Bisimulation Metric Computation for Markov Decision Processes

Sherry Shanshan Ruan

Supervisor: Prakash Panangaden, Gheorghe Comanici

School of Computer Science

McGill University

August 29, 2014

Background

- Markov Decision Process (MDPs) are a powerful mathematical model in planning and learning under uncertainty
- A major task in the study of MDPs is to compute the optimal value function
- The state space of MDPs is often too large to apply dynamic programming algorithms
- Use *probabilistic bisimulation* to aggregate states and reduce the size of the state space

Probabilistic bisimulation relations

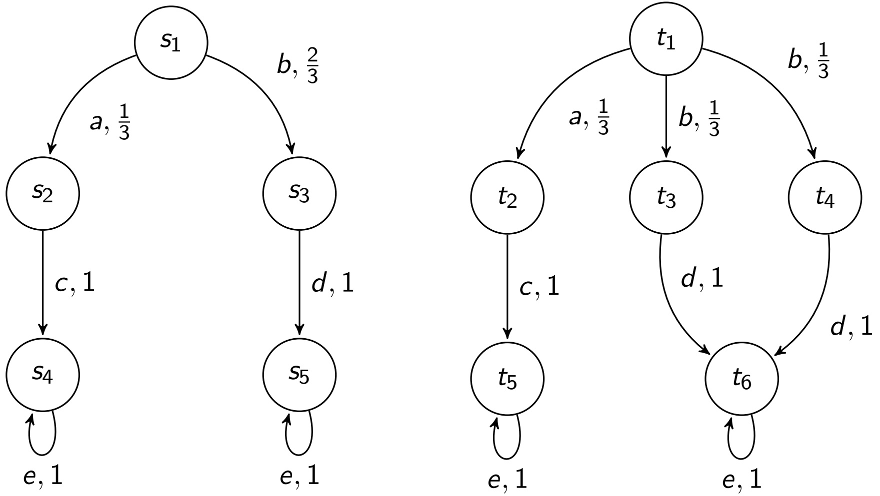
Definition (probabilistic bisimulation relations)

Given a Markov Decision Process (S, A, R, P, γ), a *probabilistic bisimulation relation* is an equivalence relation $\sim$ on S such that if $s \sim t$ then

$$(1)\ \forall a \in A, R_{sa} = R_{ta}$$

$$(2)\ \forall a \in A, \forall c \in S/\sim, P^a_{sc} = P^a_{tc}$$

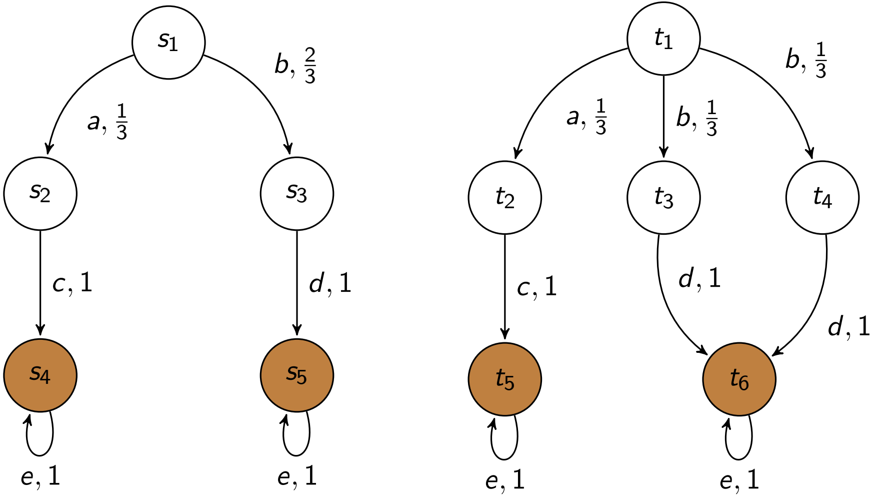
An example of probabilistic bisimulations



Quick Question:

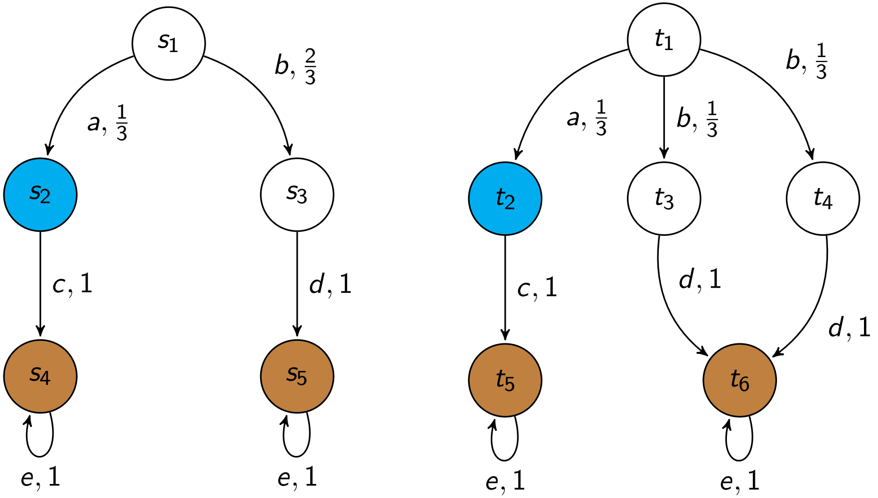
Which states are bisimilar?

An example of probabilistic bisimulations



$$s_4 \sim s_5 \sim t_5 \sim t_6$$

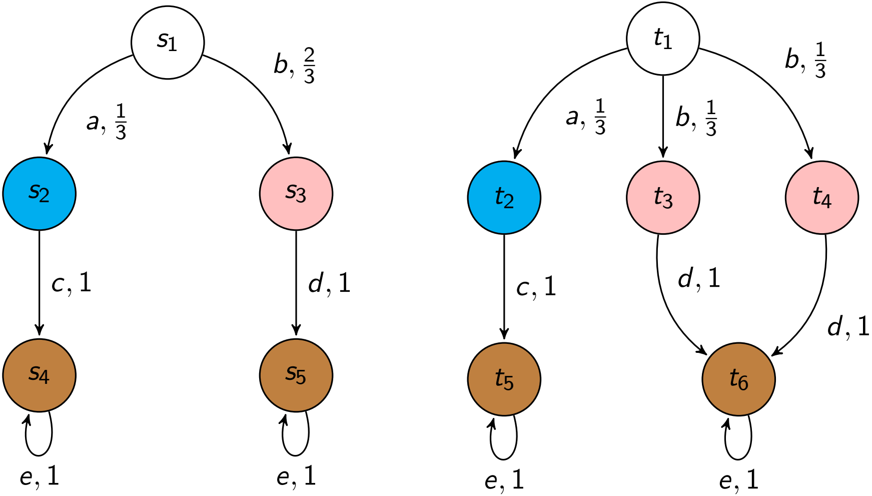
An example of probabilistic bisimulations



$$s_4 \sim s_5 \sim t_5 \sim t_6$$

$$s_2 \sim t_2$$

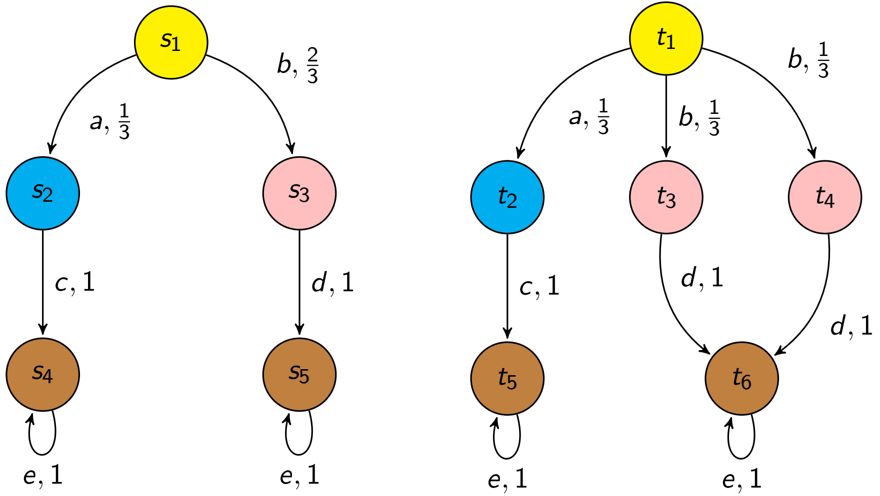
An example of probabilistic bisimulations



$$s_4 \sim s_5 \sim t_5 \sim t_6$$

$$s_2 \sim t_2 \quad s_3 \sim t_3 \sim t_4$$

An example of probabilistic bisimulations



$$s_4 \sim s_5 \sim t_5 \sim t_6$$

$$s_2 \sim t_2 \quad s_3 \sim t_3 \sim t_4 \quad s_1 \sim t_1$$

Bisimulation metrics

- *Bisimulation metrics* are a quantitative analogue of *bisimulation relations*
- Metrics are smooth with respect to changes on the transition probabilities over the state space
- *Bisimulation metrics* can also be used to aggregate states (ϵ-neighborhood)

Bisimulation metrics

Definition (bisimulation metrics)

Let s and s' be two states, the bisimulation metric is a fixed point of of F defined as follows

$$F(d)(s, s') := \max_{a} \left(|R_{sa} - R_{s'a}| + \gamma \mathcal{T}(d, P_{sa}, P_{s'a}) \right)$$

- To compute *bisimulation metrics*, we need to measure the distance between probability distributions
- We can use different metrics to compare probability distributions

Related work on bisimulation metric computation

Bisimulation metric

An iteratively improving algorithm [Ferns et al. 2004]

Optimal value function

Related work on bisimulation metric computation

Bisimulation metric

↓

An iteratively improving algorithm [Ferns et al. 2004]

√ Prove convergence

× Must compare all pairs of states at every iteration

Optimal value function

Related work on bisimulation metric computation

Heuristic algorithm

An on-the-fly technique [Comanici et al. 2012]

Bisimulation metric

An iteratively improving algorithm [Ferns et al. 2004]

$\surd$ Prove convergence

$\times$ Must compare all pairs of states at every iteration

Optimal value function

Related work on bisimulation metric computation

Heuristic algorithm

↓

An on-the-fly technique [Comanici et al. 2012]

$\surd$ Use heuristics to select states and update distances

$\times$ Metric is still over the entire state space

Bisimulation metric

↓

An iteratively improving algorithm [Ferns et al. 2004]

$\surd$ Prove convergence

$\times$ Must compare all pairs of states at every iteration

Optimal value function

Related work on bisimulation metric computation

Heuristic algorithm

↓

Our work: a much more efficient algorithm

↓

Bisimulation metric

↓

An iteratively improving algorithm [Ferns et al. 2004]

$\surd$ Prove convergence

$\times$ Must compare all pairs of states at every iteration

↓

Optimal value function

Related work on bisimulation metric computation

Heuristic algorithm

↓

Our work: a much more efficient algorithm

√ Develop metrics over partitions

√ Provide theoretical guarantees

Bisimulation metric

↓

An iteratively improving algorithm [Ferns et al. 2004]

√ Prove convergence

× Must compare all pairs of states at every iteration

Optimal value function

First contribution: metrics over partitions

- At each time step, we partition the state space into **blocks** based on the *approximate bisimulation*
- The metric is over partitions instead of the entire state space
- We adopt *heuristics* to select certain pairs of **blocks** at each iteration and only update distances between them
- The *heuristics* provides more flexibility in designing strategies for a variety of MDPs

Second contribution: Kantorovich metric over partitions

- We use the *Kantorovich metric* to compare probability distributions
- The metric finds the **optimal** coupling of two probability distributions

Definition (discrete Kantorovich metric)

Let Ω be a discrete state space , μ, ν be two probability measures on Ω, d be a given metric on Ω

$$\mathcal{T}(d, \mu, \nu) := \max \left\{ \left| \sum_{x \in \Omega} h(x)\mu(x) - \sum_{x \in \Omega} h(x)\nu(x) \right| : \|h\|_L \leq 1 \right\}$$

The max is taken over all h s.t. $|h(x) - h(y)| \leq d(x, y)$

Second contribution: Kantorovich metric over partitions

- *Kantorovich metric* can be transformed into transportation problems by Kantorovich-Rubinstein Duality (max $\rightarrow$ min)
- The minimal cost for transforming one histogram into the other is defined as *the Earth Mover's Distance (EMD)*
- We adopt *the Earth Mover's Distance (EMD)* algorithms [Rubner et al., 2000] to find the best coupling of probability distributions
- By using our partition strategy, the algorithm only needs to search couplings over partitions instead of the entire state space

Third contribution: theoretical guarantees

- We formalize the asynchronous partition algorithm in a mathematical framework
- We establish the correspondence between the theoretical characterization and the algorithm

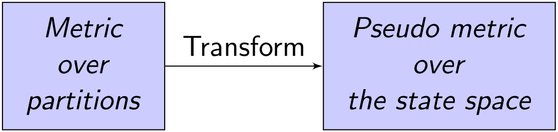
Third contribution: theoretical guarantees

- We formalize the asynchronous partition algorithm in a mathematical framework
- We establish the correspondence between the theoretical characterization and the algorithm

Metric over partitions

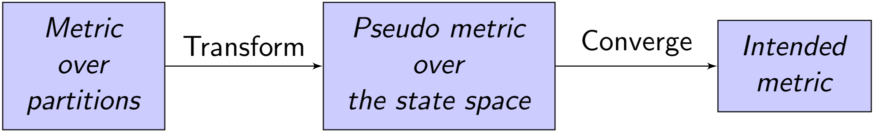
Third contribution: theoretical guarantees

- We formalize the asynchronous partition algorithm in a mathematical framework
- We establish the correspondence between the theoretical characterization and the algorithm
  - We transform the metric over partitions into a pseudo metric over the state space



Third contribution: theoretical guarantees

- We formalize the asynchronous partition algorithm in a mathematical framework
- We establish the correspondence between the theoretical characterization and the algorithm
  - We transform the metric over partitions into a pseudo metric over the state space
  - We prove that the transformed metric converges to the intended bisimulation metric



Fourth contribution: experiments on synthetic data

- We implement the iterative asynchronous partition algorithm described previously
- We run experiments on synthetic data
- The experimental results are consistent with the theory

Future work

- Extend experiments on synthetic data to real data with large state space
- Design elaborate heuristics based on different data to further optimize the algorithm

Thank You