Reasoning for reinforcement learning

Silviu Pitis
Georgia Institute of Technology
spitis@gatech.edu

Abstract

This paper presents preliminary work on a framework for reasoning over multiple competing representations of the value function during reinforcement learning, with a focus on landmark and factor-based reasoning. This approach enables consistency-based learning, explicit justification of actions and learning by partial supervision, and has the potential to improve agent performance.

1 Introduction

Some human judgements are strictly intuitive; others involve reasoning. Given a choice between two research projects, we may go with our gut instinct, or we may value each independently and choose the higher value option. Those valuations may themselves be intuited or reasoned to. For instance, we might value a project based on its difficulty and significance. Once again, each factor may be intuited or reasoned to. In this way, *reasoning forms a hierarchical structure over primitive intuitions*.

A reinforcement learning (RL) agent may "intuit" choice through direct policy approximation. Alternatively, the agent can apply a minimal form of reasoning to find the greedy policy with respect to "intuited" state-action values. Can these values be reasoned to rather than intuited? If so, the ability to reason to and use these multiple representations offers many potential benefits: improved performance (if certain representations are known to be "better" than others), inter-representational consistency-based learning, explicit justification, and learning by partial supervision.

To augment our agents with reasoning, we need only endow them with (1) primitives with which to reason (i.e., measures or classifications produced by a black-box function like a deep neural network), (2) rules of reasoning for combining them, and (3) a mechanism by which reasoning is applied and representational conflicts are resolved (metacognition). This paper illustrates this three-part framework with landmark and factor-based reasoning using primitive distance and density measures. Landmark-based reasoning decomposes the value function temporally (cf. options [24]), whereas factor-based reasoning decomposes the value function as a sum over factors (cf. the successor representation [5]). Distance is the spatial, temporal, value-related, or abstract difference between two states, whereas density measures frequency within a time interval. Here, we consider only infinite-time horizon density, which is a measure of total future expectation.

2 Primitive learning and representation

Let us consider the measures defined in Table 1 in context of a infinite-time horizon Markov Decision Process (MDP) consisting a state space S, an action space A, a transition function $T : S \times A \times S \to [0, 1]$, a reward function $R : S \times A \to \mathbb{R}$ and a discount factor $\gamma \in [0, 1)$. As used in the table, $s, x, g \in S$ (but see Appendix A—Abstraction), $a \in A$, and π is a policy function mapping current states to next actions. As defined, the measures operate only on states and do not include an action (a_t) input—model-free RL learners will need to use the state-action equivalents.

Hierarchical Reinforcement Learning Workshop at the 31st Conference on Neural Information Processing Systems (HRL@NIPS 2017), Long Beach, CA, USA.

Table 1: Primitive measures (distances & densities)

Name	Notation	Recursive definition
Time similarity	$\Gamma(s_t, g)$	$\max_a \mathbb{E}^a[\gamma \cdot \Gamma'(s_{t+1}, g)]$, where $\Gamma'(s, g) := \Gamma(s, g)$ unless $s = g$, in which case $\Gamma'(g, g) := 1$.
Value distance[†]	$\texttt{vdist}(s_t, g)$	$\mathbb{E}^{\pi(s_t)}[r_t + \gamma \cdot \texttt{vdist}'(s_{t+1}, g)]$.
Event distance[†*]	$\texttt{xdist}(s_t, x, g)$	$I(s_t, x) + \mathbb{E}^{\pi(s_t)}[\gamma \cdot \texttt{xdist}'(s_{t+1}, x, g)]$.
Event density[††*]	$\texttt{maxsr}(s_t, x)$	$I(s_t, x) + \max_a \mathbb{E}^a[\gamma \cdot \texttt{maxsr}(s_{t+1}, g)]$.
Value	$V(s_t)$	$\max_a \mathbb{E}^a[r_t + \gamma \cdot V(s_{t+1})]$.

[*] $I(s, x)$ is 1 if $s = x$ and 0 otherwise. To generalize in the same way as [1] generalizes the SR to successor features, define x as a feature index (rather than a state) and redefine $I(s, x)$ as the xth feature of s.

[†] In each case, $\pi(s)$ is greedy with respect to $\mathbb{E}[\Gamma(s_{t+1}, g)]$ and $f' := f$ unless $s_{t+1} = g$, in which case $f' := 0$.

[††] Technically, this should be *maximum* event density. The sr in maxsr stands for successor representation [5].

Each measure represents interpretable world knowledge. For example, event distance, which is the discounted frequency of event x on the shortest path to goal g, could be used to represent an expectation about the number of red lights on the drive to work. Upon fixing the goal g and event x, each measure is the solution, in form of a value function, to a derivative MDP that inherits the world dynamics, but assumes a specific policy and has its own pseudo reward and termination functions. Value functions encoding world knowledge in this way are known as Generalized Value Functions (GVFs) and were proposed alongside the Horde architecture [23].

Horde endows an RL agent with a number of "demons", each of which learns an independent GVF in parallel with the agent. Universal Value Function Approximators (UVFAs) [19] generalize across GVFs by including g in their domain rather than definition, thereby representing an infinite Horde in a single function. This idea is easily extended to include x in the domain, so we can think of the defined measures as specific "interesting" types of universal value functions. Schaul et al. [19] describe two methods by which such functions can be learned in an RL setting: first, by training an initial Horde and then generalizing between GVFs, and second, by direct RL (i.e., using the state-action, stochastic approximation equivalents of the recursive definitions in Table 1). Their empirical results demonstrate that the measures defined in Table 1 can be learned as primitives.

3 Rules of reasoning

Table 2 proposes a small set of rules by which the measures defined in Table 1 can be used to reason about value. It is divided into rules for landmark-based and factor-based reasoning.

Landmark-based reasoning. Landmark bounds—inspired by A* with landmarks [9]—decompose a distance or density into what comes before a landmark state, ℓ, and what comes after. The validity of the lower bounds follows from their achievability: an agent has the "option" to first seek out the landmark, and then return to its greedy policy. Viewed in this way, the first leg of a landmark lower bound implicitly represents an *interruptible option* (cf., esp., pp. 196-200 of [24]).

Landmark lower bounds (e.g., on V) can be composed hierarchically, as in Figure 1 (left), by splitting either vdist or V. The latter split is a direct application of the landmark lower bound and produces an equal or looser bound. The former—a split of $\texttt{vdist}(s, \ell_1)$ with ℓ_2—increases the time to reach the original landmark and requires the discount factor in second component to be adjusted from $\Gamma(s, \ell_1)$ to $\Gamma(s, \ell_2) \cdot \Gamma(\ell_2, \ell_1)$.

Each landmark lower bound can be rearranged into two upper bounds (one shown for V and xdist), which are useful for reducing maximization bias [10], consistency-based learning (Section 4) and direct value estimation if the bound is close to equality (Section 4).

Factor-based reasoning. The bottom half of Table 2 is concerned with factoring values into the sources of rewards. This type of decomposition is common in human reasoning and, in absence of state and event abstraction (Appendix A), is best exemplified in RL by the successor representation

Table 2: Landmark and factor-based rules of reasoning

Landmark lower bound, V	$V(s)$	$\geq$	$\texttt{vdist}(s,\ell) + \Gamma(s,\ell) \cdot V(\ell)$
Landmark lower bound, Γ	$\Gamma(s,g)$	$\geq$	$\Gamma(s,\ell) \cdot \Gamma(\ell,g)$
Landmark upper bound, V	$V(s)$	$\leq$	$(V(l) - \texttt{vdist}(l,s))/\Gamma(l,s)$
Landmark upper bound, xdist	$\texttt{xdist}(s,x,g)$	$\leq$	$\texttt{maxsr}(s,x) - \Gamma(s,g) \cdot \texttt{maxsr}(g,x)$
Event-reward decomposition	$\texttt{vdist}(s,g)$	$\approx$	$\sum_{x_i \in X \subseteq S} \texttt{xdist}(s,x_i,g) \cdot r(x_i)$
Distance-density duality	$\texttt{maxsr}(s,g)$	$=$	$I(s,g) + \Gamma(s,g) \cdot (1/(1-\Gamma(g,g)))$
Distance-density lower bound	$V(s)$	$\geq$	$\texttt{maxsr}(s,s) \cdot \texttt{vdist}(s,s)$

[5, 17]. The successor representation (SR), however, is limited in the same way as V: it is a monolithic long-run estimator that depends critically on policy [1, 14].

The first rule—event-reward decomposition of vdist—allows the SR to be decomposed into segments for any landmark-delineated policy (cf. a policy over options). It is a substantial simplification in representation of the SR that comes at the cost of additional reasoning during action selection (a search over landmark-delineated policies). If all "major" sources x_i of reward are considered, event-reward decomposition will provide a good approximation of vdist. Cf., esp., universal option models (UOMs) [25] (xdist is directly comparable to the UOM occupancy function).

4 Metacognition

Multiple value function representations. The rules of Section 3 combine the primitives of Section 2 to make various statements about (form multiple representations of) themselves and, in particular, about the value function, V. The utility of this rests on certain of these statements being "better" than others. It is easy to see why this might be the case. In the extreme, consider that if a human teacher provides a known value for $\texttt{vdist}(s,\ell)$ ("partial supervision"), any statement about $V(s)$ that uses the labeled value can now be made with more confidence. Alternatively, in an environment with shifting rewards, xdist remains constant and can be learned with greater accuracy than vdist. These examples suggest one aspect of better representations: they can be stated with greater confidence.

Recursive reasoning; confidence. As humans, we apply reasoning to produce alternative representations when we are unsure of something. Our reasoning is idiosyncratic, depending greatly on our past experiences and personal characteristics. For RL agents to do the same (rather than apply some predefined form of fixed reasoning such as greedy policy selection), they must be able to evaluate their level of confidence in a given representation in context of their individual circumstances (e.g., their past experiences and computational budget). Then, an agent that is uncertain about a primitive estimate could attempt to improve its estimate by recursively applying known rules of reasoning. One approach to implementing this process is outlined in Algorithm Sketch 1.

Algorithm Sketch 1 Recursive reasoning

```
function REASON_TO(prim, budget)
    if CONFIDENT(prim, budget) then
        return prim
    end if
    estimates ← {prim}
    for rule in GENERATE(prim, budget) do
        REASON_TO required primitives
        Compute estimate using rule and
            add it to estimates
    end for
    return RESOLVE(estimates, budget)
end function
```

```
function CONFIDENT(prim, budget):
    Returns true if agent sufficiently confident in
    prim given budget.

function GENERATE(prim, budget):
    Generates rules (e.g., landmark bounds) that
    might produce useful representations of prim.

function RESOLVE(estimates, budget):
    Returns single estimate of prim based on the set
    estimates (not necessarily a member thereof).
    Optionally invokes consistency-based learning.
```

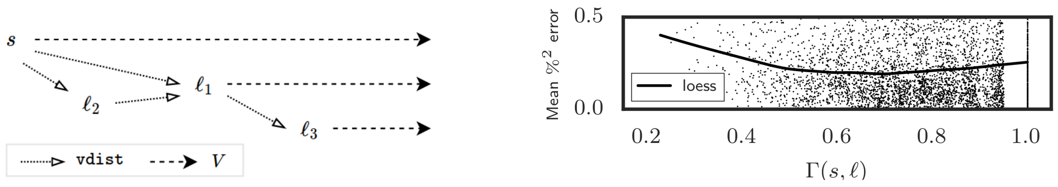

Figure 1: *Left*: A two-level decomposition of V using landmark bounds. *Right*: In a toy 25x25 wraparound gridworld transfer task ($\gamma = 0.95$, with stochastic movement in 2 directions only), intermediate landmarks (learned by Horde) offering true bounds on $V(s)$ within 10% of the true $V(s)$ provided more accurate estimates (in terms of mean squared percentage error to the true bound) than nearby or far landmarks after 100,000 steps.

But how confident is an agent in its estimates? How can the CONFIDENT function be evaluated? An answer requires knowledge about *primitive distributions* [20, 6, 7, 26, 2] and an augmented list of rules for its composition. This work is left for the full paper, but it is stressed that this is truly fundamental: without some measure of confidence in its estimates, an agent has no way of deciding when to reason and when not to, or how to choose amongst conflicting representations (in the RESOLVE function), *unless* the agent uses a fixed reasoning pattern.

Confidence estimates also have the potential to guide exploration and enable question-driven learning. An agent that can usefully bound an important long-run value estimate with three landmark-delineated segments could focus exploration on the segment of lowest confidence. Alternatively, an NLP-augmented agent could query a (possibly human) teacher about the value of that segment.

Landmark selection, tightness of bounds, and generation. Landmark selection (in the GENERATE function) impacts confidence in two ways. First, familiar landmarks offer better accuracy. The agent can spend time to get to know a few critical landmarks well (e.g., by targeted exploration, MCTS [3], or focused experience replay or Dyna [16, 22]), so that these landmarks produce high-confidence bounds. Second, the landmark bounds distribute uncertainty between two segments, reducing it when short distances are easier to estimate. This may often be the case; for instance, see Figure 1 (right).

Closely related to confidence is the "tightness" of landmark bounds, which also depends critically on landmark selection: a lower bound is tightest when the first leg of the bound is on the optimal path, and an upper bound is tightest when the landmark is directly behind the state or goal, as appropriate. A similar comment applies to factor-based reasoning: the event-reward decomposition will be more accurate the more representative the set X is of the reward sources between s and g.

It is clear then that reasoning-enabled agents should possess a high-quality generative model of landmarks (and of X sets), contingent on endpoints. Agents in possession of an `xdist` primitive are at advantage when training such a generator. A good landmark ℓ (or member of X) is often "on the way" to a goal g; in other words, $\texttt{xdist}(s, \ell, g) > 0$, so that the agent has an internal training target.

Consistency-based learning. Learning to generate landmarks from internal targets is an example of *consistency-based learning*, which is available whenever there are multiple representations of the same value, so that an agent can optimize for consistency. Learning might be invoked in one or both directions (e.g., if a bound is violated, apply one step of gradient descent to bring the less confident estimate toward the other). Popular examples include actor-critic methods [12] and TD learning [21].

It is notable that good rules of reasoning will be satisfied whenever primitives are accurate. E.g., Monte carlo value estimation will eventually satisfy the Bellman equation. Invoking the Bellman update, however, can speed this process up [21]. This advantage of consistency-based learning makes it particularly relevant in a multitask learning (MTL) setting [4]: we might estimate all primitives in a single neural MTL architecture to improve generalization (e.g., as in [19]); ideally, joint estimation will produce consistent estimates that *implicitly* satisfy the rules of reasoning. To train a such an architecture, however, explicit reasoning can help (cf. habit acquisition, human learning in general).

Language; interpretation and justification. Each rule of reasoning has a well-defined, explicit statement that interfaces to an agent's implicit understanding. This allows reasoning-enhanced agents to interpret their own behaviors and provide justification for their actions: "*I took action 2 because it takes me in the direction of landmark A, which is a good place to be; further, the path to A has a reasonable chance of encountering event X, which has been very rewarding.*"

References

[1] A. Barreto, R. Munos, T. Schaul, and D. Silver. Successor features for transfer in reinforcement learning. *arXiv preprint arXiv:1606.05312*, 2016.

[2] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458, 2017.

[3] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[4] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[5] P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

[6] R. Dearden, N. Friedman, and S. Russell. Bayesian q-learning. In *AAAI/IAAI*, pages 761–768, 1998.

[7] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208. ACM, 2005.

[8] R. Feynman. Richard Feynman on the scientific method. *Recorded lecture available on Youtube*, 1964.

[9] A. V. Goldberg and C. Harrelson. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 156–165. Society for Industrial and Applied Mathematics, 2005.

[10] H. V. Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

[11] N. Jiang, A. Kulesza, and S. Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 179–188, 2015.

[12] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[13] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez. Constructing symbolic representations for high-level planning. In *AAAI*, pages 1932–1938, 2014.

[14] L. Lehnert, S. Tellex, and M. L. Littman. Advantages and limitations of using successor features for transfer in reinforcement learning. *arXiv preprint arXiv:1708.00102*, 2017.

[15] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for MDPs. In *ISAIM*, 2006.

[16] L.-J. Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

[17] I. Momennejad, E. M. Russek, J. H. Cheong, M. M. Botvinick, N. Daw, and S. J. Gershman. The successor representation in human reinforcement learning. *Nature Human Behavior*, 1:680–692, 2017.

[18] M. Ponsen, M. E. Taylor, and K. Tuyls. Abstraction and generalization in reinforcement learning: A summary and framework. In *International Workshop on Adaptive and Learning Agents*, pages 1–32. Springer, 2009.

[19] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320, 2015.

[20] M. J. Sobel. The variance of discounted markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

[21] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[22] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the seventh international conference on machine learning*, pages 216–224, 1990.

[23] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.

[24] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[25] C. Szepesvari, R. S. Sutton, J. Modayil, S. Bhatnagar, et al. Universal option models. In *Advances in Neural Information Processing Systems*, pages 990–998, 2014.

[26] A. Tamar, D. D. Castro, and S. Mannor. Temporal difference methods for the variance of the reward to go. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 495–503, 2013.

A Extensions

Rule acquisition and validation. The rules in Table 2 are predefined by the human designer. Would it be possible for an agent to discover rules on its own? One might find inspiration for a method of autonomous rule acquisition in the following quote by physicist Richard Feynman on the scientific method: to discover a new rule, "*first, we guess it; then we compute the consequences of the guess; and then we compare those computation results to nature, or experiment, or experience ... if [the rule] disagrees with experiment, it's wrong*" [8]. As an alternative to guessing, an agent might learn of a rule through verbal discussion with humans, after which it could apply consistency-based validation to accept or reject the rule. If the rule is accepted, the agent can able consistency-based learning offline in order to integrate it into its current base of knowledge and experience.

Abstraction. As practical settings require function approximation and state abstraction, the definitions in Tables 1 and 2 should be reformulated with respect to *sets of states*. The best way to learn and represent useful sets of states is not yet clear, but a growing body of work offers insights [15, 18, 11, 13]. Regardless of the approach, it should be applied using the same three-part framework (primitives + rules + metacognition). This would entail a rich set of definition ($=$), negation ($\neg$), inclusion ($\in, \subseteq$), exclusion ($\notin, \nsubseteq$), preference ($\succ, \nsucc$) and composition ($\cup, \cap$) rules for operating on set primitives and allow the agent to reason recursively at multiple levels of abstraction. The points above—confidence, generation, consistency-based learning, rule acquisition and language—would apply *mutatis mutandis*. Relatedly, full human-like utilization of factor-based reasoning requires a method for event abstraction (a red light observed over multiple steps should be counted only once).

Continuation primitives. The described mechanism strictly adds complexity to ordinary action selection. To simplify, as do temporally abstract options by applying a fixed in-option policy, one could introduce a *plan-continuation primitive*. Also potentially useful given a computation budget is a *reasoning-continuation primitive* that would allow the agent to stop and think.