2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

A Teacher-Student Markov Decision Process-based Framework for Online Correctional Learning*

Inês Lourenço, Rebecka Winqvist, Cristian R. Rojas, and Bo Wahlberg[1]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992858

***Abstract*— A classical learning setting typically concerns an agent/student who collects data, or observations, from a system in order to estimate a certain property of interest. *Correctional learning* is a type of cooperative teacher-student framework where a teacher, who has partial knowledge about the system, has the ability to observe and alter (correct) the observations received by the student in order to improve the accuracy of its estimate. In this paper, we show how the variance of the estimate of the student can be reduced with the help of the teacher. We formulate the corresponding *online* problem – where the teacher has to decide, at each time instant, whether or not to change the observations due to a limited budget – as a Markov decision process, from which the optimal policy is derived using dynamic programming. We validate the framework in numerical experiments, and compare the optimal online policy with the one from the *batch* setting.**

I. Introduction

With the rapid growth of smart systems and IoT, we are able to collect enormous amounts of data like never before. These data may range from medical images captured by camera sensors to distance measures from e.g. lidars and radars. From this data, agents can learn to perform tasks such as cancer detection and prognosis [1], and autonomous driving [2]. These are only some examples and the application domain is much more extensive.

In the Oxford dictionary [3], the term *learning* is defined as the “acquisition of knowledge or skills through study, experience, or being taught”. In this work we consider a combination of the latter two; “experience”, by using dynamic programming to train a teacher, and “being taught”, by letting the trained teacher transfer its knowledge to a student agent. Setups that involve the presence of aiding expert agents are commonly denoted cooperative learning problems.

Cooperative problems play an important role in our lives. Indeed, most tasks we perform require some sort of collaboration; acquiring a new skill, such as learning how to drive, social learning, search and rescue operations, and much more. Solving these tasks is, however, not trivial, and the aid of an external agent can be very helpful. In the literature, the two most famous paradigms of cooperative learning that use a teacher-student framework are *learning from demonstration* [4] and *imitation learning* [5], in which the role of the teacher is to accelerate the learning of the student by means of showing it the optimal policy.

In this work, we consider a different teacher-student paradigm. We study how the teacher can assist the student by intercepting, and altering, the data collected by the student. This approach is denoted *correctional learning* and was recently proposed by [6], in an effort to tackle the problem of assisting an agent in situations where transmitting knowledge directly might be impossible or undesirable. This correctional learning framework opens up for new possibilities. In the traditional learning setting, helping an agent learn a policy, the parameters of a system, or the state of the world, are some of the potential applications. These problems are called reinforcement learning, system identification, and filtering, respectively. Other examples are manual output-error correction of machine learning models, cooperative learning for task-performing, and estimation of user preferences and ratings. Alternatively, the correctional framework may be viewed as a means for diversifying the information presented to a user – in social media applications and search engines, it could tackle the growing issue of echo chambers or confirmation bias, and the spreading of “fake news”. Financial applications are another field of interest, in which the framework could be used to e.g. influence an investor’s market state predictions for stock portfolio allocation. Other application examples are discussed in Section V-C.

In most of these fields, however, immediate (online) action is typically required as observations arrive sequentially, due to the need of a learning process that adapts and rapidly changes. Online algorithms often make learning faster and computationally cheaper. In this paper, we thus present the *online* correctional learning framework where the teacher has to decide, at each time instant, whether or not to alter the corresponding observation.

The research question we answer in this paper is then:

How should a teacher modify, at each instant and under budget constraints, the data received by a student in order to assist its learning process?

The main contributions of this paper are as follows.

- Computation of a theoretical bound for how much the teacher can improve the estimation of the student in the case of discrete systems.
- Formulation of a *Markov decision process* (MDP) for the correctional learning framework performed in an online setting.
- Demonstration of the results in two numerical experiments; in particular, of the optimal policy of the

*This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), the Swedish Research Council Research Environment NewLEADS under contract 2016-06079, and the Digital Futures project EXTREMUM.

[1] The authors are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden {ineslo,rebwin,crro,bo}@kth.se

978-1-6654-6761-2/22/$31.00 ©2022 IEEE

teacher obtained in an offline fashion through dynamic programming.
- Comparison of the proposed online correctional learning framework with the batch framework.

The rest of the paper is organised as follows. In Section II, the correctional learning problem is formulated. In Section III, we derive bounds for how much the teacher can help the student, and, in Section IV, the proposed algorithm for performing online correctional learning is derived. Finally, Section V validates the presented methods in numerical experiments and Section VI concludes the paper.

A. Related work

Incorporating external information in decision-making is a commonly studied problem, not least in the teacher-student frameworks previously mentioned [4], [5]. Despite being of a similar nature, our proposed framework differs from these in its nature of correcting the observations. Below are examples of areas that inspire our work and which resemble the correctional behavior of the teacher when selecting the inputs to intercept and alter.

Feature selection is a technique used in learning and classification tasks to select the most relevant and non-redundant features to improve learning [7]. Similarly to our correctional learning framework, this family of methods has seen a shift from batch to online techniques [8] – which represent a more promising group of efficient and scalable machine learning algorithms for large-scale applications.

Our work is positioned around other important methods such as *input design for system identification* [9], [10], where the input signals are designed to guarantee a certain model accuracy; *active learning* [11], where the learner queries the teacher for the desired labels; *counterfactual explanations* [12], which is a branch within explainable artificial intelligence that uses feature importance to explain how a small perturbation of an input datapoint affects the output of a machine learning model; and *learning with side information* [13], in which additional information is provided to the learner to help its learning process. The concept of side information is also very popular in information theory (in connection to communication problems) [14]. Examples of other areas are *active fault diagnosis* [15], consisting of the design of an input signal for minimizing the time and energy required to detect and isolate faults in the outputs of a system; *anomaly detection* [16], which aims to improve the performance of the model by removing anomalies from the training sample; and *controlled sensing* [17], where the decision-maker can choose at each time instant which sensor to use to obtain the next measurement.

In the next sections we show how we use these techniques as motivation and inspiration to create a simple and efficient online mechanism for sequentially correcting observations in a wide variety of applications.

II. Problem Formulation

In this section, we formally introduce the correctional learning problem: a teacher, who has knowledge about a system of interest, aims to help a student's learning process by altering the data it receives. Throughout the paper we denote the i-th element of a vector v as $[v]_i$.

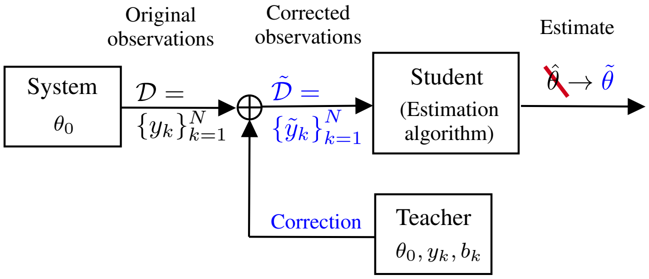

Fig. 1. Schematic representation of the correctional learning framework.

A. Introduction to Correctional Learning

A learning agent (student) is sequentially collecting information from a source in the form of observations $\mathcal{D} = \{y_k\}_{k=1}^N$, $y_k \in \mathcal{Y}$, throughout N time steps, and estimating characteristics of interest $\hat{\theta}$ about that system. An expert agent (teacher) has more knowledge about the system and its goal is to assist the estimation process of the student. However, for several reasons it might be impossible or undesirable for the teacher to transmit its knowledge directly to the student. For example, the expert's knowledge might be too abstract (teaching someone to drive), or too complex to be transmitted, the communication might be restricted due to privacy concerns, or the teacher and student might operate in different model classes or parameterizations [6]. The teacher, therefore, instead has the ability to interfere by intercepting, and altering, the observations collected by the student to $\tilde{\mathcal{D}} = \{\tilde{y}_k\}_{k=1}^N$. A schematic representation is shown in Figure 1. Improving the estimation thus means obtaining an altered estimate $\tilde{\theta}$ closer to the true estimate $(||\tilde{\theta}_N - \theta_0|| \leq ||\hat{\theta}_N - \theta_0||)$, or which converges to the true one faster ($\text{var}\{\tilde{\theta}\} < \text{var}\{\hat{\theta}\}$). This is of particular importance when the teacher has studied the system of interest for a longer time than the student and thus has a more accurate estimate. However, if this estimate is not perfect, rather than giving the student its estimate, it instead corrects the information acquired by the student to better match its own. One can also imagine that the student uses few training examples to study the behaviour of an agent, and might thus include exploratory actions in its analysis – which biases its estimation and illustrate the importance of a teacher interfering to alter these.

Additionally, altering the data might be expensive or might compromise privacy. Therefore, correctional learning includes the budget constraint

$$B(\mathcal{D}, \tilde{\mathcal{D}}) \leq b, \qquad (1)$$

where B is a distance measure between two sequences which represents the budget, b, that the teacher has on how much it can interfere with the observations obtained from the system. If the observations are discrete, B can be defined as the l_1-norm divided by N: $\frac{1}{N}\sum_{k=1}^N |y_k - \tilde{y}_k| \leq b$.

B. Batch Correctional Learning

In *batch* correctional learning, multiple observations are intercepted simultaneously. This can be the case in, for example, a communication channel, where multiple bits can be delayed on the way from the source to the receiver. In [6], the batch problem was solved by minimizing the distance between the true parameter and the empirical estimate computed by the student, $V(\theta_0, \tilde{\theta})$, according to the following optimization problem:

$$\begin{aligned} \min_{\tilde{\mathcal{D}}} \quad & V(\theta_0, \tilde{\theta}) \\ \text{s.t.} \quad & \tilde{y}_k \in \mathcal{Y}, \text{ for all } \tilde{y}_k \in \tilde{\mathcal{D}}, \\ & B(\mathcal{D}, \tilde{\mathcal{D}}) \leq b, \end{aligned} \tag{2}$$

where B is the distance measure from (1). In [6] it was shown that the resulting set $\tilde{D}$ of corrected observations was the optimal one for the case of binomial data, and by how much the variance of the corrected estimate is decreased compared to the original one. In this paper, we formulate an MDP to solve the problem in an online setting and for an extensive variety of applications.

III. Correctional learning bounds for discrete systems

In this section we analyse how much the teacher can effectively help the student, for the case when the observations are discrete.

The following theorem relates the estimates of the mean values of two sequences of observations – the original, Y/N, and the corrected one by the teacher, $\tilde{Y}/N$. Knowing how much the variance of the corrected estimate is decreased is a measure of how much the teacher can help reducing the estimation error of the student.

Theorem 1 (Variance decrease of the altered estimate): Let $X_1, \ldots, X_N$ be N independent and identically distributed (i.i.d.) random variables that take values in the set $\{0, 1, \ldots, M\}$ with mean μ, and $Y = X_1 + \cdots + X_N$. Let $B \in \{0, 1, \ldots, N\}$ denote the number of variables that can be altered from Y to Z in order to make Z as close as possible to the mean:

$$\tilde{Y} = \operatorname*{arg\,min}_{\{Z \in \{0, \ldots, N\} : |Y - Z| \leq B\}} |Z - N\mu|. \tag{3}$$

Then,

$$\operatorname{var}[\tilde{Y}/N] \leq M^2 \exp\left[-\frac{2B^2}{NM^2}\right]. \tag{4}$$

Let us further assume that $X_i \sim \operatorname{Unif}(\{0, \ldots, M\})$ (the assumption of being uniformly distributed is not crucial but provides a special case that is easier to understand). Then,

$$\frac{\operatorname{var}[\tilde{Y}/N]}{\operatorname{var}[Y/N]} \leq \frac{6M}{5M+1} \exp\left[-\frac{2B^2}{NM^2}\right]. \tag{5}$$

The proof of Theorem 1 is provided in [18] based on Hoeffding's inequality. This theorem provides an upper bound for the decrease in variance of the estimate computed by the student due to the help of the teacher, according to its budget B. It implies that:

i) the teacher's ability to improve the learning of the student increases with its budget;

ii) for a given budget, the improvement becomes less important as N grows. This is reasonable, since the average deviation of Y/N around μ is of order $\mathcal{O}(1/\sqrt{N})$, while the improvement due to the teacher can be at most B/N;

iii) for a fixed budget and N, the improvement degrades as M increases, since the variance of Y/N increases with M, which makes it increasingly harder for a teacher to compensate for "bad" samples.

After computing by how much the teacher can improve the estimation process of the student in a discrete setting, we next propose a framework for how the teacher can achieve this by altering the observations in real time.

IV. Online Correctional learning

In this section we present a framework for computing an optimal online policy for the teacher. Unlike the batch case, the online setting is a more realistic scenario in which the observations are obtained sequentially and the expert has to make, at each time instant, the decision of whether or not to change the current observation.

A. Formulation of the Markov Decision Process

We are now ready to present the second main result of our paper, which is the formulation of an MDP to describe the teacher's policy for the online correctional learning problem when the student samples discrete observations $y_k \in \{0, 1, \ldots, M-1\} = \mathcal{Y}$ from a system to estimate the true parameter θ_0:

States:	$s = (x_{1:k}, b_k, y_k)$
Actions:	$a = \{\text{keep } y_k, \text{change to } \bar{y}_k\}$
Time-horizon:	N
Reward function:	$-\,\|\|\tilde{\theta}_N \;-\; \theta_0\|\|_1$
Constraint:	number of times the action "change to $\bar{y}_k$"is taken $\leq b$
Transition probabilities:	see (8).

In more details:

- *States*: The states $s \in \mathcal{S}$ of the MDP are tuples containing: *i)* $x_{1:k}$ – an $M \times 1$ vector with the number of times each observation has been seen until time k; *ii)* $b_k \in \mathbb{N}_0$ – the current budget left to use at time k; *iii)* y_k – the observation received at time k. The number of states, $\operatorname{card}(\mathcal{S})$, is finite and upper bounded by $N^{M+1}b$. However, the constraint $\sum_{l=1}^{M} [x]_l \leq N$ renders many of these states invalid, which results in a much smaller and tight upper bound: $\operatorname{card}(\mathcal{S}) \leq \operatorname{card}(x)bN$. Here, $\operatorname{card}(x)$ can be computed using *multiset coefficients* as

$$\sum_{n=1}^{N} \left(\!\!\binom{M}{n}\!\!\right) = \sum_{n=1}^{N} \frac{M(M+1)\ldots(M+n-1)}{n!}, \tag{6}$$

which are the N-permutations of M with repetitions and which satisfy the previous constraint.

- *Terminal states*: These are the ones where all the observations have been received, that is, where

$$\sum_{l=1}^{M} [x]_l = N. \tag{7}$$

- *Actions*: The possible actions are to keep the last observation y_k or change it to a certain value $\tilde{y}_k \in \mathcal{Y}$. The number of actions is $\text{card}(A) = M$.

- *Reward function*: The reward is zero in all states except in the terminal states, in which it is inversely proportional to the error computed after the teacher's alterations.

- *Transition probabilities*: If the action is "keep y_k", the next state depends, with probability $p(y_{k+1})$, on the next received observation y_{k+1}. The value of the next state is obtained by simply replacing the last value of the previous state y_k by the new observation received, and adding one to that entry of the vector x, $[x']_{y_{k+1}} = [x]_{y_{k+1}} + 1$. If the action is "change to $\tilde{y}_k$", the next state will have the same probability as in the previous case, where one is added to $[x]_{y_{k+1}}$. However, it will now have one subtracted from the previous observation in $[x]_{y_k}$ and one added in $[x]_{\tilde{y}_k}$ (since y_k was altered to $\tilde{y}_k$), as well as a budget of $b_{k+1} = b_k - 1$. We can write these mathematically as follows, and pseudocode can be found in [18]:

$$\begin{aligned}
&\mathbb{P}\{(x', b, y_{k+1}) \mid s = (x, b, y_k), a = \text{“keep } y_k\text{”}\} = p(y_{k+1}),\\
&\quad \text{where } [x']_{y_{k+1}} = [x]_{y_{k+1}} + 1\\
&\mathbb{P}\{(x', b-1, y_{k+1}) \mid s = (x, b, y_k), a = \text{“}\tilde{y}_k\text{”}\} = p(y_{k+1}),\\
&\quad \text{where } [x']_{y_{k+1}} = [x]_{y_{k+1}} + 1, [x']_{y_k} = [x]_{y_k} - 1, \text{and}\\
&\quad [x']_{\tilde{y}_k} = [x]_{\tilde{y}_k} + 1,\\
&\text{and } \mathbb{P}\{\text{others}\} = 0.
\end{aligned} \tag{8}$$

Note that the chosen formulation of the states and actions satisfies the Markovian property.

- *Constraint*: The constraint is enforced by attributing an infinitely negative reward to transitions to states where the budget would be $b_{k+1} < 0$.

The optimal policy for the online correctional learning problem represented as the previous MDP can be obtained using dynamic programming [19].

This framework can be translated to different scenarios by adjusting the reward function to a representative description of the student's goal in the task at hand.

Remark 1: This framework can also be used when the observations are continuous, by discretizing the observation space and changing the constraint to the total amount of correction $\sum_{k=1}^{N} |y_k - \tilde{y}_k| \leq b$.

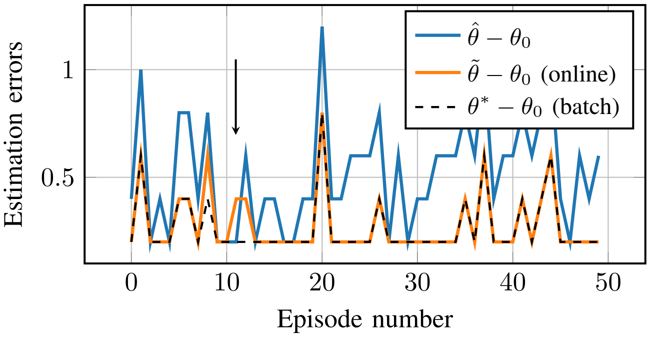

Fig. 2. The blue line shows the estimation errors obtained during 50 episodes with the original sequence of observations ($b = 0$). The orange line represents how much this error is decreased with the help of the teacher with $b = 1$, that is, by changing one observation. The online policy learned by the teacher thus allows it to reduce the errors considerably, closely approaching the batch case shown by the dashed black line. The larger the budget b, the closer the orange and black curves become to e_{min} in (9).

V. Numerical Results

In this section, we validate the framework proposed in Section IV by showing significant gains in using the teacher to improve the learning of the student in two different tasks. We first consider the simple example of computing the mean of bi- and multi-nomial data, since it grants us, due to its simplicity, the derivation of explicit solutions for the batch and online settings, as well as a thorough analysis of the intrinsic workings of the framework. We then apply the proposed framework to a problem of biological parameter estimation, to illustrate its application to more complex scenarios. The simulations were performed using Python 3.7 and a 1.90 GHz CPU.

A. Example: multinomial data

Figure 2 presents the results of the MDP proposed in Section IV for performing correctional learning in an online setting. The figure shows the estimation error of the student with, in blue, the original sequence of observations – that is, without the help of the teacher – and, in orange, the corrected sequence. The estimates θ are computed as the mean of the observations, $[\theta]_i = \sum_{k=1}^{M} I(y_k = i)/N$, and we consider that an observation is randomly sampled $N = 5$ times from a multinomial distribution with parameter $\theta_0 = [0.4; 0.3; 0.3]$, over 50 episodes. $I(/cdot)$ is the indicator function. Unlike in the binomial case, where a closed form solution for the minimum attainable error can be computed (see [18]), in the multinomial case this error is compared to the batch error obtained from (8) in [6] and using the l_1-norm in the objective function for consistency. The minimum error, independent of $\hat{\theta}$ and b, can, however, be computed as

$$e_{min}(N, \theta_0) = \left\| \theta_0 - \frac{[\theta_0 N]}{N} \right\|_1 = 0.2, \tag{9}$$

which is achieved by $\theta^* = [0.4; 0.4; 0.2]$ or $[0.4; 0.2; 0.4]$. In (9), the brackets $[\cdot]$ without subscript mean rounding to the closest integer value, subject to the constraint $\mathbb{1}^T \theta_{min} = 1$ where $\theta_{min} = \frac{[\theta_0 N]}{N}$.

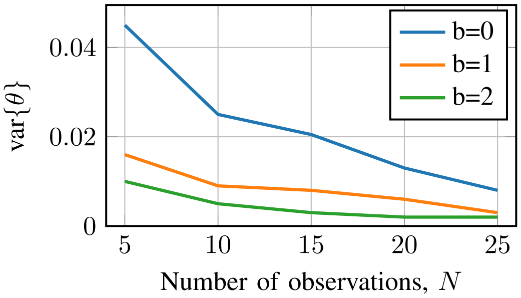

Fig. 3. Reduction of the variance of the estimate for increasing budgets b as N increases. The case $b = 0$ corresponds to when the teacher cannot assist the student.

Intuitively, one would expect the teacher's optimal policy to be delaying as much as possible spending its budget. In the binomial case the online policy learned,

$$\mu^* = \begin{cases} a_k = \text{keep } y_k, & \text{if } b_k \leq 0 \text{ or } [x]_{y_k} \leq \big[[\theta_0]_{y_k} N\big], \\ a_k = \text{alter to } \tilde{y}_k = 1 - y_k, & \text{otherwise,} \end{cases}$$

is optimal since it coincides with the policy computed using batch correctional learning, as is shown in [18]. In the multinomial case from Figure 2, both differ only in a limited amount of scenarios, when a less expected sample that has a small reward is obtained. Note that in episode 11 (marked with an arrow in the figure), the altered estimate $\tilde{\theta}$ is even worse than the original one, $\hat{\theta}$. The teacher chose to alter the fourth observation of $\underline{1}, \underline{2}, \underline{0}, \underline{\mathbf{2}}, \underline{?}$ to a 0 since the expected value was larger (receiving a 1 or a 2 at $k = 5$ had a large probability and maximum reward), but the less likely observation, 0, was received instead.

Remark 2: Risk-aversion conditions for how much the teacher is willing to sporadically risk worsening the learning could be added to the MDP by changing the cost function by a risk-averse one.

Figure 3 shows that, as expected, the variance of the estimate decreases as the number of observations increases. However, as the budget of the teacher increases, the variance is further decreased. This result illustrates the conclusions from Theorem 1.

B. More complex example: biological parameter estimation

Biological internal models have taken a major role in the exploration and validation of neuroscientific theories, e.g., when it comes to understanding the role of the Cerebelum in motor control [20], or predicting and treating neurological diseases [21]. In the next example, we apply online correctional learning to a scenario where the student estimates biological neural parameters θ from observing the actions a performed by biological agents, such as animals or other humans. The observations $\mathcal{Y}$ are in this case the history $\mathcal{H}$ of actions a, and problems like this are called inverse problems [22]. Using a forward model of behaviour, the actions distribution $p(a|\theta)$ can be computed, and, from there, the likelihood $L(\mathcal{H};\theta)$. Maximizing this likelihood (or minimizing its negative value), originates the student's estimate $\hat{\theta}$. In [23], the authors use this inverse method to estimate parameters of neural time perception mechanisms. The model [24] proposed to replicate these mechanisms generates the data from Figure 4 over 2000 episodes. As the student observes the actions of the animal throughout the task, the teacher, having more knowledge about the animal, uses the framework proposed in Section IV to correct certain observations (obtaining a corrected history $\tilde{\mathcal{H}}$), in order to improve the student's estimation of the animal's biological parameters – correct the sampled distribution to be more similar to the distribution from Figure 4 that corresponds to the true parameter θ_0. In this example, the observations are the actions performed by the animal (M is the number of possible actions and the actions correspond to different buttons that the animal can press during a time perception experiment), and θ_0 is the number of microstimuli of its time perception mechanism [25]. The reward function is given by the difference between θ_0 and $\tilde{\theta}$, where the latter is computed from

$$\tilde{\theta} = \arg\min_{\theta} -L(\tilde{\mathcal{H}};\theta), \tag{10}$$

and which corresponds to the difference between the student's corrected estimate and the true parameter.

The left plot of Figure 5 shows the total number of times each action was observed by the student in a certain episode,

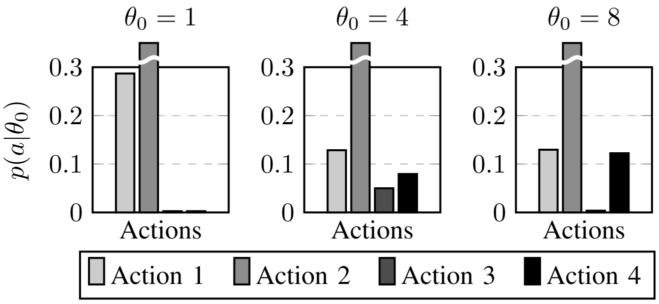

Fig. 4. Effect of the parameter θ_0 on the behaviour a of the agent. These statistics are the base of our model and were computed over one training simulation with 2000 episodes. For the rest of this example we assume that $\theta_0 = 4$ is the true parameter, being the observations sampled from the distribution on the middle.

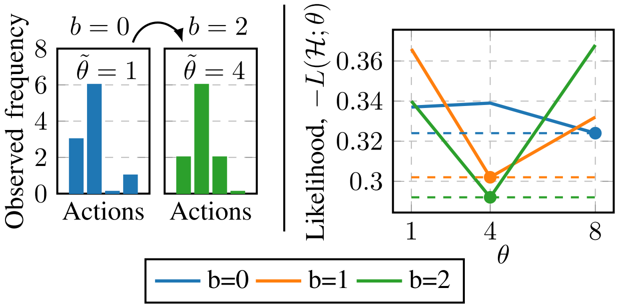

Fig. 5. On the left is shown an example of the action distributions seen by the student before and after the teacher corrections, for a budget of $b = 2$ for $N = 10$. On the right, is shown the effect of these corrections on the corresponding negative likelihood, with the estimated parameter (the one that has the minimum negative likelihood) marked with a dot. For a budget of $b = 2$, the action distribution is closer to the true distribution corresponding to $\theta_0 = 4$ from Figure 4, and therefore the minimum negative likelihood parameter becomes $\tilde{\theta} = 4$ instead of $\tilde{\theta} = 1$.

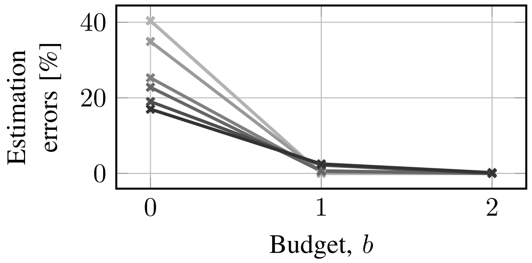

Fig. 6. The percentage of times that the student incorrectly estimates the true parameter decreases with the increase of the budget of the teacher, averaged over 1000 episodes and plotted in different shades for different sample sizes N.

and the corresponding corrections of the teacher as its budget increases. The right plot illustrates how these corrections alter the likelihood of the student estimating each parameter, converging to $\tilde{\theta} = \theta_0 = 4$ for budgets b larger than changing 1 out of $N = 10$ actions. Figure 6 shows how the estimation error decreases over multiple episodes as the budget allocated to help the student increases.

C. Other applications

The two previous examples demonstrate the application of the framework when the observations are samples from a system of interest or actions performed by an agent. By adapting the block diagram from Figure 1, these settings could extend to a large variety of problems such as assisted language learning or improved hypothesis testing, and bring together a variety of fields such as input design and active learning. When training neural networks, for example, correcting the inputs could be compared to input design methods presented in Section I-A. In reinforcement learning tasks, a teacher could use online correctional learning to accelerate the learning of the student in real time. The framework can also be easily translated to an adversarial setting, where the teacher finds the perturbation of the observations that maximizes the impact on the student's estimate – e.g., data poisoning [26, Section 6.1].

VI. Conclusion

In this work we considered that an expert agent, a teacher, can observe the observations collected by a student agent from a certain system of interest. We used *correctional learning* to study how the teacher can alter these observations in real time and under budget constraints in order to improve the learning process of the student. We bounded by how much the teacher can help the estimation of the student by reducing the variance of its estimate, and derived an MDP that gives the optimal policy to perform correctional learning in an online setting using dynamic programming. We illustrated the improvement of the estimation when using multinomial data (Figure 3), and in a biological parameter estimation setting to illustrate the success of the framework in more complex settings (Figure 6).

The way is now paved for extending this method to several interesting applications mentioned throughout the paper, such as correctional reinforcement learning and comparison with related approaches. Tackling the dimensionality problem of MDPs will be an important step along the way.

References

[1] L. Shen, L. R. Margolies, J. Rothstein, E. Fluder, R. B. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Scientific Reports*, vol. 9, 2019.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[3] "learn, v." in *OED Online*. Oxford University Press, Oct. 2021.

[4] 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.

[5] A. Hussein, M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys*, vol. 50, 2017.

[6] I. Lourenço, R. Mattila, C. R. Rojas, and B. Wahlberg, "Cooperative system identification via correctional learning," *19th IFAC Symposium on System Identification*, vol. 54, no. 7, pp. 19–24, 2021.

[7] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256.

[8] J. Wang, P. Zhao, S. C. Hoi, and R. Jin, "Online feature selection and its applications," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 3, pp. 698–710, 2013.

[9] L. Pronzato, "Optimal experimental design and some related control problems," *Automatica*, vol. 44, no. 2, pp. 303–325, 2008.

[10] H. Hjalmarsson, "System identification of complex and structured systems," *European journal of control*, vol. 15, pp. 275–310, 2009.

[11] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and S. Y. Philip, "Active learning: A survey," in *Data Classification*. Chapman and Hall/CRC, 2014, pp. 599–634.

[12] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

[13] P. Kuusela and D. Ocone, "Learning with side information: PAC learning bounds," *Journal of Computer and System Sciences*, vol. 68, no. 3, pp. 521–545, 2004.

[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory, second edition*. Wiley Interscience, 2006.

[15] T. A. N. Heirung and A. Mesbah, "Input design for active fault diagnosis," *Annual Reviews in Control*, vol. 47, pp. 35–50, 2019.

[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[17] V. Krishnamurthy, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge University Press, 2016.

[18] I. Lourenço, R. Winqvist, C. R. Rojas, and B. Wahlberg, "A teacher-student framework for online correctional learning," *arXiv preprint arXiv:2111.07818*, 2021.

[19] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[20] Q. Welniarz, Y. Worbe, and C. Gallea, "The forward model: a unifying theory for the role of the cerebellum in motor control and sense of agency," *Frontiers in Systems Neuroscience*, vol. 15, 2021.

[21] P. Lanillos, D. Oliva, A. Philippsen, Y. Yamashita, Y. Nagai, and G. Cheng, "A review on neural network models of schizophrenia and autism spectrum disorder," *Neural Networks*, vol. 122, pp. 338–363, 2020.

[22] H. W. Engl, C. Flamm, P. Kügler, J. Lu, S. Müller, and P. Schuster, "Inverse problems in systems biology," *Inverse Problems*, vol. 25, no. 12, p. 123014, 2009.

[23] I. Lourenço, R. Mattila, R. Ventura, and B. Wahlberg, "A biologically-inspired computational model of time perception," *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[24] I. Lourenço, R. Ventura, and B. Wahlberg, "Teaching robots to perceive time: A twofold learning approach," in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2020.

[25] E. A. Ludvig, R. S. Sutton, and E. J. Kehoe, "Stimulus representation and the timing of reward-prediction errors in models of the dopamine system," *Neural computation*, vol. 20, no. 12, pp. 3034–3054, 2008.

[26] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in Neural Information Processing Systems (NEURIPS)*, vol. 32, 2019.