Representation learning via metrics

Pablo Castro[1] Tyler Kastner[2,3] Prakash Panangaden [2,3]
Mark Rowland[4]

[1] Google Brain, Montreal
[2] McGill University
[3] Montreal Institute of Learning Algorithms (Mila)
[4] DeepMind, London

June 16, 2021

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Basic goals in RL

- We are often dealing with *large* or *infinite* transition systems whose behaviour is probabilistic.

Basic goals in RL

- We are often dealing with *large* or *infinite* transition systems whose behaviour is probabilistic.
- The system responds to stimuli (actions) and moves to a new state probabilistically and outputs a (possibly) random reward.

Basic goals in RL

- We are often dealing with *large* or *infinite* transition systems whose behaviour is probabilistic.
- The system responds to stimuli (actions) and moves to a new state probabilistically and outputs a (possibly) random reward.
- We seek optimal policies for extracting the largest possible reward in expectation.

Basic goals in RL

- We are often dealing with *large* or *infinite* transition systems whose behaviour is probabilistic.
- The system responds to stimuli (actions) and moves to a new state probabilistically and outputs a (possibly) random reward.
- We seek optimal policies for extracting the largest possible reward in expectation.
- A plethora of algorithms and techniques, but the cost depends on the size of the state space.

Basic goals in RL

- We are often dealing with *large* or *infinite* transition systems whose behaviour is probabilistic.
- The system responds to stimuli (actions) and moves to a new state probabilistically and outputs a (possibly) random reward.
- We seek optimal policies for extracting the largest possible reward in expectation.
- A plethora of algorithms and techniques, but the cost depends on the size of the state space.
- Can we *learn* representations of the state space that accelerate the learning process?

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- Immediate rewards.

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- Immediate rewards.
- What should be guaranteed?

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- Immediate rewards.
- What should be guaranteed?
- An equivalence relation on states so that if the equivalence classes are 'lumped' together we cannot tell that anything has changed.

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- Immediate rewards.
- What should be guaranteed?
- An equivalence relation on states so that if the equivalence classes are 'lumped' together we cannot tell that anything has changed.
- Ideally we assume exact equality of real numbers.

A bit of history

- Cantor and the back-and-forth argument

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960’s

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989
- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989
- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...
- Bisimulation metrics for Markov processes Desharnais, Gupta, Jagadeesan, P. 1999

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989
- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...
- Bisimulation metrics for Markov processes Desharnais, Gupta, Jagadeesan, P. 1999
- Fixed-point version: van Breugel and Worrell 2001

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989
- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...
- Bisimulation metrics for Markov processes Desharnais, Gupta, Jagadeesan, P. 1999
- Fixed-point version: van Breugel and Worrell 2001
- Bisimulation for MDP's : Givan and Dean 2003

A bit of history

- Cantor and the back-and-forth argument
- Lumpability in queueing theory 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation in probabilistic automata : Larsen and Skou 1989
- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...
- Bisimulation metrics for Markov processes Desharnais, Gupta, Jagadeesan, P. 1999
- Fixed-point version: van Breugel and Worrell 2001
- Bisimulation for MDP's : Givan and Dean 2003
- Bisimulation metrics for MDP's: Ferns, Precup, P. 2004

What are Markov decision processes?

- Markov decisionprocesses are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.

What are Markov decision processes?

- Markov decisionprocesses are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- There is a *reward* associated with each transition.

What are Markov decision processes?

- Markov decisionprocesses are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- There is a *reward* associated with each transition.
- We observe the interactions and the rewards - not the internal states.

Markov decision processes: formal definition

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

where

S : the state space, we will take it to be a finite set.

$\mathcal{A}$: the actions, a finite set

P^a : the transition function; $\mathcal{D}(S)$ denotes distributions over S

$\mathcal{R}$: the reward, could readily make it stochastic.

Will write $P^a(s, C)$ for $P^a(s)(C)$.

Policies

MDP

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

We control the choice of action; it is not some external scheduler.

Policies

MDP

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

We control the choice of action; it is not some external scheduler.

Policy

$$\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$$

Policies

MDP

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

We control the choice of action; it is not some external scheduler.

Policy

$$\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$$

Policies

MDP

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

We control the choice of action; it is not some external scheduler.

Policy

$$\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$$

The goal is **choose** the best policy. We do not know it in advance; we must **learn** it.

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.
- The **value function** $V^\pi : S \longrightarrow \mathbf{R}$ associated with the policy π is given by:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s)(a)[\mathcal{R}(s, a) + \gamma \sum_{s' \in S} P^a(s, s') V^\pi(s')]$$

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.
- The **value function** $V^\pi : S \longrightarrow \mathbf{R}$ associated with the policy π is given by:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s)(a)[\mathcal{R}(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V^\pi(s')]$$

- $\gamma \in (0,1)$ is a *contraction* factor.

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.
- The **value function** $V^\pi : S \longrightarrow \mathbf{R}$ associated with the policy π is given by:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s)(a)[\mathcal{R}(s, a) + \gamma \sum_{s' \in S} P^a(s, s')V^\pi(s')]$$

- $\gamma \in (0, 1)$ is a *contraction* factor.
- There is a version for the **optimal** value function V^*

$$V^*(s) = \max_{a \in \mathcal{A}}[\mathcal{R}(s, a) + \gamma \sum_{s' \in S} P^a(s, s')V^*(s')]$$

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.
- The **value function** $V^\pi : S \longrightarrow \mathbf{R}$ associated with the policy π is given by:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s)(a)[\mathcal{R}(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V^\pi(s')]$$

- $\gamma \in (0,1)$ is a *contraction* factor.
- There is a version for the **optimal** value function V^*

$$V^*(s) = \max_{a \in \mathcal{A}}[\mathcal{R}(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V^*(s')]$$

- we can extract a Bellman operator as $T^\pi(V) = \sum_{a \in \mathcal{A}} \pi(s)(a)[r(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V(s')]$

Bellman equations

- Given an MDP $(S, \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : S \times \mathcal{A} \longrightarrow \mathbf{R}^{\geq 0})$
- we define a **policy** $\pi : S \longrightarrow \mathcal{D}(\mathcal{A})$, a strategy for choosing an action in a state.
- The **value function** $V^\pi : S \longrightarrow \mathbf{R}$ associated with the policy π is given by:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s)(a)[\mathcal{R}(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V^\pi(s')]$$

- $\gamma \in (0,1)$ is a *contraction* factor.
- There is a version for the **optimal** value function V^*

$$V^*(s) = \max_{a \in \mathcal{A}} [\mathcal{R}(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V^*(s')]$$

- we can extract a Bellman operator as $T^\pi(V) = \sum_{a \in \mathcal{A}} \pi(s)(a)[r(s,a) + \gamma \sum_{s' \in S} P^a(s,s')V(s')]$
- $T^\pi(V^\pi) = V^\pi$.

Policy evaluation by iteration

- Given a policy π we have the associated Bellman operator T^{π} on the space of value functions.

Policy evaluation by iteration

- Given a policy π we have the associated Bellman operator T^{π} on the space of value functions.
- If V^{π} is the value function we write V_n for its nth iterate: $V_{n+1} = T^{\pi}(V_n)$.

Policy evaluation by iteration

- Given a policy π we have the associated Bellman operator T^π on the space of value functions.
- If V^π is the value function we write V_n for its nth iterate: $V_{n+1} = T^\pi(V_n)$.
- The Banach fixed-point theorem says that V_n converges to V^π.

Policy iteration

- Start with some policy π_0 and compute V^{π_0}

Policy iteration

- Start with some policy π_0 and compute V^{π_0}
- Inductive step: evaluate V^{π_n}, then set π_{n+1} to be equal to the greedy policy based on V^{π_n} and repeat.

Policy iteration

- Start with some policy π_0 and compute V^{π_0}
- Inductive step: evaluate V^{π_n}, then set π_{n+1} to be equal to the greedy policy based on V^{π_n} and repeat.
- This converges to π^* the optimal policy, *but not by the Banach fixed point theorem*.

- For large state spaces, learning value functions $S \times \mathcal{A} \longrightarrow \mathbf{R}$ is not feasible.

Representation learning

- For large state spaces, learning value functions $S \times \mathcal{A} \longrightarrow \mathbf{R}$ is not feasible.
- Instead we define a new space of *features* M and try to come up with an embedding $\phi : S \longrightarrow \mathbf{R}^M$.

Representation learning

- For large state spaces, learning value functions $S \times \mathcal{A} \longrightarrow \mathbf{R}$ is not feasible.
- Instead we define a new space of *features* M and try to come up with an embedding $\phi : S \longrightarrow \mathbf{R}^M$.
- Then we can try to use this to predict values associated with state,action pairs.

Representation learning

- For large state spaces, learning value functions $S \times \mathcal{A} \longrightarrow \mathbf{R}$ is not feasible.
- Instead we define a new space of *features* M and try to come up with an embedding $\phi : S \longrightarrow \mathbf{R}^M$.
- Then we can try to use this to predict values associated with state,action pairs.
- Representation learning means learning such a ϕ.

Representation learning

- For large state spaces, learning value functions $S \times \mathcal{A} \longrightarrow \mathbf{R}$ is not feasible.
- Instead we define a new space of *features* M and try to come up with an embedding $\phi : S \longrightarrow \mathbf{R}^M$.
- Then we can try to use this to predict values associated with state,action pairs.
- Representation learning means learning such a ϕ.
- The elements of M are the "features" that are chosen. They can be based on any kind of knowledge or experience about the task at hand.

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s\ R\ t$ if $(\forall\ a)$ and all equivalence classes C of R:

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s\ R\ t$ if $(\forall\ a)$ and all equivalence classes C of R:
  (i) $\mathcal{R}(a,s) = \mathcal{R}(a,t)$

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s \; R \; t$ if $(\forall \; a)$ and all equivalence classes C of R:
  (i) $\mathcal{R}(a, s) = \mathcal{R}(a, t)$
  (ii) $P^a(s, C) = P^a(t, C)$

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s\ R\ t$ if $(\forall\ a)$ and all equivalence classes C of R:
  (i) $\mathcal{R}(a, s) = \mathcal{R}(a, t)$
  (ii) $P^a(s, C) = P^a(t, C)$
- s, t are bisimilar if there is a bisimulation relation R with sRt them.

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s\ R\ t$ if $(\forall\ a)$ and all equivalence classes C of R:
  - (i) $\mathcal{R}(a,s) = \mathcal{R}(a,t)$
  - (ii) $P^a(s,C) = P^a(t,C)$
- s,t are bisimilar if there is a bisimulation relation R with sRt them.
- Basic pattern: immediate rewards match (initiation), stay related after the transition (coinduction).

Bisimulation

- Let R be an equivalence relation. R is a bisimulation if: $s\ R\ t$ if $(\forall\ a)$ and all equivalence classes C of R:
  (i) $\mathcal{R}(a,s) = \mathcal{R}(a,t)$
  (ii) $P^a(s,C) = P^a(t,C)$
- s,t are bisimilar if there is a bisimulation relation R with sRt them.
- Basic pattern: immediate rewards match (initiation), stay related after the transition (coinduction).
- Bisimulation can be defined as the *greatest fixed point* of a relation transformer.

A metric-based approximate viewpoint

- Move from equality between processes to distances between processes (Jou and Smolka 1990).

A metric-based approximate viewpoint

- Move from equality between processes to distances between processes (Jou and Smolka 1990).
- Quantitative measurement of the distinction between processes.

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that
  1. $\forall x \in X, d(x, x) = 0$

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that
  1. $\forall x \in X, d(x,x) = 0$
  2. $\forall x, y \in X, d(x,y) = d(y,x)$

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that
  1. $\forall x \in X, d(x,x) = 0$
  2. $\forall x, y \in X, d(x,y) = d(y,x)$
  3. $\forall x, y, z \in X, d(x,y) \leq d(x,z) + d(z,y)$

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that
  1. $\forall x \in X, d(x,x) = 0$
  2. $\forall x,y \in X, d(x,y) = d(y,x)$
  3. $\forall x,y,z \in X, d(x,y) \leq d(x,z) + d(z,y)$
  4. If $d(x,y) = 0$ implies $x = y$ we say that it is a *metric*

The basic setting: metric spaces

- A *pseudometric* on a set X is a function $d : X \times X \longrightarrow \mathbf{R}^{\geq 0}$ such that
  1. $\forall x \in X, d(x,x) = 0$
  2. $\forall x, y \in X, d(x,y) = d(y,x)$
  3. $\forall x, y, z \in X, d(x,y) \leq d(x,z) + d(z,y)$
  4. If $d(x,y) = 0$ implies $x = y$ we say that it is a *metric*

The setup

A set M equipped with a **metric** d obeying the above axioms (unlike, for example, KL-divergence which is **not** a metric). A metric space is **complete** if every Cauchy sequence has a limit point to which it converges.

The setup

- We will assume that we have an underlying metric space—the state space—and we are looking at probability distributions on top of this space.

The setup

- We will assume that we have an underlying metric space—the state space—and we are looking at probability distributions on top of this space.
- We will then look at ways to define a metric on the space of probability distributions.

The setup

- We will assume that we have an underlying metric space—the state space—and we are looking at probability distributions on top of this space.
- We will then look at ways to define a metric on the space of probability distributions.
- It should be, somehow, related to the metric of the underlying space.

The setup

- We will assume that we have an underlying metric space—the state space—and we are looking at probability distributions on top of this space.
- We will then look at ways to define a metric on the space of probability distributions.
- It should be, somehow, related to the metric of the underlying space.
- I will elide all measure theory issues in this discussion, but they are there, and one cannot really work on this topic without knowing basic measure theory on metric spaces.

The Kantorovich metric

- What is the observable aspect of a probability distribution?

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.

The Kantorovitch metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P, Q) = \sup_{f \in ??} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$

The Kantorovitch metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P, Q) = \sup_{f \in ??} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P, Q) = \sup_{f \in ??} | \int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.
- Nonexpansive or Lipschitz-1 functions: $d(f(x), f(y)) \leq d(x, y)$.

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P, Q) = \sup_{f \in ??} | \int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.
- Nonexpansive or Lipschitz-1 functions: $d(f(x), f(y)) \leq d(x, y)$.
- Such functions are always continuous but, clearly, continuous functions are not necessarily Lipschitz-1.

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P, Q) = \sup_{f \in ??} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.
- Nonexpansive or Lipschitz-1 functions: $d(f(x), f(y)) \leq d(x, y)$.
- Such functions are always continuous but, clearly, continuous functions are not necessarily Lipschitz-1.
- $\kappa(P, Q) = \sup_{f \in \mathrm{Lip}_1} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P,Q) = \sup_{f \in ??} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.
- Nonexpansive or Lipschitz-1 functions: $d(f(x), f(y)) \leq d(x,y)$.
- Such functions are always continuous but, clearly, continuous functions are not necessarily Lipschitz-1.
- $\kappa(P,Q) = \sup_{f \in \mathrm{Lip}_1} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- It is easy to verify all the metric conditions.

The Kantorovich metric

- What is the observable aspect of a probability distribution?
- Expectation values.
- $\kappa(P,Q) = \sup_{f\in ??} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- But what kind of functions should we allow? Not just continuous ones.
- Nonexpansive or Lipschitz-1 functions: $d(f(x), f(y)) \leq d(x,y)$.
- Such functions are always continuous but, clearly, continuous functions are not necessarily Lipschitz-1.
- $\kappa(P,Q) = \sup_{f\in \mathrm{Lip}_1} |\int f \mathrm{d}P - \int f \mathrm{d}Q|$
- It is easy to verify all the metric conditions.
- But this definition is only half the story.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a “transport plan.”

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a “transport plan.”
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a "transport plan."
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.
- There is always the independent coupling: $\gamma(A \times B) = P(A)Q(B)$.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a "transport plan."
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.
- There is always the independent coupling: $\gamma(A \times B) = P(A)Q(B)$.
- But there are many others: the convex combinations of couplings are couplings.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a “transport plan.”
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.
- There is always the independent coupling: $\gamma(A \times B) = P(A)Q(B)$.
- But there are many others: the convex combinations of couplings are couplings.
- We write $\mathcal{C}(P, Q)$ for the set of couplings of P and Q.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a “transport plan.”
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.
- There is always the independent coupling: $\gamma(A \times B) = P(A)Q(B)$.
- But there are many others: the convex combinations of couplings are couplings.
- We write $\mathcal{C}(P, Q)$ for the set of couplings of P and Q.
- We can also define a coupling to be a pair of random variables R, S with distributions P, Q respectively.

Couplings

- How to relate two distributions? Think of a distribution as a pile of sand.
- We need to move some sand around to make the pile P look like Q.
- There are many different ways to do it. Each way is a “transport plan.”
- A **coupling** of two distributions P, Q defined on X is a *joint* distribution γ on $X \times X$ such that the *marginals* of γ are P and Q.
- There is always the independent coupling: $\gamma(A \times B) = P(A)Q(B)$.
- But there are many others: the convex combinations of couplings are couplings.
- We write $\mathcal{C}(P, Q)$ for the set of couplings of P and Q.
- We can also define a coupling to be a pair of random variables R, S with distributions P, Q respectively.
- We can also define couplings easily between two different underlying spaces X and Y.

The W metrics

- A coupling γ defines a transport plan, how much does it cost?

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X \times X} d(x,y) \mathrm{d}\gamma$

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X \times X} d(x,y) \mathrm{d}\gamma$
- We define a metric: $W_1(P,Q) = \inf_{\gamma \in \mathcal{C}(P,Q)} \int_{X \times X} d(x,y) \mathrm{d}\gamma$.

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X\times X} d(x,y)\mathrm{d}\gamma$
- We define a metric: $W_1(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} \int_{X\times X} d(x,y)\mathrm{d}\gamma$.
- Kantorovich-Rubinstein duality: $\kappa = W_1$.

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X\times X} d(x,y)\mathrm{d}\gamma$
- We define a metric: $W_1(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} \int_{X\times X} d(x,y)\mathrm{d}\gamma$.
- Kantorovich-Rubinstein duality: $\kappa = W_1$.
- $W_p(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} [\int_{X\times X} [d(x,y)]^p \mathrm{d}\gamma]^{\frac{1}{p}}$.

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X\times X} d(x,y)\mathrm{d}\gamma$
- We define a metric: $W_1(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} \int_{X\times X} d(x,y)\mathrm{d}\gamma$.
- Kantorovich-Rubinstein duality: $\kappa = W_1$.
- $W_p(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} [\int_{X\times X} [d(x,y)]^p \mathrm{d}\gamma]^{\frac{1}{p}}$.
- Crucial point: if I find *any* coupling it gives an *upper bound* on W_1.

The W metrics

- A coupling γ defines a transport plan, how much does it cost?
- If we measure the cost by a metric d we get
- cost $= \int_{X\times X} d(x,y)\mathrm{d}\gamma$
- We define a metric: $W_1(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} \int_{X\times X} d(x,y)\mathrm{d}\gamma$.
- Kantorovich-Rubinstein duality: $\kappa = W_1$.
- $W_p(P,Q) = \inf_{\gamma\in\mathcal{C}(P,Q)} [\int_{X\times X} [d(x,y)]^p \mathrm{d}\gamma]^{\frac{1}{p}}$.
- Crucial point: if I find *any* coupling it gives an *upper bound* on W_1.
- We can define a map from a metric space (M,d) to the space $(\mathcal{P}(M), W_1)$ by $x \mapsto \delta_x$. This map is an *isometry*.

Bisimulation via couplings

- Recall MDP's

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

Bisimulation via couplings

- Recall MDP's

$$(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \longrightarrow \mathcal{D}(S), \mathcal{R} : \mathcal{A} \times S \longrightarrow \mathbf{R})$$

- An equivalence relation R on S is a **bisimulation** if sRt implies that $\forall a \in \mathcal{A}$ there is a *coupling* ω of $P^a(s)$ and $P^a(t)$ such that the *support* of ω is contained in R.

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.
- We define $T_K : \mathcal{M} \to \mathcal{M}$ by

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.
- We define $T_K : \mathcal{M} \longrightarrow \mathcal{M}$ by
- $T_K(d)(x, y) = \max_a[|\mathcal{R}(x, a)\mathcal{R}(y, a)| + \gamma W_d(P^a(x), P^a(y))]$

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.
- We define $T_K : \mathcal{M} \longrightarrow \mathcal{M}$ by
- $T_K(d)(x, y) = \max_a[|\mathcal{R}(x, a)\mathcal{R}(y, a)| + \gamma W_d(P^a(x), P^a(y))]$
- This is a monotone function on $\mathcal{M}$.

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.
- We define $T_K : \mathcal{M} \longrightarrow \mathcal{M}$ by
- $T_K(d)(x, y) = \max_a[|\mathcal{R}(x, a)\mathcal{R}(y, a)| + \gamma W_d(P^a(x), P^a(y))]$
- This is a monotone function on $\mathcal{M}$.
- We can find the bisimulation as the fixed point of T_K by iteration: $d^{\sim}$.

Computing the bisimulation metric

- Let $\mathcal{M}$ be the space of 1-bounded pseudometrics over S, ordered by $d_1 \leq d_2$ if $\forall x, y; d_2(x, y) \leq d_1(x, y)$.
- This is a complete lattice.
- We define $T_K : \mathcal{M} \longrightarrow \mathcal{M}$ by
- $T_K(d)(x, y) = \max_a [|\mathcal{R}(x, a)\mathcal{R}(y, a)| + \gamma W_d(P^a(x), P^a(y))]$
- This is a monotone function on $\mathcal{M}$.
- We can find the bisimulation as the fixed point of T_K by iteration: $d^{\sim}$.
- An important bound proved by Ferns et al. $|V^*(x) - V^*(y)| \leq d^{\sim}(x, y)$.

Computational complexity

- Iteration of T_K to obtain an ε-approximation to the metric requires $O(\log(\varepsilon)/\log(\gamma))$ iterations.

Computational complexity

- Iteration of T_K to obtain an ε-approximation to the metric requires $O(\log(\varepsilon)/\log(\gamma))$ iterations.
- Each iteration requires the computation of $|S|^2|\mathcal{A}|$ distances.

Computational complexity

- Iteration of T_K to obtain an ε-approximation to the metric requires $O(\log(\varepsilon)/\log(\gamma))$ iterations.
- Each iteration requires the computation of $|S|^2|\mathcal{A}|$ distances.
- Each W_d distance computation is $O(|S|^3)$.

Computational complexity

- Iteration of T_K to obtain an ε-approximation to the metric requires $O(\log(\varepsilon)/\log(\gamma))$ iterations.
- Each iteration requires the computation of $|S|^2|\mathcal{A}|$ distances.
- Each W_d distance computation is $O(|S|^3)$.
- So the overall cost is $O(|S|^5|\mathcal{A}|\log(\varepsilon)/\log(\gamma)$.

Computational complexity

- Iteration of T_K to obtain an ε-approximation to the metric requires $O(\log(\varepsilon)/\log(\gamma))$ iterations.
- Each iteration requires the computation of $|S|^2|\mathcal{A}|$ distances.
- Each W_d distance computation is $O(|S|^3)$.
- So the overall cost is $O(|S|^5|\mathcal{A}|\log(\varepsilon)/\log(\gamma)$.
- Too high in practice!

Bias

- Computating T_K requires access to $P^a(x)$ for each x and a; typically not available.

Bias

- Computating T_K requires access to $P^a(x)$ for each x and a; typically not available.
- So we use sampling to estimate these quantities.

Bias

- Computating T_K requires access to $P^a(x)$ for each x and a; typically not available.
- So we use sampling to estimate these quantities.
- Unfortunately it is not easy to obtain these samples and in particular most methods used give biased samples.

Non-optimal policies

- We have $|V^*(x) - V^*(y)| \leq d^{\sim}(x, y)$.

Non-optimal policies

- We have $|V^*(x) - V^*(y)| \leq d^{\sim}(x, y)$.
- But if we have a fixed policy π, which may not be optimal, we do not have the inequality $|V^{\pi}(x) - V^{\pi}(y)| \leq d^{\sim}(x, y)$.

Non-optimal policies

- We have $|V^*(x) - V^*(y)| \leq d^{\sim}(x, y)$.
- But if we have a fixed policy π, which may not be optimal, we do not have the inequality $|V^{\pi}(x) - V^{\pi}(y)| \leq d^{\sim}(x, y)$.
- We often need V^{π} for non-optimal policies and the bismulation metric does not help us bound it.

The MICo distance

- MICo: matching under independent couplings.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \to \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$
- $(T_M^\pi U)(x,y) = |r^\pi(x) - r^\pi(y)| + \gamma\mathbb{E}_{x'\sim P^\pi(x), y'\sim\, P^\pi(y)}[U(x',y')]$.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S \times S} \longrightarrow \mathbf{R}^{S \times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a \sim \pi(s)}[\mathcal{R}(x, a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a) P^a(x)$
- $(T^\pi_M U)(x, y) = |r^\pi(x) - r^\pi(y)| + \gamma \mathbb{E}_{x' \sim P^\pi(x), y' \sim P^\pi(y)}[U(x', y')]$.
- If we use the L^∞ norm, T_M is a contraction so we have a fixed point by Banach's fixed point theorem.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$
- $(T_M^\pi U)(x,y) = |r^\pi(x) - r^\pi(y)| + \gamma\mathbb{E}_{x'\sim P^\pi(x), y'\sim\ P^\pi(y)}[U(x',y')]$.
- If we use the L^∞ norm, T_M is a contraction so we have a fixed point by Banach's fixed point theorem.
- Call the fixed point U^π.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$
- $(T_M^\pi U)(x,y) = |r^\pi(x) - r^\pi(y)| + \gamma\mathbb{E}_{x'\sim P^\pi(x), y'\sim P^\pi(y)}[U(x',y')]$.
- If we use the L^∞ norm, T_M is a contraction so we have a fixed point by Banach's fixed point theorem.
- Call the fixed point U^π.
- Of course this will not give us a metric!

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$
- $(T_M^\pi U)(x,y) = |r^\pi(x) - r^\pi(y)| + \gamma\mathbb{E}_{x'\sim P^\pi(x), y'\sim\ P^\pi(y)}[U(x',y')]$.
- If we use the L^∞ norm, T_M is a contraction so we have a fixed point by Banach's fixed point theorem.
- Call the fixed point U^π.
- Of course this will not give us a metric!
- But who knows, maybe it tells us something good.

The MICo distance

- MICo: matching under independent couplings.
- Do not try to find the optimal coupling use a simple known coupling, the independent coupling.
- We define a new update $T_M : \mathbf{R}^{S\times S} \longrightarrow \mathbf{R}^{S\times S}$ instead of T_K.
- We define $r^\pi(x) := \mathbb{E}_{a\sim\pi(s)}[\mathcal{R}(x,a)]$ and
- $P^\pi(x) = \sum_a \pi(x)(a)P^a(x)$
- $(T_M^\pi U)(x,y) = |r^\pi(x) - r^\pi(y)| + \gamma\mathbb{E}_{x'\sim P^\pi(x), y'\sim\, P^\pi(y)}[U(x',y')]$.
- If we use the L^∞ norm, T_M is a contraction so we have a fixed point by Banach's fixed point theorem.
- Call the fixed point U^π.
- Of course this will not give us a metric!
- But who knows, maybe it tells us something good.
- Complexity is $O(|S|^4)$ still not good but Google has fancy hardware!

What good is MICo?

- Computational complexity down to $O(|S|^4)$, a bit better. Also no factor of $|\mathcal{A}|$ since we are sticking to a particular policy.

What good is MICo?

- Computational complexity down to $O(|S|^4)$, a bit better. Also no factor of $|\mathcal{A}|$ since we are sticking to a particular policy.
- We can use online updates rather than iterating the actual T_M operator.

What good is MICo?

- Computational complexity down to $O(|S|^4)$, a bit better. Also no factor of $|\mathcal{A}|$ since we are sticking to a particular policy.
- We can use online updates rather than iterating the actual T_M operator.
- If stepsizes ($\varepsilon_t(x, y)$) decrease according to some specific conditions (Robbins-Munro) then we get convergence for the following sequence of updates

$$U_{t+1}(x, y) \to (1 - \varepsilon_t(x, y))U_t(x, y) + \varepsilon_t(x, y)(|r - \tilde{r}| + \gamma U_t(x', y'))$$

What good is MICo?

- Computational complexity down to $O(|S|^4)$, a bit better. Also no factor of $|\mathcal{A}|$ since we are sticking to a particular policy.
- We can use online updates rather than iterating the actual T_M operator.
- If stepsizes ($\varepsilon_t(x,y)$) decrease according to some specific conditions (Robbins-Munro) then we get convergence for the following sequence of updates

$$U_{t+1}(x,y) \rightarrow (1-\varepsilon_t(x,y))U_t(x,y) + \varepsilon_t(x,y)(|r-\tilde{r}| + \gamma U_t(x',y'))$$

- where we are updating using a pair of transitions (x_t, a_t, r_t, x_t') and $(y_t, b_t, \tilde{r}_t, y_t')$.

What good is MICo?

- Computational complexity down to $O(|S|^4)$, a bit better. Also no factor of $|\mathcal{A}|$ since we are sticking to a particular policy.
- We can use online updates rather than iterating the actual T_M operator.
- If stepsizes $(\varepsilon_t(x,y))$ decrease according to some specific conditions (Robbins-Munro) then we get convergence for the following sequence of updates

$$U_{t+1}(x,y) \to (1-\varepsilon_t(x,y))U_t(x,y) + \varepsilon_t(x,y)(|r-\tilde{r}| + \gamma U_t(x',y'))$$

- where we are updating using a pair of transitions (x_t, a_t, r_t, x'_t) and $(y_t, b_t, \tilde{r}_t, y'_t)$.
- $|V^\pi(x) - V^\pi(y)| \leq U^{\pi(x,y)}$.

A new type of distance

Diffuse metric

A new type of distance

Diffuse metric

1. $d(x, y) \geq 0$

A new type of distance

Diffuse metric

1. $d(x, y) \geq 0$
2. $d(x, y) = d(y, x)$

A new type of distance

Diffuse metric

1. $d(x, y) \geq 0$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$

A new type of distance

Diffuse metric

1. $d(x, y) \geq 0$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$
4. Do not require $d(x, x) = 0$

What is MICo?

Similar to, but not the same as, partial metrics (Matthews) or weak partial pseudometrics (Heckmann). They require stronger conditions than our triangle and they can then extract a real metric and something like a “norm”. Our examples violate their conditions.

What is MICo?

Similar to, but not the same as, partial metrics (Matthews) or weak partial pseudometrics (Heckmann). They require stronger conditions than our triangle and they can then extract a real metric and something like a “norm”. Our examples violate their conditions.

MICo distance is a diffuse metric.

MICo loss

- Nearly all machine learning algorithms are optimization algorithms.

MICo loss

- Nearly all machine learning algorithms are optimization algorithms.
- One often introduces extra terms into the objective function that push the solution in a desired direction.

MICo loss

- Nearly all machine learning algorithms are optimization algorithms.
- One often introduces extra terms into the objective function that push the solution in a desired direction.
- We defined a loss term based on the fixed point of the MICo update operator.

MICo loss

- Nearly all machine learning algorithms are optimization algorithms.
- One often introduces extra terms into the objective function that push the solution in a desired direction.
- We defined a loss term based on the fixed point of the MICo update operator.
- We assume a value-based agent learning as estimate based on two function approximators ψ, ϕ with their own sets of parameters.

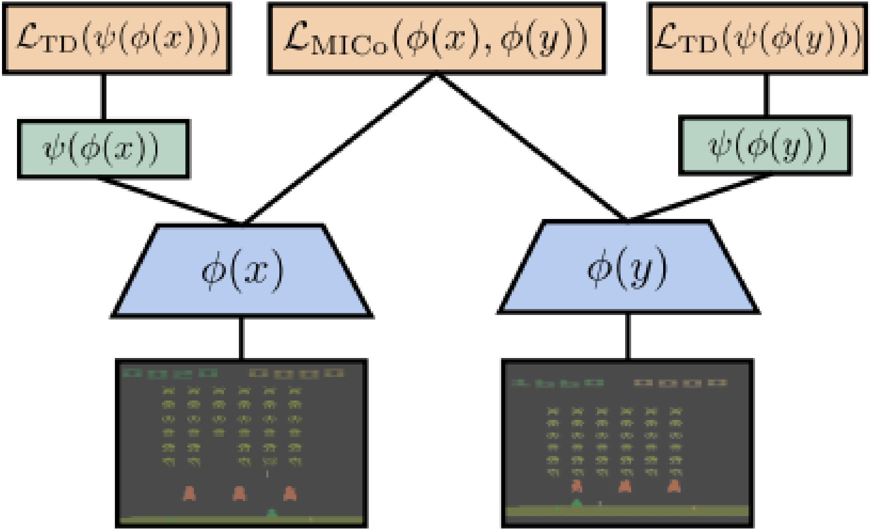
MICo loss

- Nearly all machine learning algorithms are optimization algorithms.
- One often introduces extra terms into the objective function that push the solution in a desired direction.
- We defined a loss term based on the fixed point of the MICo update operator.
- We assume a value-based agent learning as estimate based on two function approximators ψ, ϕ with their own sets of parameters.
- We then define a loss term based on the MICo distance.

MICo loss

- Nearly all machine learning algorithms are optimization algorithms.
- One often introduces extra terms into the objective function that push the solution in a desired direction.
- We defined a loss term based on the fixed point of the MICo update operator.
- We assume a value-based agent learning as estimate based on two function approximators ψ, ϕ with their own sets of parameters.
- We then define a loss term based on the MICo distance.
- For details read `https://psc-g.github.io/posts/research/rl/mico/`

Experimental setup



Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.

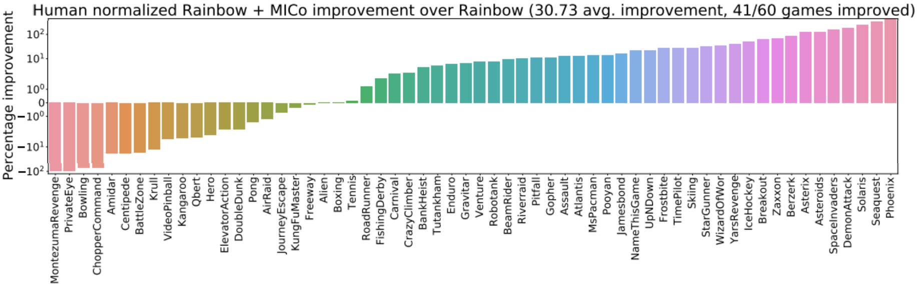
Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Hyperparamemters settings were taken from the Library.

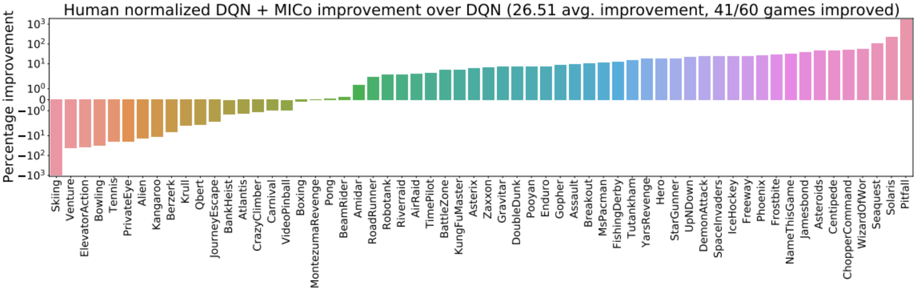
Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Hyperparamemters settings were taken from the Library.
- The learning algorithms tried to learn good strategies for Atari games. We tried each agent with and without the MICo loss term on 60 different Atari games.

Results for Rainbow



Results for DQN



Conclusions

- Explored the use of state-similarity metrics in improving representation learning.

Conclusions

- Explored the use of state-similarity metrics in improving representation learning.
- Variations of the concept of metric seem to be important.

Conclusions

- Explored the use of state-similarity metrics in improving representation learning.
- Variations of the concept of metric seem to be important.
- Connections to Reproducing Kernel Hilbert Space theory is being explored.