Explainability of Probabilistic Bisimilarity Distances in Labelled Markov Chains

Amgad Rady, Fank van Breugel

Chris Römmer (Presentation)

2025-11-25

Outline

Labelled Markov Chains

What is a Labelled Markov Chain?

Labelled Markov Chains

What is a Labelled Markov Chain?

Definition

A Labelled Markov Chain is a tuple $\mathcal{M} = (S, L, \tau, l)$ consisting of

- a finite set S of states
- a finite set L of labels
- a transition probability function $\tau : S \to \mathcal{D}_{\mathbb{Q}}(S)$
- a labeling function $l : S \to L$

$\mathcal{D}_{\mathbb{Q}}(X) = \{\mu : X \to \mathbb{Q} \mid \mu \text{ is a probability distribution}\}$

Probabilistic Bisimilarity

Definition

For all $\mu, \nu \in \mathcal{D}_{\mathbb{R}}(S)$, the set $\Omega_{\mathbb{R}}(\mu, \nu)$ is defined by

$$\Omega_{\mathbb{R}}(\mu, \nu) = \{\omega \in \mathcal{D}_{\mathbb{R}}(S \times S) \mid \forall s \in S : \omega(s, S) = \mu(s) \wedge \omega(S, s) = \nu(s)\}$$

Probabilistic Bisimilarity

Definition

For all $\mu, \nu \in \mathcal{D}_{\mathbb{R}}(S)$, the set $\Omega_{\mathbb{R}}(\mu, \nu)$ is defined by

$$\Omega_{\mathbb{R}}(\mu, \nu) = \{\omega \in \mathcal{D}_{\mathbb{R}}(S \times S) \mid \forall s \in S : \omega(s, S) = \mu(s) \wedge \omega(S, s) = \nu(s)\}$$

Definition

A relation $R \subset S \times S$ is a probabilistic bisimulation if for all $(s, t) \in R$, $l(s) = l(t)$ and there exists $\omega \in \Omega_{\mathbb{R}}(\tau(s), \tau(t))$ with $support(\omega) \subseteq R$.
$s \sim t \Leftrightarrow (s, t) \in R$ for some probabilistic bisimulation R.

Probabilistic Bisimilarity Distances

Definition

The function $\Delta : (S \times S \to [0,1]) \to (S \times S \to [0,1])$ is defined by

$$\Delta(d)(s,t) = \begin{cases} 0 & \text{if } s \sim t \\ 1 & \text{if } l(s) \neq l(t) \\ \inf\limits_{\omega \in \Omega_{\mathbb{R}}(\tau(s),\tau(t))} \omega \cdot d & \text{otherwise} \end{cases}$$

Notation: $\omega \cdot d = \sum_{u,v \in S} \omega(u,v) \cdot d(u,v)$

Fixpoint of Δ

Δ has a least fixed point δ, i.e. $\Delta(\delta) = \delta$. $\delta(s, t)$ is the *probabilistic bisimilarity distance* of s and t.

Fixpoint of Δ

Δ has a least fixed point δ, i.e. $\Delta(\delta) = \delta$. $\delta(s,t)$ is the *probabilistic bisimilarity distance* of s and t.

Definition

For $n \geq 0$ the function $\delta_n : S \times S \to [0,1]$ is defined by

$$\delta_n = \begin{cases} 0 & \text{if } n = 0 \\ \Delta(\delta_{n-1}) & \text{otherwise} \end{cases}$$

Fixpoint of Δ

Δ has a least fixed point δ, i.e. $\Delta(\delta) = \delta$. $\delta(s,t)$ is the *probabilistic bisimilarity distance* of s and t.

Definition

For $n \geq 0$ the function $\delta_n : S \times S \to [0,1]$ is defined by

$$\delta_n = \begin{cases} 0 & \text{if } n = 0 \\ \Delta(\delta_{n-1}) & \text{otherwise} \end{cases}$$

Theorem

(Kleene's fixed point theorem)

$$\lim_{n\to\infty} \delta_n \to \delta$$

The Logic $\mathcal{L}_\neg$ - Grammar

Definition

The logic $\mathcal{L}_\neg$ is defined by the grammar

$$\varphi, \psi ::= a \mid \bigcirc\varphi \mid \neg\varphi \mid \varphi \ominus q \mid \varphi \vee \psi$$

with $a \in L$ and $q \in \mathbb{Q} \cap [0, 1]$

The Logic $\mathcal{L}_\neg$ - Semantics

Definition

The semantics of $\mathcal{L}_\neg$ are defined as follows:

- $[\![a]\!](s) = \begin{cases} 1 & \text{if } l(s) = a \\ 0 & \text{otherwise} \end{cases}$
- $[\![\bigcirc\varphi]\!](s) = [\![\varphi]\!] \cdot \tau(s)$
- $[\![\neg\varphi]\!](s) = 1 - [\![\varphi]\!](s)$
- $[\![\varphi \ominus q]\!](s) = \max\{[\![\varphi]\!](s) - q, 0\}$
- $[\![\varphi \vee \psi]\!](s) = \max\{[\![\varphi]\!](s), [\![\psi]\!](s)\}$

The Logic $\mathcal{L}$

Definition

The logic $\mathcal{L}$ is defined by the grammar

$$\varphi, \psi ::= a \mid \bigcirc\varphi \mid \varphi \ominus q \mid \varphi \oplus q \mid \varphi \vee \psi \mid \varphi \wedge \psi$$

with $a \in L$ and $q \in \mathbb{Q} \cap [0, 1]$

The Logic $\mathcal{L}$

Definition

The logic $\mathcal{L}$ is defined by the grammar

$$\varphi, \psi ::= a \mid \bigcirc\varphi \mid \varphi \ominus q \mid \varphi \oplus q \mid \varphi \vee \psi \mid \varphi \wedge \psi$$

with $a \in L$ and $q \in \mathbb{Q} \cap [0, 1]$

- $[\![\varphi \oplus \psi]\!](s) = \min\{[\![\varphi]\!](s) + q, 1\}$
- $[\![\varphi \wedge \psi]\!](s) = \min\{[\![\varphi]\!](s), [\![\psi]\!](s)\}$

The Logic $\mathcal{L}$

Definition

The logic $\mathcal{L}$ is defined by the grammar

$$\varphi, \psi ::= a \mid \bigcirc\varphi \mid \varphi \ominus q \mid \varphi \oplus q \mid \varphi \vee \psi \mid \varphi \wedge \psi$$

with $a \in L$ and $q \in \mathbb{Q} \cap [0, 1]$

- $[\![\varphi \oplus \psi]\!](s) = \min\{[\![\varphi]\!](s) + q, 1\}$
- $[\![\varphi \wedge \psi]\!](s) = \min\{[\![\varphi]\!](s), [\![\psi]\!](s)\}$

$\mathcal{L}$ can be fully expressed by $\mathcal{L}_\neg$

Distinguishing formula

Definition (Distinguishing formula)

A formula φ_{st} is called a distinguishing formula, if

$$\delta(s,t) = [\![\varphi_{st}]\!](s) - [\![\varphi_{st}]\!](t)$$

Distinguishing formula

Definition (Distinguishing formula)

A formula φ_{st} is called a distinguishing formula, if

$$\delta(s, t) = [\![\varphi_{st}]\!](s) - [\![\varphi_{st}]\!](t)$$

Question: Does φ_{st} always exist?

Distinguishing formula

Definition (Distinguishing formula)

A formula φ_{st} is called a distinguishing formula, if

$$\delta(s,t) = [\![\varphi_{st}]\!](s) - [\![\varphi_{st}]\!](t)$$

Question: Does φ_{st} always exist? **Answer:** No

Distinguishing formula

Definition (Distinguishing formula)

A formula φ_{st} is called a distinguishing formula, if

$$\delta(s, t) = [\![\varphi_{st}]\!](s) - [\![\varphi_{st}]\!](t)$$

Question: Does φ_{st} always exist? **Answer:** No

But $\forall s, t \in S$:

$$\delta(s, t) = \sup_{\varphi \in \mathcal{L}} [\![\varphi]\!](s) - [\![\varphi]\!](t)$$

Convex polytope of non-expansive functions

Definition

A function $f \in S \to [0, 1]$ is non-expansive if for all $s, t \in S$:

$$|f(s) - f(t)| \leq \delta_n(s, t)$$

We denote the set of all non-expansive functions by $(S, \delta_n) \hookrightarrow [0, 1]$.

Convex polytope of non-expansive functions

Definition

A function $f \in S \to [0,1]$ is non-expansive if for all $s, t \in S$:

$$|f(s) - f(t)| \leq \delta_n(s,t)$$

We denote the set of all non-expansive functions by $(S, \delta_n) \hookrightarrow [0,1]$.

Theorem

For all $s, t \in S$ with $s \not\sim t$ and $l(s) = l(t)$ and $n \geq 0$ there exists $f^n_{st} \in (S, \delta_n) \hookrightarrow [0,1]$ such that

$$\delta_{n+1}(s,t) = f^n_{st} \cdot (\tau(s) - \tau(t))$$

Proof.

$$
\begin{aligned}
&\delta_{n+1}(s,t) \\
&= \inf_{\omega \in \Omega_{\mathbb{R}}(\tau(s),\tau(t))} \omega \cdot \delta_n && \text{Definition} \\
&= \sup_{f \in (S,\delta_n) \hookrightarrow [0,1]} f \cdot (\tau(s) - \tau(t)) && \text{Kanotorich-Rubinstein} \\
&= \max_{f \in V((S,\delta_n) \hookrightarrow [0,1])} f \cdot (\tau(s) - \tau(t)) && \text{Convex polytope}
\end{aligned}
$$

□

Expressing f^n_{st} as a formula

Assume f^n_{st} can be expressed by $[\![\psi^n_{st}]\!]$:

$$\begin{aligned}
[\![(\bigcirc\psi^n_{st}) \ominus (f^n_{st} \cdot \tau(t))]\!](s) &= \max\{([\![\psi^n_{st}]\!] \cdot \tau(s)) - (f^n_{st} \cdot \tau(t)), 0\} \\
&= \max\{(f^n_{st} \cdot \tau(s)) - (f^n_{st} \cdot \tau(t)), 0\} \\
&= \max\{f^n_{st} \cdot (\tau(s) - \tau(t)), 0\} \\
&= \max\{\delta_{n+1}(s,t), 0\} \\
&= \delta_{n+1}(s,t)
\end{aligned}$$

Expressing f_{st}^n as a formula

Assume f_{st}^n can be expressed by $[\![\psi_{st}^n]\!]$:

$$
\begin{aligned}
[\![(\bigcirc\psi_{st}^n) \ominus (f_{st}^n \cdot \tau(t))]\!](s) &= \max\{([\![\psi_{st}^n]\!] \cdot \tau(s)) - (f_{st}^n \cdot \tau(t)), 0\} \\
&= \max\{(f_{st}^n \cdot \tau(s)) - (f_{st}^n \cdot \tau(t)), 0\} \\
&= \max\{f_{st}^n \cdot (\tau(s) - \tau(t)), 0\} \\
&= \max\{\delta_{n+1}(s,t), 0\} \\
&= \delta_{n+1}(s,t)
\end{aligned}
$$

$$
[\![\psi_{st}^n]\!] = [\![\bigwedge_{u\in S} \bigvee_{v\in S} \psi_{stuv}^n]\!] = f_{st}^n
$$

Construction of φ^n_{st}

For all $s, t \in S$ and $n \geq 2$:

$$\varphi^n_{st} = \begin{cases} \mathit{false}, & \text{if } s \sim t \\ l(s), & \text{if } l(s) \neq l(t) \\ (\bigcirc \psi^{n-1}_{st}) \ominus (f^{n-1}_{st} \cdot \tau(t)) & \text{otherwise} \end{cases}$$

Construction of φ_{st}^{n}

For all $s, t \in S$ and $n \geq 2$:

$$\varphi_{st}^{n} = \begin{cases} \mathit{false}, & \text{if } s \sim t \\ l(s), & \text{if } l(s) \neq l(t) \\ (\bigcirc \psi_{st}^{n-1}) \ominus (f_{st}^{n-1} \cdot \tau(t)) & \text{otherwise} \end{cases}$$

$$\psi_{st}^{n} = \bigwedge_{u \in S} \bigvee_{v \in S} \psi_{stuv}^{n}$$

Construction of φ^n_{st}

For all $s, t \in S$ and $n \geq 2$:

$$\varphi^n_{st} = \begin{cases} \mathit{false}, & \text{if } s \sim t \\ l(s), & \text{if } l(s) \neq l(t) \\ (\bigcirc \psi^{n-1}_{st}) \ominus (f^{n-1}_{st} \cdot \tau(t)) & \text{otherwise} \end{cases}$$

$$\psi^n_{st} = \bigwedge_{u \in S} \bigvee_{v \in S} \psi^n_{stuv}$$

$$\psi^n_{stuv} = \begin{cases} \mathit{false} \oplus f^n_{st}(u) & \text{if } f^n_{st}(u) = f^n_{st}(v) \\ (\varphi^n_{uv} \ominus (\delta_n(u,v) - (f^n_{st}(u) - f^n_{st}(v)))) \oplus f^n_{st}(v) & \text{if } f^n_{st}(u) > f^n_{st}(v) \\ (\varphi^n_{vu} \ominus (\delta_n(u,v) - (f^n_{st}(v) - f^n_{st}(u)))) \oplus f^n_{st}(u) & \text{otherwise} \end{cases}$$

Computing f_{st}^n

Data: $d \in S \times S \to \mathbb{Q} \cap [0,1], \mu, \nu \in \mathcal{D}_Q(S)$
Result: $\arg\max_{f \in (S,d) \hookrightarrow [0,1]} f \cdot (\mu - \nu)$
$d_{\mu\nu} = \inf_{\omega \in \Omega_{\mathbb{R}}(\mu,\nu)} \omega \cdot d$
$f_{\mu\nu} = \text{vertex of } \{f \in (S,d) \hookrightarrow [0,1] \mid f \cdot (\mu - \nu) = d_{\mu\nu}\}$
return $f_{\mu\nu}$

Algorithm 1: FindVertex(d, μ, ν)

Algorithm

Data: $\tau : S \to \mathcal{D}_{\mathbb{Q}}(S), l : S \to L, s, t \in S, N \geq 0$
Result: $(\varphi_{st}^n)_{n=0}^N \forall s, t \in S$
`// Initialization`
declare $formula[|S|][|S|][N+1] = \varphi_{st}^1$
$formula[|S|][|S|][0] = false$
declare $distance[|S|][|S|] = 0[|S|][|S|]$
declare $function[|S|][|S|][|S|] = 0[|S|][|S|][|S|]$

Algorithm 2: ExplainDistances

```
for n = 1 to N do
    UpdateDistances(distance, function, τ, l)
    for s, t ∈ S do
        if s ≁ t and l(s) = l(t) then
            function[s][t] = FindVertex(distance, τ(s), τ(t))
            declare ψ^n_st
            for u, v ∈ S do
                declare ψ^n_stuv =
                  ComputeSubformula(distance[u][v], function[s][t], formula, n)

                ψ^n_st.insert(ψ^n_stuv)
            end
            formula[s][t][n + 1] = (◯ψ^n_st) ⊖ (function[s][t] · τ(t))
        end
    end
end
```