STATE DECOMPOSITION FOR MODEL-FREE PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents an easy and fast algorithm to measure the gap between states and observations in a model-free partially observable Markov Decision Process (POMDP) with a stationary environment and a few state-missing conditions. We invented a proposition that states and observations can be decomposed into dimension sets, and dimension melting experiments are designed based on Cliff Walking. According to the experimental results, the algorithm can measure the general gap between states and observations.

1 INTRODUCTION

1.1 BACKGROUND

Markov decision processes exist everywhere in the real world, but the current research trend is to model the natural world through simulators. In other words, people try to abstract real world Markov decision processes into simulators and then build reinforcement learning to train agents. Simulators cannot fully model the reality because we cannot represent every detail of the natural environment. Consequently, all simulators that model the reality naturally become partially observable Markov decision processes (POMDP). Naturally, this problem becomes one of the challenges in landing reinforcement learning in the industrial sessions. Building more accurate reinforcement learning models is impossible unless if we can evaluate the simulator states validity in POMDP. When our reinforcement learning algorithms are running with low accuracy, it may not be a problem with the algorithm, but the gap between the simulator model and the real-world model is too large, but we do not know that the gap exists. The study is motivated by the fact that if the observation and the state differs significantly, it brings the bias to the relevant calculations, such as the calculation of Q-values. We think that when a reinforcement learning model does not work well in practice, it may not be a problem with the model but because the state of the simulator for training reinforcement learning differs significantly from the state of the natural environment.

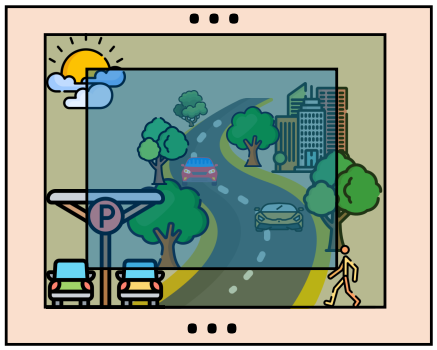
Most researches focus on how to solve the POMDP problems. (Chadès et al., 2021) introduces three methods to solve the POMDP problem: belief MDP (Åström, 1965), solution representation (Smallwood & Sondik, 1973) (Cassandra), and interpretation and visualisation (Chadès et al., 2008) (Chadès et al., 2011). They are both model-based POMDP solutions. Model-based means that the algorithm needs to know all the system states before solving the POMDP problem. We are the first to propose and define the theorem that the gap between states and observations is measurable in the model-free POMDP (it is not necessary to know all the states beforehand).

1.2 MAIN IDEAS

All the ideas in this paper are based on the following two conjectures:

Conjecture 1 *Set $s_{\mathbb{D}}$ is infinite, where s is a state from the real world, $s_{\mathbb{D}}$ is the dimensions of a state.*

We all know that the objects and things in the real world consist of infinite dimensions, and every element in the world may affect a thing that will happened. Thus, the dimensions of objects and things are uncountable.



Real-world

Simulator

Dimension / Dimension Space	Tree	Pedestrian	Yellow Car (Road)	Yellow Car (Park)	Red Car	Green Car	...
State-dimension	Yes	Yes	Yes	Yes	Yes	Yes	...
Observation-dimension	Yes	No	Yes	No	Yes	No	

Figure 1: The comparison state-dimension and observation-dimension. The left picture shows state dimensions in the real world, the blue area represents the observed by simulator; the green area represents the valid region for POMDP; the red area represents the states are infinite in the real world. The right picture shows the observed region by simulators. The table shows the state dimension set and the observation dimension set.

Conjecture 2 *Set $o_{\mathbb{D}} \subset s_{\mathbb{D}}$, where o is a observation from simulator, $o_{\mathbb{D}}$ is the dimensions of a observation.*

According to Conj. 1, in the real world, $s_{\mathbb{D}}$ are uncountable and partially observable. Thus, humans cannot build a full real world model in a simulator or an artificial environment. $\forall o_{\mathbb{D}}$ from simulators are a subset of $\forall \hat{s}_{\mathbb{D}}$ in the real world.

The difference in dimensions determines the difference between states and observations. As seen in Fig. 1, the natural environment has all the environmental information, but the agent can only observe part of it. Because the role of the state is used to distinguish the environmental information at different time, the lack of dimensions of the observed values leads to the inability to obtain the environmental state accurately. Although in many applications, the representation of the environment state is not composed of dimensions. For example, Super Mario Brothers can return a matrix of image pixels as states. However, it does not prevent us from using the environmental state dimension as a criterion to distinguish environmental information.

The dimensions have an order of the importance. Obviously, in Fig 1, it is crucial to obtain information about the turtle's location because this information can help Mario avoid death due to touching the turtle. However, the information about the cloud's location is unimportant because the cloud will not determine Mario's survival rate.

2 RELATED WORKS

2.1 POMDP (*)

Definition 1 *Formally, a POMDP is a 7-tuple $\mathcal{G} = \langle \mathbb{S}, \mathbb{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathbb{O}, \gamma \rangle$, where*

- $\mathbb{S}$*: a finite set of states*
- $\mathbb{A}$*: a set of actions*
- $\mathcal{P} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to [0, 1]$ *is a set of conditional transition probabilities between states*
- $\mathcal{R} : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ *is the reward function*
- Ω*: a set of observations*

- $\mathbb{O}$*: a set of conditional observation probabilities*
- $\gamma \in [0, 1)$*: discount factor*

Extended Parameters:

- $\mathbb{T}$: a set of action-observation history

The process of MDP is: at a timestep, the environment is in a state $s \in \mathbb{S}$. Then the agent takes an action $a \in \mathbb{A}$, and leads the environment to transition from s to s' with probability $\mathcal{P}(s' \mid s, a)$. After that, the agent observed an observation o from s' with probability $\mathbb{O}(o \mid s', a)$. At the same time, the agent gets the reward r by the reward function $\mathcal{R}(s, a)$. This process will repeat continuously. During this process, all interactions $\langle o^t, a, r_t \rangle$ are recorded in $\mathbb{T}$. The ultimate goal is to maximize the expectation of the discounted rewards: $E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where r_t is the reward at time t and γ is the discount factor.

2.2 Value Decomposition

The concept of decomposition was first used in the value decomposition, where (Sunehag et al., 2017) decomposed the joint value into:

$$Q_{tot}(\tau, \mathbf{u}) = \sum_{i=1}^{n} Q_i\left(\tau^i, u^i; \theta^i\right).$$

(Rashid et al., 2018) then introduced the QMIX for the value decomposition:

$$\underset{\mathbf{u}}{\operatorname{argmax}} Q_{tot}(\tau, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1\left(\tau^1, u^1\right) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n\left(\tau^n, u^n\right) \end{pmatrix}.$$

In (Son et al., 2019), the two equations were defined as IGM (Individual-Global-Max), and a new factorization equation was given based on IGM. (Rashid et al., 2020) and (Wang et al., 2020) also brought in their own optimization for the above theory.

3 Observation Decomposition

3.1 Information Theory(*)

In general, Shannon Entropy(Shannon, 1948) of a probability distribution P is defined as:

$$\text{Entropie}(P) = -\sum_{i=1}^{n} p_i \times \log\left(p_i\right). \tag{1}$$

Where $P = (p_1, p_2, \ldots, p_n)$ and $\sum_{i=1}^{N} p_i = 1$ (Bromiley et al., 2004). The higher the entropy, the more information can be transmitted; the lower the entropy, the less information is implied.

3.2 Observation Decomposition Theory

Fig. 2 describes the process of an $\langle o, a, r \rangle$. When an observation generates multiple state possibilities, the observation and the state are in a one-to-many relationship. The role of states is to record changes in the environment, and a decrease in the dimension of states means that states will not work correctly. Since the set of observation dimensions is a subset of the dimensions of the state, the POMDP can be defined as a formula (see Def. 2).

A state or observation is made up of multiple dimensions and is regarded as a collection of dimensions. Dimensions can be understood as objects or elements in a scene or environment. However,

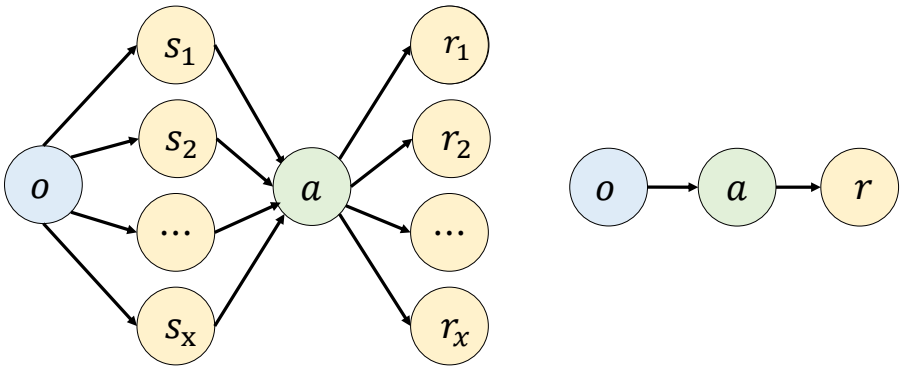


Figure 2: The relationship between states and observations in stationary environments. The left picture shows $\langle o, a \rangle$ in POMDP; the right picture shows $\langle o, a \rangle$ in MDP.

there are scenes (Brockman et al., 2016) where pixels are used as states or observations, and pixels are determined not to meet the definition of dimensions. One characteristic of the dimensions is that their properties do not change over time. For examples, pixel location is not regarded as a dimension as they changes over time.

Definition 2 *Specifically, a POMDP is a 7-tuple* $\mathcal{G} = \langle \mathbb{S}, \mathbb{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathbb{O}, \gamma \rangle$*, where*

- $_{\mathbb{D}}$*: a state dimension set, where* $\hat{s}_{\mathbb{D}} = \{\omega_n d_n \mid \omega_n \geq \omega_{n+1}, n \to \infty\}$, $s \subset \mathbb{S}$
- $\hat{s}_{\mathbb{D}}$*: an estimate state dimension set, where* $\hat{s}_{\mathbb{D}} = \{\omega_m d_m \mid \omega_m \geq \omega_{m+1}, m \geq n\}$, $s \subset \mathbb{S}$
- $o_{\mathbb{D}}$*: a observation dimension set, where* $\hat{s}_{\mathbb{D}} = \{\omega_n d_n \mid \omega_n \geq \omega_{n+1}, n > 0\}$, $s \subset \mathbb{S}$

Extended Parameters:

- d*: a dimension, where* $d \subset \mathbb{D}$
- ω*: a weight, where* $\omega \in \mathbb{R}$
- $\mathbb{D}$*: a set of dimensions*

According to Conj. 1, the state dimension space $s_{\mathbb{D}}$ is infinite, it is impossible to realized. Thus, we propose $\hat{s}_{\mathbb{D}}$ as the estimate of $s_{\mathbb{D}}$ when $_{\mathbb{D}} - \hat{s}_{\mathbb{D}} < \epsilon$, where $\epsilon \in \mathbb{R}$. ω is the importance of dimension, there is exist a zero Boundary ω_z that $\{\omega_v = 0 \mid v > z\}$ for a specific task.

Gap The gap between states and observations is defined as $\Delta = \hat{s}_{\mathbb{D}} - o_{\mathbb{D}}$. Δ determines how much probabilities $o = s$ in POMDP:

Proposition 1 *Set* $\Delta = \hat{s}_{\mathbb{D}} - o_{\mathbb{D}}$*, then* $\forall \Delta$, $p(s \mid o) \propto 1/\Delta$.

According to Fig. 2, the state s and observation o has one-to-many relationship, we define the mapping as below:

Definition 3 *If* $\hat{s}_{\mathbb{D}} \cap o_{\mathbb{D}} = o_{\mathbb{D}}$*, then* $f_{s2o} : s \to o$.

Def. 3 measures the error of partial observation, and states are disguised as an observation. This defines the scientific illustration of the bias in POMDP with the representation dimension sets.

3.3 WEIGHTING

The weighting style determines the representation of dimension sets. We propose two weighting types for measuring the gap, which are the real gap and the gap in the simulator between states and

observations. The real gap is the gap defined by Def. 2. The gap in the simulator is the dimension of the gap has different importance in different simulators. Thus we should quantify the gap in every simulator.

Boolean Weighting This weighting style, which we call Boolean Weighting, is a general representation:

$$w = \begin{cases} 1 & d \in \mathbb{D} \\ 0 & d \notin \mathbb{D} \end{cases} \tag{2}$$

To ensure the general gap, we simply set the dimension set to be the index set (equation 2), which means choosing which dimension exists in the dimension set. Because in a stationary environment, the reward r in $\langle s, a, r\rangle$ must be the same as the one in the history $\mathbb{T}$, but the reward in $\langle o, a, r\rangle$ may be changed (as Fig. 2). Thus, we propose λ for Boolean Weighting. λ is the metric for measuring the gap:

$$\lambda = \sum_{j=1}^{J} \sum_{i=1}^{I} \theta_{ji} \tag{3}$$

where θ is a reward class that always equals one here, $J \in \mathbb{Z}^+$ is the number of tuple $\langle o, a, r\rangle$ in the history, and $I \in \mathbb{Z}^+$ is the number of different reward classes in a tuple $\langle o, a, r\rangle$.

Theorem 3 *Let ω be the Boolean Weighting from equation 4, λ be a metric from equation 3. Set $\Delta = \hat{s}_{\mathbb{D}} - o_{\mathbb{D}}$, then $\exists \alpha \geq 0$ such that $\lambda = \alpha \mid \Delta \mid$.*

Theorem 3 provides a guarantee that λ and Δ are positively correlated. λ can measure the gap when ω is the Boolean Weighting.

Continuous Weighting This weighting style, which we call Continuous Weighting, is a specific representation:

$$w = \begin{cases} \beta & d \in \mathbb{D} \\ 0 & d \notin \mathbb{D} \end{cases} \tag{4}$$

Because Continuous Weighting is designed for the specific simulator, different simulators and tasks bring in the gap variance. Thus, we use $\beta \geq 0$ to represent the importance of the dimension differently in the specific simulator.

$$\lambda = -\sum_{j=1}^{J} \sum_{i=1}^{I} p_{ji} \times \log_2 (p_{ji}) \tag{5}$$

where p is the the ratio of a reward class in the same tuple $\langle o, a, r\rangle$, $J \in \mathbb{Z}^+$ is the number of tuple $\langle o, a, r\rangle$ in the history $\mathbb{T}$, and $I \in \mathbb{R}$ is the ratio of different reward classes in a tuple $\langle o, a, r\rangle$. Then, we propose λ for Continuous Weighting.

Theorem 4 *Let ω be the Continuous Weighting from equation 4, λ to be a metric from equation 5. Set $\Delta = \hat{s}_{\mathbb{D}} - o_{\mathbb{D}}$, then $\exists \alpha \geq 0$ such that $\lambda = \alpha \mid \Delta \mid$.*

Theorem 4 shows λ and Δ have positive correlations, and λ can measure the gap under Continuous Weighting.

Proposition 2 *When $\lambda > 0$, it will lead to a biased Q-value.*

Algorithm 1 λ-list Generate Function

Input: $\mathbb{T}, o_{select}, a_{select}$
Output: a λ-table
Process: Function λ-ListGenerate
Initialisation : Set a empty list λ-list
begin
set H is the length of $\mathbb{T}$
for $h := 1$ to $\dots H$ **do**
  if $o_h == o_{select}$ and $a_h == a_{select}$ **then**
    insert r_h into λ-list
  end if
end for

Algorithm 2 λ Calculation Algorithm

Input: $\mathbb{T}$
Output: λ
Process: λ-Function
begin
set $U \leftarrow$ empty list.
set H is the length of $\mathbb{T}$
for $h := 1$ to $\dots H$ **do**
  if $\{o_h, a_h\}$ not in U **then**
    U append $\{o_h, a_h\}$
  end if
end for
set K is the length of U
set $\lambda = 0$
for $k := 1$ to $\dots K$ **do**
  Set $\lambda_{sub} = 0$
  λ-list $\longleftarrow$ λ-ListGenerate$\langle o_k, a_k \rangle$
  calculate λ_{sub} using λ-list (by equation 3 λ for Boolean Weighting, by equation 5 (λ for Continuous Weighting)
  $\lambda \longleftarrow \lambda + \lambda_{sub}$
end for

3.4 ALGORITHM

Algorithm 1 introduces the process of generating the λ-list. The input is the selected tuple $\langle o_{select}, a_{select} \rangle$ which be extracted from the history $\mathbb{T}$, and the output is the λ-list. During the process, the algorithm searches through all histories for records which are to the tuple $\langle o_{select}, a_{select} \rangle$. When a match is fetched, the reward of the record is inserted into the λ-list.

Algorithm 2 describes the process of computing the λ. The de-duplicated tuple $\langle o, o', a \rangle$ is firstly extracted from the history, after which the de-duplicated elements are looped into Alg. 1 to calculate λ-list. Finally, the entropy corresponding to each de-duplicated tuple is calculated by Equ. 1, and these entropies are summed to output λ.

4 EXPERIMENT

4.1 PREPARATION

We chose Cliff Walking Environment from Gym OpenAI (Brockman et al., 2016) as our experiment. This is the grid-world example mainly for comparing the Sarsa (Sutton et al., 1998) (on-policy) and Q-learning (Watkins & Dayan, 1992) (off-policy) methods.

Fig. 3 shows the environment map composed of 4×12 grids (Sutton & Barto, 1999), where the entire fourth row of columns are cliffs, $[3, 0]$ is the starting point, and $[3, 11]$ is the goal point. Action

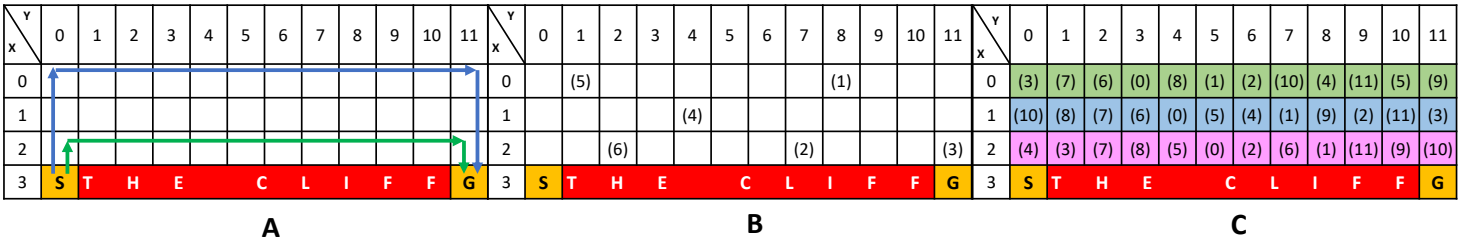


Figure 3: The cliff walking environment and three examples of experimental designs for the dimension melting. **A** panel shows the base environment of cliff walking and the optimal choice of route for Q Learning and Sarsa; **B** panel shows global random melting; **C** panel figure shows row traversal melting with two-part.

space $\mathbb{S}$ consists of four discrete actions, which are up, down, left and right. If the agent steps onto the cliff, it will incur a -100 reward; immediately terminate the episode and send the agent back to the starting point. Each step will incur a -1 reward. The goal of cliff-walking is to obtain the smallest cumulative reward during the episode.

Because Cliff Walking is an artificial environment, we regard that the states of the environment are fully observable and finite. Hence, the total number of states is $3 \times 12 + 1$ (excluding them from the state because $[3, 1 \ldots 10]$ are cliffs), and we decompose the state and full observation into $\hat{s}_{\mathbb{D}} = \{\omega_1 d_1, \omega_2 d_2, \ldots, \omega_{37} d_{37}\}$ and $o_{\mathbb{D}} = \{\omega_1 d_1, \omega_2 d_2, \ldots, \omega_{37} d_{37}\}$.

Stationary Environment

Both Q-Learning and Sarsa are $\varepsilon-$greedy algorithms, and both follow the Bellman equation. But the action used by Sarsa to update the Q value and the action chosen in the next step must be the same:

$$\mathrm{Q}^{*}\left(\mathrm{s_t}, \mathrm{a_t}\right) = \mathrm{Q}\left(\mathrm{s_t}, \mathrm{a_t}\right) + \alpha\left[\mathrm{r_t} + \gamma \mathrm{Q}\left(\mathrm{s_{t+1}}, \mathrm{a_{t+1}}\right) - \mathrm{Q}\left(\mathrm{s_t}, \mathrm{a_t}\right)\right] \tag{6}$$

On the other hand, Q-Learning selects the next action using the $\varepsilon-$greedy algorithm and updates the Q value using the action with the maximum Q value:

$$\mathrm{Q}^{*}\left(\mathrm{s_t}, \mathrm{a_t}\right) = \mathrm{Q}\left(\mathrm{s_t}, \mathrm{a_t}\right) + \alpha\left[\mathrm{r_t} + \gamma \max_{\mathrm{a}'}\left(\mathrm{Q}\left(\mathrm{s_{t+1}}, \mathrm{a}'\right)\right) - \mathrm{Q}\left(\mathrm{s_t}, \mathrm{a_t}\right)\right] \tag{7}$$

4.2 Dimension Melting

In order to better explore the difference between states and observations, we design a new experimental approach called **Dimension Melting**. The meaning of Dimension Melting is to simulate the result of λ when systematically eliminating the dimension of states. The gap between state and observation is measured by continuously melting dimensions.

For the cliff-walking task, we design two melting schemes for observations. In Fig. 3, the **B** panel is to pick 16 grids in the entire grid map and randomly melt their corresponding dimensions. The **C** panel is the dimensions of each row are melted gradually and independently.

The experiment is set up with 2000 episodes for every observation melting, and the maximum number of steps per episode is 50. At each observation melting, we record (s, a, r) and later use Alg. 2 to calculate the λ and the sum of rewards. Each melting experiment is run for ten times, and the results of the ten runs are averaged.

We design a state replacement mechanism according to Fig. 2. One state has and only corresponds to one observation, so we simulate this feature with the observation substitution mechanism. When a dimension corresponds to a state that is melted, this state is replaced by the upper, lower, left, or right states. If neither the upper nor the lower left state is available, one of the available states will be randomly selected for replacement.

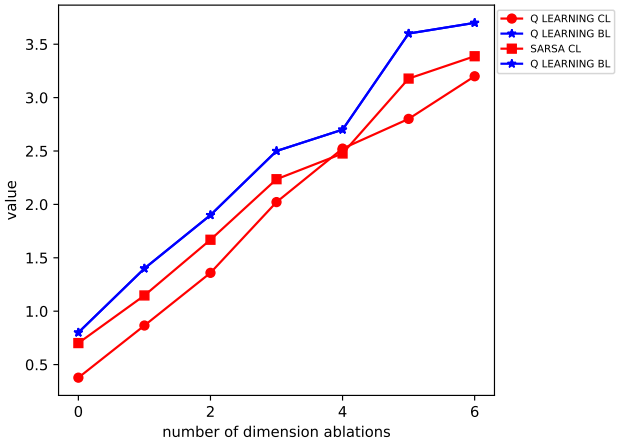


Figure 4: The result of **B** panel of Fig. 3. BL is the λ for Boolean Weighting, CL is the λ for Continuous Weighting.

5 RESULT

Fig. 4, the overall trend of λ based on Q Learning and Sarsa goes up as the dimension melts more. all the λ for Boolean Weighting are almost overlap, that proof Boolean Weighting is for the real state instead of the specific state.

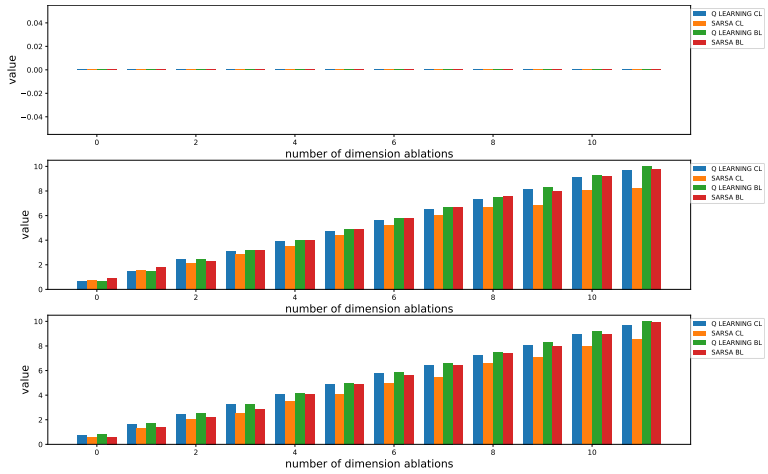


Figure 5: The result of **C** panel of Fig. 3. BL is the λ for Boolean Weighting, CL is the λ for Continuous Weighting.

The first graph of Fig. 5 comes from **A** panel of Fig. 3. All λ is equal to zero because the entire row 1 will be replaced by row 2, so both of their rewards are -1. Theorem 3 and 4 do not apply to the situation that the distribution of rewards is too concentrated because I in equation 3 and equation 5 represents different reward classes should have different value. Sarsa does not detect this change because it chooses a shorter path than the original.

In the second and third graphs of Fig. 5, all λ steadily increases with the dimension melting. The reason is that rows 1 and 2 are the critical paths for Sarsa, so λ for Sarsa goes up. For Q Learning, no doubt, row 2 is the optimal route. However, λ of Q Learning increases because according to **state replacement principle** in Sec. 4.2, when states in row 1 are deleted, they will be replaced with their lower states which are located in row 1.

Overall, λ performs consistently and can determine the gap between states and observations as the dimension melts. However, the weakness exists in the high algorithm dependence (all $\langle o, a, r \rangle$ need to be explored by algorithms) and the random state replacement principle, they may lead to the malfunction of λ.

CONCLUSION

The first time we propose for the first time that states and observations can be decomposed into sets of dimensions and use λ to measure the gap between them. Experiments prove that λ can be effective in general. The future work is to solve the high algorithm depend, non-stationary environment, and the distribution of rewards is too concentrated problems.

REPRODUCIBILITY STATEMENT

Although the theorem and algorithms presented in this paper are originally designed for gap assessment between natural and artificial environments, they can also experiment in any scenario with Markov properties. However, the volume of data required varies depending on the environment. Only $\langle o, a, r \rangle$ needs to be stored to calculate both Boolean Weighting and Continuous Weighting styles of λ.

REFERENCES

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10(1):174–205, 1965.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

PA Bromiley, NA Thacker, and E Bouhova-Thacker. Shannon entropy, renyi entropy, and information. *Statistics and Inf. Series (2004-004)*, 9:10–42, 2004.

A.R. Cassandra. The pomdp page. https://www. pomdp.org/.

Iadine Chadès, Luz V Pascal, Sam Nicol, Cameron S Fletcher, and Jonathan Ferrer-Mestres. A primer on partially observable markov decision processes (pomdps). *Methods in Ecology and Evolution*, 12(11):2058–2072, 2021.

Iadine Chadès, Eve McDonald-Madden, Michael A. McCarthy, Brendan Wintle, Matthew Linkie, and Hugh P. Possingham. When to stop managing or surveying cryptic threatened species. *Proceedings of the National Academy of Sciences*, 105(37):13936–13940, 2008. doi: 10.1073/pnas.0805265105. URL https://www.pnas.org/doi/abs/10.1073/pnas.0805265105.

Iadine Chadès, Tara G. Martin, Samuel Nicol, Mark A. Burgman, Hugh P. Possingham, and Yvonne M. Buckley. General rules for managing and surveying networks of pests, diseases, and endangered species. *Proceedings of the National Academy of Sciences*, 108(20):8323–8328, 2011. doi: 10.1073/pnas.1016846108. URL https://www.pnas.org/doi/abs/10.1073/pnas.1016846108.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pp. 4295–4304. PMLR, 2018.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *Robotica*, 17(2): 229–235, 1999.

Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

A Appendix

A.1 Proof of Prop. 1

Set $\Delta = \hat{s}_{\mathbb{D}} - o_{\mathbb{D}}$. Define f is the function which mapping: $f : \hat{s}_{\mathbb{D}} \to s$ and $f : o_{\mathbb{D}} \to o$. Then we have:

$$\begin{aligned} p(\hat{s}_{\mathbb{D}} \mid o_{\mathbb{D}}) &= p(\hat{s}_{\mathbb{D}} \mid \hat{s}_{\mathbb{D}} - \Delta) \propto 1/\Delta \\ &\Rightarrow p(f(\hat{s}_{\mathbb{D}}) \mid f(o_{\mathbb{D}})) = p(f(\hat{s}_{\mathbb{D}}) \mid f(\hat{s}_{\mathbb{D}} - \Delta)) \propto 1/\Delta \\ &\Rightarrow p(s \mid o) \propto 1/\Delta \end{aligned} \tag{8}$$

A.2 Proof of Theorem 3 and Theorem 4

Assume $o_1, o_2 \in \Omega$, $s_1, s_2 \in \mathbb{S}$. Define

$$\hat{s}^b_{\mathbb{D}} := \left\{\omega^b_n \mid n \in \mathbb{R}, b \in \mathbb{R}\right\}$$

, where $n \to \infty$ (according to Def. 2) is the number of dimensions, b is the index of states. Moreover, $\hat{s}^1_{\mathbb{D}} = \hat{s}^2_{\mathbb{D}}$.

$$\mathrm{o}^b_{\mathbb{D}} := \left\{\omega^b_m \mid m \in \{m_1, m_2, ..., m_b\}, b \in \mathbb{R}\right\}$$

where m is the observation dimension set consist with finite numbers (according to Def. 2), b is the index of observations. Set $o^b_{\mathbb{D}} \subset \hat{s}^b_{\mathbb{D}}$.

Thus, we have the expression of dimension sets for s_1, s_2, o_1, o_2:

$$\hat{s}^1_{\mathbb{D}} := \left\{\omega^1_1 d^1_1, \omega^1_2 d^1_2, \omega^1_3 d^1_3, \cdots, \omega^1_n d^1_n\right\}$$

$$\hat{s}^2_{\mathbb{D}} := \left\{\omega^2_1 d^2_1, \omega^2_2 d^2_2, \omega^2_3 d^2_3, \cdots, \omega^2_n d^2_n\right\}$$

$$o^1_{\mathbb{D}} := \left\{\omega^1_1 d^1_1, \omega^1_2 d^1_2, \omega^1_3 d^1_3, \cdots, \omega^1_x d^1_x\right\}$$

$$o^2_{\mathbb{D}} := \left\{\omega^2_1 d^2_1, \omega^2_2 d^2_2, \omega^2_3 d^2_3, \cdots, \omega^2_y d^2_y\right\}$$

Define the gap between the observation and the state is Δ, formulas are:

$$\Delta_1 := \hat{s}^1_{\mathbb{D}} - o^1_{\mathbb{D}} = \left\{\omega^1_{m+1} d^1_{m+1}, \omega^1_{m+2} d^1_{m+2}, \cdots, \omega^1_n d^1_n\right\}$$

where $m \in \mathbb{R}$ and $m \neq 0$.

$$\Delta_2 := \hat{s}^2_{\mathbb{D}} - o^2_{\mathbb{D}} = \left\{\omega^2_{k+1} d^2_{k+1}, \omega^2_{k+2} d^2_{k+2}, \cdots, \omega^2_n d^2_n\right\}$$

where $k \in \mathbb{R}$ and $K \neq 0$.

After z **iterations**, we extract $\langle s_1, a, r\rangle$, $\langle s_2, a, r\rangle$, $\langle o_1, a, r\rangle$ and $\langle o_2, a, r\rangle$ from $\mathbb{T}$. Then we get:

$$\begin{bmatrix} s_1 \\ \vdots \\ s_u \end{bmatrix} \times a = \begin{bmatrix} r_1 \\ \vdots \\ r_u \end{bmatrix} = \mathbb{B}^1_s$$

$$\begin{bmatrix} s_1 \\ \vdots \\ s_u \end{bmatrix} \times a = \begin{bmatrix} r_1 \\ \vdots \\ r_u \end{bmatrix} = \mathbb{B}^2_s$$

$$\begin{bmatrix} o_1 \\ \vdots \\ o_v \end{bmatrix} \times a = \begin{bmatrix} r_1 \\ \vdots \\ r_v \end{bmatrix} = \mathbb{B}^1_o$$

$$\begin{bmatrix} o_1 \\ \vdots \\ o_w \end{bmatrix} \times a = \begin{bmatrix} r_1 \\ \vdots \\ r_w \end{bmatrix} = \mathbb{B}^2_o$$

According to Def. 3,

Thus, in equation 3 and equation 5, $J = z$. After that, we collect the unique reward separately into sets

$$\mathbb{W}^1_s := \{a, b \mid a \neq b, \forall a \in \mathbb{B}^1_s, \forall b \in \mathbb{B}^1_s\}$$
$$\mathbb{W}^2_s := \{a, b \mid a \neq b, \forall a \in \mathbb{B}^2_s, \forall b \in \mathbb{B}^2_s\}$$
$$\mathbb{W}^1_o := \{a, b \mid a \neq b, \forall a \in \mathbb{B}^1_o, \forall b \in \mathbb{B}^1_o\}$$
$$\mathbb{W}^2_o := \{a, b \mid a \neq b, \forall a \in \mathbb{B}^1_o, \forall b \in \mathbb{B}^2_o\}$$

Because the reward r is unchanged for (s,a,r) in stationary POMDP after z iterations. Thus, $\mid \mathbb{W}^1_s \mid = \mid \mathbb{W}^2_s \mid$.

For the proof of the Boolean Weighting, we set $\omega = 0$ or $\omega = 1$, and according to equation 3, we get:

Whereas if $\mid \Delta_1 \mid > \mid \Delta_2 \mid$, then according to Prop. 1

$$p(s_1 \mid o_1) < p(s_2 \mid o_2)$$

This means if the Δ

According to equation 3, we know Δ has no affect to mapping from $\hat{s}_{\mathbb{D}}$ to $o_{\mathbb{D}}$:

$$\begin{aligned} \hat{s}_{\mathbb{D}} \cap o_{\mathbb{D}} &\Rightarrow (o_{\mathbb{D}} \cup \Delta) \cap o_{\mathbb{D}} = o_{\mathbb{D}} \\ &\Rightarrow (o_{\mathbb{D}} \cap o_{\mathbb{D}}) \cup (\Delta \cap o_{\mathbb{D}}) \\ &\Rightarrow o_{\mathbb{D}} \cup \varnothing = o_{\mathbb{D}} \end{aligned} \tag{9}$$

Thus, $C^1_{\mu^{|\Delta|}}$, where μ is the classes of ω, is the maximum combination of Δ, and this is the error range. Because $C^1_{\mu^{|\Delta|}} = C^1_{\mu^{|s_{\mathbb{D}}-o_{\mathbb{D}}|}}$ represent potential s when o is fixed, and this is equal to classes of rewards because s is mapping a r in $\langle s, a, r\rangle$.

For Boolean Weighting:

Set $j = z$, $i \in \mathbb{W}$, $\mu = 0$ or $\mu = 1$, $c^1_\Delta(1) = C^1_{\mu|\Delta_1|}$, $c^2_\Delta(1) = C^1_{\mu|\Delta_2|}$, $x > 0$ and $x \in \mathbb{R}$.

Whereas if $\mid \Delta_1 \mid = \mid \Delta_2 \mid$, then $\Sigma^z_{i=1}\Sigma^{c^1_\Delta}_{i=1}\theta_{ji} = \Sigma^z_{i=1}\Sigma^{c^2_\Delta}_{i=1}\theta_{ji}$, so $\lambda^1_o = \lambda^2_o$.

Whereas if $\mid \Delta_1 \mid > \mid \Delta_2 \mid$, then $\Sigma^z_{i=1}\Sigma^{c^1_\Delta}_{i=1}\theta_{ji} > \Sigma^z_{i=1}\Sigma^{c^2_\Delta}_{i=1}\theta_{ji}$, so $\lambda^1_o > \lambda^2_o$.

Whereas if $\mid \Delta_1 \mid < \mid \Delta_2 \mid$, then $\Sigma^z_{i=1}\Sigma^{c^1_\Delta}_{i=1}\theta_{ji} < \Sigma^z_{i=1}\Sigma^{c^2_\Delta}_{i=1}\theta_{ji}$, so $\lambda^1_o < \lambda^2_o$.

For Continuous Weighting:

Set $j = z$, $i \in \mathbb{W}$, $\mu = \alpha \in \mathbb{R}$, $c^1_\Delta(1) = C^1_{\mu|\Delta_1|}$, $c^2_\Delta(1) = C^1_{\mu|\Delta_2|}$, $x > 0$ and $x \in \mathbb{R}$.

Whereas if $\mid \mathbb{W}^1_o \mid = \mid \mathbb{W}^2_o \mid$, then $\lambda^1_o = \lambda^2_o \Rightarrow \Sigma^{c^1_\Delta}_{i=1} p_{ji} \times log_2(p_{ji}) = \Sigma^{c^2_\Delta}_{i=1} p_{ji} \times log_2(p_{ji})$, so $\Delta_1 = \Delta_2$.

Whereas if $\mid \mathbb{W}^1_o \mid > \mid \mathbb{W}^2_o \mid$, then $\lambda^1_o > \lambda^2_o \Rightarrow \Sigma^{c^1_\Delta}_{i=1} p_{ji} \times log_2(p_{ji}) > \Sigma^{c^2_\Delta}_{i=1} p_{ji} \times log_2(p_{ji})$, so $\Delta_1 > \Delta_2$.

Whereas if $\mid \mathbb{W}^1_o \mid < \mid \mathbb{W}^2_o \mid$, then $\lambda^1_o < \lambda^2_o \Rightarrow \Sigma^{c^1_\Delta}_{i=1} p_{ji} \times log_2(p_{ji}) < \Sigma^{c^2_\Delta}_{i=1} p_{ji} \times log_2(p_{ji})$, so $\Delta_1 < \Delta_2$.

A.3 Proof of Prop. 2

Bellman Equation is:

$$V(s) = R(s) + \gamma \sum_{s' \in S} p\left(s' \mid s\right) V\left(s'\right) \tag{10}$$

The analytic solution of the Bellman Equation is:

$$\boldsymbol{V} = (\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} \boldsymbol{R} \tag{11}$$

Because according to Prop. 3 and 4, the larger λ, the larger Δ. Let $\Delta > 0$. Since here both states and observations are sets, $s = s' = \{\omega_1 d_1, \omega_2 d_2, \omega_3 d_3, \cdots, \omega_n d_n, \cdots\}$ and $o = o' = \{\omega_1 d_1, \omega_2 d_2, \omega_3 d_3, \cdots, \omega_n d_n\}$. Since $s = o \cup \Delta$ and $o \cap \Delta = \varnothing$. Therefore, the size of $\boldsymbol{V}$ has changed from infinite to finite. Because the association between the state value function and the Q function is:

$$Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} p\left(s' \mid s, a\right) V_\pi\left(s'\right) \tag{12}$$

Both parts $R(s, a)$ and $\gamma \sum_{s' \in S} p\left(s' \mid s, a\right) V_\pi\left(s'\right)$ change from infinite to finite, so $Q_\pi(s, a)$ also changes from infinite to finite. Therefore, lambda causes a compression of the Q-value space, which leads to a possible Q-value collection impact.

The reason is that the observation cannot serve to distinguish between environments, resulting in reward updates for different environmental states in the same state and directly leading to a biased Q-value.