PAC Optimal Exploration in Continuous Space Markov Decision Processes

Jason Pazis and Ronald Parr
Department of Computer Science, Duke University
Durham, NC 27708
{jpazis,parr}@cs.duke.edu

Abstract

Current exploration algorithms can be classified in two broad categories: Heuristic, and PAC optimal. While numerous researchers have used heuristic approaches such as ϵ-greedy exploration successfully, such approaches lack formal, finite sample guarantees and may need a significant amount of fine-tuning to produce good results. PAC optimal exploration algorithms, on the other hand, offer strong theoretical guarantees but are inapplicable in domains of realistic size. The goal of this paper is to bridge the gap between theory and practice, by introducing C-PACE, an algorithm which offers strong theoretical guarantees and can be applied to interesting, continuous space problems.

1 Introduction and motivation

Efficient exploration is a central concept in Reinforcement Learning (RL). Contrary to the more straightforward, non-sequential, supervised active learning setting, we are often unable to collect samples from different parts of the state-action space at will. Samples must come from trajectories which depend on the dynamics of the underlying Markov Decision Process (MDP), which means that sampling certain states can be quite improbable unless we are actively trying to reach those states. Furthermore, even in cases where we have a generative model of the environment, it may be the case that only a tiny percentage of the state space is reachable from the starting state, and learning by collecting samples from the entire space would be very inefficient.

A number of relatively recent papers have proposed new algorithms, or analyzed known methods, to explore efficiently in unknown MDPs. Unfortunately, even after more than a decade of incremental improvements on complexity bounds, experimental results on challenging, realistic applications are absent. Instead, it appears that research on near optimal exploration is still focused on small, discrete state-action spaces. This is in stark contrast with the rest of the field, which has emphasized continuous and/or large MDPs.[1]

This paper contributes C-PACE, a new algorithm for exploration in continuous state MDPs, which is guaranteed to perform within some constant ϵ of the optimal policy on all but a small number of steps with high probability. In addition to our theoretical contribution, to demonstrate the applicability of the proposed approach in realistic problems, we present experimental results on a challenging six dimensional continuous HIV treatment domain.

2 Background

A *Markov Decision Process* (MDP) is a 5-tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space of the process, $\mathcal{A}$ is the action space, P is a Markovian transition model $\big(p(s'|s,a)$ denotes the probability density of a transition to state s' when taking action a in state $s\big)$, R is a reward function $\big(R(s,a,s')$ is the expected reward for taking action a in state s and transitioning to state $s'\big)$, and $\gamma \in [0,1)$ is a discount factor for future rewards. A *deterministic policy* π for an MDP is a mapping $\pi : \mathcal{S} \mapsto \mathcal{A}$ from states to actions; $\pi(s)$ denotes the action choice in state s. The value $V^\pi(s)$ of a state s under a policy π is defined as the expected, total, discounted reward when the process begins in state s and all decisions are made according to policy π. There exists an optimal policy π^* for choosing actions which yields the optimal value function $V^*(s)$, defined recursively via the Bellman optimality equation: $V^*(s) = \max_a\{\int_{s'} p(s'|s,a)\left(R(s,a,s') + \gamma V^*(s')\right)\}$. $Q^\pi(s,a)$ and $Q^*(s,a)$ are similarly defined when action a is taken at the first step.

In reinforcement learning, a learner interacts with a stochastic process modeled as an MDP and typically observes the state and immediate reward at every step; however, the transition model P and the reward function R are not accessible. The goal is to learn an optimal policy using the experience collected through interaction with the process. At each step of interaction, the learner observes the current state s, chooses an action a, and observes the resulting next state s' and the reward received r, essentially sampling the transition model and the reward function of the process. Thus experience comes in the form of (s,a,r,s') samples.

There have been many definitions of sample complexity in various RL settings. For the purposes of this paper, we employ the following definition due to Kakade (2003):

[1] While some research in PAC optimal exploration for continuous spaces exists (Kakade, Kearns, and Langford 2003), it does not offer a concrete, practical algorithm. See section 4 for more details.

Definition 2.1. *The sample complexity of exploration of an algorithm is the number of time steps t such that $V^{\pi}(s_t) < V^*(s_t) - \epsilon$.*

Discrete, PAC optimal exploration algorithms measure their efficiency in terms of the number of states and actions. In continuous state-action spaces we have to use a definition which relates to the covering number of the space:

Definition 2.2. *An exploration algorithm is said to be **efficient** if its sample complexity is polynomial in the covering number of the state-action space $\mathcal{N}_{SA}(L_{\tilde{Q}}, \epsilon_d)$ (see Definition 3.2).*

3 C-PACE: Continuous PAC optimal exploration

In many interesting problems, we do not start with a representative set of samples. Instead, acquiring samples is part of the learning process, and we are not only evaluated on the quality of the resulting policy, but also on the sample complexity of the algorithm.

While such criteria can be used for almost any problem, the most interesting cases from a practical perspective are problems with complicated dynamics and large state spaces, which cannot be sufficiently covered with a reasonable number of samples from a policy which selects actions uniformly at random (which is often what is used when we lack an expert policy).[2] Additionally, while not strictly necessary for such problems to be interesting, we will assume that in general we cannot reset the process to any desired state (we do not have a generative model).

This section introduces C-PACE, an algorithm for PAC optimal exploration in continuous state spaces. Instead of discretizing the state-action space or using a parametric approximation technique, C-PACE assumes that there exists some distance metric in which the state-action value function is smooth. For simplicity of exposition, our choice of smoothness measure will be Lipschitz continuity, although C-PACE can easily be extended to support other forms of smoothness (such as Hölder continuity), simply by pushing the complexity into the distance function. A Lipschitz continuous Q-value function means that for any two state-action pairs we have $|Q(s,a) - Q(s',a')| \leq L_Q d(s,a,s',a')$. Even though it may be statistically impossible to sample transitions out of a particular state-action pair more than once, Lipschitz continuity allows us to use samples from nearby state-actions to approximate Bellman's equation without introducing too much error. We will make this notion more concrete later in this section.

Similarly to other work in the field of PAC optimal exploration (Strehl and Littman 2005), C-PACE is based on the intuition that if we act according to the most optimistic scenario consistent with our observations, we'll either achieve good performance (if our optimistic assumption turns out to be true), or learn something new about the environment.

C-PACE's optimism in the face of uncertainty comes in the way the estimate of Bellman's equation is calculated. Instead of performing a simple average over nearby samples, an additional distance-dependent term is added. This term accounts for the maximum difference between the value of the state-action each sample originated from, and the value of the state-action whose Bellman equation we are trying to estimate.

Definitions and assumptions

For simplicity we assume that all rewards lie in $[0, R_{\max}]$ and $0 \leq Q_{\max} \leq \frac{R_{\max}}{1-\gamma}$.[3] $V^{\pi}(s,T)$ denotes the T step truncated, discounted value function for policy π.

Assumption 3.1. *We are given access to an approximate nearest neighbor algorithm*[4] *which when queried about the k nearest neighbors from a set of N points, returns one of cN sets of k distinct approximate nearest neighbors, where if d is the distance of the true k-th nearest neighbor, the approximate k-th nearest neighbor is no more than $c_m(d + c_a)$ away.*[5]

Definition 3.2. *The approximate covering number $\mathcal{N}_{SA}(L_{\tilde{Q}}, \epsilon_d)$ of a state-action space is the size of the largest minimal set C of state-action pairs, such that for any (s,a) reachable from the starting state(s), there exists $(s',a') \in C$ such that $L_{\tilde{Q}} c_m \left[d(s,a,s',a') + c_a\right] \leq \epsilon_d$.*

Definition 3.3. *A state-action pair (s_i, a_i) is considered "known" if $L_{\tilde{Q}} d(s_i, a_i, s_k, a_k) \leq \epsilon_d$, where (s_k, a_k) is its k-th approximate nearest neighbor in the sample set.*

Definition 3.4. *The value x_i of each sample of Bellman's equation is defined as the sampled reward plus the discounted, approximate, optimistic value of the sampled next state $\tilde{V}(s) = \max_a \tilde{Q}(s,a)$:*

$$x_i = x_{(s_i, a_i, r_i, s'_i)} = r_i + \gamma \tilde{V}(s'_i)$$

Definition 3.5. *For state-action (s_i, a_i), the approximate optimistic Q value function is defined as:*

$$\tilde{Q}(s_i, a_i) = \frac{\sum_{j=1}^{k} \left(\min \left\{ \tilde{Q}_{\max}, x_j + L_{\tilde{Q}} d_{ij} \right\} \right)}{k} \quad (1)$$

where $\tilde{Q}_{\max} = R_{\max} + \gamma Q_{max}$, $d_{ij} = d(s_i, a_i, s_j, a_j)$ and $j = 1$ to k are selected to be the k approximate nearest sampled neighbors to state-action (s_i, a_i) with $d_{ij} \leq \frac{Q_{\max}}{L_{\tilde{Q}}}$. If fewer than k approximate neighbors exist within $\frac{Q_{\max}}{L_{\tilde{Q}}}$, substitute the value of each missing neighbor with $\tilde{Q}_{\max}$.[6]

[2] It is easy to prove that for some MDPs, certain states can be exponentially unlikely (in the size of the state-action space) to be reached by a policy which selects actions uniformly at random.

[3] It is easy to satisfy the above assumption in all MDPs with bounded rewards by simply shifting the reward space.

[4] In this work we are interested in large multi-dimensional sample sets, for which the cost of exact nearest neighbor queries quickly becomes prohibitive.

[5] A very simple scheme for which $c = 1$, $c_m = 2$ and $c_a = 0$ is to cache the k-1 nearest neighbors for each sample, and answer k-NN queries by returning the 1-NN and its k-1 cached nearest neighbors.

[6] Note that if we have a bound better than $\frac{R_{\max}}{1-\gamma}$ for $Q_{\max}$, we may have $R_{\max} + \gamma Q_{\max} > Q_{\max}$.

The C-PACE algorithm

Given the above, C-PACE can be summarized as follows:

1. From state s, select and perform an action according to $\arg\max_a \tilde{Q}(s,a)$.
2. If (s,a) executed in step 1 is not known, add (s,a,r,s') to the sample set, and find the fixed point solution to $\tilde{Q}$.
3. Go to step 1.

We will not try to categorize C-PACE as a model-free or model-based algorithm. On the one hand, C-PACE integrates experience from all samples at every step, a trait commonly associated with model-based algorithms (Szita and Szepesvári 2010). On the other hand, it does not explicitly build a model, and its space requirements are less than what is typically required to build an accurate model, which have been defined as sufficient conditions for an algorithm to be considered model free by some authors (Strehl et al. 2006).

The rest of this section is devoted to proving that C-PACE explores efficiently. Intuitively, the proof is based on two key facts. The first is that at each step, C-PACE will either perform near optimally, or learn something new about the environment with high probability. The second is that there is only a finite number of things C-PACE can learn about the environment, so the number of suboptimal steps must be bounded.

Basic lemmata

Lemma 3.6. *(Lemma 4.5 in Kakade, Kearns, and Langford (2003)) All state-actions reachable from the starting state(s) will become known after adding at most* $k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)$ *samples.*

Lemma 3.7. *(Lemma 2 in Kearns and Singh (2002)) If* $T \geq \frac{1}{1-\gamma}\ln\frac{R_{\max}}{\epsilon_T}$, *then* $|V^\pi(s,T) - V^\pi(s)| \leq \frac{\epsilon_T}{1-\gamma}$.

Although the following lemma was first stated for discrete state-action spaces, the proof readily extends to continuous state-action spaces if we allow for K to be an infinite set:

Lemma 3.8. *(Generalized Induced Inequality) (Lemma 8 in Strehl and Littman 2005) Let M be an MDP, K a set of state-action pairs, M′ an MDP equal to M on K (identical transition and reward functions),* π *a policy, and T some positive integer. Let* A_M *be the event that a state-action pair not in K is encountered in a trial generated by starting from state* s_1 *and following* π *for T steps in M. Then:*

$$V^\pi_M(s_1,T) \geq V^\pi_{M'}(s_1,T) - Q_{\max}Pr(A_M)$$

Lemma 3.9. *(Lemma 56 in Li 2009) Let* $x_1, x_2, x_3, \dots \in \mathcal{B}$ *be a sequence of m independent Bernoulli trials, each with a success probability at least* μ: $E[x_i] \geq \mu$, *for some constant* $\mu > 0$. *Then for any* $l \in N$ *and* $\delta \in (0,1)$, *with probability at least* $1-\delta$, $x_1 + x_2 + \dots + x_m \geq l$ *if* $m \geq \frac{2}{\mu}\left(l + \ln\frac{1}{\delta}\right)$.

In the following, B denotes the (exact) Bellman operator.

Lemma 3.10. *Let* $\epsilon \geq 0$ *be a constant such that:* $\forall(s,a) \in (\mathcal{S},\mathcal{A}), BQ(s,a) \leq Q(s,a) + \epsilon$. *Then:*

$$\forall(s,a) \in (\mathcal{S},\mathcal{A}), Q^*(s,a) \leq Q(s,a) + \frac{\epsilon}{1-\gamma}$$

Proof. We will prove our claim by induction. All we need to prove is that $B^iQ_0(s,a) \leq Q_0(s,a) + \sum_{j=0}^{i-1}\gamma^j\epsilon$, and then take the limit as $i \to \infty$.

The base is given by hypothesis. Assuming that $B^iQ_0(s,a) \leq Q_0(s,a) + \sum_{j=0}^{i-1}\gamma^j\epsilon$, we'll prove that the inequality also holds for $i+1$:

$$\begin{aligned}
B^{i+1}Q_0(s,a) &= BB^iQ_0(s,a) \\
&\leq \sum_{s'} P(s'|s,a)\Big(\\
&\quad R(s,a,s') + \gamma\max_{a'} B^iQ_0(s',a')\Big) \\
&\leq \sum_{s'} P(s'|s,a)\Bigg(R(s,a,s') + \\
&\quad \gamma\left(\max_{a'} Q_0(s',a') + \sum_{j=0}^{i-1}\gamma^j\epsilon\right)\Bigg) \\
&= \sum_{s'} P(s'|s,a)\Big(R(s,a,s') + \\
&\quad \gamma\max_{a'} Q_0(s',a')\Big) + \gamma\sum_{j=0}^{i-1}\gamma^j\epsilon \\
&\leq Q_0(s,a) + \epsilon + \gamma\sum_{j=0}^{i-1}\gamma^j\epsilon \\
&= Q_0(s,a) + \sum_{j=0}^{i}\gamma^j\epsilon
\end{aligned}$$

If we now take the limit as $i \to \infty$ we have the original claim:

$$\begin{aligned}
\lim_{i\to\infty} B^iQ_0(s,a) &\leq Q_0(s,a) + \sum_{j=0}^{i-1}\gamma^j\epsilon \to Q^*(s,a) \\
&\leq Q_0(s,a) + \frac{\epsilon}{1-\gamma}
\end{aligned}$$

□

Lemma 3.11. *(Second part of theorem 4.1 in Williams and Baird (1993)). Let* $\epsilon = ||Q - BQ||_\infty$ *denote the Bellman error magnitude for Q, and* $V^* \leq V_Q$. *The return* V^π *from the greedy policy over Q satisfies:*

$$\forall s \in \mathcal{S},\ V^\pi(s) \geq V^*(s) - \frac{\epsilon}{1-\gamma}$$

Theorem 3.12. *Let* $\epsilon_- \geq 0$ *and* $\epsilon_+ \geq 0$ *be constants such that:* $\forall(s,a) \in (\mathcal{S},\mathcal{A}), -\epsilon_- \leq Q(s,a) - BQ(s,a) \leq \epsilon_+$. *The return* V^π *from the greedy policy over Q satisfies:*

$$\forall s \in \mathcal{S},\ V^\pi(s) \geq V^*(s) - \frac{\epsilon_- + \epsilon_+}{1-\gamma}$$

Proof. We set $Q'(s,a) = Q(s,a) + \frac{\epsilon_-}{1-\gamma}$, $\forall(s,a) \in (\mathcal{S},\mathcal{A})$. It's easy to see that the performance achieved by the one step greedy policy π over Q and π' over Q' is the same: $V^\pi(x) = V^{\pi'}(x)$. From lemma 3.10 we have $\forall(s,a) \in$

$(\mathcal{S}, \mathcal{A}), Q'(s,a) \geq Q^*(s,a)$.

$\forall (s,a) \in (\mathcal{S}, \mathcal{A})$,

$$\begin{aligned} BQ'(s,a) &= \sum_{s'} P(s'|s,a) \Bigg(R(s,a,s') + \\ &\quad \gamma \max_{a'} \left(Q(s',a') + \frac{\epsilon_-}{1-\gamma} \right) \Bigg) \\ &= BQ(s,a) + \gamma \frac{\epsilon_-}{1-\gamma} \\ \Rightarrow Q'(s,a) - BQ'(s,a) &= Q(s,a) - BQ(s,a) + \epsilon_- \\ &\leq \epsilon_- + \epsilon_+ \end{aligned}$$

From lemma 3.11 we have $\forall s \in \mathcal{S},\ V^\pi(s) \geq V^*(s) - \frac{\epsilon_- + \epsilon_+}{1-\gamma}$. □

Approximation

Given a Lipschitz continuous value function, the value of any state-action pair can be expressed in terms of any other state-action pair as $Q(s_j, a_j) = Q(s_i, a_i) + \xi_{ij} L_Q d_{ij}$, where $d_{ij} = d(s_i, a_i, s_j, a_j)$ and ξ_{ij} is a fixed but possibly unknown constant in $[-1, 1]$. For sample (s_i, a_i, r_i, s_i'), define:

$$x_{(s_i,a_i,r_i,s_i'),j} = r_i + \gamma V(s_i') + \xi_{ij} L_Q d_{ij}.$$

Then:

$$\begin{aligned} E_{s_i'}[x_{(s_i,a_i,r_i,s_i'),j}] &= E_{s_i'}[r_i + \gamma V(s_i')] + \xi_{ij} L_Q d_{ij} \\ &= Q(s_i, a_i) + \xi_{ij} L_Q d_{ij}. \end{aligned}$$

Consider a state-action pair (s_0, a_0) and its k approximate nearest neighbors (s_i, a_i) for $i = 1, \ldots, k$. We can arrive at an estimate of its value by averaging over the predicted value of all its neighbors (which may include itself if (s_0, a_0) is a sampled state-action pair): $\hat{Q}(s_0, a_0) = \frac{\sum_{i=1}^k x_{(s_i,a_i,r_i,s_i'),0}}{k}$. Setting $\xi_{ij} = 1\ \forall\ i, j$ we arrive at the definition of the approximate optimistic value function $\tilde{Q}$ (Definition 3.5).

In the following, B denotes the (exact) Bellman operator, $\hat{B}$ denotes the approximate Bellman operator corresponding to the definition of $\hat{Q}$ above, and $\tilde{B}$ denotes the approximate Bellman operator defined by the right hand side of equation 1.

The Bellman error can be decomposed into two pieces: the maximum absolute overestimation and underestimation error ϵ_s caused by using a finite number of neighbors, and the overestimation error caused by using neighbors at a nonzero distance from the point of interest ϵ_d (remember that ϵ_d is an input used at step 2 of the algorithm).

The following lemma helps bound the minimum number of neighbors k, required to guarantee a particular ϵ_s with probability $1 - \delta$:

Lemma 3.13.
If $\frac{Q^2_{\max}}{\epsilon_s^2} \ln \left(\frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta} \right) \leq k \leq \frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta}$*:*
$-\epsilon_s \leq \hat{B}\tilde{Q}(s,a) - B\tilde{Q}(s,a) \leq \epsilon_s$*, w. p.* $1 - \delta$, $\forall (s,a)$.

Proof. $\tilde{Q}$ is the (fixed) solution to the equations in Definition 3.5, and $\hat{B}$ differs from B in that it is the mean over k samples instead of the true expectation. Thus we can use Hoeffding's inequality to bound the difference between applying $\hat{B}$ and B to $\tilde{Q}$ for any (s,a) (note that values in $\tilde{Q}$ will always lie in $[0, \tilde{Q}_{\max}]$):[7]

$P(|\hat{B}\tilde{Q}(s,a) - B\tilde{Q}(s,a)| \leq \epsilon_s) \leq 2e^{-\frac{2\epsilon_s^2 k}{\tilde{Q}^2_{\max}}}$.

From the union bound, we have that the probability δ of the mean over k samples being more than ϵ_s away in any of the $k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)$ possible combinations, is no more than the sum of the individual probabilities:

$\delta \leq k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d) 2e^{-\frac{2\epsilon_s^2 k}{\tilde{Q}^2_{\max}}}$.

Solving for k, we have that for a given probability of failure δ and error ϵ_s, assuming $k \leq \frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta}$, k needs to be at most: $k = \frac{\tilde{Q}^2_{\max}}{\epsilon_s^2} \ln \left(\frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta} \right)$. □

The following lemma translates the result from Lemma 3.13 from $\hat{B}$ to $\tilde{B}$ for known states:

Lemma 3.14.
If $\frac{\tilde{Q}^2_{\max}}{\epsilon_s^2} \ln \left(\frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta} \right) \leq k \leq \frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}}, \epsilon_d)}{\delta}$*:*
$-\epsilon_s \leq \tilde{Q}(s,a) - B\tilde{Q}(s,a) \leq \epsilon_s + \epsilon_d$*, w. p.* $1 - \delta$*, for all known* (s,a)*.*

Proof. Follows directly from Lemma 3.13, the fact that $0 \leq \tilde{B}\tilde{Q}(s,a) - \hat{B}\tilde{Q}(s,a) \leq \epsilon_d$ for all known (s,a), and the fact that since $\tilde{Q}$ is fixed under $\tilde{B}$, $\tilde{B}\tilde{Q}(s,a) = \tilde{Q}(s,a),\ \forall (s,a)$. □

Lemma 3.15. *The set of equations in definition 3.5 has a unique fixed point.*

Proof. Assuming Q is in a complete metric space, all we need to prove is that $\tilde{B}$ is a contraction in maximum norm. Suppose $||Q_1 - Q_2||_\infty = \epsilon$. For any (s,a) we have:

$\tilde{B}Q_1(s,a) =$

$$\begin{aligned} &= \frac{1}{k} \sum_{j=1}^{k} \left(\min \left\{ \tilde{Q}_{\max}, r_j + \gamma \max_{a'} Q_1(s',a') + L_{\tilde{Q}} d_{ij} \right\} \right) \\ &\leq \frac{1}{k} \sum_{j=1}^{k} \left(\min \left\{ \tilde{Q}_{\max}, r_j + \gamma \max_{a'} (Q_2(s',a') + \epsilon) + L_{\tilde{Q}} d_{ij} \right\} \right) \\ &\leq \frac{1}{k} \sum_{j=1}^{k} \left(\min \left\{ \tilde{Q}_{\max}, r_j + \gamma \max_{a'} Q_2(s',a') + L_{\tilde{Q}} d_{ij} \right\} \right) + \gamma\epsilon \\ &= \tilde{B}Q_2(s,a) + \gamma\epsilon \\ &\Rightarrow \tilde{B}Q_1(s,a) \leq \tilde{B}Q_2(s,a) + \gamma\epsilon \end{aligned}$$

Similarly we have that $\tilde{B}Q_2(s,a) \leq \tilde{B}Q_1(s,a) + \gamma\epsilon$ which completes our proof. □

[7] This is not part of the algorithm. The lemma calculates what would be the difference between these two operators if we were to apply them to the fixed point solution to $\tilde{Q}$. Also note that the manner in which the Hoeffding bound is used (to bound ϵ_s) requires that the noise between different samples be independent, a fact which is guaranteed by the Markov property. The position of the samples is not required to be (and in practice will not be) independent. ϵ_d covers the worst case errors which may be introduced by biased placement of samples in the state-action space.

Efficient exploration

The following theorem is the main theorem of this section. It allows us to guarantee that the number of steps in which the performance of C-PACE is significantly worse than that of an optimal policy starting from the current state is at most log-linear in the covering number of the state-action space with probability $1-\delta$.

Theorem 3.16. *Let M be an MDP, $\tilde{\pi}_t$ be the greedy policy over $\tilde{Q}$ at time t, s_t be the state at time t and $\frac{\tilde{Q}^2_{\max}}{\epsilon_s^2}\ln\left(\frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}},\epsilon_d)}{\delta}\right) \leq k \leq \frac{2\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}},\epsilon_d)}{\delta}$. For a trajectory of arbitrary length, with probability at least $1-\delta$, there will be at most: $\frac{2Q_{\max}}{\epsilon_K}\ln\left(\frac{R_{\max}}{\epsilon_T}\right)\left(k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}},\epsilon_d)+\ln\frac{2}{\delta}\right)$ time-steps t, where $\epsilon_K=(1-\gamma)Q_{\max}Pr(A_M)$, such that:*[8]

$$\tilde{V}^{\tilde{\pi}_t}_M(s_t) < V^*_M(s) - \frac{2\epsilon_K+2\epsilon_T+\epsilon_d+2\epsilon_s}{1-\gamma}. \tag{2}$$

Proof. Let M' be an MDP that is equal to M on all known state-action pairs. All other state-action pairs transition deterministically with reward $R(s,a)=\tilde{Q}(s,a)$ to an absorbing state with reward 0.[9] Let A_M be the event that $\tilde{\pi}_t$ encounters an unknown state-action in T steps. At every time step, exactly one of two things can happen:

1. $Pr(A_M) \geq \frac{\epsilon_K}{Q_{\max}(1-\gamma)}$. From Lemma 3.9, with probability $1-\frac{\delta}{2}$, this can happen to no more than $\frac{2Q_{\max}(1-\gamma)}{\epsilon_K}\left(k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}},\epsilon_d)+\ln\frac{2}{\delta}\right)$ non-overlapping trajectories of length T before all state-actions become known. Setting $T=\frac{1}{1-\gamma}\ln\frac{R_{\max}}{\epsilon_T}$ we have that this can happen to no more than $\frac{2Q_{\max}}{\epsilon_K}\ln\left(\frac{R_{\max}}{\epsilon_T}\right)\left(k\mathcal{N}_{\mathcal{SA}}(L_{\tilde{Q}},\epsilon_d)+\ln\frac{2}{\delta}\right)$ time-steps.
2. $Pr(A_M) < \frac{\epsilon_K}{Q_{\max}(1-\gamma)}$ for state s at time t. With probability $1-\frac{\delta}{2}$:

$$\begin{aligned}
V^{\tilde{\pi}_t}_M(s_t) &\geq V^{\tilde{\pi}_t}_M(s_t,T)\\
&\geq V^{\tilde{\pi}_t}_{M'}(s_t,T)-\frac{\epsilon_K}{1-\gamma}\\
&\geq V^{\tilde{\pi}_t}_{M'}(s_t)-\frac{\epsilon_K+\epsilon_T}{1-\gamma}\\
&\geq V^*_{M'}(s)-\frac{\epsilon_K+\epsilon_T+\epsilon_d+2\epsilon_s}{1-\gamma}\\
&\geq V^*_{M'}(s,T)-\frac{\epsilon_K+\epsilon_T+\epsilon_d+2\epsilon_s}{1-\gamma}\\
&\geq V^*_{M}(s,T)-\frac{2\epsilon_K+\epsilon_T+\epsilon_d+2\epsilon_s}{1-\gamma}\\
&\geq V^*_{M}(s)-\frac{2\epsilon_K+2\epsilon_T+\epsilon_d+2\epsilon_s}{1-\gamma},
\end{aligned}$$

where step 1 made use of the fact that rewards are assumed to be non-negative, step 2 made use of Lemma 3.8, step 3 made use of Lemma 3.7, step 4 made use of Lemma 3.14, theorem 3.12 and the definition of M', step 5 made use of the fact that rewards are assumed to be non-negative, step 6 made use of Lemma 3.8 and step 7 made use of Lemma 3.7.

□

4 Related Work

PAC optimal exploration

During the last decade, there has been a significant body of work addressing PAC optimal exploration. However, the vast majority of papers only address the discrete case, providing incremental improvements on complexity bounds. The best of these bounds offer log-linear dependence on the size of the state-action space. Unfortunately, the huge constants associated with these bounds preclude the use of the associated algorithms in non-trivially sized, discrete MDPs, which coupled with their inability to handle continuous spaces makes them inapplicable in challenging, realistic domains.

Exploration in Metric State Spaces (Kakade, Kearns, and Langford 2003) is the only example we are aware of in the PAC MDP literature which tries to address exploration in continuous state spaces. While definitely a step in the right direction, the paper did not offer a concrete algorithm. Instead, an efficient black box approximate planner and local approximate modeling algorithm are assumed to exist. Unfortunately some of the assumptions regarding the abilities of these black box algorithms are overly optimistic (e.g. good model approximation everywhere not just with some probability $1-\delta$).

Nouri and Littman (2008) and Jong and Stone (2007) present interesting algorithms for exploration in continuous state spaces, but stop short of providing PAC-optimal bounds for them. For cases where the user can come up with a good set of features but not an exploration strategy, Strehl and Littman (2008) provide an algorithm that can explore in polynomial time in environments whose dynamics can be accurately modeled by linear regression. Similarly, Brunskill et al. (2009) also use parametric models, but allow for the state-action space to be partitioned into multiple models, rather than a monolithic one.

An interesting algorithm for discrete MDPs which has a number of similarities with C-PACE is Model-Based Interval Estimation (Strehl and Littman 2005). Similarly to C-PACE, Model-Based Interval Estimation integrates new samples as soon as they are available, instead of integrating in batches. In addition, like C-PACE it always acts assuming the best of all possible worlds consistent with its hypothesis is true, achieving exploration in an implicit manner, rather than explicitly choosing to explore or exploit at each timestep.

Delayed Q-learning (Strehl et al. 2006) was the first model-free algorithm for PAC optimal reinforcement learning with log-linear dependence on the number of state-actions. Its bounds have since been improved upon by a model based algorithm (Szita and Szepesvári 2010).

[8] Note that at the moment we assume that we don't underestimate the Lipschitz constant (cases where the Lipschitz constant is overestimated are accounted for). If we were to bound the Bellman error due to underestimations of the Lipschitz constant by ϵ_L, an extra term of $2\epsilon_L$ would appear in the numerator of equation 2.

[9] This may require the addition of an extra state. One could assume that this state exists in the original MDP but is unreachable.

Other forms of exploration

While PAC optimal exploration is arguably one of the most theoretically interesting forms of exploration, other forms of exploration have been proposed and used over the years.

One of the simplest and most commonly used approaches to exploration is the so called ϵ-greedy family of algorithms. Although they are guaranteed to explore the entire state-action space eventually, the time required may be exponential in the size of the state-action space.

Another approach to exploration is that of Bayesian or PAC-Bayesian exploration (Kolter and Ng 2009). Bayesian exploration tries to optimize for a very different goal from that of typical PAC optimal methods. Its assumption is that all that matters is the cumulative discounted reward from the current state, and as such it chooses to explore unknown state-actions only when those state-actions are expected to perform better that the known state-actions. This leads to a significantly more myopic algorithm, which explores far less than other PAC optimal methods. Such an approach would be appropriate in situations where its assumptions are true, which, for example, may include the testing phase in a learning scenario with both learning and testing phases.

5 Simulated HIV treatment

The simulated HIV treatment problem (Adams et al. 2004), introduced to reinforcement learning by Ernst et al. (2006), is a six dimensional continuous state space, two dimensional discrete action space problem, modeled after clinical data. The state space of the process, abbreviated as $(T_1, T_2, T_1^*, T_2^*, V, E)$ measures the concentrations of healthy $CD4+$ T-lymphocytes, healthy macrophages, infected $CD4+$ T-lymphocytes, infected macrophages, free virus particles and HIV-specific cytotoxic T-cells, updated every five days. The action space of the process is comprised of the administration of two drugs over the next five days, a reverse transcriptase inhibitor and a protease inhibitor.

The reward of the process at time t is: $1{,}000E_t - 0.1V_t - 20{,}000\epsilon_{1t}^2 - 2{,}000\epsilon_{2t}^2$. ϵ_{1t} and ϵ_{2t} are equal to 0.7 and 0.3 respectively when the reverse transcriptase inhibitor and protease inhibitor are administered and 0 otherwise. The agent is rewarded for having a large number of HIV-specific cytotoxic T-cells, while it is penalized for having a large number of free virus particles, and for using each of the drugs, as these drugs have severe, adverse side effects.

In the absence of treatment, the system of ordinary differential equations exhibits three physical equilibrium points (and several non physical ones for which one or more state variables are negative): The uninfected state $S = (10^6, 3198, 0, 0, 0, 10)$, which is an unstable equilibrium point (even slight perturbations lead away from this equilibrium), the "healthy" locally stable equilibrium point, $S = (967839, 621, 76, 6, 415, 353108)$ and the "non-healthy" locally stable equilibrium point, $S = (163573, 5, 11945, 46, 63919, 24)$. Simulations show that the basin of attraction of the healthy steady-state is relatively small compared to the one of the non-healthy steady state, and in the absence of drugs, perturbation of the uninfected steady state by adding as little as one single particle of virus per ml of blood plasma leads to asymptotic convergence towards the non-healthy steady state. For further details readers can consult earlier work on this domain (Ernst et al. 2006).

In all experiments below, the Lipschitz constant was set to 10^3, the distance function was set to be the max-norm over the state-action space, and $k = 1$ (the model for this domain is deterministic).[10] At the start of every episode, the domain was initialized to the "non-healthy" steady state.

A policy which selects actions uniformly at random, often used to obtain samples from typical RL domains was unable to generate samples anywhere but in the vicinity of the "non-healthy" steady state. Additionally several discretization based PAC-optimal exploration algorithms were unable to yield meaningful performance improvement after several days of computation, for various discretization thresholds. As such, we consider this domain a good testbed for an exploration algorithm.[11] Figure 1 (a) shows total accumulated discounted reward achieved by C-PACE as a function of training episodes. Although the performance is very poor at first, C-PACE is able to find a good policy in less than 500 episodes.

Figure 1 (b) shows the policy followed at the 1000th episode while figures 1 (c) through (h) show the path through the state space for the the 1000th episode. By repeatedly cycling the drugs, the agent is able to take the patient from the "non-healthy" steady state to the "healthy" steady state, where the number of HIV-specific cytotoxic T-cells is high, the number of free virus particles is low, and drugs are no longer required to manage the disease.

Since our choice of reward function and dynamics are the same as the ones used in earlier work, our results are directly comparable. Of course one would be comparing the combination of human effort to find a good exploration schedule and a "best estimate" algorithm (Ernst et al. 2006), or a pure planning algorithm using the model (Busoniu et al. 2011), versus the fully automated exploration of C-PACE, which needs to convince itself that no better policy exists before it starts to follow the same trajectory to the goal consistently. To the best of our knowledge, this is the first time this problem (or any other problem of comparable difficulty) has been tackled by a PAC optimal exploration algorithm.

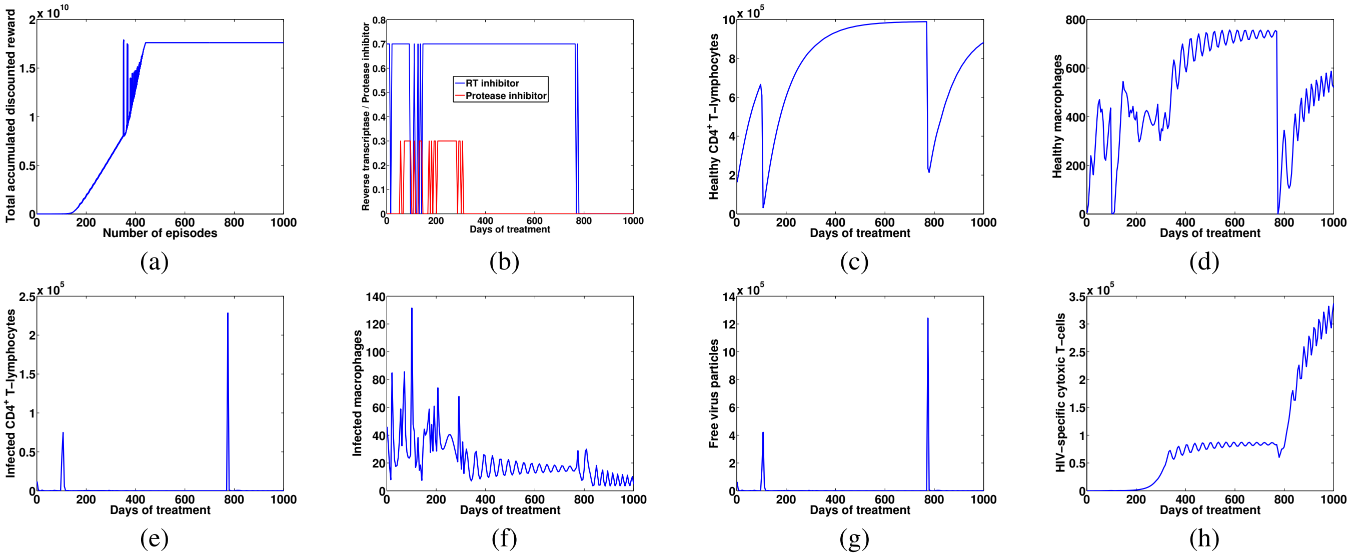
6 Future work

While these simulation results are very encouraging, there are still aspects of the real world as they would apply to the HIV treatment and many other domains, that we have not yet considered.

For the HIV treatment problem the number of exploration episodes needed to arrive at a good treatment strategy are

[10] The max-norm over the state-action space was chosen because of its simplicity, and the Lipschitz constant via cross-validation. While our choice of distance function and Lipschitz constant is likely suboptimal, it proved to be sufficient, with performance being robust over several orders of magnitude of the Lipschitz constant.

[11] We also tested C-PACE in more traditional domains such as the noisy pendulum swing-up with good results. The HIV treatment results are presented here as this domain is much more challenging.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: (a) Total accumulated discounted reward per episode achieved by C-PACE versus training episodes. (b) Policy for the the 1000th episode: use of the reverse transcriptase inhibitor (blue) and protease inhibitor (red). Path through the state space for the the 1000th episode: (c) number of healthy $CD4+$ T-lymphocytes, (d) healthy macrophages, (e) infected $CD4+$ T-lymphocytes, (f) infected macrophages, (g) free virus particles, (h) HIV-specific cytotoxic T-cells.

fairly modest, even by real world standards. However, in our simulations, the trials were done in succession. Considering that for the HIV domain each episode lasts two years, arriving at a good policy in this way would require several centuries. An important step forward would be to develop algorithms that can exploit multiple parallel executions on the same MDP, integrating information and choosing actions to maximize global knowledge and performance.

Another aspect of the real world which our exploration algorithm did not take into account is that the true cost of exploration is more than just the number of steps for which we perform worse than near optimally. This is obviously true when dealing with patients, but it is also true with mechanical systems. Even if, for example our budget allows us to crash a number of autonomous helicopters while learning to fly, that number will be prohibitively small compared to most PAC bounds. Consequently methods for safe exploration are of great real world interest (Moldovan and Abbeel 2012).

Our analysis applies to both discrete and continuous/multidimensional action spaces. Unfortunately, selecting the action which maximizes $\tilde{Q}$ over large/infinite sets of actions is far from trivial, and approximate methods could break any PAC guarantees. An effective method for efficient action selection which maintains PAC guarantees would extend the applicability of C-PACE to many interesting problems.

Finally, the central assumption made by C-PACE is that there exists some distance function in which the value function is smooth (Lipschitz continuous in our analysis). While it is reasonable to expect that such a distance function exists, many obvious distance functions that a user might try to use may not satisfy this requirement, or may have a large Lipschitz constant. Automatic discovery of suitable distance functions is an important next step.

7 Conclusion

In this paper, we presented C-PACE, the first *practical*, model-free, PAC-optimal algorithm for exploration in continuous spaces using real trajectories for learning. C-PACE requires only a guess of the Lipschitz constant of the Q-function for the chosen distance metric and the maximum tolerable error due to sampling as input. Using these and an approximation to the Bellman operator, it provides a straightforward and easily implemented exploration algorithm with strong performance guarantees. We have demonstrated the ability of C-PACE to explore and learn good policies for a challenging, six-dimensional problem, the first time a PAC-optimal algorithm has been applied to a problem of such size and difficulty.

Acknowledgments

We would like to thank Vincent Conitzer, Mauro Maggioni and the anonymous reviewers for helpful comments and suggestions. This work was supported by NSF IIS-1147641 and NSF IIS-1218931. Opinions, findings, conclusions or recommendations herein are those of the authors and not necessarily those of NSF.

References

Adams, B.; Banks, H.; Kwon, H.-D.; and Tran, H. 2004. Dynamic multidrug therapies for HIV: Optimal and STI control approaches. In *Mathematical Biosciences and Engineering*, volume 1, 223 – 241.

Brunskill, E.; Leffler, B.; Li, L.; Littman, M.; and Roy, N. 2009. Provably efficient learning with typed parametric models. *The Journal of Machine Learning Research* 10:1955–1988.

Busoniu, L.; Munos, R.; Schutter, B. D.; and Babuska, R. 2011. Optimistic planning for sparsely stochastic systems. In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning*.

Ernst, D.; Stan, G.-B.; Goncalves, J.; and Wehenkel, L. 2006. Clinical data based optimal STI strategies for HIV; a reinforcement learning approach. In *Machine Learning Conference of Belgium and The Netherlands (Benelearn)*, 65–72.

Jong, N., and Stone, P. 2007. Model-based exploration in continuous state spaces. *Abstraction, Reformulation, and Approximation* 258–272.

Kakade, S.; Kearns, M. J.; and Langford, J. 2003. Exploration in metric state spaces. In *ICML*, 306–312.

Kakade, S. M. 2003. *On the sample complexity of reinforcement learning*. Ph.D. Dissertation, Gatsby Computational Neuroscience Unit, University College London.

Kearns, M. J., and Singh, S. P. 2002. Near-optimal reinforcement learning in polynomial time. *Machine Learning* 49(2-3):209–232.

Kolter, J. Z., and Ng, A. Y. 2009. Near-bayesian exploration in polynomial time. In *ICML '09*, 513–520.

Li, L. 2009. *A unifying framework for computational reinforcement learning theory*. Ph.D. Dissertation, Rutgers University, New Brunswick, NJ, USA. AAI3386797.

Moldovan, T. M., and Abbeel, P. 2012. Safe exploration in Markov decision processes. *CoRR* abs/1205.4810.

Nouri, A., and Littman, M. 2008. Multi-resolution exploration in continuous spaces. *Advances in neural information processing systems* 21:1209–1216.

Strehl, A. L., and Littman, M. L. 2005. A theoretical analysis of model-based interval estimation. In *ICML '05*, 856–863. New York, NY, USA: ACM.

Strehl, A., and Littman, M. 2008. Online linear regression and its application to model-based reinforcement learning. *Advances in Neural Information Processing Systems* 20:1417–1424.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. PAC model-free reinforcement learning. In *ICML*, 881–888.

Szita, I., and Szepesvári, C. 2010. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *ICML*, 1031–1038.

Williams, R., and Baird, L. 1993. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Northeastern University, College of Computer Science.