Detection of Important States through an Iterative Q-value Algorithm for Explainable Reinforcement Learning

Rudy Milani
Faculty of Computer Science
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
85577 Neubiberg, Germany
rudy.milani@unibw.de

Maximilian Moll
Faculty of Computer Science
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
85577 Neubiberg, Germany
maximilian.moll@unibw.de

Renato De Leone
School of Science and Technology
University of Camerino
via Madonna delle Carceri 9
62032 Camerino, Italy
renato.deleone@unicam.it

Abstract

To generate safe and trustworthy Reinforcement Learning agents, it is fundamental to recognize meaningful states where a particular action should be performed. Thus, it is possible to produce more accurate explanations of the behaviour of the trained agent and simultaneously reduce the risk of committing a fatal error. In this study, we improve existing metrics using Q-values to detect essential states in Reinforcement Learning by introducing a scaled iterated algorithm called IQVA. The key observation of our approach is that a state is important not only if the action has a high impact but also if it often appears in different episodes. We compared our approach with the two baseline measures and a newly introduced value in grid-world environments to demonstrate its efficacy. In this way, we show how the proposed methodology can highlight only the meaningful states for that particular agent instead of emphasizing the importance of states that are rarely visited.

Keywords: Explainable Reinforcement Learning, Safe Reinforcement Learning, Importance Analysis, Important States.

1. Introduction

Reinforcement Learning (RL) has gained a lot of popularity after superhuman performances have been achieved in solving problems with high complexity, for example, playing chess and driving autonomous cars (Isele et al., 2018; Silver et al., 2017). Since then, its application to real-world problems relative to logistics, supply chain and intelligent transportation systems have been addressed (Milani, Moll, and Pickl, 2023; Yan et al., 2022). However, RL agents do not reason about their choices but depend principally on trial-and-error interactions with the environment. Therefore, this creates a problematic situation for human users, who are not able to completely understand and explain their behaviour. In this way, instead of helping in the decision-making process, the algorithm leads to an increase in confusion (Simkute et al., 2021). Moreover, RL agents do not know why a particular action should be preferred over a different one or what the most favourable states to visit are (Sequeira and Gervasio, 2020). Hence, identifying relevant situations to be summarized or focused on to generate a trustworthy explanation becomes a fundamental task. It stands to reason that in many occasions it is not important to know what the RL algorithm would do in all states, but there are a few important circumstances that really matter.

To this end, in this paper, we present the Iterated Q-Value Algorithm (IQVA), an iterated extension of metrics already used in the literature (Bellemare et al., 2016; Torrey and Taylor, 2013) to detect critical states. Through our approach, we were able to discover the states that are, at the same time, important in terms of choice of action and with a high frequency of visits. The proposed methodology was evaluated in grid-world environments and compared to two baseline importance metrics: *advising* and *action gap importance*. Afterwards, a new metric is derived and compared to the ones present in the literature, in both cases, i.e., static and iterative. The results show that using our algorithm, we can drastically reduce the states identified as essential and, in particular, we can discriminate high-risk situations through a simple comparison of values.

URI: https://hdl.handle.net/10125/106551
978-0-9981331-7-1
(CC BY-NC-ND 4.0)

2. Background

We introduce in this section the RL framework for the discrete scenarios that we analyse in the following. At any time step $t = 0, 1, 2, \ldots, T$ the RL agent gets the state $s_t \in \mathbb{S}$ from the environment and chooses action $a_t \in \mathbb{A}$, where $\mathbb{S}$ and $\mathbb{A}$ are the state and action spaces, respectively (Sutton and Barto, 2018). Consequently, the agent receives a numerical reward $r_{t+1} \in \mathbb{R}$ and the environment provides a new state s_{t+1}. The choice of the actions is performed by the agent to maximize the sum of discounted rewards $R_t = \sum_{i=0}^{T-1} \gamma^i r_{t+i+1}$ where $\gamma \in [0, 1]$ is a discounting factor (Sutton and Barto, 2018). Fundamental for the description of the proposed algorithm is the *Q-function* $Q^t_\pi(s, a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$ evaluated under policy π, which is defined as a distribution over actions in given states: $\pi(a|s) = \mathbb{P}(a_t = a|s_t = s)$, where $\mathbb{P}(\cdot)$ represents the probability.

The RL methods considered for our applications are two model-free algorithms: *Q-learning* (Watkins and Dayan, 1992) and *SARSA* (Sutton and Barto, 2018). The general concept behind these methods is the same. At the beginning of the learning process, a matrix defined as *Q-table* ($Q(s_t, a_t) \approx Q^t_\pi(s_t, a_t)$), with dimensions $|\mathbb{S}| \times |\mathbb{A}|$, is initialized as a null matrix. After taking action a_t at state s_t, the Q-table is updated using the following rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Bigg[r_{t+1} + \gamma \max_{a \in \mathbb{A}} Q(s_{t+1}, a) - Q(s_t, a_t) \Bigg],$$

where $\alpha \in (0, 1]$ denotes the learning rate. In the case of SARSA, because we use an on-policy method, we consider the behaviour policy as a target policy in the updating formula. Therefore, the rule is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Bigg[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \Bigg],$$

where a_{t+1} is the action chosen following the behavioural policy.

3. Literature Review

The first studies that chronologically dealt with the introduction of metrics to evaluate the importance of a state were Torrey and Taylor, 2013 and Bellemare et al., 2016. In both cases, they relied on the Q-values related to the specific state to calculate their importance. In particular, we focused on these two measures to build our approach.

Specifically, Torrey and Taylor, 2013 considered the difference between the maximum Q-value and its minimum, that is, $I_A(s_t) = \max_{a \in \mathbb{A}} Q(s_t, a) - \min_{a \in \mathbb{A}} Q(s_t, a)$ as an indicator of state importance. This measure was first introduced by Clouse, 1996 but it was used to approximate a learner's confidence. In fact, this idea was developed in the context of student-teacher Reinforcement Learning as *advising importance* (O. Amir et al., 2016; Torrey and Taylor, 2013), and later applied to the extraction of important trajectories for summarizing the RL agent's behaviour (D. Amir and Amir, 2018).

In the other case, Bellemare et al., 2016 used the *action gap importance* of a state which represents the difference between the maximum and second maximum Q-value, that is, $I_{AG}(s_t) = Q(s_t, a^*) - \max_{a \in \mathbb{A}^*} Q(s_t, a)$ where $a^* = \arg\max_{a \in \mathbb{A}} Q(s_t, a)$, and $\mathbb{A}^* = \mathbb{A} \setminus \{a^*\}$. However, in this case, the initial scope of this measure is to increase the consistency of tabular methods and lessen the effects of approximations and estimation errors using induced greedy policies.

A similar metric was later developed (Huang et al., 2018) by considering the difference between the maximum Q-value and the mean, that is, $I_{\text{MM}}(s_t) = \max_{a \in \mathbb{A}} Q(s_t, a) - \sum_{a \in \mathbb{A}} Q(s_t, a)$. In this manner, Huang et al., 2018 defined critical states as those where acting randomly generates a much worse result than acting optimally.

The last approach relevant for our paper was proposed by Jacq et al., 2022, where the authors introduced the concept of *laziness*, which involves deferring the decision-making process to a default policy in particular simple states. Specifically, they considered as a metric for determining these critical states the *lazy-gap*: the difference between the maximum Q-value and the expected value when the next action is chosen following the default policy, that is, $I_{LG}(s_t) = \max_{a \in \mathbb{A}} Q(s_t, a) - \mathbb{E}_{\pi_D}[Q(s_t, a)]$, where π_D is the default policy.

4. Methodology

In this section, first, we introduce the Iterated Q-value Algorithm and the results obtained using the proposed approach. Afterwards, it is possible to present a new metric derived from the previous ones, in both the static and iterated forms.

4.1. Iterated Q-Value Algorithm

While interesting on their own, none of the aforementioned works explicitly try to identify important states that are recurrently visited in different episodes. In fact, these metrics only focus on the importance of taking the correct action in high-risk states without considering that failing an action that would cause a small change in the return in a state that is frequently visited would lead to an increase in the number of trajectories that are not at the optimal value. Consequently, it is fundamental to pay attention to the frequent states that can occur during multiple episodes.

Algorithm 1 IQVA

```
Require: Number of episodes $E$, discounting factor $\hat{\gamma}$.
I(s) ← 0,   $\forall s \in \mathbb{S}$
for episode $= 1, \cdots, E$ : do
    for $t = 0, 1, \cdots, T$ : do
        Read state $s_t$
        if $train$ then
            Update Q-table
        end if
        Update Importance Vector:
            $I(s_t) \leftarrow I(s_t) + \hat{\gamma}^t I_M(s_t).$
    end for
end for
Normalization:
    $I(s) \leftarrow \frac{I(s)}{E}, \quad \forall s \in \mathbb{S}.$
Return $I$.
```

Therefore, we introduce an iterated algorithm that can bear in mind this aspect. In particular, we define the Iterated Q-Value Algorithm (IQVA), shown in Algorithm 1, considering any Q-value importance metrics I_M. The major assumption of this methodology is that the Q-function must be available; otherwise, it is impossible to compute the presented metrics. At the beginning of our algorithm, we initialize a column vector $I \in \mathbb{R}^{|\mathbb{S}|}$, with the same dimensions as the state space, as a null vector. Successively, we can start an episode. At each time step t, the agent will visit a state $\bar{s}$ then we will update the importance vector in that component following the rule:

$$I(\bar{s}) \leftarrow I(\bar{s}) + \hat{\gamma}^t I_M(\bar{s})$$

where $\hat{\gamma} \in (0, 1]$ is a discounting factor (that could be different from the discounting factor γ), and $I_M(\bar{s})$ is an importance metric value of the state $\bar{s}$ (e.g., $M \in \{A, AG\}$). The exponent t of the discounting factor was used to decrease the contribution of states that appear far in the future. In fact, at the end of the episodes, the states should be nearer to the end goal of the problem; thus, the Q-value differences should be greater. This concept is taken from the *exponential smoothing*, a rule-of-thumb technique for assigning larger weights to more recent observations (Gardner Jr, 1985). We continued this process until the end of the episode and iterated it for a fixed number of episodes. Finally, we normalized the values obtained by dividing the importance vector I by the number of episodes. In this way, we can generate a vector storing the importance relative to each state and depending on both the amount of time we visited a state and the importance of taking the correct action.

For the computational analysis reported in the next subsection, we focused on using the two metrics, advising and action gap importance, in the iterated form. In particular, we will call IQVA-A our approach considering the advising metric and IQVA-AG for the action gap case.

4.2. IQVA Results

To test this novel methodology, we consider three grid-world environments (Lava lake, Key-door and Taxi), as shown in Figure 1, to directly visualize the results. Then, we compared the computational results obtained by our approach, applied during and after the training of the RL agents, with the metrics presented. In this manner, we can characterize and differentiate important states in both situations.

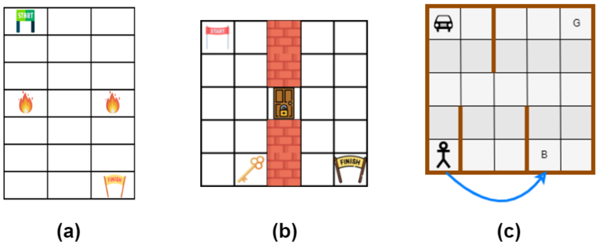

Figure 1. Environments used for the computational analysis of the proposed methodology: (a) Lava-lake, (b) Key-door, (c) Taxi.

In Lava Lake, a custom map derived from the well-known Frozen Lake, the agent, starting from the top-left corner, must reach the opposite corner of the map by traversing a lava lake. At any step, it receives 0 as a reward and 1 if it reaches its goal state. In this environment, we used SARSA for training the RL agent, and we considered as parameters for IQVA 10000 episodes and a discounting factor $\hat{\gamma} = 0.95$. The results are shown in Figure 2, where we used the suffixes *post* and *train* to describe whether the method was applied after or during the learning process. In particular, it is possible to notice how using IQVA-A, we are able to identify the position between the lava in the post-training phase, while for the training phase, the states that are near the lava in the upper part are more important. On the other hand, only during the learning procedure, it is possible to recognise the important state in the centre of the map using IQVA-AG. However, a significant difference is visible when compared to the usual metrics, where the focus is on the states next to the goal position.

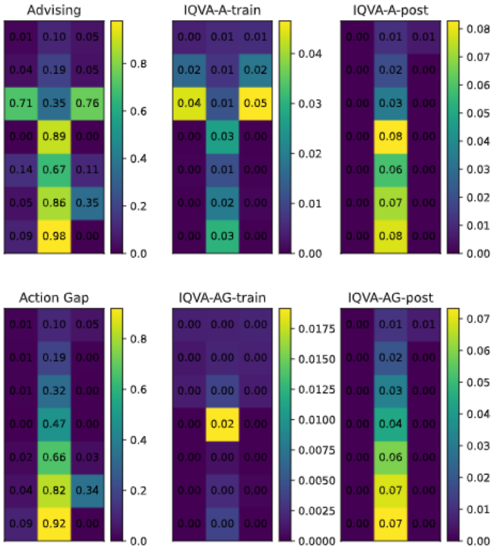

Figure 2. Importance Heat-maps for each state in the Lava lake from Figure 1a environment calculated using different metrics.

The second environment considered is the Key-door. In this case, the goal of the RL agent is to first pick up the key that is in one random position in the first room, and then be able to enter the second room and reach the end position. In this case, the reward is 0 for each time step, forcing the agent to complete the task to receive 1. In this scenario, we used a Q-learning algorithm for solving this problem and, for the evaluation of the states' importance, we chose 1000 episodes and $\hat{\gamma} = 0.95$. The results, divided into before and after picking up the key, are presented in Figure 3. For conciseness, we restricted the analysis of the "before pick up" scenario when the key is in the same position as in Figure 1b. However, the major differences between the metrics are in the second part ("after pick up"). In fact, we can notice that for both measurements (using the advising importance or action gap), the iterated approach is able to find the bottleneck caused by the door and consider it a crucial state. The focus was on the position next to the terminal state only in the case of IQVA-AG-train. This is principally related to the choice of a high value for the discounting factor $\hat{\gamma}$.

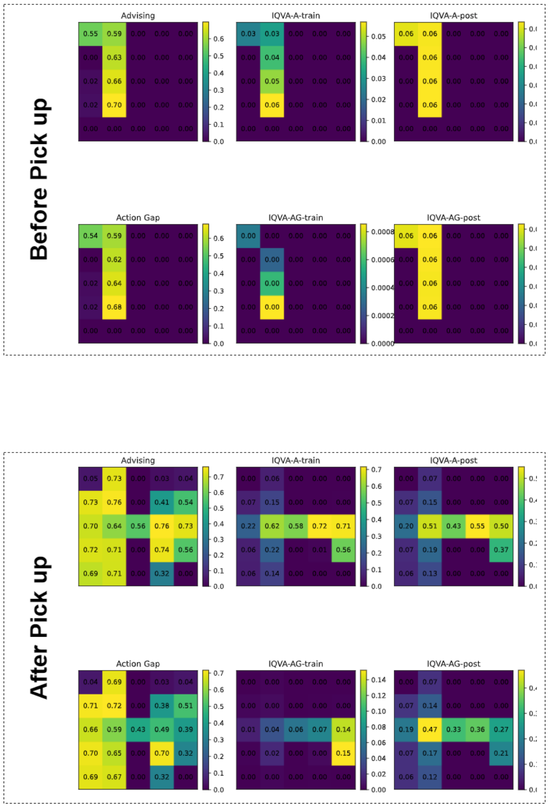

Figure 3. Importance Heat-maps for each state in the Key-door environment (before and after picking up the key) from Figure 1b calculated using different metrics.

The last environment is a simplified version of a real-world problem in which a taxi must pick up a passenger and drop it off at the correct destination. In the Taxi environment, there are four possible positions for the passenger and the destination that are marked by a letter on the map. Here, the reward is always -1 at each step, unless we try to pick up or drop the passenger in the wrong place, receiving in this way a reward of -10 or if we complete our task, earning 20. We trained a SARSA agent for this task using 5000 episodes and $\hat{\gamma} = 0.95$. As in the previous case, we divided the analysis before and after picking up the passenger to obtain a complete overview. In both situations, the iterated approach can identify the subset of states that are more interesting during training and after the learning process. In fact, the baseline metrics are giving high importance to states that are not often visited by the agent, and, therefore, characterizing these side states as central.

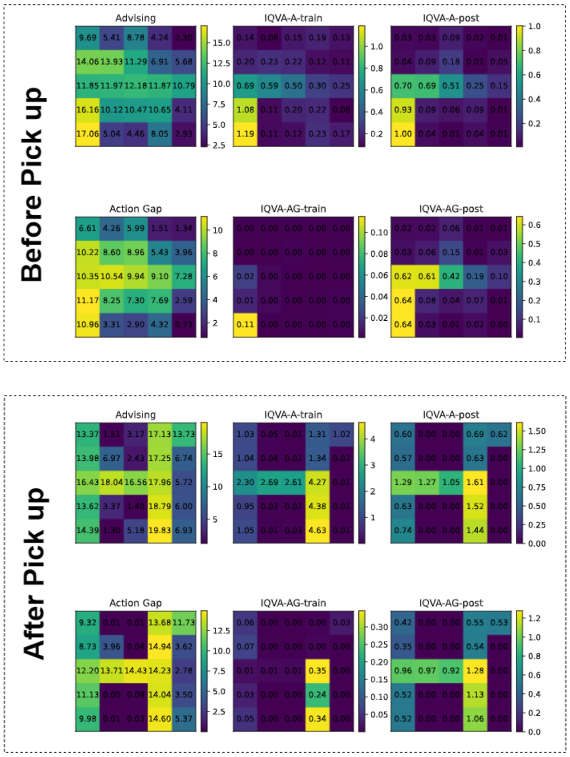

Figure 4. Importance Heat-maps for each state in the Taxi environment (before and after picking up the passenger) from Figure 1c calculated using different metrics.

4.3. Difference Metric

In the previous subsection, we focused on the analysis of metrics concerning the difference between the maximum Q-value and the second-best or minimum. In this way, it is possible to consider the disparity of taking the best action instead of an alternative one. However, they take into account only the best and worst-case scenarios, without going deep into the comparison of both of them. Therefore, we introduce a third metric that is strictly related to the advising and action gap measures considered before. In particular, we define the *difference* measure as follow:

$$\begin{aligned} I_D(s_t) &= \max_{a \in \mathbb{A}^*} Q(s_t, a) - \min_{a \in \mathbb{A}} Q(s_t, a) \\ &= \max_{a \in \mathbb{A}^*} Q(s_t, a) - \max_{a \in \mathbb{A}} Q(s_t, a) \\ &\qquad + \max_{a \in \mathbb{A}} Q(s_t, a) - \min_{a \in \mathbb{A}} Q(s_t, a) \\ &= -I_{AG}(s_t) + I_A(s_t), \end{aligned}$$

where it can be written as the difference between the second-best and minimum values or the advising and action gap measures. Using the last formulation can help us to find directly the *difference* importance values I_D in the environments already examined. In the same way as before, the iterated version of this new metric, called IQVA-D, is derived for each of the two phases considered, i.e., during the training and post-training.

Comparing the results obtained in the Lava lake environment, shown in Figure 5, we can notice that the bottleneck in the middle of the map is already recognised as being one of the most salient states, together with the positions next to the lava in the upper part. The latter ones are recognised also during the training phase from the IQVA-D, while the central location has been addressed as important only for the IQVA-D-post. In this scenario, the static formulation performs better than the iterated version due to the high-risk states present in the environment.

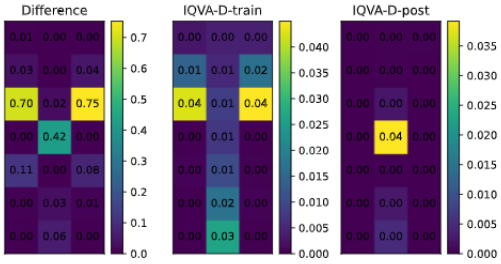

Figure 5. Importance Heat-maps for each state in the Lava lake from Figure 1a environment calculated using the difference metric.

However, the same can not be stated for the other cases. In the Key door environment, after picking up the key, we obtain that the bottleneck of the door is recognized by the IQVA-D-train but not from the simple difference metric, as represented in Figure 6. For the post-training version, we observe that the central path is highlighted from the other states, nonetheless, the focus is still on the last states before the goal. This could be caused by a large scaling factor for the iterated algorithm.

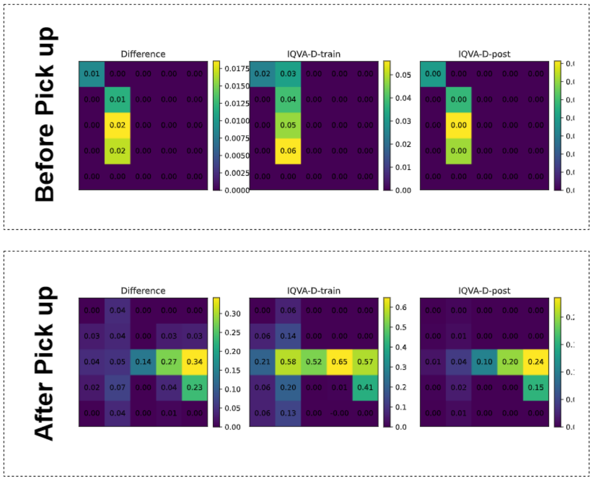

Figure 6. Importance Heat-maps for each state in the Key-door environment (before and after picking up the key) from Figure 1b calculated using the difference metric.

Finally, the major distinctions between the static approach and the iterative for the difference measure are obtained in the Taxi environment, shown in Figure 7. In fact, both the states preceding picking up the passenger and dropping off are relevant for the IQVA-D while, for the pure metric, the results are still not convincing, since the focus is spread all across the map. Moreover, the bottlenecks are also considered as important due to the high visitation rate discovered by our proposed approach.

5. Discussion

In the previous section, we noticed how IQVA can precisely detect the important states, not giving relevance to less visited states. Thus, we can focus on the real subset of states that must be analyzed and considered carefully. However, all approaches related to Q-values present a downfall. In fact, different agents could have different approximations for the action-state value function, resulting in a diverse representation of the important states. Nonetheless, to this extent, we are able to strictly characterize the choices of that particular agent in order to generate a specific explanation for the RL algorithm that we trained and that we will exploit.

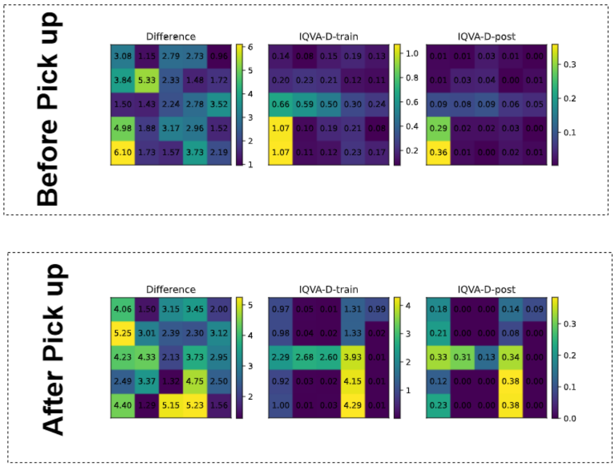

Figure 7. Importance Heat-maps for each state in the Taxi environment (before and after picking up the passenger) from Figure 1c calculated using different metrics.

Moreover, we considered two tabular value-based approaches, i.e., an off-policy (Q-learning) and an on-policy (SARSA), obtaining in both cases satisfying results in terms of recognition of the important states and restriction of the subset of relevant information. Therefore, our approach has been tested to be independent of the type of techniques adopted for finding the optimal policy. In addition, this methodology can be applied directly to more complex algorithms, e.g., deep learning methods like Deep Q-Network.

Another remark concerns the representation of these important states. For our experiments, we considered only the grid world environments, where each state corresponds to a particular position on a map. However, we can extend this idea by considering the network associated with the states visited during the training. In this way, we can rely on the measurements of the importance that we found with the previously presented methods in order to visualize it. Moreover, we can notice that, in the reported experiments, a chain of important states is found. This can be strictly related to the concept of distal action (Madumal et al., 2020), and, more specifically, of distal information (Milani, Moll, De Leone, et al., 2023), where, in order to achieve a particular state or perform a particular action, we first need to enable it by generating a sequence of actions or visiting determined circumstances. With our approach, this information is quickly detected owing to the iterated procedure adopted.

Furthermore, considering the advising and action gap importance in both the original and iterated versions can lead to the classification of important states. In fact, we can characterize the *physically important states* as those that present high values in the measurements obtained by IQVA and lower significance from the static ones. Moreover, we can rank the risk in each of these states by considering the difference between the advising and action gap metrics. Thus, we can measure the difference between the second-best Q-value and minimum. Therefore, great values of the difference will be related to *high-risk* situations, where choosing an incorrect action can lead to huge troubles, and *low-risk* states, where the choice made is not so relevant. Consequently, using advising and action gap metrics can lead to a general classification of important states, relying on a third possible importance measure that can be directly obtained by subtracting the previous values.

However, having a wide set of different metrics can help human users to characterise better the problem studied. Nonetheless, it is fundamental to recognise the advantages or disadvantages of using a measure instead of a different one. This is a well-known problem in the network analytics field, where there is a multitude of measures for concepts like centrality (Vignery and Laurier, 2021).

6. Conclusion and Future works

As more problems become solvable by RL agents, it is important to identify crucial states that can help in the generation of complete and trustworthy explanations. In this way, the human-artificial intelligent agent interactions can be improved substantially, drastically reducing the risk of taking a wrong decision. This paper introduces an iterative algorithm called IQVA for the detection of important states by relying on well-known metrics obtained from the action-state value function. We present an approach that can identify the information, that is relevant to a particular trained agent by considering the number of visits in a specific situation. Moreover, we introduce a new measure derived from the ones present in the literature and we apply to it our methodology. Our computational results on three grid-world environments show that significant benefits in terms of conciseness and focus on critical states can be achieved for all three measures considered.

There are several potential directions for future research. For example, being able to automatically tune the parameters relative to the number of episodes to operate and the discounting factor is a possible starting point. In fact, both of these values have a great influence on the results, and changing them can lead to a different characterization of the important states. Moreover, it may be useful to consider these measurements during the training of RL agents. Thus, it is possible to limit exploration on occasions where the risk importance values are high. Therefore, the agent can focus on deviating from safer trajectories instead of going into critical scenarios.

Acknowledgments

Rudy Milani is funded by dtec.bw—Digitalization and Technology Research Center of the Bundeswehr project RISK.twin. dtec.bw is funded by the European Union—NextGenerationEU.

References

Amir, D., & Amir, O. (2018). Highlights: Summarizing agent behavior to people. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 1168–1176.

Amir, O., Kamar, E., Kolobov, A., & Grosz, B. (2016). Interactive teaching strategies for agent training. *In Proceedings of IJCAI 2016*.

Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P., & Munos, R. (2016). Increasing the action gap: New operators for reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Clouse, J. A. (1996). *On integrating apprentice learning and reinforcement learning*. University of Massachusetts Amherst.

Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of forecasting*, *4*(1), 1–28.

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

Isele, D., Rahimi, R., Cosgun, A., Subramanian, K., & Fujimura, K. (2018). Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. *2018 IEEE international conference on robotics and automation (ICRA)*, 2034–2039.

Jacq, A., Ferret, J., Pietquin, O., & Geist, M. (2022). Lazy-mdps: Towards interpretable reinforcement learning by learning when to act. *arXiv preprint arXiv:2203.08542*.

Madumal, P., Miller, T., Sonenberg, L., & Vetere, F. (2020). Distal explanations for model-free explainable reinforcement learning. *arXiv preprint arXiv:2001.10284*.

Milani, R., Moll, M., De Leone, R., & Pickl, S. (2023). A bayesian network approach to explainable reinforcement learning with distal information. *Sensors*, *23*(4), 2013.

Milani, R., Moll, M., & Pickl, S. (2023). Advances in explainable reinforcement learning: An intelligent transportation systems perspective.

In *Explainable ai for intelligent transportation systems*. CRC Press.

Sequeira, P., & Gervasio, M. (2020). Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. *Artificial Intelligence*, *288*, 103367.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Simkute, A., Luger, E., Jones, B., Evans, M., & Jones, R. (2021). Explainability for experts: A design framework for making algorithms supporting expert decisions more explainable. *Journal of Responsible Technology*, *7*, 100017.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

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

Vignery, K., & Laurier, W. (2021). A methodology and theoretical taxonomy for centrality measures: What are the best centrality indicators for student networks? *PLOS ONE*, *15*(12), 1–32. https://doi.org/10.1371/journal.pone.0244377

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279–292.

Yan, Y., Chow, A. H., Ho, C. P., Kuo, Y.-H., Wu, Q., & Ying, C. (2022). Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. *Transportation Research Part E: Logistics and Transportation Review*, *162*, 102712.