Learning Successor States and Reward-Free RL through the Forward-Backward Model

A Mathematical Viewpoint

Alessio Russo

March 2025

Boston University

Introduction

Motivation

Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint

Léonard Blier, Corentin Tallec, Yann Ollivier

January 19, 2021

Abstract

In reinforcement learning, temporal difference-based algorithms can be sample-inefficient: for instance, with sparse rewards, no learning occurs until a reward is observed. This can be remedied by learning richer objects, such as a model of the environment, or *successor states*. Successor states model the expected future state occupancy from any given state [Dayan, 1993, Kulkarni et al., 2016], and summarize all paths in the environment for a given policy. They are related to *goal-dependent value functions*, which learn how to reach arbitrary states.

`https://controllable-agent.metademolab.com/` Ref. [BTO21, TO21, TRO22].

Motivation

Today we will talk about...

1. Successor states in tabular domains
2. Successor states in continuous domains
   - i.e., how to learn sparse rewards in continuous environments
3. The forward-backward model $\Rightarrow$ very important for extending Reward-Free RL to Deep RL!

Overview i

Introduction

- Successor states capture the expected future occupancy of each state, starting from a given state, under a fixed policy.
- They generalize the notion of *value functions* to goal-reaching problems: each potential *goal state* is treated as providing a reward upon arrival.
- Learning successor states can be more efficient than learning separate value functions for each goal, especially in environments with sparse rewards.

Introduction

- Successor states capture the expected future occupancy of each state, starting from a given state, under a fixed policy.
- They generalize the notion of *value functions* to goal-reaching problems: each potential *goal state* is treated as providing a reward upon arrival.
- Learning successor states can be more efficient than learning separate value functions for each goal, especially in environments with sparse rewards.

Introduction

- Successor states capture the expected future occupancy of each state, starting from a given state, under a fixed policy.
- They generalize the notion of *value functions* to goal-reaching problems: each potential *goal state* is treated as providing a reward upon arrival.
- Learning successor states can be more efficient than learning separate value functions for each goal, especially in environments with sparse rewards.

Markov Process and Successor States

We consider a **Markov reward process** with state space S (possibly infinite or continuous), **transition operator** P^π (for some policy π), and discount factor γ.

Successor state operator M

For a fixed policy π, the successor state operator M maps any starting state s to a *measure* over future states, defined by:

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

for any measurable set of states A.

In a finite state space, M can be viewed as a matrix in $\mathbb{R}^{|S|\times|S|}$ given by

$$M = (I - \gamma P^\pi)^{-1}.$$

$M_{s,s'}$ is the discounted expected number of visits to state s' starting from s.

Markov Process and Successor States

We consider a **Markov reward process** with state space S (possibly infinite or continuous), **transition operator** P^π (for some policy π), and discount factor γ.

Successor state operator M

For a fixed policy π, the successor state operator M maps any starting state s to a *measure* over future states, defined by:

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

for any measurable set of states A.

In a finite state space, M can be viewed as a matrix in $\mathbb{R}^{|S|\times|S|}$ given by

$$M = (I - \gamma P^\pi)^{-1}.$$

$M_{s,s'}$ is the discounted expected number of visits to state s' starting from s.

Markov Process and Successor States

We consider a **Markov reward process** with state space S (possibly infinite or continuous), **transition operator** P^π (for some policy π), and discount factor γ.

Successor state operator M

For a fixed policy π, the successor state operator M maps any starting state s to a *measure* over future states, defined by:

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

for any measurable set of states A.

In a finite state space, M can be viewed as a matrix in $\mathbb{R}^{|S| \times |S|}$ given by

$$M = (I - \gamma P^\pi)^{-1}.$$

$M_{s,s'}$ is the discounted expected number of visits to state s' starting from s.

Markov Process and Successor States: Feature Maps

Successor state operator M

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

For any feature map $\phi : S \to \mathbb{R}^d$ we can define the expected discounted sum of future state features

$$\psi(s) = \mathbb{E}\left[\sum_{t \geq 1} \gamma^{t-1} \phi(s_{t+1}) | s_1 = s\right].$$

Hence[1]

$$\psi(s) = \int_S \phi(s') M(s, \mathrm{d}s')$$

[1] For tabular domains we instead have $\psi(s) = \sum_{s'} (I - \gamma P^{\pi})_s^{-1} \phi(s')$

Markov Process and Successor States: how to learn this object?

Successor state operator M

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

- How do we learn M?
- For continuous domains the question is tricky. **We need to learn a density!**
- Consider a dominating measure ρ over S [2]. We define the density as

$$m(s, s') = \frac{M(s, \mathrm{d}s')}{\rho(\mathrm{d}s')}$$

You can think of ρ as the data distribution induced by π.

[2] E.g., Lebesgue, Gaussian. See also Radon-Nykodim derivative.

Markov Process and Successor States: how to learn this object?

Successor state operator M

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

- How do we learn M?
- For continuous domains the question is tricky. **We need to learn a density!**
- Consider a dominating measure ρ over S [2]. We define the density as

$$m(s, s') = \frac{M(s, \mathrm{d}s')}{\rho(\mathrm{d}s')}$$

You can think of ρ as the data distribution induced by π.

[2] E.g., Lebesgue, Gaussian. See also Radon-Nykodim derivative.

Markov Process and Successor States: how to learn this object?

Successor state operator M

$$M(s, A) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t \in A \mid s_0 = s, \pi),$$

- How do we learn M?
- For continuous domains the question is tricky. **We need to learn a density!**
- Consider a dominating measure ρ over S [2]. We define the density as

$$m(s, s') = \frac{M(s, \mathrm{d}s')}{\rho(\mathrm{d}s')}$$

You can think of ρ as the data distribution induced by π.

[2] E.g., Lebesgue, Gaussian. See also Radon-Nykodim derivative.

Forward TD for Successor States: Tabular Case

The Forward Bellman Equation

Theorem (Bellman Equation for Successor States)

The successor state operator M is the unique operator satisfying

$$M = I + \gamma P^{\pi} M,$$

on the state space (M is a fixed point of the Bellman operator $\mathcal{T}M := I + \gamma P^{\pi} M$)

Proof Sketch.

Rearranging $M = I + \gamma P^{\pi} M$ gives $(I - \gamma P^{\pi})M = I$. Thus M is a right-inverse of $(I - \gamma P^{\pi})$. By standard results (e.g. Neumann series), $I - \gamma P^{\pi}$ is invertible for $0 \leq \gamma < 1$, with inverse $(I - \gamma P^{\pi})^{-1}$. Therefore $M = (I - \gamma P^{\pi})^{-1}$ is the unique solution. $\square$

The Forward Bellman Equation

Theorem (Bellman Equation for Successor States)

The successor state operator M is the unique operator satisfying

$$M = I + \gamma P^{\pi} M,$$

on the state space (M is a fixed point of the Bellman operator $\mathcal{T}M := I + \gamma P^{\pi} M$)

Proof Sketch.

Rearranging $M = I + \gamma P^{\pi} M$ gives $(I - \gamma P^{\pi})M = I$. Thus M is a right-inverse of $(I - \gamma P^{\pi})$. By standard results (e.g. Neumann series), $I - \gamma P^{\pi}$ is invertible for $0 \leq \gamma < 1$, with inverse $(I - \gamma P^{\pi})^{-1}$. Therefore $M = (I - \gamma P^{\pi})^{-1}$ is the unique solution. $\square$

Bellman Operator Contractivity

Proposition (Contraction of Bellman Operator on M)

Equip the space of bounded operators on S with the sup-norm $\|\cdot\|_\infty$. The Bellman update $\mathcal{T}M = I + \gamma P^\pi M$ is a γ-contraction in this norm[3].

Proof Sketch.

For any two operators M_1, M_2:

$$\|\mathcal{T}M_1 - \mathcal{T}(M_2)\|_\infty = \|\gamma P^\pi (M_1 - M_2)\|_\infty \leq \gamma \|M_1 - M_2\|_\infty.$$

$\square$

For a learning rate $\eta \leq 1$, repeated updates $M_{n+1} \leftarrow (1-\eta)M_n + \eta(I + \gamma P^\pi M_n)$ will converge to M.

[3] Consequently, iterated application of $\mathcal{T}$ converges to the unique fixed point M.

Bellman Operator Contractivity

Proposition (Contraction of Bellman Operator on M)

Equip the space of bounded operators on S with the sup-norm $\|\cdot\|_\infty$. The Bellman update $\mathcal{T}M = I + \gamma P^\pi M$ is a γ-contraction in this norm[3]*.*

Proof Sketch.

For any two operators M_1, M_2:

$$\|\mathcal{T}M_1 - \mathcal{T}(M_2)\|_\infty = \|\gamma P^\pi(M_1 - M_2)\|_\infty \leq \gamma\|M_1 - M_2\|_\infty.$$

$\square$

For a learning rate $\eta \leq 1$, repeated updates $M_{n+1} \leftarrow (1-\eta)M_n + \eta(I + \gamma P^\pi M_n)$ will converge to M.

[3] Consequently, iterated application of $\mathcal{T}$ converges to the unique fixed point M.

Forward TD for Successor States: Tabular Case

The Bellman equation $M = I + \gamma P^{\pi} M$ suggests a TD-style iteration to learn M.

Definition (Tabular TD Update for M)

In a finite state space, maintain an estimate M as an $|S| \times |S|$ matrix. Upon observing a transition $s \to s'$ in the Markov process, update for all $s_2 \in S$:

$$M_{s,s_2} \leftarrow M_{s,s_2} + \eta\, \delta M_{s,s_2}, \text{ where } \delta M_{s,s_2} := \mathbf{1}_{\{s=s_2\}} + \gamma\, M_{s',s_2} - M_{s,s_2},$$

and η is the learning rate.

- Here $\mathbf{1}_{\{s=s_2\}}$ serves as a "reward" signal indicating if we have reached the target state s_2.
- This update is equivalent to performing one-step TD for each possible goal s_2 simultaneously.

Forward TD for Successor States: Tabular Case

The Bellman equation $M = I + \gamma P^\pi M$ suggests a TD-style iteration to learn M.

Definition (Tabular TD Update for M)

In a finite state space, maintain an estimate M as an $|S| \times |S|$ matrix. Upon observing a transition $s \to s'$ in the Markov process, update for all $s_2 \in S$:

$$M_{s,s_2} \leftarrow M_{s,s_2} + \eta\, \delta M_{s,s_2}, \text{ where } \delta M_{s,s_2} := \mathbf{1}_{\{s=s_2\}} + \gamma\, M_{s',s_2} - M_{s,s_2},$$

and η is the learning rate.

- Here $\mathbf{1}_{\{s=s_2\}}$ serves as a "reward" signal indicating if we have reached the target state s_2.
- This update is equivalent to performing one-step TD for each possible goal s_2 simultaneously.

Forward TD for Successor States: Tabular Case

The Bellman equation $M = I + \gamma P^\pi M$ suggests a TD-style iteration to learn M.

Definition (Tabular TD Update for M)

In a finite state space, maintain an estimate M as an $|S| \times |S|$ matrix. Upon observing a transition $s \to s'$ in the Markov process, update for all $s_2 \in S$:

$$M_{s,s_2} \leftarrow M_{s,s_2} + \eta\, \delta M_{s,s_2}, \text{ where } \delta M_{s,s_2} := \mathbf{1}_{\{s=s_2\}} + \gamma\, M_{s',s_2} - M_{s,s_2},$$

and η is the learning rate.

- Here $\mathbf{1}_{\{s=s_2\}}$ serves as a "reward" signal indicating if we have reached the target state s_2.
- This update is equivalent to performing one-step TD for each possible goal s_2 simultaneously.

Interpretation: State-Goal Process

The tabular TD update for M is equivalent to ordinary TD learning on an augmented *state-goal MDP*.

Equivalence with goal-conditioned value functions

- In this interpretation, we consider a process on pairs (s, g) where g is a fixed "goal" state:
  - Transition: $(s, g) \to (s', g)$ with $s' \sim P^\pi(\cdot|s)$.
  - Reward: $R^\pi(s, g) = 1$ if $s = g$ (and 0 otherwise).
- Let $Q(s, g)$ be the value (expected return) for this process.
  - Then the tabular successor representation $M_{s,g}$ learned using the TD-style approach is **exactly** $Q(s, g)$.
  - In other words, $M(s, \cdot)$ **learns the value of state** s **for every possible goal** g in parallel.

Interpretation: State-Goal Process

The tabular TD update for M is equivalent to ordinary TD learning on an augmented *state-goal MDP*.

Equivalence with goal-conditioned value functions

- In this interpretation, we consider a process on pairs (s, g) where g is a fixed "goal" state:
  - Transition: $(s, g) \to (s', g)$ with $s' \sim P^\pi(\cdot|s)$.
  - Reward: $R^\pi(s, g) = 1$ if $s = g$ (and 0 otherwise).
- Let $Q(s, g)$ be the value (expected return) for this process.
  - Then the tabular successor representation $M_{s,g}$ learned using the TD-style approach is **exactly** $Q(s, g)$.
  - In other words, $M(s, \cdot)$ **learns the value of state** s **for every possible goal** g in parallel.

Forward TD for Successor States: Continuous Case

Representing M in continuous state spaces

In continuous or large state spaces, storing M as a full matrix is infeasible.

- Instead, the idea is to represent M with a parameterized function $M_\theta(s,g)$. For example, assume:

$$m_\theta(s,s') \approx \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}$$

  where m_θ is a parametric function (e.g. a neural network) approximating the density of reaching s' from s.

In continuous spaces, M has a singular part due to the term I $(M = I + \cdots)$: for each s, the measure $M(s,\cdot)$ comprises a Dirac mass at s.

Representing M in continuous state spaces

In continuous or large state spaces, storing M as a full matrix is infeasible.

- Instead, the idea is to represent M with a parameterized function $M_\theta(s, g)$. For example, assume:

  $$m_\theta(s, s') \approx \frac{M(s, \mathrm{d}s')}{\rho(\mathrm{d}s')}$$

  where m_θ is a parametric function (e.g. a neural network) approximating the density of reaching s' from s.

In continuous spaces, M has a singular part due to the term I ($M = I + \cdots$): for each s, the measure $M(s, \cdot)$ comprises a Dirac mass at s.

Representing M in continuous state spaces

In continuous or large state spaces, storing M as a full matrix is infeasible.

- Instead, the idea is to represent M with a parameterized function $M_\theta(s, g)$. For example, assume:

  $$m_\theta(s, s') \approx \frac{M(s, \mathrm{d}s')}{\rho(\mathrm{d}s')}$$

  where m_θ is a parametric function (e.g. a neural network) approximating the density of reaching s' from s.

In continuous spaces, M has a singular part due to the term I ($M = I + \cdots$): for each s, the measure $M(s, \cdot)$ comprises a Dirac mass at s.

Forward TD with Function Approximation: Preamble

Goal

Find θ such that M_θ (parameterized by m_θ) satisfies the Bellman equation $M_\theta = I + \gamma P^\pi M_\theta$.

- We need some notion of distance. Define the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$.
- We can do this by minimizing the *Bellman error*

$$J(\theta) := \frac{1}{2}\|M_\theta - (I + \gamma P^\pi M_\theta)\|_\rho^2,$$

  and performing gradient descent on $J(\theta)$.
- However, **we know** from DQN that it's better to use a semi-stationary target. Therefore, we instead consider

$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2,$$

  for some target $M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar{\theta}}$ for some fixed $\bar{\theta}$.

Forward TD with Function Approximation: Preamble

Goal

Find θ such that M_θ (parameterized by m_θ) satisfies the Bellman equation $M_\theta = I + \gamma P^\pi M_\theta$.

- We need some notion of distance. Define the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$.
- We can do this by minimizing the *Bellman error*

$$J(\theta) := \frac{1}{2}\|M_\theta - (I + \gamma P^\pi M_\theta)\|_\rho^2,$$

  and performing gradient descent on $J(\theta)$.
- However, we know from DQN that it's better to use a semi-stationary target. Therefore, we instead consider

$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2,$$

  for some target $M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar\theta}$ for some fixed $\bar\theta$.

Forward TD with Function Approximation: Preamble

Goal

Find θ such that M_θ (parameterized by m_θ) satisfies the Bellman equation $M_\theta = I + \gamma P^\pi M_\theta$.

- We need some notion of distance. Define the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$.
- We can do this by minimizing the *Bellman error*

$$J(\theta) := \frac{1}{2}\|M_\theta - (I + \gamma P^\pi M_\theta)\|_\rho^2,$$

  and performing gradient descent on $J(\theta)$.
- However, we know from DQN that it's better to use a semi-stationary target. Therefore, we instead consider

$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2,$$

  for some target $M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar\theta}$ for some fixed $\bar\theta$.

Forward TD with Function Approximation: Preamble

Goal

Find θ such that M_θ (parameterized by m_θ) satisfies the Bellman equation $M_\theta = I + \gamma P^\pi M_\theta$.

- We need some notion of distance. Define the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s' \sim \rho}[f(s, s')]$ with $f(s, s') := M(s, \mathrm{d}s')/\rho(\mathrm{d}s')$.
- We can do this by minimizing the *Bellman error*

$$J(\theta) := \frac{1}{2}\|M_\theta - (I + \gamma P^\pi M_\theta)\|_\rho^2,$$

  and performing gradient descent on $J(\theta)$.
- However, **we know** from DQN that it's better to use a semi-stationary target. Therefore, we instead consider

$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2,$$

  for some target $M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar{\theta}}$ for some fixed $\bar{\theta}$.

Infinitely Sparse Rewards: A Cautionary Tale

- In a continuous state space, the reward $\mathbf{1}_{\{s=g\}}$ becomes a Dirac delta, which is zero with probability 1 for any given g.
- **Key insight:** do *not* rely on sampling this rare event directly. when we are at state s, we know that the goal $g = s$ was just achieved (for that particular g). Thus every transition provides some learning signal.
- We should be able to exploit this fact to compute a gradient. Expect $\partial_\theta m_\theta(s, s)$ to appear in the gradient update: it captures the self-transition (s to itself) which ensures the sparse Dirac reward still contributes.

Infinitely Sparse Rewards: A Cautionary Tale

- In a continuous state space, the reward $\mathbf{1}_{\{s=g\}}$ becomes a Dirac delta, which is zero with probability 1 for any given g.
- **Key insight:** do *not* rely on sampling this rare event directly. when we are at state s, we know that the goal $g = s$ was just achieved (for that particular g). Thus every transition provides some learning signal.
- We should be able to exploit this fact to compute a gradient. Expect $\partial_\theta m_\theta(s, s)$ to appear in the gradient update: it captures the self-transition (s to itself) which ensures the sparse Dirac reward still contributes.

Infinitely Sparse Rewards: A Cautionary Tale

- In a continuous state space, the reward $\mathbf{1}_{\{s=g\}}$ becomes a Dirac delta, which is zero with probability 1 for any given g.
- **Key insight:** do *not* rely on sampling this rare event directly. when we are at state s, we know that the goal $g = s$ was just achieved (for that particular g). Thus every transition provides some learning signal.
- We should be able to exploit this fact to compute a gradient. Expect $\partial_\theta m_\theta(s, s)$ to appear in the gradient update: it captures the self-transition (s to itself) which ensures the sparse Dirac reward still contributes.

Forward TD with Function Approximation

Theorem (TD Update for Successor States with Approximation)

Consider the model $M_\theta(s, \mathrm{d}s') = m_\theta(s, s')\rho(\mathrm{d}s')$. *For the loss* $J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2$, *the gradient of* $J(\theta)$ *is*

$$-\partial_\theta J(\theta) = \mathbb{E}_{s, s_2 \sim \rho, s' \sim P^\pi(s, \mathrm{d}s')} \left[\underbrace{\partial_\theta m_\theta(s, s)}_{\textit{Singular term}} + \underbrace{\partial_\theta m_\theta(s, s_2)(\gamma m_{\bar{\theta}}(s', s_2) - m_\theta(s, s_2))}_{\textit{Nxt step error}} \right]$$

where ρ *is a sampling distribution over states (e.g. stationary distribution, stationary buffer).*

- This means we update θ by sampling a transition $s \to s'$ and an independent random state s_2, and then computing the above gradient term.
- This algorithm has the *same expected update* as naive parallel TD, but avoids the problem of getting zero reward in continuous state spaces.

Forward TD with Function Approximation: Proof [1/3]

We now look at the proof of the theorem. Recall that $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')^2]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$. Also let $\langle M_1, M_2\rangle_\rho = \mathbb{E}_{s,s'\sim\rho}[f_1(s,s')f_2(s,s')]$.

- Recall that
$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2 \quad \text{and} \quad M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar\theta}$$
for some fixed $\bar\theta$.
- M_θ is absolutely continuous with respect to ρ while $M_{\bar\theta}$, is not, due to the I term. This makes the norm infinite, but the gradient is still well defined.
- To see this, note that
$$J(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho + \frac{1}{2}\|M^{\mathrm{tar}}\|_\rho^2.$$
$J(\theta)$ has the same minima as $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$! Namely, they differ by an "infinite" constant.

Forward TD with Function Approximation: Proof [1/3]

We now look at the proof of the theorem. Recall that $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')^2]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$. Also let $\langle M_1, M_2\rangle_\rho = \mathbb{E}_{s,s'\sim\rho}[f_1(s,s')f_2(s,s')]$.

- Recall that

  $$J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2 \quad \text{and} \quad M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar\theta}$$

  for some fixed $\bar\theta$.
- M_θ is absolutely continuous with respect to ρ while $M_{\bar\theta}$, is not, due to the I term. This makes the norm infinite, but the gradient is still well defined.
- To see this, note that

  $$J(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho + \frac{1}{2}\|M^{\mathrm{tar}}\|_\rho^2.$$

  $J(\theta)$ has the same minima as $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$! Namely, they differ by an "infinite" constant.

Forward TD with Function Approximation: Proof [1/3]

We now look at the proof of the theorem. Recall that $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')^2]$ with $f(s,s') \coloneqq M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$. Also let $\langle M_1, M_2\rangle_\rho = \mathbb{E}_{s,s'\sim\rho}[f_1(s,s')f_2(s,s')]$.

- Recall that
$$J(\theta) \coloneqq \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2 \quad \text{and} \quad M^{\mathrm{tar}} = I + \gamma P^\pi M_{\bar\theta}$$
for some fixed $\bar\theta$.
- M_θ is absolutely continuous with respect to ρ while $M_{\bar\theta}$, is not, due to the I term. This makes the norm infinite, but the gradient is still well defined.
- To see this, note that
$$J(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho + \frac{1}{2}\|M^{\mathrm{tar}}\|_\rho^2.$$
$J(\theta)$ has the same minima as $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$! Namely, they differ by an "infinite" constant.

Forward TD with Function Approximation: Proof [1/3]

We now look at the proof of the theorem. Recall that $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')^2]$ with $f(s,s') \coloneqq M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$. Also let $\langle M_1, M_2\rangle_\rho = \mathbb{E}_{s,s'\sim\rho}[f_1(s,s')f_2(s,s')]$.

- Recall that
$$J(\theta) \coloneqq \frac{1}{2}\|M_\theta - M^{\text{tar}}\|_\rho^2 \quad \text{and} \quad M^{\text{tar}} = I + \gamma P^\pi M_{\bar\theta}$$
for some fixed $\bar\theta$.
- M_θ is absolutely continuous with respect to ρ while $M_{\bar\theta}$, is not, due to the I term. This makes the norm infinite, but the gradient is still well defined.
- To see this, note that
$$J(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho + \frac{1}{2}\|M^{\text{tar}}\|_\rho^2.$$
$J(\theta)$ has the same minima as $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$! Namely, they differ by an "infinite" constant.

Forward TD with Function Approximation: Proof [1/3]

We now look at the proof of the theorem. Recall that $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[f(s,s')^2]$ with $f(s,s') := M(s,\mathrm{d}s')/\rho(\mathrm{d}s')$. Also let $\langle M_1, M_2\rangle_\rho = \mathbb{E}_{s,s'\sim\rho}[f_1(s,s')f_2(s,s')]$.

- Recall that
$$J(\theta) := \frac{1}{2}\|M_\theta - M^{\text{tar}}\|_\rho^2 \quad \text{and} \quad M^{\text{tar}} = I + \gamma P^\pi M_{\bar\theta}$$
for some fixed $\bar\theta$.
- M_θ is absolutely continuous with respect to ρ while $M_{\bar\theta}$, is not, due to the I term. This makes the norm infinite, but the gradient is still well defined.
- To see this, note that
$$J(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho + \frac{1}{2}\|M^{\text{tar}}\|_\rho^2.$$
$J(\theta)$ has the same minima as $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$! Namely, they differ by an "infinite" constant.

Forward TD with Function Approximation: Proof [2/3]

We work with $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$.

Next, recall that $M_\theta(s, \mathrm{d}s_2) = m_\theta(s, s_2)\rho(\mathrm{d}s_2)$, while M^{tar} is

$$M^{\text{tar}}(s, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \underbrace{P_s^\pi M_{\bar{\theta}}(\cdot, \mathrm{d}s_2)}_{P \text{ is like an integral operator: next state transition}},$$

$$= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} M_{\bar{\theta}}(s', \mathrm{d}s_2) P^\pi(\mathrm{d}s'|s),$$

$$= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} m_{\bar{\theta}}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s).$$

Therefore

$$\langle M_\theta, M^{\text{tar}}\rangle_\rho = \int_{s,s_2} \frac{M_\theta(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \frac{M^{\text{tar}}(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \int_{s,s_2} m_\theta(s, s_2) M^{\text{tar}}(s, \mathrm{d}s_2)\rho(\mathrm{d}s),$$

$$= \int_s m_\theta(s, s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s, s_2) m_{\bar{\theta}}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s).$$

Forward TD with Function Approximation: Proof [2/3]

We work with $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$.

Next, recall that $M_\theta(s, \mathrm{d}s_2) = m_\theta(s, s_2)\rho(\mathrm{d}s_2)$, while M^{tar} is

$$
\begin{aligned}
M^{\text{tar}}(s, \mathrm{d}s_2) &= \delta_s(\mathrm{d}s_2) + \gamma \underbrace{P_s^\pi M_{\bar\theta}(\cdot, \mathrm{d}s_2)}_{P\text{ is like an integral operator: next state transition}}, \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} M_{\bar\theta}(s', \mathrm{d}s_2) P^\pi(\mathrm{d}s'|s), \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\langle M_\theta, M^{\text{tar}}\rangle_\rho &= \int_{s,s_2} \frac{M_\theta(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \frac{M^{\text{tar}}(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \int_{s,s_2} m_\theta(s, s_2) M^{\text{tar}}(s, \mathrm{d}s_2)\rho(\mathrm{d}s), \\
&= \int_s m_\theta(s, s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s, s_2) m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s).
\end{aligned}
$$

Forward TD with Function Approximation: Proof [2/3]

We work with $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$.

Next, recall that $M_\theta(s, \mathrm{d}s_2) = m_\theta(s, s_2)\rho(\mathrm{d}s_2)$, while M^{tar} is

$$
\begin{aligned}
M^{\mathrm{tar}}(s, \mathrm{d}s_2) &= \delta_s(\mathrm{d}s_2) + \gamma \underbrace{P_s^\pi M_{\bar\theta}(\cdot, \mathrm{d}s_2)}_{P \text{ is like an integral operator: next state transition}}, \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} M_{\bar\theta}(s', \mathrm{d}s_2) P^\pi(\mathrm{d}s'|s), \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\langle M_\theta, M^{\mathrm{tar}}\rangle_\rho &= \int_{s,s_2} \frac{M_\theta(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \frac{M^{\mathrm{tar}}(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \int_{s,s_2} m_\theta(s, s_2) M^{\mathrm{tar}}(s, \mathrm{d}s_2)\rho(\mathrm{d}s), \\
&= \int_s m_\theta(s, s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s, s_2) m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s).
\end{aligned}
$$

Forward TD with Function Approximation: Proof [2/3]

We work with $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$.

Next, recall that $M_\theta(s, \mathrm{d}s_2) = m_\theta(s, s_2)\rho(\mathrm{d}s_2)$, while M^{tar} is

$$
\begin{aligned}
M^{\mathrm{tar}}(s, \mathrm{d}s_2) &= \delta_s(\mathrm{d}s_2) + \gamma \underbrace{P^\pi_s M_{\bar\theta}(\cdot, \mathrm{d}s_2)}_{P \text{ is like an integral operator: next state transition}}, \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} M_{\bar\theta}(s', \mathrm{d}s_2) P^\pi(\mathrm{d}s'|s), \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\langle M_\theta, M^{\mathrm{tar}}\rangle_\rho &= \int_{s,s_2} \frac{M_\theta(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \frac{M^{\mathrm{tar}}(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \int_{s,s_2} m_\theta(s, s_2) M^{\mathrm{tar}}(s, \mathrm{d}s_2)\rho(\mathrm{d}s), \\
&= \int_s m_\theta(s, s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s, s_2) m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s).
\end{aligned}
$$

Forward TD with Function Approximation: Proof [2/3]

We work with $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$.

Next, recall that $M_\theta(s, \mathrm{d}s_2) = m_\theta(s, s_2)\rho(\mathrm{d}s_2)$, while M^{tar} is

$$
\begin{aligned}
M^{\text{tar}}(s, \mathrm{d}s_2) &= \delta_s(\mathrm{d}s_2) + \gamma \underbrace{P_s^\pi M_{\bar\theta}(\cdot, \mathrm{d}s_2)}_{P \text{ is like an integral operator: next state transition}}, \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} M_{\bar\theta}(s', \mathrm{d}s_2) P^\pi(\mathrm{d}s'|s), \\
&= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\langle M_\theta, M^{\text{tar}}\rangle_\rho &= \int_{s,s_2} \frac{M_\theta(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \frac{M^{\text{tar}}(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \int_{s,s_2} m_\theta(s, s_2) M^{\text{tar}}(s, \mathrm{d}s_2)\rho(\mathrm{d}s), \\
&= \int_s m_\theta(s, s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s, s_2) m_{\bar\theta}(s', s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s).
\end{aligned}
$$

Forward TD with Function Approximation: Proof [3/3]

1. $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$
2. $\langle M_\theta, M^{\text{tar}}\rangle_\rho = \int_s m_\theta(s,s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s,s_2) m_{\bar\theta}(s',s_2)\rho(\mathrm{d}s_2)P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s)$.

We also have

$$\|M_\theta\|_\rho^2 = \int_{s,s_2} m_\theta(s,s_2)^2 \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \mathbb{E}_{s,s_2\sim\rho}[m_\theta(s,s_2)^2]$$

Hence

$$\begin{aligned}\partial_\theta J'(\theta) &= \mathbb{E}_{s,s_2\sim\rho, s'\sim P^\pi(\cdot|s)}\left[m_\theta(s,s_2)\partial_\theta m_\theta(s,s_2) - \partial_\theta m_\theta(s,s) - \gamma\partial_\theta m_\theta(s,s_2) m_{\bar\theta}(s',s_2)\right], \\ &= -\mathbb{E}_{s\sim\rho}\left[\partial_\theta m_\theta(s,s)\right] + \mathbb{E}_{s,s_2\sim\rho,s'\sim P^\pi(\cdot|s)}\left[\partial_\theta m_\theta(s,s_2)\left(m_\theta(s,s_2) - \gamma m_{\bar\theta}(s',s_2)\right)\right]. \square\end{aligned}$$

Forward TD with Function Approximation: Proof [3/3]

1. $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$
2. $\langle M_\theta, M^{\text{tar}}\rangle_\rho = \int_s m_\theta(s,s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s,s_2) m_{\bar\theta}(s',s_2)\rho(\mathrm{d}s_2)P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s)$.

We also have

$$\|M_\theta\|_\rho^2 = \int_{s,s_2} m_\theta(s,s_2)^2 \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \mathbb{E}_{s,s_2\sim\rho}[m_\theta(s,s_2)^2]$$

Hence

$$\begin{aligned}\partial_\theta J'(\theta) &= \mathbb{E}_{s,s_2\sim\rho,s'\sim P^\pi(\cdot|s)}\left[m_\theta(s,s_2)\partial_\theta m_\theta(s,s_2) - \partial_\theta m_\theta(s,s) - \gamma\partial_\theta m_\theta(s,s_2)m_{\bar\theta}(s',s_2)\right], \\ &= -\mathbb{E}_{s\sim\rho}\left[\partial_\theta m_\theta(s,s)\right] + \mathbb{E}_{s,s_2\sim\rho,s'\sim P^\pi(\cdot|s)}\left[\partial_\theta m_\theta(s,s_2)\left(m_\theta(s,s_2) - \gamma m_{\bar\theta}(s',s_2)\right)\right]. \square\end{aligned}$$

Forward TD with Function Approximation: Proof [3/3]

1. $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\text{tar}}\rangle_\rho$
2. $\langle M_\theta, M^{\text{tar}}\rangle_\rho = \int_s m_\theta(s,s)\rho(\mathrm{d}s) + \gamma \int_{s,s_2,s'} m_\theta(s,s_2) m_{\bar\theta}(s',s_2)\rho(\mathrm{d}s_2) P^\pi(\mathrm{d}s'|s)\rho(\mathrm{d}s)$.

We also have

$$\|M_\theta\|_\rho^2 = \int_{s,s_2} m_\theta(s,s_2)^2 \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) = \mathbb{E}_{s,s_2\sim\rho}[m_\theta(s,s_2)^2]$$

Hence

$$\begin{aligned}\partial_\theta J'(\theta) &= \mathbb{E}_{s,s_2\sim\rho, s'\sim P^\pi(\cdot|s)}\left[m_\theta(s,s_2)\partial_\theta m_\theta(s,s_2) - \partial_\theta m_\theta(s,s) - \gamma \partial_\theta m_\theta(s,s_2) m_{\bar\theta}(s',s_2)\right], \\ &= -\mathbb{E}_{s\sim\rho}\left[\partial_\theta m_\theta(s,s)\right] + \mathbb{E}_{s,s_2\sim\rho, s'\sim P^\pi(\cdot|s)}\left[\partial_\theta m_\theta(s,s_2)\left(m_\theta(s,s_2) - \gamma m_{\bar\theta}(s',s_2)\right)\right]. \square\end{aligned}$$

Forward TD with Function Approximation

Theorem (TD Update for Successor States with Approximation)

Consider the model $M_\theta(s, \mathrm{d}s') = m_\theta(s, s')\rho(\mathrm{d}s')$*. For the loss* $J(\theta) := \frac{1}{2}\|M_\theta - M^{\mathrm{tar}}\|_\rho^2$*, the gradient of* $J(\theta)$ *is*

$$-\partial_\theta J(\theta) = \mathbb{E}_{s, s_2 \sim \rho, s' \sim P^\pi(s, \mathrm{d}s')} \left[\underbrace{\partial_\theta m_\theta(s, s)}_{\text{Singular term}} + \underbrace{\partial_\theta m_\theta(s, s_2)(\gamma m_{\bar{\theta}}(s', s_2) - m_\theta(s, s_2))}_{\text{Nxt step error}} \right]$$

where ρ *is a sampling distribution over states (e.g. stationary distribution, stationary buffer).*

Matrix Factorization: The Forward-Backward Representation

Matrix Factorization: Forward-Backward (FB) Representation

Even with function approximation, learning $m_\theta(s, s')$ for all pairs can be challenging. A useful approach is to restrict M to a low-rank form:

Definition (Forward-Backward Factorization)

We approximate the successor operator by a rank-r factorization:

$$M(s, \mathrm{d}s') = F(s)^\top B(s')\rho(\mathrm{d}s'),$$

where $F : S \to \mathbb{R}^r$ and $B : S \to \mathbb{R}^r$ are learnable feature vectors (with parameters θ_F, θ_B). In matrix form, $M \approx F\,B^\top$. We call $F(s)$ the *forward representation* of state s and $B(s')$ the *backward representation* of state s'.

Intuitively, $F(s)$ encodes the long-term dynamics starting from s, while $B(s')$ encodes how reachable state s' is (acting like a representation of "goals").

Matrix Factorization: Forward-Backward (FB) Representation

Even with function approximation, learning $m_\theta(s, s')$ for all pairs can be challenging. A useful approach is to restrict M to a low-rank form:

Definition (Forward-Backward Factorization)

We approximate the successor operator by a rank-r factorization:

$$M(s, \mathrm{d}s') = F(s)^\top B(s')\rho(\mathrm{d}s'),$$

where $F : S \to \mathbb{R}^r$ and $B : S \to \mathbb{R}^r$ are learnable feature vectors (with parameters θ_F, θ_B). In matrix form, $M \approx F\,B^\top$. We call $F(s)$ the *forward representation* of state s and $B(s')$ the *backward representation* of state s'.

Intuitively, $F(s)$ encodes the long-term dynamics starting from s, while $B(s')$ encodes how reachable state s' is (acting like a representation of "goals").

Matrix Factorization: Forward-Backward (FB) Representation

Even with function approximation, learning $m_\theta(s, s')$ for all pairs can be challenging. A useful approach is to restrict M to a low-rank form:

Definition (Forward-Backward Factorization)

We approximate the successor operator by a rank-r factorization:

$$M(s, \mathrm{d}s') = F(s)^\top B(s')\rho(\mathrm{d}s'),$$

where $F : S \to \mathbb{R}^r$ and $B : S \to \mathbb{R}^r$ are learnable feature vectors (with parameters θ_F, θ_B). In matrix form, $M \approx F\, B^\top$. We call $F(s)$ the *forward representation* of state s and $B(s')$ the *backward representation* of state s'.

Intuitively, $F(s)$ encodes the long-term dynamics starting from s, while $B(s')$ encodes how reachable state s' is (acting like a representation of "goals").

Why Low Rank?

Why the matrix $M = (I - \gamma P^\pi)^{-1}$ should be low rank?

1. Denote by s_t the state at time t. In many systems we have that $s_{t+\Delta} \approx s_t$ for Δ sufficiently small.
2. Example: **dynamical systems**

$$\frac{dx}{dt} = Ax(t) \Rightarrow x(t+\Delta) = e^{A\Delta}x(t) = (I + A\Delta + \dots)x(t).$$

3. For **continuous-time operators associated with random diffusions**, $I - P^\pi$ has few small and many large eigenvalues.
4. For such property to hold we must have $P \approx I + X^{-1}$, where X^{-1} is like a second-order term ($\approx$ low rank).
5. Results that use P to be low rank usually do not work well in practice (they miss the identity term!)

Why Low Rank?

Why the matrix $M = (I - \gamma P^\pi)^{-1}$ should be low rank?

1. Denote by s_t the state at time t. In many systems we have that $s_{t+\Delta} \approx s_t$ for Δ sufficiently small.
2. Example: **dynamical systems**

$$\frac{dx}{dt} = Ax(t) \Rightarrow x(t+\Delta) = e^{A\Delta}x(t) = (I + A\Delta + \dots)x(t).$$

3. For **continuous-time operators associated with random diffusions**, $I - P^\pi$ has few small and many large eigenvalues.
4. For such property to hold we must have $P \approx I + X^{-1}$, where X^{-1} is like a second-order term ($\approx$ low rank).
5. Results that use P to be low rank usually do not work well in practice (they miss the identity term!)

Why Low Rank?

Why the matrix $M = (I - \gamma P^\pi)^{-1}$ should be low rank?

1. Denote by s_t the state at time t. In many systems we have that $s_{t+\Delta} \approx s_t$ for Δ sufficiently small.
2. Example: **dynamical systems**

$$\frac{dx}{dt} = Ax(t) \Rightarrow x(t+\Delta) = e^{A\Delta}x(t) = (I + A\Delta + \dots)x(t).$$

3. For **continuous-time operators associated with random diffusions**, $I - P^\pi$ has few small and many large eigenvalues.
4. For such property to hold we must have $P \approx I + X^{-1}$, where X^{-1} is like a second-order term ($\approx$ low rank).
5. Results that use P to be low rank usually do not work well in practice (they miss the identity term!)

Why Low Rank?

Why the matrix $M = (I - \gamma P^\pi)^{-1}$ should be low rank?

1. Denote by s_t the state at time t. In many systems we have that $s_{t+\Delta} \approx s_t$ for Δ sufficiently small.
2. Example: **dynamical systems**

$$\frac{dx}{dt} = Ax(t) \Rightarrow x(t+\Delta) = e^{A\Delta}x(t) = (I + A\Delta + \dots)x(t).$$

3. For **continuous-time operators associated with random diffusions**, $I - P^\pi$ has few small and many large eigenvalues.
4. For such property to hold we must have $P \approx I + X^{-1}$, where X^{-1} is like a second-order term ($\approx$ low rank).
5. Results that use P to be low rank usually do not work well in practice (they miss the identity term!)

Connections with SVD

Is there some connection with **SVD**? Let $M_{\text{approx}} = F^\top B\rho$.

$$
\begin{aligned}
J &= \frac{1}{2}\|M_{\text{approx}}\|_\rho^2 - \langle M_{\text{approx}}, M\rangle_\rho, \\
&\approx \frac{1}{2}\|M_{\text{approx}} - M\|_\rho^2, \\
&= \frac{1}{2}\mathbb{E}_{s,s'\sim\rho}\left[\left(F(s,s')^\top B(s,s') - \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right)^2\right].
\end{aligned}
$$

We are looking for the d-rank best approximation of M/ρ (SVD in $L^2(\rho)$). See Eckart–Young–Mirsky theorem.

Connections with SVD

Is there some connection with **SVD**? Let $M_{\text{approx}} = F^\top B\rho$.

$$
\begin{aligned}
J &= \frac{1}{2}\|M_{\text{approx}}\|_\rho^2 - \langle M_{\text{approx}}, M\rangle_\rho, \\
&\approx \frac{1}{2}\|M_{\text{approx}} - M\|_\rho^2, \\
&= \frac{1}{2}\mathbb{E}_{s,s'\sim\rho}\left[\left(F(s,s')^\top B(s,s') - \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right)^2\right].
\end{aligned}
$$

We are looking for the d-rank best approximation of M/ρ (SVD in $L^2(\rho)$). See Eckart–Young–Mirsky theorem.

Connections with SVD

Is there some connection with **SVD**? Let $M_{\text{approx}} = F^\top B\rho$.

$$J = \frac{1}{2}\|M_{\text{approx}}\|_\rho^2 - \langle M_{\text{approx}}, M\rangle_\rho,$$
$$\approx \frac{1}{2}\|M_{\text{approx}} - M\|_\rho^2,$$
$$= \frac{1}{2}\mathbb{E}_{s,s'\sim\rho}\left[\left(F(s,s')^\top B(s,s') - \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right)^2\right].$$

We are looking for the d-rank best approximation of M/ρ (SVD in $L^2(\rho)$). See Eckart–Young–Mirsky theorem.

Connections with SVD

Is there some connection with **SVD**? Let $M_{\text{approx}} = F^\top B\rho$.

$$
\begin{aligned}
J &= \frac{1}{2}\|M_{\text{approx}}\|_\rho^2 - \langle M_{\text{approx}}, M\rangle_\rho, \\
&\approx \frac{1}{2}\|M_{\text{approx}} - M\|_\rho^2, \\
&= \frac{1}{2}\mathbb{E}_{s,s'\sim\rho}\left[\left(F(s,s')^\top B(s,s') - \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right)^2\right].
\end{aligned}
$$

We are looking for the d-rank best approximation of M/ρ (SVD in $L^2(\rho)$). See Eckart–Young–Mirsky theorem.

Connections with SVD

Is there some connection with **SVD**? Let $M_{\text{approx}} = F^\top B\rho$.

$$
\begin{aligned}
J &= \frac{1}{2}\|M_{\text{approx}}\|_\rho^2 - \langle M_{\text{approx}}, M\rangle_\rho, \\
&\approx \frac{1}{2}\|M_{\text{approx}} - M\|_\rho^2, \\
&= \frac{1}{2}\mathbb{E}_{s,s'\sim\rho}\left[\left(F(s,s')^\top B(s,s') - \frac{M(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right)^2\right].
\end{aligned}
$$

We are looking for the d-rank best approximation of M/ρ (SVD in $L^2(\rho)$). See Eckart–Young–Mirsky theorem.

Advantages of the FB Representation

- **Direct value estimation:** F and B together allow immediate computation of the value for any reward function. For example, if $R(s)$ is a reward function

$$V(R)(s) \approx F(s)^\top B(R),$$

  where $B(R) := \mathbb{E}_{s\sim\rho}[\,R(s)\,B(s)\,]$. Thus, we get a value function estimate at *every* state without explicitly learning a separate value network.
- **Generalization across states:** The low-rank assumption provides a form of regularization or prior: states that have similar long-term dynamics will learn similar F and B representations.
- **Implicit second-order effects:** The forward-backward model tends to prioritize learning the major dynamical modes of the Markov chain (eigenvectors corresponding to large eigenvalues of P).

Advantages of the FB Representation

- **Direct value estimation:** F and B together allow immediate computation of the value for any reward function. For example, if $R(s)$ is a reward function

$$V(R)(s) \approx F(s)^\top B(R),$$

where $B(R) := \mathbb{E}_{s\sim\rho}[\, R(s)\, B(s)\,]$. Thus, we get a value function estimate at *every* state without explicitly learning a separate value network.

- **Generalization across states:** The low-rank assumption provides a form of regularization or prior: states that have similar long-term dynamics will learn similar F and B representations.
- **Implicit second-order effects:** The forward-backward model tends to prioritize learning the major dynamical modes of the Markov chain (eigenvectors corresponding to large eigenvalues of P).

Advantages of the FB Representation

- **Direct value estimation:** F and B together allow immediate computation of the value for any reward function. For example, if $R(s)$ is a reward function

$$V(R)(s) \approx F(s)^\top B(R),$$

where $B(R) := \mathbb{E}_{s\sim\rho}[\,R(s)\,B(s)\,]$. Thus, we get a value function estimate at *every* state without explicitly learning a separate value network.

- **Generalization across states:** The low-rank assumption provides a form of regularization or prior: states that have similar long-term dynamics will learn similar F and B representations.
- **Implicit second-order effects:** The forward-backward model tends to prioritize learning the major dynamical modes of the Markov chain (eigenvectors corresponding to large eigenvalues of P).

Shortcomings of the FB Representation

- **Limited capacity (rank-r approximation):** By constraining M to rank r, we cannot perfectly represent the true M. Important dynamics corresponding to smaller singular values of M may be neglected.
- This means fine-grained or rapidly changing aspects of the reward structure (e.g. very localized rewards that vary quickly from state to state) might be smoothed out or underrepresented. The FB model tends to focus on the dominant, "long-range" structures in the state space.
- **Mitigation:** One can combine the low-rank SSR with a standard value function to capture the residual. For example, use $V(s) = F(s)^\top B(R) + v_\phi(s)$, where $v_\phi(s)$ is a separate value function learned for the specific reward.

Shortcomings of the FB Representation

- **Limited capacity (rank-r approximation):** By constraining M to rank r, we cannot perfectly represent the true M. Important dynamics corresponding to smaller singular values of M may be neglected.
- This means fine-grained or rapidly changing aspects of the reward structure (e.g. very localized rewards that vary quickly from state to state) might be smoothed out or underrepresented. The FB model tends to focus on the dominant, "long-range" structures in the state space.
- **Mitigation:** One can combine the low-rank SSR with a standard value function to capture the residual. For example, use $V(s) = F(s)^\top B(R) + v_\phi(s)$, where $v_\phi(s)$ is a separate value function learned for the specific reward.

Shortcomings of the FB Representation

- **Limited capacity (rank-r approximation):** By constraining M to rank r, we cannot perfectly represent the true M. Important dynamics corresponding to smaller singular values of M may be neglected.
- This means fine-grained or rapidly changing aspects of the reward structure (e.g. very localized rewards that vary quickly from state to state) might be smoothed out or underrepresented. The FB model tends to focus on the dominant, "long-range" structures in the state space.
- **Mitigation:** One can combine the low-rank SSR with a standard value function to capture the residual. For example, use $V(s) = F(s)^\top B(R) + v_\phi(s)$, where $v_\phi(s)$ is a separate value function learned for the specific reward.

Deep Reward-Free RL

Forward-Backward for Deep-RL

Starting point

$$M^{\pi}(s,a,\mathrm{d}s') = F(s,a)^{\top}B(s')\rho(\mathrm{d}s').$$

- Consider a family of policies $\{\pi_z\}$ parameterized by a vector $z \in \mathbb{R}^d$.
- Assume that for all z, we can find (F_z, B) (F is parameterized by z) such that

$$M^{z}(s,a,\mathrm{d}s') = F_z^{\top}(s,a)B(s')\rho(\mathrm{d}s')$$

- Then, if π_z induces a distribution ρ

$$\begin{aligned} Q^{z}(s,a) &= \mathbb{E}^{\pi_z}\left[\sum_{t\geq 1}\gamma^{t-1}r_t | s_1 = s, a_1 = as\right], \\ &= \left[(I-\gamma P^{\pi_z})^{-1}r\right](s,a), \\ &= \left[M^{z}r\right](s,a) = \int_{s'} F_z^{\top}(s,a)B(s')r(s')\rho(\mathrm{d}s'). \end{aligned}$$

Forward-Backward for Deep-RL

Starting point

$$M^{\pi}(s,a,\mathrm{d}s') = F(s,a)^{\top} B(s')\rho(\mathrm{d}s').$$

- Consider a family of policies $\{\pi_z\}$ parameterized by a vector $z \in \mathbb{R}^d$.
- Assume that for all z, we can find (F_z, B) (F is parameterized by z) such that

$$M^{z}(s,a,\mathrm{d}s') = F_z^{\top}(s,a) B(s')\rho(\mathrm{d}s')$$

- Then, if π_z induces a distribution ρ

$$\begin{aligned} Q^z(s,a) &= \mathbb{E}^{\pi_z}\left[\sum_{t\geq 1} \gamma^{t-1} r_t | s_1 = s, a_1 = as\right], \\ &= \left[(I - \gamma P^{\pi_z})^{-1} r\right](s,a), \\ &= \left[M^z r\right](s,a) = \int_{s'} F_z^{\top}(s,a) B(s') r(s') \rho(\mathrm{d}s'). \end{aligned}$$

Forward-Backward for Deep-RL

Starting point

$$M^{\pi}(s,a,\mathrm{d}s') = F(s,a)^{\top}B(s')\rho(\mathrm{d}s').$$

- Consider a family of policies $\{\pi_z\}$ parameterized by a vector $z \in \mathbb{R}^d$.
- Assume that for all z, we can find (F_z, B) (F is parameterized by z) such that

$$M^{z}(s,a,\mathrm{d}s') = F_z^{\top}(s,a)B(s')\rho(\mathrm{d}s')$$

- Then, if π_z induces a distribution ρ

$$\begin{aligned} Q^z(s,a) &= \mathbb{E}^{\pi_z}\left[\sum_{t\geq 1}\gamma^{t-1}r_t | s_1 = s, a_1 = as\right], \\ &= \left[(I-\gamma P^{\pi_z})^{-1}r\right](s,a), \\ &= \left[M^z r\right](s,a) = \int_{s'} F_z^{\top}(s,a)B(s')r(s')\rho(\mathrm{d}s'). \end{aligned}$$

Forward-Backward for Deep-RL

Starting point

$$M^\pi(s,a,\mathrm{d}s') = F(s,a)^\top B(s')\rho(\mathrm{d}s').$$

- Consider a family of policies $\{\pi_z\}$ parameterized by a vector $z \in \mathbb{R}^d$.
- Assume that for all z, we can find (F_z, B) (F is parameterized by z) such that

$$M^z(s,a,\mathrm{d}s') = F_z^\top(s,a)B(s')\rho(\mathrm{d}s')$$

- Then, if π_z induces a distribution ρ

$$\begin{aligned} Q^z(s,a) &= \mathbb{E}^{\pi_z}\left[\sum_{t\geq 1}\gamma^{t-1}r_t | s_1 = s, a_1 = as\right], \\ &= \left[(I-\gamma P^{\pi_z})^{-1}r\right](s,a), \\ &= \left[M^z r\right](s,a) = \int_{s'} F_z^\top(s,a)B(s')r(s')\rho(\mathrm{d}s'). \end{aligned}$$

Forward-Backward for Deep-RL

Starting point

$$M^\pi(s,a,\mathrm{d}s') = F(s,a)^\top B(s')\rho(\mathrm{d}s').$$

- Consider a family of policies $\{\pi_z\}$ parameterized by a vector $z \in \mathbb{R}^d$.
- Assume that for all z, we can find (F_z, B) (F is parameterized by z) such that

$$M^z(s,a,\mathrm{d}s') = F_z^\top(s,a)B(s')\rho(\mathrm{d}s')$$

- Then, if π_z induces a distribution ρ

$$\begin{aligned} Q^z(s,a) &= \mathbb{E}^{\pi_z}\left[\sum_{t\geq 1}\gamma^{t-1}r_t | s_1 = s, a_1 = as\right], \\ &= \left[(I-\gamma P^{\pi_z})^{-1}r\right](s,a), \\ &= [M^z r](s,a) = \int_{s'} F_z^\top(s,a)B(s')r(s')\rho(\mathrm{d}s'). \end{aligned}$$

Forward-Backward for Deep-RL

$$Q^z(s,a) = \int_{s'} F_z^\top(s,a)B(s')r(s')\rho(\mathrm{d}s').$$

Let

$$z_r = \mathbb{E}_{s\sim\rho}\big[B(s)r(s)\big].$$

If we define π_z by

$$\pi_z(s) \;=\; \arg\max_a \big(F_z(s,a)^\top z\big),$$

then π_{z_r} is the optimal policy for reward r, since $Q^{z_r} = F_{z_r}^\top(s,a)z_r$.

Forward-Backward for Deep-RL

$$Q^z(s,a) = \int_{s'} F_z^\top(s,a) B(s') r(s') \rho(\mathrm{d}s').$$

Let

$$z_r = \mathbb{E}_{s\sim\rho}\left[B(s) r(s)\right].$$

If we define π_z by

$$\pi_z(s) \;=\; \arg\max_a \left(F_z(s,a)^\top z\right),$$

then π_{z_r} is the optimal policy for reward r, since $Q^{z_r} = F_{z_r}^\top(s,a) z_r$.

Forward-Backward for Deep-RL

$$Q^z(s,a) = \int_{s'} F_z^\top(s,a) B(s') r(s') \rho(\mathrm{d}s').$$

Let

$$z_r = \mathbb{E}_{s\sim\rho}\big[B(s) r(s)\big].$$

If we define π_z by

$$\pi_z(s) \;=\; \arg\max_a \big(F_z(s,a)^\top z\big),$$

then π_{z_r} is the optimal policy for reward r, since $Q^{z_r} = F_{z_r}^\top(s,a) z_r$.

Forward-Backward for Deep-RL

Let's slow down...

$z_r = \mathbb{E}_{s\sim\rho}\big[B(s)r(s)\big]$ is like a representation.

- If we think of the reward function r as a vector in some Hilbert space, then we are simply projecting r onto the space spanned by B
- For the method to make sense we need to have $\mathbb{E}_{s\sim\rho}[B(s)B(s)^\top]$ to be full rank

$$z = B^\top r \Rightarrow r = (BB^\top)^{-1}B^r$$

- The eigenvectors of $\mathbb{E}_{s\sim\rho}[B(s)B(s)^\top]$ define a basis in this space.

Goal

We need to learn F_z, B. **How?**

$$M^z(s, a, \mathrm{d}s') = F_z(s, a)^\top B(s')\rho(\mathrm{d}s').$$

- Remember the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[m(s, s')^2]$.
- **Goal**: find (F_z, B) such that, for all z, we have

$$M^z = I + \gamma P^{\pi_z} M^z$$

  and

$$\pi_z(s) \;=\; \arg\max_a \big(F_z(s, a)\big).$$

- **How**? TD-Learning on $J(\theta) = \|M_\theta^z - (I + \gamma P^{\pi_z} M_{\bar\theta}^z)\|_\rho^2$.

Goal

We need to learn F_z, B. **How?**

$$M^z(s, a, \mathrm{d}s') = F_z(s, a)^\top B(s')\rho(\mathrm{d}s').$$

- Remember the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[m(s, s')^2]$.
- **Goal**: find (F_z, B) such that, for all z, we have

$$M^z = I + \gamma P^{\pi_z} M^z$$

and

$$\pi_z(s) = \arg\max_a (F_z(s, a)).$$

- **How**? TD-Learning on $J(\theta) = \|M_\theta^z - (I + \gamma P^{\pi_z} M_{\bar{\theta}}^z)\|_\rho^2$.

Goal

We need to learn F_z, B. **How?**

$$M^z(s,a,\mathrm{d}s') = F_z(s,a)^\top B(s')\rho(\mathrm{d}s').$$

- Remember the norm $\|M\|_\rho^2 = \mathbb{E}_{s,s'\sim\rho}[m(s,s')^2]$.
- **Goal**: find (F_z, B) such that, for all z, we have

$$M^z = I + \gamma P^{\pi_z} M^z$$

  and

$$\pi_z(s) \;=\; \arg\max_a \big(F_z(s,a)\big).$$

- **How**? TD-Learning on $J(\theta) = \|M_\theta^z - (I + \gamma P^{\pi_z} M_{\bar{\theta}}^z)\|_\rho^2$.

Forward-backward method: training

- Parametrize F_z, B by θ. Let $m^z_\theta = (F^\theta_z)^\top B^\theta_z$, thus

$$M^z_\theta(s, a, \mathrm{d}s') = m^z_\theta(s, a, s')\rho(\mathrm{d}s').$$

- For $(\theta, \bar{\theta})$, define the learning objective:

$$J'(z; \theta) = \frac{1}{2}\|M^z_\theta\|^2_\rho - \langle M^z_\theta, M^{\mathrm{tar}}\rangle_\rho$$

where $M^{\mathrm{tar}} = I + \gamma P^{\pi_z} M^z_{\bar{\theta}}$.

Forward-backward method: training

- Parametrize F_z, B by θ. Let $m_\theta^z = (F_z^\theta)^\top B_z^\theta$, thus

$$M_\theta^z(s, a, \mathrm{d}s') = m_\theta^z(s, a, s')\rho(\mathrm{d}s').$$

- For $(\theta, \bar{\theta})$, define the learning objective:

$$J'(z; \theta) = \frac{1}{2}\|M_\theta^z\|_\rho^2 - \langle M_\theta^z, M^{\mathrm{tar}}\rangle_\rho$$

where $M^{\mathrm{tar}} = I + \gamma P^{\pi_z} M_{\bar{\theta}}^z$.

Gradient with replay buffer

Gradient step

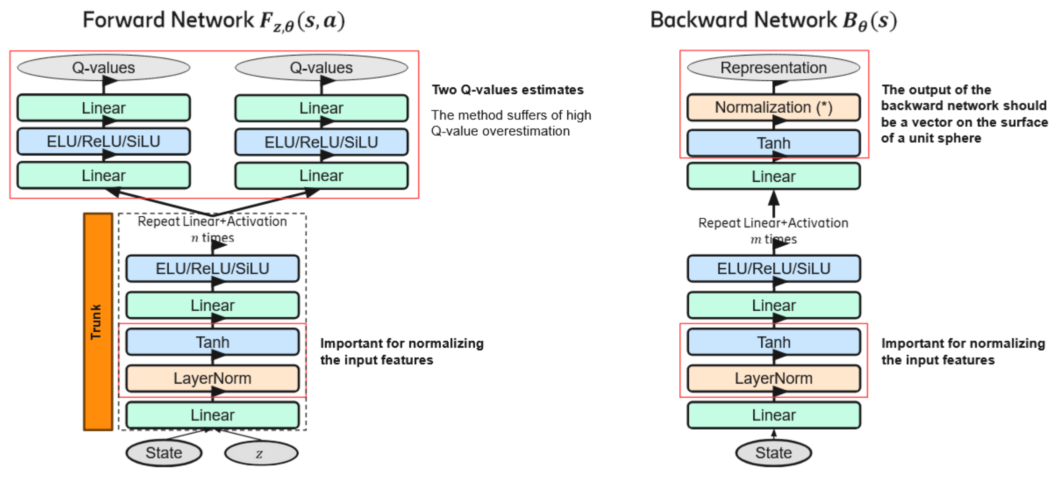
We know very well that $\partial_\theta J'(\theta) = \partial_\theta J(\theta)$, therefore

$$-\partial_\theta J(\theta) = \mathbb{E}_{(s,a,s')\sim D,(s_2,a_2)\sim D}\left[\partial_\theta m_\theta(s,a,s) + \partial_\theta m_\theta(s,a,s')(\gamma m_{\bar\theta}(s_2,a_2,s') - m_\theta(s,a,s'))\right]$$

where D is the data distribution induced by the policy and $m^z_\theta(s,a,s') = F^\theta_z(s,a)^\top B^\theta_z(s')$

- Practically speaking, D is the replay buffer. At every training step we sample two random independent samples from the replay buffer D.
- One contributes due to visiting s' from s.
- Another from a different random state s''.

Training procedure: networks



Training

Algorithm 1 Off-policy Training Procedure

```
Initialize θ, θ̄
while not converged do
    Sample z ∼ S^{d−1}                                  ▷ from the surface of a unit sphere
    Collect data using policy π_z and add to replay buffer D
    Sample batches (B, B′) ∼ D and compute gradient:
```

$$-\partial_\theta J(\theta) \approx \hat{\mathbb{E}}_{(s,a,s')\sim B,(s_2,a_2)\sim B'}\left[\partial_\theta m_\theta(s,a,s') + \partial_\theta m_\theta(s,a,s')\Big(\gamma\, m_\theta(s_2,a_2,s') - m_\theta(s,a,s')\Big)\right]$$

```
    where m^z_θ(s, a, s′) = F^z_θ(s, a)^⊤ B^z_θ(s′)
    Update parameters θ ← θ − η∂_θ J(θ) using the computed gradient
    if every N steps then
        Set θ̄ ← θ.
    end if
end while
```

Other tricks and technicalities

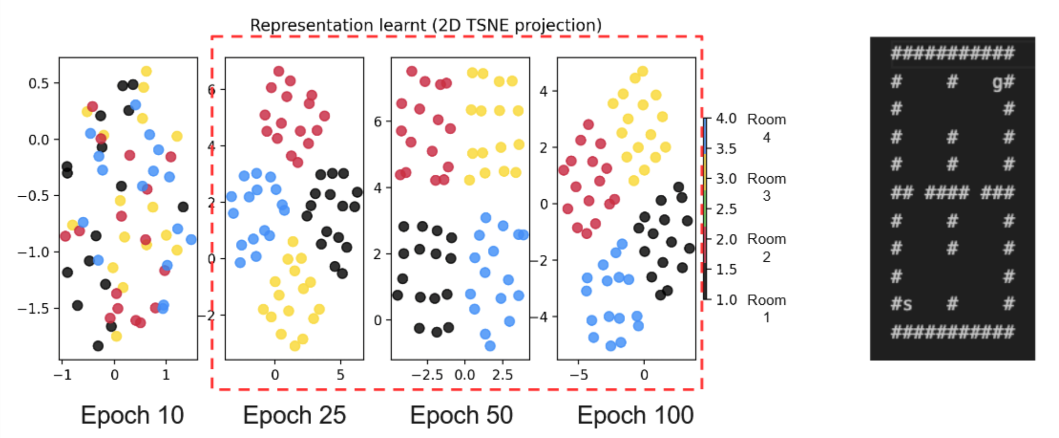
- We add a regularization term $\left\|E(B^\top B) - I\right\|_\rho^2$ to ensure $\operatorname{cov}(B) = I$.
- The computation of the target value can be numerically unstable (large gradients).
- One way to solve the issue: replace the greedy $\pi_z = \arg\max_a F_z(s,a)^\top z$ with a regularized version
$$\pi_z = \operatorname{softmax}\big(F_z(s,a)^\top z/\tau\big).$$
- Use a combination of linear + layer normalization + $\tanh$ layers to extract features.

Test time

How does the method work at test time?

- For a specific reward r we need to compute z.
- Recall $z = \mathbb{E}_{s\sim\rho}[B(s)r(s)]$.
- Sample a batch D from the buffer, and compute $z_r = \frac{1}{|D|}\sum_{s\in D} B(s)r(s)$.
- Obtain the policy $\pi_{z_r}(s) = \arg\max_a F_{z_r}(s)^\top z_r$.

4-Rooms Example



Random initial state/goal at every episode.

Conclusion

Conclusions

- **Takeaway:** Successor representations offer a powerful framework for goal-conditioned value learning and reward-free RL in a model-free fashion.
- The theoretical insights can guide the design of efficient RL algorithms in continuous state spaces.

Conclusions

- **Takeaway:** Successor representations offer a powerful framework for goal-conditioned value learning and reward-free RL in a model-free fashion.
- The theoretical insights can guide the design of efficient RL algorithms in continuous state spaces.

References i

- Léonard Blier, Corentin Tallec, and Yann Ollivier, *Learning successor states and goal-dependent values: A mathematical viewpoint*, arXiv preprint arXiv:2101.07123 (2021).
- Ahmed Touati and Yann Ollivier, *Learning one representation to optimize all rewards*, Advances in Neural Information Processing Systems **34** (2021), 13–23.
- Ahmed Touati, Jérémy Rapin, and Yann Ollivier, *Does zero-shot reinforcement learning exist?*, arXiv preprint arXiv:2209.14935 (2022).