Towards Feature Selection In Actor-Critic Algorithms

Khashayar Rohanimanesh KHASH@CS.UMASS.EDU
University of Massachusetts, Amherst, MA 01003

Nicholas Roy and Russ Tedrake {NICKROY,RUSST}@CSAIL.MIT.EDU
Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139

Abstract

Choosing features for the critic in actor-critic algorithms with function approximation is known to be a challenge. Too few critic features can lead to degeneracy of the actor gradient, and too many features may lead to slower convergence of the learner. In this paper, we show that a well-studied class of actor policies satisfy the known requirements for convergence when the actor features are selected carefully. We demonstrate that two popular representations for value methods - the barycentric interpolators and the graph Laplacian proto-value functions - can be used to represent the actor in order to satisfy these conditions. A consequence of this work is a generalization of the proto-value function methods to the continuous action actor-critic domain. Finally, we analyze the performance of this approach using a simulation of a torque-limited inverted pendulum.

1. Introduction

Actor-Critic (AC) algorithms, initially proposed by (Barto et al., 1983), aim at combining the strong elements of the two major classes of reinforcement learning algorithms – namely the value-based methods and the policy search methods. As in value-based methods, the critic component maintains a value function, and as in policy search methods, the actor component maintains a separate parameterized stochastic policy from which the actions are drawn. This combination may offer the convergence guarantees which are characteristic of the policy gradient algorithms as well as an improved convergence rate because the critic can be used to reduce the variance of the policy update (Konda and Tsitsiklis, 2003).

Appearing in Proceedings of the ICML/UAI/COLT Workshop on Abstraction in Reinforcement Learning, Montreal, Canada, 2009.
Copyright 2009 by the author(s)/owner(s).

Recent AC algorithms use a function approximation architecture to maintain both the actor policy and the critic (state-action) value function, relying on Temporal Difference (TD) learning methods to update the critic parameters. Konda and Tsitsiklis (2000) and Sutton et al. (2000) showed that in order to compute the gradient of the performance function (typically using the average cost criterion) with respect to the parameters of a stochastic policy $\mu_\theta(\mathbf{x}, \mathbf{u})$ it suffices to compute the projection of the state-action value function onto a sub-space $\mathbf{\Psi}$ spanned by the vectors $\psi_\theta^i(x,u) = \frac{\partial}{\partial\theta_i}\log\mu_\theta(\mathbf{x}, \mathbf{u})$. Konda and Tsitsiklis (2003) also noted that for certain values of the policy parameters θ, it is possible that the vectors ψ_θ^i are either close to zero, or almost linearly dependent. In these situations the projection onto $\mathbf{\Psi}$ becomes ill-conditioned, providing no useful gradient information, and the algorithm can become unstable. As a remedy for this problem the authors suggested the use of a richer, higher dimensional set of critic features which contain the space $\mathbf{\Psi}$ as a proper subset.

In this paper, we attempt to design features which span $\mathbf{\Psi}$ and preserve linear independence without increasing the dimensionality of the critic. In particular, we investigate stochastic actor policies represented by a family of Gaussian distributions where the mean of the distribution is linearly parameterized using a set of a fixed basis functions. For this parameterization, we show that if the basis functions in the actor are selected to be linearly independent, then the minimal set of critic features which naturally satisfy the containment condition also form a linearly independent basis set.

2. Preliminaries

In this section we present a brief overview of the AC algorithms with function approximation adapted from (Konda and Tsitsiklis, 2003). Assume that the problem is modeled as a Markov decision process $\mathcal{M} = \langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{C} \rangle$, where $\mathcal{X}$ is the state space, $\mathcal{U}$ is the action space, $\mathcal{P}(x'|x,u)$ is the transition probability function, $\mathcal{C} : \mathcal{X} \times \mathcal{U} \to \Re$ is the one step cost function, and μ_θ is a stochastic policy parameterized by $\theta \in \Re^n$ where $\mu_\theta(u|x)$ gives the probability of selecting an action u in state x, parameterized by the vector $\theta \in \Re^n$. We also assume that for every $\theta \in \Re^n$, the Markov chains $\{X_k\}$ and $\{X_k, U_k\}$ are irreducible and aperiodic, with stationary probabilities $\pi_\theta(x)$ and $\eta_\theta(x,u) = \pi_\theta(x)\mu_\theta(u|x)$ respectively.

The average cost function $\bar{\alpha}(\theta) : \Re^n \rightarrow \Re$ can be defined:

$$\bar{\alpha}(\theta) = \sum_{x \in \mathcal{X}, u \in \mathcal{U}} c(x,u)\eta_\theta(x,u)$$

For each $\theta \in \Re^n$, let $\mathcal{V}_\theta : \mathcal{X} \rightarrow \Re$, and $\mathcal{Q}_\theta : \mathcal{X} \times \mathcal{U}$ be the differential state, and the differential state-action cost functions that are solution to the corresponding Poisson equations in a standard average cost setting. Then, following the results of (Marbach and Tsitsiklis, 1998), the gradient of the average cost function can be expressed as:

$$\nabla_\theta \bar{\alpha}(\theta) = \sum_{x,u} \eta_\theta(x,u)\mathcal{Q}_\theta(x,u)\psi_\theta(x,u) \qquad (1)$$

where:

$$\psi_\theta(x,u) = \nabla_\theta \ln \mu_\theta(u|x) \qquad (2)$$

The ith component of ψ_θ, $\psi_\theta^i(x,u)$ is the one-step *eligibility* of parameter i in state-action pair x and u given by $\psi_\theta^i(x,u) = \frac{\partial}{\partial \theta_i} \ln \mu_\theta(u|x)$. We will therefore refer to ψ_θ^i as the *actor eligibility vector*, a vector in $\Re^{|\mathcal{X}||\mathcal{U}|}$. For any $\theta \in \Re^n$, the inner product $\langle \cdot, \cdot \rangle_\theta$ of two real-valued functions $\mathcal{Q}_1, \mathcal{Q}_2$ on $\mathcal{X} \times \mathcal{U}$, also viewed as vectors in $\Re^{|\mathcal{X}||\mathcal{U}|}$, can be defined by:

$$\langle \mathcal{Q}_1, \mathcal{Q}_2 \rangle_\theta = \sum_{x,u} \eta_\theta(x,u)Q_1(x,u)Q_2(x,u)$$

and let $\|\cdot\|_\theta$ denote the norm induced by this inner product on $\Re^{|\mathcal{X}||\mathcal{U}|}$. Now, we can rewrite Equation 1 as:

$$\frac{\partial}{\partial \theta_i}\bar{\alpha}(\theta) = \langle \mathcal{Q}_\theta, \psi_\theta^i \rangle_\theta, \quad i = 1, \ldots, n.$$

For each $\theta \in \Re^n$, let Ψ_θ denote the span of the vectors $\{\psi_\theta^i; 1 \leq i \leq n\}$ in $\Re^{|\mathcal{X}||\mathcal{U}|}$. An important observation is that although the gradient of $\bar{\alpha}$ depends on the function $\mathcal{Q}_\theta$, which is a vector in a possibly very high-dimensional space $\Re^{|\mathcal{X}||\mathcal{U}|}$, the dependence is only through its inner products with vectors in Ψ_θ. Thus, instead of "learning" the function $\mathcal{Q}_\theta$, it suffices to learn its projection on the low-dimensional sub-space Ψ_θ.

Konda and Tsitsiklis (2003) consider actor-critic algorithms where the critic is a TD algorithm with a linearly parameterized approximation architecture for the $\mathcal{Q}$-value function that admits the linear-additive form:

$$\mathcal{Q}_\theta^r(x,u) = \sum_{j=1}^{m} r^j \phi_\theta^j(x,u) \qquad (3)$$

where $r = (r^1, \ldots, r^m) \in \Re^m$ is the parameter vector of the critic. The critic features $\phi_\theta^j, j = 1, \ldots, m$ depend on the actor parameter vector and are chosen so that the following assumptions are satisfied: (1) For every $(x,u) \in \mathcal{X} \times \mathcal{U}$, the map $\theta \rightarrow \phi_\theta(x,u)$ is bounded and differentiable; (2) The span of the vectors ϕ_θ^j $(j = 1, \ldots, m)$ in $\Re^{|\mathcal{X}||\mathcal{U}|}$ denoted by Φ_θ, contains Ψ_θ.

As noted by (Konda and Tsitsiklis, 2003), one trivial choice for satisfying the second condition would be to set $\Psi = \Phi$, or in other words to set critic features as $\phi_\theta^i = \psi_\theta^i$. However, it is possible that for some values of θ, the features ψ_θ^i are either close to zero, or almost linearly dependent. In these situations the projection of $\mathcal{Q}_\theta^r$ onto $\mathbf{\Psi}$ becomes ill-conditioned, providing no useful gradient information, and therefore the algorithm may become unstable. Konda and Tsitsiklis (2003) suggest some ideas to remedy to this problem. In particular, the troublesome situations are avoided if the following condition is satisfied: (3) There exists $a > 0$, such that for every $r \in \Re^m$, and $\theta \in \Re^n$:

$$\| r' \hat{\phi}_\theta \|_\theta^2 \;\geq\; a|r|^2$$

where $\hat{\phi} = \{\hat{\phi}^i\}_{i=1}^m$ are defined as:

$$\hat{\phi}_\theta^i(x,u) = \phi_\theta^i(x,u) - \sum_{\bar{x},\bar{u}} \eta_\theta(\bar{x},\bar{u})\,\phi_\theta^i(\bar{x},\bar{u}) \qquad (4)$$

This condition can be roughly explained as follows: the new functions $\hat{\phi}_\theta^i$ can be viewed as the original critic features with their expected value (with respect to the distribution $\eta_\theta(x,u)$) removed. In order to ensure that the projection of $\mathcal{Q}_\theta^r$ onto $\mathbf{\Psi}$ contains some gradient information for the actor (and to avoid instability), the set $\hat{\phi}_\theta$ must be uniformly bounded away from zero. Given these conditions, Konda and Tsitsiklis (2003) prove convergence for of the most common form for the actor-critic update (see (Konda and Tsitsiklis, 2003, p.1148) for the updates).

Konda and Tsitsiklis (2003) go on to propose adding additional features to the critic as a remedy, but satisfying this condition is still a difficult problem. To the best of our knowledge there is no general systematic approach for choosing a set of critic features that satisfies this third condition. In the next section, we will address this issue for one commonly used policy class.

3. Our Approach

We consider the following popular Gaussian probabilistic policy structure parameterized by θ:

$$\mu_\theta(\mathbf{u}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{u} - \mathbf{m}_\theta(\mathbf{x}))^T \Sigma^{-1} (\mathbf{u} - \mathbf{m}_\theta(\mathbf{x}))\} \qquad (5)$$

where $\mathbf{u} \in \Re^k$ is a multi-dimensional action vector. The vector $\mathbf{m}_\theta(\mathbf{x}) \in \Re^k$ is the mean of the distribution that is parameterized by θ:

$$\mathbf{m}_\theta^i(\mathbf{x}) = \sum_{j=1}^{n} \theta^{ij}\,\rho^j(\mathbf{x}), \quad i = 1, \ldots, k \tag{6}$$

where in this setting $\theta \in \Re^{k\times n}$. The functions $\rho^j(\mathbf{x})$ are a set of actor features defined over the states. For simplicity, in this paper we only investigate the case where $\Sigma = \sigma_0^2\,\mathbf{I}$. In this case Equation 5 simplifies to:

$$\mu_\theta(\mathbf{u}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{k}{2}}\sigma_0^k}\,\exp\{-\frac{1}{2\sigma_0^2}(\mathbf{u}-\mathbf{m}_\theta(\mathbf{x}))^T(\mathbf{u}-\mathbf{m}_\theta(\mathbf{x}))\} \tag{7}$$

Using Equation 2 we can compute the actor eligibility vectors as follows:

$$\begin{aligned} \psi_\theta^{ij}(\mathbf{x},\mathbf{u}) &= \frac{\partial}{\partial\theta^{ij}}\ln\mu_\theta(\mathbf{u}|\mathbf{x}) \\ &= \frac{\partial}{\partial\theta^{ij}}\Big[-\ln((2\pi)^{\frac{k}{2}}\sigma_0^k)- \\ &\quad \frac{1}{2\sigma_0^2}(\mathbf{u}-\mathbf{m}_\theta(\mathbf{x}))^T(\mathbf{u}-\mathbf{m}_\theta(\mathbf{x}))\}\Big] \\ &= \frac{1}{\sigma_0^2}(\mathbf{u}-\mathbf{m}_\theta(\mathbf{x}))^T\frac{\partial}{\partial\theta^{ij}}\mathbf{m}_\theta(\mathbf{x}) \\ &= \frac{1}{\sigma_0^2}(u_i-\mathbf{m}_\theta^i(\mathbf{x}))\rho^j(\mathbf{x}) \\ &= \kappa_\theta^i(\mathbf{x},\mathbf{u})\rho^j(\mathbf{x}) \end{aligned} \tag{8}$$

where $\kappa_\theta^i(\mathbf{x},\mathbf{u}) = \frac{(u_i-\mathbf{m}_\theta^i(\mathbf{x}))}{\sigma_0^2}$. In order to satisfy the condition (2) in the previous section ($\mathbf{\Phi}$ should properly contain $\mathbf{\Psi}$), we apply the straightforward solution of setting $\phi^{ij} = \psi^{ij}$ for $i = 1,\ldots,k$ and $j = 1,\ldots,n$. This selection also guarantees that the mapping from θ to ϕ_θ is bounded and differentiable, from condition (1). In Proposition 1, we show that for the particular choice of policy structure that we have chosen, if the actor features, $\rho^j(\mathbf{x})$, are linearly independent, then the critic features will also be linearly independent.

Proposition 1: If the functions $\rho = \{\rho^j\}_{j=1}^n$ are linearly independent, then the set of critic feature functions ϕ^{ij} will form a linearly independent set of functions.

Proof: We prove by contradiction that if the above condition holds, then the set of actor eligibility functions ψ^{ij} (and therefore also ϕ^{ij}) are linearly independent. Assume that the functions ψ^{ij} are linearly dependent. Then there exists $\alpha = \{\alpha_{ij} \in \Re\}_{i=1,j=1}^{k,n}$ such that

$$\sum_{i=1,j=1}^{k,n}\alpha_{ij}\,\psi^{ij}(\mathbf{x},\mathbf{u}) = 0, \;\; \forall\mathbf{x}\in\mathcal{X},\,\forall\mathbf{u}\in\mathcal{U},$$

and $\parallel\alpha\parallel^2 > 0$. Substituting the right hand side of the Equation 8 for $\psi^{ij}(\mathbf{x},\mathbf{u})$ yields:

$$\sum_{i=1,j=1}^{k,n}\alpha_{ij}\,\kappa_\theta^i(\mathbf{x},\mathbf{u})\rho^j(\mathbf{x}) = 0, \;\; \forall\mathbf{x}\in\mathcal{X},\,\forall\mathbf{u}\in\mathcal{U}.$$

By regrouping terms we obtain:

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{k}\alpha_{ij}\kappa_\theta^i(\mathbf{x},\mathbf{u})\right)\rho^j(\mathbf{x}) = 0, \;\; \forall\mathbf{x}\in\mathcal{X},\,\forall\mathbf{u}\in\mathcal{U}.$$

Since according to the assumption the functions $\rho = \{\rho^j\}_{j=1}^n$ are linearly independent, then the following condition must hold:

$$\sum_{i=1}^{k}\alpha_{ij}\kappa_\theta^i(\mathbf{x},\mathbf{u}) = 0, \;\; \forall j=1,\ldots,n,\,\forall\mathbf{x}\in\mathcal{X},\,\forall\mathbf{u}\in\mathcal{U} \tag{9}$$

But for every i, there exists an $(\mathbf{x},\mathbf{u})$ such that $\kappa_\theta^i(\mathbf{x},\mathbf{u}) \neq 0$:

$$\kappa_\theta^i(\mathbf{x},\mathbf{m}_\theta(\mathbf{x})+\epsilon_i\mathbf{1}) = \frac{\epsilon_i}{\sigma_0^2}, \tag{10}$$

where $\mathbf{1}$ is the $k\times 1$ vector of ones, and $\epsilon_i \neq 0$. Note that the above condition holds for all $\epsilon_i \in \Re - \{0\}$. Now, define a $(k\times 1)$ vector $\mathbf{h}_l$ (for $l = 1,\ldots,k$) as:

$$\mathbf{h}_l(j) = \{ \begin{array}{ll} \epsilon & \text{if } j \neq l \\ 2\epsilon & \text{if } j = l \end{array} \tag{11}$$

for some $\epsilon > 0$. Based on Equation 10, if we choose $\mathbf{u} = \mathbf{m}_\theta(\mathbf{x}) + \mathbf{h}_l$ in Equation 9, we obtain:

$$\frac{1}{\sigma_0^2}\mathbf{h}_l{}^T\bar{\alpha}_{ij} = 0, \quad \forall l = 1,\ldots,k$$

where $\bar{\alpha}_{ij} = [\alpha_{1j},\alpha_{2j},\ldots,\alpha_{kj}]^T$. This gives us the following system of equations (for a fixed value of j):

$$\mathbf{A}\,\bar{\alpha}_{ij} = \mathbf{0}, \quad i = 1,\ldots,k \tag{12}$$

where $\mathbf{A}_{k\times k} = [\mathbf{h}_1\mathbf{h}_2\ldots\mathbf{h}_k]^T$. Note that for the particular choice of the vectors $\mathbf{h}_l$ (Equation 11), the matrix $\mathbf{A}$ has a full-rank (since the vectors $\mathbf{h}_l$ are linearly independent), and thus the only solution to the Equation 12 is $\bar{\alpha}_{ij} = \mathbf{0}$. This means that $\alpha_{ij} = 0$ (for all i,j), and thus $\parallel\alpha\parallel^2 = 0$. By contradiction, ψ^{ij} (and therefore ϕ^{ij}) must be linearly independent.

Proposition 1 provides a mechanism for ensuring that the θ-dependent critic features remain linearly independent for all θ's, thereby avoiding a major source of potential instabilities in the AC algorithm. However, to meet the strict conditions from (Konda and Tsitsiklis, 2003), we should also demonstrate that the critic features are uniformly bounded away from zero. Proposition 2 allows us to demonstrate that a set of actor features that is also linearly independent with the function $\underline{\mathbf{1}}$ satisfies the weak form of condition (3).

Proposition 2: If the functions $\underline{\mathbf{1}}$ and $\rho = \{\rho^j\}_{j=1}^n$, $j = 1, \ldots, n$ are linearly independent, then the set of critic feature functions ϕ^{ij} and the function $\underline{\mathbf{1}}$, will also form a linearly independent set of functions.

Proof (sketch): We follow the proof of the proposition 1. Assume that the functions ψ^{ij} are linearly dependent. Then there exists $\alpha = \{\alpha_{ij} \in \Re\}_{i=1,j=1}^{k,n} \cup \{\alpha_1 \in \Re\}$ such that:

$$\sum_{i=1,j=1}^{k,n} \alpha_{ij}\, \psi^{ij}(\mathbf{x},\mathbf{u}) + \alpha_1 \underline{\mathbf{1}} = 0, \;\; \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{u} \in \mathcal{U},$$

and $\parallel \alpha \parallel^2 > 0$. Following the same steps as in proof of proposition 1, we obtain:

$$\sum_{j=1}^{n} \left(\sum_{i=1}^{k} \alpha_{ij} \kappa_\theta^i(\mathbf{x},\mathbf{u}) \right) \rho^j(\mathbf{x}) + \alpha_1 \underline{\mathbf{1}} = 0, \;\; \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{u} \in \mathcal{U}. \tag{13}$$

Since according to the assumption the functions $\rho = \{\rho^j\}_{j=1}^n$ and $\underline{\mathbf{1}}$ are linearly independent, then $\alpha_1 = 0$. Following the rest of the steps in proof of the proposition 1, it can be also established that $\alpha_{ij} = 0$ (for all i,j). This completes the proof.

Konda and Tsitsiklis (2003) prove that if the functions $\underline{\mathbf{1}}$ and the critic features ϕ_θ^i are linearly independent for each θ, then there exists a positive function $a(\theta)$ such that:

$$\parallel r' \, \hat{\phi}_\theta \parallel_\theta^2 \;\; \geq \; a(\theta) \parallel r \parallel^2 \tag{14}$$

(refer to Section 2, Equation 4 for the definition of $\hat{\phi}_\theta$). This is the weak form of the non-zero projection property.

Finally, it should be noted that it is also possible to tune the standard-deviation of the policy distribution, σ_0, as a function of state using additional policy parameters, $\mathbf{w}$. If we parameterize the variance as:

$$\sigma_0(x) = [1 + \exp(-\sum_i w_i \rho^i(x))]^{-1}$$

then the eligibility of this actor parameter takes the form:

$$\begin{aligned} \frac{\partial}{\partial w_i} \ln \mu_{\theta,w}(x,u) = \\ \left((u - m_\theta(x))^2 - \sigma_0^2(x)\right)(1 - \sigma_0(x))\, \rho^i(x) \\ = \kappa_{\theta,w}^i(x,u) \rho^j(x) \end{aligned}$$

It can be shown that this set of vectors forms a linearly independent basis set, which is also independent of the bases Ψ.

4. Candidate Features

In this section we investigate two different approaches for choosing linearly independent actor features, $\rho^j(\mathbf{x})$.

4.1. Unit Basis Functions

Unit Basis Functions are the simplest linear independent basis set. For a random walk of size N in the state space, we can define a unit basis set $\mathcal{U} = \{u^i\}_{i=1}^m$, for some $m \leq N$, where u^i is a unit vector of size N, with a 1 at i^{th} position, and zero elsewhere (note that by reordering i we can select different subset of the nodes of the random walk).

Proposition 3: For any given set of unit basis functions $\mathcal{U}$, if $|\mathcal{U}| < N$, then $\mathcal{U}$ will satisfy the weak form of the non-zero projection property presented in Equation 14.

Proof (sketch): We know that the unit basis functions defined over a space of dimension N are linearly independent. Since $|\mathcal{U}| < N$, then they are also linearly independent of the function $\underline{\mathbf{1}}$. Based on the results of the Proposition 2, the critic feature functions ϕ^{ij} are also linearly independent and also linearly independent of the function $\underline{\mathbf{1}}$, and the features will satisfy the weak non-zero projection property.

4.2. Barycentric Interpolation

Barycentric interpolants described in (Munos and Moore, 1998, 2002) are defined as an arbitrary set of (non-overlapping) mesh points ξ_i distributed across the state space. We denote the vector-valued output of the function approximator at each mesh point as $\mathbf{m}(\xi_i)$. For an arbitrary $\mathbf{x}$, if we define a simplex $S(\mathbf{x}) \in \{\xi_1, ..., \xi_N\}$ such that $\mathbf{x}$ is in the interior of the simplex, then the output at $\mathbf{x}$ is given by interpolating the mesh points $\xi \in S(\mathbf{x})$: $\mathbf{m}_\theta(\mathbf{x}) = \sum_{\xi_i \in S(\mathbf{x})} \mathbf{m}(\xi_i) \lambda_{\xi_i}(\mathbf{x})$. Note that the interpolation is called *barycentric* if the positive coefficients $\lambda_{\xi_i}(\mathbf{x})$ sum to one, and if $\mathbf{x} = \sum_i \xi_i \lambda_{\xi_i}(\mathbf{x})$ (Munos and Moore, 1998). In addition, Munos and Moore (1998) denote the *piecewise linear* barycentric interpolation functions as functions for which the interpolation uses exactly $dim(\mathbf{x}) + 1$ mesh points such that the simplex for state $\mathbf{x}$ is the simplex which forms a triangulation of the state space and does not contain any interior mesh points. Barycentric interpolators are a popular representation for value functions, because they provide a natural mechanism for variable resolution discretization of the value function, and the barycentric co-ordinates allow the interpolators to be used directly by value iteration algorithms.

These interpolators also represent a linear function approximation architecture; we confirm here that the feature vectors are linearly independent. Let us consider the output at the mesh points as the parameters, $\theta_i = \mathbf{m}(\xi_i)$, and the interpolation function as the features $\rho_i(\mathbf{x}) = \lambda_{\xi_i}(\mathbf{x})$.

Proposition 4: The features $\rho_i(\mathbf{x})$ formed by the piecewise linear barycentric interpolation of a non-overlapping mesh ($\xi_i \neq \xi_j, \forall i \neq j$) form a linearly independent basis set.

Proof (sketch): For a non-overlapping mesh, consider the solution for the barycentric weights of a piecewise linear barycentric interpolation evaluated at $\mathbf{x} = \xi_i$. There are multiple simplices $S(\mathbf{x})$ that contain $\mathbf{x}$, but for each such simplex, $\mathbf{x}$ is a vertex of that simplex. By definition of a simplex, $\mathbf{x}$ is linearly independent of all other vertices of each simplex. As a result, the unique solution for the barycentric weights is $\rho_i(\mathbf{x}) = 1, \rho_j(\mathbf{x}) = 0, \forall j \neq i$. Since for each feature we can find an $\mathbf{x}$ which is non-zero for only that feature, the basis set must be linearly independent. Note that the traditional barycentric interpolators are *not* constrained to be linearly independent from the function $\underline{\mathbf{1}}$.

4.3. Graph Laplacian

Proto-Value Functions (PVFs) (Mahadevan and Maggioni, 2007) have recently shown some success in automatic learning of representations in the context of function approximation in MDPs. In this approach, the agents learn global task-independent basis functions that reflect the large-scale geometry of the state-action space that all task-specific value functions must adhere to. Such basis functions are learned based on the topological structure of graphs representing the state (or state-action) space manifold. PVFs are essentially a subset of eigenfunctions of the graph *Laplacian* computed from a random walk graph generated by the agent. We show here that if the proto-value functions are used instead to represent features of the actor, instead of the critic, then this representation satisfies our Proposition 2.

Proposition 5: If the functions $\{\rho^j\}_{j=1}^n$ are the proto-value functions computed from the graph generated by a random walk in state space, then the set of critic features and the function $\underline{\mathbf{1}}$ will form a linearly independent basis set, and will satisfy the weak form of the non-zero projection property presented in Equation 14.

Proof (sketch): Since the functions $\{\rho^j\}_{j=1}^n$ are the eigenfunctions of the graph Laplacian computed from the graph generated by a random walk in state space, they are linearly independent. Note that the function $\underline{\mathbf{1}}$ is always the eigenfunction of any graph Laplacian associated with the eigenvalue 0. That implies the functions $\{\rho^j\}_{j=1}^n$ are also linearly independent of the function $\underline{\mathbf{1}}$. Based on the results of the Proposition 2, the critic feature functions ϕ^{ij} are also linearly independent and also linearly independent of the function $\underline{\mathbf{1}}$, and the features will satisfy the weak non-zero projection property.

5. Experiments

We demonstrate the effectiveness of our feature selections by learning a control policy for the swing-up task on a torque-limited inverted pendulum, governed by $\ddot{q} = \frac{1}{ml^2}\left[\tau - b\,\dot{q} - m\,g\,l\,\cos(q)\right]$, with $m = 1, l = 1, b = 1, g = 9.8, |\tau| < 1$, and initial conditions $q = -\frac{\pi}{2}, \dot{q} = 0$. We use an infinite-horizon, average reward formulation (no resetting) with the instantaneous cost function:

$$g(q, \dot{q}, \tau) = \frac{1}{2}(q - \frac{\pi}{2})^2 + \frac{1}{2}\dot{q}^2 + \frac{1}{10}\tau^2$$

The policy is evaluated every $dt = 0.1$ seconds; τ is held constant (zero-order hold) between evaluations.

BASIS FUNCTIONS: We employed a variety of basis functions parameterizing the actor's policy (i.e., the ρ basis functions in Equation 6) as follows:

Barycentric Interpolators: We use linear barycentric features on a uniform mesh over the state space, with 16 bins on θ over the interval $[-\pi, \pi]$, and 10 bins on $\dot{\theta}$ over the interval $[-1, 1]$.

Proto Value Functions (PVFs): We generated a random walk of size 4781 using rapidly-exploring randomized trees (RRTs) (LaValle and Kuffner, 2000) for coverage. Note that this is in place of the traditional "behavioral policy" used to identify the proto-value functions; it provides a fast and efficient coverage of our continuous state space. We then computed the Laplacian eigen-vectors and used a set of 10 eigen functions (eigen-vectors 2-11) in all of our experiments. For generalizing to unseen states, we used a weighted average of 20 nearest neighbors of that state to approximate the policy in that state.

Perturbed PVFs: We also generated a set of perturbed PVF basis functions, by perturbing the original PVF basis set computed as above using a Gaussian noise. In our experiments the model $PVF + \mathcal{N}(0, \sigma_p^2)$ refers to an experiment where the original PVFs (consisting of 10 eigen functions (eigen vectors 2-11)) are perturbed using a Gaussian noise $\mathcal{N}(0, \sigma_p^2)$. The primary reason for using a noisy PVF basis set is to investigate the convergence properties of the model as a function of noise in the original basis set.

Unit Basis Functions: We used a set of 10 unit basis functions defined at 10 random nodes of the random walk over the state space.

Polynomial Basis Functions: We used a polynomial function of degree 4 (i.e., $(1, q, \dot{q}, q\,\dot{q}, \ldots, q^4, \dot{q}^4)$) for a total of 15 for approximating the actor's policy.

Radial Basis Functions (RBFs): We used a set of 10 radial basis functions for approximating the actor's policy . These 10 basis functions included a constant term and 9 radial basis functions (Gaussians):

$$\left(1, exp\{-\frac{\parallel s - \mu_{rbf}^1 \parallel^2}{2\sigma_{rbf}^2}\}, \ldots, exp\{-\frac{\parallel s - \mu_{rbf}^9 \parallel^2}{2\sigma_{rbf}^2}\right)$$

Basis Functions	Converged (percentage)
Unit	100 %
PVF	100 %
Barycentric*	100 %
RBF	10 %
PVF + $\mathcal{N}(0, 0.10)$	95 %
PVF + $\mathcal{N}(0, 0.50)$	45 %
PVF + $\mathcal{N}(0, 1.0)$	0 %
PVF + $\mathcal{N}(0, 1.5)$	0 %
PVF + $\mathcal{N}(0, 2.0)$	0 %
Polynomial	0 %

Table 1. Percentage of convergence of AC with various function approximation methods computed over 20 runs of each method.

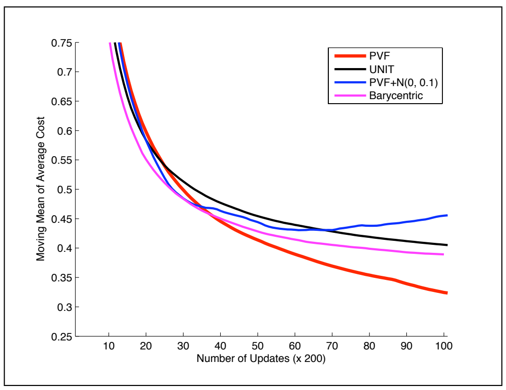

Figure 1. Performance comparison among AC with using PVF basis functions, AC with unit basis functions, and AC using perturbed set of PVF basis functions.

for a state $s = (q, \dot{q})$. Here the $\{\mu_{rbf}^i\}_{i=1}^9$ are the 9 points of the grid covering the state space located at $\{\pi/2, 0, \pi/2\} \times \{-1, 0, 1\}$. In order to evaluate the convergence properties of the AC with the above basis functions, we setup up an experiment where we measured the percentage of convergence of each method over 20 runs of each, for total of 4000 steps. The results are reported in Table 1. As we can see in this table, the representations on which our convergence results apply (Unit, Barycentric, and PVF), the algorithm did converge experimentally on every run. More surprising was the observation that other commonly used features actually diverge in relatively benign experiments. The RBF experiments only converged on 10% of the experiments, and AC with polynomial basis functions diverged at every single experiment (with polynomial basis functions, we also tried different learning rates in AC algorithm; the experiments diverged quickly).

Figure 1 shows the moving mean of the average cost for four different approaches for every 200 steps, namely AC with PVF, AC with perturbed PVF (with Gaussian noise $\mathcal{N}(0, 0.10)$), AC with unit basis functions, and finally AC with barycentric interpolation. In all methods the policy is parameterized as in Equation 5 using the corresponding basis set. Each trial starts the pendulum from the initial condition, with the parameters of the actor and critic initialized to small random values. AC with PVF achieves the best performance, followed closely by barycentric. Note that the AC with perturbed PVF basis set initially performs better then the AC with unit basis functions, however it the performance degrades by time and it slowly diverges.

6. Conclusions

In this paper, we provide some insights for designing features for actor-critic algorithms with function approximation. For a limited policy class, we demonstrate that a linearly independent feature set in the actor permits a linearly independent feature set in the critic. This condition is satisfied by the piecewise linear barycentric interpolators, and by the features based on a graph Laplacian. When combined with an additional linear independence with the function $\underline{\mathbf{1}}$, the critic features for any particular θ are uniformly bounded away from zero. This condition is satisfied by the graph Laplacian features. Finally, our experimental results demonstrate that our proposed representation smoothly and efficiently converges to a local minimum for a simulated inverted pendulum control task.

References

Barto, A., Sutton, R., and Anderson, C. (1983). Neuron-like elements that can solve difficult learning control problems. In *IEEE Trans. on Systems, Man and Cybernetics*, volume 13, pages 835–846.

Konda, V. and Tsitsiklis, J. (2000). Actor-critic algorithms. In *Advances in Neural Information Processing Systems 12*.

Konda, V. and Tsitsiklis, J. (2003). Actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166.

LaValle, S. and Kuffner, J. (2000). Rapidly-exploring random trees: Progress and prospects. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

Mahadevan, S. and Maggioni, M. (2007). Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *J. Mach. Learn. Res.*, 8:2169–2231.

Marbach, P. and Tsitsiklis, J. (1998). Simulation-based optimization of markov reward processes.

Munos, R. and Moore, A. (1998). Barycentric interpolators for continuous space and time reinforcement learning. In Kearns, M. S., Solla, S. A., and Cohn, D. A., editors, *Advances in Neural Information Processing Systems*, volume 11, pages 1024–1030. NIPS, MIT Press.

Munos, R. and Moore, A. (2002). Variable resolution discretization in optimal control. *Machine Learning*, 49(2/3):291–323.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063.