Measuring Structural Similarities in Finite MDPs

Hao Wang[1*], **Shaokang Dong**[2] and **Ling Shao**[1]
[1]Inception Institute of Artificial Intelligence, UAE
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
[1]{hao.wang, ling.shao}@inceptioniai.org
[2]shaokangdong@smail.nju.edu.cn

Abstract

In this paper, we investigate the structural similarities within a finite Markov decision process (MDP). We view a finite MDP as a heterogeneous directed bipartite graph and propose novel measures for the state and action similarities, in a mutually reinforced manner. We prove that the state similarity is a metric and the action similarity is a pseudometric. We also establish the connection between the proposed similarity measures and the optimal values of the MDP. Extensive experiments show that the proposed measures are effective.

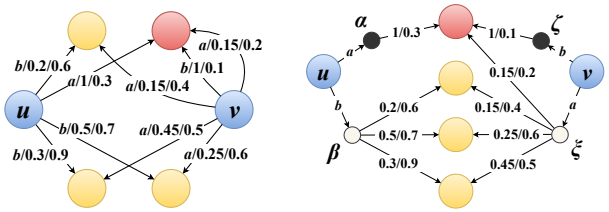
1 Introduction

The Markov decision process (MDP) is a useful mathematical model to support decision making, which has practical applications in many areas, such as intelligent control systems, finance, energy management, online advertising, etc. [Cai *et al.*, 2017; Han *et al.*, 2016] A finite MDP models the interaction between an agent and the outside environment. The environment is described as a set of discrete states, where, on each state, the agent has a set of available actions. By observing the current state and deciding which action to take, the agent may change the state of the environment. Driven by a properly designed reward scheme, the agent may thus develop a good policy for making wise decisions in the environment.

In this paper, we investigate how to measure the state and action similarities in a finite MDP. Such similarities are important as they may provide a principled way of designing solutions in various other research areas, such as

- Transfer in reinforcement learning [Taylor and Stone, 2009], which aims to use past learning experiences to accelerate a current learning process; and
- MDP abstraction [Abel, 2019], where the goal is to construct an abstracted MDP with a smaller state-action space, while still maintaining certain properties of the original MDP.

The research most related to ours is the bisimulation metric, proposed in [Ferns *et al.*, 2004], which smoothly extends the notion of bisimulation [Givan *et al.*, 2003]. Briefly, bisimulation defines an equivalence relation between states and thereby induces a partition of the state space into equivalence classes. Whenever the *same* action is taken, bisimilar states perform exactly the same (probabilistic) transition into other equivalence classes, receiving exactly the same reward. Based on the observation that *two states are similar if the same action has similar effects*, [Ferns *et al.*, 2004] developed the bisimulation metric between states to extend the rigorous notion of bisimulation. Nonetheless, one problem with the bisumlation metric is that it overlooks the role of actions. Consider the MDP example in Fig. 1(a). The bisimulation metric cannot capture the fact that taking action b on state u is, to some extent, similar to taking action a on state v. In this paper, we tackle this problem by explicitly considering the similarity between *different* actions.



(a) An example MDP M (b) MDP graph G_M

Figure 1: An MDP example and its graph representation. Labels on the edge are of the form "(action/)transition probability/reward".

Another related notion is the MDP homomorphism [Ravindran and Barto, 2003], which consists of two algebraic mappings, one between states and the other between actions. The two mappings should *exactly* preserve the reward and transition probabilities. It has been pointed out that homomorphisms are often too strict and computationally difficult to be practically useful [Taylor and Stone, 2011]. [Sorg and Singh, 2009] proposed *soft* homomorphism, allowing the state mapping to be probabilistic as long as it exactly preserves transition probabilities and rewards in expectation. We argue that the soft homomorphism is still too rigorous in many practical situations. For example, there is no nontrivial (soft) homomorphism in the example of Fig. 1(a), even though states u and v are intuitively quite similar.

Instead of the algebraic view adopted by bisimulation and homomorphism, in this paper, we take a graph-theoretical

*Part of this work was done when this author stayed at the State Key Laboratory for Novel Software Technology, Nanjing University.

perspective to reveal the structural similarities within a finite MDP. Our methodology originates in the studies on measuring the structural-contextual similarities between nodes in a general graph. [Jeh and Widom, 2002] developed SimRank, of which the basic idea is that *two nodes are similar if and only if their neighbors are similar*. Although SimRank is not directly applicable to MDPs, as it is unaware of MDP-specific characteristics, we are nevertheless able to tailor its idea to develop similarity measures in MDPs. To sum up, the contributions of this paper are the following.

1. We propose a heterogeneous bipartite graph representation for finite MDPs (Sec. 3). The graph contains state and action nodes interconnected by decision and transition edges. With such a representation, the roles of states and actions are properly captured.
2. We propose recursive definitions for the state and action similarities (Sec. 4.1). The similarities can be efficiently computed by an iterative algorithm (Sec. 4.2).
3. We prove that the induced distance measures have good metric properties (Sec. 4.3). We also show that the proposed measures can be used to bound the difference between optimal values of the MDP (Sec. 4.4).
4. We show, via extensive experiments on randomly generated MDPs, that the proposed measures are effective in capturing structural similarities (Sec. 5).

2 Related Work

2.1 Transfer in Reinforcement Learning

Transfer learning aims at using past learning experience to accelerate the learning of a current task. Many transfer learning techniques are based on the notion of task similarity. [Lazaric *et al.*, 2008] measured the similarity between two MDPs from a sample-oriented perspective. [Ramamoorthy *et al.*, 2013] proposed a value-preserving Lipschitz metric between two MDPs within the same state-action space, which was calculated as the maximum possible difference between the reward and state transition functions. [Ammar *et al.*, 2014] used a restricted Boltzmann machine to measure the distance between two batches of samples from two MDPs, which was later used as the distance between the two MDPs. [Sinapov *et al.*, 2015] represented MDPs as feature vectors and trained a regression model to evaluate the closeness of two MDPs. [Song *et al.*, 2016] recently employed several metrics to measure the distance between two MDPs within the same state-action space. The above works mainly focus on inter-task similarities rather than structural similarities within one single task.

2.2 MDP Abstraction

The research field of MDP abstraction is also relevant to our work. [Li *et al.*, 2006] proposed a unified theory of state abstraction, covering bisimulation [Givan *et al.*, 2003], homomorphism [Ravindran and Barto, 2003], policy irrelevance [Jong and Stone, 2005], etc. The focus of the theory was mainly on property preservation during the process of state abstraction. Structural similarities (especially similarities between actions) were not fully explored. On the other hand, there have been several research works on planning and learning with abstract MDPs [Abel *et al.*, 2019; Gopalan *et al.*, 2017]. Their focus is different from ours. However, our work may contribute to this area by introducing a novel perspective towards MDP abstraction.

2.3 Structural Similarity in Graphs

Besides the SimRank measure [Jeh and Widom, 2002] introduced in Sec. 1, node-to-node proximities in graphs have been extensively studied in the literature. [Jeh and Widom, 2003] suggested using personalized PageRank, which is essentially an asymmetric random walk distance. Since SimRank is counterintuitive and inflexible in certain cases, many studies have been done to improve SimRank. [Xi *et al.*, 2005] proposed SimFusion, based on a unified relationship matrix representation of the graph. [Antonellis *et al.*, 2008] revised SimRank to Simrank++, by weighting the neighbors of graph nodes. [Zhao *et al.*, 2009] proposed P-Rank, extending SimRank to work for information networks. [Lin *et al.*, 2012] introduced maximum matching into the similarity measure and thereby developed MatchSim. Recently, [Jin *et al.*, 2014] proposed a metric, RoleSim, that complies with a set of admissible properties. As with SimRank, the above-mentioned measures are not directly applicable to our problem because they are not tailored to finite MDPs.

3 Graph Representation of MDPs

In the literature, an MDP is typically described as a tuple $M = (S, A, T, R)$. S and A are the finite sets of *states* and *actions*, respectively. In particular, for any $s \in S$, we denote by $A_s \subseteq A$ the set of *available actions* on state s. $T : S \times A \times S \to [0, 1]$ and $R : S \times A \times S \to [0, 1]$ are the *state transition function* and the *reward function*, respectively;[1] so, $T(s, a, s')$ gives the probability of ending up on state s' after taking action a on state s, and $R(s, a, s')$ is the reward for such a state transition. One problem with this algebraic representation of an MDP is that it does not distinguish between actions that have the same name but are taken on different states. To tackle this problem, we instead consider the following graph-theoretical representation of MDPs.

The *MDP graph* for an MDP $M = (S, A, T, R)$ is defined as $G_M = (V, \Lambda, E, \Psi, p, r)$, which is a heterogeneous directed bipartite graph with two types of nodes, the *state nodes* (V) and the *action nodes* (Λ). E is the set of *decision edges* from state nodes to action nodes and Ψ contains all the *transition edges* from action nodes to state nodes. While the decision edges are unweighted, each transition edge $(\alpha, v) \in \Psi$ is weighted by a transition probability $p(\alpha, v)$ and a reward $r(\alpha, v)$. The following procedure constructs the graph G_M from a given MDP M.

1. For each $s \in S$, create a new node v_s into V.
2. For each $s \in S$ and each $a \in A_s$, create a new node α_a into Λ and a new edge (v_s, α_a) into E.

[1] In principle, the value of R may take arbitrary reals. Nonetheless, in most practical cases, it is reasonable to assume that R is both lower and upper bounded and can thus be rescaled into $[0, 1]$ without sacrificing the representation power of the MDP model.

3. For each $(s, a, s') \in S \times A \times S$ such that $T(s, a, s') > 0$, create a new edge $(\alpha_a, v_{s'})$ into Ψ; set $p(\alpha_a, v_{s'}) = T(s, a, s')$ and $r(\alpha_a, v_{s'}) = R(s, a, s')$.

It is clear that there is a one-to-one correspondence between an MDP M and its graph G_M. The graph G_M is always bipartite, with $|V| = |S|$ and $|\Lambda| = |E| = \sum_{s \in S} |A_s|$. Fig. 1(b) shows the graph of the MDP in Fig. 1(a), which contains 6 state nodes, 4 action nodes, 4 decision edges, and 9 transition edges. Note that the 4 action nodes distinguish the two actions $a, b \in A$ on states u and v.

4 The Structural Similarities

For any node $x \in V \cup \Lambda$ in a given MDP graph $G_M = (V, \Lambda, E, \Psi, p, r)$, let N_x be the set of all the out-neighbors of x. Note that G_M is bipartite, thus the out-neighbors of a state node are always action nodes, whereas those of an action node are always state nodes (see Fig. 1(b)). We now define the state similarity σ_{S} and the action similarity σ_{A}. The goal is to make the induced distance measures

$$\delta_{\text{S}}(u, v) \stackrel{\text{def}}{=} 1 - \sigma_{\text{S}}(u, v), \quad \forall u, v \in V, \tag{1}$$

$$\delta_{\text{A}}(\alpha, \beta) \stackrel{\text{def}}{=} 1 - \sigma_{\text{A}}(\alpha, \beta), \quad \forall \alpha, \beta \in \Lambda, \tag{2}$$

have desirable properties.

4.1 The Recursive Similarity Measure

We adopt the basic idea of SimRank [Jeh and Widom, 2002] to define σ_{S} and σ_{A}; namely, that *two nodes are similar if and only if their neighbors are similar*. The idea is implemented as a recursion.

The Base Cases

As the base cases, we define

$$\delta_{\text{S}}(u, v) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } u = v, \\ 1, & \text{if } u \text{ or } v, \text{ but not both, is absorbing,} \\ d_{u,v}, & \text{if both } u \text{ and } v \text{ are absorbing.} \end{cases}$$

Here, a state is *absorbing* if there is no out-neighbors, which is typically a goal state in practice. A configuration of $d_{u,v} \in [0, 1]$ is thus an application-dependent description of the relationships among goal states. Two special cases are $d_{u,v} \equiv 1$ and $d_{u,v} \equiv 0$, indicating that any two goal states should be regarded completely different or identical, respectively.

The Recursion for State Similarity

For any two state nodes $u, v \in V$, the similarity $\sigma_{\text{S}}(u, v)$ is essentially the similarity between their out-neighbors N_u and N_v, which are two *sets* of action nodes. The similarity between N_u and N_v should in turn be based on the pairwise similarity $\sigma_{\text{A}}(\alpha, \beta)$, for every $\alpha \in N_u$ and $\beta \in N_v$. Note that simply averaging over all the pairwise similarities $\sigma_{\text{A}}(\alpha, \beta)$ (as SimRank does) will prevent δ_{S} from being a metric, since the triangle inequality is compromised. Hence, we consider using the Hausdorff distance [Delfour and Zolésio, 2011].[2]

[2]Other options do exist. For example, the *minimum pairwise distance* (mindist), $\min_{\alpha,\beta} \delta_{\text{A}}(\alpha, \beta)$, is a metric. Yet, one problem is that a single small $\delta_{\text{A}}(\alpha, \beta)$ dominates all the other pairwise distances even if they are all very large. This may be inconsistent with the intuition that similar states should have similar decision options.

Specifically, given an action node α and a set of action nodes N, the distance between α and N, abusing the symbol δ_{A}, is $\delta_{\text{A}}(\alpha, N) \stackrel{\text{def}}{=} \min_{\beta \in N} \delta_{\text{A}}(\alpha, \beta)$. The Hausdorff distance, given δ_{A}, is then the maximum of all element-to-set distances:

$$\delta_{\text{Haus}}(N_u, N_v; \delta_{\text{A}}) = \max_{\substack{\alpha \in N_u \\ \beta \in N_v}} \{\delta_{\text{A}}(\alpha, N_v), \delta_{\text{A}}(\beta, N_u)\}. \tag{3}$$

The Hausdorff distance $\delta_{\text{Haus}}(N_u, N_v; \delta_{\text{A}}) = \Delta$ provides an upper bound on the pairwise distances between the actions available on states u and v. Specifically, within a distance Δ from any action in N_u (resp. N_v), there is always another action from N_v (resp. N_u); thereby, Δ bounds the overall dissimilarity between N_u and N_v. Using δ_{Haus}, we obtain

$$\sigma_{\text{S}}(u, v) = C_{\text{S}} \cdot (1 - \delta_{\text{Haus}}(N_u, N_v; \delta_{\text{A}})), \tag{4}$$

where $u, v \in V$ are two distinct non-absorbing states and $0 < C_{\text{S}} < 1$ is a constant discounting the impact of the neighbors N_α and N_β on the pair of state nodes (u, v).

The Recursion for Action Similarity

The state similarity σ_{S} relies on a well-defined δ_{A} (see Eq. 4). As shown in Fig. 1(b), in an MDP graph, an action node itself conveys limited information – what really matters is its *consequences* or *effects*. An action node α is essentially both a distribution $p_\alpha = p(\alpha, *)$ over the state nodes (probabilistic transition) and a distribution $r_\alpha = r(\alpha, *)$ over $[0, 1]$ (stochastic reward). The distance between rewards is relatively simple:

$$\delta_{\text{rwd}}(\alpha, \beta) = |\mathbb{E}[r_\alpha] - \mathbb{E}[r_\beta]|, \tag{5}$$

i.e., it is the difference between their expectations. The rest of the task involves measuring the distance between two probabilistic distributions over state nodes. To that end, we employ the earth mover's distance (EMD) [Rubner *et al.*, 1998][3]:

$$\begin{aligned} \delta_{\text{EMD}}(p_\alpha, p_\beta; \delta_{\text{S}}) = \min_{\boldsymbol{F}} & \sum_{u \in N_\alpha} \sum_{v \in N_\beta} f_{u,v} \cdot \delta_{\text{S}}(u, v) \\ \text{s.t.} \quad & \forall u, v \in V : f_{u,v} \geq 0, \\ & \forall u \in V : \sum_{v \in V} f_{u,v} = p(\alpha, u), \\ & \forall v \in V : \sum_{u \in V} f_{u,v} = p(\beta, v). \end{aligned} \tag{6}$$

EMD quantifies the effort for transforming one distribution into another by moving the "earth" (probabilities, in our case) around. The value $f_{u,v}$ in the matrix $\boldsymbol{F}$ is the "earth" moved from the state node $u \in N_\alpha$ to the state node $v \in N_\beta$. Such a movement of the "earth" incurs a cost of $\delta_{\text{S}}(u, v)$. $\delta_{\text{EMD}}(p_\alpha, p_\beta; \delta_{\text{S}})$ is thus the minimum possible effort for transforming the distribution p_α to the distribution p_β.

The recursion for σ_{A}, based on δ_{EMD}, is designed to be

$$\begin{aligned} \sigma_{\text{A}}(\alpha, \beta) = & 1 - (1 - C_{\text{A}}) \delta_{\text{rwd}}(\alpha, \beta) \\ & - C_{\text{A}} \delta_{\text{EMD}}(p_\alpha, p_\beta; \delta_{\text{S}}), \end{aligned} \tag{7}$$

where $0 < C_{\text{A}} < 1$ is the parameter to weight the importance of the reward similarity and the transition similarity.

[3]There are other statistical distance measures (e.g., Kullback-Leibler divergence, Hellinger distance, Jensen-Shannon divergence, etc. [Venturini, 2015]). They are not applicable here as they require a fixed matching between N_α and N_β to get nontrivial evaluations.

4.2 Computation

Algorithm 1 shows the iterative algorithm for computing σ_{S}^* and σ_{A}^* by simulating the recursion of Sec. 4.1. The algorithm

Algorithm 1 Structural similarities of an MDP graph

Input: MDP graph $G_M = (V, \Lambda, E, \Psi, p, r)$
Parameter: Discount factors $C_{\mathrm{S}}, C_{\mathrm{A}} \in (0, 1)$
Output: Solution $(\sigma_{\mathrm{S}}^*, \sigma_{\mathrm{A}}^*)$ to the recursion of Sec. 4.1

▷ *Initialization & base cases.*
1: $\boldsymbol{S} \leftarrow \boldsymbol{I}_{|V|\times|V|}$, $\boldsymbol{A} \leftarrow \boldsymbol{I}_{|\Lambda|\times|\Lambda|}$
▷ *Iterative computation.*
2: **repeat**
3: **for** all $\alpha \in N_u$ and $\beta \in N_v$ $(u, v \in V, u \neq v)$ **do**
4: $d \leftarrow \mathrm{EMD}\,(p_\alpha, p_\beta; G_M, \mathbf{1} - \boldsymbol{S})$
5: Compute $\boldsymbol{A}_{\alpha,\beta}$ using Eq. 7 with C_{A}, d and $\boldsymbol{S}$
6: **for** all $u, v \in V$ with $N_u \neq \emptyset$ and $N_v \neq \emptyset$ **do**
7: Compute $\boldsymbol{S}_{u,v}$ using Eq. 4 with C_{S} and $\boldsymbol{A}$
8: **until** $\boldsymbol{S}$ and $\boldsymbol{A}$ converge
9: **return** $(\sigma_{\mathrm{S}}^*, \sigma_{\mathrm{A}}^*) \leftarrow (\boldsymbol{S}, \boldsymbol{A})$

is mostly straightforward except for Line 4, where a subroutine EMD is invoked to compute $\delta_{\mathrm{EMD}}(p_\alpha, p_\beta; \delta_{\mathrm{S}})$. The computation of the EMD can be seen as solving an instance of the *minimum-cost network flow* (MCF) problem, which can be done by using, for example, the *successive shortest path* (SSP) algorithm [Jewell, 1962].

Space and time complexity. Given the graph $G_M = (V, \Lambda, E, \Psi, p, r)$ of the MDP $M = (S, A, T, R)$, Algorithm 1 requires $\Theta\left(|V|^2 + |\Lambda|^2\right) = O\left(|S|^2|A|^2\right)$ space to store $\boldsymbol{S}$ and $\boldsymbol{A}$. SSP takes $O\left(K_{\max}^2\right)$ working memory, where $K_{\max} \leq |V|$ is the maximum out-degree of action nodes in G_M. Given a predefined precision ϵ (e.g., $\epsilon = .001$), SSP is guaranteed to terminate in $O\left(\frac{1}{\epsilon^2} \cdot \left(K_{\max}^2 + K_{\max} \log K_{\max}\right)\right) = O\left(\frac{1}{\epsilon^2} \cdot K_{\max}^2\right)$ time, using Dijkstra's algorithm with a Fibonacci heap.[4] Each iteration of Algorithm 1 (Lines 3-7) makes $\Theta\left(|\Lambda|^2\right)$ calls to SSP. Computing the Hausdorff distances takes $\Theta\left(|V|^2 L_{\max}^2\right)$ time, where $L_{\max} \leq |A|$ is the maximum out-degree of state nodes in G_M. The overall time cost is therefore $O\left(N \cdot |S|^2|A|^2 K_{\max}^2/\epsilon^2\right)$, where N is the number of iterations before convergence.

4.3 Mathematical Properties

We now prove that σ_{S}^* and σ_{A}^* are well-defined by showing that Algorithm 1 always terminates. Let $\boldsymbol{S}^{(k)}$ and $\boldsymbol{A}^{(k)}$ be versions of the matrices $\boldsymbol{S}$ and $\boldsymbol{A}$ after the k-th execution of Lines 3-7 of Algorithm 1 ($k = 1, 2, \cdots$). In addition, let $\boldsymbol{S}^{(0)}$ and $\boldsymbol{A}^{(0)}$ be the contents of $\boldsymbol{S}$ and $\boldsymbol{A}$ right before the algorithm enters the main loop. Let the symbol $\preceq$ denote the pairwise-less-than-or-equal-to relationship between matrices.

Lemma 1 (Boundedness) *For all $k \geq 0$, $\mathbf{0} \preceq \boldsymbol{S}^{(k)} \preceq \mathbf{1}$ and $\mathbf{0} \preceq \boldsymbol{A}^{(k)} \preceq \mathbf{1}$.*

[4] It is known that such a worst-case analysis is too pessimistic. [Brunsch *et al.*, 2013] developed a smoothed analysis of SSP to better explain its practical efficiency. The discussion is, however, out of the scope of this paper as it does not affect how we use SSP.

Lemma 2 (Monotonicity) *For all $k \geq 0$, $\boldsymbol{S}^{(k)} \preceq \boldsymbol{S}^{(k+1)}$ and $\boldsymbol{A}^{(k)} \preceq \boldsymbol{A}^{(k+1)}$.*

Sketch of Proof. Lemmata 1 & 2 can be proved by induction, using the boundedness of δ_{Haus} and δ_{EMD}, as well as their monotonicity with respect to δ_{A} and δ_{S}, respectively. □

Theorem 1 (Unique Existence & Nontrivialness) *Algorithm 1 terminates correctly with the unique solution (σ_S^*, σ_A^*) to the recursion of Sec. 4.1. The solution is nontrivial in the sense that $\sigma_S^* \not\equiv 1$ and $\sigma_A^* \not\equiv 1$, even if there is no absorbing state in the input MDP graph G_M.*

Proof. It is a direct corollary of Lemmata 1 & 2 that

$$\lim_{k\to\infty} \boldsymbol{S}^{(k)} = \sigma_{\mathrm{S}}^* \in [0, 1],$$
$$\lim_{k\to\infty} \boldsymbol{A}^{(k)} = \sigma_{\mathrm{A}}^* \in [0, 1].$$

To see the nontrivialness of $(\sigma_{\mathrm{S}}^*, \sigma_{\mathrm{A}}^*)$, it suffices to verify that $\sigma_{\mathrm{S}} \equiv 1$ or $\sigma_{\mathrm{A}} \equiv 1$ cannot constitute the solution due to the discount factor $0 < C_{\mathrm{S}} < 1$. □

Next, we consider the matrices $\boldsymbol{D}^{(k)} \stackrel{\text{def}}{=} \mathbf{1} - \boldsymbol{S}^{(k)}$ and $\boldsymbol{L}^{(k)} \stackrel{\text{def}}{=} \mathbf{1} - \boldsymbol{A}^{(k)}$ for $k = 0, 1, 2, \cdots$.

Lemma 3 (Triangle Inequality) *The triangle inequalities $\boldsymbol{D}_{u,v}^{(k)} \leq \boldsymbol{D}_{u,w}^{(k)} + \boldsymbol{D}_{w,v}^{(k)}$ and $\boldsymbol{L}_{\alpha,\beta}^{(k)} \leq \boldsymbol{L}_{\alpha,\gamma}^{(k)} + \boldsymbol{L}_{\gamma,\beta}^{(k)}$ hold for arbitrary state nodes $u, v, w \in V$, arbitrary action nodes $\alpha, \beta, \gamma \in \Lambda$ and every $k \geq 0$, if the configuration of the base cases, $d_{u,v}$, observes the triangle inequality (i.e., if for any absorbing states u, v and w, $d_{u,v} \leq d_{u,w} + d_{w,v}$).*

Sketch of Proof. This proof is also done by induction. Verifying for $\boldsymbol{L}^{(0)} = \mathbf{1}$ and $\boldsymbol{D}^{(0)}$ is trivial, given the initial configuration of $d_{u,v}$ values. The induction can then be completed since $\delta_{\mathrm{Haus}}\left(*, *; \boldsymbol{L}^{(k)}\right)$ and $\delta_{\mathrm{EMD}}\left(*, *; \boldsymbol{D}^{(k)}\right)$ are metrics. □

Lemma 3 immediately leads to the following result.

Theorem 2 (Metric Properties) *If (σ_S^*, σ_A^*) is the solution to the recursion of Sec. 4.1, formulated on the MDP graph $G_M = (V, \Lambda, E, \Psi, p, r)$, then $\delta_S^* = 1 - \sigma_S^*$ is a metric on V and $\delta_A^* = 1 - \sigma_A^*$ is a pseudometric on Λ.*

Proof. It is clear that both δ_{S}^* and δ_{A}^* are nonnegative and symmetric. The triangle inequalities of δ_{S}^* and δ_{A}^* are obtained by taking limits on both sides of the inequalities of Lemma 3. Then, δ_{S}^* is a metric because $\delta_{\mathrm{S}}^*(u, v) = 0$ if and only if $u = v$. Indeed, for any $u \neq v$, $\sigma_{\mathrm{S}}^*(u, v) < 1$ due to the discount factor C_{S}, even if $\delta_{\mathrm{Haus}}(u, v; \delta_{\mathrm{A}}^*) = 0$ (see Eq. 4). In contrast, it is possible that $\delta_{\mathrm{A}}^*(\alpha, \beta) = 0$ for two different actions $\alpha \neq \beta \in \Lambda$, which happens when $r_\alpha \equiv r_\beta$ and $p_\alpha \equiv p_\beta$. □

4.4 Bounding Differences Between Optimal Values

Given an MDP graph $G_M = (V, \Lambda, E, \Psi, p, r)$ and an arbitrary initial state $u_0 \in V$, following a probabilistic *policy* $\pi : V \times \Lambda \to [0, 1]$, there will be a trajectory of state transitions (of which the length could potentially be infinite):

$$u_0 \xrightarrow[r_1]{\alpha_0 = \pi(u_0)} u_1 \xrightarrow[r_2]{\alpha_1 = \pi(u_1)} u_2 \xrightarrow[r_3]{\alpha_2 = \pi(u_2)} \cdots$$

Given a discount factor $\rho \in (0, 1)$, the *state value* of $u \in V$ under policy π, written $\mathcal{V}_u^\pi$, is the expected total accumulative return starting from the state node u, i.e.,

$$\mathcal{V}_u^\pi \stackrel{\text{def}}{=} \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \rho^k r_{k+1} \middle| u_0 = u \right]. \tag{8}$$

Similarly, the *action value* of $\alpha \in \Lambda$ under policy π is

$$\mathcal{Q}_\alpha^\pi \stackrel{\text{def}}{=} \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \rho^k r_{k+1} \middle| \alpha_0 = \alpha \right]. \tag{9}$$

Now, consider the optimal value functions $\mathcal{V}^*$ and $\mathcal{Q}^*$ under the optimal policy π^*. The well-known Bellman equations [Sutton and Barto, 2018] state that

$$\mathcal{V}_u^* = \max_{\alpha \in N_u} \mathcal{Q}_\alpha^*,$$
$$\mathcal{Q}_\alpha^* = \sum_{u \in N_\alpha} p(\alpha, u) \left(r(\alpha, u) + \rho \mathcal{V}_u^* \right).$$

We show that the proposed distance measures, δ_{S}^* and δ_{A}^*, can be used to bound the difference between the optimal values.

Theorem 3 (Bounds on Differences of Optimal Values) *For any state nodes $u, v \in V$ and any action nodes $\alpha, \beta \in \Lambda$, if we choose $C_S = 1$ and $C_A = \rho$, then we have*[5]

$$|\mathcal{V}_u^* - \mathcal{V}_v^*| \le \frac{1}{1-\rho} \cdot \delta_S^*(u, v),$$
$$\left| \mathcal{Q}_\alpha^* - \mathcal{Q}_\beta^* \right| \le \frac{1}{1-\rho} \cdot \delta_A^*(\alpha, \beta).$$

Proof. Consider the sequences $\left\{ \mathcal{V}^{(k)} \right\}$ and $\left\{ \mathcal{Q}^{(k)} \right\}$ $(k \ge 0)$:

$$\mathcal{V}_u^{(k)} = \max_{\alpha \in N_u} \mathcal{Q}_\alpha^{(k)},$$
$$\mathcal{Q}_\alpha^{(k)} = \sum_{u \in N_u} p(\alpha, u) \left(r(\alpha, u) + \rho \mathcal{V}_u^{(k-1)} \right).$$

It suffices to first prove by induction that, for any k, $\left| \mathcal{V}_u^{(k)} - \mathcal{V}_v^{(k)} \right| \le \frac{1}{1-\rho} \boldsymbol{D}_{u,v}^{(k)}$ and $\left| \mathcal{Q}_\alpha^{(k)} - \mathcal{Q}_\beta^{(k)} \right| \le \frac{1}{1-\rho} \boldsymbol{L}_{\alpha,\beta}^{(k)}$ and then take limits on both sides of the inequalities. After verifying for $k = 0$, we see that

$$\left| \mathcal{Q}_\alpha^{(k+1)} - \mathcal{Q}_\beta^{(k+1)} \right|$$
$$\le \delta_{\text{rwd}}(\alpha, \beta) + \rho \left| \sum_{u \in V} \left(p(\alpha, u) - p(\beta, u) \right) \mathcal{V}_u^{(k)} \right|.$$

Due to the induction hypothesis $\left| \mathcal{V}_u^{(k)} - \mathcal{V}_v^{(k)} \right| \le \frac{1}{1-\rho} \boldsymbol{D}_{u,v}^{(k)}$, the second term is exactly the dual problem of Eq. 6 with cost function $\frac{1}{1-\rho} \boldsymbol{D}^{(k)}$. Hence,

$$\left| \mathcal{Q}_\alpha^{(k+1)} - \mathcal{Q}_\beta^{(k+1)} \right|$$
$$\le \delta_{\text{rwd}}(\alpha, \beta) + \rho \delta_{\text{EMD}} \left(p_\alpha, p_\beta; \frac{1}{1-\rho} \boldsymbol{D}^{(k)} \right) \qquad \text{(I.H.)}$$
$$= \frac{1}{1-\rho} \boldsymbol{L}_{\alpha,\beta}^{(k+1)}. \qquad \text{(Line 5 of Algorithm 1)}$$

[5]The main purpose of C_{S} is to, theoretically, guarantee a nontrivial $(\sigma_{\text{S}}^*, \sigma_{\text{A}}^*)$; any $C_{\text{S}} < 1$ serves its purpose. Here, it is convenient to choose a C_{S} very close to 1 such that it can be ignored in Eq. 7.

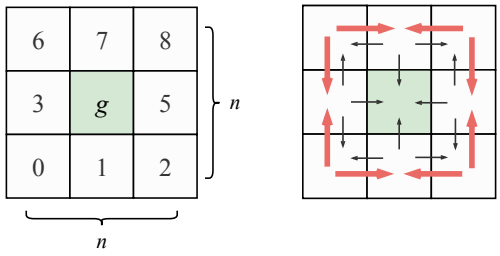
Define $d^{(k)}(\alpha, \beta) = \left| \mathcal{Q}_\alpha^{(k)} - \mathcal{Q}_\beta^{(k)} \right|$ for $k \ge 0$, and let $\alpha^* = \arg\max_{\alpha \in N_u} \mathcal{Q}_\alpha^{(k+1)}$ and $\beta^* = \arg\max_{\beta \in N_v} \mathcal{Q}_\beta^{(k+1)}$. It can be shown that either α^* finds β^* as the closest counterpart, with respect to the distance measure $d^{(k+1)}$, or the other way around. Hence,

$$\left| \mathcal{V}_u^{(k+1)} - \mathcal{V}_v^{(k+1)} \right| = d^{(k+1)}(\alpha^*, \beta^*)$$
$$\le \delta_{\text{Haus}} \left(N_u, N_v; d^{(k+1)} \right) \le \delta_{\text{Haus}} \left(N_u, N_v; \frac{1}{1-\rho} \boldsymbol{L}^{(k+1)} \right)$$
$$= \frac{1}{1-\rho} \boldsymbol{D}_{u,v}^{(k+1)}. \qquad \text{(Line 7 of Algorithm 1)} \qquad \square$$

Note that, since the reward function r is bounded within $[0, 1]$, there is a trivial upper bound of $\sum_{k=0}^{\infty} \rho^k = \frac{1}{1-\rho}$ on both $\mathcal{V}^*$ and $\mathcal{Q}^*$ (see Eq. 8-9). Thm. 3 reveals a connection between the proposed similarity/distance measure and the optimal values by tightening the trivial upper bound.

5 Experiments

We experimentally evaluate our proposed similarity measures using a series of designed MDPs. The state space of each MDP is an $n \times n$ grid of cells, as shown in Fig. 2(a), where n is an odd integer. States are indexed with natural numbers



(a) The state space (b) Available actions

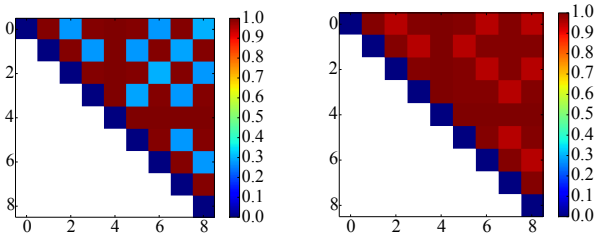
Figure 2: An $n \times n$ grid ($n = 3$) as an MDP $M = (S, A, T, R)$.

in a left-to-right and bottom-to-top manner. There is a sole goal state $g \in S$, indexed $\frac{n^2-1}{2}$, at the center of the grid. The action space consists of 4 actions intended towards 4 different directions: left, up, right, and down. On any state, an action is available as long as its intention is physically possible. If action $a \in A_s$ is taken on state $s \in S$, then it has a $\tau_{s,a}$ probability of going along the intended direction and $1 - \tau_{s,a}$ probability of entering a random adjacent state. After entering a state s', the agent receives a random reward of $r_{s,a,s'}$. We set $r_{s,a,g} \equiv 1$ for any $s \in S$ and $a \in A$.

5.1 A Case Study

We first carry out a simple case study to show the ability of our measures in capturing structural information for MDPs. We use the 3×3 grid shown in Fig. 2, in which $\tau_{s,a} \equiv 0.9$ for any $s \in S$ and $a \in A$, and $r_{s,a,s'} \equiv 0$ whenever $s' \neq g$. As shown in Fig. 2(b), in this 9-state MDP there are 20 available actions. Note that states 0, 2, 6, and 8, together with the eight actions highlighted/in bold, are structurally symmetric. The same is true for states 1, 3, 5, and 7.

We compare our results on this 9-state MDP with those of d_{bis}, the bisimulation metric using EMD [Ferns *et al.*, 2004]. d_{bis} requires that every state has the same set of available actions. To meet this requirement, for any action of which the intention is physically impossible, we define its EMD for other actions to be 1. In all the experiments, we set $C_{\text{S}} = C_{\text{A}} = 0.95$ for our solution and $c_R = 0.05$ and $c_T = 0.95$ for d_{bis}.



(a) Our solution δ^*_{S} (b) Bisimulation metric d_{bis}

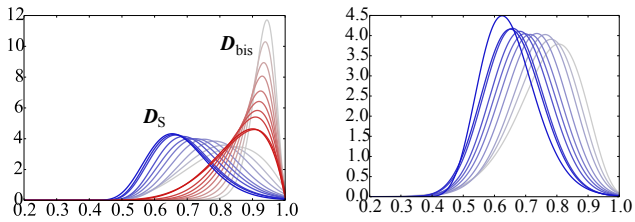
Figure 3: Visualization of state distance matrices.

Fig. 3 visualizes the resulting (upper-triangular) matrices of the distances between the 9 states. As can be seen, our results show a clear structural pattern in the distance matrix. The smallest distances are $\delta^*_{\text{S}}(0,2) = \delta^*_{\text{S}}(0,6) = 0.27489$ and $\delta^*_{\text{S}}(1,3) = \delta^*_{\text{S}}(1,5) = 0.27492$, followed by $\delta^*_{\text{S}}(1,7) = 0.28627$ and $\delta^*_{\text{S}}(0,8) = 0.29873$. In contrast, d_{bis} generates very large distance values, with 0.95 being the minimum. In particular, with $d_{\text{bis}}(1,7) = d_{\text{bis}}(3,5) = 0.995$, d_{bis} fails to capture certain structural similarities in the MDP.

5.2 Distribution of Distance Values

We now analyze the distribution of the calculated distance values. We generate a series of $n \times n$ grid MDPs $\{M_n\}$ for $n = 11, 13, \cdots, 33$. In each MDP M_n, for every $s \in S$ and $a \in A_s$, we randomly draw the success probability $\tau_{s,a}$ from the Gaussian distribution $\mathcal{N}(0.9, 0.05^2)$ and the reward $r_{s,a,s'}$ from the uniform distribution $\mathcal{U}(0, 0.01)$ when $s' \neq g$. The reward for entering the goal state remains 1. The size of the state space of $\{M_n\}$ ranges from 121 to 1,089, which adequately supports practical applications of finite MDPs. On each MDP M_n, Algorithm 1 calculates two similarity matrices, $\boldsymbol{S}_n$ and $\boldsymbol{A}_n$, with parameters $C_{\text{S}} = C_{\text{A}} = 0.95$, from which we obtain two distance matrices, $\boldsymbol{D}^n_{\text{S}} = \mathbf{1} - \boldsymbol{S}_n$ and $\boldsymbol{D}^n_{\text{A}} = \mathbf{1} - \boldsymbol{A}_n$. We also calculate a matrix $\boldsymbol{D}^n_{\text{bis}}$ for d_{bis}, using parameters $c_R = 0.05$ and $c_T = 0.95$.

For each of $\boldsymbol{D}^n_{\text{S}}$, $\boldsymbol{D}^n_{\text{bis}}$, and $\boldsymbol{D}^n_{\text{A}}$, we fit the values as a Beta distribution. Fig. 4 shows the fitted probability density functions (PDFs). There are clearly three series of PDFs, fitted from $\boldsymbol{D}^n_{\text{S}}$, $\boldsymbol{D}^n_{\text{bis}}$ (Fig. 4(a)) and $\boldsymbol{D}^n_{\text{A}}$ (Fig. 4(b)), for $n = 11, 13, \cdots, 33$. In each series, a darker line corresponds to a larger state-action space. As can be seen, the $\boldsymbol{D}_{\text{S}}$ and $\boldsymbol{D}_{\text{A}}$ series have consistent trends. The $\boldsymbol{D}_{\text{bis}}$ series is heavily skewed towards 1, while the $\boldsymbol{D}_{\text{S}}$ and $\boldsymbol{D}_{\text{A}}$ series are more flattened. This indicates that our measures generate more reasonably distributed distance values, whereas the bisimulation metric constantly underestimates the structural similarities in $\{M_n\}$.

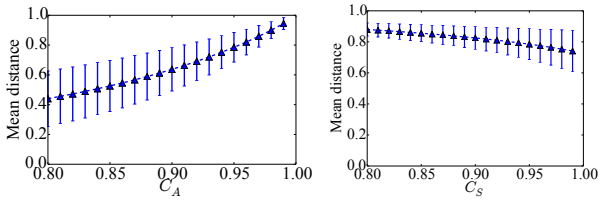


(a) Fitted from $\boldsymbol{D}_{\text{S}}$ and $\boldsymbol{D}_{\text{bis}}$ (b) Fitted from $\boldsymbol{D}_{\text{A}}$

Figure 4: PDFs of the fitted Beta distributions.

5.3 Effect of Parameters

Our measures rely on two parameters, C_{S} and C_{A}, so we perform experiments to investigate the impact of each. We randomly generate a moderate-sized MDP, M_{15}, using the same methodology as in Sec. 5.2. We first fix $C_{\text{S}} = 0.95$ and vary C_{A} from 0.80 to 0.99 with a step-size of 0.01. Then, we swap the roles of C_{S} and C_{A}. For each pair of $(C_{\text{S}}, C_{\text{A}})$, we run Algorithm 1 to obtain the state similarity matrix $\boldsymbol{S}$, from which we calculate the mean distance value and the standard deviation of the distance matrix $\boldsymbol{D} = \mathbf{1} - \boldsymbol{S}$.



(a) Effect of C_{A} ($C_{\text{S}} = 0.95$) (b) Effect of C_{S} ($C_{\text{A}} = 0.95$)

Figure 5: Effect of the parameters C_{A} and C_{S} on M_{15}.

Fig. 5 illustrates the impact of each parameter. As C_{A} increases, the mean distance value also increases and the distribution becomes more concentrated. It is worth mentioning that, as suggested by Thm. 3, the value of C_{A} is closely related to the discount factor ρ, which is determined by the application and of which 0.95 is a common practice [Sutton and Barto, 2018]. In contrast, C_{S} shows the opposite impact. As C_{S} increases, the mean distance decreases, though it remains around 0.8, and the distribution becomes flattened. This is consistent with the implication of Thm. 3 that $C_{\text{S}} \approx 1$.

6 Conclusion and Future Work

In this paper, we studied the structural similarities within a finite MDP. By representing MDPs as graphs, we described the similarities between states and actions by measuring the proximity between graph nodes. We proved the metric properties of the proposed measures, based on which we also derived upper bounds on the difference of optimal values. Extensive experiments showed the advantages of our measures. In the future, we plan to accelerate the computation of the proposed measures via parallelism, which is commonly used to handle large graphs. We are also interested in similarity search queries on MDPs, which aim to efficiently identify the most similar pairs of states/actions.

References

[Abel *et al.*, 2019] David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L. Littman, and Lawson L.S. Wong. State abstraction as compression in apprenticeship learning. In *AAAI*, 2019.

[Abel, 2019] David Abel. A theory of state abstraction for reinforcement learning. In *AAAI*, Doctoral Consortium, 2019.

[Ammar *et al.*, 2014] Haitham Bou Ammar, Eric Eaton, Matthew E. Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of MDP similarity for transfer reinforcement learning. In *AAAI*, 2014.

[Antonellis *et al.*, 2008] Ioannis Antonellis, Hector Garcia Molina, and Chi Chao Chang. Simrank++: query rewriting through link analysis of the click graph. In *VLDB*, 2008.

[Brunsch *et al.*, 2013] Tobias Brunsch, Kamiel Cornelissen, Bodo Manthey, and Heiko Röglin. Smoothed analysis of the successive shortest path algorithm. In *SODA*, 2013.

[Cai *et al.*, 2017] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. Real-time bidding by reinforcement learning in display advertising. In *WSDM*, 2017.

[Delfour and Zolésio, 2011] Michel C. Delfour and Jean-Paul Zolésio. *Shapes and Geometrics: Metrics, Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM, 2nd edition, 2011.

[Ferns *et al.*, 2004] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *UAI*, 2004.

[Givan *et al.*, 2003] Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147:163–223, 2003.

[Gopalan *et al.*, 2017] Nakul Gopalan, Marie desJardins, Michael L. Littman, James MacGlashan, Shawn Squire, Stefanie Tellex, John Winder, and Lawson L.S. Wong. Planning with abstract Markov decision processes. In *ICAPS*, 2017.

[Han *et al.*, 2016] Zhenhua Han, Haisheng Tan, Guihai Chen, Rui Wang, Yifan Chen, and Francis C.M. Lau. Dynamic virtual machine management via approximate Markov decision process. In *INFOCOM*, 2016.

[Jeh and Widom, 2002] Glen Jeh and Jennifer Widom. SimRank: a measure of structural-context similarity. In *KDD*, 2002.

[Jeh and Widom, 2003] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW*, 2003.

[Jewell, 1962] William S. Jewell. Optimal flow through networks. *Operations Research*, 10(4):476–499, 1962.

[Jin *et al.*, 2014] Ruoming Jin, Victor E. Lee, and Longjie Li. Scalable and axiomatic ranking of network role similarity. *TKDD*, 8(1):Article No. 3, 2014.

[Jong and Stone, 2005] Nicholas K. Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *IJCAI*, 2005.

[Lazaric *et al.*, 2008] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *ICML*, 2008.

[Li *et al.*, 2006] Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *ISAIM*, 2006.

[Lin *et al.*, 2012] Zhenjiang Lin, Michael R. Lyu, and Irwin King. MatchSim: a novel similarity measure based on maximum neighborhood matching. *KAIS*, 32(1):141–166, 2012.

[Ramamoorthy *et al.*, 2013] Subramanian Ramamoorthy, M. M. Hassan Mahmud, Majd Hawasly, and Benjamin Rosman. Clustering Markov decision processes for continual transfer. Technical report, University of Edinburgh, 2013.

[Ravindran and Barto, 2003] Balaraman Ravindran and Andrew G. Barto. SDMP homomorphisms: an algebraic approach to abstraction in semi-Markov decision processes. In *IJCAI*, 2003.

[Rubner *et al.*, 1998] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *ICCV*, 1998.

[Sinapov *et al.*, 2015] Jivko Sinapov, Sanmit Narvekar, Matteo Leonetti, and Peter Stone. Learning inter-task transferability in the absence of target task samples. In *AAMAS*, 2015.

[Song *et al.*, 2016] Jinhua Song, Yang Gao, Hao Wang, and Bo An. Measuring the distance between finite Markov decision processes. In *AAMAS*, 2016.

[Sorg and Singh, 2009] Jonathan Sorg and Satinder Singh. Transfer via soft homomorphisms. In *AAMAS*, 2009.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[Taylor and Stone, 2009] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: a survey. *JMLR*, 10:1633–1685, 2009.

[Taylor and Stone, 2011] Matthew E. Taylor and Peter Stone. An introduction to intertask transfer for reinforcement learning. *AI Magzine*, 32(1):15–34, 2011.

[Venturini, 2015] Gabriel Martos Venturini. *Statistical distances and probability metrics for multivariate data, ensembles and probability distributions*. PhD thesis, Universidad Carlos III de Madrid, 2015.

[Xi *et al.*, 2005] Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, and Dong Zhuang. SimFusion: measuring similarity using unified relationship matrix. In *SIGIR*, 2005.

[Zhao *et al.*, 2009] Peixiang Zhao, Jiawei Han, and Yizhou Sun. P-Rank: a comprehensive structrual similarity measure over information networks. In *CIKM*, 2009.