Distributional analysis of sampling-based RL algorithms

Philip Amortila [1] Doina Precup [2,3,4] Prakash Panangaden [2,3]
Marc Bellemare [5]

[1] University of Illinois Urbana-Champaign
[2] McGill University; [3] MILA
[4] DeepMind; [5] GoogleBrain

Distinguished Lecture Series
Max Planck Institute Saarbrucken
5th May 2021

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 ourputs a 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 ourputs a 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 ourputs a 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 ourputs a 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 shrink it?

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

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.

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

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 (induction).

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 (induction).
- 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.

Banach fixed-point theorem

The grandmother of convergence arguments

If (M, d) is a *complete* metric space and $f : M \longrightarrow M$ is a *contractive* function (*i.e.* there is a $c \in (0, 1)$ such that for *every* $x, y \in M$ $d(f(x), f(y)) \leq c \cdot d(x, y)$) there *is* a *unique* $x_0 \in M$ such that $f(x_0) = x_0$.

Banach fixed-point theorem

The grandmother of convergence arguments

If (M, d) is a *complete* metric space and $f : M \longrightarrow M$ is a *contractive* function (*i.e.* there is a $c \in (0, 1)$ such that for *every* $x, y \in M$ $d(f(x), f(y)) \leq c \cdot d(x, y)$) there *is* a *unique* $x_0 \in M$ such that $f(x_0) = x_0$.

proof idea

Start *anywhere* and keep iterating f. The sequence $x, f(x), f(f(x)), f(f(f(x))), \ldots$ gets closer and closer because of the contractive property. Thus it has a limit (because of completeness) which is the desired fixed point.

Contractive functions and iteration

- Contractive functions are automatically continuous but continuous functions may or may not be contractive.

Contractive functions and iteration

- Contractive functions are automatically continuous but continuous functions may or may not be contractive.
- The Banach fixed-point theorem is used to justify the existence of solutions to Bellman equations.

Contractive functions and iteration

- Contractive functions are automatically continuous but continuous functions may or may not be contractive.
- The Banach fixed-point theorem is used to justify the existence of solutions to Bellman equations.
- One has usually to do some work to show that the function of interest is contractive.

Contractive functions and iteration

- Contractive functions are automatically continuous but continuous functions may or may not be contractive.
- The Banach fixed-point theorem is used to justify the existence of solutions to Bellman equations.
- One has usually to do some work to show that the function of interest is contractive.
- The proof essentially says, "iterative algorithms converge."

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*.

Convergence of monotone functions

- A *lattice* is a partially ordered set in which every subset (even the empty set) has a least upper bound (sup) and a greatest lower bound (inf).

Convergence of monotone functions

- A *lattice* is a partially ordered set in which every subset (even the empty set) has a least upper bound (sup) and a greatest lower bound (inf).
- A *monotone* function f from a complete lattice L to itself is a function such that for every $x, y \in L$ if $x \leq y$ then $f(x) \leq f(y)$.

Convergence of monotone functions

- A *lattice* is a partially ordered set in which every subset (even the empty set) has a least upper bound (sup) and a greatest lower bound (inf).
- A *monotone* function f from a complete lattice L to itself is a function such that for every $x, y \in L$ if $x \leq y$ then $f(x) \leq f(y)$.
- A monotone function from a complete lattice to itself has a *least* fixed point and a *greatest* fixed point.

Convergence of monotone functions

- A *lattice* is a partially ordered set in which every subset (even the empty set) has a least upper bound (sup) and a greatest lower bound (inf).
- A *monotone* function f from a complete lattice L to itself is a function such that for every $x, y \in L$ if $x \leq y$ then $f(x) \leq f(y)$.
- A monotone function from a complete lattice to itself has a *least* fixed point and a *greatest* fixed point.
- Actually the collection of fixed points itself is a complete lattice but that does not concern us here.

Convergence of monotone functions

- A *lattice* is a partially ordered set in which every subset (even the empty set) has a least upper bound (sup) and a greatest lower bound (inf).
- A *monotone* function f from a complete lattice L to itself is a function such that for every $x, y \in L$ if $x \leq y$ then $f(x) \leq f(y)$.
- A monotone function from a complete lattice to itself has a *least* fixed point and a *greatest* fixed point.
- Actually the collection of fixed points itself is a complete lattice but that does not concern us here.
- The convergence to the optimal policy follows from the montonicity of T^{π}.

RL algorithms

- The Bellman operator for an MDP depends on details of the model.

RL algorithms

- The Bellman operator for an MDP depends on details of the model.
- In the RL setting MDPs are usually not known so we cannot just apply Bellman operators.

RL algorithms

- The Bellman operator for an MDP depends on details of the model.
- In the RL setting MDPs are usually not known so we cannot just apply Bellman operators.
- We have to update based on *sampling*.

RL algorithms

- The Bellman operator for an MDP depends on details of the model.
- In the RL setting MDPs are usually not known so we cannot just apply Bellman operators.
- We have to update based on *sampling*.
- For example in $TD(0)$:
  $V_{n+1}(s) = (1-\alpha)V_n(s) + \alpha(r + \gamma V_n(s'))$

RL algorithms

- The Bellman operator for an MDP depends on details of the model.
- In the RL setting MDPs are usually not known so we cannot just apply Bellman operators.
- We have to update based on *sampling*.
- For example in $TD(0)$:
  $V_{n+1}(s) = (1-\alpha)V_n(s) + \alpha(r + \gamma V_n(s'))$
- where the action a is sampled according to the policy and the reward r and next state s' are sampled from the MDP.

RL algorithms

- The Bellman operator for an MDP depends on details of the model.
- In the RL setting MDPs are usually not known so we cannot just apply Bellman operators.
- We have to update based on *sampling*.
- For example in $TD(0)$:
  $V_{n+1}(s) = (1-\alpha)V_n(s) + \alpha(r + \gamma V_n(s'))$
- where the action a is sampled according to the policy and the reward r and next state s' are sampled from the MDP.
- Proof of convergence now involves stochastic approximation theory.

Value distributions

- The functions obtained by sampling are random variables.

Value distributions

- The functions obtained by sampling are random variables.
- We should study the distributions not just the expectation values.

Value distributions

- The functions obtained by sampling are random variables.
- We should study the distributions not just the expectation values.
- Distributional approach to RL: Marc Bellemare, Will Dabney and Rémi Munos.

Value distributions

- The functions obtained by sampling are random variables.
- We should study the distributions not just the expectation values.
- Distributional approach to RL: Marc Bellemare, Will Dabney and Rémi Munos.
- The sequence of distributions forms a Markov chain over the space of value functions.

Value distributions

- The functions obtained by sampling are random variables.
- We should study the distributions not just the expectation values.
- Distributional approach to RL: Marc Bellemare, Will Dabney and Rémi Munos.
- The sequence of distributions forms a Markov chain over the space of value functions.
- Does this converge? To what limit?

Stochastic Approximation Algorithms as Markov Chains

- Algorithms like $TD(0)$ are updating random variables.

Stochastic Approximation Algorithms as Markov Chains

- Algorithms like $TD(0)$ are updating random variables.
- A random variable induces a distribution so we are updating distributions.

Stochastic Approximation Algorithms as Markov Chains

- Algorithms like $TD(0)$ are updating random variables.
- A random variable induces a distribution so we are updating distributions.
- We view the *algorithm as a Markov chain* with the space of distributions as the *state space*.

Stochastic Approximation Algorithms as Markov Chains

- Algorithms like $TD(0)$ are updating random variables.
- A random variable induces a distribution so we are updating distributions.
- We view the *algorithm as a Markov chain* with the space of distributions as the *state space*.
- How do we reason about convergence in such a space?

Stochastic Approximation Algorithms as Markov Chains

- Algorithms like $TD(0)$ are updating random variables.
- A random variable induces a distribution so we are updating distributions.
- We view the *algorithm as a Markov chain* with the space of distributions as the *state space*.
- How do we reason about convergence in such a space?
- We need a *metric* on the space of probability distributions.

The basic 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 basic 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 basic 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 basic 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 total variation metric

- Let (X, d) be a metric space and let P, Q be probability distributions defined on (the Borel sets of) X.

The total variation metric

- Let (X, d) be a metric space and let P, Q be probability distributions defined on (the Borel sets of) X.
- If E is any (measurable) subset of X we can compare $P(E)$ and $Q(E)$.

The total variation metric

- Let (X, d) be a metric space and let P, Q be probability distributions defined on (the Borel sets of) X.
- If E is any (measurable) subset of X we can compare $P(E)$ and $Q(E)$.
- We define $TV(P, Q) = \sup_E |P(E) - Q(E)|$.

The total variation metric

- Let (X, d) be a metric space and let P, Q be probability distributions defined on (the Borel sets of) X.
- If E is any (measurable) subset of X we can compare $P(E)$ and $Q(E)$.
- We define $TV(P, Q) = \sup_E |P(E) - Q(E)|$.
- Why I love the TV metric: easy to define, relatively easy to compute, provides all kinds of useful bounds.

The total variation metric

- Let (X, d) be a metric space and let P, Q be probability distributions defined on (the Borel sets of) X.
- If E is any (measurable) subset of X we can compare $P(E)$ and $Q(E)$.
- We define $TV(P, Q) = \sup_E |P(E) - Q(E)|$.
- Why I love the TV metric: easy to define, relatively easy to compute, provides all kinds of useful bounds.
- Why I hate the TV metric: completely insensitive to the underlying metric.

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 Kantorovich metric

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

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.

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* γ of $P^a(s)$ and $P^a(t)$ such that the *support* of γ is contained in R.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.
- The update rule $\mathcal{U}$ takes an estimate f for the value function and produces a new estimate f'. This is **not** a function $f \mapsto f'$.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.
- The update rule $\mathcal{U}$ takes an estimate f for the value function and produces a new estimate f'. This is **not** a function $f \mapsto f'$.
- It is a probabilistic mapping called a *Markov kernel*: $K : \mathbf{R}^d \times \mathcal{B} \longrightarrow [0, 1]$, where $\mathcal{B}$ are the (Borel) subsets of $\mathbf{R}^d$.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.
- The update rule $\mathcal{U}$ takes an estimate f for the value function and produces a new estimate f'. This is **not** a function $f \mapsto f'$.
- It is a probabilistic mapping called a *Markov kernel*: $K : \mathbf{R}^d \times \mathcal{B} \longrightarrow [0, 1]$, where $\mathcal{B}$ are the (Borel) subsets of $\mathbf{R}^d$.
- $K(f, B) = \text{Prob}\{f' \in B\}$, where B is a Borel set.

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.
- The update rule $\mathcal{U}$ takes an estimate f for the value function and produces a new estimate f'. This is **not** a function $f \mapsto f'$.
- It is a probabilistic mapping called a *Markov kernel*: $K : \mathbf{R}^d \times \mathcal{B} \longrightarrow [0, 1]$, where $\mathcal{B}$ are the (Borel) subsets of $\mathbf{R}^d$.
- $K(f, B) = \text{Prob}\{f' \in B\}$, where B is a Borel set.
- The kernel will depend on the update rule (and step size).

Markov chains on the space of functions

- In RL algorithms the update rule *usually* depends only on the current estimate and the random samples.
- We take the MDP state space to be a *finite* set S.
- The space of value functions is a finite-dimensional vector space $\mathbf{R}^{|S|} = \mathbf{R}^d$.
- The update rule $\mathcal{U}$ takes an estimate f for the value function and produces a new estimate f'. This is **not** a function $f \mapsto f'$.
- It is a probabilistic mapping called a *Markov kernel*: $K : \mathbf{R}^d \times \mathcal{B} \longrightarrow [0, 1]$, where $\mathcal{B}$ are the (Borel) subsets of $\mathbf{R}^d$.
- $K(f, B) =$ Prob$\{f' \in B\}$, where B is a Borel set.
- The kernel will depend on the update rule (and step size).
- We can apply a kernel to a distribution over value functions: $K(P, B) = \int_{\mathbf{R}^d} K(\vec{x}, B)\mathrm{d}\vec{x}$.

Stochastic operators

- We want a general formalism to describe many update rules.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \text{Pr})$.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \mathsf{Pr})$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \mathrm{Pr})$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.
- A generic form for an update rule:
  $f_{n+1} = (1 - \alpha) f_n + \alpha \mathcal{T}(f_n, \omega)$.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \Pr)$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.
- A generic form for an update rule:
  $f_{n+1} = (1-\alpha)f_n + \alpha\mathcal{T}(f_n, \omega)$.
- Here α is the step size and $\mathcal{T}$ will depend on the algorithm.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \mathrm{Pr})$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.
- A generic form for an update rule: $f_{n+1} = (1 - \alpha) f_n + \alpha \mathcal{T}(f_n, \omega)$.
- Here α is the step size and $\mathcal{T}$ will depend on the algorithm.
- We say $\mathcal{T}$ is an *empirical Bellman operator* for a policy π if $\mathbb{E}_{\omega \sim \mathrm{Pr}}[\mathcal{T}(f, \omega)] = \mathcal{T}^{\pi}(f)$.

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \mathsf{Pr})$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.
- A generic form for an update rule:
  $f_{n+1} = (1-\alpha) f_n + \alpha \mathcal{T}(f_n, \omega)$.
- Here α is the step size and $\mathcal{T}$ will depend on the algorithm.
- We say $\mathcal{T}$ is an *empirical Bellman operator* for a policy π if
  $\mathbb{E}_{\omega \sim \mathsf{Pr}}[\mathcal{T}(f, \omega)] = \mathcal{T}^{\pi}(f)$.
- For $TD(0)$ the stochastic operator is:
  $\mathcal{T}(V, (a_s, r_s, s'_s)_{s \in S} = r_s \gamma V(s'_s)$

Stochastic operators

- We want a general formalism to describe many update rules.
- We have a source of randomness: $(\Omega, \mathcal{F}, \mathsf{Pr})$.
- A stochastic operator $\mathcal{T} : \mathbf{R}^d \times \Omega \longrightarrow \mathbf{R}^d$.
- A generic form for an update rule:
  $f_{n+1} = (1-\alpha)f_n + \alpha\mathcal{T}(f_n, \omega)$.
- Here α is the step size and $\mathcal{T}$ will depend on the algorithm.
- We say $\mathcal{T}$ is an *empirical Bellman operator* for a policy π if
  $\mathbb{E}_{\omega\sim\mathsf{Pr}}[\mathcal{T}(f,\omega)] = \mathcal{T}^{\pi}(f)$.
- For $TD(0)$ the stochastic operator is:
  $\mathcal{T}(V, (a_s, r_s, s'_s)_{s\in S} = r_s\gamma V(s'_s)$
- Here (a_s, r_s, s'_s) is sampled at every state s.

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma) W_1(P_1, P_2)$.

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma)W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y)\in\mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma)W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y)\in\mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$
- Let us start with any two distributions P, Q and we assume that (X_0, Y_0) is the optimal coupling: $W_1(P, Q) = \mathbb{E}[\|X_0 - Y_0\|]$.

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma) W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y) \in \mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$
- Let us start with any two distributions P, Q and we assume that (X_0, Y_0) is the optimal coupling: $W_1(P, Q) = \mathbb{E}[\|X_0 - Y_0\|]$.
- Now we define the coupling of the next estimates by forcing them to sample the same transitions at each state: $a \sim \pi(\cdot|s), r_s \sim ...$

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma) W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y) \in \mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$
- Let us start with any two distributions P, Q and we assume that (X_0, Y_0) is the optimal coupling: $W_1(P, Q) = \mathbb{E}[\|X_0 - Y_0\|]$.
- Now we define the coupling of the next estimates by forcing them to sample the same transitions at each state: $a \sim \pi(\cdot|s), r_s \sim \ldots$
- $X_1(s) = (1 - \alpha) X_0(s) + \alpha(r_s + \gamma X_0(s'_s))$
  $Y_1(s) = (1 - \alpha) Y_0(s) + \alpha(r_s + \gamma Y_0(s'_s))$

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma) W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y) \in \mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$
- Let us start with any two distributions P, Q and we assume that (X_0, Y_0) is the optimal coupling: $W_1(P, Q) = \mathbb{E}[\|X_0 - Y_0\|]$.
- Now we define the coupling of the next estimates by forcing them to sample the same transitions at each state: $a \sim \pi(\cdot|s), r_s \sim ...$
- $X_1(s) = (1 - \alpha) X_0(s) + \alpha(r_s + \gamma X_0(s'_s))$
  $Y_1(s) = (1 - \alpha) Y_0(s) + \alpha(r_s + \gamma Y_0(s'_s))$
- One can verify that this is a valid coupling of the updated distributions; nobody claims that this is the optimal coupling.

Updates in $TD(0)$

- We will show that $TD(0)$ defines a *contractive* Markov kernel: $W_1(K(P_1), K(P_2)) \leq (1 - \alpha + \alpha\gamma) W_1(P_1, P_2)$.
- If our coupling is given in terms of random variables X, Y $W_1(P, Q) = \inf_{(X,Y) \in \mathcal{C}(P,Q)} \mathbb{E}[\|X - Y\|_\infty]$
- Let us start with any two distributions P, Q and we assume that (X_0, Y_0) is the optimal coupling: $W_1(P, Q) = \mathbb{E}[\|X_0 - Y_0\|]$.
- Now we define the coupling of the next estimates by forcing them to sample the same transitions at each state: $a \sim \pi(\cdot|s), r_s \sim \ldots$
- $X_1(s) = (1 - \alpha) X_0(s) + \alpha(r_s + \gamma X_0(s'_s))$
  $Y_1(s) = (1 - \alpha) Y_0(s) + \alpha(r_s + \gamma Y_0(s'_s))$
- One can verify that this is a valid coupling of the updated distributions; nobody claims that this is the optimal coupling.
- However, simple inequality arguments shows that the upper bound on W_1 obtained with this coupling is enough to show contractivity.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.
- The key point is finding the proper coupling.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.
- The key point is finding the proper coupling.
- This simple idea works with little effort for MC, $TD(\lambda)$, SARSA, Q-learning.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.
- The key point is finding the proper coupling.
- This simple idea works with little effort for MC, $TD(\lambda)$, SARSA, Q-learning.
- It does not work for optimistic policy iteration where deeper techniques are needed.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.
- The key point is finding the proper coupling.
- This simple idea works with little effort for MC, $TD(\lambda)$, SARSA, Q-learning.
- It does not work for optimistic policy iteration where deeper techniques are needed.
- In the paper we analyze the stationary distributions attained and also discuss OPI with decreasing step size where we use monotonicity arguments.

Reaping the rewards

- The sequence of updates for $TD(0)$ converges in W_1 to a unique stationary distribution.
- The key point is finding the proper coupling.
- This simple idea works with little effort for MC, $TD(\lambda)$, SARSA, Q-learning.
- It does not work for optimistic policy iteration where deeper techniques are needed.
- In the paper we analyze the stationary distributions attained and also discuss OPI with decreasing step size where we use monotonicity arguments.
- Deeper analysis of OPI is underway with Philip, Marc and Rosie Zhao.

Thanks!

Paper and supplement available from AISTATS 2020 website.