# Analyzing Feature Generation for Value-Function Approximation

**Ronald Parr** PARR@CS.DUKE.EDU
**Christopher Painter-Wakefield** PAINT007@CS.DUKE.EDU
Department of Computer Science, Duke University, Durham, NC 27708 USA

**Lihong Li** LIHONG@CS.RUTGERS.EDU
**Michael Littman** MLITTMAN@CS.RUTGERS.EDU
Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA

## Abstract

We analyze a simple, Bellman-error-based approach to generating basis functions for value-function approximation. We show that it generates orthogonal basis functions that provably tighten approximation error bounds. We also illustrate the use of this approach in the presence of noise on some sample problems.

## 1. Introduction

In many areas of machine learning, the automatic discovery of features remains an important challenge. In the area of value-function approximation for reinforcement learning or the approximate solution of Markov decision processes (MDPs), which have reaped few of the benefits of techniques such as boosting (Freund & Schapire, 1995) or the kernel trick (Vapnik et al., 1997), the challenge of feature selection or discovery remains particularly acute.

Recent efforts in feature discovery for value-function approximation have been along two main lines. The first is based upon graph structures built up from observed trajectories through the state space (Mahadevan & Maggioni, 2006). The second is based upon the Bellman error (Menache et al., 2005; Keller et al., 2006). In this paper, we consider approaches of the second type in the context of linear value-function approximation. Specifically, we consider a general family of approaches that iteratively add basis functions to a linear approximation architecture in a manner where each new basis function is derived from the Bellman error of the previous set of basis functions. We call these Bellman Error Basis Functions (BEBFs).

Our main theoretical contribution is to show that BEBFs form an orthonormal basis with guaranteed improvement in approximation quality at each iteration. Since the Bellman error can be a quite complicated function that may not be any easier to represent than the true value function, we consider the use of a Bellman error approximator to represent the new basis function. A similar approach was taken by Keller et al. (2006), who used a form of state aggregation or clustering to estimate the Bellman error. We prove a general result showing that the approximation quality can still improve even if there is significant error in the estimate of the Bellman error.

We distinguish between two applications of the general BEBF scheme. One is exact, in which all computations are made with respect to precise representations of the underlying MDP or Markov chain model. In this setting, we show that BEBFs result in steadily improving bounds on the distance from the optimal value function, a significant open problem in previous work. The second is approximate in which Markov dynamics are experienced only via samples and the functions are represented using an approximation scheme. In this setting, we prove some conservative conditions that suffice to ensure that new BEBFs that approximate the Bellman error will improve the linear value function. Finally, we provide experiments demonstrating the use of approximate BEBFs in policy iteration.

## 2. Formal Framework and Notation

We are concerned with controlled and uncontrolled Markov processes consisting of a set of states $s_1 \ldots s_n$. Actions, when applicable, are chosen from the set $a_1 \ldots a_m$; however, much of our theory applies only to uncontrolled systems—essentially ones with a single action ($m = 1$).

Given a state $s_i$, the probability of a transition to a state $s_j$ given action $a$ is given by $P^a_{ij}$ and results in an expected reward of $R^a_i$. In the uncontrolled case, we use $P$ and $R$ to stand for the transitions and rewards.

Appearing in *Proceedings of the $24^{th}$ International Conference on Machine Learning*, Corvallis, OR, 2007. 

We are interested in finding value functions $V$ that map each state $s_i$ to the expected total $\gamma$-discounted reward for the process. In particular, we would like the solution to the Bellman equation

$$V[s_i] = \max_a (R_i^a + \gamma \sum_j P_{ij}^a V[s_j])$$

in the controlled case (the "max" is eliminated from the equation in the uncontrolled case).

To simplify notation for manipulating value functions, we define the Bellman operator $T^*$ on value functions as

$$(T^*V)[s_i] = \max_a (R_i^a + \gamma \sum_j P_{ij}^a V[s_j]).$$

In this notation, the Bellman equation becomes $V^* = T^*V^*$. Of particular interest in this paper is the Bellman operator in the uncontrolled case, $T$, which is, again, simply the $T^*$ operator without the "max":

$$(TV)[s_i] = R_i + \gamma \sum_j P_{ij} V[s_j].$$

The Bellman operator is well known to be a contraction in maximum norm:

$$\|V_1 - V_2\|_\infty = \epsilon \Rightarrow \|TV_1 - TV_2\|_\infty \leq \gamma\epsilon.$$

A less known property of the Bellman operator is that it is a contraction in the weighted $L_2$ norm:

$$\|V\|_\rho = \sqrt{\sum_{i=1}^n V[s_i]^2 \rho[s_i]},$$

where $\rho$ is the stationary distribution of $P$: $\rho = P^T\rho$. As noted by Van Roy (1998),

$$\|V_1 - V_2\|_\rho = \epsilon \Rightarrow \|TV_1 - TV_2\|_\rho \leq \gamma\epsilon.$$

Unless otherwise indicated, we will use $\|\cdot\|$ for $\|\cdot\|_\rho$.

Note that defining $V^0 = R$, then $V^{t+1} = TV^t$, results in the value-iteration algorithm. Based on the results above, $V^t \rightarrow V^*$ as $t$ increases.

The *Bellman error* of a value function $V$ is the difference between the value function and the result of applying the Bellman operator: $TV - V$.

In cases where the value function cannot be represented exactly, it is common to use some form of parametric value-function approximation, such as a linear combination of features or basis functions:

$$\hat{V} = \sum_{i=1}^k w_i \phi_i,$$

where $\Phi = \{\phi_1 \ldots \phi_k\}$ is a set of linearly independent basis functions of the state, and $\mathbf{w} = \{w_1 \ldots w_k\}$ is a set of scalar weights. We can think of $\Phi$ as a design matrix with $\Phi[i,j] = \phi_j(s_i)$, that is, the basis functions span the columns of $\Phi$ and the states span the rows. For a set of weights $\mathbf{w}$ expressed as a column vector, $\hat{V} = \Phi\mathbf{w}$.

Methods for finding reasonable $\mathbf{w}$ given $\Phi$ and a set of samples include linear TD (Sutton, 1988), LSTD (Bradtke & Barto, 1996) and LSPE (Yu & Bertsekas, 2006). If the model can be expressed as a factored MDP, then the weights can be found directly (Koller & Parr, 1999). We refer to this family of methods as *linear fixed point* methods because they all solve for the same fixed point:

$$\hat{V} = \Phi\mathbf{w} = \Pi_\rho(R + \gamma P \Phi\mathbf{w}), \quad (1)$$

where $\Pi_\rho$ is an operator that is the $\rho$-weighted $L_2$ projection into the span of $\Phi$, that is, if $\Delta = diag(\rho)$,

$$\Pi_\rho = \Phi(\Phi^T\Delta\Phi)^{-1}\Phi^T\Delta.$$

We use $\Pi$ as shorthand for $\Pi_\rho$ unless otherwise indicated. The closest point (in $\|\cdot\|_\rho$) in the span of $\Phi$ to $V^*$ is $\Pi V^*$, but linear fixed point methods are not guaranteed to find this point. However, the distance from $\hat{V}$ to $V^*$ is bounded in terms of the distance from $\Pi V^*$ to $V^*$ (Van Roy, 1998):

$$\|V^* - \hat{V}\| \leq \frac{1}{\sqrt{1-\kappa^2}} \|V^* - \Pi V^*\|. \quad (2)$$

The effective contraction rate $\kappa$ arises from the combination of the Bellman operator, $T$, with contraction rate $\gamma$, and the $L_2$ projection, which is non-expansive, and could possibly introduce some additional contraction. For this paper, we conservatively assume $\kappa = \gamma$.

## 3. Basis Expansion

Basis expansion in the context of linear fixed point methods addresses the following question: Given a set of basis functions $\phi_1 \ldots \phi_k$ and a linear fixed point solution $\hat{V}$, what is a good $\phi_{k+1}$ to add to the basis? This question is asked both in the context of graph-based approaches (Mahadevan & Maggioni, 2006) and Bellman-error-based methods (Keller et al., 2006).

The Bellman error is an intuitively appealing approach to expanding the basis since it is, loosely speaking, pointing towards $V^*$. We say that $\phi_{k+1}$ is a Bellman Error Basis Function (BEBF) for $\hat{V} = \Phi\mathbf{w}$ if $\phi_{k+1} = T\hat{V} - \hat{V}$. Constructing $\Phi' = [\Phi, \phi_{k+1}]$ (concatenating column vector $\phi_{k+1}$ to design matrix $\Phi$) ensures that $T\hat{V}$ is in the span of $\Phi'$ (trivially by picking new weights $w'_i = w_i$ for $1 \leq i \leq k$, and $w'_{k+1} = 1$). While this formulation ensures that $\hat{V}$ can be represented, it leaves many open questions such as:

- How does increasing the expressive power to include $T\hat{V}$ affect the performance bound for the fixed point error bound in Eq. (2)? Adding $T\hat{V}$ to the space of representable value functions doesn't necessarily mean that a linear fixed point method will choose $T\hat{V}$. Even if a linear fixed point method did pick $T\hat{V}$, it might be no closer to $V^*$ than $\Pi V^*$ is.

- How does performance degrade if $\widehat{\phi_{k+1}} \approx \phi_{k+1}$ is used instead? This question is important because it typically will be difficult to represent $\phi_{k+1}$ exactly for large problems.

The subsequent two subsections address these questions.

### 3.1. Exact BEBFs

In this subsection, our analysis continues from the perspective of the $\rho$-weighted $L_2$ norm. The inner product between two vectors $V_1$ and $V_2$ is therefore defined as:

$$(V_1 \cdot V_2)_\rho = \sum_{i=1}^{n} V_1[s_i]V_2[s_i]\rho[s_i].$$

Two vectors are orthogonal if $(V_1 \cdot V_2)_\rho = 0$. Many properties of the usual, unweighted, $L_2$ norm remain true in a weighted $L_2$ norm. For example, the Pythagorean theorem remains true, which means that if $(A \cdot B)_\rho = 0$, then

$$\|A + B\|_\rho^2 = \|A\|_\rho^2 + \|B\|_\rho^2.$$

In the sequel, when we indicate that two vectors are orthogonal, the $\rho$-weighted $L_2$ norm will be implicit. When we say that a vector is normalized, we mean that the entries have been divided by a suitable constant to ensure that the $\rho$-weighted $L_2$ norm is 1.

**Lemma 3.1** *If $\hat{V}$ is a linear fixed point solution using the basis $\Phi = \{\phi_1 \dots \phi_k\}$, then the BEBF $\phi' = T\hat{V} - \hat{V}$ is orthogonal to the span of $\Phi$.*

**Proof:** This result follows immediately from the definition of the fixed point in Eq. (1), since $\hat{V}$ is, by definition, the orthogonal projection of $T\hat{V}$ into $\Phi$. ∎

A sequence of BEBFs is a set of BEBFs $\phi_1 \dots \phi_k$, generated with an arbitrary, but nonzero, $\phi_1$, and $\phi_i$ for $i > 1$ as the BEBF for the basis $\phi_1 \dots \phi_{i-1}$.

**Corollary 3.2** *A sequence of normalized BEBFs $\phi_1 \dots \phi_k$ forms an orthonormal basis.*

**Corollary 3.3** *For a system with $n$ states, $V^*$ can be represented exactly using a sequence of no more than $n$ BEBFs.*

Without loss of generality, for normalized sequence of BEBFs $\phi_1 \dots \phi_n$,

$$V^* = \sum_{i=1}^{n} w_i\phi_i$$

for some $w_1 \dots w_n$. This representation of $V^*$ is critical to our analysis, so we describe its significance in more detail: For the purposes of analysis, we will assume a representation of $V^*$ *as if* the BEBF procedure had been run to completion. We are *not* assuming that these functions are produced in practice. Rather, we are adopting this representation to analyze the effect of adding an additional BEBF to an existing sequence of $k$ BEBFs for $k < n$.

Informally, our main result states that if the Bellman operator moves $\hat{V}$ closer to $V^*$ by some amount $x$, and we use this Bellman error to generate a new BEBF, then the closest function in the new space to $V^*$ (the term in the right hand side of Eq. (2)) must improve by at least $x$. Alternatively, we can say that the bound is tightening at least as quickly as it does in value iteration.

**Theorem 3.4** *Let $\hat{V}$ be the linear fixed point solution using a sequence of normalized BEBFs $\phi_1 \dots \phi_k$. If $\|V^* - \hat{V}\| - \|V^* - T\hat{V}\| = x$, then for new BEBF $\phi_{k+1}$, with $\Phi' = [\Phi, \phi_{k+1}]$, and corresponding $\Pi'$, the improvement in the approximation bound is $\|V^* - \Pi V^*\| - \|V^* - \Pi' V^*\| \geq x$.*

**Proof:** The contraction property of $T$ guarantees that $x > 0$ for $\hat{V} \neq V^*$. By the orthonormality of the basis,

$$\Pi V^* = \sum_{i=1}^{k} w_i\phi_i.$$

For $\hat{V} = \sum_{i=1}^{k} \alpha_i\phi_i$,

$$V^* - \hat{V} = \sum_{i=1}^{k}(w_i - \alpha_i)\phi_i + \sum_{i=k+1}^{n} w_i\phi_i,$$

where the first term is the error introduced by choosing other than the closest point in the span of $\Phi$, and the second term is the residual left because the basis may not be able to express $V^*$ exactly. Since $\Phi$ is an orthonormal basis, by the Pythagorean theorem:

$$\|V^* - \hat{V}\|^2 = \sum_{i=1}^{k}(w_i - \alpha_i)^2 + \sum_{i=k+1}^{n} w_i^2,$$

and since $\phi_{k+1}$ is orthogonal to the span of $\Phi$,

$$\begin{aligned} &\|V^* - T\hat{V}\|^2 \\ &= \|V^* - (\hat{V} + \beta\phi_{k+1})\|^2 \\ &= (w_{k+1} - \beta)^2 + \sum_{i=1}^{k}(w_i - \alpha_i)^2 + \sum_{i=k+2}^{n} {w_i}^2, \end{aligned}$$

where $\beta$ is the normalizing constant used to normalize the Bellman error in producing $\phi_{k+1}$.

Since the only difference between these expressions is the $(w_{k+1} - \beta)^2$ term, the reduction in squared distance is attributed entirely to the fact that $(w_{k+1} - \beta)^2 < w_{k+1}^2$:

$$\|V^* - \hat{V}\|^2 - \|V^* - T\hat{V}\|^2 = w_{k+1}^2 - (w_{k+1} - \beta)^2.$$

Reintroducing $x$ into the expression:

$$\begin{aligned}
&\|V^* - \hat{V}\|^2 - \|V^* - T\hat{V}\|^2 \\
&= (\|V^* - \hat{V}\| - \|V^* - T\hat{V}\|)(\|V^* - \hat{V}\| + \|V^* - T\hat{V}\|) \\
&= x(\|V^* - \hat{V}\| + \|V^* - T\hat{V}\|).
\end{aligned}$$

Solving for $x$:

$$\begin{aligned}
x &= \frac{\|V^* - \hat{V}\|^2 - \|V^* - T\hat{V}\|^2}{\|V^* - \hat{V}\| + \|V^* - T\hat{V}\|} \\
&= \frac{w_{k+1}^2 - (w_{k+1} - \beta)^2}{\|V^* - \hat{V}\| + \|V^* - T\hat{V}\|}. \qquad (3)
\end{aligned}$$

Now, consider the difference between $\|V^* - \Pi V^*\|^2 = \sum_{i=k+1}^{n} w_i^2$, and $\|V^* - \Pi' V^*\|^2 = \sum_{i=k+2}^{n} w_i^2$:

$$\|V^* - \Pi V^*\|^2 - \|V^* - \Pi' V^*\|^2 = w_{k+1}^2,$$

which implies

$$\begin{aligned}
&\|V^* - \Pi V^*\| - \|V^* - \Pi' V^*\| \\
&= \frac{w_{k+1}^2}{\|V^* - \Pi V^*\| + \|V^* - \Pi' V^*\|}. \qquad (4)
\end{aligned}$$

Finally, observe that the numerator in (4) is greater than or equal to the numerator in (3), and that the denominator in (4) is less than or equal to the denominator in (3). ∎

### 3.2. Approximation

In the previous subsection, we assumed that it was possible to represent the Bellman error exactly over the entire state space. As with many of the early efforts to discover basis functions (Mahadevan & Maggioni, 2006), this exact representation could be as difficult to find as the value function itself. In practice, we may be forced to use an approximate representation, as in, for example, the work of Keller et al. (2006). For $\widehat{\phi_{k+1}} \approx \phi_{k+1}$, we can state some qualitative results. The first is that expanding the basis in the general direction of $V^*$ ensures progress:

**Lemma 3.5** *If $\widehat{\phi_{k+1}}$ is not orthogonal to $V^* - \hat{V}$, then there exists a positive $\beta$ such that $\|V^* - (\hat{V} + \beta\widehat{\phi_{k+1}})\| < \|V^* - \hat{V}\|$. Moreover, if $\widehat{\phi_{k+1}}$ is not in the span of $\Phi$, then for $\Phi' = \Phi \cup \widehat{\phi_{k+1}}$, and corresponding $\Pi'$, $\|V^* - \Pi' V^*\| < \|V^* - \Pi V^*\|$.*

**Proof:** Assume

$$\beta = 0 = \arg\min_\sigma \|V^* - (\hat{V} + \sigma\widehat{\phi_{k+1}})\|.$$

By definition, $\hat{V}$ is then the orthogonal projection of $V^*$ onto the line $\hat{V} + \sigma\widehat{\phi_{k+1}}$, and $V^* - \hat{V}$ is therefore orthogonal to $\widehat{\phi_{k+1}}$. Thus, unless $\widehat{\phi_{k+1}}$ is orthogonal to $V^* - \hat{V}$, there exists a $\beta$ such that $\hat{V} + \beta\widehat{\phi_{k+1}}$ is closer to $V^*$ than $\hat{V}$ is.

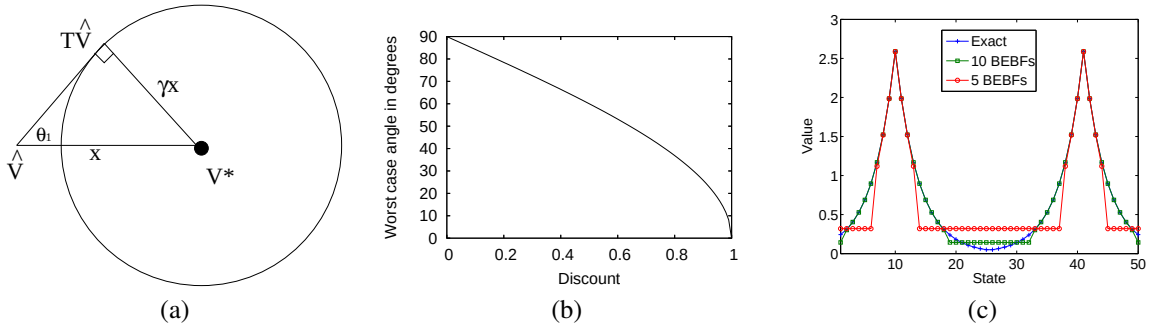
We only sketch the result for $\|V^* - \Pi' V^*\| < \|V^* - \Pi V^*\|$ because the analysis is very similar to Theorem 3.4. $\widehat{\phi_{k+1}}$ can be expressed as weighted sum of $\phi_1 \ldots \phi_n$. The $\phi_1 \ldots \phi_k$ components can be ignored because they are already in the span of $\Phi$ and they do not affect the analysis. It is then easy to show that any reduction in the squared distance from $\hat{V}$ to $V^*$ can be mirrored with an equivalent reduction in the squared distance from $\Pi V^*$. ∎

This lemma is encouraging, but the ease or difficulty in obtaining a $\widehat{\phi_{k+1}}$ that points towards $V^*$ may not be obvious since the true direction of $V^*$ typically isn't known until the problem is solved exactly. The angle between $\widehat{\phi_{k+1}}$ and $\phi_{k+1}$ provides a weaker, sufficient (though not necessary) condition for ensuring progress:

**Theorem 3.6** *If (1) the angle between $\phi^{k+1}$ and $\widehat{\phi_{k+1}}$ is less than $\cos^{-1}(\gamma)$ radians and (2) $\hat{V} \neq V^*$, then there exists a $\beta$ such that $\|V^* - (\hat{V} + \beta\widehat{\phi_{k+1}})\| < \|V^* - \hat{V}\|$. Moreover, if conditions (1) and (2) hold and $\widehat{\phi_{k+1}}$ is not in the span of $\Phi$, then for $\Phi' = \Phi \cup \widehat{\phi_{k+1}}$, and corresponding $\Pi'$, $\|V^* - \Pi' V^*\| < \|V^* - \Pi V^*\|$.*

**Proof:** First, we identify the maximum possible angle, $\theta_1$, between $T\hat{V} - \hat{V}$ and $V^* - \hat{V}$. In the worst case, adding an additional $\theta_2 = \pi/2 - \theta_1$ radians suffices to make $\widehat{\phi_{k+1}}$ orthogonal to $V^* - \hat{V}$. If $V^* - \hat{V} = x$, then $T\hat{V}$ can lie on a circle of radius $\gamma x$ from $V^*$. The angle between $T\hat{V} - \hat{V}$ and $V^* - \hat{V}$ is maximized when $T\hat{V} - \hat{V}$ is tangent to the radius $\gamma x$ circle, as illustrated in Figure 1a. Since $\theta_1 = \sin^{-1}(\gamma)$, $\theta_2 < \pi/2 - \sin^{-1}(\gamma) = \cos^{-1}(\gamma)$ is sufficient to ensure that $\theta_1 + \theta_2 < \pi/2$ and that $\widehat{\phi_{k+1}}$ is not orthogonal to $V^* - \hat{V}$. The conditions of the preceding lemma are then satisfied, completing the proof. ∎

In Figure 1b, we show a graph of $\cos^{-1}(\gamma)$ in degrees vs. discount. This graph shows the smallest angular error in $\widehat{\phi_{k+1}}$ that could, in the worst case, prevent $\widehat{\phi_{k+1}}$ from improving the value function. In practice, much larger errors could be tolerated if they are not in the same direction away from $V^*$ as $T\hat{V}$ is. Qualitatively, the graph gives insight into the impact of the discount factor on BEBF approximation. In domains without noise, this result could be used to test if an approximate BEBF should be accepted by computing the dot product between the training data for the BEBF and the output of the learned BEBF.




(a) (b) (c)

Figure 1. (a) The worst case angle of $\phi_{k+1} = T\hat{V} - \hat{V}$, (b) Worst case angle $\theta_2$ vs. discount $\gamma$, (c) Value functions for the 50-state chain varying the number of exact BEBFs.

These results do not rely upon the assumption that $\phi_1 \ldots \phi_k$ are orthonormal, and generalize to the case where multiple approximate BEBFs are added, and not just a single $\widehat{\phi_{k+1}}$.

### 3.3. Comparison with Fitted Value Iteration

These BEBF results should be considered in the light of the known, negative results on the use of supervised learning for value-function approximation in the context of Fitted Value Iteration (Boyan & Moore, 1995), or FVI. At iteration $k+1$, $\hat{V}_{k+1}$ is produced by using a function approximator to fit $T\hat{V}_k$. FVI is a frustrating algorithm to use in practice due to its tendency to diverge for fairly inscrutable (to the user) reasons. The difficulty arises when errors compound. In contrast, the BEBF approach always finds a fixed point for any basis functions it produces. While this does not ensure strictly improving performance as basis functions are added, it ensures a tightening bound on the value-function error as basis functions are added.

A possible advantage of FVI is that it is more compatible with a maximum norm analysis, while the analysis of BEBF is, thus far, entirely in $\|\cdot\|_\rho$. This difference potentially makes fitted value iteration more compatible with policy-improvement approaches. A similarity between fitted value iteration and BEBF is that both methods can get "stuck" when the function approximation error for a new basis function (or new iteration of value iteration) is large enough to cancel out the contraction from $T$. We expect that BEBF will be much more robust in practice since any basis function with positive dot product with $\phi_{k+1}$ ensures a tightening of the bound while fitted value iteration can oscillate indefinitely in a region circumscribed by the worst value-function approximation error. Our expectation of greater robustness for BEBF vs. FVI was met by our experiments in Section 4.2.2.

### 3.4. Comparison with Graph-based methods

Graph-based methods have the *advantage* of being less directly connected to the reward and value function. The basis functions produced by these methods are derived from connectivity properties of the state space and could, potentially, be more generally applicable to a wide class of problems with similar state space connectivity but different reward functions. Graph-based methods have the *disadvantage* of being less directly connected to the reward and value functions. The disconnect from the reward and value function makes it difficult to guarantee a specific amount of progress with each iteration.

### 3.5. Comparison with Matching Pursuits

Matching pursuits (Mallat & Zhang, 1993) (or MP) is a family of approaches for basis selection and synthesis with a high level structure very similar to BEBF's. MP was developed in the context of time-series reconstruction and iteratively adds basis functions to a set by finding the function that best matches the residual between the target function and the reconstruction of the function using the basis functions added so far. The target function is sampled directly, and the basis functions are chosen from a restricted class of functions for generalization or compression purposes. While BEBFs can be explained in similar terms, the details and motivation are quite different. With BEBFs, state transitions are sampled from a Markov chain, and the target function is defined implicitly through the fixed-point equations. As such, taking the current residual itself as a basis does not reduce the error to zero, as it would in MP. With BEBFs, each additional basis function not only adds to the expressive power of the basis, but moves the span of the basis closer to the implicitly defined target.

## 4. Experimental Results

Our experimental results are of two types. We first consider the case where the model is available and exact BEBFs can

be computed. This case is not very realistic because the effort to compute and represent a BEBF is scarcely less than that required to compute the exact value function. However, it does demonstrate the general viability of the representation. Our second type of experimental results uses BEBFs within the context of Least Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003). These experiments demonstrate the robustness of the BEBF approach with Bellman error estimation and goes beyond the theoretical framework established above by using projections that are not weighted by the stationary distribution, and by performing policy improvement within the LSPI framework.

### 4.1. Exact BEBFs

As a simple test to validate the use of BEBFs to represent value functions, we considered the $50$-state chain model introduced by Lagoudakis and Parr (2003) and used by Mahadevan and Maggioni (2006) as a test for basis-function discovery. In this problem, there are $50$ states numbered $1$ through $50$ with a reward of $1.0$ at states $10$ and $41$. Actions are right/left $(+1/-1)$ moves, which succeed with probability $0.9$ and move in the opposite direction with probability $0.1$. We directly encoded the optimal policy into an exact transition matrix and generated exact BEBFs starting with an initial BEBF of $1$. The experiment terminated with a total of $16$ BEBFs, which yielded an exact representation of the value function for all $50$ states. Figure 1c shows the value functions for $5$, $10$, and all $16$ basis functions. To give some sense of the shape of the BEBFs for this problem, we show the first nine basis functions in Figure 2a. The difference between the conditions of this experiment and those of Theorem 3.4 is that we used an unweighted projection because the Markov chain is periodic and does not have a stationary distribution. We did try using an approximate stationary distribution that smoothed over some of the periodicity, and found that it produced a slightly worse fit at the tails and slightly better fit at the peaks for low numbers of basis functions. The performance is comparable to that of other methods that learn features for this task (Mahadevan & Maggioni, 2006).

### 4.2. Approximate BEBFs

LSPI (Lagoudakis & Parr, 2003) is a batch, approximate policy-iteration algorithm that uses a variant of LSTD (Bradtke & Barto, 1996) in its inner loop. For LSPI, basis functions are Q-functions that map from state-action pairs to reals. LSPI typically is initialized with a master set of basis functions from which LSPI makes $a$ copies, one for each action. LSPI is an off-policy algorithm that uses every sample in a corpus of $(s, a, r, s')$ samples to evaluate every policy it considers.

Our modified version of LSPI computes a completely new set of basis functions at each policy-evaluation phase using function approximation. We initialize LSPI with a BEBF estimated from the immediate reward. Within each phase of policy evaluation, our modified LSPI adds new basis functions for each action by training a function approximator on the Bellman error of the current solution. The training data come from evaluating $\hat{Q}_k(s', \pi_k(s')) + r - \hat{Q}_k(s, a)$ for each $(s, a, r, s')$ sample in the corpus. This procedure produces noisy estimates of the Bellman error, which must be smoothed out by the function approximator. We stop adding new basis functions when the maximum norm of the most recently added BEBF is below a threshold of $10^{-5}$, and terminate LSPI when the most recent policy is identical to any previous policy. In the spirit of LSPI, we used a single set of samples for all policy-evaluation iterations and all function approximation BEBF training.

The choice of function-approximation technique for BEBF approximation is, of course, quite important for the overall performance of the method. The function approximator must be generic and expressive enough to capture a wide range of possible functions on the state space. For example, a simple choice of a fixed degree polynomial basis would be useless beyond the first iteration of the BEBF approach, since subsequent basis functions would necessarily lie in the same space as the first and would add nothing to the expressive power of the basis. This observation suggests the use of a non-parametric or semi-parametric function approximator. For our initial experiments in this framework, we made the somewhat naive choice of locally weighted regression (Atkeson et al., 1997), or LWR, to estimate the BEBFs. In retrospect, this choice turned out to be somewhat more data hungry, CPU hungry, and sensitive to parameter settings than we might have liked, the biggest problem being LWR's tendency to underfit or overfit the Bellman error if given a poor choice of precision parameter. It is possible that an automated method for tuning the precision based upon cross validation could have helped. These limitations notwithstanding, we believe our results demonstrate the soundness of the overall approach since our goal is to demonstrate the BEBF technique with a generic function approximator, and not the user friendliness of LWR.

#### 4.2.1. 50-STATE CHAIN

Our first experiments with the 50-state chain used an exact model and randomly generated basis functions. The goal of these experiments was to evaluate the importance of matching an approximate BEBF to the true Bellman error. With random basis functions, we found that quite large Bellman errors could persist until a nearly complete set of $50$ basis functions was constructed. For our experiments with sampled data, we used $8000$ samples from randomly selected actions and a degree-2 polynomial for LWR with a precision of $1.0$. Since the problem is small enough to

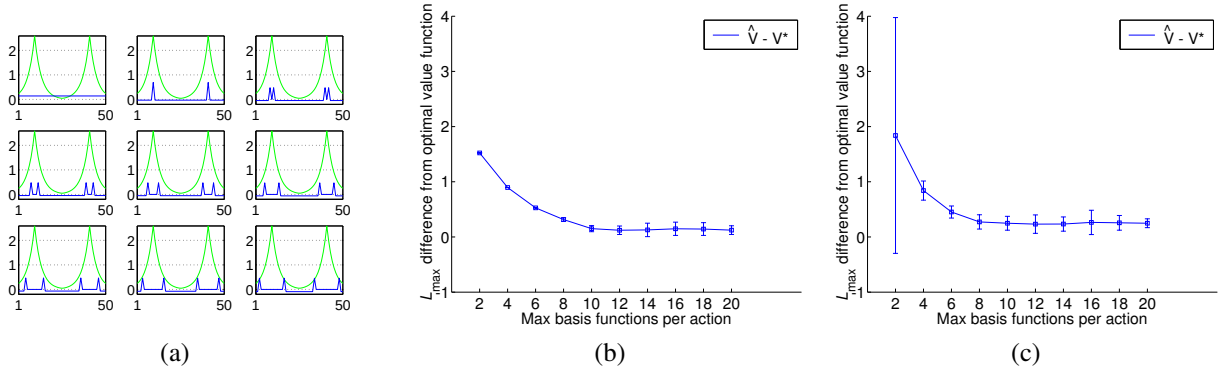



Figure 2. The 50-state chain: (a) The first 9 BEBFs shown beneath the optimal value function. The first BEBF is on the top left, second on the top middle, third on the top right, etc., (b) LSTD policy quality vs. number of training samples (c) LSPI with approximate BEBF performance vs. number of samples.

permit an exact value-function computation, we report the max-norm difference between the estimated value function and the true value function. In Figure 2b, we show the value function quality vs. the maximum number of basis functions permitted per action for policy evaluation using LSTD with BEBFs to evaluate the optimal policy. We show the $\|\cdot\|_\infty$ distance between LSPI's final value function, $\hat{V}$, and $V^*$ in Figure 2c. Even though LSPI has a higher value-function error than LSTD, the value of the resulting $V^\pi$ (not shown due to space limitations) is much closer to $V^*$

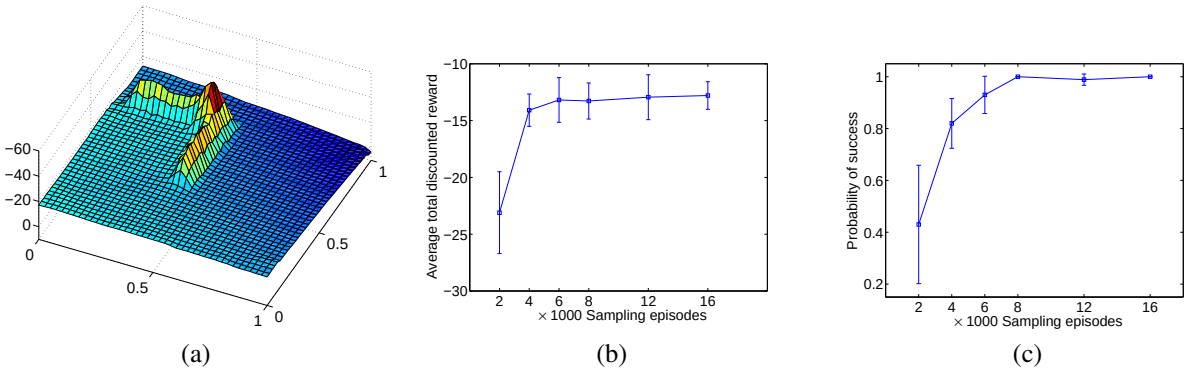
#### 4.2.2. Puddle World

Finally, we report results on the Puddle World problem from Boyan and Moore (1995). Puddle World is a two-dimensional, cost-minimization, navigation problem that requires an agent to move to a corner goal state while avoiding "puddles," which are regions of high cost. This problem is particularly difficult for function approximation methods because the reward and value function have very steep gradients. For this reason, we used a degree-0 polynomial for LWR (equivalent to kernel regression) to help minimize extrapolation errors near the borders of areas with steep gradients. Even with kernel regression, fitted value iteration was not able to produce good policies in our experiments. In Figure 3a, we show a sample value function produced using LSPI with approximately $160,000$ samples, a precision parameter of $5000$, and a max of $40$ basis functions per action. The ridges and peaks in the graph correspond to the puddles in the domain. Readers familiar with this domain will recognize the shape as consistent with previously reported examples of good value functions for this problem. Figure 3b shows the discounted sum of rewards for the learned policy vs. the number of training samples and Figure 3c shows the percentage of trials that reach the goal in less than 200 steps under the learned policy. In Figure 3b and Figure 3c, results are averaged over 10 experiments, except for the 12,000 episode data point, which is averaged over 9 experiments. Each experiment learns a policy using a new sample set, then reports average performance and success rates over 10 trial simulations following the policy from a random starting position. Samples were collected in short episodes starting from a random (non-goal) position and following a random policy for a maximum of ten steps or until reaching the goal area.

## 5. Future Work

In this paper, we have shown the flexibility of the BEBF framework for automatic feature generation within the context of some relatively well understood problems. The theoretical and initial empirical results suggest a fairly flexible framework that could potentially be used to expand the range of problems that are considered within reach of reinforcement-learning techniques. Such steps could require the use of more powerful function-approximation techniques for approximating the Bellman error. An advantage of the BEBF approach is that it provides well circumscribed function-approximation problems at each stage and guaranteed tightening of approximation error bounds.

Although we have demonstrated the use of the BEBF approach with unweighted projections and policy improvement in LSPI, this application goes beyond the theoretical analysis, so there is room for further theoretical development, perhaps drawing inspiration from the work of Munos (2003). In the area of Bellman-error estimation, there is some potential for theoretical development in the direction of sample complexity bounds, perhaps by making some assumptions about the smoothness of the model. Another worthwhile extension would be a strengthening of the approximate BEBF results that quantified the weakening of the exact BEBF guarantees in terms of the error in the BEBF approximator.




(a) (b) (c)

Figure 3. Puddle World: (a) Learned Puddle World value function with approximately 160,000 samples (16,000 sampling episodes) (b) The discounted total reward for the learned policy with LSPI and BEBF vs. the number of sampling episodes (c) The percentage of trials that reach the goal in less than 200 steps vs. the number of sampling episodes.

## 6. Conclusion

We have presented a theoretical analysis of the effects of generating basis functions based upon the Bellman error in the context of linear value-function approximation. Our results show guaranteed tightening of error bounds when the exact Bellman error is used, and give conservative conditions under which improvement can be guaranteed when an approximation of the Bellman error is used. Our experimental results demonstrate the use of a Bellman error approximation based upon locally weighted regression as a means of basis-function generation in the context of least squares policy iteration.

## Acknowledgment

This work was supported by NSF IIS awards 029088 and 0329153. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*, 11–73.

Boyan, J. A., & Moore, A. W. (1995). Generalization in reinforcement learning: Safely approximating the value function. *Advances in Neural Information Processing Systems 7* (pp. 369–376). Cambridge, MA: The MIT Press.

Bradtke, S., & Barto, A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, *2*, 33–58.

Freund, Y., & Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Proc. of the Second European Conference on Computational Learning Theory*. LNCS.

Keller, P., Mannor, S., & Precup, D. (2006). Automatic basis function construction for approximate dynamic programming and reinforcement learning. *Proceedings of the Twenty-third International Conference on Machine Learning*.

Koller, D., & Parr, R. (1999). Computing factored value functions for policies in structured MDPs. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)* (pp. 1332 – 1339). Morgan Kaufmann.

Lagoudakis, M., & Parr, R. (2003). Least squares policy iteration. *Journal of Machine Learning Research*, *4*, 1107–1149.

Mahadevan, S., & Maggioni, M. (2006). *Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes* (Technical Report 2006-35). University of Massachusetts, Amherst.

Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, *41*.

Menache, I., Mannor, S., & Shimkin, N. (2005). Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, *134*.

Munos, R. (2003). Error bounds for approximate policy iteration. *Proceedings of the Twentieth International Conference on Machine Learning*.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Van Roy, B. (1998). *Learning and value function approximation in complex decision processes*. Doctoral dissertation, Massachusetts Institute of Technology.

Vapnik, V., Golowich, S., & Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems 9* (pp. 281–287). Cambridge, MA: MIT Press.

Yu, H., & Bertsekas, D. (2006). *Convergence results for some temporal difference methods based on least squares* (Technical Report LIDS-2697). Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.