Samuel meets Amarel: Automating Value Function Approximation using Global State Space Analysis

Sridhar Mahadevan *
Department of Computer Science
University of Massachusetts
140 Governor's Drive
Amherst, MA 01003
mahadeva@cs.umass.edu

Abstract

Most work on value function approximation adheres to Samuel's original design: agents learn a task-specific value function using parameter estimation, where the approximation architecture (e.g, polynomials) is specified by a human designer. This paper proposes a novel framework generalizing Samuel's paradigm using a *coordinate-free* approach to value function approximation. Agents learn both representations and value functions by constructing geometrically customized task-independent basis functions that form an orthonormal set for the Hilbert space of smooth functions on the underlying state space manifold. The approach rests on a technical result showing that the space of smooth functions on a (compact) Riemannian manifold has a discrete spectrum associated with the Laplace-Beltrami operator. In the discrete setting, spectral analysis of the graph Laplacian yields a set of geometrically customized basis functions for approximating and decomposing value functions. The proposed framework generalizes Samuel's value function approximation paradigm by combining it with a formalization of Saul Amarel's paradigm of representation learning through global state space analysis.

Introduction

Arthur Samuel (Samuel 1959) pioneered the study of value function approximation: his checkers program adjusted the coefficients of a fixed polynomial approximator so that values of states earlier in a game reflected outcomes experienced later during actual play. Samuel's pioneering ideas were formalized using the framework of Markov decision processes (MDP) (Puterman 1994), leading to the field of reinforcement learning (RL) (Sutton & Barto 1998). Substantial expertise has been gained in value function approximation for linear and nonlinear architectures (Bertsekas & Tsitsiklis 1996). However, most systems remain constrained by Samuel's paradigm where agents do not learn the underlying representation. This paper proposes a novel generalization of Samuel's paradigm, where the basis representations for value function approximation are learned by analyzing the large-scale topological structure of the underlying environment. This approach formalizes Saul Amarel's (Amarel 1968) paradigm where agents learn representations through *global* analysis of a state space. Amarel's ideas motivated much subsequent research on representation discovery (Subramanian 1989; Utgoff & Stracuzzi 2002), and many methods for discovering global state space properties like "bottlenecks" and "symmetries" have been studied (McGovern 2002; Ravindran & Barto 2003; Mannor *et al.* 2004; Simsek & Barto 2004). However, this past research lacked a formal framework showing how state space geometry can be transformed into representations for approximating value functions: this paper provides such a unifying framework.

The proposed framework is based on a coordinate-free operator model, where representations emerge from an abstract harmonic analysis of the *topology* of the underlying state space. Value functions are viewed as elements of the Hilbert space of smooth functions on a *Riemannian manifold* (Rosenberg 1997). Hodge theory (Rosenberg 1997) shows that the Hilbert space of smooth functions on a Riemannian manifold has a discrete spectrum captured by the eigenfunctions of the *Laplacian*, a self-adjoint operator on differentiable functions on the manifold. In the discrete setting, the eigenspace of the self-adjoint graph Laplacian operator provides an orthonormal set of basis functions that can approximate any function on the graph (Chung 1997).

Informally, agents learn representations that reflect the agent's experience and an environment's *large-scale* geometry. An agent "living" in a *one-dimensional* environment (e.g., the "chain" MDP in (Lagoudakis & Parr 2003; Koller & Parr 2000)) should "see" the world differently from an agent "inhabiting" a *figure-of-eight* environment or a *closed chain* or a *two-dimensional* grid world. The unexpected result from applying the coordinate-free approach is that Laplacian eigenfunctions resemble actual value functions, and appear remarkably adept at value function approximation. This similarity suggests a new framework for RL where agents learn a suite of "proto-value" or task-independent value functions, which can then subsequently approximate task-specific value functions using rewards (Mahadevan 2005a).

A Markov decision process (MDP) $M = \langle S, A, P^a_{ss'}, R^a_{ss'} \rangle$ is defined by a set of states S, a set of actions A, a transition model $P^a_{ss'}$ specifying the distri-

*This research was supported in part by the National Science Foundation under grant ECS-0218125.

bution over future states s' when an action a is performed in state s, and a corresponding reward model $R^a_{ss'}$ specifying a scalar cost or reward. Abstractly, a value function is a mapping $S \rightarrow \mathcal{R}$ or equivalently a vector $\in \mathcal{R}^{|S|}$. Given a policy $\pi : S \rightarrow A$ mapping states to actions, its corresponding value function V^π specifies the expected long-term discounted sum of rewards received by the agent in any given state s when actions are chosen using the policy. Any optimal policy π^* defines the same unique optimal value function V^* which satisfies the nonlinear constraints

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

Classical techniques, including *value iteration* and *policy iteration* (Puterman 1994), represent value functions exactly using the orthonormal basis $(\phi_1, \ldots, \phi_{|S|})$ of the Euclidean space $\mathcal{R}^{|S|}$, where $\phi_i = [0 \ldots 1 \ldots 0]$ has a 1 only in the i^{th} position. Linear approximation techniques, such as least-squares policy iteration (Lagoudakis & Parr 2003) and linear programming methods for factored MDPs (Guestrin *et al.* 2003), use a set of *handcoded* basis functions $\phi_i(s)$, where the number of basis functions $k \ll |S|$. The proposed approach differs in that value functions are decomposed into a linear sum of learned global basis functions using spectral analysis of the state space graph topology. This approach differs from methods for tuning or adapting basis functions from a predefined set for a specific task (Menache, Shimkin, & Mannor 2005; Poupart *et al.* 2002). Second, since basis functions are learned and represented using a coordinate-free model of the underlying manifold, they reflect large-scale *geodesic* constraints: states close in Euclidean (or some other normed) distance can be assigned very different values if they are far apart in manifold space (e.g., two states on opposite sides of a wall).

Laplace Operator on Riemannian Manifolds

This section introduces the Laplace-Beltrami operator in the general setting of Riemannian manifolds (Rosenberg 1997), as a prelude to describing the Laplace-Beltrami operator in the more familiar setting of graphs, namely spectral graph theory (Chung 1997). These topics are increasingly finding applications in AI, from image segmentation (Shi & Malik 2000) and clustering (Ng, Jordan, & Weiss 2002) to semi-supervised learning (Belkin & Niyogi 2004). However, what is novel to this paper is the use of Laplacian methods for function approximation, by modeling value functions as real-valued functions on a manifold. Formally, a *manifold* $\mathcal{M}$ is a *locally Euclidean* set, with a *homeomorphism* (a bijective or one-to-one and onto mapping) from any open set containing an element $p \in \mathcal{M}$ to the n-dimensional Euclidean space $\mathcal{R}^n$. Manifolds with *boundaries* are defined using a homeomorphism that maps elements to the upper half plane $\mathcal{H}^n$. A manifold is a topological space, i.e. a collection of open sets closed under finite intersection and arbitrary union. In smooth manifolds, the homeomorphism becomes a *diffeomorphism*, or a continuous bijective mapping with a continuous inverse mapping, to the Euclidean space $\mathcal{R}^n$. In a smooth manifold, a diffeomorphism mapping any point $p \in \mathcal{M}$ to its *coordinates* $(\rho_1(p), \ldots, \rho_n(p))$ should be a differentiable function with a differentiable inverse. Given two coordinate functions $\rho(p)$ and $\xi(p)$, or *charts*, the induced mapping $\psi : \rho \circ \xi^{-1} : \mathcal{R}^n \rightarrow \mathcal{R}^n$ must have continuous partial derivatives of all orders. *Riemannian* manifolds are smooth manifolds where the Riemann metric defines the notion of length. Given any element $p \in \mathcal{M}$, the *tangent space* $T_p(\mathcal{M})$ is an n-dimensional vector space that is isomorphic to $\mathcal{R}^n$. A Riemannian manifold is a smooth manifold $\mathcal{M}$ with a family of smoothly varying positive definite inner products $g_p, p \in \mathcal{M}$ where $g_p : T_p(\mathcal{M}) \times T_p(\mathcal{M}) \rightarrow \mathcal{R}$. For the Euclidean space $\mathcal{R}^n$, the tangent space $T_p(\mathcal{M})$ is clearly isomorphic to $\mathcal{R}^n$ itself. One example of a Riemannian inner product on $\mathcal{R}^n$ is simply $g(x, y) = \langle x, y \rangle_{\mathcal{R}^n} = \sum_i x_i y_i$, which remains the same over the entire space. If the space is defined by probability distributions $P(X|\theta)$, then one example of a Riemann metric is given by the Fisher information $\mathcal{I}(\theta)$.

Hodge's theorem states that any smooth function on a compact manifold has a discrete spectrum mirrored by the *eigenfunctions* of Δ, the Laplace-Beltrami self-adjoint operator. On the manifold $\mathcal{R}^n$, the Laplace-Beltrami operator is $\Delta = \sum_i \frac{\partial^2}{\partial x_i^2}$ (often written with a $-$ sign for convention). Functions that solve the equation $\Delta f = 0$ are called *harmonic functions* (Axler, Bourdon, & Ramey 2001). For example, on the plane $\mathcal{R}^2$, the "saddle" function $x^2 - y^2$ is harmonic. *Eigenfunctions* of Δ are functions f such that $\Delta f = \lambda f$, where λ is an eigenvalue of Δ. If the domain is the unit circle S^1, the trigonometric functions $\sin(\theta)$ and $\cos(\theta)$ form eigenfunctions, which leads to *Fourier* analysis. Abstract harmonic analysis generalizes Fourier methods to smooth functions on arbitrary Riemannian manifolds. The *smoothness functional* for an arbitrary real-valued function on the manifold $f : \mathcal{M} \rightarrow \mathcal{R}$ is given by

$$S(f) \equiv \int_{\mathcal{M}} \mid \nabla f \mid^2 d\mu = \int_{\mathcal{M}} f \Delta f d\mu = < \Delta f, f >_{\mathcal{L}^2(\mathcal{M})}$$

where $\mathcal{L}_2(\mathcal{M})$ is the space of smooth functions on $\mathcal{M}$, and ∇f is the gradient vector field of f. For a Riemannian manifold $(\mathcal{M}, g)$, where the Riemannian metric g is used to define distances on manifolds, the Laplace-Beltrami operator is given as

$$\Delta = \frac{1}{\sqrt{\det\, g}} \sum_{ij} \partial_i \left(\sqrt{det\, g}\ \ g^{ij} \partial_j \right)$$

where g is the Riemannian metric, $\det g$ is the measure of volume on the manifold, and ∂_i denotes differentiation with respect to the i^{th} coordinate function.

Theorem 1 *(Hodge (Rosenberg 1997)): Let $(\mathcal{M}, g)$ be a compact connected oriented Riemannian manifold. There exists an orthonormal basis for all smooth (square-integrable) functions $\mathcal{L}^2(\mathcal{M}, g)$ consisting of eigenfunctions of the Laplacian. All the eigenvalues are positive, except that zero is an eigenvalue with multiplicity 1.*

In other words, Hodge's theorem shows that a smooth function $f \in \mathcal{L}^2(\mathcal{M})$ can be expressed as $f(x) = \sum_{i=0}^{\infty} a_i e_i(x)$, where e_i are the eigenfunctions of Δ, i.e. $\Delta e_i = \lambda_i e_i$. The smoothness $S(e_i) = < \Delta e_i, e_i >_{\mathcal{L}^2(\mathcal{M})} = \lambda_i$.

Graph Laplacian and Spectral Graph Theory

The continuous manifold setting provides a motivation for the discrete case studied in spectral graph theory. The Laplace-Beltrami operator now becomes the graph Laplacian (Chung 1997), from which an orthonormal set of basis functions $\phi_1^G(s), \ldots, \phi_k^G(s)$ can be constructed that asymptotically capture any real-valued function on G. The graph Laplacian can be defined in several ways, such as the *combinatorial* Laplacian and the *normalized* Laplacian, in a range of models from undirected graphs with $(0, 1)$ edge weights to directed arbitrary weighted graphs with loops (Chung 1997). For simplicity, consider an undirected graph $G = (V, E)$ without self-loops, where d_v denote the degree of vertex v. Define T to be the diagonal matrix where $T(v, v) = d_v$. Note that for an unweighted graph, the operator $T^{-1}A$, where A is the adjacency matrix, induces a random walk on the graph. The *combinatorial Laplacian* operator is defined as the matrix $L = T - A$:

$$L(u,v) = \begin{cases} d_v & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

whereas the *normalized Laplacian* $\mathcal{L}$ of the graph G is defined as

$$\mathcal{L}(u,v) = \begin{cases} 1 & \text{if } u = v \text{ and } d_v \neq 0 \\ -\frac{1}{\sqrt{d_u d_v}} & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

These definitions can be extended to weighted (directed) graphs, where weights can reflect any local distance measure (Chung 1997). Since almost any function approximator studied previously in MDPs and RL is *isotropic*, e.g, polynomials or CMAC or RBF, modeling manifolds using undirected graphs gives sufficient generality to handle most cases. Since $\mathcal{L}$ is symmetric, its eigenvalues are all real and non-negative. It easily follows from the above definition that

$$\mathcal{L} = T^{-\frac{1}{2}} L T^{-\frac{1}{2}}$$

If G is a constant degree k graph, then it follows that $\mathcal{L} = I - \frac{1}{k}A$, where A is the adjacency matrix of G. For a general graph G, $\mathcal{L} = T^{-\frac{1}{2}} L T^{-\frac{1}{2}} = I - T^{-\frac{1}{2}} A T^{-\frac{1}{2}}$. The Laplacian $\mathcal{L}$ is an *operator* on the space of functions defined on the graph $g : V \to \mathcal{R}$, where ($u \sim v$ means u and v are neighbors):

$$\mathcal{L}g(u) = \frac{1}{\sqrt{d_u}} \sum_{v:u\sim v} \left(\frac{g(u)}{\sqrt{d_u}} - \frac{g(v)}{\sqrt{d_v}} \right) \qquad (1)$$

The *Rayleigh quotient* provides a variational characterization of eigenvalues of $\mathcal{L}$. Eigenvalues can be found by projections of an arbitrary function $g : V \to \mathcal{R}$ onto the subspace $\mathcal{L}g$. The quotient gives the eigenvalues and the functions satisfying orthonormality are the eigenfunctions (here $\langle f, g \rangle_G = \sum_u f(u)g(u)$ denotes the inner product on graph G):

$$\frac{\langle g, \mathcal{L}g \rangle}{\langle g, g \rangle} = \frac{\langle g, T^{-\frac{1}{2}} L T^{-\frac{1}{2}} g \rangle}{\langle g, g \rangle} = \frac{\sum_{u\sim v} (f(u) - f(v))^2}{\sum_u f^2(u) d_u}$$

where $f \equiv T^{-\frac{1}{2}} g$. The first eigenvalue is $\lambda_0 = 0$, and is associated with the constant function $f(u) = \mathbf{1}$, which means the first basis function $g_o(u) = \sqrt{T}\ \mathbf{1}$. Note that the first eigenfunction (associated with eigenvalue 0) of the combinatorial Laplacian L is just the constant function $\mathbf{1}$. The second eigenfunction is the infimum over all functions $g : V \to \mathcal{R}$ that are perpendicular to $g_o(u)$, which gives us a formula to compute the first non-zero eigenvalue λ_1, namely

$$\lambda_1 = \inf_{f \perp \sqrt{T}\mathbf{1}} \frac{\sum_{u\sim v} (f(u) - f(v))^2}{\sum_u f^2(u) d_u} \approx \inf \frac{\int_M \|\Delta f\|^2}{\int_m \mid f \mid^2}$$

The last term characterizes the eigenvalues of the Laplace-Beltrami operator. The Rayleigh quotient for higher-order eigenfunctions is similar: each function is perpendicular to the subspace spanned by previous functions. The *Cheeger* constant h_G of a graph G is defined as

$$h_G(S) = \min_S \frac{|E(S, \tilde{S})|}{\min(\text{vol } S, \text{vol } \tilde{S})}$$

Here, S is a subset of vertices, $\tilde{S}$ is the complement of S, and $E(S, \tilde{S})$ denotes the set of all edges (u, v) such that $u \in S$ and $v \in \tilde{S}$. The volume of a subset S is defined as vol $S = \sum_{x \in S} d_X$. The sign of the basis functions can be used to decompose state spaces (see the Missionaries and Cannibals problem in Figure 1 and the MDP in Figure 4). Consider the problem of finding a subset S of states such that the edge boundary ∂S contains as few edges as possible, where $\partial S = \{(u, v) \in E(G) : u \in S \text{ and } v \notin S\}$. The relation between ∂S and the Cheeger constant is given by $|\partial S| \geq h_G \ \text{vol}\, S$. In the Missionaries and Cannibals task, the Cheeger constant is minimized by setting S to be the states from 1 through 8, since this will minimize the numerator $E(S, \tilde{S})$ and maximize the denominator $min(\text{vol } S, \text{vol } \tilde{S})$. A remarkable identity connects the Cheeger constant with the spectrum of the Laplace-Beltrami operator. This theorem underlies the reason why basis functions associated with eigenvalues of the Laplace-Beltrami operator reflect the intrinsic geometry of environments (see also Figure 5).

Theorem 2 *(Chung 1997): Define λ_1 to be the first (non-zero) eigenvalue of the Laplace-Beltrami operator $\mathcal{L}$ on a graph G. Let h_G denote the Cheeger constant of G. Then, we have $2h_G \geq \lambda_1$.*

Algorithms

The proposed approach suggests a range of algorithms, varying in their complexity. This section presents the simplest methods used in the experiments described below, and more elaborate extensions are discussed in the concluding section. Algorithms derived from this framework result from implementation choices for the four main steps: exploration, graph construction and analysis, basis function construction, and value function approximation. In the first step, agents explore the environment and record an experience sample of tuples (s, a, s', r). The exploration policy can be a random walk, or it can be guided by actual or intrinsically motivated rewards (Singh, Barto, & Chentanez 2005). Methods like least-squares policy iteration (LSPI) (Lagoudakis &

Parr 2003) assume an initial data set of sample transitions to learn policies; this same sample can also be used to build the graph. The second step involves constructing and analyzing the graph. A simple approach is to build an undirected graph with edges (s, s') based on observed state transitions. A more sophisticated approach is to use some positive-definite or even indefinite weight matrix, where weights are estimated transition probabilities or can even include rewards. Graphs analysis comprises of computing the combinatorial or normalized graph Laplacian, and solving the eigenvector problem $\mathcal{L}v = \lambda v$. In the experiments reported below, the combinatorial Laplacian was used, although both approaches have been implemented and tested. Step 3 constructs the basis functions, which in the simplest case are the low-order eigenfunctions of the graph Laplacian. A more sophisticated choice is discussed later. Finally, in Step 4, rewards are combined with the learned basis functions to approximate task-specific value functions. Denote the basis function set by $\Phi_G = \{v_1, \ldots, v_k\}$. Assume noisy samples of the target value function V^π or V^* are known on a subset of states, so that $\hat{V} = (\hat{V}(s_1), \ldots, \hat{V}(s_m))^T$, where $S_G = \{s_1, \ldots, s_m\}$. The low-dimensional reconstruction of a value function V of dimension $\mathcal{R}^{|S|}$ into $\mathcal{R}^k$ for $k \ll |S|$ is computed as follows. Define the Gram matrix $K_G = (\Phi_m^G)^T \Phi_m^G$, where Φ_m^G is the component wise projection of the basis functions onto the states in S_G, and $K_G(i,j) = \sum_k v_i^k v_j^k$. The coefficients are found using a least-squares approach, by solving the equation $\alpha = K_G^{-1}(\Phi_M^G)^T \hat{V}$ where $\alpha = (\alpha_1, \ldots, \alpha_{|S_G|})$ are the coefficients. Control learning methods such as Q-learning or least-squares policy iteration (LSPI) (Lagoudakis & Parr 2003) are easily combined with the proposed framework. In particular, a new algorithm called *Representation Policy Iteration* (RPI) has been developed, which iterates between using the current policy to learn a new representation, and using the learned representation to find a new policy (Mahadevan 2005b). In initial experiments, RPI outperformed LSPI on the classic chain problem (Koller & Parr 2000) using two handcoded state embeddings (polynomials and radial basis functions).

Illustrative Experiments

This section illustrates the framework using experiments on simple deterministic MDPs, as these suffice to highlight the main ideas. The experiments also assume step 1 has been completed yielding a complete graph of the environment for analysis (the problem of analyzing partial graphs is discussed below). It is instructive to begin with Amarel's classic Missionaries and Cannibals problem shown in Figure 1. This environment is modeled as an undirected graph. The initial state is **3300L** (top left node numbered 1) indicating that all the three missionaries and cannibals are on the left bank, and the boat is on the left bank as well. The goal state is **0033R** (top right node numbered 16), where the missionaries and cannibals are safely on the other side.

The proposed approach of learning representations for function approximation can be contrasted with handcoded approaches such as the *polynomial encoding* studied in

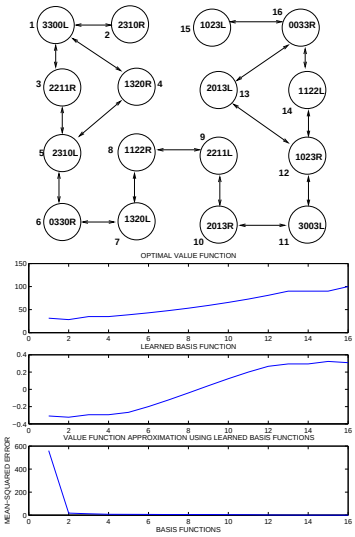

Figure 1: Shown here on top is the graph representing the missionaries and cannibals problem. The plots below show the optimal value function (top plot), and a basis eigenfunction from the orthonormal set spanning the Hilbert space of smooth functions on this graph. These basis functions can look surprisingly similar to value functions; their sign mirrors the two-sided symmetry of this state space. The bottom figure shows the optimal value function is almost exactly approximated with just two learned basis functions, achieving a dimensionality reduction from $\mathcal{R}^{16} \rightarrow \mathcal{R}^2$.

(Koller & Parr 2000; Lagoudakis & Parr 2003). Here, a state s is mapped to the monomials $\phi(s) = [1\ s \ldots s^i]^T$. This encoding easily extends to a state action encoding $\phi(s,a)$ by adding $\log_2 |A|$ bits for encoding actions. Interestingly, this basis set is a special case of the proposed framework which builds customized orthonormal basis sets for an arbitrary graph (manifold). For example, choosing $i = 3$ would map state 1 above to $\phi(1) = [1\ 1\ 1]^T$, state 2 to $\phi(2) = [1\ 2\ 4]^T$, state 3 to $\phi(s) = [1\ 3\ 9]^T$ and so on, reducing the value function dimensionality from $\mathcal{R}^{16} \rightarrow \mathcal{R}^3$. As it happens, this polynomial embedding works well on the Missionaries and Cannibals problem, but not for the MDPs shown in Figure 2 and Figure 3. In contrast, the proposed approach automatically builds the basis functions $\phi(s)$ using global state space analysis, achieving a dimensionality reduction as good as the polynomial encoding for the Missionaries and Cannibals problem, and far superior to it for the MDPs shown in Figure 2 and Figure 3. Figure 1 shows that the shape of the second basis function of the combinatorial Laplacian (shown for convenience with the sign inverted) resembles the value function. This is no coincidence: the Laplacian is an operator on the Hilbert space of functions on the graph that enforces *geodesic* smoothness in a manner analogous to the Bellman backup operator on the space of value functions in an MDP. Both map neighboring vertices on the graph to adjacent real values.

Figure 3 demonstrates that Laplacian eigenfunctions excel on standard RL benchmark problems: the mean-squared

error using the Laplacian basis functions on a 30×30 grid world environment is substantially less than the handcoded polynomial state encoding. Figure 4 shows Laplacian eigenfunctions can recursively decompose larger MDPs into smaller ones. This figure also shows that eigenfunctions derived from the right topology are much more effective than those produced from a dramatically incorrect topology (a complete graph). Figure 5 shows geometric structure discovery and value function approximation for a larger five room grid world MDP.

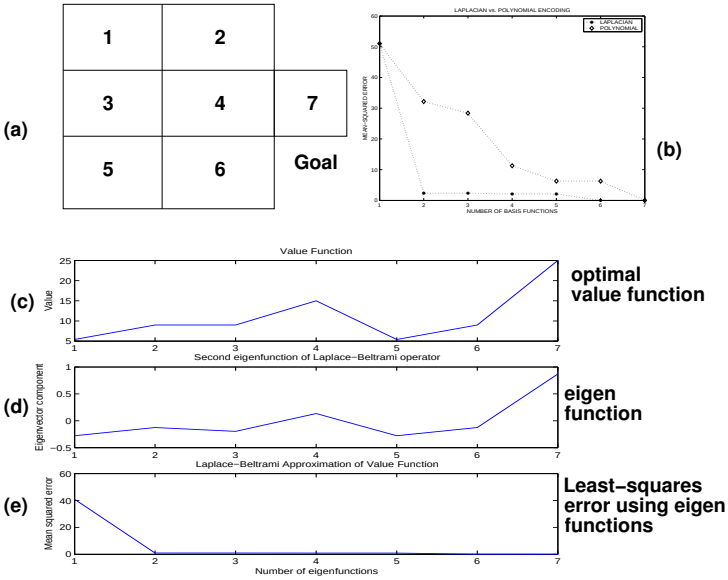

Figure 2: For the environment shown in (a), an eigenfunction of the (graph) Laplacian shown in (d), from the orthonormal basis set of smooth functions on the manifold, closely resembles the value function shown in (c). (b) and (e) show the optimal value function can be approximated with just two learned basis functions. The plot in (b) compares the mean-squared error using the learned representation (bottom curve) with a fixed polynomial encoding (top curve) for varying numbers of basis functions.

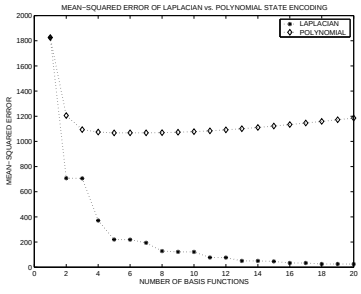

Figure 3: Mean squared error in approximating the optimal value function for a 30×30 grid world for varying numbers of learned Laplacian basis functions (bottom curve) and varying degrees of handcoded polynomials (top curve).

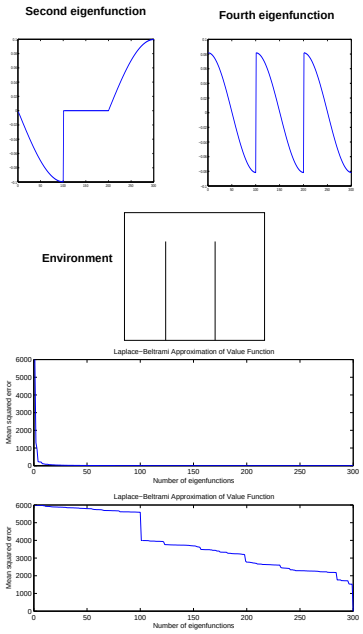

Figure 4: Laplacian eigenfunctions decompose the state space of a MDP into smaller units. Here, the second eigenfunction splits the environment into three "arms". The fourth eigenfunction splits each arm into two symmetric pieces. The bottom plot shows mean-squared error using Laplacian eigenfunctions from the right topology (top curve) is much lower than from an incorrect (complete graph) topology (bottom curve).

Analysis and Future Work

The proposed approach can be extended to weighted graphs, where the weights reflect estimated transition probabilities or rewards. Learning such graphs will require more samples. Once a graph is learned, the complexity of spectral analysis is $O(N^3)$, where N is the number of nodes in the graph. However, sample-based approximations can significantly reduce this complexity. The approach can be extended to the more realistic case where agents can only build partial graphs as discussed below. In large state spaces, exploration, graph construction, and spectral analysis can be interleaved.

A number of specific directions are being investigated to scale the approach. The state space can be modeled at multiple levels of abstraction, where higher level graphs can be viewed as a SMDP-homomorphism of lower-level graphs (Ravindran & Barto 2003). Laplacian eigenfunctions capture symmetries and other geometric regularities for automatically learning homomorphisms. *Nystrom* approximations for solving integral equations reduce the complexity of spectral analysis from $O(N^3)$ to $O(m^2N)$ where $m \ll N$ is the number of samples for which complete local distance information is available (Fowlkes *et al.* 2004). A number of other randomized low-rank approximations show that interesting linear algebra can be performed in time independent of the size of the matrix (Achlioptas, McSherry, & Scholkopff 2002; Frieze, Kannan, & Vempala 1998). An-

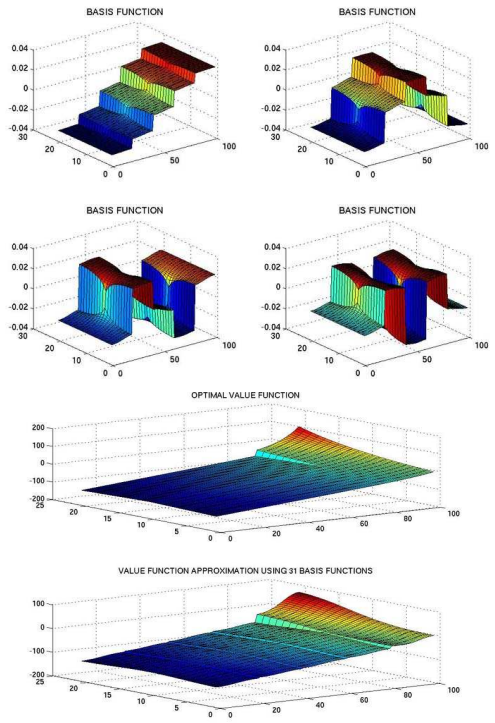

Figure 5: Top: eigenfunctions learned for a five-rooom environment with $5 \times 21 \times 20 = 2100$ states. Middle: the optimal value function; Bottom: approximation using 31 learned eigenfunctions.

other direction being investigated is to build a sparse hierarchical representation of the Laplace-Beltrami operator using *diffusion wavelets* (Coifman & Maggioni). This approach yields a multi-scale hierarchical tree of learned basis functions, which can be efficiently computed in $O(N \log^2 N)$. Unlike Fourier methods, which are based on differential equations, wavelets are based on dilation equations and use basis functions with compact support. A detailed comparison of diffusion wavelets and Laplacian eigenfunctions is underway (Mahadevan & Maggioni 2005).

References

Achlioptas, D.; McSherry, F.; and Scholkopff, B. 2002. Sampling techniques for kernel methods. In *Proceedings of the International Conference on Neural Information Processing Systems*. MIT Press.

Amarel, S. 1968. On representations of problems of reasoning about actions. In Michie, D., ed., *Machine Intelligence 3*, volume 3, 131–171. Elsevier/North-Holland.

Axler, S.; Bourdon, P.; and Ramey, W. 2001. *Harmonic Function Theory*. Springer.

Belkin, M., and Niyogi, P. 2004. Semi-supervised learning on Riemannian manifolds. *Machine Learning* 56:209–239.

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

Chung, F. 1997. *Spectral Graph Theory*. American Mathematical Society.

Coifman, R., and Maggioni, M. Diffusion wavelets. *Applied Computational Harmonic Analysis*. To Appear.

Fowlkes, C.; Belongie, S.; Chung, F.; and Malik, J. 2004. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(2):1373–1396.

Frieze, A.; Kannan, R.; and Vempala, S. 1998. Fast monte-carlo algorithms for finding low-rank approximations. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 370–378.

Guestrin, C.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient solution algorithms for factored MDPs. *Journal of AI Research* 19:399–468.

Koller, D., and Parr, R. 2000. Policy iteration for factored MDPs. In *Proceedings of the 16th Conference on Uncertainty in AI*.

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

Mahadevan, S., and Maggioni, M. 2005. Automating value function approximation using diffusion wavelets. Under preparation.

Mahadevan, S. 2005a. Proto-value functions: Developmental reinforcement learning. Submitted.

Mahadevan, S. 2005b. Representation policy iteration. In *Proceedings of the* 21^{st} *Conference On Uncertainty In Artificial Intelligence*.

Mannor, S.; Menache, I.; Hoze, A.; and Klein, U. 2004. Dynamic abstraction in reinforcement learning via clustering. In *ICML*.

McGovern, A. 2002. *Autonomous Discovery of Temporal Abstractions from Interactions with an Environment*. Ph.D. Dissertation, University of Massachusetts, Amherst.

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

Ng, A.; Jordan, M.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS*.

Poupart, P.; Patrascu, R.; Schuurmans, D.; Boutilier, C.; and Guestrin, C. 2002. Greedy linear value function approximation for factored markov decision processes. In *AAAI*.

Puterman, M. L. 1994. *Markov decision processes*. New York, USA: Wiley Interscience.

Ravindran, B., and Barto, A. 2003. SMDP homomorphisms: An algebraic approach to abstraction in semi-markov decision processes. In *Proceedings of the 18th IJCAI*.

Rosenberg, S. 1997. *The Laplacian on a Riemannian Manifold*. Cambridge University Press.

Samuel, A. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3(3):210–229.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE PAMI* 22:888–905.

Simsek, Ö., and Barto, A. G. 2004. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *ICML*.

Singh, S.; Barto, A.; and Chentanez, N. 2005. Intrinsically-motivated reinforcement learning. In *NIPS*.

Subramanian, D. 1989. *A theory of justified reformulations*. Ph.D. Thesis, Stanford University.

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

Utgoff, P., and Stracuzzi, D. 2002. Many-layered learning. *Neural Computation* 14:2497–2529.