State Representation Learning for Robotics

David Filliat
U2IS - ENSTA ParisTech/INRIA FLOWERS

With N. Díaz-Rodríguez, T. Lesort, A. Raffin, A. Hill, R. Traoré, M. Seurin, J-F. Goudou, ...
**DREAM Project**

**Deferred Restructuring of Experience in Autonomous Machines**
- Learning representations from experience
- Improve learning speed for next ‘day’
States?

- Often, robot controllers require simple, ‘high-level’, low dimension inputs (the ‘state’ of the world)
  - E.g., *grasping*: object position; *driving*: road direction, obstacle

- Vision based control requires filtering to get this information
  - Many solutions, often hand-crafted, task specific
Why learning states?

- Automatic adaptation to new task
  - Discover the relevant state from exploration/demonstrations

- Visual processing made huge progress with deep learning
  - Possibly better robustness in changing environments
  - Specificity of the robotics context?

- Controllers are easier to train in such lower dimension
  - Possibly faster than end-to-end approaches
  - Could help transfer across tasks
State Representation Learning vs Representation Learning

- SRL is a particular case of Representation Learning
- SRL entails a control/robotics context where actions are possible
- Often SRL looks for interpretable info./ info. with a *physical* meaning
  - e.g., position, speed, angle...
- Like RL, SRL can be an objective by itself, but is often present in more general approaches
  - E.g., as an auxiliary task in RL
What is a good state?
A good state representation is:

1. Markovian, i.e. it summarizes all the necessary information to be able to choose an action within the policy, by looking only at the current state.

2. Able to:
   a. Represent the true value of the current state well enough for policy improvement.
   b. Generalize the learned value-function to unseen states with similar futures.

3. Low dimensional for efficient estimation.

[Böhmer’15]
Other perspective

[Achille and Soatto, 2017]

A good state representation should be:

- Sufficient
- As efficient as possible (i.e., easy to work with, e.g., factorizing the data-generating factors)
- Minimal (from all possible representations, take the most efficient one).
How to learn states?
State Representation Learning (SRL) in RL context

SRL

Reinforcement Learning

Motor Commands

CNN

State

NN

$O_t$

$S_t$

$A_t$

Robot

$s_t = \phi(O_t)$

$a_t = \pi(s_t)$

World
Learning states: challenges

- Learning without direct supervision
  - In a developmental/autonomous perspective
  - True state not available

- Solution: Use ‘self-supervision’
  - RL framework
  - Exploit observations, actions, rewards
Overview of existing approaches

Autoencoders

Forward Models

Inverse models

Priors
Reconstructing the observation

- Train a state that is sufficient for reconstructing the input observation
  - AE, DAE, VAE
  - GANs
- Downside: sensitive to irrelevant variations (vs actions)

![Variational Autoencoder Diagram]

\[
\begin{align*}
s_t &= \phi(o_t; \theta_\phi) \\
\hat{o}_t &= \phi^{-1}(s_t; \theta_{\phi^{-1}})
\end{align*}
\]
Deep Spatial Autoencoders

[Finn’15]

- Learns a representation that corresponds to objects coords. (image space):

\[
\text{feature encoder } h_{\text{enc}}(I_t) = f_t
\]

- Torque control & 2nd-order dynamical system requires features \( \dot{f}_t \) and their velocities

\[
\text{full state } x_t = [\ddot{x}_t; \dot{f}_t; \dot{f}_t]
\]
Forward models

- Find state from which it is easy to predict next state
  - Impose constraints on forward model (e.g., linear model)
- Naturally discard irrelevant features
- May be useful in model based RL

\[ \hat{s}_{t+1} = f(s_t, a_t; \theta_{fwd}) \]
Inverse models

- Find a state sufficient to recover action from 2 observations
  - Impose constraints on model (e.g., linear model)
- Focus on states that can be controlled
- Useful for a direct control model

\[ \hat{a}_t = g(s_t, s_{t+1}; \theta_{inv}) \]
Intrinsic Curiosity Module (ICM)

- Build states using forward / inverse models as part of RL
- Exploit error as a curiosity signal to improve exploration in RL

Curiosity-driven Exploration by Self-supervised Prediction

(a) learn to explore on Level-1  (b) explore faster on Level-2
Priors models

- Encode high-level constraints on the states
  - Temporal continuity
  - Controllability
  - Inertia, etc.

- May exploit rewards

\[
Loss = \mathcal{L}_{prior}(s_1, \ldots, s_n; \theta_\phi | c)
\]
Robotic Priors

[Jonschkowski et. al. 2015]

- Use *a priori* knowledge to learn representations relevant to the task

\[ L_{Temp}(D, \hat{\phi}) = \mathbb{E}[(\| \Delta \hat{s}_t \|)^2], \]  

\[ L_{Prop}(D, \hat{\phi}) = \mathbb{E}[(\| \Delta \hat{s}_{t_2} \| - \| \Delta \hat{s}_{t_1} \|)^2 | a_{t_1} = a_{t_2}], \]  

\[ L_{Rep}(D, \hat{\phi}) = \mathbb{E}[e^{-\| \Delta s_{t_2} - \Delta s_{t_1} \|^2} | \Delta s_{t_2} - \Delta s_{t_1} \|^2 | a_{t_1} = a_{t_2}], \]  

\[ L_{Caus}(D, \hat{\phi}) = \mathbb{E}[e^{-\| \Delta s_{t_2} - \Delta s_{t_1} \|^2} | a_{t_1} = a_{t_2}, r_{t_1+1} \neq r_{t_2+1}], \]
Robotic Priors

[Jonschkowski et. al. 2015]
Overview of existing approaches

- Integrating several approaches
Embed to Control (E2C)

[Watter’18] [Karl et al., 2016] [Krishnan et al., 2015]

- Reconstruct observation using VAE
- Learn a locally linear forward model
- Exploit this forward model in optimal control setting

\[ \hat{s}_{t+1} \sim \mathcal{N}(\mu = W \ast \hat{s}_t + U \ast a_t + V, \sigma) \]
SRL Evaluation

● Tasks
  ○ Each paper evaluate on its tasks
    ○ Atari games (2D), VizDoom (3D)
    ○ Mobile robots, labyrinths, navigation grids
    ○ Robotics manipulation skills: Grasping, pushing ...

● Metrics
  ○ Best metric is usually RL performance, but unstable, time consuming
  ○ Distortion/correlation with Ground truth
  ○ We propose : KNN-MSE : Measuring the dispersion of learned states vs Ground truth
SRL: state of the art

State Representation Learning for Control: An Overview
Timothée Lesort\textsuperscript{1,2}, Natalia Díaz-Rodríguez\textsuperscript{1}, Jean-François Goudou\textsuperscript{2}, and David Filliat\textsuperscript{1}

\textsuperscript{1}U2IS, ENSTA ParisTech, Inria FLOWERS team, Université Paris Saclay, Palaiseau, France, \{timothée.lesort, natalia.diaz, david.filliat\}@ensta-paristech.fr
\textsuperscript{2}Thales, Thales Laboratory, Palaiseau, France, \{jean-francois.goudou@thalesgroup.com

Contents

1 Introduction

2 Formalism and definitions
2.1 SRL Formalism
2.2 State representation characteristics
2.3 State representation learning applications

3 Learning objectives
3.1 Reconstructing the observation
3.2 Learning a forward model
3.3 Learning an inverse model
3.4 Using feature adversarial learning
3.5 Exploiting rewards
3.6 Other objective functions
3.7 Using hybrid objectives

4 Building blocks of State Representation Learning
4.1 Learning tools
4.1.1 Auto-encoders
4.1.2 Denoising auto-encoders (DAE)
4.1.3 Variational auto-encoders (VAE)
4.1.4 Siamese networks
4.2 Observation/action spaces
4.3 Evaluating learned state representations
4.4 Evaluation scenarios

5 Discussion and future trends
5.1 SRL models for autonomous agents
5.2 Assessment, comparison and reproducibility in SRL
5.3 Providing interpretable systems

6 Conclusion

7 Acknowledgements
SRL Toolbox

- A set of tools for SRL
- Tasks / dataset
  - 2 different tasks with variants
  - Based on a fast simulator (pybullet)
  - Ground truth for each task
  - Real robot dataset

[Raffin et al. 18]
SRL Toolbox

- A set of baselines
  - Auto Encoders
  - Variational Auto Encoders
  - Robotic priors
  - Forward Models
  - Inverse models
  - ...

- A set of evaluation tools
  - RL (OpenAI baselines)
    - PPO, CMA-ES, ARS, ...
  - KNN-MSE
  - Ground truth correlation

![Ground Truth States vs Learned States](image1)

**RL Performance**

![Learning Curve](image2)
SRL Toolbox

- A set of visualization tools

![Correlation Matrix: $\hat{s} = \text{Predicted states} | \bar{s} = \text{Agent's position}$](image)

State / GT correlation

![State vs State plot](image)
## SRL Toolbox

- A set of visualization tools

<table>
<thead>
<tr>
<th>Real-time SRL</th>
<th>Interactive scatter</th>
<th>Latent visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Real-time SRL" /></td>
<td><img src="image2.png" alt="Interactive scatter" /></td>
<td><img src="image3.png" alt="Latent visualization" /></td>
</tr>
</tbody>
</table>
SRL Toolbox

- A set of visualization tools
SRL Toolbox

● Some lessons learned
  ○ Many method performances are quite task specific
    ■ E.g. robotics prior fail on robotic arm
  ○ Autoencoders/VAE work quite well if extreme (small or large) noise
  ○ Predicting a forward and inverse model often efficient, but don’t focus on goal (not using rewards)
  ○ SRL + RL usually more efficient than end-to-end RL

● Some perspectives
  ○ Exploit GANs (BiGAN, CycleGAN, …)
Questions ?