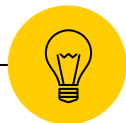


Finding State-Action Similarities in Tabular Reinforcement Learning Using Low-Dimensional Embeddings



Gabriel Rubin



Abstract

Reinforcement Learning Training is Slow

The training process can often be expensive resource-wise and time consuming.

My proposal

A method to automatically detect state-action similarities in temporal difference learning methods and use the same benefits of SASS without a domain-specific function.

Previous Work On State Similarities (SASS)

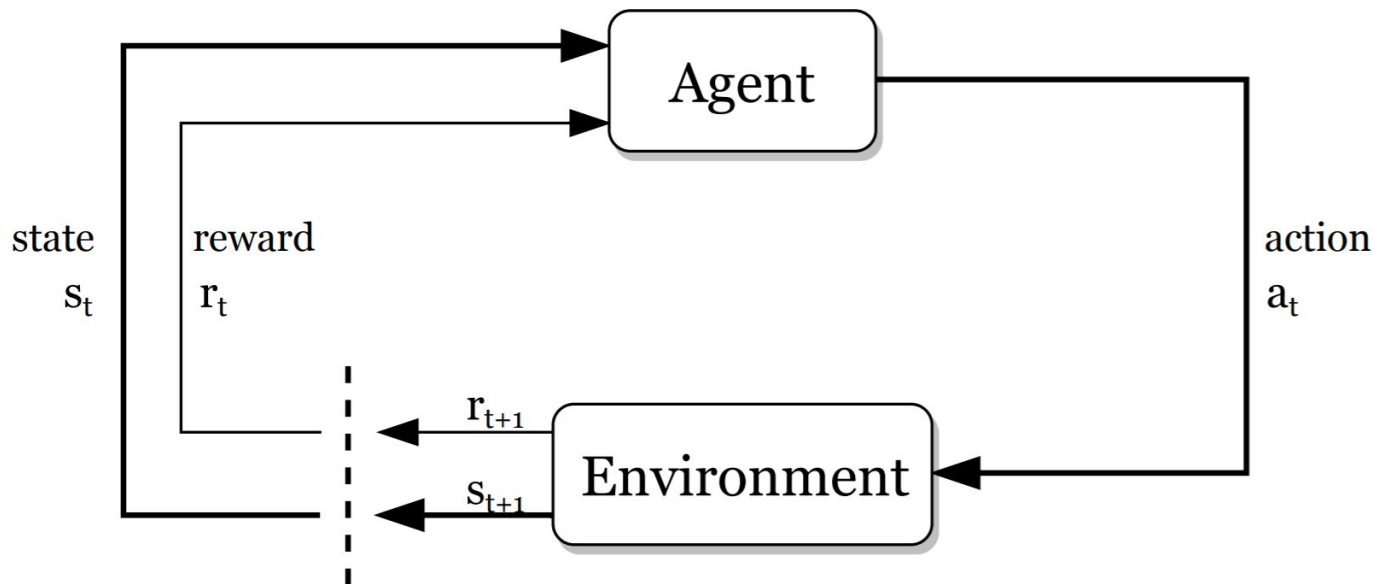
Previous work showed that domain-specific state-action similarity functions can be used to speed up the training process of these learning methods on many domains.

Super Mario Benchmark

In order to evaluate my similarity function, I will compare its performance to that of the custom made similarity function from previous work on an SASS agent that will be trained on the Super Mario Bros domain.



Reinforcement Learning





Reinforcement Learning

Markov Decision Process

- Mathematical formulation of the RL problem
- **Markov property**: Current state completely characterises the state of the world

Defined by: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{P}, \gamma)$

\mathcal{S} : set of possible states

\mathcal{A} : set of possible actions

\mathcal{R} : distribution of reward given (state, action) pair

\mathbb{P} : transition probability i.e. distribution over next state given (state, action) pair

γ : discount factor



Reinforcement Learning

actions = {

1. right →

2. left ←

3. up ↑

4. down ↓

}

states

★			
			★





Set a negative “reward”
for each transition
(e.g. $r = -1$)

Objective: reach one of terminal states (greyed out) in
least number of actions



Reinforcement Learning

actions = {

1. right 
2. left 
3. up 
4. down 

}

states

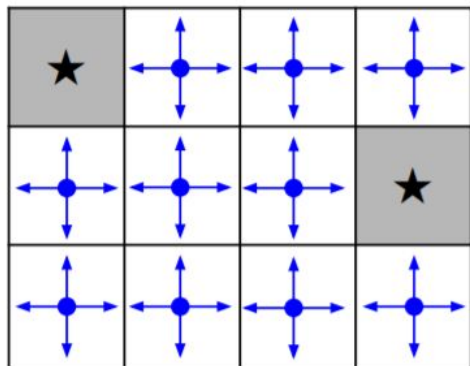
★			
			★

Set a negative “reward”
for each transition
(e.g. $r = -1$)

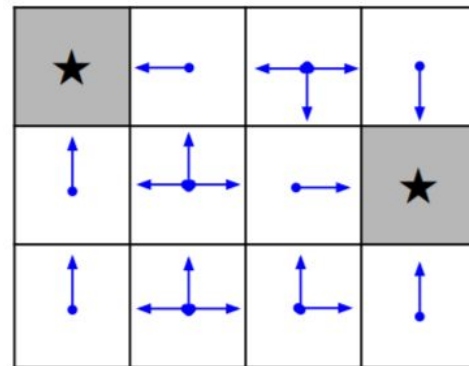
Objective: reach one of terminal states (greyed out) in
least number of actions



Reinforcement Learning



Random Policy



Optimal Policy



Previous work on Speeding RL with Similarities

SASS

New method that can speed up the training process significantly for temporal difference methods, such as Q-Learning, by “spreading” the Q-function estimates of an state to other similar states

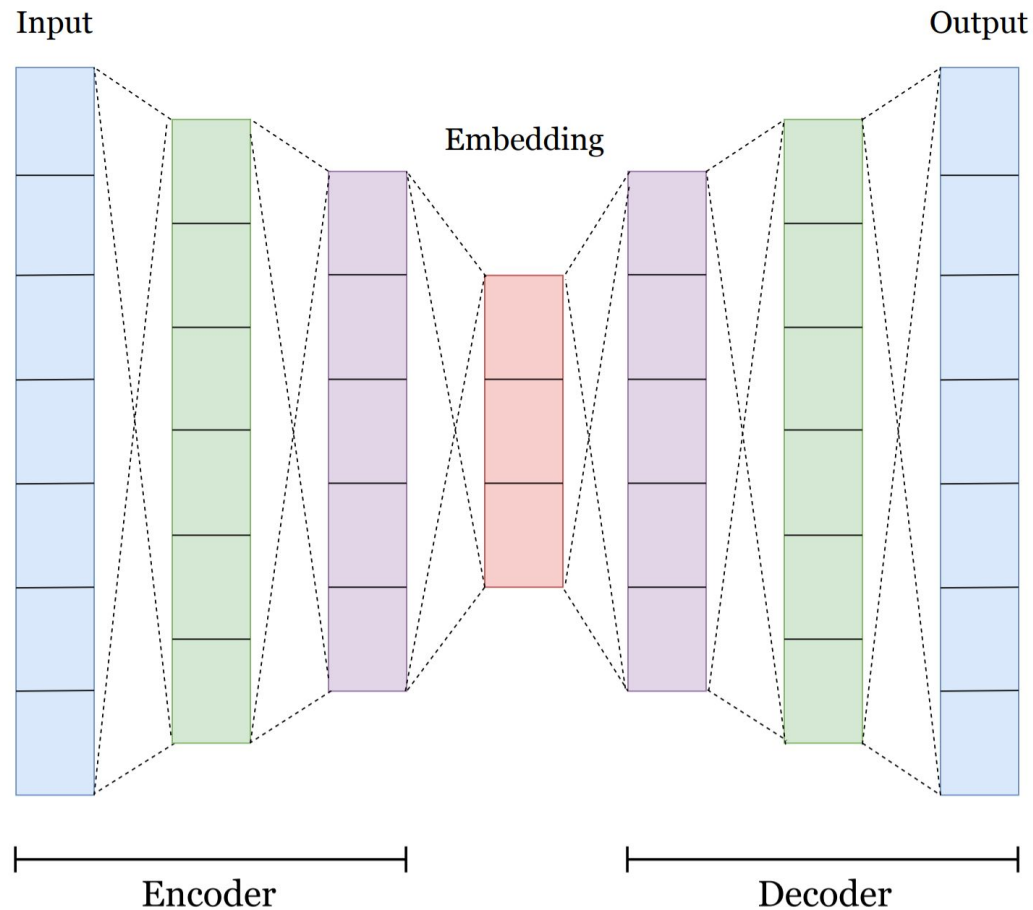
Similarity Functions

SASS considers a custom similarity function that relies on a designer's input in order to compare state-action pairs

$$\sigma : S \times A \times S \times A \mapsto [0, 1]$$



Autoencoders and Embeddings





Super Mario Domain





Project Management

TIMELINE 1: *Project Milestones*

-
- Oct. 11 • Project start
 - Oct. 21 • Super Mario domain implementation
 - Nov. 1 • Autoencoder implementation
 - Nov. 8 • Similarity method implementation
 - Nov. 19 • Optimization and fine-tuning
 - Nov. 26 • Paper and Results
 - Nov. 29 • Presentation
-



Thanks!

Any questions ?