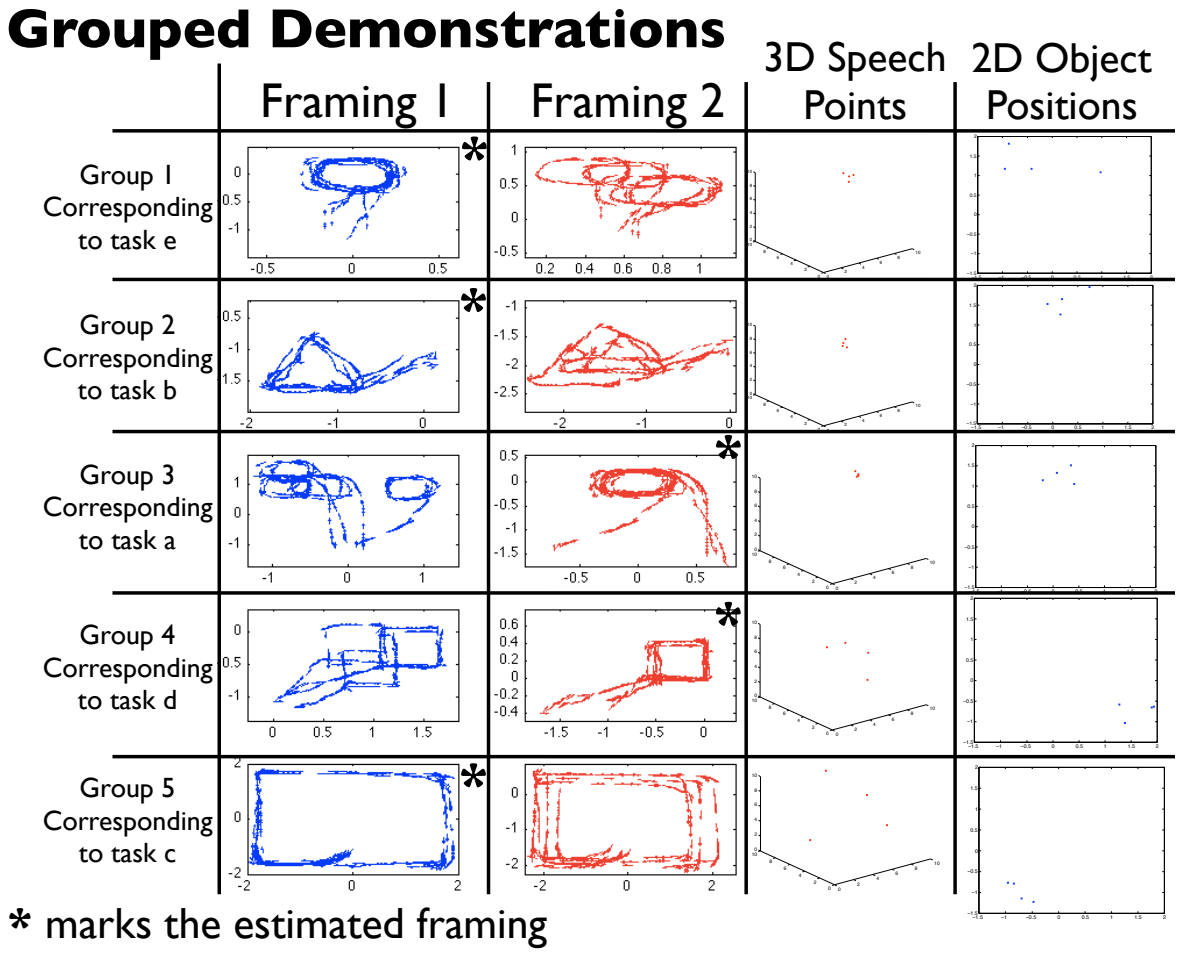

Fig. 6.4: This shows the result of the grouping algorithm applied to the data in experiment 1. The five task groups that where found are shown in framing 1 to the left and in framing 2 to the right. We can see that each group found does correspond to one of the tasks described in the task descriptions, and we can also see that the correct framings were found (also notice that the demonstrations look more coherent when viewed in the correct framing). The ordering of the tasks are random and will be different each time (this time it is e,b,a,d,c) but each time the same set of region-framing-data tuples are found. In order to avoid duplication, this figure also serves to show what was demonstrated (since each of the task groups found consists of the demonstrations of one task, the only difference of showing the task demonstrations separately would be in the ordering).

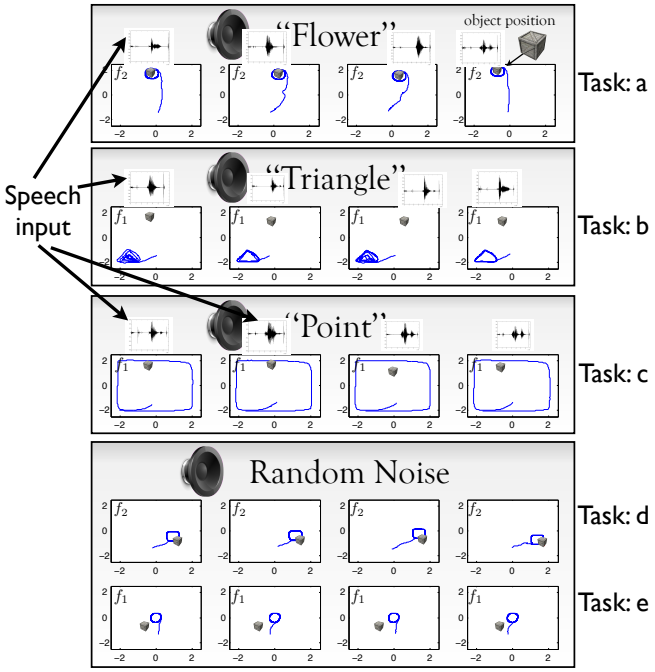

Fig. 6.5: This shows 4 reproductions each for the five reproduced tasks of experiment 1 in the absolute reference frame (the f_1 or f_2 indicates what framing the imitator estimated during the reproductions; each time the framing found is correct). Comparing to the demonstrations and the task descriptions, we can see that the reproduced trajectories correspond reasonably well with what the imitator was supposed to do. We see that the triangle task (task b) has a tendency to sometimes go into the middle of the triangle and circulate in a deformed trajectory after having completed a few correct laps. This problem is not due to the grouping algorithm and demonstrates a shortcoming of the ILO-GMR algorithm that is presented with only relevant data in the correct framing (and it is presented with all the relevant data).

happening as a response to language being a predictable part of the environment is outside the scope of this thesis).

6.5.2 Experiment 2: Relaxing the assumption of a single channel of communication

The context of this experiment includes interactant speech, interactant hand gestures/signs, the position of an object and the starting position of the robots hand. There are tasks that are to executed in case of events in each of these 4 domains (when the object is to the left of the robot task 1 should be performed and when speech include "circle", task 4 is to be performed, etc.). Those tasks that are triggered by speech or hand signs of the interactant can be seen as linguistic while the other tasks are ordinary sensorimotor tasks. From the point of view of the robot, there is however no difference as it in all cases notices that certain states in a low dimensional space should trigger certain behavior. The tasks are also chosen in a way that demonstrates the possibility for reciprocal communication. For example the correct response to the object being to the left is to draw a "L" shape, and the correct response to the utterance "dubleve" is to draw a "w" shape (other times the task is a non communicative response to the interactants communicative act, such as when the interactant makes a "s" shaped hand sign and the correct response is for the imitator to move its hand in a square). The imitator thus performs acts that an outside observer might consider communicative (mixed with other responses). From the perspective of the demonstrator, the "L" shape has a communicative meaning (in this case that an object is to the left from the robots point of view), but from the point of view of the robot it is just a hand movement similar to the "encircle the object" movement. It has learned a normative rule that a certain context should trigger a certain response, and the "linguistic" nature of its response comes from the fact that the demonstrator is a linguistic human that tried to demonstrate a linguistic convention. This is possible since the type of imitation performed is not "if I do x in world state y, then z will happen" but in the shared intentionality sense of "I should do x in word state y". This is similar in principle to learning that the correct response to seeing berries is to shout "berries" simply because a demonstrator is doing this (and not as a response to observing benefits of shouting "berries" due to some tribal enforcement mechanism).

Setup

The setup is similar to the previous experiment, with the addition that the interactant not only produces speech utterances, but can also perform hand movements/signs. The hand signs are also projected into a three dimensional space (as described earlier).

Tasks

There are 7 tasks defined and the demonstrations of them can be seen in figure 6.6. Two of the tasks are to be performed as a response to a specific object position, two tasks as a response to a speech command, two as a response to a gesture and one should be triggered when the robot hand starting position is in a certain zone. Just as in experiment one, it never occurs, neither during demonstration or reproductions, that the context contains two such conditions.

- **task 1)** When the object is to the left, task 1 is to be performed: drawing an L shape.
- **task 2)** When the object is to the right task 2 is to be performed: drawing an R shape. Tasks 1 and 2 are meant to demonstrate that it is possible to learn to give a sign as a response to a world state (something that might look like a description of the world to an external observer).
- **task 3)** When the word "dubleve" (french for w) is uttered by the interactant task 3 is to be performed: drawing a w shape.
- **task 4)** When the word "circle" is uttered by the interactant task 4 is to be performed: going around in a circle around the point 0,0 in the reference frame of the robot. Tasks 3 and 4 shows that verbal commands can be used either to draw a shape or do an action.
- **task 5)** When an "S" shape is made by the interactant task 5 is to be performed: go around in a square with the lower left corner of the square coinciding with the object.
- **task 6)** When the interactant makes a "P" shape with its hand, task 6 is to be performed: push the object. Tasks 5 and 6 tasks shows that it is possible for the architecture to handle different forms of symbolic communication; a sign can also be used to command an action. In these two tasks the approximate shape of the gesture determines what to do so it might look symbolic; as long as the shape is similar to "S" the square task is performed, and the exact shape have no influence on how it is performed. The position of the object also affect the task execution but here it smoothly modifies the policy.
- **task 7)** When the starting position of the robot hand is far away task 7 should be performed: go to the point 0,0 in the robots reference frame. This task was not correctly handled by the grouping algorithm (the reasons for the failure are discussed further below), but was meant to demonstrate a traditional sensorimotor task without any for of communication (the interactant is ignored and the demonstrator does not view its behavior as communicative).

Together these tasks show descriptions of the environment (the "R" and "L" gestures) as well as responses to both the environment and interactant actions (speech and gestures). This setup was simulated in the same manner as in experiment one.

Results

The results of the grouping algorithm can be seen in figure 6.7. In order to make viewing of the results easier the first four demonstration (1 to 4) are of the first task, the next four demonstrations (5 to 8) are of the second task, and so on. The fact that they are demonstrated in this pattern has no impact on the algorithm but makes it possible to immediately determine visually if the algorithm was successful. The demonstrations of task 7 is not identified as a task, which is a failure of the algorithm, but the demonstrations of the other 6 tasks is grouped correctly.

Why task 7 is not grouped correctly

The 4 demonstrations of task 7 can be seen in figure 6.8 in different colors. The green demonstration might look similar to the other demonstrations from a human observers point of view, however to the policy similarity measure defined it is actually quite different from the red and blue demonstrations (the red demonstration is actually more similar to the demonstrations of task 1 than to the green demonstration). Remember that the similarity measure relies on the policy (the direction) being similar in points close to each other in the same task, this is however not true for these demonstrations. The framing for this task is the coordinate system relative to the robot (framing 1) and this input is indeed all that is needed to define a consistent policy. For the grouping algorithm to see the policies as similar it would however be necessary to view the output in terms of speed towards the point $posHx_r = 0$, $posHy_r = 0$, or movement in a coordinate system with one axis intersecting the starting position and the point $posHx_r = 0$, $posHy_r = 0$ as suggested in [105]. If there are intermediate demonstrations the grouping algorithm can succeed anyway according to the principle A is similar to B and C, B is similar to A,C and D, C is similar to A,B,D, and E and D is similar to B,C E and F, E is similar to C,D and F. The starting positions are generated randomly and often the demonstrations will be similar enough to be grouped together. With these 4 specific demonstrations it sometimes happens that demonstrations 1, 2 and 3 or demonstrations 1 and 2 are grouped together to form a task. The proper way to fix this problem would be to either provide a framing where the demonstrations look the same or to give the imitator the ability to find such a framing by itself.

Finding back the correct task and the correct framing from the current state

.

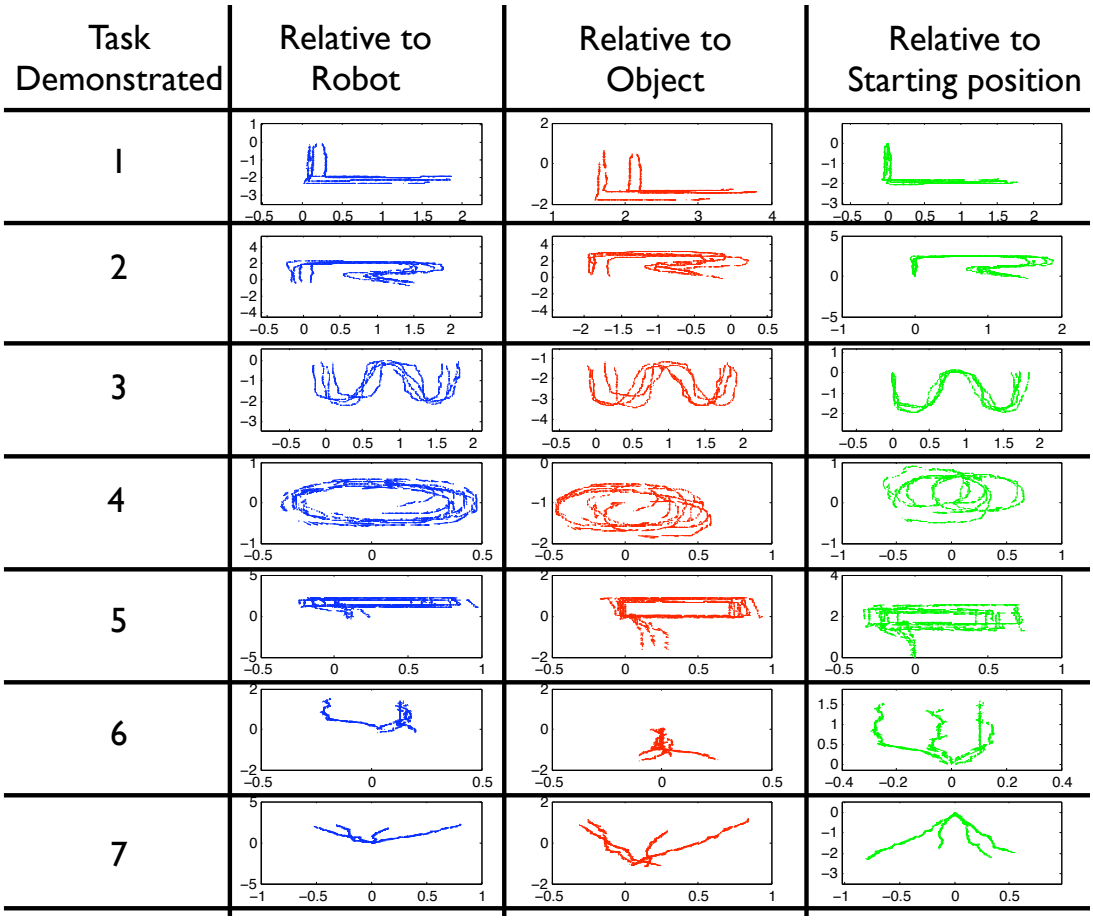

Fig. 6.6: The 7 tasks demonstrated in experiment 2. In the column to the left, in blue, the data is presented in framing 1 (relative to the robot). In the middle, in red, the data is presented in framing 2 (relative to the object). And finally in the column to the right, in green, the data is presented in framing 3 (relative to the starting position). The demonstrations of a task will look like several instances of a consistent policy in the correct framing but might look incoherent in the other framings. Task 1 and 2 ("L" and "R") is to be executed as a responce to the object being to the left (task 1) and right (task 2). Task 3 and 4 is to be executed as a response to specific speech acts ("dubleve" and "circle" respectively). Task 5 and 6 is to be executed as a response to hand signs (and "S" and a "P" respectively) and task 7 is to be executed in case of a starting position far away from the robot (roughly "when the arm is extended; move close to body").

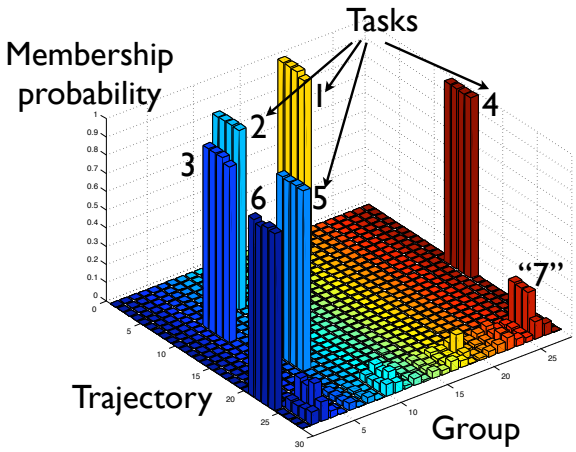

Fig. 6.7: The memberships found in experiment 2. The height of a pillar show the membership value of demonstration nr indicated on the left axis of the task indicated on the right axis. The 6 groups of 4 high pillars show tasks 1 to 6. There are several values on the right task axis that have no high values and those correspond to empty groups. For values 25 to 28 on the left demonstration axis we can see that the demonstrations of task 7 is not grouped together. The demonstrations of task number 7 has not been correctly grouped and when utilizing a cutoff value of 50% there are 6 groups formed (all of them with the correct data associated to them) but the demonstrations of task 7 are discarded (meaning that reproduction attempts of task 7 results in some other task being selected). In some runs demonstrations 1, 2 and 3 or demonstrations 1 and 2 of task 7 are grouped together as a 7th task (which also represent a failure since while reproducing task 7, the algorithm does not have access to all the relevant information).

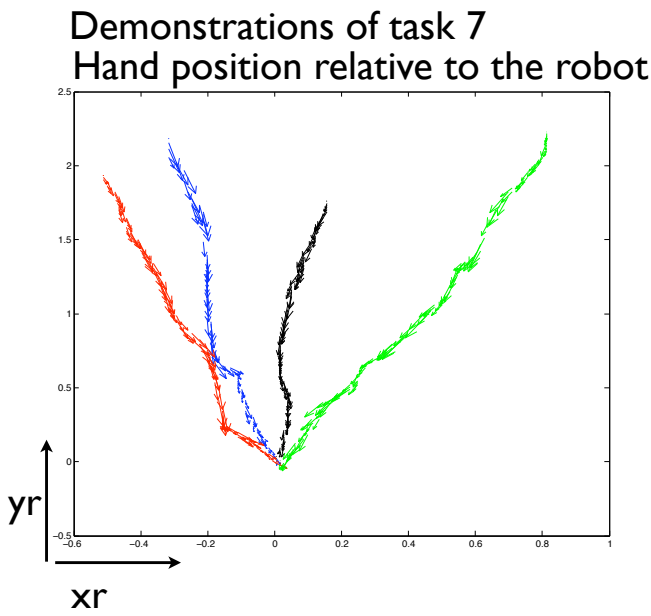

Fig. 6.8: Here we can see why task 7 of experiment 2 has not been correctly grouped. The similarity measure will simply not classify the red and the green demonstration as similar in any of the three framing available (they never move in the same direction, no matter how you pick points from the demonstrations). A framing that considers positions in a coordinate frame where one axis goes between the point $H_{x_r} = 0$, $H_{y_r} = 0$ and the starting point would work since the demonstrations would look the same plotted in this framing.

For the 6 tasks that are correctly grouped, the reproductions are successful except for around 5% failure rate for task 4[5]. The 6 tasks that were found all have the correct framing attached to them so when the correct task is found during reproduction the correct framing is also found. During each of the reproductions of the 6 tasks found by the grouping algorithm, the ILO-GMR algorithm was supplied by the grouping algorithm with only relevant data in only the correct framing and, as can be seen in figure 6.9, generally performs well. Sometimes the imitator acts "twitchy" at the top of task 4 during the second time around the circle if it gets too high (this is hard to see in the figure but is apparent when watching the simulated hand move during a reproduction). The push task stops slightly to the left of the object and drifts a bit when this point is reached even if the speed is greatly reduced. The path to the object in task 6 is also not completely straight (its not straight in the demonstrations but an optimal algorithm should average the directions and smooth out these differences). The reproductions of the three tasks where framing f_s (hand position relative to the starting position) is the relevant one looks very similar since the relevant part of the starting conditions are always the same (the relevant state is position relative to the starting position so even if starting position and object position differ each time, everything that affects policy stays the same).

Discussion

The main contribution of this experiment was to show that the channel of communication does not have to be known initially to the imitator or be confined to a single channel, but can be estimated. It also demonstrated that the actions of the imitator can be communicative acts. It also showed a situation in which the proposed similarity measure can fail if provided with only a small number of demonstrations.

[5]If looking only at the speech part of the context the task is correctly identified, but since we look at the entire context (we do not know that speech is the relevant thing to look at in the current context) sometimes the other parts of the context simply correspond better to task 3 that it outweighs the information from the speech part of the context. For tasks 3 and 4 the other parts of the contexts are drawn from the exact same distribution but since there are only 4 demonstrations each there is still some regularities (if, during reproductions, the starting position happens to be more similar to those in task to 3 than those in the demonstrations of task 4 this will tilt in favor of task 3. The speech is not different enough to always offset this). The problem would decrease with more demonstrations (as the demonstrations increase the estimated regions of the tasks would look more and more the same in those dimensions from which the starting position is drawn from the same distribution). However the problem would get worse as more irrelevant dimensions are added as each new irrelevant dimension adds a noise term in a random direction and increasing the expected distortion (although growing sub-linearly in the number of irrelevant dimensions)

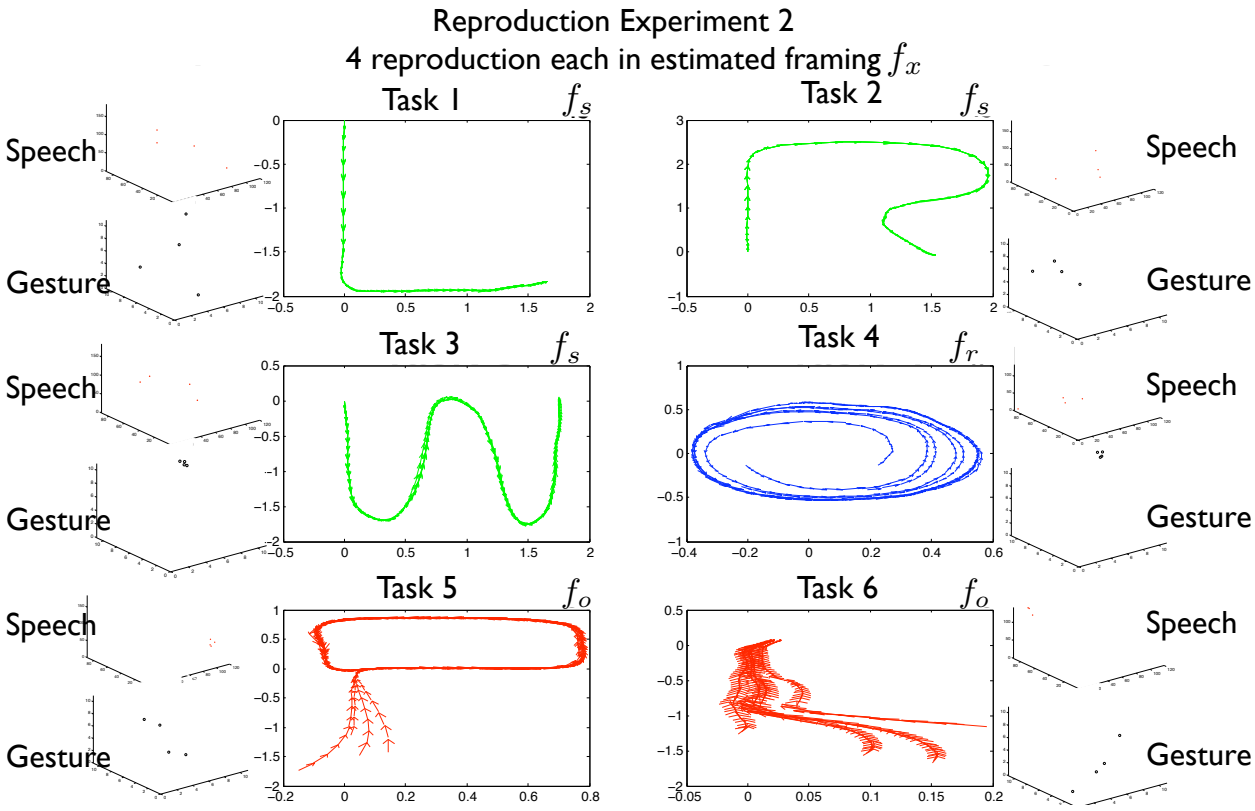

Fig. 6.9: Here we can see the reproduction of the 6 tasks that was correctly found by the grouping algorithm in experiment 2 (Task 7 was not found by the grouping algorithm, and due to this, no reproduction attempts of this task where made). Each task is reproduced four times with different starting conditions (the reproductions of each task is viewed in the inferred framing indicated to the top right of each subfigure). Tasks 1,2 and 3 are defined in the framing relative to the starting position of the imitators hand, meaning that in this framing the starting position is always at 0,0 (the imitator always start at 0,0 relative to the starting position), resulting in more homogenous reproductions. Comparing the reproduced trajectories with the task descriptions and the demonstrations we can see that they match fairly well and comparing the inferred framings with the framings of the task descriptions we can see that all framings where inferred successfully.

6.5.3 Experiment 3: Extending the types of word meanings that can be understood and learning simple word order syntax

This experiment explores more advanced types of words than direct action requests. Words corresponding to requests for attention to be on a specific object (i.e words that explicitly expresses framing), and thus corresponding to requests of internal cognitive operations, are learnt along with action request words. To imitate the response to the focus requesting words, this focusing operation must first be inferred in each demonstration, so that the problem reduces to finding a general policy from a set of known context-action pairs. Inferring this internal "focus on object" cognitive operation is done without introducing assumptions about how specifically an object focus will influence policy. The change in object focus is here viewed as any other action, like a hand movement or a speech utterance. This type of action is however not visible, and the change is only to internal cognitive structures of the teacher (the imitator uses a theory of how the internal structures of teachers mind works in order to imitate this action). The only assumption that is made about the internal structure is that a specific state will have a consistent influence on the hand movements of the demonstrator. This means that, while not directly observable, internal states are completely defined by the demonstrators behavior (instead of starting with a definition of how an internal state influences actions, and then infer what words leads to what state). An internal state that does not have a consistent influence on policy could not be imitated even if some context consistently resulted in this state.

Setup

In this experiment, the interactant uses a sign language with a simple 2-gesture syntax that has to be learnt by the robot. The syntax is as follows: the first gesture requests the internal operation of a specific object focus; a gesture drawing a "1" requests focus on the red object (i.e. the movement in the task should be defined in the coordinate system of the red object), a gesture drawing a "2" requests focus on the green and a "3" the blue object. The second sign requests a specific type of movement, defined in relation to whatever object is focused on[6]; a "4" requests performing the "triangle up" movement, a "5" the "triangle down" movement and a "6" requests the "circle" movement. For the imitator to find this word order, it needs to infer the internal operations performed and it needs to know what trajectories are instances of the same movement. Each gesture ("1", "2", "3", ...) is transformed into a low-dimensional (3D) point as described above.

The robot knows that each of the two hand signs have some meaning, and that there exist some form of word order syntax. This is an additional assumption that was not present in experiment 1 and 2 where the relevance of speech (or gestures, or object

[6]The policy maps hand positions in a coordinate system with (0,0) at the center of the object focused on to outputs.

positions) had to be determined. Since the robot here knows that the hand signs are communicative we have not simultaneously solved the two problems of: (i) finding out that the demonstrator is communicating and how (hand signs or speech) and (ii) the problem of finding word order syntax and imitating internal cognitive operations. The fact that the algorithm of experiment 3 makes use of this assumption that hand signs are communicative means that it can not be used to solve (i) and (ii) simultaneously without modification. The number of words is however still initially unknown to the imitator as well as which specific hand gestures are instances of the same hand sign (as before; a set of hand gestures can be 2 hand signs done many times each, or several hand sign done only a few times each).

Algorithm

When the grouping algorithm is successful we know what demonstrations are instances of the same movements. The new step is to find out what object was focused on during a movement. If we know that two demonstrations are instances of the same movement, for example a circle around the focused on object, we also know that conditioned on the correct object focus assumption, they will (on average) look similar.

The coordinate system in which a trajectory is the most similar to the other trajectories of the same movement is set as the coordinate system of that demonstration. For example; let's say that trajectories #1 and #3 are known to be instances of the same movement, and that they get the highest similarity score under the assumptions that the object focus during demonstration #1 was the blue object and the object focus during demonstration #3 was the green object. Then it is estimated that these were indeed the actual objects focused on (but taking into account comparisons with all instances of the same movement).

To find the word order we use the assumption that both hand signs are communicative acts and meaningful for the task. The within group distances of the first signs and the second signs are compared and the one that has the smallest distance is assumed to request a movement type, meaning that the one that has the biggest distance is assumed to designate the coordinate system (both are known to be meaningful, so if one of them requests a movement type, then the other must request an object focus). If the grouping algorithm is successful then each group consists of trajectories that are instances of the same movement. This in turn means that for each group, the same movement was requested but not the same object focus. Thus, the interactants hand signs which requests movements should be similar within each group (each group contains several instances of the same "request movement sign", but instances of different "request object focus" signs). If the movement is requested in the second of the two signs then, for each group, the within group distance of those signs will be smaller (since they are all the same sign) but the within group distance of the first signs will be bigger (since they are not all the same signs, requesting different object focus). If this is successful the imitator knows which of the signs designates the coordinate system and which one designates

the movement.

To determine what object focus, and what movement was requested during reproduction, the interactant hand sign that has been found to trigger a movement response is compared to the corresponding signs of all demonstrations and the group of the demonstrations whose sign is closest is assumed to be demonstrations of the correct movement. This results in a smaller dataset that is used for the rest of the reproduction.

The same is done to find the object focus: The interactant hand sign that has been found to trigger an object focus response is compared to the corresponding signs of all demonstrations and the object focus of the demonstration whose sign is closest is assumed to be the correct coordinate system.

A simple regression algorithm is used that, at each timestep during the reproduction, finds the 50 points that are closest to the current state (using the estimated framing) in the reduced dataset found earlier (consisting of only the relevant trajectories). The average of the output of these points is used.

Results

In figure 6.10 the 12 demonstrations are shown relative to the three different objects. The appropriate response to six of the total nine possible combinations of communicative inputs are demonstrated. The algorithm finds the number of movements and correctly infers all the internal actions as well as the word order. In figure 6.11 we can see that the imitator successfully reproduces in all nine combinations, which shows it has learnt to master the combinatorial structure of the sign language of the interactant.

A total of 36 successful reproductions, where the top left, middle middle and bottom right each show 4 correct reproductions of an unseen task (an unseen combination of hand signs). The edges of the triangles are not as sharp as they should be and, when the starting position in the circle movement is far to the right of the object, the imitator initially makes a too big semi circle before falling into the correct small circle movement (more sophisticated methods for the reproduction could be used on the data obtained, but that is not the focus of the current experiments and the reproduction ability was enough for our purposes).

Discussion

We have shown 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. We have also shown that the meanings of the signs learnt can include requests to perform unseen internal operations (focus on object) of a demonstrator under a set of conditions. First, it is neccesary that the unseen operation is performed as a predictable

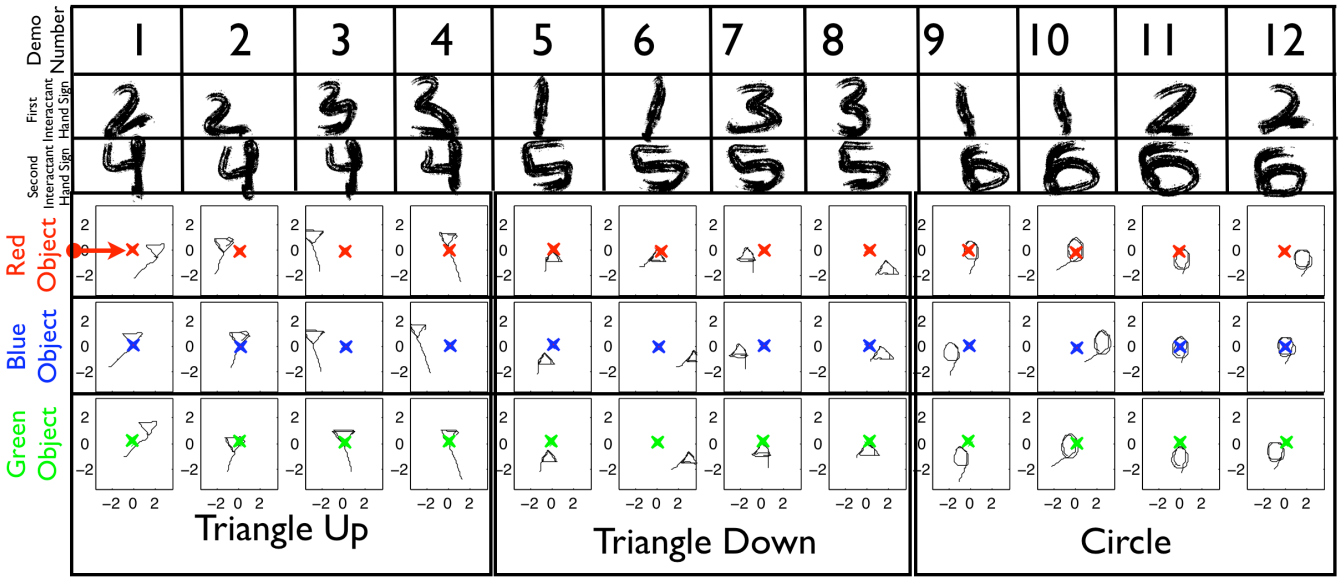

Fig. 6.10: Here we see the 12 demonstrations relative to the three objects of experiment 3. The demonstrator observes the first sign, the second sign and then performs the hand movement presented under. Each of the trajectories are shown relative to the three objects. The rows marked "red", "blue" and "green" respectively show the same trajectory, but each row show the trajectory in the a coordinate system centered on the respective object.

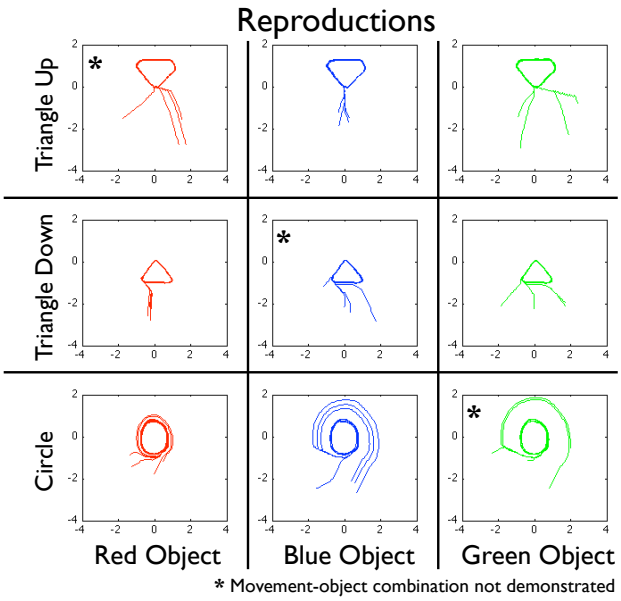

Fig. 6.11: Here we can see the 36 reproduction attempts. The rows indicate movement and the columns indicate coordinate system. In each case the signs given to the imitator led to correctly finding the correct data and the correct coordinate system. Comparing the reproduced trajectories with the task descriptions and the demonstrations, we can see that the corners of the triangles are a bit to round and that in some of the circle task reproductions there is some odd behavior before settling into the correct circular movement, but that overall the reproductions where reasonably accurate. There is also no apparent degradation in performance in the three combinations that were not demonstrated, showing the ability to generalize.

response to a part of the context visible to the imitator. Second, it is also necessary that the operation resulted in a state that had a consistent influence on a policy of the demonstrator which determined actions that were observable by the imitator. We have further shown how imitating these internal operations resulted in a policy that is able to generalize correctly and results in successful reproductions in situations where there are no demonstrations.

6.5.4 Further investigation of the grouping algorithm

To examine the main aspect of the grouping algorithm, we generate simulated data and examine under what conditions the algorithm will work. Data is generated with randomized similarity for each trajectory pair. Then the similarity of those pairs that are instances of the same task is multiplied by a constant k. The performance of the algorithm is then checked as a function of k.

Setup

For each trajectory pair, 200 values are generated and summed, each drawn from a uniform distribution between 0 and 1. If the trajectories are instances of the same task, the sum is multiplied by k. The membership matrix is initialized as usual and then the algorithm step where similarity is used to update the membership matrix is performed 200 times.

The experiments are performed on 28 trajectories, consisting of 4 trajectories that are not instances of any task and 6 groups of 4 trajectories each.

Performance is measured in two values: the proportion of correctly classified tasks, and the proportion of correctly classified non task trajectories. If all the trajectories of a task has membership values higher than 0.5 in the same group, and no other trajectory has a membership value higher than 0.5 of that group, the task is considered correctly classified (and otherwise a failure). If a trajectory that is not an instance of any task has any membership value higher than 0.5 then it is considered a failure, otherwise a success (if it is not a member of an actual task group, the algorithm will create a false task group that might get reproduced, and if it is a member of an actual task group, it will introduce corrupted data during reproduction. Each case would be a failure).

13 different k values are tested: 1,1.05,1.1,1.15, ..., 1.6. Each value is tested 100 times (new randomized similarities are generated for each run).

Result

Of 1300 runs, two instances of incorrectly grouped non task trajectories occurred, one misclassified non task trajectory at $k = 1.2$ and one at $k = 1.25$, meaning that the

grouping algorithm very rarely considers a non task trajectory as being an instance of a task.

In figure 6.12 we see task success as a function of similarity, with the bars indicating variance. When the similarity between two points is 15% higher for trajectory pairs that are of the same task, the algorithm starts to correctly classify a few tasks correctly, and at 50%, all tasks are consistently correctly classified.

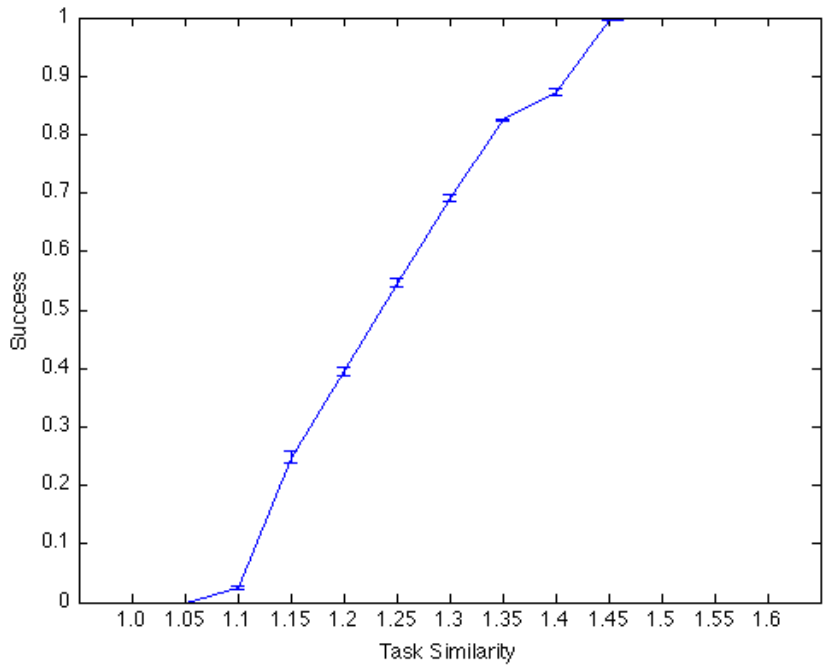

Fig. 6.12: Here we can see what level of similarity between trajectory pairs of the same task is needed for the grouping algorithm to succeed. The explanation for the low variance can be seen in figures 6.13 and 6.14 (correctly grouping one task makes it more difficult to correctly group the remaining tasks, which leads to the oddly low variance).

The very low variance needs to be explained. In figure 6.13 we see the success of each of the 1300 individual runs, showing that there are essentially no outliers. Each run correctly classifies a number of tasks in a narrow band, something that we would not expect to see if each task group had a certain percentage of being correctly classified, dependent on k, but independent on how the other tasks are classified. If that were the case we would expect to see a much higher number of outliers. This strongly suggests that correctly classifying one task group makes it more difficult to correctly classify another task group. Investigating the membership values directly in figure 6.14 we can see why this is. The trajectories of the same task have a strong tendency to form several identical groups, each with memberships values of lower than 0.5, a situation classified as a failure. Each time a task is correctly grouped, those trajectories disappear from all but one group, and it becomes easier for the other tasks to form multiple groups.

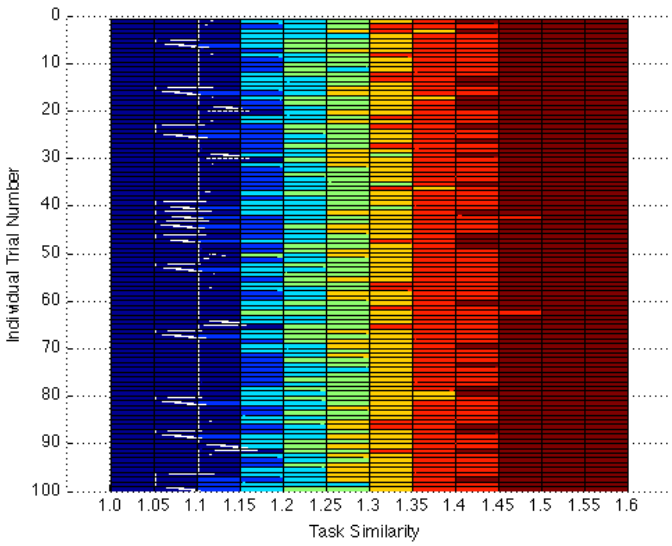

Fig. 6.13: Here we can see the results of each individual run. The success is color coded from low success at dark blue to high success at red. There are only 7 possible values corresponding to 0 correct groupings, 1 correct, etc. We can see that there are no outliers. For example at $k = 1.35$ there are 95 runs with 5 correct groupings and 5 runs with 4 correct groupings, and not a single run with 3 correct, or 6 correct groupings. At $k = 1.3$ all are either 3,4 or 5 correct groupings, with no run outside this band (and for each instance, all runs are within a narrow band, with the highest variation being at $k = 1.15$ where each run is between 0 and 3). This strongly suggest that the success of two task groups are not independent of each other. In figure 6.14 we can see why the difficulty of grouping a task is increased every time another task is correctly grouped.

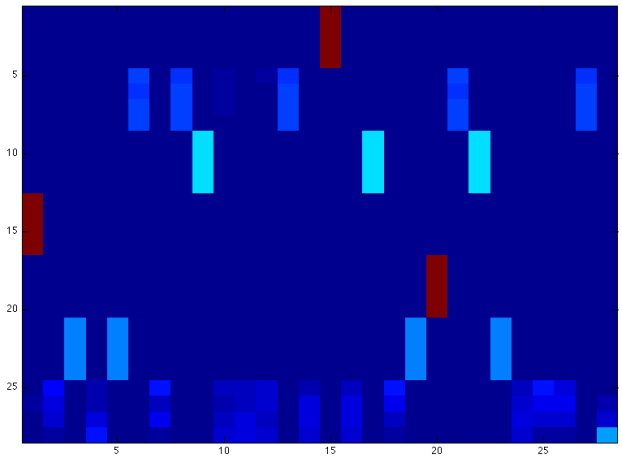

Fig. 6.14: Here we can see the memberships of one run with $k = 1.25$. The y axis indicate trajectory number and the x axis indicate group number. For visual convenience, task 1 consists of the first 4 trajectories (at the top), task 2 of the following 4, etc, and the non task trajectories are the last 4 (at the bottom). Red indicates a membership value of 1 and dark blue indicate a membership value of 0. For example, the light blue square in the bottom right corner indicates that trajectory 28 have fairly high membership in group 28, and the dark blue square to the left of this indicate that it has no membership in group 27. And from the red rectangle at the top middle of the figure, we can see that trajectories 1 to 4 all have membership value 1 in task group number 15. In general we can see that trajectories of the same task tend to either be correctly classified, or form several identical groups (trajectories 5 to 8 of task 2, trajectories 9 to 12 of task 3 and trajectories 25 to 28 of task 6).

Unassigned demonstrations "crowd" the existing tasks groups, making the formation of multiple identical groups more difficult to maintain. In the figure, tasks 2,3 and 6 compete for the same space.

6.6 Overall discussion

The first experiment of this chapter showed that the imitation learning context can indeed be extended to include a social partner who produces communicative acts. It was further shown that the appropriate way to respond to communicative acts can be learnt without discrete behaviors or discrete words, by treating them as any other part of the multimodal context. Treating the communicative acts as any other part of the context means that the imitator does not need to be told specifically that it is being talked to or given a special predefined "channel of communication". This is then further exemplified in experiment 2 were the interactant is sometimes communicating using hand signs, and sometimes using speech (and sometimes not communicating at all, in which case the demonstrator reacts to the more traditional parts of the context). The last experiment showed that the types of actions imitated does not need to be direct physical responses, but can be cognitive operations instead. If the internal cognitive operations of the demonstrator are the predictable result of a part of the environment that is visible to the imitator, and the operations have a consistent impact on observable teacher behavior, then the imitator can infer when to perform the operation (and the operation can be represented in terms of the behavioral changes that it induces).

Chapter 7

Conclusion

A formalism was proposed that makes clear what a successful learner behavior is, and that can be used to re interpret existing learning algorithms. New research avenues was made obvious by this reinterpretation, focusing on how several learning algorithms, that each learn from different types of input, can modify each other based on observations. An experiment then showed one way in which multiple tasks can be learned from unlabeled demonstrations. Finally, three experiments where presented where the context was extended to include another agent, allowing the exploration of language learning within the imitation learning framework and showing that the symbol grounding problem might be better dissolved then solved. Finding optimal solutions to the problem formalized is essentially impossible, and so is finding a way to be completely certain of how successful the learner is. This is however due to inherent difficulties in the setup where an agent is trying to figure out what a human wants it to do in an unstructured environment, and there is good hope of finding approximate solutions.

A very illustrative example is to wait to sweep the dust under the rug when the teacher is looking, so that the feedback is more informative (if the learner has two hypotheses saying that the teacher likes a clean apartment/a clean looking apartment, then the feedback can help distinguish between them iff the teacher is aware of the dust being swept under the rug). The full problem of modeling an informed version of a human teacher in an unstructured environment is completely intractable, but it is still possible to come up with simple solutions that improves performance over naive strategies maximizing a simplified measure (that is known to be unsuitable in some situations). The informedness of the teacher that is performing a demonstration or giving feedback can be treated in the same way as the lighting conditions of a map building robot (that moves around some environment, directs its sensors in different directions and that has access to some algorithm translating sensor readings into walls, doors, etc). If it knows that the estimates is more uncertain in bad lighting conditions, it can take that into account when making updates, just like an learner can estimate how informed a teacher was when giving feedback, and take that into account. The map building robot could also turn on the light switch whenever possible to improve the accuracy of its estimates, just like an learner can wait until the teacher is looking to sweep dust under the rug. The map building robot might never be able to get a perfect map, an might never be certain of how successful it was (it might never be able to be sure of how accurate its maps are), but it can still do better than a map building robot that simply ignores lighting conditions (building a perfect map might be impossible, but it can still outperform

a naive robot since updating under the assumption that the estimates are flawless will reduce performance).

One can use the formalism as inspiration to extend existing research either by combining and introducing parameters to a certain class of existing learning algorithms or by solving increasingly realistic inference problems (starting from algorithms dealing with more simplified success criteria, and drawing inspiration from a large existing literature). An example experiment was presented where a the learner started with an incomplete understanding of how to interpret the teachers evaluative comments. This limited interpretation ability was however enough to start building a task model. A task model allows the re-estimation of the learners interpretation hypothesis, and a better interpretation hypothesis allows for a faster learning of the task model. This situation is suitable for concurrently learning both elements (concurrent updates are suitable when each element makes learning the others easier). As could be expected, it was shown experimentally that the concurrent updates made learning of the task faster. Active learning was also applied and shown to improve performance.

The formalism offers a principled way of taking any learning algorithm that is trying to find out what a human wants an agent to do, and reinterpreting it as an interpretation hypothesis. For any such learning algorithm there must be an hypothesis, even if it is implicit, of how the inputs of the algorithm is related to what the human wants the agent to do. The formalism has offered a principled way of quantifying these types of hypotheses and to check their validity. The check of validity can be done since different hypotheses make different predictions regarding interaction histories (an hypothesis that a reward button is a very accurate evaluation of how successful the learner was will predict certain interaction histories, and another hypothesis that the reward button is a very noisy comparison between the success of the most recent learner action and the three previous learner actions will predict very different interaction histories). Two competing interpretation hypotheses regarding the same inputs can thus be tested, and one can be discarded in favor of the other (or it could be found that one hypothesis is good for certain types of tasks or in certain types of situations, and that other hypotheses are good in other situations). If some of the uncertainty of an hypothesis can be described as parameters, then the validity test can be used to determine which of two parameter values are better, and thus allows a parameter search. If there are several different interpretation hypotheses over different types of inputs, then finding out which one is the most accurate can be useful even if no updates are made since this can allow the learner to choose which types of interaction are the most useful (asking for feedback, or asking for another demonstration or just performing reproductions and observing facial expressions).

The simplified setup was reduced to an inference problem, opening up a whole new set of research avenues. One can for example start from any reinforcement learning algorithm and adapt it to suit a case where the rewards are not necessarily accurate, and where it is possible to for example to figure out how accurate they are, when they are accurate, and how to make them more accurate. The rewards are treated similar to inaccurate

sensor readings, allowing the learner to disregard certain signals (for example if the teacher missed something important) and to improve the informedness of the signals. One can also adapt the basic ideas behind existing methods such as particle swarm optimization (where each particle can for example be an interpretation hypothesis moving in an internal hypothesis parameter space and where the direction in that space is based on prediction quality on new data and the position of other particles, performing a reasonably effective search of the parameter space). Even if optimal solutions to the least simplified versions of the problem is far from possible, this approach is still suitable for guiding incremental extensions of existing work. Good solutions can be modified to problems that are slightly less simplified than what they are currently solving, and standard ideas can be adapted to find better and better approximate solutions to the most difficult problems. The central key to finding reasonably good solutions might very well be in allowing the learner to perform actions designed to gather information, which is already an active research area. This makes optimal solutions basically impossible to find (even if optimal is defined in terms of "optimal according to priors and information"), but makes reasonably functional solutions easier to find (it allows the learner to actively search for forms of interactions and those types of tasks that is possible to build reasonable models of given the current situation and type of teacher).

Experiments showed that it is possible to extend the usual learning from demonstration setup to explore new types of learning situations. The first experiment conducted showed that symbolic information labeling demonstrations is not needed. By using ILO-GMR, an incremental and local version of Gaussian Mixture Regression, it was possible to learn several different tasks from demonstration even when there was no symbolic information attached to them (the learner was not told how many tasks the teacher was demonstrating, and was not told which demonstration was of what task).

Experiments which included another agent showed that linguistic and non linguistic skills can be learnt using a single imitation learning system. Adopting normative rules from observations of others can result in rules for how one should react to inanimate objects and other non communicative parts of the context as well as how one should react to someone saying something or making a hand gesture/sign. The appropriate actions learnt (as a response to for example: seeing a blueberry, hearing a speech utterance or seeing a facial expression) can be both communicative or non communicative from the point of view of an external observer (for example drawing an "L" shape when an object is to the left). All this suggest that it is possible to investigate language acquisition without encountering anything resembling symbols that are in need of grounding.

Any learner of the type described here must obviously have a will to imitate (a motivation to adopt normative rules as opposed to only using observations of a teacher to update a world model). Another assumption made in experiment 3 is that the learner has a theory of mind, the learner is assumed to understand that the teacher is a special type of thing that can have internal states.

The presented approach can be seen as related to the hypothesis of Corballis that language evolved from the ability to use tools [100]. However, instead of language

evolving from an ability to use tools, language learning and learning to use tools results from a general sensorimotor imitation system in our account. Tool use is also not necessary for our account, and the selection pressure in favor of a normative imitation learning system could for example come from adopting rules regarding things such as: who to back in a fight, when to back down if challenged, when to start a fight, how to build social alliances/relationships in general[1].

This imitation system allows each generation to understand a big part of the practices of the previous generation, and either through invention or through non intentional generalization modify those practices. For example, assuming that some statistically common word order is a normative rule might result in such a rule being actually adopted. Once established, later generations can use it to reduce cognitive load when parsing sentence (and the benefit comes from being able to assume that everyone uses the same syntax, not the details of which specific syntax happened to be common/was adopted as a normative rule). As pointed out previously, our position is that capabilities and motives of shared intentionality is absolutely central to this general imitation learning system, making our system very compatible with Tomasello's view of shared intentionality as important to language. The learner would be a shared intentionality learner that adopts normative rules of how it should behave (as opposed to exclusively using imitation to learn predictive models of the physical world and other minds), and does so regardless of the types of contexts that should trigger a behavior (ignoring our judgement of whether the context makes the skill communicative or not).

The evolutionary accounts of language offered by Tomasello and Corballis centers on a gradual shift from action/gesture skills into a more sophisticated linguistic system (that at some point shifts to speech). This resonates well with the idea presented here of a single strategy/system used for all learning (this single learning strategy would simply be used on a gradually expanding set of contexts).

The difference between completely non communicative skills (for example how to manipulate an inanimate object by yourself for the purpose of getting food) and language skills is not only continuous. We further propose that each region on the spectrum has plenty of interesting skills. Close on the spectrum to manipulating inanimate objects are how to catch prey or avoiding a predator (by yourself). This introduces another agent in the context (the prey or the predator) and makes the correct actions dependent on the actions of another mind (as well as what that mind sees and hears, etc). Taking another step in this direction would be the skill of hunting an animal at the same time as a collaborator is hunting it. The correct actions are now also dependent on your collaborators movements (who will in turn take your actions into account). At

[1] If following the same set of rules as everyone else is more important than the content of the rules (and those rules are not always the same), then this creates a selection pressure for an imitation learning system that adopts the normative rules of the adults it encounters. When being born into a linguistic community is a predictable part of the environment, this could of course lead to further adaptations, something that this reasoning says nothing about either way (we are concerned with how a linguistic community could become a predictable part of the environment in the first place, not what secondary effects such an environment might have had).

the next level we have social skills where body language, facial expressions and eye gaze determine whether to take food openly as opposed to to stealing it or leaving it alone, or when to give away food as opposed to fight for it, or how to react when two other individuals are fighting. Overgeneralizing imitation learning by a new generation could now lead to a "linguistic" hand gesture for "go away / leave me alone / give it to me" as opposed to the "mere body language of someone involved in social conflict" in the previous generation.

Imitating internal cognitive operations and extending the context to include states in internal cognitive structures of the learner (or alternatively: estimated states of the teacher/interactant) can be useful in many types of settings and opens up entirely new types of tasks to imitation learning. A non specialized learner learning to drive a car (non specialized in the sense that the programmer had no idea this would be imitated) could learn how to follow the speed limit using this approach. It is of course possible to find complete visual histories and map them to driving behavior (since those histories will include the speed limit signs, at each instant it is possible to use them to determine the current speed limit). A more natural way of imitating a teacher that is following the speed limit would be to assume that there is some sort of internal cognitive structure that the teacher performs an operation on as a response to a speed limit sign. Then, at each moment, the current speed is a result of the current visible context as well as this internal structure.

If the learner is to drive in a country where rain can change the speed limit it will not be possible to learn this by experimenters defining some channel of communication and linking symbolic input in that channel to changes in internal states. What would be needed is a learner that is capable of treating rain, a traffic sign or a passenger saying "the speed limit here is 90" in the same way (and that is capable of discovering that states in these spaces apparently should influence the state of the "current speed limit" structure). The number of internal states, or the impact it has on driving, does not have to be known since it can be inferred from observation (given that the it is possible to detect that there are X number of influences on behavior coming from internal states). One language system that learns how to respond to the speech of the passenger and one **separate** action system that learns how to respond to the rain would be awkward (how to deal with the speed limit sign would depend on how the programmers chose to divide work between the two systems, and nicely exemplifies how arbitrary such a division becomes).

All these situations involve some aspect of the extended context that should result in an operation on an internal cognitive structure, whose state then should influence driving speed.

In the case of an learner learning to pick blueberries, all the following contexts would be treated the same way

- Directly observing blueberries.

- Observing another human bending down, picking something up, and then eating a blueberry (if there is one there are many).
- Observing another human bending down, picking up, and then holding up a blueberry.
- Observing another human shouting "blueberries!".
- Observing another human holding up a blueberry and pointing towards the lake.
- Observing another human saying "blueberries" and pointing towards the lake.
- Observing another human holding up a blueberry and saying "from the lake".
- Observing another human saying "there are blueberries at the lake".

When analyzing the teachers behavior (in each case: go and pick up the blueberries) it would make sense to assume that some internal state is triggered by each of these situations. Then that internal state has the same influence on behavior, no matter what caused the state in the first place. It does not seem necessary to build a learner with one separate language system, and one separate action system.

The same can be said about ownership, where a teacher might get the same "now Bill owns the apple" internal state from either of these situations:

- Bill finds and picks an apple.
- Steve owns an apple and Bill takes the apple from Steve when Steve is not looking.
- Steve hands the apple to Bill.
- Steve tells Bill "you can have that apple".
- Steve is holding an apple and Bill tells him "thats mine" and Steve says "Ok".
- Bill says "I will get you two apples tomorrow if I can have that now" and Steve says "Ok".

It is again reasonable to think of the teacher as having one internal state and there being many different contexts that might cause this state. It might be useful to approximate the teacher as having a "ownership state space" where many different types of observable contexts results the same states in this space and where the state in this space has a consistent influence on policy (where the influence is not dependent on what type of context caused it). A general ownership concept is useful for modeling many people if a significant subset have similar rules for when the state is triggered, and it has a similar influence on policy.

To exemplify how communicative goals can be described as different action requests, we can look at three different pointing behaviors; (i) "give me that", (ii) "notice that" (iii), and "share my attitude about that". The first type is a traditional action request, and the second is straightforwardly described as a request to change the state of knowledge about the world (make a change to a world model). The third type of pointing can be analyzed as requesting that a certain attitude is adopted towards the object. When this attitude is visibly adopted, the common ground can be changed to encode this shared attitude as an objective property about the object, which greatly facilitates communication[2]. To imitate the adoption of some specific attitude towards an object as a response to a communicative act is not different in principle from imitating the adoption of an attitude as a response to other aspects of the environment (some types of big heavy objects that suddenly start moving fast towards you should trigger one state while small things that look at you with two oversized eyes should trigger another state). The practical problems are the same no matter what triggers the internal operation; the part of the context that determines the internal operation must be isolated, the effects on behavior of the different possible internal states must be inferred, etc. In [24], experiments are presented and it is argued that the first and second type is present in one year old humans and in [25] the same is done for the second and third type of pointing.

It would also be possible to modify the setup to investigate the effect of shared intentionality on common ground and in restricting the hypothesis space of possible meaning of utterances. In the presented experiments there is no explicitly modeled common goal of the interactant and the teacher, and the teacher only performs one behavior as a response to a communicative act. If the learner assumes shared intentionality of the interactant and a teacher that performs several behaviors it can use this assumption to estimate what behaviors was requested. Even though the full hypothesis space of every combination of goals and requested behavior is probably impossible to work with it could concurrently: (i) use the current best estimate of the requested behaviors to build an explicit model of the common goal and (ii) use the current best estimate of the common goal to find the requested behaviors (preferring those that are relevant to that goal).

One of the limitations of the model is that it deals exclusively with a learner observing two interacting humans, while human infants can learn from interactions with a single human. The problem that is avoided by imitating one out of two interacting humans is that it is easier to divide what has been observed into context and response. If a

[2]Instead of requiring that the state of knowledge and the opinions of the other interlocutors are modeled explicitly one can simply talk and think as if the object has some additional properties (perhaps the huge positive effects of this type of communication and thinking can explain why humans tend to do this even when it does not really apply. It is perhaps such a good heuristic that the benefits of instinctively and without thinking using this strategy outweighs the problems that arise in the few cases where it fails). This is yet another aspect of the point made in [23] that sharing intentionality, perspective, attention, attitude, opinion, etc, etc will facilitate communication if this alignment is common knowledge (everyone is aware that everyone else is aware, etc).

single human says “throw” and then throws an object, it is not certain what is the context and what is the response, possible rules are: (i) “If I happen to be throwing an object, I should say “throw” first” (compare to saying “fore!”, before taking a golf swing), (ii) “if I happen to say “throw”, I should throw something”, (iii) “if another person is about to throw something I should say “throw” first”, (iv) “if someone says “throw” I should throw something”, etc. A robot would have to learn a lot or use extensive hand coded rules in order to disambiguate this (this is true even if the hand coded rules are not explicitly stated, but instead built into the setup). This ambiguity is apparently also difficult for parrots which is why Alex learns by observing two humans instead of learning from interactions with a single human. As mentioned earlier,[107] deals with parrots imitating humans, and it offers an explanation of the setup used here and a motivation for why it is needed in the case of parrots. This motivation works equally well for an artificial mind since it, just like the parrot Alex, is so different from a human teacher that it can not use the simulation theory of mind (it cannot determine what the teacher means by asking itself “what would I mean if I did that”). A complete model of language acquisition must include some method for estimating what behavior should be imitated and what is the context when both are performed by a single individual, making this setup a limitation (it removes complexities that a human child is able to deal with).

Another limitation is that the linguistic behavior is not very complex. Yet, we believe the models presented here are still interesting even though more complex linguistic behavior have already been investigated. One main interesting feature is that some abilities that are usually assumed is here explicitly modeled, for example finding the number of different utterances as opposed to symbolic input. The other interesting feature is that since there is no need to solve a symbol grounding problem, the behaviors that can be learnt becomes essentially open ended.

The symbol grounding problem can be avoided since there is no separate linguistic system whose symbols needs to be grounded. It will have to learn how and when to perform operations on internal cognitive structures, and how the state of those internal structures should influence policy, but this is learnt in a way that treats those operations as any other action, and that treats those states as any other part of the context.

The types of internal operations that can be learned this way thus becomes open ended. The practical problems become more severe when the requested operations become more complicated and do not effect behavior immediately. This is however a feature of the problem and can not be solved by other approaches either, unless the language learner starts with a hand coded version of the rule set. The imitation learning approach suggested does offer a general strategy for teaching a robot how to perform a more complex internal operation as a response to communicative acts. The learner’s current linguistic level, current theory of mind and current ability to observe the non linguistic parts of the situation defines what contexts can be observed. This will include “now Steve is focusing on the blue object” at one level of linguistic competence, but not at a

lower level[3]. The observable actions are also to some extent dependent on the current linguistic competence of the learner[4], and together this defines how the language can be extended. When the learner can predict an internal state from the currently observable context, this state can be inferred and now becomes part of the observable context. The noise is in principle not more problematic than other sources of noise, and might in fact be easier to remedy (as the noise to a degree comes from imperfect models of the teacher, a set of demonstrations that are to noisy to learn from can be made more clear by improving the models of the teacher that produced the data[5]).

Consider the rule "when getting an apple that belongs to Bill, taking it must be done silently and when Bill is not looking". If "that apple belongs to Bill" is already representable as a state in some internal structure from an earlier interaction and similarly "Bill being aware of what happens to the apple" as well as the fact that sound levels might be an important value to pay attention to, a demonstration of the task can be seen as mapping states in a simple context space to actions in a simple output space. If the learner does not already know about these internal structures, the task will look completely different and learning the policy will be a completely different type of problem. It would require a larger amount of training examples, a longer interaction history (to establish what it was that made the apple belong to Bill), and a larger amount of computation to find the relevant spaces. If the internal structures are known, learning this task can modify them: the effect of the "ownership" structure on behavior can be modified (for example be made dependent on who is observing), the way in which ownership is changed can be extended, the estimate of when "Bill being aware of what happens to the apple" can be modified, the types of context in which being silent is important can be modified, etc.

The use of non symbolic input also opens up different ways the interactant can use to communicate (for example: facial expressions, body language, tone of voice, traffic signs and written symbols), which is now limited only by the learners ability to transform its raw sensory experience into a low dimensional space, and that the learner has enough computational resources to pay attention to them (requiring good prior information of what is important or the time required to figure out what parts of the context is important[6]). The low dimensional space is a simple way of transforming data into a

[3]If the learner is observing Steve, and an interactant says "look at the blue object", then a linguistically competent learner can conclude that Steve is now focusing on the blue object. This can now be used as any other part of the context, such as "The interactant says "give me", the blue box is not heavy, Steve is focusing on the blue box"

[4]Steve shaking his head might be seen as a relevant action at one level of linguistic competence, but not at a lower level.

[5]If the relevant part of the context of a set of demonstrations are internal cognitive states, and the data is to noisy too learn from, the learner might gather more of the type of data that was used to build that model in the first place, or perhaps do more expensive calculations, etc

[6]If the learner has access to rich sensors and is able to generate large numbers of possible transforms from this rich input space to low dimensional spaces, it can use this method as an evaluation criteria (if the observed states in the resulting spaces predict what task was performed, it is possible that the evaluated transform is indeed capturing something about the interactant that the teacher considers

format where standard techniques can be used to what group a new point belongs to (in these experiments the observed teacher behavior is used to find groups and the position in the low dimensional space is used to find the group of a new point during reproduction).

Hopefully this thesis have exemplified the approach of systematically making assumptions explicit, and thereby finding new avenues of research by various ways of relaxing them. This has been attempted by theoretical contributions as well as by use of illustrative experiments dropping the assumption of task labels, dropping the assumption of a separate language learning faculty and by dropping the assumption of a known interaction protocol for feedback. There are many ways in which this research can be extended and hopefully this research has opened up new ways of looking at existing research in addition to illustrating new paths.

important). The limiting factor would not be in demonstrations (as we have shown that relatively few demonstrations is enough to evaluate channels of communication) but is instead limited by off-line computational resources (after observing demonstrations, the learner would be able to search for transforms without the help of a teacher, even if the hypotheses that is generated might at some point need to be verified in new demonstrations).

Bibliography

[1] Rizzolatti, G. and Arbib, M. A. Language within our grasp. *Trends in Neurosciences*, 1998, 21(5):188–194 18, 19, 99

[2] Glenberg, A. M. and Kaschak, M. P. Grounding language in action. *Psychonomic Bulletin and Review*, Vol 9(3), Sep 2002. pp. 558-565 18, 99

[3] Pulvermuller, F., Hauk, O., Shtyrov, Y., Johnsrude, I., Nikulin, V., and Ilmoniemi, R.Ê Interactions of language and actions. *Psychophysiology*, 40, 2003 18, 99

[4] Hauk, O., Johnsrude, I., and Pulvermuller, F. Somatotopic representation of action words in human motor and premotor cortex. *Neuron*, 2004, 41(2), 301-307. 18, 99

[5] Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Luciano Fadiga, Britta Wrede, Katharina Rohlfing, Elio Tuci, Kerstin Dautenhahn, Joe Saunders, Arne Zeschel: Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. *IEEE Transactions on Autonomous Mental Development*, 2010, In press. 15, 18, 23, 99

[6] D Perani, S.F. Cappa, M. Tettamanti, M. Rosa, P. Scifo, A. Miozzo, A. Basso, F. Fazio. A fMRI study of word retrieval in aphasia *Brain Lang* 2003, 85:357-68 18

[7] Massera, G., Tuci, E., Ferrauto, T., and Nolfi, S. The facilitatory role of linguistic instructions on developing manipulation skills, *Comp. Intell. Mag.*, 2010, 5(3): 33-42 12, 13, 22, 23, 24, 99

[8] Pecher, D. and R. Zwaan (eds). Grounding Cognition. Cambridge: *Cambridge University press* , 2005 19

[9] Barsalou, L.W. Perceptual symbol systems. *Behavioral and Brain Sciences*, 1999, 22, 577-609. 19

[10] Lakoff, G. *Women, fire, and dangerous things: What categories reveal about the mind* 1987 Chicago: University of Chicago. 19

[11] Mussa-Ivaldi, F.A. and Giszter,S.F. Vector Field Approximation: A computational paradigm for motorcontrol and learning. *Biological Cybernetics*, 1992, 67, 491-500. 19

[12] Piaget, J. *Intelligence and affectivity: Their relationship during child development*, 1954. Palo Alto, CA: Annual Review, Inc. 19

[13] Gibson, J.J. The Theory of Affordances. *In R. Shaw and J. Bransford (Eds.). Perceiving, Acting, and Knowing: Toward an Ecological Psychology. Hillsdale, NJ: Lawrence Erlbaum.* 1977 pp. 67-82 19

[14] Tomasello, M. *Constructing a Language: A Usage-Based Theory of Language Acquisition.* Harvard University Press. 2003 19, 24

[15] W. Croft and D. A. Cruse *Cognitive linguistics. (Cambridge Textbooks in Linguistics.)* Cambridge: Cambridge University Press 2004 19

[16] Harnad, S. The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*, 1990, 42:335–346. 19, 22

[17] Hoijer. H., The Sapir-Whorf hypothesis. *In: Language in Culture (ed. H. Hoijer)* 1954 pp 92-105. 19

[18] Steels, L. and Spranger, M. Can Body Language Shape Body Image?. *In Bullock, S. Noble, J. Watson, R. and Bedau, M. A., editor, Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 577-584, Cambridge, MA, 2008. MIT Press. 19, 22, 23

[19] F. A. Mussa-Ivaldi, E. Bizzi. Motor learning through the combination of primitives. *Philos Trans R Soc Lond B Biol Sci* Vol. 355 (2000), 1755-1769. 19

[20] Rizzolatti, G., Fogassi, L., and Gallese, V. Parietal cortex: from sight to action. *Currrent Opinion in Neurobiology* , vol. 7, 1997, pp 562-567. 19

[21] Graziano, M.S.A., Hu, X., and Gross, C.G. Visuo-spatial properties of ventral premotor cortex. *J. Neurophysiol.*, 1997, 77: 2268-2292. 19

[22] W, V, O Quine, *word and object*, MIT Press, 1960. 19, 23, 101, 102

[23] Tomasello, M. Origins of human communication. *MIT press* , 2008 19, 21, 24, 144

[24] Liszkowski, U., Carpenter, M., Striano, T., and Tomasello, M. 12- and 18-Month-Olds Point to Provide Information for Others *Journal of Cognition and Development* 7. 2006, pp. 173-187. 19, 144

[25] Liszkowski, U., Carpenter, M., and Tomasello, M. Reference and attitude in infant pointing *Journal of Child Language* 34. 2007, pp. 1-20. 19, 144

[26] Tomasello, M., Hare, B., Lehmann, H. and Call, J. *Reliance on head versus eyes in the gaze following of great apes and human infants: The cooperative eye hypothesis. Journal of Human Evolution* . 2007, 52, 314-320. 19

[27] Call, J., Hare, B., Carpenter, M., and Tomasello, M. Unwilling versus unable: chimpanzees' understanding of human intentional action *Developmental Science* 7:4 (2004), pp 488-498 21

[28] Schmelz, M., Call, J., and Tomasello, M. Chimpanzees know that others make inferences. *Proceedings of the National Academy of Sciences*, 2011. 108, 17284-17289. 21

[29] Call, J. and Tomasello, M. Does the chimpanzee have a theory of mind? 30 years later. *Trends in Cognitive Science*, 2008. 12, pp 187-192 21

[30] Tuomela, R. *The Philosophy of Sociality: The Shared Point of View*, Oxford University Press, 2007 21

[31] Grice. H. P. Logic and conversation. *In Cole, P. and Morgan, J. (eds.) Syntax and semantics*, vol 3. 1975. New York: Academic Press. 21

[32] Tomasello, M. Why don't apes point? *In N. Enfield and S.C. Levinson (Eds.), Roots of human sociality: Culture, cognition and interaction* 2006. pp. 506-524. 21

[33] Call, J. and Tomasello, M. The production and comprehension of referential pointing by orangutans. *Journal of Comparative Psychology*, 1994. 108, pp 307-317 21

[34] Langacker, R. W. *Foundations of cognitive grammar: Theoretical Prerequisites*. Stanford, CA: Stanford University Press. 1987

[35] Westermann G., Mareschal D., Johnson M.H., Sirois S., Spratling M. and Thomas M., Neuroconstructivism, *Developmental Science*, vol. 10(1), pp 75-83, 2007.

[36] Sirois S., Spratling M., Thomas M. S. C., Westermann G., Mareschal D., and Johnson M. H., Precis of neuroconstructivism: How the brain constructs cognition, *Behavioral and Brain Sciences*, vol. 31, pp. 321-356, 2008.

[37] Quartz S. R., and Sejnowski T. J., The neural basis of cognitive development: A constructivist manifesto,*Behavioral and Brain Sciences*, vol. 20, pp. 537-596, 1997.

[38] Steels, L. The symbol grounding problem has been solved. so what's next. *Symbols Embodiment and Meaning* Oxford University Press Oxford UK, 2007 1-18. 22, 103

[39] Oudeyer, P-Y. The Self-Organization of Speech Sounds, *Journal of Theoretical Biology*. 2005. 233(3), pp. 435–449 22

[40] S, M, Nguyen. P-Y Oudeyer. Socially Guided Intrinsic Motivation for Robot Learning of Motor Skills. 2013, submitted. 10, 17, 52

[41] Louis ten Bosch, Hugo Van hamme , Lou Boves and Roger K. Moore. A computational model of language acquisition: the emergence of words, *Fundamenta Informaticae*, Vol. 90, (2009), pp. 229-249 22

[42] Okko J. Räsänen, Unto K. Laine and Toomas Altosaar. Self-learning Vector Quantization for Pattern Discovery from Speech, *Proc. Interspeech* 2009 22

[43] Alex Park and James R. Glass. Towards unsupervised pattern discovery in speech. *IEEE Workshop on Automatic Speech Recognition and Understanding* , 2005, pp 53-58 22

[44] Paul Ruvolo, Ian Fasel and Javier R. Movellan. A Learning Approach to Hierarchical Feature Selection and Aggregation for Audio Classification. *Pattern Recognition Letters* (2010). 22

[45] Lallee S, Yoshida E, Mallet A, Nori F, Natale L, Metta G, Warneken F, Dominey PF. Human-Robot Cooperation Based on Learning and Spoken Language Interaction From Motor Learning to Interaction Learning *Robots, Studies in Computational Intelligence*, (2010) vol. 264, Springer-Verlag 23

[46] Saunders, J., Lehmann, H., Forster, F., Nehaniv, C.L. Robot acquisition of lexical meaning moving towards the two-word stage. ICDL 2012. 22

[47] Lyon C, Nehaniv CL, Saunders J (2012) Interactive Language Learning by Robots: The Transition from Babbling to Word Forms. PLoS ONE 7(6) 22

[48] Tikhanoff V., Cangelosi A and Metta G. Language understanding in humanoid robots: iCub simulation experiments. *IEEE Transactions on Autonomous Mental Development*. 2011 3(1), 17-29 12, 22, 23, 24

[49] Sugita, Y. and Tani, J. Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes. *Adaptive Behavior*, 2005 13(1):33–52. 12, 13, 22, 23, 24, 25, 26

[50] E. Bicho, L, Louro. W, Erlhagen (2010). Integrating verbal and nonverbal communication in a dynamic neural fi eld architecture for human-robot interaction. Frontiers in Neurorobotics 26

[51] Steels, L. and Loetzsch, M. Perspective Alignment in Spatial Language. *In Coventry, K.R., Tenbrink, T. and Bateman, J.A., editor, Spatial Language and Dialogue*, Oxford University Press. Oxford, 2008. 22, 23

[52] Steels, L. and Spranger, M. The Robot in the Mirror. *Connection Science*, 20(4):337-358 2008. 22, 23

[53] Wrede B., Rohlfing K., Steil J., Wrede S., Oudeyer P-Y., Tani J. Towards Robots with Teleological Action and Language Understanding, *in Workshop on Developmental Robotics of IEEE RAS Conference on Humanoid Robotics, Osaka, Japan.* 2012. 23

[54] Steels, L. Is sociality a crucial prerequisite for the emergence of language? *In: Botha, R. (2008) (ed.) The Prehistory of Language. Oxford University Press*, Oxford. pp. 18-51. 22, 23

[55] Steels, L. Experiments on the emergence of human communication. *Trends in Cognitive Sciences* 2006 10(8), pp. 347-349. 22, 23, 79, 103

[56] Rolf M, Hanheide M, Rohlfing K. Attention via synchrony. Making use of multimodal cues in social learning. IEEE Transactions on Autonomous Mental Development. 2009;1:55-67. 103

[57] Wrede, B., Schillingmann, L. Rohlfing, K. J. (2013): Making use of multi-modal synchrony: A model of acoustic packaging to tie words to actions. In: Gogate, L. J. Hollich, G. (Eds.): Theoretical and computational models of word learning: Trends in Psychology and Artificial Intelligence. Hershey: Information Science Reference: 224-240. 103

[58] Steels, L. Modeling the Formation of Language in Embodied Agents: Methods and Open Challenges. *In Nolfi, S. and Mirolli, M., editor, Evolution of Communication and Language in Embodied Agents*, pages 223-233, Springer. Berlin, 2010. 22, 23, 103

[59] Xu, F. and Tenenbaum, J. B. Word learning as Bayesian inference: Evidence from preschoolers. *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*. 2005 23

[60] Yu, C., and Ballard, D. A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, 70, pp 2149-2165 2007 22, 23

[61] Adele E. Goldberg. *Constructions at Work: the nature of generalization in Language*. Oxford: Oxford University Press. 2006 24

[62] Steels, L. The Recruitment Theory of Language Origins. *In: Lyon, C., C. Nehaniv, and A. Cangelosi (eds) Emergence of Communication and Language*. Springer Verlag, Berlin. 2007 pp. 129-151. 24

[63] Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 2005. 28, 675 - 691. 101

[64] Billard, A., Calinon, S., Dillmann R. and Schaal, S. Robot Programming by Demonstration. *In Siciliano, B. and Khatib, O. (eds.). Handbook of Robotics*, 2008 pp. 1371-1394. Springer. 10, 12, 99

[65] Calinon, S. and D'halluin, F. and Sauser, E. L. and Caldwell, D. G. and Billard, A. G., Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression, Robotics and Automation Magazine, vol. 17, 2010, pp 44-54. 13, 114

[66] Calinon, S., D'halluin, F., Caldwell, D.G. and Billard, A. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids), Paris, France, 2009. 12

[67] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous Helicopter Aerobatics through Apprenticeship Learning *International Journal of Robotics Research*, Nov. 2010, vol. 29, no. 13, 1608-1639 11, 12, 14, 16, 99

[68] F. Guenter, M. Hersch, S. Calinon, and A. Billard, Reinforcement learning for imitating constrained reaching movements, *Advanced Robotics*, vol. 21, no. 13, pp. 1521-1544, 2007. 104

[69] Calinon, S *Robot Programming by Demonstration: A Probabilistic Approach.*,EPFL/CRC Press , 2009. 12, 60, 81

[70] Chalodhorn, R., Grimes, D., Maganis, G., Rao, R., and Asada, M. Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces. *Proceedings of the IEEE international conference on robotics and automation* 2006. pp. 3693-3698. 12, 13

[71] Hyvarinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*,1999, 10 (3), pp 626-634. 12, 13

[72] Jenkins, O., and Mataric, M. A spatio-temporal extension to isomap non- linear dimension reduction. *Proceedings of the international conference on machine learning (ICML)* 2004 pp. 56-63. 13

[73] Vijayakumar, S., and Schaal, S. Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional spaces. *Proceedings of the international conference on machine learning (ICML)* 2000. pp. 288-293. 13

[74] D. Nguyen-Tuong and J. Peters. Local gaussian process regression for real-time model-based robot control. *Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 380-385. IEEE, 2008. 13

[75] Schneider, M., and Ertel, W. Robot Learning by Demonstration with local Gaussian process regression *International Conference on Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ 13

[76] Duy Nguyen-Tuong and Matthias W. Seeger and Jan Peters, Real-Time Local GP Model Learning, *in From Motor Learning to Interaction Learning in Robots*, 2010, pp 193-207. 13, 14

[77] Schaal, S. Peters, J. Nakanishi, J. and Ijspeert, A. learning movement primitives in, *international symposium on robotics research*. springer. 2004 12, 14

[78] Ng, A.Y. and Russell, S. Algorithms for inverse reinforcement learning *Proceedings of the Seventeenth International Conference on Machine Learning* 2000. pp 663-670 14

[79] Neu, G. and Szepesvári, C. Training parsers by inverse reinforcement learning. *Machine learning*, 2009 vol 77, number 2, pp 303-337, 14

[80] Thomaz, A., Berlin, M., and Breazeal, C.. Robot science meets social science: An embodied computational model of social referencing. *Workshop toward social mechanisms of android science (CogSci)* (2005, July) pp. 7-17. 12, 15

[81] Zukow-Goldring P., Assisted imitation: Affordances, effectivities, and the mirror system in early language development *in From Action to Language, Arbib, M.A. Ed. Cambridge: CUP*, 2006, pp. 469-500. 15

[82] Csibra, G., and G. Gergely. Social learning and social cognition: The case of pedagogy. *In Progress of Change in Brain and Cognitive Development. Attention and Performance, vol. XXI, edited by Y. Munakata and M. H. Johnson. Oxford: Ofxord University Press.* 2006 pp 249-274 15

[83] Yasser Mohammad and Toyoaki Nishida, Learning Interaction Protocols using Augmented Baysian Networks Applied to Guided Navigation, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* 2010. 14, 25, 64, 104

[84] Yasser Mohammad, Toyoaki Nishida, and Shogo Okada, Unsupervised Simultaneous Learning of Gestures, Actions and their Associations for Human-Robot Interaction *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pp. 2537-2544. 2009. 14, 25, 104

[85] Evrard, P., Gribovskaya, E., Calinon, S., Billard, A. and Kheddar, A. Teaching physical collaborative tasks: Object-lifting case study with a humanoid. *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009 Paris, France, pp. 399-404. 26

[86] Calinon, S., Evrard, P., Gribovskaya, E., Billard A. and Kheddar, A. Learning collaborative manipulation tasks by demonstration using a haptic interface. *Proceedings of the Intl Conf. on Advanced Robotics (ICAR)*, 2009 Munich, Germany, pp. 1-6. 26

[87] Calinon, S. and Billard, A. Teaching a humanoid robot to recognize and reproduce social cues. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2006 pages 346-351. 12, 26, 99, 104

[88] Lopes M, Cederborg T, Oudeyer PY (2011) Simultaneous acquisition of task and feedback models. IEEE International Conference on Development and Learning. 8, 15, 63

[89] A.L. Thomaz C. Breazeal. Understanding Human Teaching Behavior to Build More Effective Robot Learners." Artificial Intelligence Journal, 2008. 15, 64

[90] J. Grizou, M. Lopes and P-Y. Oudeyer. Learning the Meaning of Instruction Signals in Interactive Task Estimation in *J. Artificial Intelligence Research* 2013, Submitted 8, 15

[91] Manuel Lopes, Luis Montesano, Francisco Melo and Jos Santos-Victor. Abstraction Levels for Robotic Imitation: Overview and Computational Approaches. *in Olivier Sigaud Jan Peters editors, From motor to interaction learning in robots, Studies in Computational Intelligence, Springer Berlin / Heidelberg*, Volume 264, pp. 313 - 355, 2010. 14

[92] Manuel Lopes, Francisco S. Melo, Ben Kenward and Jos Santos-Victor. A Computational Model of Social-Learning Mechanisms. Adaptive Behaviour, 467(17), 2009. 14

[93] Cynthia Breazeal, Jesse Gray and Matt Berlin, An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots, *I. J. Robotic Res.* vol. 28, 5, 2009, pp 656-680. 26, 43

[94] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. L. Thomaz, Using Perspective Taking to Learn from Ambiguous Demonstrations, *Robotics and Autonomous Systems (RAS) Special Issue on The Social Mechanisms of Robot Programming by Demonstration* 2006, pp 385-393. 26

[95] K. Dautenhahn and C. L. Nehaniv. The agent-based perspective on imitation, *Imitation in animals and artifacts*, pages 1-40. MIT Press, 2002. 9, 105

[96] B.D. Argall, S. Chernova, M. Veloso and B. Brett. A survey of robot learning from demonstration in *Robot. Auton. Syst.*, 2009, 57, 5, pp 469-483 10, 16, 17

[97] M. Stolle, C.G. Atkeson, Knowledge transfer using local features, in: Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, ADPRL'07, 2007. 17

[98] B.D. Argall, B. Browning, M. Veloso. Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robotics and Autonomous Systems.* 2011, 59(3-4). pp 243-255. 15

[99] Kober J, Wilhelm A, Oztop E, Peters J. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots.* 2012, pp 1-19. 15

[100] Corballis M.C., *From Hand to Mouth: The Origins of Language.* Princeton University Press, 2002. 19, 140

[101] Calinon, S. and Billard, A, Incremental Learning of Gestures by Imitation in a Humanoid Robot, *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2007, pp255-262. 12, 13, 60, 81, 104, 114

[102] Baranes, A., Oudeyer, P-Y. (2009) R-IAC: Robust Intrinsically Motivated Exploration and Active Learning, *IEEE Transactions on Autonomous Mental Development*, 1(3), pp. 155-169. 6, 14, 82, 104, 114

[103] Thomas Cederborg and Piere-Yves Oudeyer. From Language to Motor Gavagai: Unified Imitation Learning of Multiple Linguistic and Non-linguistic Sensorimotor Skills, *IEEE Transactions on Autonomous Mental Development*, 2013, accepted. 18, 98

[104] Cederborg, T., Ming, L., Baranes, A., Oudeyer, P-Y: Incremental Local Online Gaussian Mixture Regression for Imitation Learning of Multiple Tasks, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* 2010. 6, 14, 82, 104, 114

[105] Komei Sugiura, Naoto Iwahashi, Hideki Kashioka, and Satoshi Nakamura, Learning, Generation, and Recognition of Motions by Reference-Point-Dependent Probabilistic Models, *Advanced Robotics*, Vol. 25, No. 5, 2011. 122

[106] G. Schwarz, Estimating the dimension of a model, *Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

[107] Irene M. Pepperberg and Diane V. Sherman 18 - Training behavior by imitation: from parrots to people ... to robots? *In, Imitation and Social Learning in Robots, Humans and Animals Behavioural, Social and Communicative Dimensions. Edited by C. L. Nehaniv and K. Dautenhahn* pp. 383-406 13, 145

[108] 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* , vol. 37, 2007, pp 286-298. 12, 60, 81, 82, 84, 87

[109] A. Dempster., N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society.* B, 39(1):, 1977, pp 1-38. 12, 35, 39, 83

[110] Angeli, A., Filliat, D., Doncieux, S., Meyer, J-A., A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. IEEE Transactions On Robotics, Special Issue on Visual SLAM. 2008. 82, 83, 84

[111] Yeung, D.Y. and Zhang, Y., Learning inverse dynamics by Gaussian process regression under the multi-task learning framework. In The Path to Autonomous Robots, G.S. Sukhatme (ed.), pp.131-142, Springer, 2009. 84

[112] Vijayakumar, S. and Schaal, S., LWPR : An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space, Proc. of Seventeenth International Conference on Machine Learning (ICML2000) Stanford, California, pp.1079-1086, 2000. 84

[113] Nguyen-Tuong, D. and J. Peters: Local Gaussian Processes Regression for Real-time Model-based Robot Control. Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), 380-385, IEEE Service Center, Piscataway, NJ, USA, 2008. 84

[114] Chang, C.C. and Lin, C.J., LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001. 84

[115] Angeli, A., Filliat, D., Doncieux, S., Meyer, J-A., A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. IEEE Transactions On Robotics, Special Issue on Visual SLAM. 2008. 82, 83, 84

[116] M, Ogino., H, Toichi., Y, Yoshikawa., M, Asada., Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. In *Robotics and Autonomous Systems*, 54, 5, 2006, pp 414-418 11

[117] M.N. Nicolescu and M.J. Mataric, Natural methods for robot task learning: Instructive demonstrations, generalization and practice, *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS).*, 2003, pp 241-248.

[118] M. Pardowitz, R. Zoellner, S. Knoop, and R. Dillmann, Incremental learning of tasks from user demonstrations, past experiences and vocal comments, *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 2007, pp 322-332.

[119] S. Ekvall and D. Kragic. Learning task models from multiple human demonstrations, Title, *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).*, 2006, pp 358-363.

[120] A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn, Correspondence mapping induced state and action metrics for robotic imitation, *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation.*, vol. 4, 2007, pp 299-307. 11

[121] A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn, Action, State and Effect Metrics for Robot Imitation. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).*, 2006, pp 232-237. 11

[122] A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn, Achieving Corresponding Effects on Multiple Robotic Platforms: Imitating in Context Using Different Effect Metrics. *In Proceedings of the Third International Symposium on Imitation in Animals Artifacts, AISB* 2005 Convention, pages 10-19. 11

[123] Nehaniv, C., and Dautenhahn, K., Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications, *Interdisciplinary approaches to robot learning.*,World Scientific Press, vol. 24, 2000, pp 136-161. 11, 16

[124] Akgun B, Cakmak M, Yoo J, Thomaz A. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In: *International Conference on Human-Robot Interaction.* 2012 11

[125] Aude Billard, Sylvain Calinon, Ruediger Dillmann, Stefan Schaal, Robot Programming by Demonstration, *Springer Handbook of Robotics.*, 2008, pp 1371-1394. 11

[126] N. Delson and H. West., Robot programming by human demonstration: Adaptation and inconsistency in constrained motion., *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*, 1996, pp 30-36. 16

[127] A. Billard., S. Calinon., F. Guenter. Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. *In Robotics and Autonomous Systems", 2006. volume 54", number 5, pages 370-384* 15

[128] Ude, A, Trajectory generation from noisy positions of object features for teaching robot paths, *Robotics and Autonomous Systems.*, 1993, pp 113-127. 12, 60, 81

11

[129] Ito, M., Noda, K., Hoshino, Y., and Tani, J, Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model, *Neural Networks.* , 19 (3), 2006, pp 323-337. 11

[130] Zoubin Ghahramani and Michael I. Jordan, Supervised learning from incomplete data via an EM approach, *Advances in Neural Information Processing Systems.* vol. 6, 1994, pp 120-127. 11

[131] Thomas P Minka. A family of algorithms for approximate bayesian inference. PhD thesis, 2001. 11

[132] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942-1948. 1995. 12, 84

[133] Baranes A, Oudeyer PY (2013) Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems* 61(1):49-73. 31

[134] Schmidhuber J (2010) Formal theory of creativity, fun, and intrinsic motivation. *IEEE Transactions on Autonomous Mental Development* 2(3): pp: 230-247 31

[135] J. Schmidhuber. Curious model-building control systems. In: *Proc. Int. Joint Conf. Neural Netw.*, vol 2, pp: 1458-1463. 1991 52

[136] V. Fedorov. Theory of Optimal Experiment. *Academic Press, Inc., New York, NY.* 1972. 52

52

[137] Nehaniv, C., and Dautenhahn, K., The correspondence problem, *Imitation in animals and artifacts.*, MIT Press, 2002, pp 41-61. 52

[138] Chrystopher L. Nehaniv, Nine Billion Correspondence Problems, *In C. L. Nehaniv and K. Dautenhahn (Eds.), Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, Cambridge University Press, 2007. 8, 10

[139] E. A. Billing, T. Hellstrm., A formalism for learning from demonstration. Paladyn , vol. 1, no. 1, pp. 1-13, 2010 17

[140] R. Sutton and A. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998. 17

[141] M. Cakmak, C. Chao, and A. Thomaz, "Designing interactions for robot active learners," IEEE Transactions on Autonomous Mental Development, vol. 2, no. 2, pp. 108-118, 2010. 17, 65

[142] A. Y. Ng and S. J. Russel, Algorithms for inverse reinforcement learning, in *Proc. 17th Int. Conf. Machine Learning.* 2000. 64

[143] D. Ramachandran and E. Amir,. Bayesian inverse reinforcement learning. In *20th Int. Joint Conf. Artificial Intelligence.* India, 2007. 65

[144] D. Fox, S. Thrun, W. Burgard, and F. Dellaert,. Particle filters for mobile robot localization. In *Sequential Monte Carlo Methods in Practice, A. Doucet, N. de Freitas, and N. Gordon, Eds.* Springer Verlag, 2001. 65, 66

[145] P.-Y. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation systems for autonomous mental development, *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265-286, 2007. 66

[146] M. Lopes, F. S. Melo, and L. Montesano. Active learning for reward estimation in inverse reinforcement learning. In *European Conference on Machine Learning (ECML/PKDD)*, Bled, Slovenia, 2009. 71

[147] R. Cohn, M. Maxim, E. Durfee, and S. Singh. Selecting Operator Queries using Expected Myopic Gain. In *IEEE/WIC/ACM RInternational Conference on Web Intelligence and Intelligent Agent Technology.* IEEE, 2010, pp. 40-47 71

71