State Representation Learning in Robotics: Using Prior Knowledge about Physical Interaction

Rico Jonschkowski and Oliver Brock
Robotics and Biology Laboratory, Technische Universität Berlin, Germany

CS330 Student Presentation

Background

State representation: a useful mapping from observations to features that can be acted upon by a policy

State representation learning (SRL) is typically done with the following learning objective categories:

- Compression of observations, i.e. dimensionality reduction[1]
- Temporal coherence[2,3,4]
- Predictive/predictable action transformations[5,6,7]
- Interleaving representation learning with reinforcement learning[8]
- Simultaneously learning the transition function[9]
- Simultaneously learning the transition and reward functions[10, 11]

Motivation & Problem

Many robotics problems solved using reinforcement learning until recently with using **task-specific** priors, i.e. *feature engineering*.

Need for state representation learning:

- Engineered features tend to not generalize across tasks, which limits the usefulness of our agents
- Want to get states that adhere to real-world/robotic priors
- Want to act using raw image observations

Robotic Priors

1. Simplicity: only a few world properties are relevant for a given task
2. Temporal coherence: task-relevant properties *change gradually* through time
3. Proportionality: change in task-relevant properties wrt action is proportional to magnitude of action
4. Causality: task-relevant properties with the action determine the reward
5. Repeatability: actions in similar situations have similar consequences

- Priors are defined using reasonable limitations applying to the physical world

Methods

Robotic Representation Setting: RL

Jonschkowski and Brock (2014)

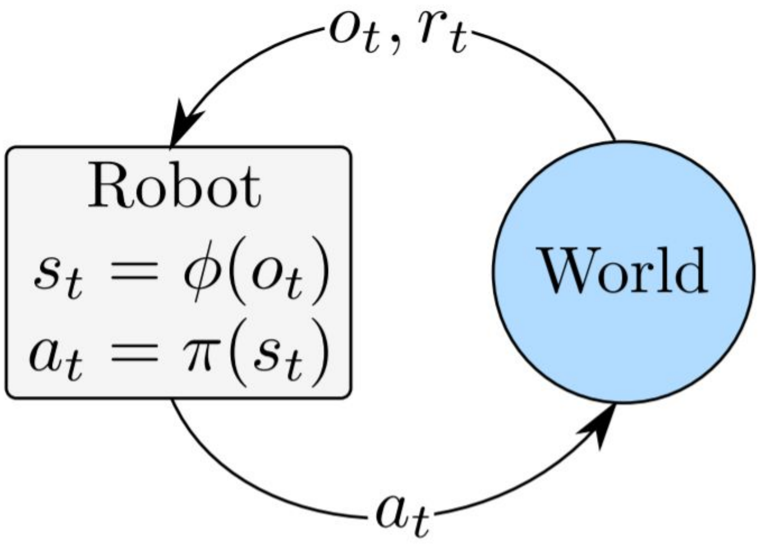
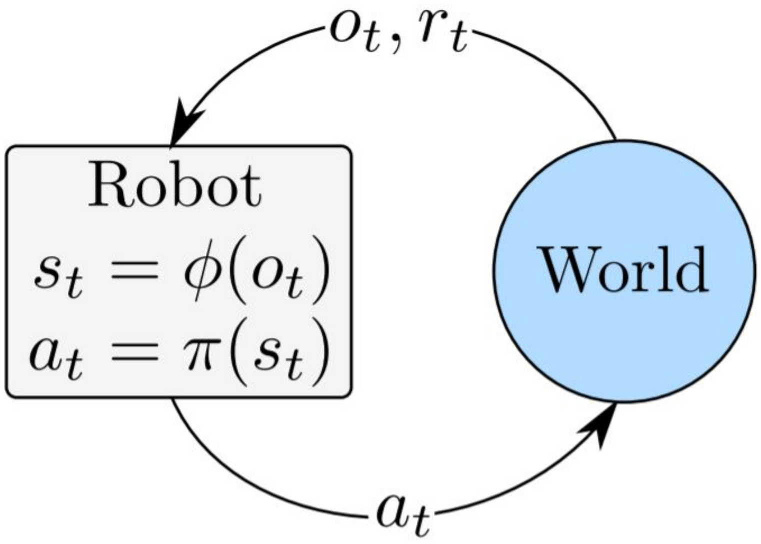


Fig. 2. The robot-world-interaction. At time t, the robot computes the state s_t from its observation o_t using observation-state-mapping ϕ. It chooses action a_t according to policy π with the goal to maximize future rewards $r_{t+1:\infty}$.

Robotic Representation Setting: RL

- State representation: $s_t = \phi(o_t)$
  - Linear state mapping
  - Learned intrinsically from robotic priors
  - Full observability assumed

- Policy: $\pi(s_t) = a_t$
  - Learned on top of representation s_t
  - Two FC layers with sigmoidal activations
  - RL method: Neural-fitted Q-iteration (Riedmiller, 2005)



Jonschkowski and Brock (2010)

Robotic Priors

$$L(D, \hat{\phi}) = L_{\text{temporal coherence}}(D, \hat{\phi}) + L_{\text{proportionality}}(D, \hat{\phi})$$
$$+ L_{\text{causality}}(D, \hat{\phi}) + L_{\text{repeatability}}(D, \hat{\phi}) \ .$$

Data set D obtained from random exploration

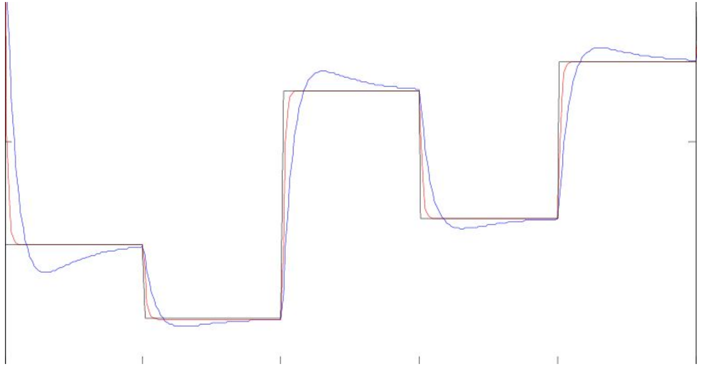
Learns state encoder: $\hat{\phi}(o_t) = s_t$

Simplicity prior implicit in compressing observation to lower dimensional space

Robotic Priors: Temporal Coherence

$$L_{\text{temporal coherence}}(D, \hat{\phi}) = \mathbf{E}\left[\|\Delta \hat{s}_t\|^2 \right]$$

- Enforces finite state “velocity”: $\Delta \hat{s}_t = \hat{s}_{t+1} - \hat{s}_t$
    - Smoothing effect



- i.e. represents state continuity
    - Intuition: physical objects cannot move from A to B in zero time
    - Newton’s First Law: Inertia

Robotic Priors: Proportionality

$$L_{\text{proportionality}}(D, \hat{\phi}) = \mathbf{E}\left[(\|\Delta \hat{s}_{t_2}\| - \|\Delta \hat{s}_{t_1}\|)^2 \;\middle|\; a_{t_1} = a_{t_2} \right]$$

- Enforces proportional responses to inputs
  - Similar actions at different times, similar magnitude of changes
  - Intuition: push harder, go faster
  - Newton's Second Law: *F* = *ma*
- Computational limitations:
  - Cannot compare all O(N^2) pairs of prior states
  - Instead only compare states K time steps apart
  - Also, $\pi_{explore}(s_t) = \pi_{explore}(s_{t+k})$ for more proportional responses in data

Robotic Priors: Causality

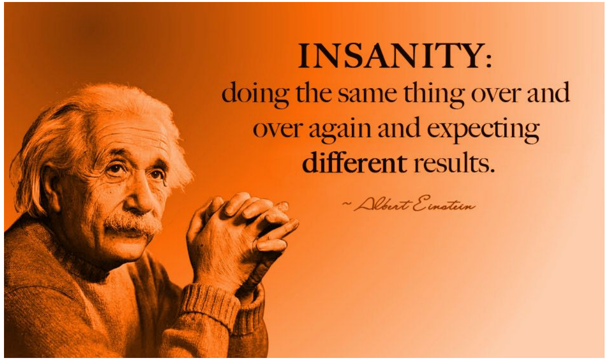
$$L_{\text{causality}}(D, \hat{\phi}) = \mathbf{E}\left[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|} \;\middle|\; a_{t_1} = a_{t_2}, r_{t_1+1} \neq r_{t_2+1} \right]$$

- Enforces state differentiation for different rewards
  - Similar actions at different times, but different rewards → different states
  - Same computational limitations

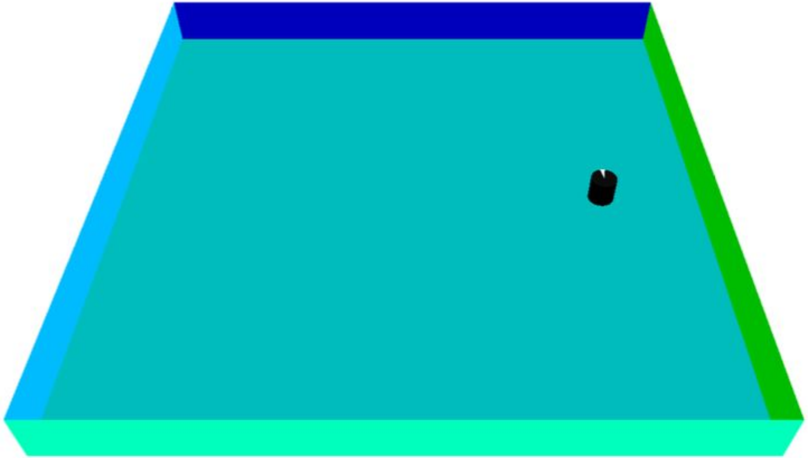
Robotic Priors: Repeatability

$$L_{\text{repeat.}}(D, \hat{\phi}) = \mathbf{E}\left[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|} \|\Delta\hat{s}_{t_2} - \Delta\hat{s}_{t_1}\|^2 \;\middle|\; a_{t_1} = a_{t_2}\right]$$

- Closer states should have similar reactions for same action at different times
  - Another form of coherence across time
  - If there are different reactions to same action from similar states, separate states more
  - Assumes determinism with full observability



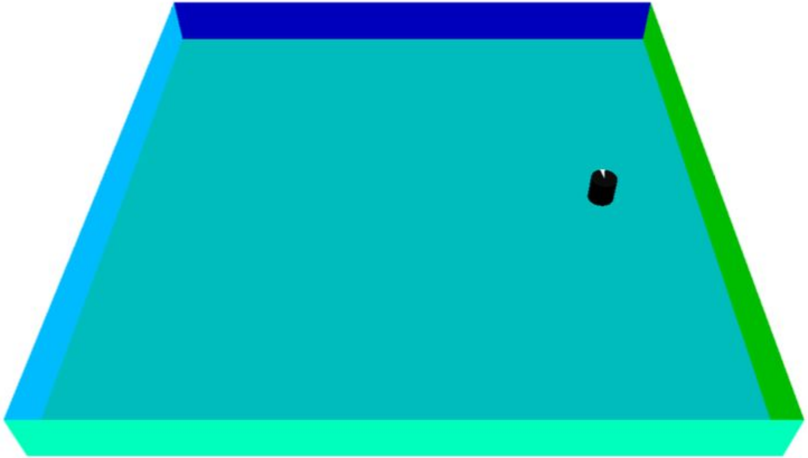
Experiments



Robot Navigation

Slot Car Racing

Experiments: Robot Navigation



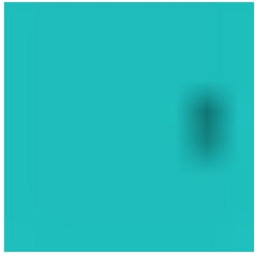
Robot Navigation

State:

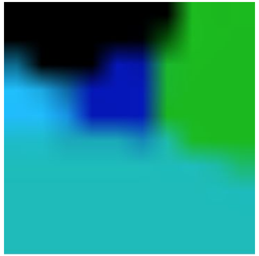
(x,y)

Observation:

10x10 RGB (Downsampled)



OR



Top-Down Egocentric

Action:

(Up, Right) Velocities ∈ [-6, -3, 0, 3, 6]

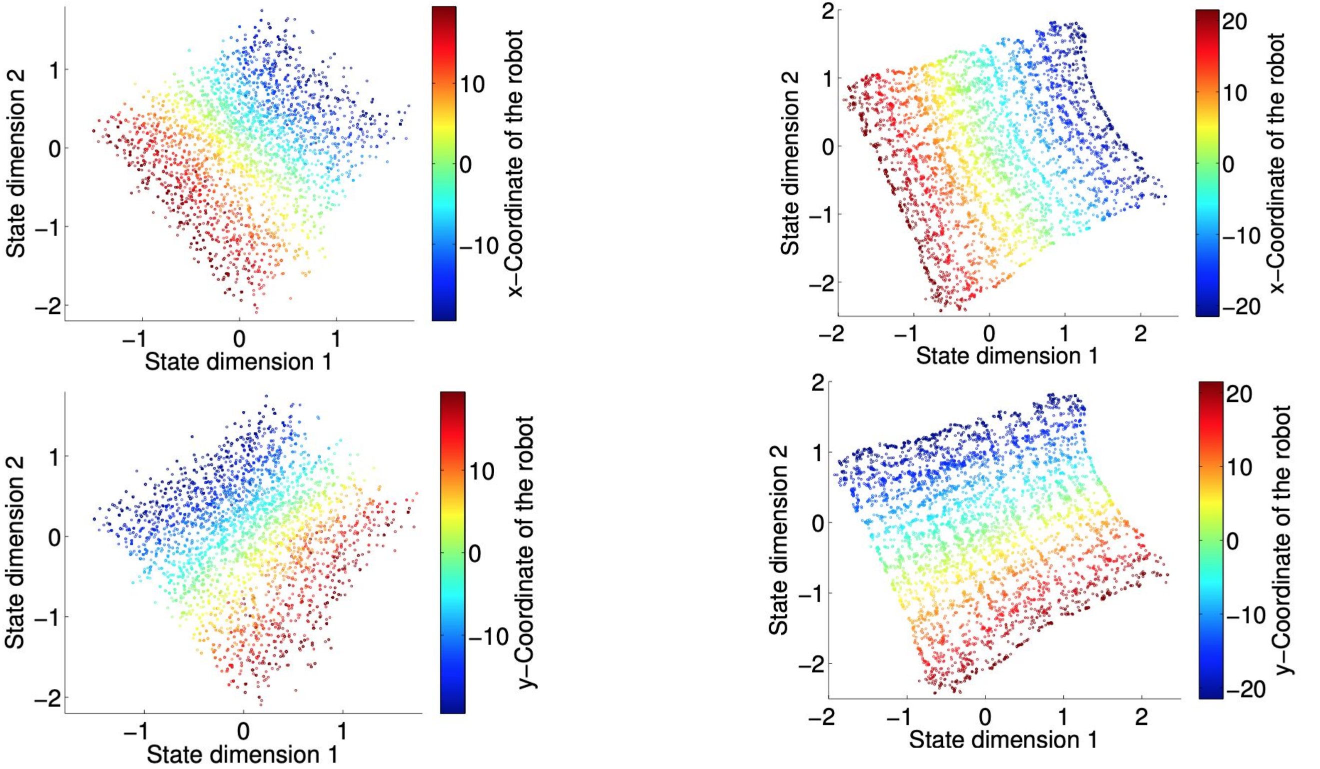
Reward:

+10 for goal corner, -1 for hitting wall

Learned States for Robot Navigation

x_{gt}

y_{gt}



Top-Down View

Egocentric View

Experiments: Slot Car Racing

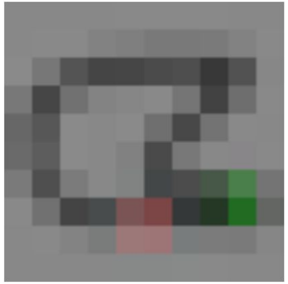
Slot Car Racing

State:

Θ (Red car only)

Observation:

10x10 RGB (Downsampled)



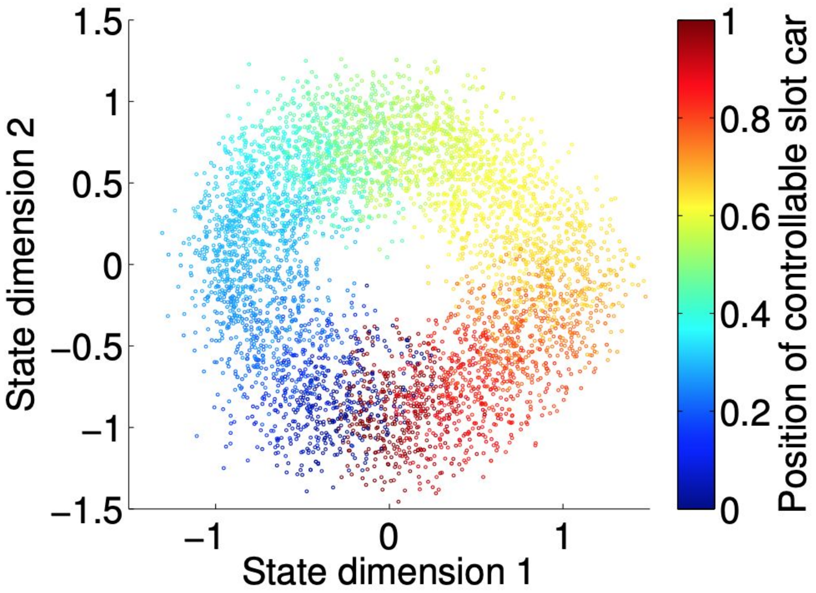
Action:

Velocity $\in$ [.01, .02, ..., 0.1]

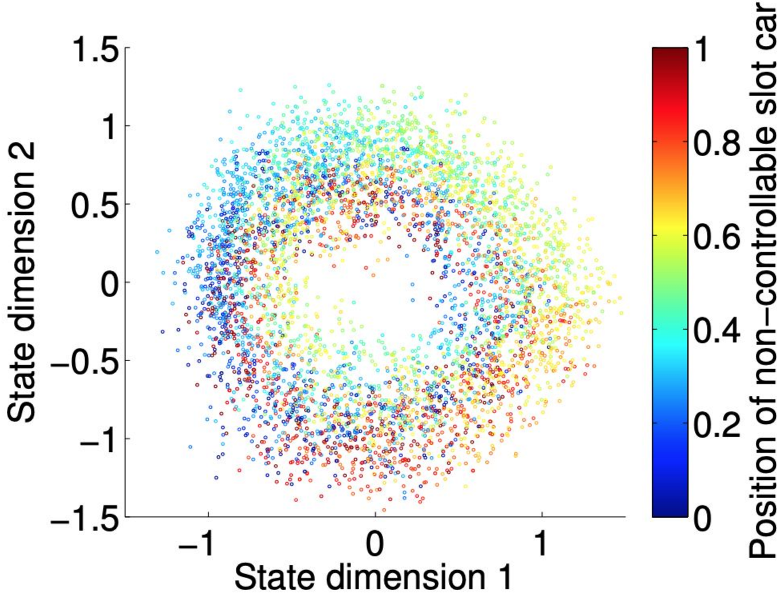
Reward:

Velocity, or -10 for flying off a sharp turn

Learned States for Slot Car Racing

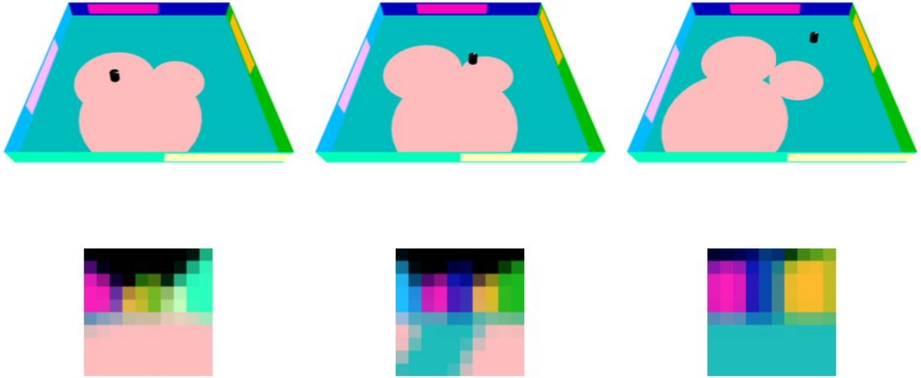


Red (Controllable) Car



Green (Non-Controllable) Car

Reinforcement Learning Task: Extended Navigation



State:

(x, y, θ)

Observation:

10x10 RGB (Downsampled)

Egocentric

Action:

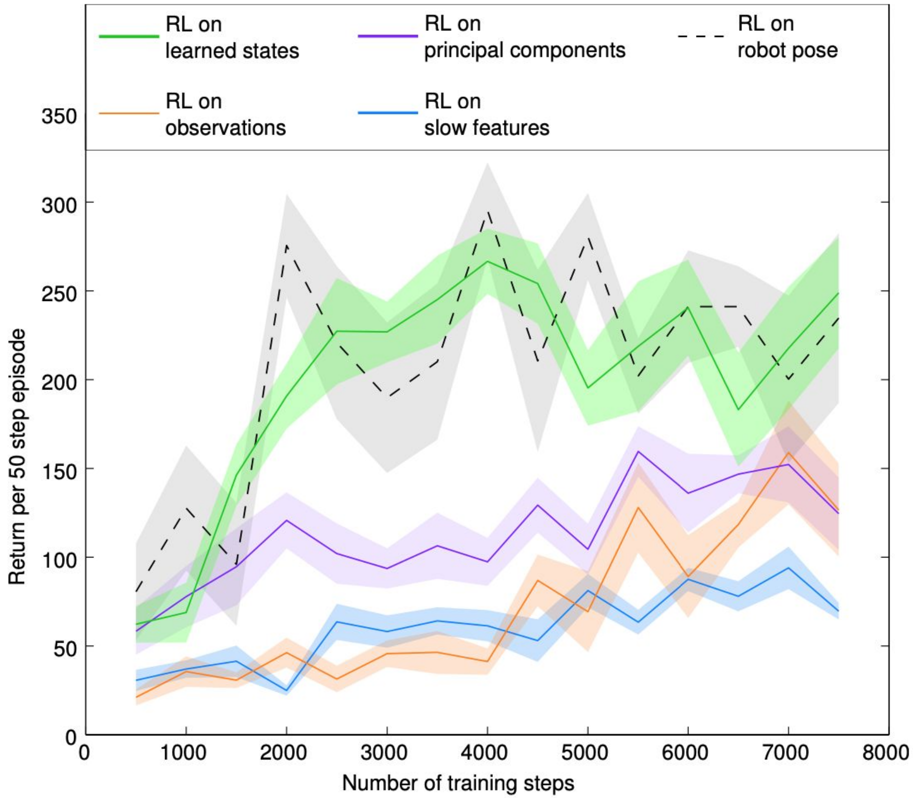
Translational Velocity ∈ [-6, -3, 0, 3, 6]

Rotational Velocity ∈ [-30,-15,0,15, 30]

Reward:

+10 for goal corner, -1 for hitting wall

RL for Extended Navigation Results



Takeaways

- State representation is an inherent sub-challenge in learning for robotics
- General priors can be useful in learning generalizable representations
- Physical environments have physical priors
- Many physical priors can be encoded in simple loss terms

Strengths and Weaknesses

Strengths:

- Well-written and organized
  - Provides a good summary of related works
- Motivates intuition behind everything
- Extensive experiments (within the tasks)
- Rigorous baselines for comparison

Weaknesses:

- Experiments are limited to toy tasks
  - No real robot experiments
- Only looks at tasks with slow-changing relevant features
- Fully-observable environments
- Does not evaluate on new tasks to show feature generalization
- Lacks ablative analysis on loss

Discussion

- Is a good representation sufficient for sample efficient reinforcement learning?
  - A. No, in worst case, it is still lower-bounded by exploration time exponential in time horizon
  - This is even true in the case where Q* or pi* is a linear mapping of states
- Does this mean SRL or RL is useless?
  - Not necessarily:
    - Unknown r(s, a) is what makes problem difficult
    - Most feature extractors induce a “hard MDP” instance
    - If data distribution fixed, can achieve polynomial upper bound in sample complexity
- For efficient value-based learning, are there necessary assumptions in reward distribution structure necessary for efficient learning?
  - What are types of reward functions or policies that could impose this structure?
- What are some important tasks that are counterexamples to these priors?

References

Rico Jonschkowski and Oliver Brock. State Representation Learning in Robotics: Using Prior Knowledge about Physical Interaction. Robotics: Science and Systems, 2014.

Martin Riedmiller. Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method. In 16th European Conference on Machine Learning (ECML), pages 317–328, 2005.

Du, Simon S., et al. "Is a Good Representation Sufficient for Sample Efficient Reinforcement Learning?." arXiv preprint arXiv:1910.03016 (2019).

References

[1] Lange, Sascha, Martin Riedmiller, and Arne Voigtländer. "Autonomous reinforcement learning on raw visual input data in a real world application." *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012.

[2] Legenstein, Robert, Niko Wilbert, and Laurenz Wiskott. "Reinforcement learning on slow features of high-dimensional input streams." *PLoS computational biology* 6.8 (2010): e1000894.

[3] Höfer, Sebastian, Manfred Hild, and Matthias Kubisch. "Using slow feature analysis to extract behavioural manifolds related to humanoid robot postures." *Tenth International Conference on Epigenetic Robotics*. 2010.

[4] Luciw, Matthew, and Juergen Schmidhuber. "Low complexity proto-value function learning from sensory observations with incremental slow feature analysis." *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2012.

[5] Bowling, Michael, Ali Ghodsi, and Dana Wilkinson. "Action respecting embedding." *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005.

[6] Boots, Byron, Sajid M. Siddiqi, and Geoffrey J. Gordon. "Closing the learning-planning loop with predictive state representations." *The International Journal of Robotics Research* 30.7 (2011): 954-966.

[7] Sprague, Nathan. "Predictive projections." *Twenty-First International Joint Conference on Artificial Intelligence*. 2009.

[8] Menache, Ishai, Shie Mannor, and Nahum Shimkin. "Basis function adaptation in temporal difference reinforcement learning." *Annals of Operations Research* 134.1 (2005): 215-238.

[9] Jonschkowski, Rico, and Oliver Brock. "Learning task-specific state representations by maximizing slowness and predictability." *6th international workshop on evolutionary and reinforcement learning for autonomous robot systems (ERLARS)*. 2013.

[10] Hutter, Marcus. "Feature reinforcement learning: Part I. unstructured MDPs." *Journal of Artificial General Intelligence* 1.1 (2009): 3-24.

[11] Martin Riedmiller. Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method. In 16th European Conference on Machine Learning (ECML), pages 317–328, 2005.

Priors

- **Simplicity**: For a given task, only a small number of world properties are relevant
- **Temporal Coherence**: Task-relevant properties of the world change gradually over time
- **Proportionality**: The amount of change in task-relevant properties resulting from an action is proportional to the magnitude of the action
- **Causality**: The task-relevant properties together with the action determine the reward
- **Repeatability**: The task-relevant properties and the action together determine the resulting change in these properties

Regression on Learned States