Learning Who to Trust: Policy Learning in Single-Stage Decision Problems with Unreliable Expert Advice

Anonymous Author(s)

Affiliation
Affiliation Line 2
email@email.com

Abstract

Work in the field of Assisted Reinforcement Learning (ARL) has shown that the incorporation of external information in problem solving can greatly increase the rate at which learners can converge to an optimal policy and aid in scaling algorithms to larger, more complex problems. However, these approaches rely on a single, reliable source of information; the problem of learning with information from multiple and/or unreliable sources of information is still an open question in ARL. We present CLUE (Cautiously Learning with Unreliable Experts), a framework for learning single-stage decision problems with policy advice from multiple, potentially unreliable experts. We compare CLUE against an unassisted agent and an agent that naïvely follows advice, and our results show that CLUE exhibits faster convergence than an unassisted agent when advised by reliable experts, but is nevertheless robust against incorrect advice from unreliable experts.

1 Introduction

Single-Stage Decision Problems (SSDPs) are a type of Reinforcement Learning (RL) problem with a wide range of useful applications, including recommendation systems (Li et al. 2010), investment portfolio selection (Huo and Fu 2017) and clinical trials (Varatharajah et al. 2018). For example, consider the problem of a doctor who can observe a patient's symptoms and medical history and must prescribe the right set of treatments to improve the patient's condition and avoid harmful side effects. These types of problems have attracted research looking to augment the doctor with a software agent, with the long-term goal of making such diagnoses more comprehensive and widely available (Lauritzen and Spiegelhalter 1988; Heckerman and Nathwani 1992; Kao, Tang, and Chang 2018). However, these types of problems can be very complex, requiring a large amount of time and data for a software agent to adequately solve. For example, a medical diagnosis problem may involve hundreds of potential symptoms and treatments. Furthermore, certain domains may make data acquisition difficult, especially when an agent must interact with the real world. There may also be ethical or safety concerns surrounding data acquisition, especially in medical domains.

One approach to tackling this complexity and the need for sample efficiency is to incorporate external information in the learning process (Bignold et al. 2020). For example, an autonomous medical diagnosis system could be advised by a doctor who instructs the agent to prescribe certain treatments in response to certain combinations of symptoms and medical history. Indeed, previous work has shown that the incorporation of expert advice can improve the rate at which an RL agent converges to a given performance threshold, provided that said advice is correct (Torrey and Taylor 2013).

It may be desirable to incorporate advice from multiple experts, either because a single expert does not have enough expertise to cover the full breadth of the problem, because an expert may not be infallible, or simply because being able to incorporate more advice results in better sample efficiency (Shelton 2000). For example, an agent learning the medical diagnosis problem could be advised by a whole panel of doctors. Incorporating advice from multiple experts introduces its own problems, however, when multiple experts offer conflicting advice for the same situation. Here the agent must decide which advice to follow and which to ignore. In general, expert advisers, especially humans, can give incorrect advice, either in error or through active malice (Efthymiadis, Devlin, and Kudenko 2013). Overcoming these problems has been identified as an open problem in the field of Assisted Reinforcement Learning (Bignold et al. 2020).

In order to address these issues, we present CLUE, a framework for learning SSDPs with policy advice from multiple, potentially unreliable experts. Our contributions consist of the framework itself, as well as Bayesian approaches to modelling expert reliability and pooling advice from multiple experts to facilitate decision-making. We demonstrate that CLUE, when advised by reliable experts, converges faster than an equivalent agent that does not incorporate advice, but is robust to advice given by experts that may be unreliable to some degree.

2 Background and Related Work

2.1 Single-Stage Decision Problems

Reinforcement Learning (RL) is a field of machine learning in which decision-making entities, known as *agents*, learn how to interact with an environment in order to maximise some cumulative reward (Sutton and Barto 2018). Of

the many types of RL problems, this paper concerns itself with *single-stage decision problems* (SSDPs), also known as *contextual bandits* (Langford and Zhang 2007), with discrete states and actions. In this setting, the agent observes some state $s \in S$, selects some action $a \in A$, and receives some reward or utility $r(s, a) \in \mathbb{R}$ from the environment. Each round of observation, action-selection and environment feedback is referred to as a *trial*, and each trial is independent from previous trials. The medical example presented in Section 1 can be posed in this way, with each state being some combination of symptoms and medical history, each action being some combination of treatments, and the reward deriving from factors such as the patient's health, harmful side effects, etc.

A policy $\pi : S \rightarrow A$ is a function that maps each state to an action, and the goal of an agent within an SSDP is to learn the optimal policy π^* that maximises $EU(\pi(s)|s)$ $\forall s \in S$, where $EU(a|s)$ denotes the expected utility (i.e. expected reward) of choosing an action a in state s. The expected utility function is typically not given, and must be learned by the agent.

There are many approaches to learning the optimal policy and selecting actions to facilitate this learning. A common family of approaches are the action-value methods with ϵ-greedy action selection (Sutton and Barto 2018). In these methods, the agent maintains an estimate $Q(s, a) \approx EU(a|s)$. After observing state s_t, with some probability ϵ the agent randomly selects an action $a \in A$ (known as "exploration"), or otherwise selects the action a^* that maximises $Q(s_t, a)$ (known as "exploitation"), and then receives some reward r_t. At the end of each trial, the agent updates $Q(s_t, a_t)$ with the following update rule

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t - Q(s_t, a_t)), \qquad (1)$$

where $\alpha \in (0, 1]$, known as the step size parameter, controls the rate at which the agent learns (Sutton and Barto 2018). Other popular SSDP algorithms include LinUCB (Li et al. 2010), NeuralBandit (Allesiardo, Féraud, and Bouneffouf 2014) and Contextual Thompson Sampling (Agrawal and Goyal 2013).

2.2 Assisted Reinforcement Learning

Assisted Reinforcement Learning (ARL) is a framework that encompasses a wide range of RL methods that incorporate information external to the environment in the learning process (Bignold et al. 2020). Some examples of ARL approaches include Heuristic RL (Bianchi, Ribeiro, and Costa 2004), RL from Demonstration (Taylor and Chernova 2010) and Transfer Learning in RL (Taylor and Stone 2009). Of particular relevance to this work is Interactive RL (IRL), in which an expert (either human of software-based) provides information to the agent during the learning process, usually as a response to the behaviour of the agent (Thomaz, Hoffman, and Breazeal 2005).

IRL methods can be classified based on the type of advice the expert gives. In *reward-shaping* approaches (Knox and Stone 2009; Gimelfarb, Sanner, and Lee 2018), the expert modifies the reward signal provided to the agent (e.g. by providing positive or negative feedback when the agent selects certain actions). In *policy-shaping* approaches (Fernández and Veloso 2006; Griffith et al. 2013), the expert modifies the agent's policy, typically by advising an action for a given state and having this action override the agent's policy whenever that state is encountered. Both approaches are preferred for different situations and domains. In this paper, we focus on policy-shaping, as state-action advice can be more easily elicited from human experts in our domains of interest (particularly the medical diagnosis domain), requires minimal similarity between the agent and expert (Torrey and Taylor 2013), and is more robust to infrequent and inconsistent feedback (Griffith et al. 2013).

Many (but not all) approaches in ARL assume the advice to be coming from a single, infallible expert. However, this assumption does not always hold, especially when the expert is human (Efthymiadis, Devlin, and Kudenko 2013). Suboptimal advice could be the result of communication error, erroneous domain knowledge or a malicious expert. Furthermore, incorporating advice from multiple experts introduces the possibility of two or more experts offering contradicting advice, requiring the agent to choose which advice is more likely to be correct (Shelton 2000). The problems of incorporating advice from unreliable experts and incorporating advice from multiple experts are considered open questions in ARL (Bignold et al. 2020).

Several approaches deal with these problems in different ways. Gimelfarb, Sanner, and Lee (2018) combine reward-shaping advice from multiple experts as a weighted sum of potential functions, where the weights are updated as the agent learns. Griffith et al. (2013) account for incorrect advice by modelling the probability of an expert giving correct advice (here in the form of a label of "right" or "wrong" rather than explicitly telling the agent what to do) with a single, static parameter $C \in [0, 1]$, where $C = 0$ corresponds to always giving suboptimal advice and $C = 1$ corresponds to always giving optimal advice. Both approaches are incompatible with the state-action advice we consider in this work. Nevertheless, these approaches contain elements which are applicable to the state-action advice we consider. The combination of advice weighted by the reliability of each expert forms the basis of the decision-making process outlined in Section 3.3, and the model of reliability as the probability of advising optimally is discussed in Section 3.2.

Fernández and Veloso (2006) account for potential unreliability in a transferred policy by initially relying on the policy with some probability ψ, and subsequently decaying ψ over time. Although the original approach considers advice given at the beginning of the learning process rather than during the process, this approach can be adapted to the setting this work considers. A comparison of performance with a variety of experts can be found in the supplementary document.

Other approaches that consider unreliable information (albeit with different temporalities and types of advice) include the Normalised Actor-Critic algorithm, an RL from demonstration approach which refines an initial policy obtained from potentially imperfect demonstrations (Gao et al. 2018), and the joint learning framework of Keswani, Lease, and Kenthapadi (2021), a classification algorithm, in which

a classifier is learnt together with a deferrer which learns when to defer to one or more experts, which may have incorrect domain knowledge or biases.

3 Methodology

As stated in the previous section, the aim of this paper is to devise an algorithm that can model the reliability of multiple experts and use these models to combine the policy-shaping advice given by the experts to calculate an optimal policy for an SSDP. To that end, in this section we present **CLUE** (Cautiously Learning with Unreliable Experts), a framework for learning to solve SSDPs with policy advice from multiple, potentially unreliable experts.

We begin by defining what it means for an expert to be reliable or unreliable. The advice given by an expert e takes the form of a state-action pair $(s, a^{(e)})$, and is interpreted as the expert asserting that $a^{(e)}$ is the optimal action for state s. If that assertion is true, the advice is said to be correct. If, for all states across all trials, an expert always gives correct advice, that expert is said to be *reliable*; otherwise, it is *unreliable*.

In order to make the problem of learning with the advice of multiple, potentially unreliable experts tractable, we make three key assumptions. Firstly, we assume the expert to be uniformly reliable across S. While not holding in the general case, this assumption can be expected to hold when S represents a particular domain of expertise. For example, if we limit the medical diagnosis problem from Section 1 to a single domain of expertise (e.g. heart disease), then any given expert may reasonably be expected to have uniform expertise across the domain, although that level of expertise may differ between experts (e.g. a seasoned cardiologist will be more reliable than a medical student). Relaxing this assumption requires dividing the state space into domains of expertise, which could range in size from a single state to the entire state space and could potentially overlap with each other. This division is non-trivial, and lies outside the scope of this work.

Secondly, we assume that each expert is uniformly reliable across all trials. Thus, the probability of an expert giving correct advice does not change between trials. This may not hold for all situations. For example, a human expert may get tired and start making more mistakes in later trials. However, solving a non-stationary problem such as this lies outside the scope of this work.

Finally, we assume that each expert gives advice independently, which allows for each expert to be rated on their own merits and eliminates factors external to the problem such as deferring to a more persuasive or senior expert, or dismissing the advice of an expert due to biases.

3.1 CLUE

Having discussed key definitions and assumptions, we now outline how CLUE operates and the specific contributions this work makes. CLUE involves three actors: an environment, an agent and a panel E of one or more experts. The high-level component view of CLUE is provided in Figure 1, and pseudocode is provided in Algorithm 1. More detailed

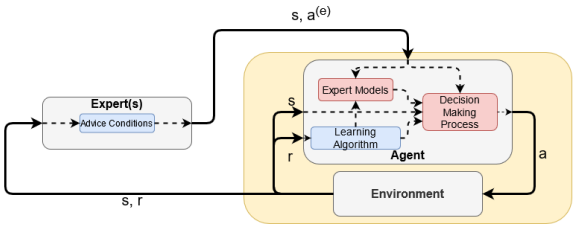

Figure 1: A high-level overview of CLUE, showing the interactions between the environment, the agent and expert(s). Components depicted in red represent contributions made by this work.

Algorithm 1: Cautiously Learning with Unreliable Experts

procedure CLUE($environment, E, N$) ▷ E = panel of experts, N = number of trials
  for $t \in [0, ..., N-1]$ **do**
    $s_t \leftarrow$**sample_state**($environment$)
    $a_t \leftarrow$**select_action**(s_t) ▷ Sec. 3.3
    $r_t \leftarrow$**execute_action**($a_t, environment$)
    $advice_t \leftarrow$**advise**(E, s_t, a_t, r_t) ▷ e.g. Sec. 4.1
    learn(s_t, a_t, r_t) ▷ Sec. 2.1
    update_estimates($s_t, a_t, r_t, advice_t$) ▷ Sec. 3.4

pseudocode for each component can be found in the supplementary material.

The environment is identical to any SSDP environment, as discussed in Section 2.1. Namely, for trial t, it samples state s_t, accepts action a_t from the agent and returns reward r_t. At the end of the trial, each expert e in panel E receives $\langle s_t, a_t, r_t \rangle$ and may offer their own advice, $(s_t, a^{(e)})$ on what action the agent should have taken this trial. How exactly an expert decides whether or not to offer advice and which advice to give differs between experts; our approach is outlined in Section 4.1.

The agent is composed of three components which facilitate decision making and policy learning. The first of these components is a learning algorithm, which uses the information $\langle s_t, a_t, r_t \rangle$ to learn a policy, such as the procedure presented in Section 2.1.

The second component, and one of the contributions of this work, is a model of the reliability of each expert (see Section 3.2). This model is necessary for learning which pieces of advice are to be followed and which are to be ignored. When an expert utters a piece of advice at the end of a trial, the agent uses its own information about the environment (such as a Q function) to evaluate the advice and update the model (see Section 3.4).

The third component, and another contribution of this work, is a decision making process which uses the information learned by the learning algorithm and the models of each expert to select an action for a state, given any advice it has previously received for that state (see Section 3.3).

3.2 Modelling Experts

The first contribution we address is how an agent working within the CLUE framework models the reliability of each expert. Intuitively, we can think of an expert as being unreliable to some degree. For example, an expert that offers correct advice in 95% of trials, while still unreliable, is more reliable than an expert that is always wrong. Following Griffith et al. (2013), we model an expert's reliability, $\rho \in [0, 1]$, as the probability of the expert giving correct advice, where $\rho = 0$ corresponds to an expert whose advice is always wrong and $\rho = 1$ corresponds to a reliable expert. Rather than maintaining a static value for each expert, we can model a probability distribution of the value of ρ using a Beta distribution $Beta_\rho[\alpha, \beta]$, whose shape is determined by the parameters $\alpha, \beta > 0$ (Owen 2008). These parameters can be thought of as counts, with α and β recording the number of times correct or incorrect advice was given respectively.

The best estimate of the reliability of an expert is therefore the expected value $\mathbb{E}[\rho] = \frac{\alpha}{\alpha+\beta}$. Thus, for each expert $e \in E$, the agent maintains a distribution $Beta_{\rho^{(e)}}[\alpha^{(e)}, \beta^{(e)}]$ from which the reliability estimate $\mathbb{E}[\rho^{(e)}]$ is calculated.

3.3 Making Decisions

We now turn our attention to the problem of how $\mathbb{E}[\rho]$ can inform the decision-making process. Suppose that, at the start of trial t, the agent observes state s_t and recalls any advice that some subset $E_t \subseteq E$ of experts offered for state s_t in trials $[0, ..., t-1]$. The agent decides whether to "exploit" - selecting an action according to its policy to maximise reward - or "explore" - selecting some other action to improve estimates of expected reward. If exploring, the agent must choose between randomly selecting an action or following advice, in which case it must choose which advice to follow.

If $E_t = \varnothing$, no advice has been offered, such as may happen at the beginning of the learning process, and the agent must act without advice. The approach adopted in this work is the action-value ϵ-greedy algorithm, outlined in Section 2.1.

If $|E_t| \geq 1$, at least one expert has offered advice. A naïve approach may be to follow the advice of the expert with the highest expected reliability. However, this approach loses information that could be provided by consensus among experts (so-called "wisdom of the crowd" (Yi et al. 2012)) or by adversarial experts (whose advice is almost always wrong, thus informing the agent which actions not to take).

In order to take advantage of this information, we employ a Bayesian method of pooling advice, inspired by similar approaches in potential-based reward shaping (Gimelfarb, Sanner, and Lee 2018) and in crowd-sourced data labelling (Burke and Klein 2020). Let a^* denote the optimal action for state s_t and $v_t^{(e)}$ denote the advice utterance given by expert e for s_t, with V_t denoting the set $\{v_t^{(e)} | e \in E_t\}$. Our aim, therefore, is to calculate $P(a_j = a^*|V_t)$ for each $a_j \in A$. To do this, we employ Bayes' rule as follows:

$$P(a_j = a^*|V_t) = \frac{P(V_t|a_j = a^*)P(a_j = a^*)}{\sum_{k=0}^{|A|} P(V_t|a_k = a^*)P(a_k = a^*)}. \quad (2)$$

If nothing is known about the environment prior to learning, a reasonable assumption would be to assume that each action has a uniform prior probability of being optimal. Under this assumption, Equation 2 reduces to

$$P(a_j = a^*|V_t) = \frac{\prod_{e \in E_t} P(v_t^{(e)}|a_j = a^*)}{\sum_{k=0}^{|A|} \prod_{e \in E_t} P(v_t^{(e)}|a_k = a^*)}, \quad (3)$$

which combines all the available advice for s_t to calculate the probability of each a_j being optimal. It is worth noting that, if for a particular domain one can reasonably assume a non-uniform prior distribution of $P(a = a^*)$, this distribution can be incorporated into Equation 2 without fundamentally changing this decision-making process.

All that remains is to calculate $P(v_t^{(e)}|a_j = a^*)$. Recalling that the probability of the advice being correct is equal to $\mathbb{E}[\rho^{(e)}]$ and assuming that, if the advice is incorrect, the expert is equally likely to advise any suboptimal action, as in Masegosa and Moral (2013), then it follows that

$$P(v_t^{(e)}|a_k = a^*) = \begin{cases} \mathbb{E}[\rho^{(e)}] & v_t^{(e)} = a_k \\ \frac{1-\mathbb{E}[\rho^{(e)}]}{|A|-1} & v_t^{(e)} \neq a_k \end{cases} \quad (4)$$

Substituting Equation 4 into Equation 3, we can calculate the posterior probability of each action in A being optimal, and can set $a_{best} = \arg\max_a P(a = a^*|V_t)$. In an approach reminiscent of both ϵ-greedy exploration and probabilistic policy reuse (Fernández and Veloso 2006), the agent selects action a_{best} with probability $P(a_{best} = a^*|V_t)$, and otherwise acts as if $E_t = \varnothing$. This allows for a trade-off between following advice and exploring as normal, where the former is more likely if the agent is confident that a_{best} is optimal.

In the above formulations, we have assumed that the estimated $\mathbb{E}[\rho^{(e)}]$ accurately represents the underlying reliability of the expert e. As we discuss in Section 3.4 however, such an assumption is not always practical. Erring on the side of caution, we can compensate for the overestimation of the reliability of particularly bad experts by introducing a threshold parameter $T \in [0, 1]$, such that if $P(a_{best} = a^*|V_t) < T$, the agent acts without advice. This approach ensures that the agent will only follow advice if it is sufficiently confident that the advice is correct.

3.4 Updating Reliability Estimates

Finally we discuss how each expert's reliability estimates are updated as they advise the agent and as the agent interacts with the environment. After selecting some action a_t, the agent receives reward r_t and some subset of experts offer their advice for state s_t. The learning algorithm then uses $\langle s_t, a_t, r_t \rangle$ to update its policy. The agent must now update $P(\rho^{(e)})$ for each expert (if any) that offered its advice for s_t. Suppose expert e advises action $a^{(e)}$. Using the agent's own learned information (e.g. a Q function), it can estimate $EU(a|s_t)\ \forall a \in A$ to determine if $a^{(e)}$ is the optimal action for s_t. Early in training the agent's own understanding of the environment is limited, and so these evaluations will be poor. As the agent learns however, the accuracy of these evaluations will improve. Poor estimates may also be the result of violating the assumptions listed in Section 3.

Across t trials, let $n^{(e)}$ denote the number of times the advice of expert e has been evaluated, with $x^{(e)}$ denoting the number of optimal evaluations and $n^{(e)} - x^{(e)}$ denoting the number of suboptimal evaluations. For ease of readability we omit the superscript denoting expert e. In order to update the reliability estimate, we wish to set the beta distribution $Beta_\rho[\alpha, \beta]$ to be equal to $P(\rho|x)$, which by Bayes rule equals the following:

$$Beta_\rho[\alpha, \beta] = P(\rho|x) = \frac{P(x|\rho)P(\rho)}{\int_0^1 P(x|\rho)P(\rho)d\rho}. \quad (5)$$

As x and $n - x$ represent counts of optimal and suboptimal evaluations respectively, a natural choice is to model the likelihood $P(x|\rho)$ as a binomial distribution $B_x[n, \rho]$ (Etz 2018). As we wish to model the posterior $P(\rho|x)$ as a beta distribution, we can model the prior $P(\rho)$ as a beta distribution $Beta_\rho[\alpha_0, \beta_0]$, which is conjugate to the binomial likelihood $P(x|\rho)$ (Etz 2018). The prior parameters α_0 and β_0 can be thought of as prior counts of x and $n - x$ respectively. Therefore, α_0 may be thought of as the number of times (prior to the start of training) that the expert's advice was evaluated to be correct, and β_0 the number of times it was evaluated to be incorrect.

Substituting these distributions into Equation 5 and taking advantage of the fact that $P(\rho)$ is conjugate to $P(x|\rho)$, we arrive at

$$P(\rho|x) = Beta_\rho[x + \alpha_0, n - x + \beta_0]. \quad (6)$$

Given the parameters of Equation 6, we can calculate the expected value

$$\mathbb{E}[\rho] = \frac{x + \alpha_0}{n + \alpha_0 + \beta_0}.$$

Thus, as the agent encounters states for which expert e has given advice, it need only update $n^{(e)}$ and $x^{(e)}$ and recompute $\mathbb{E}[\rho^{(e)}]$, which can be used in future decision-making.

4 Experiments

4.1 Experiment Set-Up

Having outlined the CLUE framework, we now present a number of experiments to show that **a)** when advised by at least one reliable expert, CLUE outperforms an equivalent unassisted agent, **b)** when advised by unreliable experts who are likely to give incorrect advice, CLUE asymptotically converges to the same threshold of convergence as that achieved by an equivalent unassisted agent, thereby being robust against incorrect advice, and **c)** when advised by multiple experts with different degrees of reliability, CLUE is correctly able to rank experts by their reliability and exploit the information obtained from consensus and contradiction to improve performance.

Additional experiments are presented in the supplementary document.

Environment To show that the performance of CLUE generalises, we run experiments on multiple, randomly generated environments. To generate these environments, we create influence diagrams (IDs) (Howard and Matheson 2005), whose state variables V_S and action variables V_A define the state- and action-space respectively, with random conditional probability distributions, utility functions and graph structures. We use a modified version of the ID implementation provided by Poole and Mackworth (2017). Each variable has a binary domain $\{0, 1\}$, so that $|S| = 2^{|V_S|}$ and $|A| = 2^{|V_A|}$. In order to ensure that each ID represents a well-formed SSDP, we restrict the graph structure to a directed acyclic graph and ensure that all state nodes are parents of action nodes (so that the problem is fully observable), all action nodes are parents of the reward node (so that all actions have some effect on the reward), no action nodes are descendants of another action node (so that only a single round of decision-making occurs), and that no state nodes are children of an action node (as states are observed before decision-making). Rewards are scaled between -1 and 1, so that results across environments are easily comparable.

Agents We compare a number of agents. The *True Policy Agent* always selects the optimal action for any state, thus representing the upper-bound performance any agent can achieve. The *Baseline Agent* is an action-value learner with ϵ-greedy exploration, as outlined in Section 2.1. $Q_0(s, a) = 0\ \forall s \in S, a \in A$, $\alpha = \frac{1}{k(s,a)}$, where $k(s, a) \geq 1$ is a count of the number of times action a has been selected for state s, and ϵ decays from 1 to 0 at a constant rate across the first 80% of trials (Sutton and Barto 2018).

As representative of the existing work in ARL, which assumes only a single reliable expert, we have the *Naïve Advice Follower* (*NAF*), which is identical to the Baseline Agent except that it will always follow any advice it has received for a given state. If it has received multiple pieces of advice for that state, it will select one of those pieces of advice to follow with uniform probability. For CLUE, each expert is modelled by a beta prior with $\alpha, \beta = 1$, and the decision-making threshold T is set to $\frac{2}{|A|}$.

Experts All experiments are conducted with software experts. Many ARL approaches limit the number of interactions between experts and agents, so as to simulate the potential cost of communication, and thus conditions are imposed upon the expert to ensure it gives advice where it is most needed (Torrey and Taylor 2013). In this work, we adopt the conditions outlined by Innes and Lascarides (2019). Firstly, we force the expert to wait at least μ trials between advice utterances, thus restricting the total number of interactions. We refer to μ as the *interval parameter*. Secondly, the expert may only offer advice if:

$$\sum_{t' \leq i \leq t} \frac{EU(a_i^*|s_i) - EU(a_i|s_i)}{t - t'} \geq \gamma,$$

where t is the current trial, t' is the last trial for which expert e gave advice, a_i^* is the optimal action for trial i, a_i is the action taken by the agent in trial i, and γ is a *tolerance parameter* that controls how tolerant an expert is of suboptimal performance by the agent. This condition ensures that the expert will only intervene if the agent is under-performing to a significant degree.

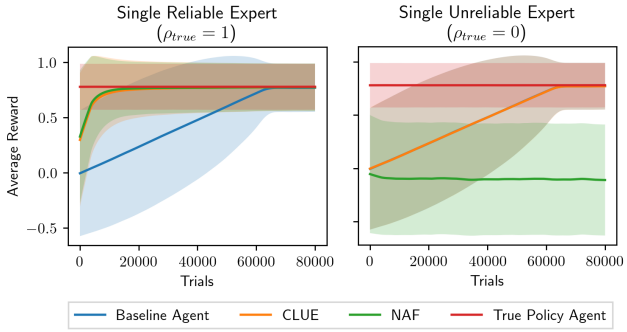

Figure 2: A comparison of agent performance, advised by two panels, a *Single Reliable Expert* ($\rho_{true} = 1$) and a *Single Unreliable Expert* ($\rho_{true} = 0$). Note that for the single unreliable expert, the Baseline Agent and CLUE have near-identical performance, and thus the CLUE curve lies on top of the Baseline curve.

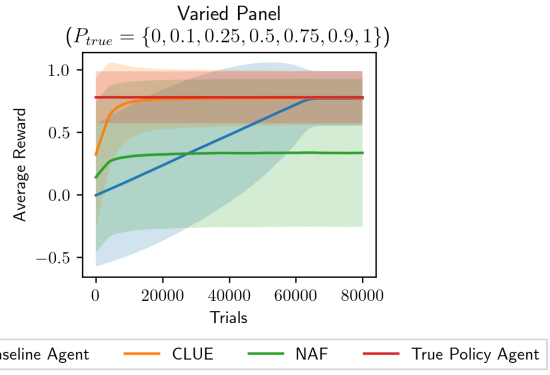

Figure 3: A comparison of agent performance, advised by a *Varied Panel* ($P_{true} = \{0, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$).

In order to simulate reliability, each expert e is controlled by a *true reliability parameter* $\rho^{(e)}_{true}$. When offering advice, the expert will advise the optimal action a^* (obtained from a "ground truth" model of the environment) with probability $\rho^{(e)}_{true}$, or else will randomly advise any other action. Thus an expert with $\rho^{(e)}_{true} = 1$ is reliable, while one with $\rho^{(e)}_{true} = 0$ never advises the optimal action.

4.2 Panel Compositions

In this set of experiments, we compare the reward obtained in each trial by the agents advised by different panels of experts. The rewards obtained by the agents training over $80,000$ trials across 100 random environments with 10 state variables ($|S| = 1024$) and 3 action variables ($|A| = 8$) are averaged and plotted against trials. For legibility, the resulting graphs are smoothed using LOWESS smoothing (Cleveland 1981). Shaded areas represent one standard deviation above and below the average curve. We compare the performance of each agent with three panels of experts. The first, a *Single Reliable Expert*, consists of one expert that always gives correct advice ($\rho_{true} = 1$). The second, a *Single Unreliable Expert*, consists of one expert that always gives incorrect advice ($\rho_{true} = 0$). The third, a *Varied Panel*, consists of seven experts with varying degrees of unreliability ($P_{true} = \{0, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$). Performance with the single-expert panels is plotted in Figure 2, and performance with the varied panel is plotted in Figure 3.

In the first experiment, where $\rho_{true} = 1$, both NAF and CLUE outperform the Baseline Agent, with NAF converging particularly quickly, demonstrating the power of existing ARL methods when the assumption of reliability holds. As CLUE does not assume reliability and is therefore more cautious, it does not converge as quickly, although its performance is very close to that of NAF. A demonstration of the robustness of CLUE comes in the second experiment, where $\rho_{true} = 0$. In this scenario, NAF exclusively follows suboptimal advice and therefore performs exceptionally poorly, failing to converge to the optimal policy. CLUE, on the other hand, correctly identifies that the advice is poor and learns to ignore it, and thus has performance almost identical to the Baseline Agent.

In the third experiment, with the varied panel, the performance of NAF lies somewhere between the two single expert cases, as it receives a mix of advice including optimal and suboptimal actions, and cannot discern which advice is advantageous to follow. However, CLUE converges to the optimal policy even faster than it did in the case of a single reliable expert, comparable to the performance of NAF in the same case. This indicates that not only is CLUE learning to assess which experts are worth following and which are not, it is also benefiting from higher confidence in that assessment as a result of more advice collected from a wide range of experts.

4.3 Reliability Estimates

In order to further examine the results obtained in Section 4.2, we now compare the value of $\mathbb{E}[\rho]$ for each expert in each panel across the same $80,000$ trials as in the previous experiments. As before, results are averaged over 100 runs in different randomly generated environments and the resulting plots are smoothed using LOWESS smoothing (Cleveland 1981). Results for the single reliable expert and single unreliable expert are presented in Figure 4, and results for the varied panel are presented in Figure 5.

For the single expert cases, the value of $\mathbb{E}[\rho]$ converges towards the correct value of ρ_{true} (1 and 0 respectively), with the final estimates being $\mathbb{E}[\rho] = 0.995$ for the single reliable expert and $\mathbb{E}[\rho] = 0.005$ for the single unreliable expert. For the varied panel, each expert is correctly ranked according to their reliability and the value of $\mathbb{E}[\rho^{(e)}]$ for each expert e correctly converges towards the true value of $\rho^{(e)}_{true}$, even faster than the single expert cases.

4.4 Alternate Approach to Expert Simulation

In this set of experiments, we consider a different approach to simulating experts. In this approach, each expert only observes a subset of state variables in the ID that represents the

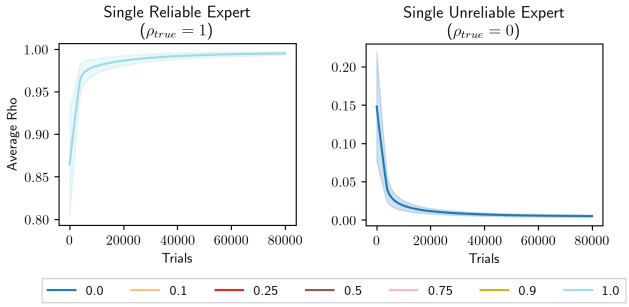

Figure 4: The value of $\mathbb{E}[\rho]$ over time as the agent learns, advised by the single reliable expert ($\rho_{true} = 1$) and single unreliable expert ($\rho_{true} = 0$).

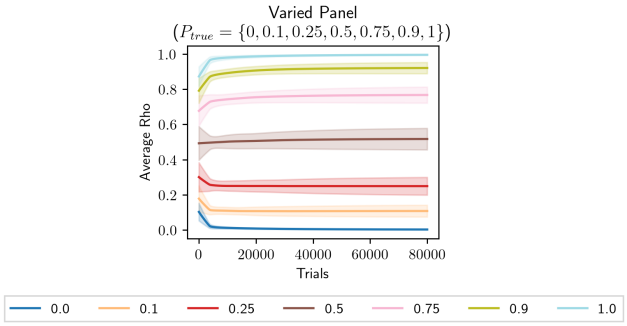

Figure 5: The value of $\mathbb{E}[\rho]$ over time as the agent learns, advised by the varied panel ($P_{true} = \{0, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$). The legend shows the value of $\rho^{(e)}_{true}$ for each expert.

environment, and must advise an action based on this partial observation. Thus an expert with 0 hidden variables is reliable and an expert with $|V_S|$ hidden variables always advises the action most likely to be optimal given no observation of the state. A comparison of the performances of CLUE and NAF in an environment where $|V_S| = 7$ and $|V_A| = 3$, averaged over 10 runs, is given in Figure 6. For the sake of clarity, the shaded areas are removed.

Here CLUE outperforms the Baseline Agent when the number of hidden state variables is less than $|V_S|$, as in these cases the expert is more likely to advise the optimal action than any other action, and performs on par with the Baseline Agent when the amount of information that can be gained from the expert is minimal. NAF, on the other hand, only converges to the optimal policy when the expert is reliable, performing poorly otherwise.

5 Conclusion and Future Work

This work presented the CLUE framework for learning SSDPs with policy advice from multiple, potentially unreliable experts. In particular, our contributions consist of a method of modelling and updating reliability estimates for each expert and, using these estimates to combine policy advice to inform action-selection. Our results show that CLUE maintains the benefits of traditional ARL approaches when advised by reliable experts but is robust to experts being unreliable to some degree. Our results also show that CLUE is able to correctly rank experts by their reliability, and exploit consensus and contradictions among experts to improve performance.

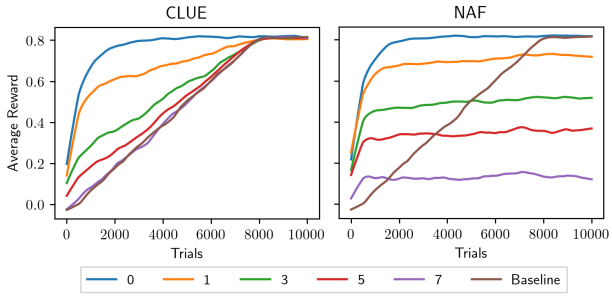

Figure 6: A comparison of agent performance advised by a single expert with varying numbers of hidden state variables. The legend denotes the number of hidden state variables. The Baseline Agent is also given for comparison.

This work may allow for easier integration of external information in the learning process, ultimately contributing towards tackling more complex problems with greater sample efficiency. The explicit modelling of expert reliability allows for a more transparent decision-making process, as it can easily be ascertained why a CLUE agent did or did not follow a given piece of advice. As with all RL approaches, care must be taken to ensure an ethical process of data gathering from real world interactions (such as in the medical diagnosis example). Additionally, any application of CLUE that uses human experts should ensure that advice is elicited ethically, without exploitation and in a manner that respects participants' privacy.

A natural future extension to CLUE would be to extend the framework to the full, episodic RL problem. Such an extension would need to take into account delayed rewards when evaluating the advice given by experts. An agent in this setting would also need to decide between following entire policies advised by an expert and following single actions or small sequences of actions.

Other possible future research includes relaxing the assumptions given in Section 3, particularly the assumption of uniform reliability across the state space. Relaxing this assumption would almost certainly require a more complex model of the reliability of each expert.

Finally, it remains to be seen how well a CLUE agent would perform in real world environments with the advice of human experts. Further research is needed in these areas to ascertain how well CLUE generalises to these settings.

References

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135. PMLR.

Allesiardo, R.; Féraud, R.; and Bouneffouf, D. 2014. A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*, 374–381. Springer.

Bianchi, R. A.; Ribeiro, C. H.; and Costa, A. H. 2004. Heuristically Accelerated Q–Learning: a new approach to speed up Reinforcement Learning. In *Brazilian Symposium on Artificial Intelligence*, 245–254. Springer.

Bignold, A.; Cruz, F.; Taylor, M. E.; Brys, T.; Dazeley, R.; Vamplew, P.; and Foale, C. 2020. A Conceptual Framework for Externally-influenced Agents: An Assisted Reinforcement Learning Review. *arXiv preprint arXiv:2007.01544*.

Burke, P.; and Klein, R. 2020. Confident in the Crowd: Bayesian Inference to Improve Data Labelling in Crowdsourcing. In *2020 International SAUPEC/RobMech/PRASA Conference*, 1–6. IEEE.

Cleveland, W. S. 1981. LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *American Statistician*, 35(1): 54.

Efthymiadis, K.; Devlin, S.; and Kudenko, D. 2013. Overcoming erroneous domain knowledge in plan-based reward shaping. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 1245–1246. Citeseer.

Etz, A. 2018. Introduction to the concept of likelihood and its applications. *Advances in Methods and Practices in Psychological Science*, 1(1): 60–69.

Fernández, F.; and Veloso, M. 2006. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 720–727.

Gao, Y.; Xu, H.; Lin, J.; Yu, F.; Levine, S.; and Darrell, T. 2018. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313*.

Gimelfarb, M.; Sanner, S.; and Lee, C.-G. 2018. Reinforcement learning with multiple experts: A bayesian model combination approach. *Advances in Neural Information Processing Systems*, 31: 9528–9538.

Griffith, S.; Subramanian, K.; Scholz, J.; Isbell, C. L.; and Thomaz, A. L. 2013. Policy shaping: Integrating human feedback with reinforcement learning. Georgia Institute of Technology.

Heckerman, D. E.; and Nathwani, B. N. 1992. An evaluation of the diagnostic accuracy of Pathfinder. *Computers and Biomedical Research*, 25(1): 56–74.

Howard, R. A.; and Matheson, J. E. 2005. Influence diagrams. *Decision Analysis*, 2(3): 127–143.

Huo, X.; and Fu, F. 2017. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11): 171377.

Innes, C.; and Lascarides, A. 2019. Learning Structured Decision Problems with Unawareness. In *International Conference on Machine Learning*, 2941–2950.

Kao, H.-C.; Tang, K.-F.; and Chang, E. Y. 2018. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Keswani, V.; Lease, M.; and Kenthapadi, K. 2021. Towards Unbiased and Accurate Deferral to Multiple Experts. *arXiv preprint arXiv:2102.13004*.

Knox, W. B.; and Stone, P. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture*, 9–16.

Langford, J.; and Zhang, T. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 817–824. Citeseer.

Lauritzen, S. L.; and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2): 157–194.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Masegosa, A. R.; and Moral, S. 2013. An interactive approach for Bayesian network learning using domain/expert knowledge. *International Journal of Approximate Reasoning*, 54(8): 1168–1181.

Owen, C. E. B. 2008. *Parameter estimation for the beta distribution*. Master's thesis, Brigham Young University-Provo.

Poole, D. L.; and Mackworth, A. K. 2017. Python code for Artificial Intelligence: Foundations of Computational Agents. *Version 0.7*, 6.

Shelton, C. 2000. Balancing multiple sources of reward in reinforcement learning. *Advances in Neural Information Processing Systems*, 13: 1082–1088.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Taylor, M. E.; and Chernova, S. 2010. Integrating human demonstration and reinforcement learning: Initial results in human-agent transfer. In *Proceedings of the Agents Learning Interactively with Human Teachers AAMAS workshop*, 23. Citeseer.

Taylor, M. E.; and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul): 1633–1685.

Thomaz, A. L.; Hoffman, G.; and Breazeal, C. 2005. Real-time interactive reinforcement learning for robots. In *AAAI 2005 workshop on human comprehensible machine learning*.

Torrey, L.; and Taylor, M. 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 1053–1060.

Varatharajah, Y.; Berry, B.; Koyejo, S.; and Iyer, R. 2018. A Contextual-bandit-based approach for informed decision-making in clinical trials. *arXiv preprint arXiv:1809.00258*.

Yi, S. K. M.; Steyvers, M.; Lee, M. D.; and Dry, M. J. 2012. The wisdom of the crowd in combinatorial problems. *Cognitive science*, 36(3): 452–470.