Interactive Learning from Unlabeled Instructions

Jonathan Grizou
Inria Bordeaux Sud-Ouest, France
jonathan.grizou@inria.fr

Iñaki Iturrate
CNBI, EPFL, Switzerland
inaki.iturrate@epfl.ch

Luis Montesano
I3A, Univ. of Zaragoza, Spain
montesano@unizar.es

Pierre-Yves Oudeyer
Inria Bordeaux Sud-Ouest, France
pierre-yves.oudeyer@inria.fr

Manuel Lopes
Inria Bordeaux Sud-Ouest, France
manuel.lopes@inria.fr

Abstract

Interactive learning deals with the problem of learning and solving tasks using human instructions. It is common in human-robot interaction, tutoring systems, and in human-computer interfaces such as brain-computer ones. In most cases, learning these tasks is possible because the signals are predefined or an ad-hoc calibration procedure allows to map signals to specific meanings. In this paper, we address the problem of simultaneously solving a task under human feedback and learning the associated meanings of the feedback signals. This has important practical application since the user can start controlling a device from scratch, without the need of an expert to define the meaning of signals or carrying out a calibration phase. The paper proposes an algorithm that simultaneously assign meanings to signals while solving a sequential task under the assumption that both, human and machine, share the same a priori on the possible instruction meanings and the possible tasks. Furthermore, we show using synthetic and real EEG data from a brain-computer interface that taking into account the uncertainty of the task and the signal is necessary for the machine to actively plan how to solve the task efficiently.

1 INTRODUCTION

Interactive learning [1, 2] aims at developing systems that can learn by practical interaction with the user and finds applications in a wide range of fields such as human-robot interaction, tutoring systems or human-machine interfaces. This type of learning combines ideas of learning from demonstration [3], learning by exploration [4] and tutor feedback [5]. Under this approach the human teacher interacts with the machine and provides extra feedback or guidance. Approaches have considered: extra reinforcement signals [6], action requests [7], disambiguation among actions [8], preferences among states [9], iterations between practice and user feedback sessions [10], and choosing actions that maximize the user feedback [11].

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 learning agent is usually assumed to know how to interpret teaching instructions from the human. In practice, this problem is solved due to two simplifications. On the one hand, the range of accepted instructions is limited to those predefined by the system developer. This approach, commonly used in human-robot interaction, lacks flexibility and adaptation to user specificities and, consequently, may not be well accepted by non-experts users with different preferences. On the other hand, sometimes it is not enough to predefine the instruction sets and it is necessary to perform a calibration phase to map raw signals such as speech or brain activity to their meanings. This is usually done using an ad-hoc protocol to collect labeled samples of the user instruction signals. This process must be well controlled to ensure signals are associated to the true intended meaning of the user.

The previous engineering solution is needed due to the chicken egg nature of the problem. In order to teach a system a new skill, it needs to understand the human instructions. And, in order to understand this feedback, the system must have some interaction with the human (e.g. through a controlled task as done in the calibration process) to learn what the instructions mean. Few works have studied and developed interactive learning systems that can learn both the meaning of signals and the task simultaneously. In human-robot interaction Griffiths et al. [12] conducted an experiment with humans learning the meaning of unknown symbolic teaching signals. Lopes et al. [13] presented sequential task experiments considering symbolic teaching signals and requiring a bootstrap with known signals. Grizou et al. [14] extended their system for non-symbolic teaching signals while removing the need for bootstrapping with known signals. Which they later extended to noninvasive brain-computer interfaces (BCIs), proposed an uncertainty measure on both the task and the signal model for efficient planning, and performed online experiments [15]. Also, for P300 spellers, Kindermans et al. have shown that it is possible to exploit the repetition of signals [16] together with prior information (language models, information from other subjects) [17] to calibrate the EEG decoder while using the speller. They exploit the particular fact that only one event out of six encodes a P300 potential in the speller paradigm. BCIs usually require user-dependent calibration and have to deal with the EEG brain signals non-stationarities. These facts, together with the poor signal-to-noise ratio of the EEG, make the EEG self-calibration one of the most challenging ones.

This paper aims at solving the general problem of developing machines that can execute a task from human instructions and simultaneously learn the communicative signals. Our approach is based on a discretization of the possible tasks into a finite number. Each task assigns different expected meanings to the instruction provided by the user. The machine solves the most likely task according to a pseudo-likelihood function computed using the corresponding task labels. The experimental results, both synthetic and based on real EEG data, show that in order to simultaneously recover the meanings and solve the task it is of paramount importance to take into account the uncertainty on both task and signal space.

Compared to the work of Grizou et al. [14, 15], we improve the algorithm formalism for both learning and planning, the robustness to noisy high-dimensional signals (e.g. EEG), and allow to seamlessly transition from task to task without changing the algorithm paradigm. Grizou at al. methods in [14] and [15] required a different set of equations for the first task than for the further ones where only a fixed classifier, common for all hypothesis, was used. Compared to the work of Kindermans et al. [16, 17], our approach is more general and do not need to rely on specific patterns in the signal occurrences, i.e. they exploit the fact that only one event out of six encodes a P300 potential in the speller paradigm. The setup considered in this paper can not guarantee a specific ratio of meanings between received feedback signals.

In the following section, we present the set of assumptions and algorithmic details of our system. Then we introduce the specificity of the uncertainty inherent to our problem using an intuitive example and present the details of our action selection method. Finally we present a set of simulated experiment showing that a) our action selection method is reliable and improve over other methods, b) our algorithm scale to the use of high dimensional signals coming from previously recorded brain signals, and c) by being operational from the first step, as opposed to calibration procedure, we can estimate the correct task as soon as sufficient evidence has been collected.

2 ALGORITHM

2.1 Problem definition

We consider interaction sessions where a machine can perform discrete actions from a set of available actions $a \in \mathcal{A}$ in an either discrete or continuous state space $s \in \mathcal{S}$. The user, that wants to achieve a task $\hat{\xi}$, is providing feedback to the machine using some specific signal e, represented as a feature vector. The task is sequential meaning it is completed by performing a sequence of actions. The machine ignores the task the user has in mind, as well as the actual meaning of each user's signal. Its objective is to simultaneously solve the task and learn a model for the user's signals. To achieve this, it has access to a sequence of triplets in the form $D_M = \{(s_i, a_i, e_i),\ i = 1, \ldots, M\}$, where s_i, a_i and e_i represent, respectively, the state, action and instruction signals at time step i. The behavior of the machine is determined by the actions $a \in \mathcal{A}$ and the corresponding transition model $p(s' \mid s, a)$.

We make the following assumptions under this general paradigm. First, the system has access to a set of tasks $\xi_1, \ldots, \xi_T$ which includes the task the user wants to solve. We assume the instruction signals e have a finite and discrete number of meanings $l \in \{l_1, l_2, \ldots, l_L\}$ which we call labels and this is known by the user and the machine. In this work we will consider two possible meanings for the signals: correct or incorrect; but more complex meanings could be used, such as guidance instructions (go up, go left, ...). We assume that given these labels, it is possible to compute a model that generates or classifies signals e into meanings l. The parameters of such a model will be denoted by θ and we assume this mapping between signal e and their label l to be fixed. However this mapping is unknown to the agent at start.

2.2 Estimating Tasks Likelihoods

We start by assuming we are provided a signal decoder $\hat{\theta}$ and relax this assumption later on. As mentioned in the introduction, knowing $\hat{\theta}$, we can compute the probability of each task ξ_t after observation of a signal e when performing action a in state s:

$$p(\xi_t | e, s, a, \hat{\theta}) \quad \propto \quad p(e | s, a, \hat{\theta}, \xi_t) p(\xi_t) \qquad (1)$$

where $p(e|s, a, \hat{\theta}, \xi_t)$ needs to take into account the probability of each possible meaning l given the target ξ_t, the current state s and the action a executed by the machine:

$$p(e|s, a, \hat{\theta}, \xi_t) = \sum_{k \in 1, \ldots, L} p(e|l = l_k, \hat{\theta}) p(l = l_k | s, a, \xi_t) \quad (2)$$

This process can be repeated recursively for several interaction steps i:

$$\begin{aligned}\mathcal{L}_i^{\xi_t} &= p(\xi_t | D_i^{\xi_t}, \hat{\theta}) \\ &\propto p(e_i | s_i, a_i, \hat{\theta}, \xi_t) p(\xi_t | D_{i-1}^{\xi_t}, \hat{\theta}) \qquad (3)\end{aligned}$$

with $p(\xi_t | D_0^{\xi_t}, \hat{\theta})$ being the prior at time 0 (before the experiment starts) for the task ξ_t, usually uniform over the task distribution.

We now relax the assumption we are given a model $\hat{\theta}$. The natural extension from the previous models is to compute the posterior distribution over the task and the model, $p(\xi, \theta | e, s, a)$. However, the resulting distribution does not have a close form solution even when linear Gaussian likelihoods are used due to the combination of mixtures for each possible task. Another alternative is to compute the θ and ξ that maximize the data likelihood. This is prone to fail in certain scenarios due to two reasons. First, it is common that different tasks share many labels (e.g. the policies to reach neighboring cells on a grid world are almost identical and, therefore, share most of the labels l) and results on large uncertainties in the task space that require multiple actions to be disambiguated. Second, if the signals are not well separated the meaning parameters θ of different tasks will not differ much.

For instance, under Gaussian assumptions for $p(e | l = l_k, \theta)$ and deterministic task labels $p(l = l_k | s, a, \xi)$, it is possible to integrate out θ to compute the marginal likelihood $p(D_M \mid \xi)$. The resulting likelihood depends only on the traces of each $p(e | l = l_k, \theta)$. Empirical results with synthetic and EEG data for a reaching task on a grid revealed that, when the distributions over e overlap, the traces were not enough to recover the most likely task and the corresponding meaning parameters.

To cope with these problems, we define the following pseudo-likelihood function:

$$P(D_M | \xi, \theta) \approx \prod_{i=1}^{M} p(e_i | s_i, a_i, \xi, \theta_{-i}) \qquad (4)$$

$$= \prod_{i=1}^{M} \sum_{l_c} \sum_{l} p(e_i | l_c, \theta_{-i}) p(l_c | l, \theta_{-i}) p(l | s_i, a_i, \xi) \qquad (5)$$

where l represents the meaning assigned by task ξ, action a_i and state s_i and l_c is the label corrected based on what we know about our classifier θ_{-i} for a given label l.

The pseudo-likelihood is built using a leave-one-out cross-validation strategy to evaluate the likelihood $p(e_i | s_i, a_i, \xi, \theta_{-i})$ of each signal based on the meaning parameters θ_{-i} learned for each task using all the other available signals. The use of θ_{-i} indicates we use a leave one out method. If we interpret $p(e_i | s_i, a_i, \xi, \theta_{-i})$ as a classifier, its predicted labels should match the ones provided by the task for different state-actions pairs. The rationale behind it is that for the correct task, the signals and labels will be more coherent than for other tasks, which we measure as the predictive ability of a classifier trained on the signal-label pairs. Note that wrong tasks will assign wrong labels l to the signals e, therefore the learned models will have larger overlaps (see Figure 1c).

Each term of the pseudo-likelihood is computed from three terms. $p(l | s_i, a_i, \xi)$ represents the probability distributions of the meanings according to a task, the executed action and the current state. $p(l_c | l, \theta_{-i})$ encodes which label will be actually recovered by θ_{-i}. Intuitively, it models the quality of the model θ_{-i}. $p(e_i | l_c, \theta_{-i})$ is the likelihood of the signal given the meaning. The pseudo-likelihood is maximized in two steps. First, the maximum a posteriori estimate θ_{-i} of each task is computed. Then, the term $p(l_c | l, \theta_{-i})$ is approximated by the corresponding confusion matrix of the classifier based on θ_{-i}. It is the probability that the classifier itself is reliable in its prediction. Finally, the best task ξ should be the one that maximizes the pseudo-likelihood in Eq. 4.

2.3 Decision and Task Change

The machine must decide which task is the correct one. To do so, we define W^{ξ_t} the minimum of pairwise normalized likelihood between hypothesis ξ_t and each other hypothesis:

$$W^{\xi_t} = \min_{x \,\in\, 1,\ldots,T \setminus \{t\}} \frac{P(D_M | \xi_t, \theta)}{P(D_M | \xi_t, \theta) + P(D_M | \xi_x, \theta)} \qquad (6)$$

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

Once a task is identified with confidence, the robot executes it and prepares to receive instructions from the user to execute a new task. Assuming the user starts teaching a new task using the same kind of signals, we now have much more information about the signal model. Indeed, we are confident that the user was providing instructions related to the previously identified task; therefore we can infer the true labels of the past signals. We can now assign such labels to all hypothesis and by using the same algorithm we can start learning the new task faster as all hypothesis now share a common set of signal-label pairs. The meaning models for each hypothesis are still updated step after step until the new task is identified and labels reassigned.

3 PLANNING UNDER UNCERTAINTY

To solve our problem we need to identify simultaneously the task and how to interpret teaching signals. To do so the system has to explore regions that allow to disambiguate among hypothesis. There are several efficient model-based reinforcement learning exploration methods that add an exploration bonus for states that might provide more learning gains. Several theoretical results show that these approaches allow to learn tasks efficiently [18, 19]. We define an uncertainty measure and use model-based planning to select sequences of actions that guide the agent toward states that better identify the desired task.

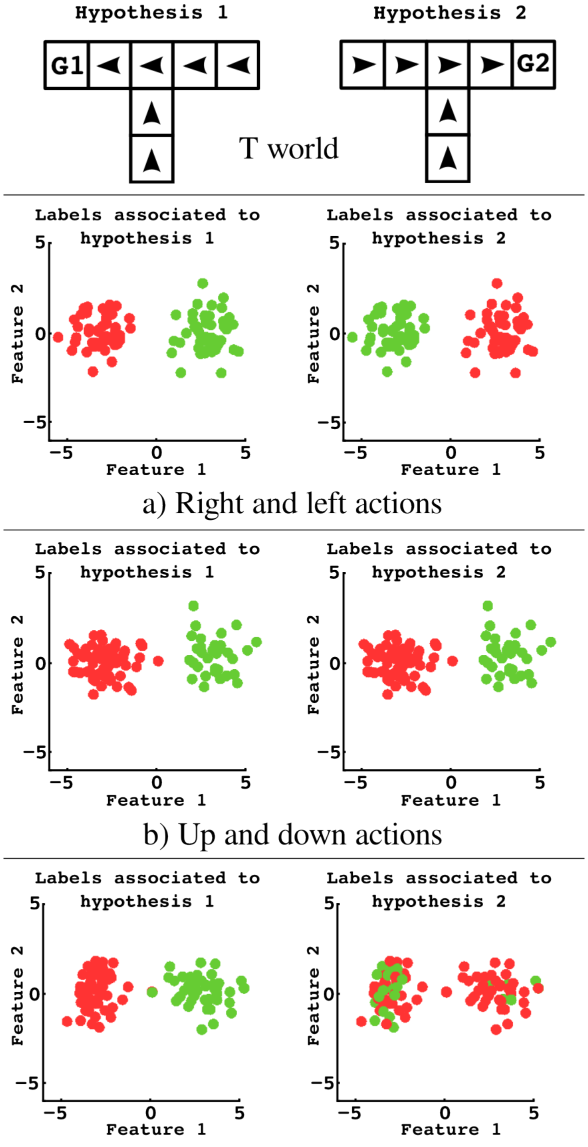
In order to exemplify the specificity of our problem in terms of planning we present a simple experiment and compare the effect of different action selection strategies. In this scenario, the agent is in a T world with 7 states and can perform 4 actions (right, left, up, and down). The user wants the robot to reach the left edge (marked by G1) of the T, (see Figure 1 top). The agent knows the users wants it to go to one of the two edges (G1 or G2) but not which one. The agent will perform some actions, and the user will assess the correctness of each agent's action by providing a two dimensional teaching signal. The agent does not known which signal means "correct" and which signal means "incorrect". As there is two possible tasks, the agent will assign labels to every user's signals according to each hypothesis. The result of the labeling process is displayed as colored dots (green for "correct"and red for "incorrect") in Figure 1 (a, b, and c), where the left part corresponds to hypothesis 1 (G1) and the right part to hypothesis 2 (G2).

If the agent knew how to interpret the signal, i.e. which signal corresponds to correct or incorrect feedback, the optimal action to differentiate between the two hypothesis would be to perform right and left actions in the top part of the T. However in our problem the classifier is not given and the agent is building a different model for each hypothesis. As a results, we end up with two opposite interpretations of the user signal, which are both as valid (see Figure 1a) and do not allow to differentiate between hypothesis.

Considering that the agent does not know the signal to meaning classifier, a sensitive option is to select actions that allow to unequivocally identify the model. In our scenario taking only up and down actions in the trunk of the T leads to identical interpretation for each hypothesis (see Figure 1b). However this method do not allow to disambiguate between hypothesis and in most setting, such as the grid world we consider later, there is no state-action pair leading to unequivocal interpretations.

However performing all the four actions allow to disambiguate between hypothesis. As shown in Figure 1c, hypothesis 1 stands out by the nice coherence between the labels and the spacial organization of the data. This informs us that hypothesis 1 is the task the user has in mind and that feedback signals in the right and left part of the feature space means "correct" and "incorrect" respectively.

For our kind of problem the agent can not just try to differentiate hypothesis by finding state-action pair where expected feedback differs but should also collect data to build a good model or at least invalidate other models. Can we



a) Right and left actions

b) Up and down actions

c) Up, down, right and left actions

Figure 1: A "T world" scenario and the interpretation results for different planning strategies. The agent knows it should reach either of the two edges of the T world (marked with the letter G). The arrows represent the optimal policy. For each move the agent receives an unlabeled two dimensional teaching signal, corresponding to user's assessments on the agent's actions. The teacher's goal is to have the agent reach G1. As the agent do not have access to this information, it interprets the signal according to each hypothesis (G1 and G2). a) shows the interpretation results if the agent only perform right and left actions in the top of the T world, b) shows the interpretation results when the agent only performs up and down actions in the trunk of the T, and c) shows the interpretation results for an agent performing all possible actions. Only the method c) allow to differentiate between hypothesis.

find a measure of uncertainty that account for both? Going back to Figure 1 (a and b), we understand that, to differentiate hypothesis in situation a) the best actions to perform are up and down in the T trunk while in situation b) the best actions to perform are right and left in the top part of the T. This corresponds to the uncertainty in the signal space. In the case of a) when going left both hypothesis agree that they will receive a signal in the right part of the feature space even if they disagree on its meaning. However for action down, both hypothesis agree they should receive a signal of meaning "incorrect" but disagree on the expected location of such signal in the feature space. In the case of b) when going up both hypothesis agree they will receive a signal in the right part of the feature space and agree on its meaning. However for action left, both hypothesis disagree about the meaning of the signal they should receive and as both share the same signal model they expect a signal in different locations of the feature space.

Estimating uncertainty in the signal space is in practice too costly as it requires to compute, for every state-action pair, the overlap between many continuous probability distributions weighted by their respective expected contribution. Following the discussion presented in previous section, we will rely on our pseudo-likelihood metric. As we cannot predict, neither control, the signal we will receive for a particular state-action, we will rely on our past history of signal and compute the expected joint probability based on previously experienced signals.

We note:

$$J^{\xi_t}(s,a,e) = \sum_{l_c}\sum_{l} p(e|l_c,\theta)p(l_c|l,\theta)p(l|s,a,\xi_t)$$

which is Eq. 5 for only one new expected observation e, so the product over iterations disappears. And $J^{\xi}(s,a,e)$ the vector $[J^{\xi_1}(s,a,e),\ldots,J^{\xi_T}(s,a,e)]$.

The uncertainty of one state-action pair given a signal e is computed as the weighted variance of the joint probability predictions with weights $W^{\xi} = [W^{\xi_1},\ldots,W^{\xi_T}]$ (see Eq. 6):

$$U(s,a|e) = weightedVariance(J^{\xi}(s,a,e), W^{\xi}) \tag{7}$$

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

$$U(s,a) \quad = \quad \int_e U(s,a|e)p(e)de \tag{8}$$

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

$$U(s,a) \quad \approx \quad \sum_e U(s,a|e)p(e) \tag{9}$$

with $p(e)$ assumed uniform.

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

This measure is then used as a classical exploration bonus method. We will switch to a pure exploitation of the task after reaching the desired confidence level.

Interestingly this approach generalizes over other active sampling method [7], if the classifier is known, equation 7 reduces to the one presented in [13] and is no longer dependent on signal e. As our uncertainty function combines uncertainty on both signal and task space, when the former is known, the latter becomes the sole source of ambiguity.

4 METHOD

In the subsequent analysis, we assume that a trainer provides feedback for the actions taken by a learner. Specifically, we consider the user is delivering signals that can be mapped into binary feedback: correct c and incorrect w.

4.1 World and Task

We consider a 5x5 grid world, where an agent can perform five different discrete actions: move up, down, left, right, or a "no move" action. The user goal is to teach the agent to reach one (unknown to the agent) of the 25 discrete positions which represent the set of possible tasks. We thus consider that the agent has access to 25 different task hypothesis (one with goal location at each of the cells). We use *Markov Decision Processes* (MDP) to represent the problem [4]. From a given task ξ, represented as a reward function, we can compute the corresponding policy π^{ξ} using, for instance, Value Iteration [4]. The policies allow us to interpret the teaching signals with respect to the interaction protocol defined. For the current work we will consider the user is providing feedback on the agent action. We define $p(l|s,a,\xi)$ as:

$$p(l|s,a,\xi) = \begin{cases} 1-\alpha & if\ a = \operatorname{argmax}_a \pi^{\xi}(s,a) \\ \alpha & \text{otherwise} \end{cases}$$

with α modeling the expected error rate of the user.

4.2 Signal properties and classifier

We aim at applying this algorithm to error-related potentials (ErrPs) for EEG-based BCI applications. These signals are generated in the user's brain after s/he assesses actions performed by an external agent [20], where correct and erroneous assessments will elicit different brain signals. Past approaches have already demonstrated that these signals can be classified online with accuracies of around 80% and translated into binary feedback, thanks to a prior calibration session that lasts for 30-40 minutes [20, 21].

Following the literature [22], we will model the signals using independent multivariate normal distributions for each class, $\mathcal{N}(\mu_c, \Sigma_c), \mathcal{N}(\mu_w, \Sigma_w)$. With θ the set of parameters $\{\mu_c, \Sigma_c, \mu_w, \Sigma_w\}$. Given the high dimensionality of the problem we will also need to regularize. For this we apply shrinkage to the covariance matrix ($\lambda = 0.5$) and compute the value of the marginal pdf function using a non-informative (Jeffreys) prior [[23], p88]:

$$p(e|l, \theta) \quad = \quad t_{n-d}(e|\mu_l, \frac{\Sigma_l(n+1)}{n(n-d)}) \tag{10}$$

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

4.3 Task Achievement

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

4.4 Evaluation scenarios

Two different evaluation scenarios were tested with two different types of signals: artificial datasets, and real ErrP datasets recorded from previous experiments [21].

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

Once the datasets were generated, two different evaluations were performed: (i) the goodness of our proposed planning strategy versus a) random action selection, b) greedy action selection, and c) a task-only uncertainty based method; (ii) the time required by the agent to learn the first task (i.e. to reach the first target), and (iii) the number of tasks that can be learned in 500 iterations.

EEG datasets Once the algorithm was evaluated with artificial datasets, we tested the feasibility of the proposed self-calibration approach using real ErrP datasets. The objective of this analysis is to study the scalability of our method to EEG data, which may have different properties than our artificial dataset.

The EEG data were recorded in a previous study [21] where participants monitored on a screen the execution of a task where a virtual device had to reach a given goal. The motion of the device could be correct (towards the goal) or erroneous (away from the goal). The subjects were asked to mentally assess the device movements as erroneous or non-erroneous. The EEG signals were recorded with a gTec system with 32 electrodes distributed according to an extended 10/20 international system with the ground on FPz and the reference on the left earlobe. The ErrP features were extracted from two fronto-central channels (FCz and Cz) within a time window of $[200, 700]$ ms (being 0 ms the action onset of the agent) and downsampled to 32 Hz. This leaded to a vector of 34 features.

Comparison with calibration methods In order to show the benefit of learning without explicit calibration, we compare our method with the standard supervised BCI calibration procedure. In this calibration procedure, which can last for up to 40 minutes, the experimenter needs to record enough data from the user from several offline runs, where the user is not controlling the agent but just passively assessing its actions. Following the literature on ErrPs [20, 21] our training data will consist of 80 percent of positive examples (associated to a correct feedback) and 20 percent of negative examples (associated to an incorrect feedback). Our proposed algorithm is compared with different (but standard) sizes of calibration datasets: 200, 300 and 400 examples.

4.5 Settings

We used $\alpha = 0.1$, $\beta = 0.9$. For dataset of dimension d, we started computing likelihoods after $d+10$ steps as equation 10 requires at least $d+1$ samples and to allow for cross validation. For the planning (Eq. 9) we selected randomly 20 signals from D_M.

5 RESULTS

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

5.1 Artificial Datasets

The first objective is to study the impact of the exploration approach proposed in Section 3. The second is to evaluate performances and robustness with respect to the dimension and the quality of each dataset.

Planning Methods Figure 2 compares the number of steps (with maximum values of 500 steps) needed to identify the first task when learning from scratch with different planning methods. Following the most probable task (i.e. going greedy) does not allow the system to explore sufficiently. On the contrary, our proposed planning method leads the system towards regions that maximize disambiguation among hypotheses. Furthermore, it also performs better than assessing uncertainty on the task space only. Given these results, the remainder of this section will only consider our planning method.

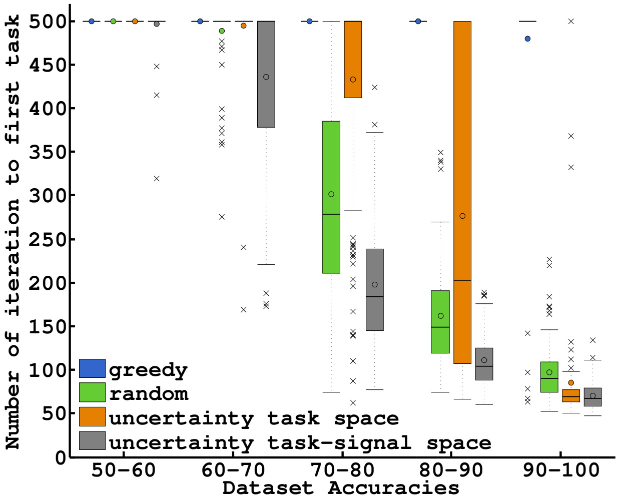


Figure 2: Number of steps to complete first task, comparison of different exploration methods with 30 dimensional artificial data. When learning from scratch, planning upon uncertainty in both task and signal space performs better than relying only on task uncertainty. Greedy action selection rarely disambiguates between hypothesis.

As depicted in Figure 1, the system needs to collect two types of information, some about the true underlying model (Fig. 1b) and some to differentiate between hypotheses (Fig. 1a). The properties of the grid world make the random strategy quite efficient at collecting those two types of information. The differences between planning methods should be more evident when navigating a complex maze since our method allows to plan in order to collect the type of information we need. Studying how different world properties affect the learning efficiency is part of our future work. Also, we note that all planning methods were switched to pure exploitation (greedy) once the confidence level was reached. Therefore the performance in Figure 2 compares the ability of the different methods to discriminate between different task hypotheses, not their ability to solve the task itself.

Dimensionality Figure 3 compares the number of steps (with maximum values of 500 steps) needed to identify the first task when learning from scratch with different dimensionality of datasets. The convergence speed is only slightly affected by the features dimensionality. On the other hand, the dataset quality (measured in terms of it associated ten-fold accuracy) is the main cause of performances decay. Furthermore, for those datasets with accuracies between 50% and 60%, the system is not able to identify a task with confidence after 500 steps.

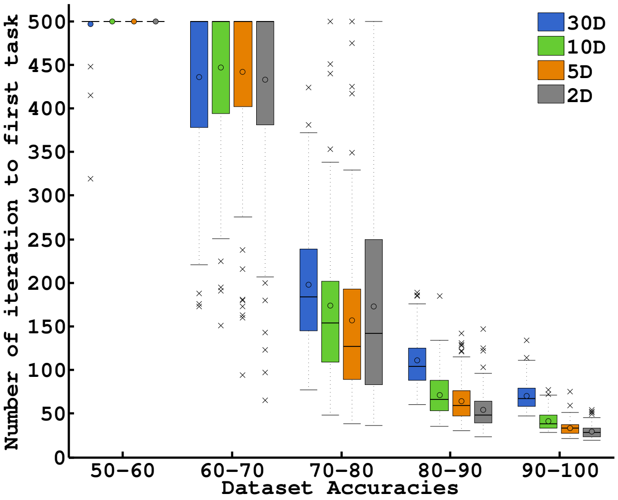


Figure 3: Number of steps to complete first task using artificial data. Under 60 percent accuracy, the confidence threshold cannot be reached in 500 steps. The dataset qualities, more than their dimensionality, impact the learning time.

Once one task is completed, a new one is selected randomly. Figure 4 compares the number of tasks that can be achieved in 500 steps. As expected, the lower the quality of the data, the less number of task can be completed. With dataset accuracies higher than 90% we can achieve more than 30 tasks on average.

An important aspect of the proposed learning approach was that the first task learned was always the correct one. We reported only 9 erroneous estimations across all simulated experiments (5 in the 70-80 group and 4 in the 80-90 group).

5.2 EEG datasets and comparison with calibration method

Example Figure 5 shows one particular run of 500 steps comparing our self-calibration method with a calibration

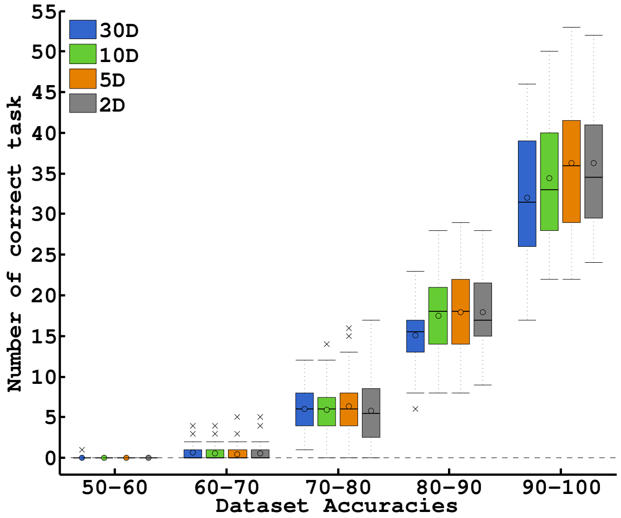


Figure 4: Number of tasks correctly achieved in 500 steps, artificial data. Quality of dataset impacts the number of task identified in 500 steps as more evidence should be collected to reach the confidence threshold.

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

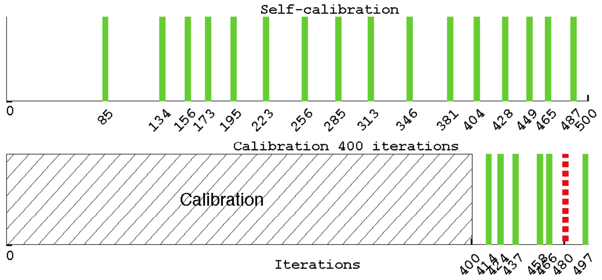


Figure 5: Time-line of one run from EEG dataset of 80 percent ten-fold classification accuracy, self-calibration (top) versus 400 steps calibration (bottom). Green (filled) and red (dashed) bars represents respectively correct and incorrect task achievement. The proposed self-calibration method allow to reach a first task faster than would take a calibration procedure.

Figure 6 shows the evolution of classification rate between our self-calibration method with a calibration procedure of 400 steps. As our method assigns different labels to each new teaching signal, the resulting classifiers have different performances, which help identifying the correct task. Once a task is identified (e.g. step 85 and 134), the corresponding labels are taken as ground truth, and all classifiers will have the same accuracies. As the agent starts exploring again to estimate the new tasks, all the classifiers except the true one will start to have worse accuracies again.

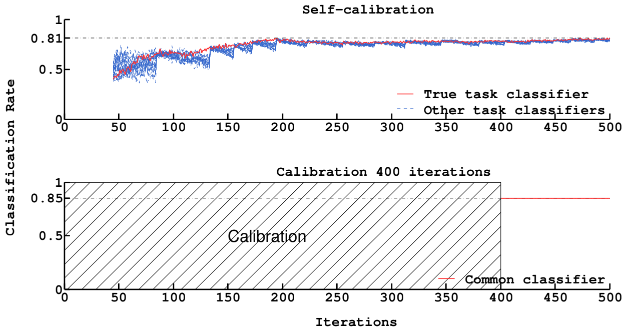


Figure 6: Evolution of classification rate of one run from EEG data, self-calibration (top) versus 400 steps calibration (bottom). On top, the red line represents the classifier corresponding to the successive tasks taught by the user, the dashed blue lines represent all others tasks. Our method updates classifiers every steps.

Time to first task Figure 7 shows the results per group of dataset. Our algorithm allows to complete the first task without errors and in a fair amount of iteration. For our method, the learning time is strongly correlated with the dataset quality. However calibration methods, which do not update their classifier once calibrated, identify more tasks incorrectly.

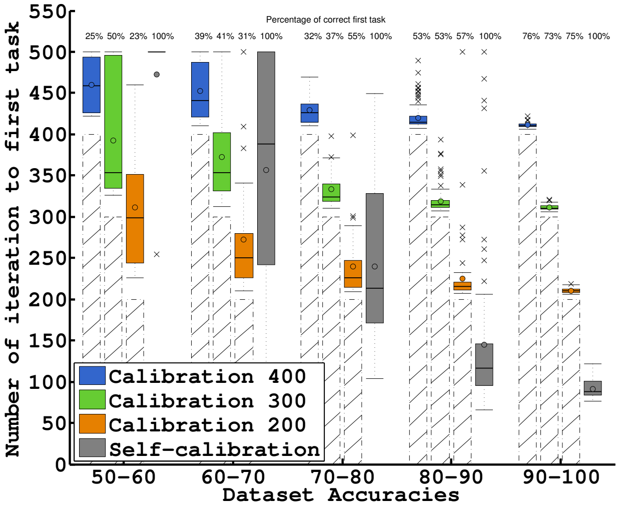


Figure 7: Number of steps to complete first task with EEG data. The method scale well to EEG data. Contrary to the standard calibration approaches, we do not make mistakes with low quality datasets.

Cumulative performances Figure 8 compares the number of tasks that can be achieved in 500 steps. With 90% and more dataset quality we can achieve about 20 tasks on average. The results are consistent with artificial dataset analysis.

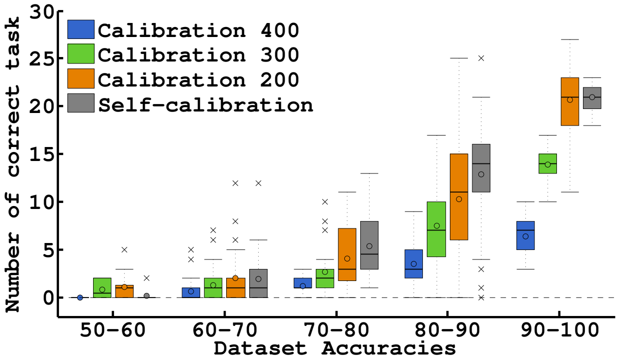


Figure 8: Number of task correctly achieved in 500 steps with EEG data. Calibration methods can not complete a significant number of task as most of the time is spent on calibration.

The calibration methods can not complete many task as a significant amount of iteration was used for calibrating the system. A calibration of 200 steps makes as many good estimation than our method, but it also makes many wrong estimation, see Figure 9. For calibration methods, the less time spent on calibration, the poorer the classifier which implies more mistakes.

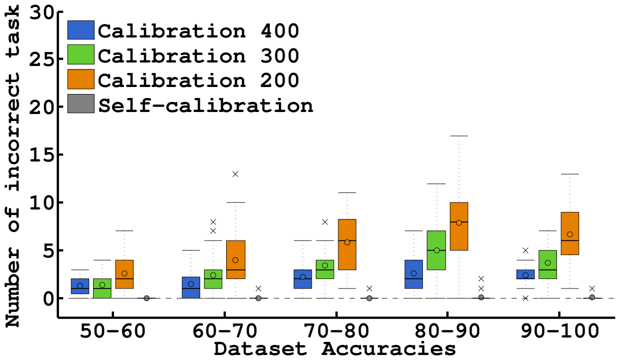


Figure 9: Number of task incorrectly achieved in 500 steps with EEG data. Calibration methods, which do not update their models once calibrated, make more errors.

6 CONCLUSION

In this paper we have shown that, given a limited number of possible tasks, it is possible to solve sequential tasks using human feedback without defining a map between feedback signals and their meaning beforehand. The proposed algorithm optimizes a pseudo-likelihood function and performs active planing according to the uncertainty in the task and meaning spaces. Indeed, taking into account this uncertainty is crucial to solve the task efficiently and to recover the actual meanings. This combination allows: a) a human to start interacting with a system without calibration; b) to automatically adapt calibration time to the user needs which can even outperform fixed calibration procedures; c) to adapt to the uncertainty of the information source from scratch. We showed the applicability of the approach to brain-machine interfaces based on error potentials which could work out of the box without calibration, a long-desired property of this type of systems.

A number of open questions remain to be addressed:

- How the task properties (symmetries, size, . . .) affect the learning properties?
- How to leverage from the finite set of hypothesis constraint? A potential avenue is to use a combination of particle filter and regularization on the task space.
- In real-world applications, users are usually told how to interact with machines. Do people want to have an open-ended choice about what signal to use? Would they be more efficient? When is it better to use a calibration procedure?
- Only prerecorded datasets have been used. However, signals may change during the learning. For instance, people can try to adapt themselves to a robot if they believe the latter is not understanding properly. Or, brain signals are sensitive to the protocol, the duration of the experiment or even the percentage of errors made by the agent [20]. To which extend the behavior of our agent changes the properties of the teaching signal? Can we adapt to such changes online?

Finally, while we only considered correct/incorrect labels, in other works we have considered the use of guidance instructions (go up, go left, ...) in human-robot interaction scenario [14]. But increasing the set of possible labels logically requires collecting more examples to obtain a good enough representation of the different signals. Hence, for BCI domains, it is reasonable to keep a limited number of labels.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. Work partially supported by INRIA, Conseil Régional d'Aquitaine, the 3rd Hand Project (funded under 7th FWP), and the ERC grant EXPLORERS 24007; and from the Spanish Ministry via DPI2011-25892 (L.M.), and DGA-FSE grupo T04.

References

[1] M. N. Nicolescu and M. J. Mataric, “Natural methods for robot task learning: Instructive demonstrations, generalization and practice,” in *Conference on Autonomous agents and multiagent systems*, pp. 241–248, ACM, 2003.

[2] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Chilongo, “Tutelage and collaboration for humanoid robots,” *International Journal of Humanoid Robotics*, vol. 1, no. 02, pp. 315–348, 2004.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

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

[5] F. Kaplan, P.-Y. Oudeyer, E. Kubinyi, and A. Miklósi, “Robotic clicker training,” *Robotics and Autonomous Systems*, 2002.

[6] A. L. Thomaz and C. Breazeal, “Teachable robots: Understanding human teaching behavior to build more effective robot learners,” *Artificial Intelligence*, vol. 172, no. 6, pp. 716–737, 2008.

[7] M. Lopes, F. Melo, and L. Montesano, “Active learning for reward estimation in inverse reinforcement learning,” in *Machine Learning and Knowledge Discovery in Databases*, pp. 31–46, Springer, 2009.

[8] S. Chernova and M. Veloso, “Interactive policy learning through confidence-based autonomy,” *Journal of Artificial Intelligence Research*, vol. 34, no. 1, p. 1, 2009.

[9] M. Mason and M. Lopes, “Robot self-initiative and personalization by learning through repeated interactions,” in *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, pp. 433–440, IEEE, 2011.

[10] K. Judah, S. Roy, A. Fern, and T. G. Dietterich, “Reinforcement learning via practice and critique advice.,” in *AAAI*, 2010.

[11] W. B. Knox and P. Stone, “Interactively shaping agents via human reinforcement: The tamer framework,” in *International conference on Knowledge capture*, pp. 9–16, ACM, 2009.

[12] S. Griffiths, S. Nolfi, G. Morlino, L. Schillingmann, S. Kuehnel, K. Rohlfing, and B. Wrede, “Bottom-up learning of feedback in a categorization task,” in *Development and Learning and Epigenetic Robotics (ICDL), 2012*, pp. 1–6, IEEE, 2012.

[13] M. Lopes, T. Cederborg, and P.-Y. Oudeyer, “Simultaneous acquisition of task and feedback models,” in *IEEE - International Conference on Development and Learning (ICDL’11)*, 2011.

[14] J. Grizou, M. Lopes, and P.-Y. Oudeyer, “Robot Learning Simultaneously a Task and How to Interpret Human Instructions,” in *Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, (Osaka, Japan), 2013.

[15] J. Grizou, I. Iturrate, L. Montesano, P.-Y. Oudeyer, and M. Lopes, “Calibration-Free BCI Based Control,” in *AAAI Conference on Artificial Intelligence*, (Quebec, Canada), 2014.

[16] 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, p. e33758, Jan. 2012.

[17] P.-J. Kindermans, M. Tangermann, K.-R. Müller, and B. Schrauwen, “Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training erp speller,” *Journal of neural engineering*, vol. 11, no. 3, p. 035005, 2014.

[18] R. Brafman and M. Tennenholtz, “R-max-a general polynomial time algorithm for near-optimal reinforcement learning,” *Journal of Machine Learning Research*, vol. 3, 2003.

[19] J. Z. Kolter and A. Y. Ng, “Near-bayesian exploration in polynomial time,” in *International Conference on Machine Learning*, ACM, 2009.

[20] R. Chavarriaga and J. Millán, “Learning from EEG error-related potentials in noninvasive brain-computer interfaces,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, no. 4, 2010.

[21] I. Iturrate, L. Montesano, and J. Minguez, “Task-dependent signal variations in eeg error-related potentials for brain–computer interfaces,” *Journal of neural engineering*, vol. 10, no. 2, p. 026024, 2013.

[22] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and M. KR, “Single-trial analysis and classification of ERP components: A tutorial,” *Neuroimage*, 2010.

[23] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2003.