Basis Adaptation for Sparse Nonlinear Reinforcement Learning

Sridhar Mahadevan, Stephen Giguere, and Nicholas Jacek
School of Computer Science
University of Massachusetts, Amherst
mahadeva@cs.umass.edu, nicolas.jacek@gmail.com, sgiguere9@gmail.com

Abstract

This paper presents a new approach to representation discovery in reinforcement learning (RL) using basis adaptation. We introduce a general framework for basis adaptation as *nonlinear separable least-squares value function approximation* based on finding Fréchet gradients of an error function using variable projection functionals. We then present a scalable proximal gradient-based approach for basis adaptation using the recently proposed *mirror-descent* framework for RL. Unlike traditional temporal-difference (TD) methods for RL, mirror descent based RL methods undertake proximal gradient updates of weights in a *dual space*, which is linked together with the primal space using a Legendre transform involving the gradient of a strongly convex function. Mirror descent RL can be viewed as a proximal TD algorithm using Bregman divergence as the distance generating function. We present a new class of regularized proximal-gradient based TD methods, which combine feature selection through sparse L_1 regularization and basis adaptation. Experimental results are provided to illustrate and validate the approach.

Introduction

There has been rapidly growing interest in reinforcement learning in *representation discovery* (Mahadevan 2008). *Basis construction* algorithms (Mahadevan 2009) combine the learning of features, or basis functions, and control. *Basis adaptation* (Bertsekas and Yu 2009; Castro and Mannor 2010; Menache, Shimkin, and Mannor 2005) enables tuning a given *parametric basis*, such as the Fourier basis (Konidaris, Osentoski, and Thomas 2011), radial basis functions (RBF) (Menache, Shimkin, and Mannor 2005), polynomial bases (Lagoudakis and Parr 2003), etc. to the geometry of a particular Markov decision process (MDP). *Basis selection* methods combine sparse feature selection through L_1 regularization with traditional *least-squares* type RL methods (Kolter and Ng 2009; Johns, Painter-Wakefield, and Parr 2010), linear complementarity methods (Johns, Painter-Wakefield, and Parr 2010), approximate linear programming (Petrik et al. 2010), or convex-concave optimization methods for sparse off-policy TD-learning (Liu, Mahadevan, and Liu 2012).

In this paper, we present a new framework for basis adaptation as nonlinear separable least-squares approximation of value functions using *variable projection functionals*. This framework is adapted from a well-known classical method for nonlinear regression (Golub and Pereyra 1973), when the model parameters can be decomposed into a *linear set*, which is fit by classical least-squares, and a *nonlinear set*, which is fit by a Gauss-Newton method based on computing the gradient of an error function based on variable projection functionals.

Mirror descent is a highly scalable online convex optimization framework (Nemirovksi and Yudin 1983). Online convex optimization (Zinkevich 2003) explores the use of first-order gradient methods for solving convex optimization problems. Mirror descent can be viewed as a first-order *proximal gradient* based method (Beck and Teboulle 2003) using a distance generating function that is a Bregman divergence (Bregman 1967). We combine basis adaptation with mirror-descent RL (Mahadevan and Liu 2012), a recently developed first-order approach to sparse RL. The proposed approach is also validated with some experiments showing improved performance compared to previous work.

Reinforcement Learning

Reinforcement learning can be viewed as a stochastic approximation framework (Borkar 2008) for solving MDPs, which are defined by a set of states S, a set of (possibly state-dependent) actions A (A_s), a dynamical system model comprised of the transition probabilities $P^a_{ss'}$ specifying the probability of transitioning to state s' from state s under action a, and a reward model R specifying the payoffs received. A policy $\pi : S \rightarrow A$ is a deterministic mapping from states to actions. Associated with each policy π is a value function V^π, which is a fixed point of the Bellman equation:

$$V^\pi = T^\pi(V^\pi) = R^\pi + \gamma P^\pi V^\pi \tag{1}$$

where $0 \leq \gamma < 1$ is a discount factor, and T^π is the Bellman operator. An optimal policy π^* is one whose associated value function dominates all others, and is defined by the following nonlinear system of equations:

$$V^*(s) = \max_a \sum_{s'} P^a_{ss'} \left(R^a_{ss'} + \gamma V^*(s')\right)$$

The *optimal action value* $Q^*(s,a)$ represents a convenient reformulation of the optimal value function, defined as the long-term value of performing a first, and then acting optimally according to V^*:

$$Q^*(s,a) = E\left(r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a\right) \quad (2)$$

where r_{t+1} is the actual reward received at the next time step, and s_{t+1} is the state resulting from executing action a in state s_t. When the set of states, S, is large, it is often necessary to approximate the value function, V, using a set of handcrafted basis functions (e.g., polynomials, radial basis functions, wavelets etc.) or automatically generated basis functions (Mahadevan 2009). In linear value function approximation, the value function is assumed to lie in the linear span of the basis function matrix Φ of dimension $|S| \times p$, where it is assumed that $p \ll |S|$.[1] Hence, $V^\pi \approx \hat{V}^\pi = \Phi w$. The (optimal) action value formulation is convenient because it can be approximately solved by a temporal-difference (TD) learning technique called Q-learning (Watkins 1989). The TD(0) rule for linear function approximated value functions is given as:

$$w_{t+1} = w_t + \beta_t \left(r_t + \gamma \langle \phi(s_{t+1}), w_t \rangle - \langle \phi(s_t), w_t \rangle\right) \phi(s_t) \quad (3)$$

where the quantity in the parenthesis is the TD error. TD can be shown to converge to the fixed point of the composition of the projector Π^Φ onto the column space of Φ and the Bellman operator T^π. TD(0) converges to the value function V^π for policy π as long as the samples are "on-policy", that is following the stochastic Markov chain associated with the policy; and the learning rate β_t is decayed according to the Robbins-Monro conditions in stochastic approximation theory: $\sum_{t=0}^{\infty} \beta_t = \infty, \sum_{t=0}^{\infty} \beta_t^2 < \infty$ (Bertsekas and Tsitsiklis 1996). l_1 regularized least-squares methods for solving MDPs attempt to find a fixed point of the l_1 penalized Bellman equation (Kolter and Ng 2009; Petrik et al. 2010; Johns, Painter-Wakefield, and Parr 2010):

$$w = f(w) = \text{argmin}_u \left(\|(R^\pi + \gamma P^\pi \Phi w - \Phi u)\|^2 + \nu \|u\|_1\right) \quad (4)$$

Another way to introduce l_1 regularization is to penalize the projected Bellman residual (Geist and Scherrer 2011), which yields a convex optimization problem:

$$\begin{aligned} w_\theta &= \text{argmin}_w \|R^\pi + \gamma P^\pi \Phi \theta - \Phi w\|^2 \\ \theta^* &= \text{argmin}_\theta \|\Phi w_\theta - \Phi \theta\|_2^2 + \nu \|\theta\|_1 \end{aligned} \quad (5)$$

where the first step is the projection step and the second is the fixed point step. Recent l_1-regularized RL methods, such as LARS-TD and LCP-TD, involve matrix inversion, requiring at least quadratic complexity in the number of (active) features. In contrast, mirror-descent based RL methods (Mahadevan and Liu 2012) can provide similar performance while requiring time per step only linear in the number of features.

[1] In practice, the full Φ matrix is never constructed, and only the p-dimensional embeddings $\phi(s)$ of sampled states are explicitly formed. Also, R^π, Φ, and $P^\pi \Phi$ are approximated by $\tilde{R}$, $\tilde{\Phi}$, and $\tilde{\Phi}'$ over a given set of samples. Finally, $\langle ., . \rangle$ denotes the dot product.

Nonlinear Value Function Approximation

A key contribution of this paper is a general framework for nonlinear value function approximation based on *nonlinear separable least-squares*. This framework helps clarify previously proposed algorithms (Castro and Mannor 2010; Bertsekas and Yu 2009; Menache, Shimkin, and Mannor 2005), which amount to particular instances of this general framework. Concretely, basis adaptation can be understood as modifying a parameterized basis, $\Phi(\alpha)$ where α denotes the *nonlinear* parameters, and w denotes the *linear* parameters, such that $\hat{V} \approx \Phi(\alpha) w$. For example, for radial basis function (RBF) bases, α would correspond to the mean (center), μ_i, and covariance (width) parameters, Σ_i, of a fixed set of bases, where these would be tuned based on optimizing some particular error, such as the Bellman error (Menache, Shimkin, and Mannor 2005). Such an approach can broadly be characterized using the framework of separable nonlinear least squares framework for regression (Golub and Pereyra 1973), which provides a framework for unifying past work on basis adaptation. The general *nonlinear regression* problem is to estimate a vector of parameters θ from a set of training examples $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to minimize (for example) the squared error:

$$J(\theta) = \sum_{i=1}^{n} \|y_i - f(x_i; \theta)\|^2.$$

In the *nonlinear separable least squares setting*, the function $f(x_i; \theta) = f(x_i; \alpha, w) = [\Phi(\alpha) w]_i$, where α are the *nonlinear variables* and w denote the *linear variables*. The squared error can now be written as:

$$J(\alpha, w) \quad = \quad \|y - \Phi(\alpha) w\|^2 = \| \left(\mathbf{I} - \mathbf{\Phi}(\alpha) \mathbf{\Phi}^\dagger(\alpha)\right) \mathbf{y} \|^2$$

Golub and Pereyra define the last (boldfaced) term above as the *variable projection functional* (VP). To explain the derivation of the second step from the first, note that for a specified value of the nonlinear parameters α, for the best linear fit, the linear parameter values can computed as $\hat{w} = \Phi^\dagger(\alpha) y$ (here, $\dagger$ denotes the Moore-Penrose pseudo-inverse). Geometrically, note that

$$\Pi^\perp_{\Phi(\alpha)} = (I - \Phi(\alpha) \Phi^\dagger(\alpha))$$

is the projector onto the complement of the column space of the parameterized basis matrix $\Phi(\alpha)$. Using this notation, the nonlinear separable least squares problem can be stated formally as:

$$\alpha_{\text{opt}} = \arg\min_\alpha \|\Pi^\perp_{\Phi(\alpha)} y\|^2 \quad (6)$$

If this problem is solved to find the optimal α_{opt}, the optimal linear parameters $w_{\text{opt}} = \Phi^\dagger(\alpha_{\text{opt}}) y$. To argue that this strategy is correct, Golub and Pereyra prove the following result:

Theorem 1 *(Golub and Pereyra 1973) Assume that $\Phi(\alpha)$ has constant rank r in every open set $\Omega \subset \mathbb{R}^k$, where $\alpha \in \Omega$. If (α^*, w^*) is the global minimizer of $J(\alpha, w)$, then $\alpha_{opt} = \alpha^*$ and $w_{opt} = w^*$.*

To solve the optimization problem given by Equation 6, Golub and Pereyra derive a modified Gauss-Newton algorithm. The key step in their algorithm involves computation of the Jacobian $\nabla \|\Pi^{\perp}_{\Phi(\alpha)} y\|^2$, which involves finding the gradient of the pseudo inverse as follows:

$$\frac{1}{2}\nabla \|\Pi^{\perp}_{\Phi(\alpha)} y\|^2 = -\langle y, \Pi^{\perp}_{\Phi(\alpha)} D\Phi(\alpha)\Phi^{\dagger}(\alpha) y\rangle$$

where $D\Phi(\alpha)$ is the *Fréchet* (or tensor) derivative of the parameterized basis matrix $\Phi(\alpha)$. If there are m basis functions ϕ_i (columns of $\Phi(\alpha)$), n training points, and k nonlinear parameters α, then $D\Phi(\alpha)$ is a *tensor* of size $n \times m \times k$.

Bellman Residual VP Functional

The *Nonlinear Bellman Residual minimization* problem is to find a set of nonlinear weights $\alpha_{BR} \in \mathbb{R}^k$ and linear weights w_{BR} such that[2]

$$(\alpha_{BR}, w_{BR}) = \text{argmin}_{w,\alpha} \|T^{\pi}(\hat{V}) - \hat{V}\|^2_{\rho^{\pi}} \qquad (7)$$

where $\hat{V} = \Phi(\alpha)w \approx V^{\pi}$, and the norm $\|x\|^2_{\rho^{\pi}} \equiv \langle x, D_{\rho^{\pi}} x\rangle$ where $D_{\rho^{\pi}}$ is a diagonal matrix whose entries reflect the stationary distribution of the Markov chain corresponding to policy π. To lighten the notation below, we will assume $D_{\rho^{\pi}} = I$, although our entire derivation carries through without change in the more general case.

Defining $\Psi^{\pi}_{\alpha} = ((I - \gamma P^{\pi})\Phi(\alpha))$, we can now eliminate the linear variables w_{NBR} and write the nonlinear Bellman residual variable projection functional as follows: $\alpha_{NBR} =$

$$\begin{aligned} &= \arg\min_{\alpha} \|R^{\pi} + \gamma P^{\pi}\hat{V} - \hat{V}\|^2 \\ &= \arg\min_{\alpha} \|R^{\pi} + (\gamma P^{\pi}\Phi(\alpha) - \Phi(\alpha))(\Psi^{\pi}_{\alpha})^{\dagger} R^{\pi}\|^2 \\ &= \arg\min_{\alpha} \| \left[I - (I - \gamma P^{\pi})\Phi(\alpha)(\Psi^{\pi}_{\alpha})^{\dagger}\right] R^{\pi}\|^2 \end{aligned}$$

We can rewrite the final result above as $\alpha_{NBR} =$

$$\arg\min_{\alpha} \|\Pi^{\perp}_{\Psi^{\pi}_{\alpha}} R^{\pi}\|^2 = \arg\min_{\alpha} \|(I - \Psi^{\pi}_{\alpha}(\Psi^{\pi}_{\alpha})^{\dagger})R^{\pi}\|^2. \qquad (8)$$

Given α_{NBR}, the linear weights w_{NBR} can be written as:

$$w_{NBR} = \Psi^{\perp}_{\Phi(\alpha_{NBR})} R^{\pi} = ((I - \gamma P^{\pi})\Phi(\alpha_{NBR}))^{\dagger} R^{\pi}$$

Comparing with Equation 6, the optimization problem for nonlinear Bellman residual minimization once again involves minimization of a projector onto the complement of a matrix column space, in this case Ψ^{π}_{α}. We can state a new result extending (Golub and Pereyra 1973) to the RL setting:

Theorem 2 *Assume that Ψ^{π}_{α} has constant rank r in every open set $\Omega \subset \mathbb{R}^k$, where $\alpha \in \Omega$. If (α_{BR}, w_{BR}) is the global minimizer of Equation 7, then $\alpha_{NBR} = \alpha_{BR}$ and $w_{NBR} = w_{BR}$.*

Nonlinear LSPI

We briefly describe a nonlinear variant of least-squares policy iteration (LSPI), where on each pass through the training data $\mathcal{D} = \{s_t, a_t, r_t, s_{t+1}\}$, a nonlinear optimization problem involving the best selection of the nonlinear parameters α is computed by solving (a sampled version of) Equation 8. We used `lsqnonlin`, a nonlinear least-squares solver within MATLAB, to solve a sampled variant of Equation 8, where $\Phi(\alpha_t)$ and $P^{\pi}\Phi(\alpha_t)$ were approximated by their sampled counterparts $\hat{\Phi}(\alpha_t)$ and $\hat{\Phi}'(\alpha_t)$. Once $\hat{\alpha}_{NBR}$ was found by the nonlinear phase, we used LSPI with these settings to find the linear weights $\hat{w}_{NBR}$. Figure 1 illustrates the application of this approach to a simple 25 state chain domain (Lagoudakis and Parr 2003). The reward is at the center, so the optimal policy is to go right in the first half of the chain and go left otherwise. The Q function is approximated by a set of radial basis functions (RBFs), whose initial centers and widths are specified as shown. The figure also shows the final modified centers and widths of the RBFs using the Bellman residual variable projection functional minimization method.

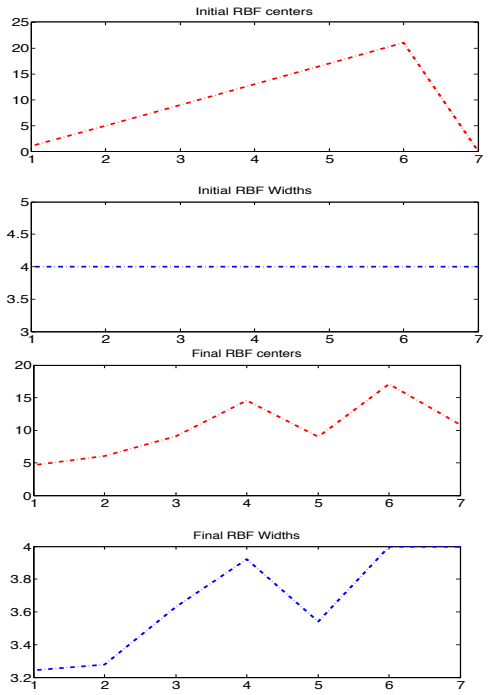

Figure 1: This figure shows the results of nonlinear basis adaptation by solving the Bellman residual variable projection functional represented by Equation 8 to find the optimal setting for the RBF centers and widths in a simple 25 state chain domain. Top two figures: initial centers and widths of RBF bases. Bottom two figures: Final centers and widths after convergence of nonlinear LSPI to the optimal policy.

Mirror Descent RL

While the above framework of nonlinear separable value function approximation using variable projection functionals is theoretically elegant, it can be computationally expensive to compute Fréchet gradients of the variable projection functional in Equation 8 in large MDPs. We describe a novel

[2] Similar VP functionals can be defined for other loss functions, e.g. for the fixed point or TD case.

mirror-descent gradient-based approach to basis adaptation that shows how to combine nonlinear value function approximation and sparsity.

Proximal Mappings and Mirror Descent

Mirror descent can be viewed as a first-order proximal gradient method (Beck and Teboulle 2003; Zinkevich 2003) for solving high-dimensional convex optimization problems. The simplest online convex algorithm is based on the classic gradient descent procedure for minimizing a convex function f, given as:

$$w_0 \in X, w_t = \Pi_X \left(w_{t-1} - \beta_t \nabla f(w_{t-1})\right) : t \geq 1 \quad (9)$$

where $\Pi_X(x) = \text{argmin}_{y \in X} \|x - y\|^2$ is the projector onto set X, and β_t is a stepsize. If f is not differentiable, then the subgradient ∂f can be substituted instead, resulting in the well-known projected subgradient method, a workhorse of nonlinear programming (Bertsekas 1999) The proximal mapping associated with a convex function h is defined as:

$$\text{prox}_h(x) = \text{argmin}_u \left(h(u) + \|u - x\|_2^2\right)$$

If $h(x) = 0$, then $\text{prox}_h(x) = x$, the identity function. If $h(x) = I_C(x)$, the indicator function for a convex set C, then $\text{prox}_{I_C}(x) = \Pi_C(x)$, the projector onto set C. For learning sparse representations, the case when $h(w) = \nu\|w\|_1$, for some scalar weighting factor ν, is particularly important. In this case, the entry-wise proximal operator is:

$$\text{prox}_h(w)_i = \begin{cases} w_i - \nu, & \text{if } w_i > \nu \\ 0, & \text{if } |w_i| \leq \nu \\ w_i + \nu & \text{otherwise} \end{cases} \quad (10)$$

An interesting observation follows from noting that the projected subgradient method (Equation 9) can be written equivalently using the proximal mapping as:

$$w_{t+1} = \text{argmin}_{w \in X} \left(\langle w, \partial f(w_t)\rangle + \frac{1}{2\beta_t}\|w - w_t\|_2^2\right) \quad (11)$$

An intuitive way to understand this equation is to view the first term as requiring the next iterate w_{t+1} to move in the direction of the (sub) gradient of f at w_t, whereas the second term requires that the next iterate w_{t+1} not move too far away from the current iterate w_t. Note that the (sub)gradient descent is a special case of Equation (11) with Euclidean distance metric. With this introduction, we can now introduce the main concept of *mirror descent* (Nemirovksi and Yudin 1983). We follow the treatment in (Beck and Teboulle 2003) in presenting the mirror descent algorithm as a nonlinear proximal method based on a distance generating function that is a Bregman divergence (Bregman 1967).

Definition 1: A distance generating function $\psi(x)$ is defined as a continuously differentiable strongly convex function (with modulus σ) which satisfies:

$$\langle x' - x, \nabla\psi(x') - \nabla\psi(x)\rangle \geq \sigma\|x' - x\|^2 \quad (12)$$

Given such a function ψ, the Bregman divergence associated with it is defined as:

$$D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y\rangle \quad (13)$$

Intuitively, the Bregman divergence measures the difference between the value of a strongly convex function $\psi(x)$ and the estimate derived from the first-order Taylor series expansion at $\psi(y)$. Many widely used distance measures turn out to be special cases of Bregman divergences, such as Euclidean distance (where $\psi(x) = \frac{1}{2}\|x\|_2^2$) and Kullback Liebler divergence (where $\psi(x) = \sum_i x_i \log_2 x_i$, the negative entropy function). In general, Bregman divergences are non-symmetric, but projections onto a convex set with respect to a Bregman divergence are well-defined.

The general mirror descent procedure can be written as:

$$w_{t+1} = \text{argmin}_{w \in X} \left(\langle w, \partial f(w_t)\rangle + \frac{1}{\alpha_t} D_\psi(w, w_t)\right) \quad (14)$$

Notice that the squared distance term in Equation 11 has been generalized to a Bregman divergence. The solution to this optimization problem can be stated succinctly as the following generalized gradient descent algorithm, which forms the core procedure in mirror descent:

$$w_{t+1} = \nabla\psi^* \left(\nabla\psi(w_t) - \alpha_t \partial f(w_t)\right) \quad (15)$$

Here, ψ^* is the Legendre transform of the strongly convex function ψ, which is defined as

$$\psi^*(y) = \sup_{x \in X} \left(\langle x, y\rangle - \psi(x)\right)$$

It can be shown that $\nabla\psi^* = (\nabla\psi)^{-1}$ (Beck and Teboulle 2003). Mirror descent is a powerful first-order optimization method that been shown to be "universal" in that if a problem is online learnable, it leads to a low-regret solution using mirror descent (Srebro, Sridharan, and Tewari 2011). It is shown in (Ben-Tal, Margalit, and Nemirovski 2001) that the mirror descent procedure specified in Equation 15 with the Bregman divergence defined by the *p-norm* function (Gentile 2003), defined below, can outperform the projected subgradient method by a factor $\frac{n}{\log n}$ where n is the dimensionality of the space. For high-dimensional spaces, this ratio can be quite large.

Sparse Learning with Mirror Descent TD

Recently, a new framework for sparse regularized RL was proposed based on mirror descent (Mahadevan and Liu 2012). Unlike regular TD, the weights are updated using the TD error in the dual space by mapping the primal weights w using a gradient of a strongly convex function ψ. Subsequently, the updated dual weights are converted back into the primal space using the gradient of the Legendre transform of ψ, namely $\nabla\psi^*$. To achieve sparsity, the dual weights θ are truncated according to Equation 10 to satisfy the l_1 penalty on the weights. Here, ν is the sparsity parameter. An analogous approach for l_1 penalized classification and regression was suggested in (Shalev-Shwartz and Tewari 2011).

One choice for Bregman divergence is the **p-norm** function, where $\psi(w) = \frac{1}{2}\|w\|_q^2$, and its conjugate Legendre transform $\psi^*(w) = \frac{1}{2}\|w\|_p^2$. Here, $\|w\|_q = \left(\sum_j |w_j|^q\right)^{\frac{1}{q}}$, and p and q are conjugate numbers such that $\frac{1}{p} + \frac{1}{q} = 1$. This

$\psi(w)$ leads to the p-norm link function $\theta = f(w)$ where $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ (Gentile 2003):

$$f_j(w) = \frac{\text{sign}(w_j)|w_j|^{q-1}}{\|w\|_q^{q-2}}, \;\; f_j^{-1}(\theta) = \frac{\text{sign}(\theta_j)|\theta_j|^{p-1}}{\|\theta\|_p^{p-2}} \tag{16}$$

Many other choices for Bregman divergence are possible.

Basis Adaptation with Mirror Descent RL

Like the approach proposed for adaptive bases for Q-learning (Castro and Mannor 2010), our method is based on the two-time scale stochastic approximation framework, whereby the linear parameters w are adapted at a faster time-scale than the nonlinear parameters α. Algorithm 1 below describes the nonlinear adaptive basis mirror-descent variant of Watkins $Q(\lambda)$ algorithm.[3] We indicate the dynamically varying nature of the bases as $\phi_t(s_t, a_t)$ where the subscript denotes the particular value of the nonlinear parameters α_t at time t. In this section, we denote β_t as the learning rate for the faster time-scale update procedure for updating the linear weights w_t and ξ_t as the learning rate for the slower time-scale parameter for updating the nonlinear basis parameters α_t. Following the approach given in (Borkar 2008; Castro and Mannor 2010), it can be shown that Algorithm 1 converges broadly under similar assumptions. The key idea underlying Algorithm 1 is to adapt the nonlinear parameters using the gradient of the TD error. As the gradient of the TD error is not easily computed due to the `max` computation in $Q(\lambda)$, what is done here is to approximate the maximum using a smoothed differentiable approximation (Castro and Mannor 2010):

$$\max_{i=1,\ldots,n} x_i \approx f_\sigma(\{x_i\}_{i=1,\ldots,n}) = \log(\sum_{i=1}^{n} e^{\sigma x_i})$$

We update the nonlinear basis parameters α_t on a slower time scale using the gradient of the smoothed TD error. The mirror-descent version of Watkins $Q(\lambda)$ with basis adaptation is given below.

Algorithm 1 Mirror Descent Q(λ) with Basis Adaptation

1: **Given:** Parametric basis $\Phi(\alpha)$
2: Initialize linear parameters w_0 and nonlinear parameters α_0.
3: **repeat**
4: Do action $a_t = \text{argmax}_{a^*}\langle\phi_t(s_t, a^*), w_t\rangle$ with high probability, else do a random action. Observe next state s_{t+1} and reward r_t.
5: Compute the actual TD error

$$\delta_t = r_t + \gamma \max_a \langle\phi_t(s_{t+1}, a), w_t\rangle - \langle\phi_t(s_t, a_t), w_t\rangle$$

6: Compute the smoothed TD error

$$\omega_t = r_t + \gamma f_\sigma(\{\langle\phi_t(s_{t+1}, a'), w_t\rangle\}_{a'}) - \langle\phi_t(s_t, a_t), w_t\rangle$$

7: Compute the gradient of the smoothed TD error w.r.t.. α

$$K_\sigma(s_t, a_t, s_{t+1}) = \gamma \sum_{a'} \frac{\partial f_\sigma(\{\langle\phi_t(s_{t+1}, a'), w_t\rangle\}_{a'})}{\partial(\langle\phi_t(s_{t+1}, a'), w_t\rangle)}$$

$$\times \nabla_\alpha(\langle\phi_t(s_{t+1}, a'), w_t\rangle) - \nabla_\alpha(\langle\phi_t(s_{t+1}, a_t), w_t\rangle)$$

8: Update the eligibility trace $e_t \leftarrow e_t + \lambda\gamma\phi_t(s_t, a_t)$
9: Update the dual weights $\theta_t = \nabla\psi_t(w_t) + \beta_t\delta_t e_t$ (e.g., $\psi(w) = \frac{1}{2}\|w\|_q^2$ is the p-norm link function).
10: Truncate weights:

$$\forall j, \;\; \theta_j^{t+1} = \text{sign}(\tilde{\theta}_j^{t+1}) \max(0, |\tilde{\theta}_j^{t+1}| - \beta_t\nu)$$

11: Update linear weight parameters: $w_{t+1} = \nabla\psi_t^*(\theta_{t+1})$ (e.g., $\psi^*(\theta) = \frac{1}{2}\|\theta\|_p^2$ and p and q are dual norms such that $\frac{1}{p} + \frac{1}{q} = 1$).
12: Update nonlinear basis parameters:

$$\alpha_{t+1} \leftarrow \alpha_t + \xi_t\omega_t K_\sigma$$

13: Set $t \leftarrow t + 1$.
14: **until done**.
Return $\hat{Q}(s, a) = (\Phi(\alpha_t)w_t)_{s,a}$ as the approximate l_1 penalized sparse optimal action value function.

Experimental Results

In this section, we provide some illustrative experiments evaluating the performance of the mirror-descent $Q(\lambda)$ method with a nonlinear generalization of the fixed Fourier basis (Konidaris, Osentoski, and Thomas 2011). The nonlinear Fourier basis that we used is as follows:

$$\phi_i(\mathbf{x}) = \cos(\pi\alpha_i\langle c_i, \mathbf{x}\rangle)$$

The nonlinear parameters here include both α_i and c_i. Note that in the previous work on Fourier bases (Konidaris, Osentoski, and Thomas 2011), the scaling factor α_i was fixed to unity, whereas in our case, it is allowed to vary (as shown in Figure 4). Also, earlier work set $c^i = [c_1, \ldots, c_d], c_j \in [0, \ldots, n], 1 \leq j \leq d$. For example, in the mountain car domain, $d = 2$, and setting $n = 3$ would result in 16 possible features (since $n = 0, 1, 2, 3$). The nonlinear parameter α_i can be seen here as a weighting of how important each c_i is. In our initial experiments, we vary both α and c_i separately to compare their effects.

Figure 2 compares the convergence and stability of $Q(\lambda)$ and mirror-descent $Q(\lambda)$ using 49 fixed Fourier bases with mirror-descent $Q(\lambda)$ using the same number of adaptive (α and C_i parameter tuned) nonlinear Fourier bases. In this case, all methods converge, but the two nonlinear mirror-descent methods (bottom curves) converge much more quickly. The mirror-descent methods have converged by 7 episodes to an average solution length at least thrice as small as the regular $Q(\lambda)$ method, which takes much longer to reliably achieve the same solution quality. For the experimental result shown in Figure 2, where we show convergence in the mountain car domain with 49 Fourier bases, we set $\beta_t = 0.03$, $\xi_t = 5 \times 10^{-7}$, and $\nu = 0.001$. The **p-norm** Bregman divergence was used, where $p =$

[3] The extension to related methods like $SARSA(\lambda)$ (Sutton and Barto 1998) is straightforward. Some details abbreviated for clarity.

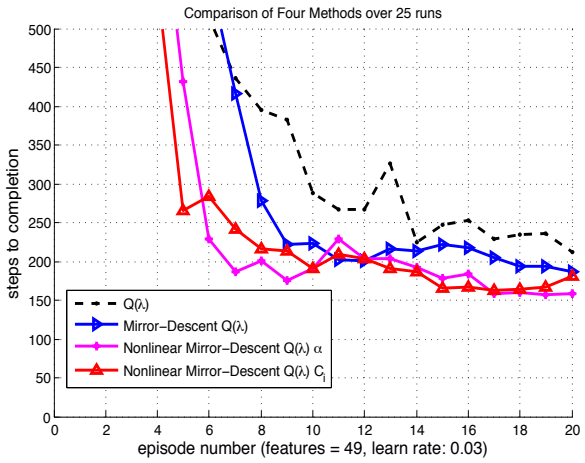

Figure 2: Comparing the convergence of two mirror-descent $Q(\lambda)$ methods over 25 runs with 49 (α and c_i parameter tuned) nonlinear Fourier bases in the mountain car task with that of $Q(\lambda)$, and mirror-descent $Q(\lambda)$, with 49 fixed Fourier bases.

$2\log_{10}(\text{num\_features x num\_actions})$, and $q = \frac{p}{p-1}$. The learning rate for $Q(\lambda)$ had to be set fairly low at 0.001 when the number of features exceeded 16 as higher rates caused instability.

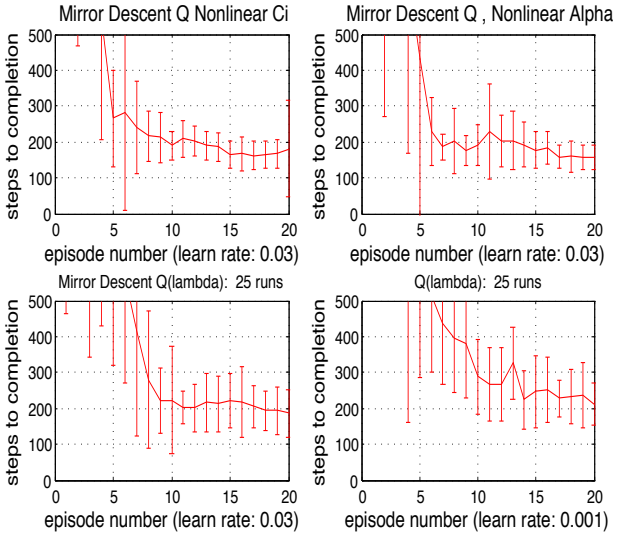

Figure 3: Variance of the four methods compared in Figure 2 across 25 runs. Note the improved variance of the two methods shown on top that combine mirror-descent updating and basis adaptation.

Figure 3 shows the variance across 25 runs of the four methods compared in Figure 2. Note the improved variance of the two mirror-descent methods that use basis adaptation, showing they not only result in improved performance, but also lower variance. Finally, Figure 4 shows the ranges of the α parameters as tuned by the basis adaptation algorithm for 49 Fourier bases over 25 runs.

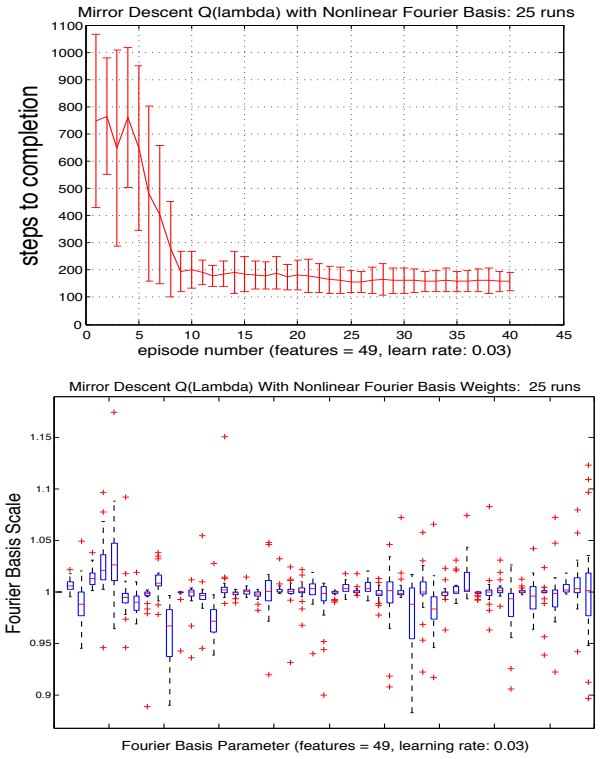

Figure 4: (a) Extended run of the mirror-descent $Q(\lambda)$ method with basis adaptation of the α parameters in the mountain car domain. (b) Boxplot of the α parameters for 49 Fourier bases for the run shown on top.

Conclusions and Future Work

This paper introduced a broad framework for basis adaptation as nonlinear separable least-squares value approximation, and described a specific gradient-based method for basis adaptation using mirror descent optimization. Currently, the learning rate for nonlinear parameters needs to be set quite low for stable convergence. More rapid tuning of the nonlinear parameters could be achieved by including an additional mirror descent step for modifying them.

Acknowledgments

This research is supported in part by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-10-1-0383, and the National Science Foundation under Grant Nos. NSF CCF-1025120, IIS-0534999, IIS-0803288, and IIS-1216467 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the AFOSR or the NSF.

References

Beck, A., and Teboulle, M. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*.

Ben-Tal, A.; Margalit, T.; and Nemirovski, A. 2001. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal of Optimization*.

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Belmont, Massachusetts: Athena Scientific.

Bertsekas, D., and Yu, H. 2009. Basis function adaptation methods for cost approximation in Markov Decision Processes. In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning*.

Bertsekas, D. P. 1999. *Nonlinear Programming*. Athena Scientific, 2nd edition.

Borkar, V. 2008. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Bregman, L. 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* 7(3):200 – 217.

Castro, D. D., and Mannor, S. 2010. Adaptive bases for Q-learning. In *49th IEEE Conference on Decision and Control*.

Geist, M., and Scherrer, B. 2011. L1-penalized projected Bellman residual. In *Proceedings of the European Workshop on Reinforcement Learning*.

Gentile, C. 2003. The robustness of the p-norm algorithms. *Mach. Learn.* 53:265–299.

Golub, G., and Pereyra, V. 1973. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal of Numerical Analysis* 10:413–32.

Johns, J.; Painter-Wakefield, C.; and Parr, R. 2010. Linear complementarity for regularized policy evaluation and improvement. *In Proceedings of Advances in Neural Information Processing Systems* 23.

Kolter, J. Z., and Ng, A. Y. 2009. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 521–528. New York, NY, USA: ACM.

Konidaris, G.; Osentoski, S.; and Thomas, P. 2011. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*.

Lagoudakis, M., and Parr, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4:1107–1149.

Liu, B.; Mahadevan, S.; and Liu, J. 2012. Regularized off-policy TD-learning. In *Neural Information Processing Systems Conference (NIPS)*.

Mahadevan, S., and Liu, B. 2012. Reinforcement learning using mirror descent. In *UAI 2012: Proceedings of the conference on Uncertainty in AI*.

Mahadevan, S. 2008. *Representation Discovery using Harmonic Analysis*. Morgan and Claypool Publishers.

Mahadevan, S. 2009. Learning Representation and Control in Markov Decision Processes: New Frontiers. *Foundations and Trends in Machine Learning* 1(4):403–565.

Menache, N.; Shimkin, N.; and Mannor, S. 2005. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research* 134:215–238.

Nemirovksi, A., and Yudin, D. 1983. *Problem Complexity and Method Efficiency in Optimization*. John Wiley 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*, 871–878.

Shalev-Shwartz, S., and Tewari, A. 2011. Stochastic methods for l1 regularized loss minimization. *Journal of Machine Learning Research*.

Srebro, N.; Sridharan, K.; and Tewari, A. 2011. On the universality of online mirror descent. *Arxiv preprint arXiv:1107.4080*.

Sutton, R., and Barto, A. G. 1998. *An Introduction to Reinforcement Learning*. MIT Press.

Watkins, C. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, King's College, Cambridge, England.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the International Conference on Machine Learning (ICML)*.