On-the-Fly Exact Computation of Bisimilarity Distances

Giorgio Bacci, **Giovanni Bacci**, Kim G. Larsen, Radu Mardare

Dept. of Computer Science, Aalborg University

Quantitative Models: Expressiveness, Analysis, and New Applications
21 January – Dagstuhl, Germany

Motivations

Probabilistic Systems

- lack of knowledge or inherent nondeterminism
- applied in various contexts (biology, security, games, A.I., ...)

Probabilistic Bisimulation is too fragile

- it only relates states with identical behaviors
- slight changes in quantities $\Longrightarrow$ systems no more bisimilar

Bisimilarity Distances

- measure the behavioral similarity between states
- support approximate reasoning on probabilistic systems

Motivations

Probabilistic Systems

+ lack of knowledge or inherent nondeterminism
+ applied in various contexts (biology, security, games, A.I., . . .)

Probabilistic Bisimulation is too fragile

+ it only relates states with identical behaviors
+ slight changes in quantities $\implies$ systems no more bisimilar

Bisimilarity Distances

+ measure the behavioral similarity between states
+ support approximate reasoning on probabilistic systems

Motivations

Probabilistic Systems

- lack of knowledge or inherent nondeterminism
- applied in various contexts (biology, security, games, A.I., ...)

Probabilistic Bisimulation is too fragile

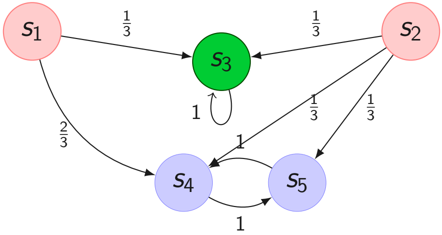
- it only relates states with identical behaviors
- slight changes in quantities $\implies$ systems no more bisimilar

Bisimilarity Distances

- measure the behavioral similarity between states
- support approximate reasoning on probabilistic systems

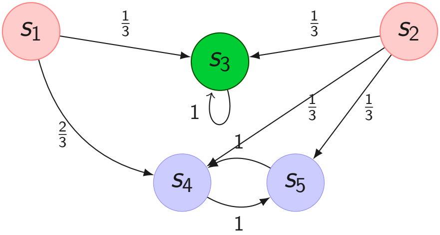
Markov Chains:

Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$



Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$

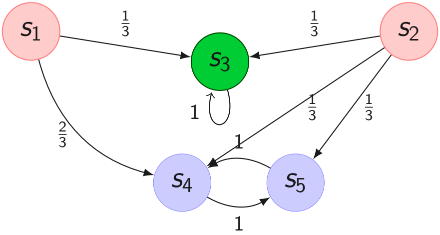
finite set of states



Markov Chains:

Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$

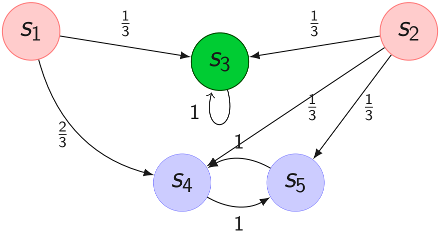
set of labels



Markov Chains:

Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$

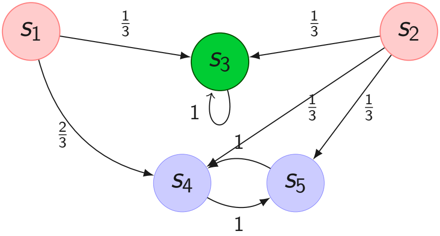
probability transition function
$\pi \colon S \times S \to [0,1]$ s.t. $\forall u \in S.\ \sum_{v \in S} \pi(u,v) = 1$



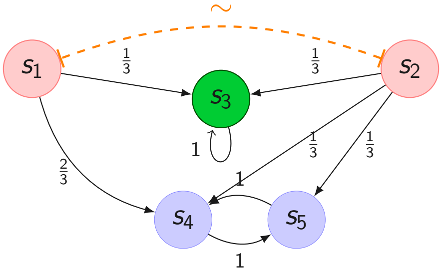
Markov Chains:

Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$

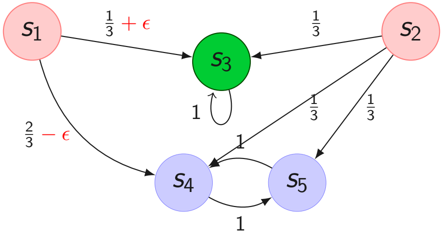
labelling function
$\ell \colon S \to \Sigma$



Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$



Markov Chain: $\mathcal{M} = (S, \Sigma, \pi, \ell)$



From equivalences to distances

Pseudometrics as the quantitative analogue of an equivalence:

Equivalence: $\equiv \subseteq S \times S$		**Pseudometric:** $d\colon S \times S \to \mathbb{R}_{\geq 0}$
$s \equiv s$	$\rightsquigarrow$	$d(s,s) = 0$
$s \equiv t \implies t \equiv s$	$\rightsquigarrow$	$d(s,t) = d(t,s)$
$s \equiv u \wedge u \equiv t \implies s \equiv t$	$\rightsquigarrow$	$d(s,u) + d(u,t) \geq d(s,t)$

Bisimilarity Pseudometric: $d(s,t) = 0 \iff s \sim t$

We consider the λ-discounted bisimilarity distances $\delta_\lambda\colon S \times S \to [0,1]$ proposed by Desharnais et al.

Bisimilarity Pseudometric [van Breugel, Worrel]

Given a parameter $\lambda \in (0,1]$, called discount factor,
the bisimilarity pseudometric δ_λ is the **least fixed point** of

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min\limits_{\omega \in \pi(s,\cdot) \otimes \pi(t,\cdot)} \lambda \sum\limits_{u,v \in S} d(u,v) \cdot \omega(u,v) & \text{otherwise} \end{cases}$$

Bisimilarity Pseudometric [van Breugel, Worrel]

Given a parameter $\lambda \in (0, 1]$, called discount factor,
the bisimilarity pseudometric δ_λ is the **least fixed point** of

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min\limits_{\omega \in \pi(s,\cdot) \otimes \pi(t,\cdot)} \lambda \sum\limits_{u,v \in S} d(u,v) \cdot \omega(u,v) & \text{otherwise} \end{cases}$$

Kantorovich distance
between $\pi(s,\cdot)$ and $\pi(t,\cdot)$

Kantorovich Metric

The distance between $\pi(s,\cdot)$ and $\pi(t,\cdot)$
is the optimal value of a Transportation Problem

$$\min \left\{ \sum_{u,v\in S} d(u,v)\cdot\omega(u,v) \;\middle|\; \begin{array}{l} \forall u\in S\ \sum_{v\in S}\omega(u,v)=\pi(s,u) \\ \forall v\in S\ \sum_{u\in S}\omega(u,v)=\pi(t,v) \end{array} \right\}$$

ω can be understood as transportation of $\pi(s,\cdot)$ to $\pi(t,\cdot)$.

(s,t)	t_1	t_2	t_3	
s_1	$\omega(s_1,t_1)$	$\omega(s_1,t_2)$	$\omega(s_1,t_3)$	$\pi(s,s_1)$
s_2	$\omega(s_2,t_1)$	$\omega(s_2,t_2)$	$\omega(s_2,t_3)$	$\pi(s,s_2)$
s_3	$\omega(s_3,t_1)$	$\omega(s_3,t_2)$	$\omega(s_3,t_3)$	$\pi(s,s_3)$
s_4	$\omega(s_4,t_1)$	$\omega(s_4,t_2)$	$\omega(s_4,t_3)$	$\pi(s,s_4)$
	$\pi(t,t_1)$	$\pi(t,t_2)$	$\pi(t,t_3)$	

Kantorovich Metric

The distance between $\pi(s,\cdot)$ and $\pi(t,\cdot)$
is the optimal value of a Transportation Problem

$$\min \left\{ \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \;\middle|\; \begin{array}{l} \forall u \in S \; \sum_{v \in S} \omega(u,v) = \pi(s,u) \\ \forall v \in S \; \sum_{u \in S} \omega(u,v) = \pi(t,v) \end{array} \right\}$$

matching

$\omega \in \pi(s,\cdot) \otimes \pi(t,\cdot)$

ω can be understood as transportation of $\pi(s,\cdot)$ to $\pi(t,\cdot)$.

(s,t)	t_1	t_2	t_3	
s_1	$\omega(s_1,t_1)$	$\omega(s_1,t_2)$	$\omega(s_1,t_3)$	$\pi(s,s_1)$
s_2	$\omega(s_2,t_1)$	$\omega(s_2,t_2)$	$\omega(s_2,t_3)$	$\pi(s,s_2)$
s_3	$\omega(s_3,t_1)$	$\omega(s_3,t_2)$	$\omega(s_3,t_3)$	$\pi(s,s_3)$
s_4	$\omega(s_4,t_1)$	$\omega(s_4,t_2)$	$\omega(s_4,t_3)$	$\pi(s,s_4)$
	$\pi(t,t_1)$	$\pi(t,t_2)$	$\pi(t,t_3)$	

Kantorovich Metric

The distance between $\pi(s,\cdot)$ and $\pi(t,\cdot)$
is the optimal value of a Transportation Problem

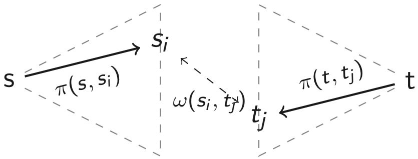
$$\min \left\{ \sum_{u,v\in S} d(u,v)\cdot\omega(u,v) \;\middle|\; \begin{array}{l} \forall u \in S\ \sum_{v\in S}\omega(u,v) = \pi(s,u) \\ \forall v \in S\ \sum_{u\in S}\omega(u,v) = \pi(t,v) \end{array} \right\}$$

matching

$\omega \in \pi(s,\cdot)\otimes\pi(t,\cdot)$

ω can be understood as transportation of $\pi(s,\cdot)$ to $\pi(t,\cdot)$.

(s,t)	t_1	t_2	t_3	
s_1	$\omega(s_1,t_1)$	$\omega(s_1,t_2)$	$\omega(s_1,t_3)$	$\pi(s,s_1)$
s_2	$\omega(s_2,t_1)$	$\omega(s_2,t_2)$	$\omega(s_2,t_3)$	$\pi(s,s_2)$
s_3	$\omega(s_3,t_1)$	$\omega(s_3,t_2)$	$\omega(s_3,t_3)$	$\pi(s,s_3)$
s_4	$\omega(s_4,t_1)$	$\omega(s_4,t_2)$	$\omega(s_4,t_3)$	$\pi(s,s_4)$
	$\pi(t,t_1)$	$\pi(t,t_2)$	$\pi(t,t_3)$	



Existing Methods for computing δ_λ

Iterative Method – Approximated:

$$\mathbf{0} \sqsubseteq \Delta_\lambda(0) \sqsubseteq \Delta_\lambda(\Delta_\lambda(0)) \cdots \sqsubseteq \Delta_\lambda^n(0) \sqsubseteq \cdots \sqsubseteq \delta_\lambda$$

Linear programming – Exact: [Chen et al. – FoSSaCS'12]

+ solution of a linear program with exponentially many constraints
+ they provided a polynomial separation algorithm

$\Longrightarrow$ ellipsoid method

proved that δ_λ can be computed in **polynomial time**

Existing Methods for computing δ_λ

Iterative Method – Approximated:

$$\mathbf{0} \sqsubseteq \Delta_\lambda(0) \sqsubseteq \Delta_\lambda(\Delta_\lambda(0)) \cdots \sqsubseteq \Delta_\lambda^n(0) \sqsubseteq \cdots \sqsubseteq \delta_\lambda$$

Linear programming – Exact: [Chen et al. – FoSSaCS'12]

+ solution of a linear program with exponentially many constraints
+ they provided a polynomial separation algorithm

$\Longrightarrow$ ellipsoid method

proved that δ_λ can be computed in **polynomial time**

On-the-fly approach

All existing methods requires to explore the entire state space

What if we only need $\delta_\lambda(s, t)$?

can we skip (part or all) the computation of some other pairs?

we propose an **on-the-fly** strategy:

- + lazy exploration of $\mathcal{M}$ (only where and when is needed)
- + saving in computational cost both in time and space

On-the-fly approach

All existing methods requires to explore the entire state space

What if we only need $\delta_\lambda(s, t)$?

can we skip (part or all) the computation of some other pairs?

we propose an **on-the-fly** strategy:

- + lazy exploration of $\mathcal{M}$ (only where and when is needed)
- + saving in computational cost both in time and space

Alternative characterization

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min_{\omega \in \pi(s,\cdot)\otimes\pi(t,\cdot)} \lambda \sum_{u,v\in S} d(u,v)\cdot\omega(u,v) & \text{otherwise} \end{cases}$$

Coupling: $\mathcal{C} = \{\omega_{s,t} \in \pi(s,\cdot) \otimes \pi(t,\cdot)\}_{s,t\in S}$

$$x_{s,t} = 1 \qquad \ell(s) \neq \ell(t)$$

$$x_{s,t} = \lambda \sum_{u,v\in S} x_{u,v} \cdot \omega_{s,t}(u,v) \qquad \ell(s) = \ell(t)$$

we call discrepancy, $\gamma_\lambda^{\mathcal{C}}$, the least solution of the linear system

Theorem: Chen, van Breugel and Worrel – FoSSaCS'12

$\delta_1 = \min\{\gamma_1^{\mathcal{C}} \mid \mathcal{C} \text{ coupling for } \mathcal{M}\}$

Alternative characterization

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min\limits_{\omega \in \pi(s,\cdot)\otimes\pi(t,\cdot)} \lambda \sum\limits_{u,v\in S} d(u,v)\cdot\omega(u,v) & \text{otherwise} \end{cases}$$

Coupling: $\mathcal{C} = \{\omega_{s,t} \in \pi(s,\cdot) \otimes \pi(t,\cdot)\}_{s,t\in S}$

$$x_{s,t} = 1 \qquad \ell(s) \neq \ell(t)$$

$$x_{s,t} = \lambda \textstyle\sum_{u,v\in S} x_{u,v} \cdot \omega_{s,t}(u,v) \qquad \ell(s) = \ell(t)$$

we call discrepancy, $\gamma_\lambda^{\mathcal{C}}$, the least solution of the linear system

Theorem: Chen, van Breugel and Worrel – FoSSaCS'12

$\delta_1 = \min\{\gamma_1^{\mathcal{C}} \mid \mathcal{C} \text{ coupling for } \mathcal{M}\}$

Alternative characterization

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min_{\omega \in \pi(s,\cdot) \otimes \pi(t,\cdot)} \lambda \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) & \text{otherwise} \end{cases}$$

Coupling: $\mathcal{C} = \{\omega_{s,t} \in \pi(s,\cdot) \otimes \pi(t,\cdot)\}_{s,t \in S}$

$$x_{s,t} = 1 \qquad \ell(s) \neq \ell(t)$$
$$x_{s,t} = \lambda \sum_{u,v \in S} x_{u,v} \cdot \omega_{s,t}(u,v) \qquad \ell(s) = \ell(t)$$

we call discrepancy, $\gamma_\lambda^{\mathcal{C}}$, the least solution of the linear system

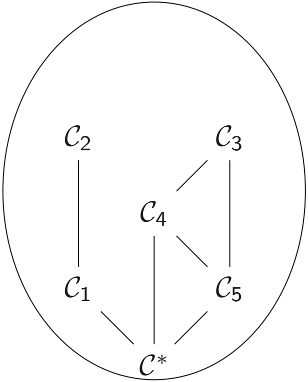
Theorem: Bacci[2], Larsen and Mardare – TACAS'13

$\delta_\lambda = \min\{\gamma_\lambda^{\mathcal{C}} \mid \mathcal{C} \text{ coupling for } \mathcal{M}\}$ for all $\lambda \in (0,1]$.

On-the-fly strategy

[Bacci[2], Larsen, Mardare – TACAS'13]

$$\mathcal{C}_1 \trianglelefteq_\lambda \mathcal{C}_2 \iff \gamma_\lambda^{\mathcal{C}_1} \sqsubseteq \gamma_\lambda^{\mathcal{C}_2}$$



Greedy strategy

Moving Criterion:
$\mathcal{C}_i = \{\dots, \omega_{u,v}, \dots\}$
$\omega_{u,v}$ not opt. w.r.t. $TP(\gamma_\lambda^{\mathcal{C}_i}, \pi(u,\cdot), \pi(v,\cdot))$

Improvement:
$\mathcal{C}_{i+1} = \{\dots, \omega^*, \dots\}$, where
ω^* optimal sol. for $TP(\gamma_\lambda^{\mathcal{C}_i}, \pi(u,\cdot), \pi(v,\cdot))$

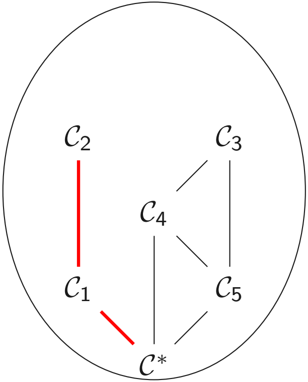
Theorem

+ each step ensures $\mathcal{C}_{i+1} \triangleleft_\lambda \mathcal{C}_i$
+ moving criterion holds until $\gamma_\lambda^{\mathcal{C}_i} \neq \delta_\lambda$
+ the method always terminates

On-the-fly strategy

[Bacci[2], Larsen, Mardare – TACAS'13]

$$\mathcal{C}_1 \trianglelefteq_\lambda \mathcal{C}_2 \iff \gamma_\lambda^{\mathcal{C}_1} \sqsubseteq \gamma_\lambda^{\mathcal{C}_2}$$



Greedy strategy

Moving Criterion:

$\mathcal{C}_i = \{\dots, \omega_{u,v}, \dots\}$
$\omega_{u,v}$ not opt. w.r.t. $TP(\gamma_\lambda^{\mathcal{C}_i}, \pi(u,\cdot), \pi(v,\cdot))$

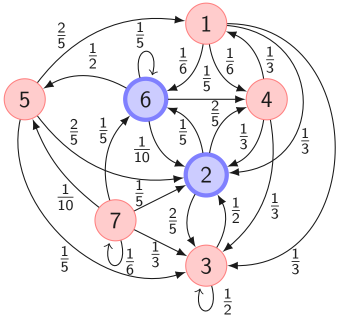
Improvement:

$\mathcal{C}_{i+1} = \{\dots, \omega^*, \dots\}$, where
ω^* optimal sol. for $TP(\gamma_\lambda^{\mathcal{C}_i}, \pi(u,\cdot), \pi(v,\cdot))$

Theorem

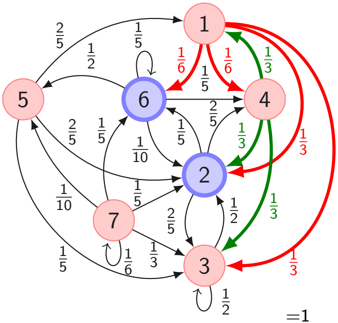
+ each step ensures $\mathcal{C}_{i+1} \triangleleft_\lambda \mathcal{C}_i$
+ moving criterion holds until $\gamma_\lambda^{\mathcal{C}_i} \neq \delta_\lambda$
+ the method always terminates

Goal: compute $\delta_1(1, 4)$



Solution:

Goal: compute $\delta_1(1,4)$

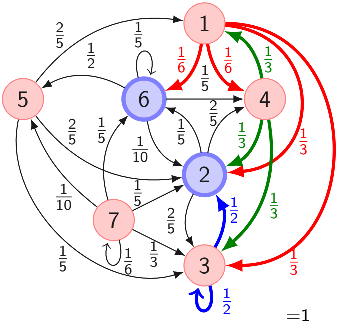


(1,4)	1	2	3	
2	$\frac{1}{3}$			$\frac{1}{3}$
3		$\frac{1}{3}$		$\frac{1}{3}$
4			$\frac{1}{6}$	$\frac{1}{6}$
6			$\frac{1}{6}$	$\frac{1}{6}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

$$d(1,4) = \frac{1}{3} \cdot \overbrace{d(1,2)}^{=1} + \frac{1}{3} \cdot \overbrace{d(2,3)}^{=1} + \frac{1}{6} \cdot d(3,4) + \frac{1}{6} \cdot \overbrace{d(3,6)}^{=1}$$
$$= \frac{1}{6} \cdot d(3,4) + \frac{5}{6}$$

Solution:

Goal: compute $\delta_1(1,4)$



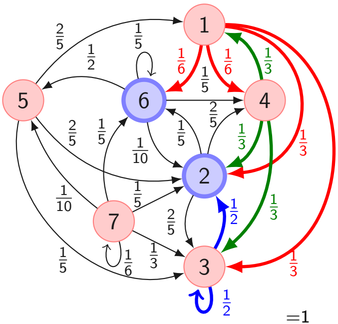
(1,4)	1	2	3	
2	$\frac{1}{3}$			$\frac{1}{3}$
3		$\frac{1}{3}$		$\frac{1}{3}$
4			$\frac{1}{6}$	$\frac{1}{6}$
6			$\frac{1}{6}$	$\frac{1}{6}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

(3,4)	1	2	3	
2	$\frac{1}{3}$	$\frac{1}{6}$		$\frac{1}{2}$
3		$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

$$d(1,4) = \frac{1}{3}\cdot\overbrace{d(1,2)}^{=1} + \frac{1}{3}\cdot\overbrace{d(2,3)}^{=1} + \frac{1}{6}\cdot d(3,4) + \frac{1}{6}\cdot\overbrace{d(3,6)}^{=1}$$
$$= \frac{1}{6}\cdot d(3,4) + \frac{5}{6}$$
$$d(3,4) = \frac{1}{3}\cdot\overbrace{d(1,2)}^{=1} + \frac{1}{6}\cdot\overbrace{d(2,2)}^{=0} + \frac{1}{6}\cdot\overbrace{d(2,3)}^{=1} + \frac{1}{3}\cdot\overbrace{d(3,3)}^{=0}$$
$$= \frac{1}{2}$$

Solution:

Goal: compute $\delta_1(1,4)$



(1,4)	1	2	3	
2	$\frac{1}{3}$			$\frac{1}{3}$
3		$\frac{1}{3}$		$\frac{1}{3}$
4			$\frac{1}{6}$	$\frac{1}{6}$
6			$\frac{1}{6}$	$\frac{1}{6}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

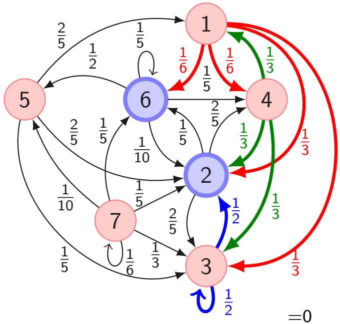
(3,4)	1	2	3	
2	$\frac{1}{3}$	$\frac{1}{6}$		$\frac{1}{2}$
3		$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

$$d(1,4) = \frac{1}{3}\cdot\overbrace{d(1,2)}^{=1} + \frac{1}{3}\cdot\overbrace{d(2,3)}^{=1} + \frac{1}{6}\cdot d(3,4) + \frac{1}{6}\cdot\overbrace{d(3,6)}^{=1}$$
$$= \frac{1}{6}\cdot d(3,4) + \frac{5}{6}$$

$$d(3,4) = \frac{1}{3}\cdot\overbrace{d(1,2)}^{=1} + \frac{1}{6}\cdot\overbrace{d(2,2)}^{=0} + \frac{1}{6}\cdot\overbrace{d(2,3)}^{=1} + \frac{1}{3}\cdot\overbrace{d(3,3)}^{=0}$$
$$= \frac{1}{2}$$

Solution: $d(1,4) = \frac{11}{12}$ and $d(3,4) = \frac{1}{2}$

Goal: compute $\delta_1(1,4)$

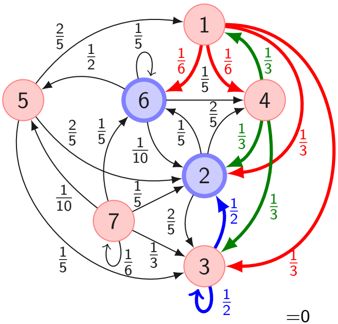


(1,4)	1	2	3	
2		$\frac{1}{3}$		$\frac{1}{3}$
3			$\frac{1}{3}$	$\frac{1}{3}$
4	$\frac{1}{6}$			$\frac{1}{6}$
6	$\frac{1}{6}$			$\frac{1}{6}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

$$\begin{aligned} d(1,4) &= \frac{1}{3} \cdot \overbrace{d(2,2)}^{=0} + \frac{1}{3} \cdot \overbrace{d(3,3)}^{=0} + \frac{1}{6} \cdot d(1,4) + \frac{1}{6} \cdot \overbrace{d(1,6)}^{=1} \\ &= \frac{1}{6} \cdot d(1,4) + \frac{1}{6} \end{aligned}$$

Solution:

Goal: compute $\delta_1(1,4)$



(1,4)	1	2	3	
2		$\frac{1}{3}$		$\frac{1}{3}$
3			$\frac{1}{3}$	$\frac{1}{3}$
4	$\frac{1}{6}$			$\frac{1}{6}$
6	$\frac{1}{6}$			$\frac{1}{6}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

$$
\begin{aligned}
d(1,4) &= \frac{1}{3}\cdot\overbrace{d(2,2)}^{=0}+\frac{1}{3}\cdot\overbrace{d(3,3)}^{=0}+\frac{1}{6}\cdot d(1,4)+\frac{1}{6}\cdot\overbrace{d(1,6)}^{=1}\\
&= \frac{1}{6}\cdot d(1,4)+\frac{1}{6}
\end{aligned}
$$

Solution: $d(1,4) = \frac{1}{5}$

Empirical Results (all-pairs)

# States	On-the-Fly (exact)		Iterating (approximated)			Approx. Error*
	Time (s)	# TPs	Time (s)	# Iterations	# TPs	
5	0.019	1.191	0.0389	1.733	26.733	0.139
6	0.059	3.046	0.092	1.826	38.133	0.146
7	0.138	6.011	0.204	2.194	61.728	0.122
8	0.255	8.561	0.364	2.304	83.028	0.117
9	0.499	12.042	0.673	2.579	114.729	0.111
10	1.003	18.733	1.272	3.111	174.363	0.094
11	2.159	25.973	2.661	3.556	239.557	0.096
12	4.642	34.797	5.522	4.042	318.606	0.086
13	6.735	39.958	8.061	4.633	421.675	0.097
14	6.336	38.005	7.188	4.914	593.981	0.118
17	11.261	47.014	12.805	5.885	908.61	0.132
19	26.635	61.171	29.654	6.961	1328.60	0.140
20	34.379	66.457	38.206	7.538	1597.92	0.142

(*) $\epsilon = \max_{s,t \in S} \delta_\lambda(s,t) - d(s,t)$

Empirical Results (single-pair)

# States	out-degree = 3		2 ≤ out-degree ≤ # States	
	Time (s)	# TPs	Time (s)	# TPs
5	0.006	0.273	0.012	0.657
6	0.012	0.549	0.031	1.667
7	0.017	0.981	0.088	3.677
8	0.025	1.346	0.164	5.301
9	0.026	1.291	0.394	8.169
10	0.058	2.038	1.112	13.096
11	0.077	1.827	2.220	18.723
12	0.043	1.620	4.940	26.096
13	0.060	1.882	10.360	35.174
14	0.089	2.794	20.123	46.077

Concluding remarks

we have seen

- on-the-fly algorithm for bisimulation metrics
  - exact
  - avoids entire exploration of the state space
- performs, on average, better than other proposals

recent achievements:

- Computing Behavioral Distances, Compositionally [MFCS'13]
- The BisimDist Library: Efficient Computation of Bisimilarity Distances for Markovian Models [QEST'13]

http://people.cs.aau.dk/giovbacci/tools.html

Ongoing & future work

- similar techniques in the timed context (e.g. CTMCs, TA)
- relating with known "trace-based" distances (e.g. TV, L_p)