

# When Policies Can Be Trusted: Analyzing a Criteria to Identify Optimal Policies in MDPs with Unknown Model Parameters

**Emma Brunskill**

Computer Science Division  
University of California, Berkeley  
emma@cs.berkeley.edu

## Abstract

Computing a good policy in stochastic uncertain environments with unknown dynamics and reward model parameters is a challenging task. In a number of domains, ranging from space robotics to epilepsy management, it may be possible to have an initial training period when suboptimal performance is permitted. For such problems it is important to be able to identify when this training period is complete, and the computed policy can be used with high confidence in its future performance. A simple principled criteria for identifying when training has completed is when the error bounds on the value estimates of the current policy are sufficiently small that the optimal policy is fixed, with high probability. We present an upper bound on the amount of training data required to identify the optimal policy as a function of the unknown separation gap between the optimal and the next-best policy values. We illustrate with several small problems that by estimating this gap in an online manner, the number of training samples to provably reach optimality can be significantly lower than predicted offline using a Probably Approximately Correct framework that requires an input  $\epsilon$  parameter.

Many real-world planning challenges take place in uncertain, stochastic domains. Such problems are even more challenging when the reward and dynamics model parameters that describe the domain are initially unknown. It is necessary to implicitly or explicitly estimate these parameters in order to compute a good plan for said domains, even though the model parameters themselves are not of interest.

In a number of interesting planning domains, such as game playing agents, space robots such as the Mars rover or Robonaut, or medical applications including epilepsy management (Guez et al. 2008), it may be reasonable to expect to have an extended training period, during which suboptimal plans can be computed and tried. However, after training completes, it is important to ensure the resulting policy is optimal with high probability. For example, a patient could be monitored in a hospital until drug dosage schedules have been tuned for the patient, but mistakes made when the patient is far from medical facilities could be severe. In such applications, we wish to know when the computed policy

can be used with high confidence in its future performance. We analyze a criteria for this optimal policy identification.

Computing a policy, a mapping of states to actions, while learning the model parameters falls under the class of reinforcement learning (RL). There has been a large body of approaches to balancing the tradeoff between model parameter identification and planning using the existing estimated models, commonly known as the tradeoff between exploration and exploitation. Perhaps the most promising approaches for this tradeoff are recent techniques that pose this challenge as a partially observable Markov decision process (POMDP) planning problem, by computing plans over the cross product of the system state and model parameters (see for example Poupart et al. (2006), Asmuth et al. (2009), and Kolter and Ng (2009)). Another popular approach for solving planning problems with initially unknown model parameters are Probably Approximately Correct (PAC) reinforcement learning algorithms, such as Brafman and Tenenholz's (2002) R-max algorithm. For an input  $\delta$  and  $\epsilon$ , PAC RL algorithms guarantee that each action selected will have a value that is  $\epsilon$ -close to the value of the optimal action, on all but a number of steps that is polynomial function of  $\delta$  and  $\epsilon$ , with probability at least  $1 - \delta$ . Neither POMDP-nor PAC-style approaches explicitly seek to identify when the optimal policy has been found with high likelihood. The criteria we discuss for identifying the optimal policy can be used in conjunction with these and a wide variety of other techniques for planning with unknown model parameters.

We consider domains described as finite-state and finite-action Markov decision processes (MDPs). The optimal policy for an MDP has been identified when the error bounds on the estimated state-action values do not change the resulting policy, with high probability. This criteria for identifying the optimal MDP policy is essentially identical to the action elimination and stopping criteria presented by Even-Dar and colleagues (2006). We extend this prior work by providing an upper bound on the number of required samples in order to identify the optimal policy as a function of the unknown separation gap between the optimal and next-best state-action values. We illustrate with several small problems that by estimating this gap in an online manner, the number of training samples to provably reach optimality can be significantly lower than predicted by PAC-RL approaches that requires an input  $\epsilon$  parameter.

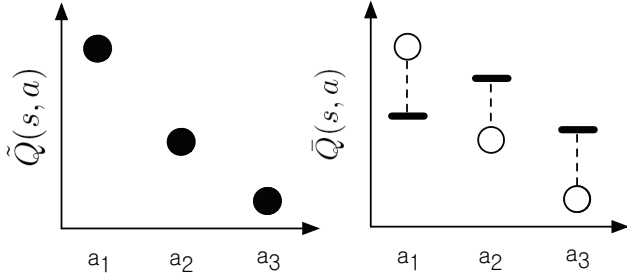


Figure 1: (left)  $\tilde{Q}$  estimates and (right) the  $\bar{Q}$  values (shown by thick horizontal lines) which use the lower bound of the best  $\tilde{Q}$  action, and the upper bounds of all other actions.  $a_2$  maximizes the  $\bar{Q}$  value, but  $a_1$  maximizes  $\tilde{Q}$ , showing the policy for this state could change given the current error bounds, and is not yet guaranteed to be optimal.

## Background

A MDP is a tuple  $\langle S, A, T_{s,a}(s'), R(s, a), \gamma \rangle$ , where  $S$  and  $A$  are the discrete set of states and actions;  $T_{s,a}(s')$  is the dynamics model that expresses the probability of starting in state  $s$ , taking action  $a$  and arriving in state  $s'$ ;  $R(s, a)$  is the deterministic<sup>1</sup> reward received from taking action  $a$  in state  $s$ ; and  $\gamma$  is the discount factor. All rewards are assumed to lie between 0 and a known  $R_{max}$ .

The goal is to learn a policy  $\pi : S \rightarrow A$ . The value of a policy  $\pi$  for a state  $s$  is the expected sum of future rewards from following policy  $\pi$  starting in state  $s$ :

$$V^\pi(s) = E \left[ \sum_{j=0}^{\infty} \gamma^j r(s_j, \pi(s_j)) \mid s_0 = s \right],$$

where  $r(s_j, \pi(j))$  is the reward received at step  $j$ ,  $s_0$  is the initial state and the expectation is taken with respect to the transition dynamics. Similarly, the state-action value  $Q^\pi(s, a)$  is :

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} T_{s,a}(s') V^\pi(s'). \quad (1)$$

Initially the parameters of the reward  $R$  and transition model  $T$  are unknown.

## Optimal Policy Identification

In this section we describe a procedure for identifying when the optimal policy has been found, with high probability. This procedure is semantically equivalent to the stopping criteria presented by Even-Dar and colleagues (2006) though our presentation is slightly different.

The key idea is to maintain uncertainty bounds around the estimates of the state-action values, and consider whether the best action for a particular state could change given these bounds: see Figure 1 for a graphical illustration. The state-action values depend on the current dynamics model parameter estimates which are computed from the observed state-action-next state transitions. Let  $\tilde{V}$  and  $\tilde{Q}$  be the state and

<sup>1</sup>We believe the results are easily extendable to unknown, stochastic rewards.

state-action values, respectively, computed using the estimated dynamics model  $\tilde{T}$ , and let  $V$  and  $Q$  be the optimal state and state-action values computed using the unknown true dynamics model  $T$ . The difference between the optimal and estimated state-action values is

$$\Delta_Q(s, a) = |Q(s, a) - \tilde{Q}(s, a)|.$$

We will use  $\Delta_Q^{max}$  to denote the maximum such difference over all state-action pairs. We can then substitute in the expression for  $Q(s, a)$  from Equation 1,

$$\begin{aligned} \Delta_Q(s, a) &= \gamma \left| \sum_{s'} T_{s,a}(s') V(s') - \sum_{s'} \tilde{T}_{s,a}(s') \tilde{V}(s') \right| \\ &= \gamma \left| \sum_{s'} T_{s,a}(s') (V(s') - \tilde{V}(s')) + \sum_{s'} \tilde{V}(s') (T_{s,a}(s') - \tilde{T}_{s,a}(s')) \right| \\ &\leq \gamma \left| \sum_{s'} T_{s,a}(s') \max_{a_2} (Q(s', a_2) - \tilde{Q}(s', a_2)) \right| + \gamma \tilde{V}_{max} L_1(T_{s,a}, \tilde{T}_{s,a}) \\ &\leq \gamma \Delta_Q^{max} + \gamma \tilde{V}_{max} L_1(T_{s,a}, \tilde{T}_{s,a}) \end{aligned} \quad (2)$$

where we have added and subtracted  $T_{s,a}(s') \tilde{V}(s')$ , used the triangle inequality, upper bounded  $\tilde{V}$  by its maximum value  $\tilde{V}_{max}$ , and used the definition of the  $L_1$  norm,  $L_1(T_{s,a}, \tilde{T}_{s,a}) \equiv \sum_{s' \in S} |T_{s,a}(s') - \tilde{T}_{s,a}(s')|$ . Equation 2 must hold for  $\Delta_Q^{max}$ ,

$$\begin{aligned} \Delta_Q^{max} &\leq \gamma \Delta_Q^{max} + \gamma \tilde{V}_{max} L_1(T_{s_m, a_m}, \tilde{T}_{s_m, a_m}) \\ \Delta_Q^{max} &\leq \frac{\gamma \tilde{V}_{max} L_1(T_{s_m, a_m}, \tilde{T}_{s_m, a_m})}{1 - \gamma}, \end{aligned} \quad (3)$$

where  $s_m$  and  $a_m$  are the state-action pair with the largest  $L_1$  error. Therefore the maximum error will be bounded above by the largest  $L_1$  difference between the transition models (over all state-action pairs) and the maximum value  $\tilde{V}_{max}$ . This is a known result that has been used in past PAC RL proofs (see e.g. Strehl and Littman, 2005).

Given a bound on the  $L_1$  norm of the transition model that holds with probability at least  $1 - \delta$ , Equation 3 can be used to determine if the policy has converged with high probability to the optimal plan (see Algorithm 1). Briefly, the algorithm returns that convergence has not occurred if any state-action pairs have not yet been sampled, since this is required in order to obtain an estimate of, and a well-defined bound on, the transition dynamics of each state-action pair. After all state-action pairs are sampled at least once, the deterministic reward model will be known exactly. On any future time steps, Algorithm 1 involves computing an error bound on the state-action values and checking if the best action for any state-action pair could change given the potential error in the estimated state-action values. If the best action stays the same for all states, then the policy has converged. To handle states where there are two or more actions with the same optimal state-action value, in addition to  $\delta$  the user should also provide an error bound  $\epsilon_{min}$ : Algorithm 1 returns true when the optimal policy has been reached or the maximum state-action value error is  $\epsilon_{min}$ .

## Bounding the Transition Model Error

To compute an upper bound on the  $L_1$  distance between the estimated and true dynamics models, we estimate con-

confidence bounds on the model parameters. Since the dynamics models are multinomials, there exist known confidence intervals, developed by Weissman et al. (2003). Strehl and Littman (2005) extended these bounds using the union bound and results from Fong (1995), to a reinforcement learning context: their bound ensures that the computed intervals are consistent over all state-action pairs, at each time step. More precisely, from Strehl and Littman we know that for a given  $\delta$ , with probability at least  $1 - \delta$ , the  $L_1$  distance between the estimated transition model for a state-action pair  $\tilde{T}_{s,a}$  and the true transition model  $T_{s,a}$  is at most:

$$L_1(\tilde{T}_{s,a}, T_{s,a}) \leq \sqrt{\frac{2 \ln\left(\frac{(2^{|S|}-2)2|S||A|\pi^2}{3\delta}\right) + 4 \ln(n_{s,a})}{n_{s,a}}}. \quad (4)$$

where  $n_{s,a}$  is the number of times action  $a$  has been taken from state  $s$ , and  $\pi$  is the circle constant, not the policy.

### Convergence to Optimal

Algorithm 1 provides a criteria for halting exploration. However, so far it is not clear how good the online criteria of Algorithm 1 is, or how it might compare to a PAC-RL algorithm which provides an offline formula for the number of required samples needed to provide accuracy guarantees on the resulting policy. We now provide promising evidence of the benefit of using the online criteria of Algorithm 1.

Let  $g$  be the minimal separation gap between the state-action values of the optimal action  $a^*$  and next-best action:

$$g \equiv \min_s \left( Q(s, a^*) - \max_{a \text{ s.t. } a \neq a^*} Q(s, a) \right).$$

Note that  $g$  will not be known in advance, which is the motivation behind using the online convergence criteria. Indeed, Algorithm 1 can identify the optimal policy when the estimated error in the state-action values,  $\Delta_Q^{max}$ , becomes equal or smaller than the gap  $g/2$ , since for this any smaller error bounds, the optimal policy does not change. Essentially  $\Delta_Q^{max}$  provides an online estimate of the gap  $g$ .

We now bound the number of samples required to achieve optimal performance with high probability as a function of the unknown separation gap.

**Theorem 1.** *Given any  $\delta > 0$ , separation  $g$ , and known maximum reward  $R_{max}$ , define*

$$N_{s,a} = \frac{8R_{max}^2\gamma^2}{g^2(1-\gamma)^4} \left( \ln\left(\frac{(2^{|S|}-2)2|S||A|\pi^2}{3\delta}\right) + 4 \ln\left(\frac{16R_{max}^2\gamma^2}{g^2(1-\gamma)^2}\right) \right). \quad (5)$$

*Then if there are at least  $N_{s,a}$  transition samples for each state-action pair  $(s, a)$  then with probability at least  $1 - \delta$  the computed policy using the estimated transition model parameters  $\tilde{T}$  will be optimal.*

*Proof.* (Sketch) If the state-value uncertainty bounds are less than or equal to  $g/2$  then the policy does not change when these the error bounds are incorporated. Therefore we

---

### Algorithm 1 OptimalPolicyReached

---

**Input:** estimated state-action values  $\tilde{Q}$ , transition counts  $n_{s,a}$  for all state-action pairs,  $\delta, \gamma, \epsilon_{min}$   
**if**  $\exists n_{s,a} < 1$  **then**  
    return False;  
**end if**  
 $\forall s \in S, \forall a \in A$ , compute  $L_1(T_{s,a}, \tilde{T}_{s,a})$  using Eqn. 4.  
Compute  $\Delta_Q^{max}$  using Eqn. 3  
**for**  $s \in S$  **do**  
     $\tilde{a}_s^* = \max_a \tilde{Q}(s, a)$   
     $\tilde{Q}(s, \tilde{a}_s^*) \equiv \tilde{Q}(s, \tilde{a}_s^*) - \Delta_Q(s, \tilde{a}_s^*)$   
     $\tilde{Q}(s, a) \equiv \tilde{Q}(s, a) + \Delta_Q(s, a) \quad \forall a \neq \tilde{a}_s^*$   
    **if**  $\text{argmax}_a \tilde{Q}(s, a) \neq \tilde{a}_s^*$  and  $\Delta_Q^{max} > \epsilon_{min}$  **then**  
        return False;  
    **end if**  
**end for**  
return True;

---

require  $\Delta_Q^{max}$  to be at most  $g/2$ . From Equation 3 we see that to ensure  $\Delta_Q^{max} \leq g/2$  it is sufficient to require:

$$\left( \gamma \tilde{V}_{max} L_1(T_{s_m, a_m}, \tilde{T}_{s_m, a_m}) \right) / (1 - \gamma) = g/2, \quad (6)$$

as the left-hand expression is an upper bound for  $\Delta_Q^{max}$ . We then substitute an upper bound for  $\tilde{V}_{max} \leq R_{max}/(1 - \gamma)$  and solve for the error in the transition model:

$$L_1(T_{s_m, a_m}, \tilde{T}_{s_m, a_m}) = \frac{g(1-\gamma)^2}{2\gamma R_{max}}. \quad (7)$$

To ensure Equation 7 holds with probability at least  $1 - \delta$  it is sufficient (from Equation 4) to ensure an upper bound on the  $L_1$  error is bounded by the right-hand side of Equation 7:

$$\sqrt{2 \ln\left(\frac{(2^{|S|}-2)2|S||A|\pi^2}{3\delta}\right) + 2 \ln(N_{s,a})} / N_{s,a} = \frac{g(1-\gamma)^2}{2\gamma R_{max}}.$$

Solving for  $N_{s,a}$  yields

$$N_{s,a} = \frac{8R_{max}^2\gamma^2}{g^2(1-\gamma)^4} \ln\left(\frac{(2^{|S|}-2)2|S||A|\pi^2}{3\delta}\right) + \frac{16R_{max}^2\gamma^2}{g^2(1-\gamma)^4} \ln(N_{s,a}). \quad (8)$$

The above expression is equivalent to  $n = D + C \ln(n)$  where  $D$  and  $C$  are positive constants. It is well known fact (used in the proofs of Strehl and Littman (2005), among others) that if  $N \geq 2C \ln(C)$  then  $N \geq C \ln(N)$ . This implies if  $N \geq C \ln(N)$  then  $2C \ln(C) \geq C \ln(N)$ , which implies  $D + 2C \ln(C) \geq D + C \ln(n)$  and therefore it is sufficient to satisfy Equation 8 to set  $N_{s,a}$  as

$$N_{s,a} = \frac{8R_{max}^2\gamma^2}{g^2(1-\gamma)^4} \ln\left(\frac{(2^{|S|}-2)2|S||A|\pi^2}{3\delta}\right) + \frac{32R_{max}^2\gamma^2}{g^2(1-\gamma)^4} \ln\left(\frac{16R_{max}^2\gamma^2}{g^2(1-\gamma)^4}\right), \quad (9)$$

which is the defined  $N_{s,a}$  in our theorem.  $\square$

## Sample Bounds

The above bound is very similar to the bounds produced in Probably Approximately Correct planning with unknown model parameters, the key difference is that our bound is defined in terms of  $g$  instead of an input parameter  $\epsilon$ . We now provide several example problems where we can directly solve the MDP and calculate  $g$  explicitly in order to demonstrate that  $g$  may be larger than an arbitrary  $\epsilon$  chosen offline for a PAC-RL-style algorithm. These results imply that for these problems, if Algorithm 1 was used to identify the optimal policy by calculating an online estimate of the separation  $g$ , the number of samples required would be fewer than the offline number of samples computed by PAC-RL algorithms that commit to an overly conservative  $\epsilon$ .

We consider three sample MDP problems. Chain is a 9-state MDP used by Dearden, Friedman and Russell (1998). PittMaze MDP (see Figure 2) is a 21-state grid maze MDP with 4 cardinal-direction actions. When Actions succeed with probability 0.6: with 0.2 probability the agent goes in a perpendicular direction, unless there is a wall. At the goal the agent transitions to a sink terminal state. Rewards are 0 for self-looping pits, 0.5 for the goal, and 0.495 for all other states. The agent can start in any non-pit state. PittMaze2 is the same as PittMaze1 with re-arranged pits and start states.

For each MDP, we computed the sample complexity bound  $N_{s,a}$  we used Equation 5 with either  $g$  or an alternate smaller  $\epsilon$  in place of  $g$  which (from the proof of Theorem 1) guarantees the resulting maximum state-action error bound  $\Delta_Q^{max}$  is at most  $g/2$  or  $\epsilon/2$ , respectively.

Table 1 shows the sample complexity results. In each case, the minimum separation  $g$  is such that the maximum number of samples  $N_{s,a}$  per state-action pair to reach the optimal policy is an order of magnitude or smaller than might be expected by a naïve selection of the  $\epsilon$  parameter.

These sample problems suggest that by estimating  $g$  online and checking repeatedly whether the optimal policy has been identified using Algorithm 1, we may need fewer samples to guarantee optimal performance than in PAC-RL approaches which offline choose an  $\epsilon$  and use this to bound the number of samples for  $\epsilon$ -optimal performance.

## Conclusion

We presented a formal bound on the number of samples required to identify an optimal policy with high probability in a MDP with initially unknown model parameters as a function of the unknown gap separation  $g$ , which is implicitly

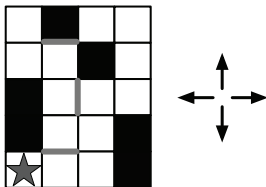


Figure 2: Pitt maze domain. Pitts are black squares, walls are grey lines, and the goal is the star state.

Table 1: The number of samples per state-action pair as a function of  $\Delta_Q^{max}(= 2\epsilon)$  or the minimal separation  $g$ .  $\gamma = 0.9$  and  $\delta = 0.5$ .

PROBLEM	G	$\Delta_Q^{max}$	$N_{s,a}$	$V_{max}$
CHAIN	0.39	$G/2$	$5 * 10^6$	4.3
		0.01	$2 * 10^9$	4.3
PITTMAZE1	0.10	$G/2$	$7 * 10^7$	3.8
		0.01	$2 * 10^9$	3.8
PITTMAZE2	1.87	$G/2$	$1 * 10^6$	10.5
		0.1	$1 * 10^8$	10.5
		0.01	$1 * 10^{10}$	10.5

estimated online in Algorithm 1. Our calculation of the sample complexity bounds for several sample problems provide evidence that Algorithm 1 may outperform alternate exploration halting criteria. Though more work is required for these bounds to be practical, our results suggest that focusing on optimal policy identification, instead of minimum errors in the optimal values, may reduce the amount of training required to be highly confident in the computed policy's future performance.

## References

- Asmuth, J.; Li, L.; Littman, M.; Nouri, A.; and Wingate, D. 2009. A Bayesian sampling approach to exploration in reinforcement learning. In *UAI*.
- Brafman, R. I., and Tennenholtz, M. 2002. R-MAX—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3:213–231.
- Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian Q-Learning. In *AAAI*.
- Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research* 7:1079–1105.
- Fong, P. 1995. A quantitative study of hypothesis selection. In *ICML*.
- Guez, A.; Vincent, R.; Avoli, M.; and Pineau, J. 2008. Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *IAAI*.
- Kolter, Z., and Ng, A. 2009. Near-Bayesian exploration in polynomial time. In *ICML*.
- Poupart, P.; Vlassis, N.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete Bayesian reinforcement learning. In *ICML*.
- Strehl, A., and Littman, M. L. 2005. A theoretical analysis of model-based interval estimation. In *ICML*.
- Weissman, T.; Ordentlich, E.; Seroussi, G.; Verdu, S.; and Weinberger, M. J. 2003. Inequalities for the L1 deviation of the empirical distribution. Technical report, HP Labs.