Strachey Lecture
From bisimulation to representation learning via metrics

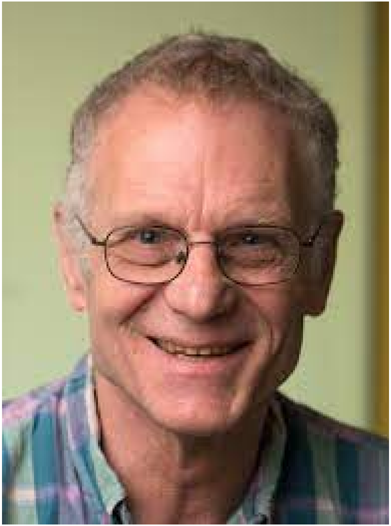
Prakash Panangaden
School of Computer Science
McGill University
and
Montreal Institute of Learning Algorithms

19 November 2024

Christopher Strachey

Heros of concurrency theory: Milner and Park

Sources of Inspiration I: Dexter Kozen



Sources of Inspiration II: Lawvere

Sources of Inspiration III: Giry

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?
- What should be guaranteed?

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- What should be guaranteed?
- (i) If two states are equivalent we should not be able to "see" any differences in observable behaviour.

Behavioural equivalence is fundamental

- When do two states have exactly the same behaviour?
- What can one observe of the behaviour?
- What should be guaranteed?
- (i) If two states are equivalent we should not be able to "see" any differences in observable behaviour.
- (ii) If two states are equivalent they should stay equivalent as they evolve.

A bit of history I

- Cantor and the back-and-forth argument

A bit of history I

- Cantor and the back-and-forth argument
- Lumpability in Markov chains: 1960's

A bit of history I

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

A bit of history I

- Cantor and the back-and-forth argument
- Lumpability in Markov chains: 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation, discrete systems: Larsen and Skou 1989

A bit of history I

- Cantor and the back-and-forth argument
- Lumpability in Markov chains: 1960's
- Bisimulation of nondeterministic automata 1970's and process algebras 1980's: Milner and Park
- Probabilistic bisimulation, discrete systems: Larsen and Skou 1989
- Adding durations: CTMC's, timed Petri nets, PEPA Hillston 1993

A bit of history II

- Bisimulation of Markov processes on continuous state spaces: Desharnais, Edalat, P. 1997...

A bit of history II

- 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 II

- 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 II

- 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 II

- 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

A bit of history II

- 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
- Representation learning using "metrics": Castro, Kastner, P. Rowland NeurIPS 2021

A bit of history II

- 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
- Representation learning using "metrics": Castro, Kastner, P. Rowland NeurIPS 2021
- Kernel perspective: Castro, Kastner, P. Rowland TMLR 2023

Labelled transition systems

- A set of states S,

Labelled transition systems

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and

Labelled transition systems

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and
- a transition relation $\subseteq S \times \mathcal{A} \times S$, usually written

$$\rightarrow_a \subseteq S \times S.$$

The transitions could be indeterminate (nondeterministic).

Labelled transition systems

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and
- a transition relation $\subseteq S \times \mathcal{A} \times S$, usually written

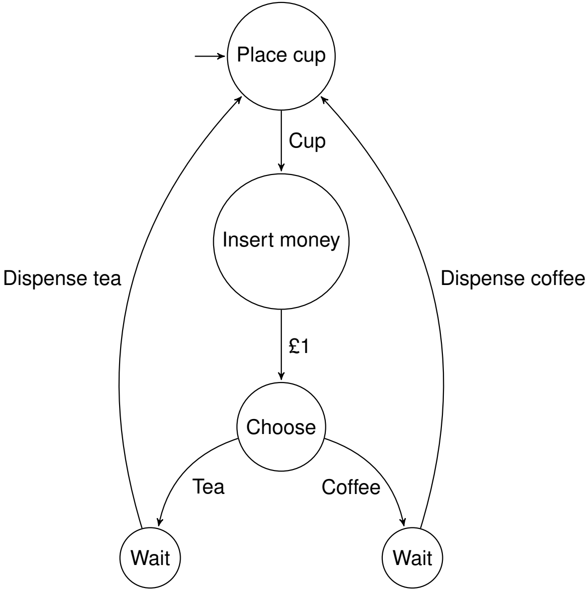
$$\rightarrow_a \subseteq S \times S.$$

The transitions could be indeterminate (nondeterministic).

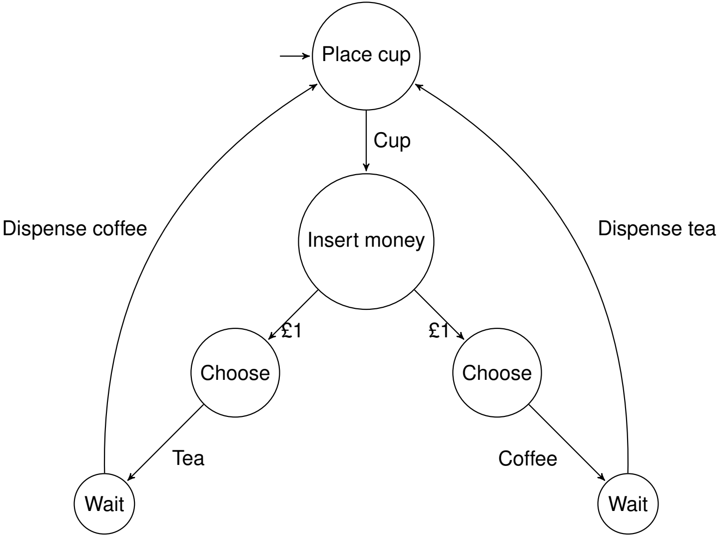
Notation

We write $s \xrightarrow{a} s'$ for $(s, s') \in \rightarrow_a$

Vending machine LTSs



Vending machine LTSs



Are the two LTSs equivalent?

- One gives *us* the choice whereas the other makes the choice *internally*.

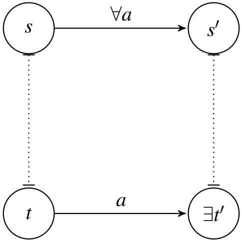
Are the two LTSs equivalent?

- One gives *us* the choice whereas the other makes the choice *internally*.
- The sequences that the machines can perform are identical: $[\mathrm{Cup};\pounds 1;(\mathrm{Cof}+\mathrm{Tea})]^*$

Are the two LTSs equivalent?

- One gives *us* the choice whereas the other makes the choice *internally*.
- The sequences that the machines can perform are identical: $[\mathrm{Cup};£1;(\mathrm{Cof}+\mathrm{Tea})]^*$
- We need to go beyond language equivalence.

Formal definition



Bisimulation definition

If $s \sim t$ then

$$\forall s \in S, \forall a \in \mathcal{A}, s \xrightarrow{a} s' \Rightarrow \exists t', t \xrightarrow{a} t' \text{ with } s' \sim t'$$

and *vice versa* with s and t interchanged.

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.

Discrete probabilistic transition systems

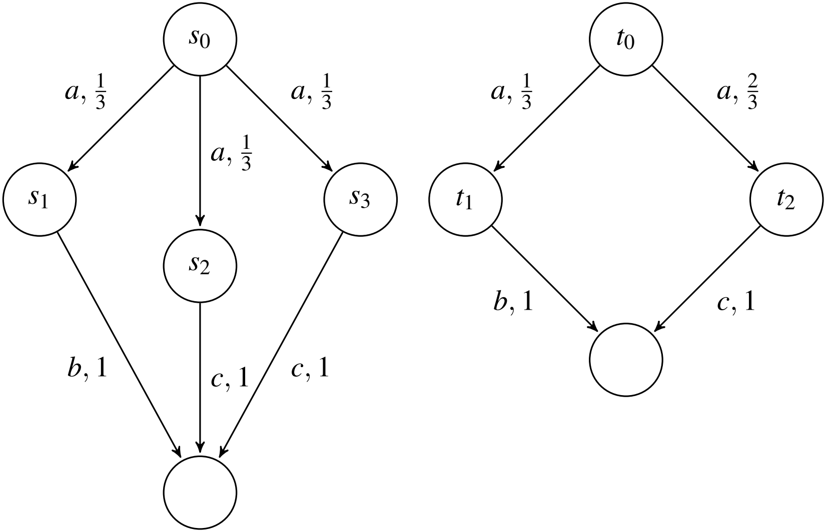
- Just like a labelled transition system with probabilities associated with the transitions.
-

$$(S, \mathcal{A}, \forall a \in \mathcal{A}\ T_a : S \longrightarrow \mathsf{Dist}(S)).$$

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathcal{A}, \forall a \in \mathcal{A}\ T_a : S \longrightarrow \mathsf{Dist}(S)).$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

Probabilistic bisimulation : Larsen and Skou



Are s_0 and t_0 bisimilar?

Yes, but one needs to add up the probabilities to s_2 and s_3.

Are s_0 and t_0 bisimilar?

Yes, but one needs to add up the probabilities to s_2 and s_3.

If s is a state, a an action and C a set of states, we write $T_a(s, C) = \sum_{s' \in S} T_a(s, s')$ for the probability of jumping on an a-action to one of the states in C.

Are s_0 and t_0 bisimilar?

Yes, but one needs to add up the probabilities to s_2 and s_3.

If s is a state, a an action and C a set of states, we write $T_a(s, C) = \sum_{s' \in S} T_a(s, s')$ for the probability of jumping on an a-action to one of the states in C.

Definition

R is a bisimulation relation if whenever sRt and C is an equivalence class of R then $T_a(s, C) = T_a(t, C)$.

Markov decision processes?

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

Markov decision processes?

- Markov decision processes 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.

Markov decision processes?

- Markov decision processes 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: numerous algorithms to find or approximate the optimal policy.

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) $P^a(s,C) = P^a(t,C)$
- 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) $P^a(s,C) = P^a(t,C)$
- 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.

Continuous state spaces: why?

- Software controllers attached to physical devices or sensors - robots, controllers.

Continuous state spaces: why?

- Software controllers attached to physical devices or sensors - robots, controllers.
- Continuous state space but discrete time.

Continuous state spaces: why?

- Software controllers attached to physical devices or sensors - robots, controllers.
- Continuous state space but discrete time.
- Applications to control systems.

Continuous state spaces: why?

- Software controllers attached to physical devices or sensors - robots, controllers.
- Continuous state space but discrete time.
- Applications to control systems.
- Applications to probabilistic programming languages.

Some remarks on the use of continuous spaces

- Can be used for reasoning - but much better if we could have a finite-state version.

Some remarks on the use of continuous spaces

- Can be used for reasoning - but much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case?

Some remarks on the use of continuous spaces

- Can be used for reasoning - but much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case?
- How can we say that our discrete approximation is "accurate"?

Some remarks on the use of continuous spaces

- Can be used for reasoning - but much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case?
- How can we say that our discrete approximation is "accurate"?
- We lose the ability to *refine* the model later.

The Need for Measure Theory

- Basic fact: There are subsets of $\mathbf{R}$ for which no sensible notion of size can be defined.

The Need for Measure Theory

- Basic fact: There are subsets of $\mathbf{R}$ for which no sensible notion of size can be defined.
- More precisely, there is no translation-invariant measure defined on all the subsets of the reals.

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]
- No finite branching assumption.

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]
- No finite branching assumption.
- No negation in the logic,

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]
- No finite branching assumption.
- No negation in the logic,
- The proof uses tools from descriptive set theory and measure theory.

Logical Characterization

- Very austere logic:

$$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$

- $s \models \langle a \rangle_q \phi$ means that if the system is in state s, then after the action a, with probability at least q the new state will satisfy the formula ϕ.
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]
- No finite branching assumption.
- No negation in the logic,
- The proof uses tools from descriptive set theory and measure theory.
- Such a theorem originally proved for (non-probabilistic) systems with finite-branching restrictions by Hennessy and Milner in 1977 and van Benthem in 1976.

The proof "engine" Josée Desharnais

But...

- In the context of probability is exact equivalence reasonable?

But...

- In the context of probability is exact equivalence reasonable?
- We say “no”. A small change in the probability distributions may result in bisimilar processes no longer being bisimilar though they may be very “close” in behaviour.

But...

- In the context of probability is exact equivalence reasonable?
- We say “no”. A small change in the probability distributions may result in bisimilar processes no longer being bisimilar though they may be very “close” in behaviour.
- Instead one should have a (pseudo)metric for probabilistic processes.

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.

In lieu of several slides of greek letters and symbols

- If two states are **not** bisimilar there is a some observation on which they disagree.

In lieu of several slides of greek letters and symbols

- If two states are **not** bisimilar there is a some observation on which they disagree.
- They may diasagree on the reward or on the probability distribution that results from a transition.

In lieu of several slides of greek letters and symbols

- If two states are **not** bisimilar there is a some observation on which they disagree.
- They may diasagree on the reward or on the probability distribution that results from a transition.
- We need to measure the latter, we use the ~~Wasserstein~~ Kantorovich metric between probability distributions.

In lieu of several slides of greek letters and symbols

- If two states are **not** bisimilar there is a some observation on which they disagree.
- They may diasagree on the reward or on the probability distribution that results from a transition.
- We need to measure the latter, we use the ~~Wasserstein~~ Kantorovich metric between probability distributions.
- Intuitively, if the difference shows up only after a long and elaborate observation then we should make the states "nearby" in the bisimulation metric.

In lieu ... continued

- All this can be formalized and was originally done by Desharnais et al. and later with a beautiful fixed-point construction by van Breugel and Worrell.

In lieu ... continued

- All this can be formalized and was originally done by Desharnais et al. and later with a beautiful fixed-point construction by van Breugel and Worrell.
- Ferns et al. added rewards and showed that the bisimulation metric bounds the difference in optimal value functions in different states.

In lieu ... continued

- All this can be formalized and was originally done by Desharnais et al. and later with a beautiful fixed-point construction by van Breugel and Worrell.
- Ferns et al. added rewards and showed that the bisimulation metric bounds the difference in optimal value functions in different states.
- $|V^*(x) - V^*(y)| \leq Cd(x, y)$.

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?

Representation learning

- 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.

The MICo distance

- The Kantorovich metric is expensive to compute and difficult to estimate from samples.

The MICo distance

- The Kantorovich metric is expensive to compute and difficult to estimate from samples.
- We (Castro et al.) invented a version that is easy to estimate from samples.

The MICo distance

- The Kantorovich metric is expensive to compute and difficult to estimate from samples.
- We (Castro et al.) invented a version that is easy to estimate from samples.
- In spirit it is closely related to the bisimulation metric but it is a crude approximation

The MICo distance

- The Kantorovich metric is expensive to compute and difficult to estimate from samples.
- We (Castro et al.) invented a version that is easy to estimate from samples.
- In spirit it is closely related to the bisimulation metric but it is a crude approximation
- and is not even technically a metric!

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 $d(x,x) = 0$

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.

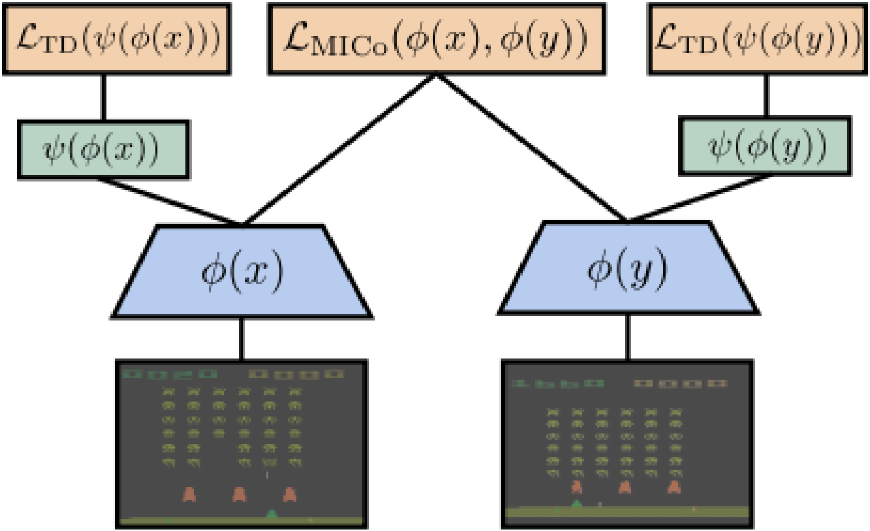
MICo loss

- 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 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 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.
- Ran each game 5 times with new seeds so 300 runs for each agent.

Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Ran each game 5 times with new seeds so 300 runs for each agent.
- Each game is run for 200 million environment interactions.

Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Ran each game 5 times with new seeds so 300 runs for each agent.
- Each game is run for 200 million environment interactions.
- We look at final scores and learning curve.

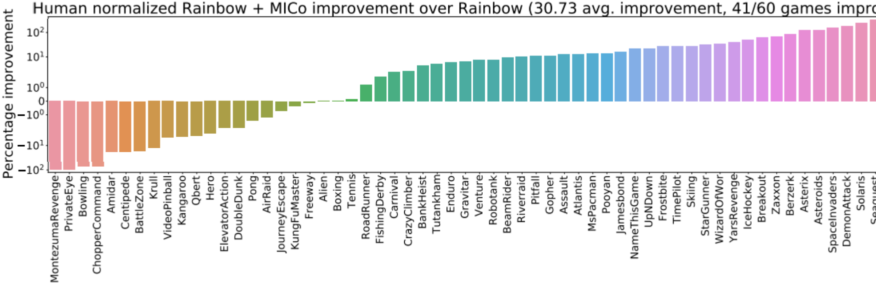
Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Ran each game 5 times with new seeds so 300 runs for each agent.
- Each game is run for 200 million environment interactions.
- We look at final scores and learning curve.
- We tried each agent with and without the MICo loss term on 60 different Atari games.

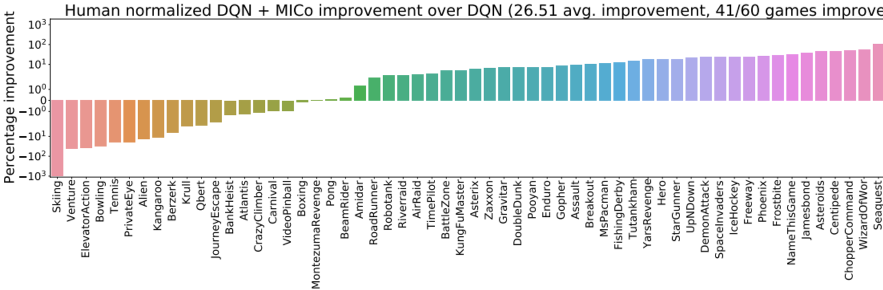
Experiments

- Added the MICo loss term to a variety of existing agents: all those available in the Dopamine Library; 5 in all.
- Ran each game 5 times with new seeds so 300 runs for each agent.
- Each game is run for 200 million environment interactions.
- We look at final scores and learning curve.
- We tried each agent with and without the MICo loss term on 60 different Atari games.
- *Every* agent performed better on about $\frac{2}{3}$ of the games.

Results for Rainbow



Results for DQN



Other developments

- Approximation of continuous-state-space systems by finite-state systems.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.
- Exploiting symmetry.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.
- Exploiting symmetry.
- Continuous space and time: Feller-Dynkin processes.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.
- Exploiting symmetry.
- Continuous space and time: Feller-Dynkin processes.
- A new kernel perspective on MiCo (TMLR 2023) shows where it “really” comes from.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.
- Exploiting symmetry.
- Continuous space and time: Feller-Dynkin processes.
- A new kernel perspective on MiCo (TMLR 2023) shows where it “really” comes from.
- Normalizing flows on hyperbolic spaces.

Other developments

- Approximation of continuous-state-space systems by finite-state systems.
- Capturing the notion of distance in an equational setting.
- Categorical mumbo jumbo.
- Exploiting symmetry.
- Continuous space and time: Feller-Dynkin processes.
- A new kernel perspective on MiCo (TMLR 2023) shows where it “really” comes from.
- Normalizing flows on hyperbolic spaces.
- Generative algorithm based on solving SDE's on Riemannian manifolds.

Conclusions

- Bisimulation has a rich and venerable history.

Conclusions

- Bisimulation has a rich and venerable history.
- The metric analogue holds promise for quantitative reasoning and approximation.

Conclusions

- Bisimulation has a rich and venerable history.
- The metric analogue holds promise for quantitative reasoning and approximation.
- Research is alive and well and there are new areas where bisimulation is being "discovered".

Some collaborators I



Some collaborators II





Special thanks