Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Sample Complexity and Performance Bounds for Non-Parametric Approximate Linear Programming

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

Abstract

One of the most difficult tasks in value function approximation for Markov Decision Processes is finding an approximation architecture that is expressive enough to capture the important structure in the value function, while at the same time not overfitting the training samples. Recent results in non-parametric approximate linear programming (NP-ALP), have demonstrated that this can be done effectively using nothing more than a smoothness assumption on the value function. In this paper we extend these results to the case where samples come from real world transitions instead of the full Bellman equation, adding robustness to noise. In addition, we provide the first max-norm, finite sample performance guarantees for any form of ALP. NP-ALP is amenable to problems with large (multidimensional) or even infinite (continuous) action spaces, and does not require a model to select actions using the resulting approximate solution.

1 Introduction and motivation

Linear programming is one of the standard ways to find the optimal value function of a Markov Decision Process. While its approximate, feature based version, Approximate Linear Programming (ALP), has been known for quite a while, until recently it had not received as much attention as approximate value and policy iteration methods. This can be attributed to a number of apparent drawbacks, namely poor resulting policy performance when compared to other methods, poor scaling properties, dependence on noise-free samples, no straightforward way to go from the resulting value function to a policy without a model and only $l1$-norm bounds. A recent surge of papers has tried to address some of these problems. One common theme among most of these papers is the assumption that the value function exhibits some type of smoothness.

Instead of using smoothness as an indirect way to justify the soundness of the algorithms, this paper takes a very different approach, extending the non-parametric approach to ALP (NP-ALP) (Pazis and Parr 2011b), which relies on a smoothness assumption on the value function (not necessarily in the ambient space). NP-ALP offers a number of important advantages over its feature based counterparts. The most obvious advantage is that because the approach is non-parametric, there is no need to define features or perform costly feature selection. Additionally, NP-ALP is amenable to problems with large (multidimensional) or even infinite (continuous) state and action spaces, and does not require a model to select actions using the resulting approximate solution.

This paper makes three contributions to the applicability and understanding of NP-ALP: 1) We extend NP-ALP to the case where samples come from real world interaction rather than the full Bellman equation. 2) We prove that NP-ALP offers significantly stronger and easier to compute performance guarantees than feature based ALP, the first (to the best of our knowledge) max-norm performance guarantees for any ALP algorithm. 3) We lower bound the rate of convergence and upper bound performance loss using a finite number of samples, even in the case where noisy, real world samples are used instead of the full Bellman equation.

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, while $P(s'|s,a)$ denotes the corresponding transition probabilities in discrete environments$\big)$, 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. The goal of the decision maker is to find 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 ob-

serves 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.

One way to solve for the optimal value function V^* in small, discrete MDPs when the model is available, is via linear programming, where every state $s \in \mathcal{S}$ is a variable and the objective is to minimize the sum of the states' values under the constraints that the value of each state must be greater than or equal to all Q-values for that state:

$$minimize \sum_s V^*(s),\ subject\ to:$$
$$(\forall s, a) V^*(s) \geq \sum_{s'} P(s'|s,a)\left(R(s,a,s') + \gamma V^*(s')\right)$$

Extracting the policy is fairly easy (at least conceptually), just by picking the action with a corresponding non-zero dual variable for the state in question (equivalently, picking the action that corresponds to the constraint that has no slack in the current state). Note that we can have a set of state-relevance weights $\rho(s)$ associated with every state in the optimization criterion; however, for the exact case every set of positive weights leads to the same V^*.

3 Non-parametric ALP

Definitions and assumptions

In the following, $\tilde{S}$ is a set of sampled state-action pairs drawn from a bounded region/volume, $\tilde{V}$ denotes the solution to the NP-ALP, $\tilde{Q}$ denotes the Q value function implied by the constraints of the NP-ALP and L_f denotes the Lipschitz constant of function f.

The main assumption required by NP-ALP is that there exists some distance function d on the state-action space of the process, for which the value function is Lipschitz continuous.[1] A Lipschitz continuous action-value function satisfies the following constraint for all (s, a) and (s', a') pairs:

$$\exists\ L_Q : |Q(s,a) - Q(s',a')| \leq L_Q d(s,a,s',a')$$

where: $d(s,a,s',a') = ||k(s,a) - k(s',a')||$ and $k(s,a)$ is a mapping from state-action space to a normed vector space.

For simplicity, we assume that the distance function between two states is minimized when the action is the same: $\forall\ (a, a'),\ d(s,s') = d(s,a,s',a) \leq d(s,a,s',a')$. Thus for a Lipschitz continuous value function: $|V(s) - V(s')| \leq L_V d(s,s')$ and it is easy to see that $L_V \leq L_Q$.

The notation $\mathcal{M}_L$ denotes the set of functions with Lipschitz constant L. For any $\tilde{L}$, $\tilde{V} \in \mathcal{M}_{\tilde{L}}$ can be enforced via linear constraints, which we'll call smoothness constraints. $\left(\forall s, s' : d(s,s') < \frac{Q_{\max} - Q_{\min}}{L_{\tilde{Q}}}\right)$:

$$\tilde{V}(s) \quad \geq \quad \tilde{V}(s') - L_{\tilde{Q}} d(s,s'), \tag{1}$$

where $Q_{\max} = \frac{R_{\max}}{1-\gamma}$ and $Q_{\min} = \frac{R_{\min}}{1-\gamma}$.

A Bellman constraint on state-action $(s_i, a_i) \in \tilde{S}$, $Bel(s_i, a_i)$ is defined as: $Bel(s_i, a_i) \rightarrow \tilde{V}(s_i) \geq \frac{1}{k}\sum_{j=1}^{k}\left(R(s_j, a_j, s'_j) + \gamma\tilde{V}(s'_j) - L_{\tilde{Q}} d(s_i, a_i, s_j, a_j)\right)$ where $j = 1$ through k are the k nearest neighbors of sample i in $\tilde{S}$ (including itself).

The algorithm

Given the above, NP-ALP can be summarized as follows:

1. Solve the following linear program:

$$minimize \sum_{s \in \tilde{S}} \tilde{V}(s),\ subject\ to:$$
$$(\forall (s_i, a_i) \in \tilde{S})\ Bel(s_i, a_i)$$
$$\tilde{V} \in \mathcal{M}_{L_{\tilde{V}}}$$

   where $\tilde{V} \in \mathcal{M}_{L_{\tilde{V}}}$ is implemented as in equation 1.

2. Let $\hat{\mathcal{S}}$ be the set of state-actions for which the Bellman constraint is active in the solution of the LP above. Discard everything except for the values of variables in $\hat{\mathcal{S}}$ and their corresponding actions.
3. Given a state s' select and perform an action a according to $\arg\max_{(s,a)\in\hat{\mathcal{S}}}\{\tilde{V}(s) - L_{\tilde{Q}} d(s,s')\}$.
4. Go to step 3.

Key properties

Sparsity Notice that the smoothness constraint on $\tilde{V}$ is defined over the entire state space, not just the states in $\tilde{S}$. However, it suffices to implement smoothness constraints only for states in $\tilde{S}$ or reachable in one step from a state in $\tilde{S}$, as smoothness constraints on other states will not influence the solution of the LP.

We will call all (primal) variables corresponding to state-action pairs in $\hat{\mathcal{S}}$ *basic* and the rest *non-basic*. Non-basic variables (and their corresponding constraints) can be discarded without changing the solution. This is useful both for sparsifying the solution to make evaluation significantly faster (as is done in step 2 of the algorithm), and can be used to solve the linear program efficiently either by constraint generation, or by constructing a homotopy method.

Consider a state-action pair s, a corresponding to a non-basic variable. This implies $\tilde{V}(s) = \tilde{V}(s') - L_{\tilde{V}} d(s,s')$ for some state s'.[2] When presented with state s'' to evaluate, we have:

$$\tilde{V}(s'') \quad \geq \quad \tilde{V}(s) - L_{\tilde{V}} d(s'', s) \tag{2}$$
$$\tilde{V}(s'') \quad \geq \quad \tilde{V}(s') - L_{\tilde{V}} d(s'', s') \tag{3}$$

[1] Note that NP-ALP can easily be extended to other forms of continuity by pushing the complexity inside the distance function. For example if $d \in [0,1]$ by defining $d' = d^\alpha$ where $\alpha \in (0,1]$ we can allow for Hölder continuous value functions.

[2] In the case where $s = s'$ this means that some other action dominates action a for state s.

Substituting $\tilde{V}(s) = \tilde{V}(s') - L_{\tilde{V}} d(s, s')$ into 2:

$$\tilde{V}(s'') \quad \geq \quad \tilde{V}(s') - L_{\tilde{V}}(d(s'', s) + d(s, s')) \qquad (4)$$

Since $d(s'', s') \leq d(s'', s) + d(s, s')$ constraint 2 does not influence the value of $\tilde{V}(s'')$.

Finally, adding states to the objective function that are not in $\tilde{S}$ or weighting the states in $\tilde{S}$ would not alter the LP solution; thus it suffices to set the objective function to be the sum over only the states in $\tilde{S}$.

The NP-ALP solution can be stored and used efficiently All we need to retain in step 2 of the algorithm are the values of variables in $\hat{S}$ and their corresponding actions. The number of such variables is at most equal to the number of samples, or significantly less in most realistic situations.

NP-ALP allows model-free continuous action selection For some query state s, the Bellman constraint that bounds the value of this state also bounds the maximal Q-value for this state. This means that actions in $\tilde{S}$ can come from a continuous range and that the maximizing action for any state can be found efficiently (as is done in step 3 of the algorithm), but it does limit actions selected at execution time to actions available for some nearby state in $\tilde{S}$.[3]

After non-basic variables have been discarded, there is only one surviving (both primal and dual) variable per basic state. For any basic state s, $\tilde{V}(s)$ is bounded by a Bellman constraint from state-action pair s, a, so $\tilde{V}(s) = \tilde{Q}(s, a)$. If s bounds the value of a non-basic state t by $\tilde{V}(t) \geq \tilde{V}(s) - L_{\tilde{V}} d(s, t)$, it also bounds $\tilde{Q}(t, a)$.[4] The predicted optimal action at t will therefore be the same as in s since the bounds from other states are lower, implying lower estimated Q-values.

The above has two important consequences. First, only actions present in the training set can ever be selected during policy execution, since the value estimation and action selection mechanisms are pessimistic. Second, action selection complexity is independent of the number of actions, allowing us to deal with spaces with infinite (continuous) or massive (multidimensional) action spaces. Sampling is of course important; however, this goes beyond the scope of this paper. See Pazis and Parr (2013) for more details.

The NP-ALP is always well defined The Lipschitz continuity constraints ensure that the solution is always bounded, even when large parts of the state-action space have been poorly sampled. This is in contrast to parametric ALP, where a single missing constraint can, in the worst case, cause the LP to be unbounded.

[3]For simplicity we assume that all actions are available in all states. When this is not the case we'd have to take the distance of the sampled actions to the closest available action at the query state into account.

[4]For simplicity of exposition, we assume that $L_{Q_a} = L_V \; \forall a \in A$. The case where different actions have different Lipschitz constants extends naturally.

Practical considerations

Some readers will have noticed that in a naive implementation, the number of constraints scales quadratically with the number of samples in the worst case (when $\frac{Q_{\max} - Q_{\min}}{L_{\tilde{Q}}}$ spans the entire space). Fortunately the NP-ALP constraints have a number of favorable properties. All the Lipschitz constraints involve exactly two variables, resulting in a very sparse constraint matrix, a property that modern solvers can exploit. Additionally, for distance functions such as the $l1$ or max-norm, most (depending on the dimensionality of the space) Lipschitz constraints can be pruned.

Even in the case of an "unfriendly" norm, we can use an iterative approach, progressively adding samples whose Bellman constraint is violated. Taking advantage of the fact that solutions tend to be very sparse, and that samples whose Bellman constraints are not tight will not influence the solution, very large problems can be solved without ever adding more than a tiny fraction of the total number of constraints. In our experiments, this technique proved to be far more effective than naive constraint generation.

Finally, for every sample either its Bellman constraint or exactly one of its Lipschitz constraints will be active, which means we can construct a homotopy method.[5] Starting from $L_{\tilde{V}} = 0$ only one Bellman constraint will be active and all other states will be bound by Lipschitz constraints to $\tilde{V} = \frac{R_{\max}}{1-\gamma}$. Progressively relaxing $L_{\tilde{V}}$, the entire space of solutions can be traversed.

4 Error bounds

In this section we bound the difference between the performance of the policy executed by NP-ALP and an optimal policy. Readers should remember that all operations applied on $\tilde{V}$ and $\tilde{Q}$ in this section are applied to the *fixed* solution of the NP-ALP so as to uncover its properties, and are not part of the algorithm.

In the following B is used to signify the (exact) Bellman operator.

Theorem 4.1. *(Theorem 3.12 in Pazis and Parr (2013)) 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:*

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

Given a Lipchitz 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:

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

[5]We have not yet implemented such a method.

Consider the (exact) Bellman operator B as it would apply to $\tilde{Q}$ for some (s_j, a_j):

$B\tilde{Q}(s_j, a_j) = \int_{s'_j} p(s'_j|s_j, a_j) \Big(R(s_j, a_j, s'_j) + \gamma \tilde{Q}(s'_j)\Big)$.

For $(s, a) \in \hat{\mathcal{S}}$, one could approximate B as $\hat{B}$ using a finite sum over k values of x: $\hat{B}\tilde{Q}(s_j, a_j) = \sum_{i=1}^{k} x_{ij}$, where $i = 1$ through k are the k nearest neighbors of sample j in $\tilde{S}$ (including itself). Let us also define $\tilde{B}$ similarly to $\hat{B}$ but by setting $\xi_{ij} = -1 \; \forall \; i, j$. It should be clear from the definition that $\forall \; (s, a) \in \hat{\mathcal{S}}, \; \tilde{B}\tilde{Q}(s, a) = \tilde{Q}(s, a)$.

$\tilde{Q}$ is the (fixed) solution to the NP-ALP and $\hat{B}$ differs from B for $(s, a) \in \hat{\mathcal{S}}$ 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 $(s, a) \in \hat{\mathcal{S}}$:[6]

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

From the union bound, we have that the probability δ of the mean over k samples being more than t away in any of the n samples, is no more than the sum of the individual probabilities: $\delta \leq n2e^{-\frac{2t^2 k}{(Q_{\max} - Q_{\min})^2}}$.

Taking logarithms on both sides and solving for t, we have that for a given probability of failure δ, the absolute error is upper bounded by: $t \leq \frac{(Q_{\max} - Q_{\min})}{\sqrt{2}} \sqrt{\frac{\ln \frac{2n}{\delta}}{k}}$.

Lemma 4.2. *Let ϵ_s^- and ϵ_s^+ denote the maximum underestimation and overestimation Bellman error respectively, such that $\forall (s, a) \in \hat{\mathcal{S}}$:*

$$-\epsilon_s^- \leq \tilde{B}\tilde{Q}(s, a) - B\tilde{Q}(s, a) \quad \leq \quad \epsilon_s^+$$

then with probability $1 - \delta$:

$$\epsilon_s^- \quad \leq \quad \frac{(Q_{\max} - Q_{\min})}{\sqrt{2}} \sqrt{\frac{\ln \frac{2n}{\delta}}{k}} + 2L_{\tilde{Q}} d_{k,\max}$$

$$\epsilon_s^+ \quad \leq \quad \frac{(Q_{\max} - Q_{\min})}{\sqrt{2}} \sqrt{\frac{\ln \frac{2n}{\delta}}{k}}$$

where $d_{k,\max}$ is the maximum distance of a sample from its k-1 sampled neighbor.

Proof. Follows directly from the discussion above, and the fact that $\forall \; (s, a) \in \hat{\mathcal{S}}, \; \hat{B}\tilde{Q}(s, a) \geq \tilde{B}\tilde{Q}(s, a)$ and $\hat{B}\tilde{Q}(s, a) \leq \tilde{B}\tilde{Q}(s, a) + 2L_{\tilde{Q}} d_{k,\max}$. □

Lemma 4.3. *Let $-\epsilon_d - \epsilon_s^- \leq \tilde{Q}(s, a) - B\tilde{Q}(s, a), \forall (s, a) \in (\mathcal{S}, \mathcal{A})$. Then $\epsilon_d \leq d_{\max}(L_{B\tilde{Q}} + L_{\tilde{Q}})$, where $d_{\max}$ is the maximum distance from a non-sampled state-action pair to the closest sampled state-action pair.*

Proof. From Lemma 4.2, $\forall (s, a) \in \hat{\mathcal{S}}, \; -\epsilon_s^- \leq \tilde{Q}(s, a) - B\tilde{Q}(s, a)$. Similarly, for all state-actions for which we have a Bellman constraint present but inactive, it must be $-\epsilon_s^- \leq \tilde{Q}(s, a) - B\tilde{Q}(s, a)$, otherwise the Bellman constraint would be active. Let there be some state-action (s, a) for which the Bellman constraint is missing, and let (s', a') be its nearest neighbor for which a Bellman constraint is present. Then:

$$\begin{aligned} B\tilde{Q}(s, a) \quad &\leq \quad B\tilde{Q}(s', a') + L_{B\tilde{Q}} d(s, a, s', a') \\ &\leq \quad \tilde{Q}(s', a') + \epsilon_s^- + L_{B\tilde{Q}} d(s, a, s', a') \\ &\leq \quad \tilde{Q}(s, a) + L_{\tilde{Q}} d(s, a, s', a') + \epsilon_s^- + \\ &\quad \; L_{B\tilde{Q}} d(s, a, s', a') \\ &\leq \quad \tilde{Q}(s, a) + \epsilon_s^- + d_{\max}(L_{B\tilde{Q}} + L_{\tilde{Q}}) \end{aligned}$$

$$\Rightarrow -\epsilon_d - \epsilon_s^- \leq \tilde{Q}(s, a) - B\tilde{Q}(s, a)$$

□

Lemma 4.4. *Let $\tilde{Q}(s, a) - B\tilde{Q}(s, a) \leq \epsilon_C + \epsilon_s^+, \forall (s, a) \in (\mathcal{S}, \mathcal{A})$. Then for $L_{\tilde{Q}} > 0$:*

$$\epsilon_C \leq \max\left(0, (Q_{\max} - Q_{\min})\left(\frac{L_{B\tilde{Q}}}{L_{\tilde{Q}}} - 1\right)\right) \tag{5}$$

Proof. From Lemma 4.2, $\forall (s, a) \in \hat{\mathcal{S}}, \; \tilde{Q}(s, a) - B\tilde{Q}(s, a) \leq \epsilon_s^+$. Let there be some state-action (s, a) that is constrained by a Lipschitz continuity constraint from another state-action (s', a'), such that its value is $\tilde{Q}(s, a) = \tilde{Q}(s', a') - L_{\tilde{Q}} d(s, a, s', a')$. Then we have that $d(s, a, s', a') \leq \frac{Q_{\max} - Q_{\min}}{L_{\tilde{Q}}}$ (otherwise we would have $Q(s, a) < Q_{\min}$) and $(s', a') \in \hat{\mathcal{S}}$. Consequently:

$$\begin{aligned} B\tilde{Q}(s, a) \quad &\geq \quad B\tilde{Q}(s', a') - L_{B\tilde{Q}} d(s, a, s', a') \\ &\geq \quad \tilde{Q}(s', a') - \epsilon_s^+ - L_{B\tilde{Q}} d(s, a, s', a') \\ &= \quad \tilde{Q}(s, a) + L_{\tilde{Q}} d(s, a, s', a') - \epsilon_s^+ - \\ &\quad \; L_{B\tilde{Q}} d(s, a, s', a') \end{aligned}$$

$$\begin{aligned} \Rightarrow \tilde{Q}(s, a) - B\tilde{Q}(s, a) \quad &\leq \quad \epsilon_s^+ + \\ &\quad \; d(s, a, s', a')(L_{B\tilde{Q}} - L_{\tilde{Q}}). \end{aligned}$$

For $L_{B\tilde{Q}} \geq L_{\tilde{Q}}$ the above is maximized for $d(s, a, s', a') = \frac{Q_{\max} - Q_{\min}}{L_{\tilde{Q}}}$, yielding $\epsilon_C = (Q_{\max} - Q_{\min})\left(\frac{L_{B\tilde{Q}}}{L_{\tilde{Q}}} - 1\right)$, and otherwise for $d(s, a, s', a') = 0$, yielding $\epsilon_C = 0$. □

We are now ready to state the main theorem of this paper:

Theorem 4.5. *Let $\tilde{V}$ be the solution to the NP-ALP. The return $\tilde{V}^\pi$ from the greedy policy over $\tilde{V}$ satisfies:*

$$\forall s \in \mathcal{S}, \; \tilde{V}^\pi(s) \geq V^*(s) - \frac{\epsilon_C + \epsilon_d + \epsilon_s^- + \epsilon_s^+}{1 - \gamma}$$

Proof. From lemmata 4.3 and 4.4 we have that $\forall (s, a) \in (\mathcal{S}, \mathcal{A})$: $-\epsilon_d - \epsilon_s^- \leq \tilde{Q}(s, a) - B\tilde{Q}(s, a) \leq \epsilon_C + \epsilon_s^+$, and the result follows directly from theorem 4.1. □

[6] Note that the values returned by the LP will always lie in $[Q_{\min}, Q_{\max}]$ (see section 3 for the definition of $Q_{\max}$ and $Q_{\min}$).

It is worth pointing out how the above bound differs from bounds typically seen in ALP literature. Theorem 4.5 bounds the performance of the greedy policy over the solution returned by NP-ALP to the performance of the optimal policy, rather than the distance between the approximate value function and V^*. In addition, theorem 4.5 is expressed in terms of max-norm, rather than $l1$-norm. Finally as we'll see later, theorem 4.5 is expressed in terms of quantities that can be bounded more easily than the ones of typical ALP bounds.

Lemma 4.8 below allows us to bound the Lipschitz constant of $B\tilde{V}$, in terms of the Lipschitz constant of the reward and transition functions, while lemma 4.9 bounds how large $L_{\tilde{V}}$ needs to be in order to guarantee $\epsilon_C = 0$. Note that while a Lipschitz continuous reward and transition function implies a Lipschitz continuous $B\tilde{V}$, it is not a requirement. One could easily come up with discontinuous reward and transition functions that still result in continuous value functions.

Definition 4.6. *If the reward function is L_r-Lipschitz continuous, it satisfies the following constraint for every two states s_1 and s_2:*

$$|r(s_1, a) - r(s_2, a)| \leq L_r d(s_1, s_2)$$

Definition 4.7. *If the transition model is L_p-Lipschitz continuous it satisfies the following constraint for every two states s_1 and s_2, and all V with $L_V = 1$:*

$$\Big| \int_{s'} (p(s'|s_1, a) - p(s'|s_2, a))V(s')ds' \Big| \leq L_p d(s_1, s_2)$$

Observe that this bounds the difference in expected next state values with respect to a normalized V. If $L_V \neq 1$, the worst case difference can be scaled appropriately.

Lemma 4.8.

$$L_{B\tilde{Q}} \leq L_r + \gamma L_p L_{\tilde{V}} \tag{6}$$

Proof. Follows directly from the definitions of L_r, L_p, $L_{\tilde{V}}$ and $L_{\tilde{Q}}$. □

Lemma 4.9. *If $\gamma L_p < 1$ and $L_{\tilde{Q}} \geq \frac{L_r}{1-\gamma L_p}$, $\epsilon_C = 0$.*

Proof. The result follows directly by substituting equation 6 in equation 5 and requiring $\frac{L_r + \gamma L_p L_{\tilde{V}}}{L_{\tilde{Q}}} \leq 1$. □

Note that $\gamma L_p < 1$ is satisfied is many noise models, e.g., actions that add a constant impulse with Gaussian noise.

In this work we are mostly interested in problems where the ambient space may be large, but the underlying manifold where the data lie is of low enough dimension as to be able to be covered with a reasonable number of samples. Cases where samples come from a truly high dimensional space are inherently difficult and hard to tackle without additional strong assumptions. The following corollary gives an idea of how the sample complexity of NP-ALP scales with the dimension of the data.

Corollary 4.10. *Assuming that n Bellman constraints are spread uniformly across the state-action space*[7] *and setting $k = \left(\ln \frac{2n}{\delta}\right)^{\frac{D}{2+D}} n^{\frac{2}{2+D}}$ the bound from theorem 4.5 becomes:*

$$\begin{aligned} &\forall s \in \mathcal{S}, \\ &\tilde{V}^{\pi}(s) \quad \geq \quad V^*(s) - \frac{\epsilon_c + C_s^{-1}(L_{B\tilde{Q}} + L_{\tilde{Q}}) \left(\frac{1}{n}\right)^{\frac{1}{D}}}{1-\gamma} \\ &\qquad - \quad \frac{\left(2\frac{(Q_{\max} - Q_{\min})}{\sqrt{2}} + 2C_s^{-1} L_{\tilde{Q}}\right) \left(\frac{\ln \frac{2n}{\delta}}{n}\right)^{\frac{1}{2+D}}}{1-\gamma} \end{aligned}$$

w. p. $1-\delta$, and $\tilde{V}^{\pi}(s) \to V^*(s) - \frac{\epsilon_c}{1-\gamma}$ as $n \to \infty$.

Proof. The volume contained by the minimum hypersphere containing k points is proportional to $\frac{k}{n}$. The radius of that hypersphere r_k is related to that volume as: $\frac{k}{n} = C_s r_k^D$, where D is the dimensionality of the space (ambient or underlying manifold). Thus in this case $d_{\max} = C_s^{-1} \left(\frac{1}{n}\right)^{\frac{1}{D}}$, $d_{k,\max} = C_s^{-1} \left(\frac{k}{n}\right)^{\frac{1}{D}}$ and the result follows by substitution. □

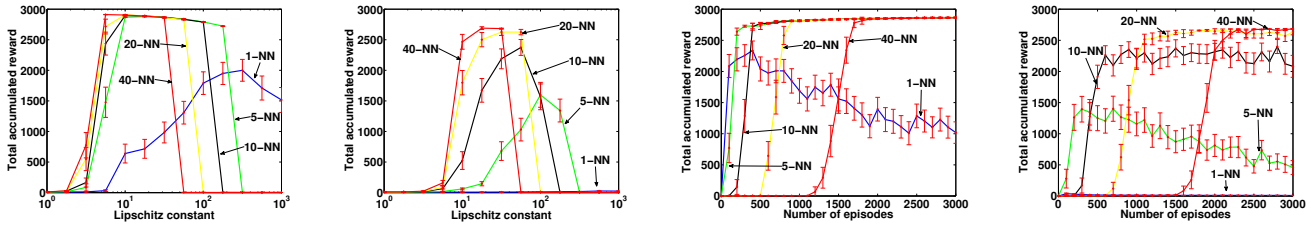
5 Related Work

The most closely related prior work to ours is that of (Pazis and Parr 2011b) where the NP-ALP algorithm was introduced. It's shortcomings addressed in this paper were that it required samples from the full Bellman equation, lacked sample complexity results, and it's performance guarantees were expressed as an $l1$-norm distance between value functions, rather than max-norm performance guarantees.

In the realm of parametric value function approximation, regularized approximate linear programming (RALP) (Petrik et al. 2010), is close to the original NP-ALP paper and, by transitivity, somewhat close to this paper. The bounds in the original NP-ALP paper were similar in form and derivation to the RALP bounds and were similarly loose. Since RALP is a parametric method, it should be viewed primarily as down-selecting features via $l1$ regularization, while non-parametric ALP methods do not require an initial set of features at all. In addition, NP-ALP incorporates infinite action spaces very naturally, while RALP requires significant extensions for large action spaces (Pazis and Parr 2011a).

Non-parametric approaches to policy iteration and value iteration have also been explored extensively in the literature. Munos and Moore (Munos and Moore 2002) considered variable resolution grids. Other approaches inspired by kernel regression have also been proposed (Ormoneit and Sen 2002). A more recent example is the work of Kroemer and Peters (2012) who also use kernel density

[7] This is a fairly strong yet necessary assumption for batch mode RL algorithms. The only meaningful way to relax this assumption is to tackle exploration directly, which goes beyond the scope of this paper (see Pazis and Parr (2013) for more details.). Weaker assumptions on the distribution of constraints can be easily reduced to this one by measuring how closely they resemble the uniform one.



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

Figure 1: Accumulated reward versus the Lipschitz constant with uniform noise in (a) $[-10, 10]N$ and (b) $[-20, 20]N$. Accumulated reward versus training episodes with uniform noise in (c) $[-10, 10]N$ and (d) $[-20, 20]N$. Averages and 95% confidence intervals for different numbers of nearest neighbors (NN) are over 100 independent runs.

estimates and Kveton and Theocharous (2012) who use cover trees to select a representative set of states. Fitted Q-Iteration with tree-based approximators (Ernst, Geurts, and Wehenkel 2005) is also a non-parametric method. Kernelized approaches (Taylor and Parr 2009) can also be viewed as non-parametric algorithms. In the family of kernelized methods, Farahmand et al. (2009) are notable for including sample complexity results and max-norm error bounds, but their bounds depend upon difficult to measure quantities, such as concentrability coefficients. In general, non-parametric approaches associated with policy iteration or value iteration tend to require more restrictive and complicated assumptions yet provide weaker guarantees.

6 Experimental Results

This section presents experimental results from applying NP-ALP to the continuous action inverted pendulum regulator problem (Wang, Tanaka, and Griffin 1996) with uniform noise in $[-10, 10]N$ and $[-20, 20]N$ applied to each action. Since both the model and a vast amount of accumulated knowledge are available for this domain, many algorithms exist that achieve good performance when taking advantage of this information. Our goal is not to claim that policies produced by NP-ALP outperform policies produced by such algorithms. Instead we want to demonstrate that we can tackle a very noisy problem even under the weakest of assumptions, with an algorithm that provides strong theoretical guarantees, providing some indication that NP-ALP would be able to perform well on domains where no such knowledge exists, and to show that the performance achieved by NP-ALP supports that which is predicted by our bounds.

Instead of the typical avoidance task, we chose to approach the problem as a regulation task, where we are not only interested in keeping the pendulum upright, but we want to do so while minimizing the amount of force a we are using. Thus a reward of $1 - (a/50)^2$ was given as long as $|\theta| \leq \pi/2$, and a reward of 0 as soon as $|\theta| > \pi/2$, which also signals the termination of the episode. The discount factor of the process was set to 0.98 and the control interval to 100ms. Coupled with the high levels of noise, making full use of the available continuous action range is required to get good performance in this setting.

The distance function was set to the two norm difference between state-actions, with the action space rescaled to $[-1, 1]$. Training samples were collected in advance by starting the pendulum in a randomly perturbed state close to the equilibrium state $(0, 0)$ and selecting actions uniformly at random.

Figure 1 shows total accumulated reward versus the Lipschitz constant with uniform noise in $[-10, 10]$ (a) and $[-20, 20]$ (b), for 3000 training episodes. Notice the logarithmic scale on the x axis. We can see that the shape of the graphs reflects that of the bounds. When $L_{\tilde{Q}}$ is too small, ϵ_C is large, while when $L_{\tilde{Q}}$ is too large, ϵ_d is large. Additionally, for small values of k, ϵ_s^- and ϵ_s^+ are large. One interesting behavior is that for small values of k, the best performance is achieved for large values of $L_{\tilde{Q}}$. We believe that this is because the larger $L_{\tilde{Q}}$ is, the smaller the area affected by each overestimation error. One can see that different values of k exhibit much greater performance overlap over $L_{\tilde{Q}}$ for smaller amounts of noise.

Figure 1 shows the total accumulated reward as a function of the number of training episodes with uniform noise in $[-10, 10]$ (c) and $[-20, 20]$ (d) with $L_{\tilde{Q}} = 1.5$. Again the observed behavior is the one expected from our bounds. While larger values of k ultimately reach the best performance even for high levels of noise, the $L_{\tilde{Q}} d_{k,\max}$ component of ϵ_s^- along with ϵ_d penalize large values of k when n is not large enough. In addition (perhaps unintuitively), for any constant k, increasing the number of samples beyond a certain point increases the probability that ϵ_s^+ will be large for some state ($\max_s \epsilon_s^+$), causing a decline in average performance and increasing variance. Thus, in practical applications the choice of k has to take into account the sample density and the level of noise. We can see that this phenomenon is more pronounced at higher noise levels, affecting larger values of k.

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

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* 6:503–556.

Farahmand, A.; Ghavamzadeh, M.; Szepesvari, C.; and Mannor, S. 2009. Regularized policy iteration. *Advances in Neural Information Processing Systems* 21:441–448.

Kroemer, O., and Peters, J. 2012. A non-parametric approach to dynamic programming. *Advances in Neural Information Processing Systems* 24:to appear.

Kveton, B., and Theocharous, G. 2012. Kernel-based reinforcement learning on representative states. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, (AAAI)*.

Munos, R., and Moore, A. 2002. Variable resolution discretization in optimal control. *Machine Learning* 49(2):291–323.

Ormoneit, D., and Sen, Ś. 2002. Kernel-based reinforcement learning. *Machine Learning* 49(2):161–178.

Pazis, J., and Parr, R. 2011a. Generalized value functions for large action sets. In *ICML-11*, 1185–1192. ACM.

Pazis, J., and Parr, R. 2011b. Non-Parametric Approximate Linear Programming for MDPs. In *AAAI-11*, 793–800. AAAI Press.

Pazis, J., and Parr, R. 2013. PAC optimal exploration in continuous space markov decision processes. In *AAAI-13*. AAAI Press.

Petrik, M.; Taylor, G.; Parr, R.; and Zilberstein, S. 2010. Feature selection using regularization in approximate linear programs for Markov decision processes. In *ICML-10*, 871–878. Haifa, Israel: Omnipress.

Taylor, G., and Parr, R. 2009. Kernelized value function approximation for reinforcement learning. In *ICML '09*, 1017–1024. New York, NY, USA: ACM.

Wang, H.; Tanaka, K.; and Griffin, M. 1996. An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transactions on Fuzzy Systems* 4(1):14–23.