Journal of Machine Learning Research 14 (2013) 2067-2118 Submitted 7/12; Revised 4/13; Published 7/13

# Construction of Approximation Spaces for Reinforcement Learning

**Wendelin Böhmer** WENDELIN@NI.TU-BERLIN.DE
*Neural Information Processing Group*
*Technische Universität Berlin*
*Marchstrasse 23, Berlin 10587, Germany*

**Steffen Grünewälder** STEFFEN@CS.UCL.AC.UK
*Centre for Computational Statistics and Machine Learning*
*University College London*
*London WC1E 6BT, United Kingdom*

**Yun Shen** YUN@NI.TU-BERLIN.DE
*Neural Information Processing Group*
*Technische Universität Berlin*
*Marchstrasse 23, Berlin 10587, Germany*

**Marek Musial** MUSIAL@CS.TU-BERLIN.DE
*Robotics Group*
*Technische Universität Berlin*
*Einsteinufer17, Berlin 10587, Germany*

**Klaus Obermayer** OBY@NI.TU-BERLIN.DE
*Neural Information Processing Group*
*Technische Universität Berlin*
*Marchstrasse 23, Berlin 10587, Germany*

**Editor:** Sridhar Mahadevan

## Abstract

Linear *reinforcement learning* (RL) algorithms like *least-squares temporal difference learning* (LSTD) require *basis functions* that span *approximation spaces* of potential value functions. This article investigates methods to construct these bases from samples. We hypothesize that an ideal approximation spaces should encode *diffusion distances* and that *slow feature analysis* (SFA) constructs such spaces. To validate our hypothesis we provide theoretical statements about the LSTD value approximation error and induced metric of approximation spaces constructed by SFA and the state-of-the-art methods *Krylov bases* and *proto-value functions* (PVF). In particular, we prove that SFA minimizes the average (over all tasks in the same environment) bound on the above approximation error. Compared to other methods, SFA is very sensitive to sampling and can sometimes fail to encode the whole state space. We derive a novel *importance sampling* modification to compensate for this effect. Finally, the LSTD and *least squares policy iteration* (LSPI) performance of approximation spaces constructed by Krylov bases, PVF, SFA and PCA is compared in benchmark tasks and a visual robot navigation experiment (both in a realistic simulation and with a robot). The results support our hypothesis and suggest that (i) SFA provides *subspace-invariant* features for MDPs with *self-adjoint* transition operators, which allows strong guarantees on the approximation error, (ii) the modified SFA algorithm is best suited for LSPI in both discrete and continuous state spaces and (iii) approximation spaces encoding diffusion distances facilitate LSPI performance.

**Keywords:** reinforcement learning, diffusion distance, proto value functions, slow feature analysis, least-squares policy iteration, visual robot navigation

## 1. Introduction

*Reinforcement learning* (RL, Sutton and Barto, 1998; Bertsekas and Tsitsiklis, 1996) provides a framework to autonomously learn *control policies* in stochastic environments and has become popular in recent years for controlling robots (e.g., Abbeel et al., 2007; Kober and Peters, 2009). The goal of RL is to compute a policy which selects *actions* that maximize the *expected future reward* (called *value*). An agent has to make these decisions based on the *state* $\boldsymbol{x} \in \mathcal{X}$ of the system. The state space $\mathcal{X}$ may be finite or continuous, but is in many practical cases too large to be represented directly. *Approximated RL* addresses this by choosing a function from *function set* $\mathcal{F}$ that resembles the true value function. Many function sets $\mathcal{F}$ have been proposed (see, e.g., Sutton and Barto, 1998; Kaelbling et al., 1996, for an overview). This article will focus on the space of *linear functions* with $p$ non-linear *basis functions* $\{\phi_i(\cdot)\}_{i=1}^p$ (Bertsekas, 2007), which we call *approximation space* $\mathcal{F}_\phi$.

The required basis functions $\phi_i(\cdot)$ are usually defined by hand (e.g., Sutton, 1996; Konidaris et al., 2011) and a bad choice can critically impede the accuracy of both the value estimate and the resulting control policy (see, e.g., Thrun and Schwartz, 1993). To address this issue, a growing body of literature has been devoted to the *construction* of basis functions and their theoretical properties (Mahadevan and Maggioni, 2007; Petrik, 2007; Parr et al., 2007; Mahadevan and Liu, 2010; Sun et al., 2011). Recently, the unsupervised method *slow feature analysis* (SFA, Wiskott and Sejnowski, 2002) has been proposed in this context (Legenstein et al., 2010; Luciw and Schmidhuber, 2012). This article presents a theoretical analysis of this technique and compares it with state-of-the-art methods. We provide theoretical statements for two major classes of automatically constructed basis functions (*reward-based* and *subspace-invariant* features, Parr et al., 2008) with respect to the induced Euclidean metric and the approximation error of *least-squares temporal difference learning* (LSTD, Bradtke and Barto, 1996). We also prove that under some assumptions SFA minimizes an average bound on the approximation error of all tasks in the same environment and argue that no better solution based on a single training sequence exists.

In practical applications (such as robotics) the state can not always be observed directly, but may be deducted from *observations*[1] $\boldsymbol{z} \in \mathcal{Z}$ of the environment. *Partial observable Markov decision processes* (POMDPs, Kaelbling et al., 1998) deal with the necessary inference of hidden states from observations. POMDPs are theoretically better suited, but become quickly infeasible for robotics. In contrast, this article focuses on another obstacle to value estimation: the *metric* associated with observation space $\mathcal{Z}$ can influence basis function construction. We assume for this purpose a unique one-to-one correspondence[2] between states $\boldsymbol{x} \in \mathcal{X}$ and observations $\boldsymbol{z} \in \mathcal{Z}$. To demonstrate the predicted effect we evaluate construction methods on a robotic visual navigation task. The observations are first-person perspective images, which exhibit a very different Euclidean metric than the underlying state of robot position and orientation. We hypothesize that *continuous SFA* (RSK-SFA, Böhmer et al., 2012) is not severely impeded by the change in observation metric and substantiate this in comparison to *continuous proto value functions* (PVF, Mahadevan and Maggioni, 2007) and *kernel PCA* (Schölkopf et al., 1998). We also confirm theoretical predictions that SFA is sensitive to the sampling policy (Franzius et al., 2007) and derive an *importance sampling* modification to compensate for these imbalances.

1. In this article the set of all possible observations $\mathcal{Z}$ is assumed to be a manifold in vector space $\mathbb{R}^d$.
2. This makes $\mathcal{Z}$ an isomorphism of $\mathcal{X}$, embedded in $\mathbb{R}^d$. The only difference is the associated *metric*. In the following we will continue to discriminate between $\mathcal{X}$ and $\mathcal{Z}$ for illustrative purposes.

### 1.1 Approximation Spaces

State spaces $\mathcal{X}$ can be finite or continuous.[3] We define the corresponding observation space $\mathcal{Z}$ to be isomorphic, but it may be governed by a very different *metric*. For example: finite $\mathcal{X}$ are usually equipped with a *discrete metric*, in which all states have equal distance to each other. Isomorphic observations $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^d$, on the other hand, might be equipped with an Euclidean metric in $\mathbb{R}^d$ instead. To approximate the *value function* $V(\cdot) : \mathcal{Z} \to \mathbb{R}$, one aims for a function $f(\cdot) \in \mathcal{F}$ that minimizes the *approximation error* w.r.t. some norm $\|V - f\|$. This article is focusing on the Euclidean $L_2$ norm[4] (see Section 2 for details), which depends on the metric's distance function. Besides different approximation errors, *generalization* to unseen states will also be very different in these spaces. This raises the question which metric is best suited to approximate value functions. Values are defined as the expected sum of *future* rewards. States with similar futures will therefore have similar values and are thus close by under an ideal metric. *Diffusion distances* compare the probabilities to end up in the same states (see, e.g., Coifman et al., 2005, and Section 4.1). It sands therefore to reason that a diffusion metric facilitates value approximation.

This article is using the term *approximation space* $\mathcal{F}_{\boldsymbol{\phi}}$ for the set of linear functions with $p$ non-linear basis functions $\phi_i : \mathcal{Z} \to \mathbb{R}$, $\mathcal{F}_{\boldsymbol{\phi}} := \left\{ f(\cdot) = \boldsymbol{w}^\top \boldsymbol{\phi}(\cdot) \,|\, \boldsymbol{w} \in \mathbb{R}^p \right\}$. Function approximation can be essentially performed by an inverse of the covariance matrix (see Section 2.3) and value estimation can be guaranteed to converge (Bertsekas, 2007). Nonetheless, the choice of basis functions $\boldsymbol{\phi} : \mathcal{Z} \to \mathbb{R}^p$ and thus approximation space $\mathcal{F}_{\boldsymbol{\phi}}$ will strongly affect approximation quality and generalization to unseen samples. An ideal approximation space should therefore (i) be able to approximate the value function well and (ii) be equipped with a Euclidean metric in $\{\boldsymbol{\phi}(\boldsymbol{z}) \,|\, \boldsymbol{z} \in \mathcal{Z}\}$ that resembles a diffusion metric. Approximation theory provides us with general functional bases that allow arbitrarily close approximation of continuous functions and thus fulfill (i), for example polynomials or a Fourier basis (Konidaris et al., 2011). However, those bases can usually not be defined on high-dimensional observation spaces $\mathcal{Z}$, as they are prone to the *curse of dimensionality*.

A straightforward approach to basis construction would extract a low-dimensional manifold of $\mathcal{Z}$ and construct a general function base on top of it. This can be achieved by manifold extraction (Tenenbaum et al., 2000; Jenkins and Mataric, 2004) or by computer vision techniques (e.g., Visual SLAM, Smith et al., 1990; Davison, 2003), which require extensive knowledge of the latent state space $\mathcal{X}$. Some approaches construct basis functions $\boldsymbol{\phi}(\cdot)$ directly on the observations $\boldsymbol{z} \in \mathcal{Z}$, but are either restricted to linear maps $\boldsymbol{\phi}(\cdot)$ (PP, Sprague, 2009) or do not generalize to unseen samples (ARE, Bowling et al., 2005). None of the above methods extracts $\mathcal{X}$ in a representation that encodes a diffusion metric.

Recent analysis of the approximation error has revealed two opposing approaches to basis construction: *reward-based* and *subspace-invariant* features (Parr et al., 2008). The former encode the propagated reward function and the latter aim for eigenvectors of the transition matrix. Section 3.1 provides an overview of the reward-based *Krylov bases* (Petrik, 2007), *Bellman error basis functions* (BEBF, Parr et al., 2007), *Bellman average reward bases* (BARB, Mahadevan and Liu, 2010) and *Value-function of the Bellman error* bases (V-BEBF, Sun et al., 2011). All of these algorithms are defined exclusively for finite state spaces. The encoded metric is investigated in Section 4.1. *Proto-value functions* (PVF, Mahadevan and Maggioni, 2007, and Section 3.3) are the state-of-the-

3. This article does not discuss discrete countable infinite state spaces, which are isomorphisms to $\mathbb{N}$.

4. Other approaches are based on the $L_\infty$ norm (Guestrin et al., 2001; Petrik and Zilberstein, 2011) or the $L_1$ norm (de Farias and Roy, 2003). However, all norms eventually depend on the metric of $\mathcal{X}$ or $\mathcal{Z}$.

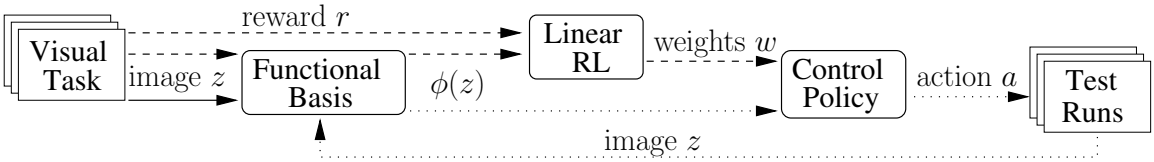


Figure 1: Scheme of a general RL architecture for visual tasks. First (solid arrow) one or many visual tasks generate images $z$ to train a *representation* $\phi : \mathcal{Z} \to \mathbb{R}^p$, which is a *functional basis* in the true state space $\mathcal{X}$. Given such a basis (dashed) one task is used to train a *control policy* with a *linear RL* algorithm. In the verification phase (dotted) the trained control policy generates multiple test trajectories.

art subspace-invariant feature construction method. In finite state spaces PVF are the eigenvectors to the smallest eigenvalues of the normalized graph Laplacian of an undirected graph representing the *transition possibility* (not *probability*) between states. As the calculation requires no knowledge of the reward, this technique has proven useful to transfer knowledge between different tasks in the same environment (Ferguson and Mahadevan, 2006; Ferrante et al., 2008). In Section 4.3 we will explain this observation by defining the class of learning problems for which this transfer is nearly optimal. To cope with continuous state or observation spaces, there also exists an extension based on the PVF of a k-nearest neighbors graph and Nyström approximation between the graph nodes (Mahadevan and Maggioni, 2007). However, as this approach is based on neighborhood relationships in $\mathcal{Z}$, the solution will not preserve diffusion distances.

An extension preserving these distances are Laplacian eigenmaps (Belkin and Niyogi, 2003) of the *transition operator*. Recently Sprekeler (2011) has shown that *slow feature analysis* (SFA, Wiskott and Sejnowski, 2002, and Section 3.4) approximates Laplacian eigenmaps. In the limit of an infinite training sequence, it can be shown (under mild assumptions) that the resulting non-linear SFA features span a Fourier basis in the unknown state space $\mathcal{X}$ (Wiskott, 2003). Franzius et al. (2007) show additionally that the *order* in which the basis functions are encoded is strongly dependent on the *relative velocities* in different state dimensions. This can lead to an insufficient approximation for *low dimensional*, but has little effect on *high dimensional* approximation spaces. Section 3.5 addresses this problem with an *importance sampling* modification to SFA.

### 1.2 Visual Tasks

Most benchmark tasks in RL have either a finite or a continuous state space with a well behaving Euclidean metric.[5] Theoretical statements in Section 4 predict that SFA encodes diffusion distances, which are supposed to facilitate generalization. Testing this hypothesis requires a task that can be solved either with a well behaving true state $x \in \mathcal{X}$ or based on observations $z \in \mathcal{Z} \subset \mathbb{R}^d$ with a disadvantageous metric. Value functions approximated w.r.t. a diffusion metric, that is, with SFA features, should provide comparable performance in both spaces. Based on a method that encodes only Euclidean distances in $\mathcal{Z}$ (e.g., PCA), on the other hand, the performance of the approximated value functions should differ.

5. An example for finite state spaces is the *50-state chain* (Section 5.2). The *puddle-world task* (Section 5.3) and the *mountain-car task* (not evaluated) have been defined with continuous and with discrete state spaces (Boyan and Moore, 1995; Sutton, 1996). Both continuous tasks are defined on well-scaled two dimensional state spaces. Euclidean distances in these spaces resemble diffusion distances closely.

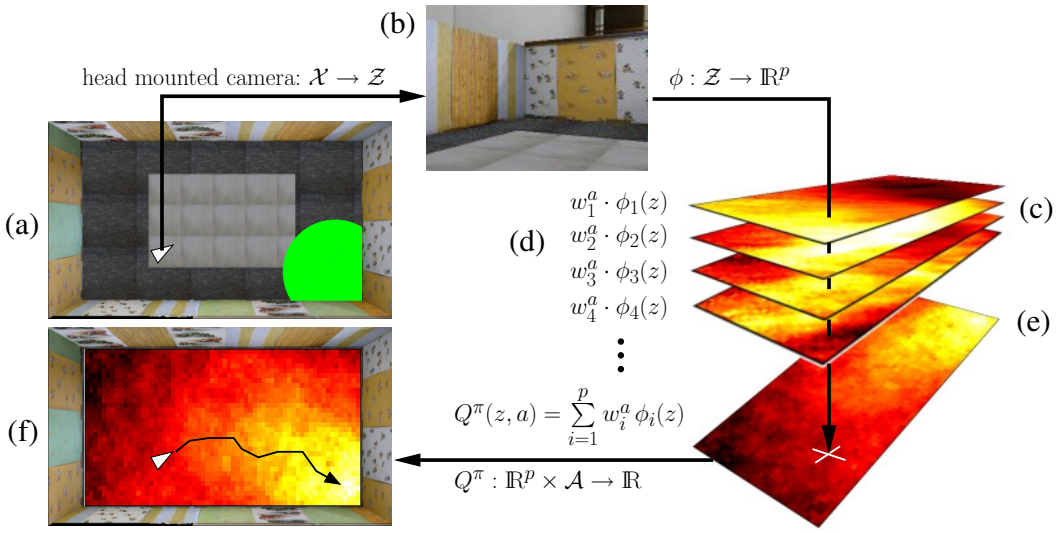


Figure 2: The figure shows the visual control process to guide a robot into the circular goal area. At its position (a), the robot observes an image with its head mounted camera (b). A function $\phi(\cdot)$, generated by one of the discussed unsupervised methods, maps the image into a $p$-dimensional feature space (c). For each action $a \in \mathcal{A}$, these features are weighted by the LSPI parameter vector $\boldsymbol{w}^a \in \mathbb{R}^p$ (d), giving rise to the Q-value function $Q^{\boldsymbol{\pi}}(\cdot,\cdot)$ (e). The control always chooses the action $a$ with the highest Q-value (f).

Applied to visual input, for example camera images, this class of problems is called *visual tasks*. Setting problems of *partial observability* aside, the true state $\boldsymbol{x}$ is usually assumed to be sufficiently *represented* by a set of hand-crafted features of $\boldsymbol{z}$. However, there is no straightforward way to extract the state reliably out of visual data without introducing artificial markers to the environment. Current approaches to visual tasks aim thus to learn a feature mapping $\boldsymbol{\phi} : \mathcal{Z} \to \mathbb{R}^p$ from observations $\boldsymbol{z}$, without loosing too much information about the true state $\boldsymbol{x}$ (see Jodogne and Piater, 2007, for an overview). Figure 1 shows a sketch of a general RL architecture to solve visual tasks with linear RL methods. Here we first learn an *image representation* $\boldsymbol{\phi} : \mathcal{Z} \to \mathbb{R}^p$ from experience (solid arrow), collected in one or more visual tasks within the same environment. To learn a *control policy* (dashed arrows) the agent treats the representation $\boldsymbol{\phi}(\boldsymbol{z}) \in \mathbb{R}^p$ of each observed image $\boldsymbol{z} \in \mathcal{Z}$ as the representation of the corresponding state $\boldsymbol{x} \in \mathcal{X}$. A linear RL algorithm can estimate future *rewards* $r \in \mathbb{R}$ by approximating the linear *Q-value function* $Q^{\boldsymbol{\pi}} : \mathbb{R}^p \times \mathcal{A} \to \mathbb{R}$ with *weight vector* $\boldsymbol{w} \in \mathbb{R}^{p|\mathcal{A}|}$. The control policy always chooses the *action* $a \in \mathcal{A}$ with the highest Q-value predicted by $Q^{\boldsymbol{\pi}}$ and can be verified by independent test runs from random start positions (dotted arrows). For example, in the context of navigation, Lange and Riedmiller (2010) employed a *deep auto-encoder* (Hinton and Osindero, 2006), Legenstein et al. (2010) hierarchical nonlinear *slow feature analysis* (SFA, Wiskott and Sejnowski, 2002) and Luciw and Schmidhuber (2012) *incremental SFA* (Kompella et al., 2012) to represent the underlying state space. The control problem was subsequently solved by different *approximate RL* algorithms. All above works verified

their approaches on a very regular observation space $\mathcal{Z}$ by providing the agent with a bird's eye view of an artificial world, in which a set of pixels determines the agents position uniquely.

To yield a less ideal observation space $\mathcal{Z}$, the visual navigation task in Section 5.4 observes first-person perspective images[6] instead. Figure 2 shows the control loop of the robot. The true state $\boldsymbol{x}$ is the robot's position at which an image $\boldsymbol{z}$ is taken by a head-mounted camera. $\mathcal{X}$ is continuous and in principle the actions $a \in \mathcal{A}$ should be continuous too. However, selecting continuous actions is not trivial and for the sake of simplicity we restricted the agent to three discrete actions: move forward and turn left or right.

### 1.3 Contributions

The contributions of this article are threefold:

1. We provide theoretical statements about the encoded *diffusion metric* (Section 4.1) and the LSTD value *approximation error* (Section 4.2) of both reward-based (Krylov bases) and subspace-invariant (SFA) features. We also prove that SFA minimizes an average bound on the approximation error of a particular set of tasks (Section 4.3). We conclude that *SFA can construct better approximation spaces for LSTD than PVF* and demonstrate this on multiple discrete benchmark tasks (Sections 5.1 to 5.3).

2. We investigate the role of the metric in approximation space $\mathcal{F}_\phi$ on a visual robot navigation experiment, both in a realistic simulation and on a robot (Sections 5.4 to 5.7). We demonstrate than SFA can sometimes fail to encode the whole state space due to its dependence on the sampling policy and address this problem with a novel *importance sampling* modification to the SFA algorithm.

3. We compare the performance of approximation spaces constructed by Krylov bases, PVF, SFA and PCA for *least-squares policy iteration* (LSPI, Lagoudakis and Parr, 2003). Results suggest that (i) the modified SFA algorithm is best suited for LSPI in both discrete and continuous state spaces and (ii) approximation spaces that encode a diffusion metric facilitate LSPI performance.

Both theoretical and empirical results leave room for interpretation and unresolved issues for future works. Section 6 discusses open questions as well as potential solutions. Finally, the main results and conclusions of this article are summarized in Section 7.

## 2. Reinforcement Learning

In this section we review *reinforcement learning* in potentially continuous state spaces $\mathcal{X}$, which require a slightly more complicated formalism than used in standard text books (e.g., Sutton and Barto, 1998). The introduced notation is necessary for Section 4 and the corresponding proofs in Appendix A. However, casual readers familiar with the RL problem can skip this section and still comprehend the more practical aspects of the article.

There exist many linear RL algorithms one could apply to our experiment, like *temporal difference learning* (TD($\lambda$), Sutton and Barto, 1998) or *Q-learning* (Watkins and Dayan, 1992). We

6. Rotating a camera by some degrees represents only a minor change in its orientation and therefore in $\mathcal{X}$, but shifts all pixels and can lead to very large Euclidean distances in $\mathcal{Z}$. Moving slightly forward, on the other hand, changes only the pixels of objects close by and thus yields a much smaller distance in $\mathcal{Z}$.

chose here the *least-squares policy iteration* algorithm (LSPI, Lagoudakis and Parr, 2003, see Section 2.4), because the underlying *least-squares temporal difference* algorithm (LSTD, Bradtke and Barto, 1996, see Section 2.3) is the most sample effective unbiased value estimator (Grünewälder and Obermayer, 2011). For practical implementation we consider *sparse kernel methods*, which are introduced in Section 2.5.

### 2.1 The Reinforcement Learning Problem

We start with the definition of a *Markov decision process* (MDP). Let $\mathcal{B}(\mathcal{X})$ denote the collection of all *Borel sets* of set $\mathcal{X}$. A *Markov decision process* is a tuple $(\mathcal{X}, \mathcal{A}, P, R)$. In our setup, $\mathcal{X}$ is a *finite* or *compact continuous*[7] state space and $\mathcal{A}$ the finite[8] action space. The *transition kernel*[9] $P : \mathcal{X} \times \mathcal{A} \times \mathcal{B}(\mathcal{X}) \to [0,1]$ represents the probability $P(A|x,a)$ to end up in set $A \in \mathcal{B}(\mathcal{X})$ after executing action $a$ in state $x$. $R : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \times \mathcal{B}(\mathbb{R}) \to [0,1]$ is a distribution over *rewards*: $R(B|x,a,y)$ is the probability to receive a reward within set $B \in \mathcal{B}(\mathbb{R})$ after a transition from state $x$ to state $y$, executing action $a$. In our context, however, we will be content with the mean *reward function* $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, defined as $r(x,a) = \int_{\mathbb{R}} \int_{\mathcal{X}} r R(dr|x,a,y) P(dy|x,a), \forall x \in \mathcal{X}, a \in \mathcal{A}$. A control *policy* $\pi : \mathcal{X} \times \mathcal{B}(\mathcal{A}) \to [0,1]$ is a conditional distribution of actions given states. The goal of *reinforcement learning* is to find a policy that maximizes the *value* $V^\pi(x)$ at each state $x$, that is the expected sum of discounted future rewards

$$V^\pi(x_0) := \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \;\middle|\; \begin{array}{rcl} a_t & \sim & \pi(\cdot|x_t) \\ x_{t+1} & \sim & P(\cdot|x_t, a_t) \end{array} \right], \quad \forall x_0 \in \mathcal{X}.$$

Here the *discount factor* $\gamma \in [0,1)$ determines the relative importance of short term to long term rewards.[10] The value function can also be expressed *recursively*:

$$V^\pi(x) = \int r(x,a)\, \pi(da|x) + \gamma \iint V^\pi(y) P(dy|x,a)\, \pi(da|x), \quad \forall x \in \mathcal{X}.$$

In finite state (and action) spaces this equation can be solved by dynamic programming. Note that for fixed $V^\pi(\cdot)$ the equation is linear in the policy $\pi(\cdot|\cdot)$ and vice versa, allowing an *expectation maximization* type algorithm called *policy iteration* (PI, Sutton and Barto, 1998) to find the best policy. To allow for continuous state spaces, however, we need to translate this formalism into a Hilbert space.

### 2.2 MDP in Hilbert Spaces

For the sake of feasibility we will restrict our discussion to value functions $v^\pi \in L^2(\mathcal{X}, \xi)$ from the space of *square-integrable functions* on $\mathcal{X}$, endowed with probability measure $\xi : \mathcal{B}(\mathcal{X}) \to [0,1]$, $\int \xi(dx) = 1$. This Hilbert space contains $\xi$-measurable functions and should suffice for all

7. Compact state spaces $\mathcal{X}$ are necessary for *ergodicity*, see Footnote 12 on Page 2074. All finite $\mathcal{X}$ are compact.
8. For generality we maintain the notation of continuous compact action spaces as long as possible.
9. Following probability theory, a *kernel* denotes here a *conditional measure* over some set, in this article $\mathcal{X} \times \mathcal{A}$ or just $\mathcal{X}$. If this measure over the whole set is always one then it is called a *transition* or *Markov kernel*. Note that the *Radon-Nikodym derivative* of a kernel w.r.t. the uniform measure is called a *kernel function* in integral calculus. Note also the difference to *positive semi-definite kernels* in the context of RKHS (see Section 3.2).
10. In classical decision theory, $\gamma$ can be interpreted as the continuous version of a maximal search depth in the decision tree. Alternatively, one can see $\gamma^t$ as shrinking certainty about predicted rewards.

continuous setups. The induced *inner product* and *norm* are

$$\langle f,g\rangle_\xi = \int f(x)\,g(x)\,\xi(dx) \quad \text{and} \quad \|f\|_\xi = \langle f,f\rangle_\xi^{1/2}, \quad \forall f,g \in L^2(\mathcal{X},\xi)\,.$$

For a fixed policy $\pi$, this yields the *transition operator*[11] $\hat{P}^\pi : L^2(\mathcal{X},\xi) \to L^2(\mathcal{X},\xi)$,

$$\hat{P}^\pi[f](x) := \iint f(y)\,P(dy|x,a)\,\pi(da|x)\,, \quad \forall x \in \mathcal{X}\,, \quad \forall f \in L^2(\mathcal{X},\xi)\,.$$

The operator is called *ergodic* if every Markov chain sampled by the underlying transition kernel $P$ and policy $\pi$ is ergodic.[12] This is a convenient assumption as it implies the existence of a *steady state distribution* $\xi$, which we will use as measure of $L^2(\mathcal{X},\xi)$. This also implies

$$\xi(B) = \iint P(B|x,a)\,\pi(da|x)\,\xi(dx)\,, \quad \forall B \in \mathcal{B}(\mathcal{X})\,.$$

Under the assumption that all rewards are *bounded*, that is, $|r(x,a)|^2 < \infty \Rightarrow \exists r^\pi \in L^2(\mathcal{X},\xi)$ : $r^\pi(x) := \int r(x,a)\,\pi(da|x), \forall x \in \mathcal{X}, \forall a \in \mathcal{A}$, we can define the *Bellman operator* in $L^2(\mathcal{X},\xi)$

$$\hat{B}^\pi[f](x) := r^\pi(x) + \gamma \hat{P}^\pi[f](x)\,, \quad \forall x \in \mathcal{X}\,, \quad \forall f \in L^2(\mathcal{X},\xi)\,,$$

which performs *recursive value propagation*. This is of particular interest as one can show[13] that $\hat{B}^\pi[f]$ is a contract mapping in $\|\cdot\|_\xi$ and an infinite application starting from any function $f \in L^2(\mathcal{X},\xi)$ converges to the *true* value function $v^\pi \in L^2(\mathcal{X},\xi)$.

### 2.3 Least-squares Temporal Difference Learning

Infinitely many applications of the Bellman operator $\hat{B}^\pi[\cdot]$ are not feasible in practice. However, there exist an efficient solution if one restricts oneself to an approximation from $\mathcal{F}_\phi = \{f(\cdot) = w^\top \phi(\cdot)\,|\,w \in \mathbb{R}^p\} \subset L^2(\mathcal{X},\xi)$. For linearly independent basis functions $\phi_i \in L^2(\mathcal{X},\xi)$ the projection of any function $f \in L^2(\mathcal{X},\xi)$ into $\mathcal{F}_\phi$ w.r.t. norm $\|\cdot\|_\xi$ can be calculated by the linear *projection operator* $\hat{\Pi}_\xi^\phi : L^2(\mathcal{X},\xi) \to L^2(\mathcal{X},\xi)$,

$$\hat{\Pi}_\xi^\phi[f](x) := \sum_{j=1}^p \overbrace{\sum_{i=1}^p \langle f,\phi_i\rangle_\xi\,(\mathbf{C}^{-1})_{ij}}^{w_j \in \mathbf{R}}\,\phi_j(x)\,, \quad C_{ij} := \langle \phi_i,\phi_j\rangle_\xi\,, \quad \forall x \in \mathcal{X}\,, \quad \forall f \in L^2(\mathcal{X},\xi)\,.$$

Instead of infinitely many alternating applications of $\hat{B}^\pi$ and $\hat{\Pi}_\xi^\phi$, one can directly calculate the fixed point $f^\pi \in \mathcal{F}_\phi$ of the combined operator

$$f^\pi \stackrel{!}{=} \hat{\Pi}_\xi^\phi\big[\hat{B}^\pi[f^\pi]\big] \qquad \Rightarrow \qquad w^\pi = \Big(\underbrace{\langle \phi, \phi - \gamma \hat{P}^\pi[\phi]\rangle_\xi}_{\mathbf{A}^\pi \in \mathbf{R}^{p\times p}}\Big)^\dagger \underbrace{\langle \phi, r^\pi\rangle_\xi}_{b^\pi \in \mathbf{R}^p}\,,$$

11. Every *kernel* $A : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0,\infty)$ induces a linear operator $\hat{A} : L^2(\mathcal{X},\xi) \to L^2(\mathcal{X},\xi), \hat{A}[f](x) := \int A(dy|x)\,f(y), \forall x \in \mathcal{X}, \forall f \in L^2(\mathcal{X},\xi)$, which in this article bears the same name with a hat.
12. A Markov chain is called *ergodic* if it is *aperiodic* and *positive recurrent*: if there is a nonzero probability to break any periodic cycle and if any infinite sequence eventually must come arbitrarily close to every state $x \in \mathcal{X}$. This is a property of the transition kernel rather than the policy. If *one* policy that assigns a nonzero probability to each action yields ergodic Markov chains, then *every* such policy does. Of course this does not hold for deterministic policies.
13. In a straightforward extension of the argument for finite state spaces (Bertsekas, 2007, Chapter 6).

where $\boldsymbol{w}^{\boldsymbol{\pi}} \in \mathbb{R}^p$ denotes the corresponding parameter vector of fixed point $f^{\boldsymbol{\pi}}(x) = (\boldsymbol{w}^{\boldsymbol{\pi}})^\top \boldsymbol{\phi}(x)$, $\forall x \in \mathcal{X}$, $(\mathbf{A}^{\boldsymbol{\pi}})^\dagger$ denotes the *Moore-Penrose pseudo-inverse* of matrix $\mathbf{A}^{\boldsymbol{\pi}}$ and we wrote $(\langle \boldsymbol{\phi}, \boldsymbol{\phi} \rangle_\xi)_{ij} = \langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle_\xi$ for convenience.

The stochastic matrices $\mathbf{A}^{\boldsymbol{\pi}}$ and $\boldsymbol{b}^{\boldsymbol{\pi}}$ can be bias-free estimated given a set of transitions $\left\{ \boldsymbol{\phi}(x_t) \xrightarrow{r_t} \boldsymbol{\phi}(x'_t) \right\}_{t=1}^n$ of start states $x_t \sim \xi(\cdot)$, executed actions $a_t \sim \boldsymbol{\pi}(\cdot|x_t)$, corresponding successive states $x'_t \sim P(\cdot|x_t, a_t)$ and received rewards $r_t \sim R(\cdot|x_t, a_t, x'_t)$. The resulting algorithm is known as *least-squares temporal difference learning* (LSTD, Bradtke and Barto, 1996). It can be shown that it converges[14] in $\|\cdot\|_\xi$ norm (Bertsekas, 2007). Note that for this property the samples must be drawn from steady state distribution $\xi$ and policy $\boldsymbol{\pi}$, usually by a long Markov chain executing $\boldsymbol{\pi}$.

Moreover, Tsitsiklis and Van Roy (1997) have proven that in $\|\cdot\|_\xi$ norm the error between true value function $v^{\boldsymbol{\pi}} \in L^2(\mathcal{X}, \xi)$ and approximation $f^{\boldsymbol{\pi}} \in \mathcal{F}_{\boldsymbol{\phi}}$ is bounded[15] by

$$\left\| v^{\boldsymbol{\pi}} - f^{\boldsymbol{\pi}} \right\|_\xi \leq \frac{1}{\sqrt{1-\gamma^2}} \left\| v^{\boldsymbol{\pi}} - \hat{\Pi}_\xi^{\boldsymbol{\phi}} \left[ v^{\boldsymbol{\pi}} \right] \right\|_\xi .$$

In Section 4 we will improve upon this bound significantly for a special case of SFA features. We will also show that for a specific class of tasks the basis functions $\boldsymbol{\phi}_i(\cdot)$ extracted by SFA minimize a mean bound on the right hand side of this equation, in other words minimize the mean approximation error over all considered tasks.

### 2.4 Least-squares Policy Iteration

Estimating the value function does not directly yield a control policy. This problem is tackled by *least-squares policy iteration* (LSPI, Lagoudakis and Parr, 2003), which alternates between *Q-value estimation* (the expectation step) and *policy improvement* (the maximization step). At iteration $i$ with current policy $\boldsymbol{\pi}_i$, the *Q-value function* $Q^{\boldsymbol{\pi}_i} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is defined as the value of state $x \in \mathcal{X}$ conditioned on the next action $a \in \mathcal{A}$:

$$Q^{\boldsymbol{\pi}_i}(x,a) := r(x,a) + \gamma \int V^{\boldsymbol{\pi}_i}(y) \, P(dy|x,a) = r(x,a) + \gamma \iint Q^{\boldsymbol{\pi}_i}(y,b) \, \boldsymbol{\pi}_i(db|y) \, P(dy|x,a) .$$

Note that *Q-value estimation* is equivalent to *value estimation* in the space of twice integrable functions over the space of state-action pairs $\mathcal{X} \times \mathcal{A}$ endowed with probability measure $\mu(B,A) := \int_B \boldsymbol{\pi}_i(A|x) \, \xi(dx)$, $\forall (B,A) \in \mathcal{B}(\mathcal{X}) \times \mathcal{B}(\mathcal{A})$, that is, $L^2(\mathcal{X} \times \mathcal{A}, \mu)$. The corresponding transition operator $\hat{P}_Q^{\boldsymbol{\pi}_i} : L^2(\mathcal{X} \times \mathcal{A}, \mu) \to L^2(\mathcal{X} \times \mathcal{A}, \mu)$ is

$$\hat{P}_Q^{\boldsymbol{\pi}_i}[f](x,a) := \iint f(y,b) \, \boldsymbol{\pi}_i(db|y) \, P(dy|x,a) , \quad \forall f \in L^2(\mathcal{X} \times \mathcal{A}, \mu) , \quad \forall x \in \mathcal{X} , \quad \forall a \in \mathcal{A} .$$

The greedy policy $\boldsymbol{\pi}_{i+1}$ in the $i$'th *policy improvement* step will for each state $x$ draw one of the actions with the highest Q-value, that is, $a^{\boldsymbol{\pi}_i}(x) \sim \arg\max_{a \in \mathcal{A}} Q^{\boldsymbol{\pi}_i}(x,a)$, and stick with it:

$$\boldsymbol{\pi}_{i+1}(a|x) := \begin{cases} 1 & ,\text{if} \quad a = a^{\boldsymbol{\pi}_i}(x) \\ 0 & ,\text{else} \end{cases} , \quad \forall x \in \mathcal{X} , \quad \forall a \in \mathcal{A} .$$

14. In a straightforward extension of the argument for finite state spaces (Bertsekas, 2007, Chapter 6).
15. Besides this bound in the weighted $L_2$ norm there exists a multitude of bounds in $L_\infty$ and sometimes $L_1$ norm. See Petrik and Zilberstein (2011) for a recent overview.

In finite state-action spaces, this procedure will provably converge to a policy that maximizes the value for all states (Kaelbling et al., 1996).

To cope with continuous state and/or action spaces, LSPI employs the LSTD algorithm to estimate approximated Q-value functions $f^{\pi_i} \in \mathcal{F}_\phi$, where the basis functions $\phi_i \in L^2(\mathcal{X} \times \mathcal{A}, \mu)$ are defined over state-action pairs rather than states alone. In difference to value estimation, any experienced set of transitions $\{(x_t, a_t) \xrightarrow{r_t} x'_t\}_{t=1}^n$ yields the necessary information for Q-value estimation with arbitrary policies $\pi_i$, in other words the LSTD training set

$$\left\{ \phi(x_t, a_t) \xrightarrow{r_t} \int \phi(x'_t, a)\, \pi_i(da|x'_t) \right\}_{t=1}^n .$$

However, convergence guarantees hold *only* when $\mu$ is the steady state distribution of $P_Q^{\pi_i}$, which usually only holds in the first iteration. Although it can thus not be guaranteed, empirically LSPI fails only for large function spaces $\mathcal{F}_\phi$ and $\gamma$ close to 1. In Section 5, Figure 8, we demonstrate this at the example of well and poorly constructed basis functions.

The easiest way to encode $p$ state-action pairs for finite action spaces $\mathcal{A}$ is to use an arbitrary $q := p/|\mathcal{A}|$ dimensional state encoding $\phi : \mathcal{X} \to \mathbb{R}^q$ and to extend it by $\bar{\phi}(x,a) := \phi(x) e_a^\top$, $\forall x \in \mathcal{X}, \forall a \in \mathcal{A}$, where $e_a \in \mathbb{R}^{|\mathcal{A}|}$ is a column vector of length 1 which is 0 everywhere except in one dimension uniquely associated with action $a$. The resulting $q \times |\mathcal{A}|$ matrix $\bar{\phi}(x,a)$ can be treated as set of $p$ state-action basis functions.

### 2.5 Reproducing Kernel Hilbert Spaces

Although $L^2(\mathcal{X}, \xi)$ is a very powerful tool for analysis, it has proven problematic in machine learning (Wahba, 1990; Schölkopf and Smola, 2002). Many algorithms employ instead the well behaving *reproducing kernel Hilbert spaces* $\mathcal{H}_\kappa \subset L^2(\mathcal{Z}, \xi)$ (RKHS, see, e.g., Schölkopf and Smola, 2002). A RKHS is induced by a *positive semi-definite kernel function* $\kappa : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$; the set $\{\kappa(\cdot, x) \mid x \in \mathcal{Z}\}$ is a full (but not orthonormal) basis of $\mathcal{H}_\kappa$. The inner product of two kernel functions in $\mathcal{H}_\kappa$ can be expressed as a kernel function itself. Take the example of the *Gaussian kernel* used in this article:

$$\langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_{\mathcal{H}_\kappa} \quad = \quad \kappa(x, y) \quad := \quad \exp\left(-\tfrac{1}{2\sigma^2} \|x - y\|_2^2\right), \qquad \forall x, y \in \mathcal{Z}.$$

Due to compactness of $\mathcal{Z}$, all continuous functions $f$ in $L^2(\mathcal{Z}, \xi)$ can be approximated arbitrarily well in $L_\infty$ (supremum) norm by functions from $\mathcal{H}_\kappa$.

Naive implementation of the *kernel trick* with $n$ observed samples $\{z_t\}_{t=1}^n$ induces a computational complexity of $\mathcal{O}(n^3)$ and a memory complexity of $\mathcal{O}(n^2)$. For large $n$ it can thus be necessary to look for approximate solutions in the subspace spanned by some *sparse subset* $\{s_i\}_{i=1}^m \subset \{z_t\}_{t=1}^n$, $m \ll n$, and thus $f(\cdot) = \sum_{i=1}^m \alpha_i \kappa(\cdot, s_i) \in \mathcal{H}_\kappa$, $\alpha \in \mathbb{R}^m$ (*projected process* matrix sparsification, Rasmussen and Williams, 2006). If subset and approximation space are chosen well, the LSTD solution $f^\pi \in \mathcal{F}_\phi$ can be approximated well too:

$$f^\pi \quad \in \quad \underbrace{\mathcal{F}_\phi \quad \subset \quad \mathcal{F}_{\{\kappa(\cdot, s_i)\}_{i=1}^m}}_{\text{approximation space}} \qquad \underset{\text{subset selection}}{\subset} \qquad \underbrace{\mathcal{H}_\kappa \quad \subset \quad L^2(\mathcal{Z}, \xi)}_{\text{all continuous functions}} .$$

However, finding a suitable subset is not trivial (Smola and Schölkopf, 2000; Csató and Opper, 2002). We employ the *matching pursuit for maximization of the affine hull* algorithm (MP-MAH,

Böhmer et al., 2012) to find a uniformly distributed subset. Section 5.5 empirically evaluates the effect of this choice.

A MDP with *discrete state space* $\mathcal{Z} := \{x_i\}_{i=1}^d$ can also be embedded as a RKHS. The kernel $\kappa(x_i, x_j) = \delta_{ij}$ induces the *discrete metric*, where $\delta_{ij}$ is the *Kronecker delta*. In this metric every state is an *open set* and thus Borel sets, integrals, ergodicity and all other concepts in this section can be extended to discrete state spaces. The discrete metric does not allow generalization to neighboring states, though. In this case the "sparse" subsets of the kernel algorithms discussed in Section 3 must contain *all* states, that is, $\{s_i\}_{i=1}^m = \{x_i\}_{i=1}^d$, which restricts the formalism to *finite* or *compact continuous* state spaces.

## 3. Basis Function Construction

Solving an MDP with the methods discussed in Section 2 requires the projection into an approximation space $\mathcal{F}_\phi = \{f(\cdot) = \boldsymbol{w}^\top \boldsymbol{\phi}(\cdot) \,|\, \boldsymbol{w} \in \mathbb{R}^p\} \subset L^2(\mathcal{Z}, \xi)$. The discussed algorithms make it necessary to specify the involved basis functions $\phi_i(\cdot) \in L^2(\mathcal{Z}, \xi), \forall i \in \{1, \dots, p\}$, *before* training, though. As the true value function $v^\pi \in L^2(\mathcal{Z}, \xi)$ is initially unknown, it is not obvious how to pick a basis that will eventually approximate it well. Classical choices (like Fourier bases, Konidaris et al., 2011) are known to approximate *any* continuous function arbitrarily well in the limit case. However, if applied on high-dimensional observations, for example, $\boldsymbol{z} \in \mathbb{R}^d$, the number of required functions $p$ scales exponentially with $d$. It would therefore be highly advantageous to exploit knowledge of task or observation space and *construct* a low dimensional basis.

In this context, recent works have revealed two diametrically opposed concepts (Parr et al., 2008). Expressed in the notation of Section 2, the *Bellman error* of the fixed point solution $f^\pi(\cdot) = (\boldsymbol{w}^\pi)^\top \boldsymbol{\phi}(\cdot) \overset{!}{=} \hat{\Pi}_\xi^\phi [\hat{B}^\pi [f^\pi]] \in \mathcal{F}_\phi$ can be separated into two types of error functions,

$$\hat{B}^\pi [f^\pi] - f^\pi \quad = \quad \underbrace{\left(\hat{I} - \hat{\Pi}_\xi^\phi\right)[r^\pi]}_{\Delta^r \in L^2(\mathcal{Z}, \xi)} \quad + \quad \gamma \sum_{i=1}^p w_i^\pi \underbrace{\left(\hat{I} - \hat{\Pi}_\xi^\phi\right)\left[\hat{P}^\pi [\phi_i]\right]}_{\Delta_i^\phi \in L^2(\mathcal{Z}, \xi)},$$

the *reward error* $\Delta^r \in L^2(\mathcal{Z}, \xi)$ and the *per-feature errors* $\Delta_i^\phi \in L^2(\mathcal{Z}, \xi)$. Correspondingly, there have been two opposing approaches to basis function construction in literature:

1. *Reward-based features* encode the reward function and how it propagates in time. $\Delta^r$ is thus zero everywhere, but $\Delta_p^\phi$ can still induce Bellman errors.

2. *Subspace-invariant features* aim for eigenfunctions of transition operator $\hat{P}^\pi$ to achieve no per-feature errors. $\Delta^r$, however, can still induce Bellman errors.

This article focuses on subspace-invariant feature sets, but reward-based features are introduced for comparison in Section 3.1. As a baseline which encodes distances in $\mathcal{Z}$ but does not attempt subspace invariance, we introduce *principal component analysis* (PCA, Section 3.2). We continue with *proto value functions* (PVF, Section 3.3), which are the current state of the art in subspace-invariant features. *Slow feature analysis* (SFA, Section 3.4) has only recently been proposed to generate basis functions for RL. Section 4 analyzes the properties of both reward-based and subspace-invariant features in detail and Section 5 empirically compares all discussed algorithms in various experiments.

### 3.1 Reward-based Basis Functions

The true value function $v^\pi \in L^2(\mathcal{Z},\xi)$ is defined as $v^\pi(z) = \sum_{t=0}^\infty \gamma^t (\hat{P}^\pi)^t[r](z), \forall z \in \mathcal{Z}$, where $(\hat{P}^\pi)^t$ refers to $t$ consecutive applications of operator $\hat{P}^\pi$ and $(\hat{P}^\pi)^0 := \hat{I}$. This illustrates the intuition of *Krylov bases* (Petrik, 2007):

$$\phi_i^K := (\hat{P}^\pi)^{i-1}[r] \in L^2(\mathcal{Z},\xi), \qquad \Phi_k^K := \{\phi_1^K, \ldots, \phi_k^K\}.$$

These bases are natural for *value iteration*, as the value functions of all iterations can be exactly represented (see also Corollary 10, Page 2087). However, the transition operator must be approximated from observations and the resulting basis $\Phi_k^K$ is not orthonormal. If employed, a projection operator must thus compute an expensive inverse (see Section 2.3). *Bellman error basis functions* (BEBF, Parr et al., 2007) rectify this by defining the $(k+1)$'th feature as the Bellman error of the fixed point solution $f^k \in \mathcal{F}_{\Phi_k^B}$ with $k$ features:

$$\phi_{k+1}^B := \hat{B}^\pi[f^k] - f^k \in L^2(\mathcal{Z},\xi), \quad f^k \overset{!}{=} \hat{\Pi}_\xi^{\Phi_k^B}\left[\hat{B}^\pi[f^k]\right] \in \mathcal{F}_{\Phi_k^B}, \quad \Phi_k^B := \{\phi_1^B, \ldots, \phi_k^B\}.$$

BEBF are orthogonal, that is, $\langle \phi_i^B, \phi_j^B \rangle_\xi = \varepsilon \delta_{ij}, \varepsilon > 0$, and scaling $\varepsilon$ to 1 yields an orthonormal basis. Parr et al. (2007) have shown that Krylov bases and BEBF span the same approximation space $\mathcal{F}_{\Phi_k^K} = \mathcal{F}_{\Phi_k^B}$. Both approaches require many features if $\gamma \to 1$.

Mahadevan and Liu (2010) have extended BEBF to *Bellman average reward bases* (BARB) by including the *average reward* $\rho$ as the first feature. This is motivated by *Drazin bases* and has been reported to reduce the number of required features for large $\gamma$. Recently, Sun et al. (2011) have pointed out that given some basis $\Phi_k$, the *best* $k+1$'th basis function is always the fixed point solution with the current Bellman error, that is, the next BEBF $\phi_{k+1}^B$, as reward. Adding the resulting *Value function of the Bellman error* (V-BEBF) to the current basis can represent the true value exactly. However, the approach has to be combined with some feature selection strategy, as finding the V-BEBF fixed point is just as hard.

All above algorithms are exclusively defined on discrete MDPs. Although an extension to general RKHSs seems possible, it is not the focus of this article. However, to give readers a comparison of available methods we will evaluate orthogonalized Krylov bases (which are equivalent to BEBF) on discrete Benchmark tasks in Sections 5.2 and 5.3.

### 3.2 Principal Component Analysis

To provide a baseline for comparison, we introduce *principal component analysis* (PCA, Pearson, 1901). As PCA does not take any transitions into account, the extracted features must therefore encode Euclidean distances in $\mathcal{Z}$. PCA aims to find subspaces of maximal variance, which are spanned by the eigenvectors to the *p largest* eigenvalues of the data covariance matrix. One interpretation of PCA features $\phi : \mathcal{Z} \to \mathbb{R}^p$ is an optimal encoding of the centered data $\{z_t\}_{t=1}^n \subset \mathcal{Z} \subset \mathbb{R}^d$ w.r.t. linear *least-squares* reconstruction, that is the optimization problem

$$\inf_{\phi \in (\mathcal{F}_{\text{lin}})^p} \underbrace{\inf_{f \in (\mathcal{F}_\phi)^d} \tilde{\mathbb{E}}_t\left[\left\|z_t - f(z_t)\right\|_2^2\right]}_{\text{least-squares reconstruction in } \mathcal{F}_\phi},$$

where $\tilde{\mathbb{E}}_t[\cdot]$ is the empirical expectation operator w.r.t. all indices $t$ and $\mathcal{F}_{\text{lin}}$ the set of linear functions in $\mathbb{R}^d$.

In the interest of a more efficient encoding one can extend the function set $\mathcal{F}_{\text{lin}}$. A popular example are *reproducing kernel Hilbert spaces*, introduced in Section 2.5. The resulting algorithm is called *kernel PCA* (Schölkopf et al., 1998) and performs an eigenvalue decomposition of the centered *kernel matrix* $K_{ij} = \kappa(z_i, z_j)$. The eigenvectors $v^i \in \mathbb{R}^n$ to the $p$ largest eigenvalues are the coefficients to the feature maps:

$$\phi_i(z) := \sum_{t=1}^{n} v_t^i \, \kappa(z, z_t), \quad \forall z \in \mathcal{Z}.$$

The classical algorithm (Schölkopf et al., 1998) is severely limited by a *computational complexity* of $\mathcal{O}(n^3)$ and a *memory complexity* of $\mathcal{O}(n^2)$. It can thus be necessary to approximate the solution by using a *sparse kernel matrix* of a subset of the data (*projected process*, Rasmussen and Williams, 2006), that is, $K_{it} = \kappa(s_i, z_t)$, with $\{s_i\}_{i=1}^m \subset \{z_t\}_{t=1}^n$, $m \ll n$. The eigenvectors $v^i \in \mathbb{R}^m$ of $\frac{1}{n}\mathbf{K}\mathbf{K}^\top$ determine the coefficients of the *sparse kernel PCA* features. If a large enough subset is distributed uniformly in $\mathcal{Z}$, the approximation is usually very good.

### 3.3 Proto-value Functions

In finite state spaces $\mathcal{Z}$, $|\mathcal{Z}| < \infty$, proto-value functions (PVF, Mahadevan and Maggioni, 2007) are motivated by *diffusion maps* on graphs (Coifman et al., 2005). For this purpose a *connection graph* is constructed out of a Markov chain $\{z_t\}_{t=1}^n \subset \mathcal{Z}$: for the first observed transition from state $x$ to $y$, the corresponding entry of connection matrix $\mathbf{W}$ is set $W_{xy} := 1$. All entries of non-observed transitions are set to zero. As diffusion maps require undirected graphs, this matrix must be symmetrized by setting $\mathbf{W} \leftarrow \frac{1}{2}(\mathbf{W} + \mathbf{W}^\top)$. PVF are the eigenvectors to the $p$ smallest eigenvalues of the *normalized graph Laplacian* $\mathbf{L} := \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\,\mathbf{D}^{-1/2}$, where $D_{xy} = \delta_{xy} \sum_{z=1}^{|\mathcal{Z}|} W_{xz}, \forall x, y \in \mathcal{Z}$, and $\delta_{xy}$ is the Kronecker delta. Section 4.1 shows that this approach, also known as *spectral encoding* (Belkin and Niyogi, 2003), yields approximation spaces in which Euclidean distances are equivalent to diffusion distances on the connection graph. Note, however, that these are not exactly the diffusion distances of the transition kernel, as the transition *possibility* rather than *probability* is encoded in matrix $\mathbf{W}$. Section 4.2 discusses this difference.

For infinite observation spaces $\mathcal{Z}$ PVF are also defined by connection graphs. However, in difference to the finite case, the construction of this graph is not straightforward. Mahadevan and Maggioni (2007) proposed a symmetrized *k-nearest neighbors graph* $\mathbf{W}$ out of a random[16] set $\{s_j\}_{j=1}^m \subset \{z_t\}_{t=1}^n$, $m \ll n$. Each node $s_i$ is only connected with the $k$ nearest nodes $\{s'_j\}_{j=1}^k \subset \{s_j\}_{j=1}^m$ (w.r.t. the Euclidean norm in $\mathcal{Z}$), with weights determined by a Gaussian kernel $\kappa(\cdot,\cdot)$ with width-parameter $\sigma$,

$$W_{ij} := \kappa(s_i, s_j) = \exp\left(-\tfrac{1}{2\sigma^2}\|s_i - s_j\|_2^2\right).$$

After symmetrization the PVF $\hat{\phi}_i$ at the nodes $s_j$ are calculated. A Nyström extension approximates the PVF for all samples $z$ by calculating the mean over the weighted PVF of the $k$ nodes $\{s'_j\}_{j=1}^k \subset \{s_j\}_{j=1}^m$ closest to $z$,

$$\phi_i(z) := \sum_{j=1}^{k} \frac{\kappa(z, s'_j)}{\sum_{l=1}^{k} \kappa(z, s'_l)} \, \hat{\phi}_i(s'_j), \quad \forall z \in \mathcal{Z}.$$

16. Ideally the nodes are uniformly drawn w.r.t. the true diffusion metric, in other words uniformly in $\mathcal{X}$. If nodes are drawn randomly or uniformly in $\mathcal{Z}$, this difference can lead to a significant deviation in the number of transitions between nodes and the resulting diffusion distances thus deviate as well.

Note that these features are no longer based on the *transitions* of the observed Markov chain, but on Euclidean distances in $\mathcal{Z}$.

### 3.4 Slow Feature Analysis

The unsupervised learning method *slow feature analysis* (SFA, Wiskott and Sejnowski, 2002) aims for a set of mappings $\boldsymbol{\phi} : \mathcal{Z} \to \mathbb{R}^p$ such that the values $\phi_i(\boldsymbol{z}_t)$ change slowly over an observed Markov chain $\{\boldsymbol{z}_t\}_{t=1}^n \subset \mathcal{Z}$. The objective (called *slowness* $\mathcal{S}$) is defined as the *expectation of the squared discrete temporal derivative*:

$$\inf_{\boldsymbol{\phi} \in (\mathcal{F})^p} \quad \sum_{i=1}^p \mathcal{S}(\phi_i) \quad := \quad \sum_{i=1}^p \tilde{\mathbb{E}}_t[\dot{\phi}_i^2(\boldsymbol{z}_t)] \quad \text{(slowness)}\,.$$

To ensure each slow feature encodes unique information and can be calculated in an iterative fashion, the following constraints must hold $\forall i \in \{1, \ldots, p\}$ :

$$\begin{aligned} \tilde{\mathbb{E}}_t[\phi_i(\boldsymbol{z}_t)] &= 0 && \text{(zero mean)}, \\ \tilde{\mathbb{E}}_t[\phi_i^2(\boldsymbol{z}_t)] &= 1 && \text{(unit variance)}, \\ \forall j \neq i : \tilde{\mathbb{E}}_t[\phi_i(\boldsymbol{z}_t)\phi_j(\boldsymbol{z}_t)] &= 0 && \text{(decorrelation)}, \\ \forall j > i : \mathcal{S}(\phi_i) &\leq \mathcal{S}(\phi_j) && \text{(order)}\,. \end{aligned}$$

The principle of slowness has been used for a long time in the context of neural networks (Földiák, 1991). Kompella et al. (2012) have proposed an incremental online SFA algorithm. Recently several groups have attempted to use SFA on a random walk of observations to generate basis functions for RL (Legenstein et al., 2010; Luciw and Schmidhuber, 2012).

Although formulated as a linear algorithm, SFA was originally intended to be applied on the space of polynomials like quadratic (Wiskott and Sejnowski, 2002) or cubic (Berkes and Wiskott, 2005). The polynomial expansion of potentially high dimensional data, however, spans an impractically large space of coefficients. Hierarchical application of quadratic SFA has been proposed to solve this problem (Wiskott and Sejnowski, 2002; Legenstein et al., 2010). Although proven to work in complex tasks (Franzius et al., 2007), this approach involves a multitude of hyper-parameters and no easy way to counteract inevitable over-fitting is known so far.

An alternative to polynomial expansions are *sparse kernel methods* (see Section 2.5). We summarize in the following the *regularized sparse kernel SFA* (RSK-SFA, Böhmer et al., 2012) which we have used in our experiments. For a given sparse subset $\{\boldsymbol{s}_i\}_{i=1}^m \subset \{\boldsymbol{z}_t\}_{t=1}^n$, the algorithm determines the mapping $\boldsymbol{\phi} : \mathcal{Z} \to \mathbb{R}^p$ in 3 steps:

i Fulfilling the zero mean constraint directly on sparse kernel matrix $K_{it} := \boldsymbol{\kappa}(\boldsymbol{s}_i, \boldsymbol{z}_t)$, that is, $\mathbf{K}' := (\mathbf{I} - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top)\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)$, where $\mathbf{1}_k \in \mathbb{R}^k$ is a column vector of ones.

ii Fulfilling unit variance and decorrelation constraints by performing an eigenvalue decomposition $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top := \frac{1}{n}\mathbf{K}'\mathbf{K}'^\top$ and projecting $\mathbf{K}'' := \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top\mathbf{K}'$.

iii Minimize the objective by another eigenvalue decomposition $\mathbf{R}\boldsymbol{\Sigma}\mathbf{R}^\top := \frac{1}{n-1}\dot{\mathbf{K}}''\dot{\mathbf{K}}''^\top$, where $\dot{K}''_{it} := K''_{it+1} - K''_{it}$. Grouping the kernel functions of the sparse subset into one multivariate function $\boldsymbol{k} : \mathcal{Z} \to \mathbb{R}^m$ with $k_i(\boldsymbol{z}) := \boldsymbol{\kappa}(\boldsymbol{z}, \boldsymbol{s}_i)$, $\forall \boldsymbol{z} \in \mathcal{Z}$, the solution is given by

$$\boldsymbol{\phi}(\boldsymbol{z}) = \mathbf{A}^\top \boldsymbol{k}(\boldsymbol{z}) - \boldsymbol{c}, \quad \forall \boldsymbol{z} \in \mathcal{Z}$$
$$\text{with } \mathbf{A} := (\mathbf{I} - \mathbf{1}_m\mathbf{1}_m^\top)\mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{R}, \text{ and } \boldsymbol{c} := \tfrac{1}{n}\mathbf{A}\mathbf{K}\mathbf{1}_n\,.$$

Böhmer et al. (2012) have demonstrated the numerical instability of this algorithm in face of insufficient sparseness and introduced a *regularization term* $\|\phi_i\|_{\mathcal{H}_\kappa}$ to the objective to stabilize the solution. In our experiments we did not face this problem, and regularization is thus omitted here.

### 3.5 Relative Velocities and Modified SFA

In the limit of an infinite Markov chain in $\mathcal{Z}$ and some mild assumptions[17] on the transition kernel in $\mathcal{X}$, the slowest possible mappings can be calculated analytically (Wiskott, 2003; Franzius et al., 2007). As the discrete temporal derivative is specified by the transition kernel in $\mathcal{X}$, the analytical solutions have domain $\mathcal{X}$ as well. Note, however, that the same transition kernel yields the same feature maps in $\mathcal{X}$, independent[18] of the actual observation space $\mathcal{Z}$. Section 4.1 demonstrates that these solutions are endowed with the diffusion metric of the symmetrized transition kernel, not unlike PVF in finite state spaces. Sprekeler (2011) has recently shown that in this case SFA solutions are (almost) equivalent to PVF. Note also that SFA encodes the actual transition *probabilities*, which requires more samples to converge than the transition *possibilities* encoded by PVF.

The analytically derived SFA features of Franzius et al. (2007) are of particular interest to the visual navigation experiment (Section 5.4 and Figure 2, Page 2071), as they assume the same setup. The solution is a *Fourier basis* on domain $\mathcal{X} := [0, L_x] \times [0, L_y] \times [0, 2\pi)$,

$$\phi_{\iota(i,j,l)}(x,y,\theta) = \begin{cases} \sqrt[3]{2}\cos(\frac{i\pi}{L_x}x)\cos(\frac{j\pi}{L_y}y)\sin(\frac{l+1}{2}\theta), & l \text{ odd} \\ \sqrt[3]{2}\cos(\frac{i\pi}{L_x}x)\cos(\frac{j\pi}{L_y}y)\cos(\frac{l}{2}\theta), & l \text{ even} \end{cases}, \quad \forall (x,y,\theta) \in \mathcal{X},$$

where $\iota : (\mathbb{N} \times \mathbb{N} \times \mathbb{N} \setminus \{(0,0,0)\}) \to \mathbb{N}^+$ is an index function, which depends on the *relative velocities* in two spatial dimensions $x$ and $y$, and the robot's orientation $\theta$. It can occur that SFA features have the same slowness, in which case the solution is no longer unique. For example, if $\phi_{\iota(1,0,0)}$ and $\phi_{\iota(0,1,0)}$ have the same slowness, then $\mathcal{S}(\phi_{\iota(1,0,0)}) = \mathcal{S}(\phi_{\iota(0,1,0)}) = \mathcal{S}(a\phi_{\iota(1,0,0)} + b\phi_{\iota(0,1,0)})$ holds as long as $a^2 + b^2 = 1$. This corresponds to an arbitrary rotation in the subspace of equally slow features. However, if we are interested in the space spanned by all features *up to a certain slowness*, every rotated solution spans the same approximation space $\mathcal{F}_\phi$.

The order $\iota(\cdot,\cdot,\cdot)$ of the analytical SFA features derived by Franzius et al. (2007, see above) depend strongly on the *relative velocities* in the state dimensions. For example, crossing the room in our experiment in Section 5.4 requires 10 movements, during which feature $\phi_{\iota(1,0,0)}$ will run through half a cosine wave. In as little as 4 rotations, on the other hand, feature $\phi_{\iota(0,0,1)}$ registers the same amount of change. Sampled evenly by a random policy, the first SFA features will therefore *not* encode the robot's orientation, which can critically impair the value representation in low dimensional approximation spaces. This article proposes a simple modification to the RSK-SFA algorithm to adjust the relative velocities by means of *importance sampling*.

Let $\{(z_t, a_t)\}_{t=0}^n$ denote a training sequence sampled by policy $\pi$ with a steady state distribution $\xi$, which induces the joint distribution $\mu(B,A) = \int_B \pi(A|z)\,\xi(dz), \forall B \in \mathcal{B}(\mathcal{Z}), \forall A \in \mathcal{B}(\mathcal{A})$. To switch to another policy $\tau$ and state distribution $\zeta$, that is, the joint distribution $\eta(B,A) = \int_B \tau(A|z)\,\zeta(dz), \forall B \in \mathcal{B}(\mathcal{Z}), \forall A \in \mathcal{B}(\mathcal{A})$, one can weight each transition with the *Radon-Nikodym*

17. In this case decorrelated Brownian motion in a multivariate state space $\mathcal{X}$ with independent boundary conditions for each dimension. Examples are rectangles, cubes, tori or spheres of real coordinates.

18. In line with SFA literature, this article does not discuss *partial observability* of the state. In other words, we assume there exist an *unknown* one-to-one mapping of states $x \in \mathcal{X}$ to observations $z \in \mathcal{Z}$.

*derivative* $\frac{d\eta}{d\mu}$. This yields the modified SFA optimization problem

$$\begin{aligned} \inf_{\phi\in(\mathcal{F})^p} \quad & \sum_{i=1}^{p} \hat{\mathbb{S}}(\phi_i,\eta) \quad := \quad \sum_{i=1}^{p} \tilde{\mathbb{E}}_t\left[\tfrac{d\eta}{d\mu}(z_t,a_t)\,\dot{\phi}_i^2(z_t)\right] \\ \text{s.t.} \quad & \begin{aligned} \tilde{\mathbb{E}}_t\left[\tfrac{d\eta}{d\mu}(z_t,a_t)\,\phi_i(z_t)\right] &= 0 \\ \tilde{\mathbb{E}}_t\left[\tfrac{d\eta}{d\mu}(z_t,a_t)\,\phi_i(z_t)\,\phi_j(z_t)\right] &= \delta_{ij} \end{aligned} \quad , \quad \forall i,j\in\{1,\ldots,p\}\,. \end{aligned}$$

However, there is no indication which distribution $\zeta$ and policy $\tau$ ensure a balanced encoding.

We propose here a simple heuristic for situations in which the actions affect only mutually independent subspaces of $\mathcal{X}$. In our robot navigation experiment, for example, rotations do not influence the robot's spatial position nor do the movements influence it's orientation. As optimal SFA features in the spatial subspace are significantly slower (see above), the first features will encode this subspace exclusively. This can be counteracted by setting $\zeta := \xi$ and defining $\frac{d\tau}{d\pi}(z,a) := \vartheta(a), \forall z\in\mathcal{Z}, \forall a\in\mathcal{A}$, where $\vartheta : \mathcal{A}\to\mathbb{R}^+$ weights each action independent of the current state. In practice, weights $\vartheta(a)$ need to be adjusted by hand for each action $a\in\mathcal{A}$: the *higher* $\vartheta(a)$, the *weaker* the influence of the corresponding subspace of $\mathcal{X}$ onto the first features. Only the last step (iii) of RSK-SFA has to be modified by redefining $\dot{K}''_{it} \leftarrow \vartheta^{\frac{1}{2}}(a_t)\,\dot{K}''_{it}$. Figure 9, Page 2101, demonstrates the effect of this modification.

## 4. Theoretical Analysis

This section analyzes the theoretical properties of *reward-based* and *subspace-invariant* features w.r.t. value function approximation. The employed formalism is introduced in Section 2. If not stated otherwise, the features are assumed to be optimized over $L^2(\mathcal{Z},\xi)$ and based on an infinite ergodic Markov chain. Proofs to all given lemmas and theorems can be found in Appendix A.

At the heart of function approximation lies the concept of *similarity*. Similar states will have similar function values. Usually this similarity is given by a metric on the observation space. Deviations of the function output from this metric must be compensated by the optimization algorithm. However, value function approximation allows for *explicit* specification of the required similarity. The definition of the value assigns similar function output to states with (i) similar immediate rewards and (ii) similar futures. As discussed in Section 3, *reward-based features* focus on encoding (i), whereas *subspace-invariant features* focus on (ii). Section 4.1 analyzes how SFA encodes similar futures as *diffusion distances* and shows some restrictions imposed onto the class of subspace-invariant features. The ramifications of these restrictions onto value function approximation are discussed in Section 4.2.

This article also presents a second, novel perspective onto value function approximation. The MDP one will face is usually not known before learning and the construction of a suitable basis is very expensive. Instead of approximating a particular MDP at hand, one could focus on a complete set $\mathcal{M}$ of *anticipated* MDPs. An *optimal* approximation space should be able to approximate any MDP $m\in\mathcal{M}$ if encountered. In difference to the classical analysis put forward by Petrik (2007), Parr et al. (2007) and Mahadevan and Maggioni (2007), this point of view puts emphasis on the reuse of prior experience, as investigated in *transfer learning* (Taylor and Stone, 2009; Ferguson and Mahadevan, 2006; Ferrante et al., 2008). Section 4.3 defines a criterion of *optimal features* for some anticipated set $\mathcal{M}$. Under some assumptions on $\mathcal{M}$, we prove that SFA optimizes a bound on

this criterion and argue that there can be no better bound based on a single Markov chain. Section 4.4 provides a summarizing conclusion and further implications can be found in Section 6.

### 4.1 Diffusion Metric

Values of observed states $x, y \in \mathcal{Z}$ depend less on their Euclidean distance in $\mathcal{Z}$ than on common *future states*. PVF are thus based on *diffusion distances* $d_t(x,y)$ of a graph representing the symmetrized *transition possibilities* $T_{xy} := W_{xy}/(\sum_{z\in\mathcal{Z}} W_{xz})$ between discrete states (see Section 3.3 or Mahadevan and Maggioni, 2007):

$$d_t^2(x,y) = \sum_{z\in\mathcal{Z}} \xi_z \left( (\mathbf{T}^t)_{xz} - (\mathbf{T}^t)_{yz} \right)^2 ,$$

where $\xi \in \mathbb{R}^{|\mathcal{Z}|}$ are arbitrary[19] non-negative weights and $\mathbf{T}^t$ denotes the $t$'th power of matrix $\mathbf{T}$. These diffusion distances are equal to Euclidean distances in a space spanned by the eigenvectors $\phi_i \in \mathbb{R}^{|\mathcal{Z}|}$ and eigenvalues $\lambda_i \in \mathbb{R}$ of connectivity matrix $\mathbf{T}$ (e.g., for general similarity matrices see Coifman et al., 2005):

$$d_t(x,y) = \left\| \psi^t(x) - \psi^t(y) \right\|_2 , \qquad \psi_i^t(x) := \lambda_i^t \phi_{ix} , \qquad \forall t \in \mathbb{N} .$$

An extension to potentially continuous observation (or state) spaces $\mathcal{Z}$ with ergodic transition kernels $P^\pi : \mathcal{Z} \times \mathcal{B}(\mathcal{Z}) \to [0,1]$ is not trivial. Mean-squared differences between distributions are not directly possible, but one can calculate the difference between *Radon-Nikodym derivatives*.[20] Due to ergodicity the derivative always exists for finite sets $\mathcal{Z}$, but for continuous $\mathcal{Z}$ one must exclude transition kernels that are not absolutely continuous.[21]

**Assumption 1** *If $\mathcal{Z}$ is continuous, the transition kernel knows no finite set of future states.*

$$P(B|z,a) \quad = \quad 0 , \qquad \forall B \in \left\{ B \in \mathcal{B}(\mathcal{Z}) \,\middle|\, |B| < \infty \right\} , \quad \forall z \in \mathcal{Z} , \quad \forall a \in \mathcal{A} .$$

This can always be fulfilled by adding a small amount of *continuous noise* (e.g., Gaussian) to each transition. Let in the following $(P^\pi)^t(\cdot,x) : \mathcal{B}(\mathcal{Z}) \to [0,1]$ denote the state distribution after $t$ transitions, starting at state $x \in \mathcal{Z}$. Note that under Assumption 1 the Radon-Nikodym derivative w.r.t. steady state distribution $\xi$ is[22] $\frac{d(P^\pi)^t(\cdot|x)}{d\xi} \in L^2(\mathcal{Z},\xi), \forall t \in \mathbb{N} \setminus \{0\}$.

**Definition 1** *The diffusion distance $d_t : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$ based on ergodic transition kernel $P^\pi$ with steady state distribution $\xi$ is defined as*

$$d_t(x,y) \; := \; \left\| \mu_x^t - \mu_y^t \right\|_\xi , \qquad \mu_x^t \; := \; \tfrac{d(P^\pi)^t(\cdot|x)}{d\xi} \in L^2(\mathcal{Z},\xi) , \qquad \forall x,y \in \mathcal{Z} , \quad \forall t \in \mathbb{N} \setminus \{0\} .$$

19. In our context these weights are the equivalent to the steady state distribution and thus named the same.
20. If Radon-Nikodym derivative $\frac{d\zeta}{d\xi}$ exists then $\int \xi(dz) \frac{d\zeta}{d\xi}(z) f(z) \; = \; \int \zeta(dz) f(z) , \; \forall f \in L^2(\mathcal{Z},\xi)$.
21. The Radon-Nikodym derivative $\frac{d\zeta}{d\xi}$ exists if distribution $\zeta$ is *absolutely continuous* w.r.t. steady state distribution $\xi$, that is if $\xi(B) = 0 \; \Rightarrow \; \zeta(B) = 0, \; \forall B \in \mathcal{B}(\mathcal{Z})$. If there exists a finite Borel set $B \in \mathcal{B}(\mathcal{Z})$ with $\zeta(B) > 0$, however, the derivative must not exist as $\xi(B) = 0$ can hold for ergodic Markov chains.
22. Assumption 1 guarantees the Radon-Nikodym derivative exists in the space of integrable functions $L^1(\mathcal{Z},\xi)$, but by compactness of $\mathcal{Z}$ the derivative is also in $L^2(\mathcal{Z},\xi) \subset L^1(\mathcal{Z},\xi)$ (Reed and Simon, 1980).

The projection methods discussed in Section 2 are based on *Euclidean distances* in the approximation space $\mathcal{F}_\phi$. These spaces are invariant to scaling of the basis functions. Given a particular scaling, however, diffusion distances can equal Euclidean distances in $\mathcal{F}_\phi$. In this case we say that the basis functions *encode* this distance.

**Definition 2** *Basis functions $\phi_i \in L^2(\mathcal{Z},\xi), i \in \{1,\dots,p\}$, are said to "encode" a distance function $d : \mathcal{Z}\times\mathcal{Z} \to \mathbb{R}^+$ if there exists a scaling vector $\varrho \in (\mathbb{R}^+)^p$ such that*

$$d(x,y) \quad = \quad \sqrt{\sum_{i=1}^{p} \varrho_i \Big(\phi_i(x) - \phi_i(y)\Big)^2} \quad = \quad \Big\|\phi(x) - \phi(y)\Big\|_\varrho , \qquad \forall x,y \in \mathcal{Z}\,.$$

If the analogy between value function generalization and diffusion distances is correct, one should aim for a set of basis functions that at least approximates an encoding of diffusion distances $d_t(\cdot,\cdot)$, if possible for all forecast parameters $t \in \mathbb{N}\setminus\{0\}$ at once.

**Lemma 3** *Let $\xi$ denote the steady state distribution of ergodic transition kernel $P^\pi$, which has a self-adjoint transition operator $\hat{P}^\pi = (\hat{P}^\pi)^* : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$. The corresponding diffusion distance equals the Euclidean distance in the space spanned by $\psi_i^t(\cdot) := \lambda_i^t \phi_i(\cdot), \forall i \in \mathbb{N}$, where $\lambda_i \in \mathbb{R}$ and $\phi_i \in L^2(\mathcal{Z},\xi)$ are the eigenvalues and eigenfunctions of $\hat{P}^\pi$, that is*

$$d_t(x,y) = \|\psi^t(x) - \psi^t(y)\|_2\,, \quad \forall x,y \in \mathcal{Z}, \forall t \in \mathbb{N}\setminus\{0\}\,.$$

**Proof** *see Appendix A, Page 2108.* ■

Note that the full set of eigenfunctions $\phi_i$ encodes *all* diffusion distances $d_t(\cdot,\cdot), \forall t \in \mathbb{N}\setminus\{0\}$. Lemma 3 shows that the above relationship between diffusion and Euclidean distances in the eigenspace of the transition operator $\hat{P}^\pi : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$ also holds, but only if this operator is *self-adjoint*.[23] This does not hold for most transition operators, however. Their eigenfunctions do not have to be orthogonal or even be real-valued functions, analogous to complex eigenvectors of asymmetric matrices. Using these eigenfunctions, on the other hand, is the declared intent of *subspace-invariant features* (see Section 3). Constructing real-valued basis functions with zero *per-feature error* thus does not seem generally possible.

In this light one can interpret the *symmetrized transition possibilities* encoded by PVF as a self-adjoint approximation of the *transition probabilities* of $P^\pi$. This raises the question whether better approximations exist.

**Lemma 4** *Let $P^\pi$ be an ergodic transition kernel in $\mathcal{Z}$ with steady state distribution $\xi$. The kernel induced by adjoint transition operator $(\hat{P}^\pi)^*$ in $L^2(\mathcal{Z},\xi)$ is $\xi$-almost-everywhere an ergodic transition kernel with steady state distribution $\xi$.*
**Proof** *see Appendix A, Page 2109.* ■

To obtain a self-adjoint transition kernel, Lemma 4 shows that the kernel of the adjoint operator $(\hat{P}^\pi)^*$ is a transition kernel as well. Intuitively, when $\hat{P}^\pi$ causes all water to flow downhill, $(\hat{P}^\pi)^*$ would cause it to flow the same way uphill. Note the difference to an *inverse* transition kernel, which could find new ways for the water to flow uphill. Although this changes the transition dynamics,

23. Each linear operator $\hat{A} : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$ has a unique *adjoint operator* $\hat{A}^*$ for which holds: $\langle f, \hat{A}[g]\rangle_\xi = \langle \hat{A}^*[f], g\rangle_\xi, \forall f,g \in L^2(\mathcal{Z},\xi)$. The operator is called *self-adjoint*, if $\hat{A} = \hat{A}^*$.

one can construct a *symmetrized transition operator* $\hat{P}_s^{\pi} := \frac{1}{2}\hat{P}^{\pi} + \frac{1}{2}(\hat{P}^{\pi})^*$ as a self-adjoint approximation of $\hat{P}^{\pi}$. Estimating $\hat{P}_s^{\pi}$ may take more samples than the connection graph $\mathbf{T}$ constructed by PVF, but it stands to reason that $\hat{P}_s^{\pi}$ is a better approximation to $\hat{P}^{\pi}$. This intuition is put to the test in Section 5.1. We find indeed that SFA features have on average smaller *per-feature errors* than PVF. For purely random transition kernels the advantage of SFA is minuscule, but the when $\hat{P}^{\pi}$ resembles a self-adjoint operator the difference is striking (see Figure 3 on Page 2092). The goal of encoding diffusion distances based on $P^{\pi}$ appears thus best served by the eigenfunctions of the symmetrized transition operator $\hat{P}_s^{\pi}$. Lemma 5 shows[24] that in the limit of an infinite training sequence, SFA extracts these eigenfunctions in the order of their largest eigenvalues:

**Lemma 5** *In the limit of an infinite ergodic Markov chain drawn by transition kernel $P^{\pi}$ in $\mathcal{Z}$ with steady state distribution $\xi$ holds $\mathcal{S}(f) = 2\langle f, (\hat{I} - \hat{P}^{\pi})[f]\rangle_{\xi}, \forall f \in L^2(\mathcal{Z}, \xi)$.*
**Proof** *see Appendix A, Page 2109.* ■

Note that the first, constant eigenfunction of $\hat{P}_s^{\pi}$ is not extracted, but has no influence on the encoded distance. Encoding any diffusion distance $d_t(\cdot,\cdot)$ would therefore need a potentially infinite number of SFA features. As the influence of each feature shrinks exponentially with the forecast parameter $t$, however, the encoding can be approximated well by the first $p$ SFA features. Except for $t = 0$, this approximation is optimal in the least-squares sense. Note also that for fixed $p$ the approximation quality *increases* with $t$. Predictions based on SFA features will therefore be more accurate in the long term than in the short term.

**Theorem 6** *SFA features $\{\phi_i\}_{i=1}^{\infty}$ simultaneously encode all diffusion distances $d_t(\cdot,\cdot), \forall t \in \mathbb{N} \setminus \{0\}$, based on the symmetrized transition kernel $P_s^{\pi} = \frac{1}{2}P^{\pi} + \frac{1}{2}(P^{\pi})^*$. The first $p$ SFA features are an optimal $p$-dimensional least-squares approximation to this encoding.*
**Proof** *The theorem follows directly from Definition 2 and Lemmas 3, 4, 5 and 15.* ■

A similar proposition can be made for PVF features and diffusion distances based on *transition possibilities*. The connection to reward-based features (Section 3.1) is less obvious. Concentrating naturally on immediate and short term reward, these basis functions depend on the *reward function* at hand. It is, however, possible to show the encoding of diffusion distances *on average*, given the reward function is drawn from a *white noise functional*[25] $\rho$.

**Theorem 7** *On average over all reward functions $r^{\pi} : \mathcal{Z} \to \mathbb{R}$ drawn from a white noise functional $\rho$, the squared norm of a Krylov basis $\{\phi_i^K\}_{i=1}^{p}$ from an ergodic transition kernel $P^{\pi}$ encodes squared diffusion distances based on $\hat{P}^{\pi}$ up to horizon $p-1$, that is*

$$d_t^2(x,y) \;=\; \mathbb{E}\left[\left\|\phi^K(x) - \phi^K(y)\right\|_{\varrho}^2 \,\middle|\, r^{\pi} \sim \rho\right], \quad \forall x,y \in \mathcal{Z},\ \exists \varrho \in (\mathbb{R}^+)^p,\ \forall t \in \{1,\ldots,p-1\}.$$

**Proof** *see Appendix A, Page 2110.* ■

Although Krylov bases are different for each reward function $r^{\pi} \in L^2(\mathcal{Z}, \xi)$ and the employed squared distances diverge slightly from Definition 1, Theorem 7 implies that on average they encode

24. Lemma 5 shows that the SFA optimization problem is equivalent to $\inf_{\phi} \langle \phi, (\hat{I} - \hat{P}^{\pi})[\phi]\rangle_{\xi} \equiv \sup_{\phi} \langle \phi, \hat{P}^{\pi}[\phi]\rangle_{\xi} = \sup_{\phi} \langle \phi, \hat{P}_s^{\pi}[\phi]\rangle_{\xi}$, due to the symmetry of the inner product.
25. A white noise functional is the Hilbert space equivalent to a Gaussian normal distribution (Holden et al., 2010). In our context it suffices to say that $\mathbb{E}[\langle f, r^{\pi}\rangle_{\xi}^2 \,|\, r^{\pi} \sim \rho] = \langle f, f\rangle_{\xi}, \forall f \in L^2(\mathcal{Z}, \xi)$.

diffusion distances up to time horizon $p-1$. The same results hold for BEBF (Petrik, 2007) and with minor modifications for BARB bases (Mahadevan and Liu, 2010).

In conclusion, reward-based features encode diffusion distances exactly up to some time horizon, whereas SFA and PVF approximate an encoding for *all* possible distances. So far the only connection to value function approximation is the intuition of a generalizing metric. However, in the next subsection we show striking parallels between diffusion distances and approximation errors.

### 4.2 Value Function Approximation

The analysis in Section 4.1 revealed a critical problem for the construction of *subspace-invariant features* (Parr et al., 2008): eigenfunctions of the transition operator $\hat{P}^{\pi}$ are not necessarily orthogonal and real-valued. Constructing a real-valued, orthonormal basis of subspace-invariant features is thus only possible in some rare cases of self-adjoint transition operators. Both SFA and PVF substitute therefore a "similar" self-adjoint transition operator for $\hat{P}^{\pi}$. SFA employs the *symmetrized* operator $\hat{P}^{\pi}_s := \frac{1}{2}\hat{P}^{\pi} + \frac{1}{2}(\hat{P}^{\pi})^*$ and PVF assigns equal probability to *all possible* neighbors[26] $\hat{T}^{\pi}$. Analytical comparison of the quality of these approximations is difficult, however.

On the other hand, the class of MDPs for which SFA features are subspace-invariant *contains* the class for which PVF are. To see this, imagine a transition kernel $T^{\pi}$ for which PVF are subspace-invariant, which implies that for each state there exists a uniform distribution to end up in the set of its neighbors, with symmetric neighborhood relationships. As $\hat{T}^{\pi}$ is thus self-adjoint, any ergodic Markov chain from this kernel will yield subspace-invariant SFA features. Reversely, one can construct a transition kernel $P^{\pi}$ with a self-adjoint transition operator but without uniform transition probabilities. PVF would no longer correspond to eigenfunctions of $\hat{P}^{\pi}$ and would thus not be subspace-invariant. SFA can in this sense be seen as a generalization of PVF.

Within the class of MDPs with self-adjoint transition operators, however, one can make some strong claims regarding *value function approximation* with LSTD (Section 2.3).

**Lemma 8** *Let $\{\phi_i\}_{i=1}^{p}$ denote any p SFA features from a MDP with self-adjoint transition operator, then the LSTD fixed point $f^{\pi} = \hat{\Pi}^{\phi}_{\xi}\left[\hat{B}^{\pi}[f^{\pi}]\right]$ and the projection of true value function $v^{\pi} = \hat{B}^{\pi}[v^{\pi}]$ coincide, that is*

$$f^{\pi}(x) \;=\; \hat{\Pi}^{\phi}_{\xi}[v^{\pi}](x) \;=\; \sum_{i=1}^{p} \langle r^{\pi}, \phi_i \rangle_{\xi}\, \tau_i\, \phi_i(x)\,, \quad \forall x \in \mathcal{Z}, \quad \tau_i := \left(1 - \gamma + \tfrac{\gamma}{2}\mathcal{S}(\phi_i)\right)^{-1}.$$

**Proof** *see Appendix A, Page 2110.* ■

Lemma 8 implies that for SFA features of symmetric transition models, the bound of Tsitsiklis and Van Roy (1997, introduced in Section 2.3) can be dramatically improved:

**Corollary 9** *The approximation error of the LSTD solution $f^{\pi}$ to the true value $v^{\pi}$ for MDPs with self-adjoint transition operators using any corresponding SFA features $\{\phi_i\}_{i=1}^{p}$ is*

$$\left\| v^{\pi} - f^{\pi} \right\|_{\xi} \quad = \quad \left\| v^{\pi} - \hat{\Pi}^{\phi}_{\xi}[v^{\pi}] \right\|_{\xi}.$$

26. In the discrete case these are all observed transitions, in the continuous case neighborhood relationships are based on similarities in observation space $\mathcal{Z}$ (Mahadevan and Maggioni, 2007).

An analogous proposition can be made for PVF, but for a smaller subset of MDPs. Equivalent fixed point solutions for $p$ reward-based features, on the other hand, do not appear generally possible as the behavior beyond $p-1$ time steps is not encoded (see Theorem 7). However, it is easy to see that *finite horizon* solutions can be computed *exactly* by projected value iteration (finite application of the projected Bellman operator, see, e.g., Bertsekas, 2007).

**Corollary 10** *Finite horizon value functions* $v_h^{\pi} := (\hat{B}^{\pi})^h[r] = (\hat{\Pi}_{\xi}^{\phi^K}[\hat{B}^{\pi}])^h[r]$ *can be computed exactly up to horizon* $h = p-1$ *by projected value iteration with Krylov base* $\{\phi_i^K\}_{i=1}^p$.

The conclusions from Theorems 6 & 7 and Corollaries 9 & 10 are very similar: SFA/PVF optimally approximate/generalize *infinite-horizon* value functions for a subset of possible MDPs, whereas reward-based features represent/generalize *finite-horizon* value functions exactly without any such restrictions.

There have also been attempts to join both types of basis functions by selecting the subspace-invariant feature most similar to the current Bellman error (Petrik, 2007; Parr et al., 2008). To motivate this, Parr et al. (2007) gave a lower bound for the approximation-bound improvement if a BEBF feature $\phi_p^B$ is added to the set $\Phi_{p-1}^B := \{\phi_i^B\}_{i=1}^{p-1}$:

$$\left\|v^{\pi} - \hat{\Pi}_{\xi}^{\Phi_{p-1}^B}[v^{\pi}]\right\|_{\xi} - \left\|v^{\pi} - \hat{\Pi}_{\xi}^{\Phi_p^B}[v^{\pi}]\right\|_{\xi} \quad \geq \quad \left\|v^{\pi} - f^{(p-1)}\right\|_{\xi} - \left\|v^{\pi} - \hat{B}\left[f^{(p-1)}\right]\right\|_{\xi},$$

where $f^{(p-1)} \in \mathcal{F}_{\Phi_{p-1}^B}$ is the LSTD fixed point solution based on $\Phi_{p-1}^B$. One can observe that for each added feature the bound shrinks by the Bellman error function $\phi_p^B$. The PVF feature with the highest correlation to $\phi_p^B$ is thus a good subspace-invariant choice.

We can provide an even stronger assertion about the approximation error of SFA features here. Theorem 11 does not rely on the knowledge of a current LSTD solution, but on the similarity of reward function $r^{\pi}$ and SFA feature $\phi_p$. Given SFA features and reward, the basis can thus be selected *before* training begins.

**Theorem 11** *Let* $\xi$ *be the steady state distribution on* $\mathcal{Z}$ *of a MDP with policy* $\pi$ *and a self-adjoint transition operator in* $L^2(\mathcal{Z},\xi)$. *Let further* $\Phi_p = \{\phi_i\}_{i=1}^p$ *be any set of p SFA features and* $v^{\pi} \in L^2(\mathcal{Z},\xi)$ *the true value of the above MDP. The improvement of the LSTD solution* $f^{(p)} := \hat{\Pi}_{\xi}^{\Phi_p}\left[\hat{B}^{\pi}[f^{(p)}]\right]$ *by including the p'th feature is bounded from below by*

$$\left\|v^{\pi} - f^{(p-1)}\right\|_{\xi} - \left\|v^{\pi} - f^{(p)}\right\|_{\xi} \quad \geq \quad \frac{1-\gamma}{2}\,\frac{\langle r^{\pi},\phi_p\rangle_{\xi}^2}{\|r^{\pi}\|_{\xi}}\,\tau_p^2, \qquad \tau_p := \left(1-\gamma+\tfrac{\gamma}{2}\mathcal{S}(\phi_p)\right)^{-1}.$$

**Proof** *see Appendix A, Page 2111.* ■

The bound improves with the similarity to reward function $r^{\pi} \in L^2(\mathcal{Z},\xi)$. The factor $\tau_p$, defined in Lemma 8, is inversely related to the slowness of the feature $\phi_p$. In $L^2(\mathcal{Z},\xi)$ we can guarantee[27] for

27. Lemma 5, Page 2085, shows that the slowness of eigenfunction $\phi_p$ of self-adjoint $P^{\pi}$ is related to the corresponding eigenvalue $\lambda_p$ by $\mathcal{S}(\phi_p) = 2(1-\lambda_p)$. Eigenvalues can be negative, but since $\lim_{p\to\infty}|\lambda_p| = 0$, every *finite* set of SFA features for *infinite* state/observation spaces will correspond to nonnegative eigenvalues only. In finite state spaces or in general RKHS with finite support, for example, in all sparse kernel algorithms, one can only guarantee $0 < \mathcal{S}(\phi_p) \leq 4$ and $\lim\limits_{p\to\infty}\lim\limits_{\gamma\to 1}\tau_p = \frac{1}{2}$.

infinite state/observation spaces that $\mathcal{S}(\phi_1) > 0$ and $\lim_{p\to\infty} \mathcal{S}(\phi_p) = 2$ that

$$\lim_{\gamma\to 0} \tau_p \quad = \quad 1\,, \qquad \lim_{\gamma\to 1} \tau_p \quad = \quad \frac{2}{\mathcal{S}(\phi_p)} \quad \geq \quad 1\,, \qquad \lim_{p\to\infty} \lim_{\gamma\to 1} \tau_p \quad = \quad 1\,.$$

One could use this bound to select the best feature set for a given MDP, similar to *matching pursuit* approaches (Mallat and Zhang, 1993). However, this is beyond the scope of this article and left for future works.

### 4.3 Encoding Anticipated Value Functions

The last subsection analyzed the properties of SFA for value function approximation of an MDP at hand. Constructed features still need to be represented somehow, for example with a RKHS or some larger set of given basis functions. Reward-based features like BEBF can reduce the value estimation time by incrementally increasing the feature set by the current Bellman error. Strictly, there is no reason to *remember* those features, though. One could instead simply remember the current value estimate. And since the features depend on the reward function, reusing them to solve another MDP is out of the question.

Subspace-invariant features, on the other hand, do not depend on the reward function but are very expensive to construct. This raises the question of *when* these features are actually computed. For example, constructing $p$ RSK-SFA features based on a Markov chain of $n$ observations with a sparse subset of $m$ support observations yields a computational complexity of $\mathcal{O}(m^2 n)$. Sparse kernel LSTD (Xu, 2006) alone exhibits the same complexity but makes use of the full span of the sparse subset, instead of only a $p$-dimensional subspace thereof. Computing features and the value function at the same time therefore does not yield any computational advantage.

Alternatively, one could rely on previously experienced "similar" MDPs to construct the basis functions (*transfer learning*, Taylor and Stone, 2009). Ferguson and Mahadevan (2006) and Ferrante et al. (2008) followed this reasoning and constructed PVF out of experiences in MDPs with the same transition, but different reward model. This section aims to analyze this transfer effect without going into the details of how to choose the MDPs to learn from.

Instead of defining "similar" MDPs, we ask how well one can approximate all value functions for a set of *anticipated tasks* $\mathcal{M}$. The set of all value functions one might encounter during *value iteration* is huge. For LSTD, however, one only has to consider fixed points $f^\pi \in L^2(\mathcal{Z}, \xi)$ of the combined operator $\hat{\Pi}_\xi^\phi \left[\hat{B}^\pi[\cdot]\right]$ (see Section 2.3). Note that there is a unique fixed point $f^\pi$ for every policy $\pi$ from the set of allowed policies $\Omega$, for example all deterministic policies. Moreover, Tsitsiklis and Van Roy (1997) have derived the upper bound

$$\left\| v^\pi - f^\pi \right\|_\xi \leq \frac{1}{\sqrt{1-\gamma^2}} \left\| v^\pi - \hat{\Pi}_\xi^\phi \left[ v^\pi \right] \right\|_\xi ,$$

which means that the approximation error (left hand side) is bounded by the projection error of *true* value function $v^\pi \in L^2(\mathcal{Z}, \xi)$ onto $\mathcal{F}_\phi$ (right hand side). It stands to reason that a set of basis functions which minimizes the right hand side of this bound for all tasks in $\mathcal{M}$ and policies from $\Omega$ according to their occurrence can be called *optimal* in this sense.

**Definition 12** *A set of p basis functions* $\{\phi_i\}_{i=1}^p \subset L^2(\mathcal{Z},\xi)$ *is called "optimal" w.r.t. the distributions* $\rho : \mathcal{B}(\mathcal{M}) \to [0,1]$ *and* $\omega : \mathcal{B}(\Omega) \to [0,1]$*, if they are a solution to*

$$\inf_{\phi \in (L^2(\mathcal{Z},\xi))^p} \mathbb{E}\left[ \underbrace{\left\| v_m^\pi - \hat{\Pi}_\xi^\phi \left[ v_m^\pi \right] \right\|_\xi^2}_{\textit{(bound)}} \;\middle|\; \begin{array}{l} m \sim \rho(\cdot) \\ \pi \sim \omega(\cdot) \end{array} \right].$$

The expectation integrates over all true value functions $v_m^\pi$ which are determined by all policies $\pi \in \Omega$, drawn from some distribution $\omega : \mathcal{B}(\Omega) \to [0,1]$ (e.g., uniform) and all tasks $m$ drawn from distribution[28] $\rho : \mathcal{B}(\mathcal{M}) \to [0,1]$. A similar definition of optimality has been proposed in the context of *shaping functions*[29] (Snel and Whiteson, 2011). Other definitions of optimality are discussed in Section 6.2.

The optimization problem in Definition 12 has no general analytic solution. In particular there is no solution for one MDP and all policies, which would be ideal for policy iteration (e.g., LSPI, Section 2.4). There is another special case, however, which demonstrates the setting under which SFA extracts nearly optimal approximation spaces.

For a fixed policy $\pi \in \Omega$ and task $m \in \mathcal{M}$ one can calculate the exact value function $v^\pi \in L^2(\mathcal{Z},\xi)$. Let $(\hat{I} - \gamma \hat{P}^\pi)^{-1}$ denote the inverse operator[30] to $(\hat{I} - \gamma \hat{P}^\pi)$, then

$$v^\pi \quad \stackrel{!}{=} \quad \hat{B}^\pi[v^\pi] \quad = \quad r^\pi + \gamma \hat{P}^\pi[v^\pi] \quad = \quad (\hat{I} - \gamma \hat{P}^\pi)^{-1}[r^\pi].$$

Substituting this into Definition 12, one can give an analytic solution $\phi_i \in L^2(\mathcal{Z},\xi), i \in \{1,\ldots,p\}$, for all tasks within the same environment,[31] that is, $\mathcal{M} := \{(\mathcal{X},\mathcal{A},P,r) \mid r \sim \rho\}$, restricted to the sampling policy $\pi$, that is, $\Omega := \{\pi\}$. The key insight is that the only allowed difference between tasks is the expected reward function $r^\pi : \mathcal{Z} \to \mathbb{R}$. If we do not constrain the possible reward functions (e.g., all states are possible goals for navigation), their statistics $\rho$ can be described as a *white noise functional* in $L^2(\mathcal{Z},\xi)$ (Holden et al., 2010, see also Footnote 25 on Page 2085).

**Theorem 13** *For any infinite ergodic Markov chain with steady state distribution* $\xi$ *over state space* $\mathcal{Z}$*, SFA selects features from function set* $\mathcal{F} \subset L^2(\mathcal{Z},\xi)$ *that minimize an upper bound on the optimality criterion of Definition 12 for sampling policy* $\pi$ *and discount factor* $\gamma > 0$*, under the assumption that the mean-reward functions* $r^\pi : \mathcal{Z} \to \mathbb{R}$ *are drawn from a white noise functional in* $L^2(\mathcal{Z},\xi)$*.*

**Proof:** see Appendix A, Page 2111. ■

The main result of Theorem 13 is that under the above assumptions, SFA approximates the *optimal basis functions* of Definition 12. To be exact, the SFA objective minimizes a *bound* on the optimality criterion. A closer look into the the proof of Theorem 13 on Page 2111 shows that the exact solution in Definition 12 requires a bias-free estimation of the term $\left\| (\hat{P}^\pi)^*[\phi_i] \right\|_\xi^2$, which is impossible without *double sampling* (see, e.g., Sutton and Barto, 1998). We argue therefore that *SFA constitutes the best approximation to optimal features one can derive using a single Markov chain*. Note that unlike the results in Sections 4.1 and 4.2, this conclusion is not restricted to self-adjoint transition operators.

28. To define a proper distribution $\rho$ one must formally define all anticipated MDPs in $\mathcal{M}$ over the union of all involved state-action spaces. See Snel and Whiteson (2011) for an example of such an approach.
29. In the context of value iteration, shaping functions are equivalent to an initialization of the value function.
30. The existence of such an operator is shown in Lemma 14, Page 2112.
31. With the same state (observation) space $\mathcal{Z}$, action space $\mathcal{A}$ and transition kernel $P$. This class of tasks is also called *variable-reward* tasks (Mehta et al., 2008) and applies for example when a flying robot needs to execute different maneuvers, but is constraint by the same aerodynamics (Abbeel et al., 2007).

### 4.4 Conclusion

Although no direct relationship has been proven, this section has provided evidence for a strong connection between diffusion distances and value function approximation. Our analysis suggests that *reward-based* features can represent finite-horizon value functions exactly. They do not generalize to different MDPs or policies and can thus as well be forgotten after the value estimate is updated. *Subspace-invariant* features, on the other hand, approximate infinite-horizon values optimally if the transition kernel adheres to some restrictions. Moreover, we argue that even in the general case, SFA provides the best possible construction method based on a single Markov chain. Computational complexity prevents this class of features from performing cost-efficient dimensionality reduction for LSTD, though. On the other hand, subspace-invariant features provide on average an optimal basis for *all* reward functions within the same environment. This optimality is still restricted to the sampling policy $\pi$, but SFA features should have an advantage over PVF here, as they are subspace-invariant for a much *larger* class of MDPs.

Using such features effectively for transfer learning or within policy iteration requires them to perform well for other policies $\pi'$, in other words to induce little *per-feature errors* when applied to $\hat{P}^{\pi'}$. In the absence of theoretical predictions a uniform sampling-policy $\pi$ appears to be a reasonable choice here. Note that depending on the transition kernel $P^{\pi}$, this can still yield an arbitrary steady state distribution $\xi$. PVF features are in the limit not affected by $\xi$ and *importance sampling* should be able to compensate the dependence of SFA (see Section 3.5). Theoretical statements about the influence of sampling policy and steady state distribution on SFA and PVF per-feature errors with other policies, however, are beyond the scope of this article and left for future works. See Section 6.1 for a discussion.

*Although SFA is more sensitive to the sampling policy than PVF, the presented analysis suggests that it can provide better approximation spaces for value estimation, that is, LSTD.*

## 5. Empirical Analysis

This section empirically evaluates the the construction of approximation spaces in light of the theoretical predictions of Section 4. Our analysis focuses on three questions:

1. How well does LSTD estimate the value function of a given Markov chain?

2. How good is the performance of policies learned by LSPI based on a random policy?

3. How does this performance depend on the approximation space metric?

We start with evidence for the relationship (hypothesized in Section 4.1) between *per-feature errors* (see Section 3) and how *self-adjoint* a transition operator is. Furthermore, Section 4.2 predicts that the set of MDPs for which PVF are *subspace-invariant* is a subset of the respective set of SFA. To test both possibilities we evaluated the first two questions on two discrete benchmark tasks: the *50-state chain* in Section 5.2 and the more complicated *puddle-world task* in Section 5.3. The third question can not be answered with a discrete metric. To test the influence of the observation space metric, we designed a simple but realistic robot navigation task with continuous state and observation spaces (Section 5.4). A robot must navigate into a goal area, using first-person video images as observations. Results are presented in Sections 5.5, 5.6 and 5.7.

### 5.1 Comparison of Subspace Invariance

In Section 4.1 we stated our intuition that the *symmetrized transition operator* of SFA approximates the true transition operator better than the *neighborhood operator* of PVF. Here we want to substantiate this intuition by testing the *per-feature errors* of theoretical SFA and PVF solutions applied to randomly generated MDPs.

#### 5.1.1 THEORETICAL FEATURES

Discrete MDPs allow an exact calculation of the objectives described in Section 3. To test the limit case of an infinite Markov chain one can generate *theoretical* features, which is straight forward[32] for SFA. These features depend on *steady state distribution* (s.s.d.) $\xi$. Changing $\xi$ has a surprising effect on per-feature errors, in particular if one assumes a uniform $\xi$. To demonstrate this effect, all experiments with theoretical features also include this case.

Theoretical PVF, on the other hand, require a proper definition of *neighborhood*. As all states could be connected by the transition kernel, *transition possibility* (as in Section 3.3) is not always an option. We followed the *k-nearest neighbor* approach of Mahadevan and Maggioni (2007) instead and defined[33] the $k$ most probable transitions as neighbors.

For each feature $\phi_i$ from a set $\Phi_p := \{\phi_1, \ldots, \phi_p\}$ one can calculate the *per feature error* $\Delta_i^{\Phi_p} \in L^2(\mathcal{X}, \xi)$ (see Section 3). To measure how how strongly $\Phi_p$ diverges from *subspace invariance*, we add the norms of all $p$ error functions together, that is, $\sum_{i=1}^{p} \|\Delta_i^{\Phi_p}\|_\xi$. This yields a measure of subspace invariance for each set of $p$ features. To compare construction methods we also calculated the mean of the above measure over $p \in \{1, 2, \ldots, 100\}$.

#### 5.1.2 SUBSPACE-INVARIANCE AND SELF-ADJOINT TRANSITION OPERATORS

To investigate whether SFA or PVF features approximate arbitrary transition models better, we tested the per-feature errors $\Delta_i^{\Phi_p}$ of random MDPs. 100 MDPs with $d = 100$ states each were created. Each state is connected with 10 random future states and each connection strength is uniformly i.i.d. drawn. The resulting matrix is converted into a probability matrix $\mathbf{P}^{\boldsymbol{\pi}}$ by normalization. SFA features are subspace-invariant for *self-adjoint* transition operators and PVF only for a subset thereof. As the above generated transition matrices are usually not self-adjoint, we repeatedly applied a *symmetrization* operator $\hat{G} : \mathbb{R}^{d \times d} \to \mathbb{R}^{d \times d}$ to each matrix $\mathbf{P}$, that is, $\hat{G}[\mathbf{P}]_{ij} := (P_{ij} + P_{ji})/(1 + \sum_k P_{jk})$. With each application the resulting transition matrices come closer and closer to be self-adjoint.

Figure 3 shows the measure for subspace invariance for theoretical PVF and SFA with both sampling distributions $\xi$. The left figure plots this measure against the feature set size $p$. One can observe that all methods show similar errors for the original asymmetric MDP (solid lines). Application of the symmetrization operator (dashed lines), on the other hand, yields a clear advantage for one SFA method. This becomes even more apparent in the right plot of Figure 3. Here the mean measure over all feature set sizes $p$ is plotted against the number of symmetrization operator applications. One can observe that (in difference to PVF) the per-feature errors of both SFA meth-

32. SFA minimizes the slowness, in the limit according to Lemma 5: $\mathcal{S}(\phi_i) = 2\langle \phi_i, (\hat{I} - \hat{P}^{\boldsymbol{\pi}})[\phi_i] \rangle_\xi$. Let $\mathbf{P}^{\boldsymbol{\pi}}$ be the transition matrix and $\boldsymbol{\Xi}$ a diagonal matrix of steady state distribution $\xi$, which is the left eigenvector to the largest eigenvalue of $\mathbf{P}^{\boldsymbol{\pi}}$. Expressing the objective in matrix notation, the theoretical SFA features are the eigenvectors to the smallest eigenvalues of the symmetric matrix $2\boldsymbol{\Xi} - \boldsymbol{\Xi}\mathbf{P}^{\boldsymbol{\pi}} - (\boldsymbol{\Xi}\mathbf{P}^{\boldsymbol{\pi}})^\top$.

33. We tested $k \in \{1, 2, 5, 10, 20, 50, 100\}$ and present here the best results for $k = 10$.

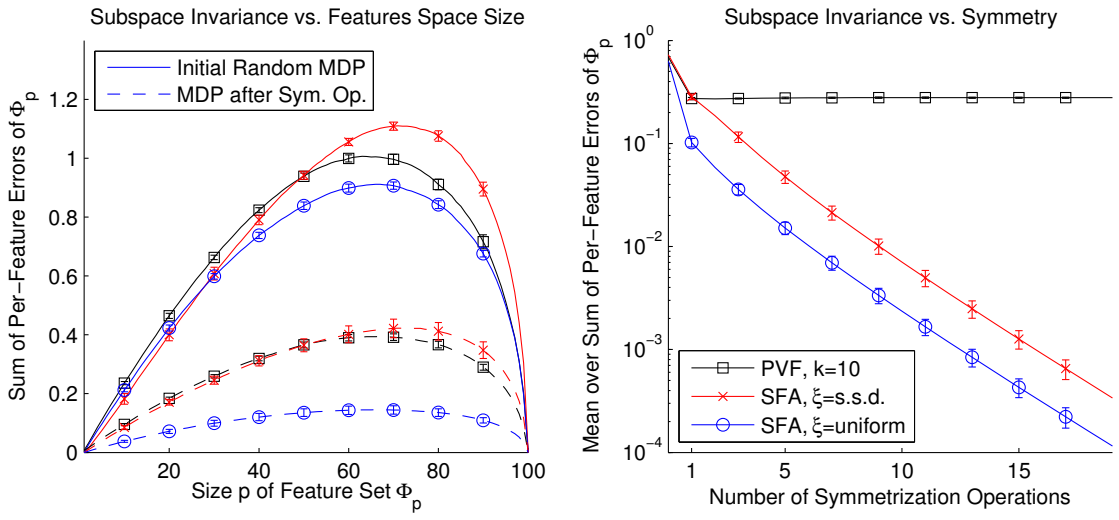


Figure 3: Subspace invariance of SFA and PVF features of random MDPs and symmetrized versions thereof. The left plot shows $\sum_{i=1}^{p} \|\Delta_i^{\Phi_p}\|_\xi$ for different feature set sizes $p$. The first symmetrization reduces per-feature errors of both methods, but all following symmetrization operations reduce only the error of SFA which is illustrated in the right plot. All means and standard deviations are w.r.t. 100 random MDPs. Note that the scale of per-feature errors differs between plots.

ods shrink the more self-adjoint the transition operator becomes. Also, SFA features with uniform distribution $\xi$ are roughly 3 times as subspace-invariant as original SFA features with steady state distribution $\xi$.

We conclude that PVF and SFA methods approximate arbitrary asymmetric MDPs equally well. However, the more self-adjoint the transition operator, the larger the advantage of SFA. Furthermore, SFA features based on a uniform distribution $\xi$ are on average more subspace-invariant than those based on the steady state distribution.

### 5.2 50-state Chain-Benchmark Task

First we investigate how well a basis constructed from a Markov chain can approximate the corresponding value function. The employed *50-state chain task* is based on a problematic 4-state MDP by Koller and Parr (2000) and has been extended in various variations by Lagoudakis and Parr (2003). Here we adopt the details from Parr et al. (2007). The task has a very similar transition- and neighborhood structure.

#### 5.2.1 TASK

50 states are connected to a chain by two actions: move left and right. Both have a 90% chance to move in their respective direction and a 10% chance to do the opposite. Non-executable transitions at both ends of the chain remain in their state. Only the 10th and 41th state are rewarded. Executing any action there yields a reward of $+1$. The task is to estimate the value function with a discount

factor $\gamma = 0.9$ from a Markov chain. Note that transition probabilities of a random policy equal the neighborhood relationships. SFA features for this policy should therefore equal PVF features.

#### 5.2.2 ALGORITHMS

We compare discrete versions of *slow feature analysis*, *proto value functions* and *Krylov bases* as described in Section 3. SFA feature sets also contained a constant feature. In this and the following experiments, higher Krylov bases have proven to be too similar for stable function approximation. We therefore orthonormalized each feature w.r.t. its predecessors, which solved the problem. After construction, the features were used to estimate the value function from the same training set with LSTD. Sampling influences value estimation here and to avoid the resulting bias we measure the difference (in some norm) to the LSTD solution with a *discrete* representation.

#### 5.2.3 RESULTS

To explore the effect of the sampling policy, we tested the (non-deterministic) uniform and the (deterministic) optimal policy. Figure 4 plots mean and standard deviation of the LSTD solution difference in $L_2$ norm w.r.t. 1000 trials and 4000 samples each. To make sure all states are visited, an optimal policy trial is sampled in 40 trajectories with random start states and 100 samples each. Training sets that did not visit all states were excluded. As reported by previous works (Petrik, 2007; Parr et al., 2008; Mahadevan and Liu, 2010), reward-based features like Krylov bases perform in this task much better than subspace-invariant features. Solutions with PVF and SFA features are virtually identical for the random policy, as the transition probabilities of SFA equal the neighborhood relations of PVF. There are distinguishable differences for the optimal policy, but one can hardly determine a clear victor. Using the $L_\infty$ norm for comparison (not shown) yields qualitatively similar results for the two feature spaces. Policy iteration did also not yield any decisive differences between SFA and PVF (not shown). In conclusion, the 50-state chain appears to belong to the class of MDPs for which features learned by SFA and PVF are not always identical, but *equally* able to estimate the value functions with LSTD.

### 5.3 Puddle-world–Benchmark Task

The puddle-world task was originally proposed by Boyan and Moore (1995), but details presented here are adapted from Sutton (1996). It is a continuous task which we discretize in order to compare reward-based features. This is a common procedure and allows to evaluate robustness by running multiple discretizations. In comparison to the 50-state chain the task is more complex and exhibits differing transition- and neighborhood-structures.

#### 5.3.1 TASK

The state space is a two dimensional square of side length 1. Four actions move the agent on average 0.05 in one of the compass directions. The original task was almost deterministic (Sutton, 1996) and to make it more challenging we increased the Gaussian noise to a standard deviation of 0.05. Centered in the $(1,1)$ corner is an absorbing circular goal area with radius 0.1. Each step that does not end in this area induces a punishment of $-1$. Additionally, there exist two puddles, which are formally two lines $(0.1,0.75) \longleftrightarrow (0.45,0.75)$ and $(0.45,0.4) \longleftrightarrow (0.45,0.8)$ with a radius of 0.1 around them. Entering a state less than 0.1 away from one of the center-lines is pun-

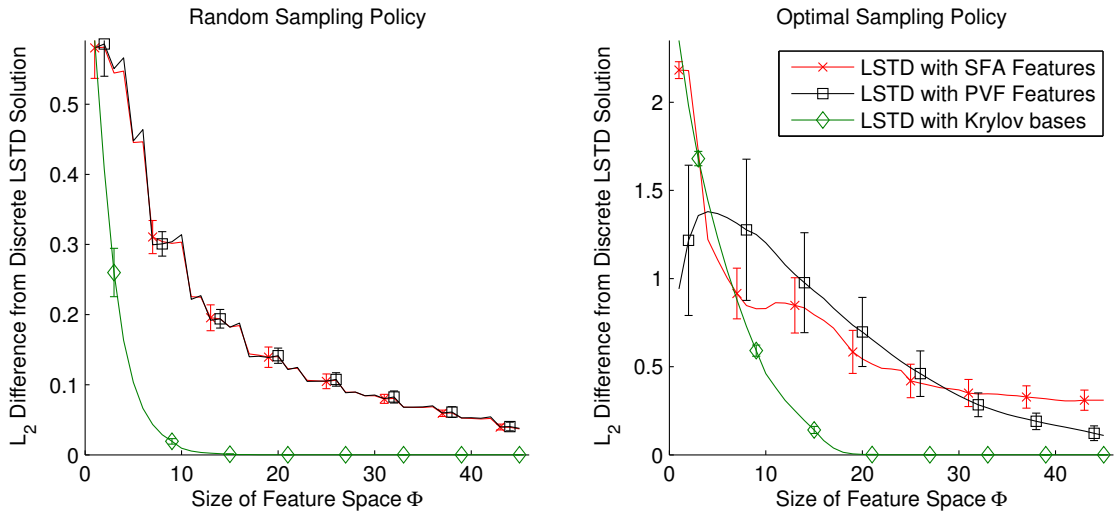


Figure 4: Difference in $L_2$ norm between approximated and discrete LSTD solutions vs. feature space size for random (left plot) and optimal (right plot) sampling policies in the 50-state chain. Means an standard deviations w.r.t. 1000 trials. $L_\infty$ differences decrease slower but show otherwise the same qualitative trends.

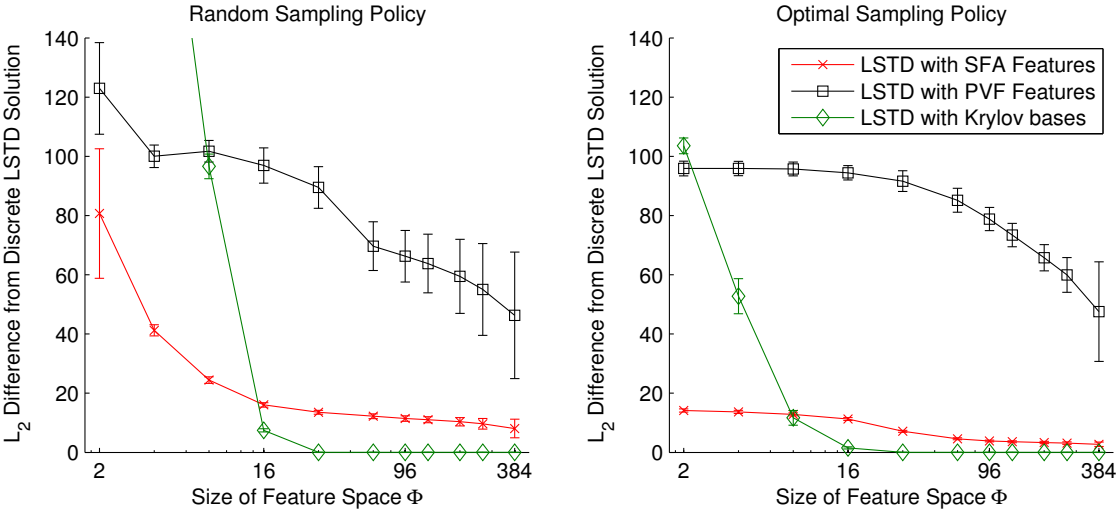


Figure 5: Difference in $L_2$ norm between approximated and discrete LSTD solutions vs. feature space size in puddle-worlds. Means and standard deviations are w.r.t. 10 training sets for each state space size $\{20\times 20, 25\times 25, \ldots, 50\times 50\}$, sampled with random (left plot) or optimal (right plot) policies. Note the logarithmic x-axis. Measuring the differences in $L_\infty$ norm yields the same qualitative trends.

ished with $-400$ times the distance into the puddle. The task is to navigate into the goal, collecting as little punishment as possible. Seven discretizations with horizontal and vertical side length of $\bar{s} \in \{20, 25, \ldots, 50\}$ states were tested. Transition probabilities of states $\boldsymbol{x}_i$ to states $\boldsymbol{x}_j$ are the normalized *continuous* probabilities (average movement $\boldsymbol{\mu}_a$ plus Gaussian noise with standard deviation $\tilde{\sigma}$), that is

$$P_{ij}^a := \frac{\exp(-\frac{1}{2\tilde{\sigma}^2}\|\boldsymbol{x}_i + \boldsymbol{\mu}_a - \boldsymbol{x}_j\|_2^2)}{\sum_{k=1}^{\bar{s}^2}\exp(-\frac{1}{2\tilde{\sigma}^2}\|\boldsymbol{x}_i + \boldsymbol{\mu}_a - \boldsymbol{x}_k\|_2^2)}, \qquad \forall i, j \in \{1, \ldots, \bar{s}^2\}, \quad \forall a \in \{1, \ldots, 4\}.$$

### 5.3.2 ALGORITHMS

We evaluated discrete versions of *slow feature analysis* (SFA), *proto value functions* (PVF) and orthonormalized *Krylov bases*. To test the influence of sampling on feature construction we also evaluated the theoretical features of all three algorithms[34] (see Section 5.1). The algorithms were trained with a long (uniform) random policy Markov chain $\{\boldsymbol{x}_t\}_{t=1}^n$ in both LSTD and LSPI evaluations. To see the effect of different policies on LSTD, we trained all three on the *optimal* policy as well. In this case the training set consists of randomly initialized trajectories of length 20 to overcome sampling problems. We chose $n = 80\,\bar{s}^2$, as large state spaces require more samples to converge.

All constructed approximation spaces were tested with LSTD and LSPI[35] under discount factor $\gamma = 0.99$ on the same training sequence $\{\boldsymbol{x}_t\}_{t=1}^n$ used in feature construction. Policy iteration ended if the value of *all* samples differed by no more than $10^{-8}$ or after 50 iterations otherwise. To investigate asymptotic behavior we also tested LSPI with all state-action pairs and *true* mean future states as training set, corresponding to the limit of an infinite Markov chain. Performance of a (deterministic) policy learned with LSPI is measured by the *mean accumulated reward* of 1000 trajectories starting at random states. A test trajectory terminates after 50 transitions or upon entering the goal area.

### 5.3.3 LSTD EVALUATION

We tested the LSTD approximation quality as in Section 5.2. Figure 5 shows the difference in $L_2$ norm between the LSTD solution based on the constructed features and a discrete state representation vs. the number of employed features $p \in \{2, 4, 8, 16, 32, 64, 96, 128, 192, 256, 384\}$. Means and standard deviations are w.r.t. 10 training sets for each state space size $\{20\times 20, 25\times 25, \ldots, 50\times 50\}$, sampled with random (left plot) or optimal (right plot) policies. Note in comparison to Figure 4 that the x-axis is logarithmic. Reward-based Krylov bases have a clear advantage for $p \geq 16$ features (and reach perfection for $p \geq 64$), similar to the 50-state chain task. Fewer SFA features, on the other hand, capture the value function much better. This has also been observed for large discount factors $\gamma$ when comparing BEBF and modified PVF (Mahadevan and Liu, 2010). The different encoding of diffusion distances (Theorems 6 & 7, Page 2085) provides a good explanation for this effect: Krylov bases represent finite-horizon value functions perfectly (Corollary 10, Page 2087),

34. We tested theoretic PVF with $k \in \{5, 10, 15, 20, 25, 50\}$ and chose $k = 10$. For larger $k$ we observed slowly degrading performance, which is more pronounced under realistic LSPI sampling.

35. As convergence to the optimal policy can not be guaranteed for LSPI, the eventual policy depends also on the initial (randomly chosen) policy $\boldsymbol{\pi}_0$. In difference to Lagoudakis and Parr (2003) we used throughout this article a *true* (non-deterministic) random policy $\boldsymbol{\pi}_0(a|\boldsymbol{z}) = \frac{1}{|\mathcal{A}|}, \forall \boldsymbol{z} \in \mathcal{Z}, \forall a \in \mathcal{A}$. This heuristic appeared to be more robust in large feature spaces.

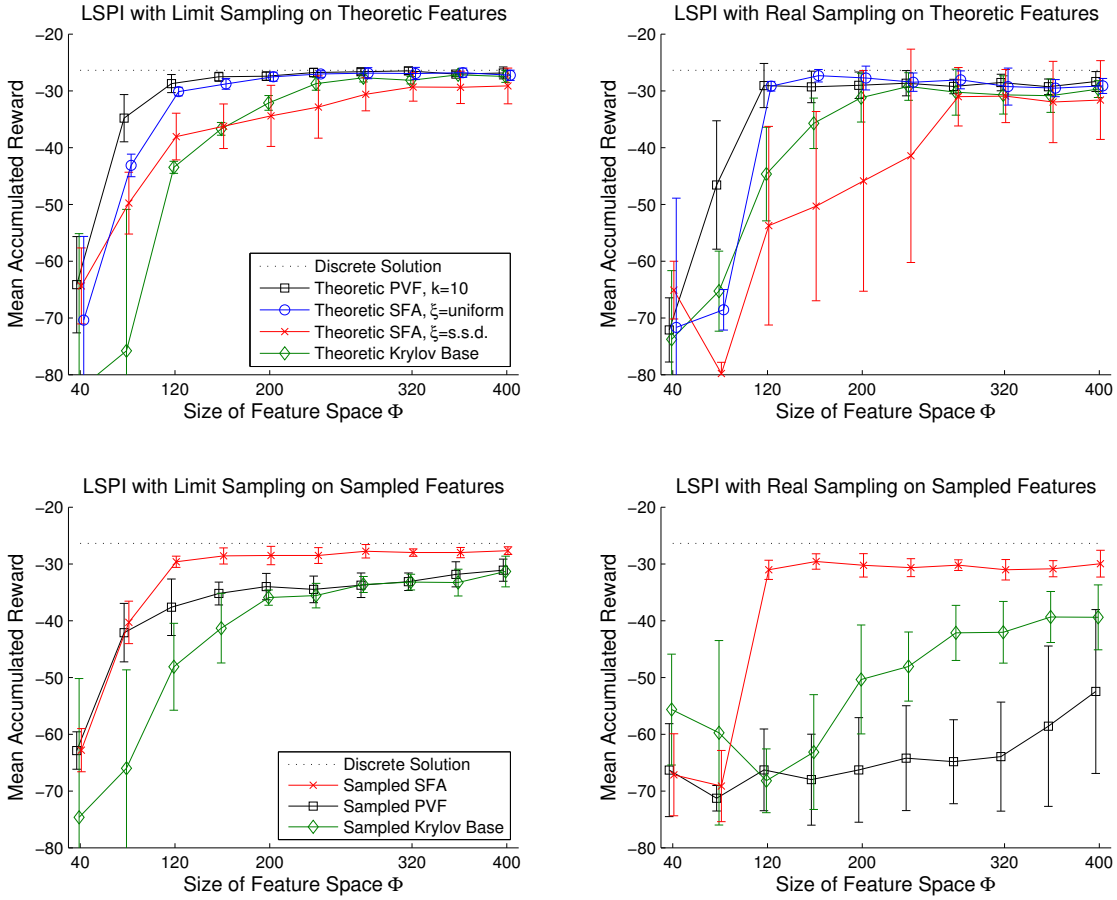


Figure 6: Performance of LSPI with optimal (left column) and realistic sampling (right column) in puddle-worlds. The upper row employed theoretical features and the lower row features constructed from samples. Means and standard deviations are w.r.t. state space sizes $\{20 \times 20, 25 \times 25, \ldots, 50 \times 50\}$.

whereas SFA represents the infinite-horizon values approximatively (Corollary 9, Page 2086). However, Figure 5 clearly supports our hypothesis (proposed in Section 4.4) that *SFA can provide better approximation spaces for value function approximation with LSTD than PVF*. With respect to the previous subsection one should extend this hypothesis by adding *when transition and neighborhood structures are dissimilar*.

### 5.3.4 LSPI EVALUATION

Figure 6 shows the LSPI performance with both theoretic (upper row) and sampled features (lower row). LSPI was trained with an optimal training set (left column) and a realistic sequence drawn by

a random policy (right column). The top-left plot is the most hypothetical and the bottom-right plot the most realistic scenario. Means and standard deviation are w.r.t. state space sizes.

First and foremost, note that the sampled SFA features (crosses, lower row) perform significantly (more than one standard deviation) better, except in the under-fitting regime of 80 features and less. They are also the only sampled features that are robust against LSPI sampling, which can be seen by similar performance in both bottom plots. In comparison with theoretic SFA features, their performance resembles the solution with uniform distribution $\xi$ (circles, upper row), which clearly outperforms SFA features based on the very similar steady state distribution $\xi$ (crosses, upper row). This is surprising, as the two distributions only differ at the borders of the square state space. Similar to Section 5.1, the uniform distribution appears nonetheless to have an advantage here.

Secondly, note that although theoretic PVF features (squares, upper row) rival the best of SFA, sampled PVF features (squares, lower row) are less successful. In the most realistic case the performance appears almost unaffected by additional features. This is consistent with our observations of the LSTD solutions in Figure 5, where sampled PVF features performed equally bad under both policies. Reward-based Krylov Features (diamonds), on the other hand, appear relatively stable through most settings and only cave in at the most realistic scenario. However, note that for LSPI *reward-based* features do not have any (empirical or theoretical) advantage over *subspace-invariant* features.

Although not exactly predicted by theory, we see this as evidence that *discrete SFA can construct better approximation spaces for LSPI than PVF or Krylov bases*. Theoretical PVF may rival SFA, but realistic sampling appears to corrupt PVF features. This advantage of SFA may be lost for other sampling policies, though. As an example, observe the strong influence of minor changes in the sampling distribution $\xi$ on the theoretical SFA solution.

### 5.4 Visual Navigation-Setup

We investigate our third question on Page 2090 with a simple but realistic continuous application task. Continuous state spaces impede the use of reward-based features, but allow an analysis of the presented metric by encoding observations with PCA. Our focal idea is to compare basis construction approaches based on different metrics in the *observation space* and the underlying *state space*. A *visual navigation task* guides a robot into a designated goal area. Observations are first-person perspective images from a head mounted video camera (see Figure 2, Page 2071, for a sketch of the control process). While robot coordinates come close to encoding diffusion distances of random policies, these observations clearly do not. A comparison between PCA and SFA features in these two cases can thus illuminate the role of the observation metric in the construction of continuous basis functions. Adequacy of SFA in this task is demonstrated with a real robot (Section 5.7) and extensively compared with PVF and PCA in a realistic simulation (Section 5.6). Section 5.5 provides an evaluation of the involved sparse subsets.

#### 5.4.1 ROBOT

We used the wheeled PIONEER 3DX robot (Figure 7a), equipped with a head mounted BUMBLEBEE camera for the experiments. The camera recorded mono RGB images from a first-person perspective of the environment in front of the robot with a 66° field of vision (Figure 7b). The robot was able to execute 3 commands: Move approximately 30cm forwards; turn approximately 45° left or right.

a) Robot b) Robot View c) Simulator View

Figure 7: a) PIONEER 3DX wheeled robot. b) A first-person perspective image from the robot camera. c) Corresponding image of the simulator.

#### 5.4.2 ENVIRONMENT

The robot had to navigate within a rectangular $3m \times 1.8m$ area surrounded by walls, approx. $1m$ in height. We covered the walls with different wallpaper to have a rich texture (sketched in Figure 10). The scenery was well-lit by artificial light. A camera installed at the ceiling allowed us to track the robot's position for analysis. We also ran simulations of the experiment for large scale comparison. Based on photographed textures, the JAVA3D engine rendered images from any position in the simulated environment. Those images were similar to the real experiment, but not photo-realistic (Figure 7c).

#### 5.4.3 TASK

Starting from a random start position, the robot has to execute a series of actions (move forward, turn left, turn right) that lead to an unmarked goal area in as few steps as possible without hitting the walls. Learning and control are based on the current camera image $\boldsymbol{z}_t$ and a corresponding reward signal $r_t \in \{-1, 0, +1\}$ indicating whether the robot is in the goal area ($r_t = +1$), close to a wall ($r_t = -1$) or none of the above ($r_t = 0$).

#### 5.4.4 ALGORITHMS

The algorithms *sparse kernel principal component analysis* (SK-PCA, Section 3.2), *k-nearest-neighbor extension of proto value functions* (kNN-PVF, Section 3.3) and *regularized sparse kernel slow feature analysis* (RSK-SFA Section 3.4), were implemented[36] using a Gaussian kernel $\boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{y}\|^2), \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Z}$. To ensure the respective *function sets* from which the basis functions were chosen are roughly equivalent, we set the Gaussian width parameter $\sigma = 5$ for all algorithms. Runtime deviated at most by a factor of 2, as SFA requires two eigenvalue decompositions. The importance sampling modification of SFA described in Section 3.5 assigned a weight[37] of $\vartheta(a_{move}) = 5$ to forward movements and $\vartheta(a_{turn}) = 1$ to rotations. This balanced out *relative velocities* and ensured that the first features encode both spatial and orientational information.

36. RSK-SFA and SK-PCA select functions from a RKHS, based on any positive semi-definite kernel, for example, the Gaussian kernel. kNN-PVF are based on a k-nearest-neighbors graph with edges weighted by a Gaussian kernel of the distance between nodes.
37. We tested other weights with less detail. The results appear stable around $\vartheta(a_{move}) = 5 \pm 1$ but exact statements require an order of magnitude more simulations than we were able to provide for this article.

All discussed algorithms also require a sparse subset of the data. For optimal coverage it appears straightforward to select this subset uniformly in observation space $\mathcal{Z}$. This can be achieved with the *matching pursuit for affine hull maximization* algorithm (MP-MAH, Böhmer et al., 2012). However, we show in Section 5.5 that this intuition is wrong and in fact one should instead select the subset uniformly distributed in the true state space $\mathcal{X}$, which can be achieved by applying MP-MAH on $\mathcal{X}$ instead of $\mathcal{Z}$. As $\mathcal{X}$ is usually not known explicitly, the comparison between algorithms in Section 5.6 is performed with randomly drawn subsets.

#### 5.4.5 SIMULATED EXPERIMENTS

Drawing actions uniformly, we generated 10 independent random walks with 35,000 samples each. The rendered images were brightness corrected and scaled down to $32 \times 16$ RGB pixels, yielding observations $\boldsymbol{z}_t \in \mathcal{Z} \subset [0,1]^{1536}$. Each training set was used to construct one feature space for each of the above algorithms with a sparse subset of 4000 samples (see Section 5.5). The resulting basis functions were applied on the corresponding training set.

The control policy was learned by LSPI on the first $p \in \{2,4,8,\ldots,2048\}$ constructed features $\phi_i : \mathcal{Z} \to \mathbb{R}$ and a constant feature $\phi_0(\boldsymbol{z}) = 1\,, \forall \boldsymbol{z} \in \mathcal{Z}$. The discount factor was $\gamma = 0.9$ and the goal area was located in the lower right corner with a radius of 50cm around x=260cm and y=40cm (see right plot of Figure 10). A distance to the walls of 40cm or less was punished.

The resulting policies were each tested on 200 test trajectories from random start positions. Navigation performance is measured[38] as *fraction of successful trajectories*, which avoid the wall and hit the goal in less than 50 steps.

#### 5.4.6 ROBOT EXPERIMENT

Running the robot requires a large amount of time and supervision, preventing a thorough evaluation. For RSK-SFA the continuous random walk video of approx. 10 hours length was sampled down to approx. 1 Hz and a sparse subset of 8,000 out of 35,000 frames was selected with the MP-MAH algorithm (Böhmer et al., 2012). The first 128 RSK-SFA and one constant feature were applied on a training set of 11,656 *transitions* sampled from the same video. LSPI was trained as in the simulator experiments and evaluated on each 20 test trajectories for the lower right and for a smaller center goal with a radius of 20cm (see Figure 10).

### 5.5 Visual Navigation-Sparse Subset Selection

The analysis of Böhmer et al. (2012) suggests that a sparse subset uniformly distributed in observation space $\mathcal{Z}$ improves the performance of RSK-SFA. We observed the same effect on the *slowness* (not shown), but interestingly not on the *navigation performance* of the respective LSPI solution. Figure 8 plots this performance against the number of features $p$ used for LSPI. Random subset selection (crosses) outperforms MP-MAH selection on $\mathcal{Z}$ with the correct kernel width (upward triangle) significantly. Moreover, random selection yields high performance reliably (small standard deviation), whereas subsets that are uniformly distributed in $\mathcal{Z}$ result in unpredictable behavior for large approximation spaces. Shrinking the kernel parameter $\sigma$ of MP-MAH (*not* of RSK-SFA) decreases the disadvantage as the algorithm converges theoretically to random selection in the limit

38. Other measures are possible. We tested "mean number of steps to goal" and also compared those to an almost optimal policy. However, the resulting plots were qualitatively so similar that we stuck to the simplest measure.

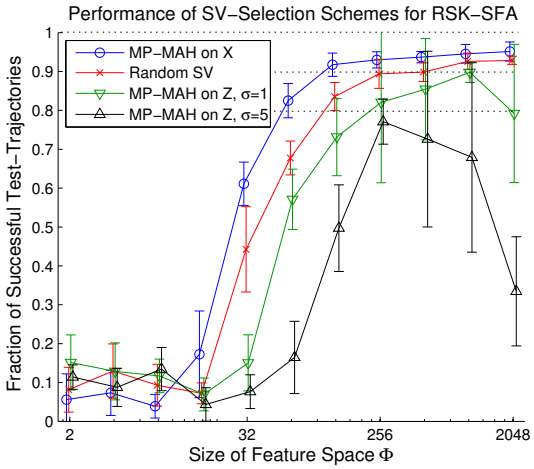


Figure 8: The influence of *sparse subset selection* for RSK-SFA on the performance of LSPI. Mean and standard deviation (over 10 independent training sets) of the navigation performance are plotted against the (logarithmic) number of features used in LSPI. Note that a subset equally distributed in the true state space $\mathcal{X}$ (circles) is most efficient and reliable, whereas for equal distribution in observation space $\mathcal{Z}$ (triangles) LSPI becomes unreliable in large approximation spaces $\mathcal{F}_\phi$.

$\sigma \to 0$ (downwards triangles). Using MP-MAH to select a subset uniformly distributed in the *true state space*[39] $\mathcal{X}$ (circles), however, demonstrates that this is not an over-fitting effect as the learned policies outperform those of random selection significantly. Section 6.3 attempts an explanation of these results and discusses some practical implications for sparse subset selection.

However, in practice $\mathcal{X}$ is usually not explicitly known. The main comparison in Section 5.6 is therefore performed with randomly drawn subsets. Note that our random walk sampled the state space $\mathcal{X}$ almost uniformly, which is the explanation for the good performance of randomly selected subsets. A random selection from biased random walks, for example, generated by other tasks, will severely decrease the navigation performance. We expect a similar effect in other sparse kernel RL methods (e.g., Engel et al., 2003; Xu, 2006).

### 5.6 Visual Navigation-Comparison of Algorithms

The left plot of Figure 9 shows a comparison of the effect of all discussed basis function construction algorithms on the control policy learned by LSPI. Note that the algorithms are based on the same randomly chosen sparse subset of 4000 samples: 4000 orthogonal features extracted by any algorithm span approximation space $\mathcal{F}_{\{\boldsymbol{\kappa}(\cdot,\boldsymbol{s}_i)\}_{i=1}^m}$. Therefore all algorithms perform equally well with enough ($p \geq 1024$) features. One can, on the other hand, observe that both the original RSK-SFA (crosses) and the modified algorithm (circles) outperform SK-PCA (triangles) and kNN-PVF[40]

39. Robot position and orientation coordinates for which the Euclidean metric resembles diffusion distances.
40. We tested the navigation performance based on kNN-PVF basis functions for the kNN parameters $k \in \{10, 25, 50\}$. As the results did not differ significantly, we omitted them here.

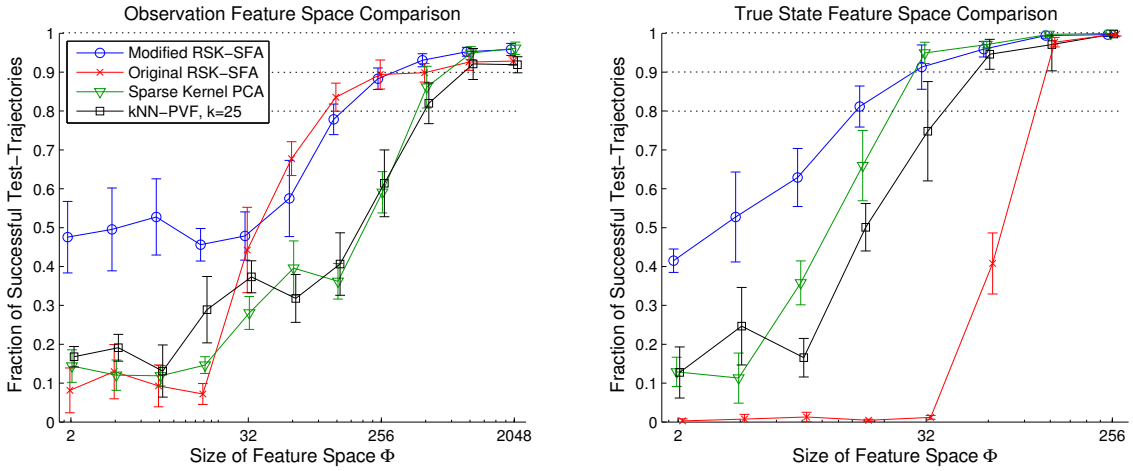


Figure 9: Comparison of subspace-invariant feature space construction algorithms on observations $\boldsymbol{z} \in \mathcal{Z}$ (camera images, left plot) and on the true state $\boldsymbol{x} \in \mathcal{X}$ (robot coordinates, right plot). Mean and standard deviation (over 10 independent training sets) of the navigation performance are plotted against the number of features used in LSPI. Note that the x-axis is logarithmic and differs between plots. The dotted lines indicate 80%, 90% and 100% performance levels.

(squares) significantly in medium sized feature spaces ($32 \leq p < 1024$ features). In particular the 80% and 90% levels of navigation performance are both reached with roughly a quarter of basis functions. We attribute this advantage to approximation spaces encoding diffusion distances rather than similarities in $\mathcal{Z}$.

Close inspection of the feature space revealed that the first RSK-SFA features encode spatial information only (not shown). This is due to the different velocities of rotations and movements of the robot in the true state space $\mathcal{X}$ (see Section 3.5 and Franzius et al., 2007). Consequentially, small feature spaces can not express policies involving rotation and thus perform poorly. The modified algorithm balances this handicap out and yields steady performance in small feature spaces ($2 \leq p < 32$ features). If the necessary orientational components are encoded in the original RSK-SFA basis functions ($p \geq 32$), both algorithms perform comparable as they span (almost) the same approximation space. We conclude that the modified RSK-SFA algorithm is the superior continuous basis function construction scheme, irrespective of feature space size $p$.

### 5.6.1 COMPARISON IN TRUE STATE SPACE $\mathcal{X}$

To confirm the above effect is due to the difference in the observed metric and the diffusion metric constructed by SFA, we run the same experiment with $\mathcal{Z} \approx \mathcal{X}$. For this purpose we applied all basis function construction algorithms on robot coordinates. The true state space $\mathcal{X}$ is supposed to be equipped with a diffusion metric, which should have a constant distance between successive states. We thus divided the spatial coordinates by an average movement of 30cm and the robots orientation by the average rotation of 45°. The kernel width was chosen $\sigma = 2$ to allow sufficient overlap.

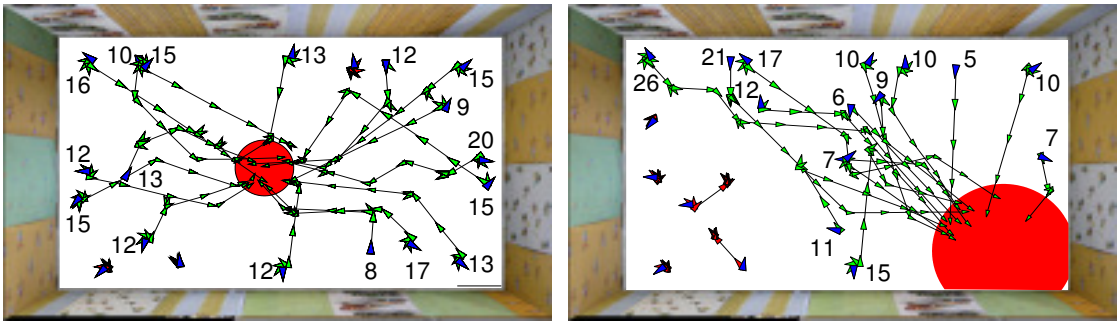


Figure 10: The recorded robot trajectories of a control policy learned by LSPI using RSK-SFA features. The plot shows 40 trajectories with random starting positions and orientation (dark triangles), aiming to hit the circular goal area. The numbers indicate the number of steps the robot required to reach the goal.

The right plot of Figure 9 shows the navigation performance of the learned LSPI policy. Due to the suitability of the observation space, a relative small number[41] of basis functions of $p \geq 128$ suffices for an almost perfect navigation performance with all algorithms. However, relative velocities influence the RSK-SFA solution as well, which renders the original algorithm (crosses) almost useless and demonstrates its sensitivity to the sampling policy. The modified algorithm (circles) performs best here as well, but demonstrates an advantage only in very small feature spaces ($2 \leq p < 32$). Together with the good performance of SK-PCA (triangles), which slightly outperforms the state-of-the-art method kNN-PVF (squares), this can be seen as evidence that methods based on Euclidean distances in $\mathcal{Z} \approx \mathcal{X}$, which are already close to diffusion distances, suffice to learn a good policy in this setup.

#### 5.6.2 CONCLUSION

The results presented in this subsection provide ample evidence for the hypothesis that *SFA can construct better approximation spaces for LSPI than PVF or PCA*, but also demonstrates its sensitivity to the sampling policy. We have empirically shown that the novel modified RSK-SFA algorithm outperforms all continuous basis function construction schemes reviewed for this article. The advantage to baseline method PCA vanishes when the observations already conform to a diffusion metric. This suggests that SFA performs essentially PCA based on diffusion rather than Euclidean distances. It also implies an answer to our third question on Page 2090: *LSPI performance is facilitated by approximation spaces that encode diffusion distances of a uniform random policy*.

### 5.7 Visual Navigation-Robot Demonstration

A thorough reproduction of the experiments from Section 5.6 on a real robot is beyond the scope of this article. Measuring the navigation performance of 200 test trajectories can not easily be automated and requires an enormous amount of supervision. However, we demonstrate how our

41. Note that, in difference to the space of images, this observation space is three dimensional. Instead of (at least potentially) dimensionality reduction, these basis functions form an overcomplete basis.

key method, that is SFA in the form of RSK-SFA, performs on a real robot. After 10 LSPI iterations with 128 RSK-SFA basis functions, the control policy achieved a success rate of 75% in two separate tasks depicted in Figure 10. 17 out of 20 test trajectories hit a centered goal within 20 steps (left plot) and 13 out of 20 trajectories reached the much farther goal area in the lower right corner (right plot). The failed trajectories made at most one forward move and then started to oscillate between right and left rotations. These oscillations also appeared often in successful trajectories whenever the robot switched from one action to another. This can be seen in the large number of steps some trajectories require to reach the goal area. We attribute these oscillations to approximation errors and noise in the basis functions. Whenever the policy changes from one action to another, the respective Q-values must be close-by and small deviations from the true value can drastically influence the greedy action selection. If the wrongfully chosen action is a rotation, the correct reaction would be to rotate back. We observed these never ending oscillations in simulations as well. The problem can usually be diminished by adding more basis functions. The fact that the robot was sometimes able to break the oscillation is an indicator for noise in images and motors. This behavior could have been easily avoided by introducing some simple heuristics, for example "never rotate back" or "in doubt select the previous action". As this article investigates approximation spaces rather than optimizing the visual task itself, we omitted those heuristics in our experiments. For practical implementations, however, they should be taken under consideration.

## 6. Discussion

This section discusses implications of the presented results and points out open questions and potential directions of future research.

### 6.1 SFA Features as Basis Functions for LSPI

Theoretical predictions in Section 4 cover value function estimation with LSTD for the current sampling policy. When LSPI changes this policy, all statements become strictly invalid. Nonetheless, the results in Sections 5.3 and 5.6 demonstrate the applicability of SFA features as basis functions for LSPI. Both experiments also show that an *importance sampling* modification to SFA can yield even better results. Maybe there exists a policy $\tau$ and a state-distribution $\zeta$ which are *optimal* (in the sense of Definition 12) for at least all deterministic policies encountered during LSPI. In the experiments this appeared to be a uniform policy and a uniform state-distribution, but we can not claim any generality based on the presented evidence alone. We still want to suggest a possible explanation:

Uniform $\tau$ and $\zeta$ might be an optimal training sets for LSPI. Koller and Parr (2000) proposed this in light of a pathological 4-state chain MDP (similar to Section 5.2). LSTD weights the importance of value approximation errors and predicts thus accurate Q-values according to $\zeta$. Using steady state distribution $\xi$ of some sampling policy $\pi$ implies that decisions at often visited states should be more reliable that those at seldom visited states. Take the optimal policy in a navigation task (like Section 5.3 or Section 5.4) as an example. Transition noise guarantees an ergodic Markov chain, but the steady state distribution will concentrate almost all mass around the goal. As a result, Q-values far away from the goal can be approximated almost arbitrarily bad and decisions become thus erratic. This is not the intended effect. To solve the task, one needs to control the approximation error of all states one will encounter until the task is complete. Without knowledge about certainty, every

decision along the way might be equally important. A uniform $\zeta$ reflects this. A similar argument can be made for a uniform $\tau$ in the context of policy iteration.

Future works might be able to identify the corresponding Radon-Nikodym derivatives $\frac{d\zeta}{d\xi} : \mathcal{Z} \to \mathbb{R}^+$ and $\frac{d\tau}{d\pi} : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}^+$ and use *importance sampling* as discussed in Section 3.5. Until then the best approach appears to be sampling with a random policy and fine tuning with the modified SFA algorithm presented in this article.

### 6.2 Optimal Basis Functions

The analysis in Section 4.3 introduces the concept of *optimal basis functions* (Definition 12, Page 2088) to construct approximation spaces for LSTD with SFA. However, Theorem 13, Page 2089, implies that SFA features are only *optimal* for sampling policy $\pi$. This optimality is lost when policy iteration varies $\pi$, but optimizing the basis functions w.r.t. policy has no analytic solution and appears not feasible. Besides the question of feasibility, there are alternatives to the definition of *optimal basis functions* in Section 4.3. Here we suggest three possibilities:

1. Given one task $m = (\mathcal{X}, \mathcal{A}, P, R)$ only, the optimal basis functions for LSPI encode the true value function $v_m^\pi(\cdot)$ of $m$ with all possible policies $\pi \in \Omega$, that is

$$\inf_{\phi \in L^2(\mathcal{Z},\xi)} \mathbb{E}\left[ \left\| v_m^\pi - \hat{\Pi}_\xi^\phi [v_m^\pi] \right\|_\xi^2 \;\middle|\; \pi \sim \omega(\cdot) \right].$$

   As LSPI is based on a dictionary of transitions, however, the distribution $\xi$ of the weighted projection operator $\hat{\Pi}_\xi^\phi$ corresponds to the sampling distribution instead of steady state distribution of policy $\pi$.

2. Definition 12 minimizes the mean approximation error over all expected task. An alternative would be to minimize the *worst case* bound instead, that is

$$\inf_{\phi \in L^2(\mathcal{Z},\xi)} \left( \sup_{m \in \mathcal{M}, \pi \in \Omega} \left\| v_m^\pi - \hat{\Pi}_\xi^\phi [v_m^\pi] \right\|_\xi^2 \right).$$

3. The presented *weighted Euclidean norm projection* $\hat{\Pi}_\xi^\phi$ is the most commonly used choice. However, Guestrin et al. (2001) have proposed an efficient algorithm based on *supremum norm projection* $\hat{\Pi}_\infty^\phi$ for approximation of the updated value function $\hat{B}^\pi[v](\cdot)$. It is straightforward to derive a bound analogous[42] to Tsitsiklis and Van Roy (1997) and thus to define a matching *optimality criterion*

$$\inf_{\phi \in L^2(\mathcal{Z},\xi)} \left( \sup_{m \in \mathcal{M}, \pi \in \Omega} \left\| v_m^\pi - \hat{\Pi}_\infty^\phi [v_m^\pi] \right\|_\infty \right).$$

As with the criterion of Definition 12, it might not be feasible to solve these optimization problems in practice. Future works could find feasible approximations thereof, though.

42. One needs to show that $\hat{\Pi}_\infty^\phi[\cdot]$ is a non-expansion and $\hat{B}^\pi[\cdot]$ a contraction in $\|\cdot\|_\infty$ (Bertsekas, 2007).

### 6.3 Sparse Subset Selection

Results in Section 5.5 suggest that sparse subsets uniformly distributed in $\mathcal{X}$ (rather than $\mathcal{Z}$) support the best basis functions for LSPI. One possible explanation is the effect of the sparse subset distribution $\chi : \mathcal{B}(\mathcal{X}) \to [0,1]$ onto the objective. In limit of an infinite subset of an infinite Markov chain, extracted features $\phi_i(z) = \int \chi(ds)\, \alpha_i(s)\, \kappa(z,s) = \langle \alpha_i, \kappa(z,\cdot)\rangle_\chi, \forall z \in \mathcal{Z}$, are determined by coefficient functions $\alpha_i \in L^2(\mathcal{X},\chi)$. To determine these functions, RSK-SFA therefore approximates *generalized eigenfunctions* in $L^2(\mathcal{X},\chi)$, which are strongly affected by norm $\|\cdot\|_\chi$. Discrete SFA, on the other hand, represents every state by one unique variable, which corresponds to a uniform distribution $\chi$ in $\mathcal{X}$. It is easy to see that a uniform $\chi$ preserves the optimization problem best. Selecting a subset uniformly in $\mathcal{Z}$, however, does not generally yield a uniform $\chi$ in $\mathcal{X}$, due to the difference in Euclidean and diffusion distances. Results presented in Sections 5.3 and 5.6 demonstrate how sensitive the LSPI performance based on SFA features is to sampling policy and sampling state-distribution. Non-uniform $\chi$ will probably decrease performance similarly. It seem therefore reasonable to attribute the results of Section 5.5 to the above effect.

Nonetheless, this raises two question of practical concern:

1. how can we *select* sparse subsets uniformly in $\mathcal{X}$ and
2. how can we guarantee uniform *support*[43] in $\mathcal{Z}$?

The first question is for sparse kernelized RL algorithms (e.g., Engel et al., 2003; Xu, 2006) of utmost importance as the reported problem will most likely affect them as well. Future works must derive such an algorithm, maybe based on slowness or diffusion distances.

The related *radial basis function networks* (RBF, see, e.g., Haykin, 1998) have found an empirically answer to the second question: each support vector $s_i$ is assigned an individual kernel width $\sigma_i$ relative to the distance to its neighbors. For the Hilbert spaces of all kernels $\kappa(\cdot,\cdot)$ holds $\mathcal{H}_\kappa \subset L^2(\mathcal{Z},\xi)$ and one could thus perform inner products between Gaussian kernel functions of different width[44] in $L^2(\mathcal{Z},\xi)$. However, the math necessary to pose kernel SFA in this framework is quite advanced and calls for further research.

### 6.4 Visual Policies for Robots

Visual tasks are an interesting field of research as they expose elemental weaknesses in current RL approaches, for example the different Euclidean distances in observation and ideal approximation spaces discussed in this article. For applications in the field of robotics, however, the assumption of *isometry between observations and states* reaches its limits.

On the one hand, partial observability of the environment (POMDPs, Kaelbling et al., 1998) will jumble the observed transition structure and therefore all discussed feature construction methods.[45] This could be avoided by including partial *histories* of observations. For example, *predictive state representations* (PSR, Littman et al., 2001; Wingate, 2012) can construct sufficient statistics of

43. Areas in $\mathcal{Z}$ with less *support vectors* $s_i$ will have an overall lower output of kernel functions and will thus exhibit worse generalization. This effect can be quantized for sample $z \in \mathcal{Z}$ by the *approximation error* of the corresponding kernel function, that is, $\inf_{a \in \mathbb{R}^m} \|\kappa(\cdot,z) - \sum_i a_i \kappa(\cdot,s_i)\|_{\mathcal{H}}$, and is called the *support* of $z$.

44. Note that $\langle \kappa_a(\cdot,s_i), \kappa_b(\cdot,s_j)\rangle_\xi = \int \xi(dz)\, \kappa_a(z,s_i)\, \kappa_b(z,s_j)$ has an analytic solution if $\xi$ is the uniform distribution, because the product of two Gaussian functions is a Gaussian function as well.

45. Basis construction might work well in a POMDP if applied on *beliefs* instead of observations, though.

observation history to solve the task without extensive knowledge of the underlying POMDP. Extensions to continuous state and observation spaces are rare (Wingate and Singh, 2007) and linear approximation not straight forward. Additionally, any traditional metric over histories will probably not reflect *diffusion distances* very well and thus perform suboptimal in techniques like LSPI (see our conclusion in Section 5.6). Therefore, the potential of feature construction techniques like *optimal basis functions* (Section 4.3) for PSR appear tremendous and should be investigated further.

Setting partial observability aside, on the other hand, every natural variable influencing the image will be treated as part of state space $\mathcal{X}$, for example angle and brightness of illumination, non-stationary objects, the view out of the window, etc. The resulting state space $\mathcal{X}$ grows exponentially in the number of these independent variables, as every combination of variables is a unique state. Besides encoding mostly useless information, this yields two problems for the method presented in this article: (i) the whole state space $\mathcal{X}$ must be *sampled* by the RL agent, which will eventually take too much time, and (ii) the basis functions must *support* the whole space, for example, a subset for sparse kernel methods must uniformly cover $\mathcal{X}$ (see Sections 5.5 and 6.3), which will eventually require too many computational resources. Both problems can not be resolved by current kernel methods to construct basis functions and/or standard linear RL approaches. *Factored MDP* approaches in combination with *computer vision* methods have the potential to solve this dilemma, though.

Using an array of highly invariant image descriptors (e.g., SIFT, Lowe, 1999), object recognition and position estimates (e.g., SLAM, Smith et al., 1990; Davison, 2003), the observation space $\mathcal{Z}$ can become much more regular. Smart choices of descriptors, for example, reacting to the window frame and not the view outside, will even make them invariant to most state dimensions in $\mathcal{X}$. If the descriptors include short-term memory, then the presented method could even be applied in a POMDP setup. However, an application of standard kernel methods takes the similarity between all descriptors at once into account and would therefore still require sampling and support on the whole space $\mathcal{X}$. *Factorizing* basis functions $\phi_i(\cdot) = \prod_j \psi_{ij}(\cdot)$, on the other hand, have full domain $\mathcal{Z}$ but are a product of multiple functions $\psi_{ij}(\cdot)$ with a domain of only a few descriptors. Integrals over $\mathcal{Z}$ break down into the product of multiple low dimensional integrals, which would each only require limited amount of sampling and support. If those factorized basis functions approximate the *optimal basis functions* discussed in Section 4.3 sufficiently close, factored MDP algorithms can be applied (Koller and Parr, 1999; Guestrin et al., 2001; Hauskrecht and Kveton, 2003; Guestrin et al., 2004). Future works must develop both the factorizing basis function construction method and some adequate factored linear RL algorithm to exploit them.

## 7. Summary

This article investigates *approximation spaces* for value estimation, in particularly the role of the *metric* in these spaces. This is relevant because this metric influences the Euclidean $L_2$ approximation error minimized by *least-squares temporal difference learning* (LSTD). We hypothesize that an *ideal Euclidean metric for LSTD should encode diffusion distances*, which reflect similar futures analogous to values. Furthermore, *slow-feature analysis (SFA) constructs the best subspace-invariant approximation spaces for LSTD*. To verify these hypotheses we compare *Krylov bases*, *proto-value functions* (PVF), *principal component analysis* (PCA) and SFA (see Section 3) theoretically and experimentally. We also derive a novel *importance sampling* modification to the SFA

algorithm to compensate for sampling imbalances of SFA. The novel algorithm showed excellent performance in Section 5.

Our theoretical analysis in Section 4 compares Krylov bases with SFA. We argue that the latter is a generalization of PVF and can construct typical *subspace-invariant* approximation spaces. Our analysis yields impressive statements for MDPs which are actually subspace-invariant under PVF or SFA. For example, Corollary 9 (Page 2086) shows a dramatically improved bound on the LSTD approximation error and Theorem 11 (Page 2087) gives a lower bound on the improvement thereof by adding another feature. However, compatible MDPs are not very common: SFA features are subspace-invariant for all MDPs with a *self-adjoint* transition operator and PVF are for all MDPs with a transition kernel visiting all neighbors uniformly. Note that the latter set is a subset of former. We argue further that *real-valued* subspace-invariant features can only be obtained for MDPs with *self-adjoint* transition operators. Both SFA and PVF can thus be interpreted as self-adjoint approximations of arbitrary MDPs, as empirical results in Section 5.1 demonstrate. This interpretation is formally supported by Theorem 13 (Page 2089). It states that SFA minimizes a mean bound over all tasks in the *same environment*, which means an arbitrary but fixed transition kernel and all possible reward functions. However, all above results hold only for the sampling policy.

It is therefore an empirical question how the discussed approximation spaces will fare when *least-squares policy iteration* (LSPI) changes the policy. We ask in Section 5:

- *How well does LSTD estimate the value function of a given Markov chain?* We predicted in Section 4.4 and verified in Sections 5.1 to 5.3: "SFA can provide better approximation spaces for LSTD than PVF".
- *How good is the performance of policies learned by LSPI based on a random policy?* Our empirical conclusion of Sections 5.3 and 5.6 is "[Modified] SFA can construct better approximation spaces for LSPI than PVF".
- *How does this performance depend on the approximation space metric?* The connection between diffusion distance and approximation error suggested itself in Sections 4.1 and 4.2. We empirically verified in Section 5.6: "LSPI performance is facilitated by approximation spaces that encode diffusion distances of a uniform random policy" because "SFA essentially performs PCA based on diffusion distances".

We see both theoretical and empirical results as evidence supporting our hypotheses. There are still too many open questions to be certain, like the undesirable dependence on the sampling distribution and other issues discussed in Section 6. However, especially the good performance with LSPI inspires hope and calls for further research.

## Acknowledgments

We would like to thank Roland Vollgraf, Hannes Nickisch and Mathias Franzius for pointing us to SFA as a pre-processing method for video data. This work was funded by the *German science foundation* (DFG) within the SPP 1527 *autonomous learning*, the *German federal ministry of education and research* (grant 01GQ0850), the EPSRC grant #EP/H017402/1 (CARDyAL) and by the integrated graduate program on *human-centric communication* at Technische Universität Berlin.

## Appendix A. Proofs of Section 4

For an introduction into the terminology see Section 2. The equivalency sign $\equiv$ is used to indicate that two optimization problems are solved by the same function.

*Lemma 3 (repeated): Let $\xi$ denote the steady state distribution of ergodic transition kernel $P^\pi$, which has a self-adjoint transition operator $\hat{P}^\pi = (\hat{P}^\pi)^* : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$. The corresponding diffusion distance equals the Euclidean distance in the space spanned by $\psi_i^t(\cdot) := \lambda_i^t \phi_i(\cdot), \forall i \in \mathbb{N}$, where $\lambda_i \in \mathbb{R}$ and $\phi_i \in L^2(\mathcal{Z},\xi)$ are the eigenvalues and eigenfunctions of $\hat{P}^\pi$, that is*

$$d_t(x,y) = \|\psi^t(x) - \psi^t(y)\|_2\,, \quad \forall x,y \in \mathcal{Z}, \forall t \in \mathbb{N}\setminus\{0\}\,.$$

**Proof** The diffusion distance $d_t(x,y)$ between states $x$ and $y$ is defined as the mean squared difference of the probability distributions after $t$ steps (see Page 2084):

$$d_t(x,y) \quad := \quad \left\|\mu_x^t - \mu_y^t\right\|_\xi .$$

Under the formal restrictions mentioned of Assumption 1, Page 2083, $\mu_x^t \in L^2(\mathcal{Z},\xi), \forall t \in \mathbb{N}\setminus\{0\}$, and one can rewrite the inner product with arbitrary functions $f \in L^2(\mathcal{Z},\xi)$:

$$\langle \mu_x^t, f\rangle_\xi = \int \xi(dy)\left(\mu_x^t(y)\right) f(y) = \int (P^\pi)^t(dy|x)\, f(y) = (\hat{P}^\pi)^t[f](x)\,,$$

where $(\hat{P}^\pi)^t$ denotes $t$ successive applications of the transition operator $\hat{P}^\pi$ in $L^2(\mathcal{Z},\xi)$.

$$\begin{aligned} d_t^2(x,y) &= \langle \mu_x^t, \mu_x^t\rangle_\xi - 2\langle \mu_x^t, \mu_y^t\rangle_\xi + \langle \mu_y^t, \mu_y^t\rangle_\xi \\ \langle \mu_x^t, \mu_y^t\rangle_\xi &= \int (P^\pi)^t(dz|x)\,\mu_y^t(z) \quad = \quad (\hat{P}^\pi)^t\left[\mu_y^t\right](x)\,. \end{aligned}$$

$\hat{P}^\pi$ is specified to be self-adjoint and due to the *Hilbert-Schmidt theorem* (e.g., Theorem 4.2.23 in Davies, 2007) holds for eigenfunctions $\hat{P}^\pi[\phi_i](\cdot) = \lambda_i \phi_i(\cdot)$, and $\langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}, \forall i,j \in \mathbb{N}$:

$$\hat{P}^\pi[f](x) = \sum_{i=0}^{\infty} \langle f, \phi_i\rangle_\xi\, \lambda_i\, \phi_i(x)\,, \qquad \forall x \in \mathcal{Z}, \quad \forall f \in L^2(\mathcal{Z},\xi)\,.$$

Applying this $t$ times, we can write

$$\begin{aligned} (\hat{P}^\pi)^t[\mu_y^t](x) &= \sum_{i=0}^{\infty} \phi_i(x)\,\lambda_i^t\,\langle \mu_y^t, \phi_i\rangle_\xi \;=\; \sum_{i=0}^{\infty} \phi_i(x)\,\lambda_i^t\,(\hat{P}^\pi)^t[\phi_i](y) \\ &= \sum_{i,j=0}^{\infty} \phi_i(x)\,\lambda_i^t\,\lambda_j^t\,\phi_j(y)\,\underbrace{\langle \phi_i, \phi_j\rangle_\xi}_{\delta_{ij}} \;=\; \psi^t(x)^\top \psi^t(y)\,. \end{aligned}$$

Therefore the diffusion distance $d_t(x,y)$ can be written as

$$d_t^2(x,y) = \psi^t(x)^\top\psi^t(x) - 2\psi^t(x)^\top\psi^t(y) + \psi^t(y)^\top\psi^t(y) = \|\psi^t(x) - \psi^t(y)\|_2^2\,.$$

■

*Lemma 4: Let $P^{\pi}$ be an ergodic transition kernel in $\mathcal{Z}$ with steady state distribution $\xi$. The kernel induced by adjoint transition operator $(\hat{P}^{\pi})^*$ in $L^2(\mathcal{Z},\xi)$ is $\xi$-almost-everywhere an ergodic transition kernel with steady state distribution $\xi$.*

**Proof** We first show that for *any* linear operator $\hat{A} : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$ with *kernel* $A : \mathcal{Z} \times \mathcal{B}(\mathcal{Z}) \to \mathbb{R}^+$, that is, $\hat{A}[f](x) = \int A(dy|x)\, f(y)$, *$\xi$-almost-everywhere* ($\xi$-a.e.) in $\mathcal{Z}$ holds

$$\underbrace{\langle f, \hat{A}[\mathbf{1}]\rangle_\xi = \langle f, \mathbf{1}\rangle_\xi, \quad \forall f \in L^2(\mathcal{Z},\xi)}_{(i)} \qquad \Leftrightarrow \qquad \int \underbrace{A(dy|x) = 1, \quad \xi\text{-a.e.}}_{(ii)}$$

where $\mathbf{1}(x) = 1, \forall x \in \mathcal{Z}$, is the constant function in $L^2(\mathcal{Z},\xi)$.

Let's assume the induction $(i) \Rightarrow (ii)$ is *not* true, that is, $(i)$ is true, but there exists a Borel set with non-zero measure $\xi$ of states $x \in \mathcal{Z}$ that violate $(ii)$. This set can be split up in $\int A(dy|x) > 1, \forall x \in B_+ \in \mathcal{B}(\mathcal{Z})$, and $\int A(dy|x) < 1, \forall x \in B_- \in \mathcal{B}(\mathcal{Z})$, with $\xi(B_+ \cup B_-) > 0$. Let furthermore $f \in L^2(\mathcal{Z},\xi)$ be defined as

$$f(x) = \begin{cases} 1 & ,\text{if } x \in B_+ \\ -1 & ,\text{if } x \in B_- \\ 0 & ,\text{otherwise} \end{cases},$$

which must adhere to claim $(i)$:

$$\begin{array}{rcl} \langle f, \hat{A}[\mathbf{1}]\rangle_\xi & = & \int_{B_+} \xi(dx) \int A(dy|x) - \int_{B_-} \xi(dx) \int A(dy|x) \\ & > & \int_{B_+} \xi(dx) - \int_{B_-} \xi(dx) \quad = \quad \langle f, \mathbf{1}\rangle_\xi . \end{array}$$

This is a contradiction and proves $(i) \Rightarrow (ii)$. The induction $(i) \Leftarrow (ii)$ is trivial, which proves $(i) \Leftrightarrow (ii)$.

Now we show that $(i)$ holds for $(\hat{P}^{\pi})^*$, which is the adjoint operator to $\hat{P}^{\pi}$.

$$\langle f, (\hat{P}^{\pi})^*[\mathbf{1}]\rangle_\xi = \langle \hat{P}^{\pi}[f], \mathbf{1}\rangle_\xi = \int \underbrace{\int \xi(dx)\, P^{\pi}(dy|x)}_{\xi(dy)\text{ (ergodicity)}} f(y) = \langle f, \mathbf{1}\rangle_\xi, \quad \forall f \in L^2(\mathcal{Z},\xi).$$

This proves that the kernel of $(\hat{P}^{\pi})^*$ is a *transition kernel* $\xi$-almost-everywhere in $\mathcal{Z}$, which means the kernel adheres to $(ii)$. Ergodicity can be proven using the same techniques as above:

$$\langle \mathbf{1}, \hat{A}[f]\rangle_\xi = \langle \mathbf{1}, f\rangle_\xi, \quad \forall f \in L^2(\mathcal{X},\xi) \qquad \Leftrightarrow \qquad \xi(B) = \int A(B|x)\, \xi(dx), \quad \forall B \in \mathcal{B}(\mathcal{X}).$$

$\langle \mathbf{1}, (\hat{P}^{\pi})^*[f]\rangle_\xi = \langle \mathbf{1}, f\rangle_\xi, \forall f \in L^2(\mathcal{X},\xi)$, and therefore the transition kernel of $(\hat{P}^{\pi})^*$ is ergodic with steady state distribution $\xi$. ■

*Lemma 5: In the limit of an infinite ergodic Markov chain drawn by transition kernel $P^{\pi}$ in $\mathcal{Z}$ with steady state distribution $\xi$ holds $\mathcal{S}(f) = 2\langle f, (\hat{I} - \hat{P}^{\pi})[f]\rangle_\xi, \forall f \in L^2(\mathcal{Z},\xi)$.*

**Proof** Due to a theorem of Jensen and Rahbek (2007) we can ensure that the *empirical mean* of functions over *sequences* of states converges in the limit to its expectation:

$$\begin{array}{rcl} \mathcal{S}(f) & = & \lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} \Big(f(x_{t+1}) - f(x_t)\Big)^2 = \iint \Big(f(y) - f(x)\Big)^2 P^{\pi}(dy|x)\, \xi(dx) \\ & = & \langle f, f\rangle_\xi - 2\langle f, \hat{P}^{\pi}[f]\rangle_\xi + \int f^2(y) \underbrace{\int P^{\pi}(dy|x)\, \xi(dx)}_{\xi(dy)\text{ due to ergodicity}} = 2\langle f, (\hat{I} - \hat{P}^{\pi})[f]\rangle_\xi . \end{array}$$

■

*Theorem 7: On average over all reward functions $r^\pi : \mathcal{Z} \to \mathbb{R}$ drawn from a white noise functional $\rho$, the squared norm of a Krylov basis $\{\phi_i^K\}_{i=1}^p$ from an ergodic transition kernel $P^\pi$ encodes squared diffusion distances based on $\hat{P}^\pi$ up to horizon $p-1$, that is*

$$d_t^2(x,y) \;=\; \mathbb{E}\left[\left\|\phi^K(x)-\phi^K(y)\right\|_{\varrho}^2 \middle| r^\pi \sim \rho\right], \quad \forall x,y \in \mathcal{Z},\, \exists \varrho \in (\mathbb{R}^+)^p,\, \forall t \in \{1,\dots,p-1\}.$$

**Proof** Let $\xi$ denote the steady state distribution of transition kernel $P^\pi$. Given a reward function $r^\pi \in L^2(\mathcal{Z},\xi)$, a Krylov feature can be posed in terms of functions $\mu_x^t \in L^2(\mathcal{Z},\xi)$ (see Definition 1, Page 2083, and the Proof of Lemma 3), that is

$$\phi_i^K(x) \;:=\; (\hat{P}^\pi)^{i-1}[r^\pi](x) \;=\; \int (P^\pi)^{i-1}(dy|x)\, r^\pi(y) \;=\; \langle \mu_x^{i-1}, r^\pi\rangle_\xi, \qquad \forall x \in \mathcal{Z}.$$

One can use a property of white noise functionals (Footnote 25, p. 2085) to prove the theorem.

$$\begin{aligned}\mathbb{E}\left[\left\|\phi^K(x)-\phi^K(y)\right\|_{\varrho}^2\right] &= \sum_{i=1}^p \varrho_i\, \mathbb{E}\left[\left(\langle \mu_x^{i-1}-\mu_y^{i-1}, r^\pi\rangle_\xi\right)^2\right] \\ &= \sum_{i=1}^p \varrho_i \left\|\mu_x^{i-1}-\mu_y^{i-1}\right\|_\xi^2 \;=\; \sum_{i=1}^p \varrho_i\, d_{i-1}^2(x,y).\end{aligned}$$

$\varrho \in (\mathbb{R}^+)^p$ can be chosen freely; all diffusion distances with $t < p$ are therefore encoded. ■

*Lemma 8: Let $\{\phi_i\}_{i=1}^p$ denote any $p$ SFA features from a MDP with self-adjoint transition operator, then the LSTD fixed point $f^\pi = \hat{\Pi}_\xi^\phi\left[\hat{B}^\pi[f^\pi]\right]$ and the projection of true value function $v^\pi = \hat{B}^\pi[v^\pi]$ coincide, that is*

$$f^\pi(x) \;=\; \hat{\Pi}_\xi^\phi[v^\pi](x) \;=\; \sum_{i=1}^p \langle r^\pi, \phi_i\rangle_\xi\, \tau_i\, \phi_i(x), \quad \forall x \in \mathcal{Z}, \quad \tau_i := \left(1-\gamma+\tfrac{\gamma}{2}\mathcal{S}(\phi_i)\right)^{-1}.$$

**Proof** Let $\psi_i \in L^2(\mathcal{Z},\xi)$ denote the (due to the Hilbert-Schmidt theorem orthonormal) eigenfunctions of $\hat{P}^\pi$ and $\lambda_i$ the corresponding eigenvalues, that is, $\hat{P}^\pi[\psi_i] = \lambda_i \psi_i$. $\{\psi_i\}_{i=1}^\infty$ is a full basis of $L^2(\mathcal{Z},\xi)$ and thus $r^\pi = \sum_{i=1}^\infty \langle r^\pi, \psi_i\rangle_\xi \psi_i$ with $\phi_i = \psi_i, \forall i \le p$ and $\langle \phi_i, \psi_j\rangle_\xi = \delta_{ij}$. From this we can conclude $\hat{\Pi}_\xi^\phi[\psi_i] = \phi_i, \forall i \le p$, and $\hat{\Pi}_\xi^\phi[\psi_i] = 0, \forall i > p$. Due to the geometric series and Lemma 5 also holds $\sum_{t=0}^\infty \gamma^t \lambda_i^t = (1-\gamma\lambda_i)^{-1} = \tau_i$. Therefore,

$$\begin{aligned}\hat{\Pi}_\xi^\phi[v^\pi] &= \sum_{t=0}^\infty \gamma^t\, \hat{\Pi}_\xi^\phi\left[(\hat{P}^\pi)^t[r^\pi]\right] \;=\; \sum_{i=1}^\infty \langle r^\pi, \psi_i\rangle_\xi \sum_{t=0}^\infty \gamma^t\, \hat{\Pi}_\xi^\phi\left[(\hat{P}^\pi)^t[\psi_i]\right] \\ &= \sum_{i=1}^p \langle r^\pi, \phi_i\rangle_\xi\, \phi_i \sum_{t=0}^\infty \gamma^t \lambda_i^t \;=\; \sum_{i=1}^p \langle r^\pi, \phi_i\rangle_\xi\, \tau_i\, \phi_i, \\ f^\pi &= \hat{\Pi}_\xi^\phi\left[\hat{B}^\pi[f^\pi]\right] \;=\; \sum_{t=0}^\infty \gamma^t \left(\hat{\Pi}_\xi^\phi[\hat{P}^\pi]\right)^t \left[\hat{\Pi}_\xi^\phi[r^\pi]\right] \;=\; \sum_{i=1}^p \langle r^\pi, \phi_i\rangle_\xi\, \phi_i \sum_{t=0}^\infty \gamma^t \lambda_i^t.\end{aligned}$$

■

*Theorem 11: Let $\xi$ be the steady state distribution on $\mathcal{Z}$ of a MDP with policy $\pi$ and a self-adjoint transition operator in $L^2(\mathcal{Z},\xi)$. Let further $\Phi_p = \{\phi_i\}_{i=1}^p$ be any set of p SFA features and $v^\pi \in L^2(\mathcal{Z},\xi)$ the true value of the above MDP. The improvement of the LSTD solution $f^{(p)} := \hat{\Pi}_\xi^{\Phi_p}\left[\hat{B}^\pi[f^{(p)}]\right]$ by including the p'th feature is bounded from below by*

$$\left\|v^\pi - f^{(p-1)}\right\|_\xi - \left\|v^\pi - f^{(p)}\right\|_\xi \quad \geq \quad \tfrac{1-\gamma}{2}\,\frac{\langle r^\pi,\phi_p\rangle_\xi^2}{\|r^\pi\|_\xi}\,\tau_p^2\,, \qquad \tau_p := \left(1-\gamma+\tfrac{\gamma}{2}\mathcal{S}(\phi_p)\right)^{-1}.$$

**Proof** Let $\{\phi_i\}_{i=1}^\infty$ denote the extension of $\Phi_p$ to a full orthonormal basis of $L^2(\mathcal{Z},\xi)$, that is, $f(\cdot) = \sum_{i=1}^\infty \langle f,\phi_i\rangle_\xi\,\phi_i(\cdot)$, $\forall f \in L^2(\mathcal{Z},\xi)$. Lemma 8 shows that $f^{(p)} = \hat{\Pi}_\xi^{\Phi_p}[v^\pi], \forall p \in \mathbb{N}$,

$$\left\|v^\pi - \hat{\Pi}_\xi^{\Phi_{p-1}}[v^\pi]\right\|_\xi^2 - \left\|v^\pi - \hat{\Pi}_\xi^{\Phi_p}[v^\pi]\right\|_\xi^2 = \left\|\sum_{i=p}^\infty \langle v^\pi,\phi_i\rangle_\xi\phi_i\right\|_\xi^2 - \left\|\sum_{i=p+1}^\infty \langle v^\pi,\phi_i\rangle_\xi\phi_i\right\|_\xi^2 = \langle v^\pi,\phi_p\rangle_\xi^2.$$

Note further that $\langle v^\pi,\phi_p\rangle_\xi^2 = \langle r^\pi,\phi_p\rangle_\xi^2\,\tau_p^2$, and that one can bound the norm of $v^\pi$:

$$\left\|v^\pi\right\|_\xi \;=\; \left\|\sum_{t=0}^\infty \gamma^t\,(\hat{P}^\pi)^t[r^\pi]\right\|_\xi \;\leq\; \sum_{t=0}^\infty \gamma^t\left\|(\hat{P}^\pi)^t[r^\pi]\right\|_\xi \;\leq\; \sum_{t=0}^\infty \gamma^t\left\|r^\pi\right\|_\xi \;=\; \frac{1}{1-\gamma}\left\|r^\pi\right\|_\xi.$$

The first equality follows from the proof of Lemma 14, the first inequality from the property of norms, the last inequality from Lemma 15 and the last equality from the geometric series. Using the identity $a^2-b^2=(a-b)(a+b)$ and inequality $\|v^\pi - \hat{\Pi}_\xi^{\Phi_p}[v^\pi]\|_\xi \leq \|v^\pi\|_\xi$,

$$\left\|v^\pi - f\right\|_\xi - \left\|v^\pi - f'\right\|_\xi \;=\; \frac{\left\|v^\pi - f\right\|_\xi^2 - \left\|v^\pi - f'\right\|_\xi^2}{\left\|v^\pi - f\right\|_\xi + \left\|v^\pi - f'\right\|_\xi} \;\geq\; \frac{\langle v^\pi,\phi_p\rangle_\xi^2}{2\,\|v^\pi\|_\xi} \;\geq\; \tfrac{1-\gamma}{2}\,\frac{\langle r^\pi,\phi_p\rangle_\xi^2}{\|r^\pi\|_\xi}\,\tau_p^2\,,$$

where $f := f^{(p-1)}$ and $f' := f^{(p)}$. ■

*Theorem 13: For any infinite ergodic Markov chain with steady state distribution $\xi$ over state space $\mathcal{Z}$, SFA selects features from function set $\mathcal{F} \subset L^2(\mathcal{Z},\xi)$ that minimize an upper bound on the optimality criterion of Definition 12 for sampling policy $\pi$ and discount factor $\gamma > 0$, under the assumption that the mean-reward functions $r^\pi : \mathcal{Z} \to \mathbb{R}$ are drawn from a white noise functional in $L^2(\mathcal{Z},\xi)$.*

**Proof** Lemma 14 shows that for all $0 \leq \gamma < 1$ the operator $(\hat{I} - \gamma\hat{P}^\pi) : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$ is invertible. Let $\hat{\Theta}^\pi$ denote this inverse operator. For any mean reward function $r^\pi \in L^2(\mathcal{Z},\xi)$ the corresponding true value function $v_r^\pi \in L^2(\mathcal{Z},\xi)$ can be determined analytically: $v_r^\pi(x) = r^\pi(x) + \gamma\hat{P}^\pi[v_r^\pi](x) = \hat{\Theta}^\pi[r^\pi](x), \forall x \in \mathcal{Z}$. According to the assumptions, the mean reward functions $r^\pi \sim \rho$ are distributed as a white noise functional, which implies

$$\int \langle f, r^\pi\rangle_\xi^2\,\rho(dr^\pi) \;=\; \langle f,f\rangle_\xi\,, \quad \forall f \in L^2(\mathcal{Z},\xi)\,.$$

We will now show that the SFA objective minimizes an upper bound on Definition 12 and thus also minimizes the bound of Tsitsiklis and Van Roy (1997).

$$
\begin{aligned}
& \inf_{\phi\in(\mathcal{F})^p} \mathbb{E}\left[\left\|v_r^\pi - \hat{\Pi}_\xi^\phi[v_r^\pi]\right\|_\xi^2 \,\middle|\, r^\pi \sim \rho\right] \\
\equiv\ & \sup_{\phi\in(\mathcal{F})^p} \mathbb{E}\left[\langle v_r^\pi, \hat{\Pi}_\xi^\phi[v_r^\pi]\rangle_\xi \,\middle|\, r^\pi \sim \rho\right] \\
\equiv\ & \sup_{\phi\in(\mathcal{F})^p} \sum_{i=1}^p \mathbb{E}\left[\langle \phi_i, \hat{\Theta}^\pi[r^\pi]\rangle_\xi^2 \,\middle|\, r^\pi \sim \rho\right] && \text{s.t. } \langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}\,,\ \forall i,j && (C_{ij}{:=}\delta_{ij}) \\
\equiv\ & \sup_{\phi\in(\mathcal{F})^p} \sum_{i=1}^p \left\|(\hat{\Theta}^\pi)^*[\phi_i]\right\|_\xi^2 && \text{s.t. } \langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}\,,\ \forall i,j && \text{(assumption)} \\
\equiv\ & \inf_{\phi\in(\mathcal{F})^p} \sum_{i=1}^p \left\|\left(\hat{I} - \gamma(\hat{P}^\pi)^*\right)[\phi_i]\right\|_\xi^2 && \text{s.t. } \langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}\,,\ \forall i,j && \text{(lemma 16)} \\
\leq\ & \inf_{\phi\in(\mathcal{F})^p} \sum_{i=1}^p \left((1+\gamma^2)\underbrace{\langle \phi_i, \phi_i\rangle_\xi}_{1} - 2\gamma\langle \phi_i, \hat{P}^\pi[\phi_i]\rangle_\xi\right) && \text{s.t. } \langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}\,,\ \forall i,j && \text{(lemmas 15\&4)} \\
\equiv\ & \inf_{\phi\in(\mathcal{F})^p} -2\gamma \sum_{i=1}^p \langle \phi_i, \hat{P}^\pi[\phi_i]\rangle_\xi && \text{s.t. } \langle \phi_i, \phi_j\rangle_\xi = \delta_{ij}\,,\ \forall i,j \\
\overset{(*)}{\equiv}\ & \inf_{\phi\in(\mathcal{F})^p} 2\sum_{i=1}^p \langle \phi_i, (\hat{I} - \hat{P}^\pi)[\phi_i]\rangle_\xi && \text{s.t. } \begin{array}{rcll} \langle \phi_i, \phi_j\rangle_\xi &=& \delta_{ij}\,, & \forall i,j \\ \langle \phi_i, 1\rangle_\xi &=& 0\,, & \forall i \end{array} && (\gamma{>}0) \\
\equiv\ & \inf_{\phi\in(\mathcal{F})^p} \sum_{i=1}^p \mathcal{S}(\phi_i) && \text{s.t. } \begin{array}{rcll} \langle \phi_i, \phi_j\rangle_\xi &=& \delta_{ij}\,, & \forall i,j \\ \langle \phi_i, 1\rangle_\xi &=& 0\,, & \forall i \end{array} && \text{(Lemma 5).}
\end{aligned}
$$

The equivalency marked $(*)$ holds because the infimum is the same for all $\gamma > 0$. In the limit $\gamma \to 1$, however, $(\hat{I} - \gamma\hat{P}^\pi)$ is not invertible. The fist (constant) right eigenfunction of $\hat{P}^\pi$ must thus be excluded by the zero mean constraint. The last equation is the SFA optimization problem, which therefore minimizes an upper bound on the optimality criterion of Definition 12. ■

**Lemma 14** *For an ergodic transition operator $\hat{P}^\pi$ in $L^2(\mathcal{Z},\xi)$ with steady state distribution $\xi$ and $0 \leq \gamma < 1$, the operator $(\hat{I} - \gamma\hat{P}^\pi)$ is invertible. Let $\hat{\Theta}^\pi$ denote the inverse,*

$$\left\|(\hat{I} - \gamma\hat{P}^\pi)\left[\hat{\Theta}^\pi[f]\right] - f\right\|_\xi = 0\,, \quad \forall f \in L^2(\mathcal{Z},\xi)\,.$$

**Proof** Let $(\hat{P}^\pi)^t$ denote the composition of $t$ operators $\hat{P}^\pi$ with $(\hat{P}^\pi)^0 = \hat{I}$ and let $\hat{\Theta}^\pi := \lim_{n\to\infty} \sum_{t=0}^{n-1} \gamma^t (\hat{P}^\pi)^t$, then $\forall f \in L^2(\mathcal{Z},\xi)$:

$$
\begin{aligned}
\left\|\left(\hat{I} - \gamma\hat{P}^\pi\right)\left[\hat{\Theta}^\pi[f]\right] - f\right\|_\xi &= \lim_{n\to\infty}\left\|\left(\hat{I} - \gamma\hat{P}^\pi\right)\left[\sum_{t=0}^{n-1}\gamma^t(\hat{P}^\pi)^t[f]\right] - f\right\|_\xi \\
= \lim_{n\to\infty}\left\|\left(\hat{I} - \gamma^n(\hat{P}^\pi)^n\right)[f] - f\right\|_\xi = \lim_{n\to\infty}\gamma^n\left\|(\hat{P}^\pi)^n[f]\right\|_\xi &\leq \lim_{n\to\infty}\gamma^n\|f\|_\xi = 0\,.
\end{aligned}
$$

The inequality holds because Lemma 15 shows that $\hat{P}^\pi$ is a non-expansion and the last equality because all $f \in L^2(\mathcal{Z},\xi)$ are bounded from above, that is, $\|f\|_\xi < \infty$. ■

**Lemma 15** *An ergodic transition operator $\hat{P}^{\pi}$ in $L^2(\mathcal{Z},\xi)$ with steady state distribution $\xi$ is a non-expansion, defined as*

$$\left\|\hat{P}^{\pi}[f]\right\|_{\xi} \leq \left\|f\right\|_{\xi}, \qquad \forall f \in L^2(\mathcal{Z},\xi).$$

**Proof** Due to Jensens inequality[46] we have

$$\left\|\hat{P}^{\pi}[f]\right\|_{\xi}^2 = \langle \hat{P}^{\pi}[f], \hat{P}^{\pi}[f]\rangle_{\xi} = \int \xi(dx)\Big(\int P^{\pi}(dy|x)f(y)\Big)^2$$
$$\leq \int \underbrace{\int \xi(dx)P^{\pi}(dy|x)}_{\xi(dy) \text{ due to ergodicity}} f^2(y) = \langle f,f\rangle_{\xi} = \left\|f\right\|_{\xi}^2, \quad \forall f \in L^2(\mathcal{Z},\xi).$$

■

**Lemma 16** *For any invertible linear operator $\hat{A} : L^2(\mathcal{Z},\xi) \to L^2(\mathcal{Z},\xi)$ holds*

$$\sup_{\substack{f\in L^2(\mathcal{Z},\xi),\\ \langle f,f\rangle_{\xi}=1}} \left\|\hat{A}[f]\right\|_{\xi} \equiv \inf_{\substack{f\in L^2(\mathcal{Z},\xi),\\ \langle f,f\rangle_{\xi}=1}} \left\|\hat{A}^{-1}[f]\right\|_{\xi}.$$

**Proof** The *operator norm* of an operator $\hat{A}$ in $L^2(\mathcal{Z},\xi)$ is defined as

$$\|\hat{A}\|_{\xi} := \sup_{f\in L^2(\mathcal{Z},\xi)} \left\{ \frac{\|\hat{A}[f]\|_{\xi}}{\|f\|_{\xi}} \,\middle|\, f \neq 0 \right\} = \sup_{f\in L^2(\mathcal{Z},\xi)} \left\{ \|\hat{A}[f]\|_{\xi} \,\middle|\, \|f\|_{\xi} = 1 \right\}.$$

Using the one-to-one transformation $f \leftarrow \hat{A}^{-1}[f]$ in the equivalency marked $(*)$,

$$\sup_{\substack{f\in L^2(\mathcal{Z},\xi),\\ \langle f,f\rangle_{\xi}=1}} \left\|\hat{A}[f]\right\|_{\xi} \equiv \|\hat{A}\|_{\xi} \equiv \sup_{f\in L^2(\mathcal{Z},\xi)} \left\{ \tfrac{\|\hat{A}[f]\|_{\xi}}{\|f\|_{\xi}} \,\middle|\, f \neq 0 \right\} \equiv \inf_{f\in L^2(\mathcal{Z},\xi)} \left\{ \tfrac{\|f\|_{\xi}}{\|\hat{A}[f]\|_{\xi}} \,\middle|\, f \neq 0 \right\}$$
$$\overset{(*)}{\equiv} \inf_{f\in L^2(\mathcal{Z},\xi)} \left\{ \tfrac{\|\hat{A}^{-1}[f]\|_{\xi}}{\|f\|_{\xi}} \,\middle|\, f \neq 0 \right\} \equiv \inf_{\substack{f\in L^2(\mathcal{Z},\xi),\\ \langle f,f\rangle_{\xi}=1}} \left\|\hat{A}^{-1}[f]\right\|_{\xi}.$$

■

## References

P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in Neural Information Processing Systems*, pages 1–8, 2007.

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

46. If $P^{\pi}$ is a transition kernel, Jensens inequality (e.g., Boyd and Vandenberghe, 2004) allows $\left(\int P^{\pi}(dy|x)f(y)\right)^2 \leq \int P^{\pi}(dy|x)f^2(y), \forall x \in \mathcal{Z}, \forall f \in L^2(\mathcal{Z},\xi)$.

P. Berkes and L. Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5:579–602, 2005.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition, 2007.

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

W. Böhmer, S. Grünewälder, H. Nickisch, and K. Obermayer. Generating feature spaces for linear algorithms with regularized sparse kernel slow feature analysis. *Machine Learning*, 89(1-2): 67–86, 2012.

M. Bowling, A. Ghodsi, and D. Wilkinson. Action respecting embedding. In *International Conference on Machine learning*, 2005.

J. A. Boyan and A. W. Moore. Generalization in reinforcement learning: safely approximating the value function. In *Advances in Neural Information Processing Systems*, pages 369–376, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1/2/3):33–57, 1996.

R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data. Part I: diffusion maps. *Proceedings of the National Academy of Science*, 102(21):7426 – 7431, May 2005.

L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

E. Brian Davies. *Linear Operators and their Spectra*. Cambridge University Press, 2007.

A. J. Davison. Real-time simultaneous localization and mapping with a single camera. In *IEEE International Conference on Computer Vision*, volume 2, page 1403, 2003.

D.P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: the Gaussian process approach to temporal difference learning. In *International Conference on Machine Learning*, pages 154–161, 2003.

K. Ferguson and S. Mahadevan. Proto-transfer learning in Markov decision processes using spectral methods. In *ICML Workshop on Transfer Learning*, 2006.

E. Ferrante, A. Lazaric, and M. Restelli. Transfer of task representation in reinforcement learning using policy-based proto-value functions. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and sparseness leads to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):e166, 2007.

S. Grünewälder and K. Obermayer. The optimal unbiased value estimator and its relation to LSTD, TD and MC. *Machine Learning*, 83:289–330, 2011.

C. Guestrin, D. Koller, and R. Parr. Max-norm projections for factored MDPs. In *International Joint Conference on Artificial Intelligence*, pages 673–682, 2001.

C. Guestrin, M. Hauskrecht, and B. Kveton. Solving factored MDPs with continuous and discrete variables. In *Uncertainty in Artificial Intelligence*, pages 235–242, 2004.

M. Hauskrecht and B. Kveton. Linear program approximations for factored continuous-state Markov decision processes. In *Advances in Neural Information Processing Systems*, pages 895–902, 2003.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1998. ISBN 978-0132733502.

G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

H. Holden, B. Øksendal, J. Ubøe, and T. Zhang. *Stochastic Partial Differential Equations*. Springer Science+Business Media, 2nd edition, 2010.

O. C. Jenkins and M. J. Mataric. A spatio-temporal extension to Isomap nonlinear dimension reduction. In *International Conference on Machine Learning*, 2004.

S. T. Jensen and A. Rahbek. On the law of large numbers for (geometrically) ergodic Markov chains. *Economic Theory*, 23:761–766, 2007.

S. Jodogne and J. H. Piater. Closed-loop learning of visual control policies. *Journal of Artificial Intelligence Research*, 28:349–391, 2007.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

J. Kober and J. Peters. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems*, 2009.

D. Koller and R. Parr. Computing factored value functions for policies in structured MDPs. In *International Joint Conference on Artificial Intelligence*, pages 1332–1339, 1999.

D. Koller and R. Parr. Policy iteration for factored MDPs. In *Uncertainty in Artificial Intelligence*, pages 326–334, 2000.

V. R. Kompella, M. D. Luciw, and J. Schmidhuber. Incremental slow feature analysis: adaptive low-complexity slow feature updating from high-dimensional input streams. *Neural Computation*, 24 (11):2994–3024, 2012.

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

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

S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *International Joint Conference on Neural Networks*, pages 1–8, 2010.

R. Legenstein, N. Wilbert, and L. Wiskott. Reinforcement learning on slow features of high-dimensional input streams. *PLoS Computational Biology*, 6(8):e1000894, 2010.

M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *In Advances In Neural Information Processing Systems 14*, 2001.

D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 1999.

M. Luciw and J. Schmidhuber. Low complexity proto-value function learning from sensory observations with incremental slow feature analysis. In *International Conference on Artificial Neural Networks and Machine Learning*, volume III, pages 279–287. Springer-Verlag, 2012.

S. Mahadevan and B. Liu. Basis construction from power series expansions of value functions. In *Advances in Neutral Information Processing Systems*, pages 1540–1548, 2010.

S. Mahadevan and M. Maggioni. Proto-value functions: a Laplacian framework for learning representations and control in Markov decision processes. *Journal of Machine Learning Research*, 8: 2169–2231, 2007.

S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions On Signal Processing*, 41:3397–3415, 1993.

N. Mehta, S. Natarajan, P. Tadepalli, and A. Fern. Transfer in variable-reward hierarchical reinforcement learning. *Machine Learning*, 73:289–312, 2008.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value-function approximation. In *International Conference on Machine Learning*, 2007.

R. Parr, L. Li, G. Taylor, C. Painter-Wakefiled, and M. L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *International Conference on Machine Learning*, 2008.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559 – 572, 1901.

M. Petrik. An analysis of Laplacian methods for value function approximation in MDPs. In *International Joint Conference on Artificial Intelligence*, pages 2574–2579, 2007.

M. Petrik and S. Zilberstein. Robust approximate bilinear programming for value function approximation. *Journal of Machine Learning Research*, 12:3027–3063, 2011.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

M. Reed and B. Simon. *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, 1980. ISBN 0-12-585050-6.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002. ISBN 978-0262194754.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

R. Smith, M. Slef, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings to the 17th International Conference Machine Learning*, pages 911–918, 2000.

M. Snel and S. Whiteson. Multi-task reinforcement learning: Shaping and feature selection. In *European Workshop on Reinforcement Learning*, pages 237–248, 2011.

N. Sprague. Predictive projections. In *International Joint Conference on Artificial Intelligence*, pages 1223–1229, 2009.

H. Sprekeler. On the relationship of slow feature analysis and Laplacian eigenmaps. *Neural Computation*, 23(12):3287–3302, 2011.

Y. Sun, F. Gomez, M. Ring, and J. Schmidhuber. Incremental basis construction from temporal difference error. In *International Conference on Machine Learning*, pages 481–488, 2011.

R. S. Sutton. Generalization in reinforcement learning: successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, pages 1038–1044, 1996.

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

M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global framework for nonlinear dimensionality reduction. *Science*, 290:2319 – 2323, 2000.

S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, 1993.

J.N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.

C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

D. Wingate. Predictively defined representations of state. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, pages 415–439. Springer-Verlag Berlin Heidelberg, 2012.

D. Wingate and S. P. Singh. On discovery and learning of models with predictive representations of state for agents with continuous actions and observations. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1128–1135, 2007.

L. Wiskott. Slow feature analysis: a theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

L. Wiskott and T. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

X. Xu. A sparse kernel-based least-squares temporal difference algorithm for reinforcement learning. In *Advances in Natural Computation*, volume 4221 of *Lecture Notes in Computer Science*, pages 47–56. Springer Berlin / Heidelberg, 2006.