Criticality-Based Advice in Reinforcement Learning

Yitzhak Spielberg (yspielb@gmail.com)
Ariel University
Israel

Amos Azaria (amos.azaria@gmail.com)
Ariel University
Israel

Abstract

One of the ways to make reinforcement learning (RL) more efficient is by utilizing human advice. Because human advice is expensive, the central question in advice-based reinforcement learning is, how to decide in which states the agent should ask for advice. To approach this challenge, various advice strategies have been proposed. Although all of these strategies distribute advice more efficiently than naive strategies (such as choosing random states), they rely solely on the agent's internal representation of the task (the action-value function, the policy, etc.) and therefore, are rather inefficient when this representation is not accurate, in particular, in the early stages of the learning process. To address this weakness, we propose an approach to advice-based RL, in which the human's role is not limited to giving advice in chosen states, but also includes hinting apriori (before the learning procedure) which sub-domains of the state space require more advice. Specifically, we suggest different ways to improve any given advice strategy by utilizing the concept of critical states: states in which it is very important to choose the correct action. Finally, we present experiments in 2 environments that validate the efficiency of our approach. **Keywords:** interactive machine learning; reinforcement learning

Introduction

The learning process of Reinforcement Learning (RL) agents in complex environments is often very slow. One of the ways to speed up this process is by providing advice to the learning agent. There exist two categories of advice strategies: General advice and contextual advice. Strategies that fall into the first category (general advice), usually utilize expert demonstrations, which should be available before the start of the learning process. In contrast, strategies that fall into the second category (contextual advice), ask a human expert for action advice in individual states during the learning process. This paper proposes an improvement for advice strategies that fall into the second category.

Human advice requires time and effort from the human expert and thus is considered expensive. Therefore, the central challenge can be formulated as follows: Given an RL agent that is learning according to a given RL algorithm and a limited advice budget (amount of available advice), distribute the advice budget such that the agent learns the given task as fast as possible. To tackle this challenge, the learning agent needs a criterion for deciding in which states it should ask for advice. The literature on advice-based RL proposes a variety of such criteria (see the "Related Work" section).

In most advice strategies found in the literature, the criteria used for selecting advice states (states in which the agent asks for advice) are based solely on *the agent's* model of the policy or the Q-function. In uncertainty-based advice (Da Silva, Hernandez-Leal, Kartal, & Taylor, 2020), for example, the selection criterion is the variance of the head outputs of the multi-headed Q-function model. Although advice strategies that use this type of criteria are usually more efficient than primitive advice strategies, such as distributing advice randomly or asking for advice in every state until the advice budget is finished, all of these strategies suffer from a major problem: they are based only on the current understanding of the task *by the agent*. This is a crucial fact because the agent's understanding of the task can be rather poor—especially during the early stage of the learning process. Consequentially, it is likely that in the early stages of the learning process, when advice is most needed, the agent will not be very good at selecting those states in which advice would be most helpful.

The approach proposed in this paper addresses the weakness of most advice strategies mentioned above by including the human expert into the advice framework more extensively. Whereas, in most advice strategies the expert is utilized solely for giving action advice in individual states, in the suggested approach the expert has the additional role to mark sub-domains of the state space in which there might be a strong need for advice. That is, the learning agent utilizes the human expert in two ways: Firstly, to receive advice in individual states; Secondly, to help selecting states in which to ask for advice.

In order to determine states in which advice might be very helpful, we use the concept of state criticality that was introduced in (Spielberg & Azaria, 2019). State criticality is a measure of variability in the expected return of the available actions. States that have a high variability in the expected returns should receive a high criticality value while states with low variability in the expected returns should receive a low criticality value. State criticality is a subjective measure, that is assigned by a human designer of the criticality function (the function that assigns a criticality value to each state from the state space) and thus does not require any estimate of the Q-function.

In summary, the major contributions of this paper are the following:

1. We introduce criticality-based advice: An approach to advice-based RL in which the human expert not only pro-

vides action advice at individual states but also helps the learning agent to select advice states using state criticality.

2. We present experiments in 2 environments (Gridworld and Atari Pong) that prove the efficiency of criticality-based advice.

Related Work

The current piece of research is closely related to multiple sub-domains of the RL domain: advice-based RL, advice strategies that are based on uncertainty metrics of the learning agent, and RL algorithms that use the notion of critical states. This section reviews literature that is related to these three sub-domains.

In the context of human-aided RL, one of the most popular techniques for speeding up the learning process is advice-based RL (Thomaz & Breazeal, 2007; Tenorio-Gonzalez, Morales, & Villaseñor-Pineda, 2010; Cruz, Twiefel, Magg, Weber, & Wermter, 2015). We discuss only several selected algorithms out of the vast amount that appears in the literature (see for example (Knox & Stone, 2009; Griffith, Subramanian, Scholz, Isbell, & Thomaz, 2013)).

Importance-based advice strategies utilize the notion of state importance to select states that require advice (Torrey & Taylor, 2013). Unfortunately, the efficiency of this strategy is compromised by the downside that the Q-function needs to be initiated with strongly negative values. Zimmer et al. succeeded in fixing this downside via an approach in which the advisor is modeled as an RL agent (Zimmer, Viappiani, & Weng, 2014).

Another remarkable approach in advice-based RL combines contextual advice with learning from demonstrations (LfD). In (Nicolescu & Mataric, 2003) and (Rybski, Yoon, Stolarz, & Veloso, 2007) a LfD system is augmented with verbal instructions, in order to make the learning agent perform certain actions during the demonstrations.

Another metric used for the selection of states that require advice is agent uncertainty (Da Silva et al., 2020). Given the many applications of agent uncertainty, several works studied how to define epistemic uncertainty measures. In some of these works, agent uncertainty is calculated via dropout schemes (Chen, Zhou, Chang, Yang, & Yu, 2017) or ensemble of networks (Clements, Robaglia, Delft, Slaoui, & Toth, 2019). In Ad-hoc advising, the uncertainty estimate is based on the number of visits in each state (Silva & Costa, 2019). Ilhan et al. propose a Deep RL version of Ad-hoc advising, estimating visit counts through a deep neural network (Ilhan, Gow, & Perez Liebana, 2019). Alternatively, it is possible to use Bayesian neural networks to estimate the epistemic uncertainty of the agent and to ask for demonstrations based on it (Thakur, Hoof, Higuera, Precup, & Meger, 2019).

While there exists a rich literature on the first two sub-domains mentioned above (Najar & Chetouani, 2020) (advice-based RL and uncertainty-based advice strategies), the notion of critical states is not yet an established notion in the RL domain. To the best of our knowledge, there exist only two papers that discuss the usage of critical states in RL. Spielberg and Azaria introduce the notion of a critical state as a state in which the choice of action has a significant influence on the agent's total reward (Spielberg & Azaria, 2019). This notion is then applied to tackle the challenge of choosing the proper step number in n-step algorithms. In another paper, critical states are utilized for a different purpose: to evaluate the safety of an AI agent or robot (Huang, Bhatia, Abbeel, & Dragan, 2018). Huang et al. advocate that safety can be achieved more efficiently by observing the robot's behaviour in critical situations.

State Criticality

In the context of reinforcement learning, the criticality of a state indicates how much the choice of action in that particular state influences the expected return (Spielberg & Azaria, 2019). State criticality can be defined as a measure of the variability of the expected return with respect to the available actions. The criticality of a state can range from 0 to 1 such that 0 represents no variability between the expected return of the actions (for example, if there is only a single action, or if all actions result in the same expected return), and 1 represents high variability between the expected return of the actions (for example when some actions result in a very high expected return, while other actions result in a very low expected return). The criticality of a state can be linked to the variance of the Q-function with respect to the action values in that state - albeit loosely. Although there the criticality of a state is not uniquely defined by any objective measure, because state criticality is subjective, it should satisfy the minimal requirement that a variance of 0 should result in a state criticality of 0, while a variance greater than 0 should result in a state criticality of greater than 0.

The notion of state criticality is particularly useful in learning situations that include a teacher and a student. An example of such a learning situation is a driving lesson. If a student driver approaches an obstacle on the road, her teacher may state to her that she must watch out, without suggesting exactly which action to take (e.g. slowing down, turning the wheel right or left, etc.). This warning will motivate the student driver to pay more attention to the situation and thereby decrease the risk of a collision. Even in the case that the car will hit that obstacle later, the student will understand that she probably took a wrong action back when the teacher has warned her and therefore, will learn more easily how to behave properly in such a situation. Clearly, the situation of a driving lesson possesses the characteristics of a human-aided reinforcement learning scenario in which the learning agent finds itself in a certain state and needs to choose one action from an array of possible actions. After having been informed about the criticality level of the current state by the human teacher, the learning agent utilizes the criticality information to adjust its learning strategy.

According to the definition above, state criticality is a human centered concept, in the sense that it is a *human* estimate of the spread of consequences with respect to the available actions. Therefore, the definition implies that the criticality function (that is, the function that assigns a criticality level to each state of the environment) of a given environment is not unique, but can be any element from a whole class of functions that are loosely defined by the variance of the expected return (as described above).

Above, state criticality was introduced as a subjective estimate of the variability of the Q-function it was not specified to which policy this Q-function should belong. Considering that the intuitive mind does not operate with explicit policies but rather with high-level intuitive representations of policies this vagueness was introduced on purpose, to ensure that state criticality will be a human-friendly concept. Yet, the optimal policy should appear in the definition of criticality at least in an implicit manner, since ultimately it is *that* policy, that the agent is supposed to learn. This might be achieved by instructing the criticality provider that the criticality levels should relate to a policy that is close to optimal. Such an instruction is likely to be friendly to the criticality function provider since it is rather natural to think about an almost optimal policy when estimating criticality levels of states.

Criticality-Based Advice

While expert advice helps RL agents to learn more efficiently, it is also rather expensive. Hence, there is a need for strategies that select states in which advice is most useful. There exists a variety of techniques that are used to execute this selection task. However, most of them only utilize the agent's knowledge and are therefore not very efficient in the early stages of the learning process. The approach that we propose, in contrast, also uses state criticality, which is an aspect of a human's knowledge about the learning environment. This section describes how to use state criticality to make advice-based RL more efficient.

The novel advice strategy that will be introduced in this paper, criticality-based advice (CBA), utilizes a criticality function which is a function that assigns a criticality level to every state in the environment, that has been generated by a human expert apriori—before the beginning of the RL agent's learning process. This paper introduces two versions of criticality-based advice: The plain version (p-CBA) and the meta version (m-CBA). p-CBA is based on criticality alone, which means that the learning agent will receive advice in a given state if and only if the criticality of that state is sufficiently high. The more complex version, m-CBA, operates on top of an underlying advice strategy. In m-CBA, the criterion that is being used to select advice states is a combination of the criterion used by the underlying advice strategy and the state criticality.

For m-CBA, there are various ways to combine state criticality with the metric of the underlying advice strategy, such as agent uncertainty in the case of (Da Silva et al., 2020). The most straightforward way to do this is to use the logical *and* operator (we will call this*logicand* approach). In this approach, a state will be selected for advice if and only if it is considered an advice state by the underlying advice strategy *and* its criticality is sufficiently high. The benefit of this approach is, that the agent will not waste its advice budget on states in which the choice of action has only a small impact on the total reward. An efficient alternative way to accomplish such a combination could be multiplication: to multiply the metric of the underlying advice strategy with state criticality. For this type of combination, the selection thresholds for agent uncertainty and state criticality should be fused into one threshold by multiplication too. Both approaches—the *logicand* approach and the multiplicative approach—are tested in this paper.

To determine whether the criticality of a state is sufficiently high it is necessary to use a threshold with a value between 0 and 1. This threshold can be either stochastic or fixed. The stochastic threshold is a threshold that is being sampled in each state that the agent visits. In the simplest case, this threshold could be sampled from a uniform distribution over the $[0,1]$ interval. When CBA is used with a fixed threshold we face the challenge of choosing an appropriate threshold. Clearly, in the case of a binary criticality function, which produces only 2 possible values - 0 or 1 - the choice of the threshold is irrelevant. However, in the case when the criticality function is continuous, it is not obvious how to choose a proper criticality threshold. In this case, one principle that might be used to determine an appropriate threshold could state that the portion of the state space that is below the threshold should be sufficiently large. Although this principle does not guarantee the efficiency of CBA, it prevents inefficient criticality thresholds: those thresholds that would rule out only a small portion of potential advice states.

```
**The CBA algorithm** (p-CBA)
trh: stochastic or deterministic criticality treshold
n=0:
budg: advice budget
RLalg: underlying RL algorithm (e.g. Q-Learning)
while $S \neq Terminal$
    if ($crit(S) > trh$) and ($n < budg$)
        ask for advice
        $a$=advice
        n+=1
    else
        select $a$ according to RLalg
    perform $a$
    update all stuff (Qfunc, policy etc.) according to RLalg
```

Experiments

This section describes experiments that prove the efficiency of criticality-based advice. Two environments serve as testbeds for the experiments: a gridworld environment and the Atari Pong environment. All experiments presented in this section were performed on a Nvidia Titan xp GPU.

Gridworld

The first set of experiments was performed in a gridworld environment (fig. 1), in which the agent starts at the bottom left corner and needs to reach the goal state located at the top left corner. The agent receives a reward of 4 when it reaches the final goal. The red circles represent radioactive states which are associated with a small negative reward of -0.01 and the black blocks represent walls. There is no negative reward for each step but the agent will strive to reduce the number of steps because the discount factor is $\gamma = 0.9$ In order to obtain the maximal total reward (~ 1.15), the agent needs to walk through the radioactive states. The total reward of the trajectory that circumvents the wall is much smaller (~ 0.4).

In the gridworld experiments, we tested p-CBA and used a stochastic criticality threshold sampled from a uniform distribution over the $[0,1]$ interval. The criticality function assigned a criticality of 1 to all radioactive states and their neighbours and a criticality of 0 to all other states. The underlying learning algorithm used for the gridworld experiments was plain Q-learning. Moreover, we used importance-based advice ((Torrey & Taylor, 2013)) as the alternative advice strategy that competed against criticality based-advice. Importance-based advice was chosen as the alternative advice strategy because it is one of the more modern advice strategies and also because this strategy performs particularly well with Q-learning. We perform two sets of experiments each one with a different advice budget (200 and 500).

To compare the different learning methods, each method was simulated 100 times such that each simulation was based on a different random seed. The plots in fig. 2 shows the average learning curves of the four learning methods: plain Q-learning, the two versions of importance-based advice with different importance thresholds (0.02 and 0.05) and p-CBA. The shaded buffers surrounding the curves represent the 95% confidence intervals. Several findings can be derived from the plots. Firstly, the plots show that all three advice-based methods outperform the plain Q-learning method. Secondly, p-CBA outperforms both versions of importance-based advice – which is the most important observation in our context. While for the smaller advice budget p-CBA dominated importance-based advice by a small margin, this margin was more significant for the larger advice budget.

Pong

The second testbed for our novel advice strategy was the Atari Pong environment [1]. In contrast to the gridworld experiments, where we tested p-CBA, here we experimented with m-CBA. The underlying advice strategy was uncertainty-based advice (Da Silva et al., 2020) which is one of the most modern and efficient advice strategies for DQN type learners (such as DDQN, Rainbow, BDQN etc.). The most important parameter in this advice strategy is the uncertainty threshold, which is used to select advice states. Only those states whose agent uncertainty is above the threshold are selected as advice states.

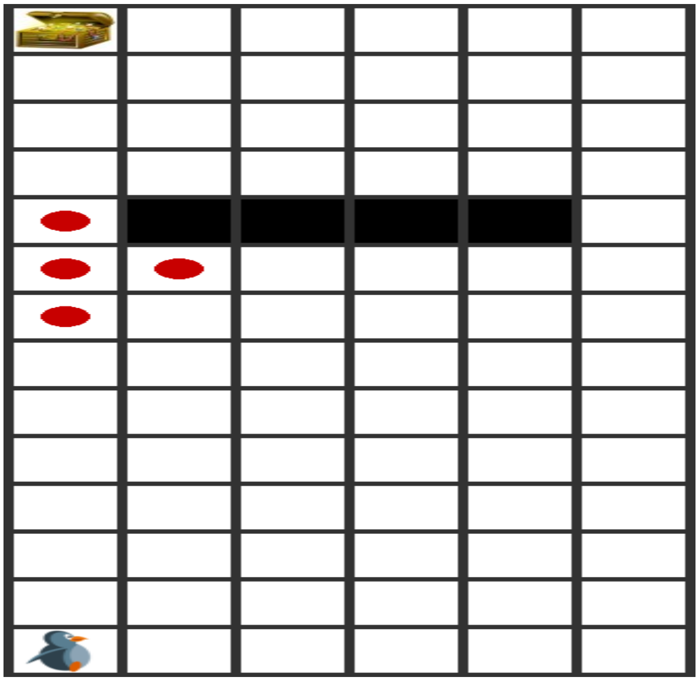

Figure 1: In this gridworld the agent starts at the bottom left corner and the goal is located at the top left corner. Red circles are radioactive states and black tiles are walls.

Before starting the main series of simulations, we first ran a separate series of experiments to determine the uncertainty threshold for BDQN in the Pong environment. The results of these experiments suggested that the agent performed particularly well with an uncertainty threshold of $trh_{uncert} = 0.04$, so we decided to use this value for the experiments. Furthermore, the advice budget was set to $150K$, which is approximately 50% of the total advice consumption of an unlimited advice agent until it reaches almost optimal performance.

Aside from the choice of the underlying advice strategy and the advice budget, another important choice is the criticality function. We use a continuous criticality function that reflects an intuitive understanding of the game dynamics. The principle that directs the design of the criticality function is that the criticality of a state should be a monotonically decreasing function of the minimal distance that the ball needs to cover to reach the learning agent. Hence, a state in which the ball was just hit by the agent has a criticality close to 0, a state in which the ball is close to the opponent's baseline has a criticality of about 0.5 and a state in which the ball is moving towards the agent and is very close to the agent's baseline has a criticality close to 1. When the ball moves towards the agent, this criticality function can be expressed by the formula:

$$crit(s) = 1 - \frac{dist(ball\ to\ agent's\ baseline) - 1}{2 * (field\ length - 1)} \qquad (1)$$

[1] https://gym.openai.com/envs/Pong-v0/

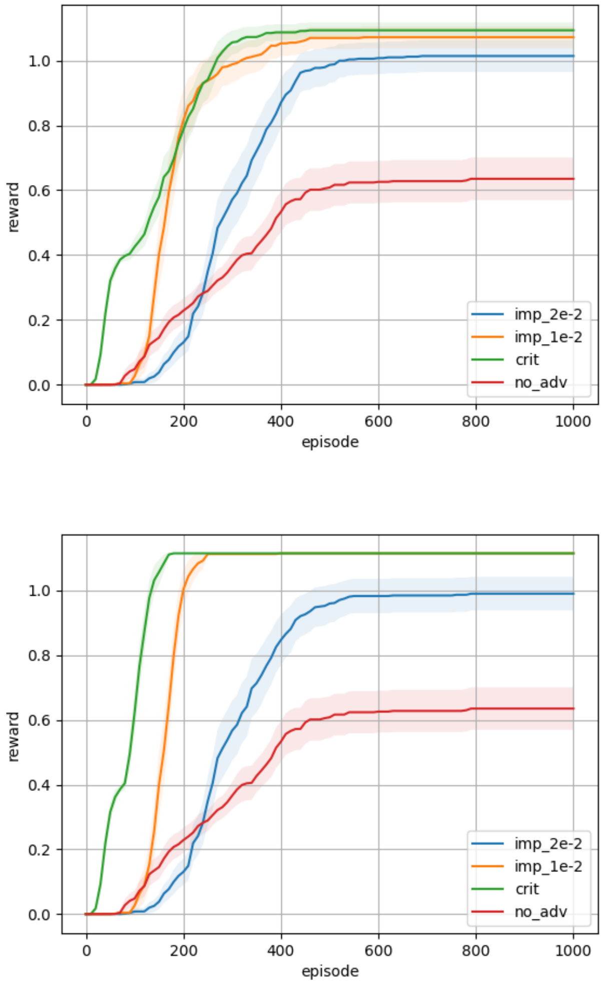

Figure 2: Learning curves for the gridworld environment for 2 advice budgets (top: 200, bottom: 500). For both budgets, the p-CBA agent (crit) outperforms the 2 importance-based agents and the plain Q-Learning agent (no_adv)

and when the ball moves away from the agent - by the formula:

$$crit(s) = \frac{dist(ball\ to\ agent's\ baseline) - 1}{2 * (field\ length - 1)}. \tag{2}$$

We use two versions of m-CBA corresponding to two different ways of integrating state criticality with the metric of the underlying advice strategy: the *logicand* version (BDQN-crit1) and the multiplicative version (BDQN-crit2).

In the *logicand* version, a state is selected for advice if both the agent uncertainty and the criticality are sufficiently high ($crit >$ criticality threshold (trh_{crit}) and $uncert >$ uncertainty threshold (trh_{uncrt})), with a criticality threshold of 0.5. In the multiplicative version, a state was selected for advice if the product $crit(s) * uncertainty(s)$ was greater than the product between the criticality threshold and the uncertainty threshold $trh_{crit} * trh_{uncert}$. This choice of the threshold accomplishes the original motivation behind the multiplicative combination: a state with sufficiently high criticality can be selected for advice, even if the uncertainty is relatively small.

To evaluate the efficiency of m-CBA, two baseline strategies are used. The first strategy is BDQN without advice (BDQN-plain), and the second is BDQN with uncertainty-based advice (BDQN-adv). Both strategies are tested experimentally.

To compare the learning curves of the different advice strategies, every strategy is executed 5 times—each time with a different random seed. The postprocessing procedure consists of two steps. First, the learning curves are smoothened, using a moving average with a window size of 5. Then, they are synthesized into a single learning curve via averaging. The resulting learning curves of the algorithms that participate in the comparison are shown in fig. 3.

There are several notable observations that can be made upon a closer look at the plot. Firstly, the plot shows that BDQN-adv outperforms BDQN-plain. This anticipated result confirms the usefulness of advice in the Atari Pong environment. The second observation is related to BDQN-adv and BDQN-crit1. It can be seen from the plot, that BDQN-crit1 outperforms BQQN-adv in the early stages of the learning process but does not retain this advantage throughout the entire learning process. The third remarkable observation is that BDQN-crit2 strongly outperformed both BDQN-adv and BDQN-crit1. This can be seen clearly, upon observing how many episodes the algorithms require to reach machine-level performance (a score of 0). While BDQN-adv requires about 700 episodes for achieving machine-level performance, BQQN-crit1 requires about 600 episodes, and BDQN-crit2 requires only about 450 episodes.

Aside from the learning curves, it might be also interesting to observe advice consumption of the various algorithms. The advice consumption curves on fig. 4 correspond to the three advice strategies that were discussed previously. There are several remarkable phenomena that can be observed in the plot. Firstly, the plot shows that BDQN-adv has a very high advice consumption, such that the advice budget is depleted at a relatively early stage of the learning process. In contrast, BDQN-crit1 has the lowest advice consumption of the three algorithms. The corresponding consumption curve is relatively steep at the beginning, flattens out later, and then gains momentum again in the more advanced stage of the learning process. The consumption curve of BDQN-crit2 is located between the two other consumption curves and from the curve it can be implied that BDQN-crit2 runs out of advice at an intermediate stage of the learning process.

Discussion & Conclusion

The current paper introduced the criticality-based advice strategy (CBA) for advice-based RL agents. The central idea of CBA is to use state criticality in order to select advice states more efficiently. In addition, the paper mentioned several

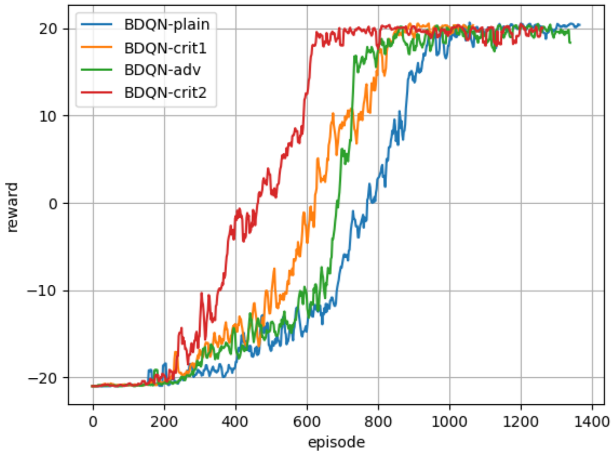

Figure 3: Learning curves of different advice strategies in the Pong environment. BDQN-adv and BDQN-crit1 outperform BDQN-plain.

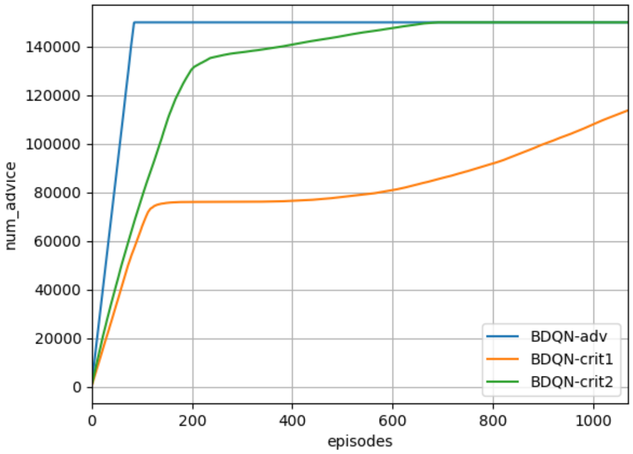

Figure 4: Advice consumption of the various advice strategies in the Pong environment. BDQN-adv runs out of advice quickly. The two other strategies use the advice budget more economically.

ways to combine state criticality with the selection criteria of the underlying advice strategy and described experiments in two environments, which were conducted to test the efficiency of the proposed approach. In this section, we will elaborate on the main conclusions that can be derived from the experiments and on a few interesting observations. We will also consider possible directions for future research.

CBA was tested in two environments. In the gridworld environment, we tested the plain version of the method whereas in the Pong environment we tested the meta version. In every experiment performed, the novel method was able to beat alternative advice strategies. Therefore, the main conclusion that can be drawn from the conducted experiments is that CBA can be considered as a promising method in the domain of advice-based RL.

It might be important to mention one remarkable observation which is related to m-CBA: the fact that the multiplicative variant (BDQN-crit2) outperformed the *logicand* variant (BDQN-crit1) by a significant margin (in Pong). A possible explanation for this phenomenon could be that especially in the beginning of the learning process, states in which advice is very useful might have low uncertainty and thus would not be considered as potential advice states by the underlying advice strategy. However, if the criticality values of these states are sufficiently high, there is a good chance that multiplying the criticality values with the uncertainty values would produce numbers that are sufficiently high to be above the CBA selection threshold used by (the underlying advice strategy augmented with state criticality). Thus BDQN-crit2 might be more successful in selecting proper states for advice in the beginning of the learning process than BDQN-crit1 and this might explain why BDQN-crit2 learns faster than BDQN-crit1.

In this paper, CBA was tested in only two learning environments. Although the experiments indicate that CBA might be an efficient way to improve advice-based RL methods, more research is needed to confirm that the novel strategy is efficient in other environments as well. It might be interesting to test the novel method in more complex environments than Pong, in which the criticality function has strong variations. Specifically, CBA should be tested in environments where the critical states constitute only a small portion of the state space, such as Pacman or Montezuma's Revenge. In these environments, it would be interesting to see whether agent uncertainty will reflect critical states properly by assigning high uncertainty to these states and whether agent uncertainty will be low in uncritical states.

In this paper, CBA operated with a static criticality function which is only a function of the state but not of the current skill level of the learning agent. Although both variants of criticality-based advice with a static criticality function were rather efficient, there might be many environments where a static criticality function might lead to redundant advice. In p-CBA, for example, a state with high criticality will keep on receiving advice even if the advice is no longer necessary. With a policy-dependent criticality function (Spielberg & Azaria, 2019), however, this negative effect could be avoided because the criticality of the state would decrease as the agent becomes more confident in his actions. Furthermore, it might be particularly interesting to compare the policy-dependent criticality to agent uncertainty since both measures are dynamic (they evolve in the course of the learning process) and agent uncertainty can be regarded as a form of policy-dependent criticality.

Acknowledgment

This research was supported in part by the Ministry of Science, Technology & Space, Israel.

References

Chen, L., Zhou, X., Chang, C., Yang, R., & Yu, K. (2017, 09). Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning.. doi: 10.18653/v1/D17-1260

Clements, W. R., Robaglia, B., Delft, B. V., Slaoui, R. B., & Toth, S. (2019). Estimating risk and uncertainty in deep reinforcement learning. *CoRR*, *abs/1905.09638*. Retrieved from http://arxiv.org/abs/1905.09638

Cruz, F., Twiefel, J., Magg, S., Weber, C., & Wermter, S. (2015). Interactive reinforcement learning through speech guidance in a domestic scenario. In *2015 international joint conference on neural networks, IJCNN 2015, killarney, ireland, july 12-17, 2015* (pp. 1–8). IEEE. Retrieved from https://doi.org/10.1109/IJCNN.2015.7280477 doi: 10.1109/IJCNN.2015.7280477

Da Silva, F. L., Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2020, Apr.). Uncertainty-aware action advising for deep reinforcement learning agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 5792-5799.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Proceedings of the 26th international conference on neural information processing systems - volume 2* (p. 2625–2633). Red Hook, NY, USA: Curran Associates Inc.

Huang, S. H., Bhatia, K., Abbeel, P., & Dragan, A. (2018). Establishing appropriate trust via critical states. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3929-3936.

Ilhan, E., Gow, J., & Perez Liebana, D. (2019, 08). Teaching on a budget in multi-agent deep reinforcement learning. In (p. 1-8). doi: 10.1109/CIG.2019.8847988

Knox, W. B., & Stone, P. (2009, September). Interactively shaping agents via human reinforcement: The TAMER framework. In *The fifth international conference on knowledge capture.*

Najar, A., & Chetouani, M. (2020). Reinforcement learning with human advice. A survey. *CoRR*, *abs/2005.11016*. Retrieved from https://arxiv.org/abs/2005.11016

Nicolescu, M., & Mataric, M. (2003, 01). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In (p. 241-248). doi: 10.1145/860575.860614

Rybski, P. E., Yoon, K., Stolarz, J., & Veloso, M. M. (2007). Interactive robot task training through dialog and demonstration. New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1228716.1228724 doi: 10.1145/1228716.1228724

Silva, F., & Costa, A. (2019, 03). A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, *64*. doi: 10.1613/jair.1.11396

Spielberg, Y., & Azaria, A. (2019). The concept of criticality in reinforcement learning. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 251-258.

Tenorio-Gonzalez, A. C., Morales, E. F., & Villaseñor-Pineda, L. (2010). Dynamic reward shaping: Training a robot by voice. In A. Kuri-Morales & G. R. Simari (Eds.), *Advances in artificial intelligence – iberamia 2010* (pp. 483–492). Berlin, Heidelberg: Springer Berlin Heidelberg.

Thakur, S., Hoof, H. V., Higuera, J., Precup, D., & Meger, D. (2019). Uncertainty aware learning from demonstrations in multiple contexts using bayesian neural networks. *2019 International Conference on Robotics and Automation (ICRA)*, 768-774.

Thomaz, A. L., & Breazeal, C. (2007). Robot learning via socially guided exploration. In *2007 ieee 6th international conference on development and learning* (p. 82-87). doi: 10.1109/DEVLRN.2007.4354078

Torrey, L., & Taylor, M. (2013). Teaching on a budget: Agents advising agents in reinforcement learning. *AAMAS*.

Zimmer, M., Viappiani, P., & Weng, P. (2014, 05). Teacher-student framework: A reinforcement learning approach..