A Multiscale Framework for Markov Decision Processes using Diffusion Wavelets

M. Maggioni & S. Mahadevan
CMPSCI TR 06-36

A Multiscale Framework for Markov Decision Processes using Diffusion Wavelets

Mauro Maggioni MAURO.MAGGIONI@YALE.EDU
Program in Applied Mathematics
Department of Mathematics
Yale University
New Haven, CT, 06510

Sridhar Mahadevan MAHADEVA@CS.UMASS.EDU
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA

Editor:

Abstract

We present a novel hierarchical framework for solving Markov decision processes (MDPs) using a multiscale method called *diffusion wavelets*. Diffusion wavelet bases significantly differ from the Laplacian eigenfunctions studied in the companion paper (Mahadevan and Maggioni, 2006): the basis functions have *compact* support, and are inherently multi-scale both spectrally and spatially, and capture *localized* geometric features of the state space, and of functions on it, at different granularities in space-frequency. Classes of (value) functions that can be compactly represented in diffusion wavelets include piecewise smooth functions. Diffusion wavelets also provide a novel approach to approximate powers of transition matrices. Policy evaluation is usually the expensive step in policy iteration, requiring $O(|S|^3)$ to directly solve the Bellman equation (where $|S|$ is the discrete state space size or sample size in continuous spaces). Diffusion wavelets compactly represent powers of transition matrices, yielding a direct policy evaluation method requiring only $O(|S|)$ operations in many cases, which is remarkable because the *Green's function* $(I - \gamma P^\pi)^{-1}$ is usually a full matrix requiring $O(|S|^2)$ just to store each entry. A range of illustrative examples and experiments, from simple discrete MDPs to classic continuous benchmark tasks like inverted pendulum and mountain car, are used to evaluate the proposed framework.

Keywords: Markov decision processes, reinforcement learning, policy iteration, multiscale analysis, diffusion wavelets

1. Introduction

This paper introduces a novel multiscale framework for solving Markov decision processes (MDPs) (Puterman, 1994), based on *diffusion wavelets* (Coifman and Maggioni, 2004). In the companion paper (Mahadevan and Maggioni, 2006) we discussed the use of eigenfunctions of the Laplacian on a state space for value functions approximation. Diffusion wavelet bases are also adapted to the geometry of the state space, and can be learned once the state space is explored. However, the wavelet framework provides significant advantages in that the constructed bases have compact support, and the approach yields a completely novel

approach to the hierarchical abstraction of Markov chains and Markov decision processes, a classical problem of much interest in reinforcement learning (Barto and Mahadevan, 2003).

The central motivation behind both Laplacian eigenfunctions and diffusion wavelet bases is to build representations that capture how the geometry of the state space affects the functions of interest on that space (e.g. the value function). Among these, the eigenfunctions of certain Laplacians on the state space are natural candidates because of their smoothness properties and their approximation properties in smoothness spaces (see Section 5.1 and 5.2). There is a growing trend in machine learning towards designing techniques that exploit the intrinsic geometry of the space that the data lie in (see for example (Belkin and Niyogi, 2003a; Coifman and Maggioni, 2005; Maggioni and Coifman, 2006; Coifman et al., 2005a,b; Coifman and Maggioni, 2004; Bremer et al., 2004)). It is natural to assume that the functions of interest have some regularity with respect to the natural geometry of the state space, and they are appropriately approximated by suitable representations via elements that are adapted to the geometry of the state space.

The Fourier analysis associated with expansions on eigenfunctions of the Laplacian is a powerful tool for *global* analysis of functions, however it is known to be relatively poor at representing or approximating local or transient properties. This motivated the construction, about 20 years ago, of classical wavelets, which allow a very efficient *multiscale* analysis, much like a tunable microscope probing the properties of a function at different locations and scales. Recently wavelet analysis has been generalized in a natural way to manifolds and graphs (Coifman and Maggioni, 2004), under the name of *diffusion wavelets*. They are associated with a diffusion process that defines the different scales, and allow a multiscale analysis of functions on manifolds and graphs, as shown in (Coifman and Maggioni, 2004; Maggioni et al., 2005b). Diffusion wavelets have desirable properties in view of applications to learning, function approximation, compression and denoising of functions on graphs and manifolds. In many applications the multiscale diffusion wavelet analysis constructed is interpretable and meaningful. For example, when applied to the analysis of document corpora, it suggests groupings of documents (or words) at different scales, corresponding to topics at different levels of specificity; when applied to Markov decision processes, it leads to new aggregate groupings of states and actions; when applied to the analysis of images, it leads to multiscale features of images.

Diffusion wavelet bases can be viewed as a multiscale extension of the notion of a *proto-value function* (Mahadevan, 2005c,a), in that these basis functions are also task-independent and collectively span the space of all possible value functions on a given state space. Like Laplacian eigenfunctions, diffusion bases incorporate geometric constraints intrinsic to the environment (see for example Figure 1). However, diffusion wavelet bases are more powerful in that their intrinsic multiresolution nature enables them to be used not just to represent value functions, but also provides a new representational framework for compact approximation of powers of transition matrices.

Diffusion wavelets enable a hierarchical analysis of MDPs by learning a multiscale tree of *wavelet*-type basis functions on the state space of a MDP, which enables an efficient hierarchical representation of not just value functions, but also yields a fast algorithm for the direct solution of the Bellman equation for policy evaluation. Bellman's equation usually involves the solution of a sparse linear system of size $|S|$, where S is the state space. A classical direct solution of the system is infeasible for large problem sizes, since it requires $\mathcal{O}(|S|^3)$ steps. One common technique is to use an iterative method, such as value iteration, which has worst case complexity $\mathcal{O}(|S|^2)$ for sparse transition matrices, $\mathcal{O}(|S| \log |S|)$ when the problem is well-conditioned and only low-precision is required. The approach to policy evaluation based on diffusion wavelets is fundamentally different, and yields a *direct* solution in time $\mathcal{O}(|S| \log^2 |S|)$. Also, it results in a multiscale structure that

depends only on the structure of the state space and on the policy, which can be reused for different reward functions and discount rates.

The rest of this paper is organized as follows. In Section 2 we review some basic ideas in multiscale analysis, both for function approximation and study of multiscale processes such as random walks, or more general Markov chains. (A very short review of classical wavelet multiscale analysis, for function approximation in Euclidean spaces, can be found in the Appendix.) We then introduce diffusion wavelets, simultaneously addressing multiscale function approximation on manifolds and graphs, as well as the multiscale representation of "diffusion processes" on such spaces. In Section 4 we describe applications of diffusion multiscale analysis to Markov Decision Processes (MDPs). Each policy in an MDP induces a Markov chain process, for which there is a natural multiscale analysis whose construction and interpretation is described. We then show how to use these multiscale decompositions for multiresolution value function approximation, for example to be included in a least-squares policy iteration scheme, very much in parallel with what we presented in the companion paper (Mahadevan and Maggioni, 2006) using eigenfunctions of the Laplacian. We also show that, remarkably, this multiscale analysis enables a direct efficient solution of Bellman's equation, providing a numerical scheme of independent interest. In Section 5 we consider function approximation in spaces of smooth and piecewise smooth functions, and compare the behavior of approximations by eigenfunctions of the Laplacian and by diffusion wavelets. In Section 6 we discuss the case of continuous state spaces, as opposed to finite state spaces. In particular we discuss how to extend basis functions, usually learned from a finite discrete sampling of the state space, to new states. We devote Section 7 to a detailed experimental study of diffusion wavelets in MDPs. We conclude with a discussion of ongoing and future research directions.

2. Multiscale Analysis: a quick overview

Due to the somewhat technical nature of the diffusion wavelet framework and construction, we begin first with some illustrative examples that provide insight into this multiresolution method. Unlike the first paper, where the value function is approximated on a subspace of eigenfunctions of a graph Laplacian, in this paper we consider a different set of bases, called *diffusion wavelets*, that are multiscale in nature. Qualitatively, each basis function has two parameters, corresponding to a scale j and a location k, and it resembles an oscillating function supported on a ball of approximately geodesic diameter 2^j. For large j (coarsest scale) they resemble the lowest frequency eigenfunctions of the Laplacian. These basis functions are learned from the data (similar to the eigenfunctions of the Laplacian) and are automatically constructed, given a matrix representing a diffusion process T^π (e.g. a "random walk", or a Markov process corresponding to a policy) on the state space $\mathcal{S}$. They are naturally defined and constructed for non-symmetric processes T^π, unlike the Laplacian.

The motivation for the introduction and use of diffusion wavelet basis functions is that they are able to represent efficiently larger classes of (value) functions than Laplacian eigenfunctions. They also enable analyzing an MDP at different levels of spatial and temporal granularity . To make these ideas more concrete, let us first illustrate the multiscale basis functions using two simple examples.

Example 1 (Room with an obstacle) *In this example, the state space $\mathcal{S}$ is a square discrete grid with a square obstacle placed in the middle (see Figure 1). We consider the natural random walk on the graph representing the state space, with reflecting boundary conditions. In the continuous setting, this would correspond to considering the heat kernel with Neumann boundary conditions. The associated diffusion wavelet basis functions, automatically constructed from a matrix (unsorted!)*

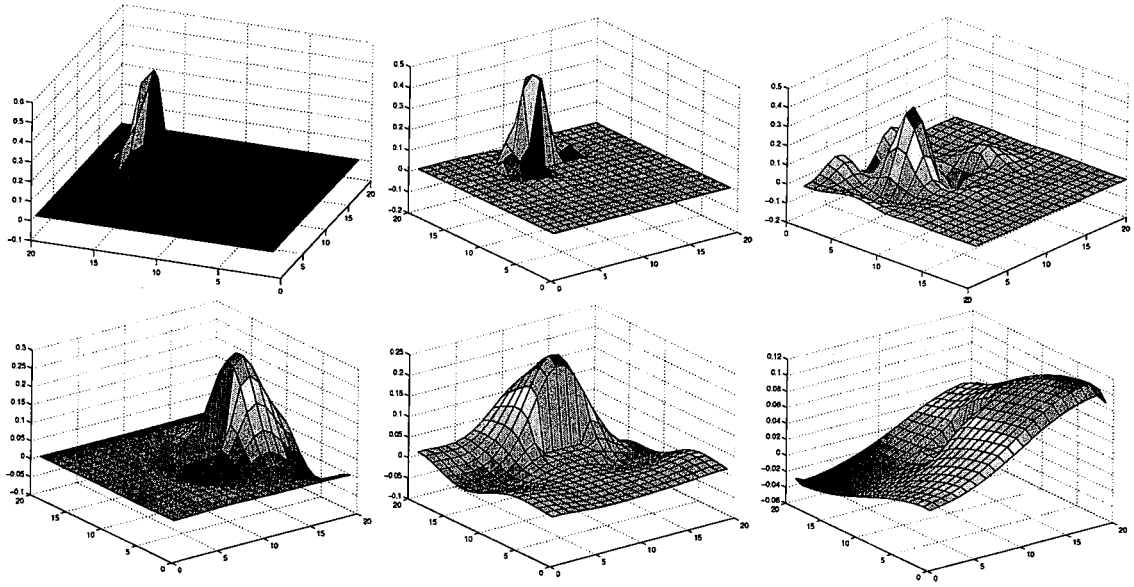


Figure 1: The state space is a room with an obstacle in the middle. The diffusion scaling functions are automatically adapted to the state space. From top to bottom, left to right, we represent $\varphi_{3,10}, \varphi_{3,23}, \varphi_{4,32}, \varphi_{5,5}, \varphi_{6,8}, \varphi_{9,2}$, where the first index denotes scale (the larger the index the coarser the scale) and the second one indexes the location. Compared to the eigenfunctions of the Laplacian, each of which has global support, the scaling functions are localized at different scales. At the coarsest scale, they are very similar to the eigenfunctions of the Laplacian.

representing the random walk process on $\mathcal{S}$, are naturally adapted to $\mathcal{S}$, as shown in Figure 1. In the construction of the basis functions, there is no assumption that the points are uniformly sampled on a grid.

Example 2 (Two rooms with a door) *The second example represents a two-room spatial domain, connected by a single door, similar to the environments analyzed in the first paper (Mahadevan and Maggioni, 2006). The state space is a discretization of two rooms, connected by one edge (representing the door). Some diffusion wavelets associated with the natural random walk on this state space are illustrated in Figure 2.*

By representing certain classes of value functions (e.g. smooth value functions) efficiently, the dimensionality of the problem is significantly reduced. Solving the reduced problem in a subspace of dimension k requires solving Bellman's equation, which has in general cost $\mathcal{O}(k^3)$: this makes any savings in k very attractive. We will discuss the approximation properties of eigenfunctions of the Laplacian in more detail in this paper, but the main observation is that eigenfunctions only efficiently represent globally smooth functions, but not functions which are only piecewise smooth, or not uniformly smooth, on the state space. See Figure 3 for a simple but revealing example. (We refer the reader to Figure 19 for an example involving classical wavelets.)

Wavelets in general efficiently represent functions which are not uniformly smooth because they perform a multiscale analysis, in such a way that the smoothness of the

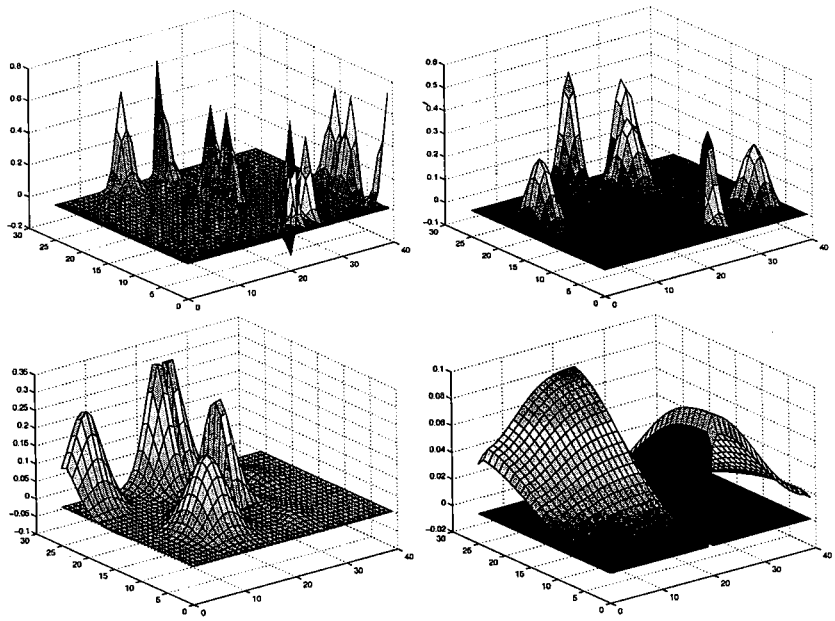


Figure 2: Diffusion wavelet scaling functions on a discrete two-room spatial domain connected by a common door. Top: some scaling functions at scale 3 and 4, bottom: some scaling functions at scale 5 and 8. All the scaling functions are naturally adapted to the state space and its topological features. Large (i.e. coarse) scale scaling functions are very similar to eigenfunctions of the Laplacian.

function at a point affects the size of the coefficients only of those wavelets whose support contain that point. The number of wavelets whose support includes any given point is proportional to the number of scales considered (which grows proportionally to $\log |\mathcal{S}|$, where $\mathcal{S}$ is the state space). Hence if a function is "generally" smooth, with a few locations of non-smoothness, only very few wavelet coefficients will be affected.

Example 3 *In Figure 3 we compare the performance of diffusion wavelet bases and Laplacian eigenfunctions in approximating a value function in a discrete two-room environment, with two positive rewards at opposite corners of the two rooms. The value function is quite peaked, and hence induces a large gradient in localized regions close to the rewards. It is thus non-smooth, at least in the sense that the gradient is large in certain regions, or not uniformly smooth in the sense that the gradient is very small in some regions and quite large in others. We can approximate the value function by choosing the best set of eigenfunctions of the Laplacian for approximating it (nonlinear approximation), and contrast it with the best set of diffusion wavelets. The difference in approximation rates is highly significant, and reveals the power of the multiscale diffusion bases.*

With these illustrative examples, we can now turn to a more detailed description of how such a multiscale analysis can be constructed on general state spaces by first reviewing the *diffusion analysis* framework (Coifman et al., 2005a,b; Coifman and Maggioni, 2004; Bremer et al., 2004; Maggioni et al., 2005b,a; Coifman and Lafon, 2004a,b; Lafon, 2004; Coifman and Maggioni, 2005).

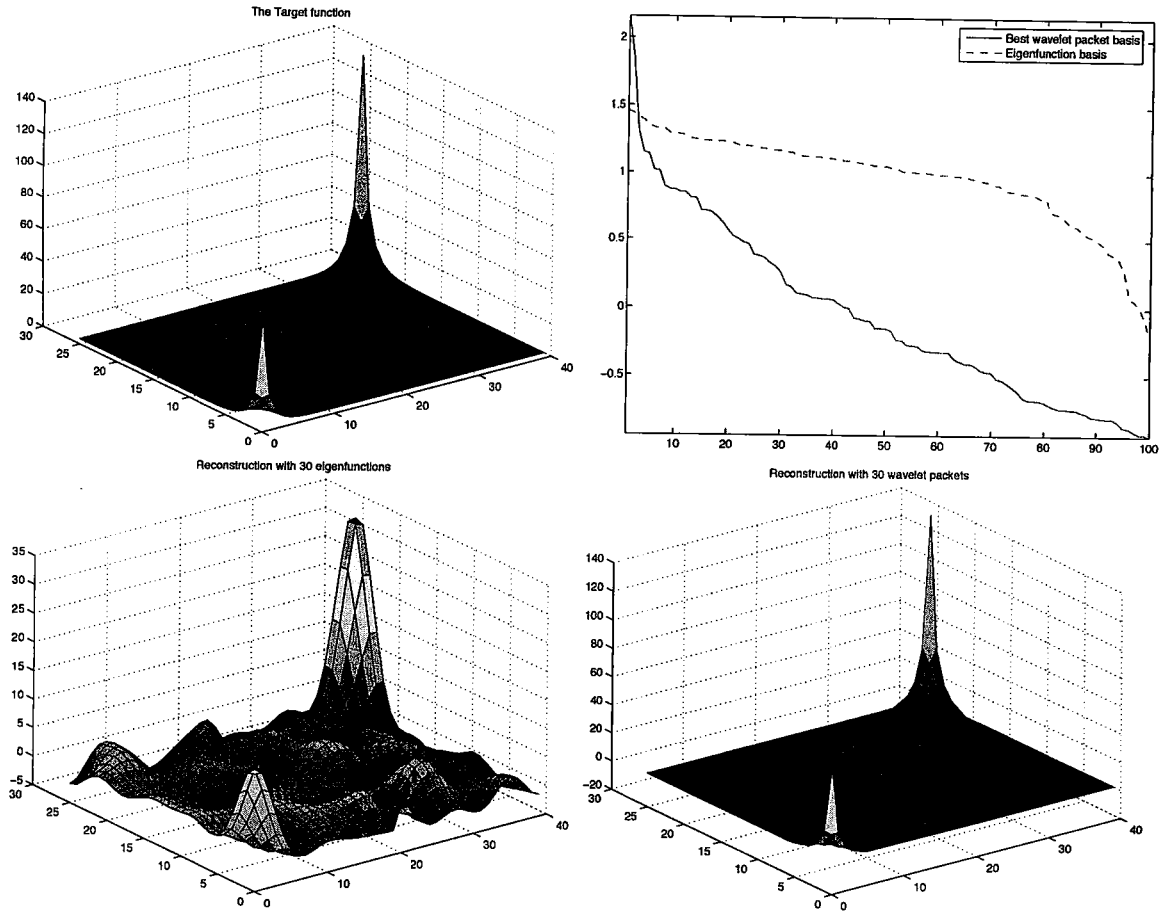


Figure 3: Approximation of a value function which is not very smooth because of localized peaks that induce local large gradients. This value function corresponds to a discrete two room domain connected by a door in the middle, with two (positive but different) rewards at the two opposite corners of the two rooms. Top left: the exact desired value function we need to approximate. Top right: decay of the coefficients onto the Laplacian eigenfunction basis and the best wavelet packet basis (in $\log_{10}$ scale). Bottom left: approximation obtained by nonlinear approximation with the 30 best eigenfunctions of the Laplacian. Bottom right: approximation obtained by nonlinear approximation with 30 best wavelet packet functions.

3. Technical background

In this section we review some background material on multiscale analysis of functions and diffusion processes, which may help the reader in better understanding the diffusion wavelet framework. The topics we discuss expand on the similar discussion in the first paper (Mahadevan and Maggioni, 2006) on Laplacian eigenfunctions. We discuss some ideas from *"diffusion analysis"*, the study of diffusion processes on graphs, which leads to a natural analysis of the geometry of a space, manifold or graph, and of functions on it

(Coifman et al., 2005a,b; Coifman and Maggioni, 2004; Bremer et al., 2004; Maggioni et al., 2005b,a; Coifman and Lafon, 2004a,b; Lafon, 2004; Coifman and Maggioni, 2005). This framework has its roots in harmonic analysis (Stein, 1970), potential theory and the study of Brownian motion (Kemeny and Snell, 1976; Doob, 1984), partial differential equations (Evans, 1998), probability, and spectral graph theory Chung (1997). It is related to work in the machine learning community (Belkin and Niyogi, 2003a, 2001; Kondor and Lafferty, 2002; Lafferty and Lebanon, 2005, 2002; Zhu et al., 2003). A global diffusion analysis leads to the study of the eigenfunctions of the Laplacian discussed in the companion paper (Mahadevan and Maggioni, 2006); here we focus on multiscale analysis. This analysis is in fact two-fold: on the one hand it applies to function approximation, on the other to the multiscale representation of certain Markov processes.

This section is organized as follows: in section 3.1 we discuss the framework of diffusion analysis, introducing the relevant basic concepts and notation. In section 3.1.2 we describe the connection with Fourier analysis on graphs and manifolds, and eigenfunctions of the Laplacian, and briefly discuss function approximation with eigenfunctions. In section 3.2 we motivate and introduce multiscale diffusion analysis of Markov processes (Coifman and Maggioni, 2004; Maggioni et al., 2005a). We briefly discuss classical multiscale function approximation with wavelets in the appendix (section 10) (excellent classical references are (Daubechies, 1992; Mallat, 1998)). In section 3.2.1 we introduce *diffusion wavelets* (Coifman and Maggioni, 2004; Coifman et al., 2005b; Maggioni et al., 2005a) and briefly discuss their construction. Section 3.2.3 summarizes various applications of diffusion wavelets.

3.1 Diffusion Analysis

An intrinsic analysis of a data set, modeled as a graph or a manifold, can be developed by considering a natural random walk P on it. The random walk can be viewed as a diffusion process on the data set, which explores it and discovers clusters and regions separated by bottlenecks (Chung, 1997; Ng et al., 2001; Belkin and Niyogi, 2001; Zha et al., 2001; Lafon, 2004; Coifman et al., 2005a,b). Recall that for a weighted undirected graph (G, E, W), the random walk is defined by $P = D^{-1}W$, where D is the diagonal matrix defined by $D_{ii} = \sum_{j \sim i} W_{ij}$. On a manifold, the natural random walk is Brownian motion on the manifold, which can be obtained by "restricting" Brownian motion in Euclidean space (Sidorova et al., 2003; Sidorova, 2003). This Markov process can be represented as an integral operator, whose kernel is heat kernel on the manifold (Rosenberg).

For the purpose of this discussion, we restrict our attention to the case of a finite undirected[1] weighted graph (G, E, W). Most of the constructions we present can be generalized, *mutatis mutandis*, to directed graphs, infinite graphs, smooth compact Riemannian manifolds, and more general spaces (Coifman and Maggioni, 2004). If P represents one step of the random walk, by the Markov property P^t represents t steps. For an initial condition δ_x (i.e, where x is the starting state), $P^t\delta_x(y)$ represents the probability of being at y at time t, conditioned on starting in state x. The matrix P encodes local similarities between points, and the matrix P^t is diffusing, or integrating, this local information for t steps to larger and larger neighborhoods of each point. The process $\{P^t\}_{t \geq 0}$ can be analyzed at different time scales. For very large times ($t \to +\infty$), the random walk can be analyzed through its top eigenvectors, which are related to those of a Laplacian on the graph/manifold. The analysis for large times leads to *Fourier analysis* of the large scale spatial and temporal regularities of the graph/manifold, and to the identification of useful structures, such as large scale clusters.

1. The construction of diffusion wavelets applies as is to *directed* graphs (Coifman and Maggioni, 2004); however the connection with the spectral theory of the operator is quite different and a delicate matter. In this section we restrict ourselves to undirected graphs for pedagogical purposes.

For small and medium times (t bounded away from infinity), the random walk in general cannot be studied effectively with Laplacian eigenfunctions, which are global and not suited for analyzing small- and medium-scale behavior. On the other hand, many interesting features of the data and of functions on the data can be expected to exist at small and medium time scales: one remarkable example is complex (computer, biological, information, social) networks, where communities (of computers, genes and/or proteins, people) of different sizes co-exist and cluster together at different scales. Another important example is the boundary between two classes in a classification problem. In the case of Markov Processes, it is easy to imagine regions of the state space where the process is smoother than others (e.g. inverted pendulum near the vertical position and zero velocity versus almost horizontal position with high velocity). The task of analyzing P^t for all times and locations seems tantalizing, since it would seem to require either large time in order to compute all powers of P (which is computationally expensive since, even if P is sparse, its powers are not), and/or large space to store those powers. However it is easy to observe is that there is redundancy in time and space in the family $\{P^t(x,y)\}_{t\geq 0; x,y\in X}$. First of all there is a spatial redundancy: if x and y are close and t^2 is large (depending on the distance between x and y), $P^t(x,\cdot)$ is very similar to $P^t(y,\cdot)$ (as distributions on X). Secondly, there is a redundancy across time scales: if we know $P^t(x,\cdot)$ and $P^t(y,\cdot)$, then by the Markov property we know $P^{2t}(x,y)$. It is remarkable that this redundancy can be eliminated, and an efficient multiscale encoding is possible. This leads to *diffusion wavelet analysis*, which is described in Section 3.2.1.

3.1.1 BASIC SETUP AND NOTATION

Fourier analysis on graphs is based on the definition of a Laplacian, and on its eigenfunctions, which constitute an orthonormal basis of Fourier modes on the graph. Spectral graph theory (Chung, 1997) studies the properties of the Laplacian on a graph and of its eigenfunctions, and has been applied to a wide range of tasks in the design of computer networks, parallel computation, clustering, manifold learning, classification.

We introduce notation that will be used throughout the paper. $x \sim y$ means that there is an edge between x and y, $d(x) = \sum_{x\sim y} w(x,y)$ is the degree of x, D is the diagonal matrix defined by $D_{xx} = d(x)$, and W the matrix defined by $W_{xy} = w(x,y) = w(y,x)$. We can assume $w(x,y) > 0$ if $x \sim y$. Sometimes G is naturally endowed with a measure (weight) μ on its vertices. A typical example is $\mu(\{x\}) = d(x)$; in some other cases μ could be a probability distribution, for example related to sampling. In most of what follows we shall assume that μ is simply the counting measure, but the construction generalize to the case of general measures μ. One defines the space of square-integrable functions

$$\mathbb{L}^2(G) := \{f : G \to \mathbb{R} \text{ s.t. } ||f||_2^2 := \sum_{x\in G} |f(x)|^2 \mu(\{x\}) < +\infty\}, \tag{1}$$

which is a Hilbert space with the associated inner product

$$\langle f, g\rangle = \sum_{x\in G} f(x)g(x)\mu(\{x\}).$$

There is a natural random walk on G, given by $P = D^{-1}W$. This Markov chain is necessarily reversible, and thus it is conjugate, together with its powers, to a symmetric operator T:

$$\begin{aligned} T^t &= D^{\frac{1}{2}} P^t D^{-\frac{1}{2}} = (D^{-\frac{1}{2}} W D^{-\frac{1}{2}})^t \\ &= (I - \mathcal{L})^t = \sum_i (1-\lambda_i)^t \xi_i(\cdot)\xi_i(\cdot). \end{aligned} \tag{2}$$

where

$$\mathcal{L} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} \tag{3}$$

is the normalized Laplacian, and $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_i \leq \ldots$ are the eigenvalues of $\mathcal{L}$ and $\{\xi_i\}$ the corresponding eigenvectors: $\mathcal{L}\xi_i = \lambda_i \xi_i$. Clearly $P^t = D^{\frac{1}{2}} T^t D^{-\frac{1}{2}}$, and hence studying T is equivalent, as far as spectral properties are concerned, to studying P.

3.1.2 FOURIER ANALYSIS ON GRAPHS AND MANIFOLDS

The eigenvectors $\{\xi_i\}$ of $\mathcal{L}$ form an orthonormal basis for $\mathbb{L}^2(G)$ and hence can be used for analysis and approximation of any such function. This has lead to novel algorithms in dimensionality reduction, classification and learning (Belkin and Niyogi, 2003c; Niyogi and Belkin, 2001; Lafon, 2004; Coifman and Lafon, 2004a; Coifman et al., 2005a; Mahadevan and Maggioni, 2005).

The Laplacian is positive semi-definite, and hence defines a (quasi-)norm, related to a notion of smoothness, and an associated function space of smooth functions on which it is naturally defined:

$$||f||^2_{\mathcal{H}^1} = \sum_x |f(x)|^2 d(x) + \sum_{x \sim y} |f(x) - f(y)|^2 w(x,y) = ||f||_2 + \langle f, \mathcal{L}f \rangle . \tag{4}$$

The first term in this norm controls the size of the function f, and the second term controls the size of the gradient. This norm measures the smoothness of f, in the sense that f is not smooth if its $\mathbb{L}^2$-norm and/or the $\mathbb{L}^2$-norm of its gradient are large.

For a function f on G, its gradient is defined as the function on the edges of G given by $\nabla f(i,j) = w(i,j)(f(i) - f(j))$ if there is an edge e connecting i to j, 0 otherwise. The gradient can thus be interpreted as a function on the edges, and the norm in $\mathcal{H}^1$ can be rewritten as

$$||f||^2_{\mathcal{H}^1} = ||f||_2 + ||\nabla f||_2 ,$$

where the second $\mathbb{L}^2$-norm is on the space of edges with measure w. The definitions in the case of a manifold are completely analogous[2].

For a function $f \in \mathbb{L}^2(G)$, one can write $f = \sum_i \langle f, \xi_i \rangle \xi_i$, with convergence at least in the $\mathbb{L}^2$-norm, as a generalization of Fourier series expansions. Ideally the sum can be restricted to few terms for a given precision. In the case of linear approximation, we restrict ourselves in choosing the coefficients $\alpha_i(f) := \langle f, \xi_i \rangle$ among the first I, in which case efficient approximation is possible if $|\alpha_i(f)|$ decays rapidly with i. It turns out that the rate of decay of $\{\alpha_i(f)\}$ is related to the global smoothness properties of f (the greater the smoothness, the faster the decay). In the case of nonlinear approximation, we allow ourselves to compute all the coefficients $\alpha_i(f)$ and pick the largest among all of them.

However it is well known that if f does not have uniform smoothness everywhere, the approximation by eigenfunctions is poor not only in regions of lesser smoothness, but the poor approximation spills widely to regions of smoothness as well. See for example Figure 3. This lack of localization can be avoided by using carefully designed kernels, in particular it is avoided with the multiscale constructions we discuss in the next section.

3.2 Multiscale Analysis of Functions and Markov Processes

In this section, we view multiscale analysis from two related, but nonetheless distinct perspectives. The first is approximation of functions, the other is approximation of (Markov)

2. One only needs some care on the boundary: boundary conditions need to be specified, and the behavior of functions in $\mathcal{H}^1$ are affected by them.

processes. Multiscale and wavelet analysis of functions is well understood in Euclidean space (Daubechies, 1992; Mallat, 1998), and it is motivated by the need for studying functions (or signals) that have different behavior at different locations at different scales.

Regarding multiscale analysis of Markov processes, many applications require representing time series data at multiple levels of resolution, for example robot navigation (Theocharous et al., 2001), sensor networks (Xu et al., 2004), and social network analysis (McCallum et al., 2004). Given the inherent uncertainty in such domains, a computational approach that automatically abstracts stochastic processes at multiple levels of abstraction is highly desirable. Parametric graphical models have been extensively studied in this regard. In the uncontrolled setting, *hierarchical hidden Markov models* are one of a general family of multi-resolution stochastic time-series models (Fine et al., 1998; Mahadevan et al., 2004). A central limitation of such parametric multi-resolution stochastic time series models is that the overall structure describing the evolution of the process at multiple levels of abstraction is usually assumed to be given. In the work on hierarchical HMMs, for example, it is typically assumed that a human hand-codes the tree structure, and most research focuses on efficient parameter estimation techniques, which are variants of expectation maximization (EM) (Dempster et al., 1977; Neal and Hinton, 1999). Similarly, in much work on hierarchical reinforcement learning, the overall decomposition mapping high-level tasks into subtasks is assumed to be given, and the focus is on parameter estimation methods such as SMDP Q-learning (Barto and Mahadevan, 2003).

In contrast, the diffusion wavelet framework provides a general and powerful way of learning multiscale structures, relieving a human of having to hand code a suitable hierarchical structure. Let P^π represent the one-step evolution of a stochastic Markov process, where P^π_{ij} represents the probability of transition from state i to state j, if the policy π is followed. This policy can be viewed as a random walk on a graph associated with the state space, where vertices are the states. Longer random walks on the graph are represented by powers of the one-step Markov process; diffusion wavelets encode the multi-step process by compactly encoding all dyadic powers of the one-step process.

Consider a random walk P on a set like the one represented in Figure 4. It is clear that several structures exist at different scales, and the set is "lumpable" at different scales. How does this affect the random walk? If we consider different powers of P, corresponding to different "time-scales" of the random walk, then we would expect to construct, for each power of P, a compressed Markov random walk on a set of "lumps" of the set. This is one of the outcomes of the multiscale diffusion analysis introduced in (Coifman and Maggioni, 2004; Maggioni et al., 2005a): it yields a multiscale set of lumps (more precisely, of probability distributions or functions on the set), and a compressed Markov chain acting at each scale on the set of lumps corresponding to that scale, in a consistent way, as discussed below. In Figure 4 we show some of the scales and the corresponding lumps, as well as the matrices representing the Markov chains at different scales acting on such lumps.

It is easy to imagine how this simple example could be extended to much more general situations, for example the case of dynamical systems with several time scales, or problems in reinforcement learning where there exist bottleneck structures at different time or space scales in the action-state space (Barto and Mahadevan, 2003).

3.2.1 DIFFUSION WAVELETS

Diffusion wavelets (Coifman and Maggioni, 2004; Bremer et al., 2004) enable a fast multiscale analysis of functions on a manifold or graph, generalizing wavelet analysis and associated signal processing techniques (such as compression or denoising) to functions on manifolds and graphs. They allow the efficient and accurate computation of high powers of

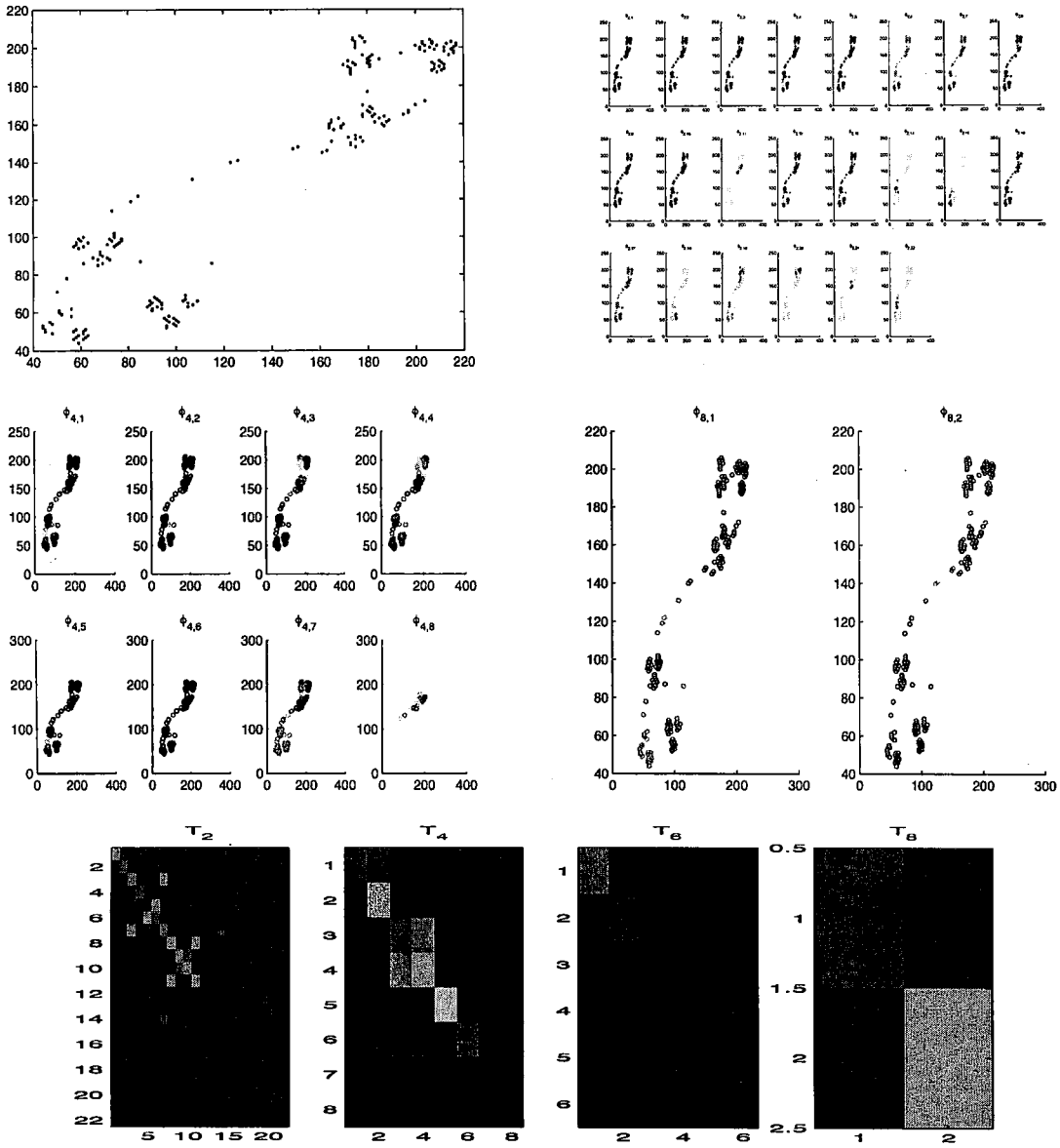


Figure 4: Top left: a multiscale lumpable set. This can be though of samples from a continuous state space of several rooms connected by corridors. Top right and second row: lumps (localized functions on the set) at increasing coarser scales, whose values are represented by colors on the set. Bottom row: compressed powers T^{2^j} of the random walk T at different time-scales.

a Markov chain P on the manifold or graph, including direct computation of the Green's function (or fundamental matrix) of the Markov chain, $(I - P)^{-1}$, which can be used to solve Bellman's equation. Space constraints permit only a brief description of the construction of diffusion wavelet trees. More details are provided in (Coifman and Maggioni, 2004;

Bremer et al., 2004). We refer the reader not familiar with wavelets to the appendix, where we briefly discuss classical multiresolution analysis.

A multiresolution decomposition of the functions on the graph is a family of nested subspaces $V_0 \supseteq V_1 \supseteq \cdots \supseteq V_j \supseteq \ldots$ spanned by orthogonal bases of diffusion scaling functions Φ_j. If we interpret T^t as an operator on functions on the graph, then V_j is defined as the numerical range, up to precision ϵ, of $T^{2^{j+1}-1}$, and the scaling functions are smooth bump functions with some oscillations, at scale roughly 2^{j+1} (measured with respect to geodesic distance). The orthogonal complement of V_{j+1} into V_j is called W_j, and is spanned by a family of orthogonal diffusion wavelets Ψ_j, which are smooth localized oscillatory functions at the same scale. The detailed construction of the multiresolution analysis is quite non-trivial, and we refer the reader interested in all the technical and algorithmic details to the paper by Coifman and Maggioni (2004). Here, we summarize the main steps below.

3.2.2 CONSTRUCTION OF DIFFUSION WAVELETS

Here and in the rest of this section we will use the notation $[L]_{B_1}^{B_2}$ to indicate the matrix representing the linear operator L with respect to the basis B_1 in the domain and B_2 in the range. A set of vectors B_1 represented on a basis B_2 will be written in matrix form $[B_1]_{B_2}$, where the rows of $[B_1]_{B_2}$ are the coordinates of the vectors B_1 in the coordinate system defined by B_2.

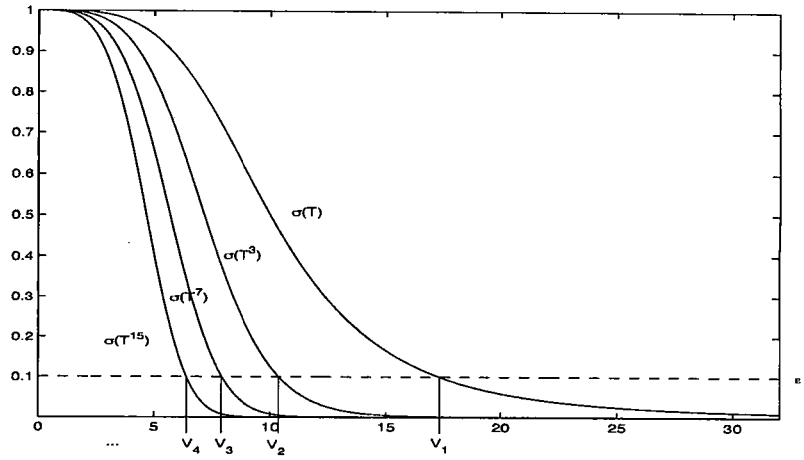


Figure 5: Spectra of powers of a diffusion operator T and corresponding multiscale eigenspace decomposition.

The input to the algorithm is a "precision" parameter $\epsilon > 0$, and a weighted graph (G, E, W). We assume that G is strongly connected and "local" (in the sense that each vertex is connected to a small number of vertices). The construction is based on using the natural random walk $P = D^{-1}W$ on a graph (where D is the out-degree matrix if the graph is directed) which we assume aperiodic. We use the powers of P to "dilate", or "diffuse" functions on the graph, and then define an associated coarse-graining of the graph. Observe that in many cases of interest P is a sparse matrix. We usually *normalize* P and consider $T = \Pi^{-1}P\Pi$, where Π is the asymptotic distribution of P. Because of the hypotheses on P, Π exists is unique and can be chosen to be a strictly positive distribution by the

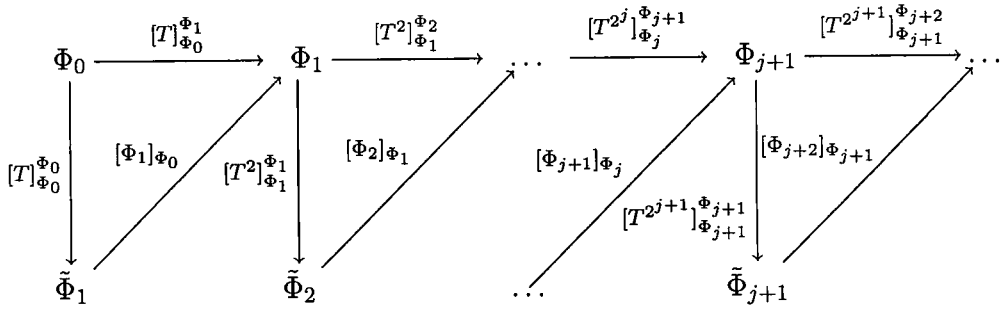


Figure 6: Figure shows downsampling, orthogonalization and operator compression. (All triangles are commutative by construction)

Perron-Fröbenius Theorem (Kemeny and Snell, 1976). We refer the reader to (Chung, 2006; Zhou et al., 2005) for details on the directed graph Laplacian. If G is undirected, P is reversible, $\Pi = D^{\frac{1}{2}}$, and T is symmetric. In the other cases, when T is not symmetric, in what follows any statement regarding (or implied by) the spectral decomposition of T should be disregarded. We take powers to obtain

$$T^t = (\Pi^{-\frac{1}{2}} P \Pi^{-\frac{1}{2}})^t = \Pi^{-\frac{1}{2}} P^t \Pi^{-\frac{1}{2}} = (I - \mathcal{L})^t = \sum_{i \geq 0} (1 - \lambda_i)^t \xi_i(\cdot) \xi_i(\cdot) \tag{5}$$

where $\{\lambda_i\}$ and $\{\xi_i\}$ are the eigenvalues and eigenfunctions of the Laplacian. Hence the eigenfunctions of T^t are again ξ_i and the i^{th} eigenvalue is $(1 - \lambda_i)^t$.

In what follows, we assume that T is a sparse matrix, and that the numerical rank of the powers of T decays rapidly with the power. For example a desirable situation is when the number of singular values of T^t larger than ϵ is smaller than $2^{-\gamma t}$.

A diffusion wavelet tree consists of orthogonal diffusion scaling functions Φ_j that are smooth bump functions, with some oscillations, at scale roughly 2^j (measured with respect to geodesic distance), and orthogonal wavelets Ψ_j that are smooth localized oscillatory functions at the same scale. The scaling functions Φ_j span a subspace V_j, with the property that $V_{j+1} \subseteq V_j$, and the span of Ψ_j, W_j, is the orthogonal complement of V_j into V_{j+1}. This is achieved by using the dyadic powers T^{2^j} as "dilations", to create smoother and wider (always in a geodesic sense) "bump" functions (which represent densities for the symmetrized random walk after 2^j steps), and orthogonalizing and downsampling appropriately to transform sets of "bumps" into orthonormal scaling functions.

We now describe the multiscale construction in detail. It may be useful to compare the description that follows with the diagram in Figure 6. T is initially represented on the basis $\Phi_0 = \{\delta_k\}_{k \in G}$; we consider the columns of T, interpreted as the set of functions $\tilde{\Phi}_1 = \{T\delta_k\}_{k \in G}$ on G. We use a local multiscale orthogonalization procedure, described in (Coifman and Maggioni, 2004), to carefully orthonormalize these columns to get a basis $\Phi_1 = \{\varphi_{1,k}\}_{k \in G_1}$ (G_1 is *defined* as this index set), written with respect to the basis Φ_0, for the range of T up to precision ϵ. This information is stored in the sparse matrix $[\Phi_1]_{\Phi_0}$. This yields a subspace that we denote by V_1. Essentially Φ_1 is a basis for the subspace V_1 which is ϵ-close to the range of T, and with basis elements that are well-localized. Moreover, the elements of Φ_1 are coarser than the elements of Φ_0, since they are the result of applying the "dilation" T once. Obviously $|G_1| \leq |G|$, but this inequality may already be strict since the numerical range of T may be approximated, within the specified precision ϵ, by a subspace of smaller dimension. Whether this is the case or not, we have computed the sparse matrix $[T]^{\Phi_1}_{\Phi_0}$, a representation of an ϵ-approximation of T with respect to Φ_0 in the domain and

$\{\Phi_j\}_{j=0}^{J}, \{\Psi_j\}_{j=0}^{J-1}, \{[T^{2^j}]_{\Phi_j}^{\Phi_j}\}_{j=1}^{J} \leftarrow$ `DiffusionWaveletTree` $([T]_{\Phi_0}^{\Phi_0}, \Phi_0, J, \mathrm{SpQR}, \epsilon)$

// Input:
// $[T]_{\Phi_0}^{\Phi_0}$: a diffusion operator, written on the o.n. basis Φ_0
// Φ_0 : an orthonormal basis which ϵ-spans V_0
// J : number of levels to compute
// SpQR : a function compute a sparse QR decomposition, template below.
// ϵ: precision

// Output:
// The orthonormal bases of scaling functions, Φ_j, wavelets, Ψ_j, and
// compressed representation of T^{2^j} on Φ_j, for j in the requested range.

for $j = 0$ **to** $J-1$ **do**

$[\Phi_{j+1}]_{\Phi_j}$, $[T]_{\Phi_0}^{\Phi_1} \leftarrow$`SpQR`$([T^{2^j}]_{\Phi_j}^{\Phi_j}, \epsilon)$

$T_{j+1} := [T^{2^{j+1}}]_{\Phi_{j+1}}^{\Phi_{j+1}} \leftarrow [\Phi_{j+1}]_{\Phi_j}[T^{2^j}]_{\Phi_j}^{\Phi_j}[\Phi_{j+1}]_{\Phi_j}^{*}$

$[\Psi_j]_{\Phi_j} \leftarrow \mathrm{SpQR}(I_{\langle\Phi_j\rangle} - [\Phi_{j+1}]_{\Phi_j}[\Phi_{j+1}]_{\Phi_j}^{*}, \epsilon)$

end

Function template for sparse QR factorization:
$Q, R \leftarrow$ `SpQR` (A, ϵ)

// Input:
// A: sparse $n \times n$ matrix
// ϵ: precision

// Output:
// Q, R matrices, possibly sparse, such that $A =_\epsilon QR$,
// Q is $n \times m$ and orthogonal,
// R is $m \times n$, and upper triangular up to a permutation,
// the columns of Q ϵ-span the space spanned by the columns of A.

Figure 7: Pseudo-code for construction of a Diffusion Wavelet Tree.

Φ_1 in the range. We can also represent T in the basis Φ_1: with the notation above this is the matrix $[T]_{\Phi_1}^{\Phi_1}$. We compute $[T^2]_{\Phi_1}^{\Phi_1} = [\Phi_1]_{\Phi_0}[T^2]_{\Phi_0}^{\Phi_0}[\Phi_1]_{\Phi_0}^{T}$. If T is self-adjoint, this is equal to $[T]_{\Phi_0}^{\Phi_1}([T]_{\Phi_0}^{\Phi_1})^*$, which has the advantage that numerical symmetry is forced upon $[T^2]_{\Phi_1}^{\Phi_1}$.

It is now clear how to proceed: we look at the columns of $[T^2]_{\Phi_1}^{\Phi_1}$, which are $\tilde{\Phi}_2 = \{[T^2]_{\Phi_1}^{\Phi_1}\delta_k\}_{k\in G_1}$. By unraveling the notation, these are functions $\{T^2\varphi_{1,k}\}_{k\in G_1}$, up to the precision ϵ. Once again we apply a local orthonormalization procedure to this set of functions, obtaining an orthonormal basis $\Phi_2 = \{\varphi_{2,k}\}_{k\in G_2}$ for the range of T_1^2 (up to precision ϵ), and also for the range of T_0^3 (up to precision 2ϵ). Observe that Φ_2 is naturally written with respect to the basis Φ_1, and hence encoded in the matrix $[\Phi_2]_{\Phi_1}$. Moreover, depending on the decay of the spectrum of T, $|X_2|$ is in general a fraction of $|X_1|$. The

matrix $[T^2]_{\Phi_1}^{\Phi_2}$ is then of size $|G_2| \times |G_1|$, and the matrix $[T^4]_{\Phi_2}^{\Phi_2} = [T^2]_{\Phi_1}^{\Phi_2}([T^2]_{\Phi_1}^{\Phi_2})^*$, a representation of T^4 acting on Φ_2, is of size $|G_2| \times |G_2|$.

After j iterations in this fashion, we will have a representation of T^{2^j} onto a basis $\Phi_j = \{\varphi_{j,k}\}_{k \in G_j}$, encoded in a matrix $T_j := [T^{2^j}]_{\Phi_j}^{\Phi_j}$. The orthonormal basis Φ_j is represented with respect to Φ_{j-1}, and encoded in the matrix $[\Phi_j]_{\Phi_{j-1}}$. We let $\tilde{\Phi}_j = T_j \Phi_j$ We can represent the next dyadic power of T on Φ_{j+1} on the range of T^{2^j}. Depending on the decay of the spectrum of T, we expect $|G_j| << |G|$, in fact in the ideal situation the spectrum of T decays fast enough so that there exists $\gamma < 1$ such that $|X_j| < \gamma |X_{j-1}| < \cdots < \gamma^j |X|$. This corresponds to downsampling the set of columns of dyadic powers of T, thought of as vectors in $\mathcal{L}^2(G)$. The hypothesis that the rank of powers of T decreases guarantees that we can downsample and obtain coarser and coarser lattices in this spaces of columns.

While Φ_j is naturally identified with the set of Dirac δ-functions on G_j, we can extend these functions living on the "compressed" (or "downsampled") graph G_j to the whole initial graph G by writing

$$[\Phi_j]_{\Phi_0} = [\Phi_j]_{\Phi_{j-1}} [\Phi_{j-1}]_{\Phi_0} = \cdots = [\Phi_j]_{\Phi_{j-1}} [\Phi_{j-1}]_{\Phi_{j-2}} \cdot \cdots \cdot [\Phi_1]_{\Phi_0} [\Phi_0]_{\Phi_0} \,. \tag{6}$$

Since every function in Φ_0 is defined on G, so is every function in Φ_j. Hence any function on the compressed space G_j can be extended naturally to the whole G. In particular, one can compute low-frequency eigenfunctions on G_j in compressed form, and then extend them to the whole G. The elements in Φ_j are at scale $T^{2^{j+1}-1}$, and are much coarser and "smoother", than the initial elements in Φ_0, which is why they can be represented in compressed form. The projection of a function onto the subspace spanned by Φ_j will be by definition an approximation to that function at that particular scale.

There is an associated fast scaling function transform: suppose we are given f on G and want to compute $\langle f, \varphi_{j,k} \rangle$ for all scales j and corresponding "translations" k. Being given f means we are given $(\langle f, \varphi_{0,k} \rangle)_{k \in G}$. Then we can compute $(\langle f, \varphi_{1,k} \rangle)_{k \in G_1} = [\Phi_1]_{\Phi_0} (\langle f, \varphi_{0,k} \rangle)_{k \in G}$, and so on for all scales. The sparser the matrices $[\Phi_j]_{\Phi_{j-1}}$ (and $[T]_{\Phi_j}^{\Phi_j}$), the faster this computation. This generalizes the classical scaling function transform. Wavelet bases for the spaces W_j can be built analogously by factorizing $I_{V_j} - Q_{j+1} Q_{j+1}^*$, which is the orthogonal projection on the complement of V_{j+1} into V_j. The spaces can be further split to obtain wavelet packets (Bremer et al., 2004). The wavelets can be considered as high-pass filters, in the sense that they capture the detail lost from going from V_j to V_{j+1}, and also in the sense that their expansion in terms of eigenfunctions of the Laplacian essentially only involves eigenfunctions corresponding to eigenvalues in $[\epsilon^{-2^j-1}, \epsilon^{-2^{j+1}-1}]$. In particular their Sobolev norm, or smoothness, is controlled.

In the same way, any power of T can be applied efficiently to a function f. Also, the Green's function $(I-T)^{-1}$ can be applied efficiently to any function, since it can represented as product of dyadic powers of T as in (9), each of which can be applied efficiently. We are at the same time compressing the powers of the operator T and the space X itself, at essentially the optimal "rate" at each scale, as dictated by the portion of the spectrum of the powers of T which is above the precision ϵ.

Observe that each point in G_j can be considered as a "local aggregation" of points in G_{j-1}, which is completely dictated by the action of the operator T on functions on G: the operator itself is dictating the geometry with respect to which it should be analyzed, compressed or applied to any vector. The algorithm is summarized in Figure 7.

A Fast Diffusion Wavelet Transform expands, in $\mathcal{O}(n)$ computations (where n is the number of vertices) any function in the wavelet, or wavelet packet, basis, and efficiently searches for the most suitable basis set. Diffusion wavelets and wavelet packets are a very efficient tool for representation and approximation of functions on manifolds and graphs

(Coifman and Maggioni, 2004; Bremer et al., 2004), generalizing to these general spaces the nice properties of wavelets that have been so successfully applied in Euclidean spaces.

Diffusion wavelets allow computing $T^{2^k} f$ for any fixed f, in order $\mathcal{O}(kn)$. This is non-trivial because while the matrix H is sparse, large powers of it are not, and the computation $T^{2^k} f = T \cdot T \cdots (T(Tf)) \dots)$ involves 2^k matrix-vector products. As a notable consequence, this yields a fast algorithm for computing the Green's function, or fundamental matrix, associated with the Markov process T, via $(I - T^1)^{-1} f = \sum_{k \geq 0} T^k = \prod_{k \geq 0} (I + T^{2^k}) f$. In a similar way one can compute $(I - P)^{-1}$. For large classes of Markov chains we can perform this computation in time $\mathcal{O}(n)$, in a direct (as opposed to iterative) fashion. This is remarkable since in general the matrix $(I - T^1)^{-1}$ is full and only writing down the entries would take time $\mathcal{O}(n^2)$. This multiscale compression scheme makes it possible to efficiently represent $(I - T^1)^{-1}$ in compressed form, taking advantage of the smoothness of the entries of the matrix. This is discussed in general in (Coifman and Maggioni, 2004). In (Maggioni and Mahadevan, 2005) we use this approach to develop an efficient direct policy evaluation algorithm, which we describe in more detail below.

Example 4 (Multiscale compression of a simple Markov chain) *To illustrate the multiscale analysis enabled by diffusion wavelets, it helps to see the results of the analysis on a simple example. We consider the Markov chain on 4 states $\{a, b, c, d\}$:*

$$T = \begin{pmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.2 & 0.75 & 0.05 & 0 \\ 0 & 0.05 & 0.75 & 0.2 \\ 0 & 0 & 0.2 & 0.8 \end{pmatrix}.$$

This chain has a "bottleneck" between the set of states $\{a, b\}$ and the set of states $\{c, d\}$, as the small weight (transition probability) on the edge $\overline{bc}$ indicates. We fix a precision $\epsilon = 10^{-10}$. See Figure 8 for the discussion that follows. The scaling functions Φ_0 are simply $\{\delta_a, \delta_b, \delta_c, \delta_d\}$. We apply T to Φ_0 and orthonormalize to get Φ_1 (Figure 8). Each function in Φ_1 is an "abstract-state", i.e. a linear combination of the original states. We represent T^2 on Φ_1, to get a matrix T_2, apply to Φ_1 and orthonormalize, and so on. At scale 5 we have the basis Φ_5 and the operator T_5, representing T^{2^5} on Φ_5. At the next level, we obtain Φ_7, which is only two dimensional, because $T_5 \Phi_5$ has ϵ-rank 2 instead of 4: of the 4 "abstract-states" $T_5 \Phi_5$, only two of them are at least ϵ-independent. Observe the two scaling functions in Φ_6 are approximately the asymptotic distribution and the function which distinguishes between the two clusters $\{a, b\}$ and $\{c, d\}$. Then T_6 represents T^{2^6} on Φ_7 and is a 2 by 2 matrix. At scale 10, Φ_{10} is one-dimensional, and is simply the top eigenvector of T (represented in compressed form, on the basis Φ_8), and the matrix T_9 is 1 by 1 and is just the top eigenvalue, 1, of T.

Already in this simple example we see that the multiscale analysis generates a sequence of Markov chains, each corresponding to a different time scale (i.e. power of the original Markov chain), represented on a set of scaling functions ("aggregates of states") in compressed form. This approach clearly provides a new way to approach the problem of hierarchical reinforcement learning (Barto and Mahadevan, 2003), although for reasons of space we cannot get into the details in this paper.

Figures 1 and 2 discussed above show several diffusion scaling functions and wavelets corresponding to diffusion operators T that are the natural random walk, with reflecting boundary conditions, on the space.

3.2.3 OTHER APPLICATIONS OF DIFFUSION WAVELETS

In brief passing, we note that diffusion wavelets and wavelet packets have been shown to be an efficient tool for representation and approximation of functions on manifolds and

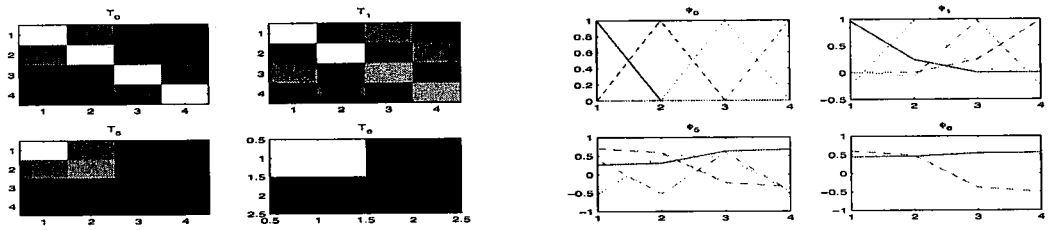


Figure 8: The four panels on the top display matrices $[T^{2^j}]_{\Phi_j}^{\Phi_j}$ representing compressed dyadic powers of T, with gray level representing entry values. Observe that the size of the matrix decays, since so does the rank of the powers of T. The four panels on the bottom illustrate some scaling function bases on the 4-state Markov chain.

graphs (Coifman and Maggioni, 2004; Bremer et al., 2004), generalizing to these general spaces the wavelets that have so successfully employed for similar tasks in Euclidean spaces. Applications include the analysis of networks, graphs, document corpora (Coifman and Maggioni, 2004, 2005), nonlinear and anisotropic image denoising, learning tasks, and value function approximation (Mahadevan and Maggioni, 2005; Maggioni and Mahadevan, 2005). We focus on their application to MDPs for the rest of the paper.

4. Applications of Diffusion Wavelets to Markov Decision Processes

In this section, we describe how to apply multiscale analysis ideas discussed above to the analysis of value functions and multiscale representation of transition matrices in MDPs. There are several motivations for such an analysis.

(i) *Multiscale view.* The multiscale approach provides a deeper understanding of the intrinsic regularities underlying an MDP, by generating a hierarchy of processes of decreasing complexity. The capacity for understanding and solving problems by hierarchical subdivision into sub-problems is one of the hallmarks of human intelligence.

(ii) *Approximation efficiency.* It is important to be able to efficiently represent the value function (or approximations of it). This is already clear in large discrete state spaces, and cannot be avoided in the case of continuous state spaces. In the companion paper (Mahadevan and Maggioni, 2006) we have discussed approximation through the eigenfunctions of the Laplacian, whose advantages over standard bases for value function approximation include adaptivity to the geometry of the state-space, automatic construction, and good approximation properties for globally smooth value functions. The multiscale bases we discuss in this second part share these same advantages, but guarantee better approximation properties for value functions which are not uniformly smooth, and are much more agile and tunable to geometric and functional properties of the problem at hand.

(iii) *Bellman's equation.* At each iteration in a policy iteration algorithm, Bellman's equation needs to be solved. The multiscale construction we discuss enables the efficient solution of such an equation: the inverse operator needed to solve Bellman's can be expressed and computed efficiently in this basis.

4.1 Multiscale analysis associated with a Markov Decision Process

Let $M = (\mathcal{S}, A, P^a_{ss'}, R^a_{ss'})$ be a discrete[3] Markov decision process. The relevant standard definitions are given in the first paper (Mahadevan and Maggioni, 2006). Let π be a policy, and denote by $\pi_s(a)$ the probability of choosing action a in state s under π. There is a natural Markov process P^π defined on $\mathcal{S}$, defined by

$$P^\pi(s_i \to s_j) = \sum_{a \in A} P^a_{s_i,s_j} \pi_{s_i}(a)\,. \tag{7}$$

We assume that P^π is reversible, [4] and let T^π be the self-adjoint operator similar to P^π. Associated with T^π, there is an associated multiscale analysis: let $\{\Phi^\pi_j\}$ and $\{\Psi^\pi_j\}$ be the associated diffusion scaling functions and wavelets, respectively, and let T^π_j be the compressed representation of $(T^\pi)^{2^j}$. We interpret $\{\Phi^\pi_j\}$ as the vertices of a coarser graph, on which the Markov process T^π_j is a compressed version of $(T^\pi)^{2^j}$, or a "coarsened" version of T^π, "viewed at time-scale 2^j". Hence each basis element of Φ^π_j is a super-state of the MDPs, corresponding to the policy π, and the compressed matrix $(T^\pi)^{2^j}$ expresses the transition probabilities among these super-states, for 2^j steps of the Markov chain. See Example 4 discussed above. The wavelets $\{\Psi_j\}$ form an orthonormal basis for $\mathbb{L}^2(\mathcal{S})$, of functions localized at different scales. The value function V^π (or an estimate thereof) and the rewards R can be expanded onto this basis.

4.2 Multiscale value function approximation

Linear approximation methods for solving MDPs, in particular LSPI (Lagoudakis and Parr, 2003), are described in the companion paper (Mahadevan and Maggioni, 2006). One of the key ingredients is the projection of the state-action value function Q^π, for a policy π, onto a set of basis functions Φ for some subspace $\langle \Phi \rangle \subseteq \mathbb{L}^2(\mathcal{S}, \mathcal{A})$. The Bellman equation

$$Q^\pi = (I - \gamma P^\pi)^{-1} R$$

can then be solved by projecting both sides of the equation onto the basis elements. One of the advantages sought is a reduction of dimensionality of the problem: one would like to select a small number of functions Φ such that $||Q^\pi - P_{\langle \Phi \rangle} Q^\pi||$ is small. Here P_V denotes the orthogonal projection onto the subspace V. In the companion paper we considered the case where Φ is a subset of eigenfunctions of a Laplacian on a graph associated with the state space of the MDP. Here, motivated by the increased approximation properties of multiscale wavelets, we consider the case where Φ is a set of diffusion wavelets. The diffusion wavelet basis has a natural ordering, from coarse to fine, as follows. The diffusion scaling functions at the coarsest scale, spanning V_J come first. They are followed by the diffusion wavelets in W_{J-1}, W_{J-2} and so on. Hence if the number of basis functions to be used is set to be equal to k, the basis functions used will be the first k extracted from Φ according to the ordering just described. Observe that the scaling functions are essentially like "Gaussian bumps" adapted to the geometry of the set.

The representation policy iteration algorithm is summarized in Figure 9. It is completely analogous to the algorithm in the companion paper, but with diffusion wavelets replacing eigenfunctions of the Laplacian.

3. The case of continuous MDPs could be treated similarly, but there are several technical points which we prefer not to address here.
4. As remarked above, the construction of diffusion wavelets is easily generalized to the non-reversible, non self-adjoint case.

RPI Algorithm $(T, N, \epsilon, P, \mathcal{O}, \delta)$:

// T: Number of initial random walk trials
// N: Maximum length of each trial
// ϵ : Convergence condition for policy iteration
// P: Number of proto-value basis functions to use
// $\mathcal{O}$: Type of graph operator used
// δ: Parameter for basis adaptation

Representation Learning Phase

1. Perform a random walk of T trials, each of maximum N steps, and store the states visited in $\mathcal{S}$.
2. Build an undirected weighted graph G from $\mathcal{S}$, in one of the ways described in the first paper (Mahadevan and Maggioni, 2006). For example, connect state i to state j if it is one of its k "nearest" neighbors, assigning a weight $w(i,j) = e^{-(\frac{\|s_i - s_j\|}{\delta})^2}$. Alternatively, connect state i to state j with a weight of 1 if the pair (i,j) form temporally successive states $\in \mathcal{S}$.
3. Construct the diffusion operator T associated with the graph built in the previous step. Construct the diffusion wavelets associated with T, ordered from coarse to fine, and including the coarsest scale scaling functions. Collect the first k of them as columns of the basis function matrix Φ, a $|\mathcal{S}| \times k$ matrix. The encoding of a state action pair (s,a) is given as $e_a \otimes \phi(s)$, where e_a is the unit vector corresponding to action a, $\phi(s)$ is the s^{th} row of Φ, and $\otimes$ is the tensor product.

Control Learning Phase

1. Initialize $w^0 \in \mathbb{R}^k$ to a random vector.
2. **Repeat** the following steps:
   (a) Set $i \leftarrow i+1$. For each transition $(s_t, a_t, s'_t, a'_t, r_t) \in \mathcal{D}$, compute low rank approximations of matrix A and b as follows:

$$\begin{aligned} \tilde{A}^{t+1} &= \tilde{A}^t + \phi(s_t, a_t)\left(\phi(s_t, a_t) - \gamma\phi(s'_t, a'_t)\right)^T \\ \tilde{b}^{t+1} &= \tilde{b}^t + \phi(s_t, a_t) r_t \end{aligned}$$

   (b) Solve the system $\tilde{A}w^i = \tilde{b}$
   (c) **Optional basis adaption step:** Modify the set of proto-value functions Φ by discarding those whose coefficients are smaller than $\delta\epsilon$.
3. **until** $\|w^i - w^{i+1}\|^2 \leq \epsilon$.
4. Return $\hat{Q}^* \approx \sum_i w^i \Phi$ as the approximation to the optimal value function.

Figure 9: Pseudo-code of the representation policy iteration algorithm with diffusion wavelets for discrete Markov decision processes.

4.3 Direct multiscale solution of Bellman's equation

An efficient algorithm for the direct (vs. iterative) solution of Bellman's equation is associated with the multiscale analysis. This algorithm can be immediately incorporated in exact policy iteration (PI) algorithms (Lagoudakis and Parr, 2003; Mahadevan, 2005b; Mahadevan and Maggioni, 2005; Maggioni and Mahadevan, 2005), and even in approximate policy iteration, under some additional conditions. The Bellman equation (which was introduced in the first paper) usually involves the solution of a (typically sparse) linear system

$$V^\pi = R + \gamma P^\pi V^\pi \tag{8}$$

of size $|S|$, where S is the state space. Here V^π is the unknown (the value function), R and P^π are given (for each step in policy iteration). We can divide the approaches for solving this equation in two families: direct or iterative.

(a) *Direct solution* involves the computation of the inverse $(I - \gamma P^\pi)^{-1}$, in some form that can be directly applied to any vector R. While the matrix $I - \gamma P$ is usually sparse, its inverse is in general a full matrix, and its computation usually takes time $\mathcal{O}(|S|^3)$. This is in general infeasible on large problems. However it presents certain computational advantages: very stable algorithms exist, computations can be done to very high precision, and once the inverse matrix has been computed, it can be applied very rapidly to any reward vector R, which is an advantage in problems where there are multiple rewards.

(a) *Iterative solution* involves iterative techniques, such as conjugate gradient or value iteration, which compute $(I - \gamma P^\pi)^{-1} R$ for a given R. These techniques have worst case complexity $\mathcal{O}(|S|^2)$ for sparse transition matrices, $\mathcal{O}(|S|)$ when the problem is well-conditioned and only low-precision is required. No structure of the problem or of previous solutions is constructed, so the computation has to be repeated for different R's.

We now introduce a fundamentally different approach, based on diffusion wavelets, that can yield a *direct* solution in time $\mathcal{O}(|S|)$, for certain classes of transition matrices P^π. This is surprising because, as we observed before, just writing down all the entries of the full inverse matrix would take time $\mathcal{O}(|S|^2)$. However, observe that the goal is not to compute the entries of this matrix, but to compute a structure that enables the rapid computation of $(I - \gamma P^\pi)^{-1} R$ for any given R. The proposed approach constructs a (multiscale) structure given which it is possible to perform this computation in only $\mathcal{O}(|S|)$ operations. Related ideas are at the core of the Fast Fourier Transform, which is exactly a full matrix multiplication by a vector, and hence would seem to necessarily require $\mathcal{O}(|S|^2)$ operations, but it can actually be performed in $\mathcal{O}(|S|)$ operations by factoring the full matrix in a product of matrices. A related principle is also behind the Fast Multipole Method (Greengard and Rokhlin, 1987) and its generalizations, which allow to evaluate the product of certain full matrices (arising from potential theory) by vectors in time $\mathcal{O}(|S|)$.

The algorithm consists of two parts:

(i) a *pre-computation* step, that depends on the structure of the state space and on the policy, and yields the multiscale analysis described above. This computation has in many cases complexity $\mathcal{O}(|S|)$, and complexity $\mathcal{O}(|S|^3)$ in general.

(ii) an *inversion* step which uses the multiscale structure built in the pre-computation step to efficiently compute the solution of Bellman's equations for a given reward function. This phase of the computation has complexity $\mathcal{O}(|S|)$ for many problems of practical importance where the transition matrix is *diffusion*-like (defined precisely below). The constants in front of this asymptotic complexity are much smaller than those in the pre-computation step.

The multiscale basis functions constructed in the pre-computation step are also very useful for approximating the value function and improve the learning process, as discussed above. The class of problems for which the complexity of the method is linear (up to logarithmic factors) includes state spaces that can be represented by a finite directed weighted graph, with all the vertices of "small" degree in which transitions are allowed only among neighboring points, and the spectrum of the transition matrix decays rapidly. The direct method we present offers several advantages.

(i) It is well-known that the number of iterations necessary for an iterative method to converge can be very large, depending on the condition number of the problem, which in general depends on the number of points, and on the precision required. Increasing precision in the direct inversion technique we propose can be done more efficiently.

(ii) When the state space and the policy are fixed, and many value functions corresponding to different rewards (tasks) need to be computed, iteration schemes do not take advantage of the common structure between the problems. In this case, the number of iterations for finding each solution is multiplied by the number of solutions sought. Our direct inversion technique efficiently encodes the common structure of the state space in the pre-computation step, and then takes advantage of this in the solution of multiple problems, thus enabling transfer across tasks in a completely novel manner.

A key advantage of the proposed approach is that direct inversion reveals interesting structure in the underlying problem.

4.3.1 MULTISCALE INVERSION

In this section we show that the multiscale construction we discussed enables a direct solution of Bellman's equation (8). The starting point are the identities

$$V^\pi = (I - \gamma P^\pi)^{-1} R = \sum_{k \geq 0} (\gamma \Pi^{-\frac{1}{2}} T^\pi \Pi^{\frac{1}{2}})^k R = \prod_{k \geq 0} (I + \gamma^{2^k} \Pi^{-\frac{1}{2}} (T^\pi)^{2^k} \Pi^{\frac{1}{2}}) R, \tag{9}$$

where $P^\pi = \Pi^{-\frac{1}{2}} T^\pi \Pi^{\frac{1}{2}}$, Π is the matrix whose diagonal is the asymptotic distribution of P, and R is the reward vector. The first identity follows by the definition of T^π, the second is the usual Neumann series expansion for the inverse, and the last identity is called the Schultz formula. The equality holds since each term of the Neumann series appears once and only once in the product, since every integer has a unique binary expansion. Observe that reordering the terms of the summation is allowed because both the sum and the product are absolutely convergent. The formulas hold for $\gamma \leq 1$ and R has no component in the kernel of $(I - \gamma P^\pi)$. Observe that since $\gamma \leq 1$ and $||P^\pi||_2 \leq 1$, the only case for which this kernel is not trivial is when $\gamma = 1$, and in this case the kernel is the span of the unique (since we assumed the state space is connected) asymptotic distribution of P^π. The product in (9) are of course finite, with $\mathcal{O}(\log |\mathcal{S}|)$ terms, for any fixed precision.

A key component in the construction of diffusion wavelets was the compression of the (quasi-)dyadic powers of the operator T^π. The product $[T^{2^{j-1}}]_{\Phi_{j-2}}^{\Phi_{j-1}} [T^{2^{j-2}}]_{\Phi_{j-3}}^{\Phi_{j-2}} \dots [T]_{\Phi_0}^{\Phi_1} [R]_{\Phi_0}$ is $T^{1+2+2^2+\dots+2^{j-1}} R = T^{2^j-1} R$, represented on Φ_{j-1}, i.e. "in compressed form". The matrices $[\Phi_{j+1}]_{\Phi_j}^*$ "un-pack" this representation back onto the basis Φ_0. In other words

$$[(T^\pi)^{2^j-1} f]_{\Phi_0} = [\Phi_1]_{\Phi_0}^* \dots [\Phi_{j-1}]_{\Phi_{j-2}}^* [T^{2^{j-1}}]_{\Phi_{j-2}}^{\Phi_{j-1}} [T^{2^{j-2}}]_{\Phi_{j-3}}^{\Phi_{j-2}} \dots [T]_{\Phi_0}^{\Phi_1} [R]_{\Phi_0}$$

To obtain $T^{2^j} f$ we only need to apply T once more. In this way the computation of $T^{2^j} f$ requires only $\mathcal{O}(j|S|)$ operations when $[T^{2^{j-1}}]_{\Phi_{j-2}}^{\Phi_{j-1}}$ contains about $\mathcal{O}(|S|)$ entries. This

cost should be compared to that of computing directly the matrix T^{2^j}, which is $\mathcal{O}(2^j|S|)$ since this matrix contains about $\mathcal{O}(2^j|S|)$ nonzero entries; this is also the cost of applying about 2^j times the matrix T to R.

Observe that the same diffusion wavelet tree can be used for this multiscale inversion for different values of the discount rate γ. In general one does not know if R does not have a component in the kernel of $(I-\gamma P^\pi)$, i.e. $R = R_1 + R_2$, with $R_2 \in \ker(I-\gamma P^\pi)$. Clearly in this case the system (8) has no solution. One usually seek the least squares solution (compare with the standard direct inversion below) of the associated normal system

$$(I-\gamma P^\pi)^*(I-\gamma P^\pi)V^\pi = (I-\gamma P^\pi)^* R \tag{10}$$

which can be re-written as

$$(I-\gamma \underbrace{(P^\pi + (P^\pi)^* - (P^\pi)^* P^\pi)}_{:=\tilde{P}^\pi} V^\pi = (I-\gamma P^\pi)^* R. \tag{11}$$

We can normalize $\tilde{P}^\pi$ as usual to obtain $\tilde{T}^\pi$. The normal equations can be solved in a multiscale fashion by applying diffusion wavelets associated with $\tilde{P}^\pi$ or $\tilde{T}^\pi$. We note in passing that the operator $\tilde{T}^\pi$ arises very naturally, but it seems it has not been much considered in the literature (see (Chung, 2006; Zhou et al., 2005) for definition and properties of a directed Laplacian).

4.3.2 COMPARISON WITH STANDARD DIRECT INVERSION AND ITERATIVE METHODS

The standard direct inversion technique consists in the explicit computation of $(I-\gamma P^\pi)^{-1}$. This typically involves the computation of the singular value decomposition of $(I-\gamma P^\pi)$: $I-\gamma P^\pi = U\Sigma V$ with U, V orthonormal and Σ diagonal, with diagonal entries $\sigma_1 \geq \cdots \geq \sigma_{|S|}$. For a fixed precision ϵ, only a partial decomposition $U_N \Sigma_N V_N$ is computed, where $N = N(\epsilon)$ is the largest n for which $\sigma_n \geq ||I-\gamma P^\pi||_2 \epsilon$. One then writes

$$(I-\gamma P^\pi)^{-1} = V^* \Sigma^{-1} U^*.$$

Very stable algorithms are available for the computation of the singular value decomposition. Optimality of the singular vectors with respect to approximation properties of the matrix itself are also well-known and are the main motivation for this technique. Unfortunately these techniques are expensive, with complexity $\mathcal{O}(|S|^2 N(\epsilon))$.

In iterative methods such as value iteration, up to $|S|$ iterations are necessary, and the cost is thus $\mathcal{O}(|S|^2)$. Our technique has cost only $\mathcal{O}(|S|)$. However in practice less than $|S|$ iterations may be needed for iterative methods to converge, especially when the problem is well-conditioned (e.g. γ far from 1), and/or low precision in the result is requested. Even when this is the case our method offers several advantages: it generates basis functions tuned to the structure of the problem that efficiently represent the value function, and once computed, this structure enables the direct fast inversion of Bellman's equation for many different rewards R. Observe that the eigenvectors of $(I-P^\pi)^{-1}$, at least when P^π is symmetric, are the same as those of P^π, and thus the eigenfunctions of the Laplacian used in the companion paper.

Formula (9) only involves, say, $K(\epsilon,\gamma)$ terms, depending on the precision ϵ and the distance between γ and the spectrum of P^π. So the total cost for computing (9) is $\mathcal{O}(K(\epsilon,\gamma)|S|\log^2|S|)$. The constant is typically very small, and depends crucially on the sparsity and the size of the filter matrices $[T^{2^j}]_{\Phi_j}^{\Phi_j}$. The best scenario is when $[T^{2^j}]_{\Phi_j}^{\Phi_j}$ has size $\mathcal{O}(2^{-j}|S|)$ and only $\mathcal{O}(2^{-j}|S|\log|S|)$ non zero entries. The analysis in (Coifman and Maggioni, 2004) shows this is the case for certain diffusion-like processes on discretized manifolds, and certain classes of random walks on graphs.

5. Function Approximation and Dimensionality Reduction

In the companion paper, we reviewed function approximation in Euclidean and Hilbert spaces. There, the main example of Hilbert space was $\mathbb{L}^2$, the space of square-integrable functions. Here we discuss Hilbert spaces of smooth functions, and more general spaces of piecewise smooth functions. We also want to make explicit the connection between the task of efficiently approximating functions and dimensionality reduction. We will keep this discussion at the minimum level necessary to appreciate its relevance to the solution of MDPs, motivate the use of the eigenfunctions of the Laplacian and diffusion wavelets, and compare between them.

Suppose we knew *a priori* that the value function lies in a certain normed vector space of functions. In the companion paper we discussed the Hilbert space $\mathbb{L}^2$ of square-integrable functions. This space is often too big, in the sense that it includes quite wild functions (e.g. functions everywhere discontinuous). Here we focus on spaces of smooth or piecewise smooth functions, which in many situations include value functions of interest. The advantage of this approach is that faster approximation, and more powerful dimensionality-reduction, is possible in these function spaces, if the appropriate basis is used.

5.1 Smoothness spaces, Fourier basis and wavelets in 1 dimension

We start from simple examples to show how smoothness is useful in order to find low-dimensional approximation of functions. This material is classical, we refer the reader to the books by Daubechies (1992); Mallat (1998).

Consider the interval $[0,1]$. We define the following natural spaces of functions:

(i) $\mathbb{L}^2([0,1]) = \{f \text{ measurable } : ||f||_2^2 := \int_0^1 |f(x)|^2 dx < +\infty\}$;

(ii) $\mathcal{H}^s([0,1]) = \{f : ||f||_s^2 := \sum_{k\in\mathbb{Z}} |\hat{f}(k)|^2(1+|k|^{2s}) < +\infty\}$.

Here $\hat{f}(n)$ is the k-th Fourier coefficient, defined below. Functions in $\mathbb{L}^2([0,1])$ could be discontinuous everywhere, functions in $\mathcal{H}^s([0,1])$ are much more regular and have s weak derivatives (when s is an integer). In fact the "natural" definition of $\mathcal{H}^s$ is through weak derivatives, with the norm defined by

$$||f||_s^2 := \sum_{k\le s} ||f^{(k)}||_2^2,$$

when s is an integer. However the definition through the Fourier transform is much more natural for our purposes, and immediately suggests generalizations to manifolds and graphs. Clearly $\mathbb{L}^2([0,1]) \supsetneq \mathcal{H}^s([0,1])$. Both $\mathbb{L}^2([0,1])$ and $\mathcal{H}^s([0,1])$, with the norms defined above, are Hilbert spaces.

Let us consider the functions

$$\phi_k = \frac{1}{\sqrt{2\pi}} e^{2\pi i k} \quad , k \in \mathbb{Z}.$$

It is well known that $\{\phi_k\}_{k\in\mathbb{Z}}$ is an orthonormal basis in $\mathbb{L}^2([0,1])$, called the Fourier basis. Observe that each complex exponential can be expressed as a sum of sine and cosine functions, at the appropriate frequency. Hence any function $f \in \mathbb{L}^2([0,1])$ can be expressed as

$$f = \sum_{k\in\mathbb{Z}} \langle f, \phi_k\rangle \phi_k = \sum_{k\in\mathbb{Z}} \hat{f}(k)\phi_k \tag{12}$$

where the limit involved in the summation is unconditional when taken with respect to the distance in $\mathbb{L}^2([0,1])$, and the k-th Fourier coefficient is given by

$$\hat{f}(k) := < f, \phi_k > = \int_0^1 f(x)\phi_k(x)dx = \frac{1}{\sqrt{2\pi}} \int_0^1 f(x)e^{2\pi ikx}dx\,.$$

This basis is very useful in characterizing the function spaces above and in efficiently representing functions in those spaces. We have the following results:

(i) $f \in \mathbb{L}^2([0,1])$ if and only if $\{\hat{f}(k)\} \in \ell^2$ [5], and in fact $||f||_{\mathbb{L}^2([0,1])} = ||\{\hat{f}(k)\}||_{\ell^2}$ (Parseval's relation);

(ii) $f \in \mathcal{H}^s([0,1])$ if and only if $\{(1+|k|^s)\hat{f}(k)\} \in \ell^2$;

In other words, the decay of the Fourier coefficients $\hat{f}(k)$ as $|k|$ approaches infinity contains information about the smoothness of f. These results generalize to spaces of functions on $\mathbb{R}^n$ and, as we shall see below, on manifolds and graphs.

A simple consequence of these results is the following result in linear approximation. Consider the subspace of band-limited functions with band K:

$$B_K = \langle \{\phi_k : |k| \le K\} \rangle\,.$$

The orthogonal projection on B_K is given by

$$P_{B_K} f = \sum_{|k| \le K} < f, \phi_k > \phi_k\,.$$

We can estimate *a priori* the error of approximating f by $P_{B_K} f$, depending on the smoothness of f and its norm. For example, if $f \in \mathcal{H}^s([0,1])$, then

$$\begin{aligned} ||f - P_{B_K} f||_2 = \left\| \sum_{|k|>K} \hat{f}(k)\phi_k \right\| &\le \left(\sum_{|k|>K} (|k|^s |\hat{f}(k)|)^2 \right)^{\frac{1}{2}} \left(\sum_{|k|>K} |k|^{-2s} \right)^{\frac{1}{2}} \\ &\le \frac{C_s ||f||_{\mathcal{H}^s}}{K^{s+\frac{1}{2}}}\,. \end{aligned} \tag{13}$$

Hence for a fixed precision ϵ, it is enough to choose $K_\epsilon \le C_s \epsilon^{\frac{1}{s+\frac{1}{2}}} ||f||_{\mathcal{H}^s}$ in order to approximate f by $P_{B_K} f$ up to error ϵ. No such result could hold if we just knew that $f \in \mathbb{L}^2$, the only statement we could make is that $||f - P_{B_K}||_2 \to 0$ as $K \to 0$, but this rate of convergence could be arbitrarily slow.

This simple example shows how decay properties of the coefficients onto a basis of a function with certain smoothness properties can be taken advantage of in determining useful subspaces in which to approximate the given function. This has far-reaching generalizations, to many different function spaces and bases (Daubechies, 1992; Mallat, 1998). Many natural questions arise: for example, is the Fourier basis the best possible for the K-term linear approximation described above? It turns out that the answer, in a certain worst-case sense, is yes!

We turn now to more complex, but natural and interesting, situations in which the Fourier basis is not the optimal basis for linear approximation. This happens when the smoothness is not global, but varies in different parts of the domain. For example if we consider a function $[0,1]$ that is infinitely differentiable, we expect its Fourier coefficients

5. A sequence $\{a_k\}$ is in ℓ^2 iff $||\{a_k\}||_{\ell^2}^2 := \sum_k |a_k|^2 < \infty$.

to decay faster than any inverse polynomials. Now if we consider a function everywhere infinitely differentiable except for a point of discontinuity (for example the function equal to 1 on $[0, \frac{1}{2}]$ and -1 on $(\frac{1}{2}, 1]$, it turns out that its Fourier coefficients decay no faster than $\frac{1}{k}$. We ought to be able to do better than this, perhaps by choosing another basis! Wavelet bases turn out to be a much better choice, the optimal in some sense, if one does not known a priori the location of the discontinuity. More generally, the optimal basis for functions of one-variable, with isolated discontinuities (of the function itself and/or in any of its (weak) derivatives) at unknown points, and in some $\mathcal{H}^s$ in every interval not containing discontinuities, is a wavelet basis.

5.2 Smoothness spaces on manifolds

We introduce various notions of smoothness and indicate how to take advantage of them in order to define interesting subspaces V and corresponding basis functions. We will treat separately two cases: when the state space is a Riemannian manifold, or when it is a graph. We will then show that there is a common approach for both cases.

Suppose the state space can be modeled as a smooth compact oriented Riemannian manifold $(\mathcal{M}, g)$ (possibly with boundary), where g denotes the Riemannian metric [6]. Simple examples are the state space of a pendulum, the circle S^1, and the state space of a double pendulum, which is $S^1 \times S^1$. There is a natural volume measure on $\mathcal{M}$, and $\mathbb{L}^2(\mathcal{M})$ is defined with respect to this measure. A Laplacian Δ can be defined intrinsically on $\mathcal{M}$, see for example (Rosenberg). Associated with Δ is the heat kernel, which can be defined as the operator $H_t = e^{-t\Delta}$. The heat kernel solves the heat equation, at time t, with initial solution given by a Dirac δ-function: $H_t(\delta_x)(y)$ is the amount of heat at y at time t when a unit mass of heat is put at x at time 0. This is connected to natural random walks in the state space, which can thought of Brownian motion on the manifold.

Under the assumptions above, the heat kernel exists, is unique, strictly positive, and is a compact self-adjoint operator. As such, there exists an orthonormal basis of eigenfunctions $\phi_j \in \mathbb{L}^2(\mathcal{M})$ corresponding to eigenvalues $e^{-t\lambda_j}$:

$$H_t \phi_j = e^{-t\lambda_j} \phi_j \,. \tag{14}$$

Of course we have

$$\Delta \phi_j = \lambda_j \phi_j \,. \tag{15}$$

The eigenvalues are nonnegative, and have no accumulation point except for infinity. We will always sort the eigenvalues so that

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_j \leq \ldots \,.$$

One can define smoothness spaces in the following way: we can let

$$\mathcal{H}^s(\mathcal{S}) = \left\{ f \in \mathbb{L}^2(\mathcal{M}) : \{(1+\lambda_j^s)\hat{f}(j)\}_j \in \ell^2 \right\} , \tag{16}$$

where $\hat{f}(j) = \langle f, \phi_j \rangle$. In particular for the space $\mathcal{H}^1$ we have

$$||f||^2_{\mathcal{H}^1(\mathcal{S})} \sim \sum_j |\hat{f}(j)|^2 + \sum_j \lambda_j^2 |\hat{f}(j)|^2 = ||f||_2^2 + ||\nabla f||_2^2 = ||f||_2^2 + < \Delta f, f >$$

One could of course repeat the simple estimates in (13).

6. The compactness assumption is not strictly necessary, but we prefer to work in this setting in order to avoid matters regarding existence, uniqueness, positiveness and compactness (as an operator) of the heat kernel.

5.3 Smoothness spaces on graphs

The smoothness of a function on a graph, can be measured by the Sobolev norm (4) exactly as above. The simple estimates in (13) can of course be applied here as well.

5.4 Eigenfunctions of the Laplacian

In the companion paper (Mahadevan and Maggioni, 2006), the top K eigenfunctions of the Laplacian of the graph associated with the state space are considered, and (at each iteration of a policy iteration algorithm) the value function V^π is approximated in the subspace B_K spanned by them. Suppose we know *a priori* at $V^\pi \in \mathcal{H}^s$ for some $s > 0$. Then for a fixed precision $\epsilon > 0$ we see from (13) that we can estimate a $K = K(\epsilon)$ such that the error introduced by the projection of V^π onto B_K is smaller than ϵ. It is natural then to choose the top K eigenfunctions as the basis in which to represent the value function in policy iteration. However it is not guaranteed that the initial policy and all the iterates will lie in this subspace.

5.5 Approximation of Piecewise-smooth Functions with Diffusion Wavelets

By using diffusion wavelets, it is possible to represents functions which are piecewise in $\mathcal{H}^s$ but not globally in $\mathcal{H}^s$. Partition the state space into a finite number of regions, separated by piecewise smooth boundaries, of total length L. Assume that a function f is in $\mathcal{H}^1$ of each region, but possibly discontinuous at the boundary between two regions. There two types of wavelet coefficients $c_{j,k} := \langle f, \psi_{j,k} \rangle$. If the support of $\psi_{j,k}$ intersects one of the boundaries, then $|c_{j,k}|$ will in general be large, while if $\psi_{j,k}$ does not intersect any boundary, then $|c_{j,k}|$ will satisfy the decay estimates as if $f \in \mathcal{H}^1$. Observe the at each scale j, only at most $C\dot{L} \cdot 2^j$ wavelets intersect one of the boundaries, and since there are only $\log |S|$, across all scales only $\sim L$ wavelet coefficients are affected by the boundary. This is radically different from what happens to the coefficients $\langle f, \phi_j \rangle$, where ϕ_j is the j^{th} eigenfunction of Δ: because of the discontinuities, no estimate better than $\{\langle f, \phi_j \rangle\} \in \ell^2$ can be expected! In this case nonlinear approximation with diffusion wavelets proves very useful.

5.6 Linear and nonlinear approximation

Let $\Phi = \{\phi_k\}_{k=0,1,\ldots}$ be a set of basis functions. For a function f, its K-term *linear approximation* on Φ is of the form

$$f \sim \sum_{i=0}^{I} \alpha_i \phi_i \,. \tag{17}$$

Here I is either a parameter, or it is chosen so that $\sim$ in the formula above indicates an error smaller than some precision ϵ, so that $I = I(\epsilon)$. Observe that the only degree of freedom here is the choice of the coefficients α_k. For example if the approximation is in the $\mathbb{L}^2$-norm and Φ is orthonormal, then the optimal choice is $\alpha_i = \langle f, \phi_i \rangle$, corresponding to orthogonally projecting f on the subspace spanned by $\{\phi_1, \ldots, \phi_I\}$. When Φ is a set of Laplacian eigenfunctions, ϕ_i is picked to be the eigenfunction corresponding to the i-th lowest eigenvalue of the Laplacian, i.e. corresponding to the i-th lowest frequency. When Φ is a set of diffusion wavelets, the set is naturally ordered by scale, from coarse to fine. So $\phi_{j,k}$ will be listed before $\phi_{j',k'}$ is $j > j'$. There is no natural ordering within a given scale. A small caveat: in the construction of diffusion wavelets of (Coifman and Maggioni, 2004), the index of the scaling functions and wavelets at a given scale is roughly proportional to the frequency at that scale, so in fact there is a "rough" sorting even within each scale.

A K-term *nonlinear approximation* of f is in the form

$$f \sim \sum_{i=0}^{I} \alpha_i \phi_{k_i} . \tag{18}$$

In other words in this expansion we are allowed to pick an arbitrary subset of I functions in Φ, rather than the first I, besides the corresponding set of coefficients $\{\alpha_i\}_{i=0}^{I}$. Nonlinear approximation is in general more powerful than linear approximation, for example if Φ is orthonormal, then for fixed I it will provide a better approximation to f unless the largest I inner products among $\{\langle f, \phi_k \rangle\}_{k=0,1,...}$ are exactly those corresponding to $k = 0, \ldots, I-1$. However the best nonlinear approximation is much harder to compute, since for example when Φ is orthonormal it may require the computation of all the inner products $\{\langle f, \phi_k \rangle\}_{k=0,1,...}$, before selecting the largest ones.

In general nonlinear approximation in a wavelet basis differs dramatically from nonlinear approximation in the eigenfunction basis.

6. Scaling Multiscale Analysis of MDPs to Continuous Domains and Factored Domains

In continuous domains, one of the critical ingredients for success of a method based on the representation of the value function on some basis of the state-action space is the capability of approximating the value function efficiently in this basis, and in extending these basis functions to novel states. In the first paper we showed how it is possible to extend eigenfunctions of the Laplacian learned on a subset of points in the state space to new states. The main tool was Nyström extension (see (Belongie et al., 2002; Fowlkes et al., 2001; Williams and Seeger, 2000)). Here we would like to extend diffusion wavelets learned and sampled on a subset of a continuous state space to new states. A simple Nyström extension formula does not apply in this case, and we need to use different local interpolation rules. A simple local averaging rule has been used in the examples of this paper. As a future extension, we propose the use of a much more refined type of extension, originally suggested in (Coifman and Maggioni, 2004) and inspired by (Coifman et al., 2005b; Coifman and Lafon, 2004b; Lafon, 2004).

6.1 Subsampling techniques

The complexity (however defined) of the value function determines the number of samples necessary to approximate the value function up to a given precision. This number of samples is independent of the number of states explored. Consider the following simple example.

Example 5 *Suppose the state space is the interval* $[0, 1]$, *and that the value function* V *is bandlimited with band* B. *This means that the Fourier transform* $\hat{V}$ *is supported in* $[-B, B]$. *Then by Shannon's theorem, only* $B/(2\pi)$ *equispaced samples are needed to recover* V *exactly.*

Suppose we have observed samples $\mathcal{S}'$ in the state space $\mathcal{S}$, and that V is smooth so that a subset $\mathcal{S}''$ much smaller than $\mathcal{S}'$ would suffice to determine V. We propose two simple methods in order to select $\mathcal{S}''$.

6.1.1 PURELY RANDOM SUBSAMPLING

We fix $|\mathcal{S}''|$, and select $|\mathcal{S}''|$ points uniformly at random in $\mathcal{S}'$. For very large $|\mathcal{S}'|$ one would expect that the points in $\mathcal{S}''$ are going to be well-spread in $\mathcal{S}'$.

6.1.2 WELL-SPREAD RANDOM NET

The previous algorithm has two main drawbacks: the first one is that it is not clear how to select $|\mathcal{S}''|$, even if in theory this can be determined by knowing the complexity of the value function to be approximated. The second one is that the points in $\mathcal{S}''$ are not going to be necessarily well-spread in $\mathcal{S}'$: while it is true that for large $|\mathcal{S}'|$, with very high-probability no two points in $\mathcal{S}''$ are going to be very close, it is not true that the points in $\mathcal{S}''$ are going to be roughly equidistant nor well equidistributed in balls contained in $\mathcal{S}'$.

In order to guarantee that the set of points is well spread, we consider the following construction. We define an ϵ-net of points in $\mathcal{S}'$ to be a subset $\mathcal{S}''$ such that no two points are closer than ϵ, and that for every point y in $\mathcal{S}'$, there is a point in $\mathcal{S}''$ which is not farther than ϵ from y. One can construct a (random) ϵ-net in $\mathcal{S}'$ as follows. Pick $x_0 \in \mathcal{S}'$ at random. By induction, for $k \geq 1$ suppose $x_0, x_1, \ldots, x_k$ have been picked so that the distance between any pair is larger than ϵ. If

$$R_k := \mathcal{S}' \setminus (\cup_{l=1}^{k} B_\epsilon(x_l))$$

is empty, stop, otherwise pick a point x_{k+1} in R_k. By definition of R_k the distance between x_{k+1} and any of the points $x_0, \ldots, x_k$ is not smaller than ϵ. When this process stops, say after k^* points have been selected, for any $y \in \mathcal{S}'$ we can find a point in $\mathcal{S}''$ not farther than ϵ, for otherwise $y \in R_{k^*}$ and the process would not have stopped.

One can prove upper bounds of the distance between the eigenfunctions of the Laplacian on $\mathcal{S}'$ and the eigenfunctions of the Laplacian on $\mathcal{S}''$, which depend on ϵ and the order of the eigenfunction (Maggioni, 2006).

6.2 Factored Domains

The case of factored domains can be treated exactly as in the companion paper. Essentially whenever the diffusion/random walk on a graph factors as a product of random walks on factor graphs, the associated diffusion wavelets factor into tensor products of diffusion wavelets on each factor. This leads to savings in both computational and storage costs. We notice however that this is not enough to defeat the "curse of dimensionality" during the function approximation case. In general a large number of basis functions will be needed in order to approximate a generic function on a factorizable space. However, if the function to be approximate satisfies particular smoothness conditions or separability conditions, then its approximation can be performed efficiently.

6.3 Randomized algorithms

Randomized algorithms can play a role in the multiscale analysis at least two respects. First of all the construction of diffusion wavelets can be randomized: instead of performing a QR-factorization $T_j = Q_j R_j$ at each scale, one can randomize the computation of such a factorization, for example by column sampling. In particular it is clear how to apply certain randomized algorithms for low-rank matrix approximation (see for example (Achlioptas et al., 2002; Chennubhotla and Jepson, 2005; Frieze et al., 1998; Drineas et al., 2004; Drineas and Mahoney, 2005)) to this context. This and finer randomized algorithms tuned to the multiscale construction are currently under investigation, and progress will be reported in a future publication.

Secondly, suppose we have a large number of samples X' of the state space. Many such samples may be "redundant" in view of the computation of the multiscale basis, and of the approximation of the value function. One can randomly subsample X' , with algorithms similar to the ones mentioned above, to a much smaller sample set X'', perform the

multiscale construction on X'', and then extend the basis functions to X' by interpolation. We have applied successfully this technique in several domains, in particular in continuous domains such as the inverted pendulum, continuous two-rooms domain, etc. Further investigation of these techniques is needed, in order to design optimal adaptive sampling and sub-sampling techniques. This is well-known to be a hard problem.

7. Experimental Results

In this section, we describe a detailed set of experiments in both discrete and continuous MDPs that illustrate the diffusion wavelet framework for policy evaluation given a fixed transition matrix, and control learning with approximate policy iteration. In all the cases, the ability of diffusion wavelets to automatically construct a hierarchy of intermediate representations provides a new framework for solving MDPs, as will be illustrated.

7.1 Control Learning using Representation Policy Iteration

7.1.1 CHAIN DOMAIN

First, we turn to control learning and compare the performance of diffusion wavelets and Laplacian eigenfunctions using the Representation Policy Iteration (RPI) algorithm described in (Mahadevan, 2005b) on the classic chain example from (Lagoudakis and Parr, 2003). RPI can be viewed as a modified LSPI algorithm where the basis functions $\phi(s,a)$ handcoded in LSPI are learned from the graph Laplacian using a random walk of 5000 steps for a 50 state chain. The chain MDP is a sequential open (or closed) chain of varying number of states, where there are two actions for moving left or right along the chain. In the experiments shown, a reward of 1 was provided in states 10 and 41. Given a fixed k, the encoding $\phi(s)$ of a state s for Laplacian eigenfunctions is the vector comprised of the values of the k^{th} lowest-order eigenfunctions on state k. For diffusion wavelets, all the basis functions at level k were evaluated at state s to produce the encoding.

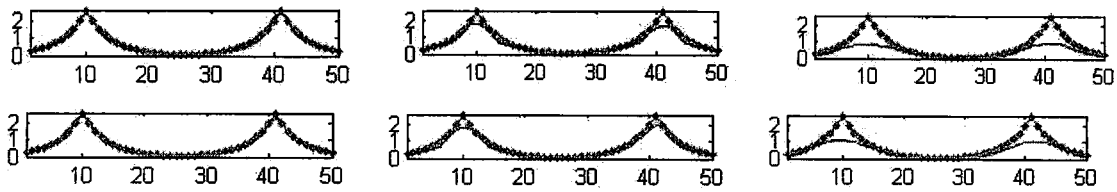


Figure 10: This experiment compares value function approximation in a 50 state chain MDP using 44 diffusion wavelet basis functions at level 4 (top left), 19 basis functions at level 6 (top middle), and 10 basis functions at level 8 (top right) of the hierarchy, and using 40 Laplacian basis functions (bottom left), 20 basis functions (bottom middle), and 10 basis functions (bottom right). For each plot, the dotted line is the exact value function, and the smooth line is the approximation.

Figure 10 illustrates the varying smoothness of the approximation produced by diffusion wavelet trees and Laplacian eigenfunctions. As the number of basis functions are reduced, the smoothness increases and the approximation gets progressively worse. As the figure shows, it is possible to get very accurate approximation using either technique provided sufficient number of basis functions is selected.

Method	#Trials	Error	Method	#Trials	Error
RPI DF (9)	4.2	2.4	LSPI RBPF (6)	3.8	20.8
RPI DF (13)	5.2	4.6	LSPI RBPF (14)	4.4	2.8
RPI DF (19)	5.4	8.2	LSPI RBF (26)	6.4	2.8
RPI Lap (5)	4.2	3.8	LSPI Poly (5)	4.2	4
RPI Lap (15)	7.2	3	LSPI Poly (15)	1	34.4
RPI Lap (25)	9.4	2	LSPI Poly (25)	1	36

Table 1: This table compares the performance of RPI using diffusion wavelets and Laplacian eigenfunctions with LSPI using handcoded polynomial and radial basis functions on a 50 state chain graph MDP.

Table 1 compares the performance of RPI using diffusion wavelets and Laplacian eigenfunctions, along with LSPI using two handcoded parametric basis functions: polynomials and radial-basis functions (RBF). Each row reflects the performance of either RPI using learned basis functions or LSPI with a handcoded basis function (values in parentheses indicate the number of basis functions used for each architecture). The two numbers reported are steps to convergence and the error in the learned policy (number of incorrect actions), averaged over 5 runs. The results show the automatically learned Laplacian and diffusion wavelet basis functions in RPI provide a more stable performance at both the low end and at the higher end, as compared to the handcoded basis functions used in LSPI. As the number of basis functions are increased, RPI with Laplacian basis functions takes longer to converge, but learns a more accurate policy. Diffusion wavelets converge quickly, but strangely the error grows as the number of bases functions is increased. This result is somewhat puzzling and will be investigated further. LSPI with RBF is unstable at the low end, converging to a very poor policy for 6 basis functions. LSPI with a 5 degree polynomial approximator works reasonably well, but its performance noticeably degrades at higher degrees, converging to a very poor policy in one step for $k = 15$ and $k = 25$.

7.1.2 INVERTED PENDULUM

This domain is described in the first paper (Mahadevan and Maggioni, 2006). We use diffusion wavelets instead of eigenfunctions of the Laplacian. The sampled state space results from 300 runs of length at most 50, and consists of 2868 points. A random subset of 900 points is used to construct a graph, with 90 nearest neighbors and $\delta = 0.9$. The diffusion wavelets associated with the natural random walk on this graph are constructed, and 27 basis functions are kept, counting them from the coarsest level down to finer levels. These are then extended to all the points via local nearest-neighbors averaging.

7.1.3 MOUNTAIN CAR DOMAIN

This domain is also described in the first paper (Mahadevan and Maggioni, 2006). We use diffusion wavelets instead of eigenfunctions of the Laplacian. The sampled state space results from 400 runs of length at most 50, and consists of 18734 points. A well-distributed subset of 1111 points is used to construct a graph, with 30 nearest neighbors and $\delta = 0.1$. The diffusion wavelets associated with the natural random walk on this graph are constructed, only the coarsest scaling functions and increasingly high frequency wavelets

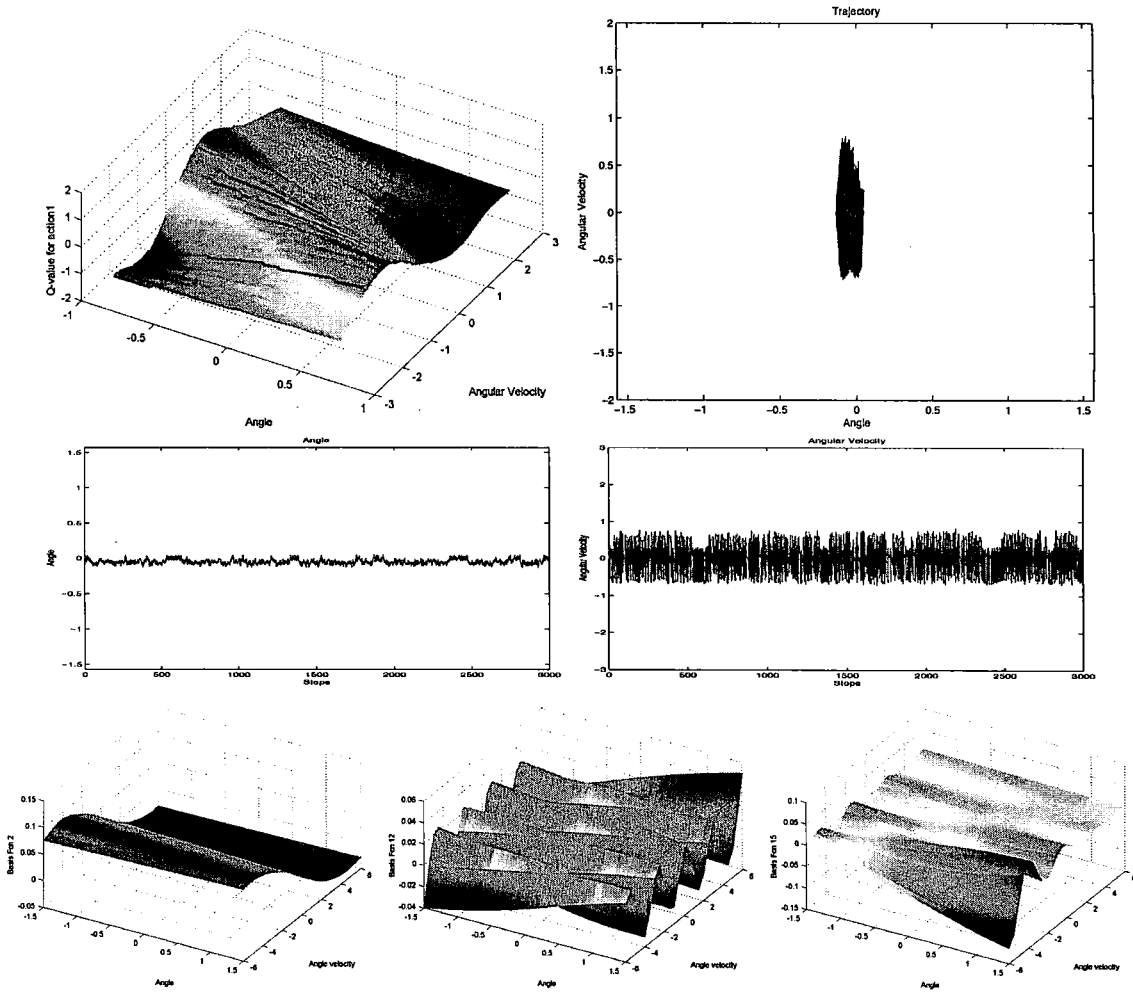


Figure 11: Top left: Q-value function for the action "left", reconstructed from its representation of the diffusion wavelet basis. Top right: trajectory of the pendulum in phase space according to the policy learned. Center row: trajectory of angle and angle velocity variables. Bottom row: some diffusion wavelets used as basis functions for representation during the learning phase.

for a total of 50 basis functions are kept. These basis functions are then extended to all the points via local averaging.

7.2 Direct multiscale inversion

We now turn to describe the direct inversion of the transition matrix using the multiscale method. We tested the multiscale analysis on several MDPs, on discrete and continuous spaces of different topologies. In fact the technique used is extremely flexible, since it essentially only needs P^π, or an estimate thereof, as an input. We consider here two examples. In the first example we consider the problem in which P^π is given, and the optimal value function, solution of Bellman's equation (8) is sought, and solve Bellman's

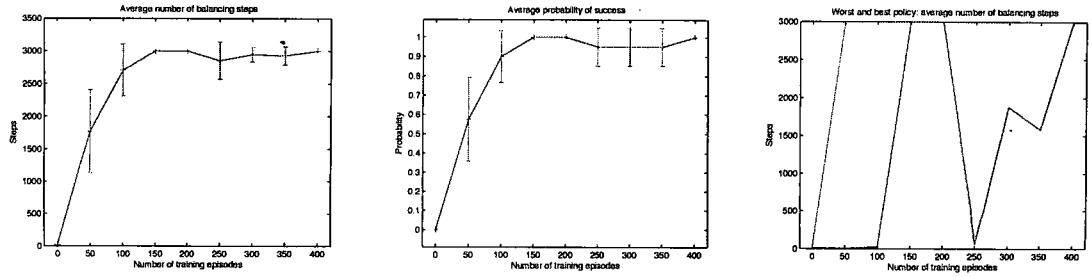


Figure 12: Measures of performance based on 20 experiments, as a function of number of training runs (each of which of length at most 100). From left to right: average number of successful steps of inverted pendulum balancing, average probability of succeeding in balancing for at least 3000 steps, and worst and best number of balancing steps. Each simulation was stopped and considered successful after 3000 steps, which biases the first and third graphs downwards.

equation using direct multiscale inversion. The domain we consider consists in a continuous two-room environment, quite arbitrarily shaped, discrete sampled, with P^π given by a random walk on the samples. In the second example we test the direct inversion on the chain domain: instead of solving the projected Bellman's equations by using least-squares methods, we use the multiscale direct inversion.

7.2.1 TWO-ROOMS ENVIRONMENT

We describe here one example in some detail. It simulates a continuous two-rooms environment, where the two rooms have an elongated shape and are connected by a corridor. The shape of the rooms and of the corridor is quite arbitrary, the bases are built automatically: we do not require any special topology or shape property for them (except connectedness, without loss of generality): we could have chosen rooms of arbitrary shapes, in arbitrary dimension, as the only input to the algorithm is the set of sampled points (vertices) and the local distances between close-by points (edge weights).

The agent has randomly explored the space, so S consists of $|S|$ randomly scattered points in the rooms (see Figure 15). We construct a natural diffusion associated with the random walk in the two rooms, restricted to the states S actually explored, by letting $W(i,j) = e^{-2\|x_i - x_j\|^2}$. This diffusion approximates the natural random walk (Brownian motion) in the continuous domain (see (Belkin and Niyogi, 2003b; Lafon, 2004)), corresponding to a policy of random moves. We then construct the corresponding multiscale analysis, with precision set to 10^{-10}. In Figure 15 we represent some of the scaling functions we obtain. In Figure 16 we represent compressed dyadic powers of this random walk. In Figure 17, left, we compare the direct computation time of $(I - \gamma P^\pi)^{-1}$ and the computation time for the multiscale structure, i.e. the pre-processing time for the two direct methods under consideration. We then pick a random reward R on S (a vector of white Gaussian noise), and compute the corresponding value function in three different ways:

(i) direct computation of the matrix $I - \gamma P^\pi$,

(ii) Schultz's method and diffusion wavelet transform as in (9),

(iii) conjugate gradient descent for symmetric matrices.

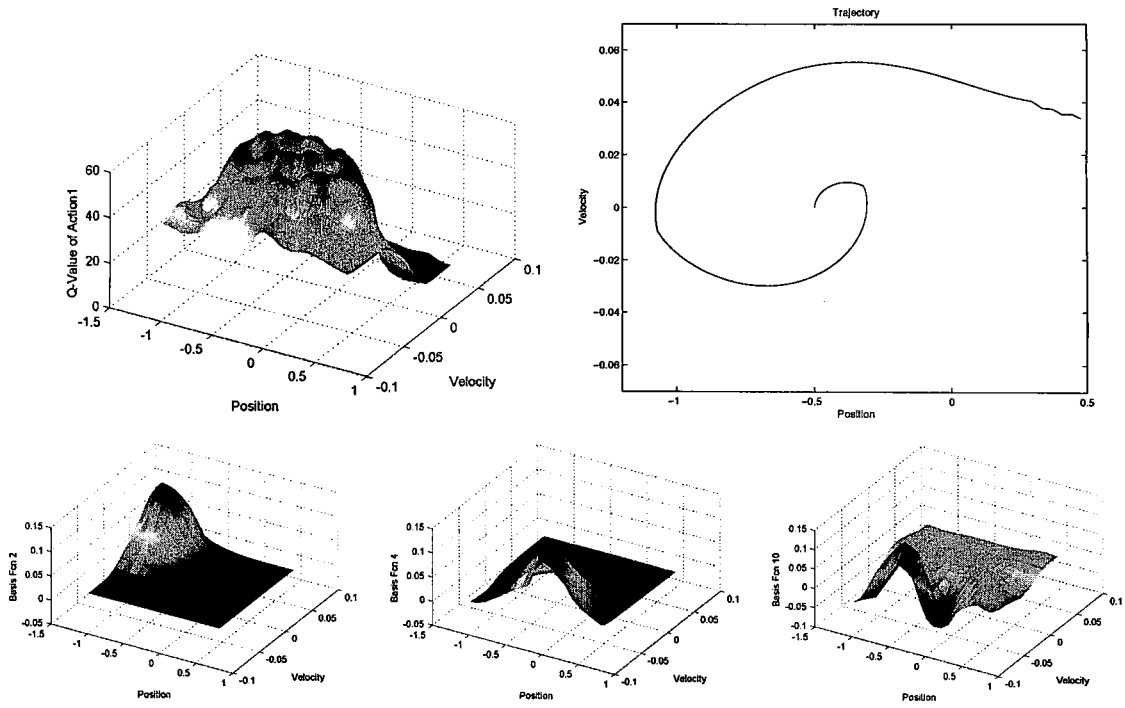


Figure 13: Top left: Q-value function for the action "left", reconstructed from its representation of the diffusion wavelet basis. Top right: trajectory of the mountain car in phase space according to the policy learned (107 steps). Bottom row: some diffusion wavelets used as basis functions for representation during the learning phase.

In this example we set $\gamma = 1$. We repeat the above for $|S| = 320 : 40 : 1040$ and, for each S, for 10 randomly generated rewards R. The first two methods are direct: we look at both the pre-processing time for computing, respectively, the inverse matrix and the diffusion wavelet tree (see Figure 17 left). Then we compare, over several random choices of the reward vector, the mean and standard deviation of the time for computing the corresponding value function, with all three methods: see Figure 17, right. Finally, in Figure 18 we show the L^2- and L^∞-norms of the Bellman residual $((I - P^\pi)\tilde{V}^\pi - R$, where $\tilde{V}^\pi$ is the estimated value function), achieved by the three methods.

We stress that the code that implements the construction of Diffusion Wavelet Tree and Schultz's formula is written mainly in Matlab and large parts of it are not optimized in any way; on the contrary the codes distributed with Matlab for conjugate gradient and direct inversion have been highly optimized. The results here are thus qualitative and not absolute, but point out that at the very least the direct solution using diffusion wavelets seems competitive with state-of-the-art methods, even before having optimized the code implementing it. Future work will be devoted for these optimizations, which will result in speed-ups and the ability to tackle larger problems. In particular factored spaces are easily tackled since the natural policies on these spaces factor, and so do the diffusion wavelets, as in the case of eigenfunctions of the Laplacian (Mahadevan and Maggioni, 2006).

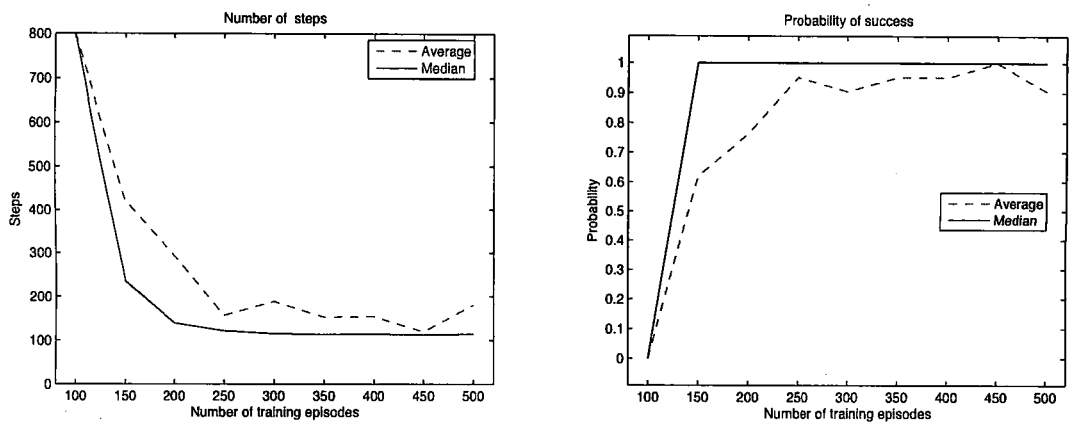


Figure 14: Measures of performance based on 23 experiments, as a function of number of training runs (each of which of length at most 100). Left: average and median number of successful steps for reaching the goal; right: average and median probability of succeeding in reaching the goal in less than 800 steps. The best policy actually finds a path in 103 steps.

7.2.2 DIRECT INVERSION FOR THE CHAIN DOMAIN

As a proof of concept, we applied the direct multiscale inversion to the solution of the projection of Bellman equation within the RPI framework. At each step of iteration in the representation policy iteration algorithm the projected Bellman equation is solved with the direct multiscale solver. Observe that in general the matrices involved are asymmetric, since the policy clearly has strong directional preferences. The difference between the solution obtained by standard least squares and direct multiscale inversion at every iteration is always smaller than 10^{-8}.

This approach is however problematic in general: while γP^π has (operator) norm ≤ 1, its approximation on the basis used for representation may have norm larger than 1, which prevents the use the Neumann series expansion for $(I - \gamma P^\pi)^{-1}$ and hence the applicability of the multiscale expansion.

8. Discussion

Many extensions of the framework proposed in this paper are being actively explored, which can only be briefly summarized due to space constraints. We have naturally tried to restrict our discussion to the simplest methods, but the scope of multiscale analysis based on diffusion can easily be extended to cover more general situations. It is natural that all the possible generalizations discussed at the end of the companion paper apply here as well. Here we discuss some other possibilities.

8.1 Adaptive Nonlinear Approximation

In the examples considered in this paper we considered bases given by the coarsest scaling functions and the wavelets in a fixed number of wavelet subspaces, counting from the coarsest toward the finest. First of all, this involves the choice of an arbitrary parameter,

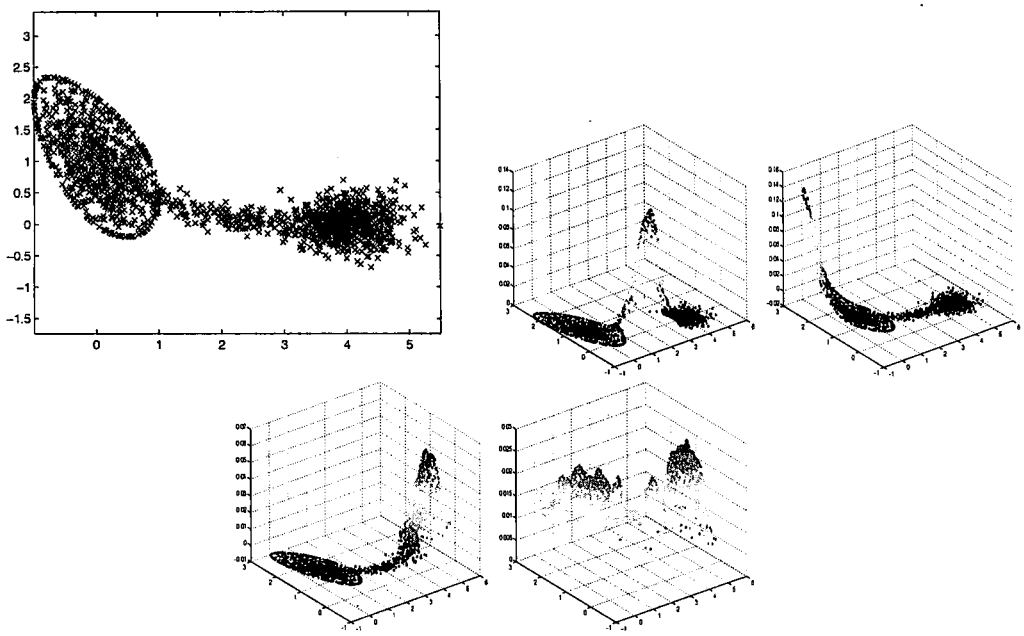


Figure 15: Top: set of samples in a two-room environment. Bottom: four diffusion scaling functions built on the set, at increasing scale. Note the localization at the finer scales, and the global support at coarser scales.

the number of wavelet subspaces to include in the basis. On the one hand the learning does not seem to be very sensitive to this parameter, on the other hand it is the case that if too few basis functions are specified, the learning is severely impaired, and if too many are specified, many more training samples seem to be needed to guarantee stability in the learning phase. It would be desirable to detect this parameter automatically. Even further, this choice of basis functions is in general suboptimal, and one would like to select the "most useful" basis functions out of the whole wavelet basis, and discarding the "less useful" basis elements as the learning proceeds. This is extremely natural in situations where the value function has very different characteristics from location to location. For example fine wavelets in regions of smoothness essentially do not contribute to the approximation, but can increase the variance of the estimation.

This kind of nonlinear approximation can be extremely powerful. Preliminary experiments suggest that not only it is possible to borrow techniques and results from approximation theory in order to develop algorithms that adaptively select the most relevant basis functions, and their number, during the learning phase, but that this also tends to improve results over parameters carefully selected by hand.

8.2 Best Basis

As a particular case of the above, one could use techniques related to best basis algorithms (Coifman and Wickerhauser, 1992; Coifman and Saito, 1994), which have been generalized to diffusion wavelet packets (Bremer et al., 2004), in order to quickly search through a rich dictionary of bases for the one which is best adapted to the problem.

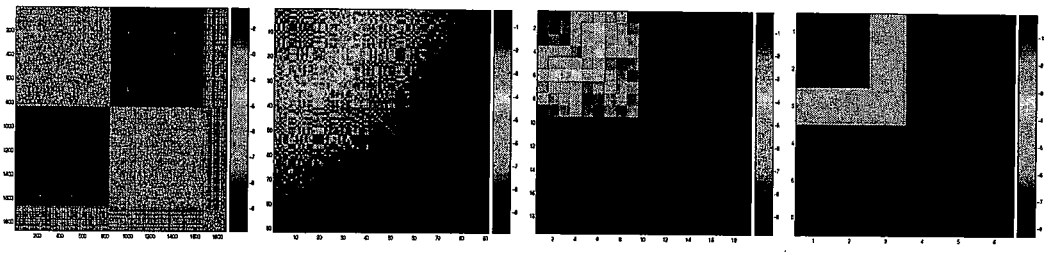


Figure 16: Compression of the powers of the symmetrized random walk T in a continuous two-room environment. From top left to bottom right by rows: T_0, T_1, T_4 and T_6. All the matrices are represented in $\log_{10}$ scale. T_0 is sorted to show the two-room and corridor structures (the algorithm is of course independent of the order of the points): the two large blocks represent transitions within each room, and the bottom-right block are transitions in the corridor, with bands at the bottom and at the right indicating the transitions from the corridor to the rooms. Notice the decreasing size of the matrices. T_6 is very small, and essentially represents only the transition between two states (the two rooms): for time scales of order 2^6 the algorithm has automatically decided this representation is faithful enough for the precision requested.

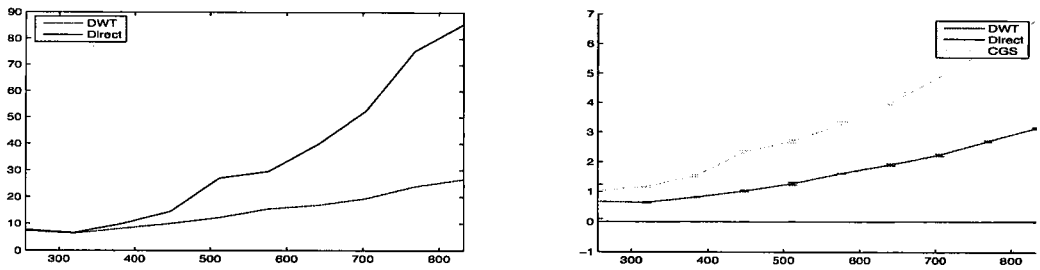


Figure 17: Left: mean and standard deviation of running time for solving a Bellman equation on a random walk in the two-room environment, as a function of the number of states explored (x-axis). We compared direct DWT inversion, iterative Conjugate Gradient Squared method (Matlab implementation) and direct inversion. Left: pre-processing time, comparing computation of the full inverse and construction diffusion wavelet tree. Right: computation time of applying the inversion scheme, comparing direct inverse, Schultz's method with diffusion wavelet transform, and symmetric conjugate gradient.

8.3 Bases for joint state and action space

While in this paper we have focused on constructing bases purely on the state space, and replicated them for each action in order to generate a basis for the joint state and action space, it is easy to generalize this approach for the construction of bases directly on the joint state action space. It is again natural to do this in a multiscale fashion, since the value

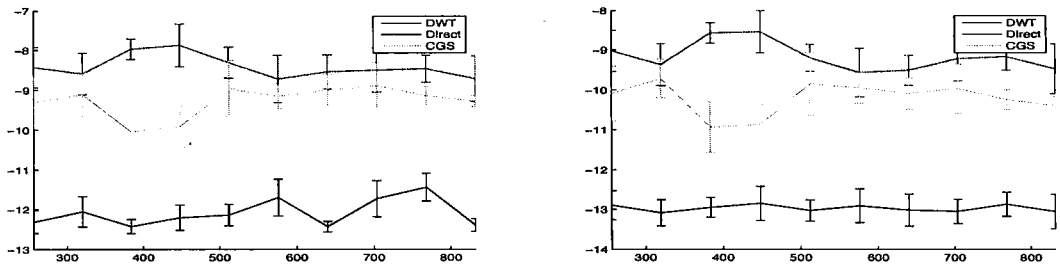


Figure 18: Precision, defined as $\log_{10}$ of the Bellman residual error $||(I - \gamma P^\pi)\tilde{V}^\pi - R||_p$, where $\tilde{V}^\pi$ is the computed solution, achieved by the different methods. The precision requested was $1e-10$. We show the results for $p = 2$ (left) and $p = \infty$ (right).

function and policy may have different smoothness properties depending on the region of state-action space.

8.4 Policy-dependent Bases

The diffusion wavelet considered in the examples were associated with the natural random walk on the sampled state space. It is natural to adapt the construction to P^π, when the policy π is not necessarily a random policy. For example one can repeat the multiscale construction at every step of a policy iteration algorithm, at step k constructing diffusion wavelets associated with P^{π_k}.

8.5 Other Multiscale Bases

Different types of multiscale bases on graphs and manifolds have been considered in (Maggioni et al., 2005b), and more are currently being investigated. These bases have properties which are different from those of diffusion wavelets, and may be better tuned (or tunable) to specific applications.

9. Acknowledgements

We would like to acknowledge the help and support of many researchers and graduate students who have inspired and helped us during the course of this research. We are indebted to Michail Lagoudakis and Ronald Parr for making available their MATLAB software for LSPI. Support for this research was provided in part by the National Science Foundation under grants ECS-0218125, IIS-0534999, DMS-0512050.

10. Appendix A. Classical Multi-Resolution Analysis

Consider a one-dimensional function f (e.g. a signal, such as a sound), and suppose we want to represent efficiently such a function, or perform tasks such as compression or denoising. *Transform methods* use a (usually linear) invertible map $f \mapsto \hat{f}$, where this map ideally has the property that simple operations on $\hat{f}$, followed by an inversion of the transformation, can be used to perform the task at hand. Typical transformations include the Fourier

transform and wavelet transform. In these cases $\hat{f}$ is the set of coefficients of f onto an orthonormal basis (resp. Fourier and wavelets), and simple operations include hard- and soft-thresholding (e.g. setting to 0 all the coefficients below a certain threshold τ, function of the noise level, smoothness etc...). When the function f is expected to have different behavior at different locations (for example a seismic wave as a function of time), it is natural to analyse and transform such a function using basis functions which are localized. In general not only the scale of localization is unknown, but it may change from location to location. It is thus desirable to have basis functions localized at all possible scales: the ones at coarse scale analyse coarse or slow variations in the signal, while the ones at fine scale analyse finer and more rapid variations. Wavelets (Daubechies, 1992; Mallat, 1998) are an example of such a basis.

A consolidated framework in wavelet analysis is the idea of Multi-Resolution Analysis (MRA). A MultiResolution Analysis (Daubechies, 1992; Mallat, 1998) of $\mathbb{L}^2(\mathbb{R})$ is a sequence of subspaces $\{V_j\}_{j\in\mathbb{Z}}$ with the following properties:

(i) $V_{j+1} \subseteq V_j$, $\overline{\cup_{j\in\mathbb{Z}} V_j} = \mathbb{L}^2(\mathbb{R})$, $\cap_{j\in\mathbb{Z}} V_j = \{0\}$;

(ii) $f \in V_{j+1}$ if and only if $f(2\cdot) \in V_j$;

(ii) there exists an orthonormal basis $\{\varphi_{j,k}\}_{k\in\mathbb{Z}} := \{2^{-\frac{j}{2}}\varphi(2^{-j}\cdot - k)\}_{k\in\mathbb{Z}}$ of V_j.

The subspace V_j is called the j^{th} approximation or scaling space. The functions $\varphi_{j,k}$ are called scaling functions. The function φ that generates, under dyadic dilations and integer translations, the family $\varphi_{j,k}$ is called the mother scaling function. The orthogonal projection on the scaling space V_j

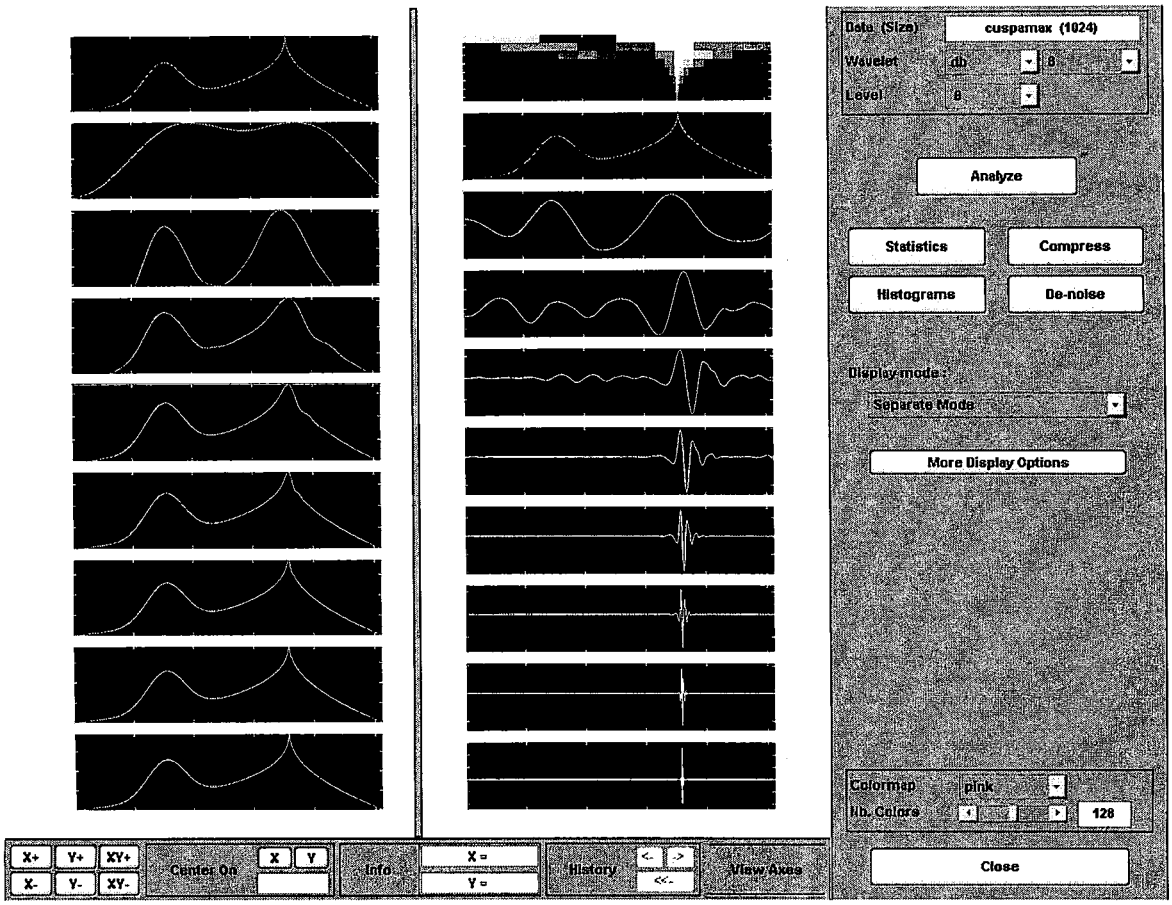
$$P_j f = \sum_{k\in\mathbb{Z}} \langle f, \varphi_{j,k}\rangle \varphi_{j,k} \tag{19}$$

gives an approximation at scale j. As j increases these approximations get coarser and coarser, while as j decreases the approximations get finer and finer, and eventually (because of (ii)), they tend to f: $P_j f \to f$ as $j \to -\infty$, where the limit is taken of course in $\mathbb{L}^2(\mathbb{R})$. One says that $P_j f$ is an approximation of f at scale j. One defines the detail or wavelet subspaces W_j as the orthogonal complement of V_{j+1} inside V_j. It turns out one can always find a function ψ such that $\{\psi_{j,k}\}_{k\in\mathbb{Z}} := \{2^{-\frac{j}{2}}\psi(2^{-j}\cdot - k)\}_{k\in\mathbb{Z}}$ is an orthonormal basis for W_j. Clearly $\oplus^{\perp}_{j\in\mathbb{Z}} W_j = \mathbb{L}^2(\mathbb{R})$ (the sum is orthogonal because $W_j \perp W_{j'}$ is $j \neq j'$), and therefore $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is an orthonormal basis for $\mathbb{L}^2(\mathbb{R})$.

The set of coefficients $\{\langle f, \psi_{j,k}\rangle\}$ is called the *wavelet transform* (WT) of f. The inclusion $V_{j+1} \subseteq V_j$ implies that $\varphi_{j+1,k}$ can be written as a linear combination of $\varphi_{j,k}$'s, and similarly the inclusion $W_j \subseteq V_j$ implies that $\psi_{j,k}$ can be written as a linear combination of $\varphi_{j,k}$'s. Moreover because of the dilation and translation structure, it is easy to see that the coefficients in these linear combinations do not depend on j and k. Hence if one knows the coefficients $\{\langle f, \varphi_{J,k}\rangle\}_{k\in\mathbb{Z}}$ of $P_J f$ for some J, it is possible to compute in a multiscale fashion all the coefficients $\{\langle f, \varphi_{j,k}\rangle\}$ and $\{\langle f, \psi_{j,k}\rangle\}$, for all $j \geq J$. This leads to a fast computation of the wavelet transform, called Fast Wavelet Transform. It can be performed in $\mathcal{O}(n \log n)$ computations, where n is the number of samples of f or of nonzero coefficients at finest scale $\{\langle f, \varphi_{J,k}\rangle\}_{k\in\mathbb{Z}}$.

The first example of MRA is due to Haar, who constructed it around 1910. He let the mother scaling function be

$$\varphi(x) = \chi_{[0,1]}(x) = \begin{cases} 1 & , x \in [0,1] \\ 0 & , \text{otherwise} \end{cases} \tag{20}$$



Example 6 Figure 19: A wavelet transform example produced with the Wavelet Toolbox in Matlab^{TM}. Top left: original signal. First column, top to bottom, are the projections of the original signal onto subspaces V_j generated by Daubechies-8 scaling functions at increasing resolution. At the top of the second column a representation of the wavelet transform of the original signal: the horizontal axis corresponds to location, the vertical axis corresponds to scales (finer scales at the bottom), the color is proportional to size of the wavelet coefficient at corresponding location and scale. The second plot is the reconstructed signal. The other plots in the second column represent the wavelet coefficients in the spaces W_j generated by wavelets at increasing resolution. This signal is highly compressible in the Daubechies-8 wavelet basis: for example with 4.65% of the coefficients it is possible to recover 99.98% of the energy ($\mathbb{L}^2$-norm) of the signal.

and the corresponding wavelet

$$\psi(x) = \begin{cases} \frac{1}{\sqrt{2}} & , x \in [0, \frac{1}{2}) \\ -\frac{1}{\sqrt{2}} & , x \in [\frac{1}{2}, 1] \end{cases} . \tag{21}$$

For a function $f \in \mathbb{L}^2(\mathbb{R})$, $P_j f$ is the function which is piecewise constant on dyadic intervals of the form $I_{k,j} = [k2^j, (k+1)2^j]$, and height equal to $2^{-\frac{j}{2}} \int_{I_{j,k}} f$ on $I_{j,k}$.

The Haar scaling functions and wavelets are discontinuous. The first construction of smoother wavelets with compact support is due to (Daubechies, 1992), and it allowed wavelets to be applied in signal processing and numerical analysis. Many generalizations of the original wavelet construction and MRA definition have been proposed. They allow to construct general families of wavelets in $\mathbb{R}^n$, with respect to a wide variety of dilation matrices. Other techniques (Sweldens, 1996, 1997) can be used for constructing wavelet-like systems on meshes, usually low-dimensional, but in principle in any number of dimensions.

The approximation properties of wavelets and their generalizations have been studied in many function spaces, for example spaces of piecewise smooth functions. A construction of wavelets on general graphs and manifolds has been recently proposed in (Coifman and Maggioni, 2004; Maggioni et al., 2005a; Bremer et al., 2004). This construction is based on natural diffusion operators defined on a graph and these wavelets have been called diffusion wavelets.

References

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

A. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Systems Journal*, 13:41–77, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 585–591. MIT Press, Cambridge, 2001.

M Belkin and P Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6(15):1373–1396, June 2003a.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003b.

M Belkin and P Niyogi. Using manifold structure for partially labelled classification. *Advances in NIPS*, 15, 2003c.

S Belongie, C Fowlkes, F Chung, and J Malik. Spectral partitioning with indefinite kernels using the Nyström extension. *ECCV*, 2002.

James C Bremer, Ronald R Coifman, Mauro Maggioni, and Arthur D Szlam. Diffusion wavelet packets. *Tech. Rep. YALE/DCS/TR-1304, Yale Univ., Appl. Comp. Harm. Anal., submitted*, Sep. 2004. doi: http://www.math.yale.edu/$\sim$mmm82/DiffusionWaveletPackets.pdf.

C. Chennubhotla and A. Jepson. Hierarchical eigensolver for transition matrices in spectral methods. In *Proceedings of the International Conference on Neural Information Processing Systems*. MIT Press, 2005.

F. G. Chung. The diameter and Laplacian eigenvalues of directed graphs. *Electronic Journal of Combinatorics*, 13(4), 2006.

Fan Chung. *Spectral Graph Theory*. Number 92. CBMS-AMS, May 1997.

Ronald R Coifman, Stephane Lafon, Ann Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data. part i: Diffusion maps. *Proc. of Nat. Acad. Sci.*, (102):7426–7431, May 2005a.

Ronald R Coifman, Stephane Lafon, Ann Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data. part ii: Multiscale methods. *Proc. of Nat. Acad. Sci.*, (102):7432–7438, May 2005b.

Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Tech. Rep. YALE/DCS/TR-1303, Yale Univ., Appl. Comp. Harm. Anal.*, Sep. 2004. doi: http://www.math.yale.edu/$\sim$mmm82/DiffusionWavelets.pdf. to appear.

Ronald R Coifman and Mauro Maggioni. Multiscale data analysis with diffusion wavelets. Tech. Rep. YALE/DCS/TR-1335, Dept. Comp. Sci., Yale University, September 2005.

Ronald R Coifman and Naoki Saito. Constructions of local orthonormal bases for classification and regression. *C. R. Acad. Sci. Paris*, 319 Série I:191–196, 1994.

Ronald R Coifman and Mladen V Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. Info. Theory*, 1992.

RR Coifman and S Lafon. Diffusion maps. *Appl. Comp. Harm. Anal.*, 2004a.

RR Coifman and S Lafon. Geometric harmonics. *Appl. Comp. Harm. Anal.*, 2004 2004b. Submitted.

I Daubechies. *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, 1992. ISBN 0-89871-274-2.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Joseph L Doob. *Classical Potential Theory and Its Probabilistic Counterpart*, volume XXV of *Classics in Mathematics.* Reprint of the 1st ed. berlin heidelberg new york 1984 2001 edition, 1984. ISBN 3-540-41206-9. Softcover.

P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. Technical Report YALEU/DCS/TR-1270, Yale University Department of Computer Science, New Haven, CT, February 2004.

P Drineas and M W Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *J. Machine Learning Research*, (6):2153–2175, 2005.

Lawrence C Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics.* American Mathematical Society, 1998.

S. Fine, Y. Singer, and N. Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1), July 1998.

C Fowlkes, S Belongie, and J Malik. Efficient spatiotemporal grouping using the nyström method. *CVPR*, 2001.

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

Lesli Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *J Comput Phys*, 73:325–348, 1987.

J. G. Kemeny and J. L. Snell. *Denumerable Markov Chains.* Number 40 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1976.

R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*, 2002.

J. Lafferty and G. Lebanon. Information diffusion kernels, 2002. URL `citeseer.ist.psu.edu/lafferty02information.html`.

John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *J. Mach. Learn. Res.*, 6:129–163, 2005. ISSN 1533-7928.

Stephane Lafon. *Diffusion maps and geometric harmonics.* PhD thesis, Yale University, Dept of Mathematics & Applied Mathematics, 2004.

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

M. Maggioni and S. Mahadevan. Fast direct policy evaluation using multiscale analysis of markov diffusion processes. In *University of Massachusetts, Department of Computer Science Technical Report TR-2005-39; submitted*, 2005.

Mauro Maggioni. Quantitative bounds on eigenfunctions of perturbed graphs. *preprint*, 2006.

Mauro Maggioni and Ronald R Coifman. Multiscale spectral analysis on data sets with diffusion wavelets. In *ICML, submitted*, 2006.

Mauro Maggioni, James C Bremer Jr, Ronald R Coifman, and Arthur D Szlam. Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs. August 2005a. Proc. SPIE Wavelet XI.

Mauro Maggioni, Arthur D Szlam, , Ronald R Coifman, and James C Bremer Jr. Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions. August 2005b. Proc. SPIE Wavelet XI.

S. Mahadevan. Proto-Value Functions: Developmental Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning*, 2005a.

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

S. Mahadevan. Samuel Meets Amarel: Automating Value Function Approximation using Global State Space Analysis. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, 2005c. AAAI Press/MIT Press.

S. Mahadevan, M. Ghavamzadeh, K. Rohanimanesh, and G. Theocharous. Hierarchical approaches to concurrency, multiagency, and partial observability. In *Learning and Approximate Dynamic Programming: Scaling up to the Real World.* 2004.

S. Mahadevan and M. Maggioni. Value function approximation with diffusion wavelets and laplacian eigenfunctions. In *University of Massachusetts, Department of Computer Science Technical Report TR-2005-38; Proc. NIPS 2005*, 2005.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *submitted*, 2006.

Stephane Mallat. *A wavelet tour in signal processing.* Academic Press, 1998.

A. McCallum, A. Corrada-Emmanuel, and X. Wang. The author-recipient-topic model for topic and role discovery in social networks: Experiments with enron and academic email. Technical Report UM-CS-2004-096, Department of Computer Science, University of Massachusetts, Amherst, 2004.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael Jordan, editor, *Learning in Graphical Models*, number 1, pages 355–368. 1999.

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm, 2001. URL `citeseer.ist.psu.edu/ng01spectral.html`.

Partha Niyogi and Mikhail Belkin. Semi-supervised learning on Riemannian manifolds. Technical Report TR-2001-30, University of Chicago, Computer Science Dept, Nov. 2001.

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

S Rosenberg. *The Laplacian on a Riemannian manifold*, volume 31 of *Student Texts.* Cambridge University Press, London Mathematical Society.

N Sidorova, O G Smolyanov, H v Weizsäcker, and O Wittich. Brownian motion close to submianifold of Riemannian manifolds. preprint, 2003.

N A Sidorova. Brownian motion on an embedded manifold as the limit of brownian motions with reflection in its tubular neighborhood. *Mathematical Notes*, 73(6):895–899, 2003.

E Stein. *Topics in Harmonic Analysis related to the Littlewood-Paley theory.* Princeton University Press, 1970.

W Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

W Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

G. Theocharous, K. Rohanimanesh, and S. Mahadevan. Learning hierarchical partially observable markov decision processes for robot navigation. In *IEEE Conference on Robotics and Automation (ICRA)*, 2001.

Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2000.

N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *In Proceedings of the ACM Conference on Embedded Networked Sensor Systems(Sensys04)*, November 2004.

H. Zha, C. Ding, M. Gu, X. He, , and H.D. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 1057–1064. MIT Press, Cambridge, 2001.

D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeld data on a directed graph. In *Proc. of the ICML*, 2005.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the ICML*, 2003.