Robot Learning Simultaneously a Task and How to Interpret Human Instructions

Jonathan Grizou
Flowers Team
INRIA / ENSTA-Paristech
France
jonathan.grizou@inria.fr

Manuel Lopes
Flowers Team
INRIA / ENSTA-Paristech
France
manuel.lopes@inria.fr

Pierre-Yves Oudeyer
Flowers Team
INRIA / ENSTA-Paristech
France
pierre-yves.oudeyer@inria.fr

***Abstract*—This paper presents an algorithm to bootstrap shared understanding in a human-robot interaction scenario where the user teaches a robot a new task using teaching instructions yet unknown to it. In such cases, the robot needs to estimate simultaneously what the task is and the associated meaning of instructions received from the user. For this work, we consider a scenario where a human teacher uses initially unknown spoken words, whose associated unknown meaning is either a feedback (good/bad) or a guidance (go left, right, ...). We present computational results, within an inverse reinforcement learning framework, showing that a) it is possible to learn the meaning of unknown and noisy teaching instructions, as well as a new task at the same time, b) it is possible to reuse the acquired knowledge about instructions for learning new tasks, and c) even if the robot initially knows some of the instructions' meanings, the use of extra unknown teaching instructions improves learning efficiency.**

I. Introduction

Robots are becoming increasingly important, targeting human assistance at home or at the workplace. Yet, such robots can not be pre-programmed to face every day problems in our open ended and dynamic environments. This challenge requires to develop learning algorithm for the robot to adapt to its environment. Among other forms of adaptation to the environment, *social learning*, where knowledge is transmitted from humans to robots through social interaction, is of primordial importance. It has the advantage of being an intuitive way for humans to instruct robots. A usual assumption in such systems is that the learner and the teacher share a mutual understanding of the meaning of each others' signals, and in particular the robot is usually assumed to know how to interpret teaching instructions from the human. In practice, the range of accepted instructions is limited to the one predifined by the system developer. However non-expert users might have very different preferences and predefined instructions might not be well accepted. We believe that robots should themselves be able to adapt progressively to every user's particular teaching behaviors at the same time as they learn new skills.

Research in robotics has long been inspired by human social learning. Among other aspects, learning by demonstration/imitation has attracted most attention. It has provided several examples of efficient learning in robotic systems [1][2]. Data from a human teacher has been used as initial condition for further self-exploration in robotics [3], bootstrapping further intrinsically motivated learning [4], information about the task solution [5], information about the task representation [6], among others. Several representations have been used to generalize the demonstration data using reinforcement learning [7], inverse reinforcement learning [6][8], or regression methods [5][9]. The different formalisms make use of various kinds of information and extract different knowledge, either direct policy information or a reward function that explains the behavior.

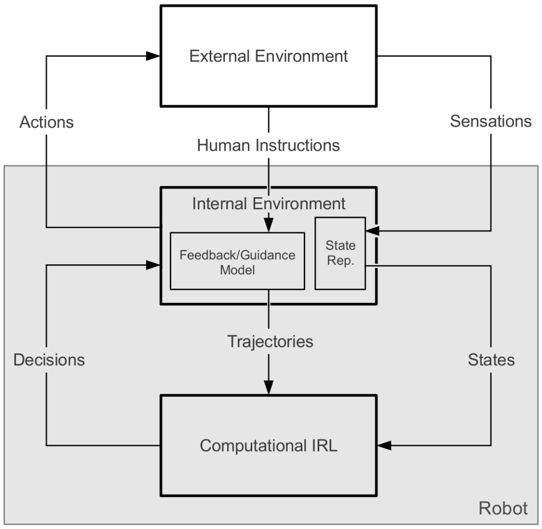

Fig. 1. Reinforcement learning oriented architecture of our problem. Humans provide instructions for learning a task whose meanings are a priori unknown. Thus, the meaning of these instructions has to be learnt by the robot in addition to learning the task itself.

For most of those systems, the human demonstrations are provided in a batch perspective where data acquisition is done before the learning phase. Recently it has been suggested that *interactive learning* [3][10] might be a new perspective on robot learning, that combines the ideas of learning by demonstration, learning by exploration and tutor feedback. Under this approach the teacher interacts with the robot and provides extra feedback or guidance. In addition, the robot can act to improve its learning efficiency. Approaches have considered: extra reinforcement signals [7], action requests [11], disambiguation among actions [9], preferences among states [12], iterations between practice and user feedback sessions [13], and choosing actions that maximize the user feedback [14].

An important challenge for such interactive systems is to deal with nonexpert humans. Several studies discuss the various behaviors naive teachers use when instructing robots [7], [15]. An important aspect is that the feedback is frequently ambiguous and deviates from the mathematical interpretation of a reward; or a sample trajectory deviates from an optimal policy. For instance, in the work of [7] the teachers frequently gave a positive reward for exploratory actions even if the signal was used by the learner as a standard reward. Also, even if we can define an optimal teaching sequence, humans do not necessarily behave according to those strategies [15].

In addition, users may have various expectations and preferences when interacting with a robot; therefore predifined protocols or teaching signals may bother the user and dramatically decrease the performance of the learning system [16]. In this paper, we present an algorithm allowing a robot to learn the meaning of human teaching instructions in the process of learning a task (as illustrated in figure 1). Importantly, the system does not need bootstraping with known instructions, but only requires knowledge about the possible structures of meanings and tasks. The learnt instruction-to-meaning association can then be reused in the learning of novel tasks, progressively increasing the knowledge of the robot. We will also show that, by combining known and unknown teaching signals, the robot is able to take advantages of unknown instructions to learn more efficiently than by relying only on known ones. We do not claim that we should not rely on predefined signals but rather that the feedback or guidance provided through predefined protocols could be completed with the particular teaching signals that each user provides.

We extend the work presented in [17], which introduced a preliminary approach to this problem considering an abstract symbolic space of instructions in simulation. Here, we allow the robot to learn the meaning of unknown instructions without the need of bootstrapping the system with known instructions and by considering real natural speech waves data instead of symbolic labels, as well as a physical human-robot interaction scenario. In [18], the robot ASIMO is taught to associate new spoken signals to visual object properties, both in noisy conditions and without the need for bootstrapping. However the robot is not learning a sequential task but correlations between clusters in speech and visual spaces. Similarly Kindermans et al. [19] proposed an unsupervised training of a P300-based BCI systems using application constraints. Their formalism is close to the one described in this paper; however, our system is able to provide a confidence about its current knowledge of the task and instruction-to-meaning association.

Our algorithm differs from typical learning by demonstration systems because data is acquired in an interactive and online setting. It improves from previous learning by interaction systems in the sense that the instructions received are continuous unlabelled signals. Our framework is generic and the signals provided by the teacher can be gestures, facial expression, or any modalities as long as we can project them into a fixed length continuous representation.

Our contribution is threefold: a) we provide an online learning algorithm which makes it possible to learn the meaning of unknown and noisy instructions, as well as a new task at the same time, b) we enable the reuse of acquired knowledge about instructions for learning new tasks, and c) in the case where the robot initially knows some of the instructions meanings, extra unknown teaching signals are used to improve learning efficiency.

In Section II, we will provide details on the algorithm. The following sections present an application of this algorithm to a particular interaction scenario. We will introduce first the robotic system, the interaction protocol and the signals processing unit. Finally, we will present results from both simulations and an experiment with a real robot.

II. Algorithm

In this section, we present our computational model by considering the following cases: 1) feedback instructions 2) guidance instructions, and 3) how to include known sources of instructions

Our goal is to learn simultaneously a task ξ and the meaning of the instructions n provided by the user. We assume such instructions are represented in a fixed length feature vector with continuous values that are generated from a probabilistic model. For each particular task ξ we only assume that we are able to compute a policy π, which represents the probability of choosing a given action a in a given state s, $\pi^{\xi}(s,a) = p(a|s,\xi)$. We consider that the human-robot interaction sessions give data in the form $\{(s_i, a_i, n_i),\ i = 1, \ldots, m\}$, i.e. a sequence of states, actions and teaching signals triplets. At each iteration the robot performs one action and waits for the instruction from the teacher.

A. Learning the Instructions Meaning

We start by assuming that the teacher provides a simple binary feedback whose meaning f can be in $F = \{\texttt{correct}, \texttt{wrong}\}$. For each feedback, the user produces a signal in natural language that might be a corresponding word (e.g. *"ok", "good", "bad", "wrong"*). In this first step we want to learn the parameters θ of the signal production model:

$$\theta^* = \arg\max_{\theta}\ p(n|s,a,\xi,\theta) \tag{1}$$

This model is very difficult to identify but if we would have access to an hidden variable z, representing the meaning of the instruction that the user said, it would be simplified and represented as $p(n|z,\theta)$ where n is the signal observed. This meaning is generated according to the following model $p(z|s,a,\xi) = p(z|f)p(f|s,a,\xi)$ where $p(f|s,a,\xi)$ represents the ideal feedback for task ξ when the teacher observes action a in state s, and $p(z|f)$ consider what the user actually provided as feedback, considering the way he likes to provide it and also the mistakes he makes. The component $p(f|s,a,\xi)$ is fixed and derives directly from the task representation used. We do not assume any particular structure for $p(z|f)$ and even allow it to be different for each sample. This allows for a larger variety of teacher behaviors including the statistics of errors made on the instructions. Due to these reasons, and without lack of generality we will always refer to $p(z|s,a,\xi)$.

Due to the uncertainty in the expected meaning z, the task model ξ, variability in the feedback signals n (e.g. words are never pronounced the same way) and occasional teaching mistakes, we are not sure if each instruction produced by the teacher corresponds to the meaning `correct` or `wrong`. As we are in the presence of a hidden information problem we will rely on an *Expectation-Maximization algorithm* (EM) to solve the problem in Eq. 1.

We start by defining the complete likelihood model:

$$\begin{aligned}\mathcal{L}(\theta,\xi) &= p(n|s,a,\xi,\theta)\\ &= \prod_i \mathcal{L}_i(\theta,\xi)\end{aligned}$$

with

$$\begin{aligned}\mathcal{L}_i(\theta,\xi) &= p(n_i|s_i,a_i,\xi,\theta)\\ &= \sum_{j\in F} p(n_i|z=j,s_i,a_i,\xi,\theta)\ p(z=j|s_i,a_i,\xi,\theta)\\ &= \sum_{j\in F} \underbrace{p(n_i|z=j,\theta)}_{\text{instruction}}\ \underbrace{p(z=j|s_i,a_i,\xi)}_{\text{meaning}}\\ &= \sum_{j\in F} p(n_i|z=j,\theta)\ z^{\xi}_{ij} \qquad (2)\end{aligned}$$

with $z^{\xi}_{ij} = p(z = j|s_i, a_i, \xi)$. Where in the second step we introduce the hidden variable z as described earlier. The meaning z depends only on the state-action pair (s, a) evaluated in the scope of the task ξ. The instruction n depends solely on the signal generation model, parameterized by θ, corresponding to the meaning (i.e. the class) z. The ML estimate of θ is found by maximizing $\log\mathcal{L}$. We first perform the *expectation* step by defining the $F(\theta|\theta^t)$ function for a given task ξ:

$$\begin{aligned}F(\theta|\theta^t) &= \mathbb{E}[\log\mathcal{L}(\theta)|n,s,a,\xi,\theta^t]\\ &= \sum_i\sum_{j\in F} \log\mathcal{L}_{ij}(\theta)\ p(z=j|n_i,s_i,a_i,\xi,\theta^t)\\ &= \sum_i\sum_{j\in F} \left(\log p(n_i|z=j,\theta)+\log z^{\xi}_{ij}\right) w_{ij} \qquad (3)\end{aligned}$$

with

$$\begin{aligned}w_{ij} &= p(z=j|n_i,s_i,a_i,\xi,\theta^t)\\ &\propto p(n_i|z=j,s_i,a_i,\xi,\theta^t)\ p(z=j|s_i,a_i,\xi,\theta^t)\\ &= p(n_i|z=j,\theta^t)\ p(z=j|s_i,a_i,\xi)\end{aligned}$$

The *M-step* is the maximization of Eq. 3:

$$\theta^{t+1} = \arg\max_{\theta} F(\theta|\theta^t) \qquad (4)$$

This step depends on the specific statistical models we use for the instruction learning, i.e. the classifier. If they are modeled as gaussian distributions then the usual equations for a gaussian mixture hold and we can solve the maximization problem analytically. As for more complex interactions the instructions produced by the teacher will be more complex we will also try learning algorithm with a higher capacity, e.g. *SVMs*. If such classifier is not able to use probabilistic labels, we approximate Eq. 3 with a hard threshold for z^{ξ}_{ij} and train the SVM on the corresponding dataset. The full process is summarized in Algorithm 1.

B. The guidance case

The version presented above is well suited to learn instructions that correspond to `correct` or `wrong`. We can devise another interaction scheme where the teacher provides the names of actions to be done and the robot has to learn which action do each instruction corresponds to. We can see these instructions as a guidance signal or a voice operated remote control. We can deal with this situation by redefining the meaning of z. Now this variable indicates the name of the optimal action in state s according to task ξ. We define G as the set of guidance meanings, i.e. the name of the possible action. Under this new definition we can change the likelihood function to:

Algorithm 1 EM for learning Instructions Meaning

```
Require: Data $\{(s_i,a_i,n_i),\ i=1,\dots,m\}$
Require: Task $\xi$
while true do
    E-Step
    $F(\theta|\theta^t) = \sum_{ij}\left(\log p(n_i|z=j,\theta)+\log z^{\xi}_{ij}\right) w_{ij}$
    $w_{ij} = p(n_i|z=j,\theta^t)\ p(z=j|s_i,a_i,\xi)$
    M-Step
    $\theta^{t+1} = \arg\max_{\theta} F(\theta|\theta^t)$
end while
```

$$\begin{aligned}\mathcal{L}_i(\theta,\xi) &= p(n_i|s_i,a_i,\xi,\theta)\\ &= \sum_{j\in G} p(n_i|z=j,\theta)\ p(z=j|s_i,\xi)\\ &= \sum_{j\in G} p(n_i|z=j,\theta)\ z^{\xi}_{ij} \qquad (5)\end{aligned}$$

with $z^{\xi}_{ij} = p(z = j|s_i, \xi)$ and where we dropped the dependence on the action.

C. Learning Simultaneously a Task and Instructions Meaning

We now relax the assumption that we have an estimation of what the task is: we consider that the learner is able to sample tasks from a finite set according to a given distribution. The goal is to find, from this distribution, the task ξ^* that is closer to the one the user is teaching to the robot. At each iteration the algorithm evaluates the likelihood of every task hypothesis. For this, it needs to apply Algorithm 1 for every task hypothesis. The global process of simultaneously estimating the task and the instruction model is shown in Algorithm 2.

Algorithm 2 Learning Simultaneously a Task and Instructions Meaning

```
Require: Set of $l$ different tasks hypothesis $\xi_1,\dots,\xi_l$
$i = 1$
$s_1 \leftarrow$ current state
while true do
    Choose and apply action $a_i$
    Observe next state $s_{i+1}$ and user instructions $n_i$
    for all $k = 1,\dots,l$ do
        From Algorithm 1 find:
        $\theta_k = \arg\max_{\theta} F(\theta|\theta^0,\xi_k)$
        $q_k(\xi_k) = \mathcal{L}(\theta_k,\xi_k)$
    end for
    Resample $\xi_k,\ k=1,\dots,l$ according to $q_k(\xi_k)$
    $i \leftarrow i+1$
end while
return $q_k(\xi_k),\ \xi_k,\ k=1,\dots,l$
```

We are now simultaneously solving two optimization problems. We are trying to select the best task hypothesis and the best instruction-to-meaning mapping. We have to rely on an approximation to avoid the computation of all possible pairs of tasks and meaning models. To do so we first optimize the meaning model for each task hypothesis using Alg. 1. Then, for the list of possible tasks we compute the likelihood of the observed data to give us the posterior distribution of tasks. As there might be no feasible task, we have to use the noiseless version of the feedback model as the likelihood in Step 7 of Alg. 2.

An intuition on how the algorithm works is to imagine the agent assigning hypothetic labels (i.e. meanings) to instructions for each task of the distribution. The agent is updating as many models as task hypothesis and is looking for the one from which emerges a coherence in the interpretation of the instructions. Here, we are assuming that if the correct labels are known, signals of same meaning (e.g. utterances of the same word) can be identified with good accuracy using the chosen classifier parameterized by θ. In case of a gaussian classifier, θ represents the mean and covariance of each class. The algorithm will start failing if signals used for different meanings cannot be differentiated by the classifier, or if the classifier is overfitting the data.

The computational complexity of the algorithm grows linearly with the number of possible hypothesis and with the number of data-points. Even with such a low complexity, for some problems the number of possible hypothesis might be very large. The complexity of this algorithm can be reduced in two ways. First we can consider a reduced set of task hypothesis and apply a resample step according to the estimated likelihoods, as shown in Step 10. Another way to reduce the complexity is to consider that the dataset does not cover the whole state-space. Because of this, Step 7 does not need to be applied to all hypothesis but only to the equivalence classes of the policies induced by the hypothesis set according to the dataset.

D. *Including prior knowledge*

Although we took such a difficult challenge of learning without assuming knowledge of the instructions, for a practical case, it is more reasonable to combine pre-specified instructions with an adaptation to new ones. For instance, the robot might be equipped with a console with a simple interface such as a green and a red button corresponding to correct and incorrect feedback and we want to combine this information with unknown sources of instructions.

The use of those extra sources of information is straightforward in our statistical formalism: we can change the likelihood model from Eq. 2 and extend the model $p(z|s, a, \xi)$ with an observed variable d representing the noisy (in terms of teaching mistakes) but known feedback. The model becomes:

$$\begin{aligned} \mathcal{L}_i(\theta, \xi) &= p(n_i, d_i | s_i, a_i, \xi, \theta) \\ &= \sum_j p(n_i | z = j, \theta)\ p(d_i | z = j)\ p(z = j | s_i, a_i, \xi) \end{aligned} \tag{6}$$

In this way we still accept that the human does not use the pre-define interface or that it makes mistakes. In the former case we just assume $p(d|z) = 1$ and we recover the original likelihood function, in the latter the complete rule will take the noise into account. Identically, if the user does not provide any known instructions, we just assume $p(n|z, \theta) = 1$.

III. Experiments and Results

In this section we present results from our algorithm both in simulation and with a real robotic system. We test different aspects of our algorithm: a) learning the associated meaning of feedback instruction words while learning a new task, b) extending it for the case of guidance words, c) combining learning from unknown instructions with pre-defined signals of known meanings, and d) reusing learnt instruction-to-meaning mapping for the learning of a new task.

A. *Experimental System*

We construct a small size pick-and-place task with a real robot. This robot is going to be programmed using a natural speech interface whose words have an unknown meaning and are **not** transformed into symbols via a voice recognizer. The robot has a prior knowledge about the distribution of possible tasks.The interaction between the robot and the human is a turn taking social behavior: the robot performs an action and waits for a feedback, or guidance, instruction to continue. This allows to synchronize a speech wave with its corresponding pair of state and action. The experimental protocol is summarized in figure 2.

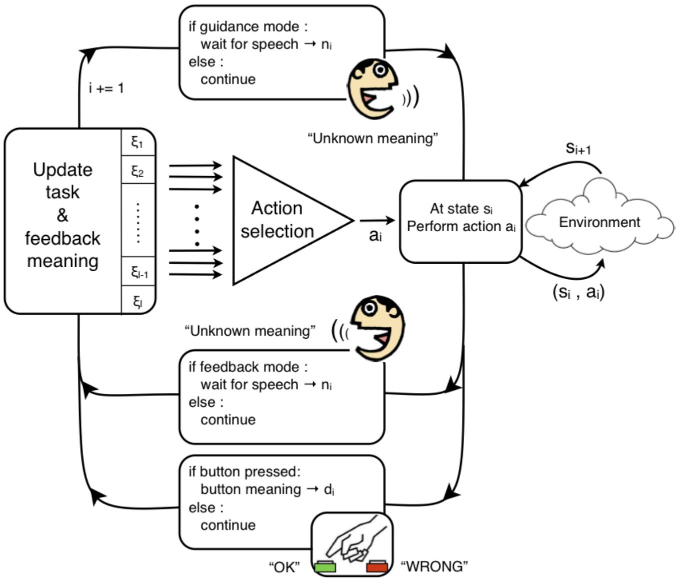

Fig. 2. Experimental protocol showing the interaction between the teacher and the learning agent. The agent has to learn a task and the meaning of the instructions signals provided by the user, here recorded speech. The teacher can use guidance or feedback instructions but also has access to buttons of known meaning for the robot.

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

2) Task Representation: We assume that for a particular task ξ we are able to compute a policy π representing the

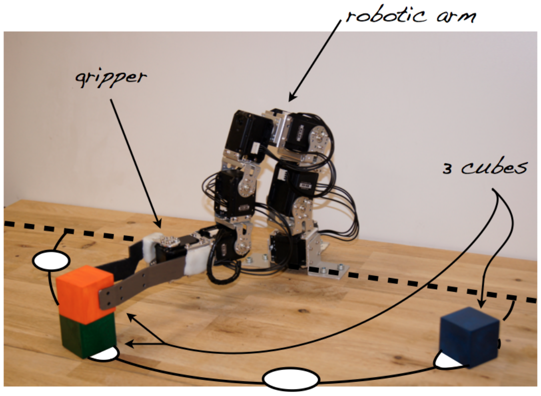

Fig. 3. Robotic System. A six d.o.f robotic arm and gripper learning to perform a pick-and-place task with three cubes.

optimal actions to perform in every state. One possibility is to use *Markov Decision Processes* (MDP) to represent the problem [20]. From a given task ξ represented as a reward function we can compute the corresponding policy using, for instance, *Value Iteration* [20]. In any case, our algorithm does not make any assumption about how tasks are represented.

For this particular representation we assume that the reward function is sparse and so we can generate possible tasks by sampling sparse reward functions. Similarly to *Bayesian Inverse Reinforcement Learning* [21] the robot learns the task by choosing among the possible space of rewards the most likely one. We approximate this process using a finite set of task hypothesis representing all the reward functions consisting of a unitary reward in one state and no reward in all the others. In other words the task is to reach one, yet unknown, of the 624 states of the MDP.

Under this formalism the action selection at runtime can be done in different ways. As different sampling methods can lead to different learning behaviors, we will compare two different methods: random and ϵ-greedy. When following random action selection the robot does not use its current knowledge of the task and randomly selects actions. Whereas with ϵ-greedy method the robot performs actions according to the current belief of what the task is, i.e. following the policy corresponding to the most likely task hypothesis. The corresponding optimal action is chosen with $1-\epsilon$ probability, otherwise, a random one is selected. In our experiment we show only results with $\epsilon = 0.1$.

3) Speech processing: As mentioned before, we consider speech as the modality for interacting with the robot. After each action we record the teaching word pronounced by the user. This data is mapped into a 20 dimensional feature space using the methodology described below.

A classical method for representing sounds is the *Mel-Frequency Cepstral Coefficients* (MFCC) [22]. It represents a sound as a time sequence MFCC vectors of dimension 12. Comparing sounds is done via *Dynamic Time Warping* (DTW) between two sequences of feature vectors [23]. This distance is a measure of similarity that takes into account possible insertions and deletions in the feature sequence and is adapted for sounds comparison of different length. Each recorded vocal signal is represented as its DTW distance to a base of 20 pre-defined spoken words which are **not** part of words used by the teacher.

By empirical testing on recorded speech samples, we estimate that a number of 20 bases words were sufficient and yet a relatively high number of dimensions to deal with a variety of people and speech. This base of 20 spoken words has been randomly sampled from a scientific book[1] and is composed of the words: *Error, Acquisition, Difficulties, Semantic, Track, Computer, Explored, Distribution, Century, Reinforcement, Almost, Language, Alone, Kinds, Humans, Axons, Primitives, Vision, Nature, Building*.

4) Classification System for the Instruction Model: As explained in Section II, any standard machine learning classifier can be used to approximate the instruction model. If such classifier is not able to use probabilistic labels then the maximization step of the EM algorithm is approximated in Eq. 3 with a hard thresholds for z_{ij}^{ξ}. We then have to rely on the generalization performances of the classifier. Indeed, if the classification algorithm is overfitting the data then no difference can be found between the hypotheses. The only required characteristic is the ability to output a confidence on the class prediction, i.e. a probability for n_i of being associated to each meaning.

In this study we decided to compare three classifiers for the instruction learning, i.e. modeling $p(n|z,\theta)$:

- Gaussian Bayesian Classifier: Computing the weighted mean μ and covariance matrix Σ, the usual equations for gaussian mixture hold.
- Support Vector Machine (SVM): Using a RBF kernel with $\sigma = 1000$ (high dimensional space) and $C = 0.1$. For SVM probabilistic prediction refer to [24].
- Logistic regression: The predictive output value ([0,1]) is used as a measure of confidence. This algorithm is usually not well suited for high dimensional spaces because of the curse of dimensionality.

B. Experimental Results

Experiments presented in this section follow the protocol described in figure 2, where at each iteration the agent performs one action and waits for the instruction from the teacher. We first present a set of simulated experiments using the same MDP as for the real word experiment. We start by assuming that the teacher provides feedback instructions without any mistake, therefore only the variability in the signals remains. We first compare the different classifiers and then the performances of ϵ-greedy versus random action selection methods both for feedback and guidance modes. Later, we present an analysis of robustness to teacher mistakes. Last simulated experiment studies the case where the teacher has also access to buttons of known meaning. Finally, we show results using a real robot where we study how instructions knowledge learned in a first run can be used in a second one to learn more efficiently.

In order to be able to compute statistically significant results for the learning algorithm, we created a database of speech signals that can be used in simulated experiments. This database allows to test our system with realistic continuous features while controlling the behavior of the teacher, e.g. by varying the amount teaching mistake. All results report

[1] RA Wilson, FC Keil, "The MIT encyclopedia of cognitive science", 2001

averages of 20 executions of the algorithm with different start and goal states. By normalizing the sum of all likelihoods estimate $(q_1, \ldots, q_l)$ to 1, we obtain the probability of each particular task hypothesis to represent the task to learn. The normalized likelihood of the task to be learned $q(\xi^*)$ is our measure of learning progress.

1) Learning feedback instructions: In this experiment we assume that the robot does not know the words being spoken by the teacher and it does not know the task either. The teacher is providing instruction of meaning being either `correct` or `wrong`. The robot will, simultaneously, learn the task and map the words that is recorded into a binary feedback signal.

The results comparing the different classification methods are shown in Figure 4. Action selection is done ϵ-greedy. Note that after 200 iterations all three methods have learned the task, i.e. the normalized goal likelihood value is greater than 0.5, meaning that the sum of all the others is inferior to 0.5. Logistic regression provides the worse results in terms of convergence rate and variance.

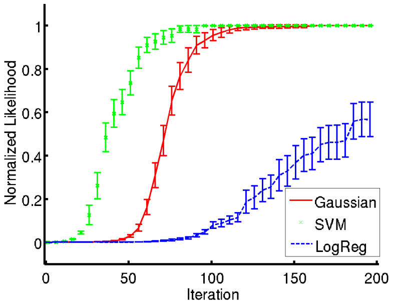

Fig. 4. Taught hypothesis normalized likelihood evolution (mean + std error) thought iteration using different kinds of classifiers. The teacher is providing feedback using one word per meaning and the agent is performing action according to ϵ-greedy strategy.

The user is not restricted to the use of one word per meaning, table I compares the goal normalized likelihood value after 100 iterations for feedback instructions composed of one, three and six spoken words per meaning. SVM has better performance when using one word per meaning but the Gaussian classifier has overall better results with less variance, see Table I. Interestingly the Gaussian classifier learns better than the other classifiers with many words per meaning. This counter intuitive result can be explain by the high dimensionality of the space where even one gaussian can differentiate several group of clusters. As expected logistic regression performs badly due to the high dimensionality of the space. For the SVM classifier, the small number of points in each cluster is probably affecting the performances. For the following experiments, we will only consider the gaussian classifier, first because it has overall better performance but also because it is by far the faster to train and thus is the only one usable for real world and real time experiments. Indeed, in this setup, at each iteration the agent has to train 624 classifiers.

TABLE I. TAUGHT HYPOTHESIS NORMALIZED LIKELIHOOD VALUES AFTER 100 ITERATIONS. COMPARISON FOR DIFFERENT CLASSIFIERS AND NUMBER OF WORDS PER MEANING. THE GAUSSIAN CLASSIFIER HAS OVERALL BETTER PERFORMANCES.

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

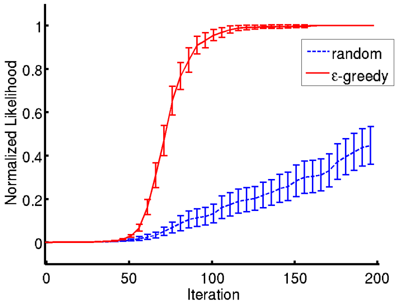

Fig. 5. Taught hypothesis normalized likelihood evolution (mean + std error) thought iteration using gaussian classifier. The teacher is providing feedback using one word per meaning. The ϵ-greedy action selection method learns faster than the random one.

We will now compare the impact of using different action selection methods. From Figure 5 we can observe that ϵ-greedy results in a faster learning with less variance. This method, at each step, leads the robot in the direction of the most probable goal. In this way it will receive more diverse feedback and will visit more relevant states than what a simple random exploration would do.

2) Learning guidance instructions: Figure 6, shows results where the teacher provides guidance instead of feedback. The number of meanings is increased from two (correct/wrong) to four (left/right/grasp/release). At each iteration the teacher first says the name of the optimal action to be performed by the robot, which then performs one action. Changes in the algorithm are described in Eq. 5. As with feedback, the robot is able to learn the task based on guidance instructions but need more iterations to reach a perfect knowledge. Indeed, even if the robot receives more informative instructions, it now needs to classify instructions in four meanings which requires more samples to identify the clusters.

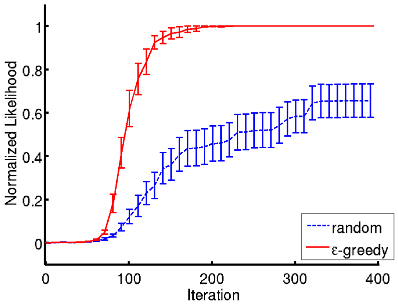

Fig. 6. Taught hypothesis normalized likelihood evolution (mean + std error) thought iteration using gaussian classifier. The teacher is providing guidance using one word per action name. The ϵ-greedy action selection method learns faster than the random one.

3) Robustness to teaching mistakes: In results presented until now, we made the assumption that the teacher is providing feedback or guidance instructions without any mistake. But real world interactions are not perfect and people can fail in providing correct feedback. An analysis of robustness is shown in figure 7 using feedback instructions, gaussian classifier and one word per meaning. Results with and without EM are compared to study if EM is improving robustness to teaching mistakes.

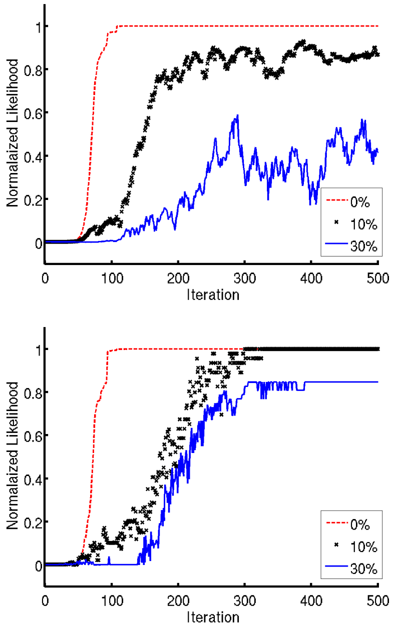

Fig. 7. Taught hypothesis normalized likelihood evolution thought iteration using gaussian classifier. Comparison of one step EM (top) versus full EM (bottom). The teacher is providing feedback using one word per meaning with different percentage of mistakes. ϵ-greedy action selection. Standard error has been omitted for readability reason.

We can observe that full EM is performing as expected and enables the agent to learn the task faster facing teaching mistakes.

4) Including prior information: Learning purely from unknown instructions is challenging for the researcher but could be restrictive for the teacher. Therefore sources of known feedback could be added, such as a green and red button, where the green button has a predefined association with a `correct` feedback meaning, as red button with a `wrong` meaning. Yet, we shall expect that even in this case, users will use more modalities than the predefined one. In this study, the teacher still provides initially unknown spoken words feedback but can also use the red and green button as described in figure 2. However, and in order to avoid possibility of direct button to instruction association, it can never use both modalities at the same time and use them alternatively with equal probability. Therefore, in average after 250 iterations the robot has received 125 known feedback and 125 unknown speech signals. This setting assumes that more information is received by the robot than the one predefined by the developer. In most systems this information is ignored but we think robots could also try learning from such unknown signals. We study three learning methods: in the first case, the robot is learning only via the known feedback, i.e. the buttons; in the second it uses only the vocal unknown signal; and in the third one, it uses both. Figure 8 shows result from this setting.

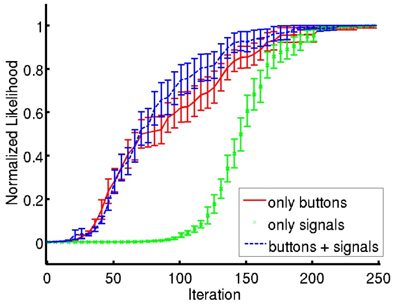

Fig. 8. Taught hypothesis normalized likelihood evolution (mean + std error) thought iteration using gaussian classifier. Comparison of using known, unknown signals and both.

As expected learning from known feedback is faster than with unknown, however taking advantage of different sources of information, even a priori unknown, can lead to slightly better performances than using only known information. Importantly, the instructions to meaning knowledge of the robot is updated and could therefore be reuse in further interaction.

5) Using a real robot: Statistical simulations have shown that our algorithm allows an agent to learn a task from unknown feedback in a limited amount of interactions. To bridge the gap of simulation we tested our algorithm in real interaction condition with our robotic arm. In this real experiment, the teacher is facing the robot and chooses a specific goal to reach (i.e. a specific arrangement of cube it wants the robot to build). It then decides one word to use as positive feedback and one as negative feedback and starts to teach the robot. For this experiment the word *'yes'* and *'no'* were respectively used for the meaning `correct` and `wrong`. Once this experiment is terminated we keep in memory the classifier corresponding to the best task, i.e. having the higher likelihood value, and start a new experiment where the human teacher is going to use the same feedback instructions to teach a new task. But this time the spoken words are first classified as of correct or wrong meaning according to the previously learnt classifier. Therefore standard IRL algorithm can be used. We study here two things, first does our system bridges the reality gap and can we reuse information learnt from a previous experience?

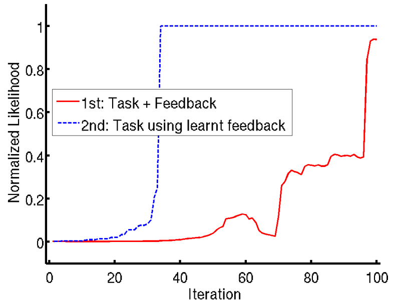

Fig. 9. Taught hypothesis normalized likelihood evolution thought iteration using gaussian classifier. Feedback using one word per action. ϵ-greedy action selection. A first run of 100 iterations is performed where the robot learns a task from unknown feedback. Then by freezing the classifier corresponding to the best task estimate, the user teaches the robot a new task.

Figure 9 shows results from this setting. In the first run it takes about 100 iterations for the robot to learn the task.

Whereas in the second run, when reusing knowledge from the first one, the robot is able to learn a new task faster, in about 30 iterations, meaning that it has well found the two clusters in our $\mathbb{R}^{20}$ dimensional space as well as the mapping to their corresponding meanings.

IV. Conclusions, Limitations And Future Work

In this work, we presented an interactive learning system that can learn the meaning of instructions while learning a new task. We considered the case of spoken words but of particular interest is the possibility to use the same system with other modalities, such as facial expressions or hand gestures. This allows different users to use the system according to their own preferences, skills, and limitations. We tested our experiment on a real robot and showed that knowledge acquired from a first experiment can be reused later as a source of known information.

Our approach assumes that the robot is equipped with planning skills and can not be used if several hypothesis are fully symmetric as they will not be distinguishable. This problem can be solved by redefining the set of hypothesis, for instance by adding a ”stop” action valid only at the goal states.

In order to make the learning problem tractable, we assumed that the robot had access to a predefined set of tasks. The robot will then find the hypothesis that best approximates the true one. We could extend this and follow a particle filter like approach to be able to generate new hypothesis online and potentially find a better one.

In the future we will study how to extend the proposed approach to more complex scenarios, e.g. how it scales to continuous domain. We will also consider how more complex instructions can be included in our formalism since the teaching models used spontaneously by people can be more complex than the simple meaning correspondences we assumed [7], [15]. Also the protocol could be enhanced to be more natural, the robot could ask questions [25] and accept asynchronous signals. An important aspect is to allow the user to teach the robot new macro-actions or macro-states and a first approach for that problem is to use the options framework [26].

Acknowledgement

The authors would like to thanks Pierre Rouanet for his useful comments, as well as Jérôme Bechu for his help with the robotic platform. Work (partially) supported by INRIA, Conseil Régional d'Aquitaine and the ERC grant EXPLORERS 24007.

References

[1] B. Argall, S. Chernova, and M. Veloso, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, 2009.

[2] M. Lopes, F. Melo, L. Montesano, and J. Santos-Victor, “Abstraction levels for robotic imitation: Overview and computational approaches,” in *From Motor to Interaction Learning in Robots*, ser. Studies in Computational Intelligence, O. Sigaud and J. Peters, Eds. Springer, 2010, vol. 264, pp. 313–355.

[3] M. Nicolescu and M. Mataric, “Natural methods for robot task learning: Instructive demonstrations, generalization and practice,” in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, 2003, pp. 241–248.

[4] S. Nguyen, A. Baranes, and P. Oudeyer, “Bootstrapping intrinsically motivated learning with human demonstration,” in *IEEE International Conference on Development and Learning (ICDL)*. IEEE, 2011.

[5] S. Calinon, F. Guenter, and A. Billard, “On learning, representing and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 2007.

[6] M. Lopes, F. S. Melo, and L. Montesano, “Affordance-based imitation learning in robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07)*, USA, Nov 2007, pp. 1015–1021.

[7] A. L. Thomaz and C. Breazeal, “Teachable robots: Understanding human teaching behavior to build more effective robot learners,” *Artificial Intelligence Journal*, vol. 172, pp. 716–737, 2008.

[8] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, 2004, pp. 1–8.

[9] S. Chernova and M. Veloso, “Interactive policy learning through confidence-based autonomy,” *J. Artificial Intelligence Research*, 2009.

[10] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, J. Lieberman, H. Lee, A. L. Thomaz, and D. Mulanda., “Tutelage and collaboration for humanoid robots,” *International Journal of Humanoid Robotics*, 2004.

[11] M. Lopes, F. S. Melo, and L. Montesano, “Active learning for reward estimation in inverse reinforcement learning,” in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ser. ECML PKDD '09, 2009, pp. 31–46.

[12] M. Mason and M. Lopes, “Robot self-initiative and personalization by learning through repeated interactions,” in *6th ACM/IEEE International Conference on Human-Robot Interaction (HRI'11)*, 2011.

[13] K. Judah, S. Roy, A. Fern, and T. Dietterich, “Reinforcement learning via practice and critique advice,” in *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010.

[14] W. Knox and P. Stone, “Interactively shaping agents via human reinforcement: The tamer framework,” in *Proceedings of the fifth international conference on Knowledge capture*. ACM, 2009, pp. 9–16.

[15] M. Cakmak and A. Thomaz, “Optimality of human teachers for robot learners,” in *Proceedings of the International Conference on Development and Learning (ICDL)*, 2010.

[16] P. Rouanet, P.-Y. Oudeyer, F. Danieau, and D. Filliat, “The impact of human–robot interfaces on the learning of visual objects,” 2013.

[17] M. Lopes, T. Cederborg, and P.-Y. Oudeyer, “Simultaneous acquisition of task and feedback models,” in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2, aug. 2011, pp. 1 –7.

[18] M. Heckmann *et al.*, “Teaching a humanoid robot: Headset-free speech interaction for audio-visual association learning,” in *Robot and Human Interactive Communication, RO-MAN*. IEEE, 2009, pp. 422–427.

[19] P.-J. Kindermans, D. Verstraeten, and B. Schrauwen, “A bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI.” *PloS one*, vol. 7, no. 4, p. e33758, Jan. 2012.

[20] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998, vol. 28.

[21] D. Ramachandran and E. Amir, “Bayesian inverse reinforcement learning,” in *20th Int. Joint Conf. Artificial Intelligence*, India, 2007.

[22] F. Zheng, G. Zhang, and Z. Song, “Comparison of different implementations of mfcc,” *Journal of Computer Science and Technology*, 2001.

[23] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.

[24] J. Platt *et al.*, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[25] M. Cakmak and A. Thomaz, “Designing robot learners that ask good questions,” *7th ACM/IEE Int. Conf. on Human-Robot Interaction*, 2012.

[26] R. Sutton, D. Precup, and S. Singh, “Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning,” *Artificial intelligence*, vol. 112, no. 1, pp. 181–211, 1999.