Milner Lecture
From bisimulation to representation learning via metrics

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

30 September 2021
Outline

1 Introduction
Outline

1. Introduction
2. Bisimulation for LTS’s
Outline

1. Introduction
2. Bisimulation for LTS’s
3. Probabilistic bisimulation
Outline

1. Introduction
2. Bisimulation for LTS’s
3. Probabilistic bisimulation
4. Continuous state spaces
Outline

1. Introduction
2. Bisimulation for LTS’s
3. Probabilistic bisimulation
4. Continuous state spaces
5. Metrics
Outline

1. Introduction
2. Bisimulation for LTS’s
3. Probabilistic bisimulation
4. Continuous state spaces
5. Metrics
6. Representation learning
Outline

1. Introduction
2. Bisimulation for LTS's
3. Probabilistic bisimulation
4. Continuous state spaces
5. Metrics
6. Representation learning
7. The MICo Distance
Outline

1. Introduction
2. Bisimulation for LTS's
3. Probabilistic bisimulation
4. Continuous state spaces
5. Metrics
6. Representation learning
7. The MICo Distance
8. Experimental results
Outline

1. Introduction
2. Bisimulation for LTS’s
3. Probabilistic bisimulation
4. Continuous state spaces
5. Metrics
6. Representation learning
7. The MICo Distance
8. Experimental results
9. Conclusions
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?

(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.
Behavioural equivalence is fundamental

- When do two states have \textit{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.
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.
Heros of concurrency theory: Milner and Park
Inspiration for my work II: Lawvere and Giry
Special thanks I
Special thanks II
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, discrete systems: 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, discrete systems: 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

Representation learning using “metrics”: Castro, Kastner, P. Rowland 2021
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, discrete systems: 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, discrete systems: 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, discrete systems: 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, discrete systems: 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
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, discrete systems: 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
- Representation learning using “metrics”: Castro, Kastner, P. Rowland 2021
The definition

- A set of states $S$, 

Panangaden  
Milner Lecture 2021  
30 September 2021  
10 / 40
The definition

- A set of states $S$,
- a set of *labels* or *actions*, $L$ or $A$ and
The definition

- 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).
The definition

- 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).

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

Vending machine LTSs

1. Place cup
2. Insert money
3. Choose
   - Coffee
   - Tea
4. Dispense coffee
5. Dispense tea
6. Wait
7. Wait
Vending machine LTSs

Place cup

Cup

Insert money

Dispense coffee

£1

Choose

Tea

Wait

Dispense tea

£1

Choose

Coffee

Wait
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: 
  
  \[
  \text{Cup, £1, (Cof + Tea)}^\ast
  \]
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: $[\text{Cup;£1;(Cof + Tea)}]^*$
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: 
  \[\text{[Cup;£1;(Cof + Tea)]}^*\]
- *We need to go beyond language equivalence.*
If \( s \sim t \) then

\[
\forall s \in S, \forall a \in A, s \xrightarrow{a} s' \implies \exists t', t \xrightarrow{a} t' \text{ with } s' \sim t'
\]

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.

\[(S, A, \forall a \in A T_a : S \times S \rightarrow [0, 1])\]
Discrete probabilistic transition systems

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

\[(S, \mathcal{A}, \forall a \in \mathcal{A} T_a : S \times S \rightarrow [0, 1])\]

- The model is reactive: All probabilistic data is internal - no probabilities associated with environment behaviour.
Probabilistic bisimulation: Larsen and Skou

\[ s_0 \rightarrow a, \frac{1}{3} \]
\[ s_1 \rightarrow b, 1 \]
\[ s_2 \rightarrow a, \frac{1}{3} \]
\[ s_3 \rightarrow a, \frac{1}{3} \]
\[ s_0 \rightarrow c, 1 \]
\[ t_0 \rightarrow a, \frac{2}{3} \]
\[ t_1 \rightarrow a, \frac{1}{3} \]
\[ t_0 \rightarrow c, 1 \]
\[ t_2 \rightarrow b, 1 \]
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

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

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 $P^a(s)(C)$. 

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

Policy

\[ \pi : S \rightarrow D(A) \]
Policies

**MDP**

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

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

**Policy**

$$\pi : S \rightarrow \mathcal{D}(\mathcal{A})$$
**MDP**

\[(S, \mathcal{A}, \forall a \in \mathcal{A}, P^a : S \rightarrow \mathcal{D}(S), R : \mathcal{A} \times S \rightarrow \mathbb{R})\]

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

**Policy**

\[\pi : S \rightarrow \mathcal{D}(\mathcal{A})\]

The goal is **choose** the best policy: numerous algorithms to find or approximate the optimal policy.
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) $R(a, s) = R(a, t)$

(ii) $P_a(s, C) = P_a(t, C)$

$s, t$ are bisimilar if there is a bisimulation relation $R$ with $sRt$.
Let $R$ be an equivalence relation. $R$ is a bisimulation if: $s \sim_R t$ if $(\forall a)$ and all equivalence classes $C$ of $R$:

(i) $R(a, s) = R(a, t)$

(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) $R(a, s) = R(a, t)$
(ii) $P^a(s, C) = P^a(t, C)$

$s$, $t$ are bisimilar if there is a bisimulation relation $R$ with $s \; R \; t$ 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.
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) $R(a, s) = R(a, t)$

(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) $R(a, s) = R(a, t)$
(ii) $P^a(s, C) = P^a(t, C)$

$s, t$ are bisimilar if there is a bisimulation relation $R$ with $s R t$ them.

Basic pattern: immediate rewards match (initiation), stay related after the transition (coinduction).
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 \textit{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.
Basic fact: There are subsets of $\mathbb{R}$ for which no sensible notion of size can be defined.
Basic fact: There are subsets of $\mathbb{R}$ 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:

\[ \mathcal{L} ::= T \phi_1 \land \phi_2 \langle a \rangle_q \phi \]
Logical Characterization

- Very austere logic:

\[ \mathcal{L} ::= T \phi_1 \land \phi_2 | \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).
Logical Characterization

- Very austere logic:
  \[ \mathcal{L} ::= T \phi_1 \land \phi_2 \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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:
  \[ \mathcal{L} ::= T \phi_1 \land \phi_2 \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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, so one can obtain a logical characterization result for simulation but it needs disjunction. The proof uses tools from descriptive set theory and measure theory. Such a theorem originally proved for LTS with finite-branching restrictions by Hennessy and Milner in 1977 and van Benthem in 1976.
Logical Characterization

- Very austere logic:
  \[ \mathcal{L} ::= T \phi_1 \land \phi_2 | \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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:

\[ \mathcal{L} ::= T | \phi_1 \land \phi_2 | \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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,
- so one can obtain a logical characterization result for simulation
Logical Characterization

- Very austere logic:
  \[ L ::= T \phi_1 \land \phi_2 \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- Two systems are bisimilar iff they obey the same formulas of \( L \).
  [DEP 1998 LICS, I and C 2002]

- No finite branching assumption.
- No negation in the logic,
- so one can obtain a logical characterization result for simulation
- but it needs disjunction.
Logical Characterization

- Very austere logic:

\[ \mathcal{L} ::= \top \phi_1 \land \phi_2 \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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,
- so one can obtain a logical characterization result for simulation
- but it needs disjunction.
- The proof uses tools from descriptive set theory and measure theory.
Logical Characterization

- Very austere logic:
  \[ \mathcal{L} ::= T \phi_1 \land \phi_2 \langle a \rangle_q \phi \]

- \( s \models \langle a \rangle_q \phi \) 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 \( \phi \).

- 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,
- so one can obtain a logical characterization result for simulation
- but it needs disjunction.
- The proof uses tools from descriptive set theory and measure theory.
- Such a theorem originally proved for LTS with finite-branching restrictions by Hennessy and Milner in 1977 and van Benthem in 1976.
The proof “engine” Josée Desharnais
In the context of probability is exact equivalence reasonable?
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.
If two states are not bisimilar there is some observation on which they disagree.
If two states are not bisimilar there is a some observation on which they disagree.

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

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

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

They may disagree 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.

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

They may disagree 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.

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.
It is possible to generalize the notion of equation to capture approximate equality.
Quantitative equational logic

- It is possible to generalize the notion of equation to capture approximate equality.
- \( s =_{\varepsilon} t \) means \( s \) is within \( \varepsilon \) of \( t \).
Quantitative equational logic

- It is possible to generalize the notion of equation to capture approximate equality.
- $s =_{\varepsilon} t$ means $s$ is within $\varepsilon$ of $t$.
- Much of the theory of equational logic carries over to this setting.
It is possible to generalize the notion of equation to capture approximate equality.

$s =_{\varepsilon} t$ means $s$ is within $\varepsilon$ of $t$.

Much of the theory of equational logic carries over to this setting.

Algebras for such equations are naturally equipped with metrics and give a way of reasoning about bisimulation metrics.
It is possible to generalize the notion of equation to capture approximate equality.

$s =_{\varepsilon} t$ means $s$ is within $\varepsilon$ of $t$.

Much of the theory of equational logic carries over to this setting.

Algebras for such equations are naturally equipped with metrics and give a way of reasoning about bisimulation metrics.

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 $S \times A \rightarrow \mathbb{R}$ is not feasible.

Instead we define a new space of features $M$ and try to come up with an embedding $\phi: S \rightarrow R^M$. Then we can try to use this to predict values associated with state, action pairs.

Representation learning means learning such a $\phi$. 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.
Representation learning

For large state spaces, learning value functions $S \times A \rightarrow \mathbb{R}$ is not feasible.

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

Instead we define a new space of features $M$ and try to come up with an embedding $\phi : S \rightarrow \mathbb{R}^M$.

Then we can try to use this to predict values associated with state,action pairs.
For large state spaces, learning value functions $S \times A \rightarrow \mathbb{R}$ is not feasible.

Instead we define a new space of features $M$ and try to come up with an embedding $\phi : S \rightarrow \mathbb{R}^M$.

Then we can try to use this to predict values associated with state, action pairs.

Representation learning means learning such a $\phi$. 
For large state spaces, learning value functions $S \times A \rightarrow \mathbb{R}$ is not feasible.

Instead we define a new space of features $M$ and try to come up with an embedding $\phi : S \rightarrow \mathbb{R}^M$.

Then we can try to use this to predict values associated with state,action pairs.

Representation learning means learning such a $\phi$.

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

\[ d(x, y) \geq 0 \]

\[ d(x, y) = d(y, x) \]

\[ d(x, y) \leq d(x, z) + d(z, y) \]

Do not require

\[ d(x, x) = 0 \]
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 \(d(x, x) = 0\)
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

\[ \mathcal{L}_{TD}(\psi(\phi(x))) \]
\[ \mathcal{L}_{MICo}(\phi(x), \phi(y)) \]
\[ \mathcal{L}_{TD}(\psi(\phi(y))) \]

\[ \psi(\phi(x)) \]
\[ \psi(\phi(y)) \]

\[ \phi(x) \]
\[ \phi(y) \]
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.
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.

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.
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.
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.
Human normalized Rainbow + MICO improvement over Rainbow (30.73 avg. improvement, 41/60 games improved)
Human normalized DQN + MiCo improvement over DQN (26.51 avg. improvement, 41/60 games improved)
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.
- Perhaps a fruitful line of research would be equation solving in quantitative algebras and automating equational reasoning in the quantitative setting.
Conclusions

- Bisimulation has a rich and venerable history.
- The metric analogue holds promise for quantitative reasoning and approximation.
- Perhaps a fruitful line of research would be equation solving in quantitative algebras and automating equational reasoning in the quantitative setting.
- Research is alive and well and there are new areas where bisimulation is being “discovered”.