Convergence Analysis of Kernel-based On-policy Approximate Policy Iteration Algorithms for Markov Decision Processes with Continuous, Multidimensional States and Actions

Jun Ma and Warren B. Powell
Department of Operations Research and Financial Engineering
Princeton University, Princeton, NJ 08544

May 11, 2010

Abstract

Using kernel smoothing techniques, we propose three different online, on-policy approximate policy iteration algorithms which can be applied to infinite horizon problems with continuous and vector-valued states and actions. Using Monte Carlo sampling to estimate the value function around the post-decision state, we reduce the problem to a sequence of deterministic, nonlinear programming problems that allow us to handle continuous, vector-valued states and actions. We provide a formal convergence analysis of the algorithms under a variety of technical assumptions.

1 Introduction

We consider the challenge of finding implementable, provably convergent algorithms for infinite horizon Markov decision processes with continuous, multidimensional states and actions. Extensive research has been conducted for convergent algorithms using linear function approximation with features or basis functions (see for example Tsitsiklis & Van Roy (1996), Bradtke & Barto (1996), Tsitsiklis & Van Roy (1997), Lagoudakis & Parr (2003), and Ma & Powell (2010)). More recently, kernel approaches for approximating nonlinear or nonparametric value functions have been drawing interest from the machine learning community, since kernel-based algorithms offer the potential of producing accurate approximations without requiring the heuristic identification of basis functions.

The objective of this paper is to prove convergence for computationally implementable algorithms which solve MDP's with continuous and multidimensional states and actions. More specifically, we propose three different implementable online, on-policy approximate policy iteration algorithms that use kernel methods to handle infinite-horizon discounted MDPs where state, action and information variables are all continuous vectors and the expectation cannot be computed exactly. We use an on-policy algorithm to avoid the need to introduce action sampling strategies, which are problematic when actions are multidimensional. The algorithms work around value functions that depend on the post-decision state instead of Q-factors that depend on states and actions, which is particularly problematic when actions are multidimensional. We then use least squares or recursive least squares methods for policy evaluation. We also provide a rigorous convergence analysis of the algorithms under a variety of technical assumptions. The first two algorithms extend algorithms that have been presented previously, so we begin with a review of the literature.

Xu et al. (2007) extends least squares policy iteration (LSPI) in Lagoudakis & Parr (2003) to a kernel algorithm (namely KLSPI) by using a kernelized version of LSTD-Q learning algorithm (KLSTD-Q) to approximate state-action value functions. In the policy evaluation of KLSPI, least squares TD learning is implemented in a kernel-induced linear feature space so that linear LSTD algorithms can be applied while having a nonlinear value-

function representation in the original space. The main concept behind the algorithm is to convert a nonlinear learning algorithm into a linear algorithm by applying Mercer's theorem (details are given in section 3). We extend the algorithm to problems with continuous state and action spaces by working around post-decision state variables. The convergence analysis in Xu et al. (2007) is done for finite state space. In addition, there is a gap in their proof of their theorem 3.1 (p. 981), where they apply two lemmas (lemmas 3.2 and 3.3 in their paper). These two lemma are inconsistent because the error terms are under different norms: the L_2 norm and the infinity norm respectively. We provide a more rigorous convergence analysis of the algorithm using the approximate policy iteration convergence result developed in Ma & Powell (2010). Furthermore, in order to reduce the computational and storage costs of kernel methods, we propose to integrate a variety of kernel sparsification approaches in existing literature into the algorithm.

Ormoneit & Sen (2002) presents an off-line, off-policy kernel-based value iteration algorithm to approximate Markov decision problems with continuous state and finite action spaces using discretization and kernel average. Following a similar idea, we apply kernel smoothing to policy evaluation in our second on-line on-policy approximate policy iteration algorithm for continuous action space and provide a rigorous convergence analysis of the algorithm. The third algorithm directly applies kernel smoothing for value function approximation using finite horizon rewards for infinite horizon problems. As a result, a rich set of kernel estimators (recursive or non-recursive) such as Nadaraya-Watson estimators of the regression function and and local polynomial estimators can be integrated into the algorithm with sound convergence properties.

The rest of the paper is organized as follows. Section 2 presents several mathematical preliminary concepts that are crucial for convergence analysis in later sections. Section 3 analyzes the kernel based recursive least squares approximate policy iteration (KRLSAPI) algorithm around post-decision states, which is a kernelized version of the RLSAPI algorithm studied in Ma & Powell (2010). In section 4, we study algorithms that apply kernel smoothing to approximate value function and their convergence properties. Section 5 presents an algorithm that directly applies kernel regression or local polynomial regression with finite

horizon reward approximation. The last section concludes and discusses future research directions.

2 Mathematical preliminaries

We consider a class of infinite horizon Markov decision processes with continuous state and action spaces. The following subsections discuss several important preliminary concepts including Markov decision processes, contraction operators, continuous-state Markov chain, post-decision state variable and policy iteration algorithms. These basics are necessary for the convergence analysis in later sections.

2.1 Markov decision process and Bellman operators

A Markov decision process (MDP) is a sequential stochastic optimization problem for finding a policy that maximizes (for our problem class) the expected infinite discounted reward. The important elements of an MDP include system states, decision/control, stochastic exogenous information, state transition function and transition probability function, contributions/rewards and objective function.

Let x_t denote the state of the system (vector valued) at time t, u_t be a vector-valued continuous decision (control), $\pi : \mathcal{X} \to \mathcal{U}$ be a policy in the stationary deterministic policy space Π, $C(x_t, u_t)$ be a contribution/reward function, and γ be a discount factor in $(0, 1)$. The system evolves according to the following state transition function

$$x_{t+1} = S^M(x_t, u_t, W_{t+1}), \tag{1}$$

where W_{t+1} represents the stochastic exogenous information that arrives during the time interval from t to $t + 1$. The goal is to find a policy that solves

$$\sup_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t=0}^{\infty} \gamma^t C(x_t, \pi(x_t)) \right\}, \tag{2}$$

where the initial state x_0 is deterministic. Due to the computational intractability of solving the objective function (2) directly, Bellman's equation is often used to compute the optimal control recursively. To handle continuous problems, we favor writing Bellman's equation with the expectation form which is equivalent to the traditional use of a one-step transition matrix in the discrete case,

$$V_t(x_t) = \sup_{u_t \in \mathcal{U}} \{C(x_t, u_t) + \gamma \mathbb{E}[V_{t+1}(x_{t+1})|x_t]\}, \tag{3}$$

where $V_t(x_t)$ is the value function representing the value of being in state x_t by following the optimal policy onward and the expectation is taken over the random information variable W_{t+1}. For infinite horizon steady state problems, the subscript t is dropped and Bellman's optimality equation becomes

$$V(x) = \sup_{u \in \mathcal{U}} \left\{C(x, u) + \gamma \mathbb{E}[V(S^M(x, u, W)]\right\}. \tag{4}$$

Value functions define a partial ordering over policies. That is, $\pi \geq \pi'$ if and only if $V^{\pi}(x) \geq V^{\pi'}(x)$ for all $x \in \mathcal{X}$. Let V^* denote the optimal value function defined as

$$V^*(x) = \sup_{\pi \in \Pi} V^{\pi}(x), \tag{5}$$

for all $x \in \mathcal{X}$. It is well-known that the optimal value function V^* satisfies equation (4).

There are two contraction operators associated with infinite-horizon Markov decision processes.

Definition 2.1 (Bellman's optimality operator) *Let M be the Bellman operator such that for all $x \in \mathcal{X}$ and $V \in \mathbf{C}^b(\mathcal{X})$,*

$$MV(x) = \sup_{u \in \mathcal{U}} \{C(x, u) + \gamma \int_{\mathcal{W}} Q(x, u, dw) V(S^M(x, u, w))\},$$

where $\mathbf{C}^b(\mathcal{X})$ is the space of all bounded continuous functions on $\mathcal{X}$ and Q is the transition probability function that is assumed to make M map $\mathbf{C}^b(\mathcal{X})$ into itself.

Definition 2.2 (Bellman's fixed policy Operator) *Let M^π be the operator for a fixed policy π such that for all $x \in \mathcal{X}$*

$$M^\pi V(x) = C(x, \pi(x)) + \gamma \int_{\mathcal{W}} Q(x, \pi(x), dw) V(S^M(x, \pi(x), w))$$

where Q and S^M have the same property as in definition 2.1.

The operators M and M^π have the desired properties of monotonicity and contraction (see Bertsekas & Shreve (1978), Ma & Powell (2010)) that are crucial in the convergence analysis of algorithms for infinite horizon policy iteration problems.

The following assumptions regarding MDPs are necessary for proving convergence of algorithm with continuous states.

Assumption 2.1 *The state space $\mathcal{X}$, the decision space $\mathcal{U}$ and the outcome space $\mathcal{W}$ are convex, compact and Borel subsets of $\mathbb{R}^m$, $\mathbb{R}^n$ and $\mathbb{R}^l$ respectively.*

Assumption 2.2 *Assume that the contribution function C, the state transition function S^M and the transition probability density function $Q : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathbb{R}_+$ are all continuous.*

It is easy to see the contribution and transition function are uniformly bounded given assumption 2.1 and 2.2. Since we work with discounted problems, the objective function (2) is also bounded.

2.2 Post-decision state variable

Computing the expectation within the Bellman operators M and M^π directly is often intractable in practice when the underlying distribution of the evolution of the stochastic system is unknown or the decision u is a vector and integration becomes burdensome. Such difficulty can be handled by using the post-decision state variable (see Van Roy et al.

(1997), Powell (2007), also known as end-of-state Judd (1998) and after-state Sutton & Barto (1998)). Suppose we can break the original transition function (1) into the two steps

$$x_t^u = S^{M,u}(x_t, u_t), \tag{6}$$

$$x_{t+1} = S^{M,W}(x_t^u, W_{t+1}). \tag{7}$$

We let x_t^u the post-decision state, which is the state immediately after we make a decision. Taking a simple resource allocation problem for example, we let x_t be a vector of supplies and demands for the resource at time t, u_t (also a vector) be how much to use at time t, and W_{t+1} be random changes in both supplies and demands. For example, in an energy storage application, u_t might be the amount of energy to withdraw from storage (or add to storage if $u_t < 0$), and W_{t+1} might be random rainfall into a reservoir. Then, the transition equations in (6) and (7) become

$$\begin{aligned} x_t^u &= x_t - u_t \\ x_{t+1} &= x_t^u + W_{t+1}. \end{aligned}$$

Let the post-decision state space be $\mathcal{X}^u$ and the post-decision value function $V^u : \mathcal{X}^u \to \mathbb{R}$ be

$$V^u(x_t^u) = \mathbb{E}\{V(x_{t+1})|x_t^u\}, \tag{8}$$

where $V^u(x_t^u)$ represents the value of being in the post decision states x_t^u. Suppose the pre-decision value function $V \in \mathcal{B}(\mathcal{X})$ and $V^u \in \mathcal{B}(\mathcal{X}^u)$. There is a simple relationship between the pre-decision value function $V(x_t)$ and post-decision value function $V^u(x_t^u)$ that is summarized as

$$V(x_t) = \max_{u_t \in \mathcal{U}} \{C(x_t, u_t) + \gamma V^u(x_t^u)\}. \tag{9}$$

By substituting (9) into (8), we have Bellman's equation using the post-decision value function

$$V^u(x_t^u) = \mathbb{E}\{\max_{u_{t+1} \in \mathcal{U}} \{C(x_{t+1}, u_{t+1}) + \gamma V^u(x_{t+1}^u)\}|x_t^u\}. \tag{10}$$

A popular strategy in the reinforcement learning community is to use Q-learning, which requires approximating the Q-factors $Q(x, u)$ which depend on states and actions. When actions are multidimensional, this greatly complicates the challenge of designing effective approximations. When both the states and actions are continuous vectors, curses of dimensionality make the application of Q-learning impractical. In our approach, we only need to approximate $V^u(x^u)$, where the dimensionality of the post-decision state x^u is often much lower than the dimensionality of the state-action pair (x, u).

For a fixed policy π, the transition steps (6) and (7) become

$$\begin{aligned} x_t^\pi &= S^{M,\pi}(x_t, \pi(x_t)), \\ x_{t+1} &= S^{M,W}(x_t^\pi, W_{t+1}). \end{aligned}$$

As a result, the Markov decision problem can be reduced to a Markov chain for post-decision states. By the simple relationships between pre- and post-decision value functions (9), we have that the post-decision value function is $V^{\pi,u}(x_t^u) = \mathbb{E}V^\pi(x_{t+1})$. More specifically, $V^{\pi,u}(x_0^u) = \mathbb{E}\left\{\sum_{t=1}^\infty \gamma^{t-1} C(x_t, \pi(x_t))\right\}$. Hence, we can write Bellman's equation (10) for the post-decision state as

$$V^{\pi,u}(x) = \int_{\mathcal{X}^\pi} P(x, dx')(C^\pi(x, x') + \gamma V^{\pi,u}(x')), \tag{11}$$

where $V^{\pi,u}$ is the value of following the fixed policy π, $P(\cdot,\cdot)$ is the transition probability function of the chain, $C^\pi(\cdot,\cdot)$ is the stochastic contribution/reward function with $C^\pi(x_t^\pi, x_{t+1}^\pi) = C(x_{t+1}, \pi(x_{t+1}))$ and $\mathcal{X}^\pi \subset \mathbb{R}^d$ is the post decision state space by following policy π. From now on, we only work with post-decision states and drop the superscript u for simplicity of presentation. It is worth noting that $\mathcal{X}^\pi$ is compact since $\mathcal{X}$ and $\mathcal{U}$ are compact and the state transition function S^M is continuous by assumptions 2.1 and 2.2 respectively and the dimensionality d of $\mathcal{X}^\pi$ is often lower than the pre-decision state dimensionality m.

2.3 Markov chains with continuous state space

The stability of the underlying process is of great importance in proving convergence results of any algorithm. In the case of fixed policy evaluation in policy iteration algorithm, the underlying process is a Markov chain. To work with Markov chains with continuous state space, we present the following definitions of irreducibility, invariant measure, recurrence and positivity that all have familiar counterparts in discrete chains. They are all helpful in defining the behavior of chains with desirable topological structure of the general state space Meyn & Tweedie (1993).

Definition 2.3 (ψ-Irreducibility for general space chains) *For any measure φ, a Markov chain Φ on state space $\mathcal{X}$ is called φ-irreducible if there exists a measure φ on $\mathcal{B}(\mathcal{X})$ such that whenever $\varphi(A) > 0$ for $A \in \mathcal{B}(\mathcal{X})$, we have*

$$\mathbb{P}_x\{\Phi \text{ ever enters } A\} > 0, \forall x \in \mathcal{X}$$

where $\mathbb{P}_x$ denotes the conditional probability on the event that the chain starts in state x. Let ψ be the maximal irreducibility measure among such measures. (For the existence of ψ, see proposition 4.2.2 of Meyn & Tweedie (1993).)

Definition 2.4 (Invariant measure) *Let $P(\cdot, \cdot)$ be the transition kernel of a chain Φ on the state space $\mathcal{X}$. A σ-finite measure μ on $\mathcal{B}(\mathcal{X})$ with the property*

$$\mu(A) = \int_{\mathcal{X}} \mu(dx) P(x, A), \forall A \in \mathcal{B}(\mathcal{X})$$

will be called invariant.

Definition 2.5 (Positive Harris chain) *The set $A \in \mathcal{B}(\mathcal{X})$ is called Harris recurrent if*

$$\mathbb{P}_x\{\Phi \in A \text{ infinitely often}\} = 1, \forall x \in \mathcal{X}.$$

A chain Φ is called Harris (recurrent) if it is ψ-irreducible, admits an invariant probability measure μ and every set in

$$\mathcal{B}^+(\mathcal{X}) = \{A \in \mathcal{B}(\mathcal{X}) : \psi(A) > 0\}$$

is Harris recurrent.

The most desired feature of a positive Harris chain is the applicability of the well-known strong law of large numbers, which is summarized in the following lemma from Meyn & Tweedie (1993). It is fundamental and essential to convergence analysis of different reinforcement learning and dynamic programming algorithms (parametric or non-parametric).

Lemma 2.1 (Law of large numbers for positive Harris chains) *If Φ is a positive Harris chain (see definition 2.5) with invariant probability measure μ, for each $f \in L_1(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$,*

$$\lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} f(x_i) = \int_{\mathcal{X}} \mu(dx) f(x)$$

almost surely.

2.4 Policy iteration

The main focus of the paper is policy iteration algorithms, which typically consist of two loops: the inner loop for fixed policy evaluation and the outer loop for policy improvement. The convergence of exact policy iteration for general state space is well known (see Bertsekas & Shreve (1978)). The primary convergence result that is derived from the aforementioned monotonicity and contraction properties of the Bellman's operators M and M^π is stated as follows.

Proposition 2.1 *Let $(\pi_n)_{n=0}^{\infty}$ be a sequence of policies generated recursively as follows: given an initial policy π_0, for $n \geq 0$,*

$$\pi_{n+1}(x) = \arg\max_u \{C(x,u) + \gamma \int_{\mathcal{W}} Q(x,u,dw) V^{\pi_n}(S^M(x,u,w))\}.$$

Then $V^{\pi_n} \to V^*$ uniformly where V^* is the optimal value function.

However, the exact policy iteration algorithm is only conceptual because in practice the expectation cannot be computed exactly. Hence, the approximate policy iteration algorithm, in which the policy evaluation stops in finite time, is considered. In approximate policy iteration, the estimated value function of a policy is random (a statistical estimate of the true value function of the policy) because it depends on the sample trajectory of the chain

by following the policy and also the iteration counter of policy evaluation. Given the state space being compact and the norm being the sup norm $||\cdot||_\infty$ for continuous functions, Ma & Powell (2010) proves convergence in mean of the approximate policy iteration algorithm, which is summarized in the following theorem.

Theorem 2.1 (Mean convergence of approximate policy iteration)

Let $\hat{\pi}_0, \hat{\pi}_1, \ldots, \hat{\pi}_n$ be the sequence of policies generated by an approximate policy iteration algorithm and let $\hat{V}^{\hat{\pi}_0}, \hat{V}^{\hat{\pi}_1}, \ldots, \hat{V}^{\hat{\pi}_n}$ be the corresponding approximate value functions. Further assume that, for each fixed policy $\hat{\pi}_n$, the MDP is reduced to a Markov chain that admits an invariant probability measure $\mu_{\hat{\pi}_n}$. Let $\{\epsilon_n\}$ and $\{\delta_n\}$ be positive scalars that bound the mean errors in approximations to value functions and policies (over all iterations) respectively, that is $\forall n \in \mathbb{N}$,

$$\mathbb{E}_{\mu^{\hat{\pi}_n}} ||\hat{V}^{\hat{\pi}_n} - V^{\hat{\pi}_n}||_\infty \leq \epsilon_n, \tag{12}$$

and

$$\mathbb{E}_{\mu^{\hat{\pi}_n}} ||M_{\hat{\pi}_{n+1}} \hat{V}^{\hat{\pi}_n} - M\hat{V}^{\hat{\pi}_n}||_\infty \leq \delta_n. \tag{13}$$

Suppose the sequences $\{\epsilon_n\}$ and $\{\delta_n\}$ converge to 0 and

$$\lim_{n\to\infty} \sum_{i=0}^{n-1} \gamma^{n-1-i} \epsilon_i = \lim_{i=0}^{n-1} \gamma^{n-1-i} \delta_i = 0,$$

e.g. $\epsilon_i = \delta_i = \gamma^i$. Then, this sequence eventually produces policies whose performance converges to the optimal performance in the mean:

$$\lim_{n\to\infty} \mathbb{E}_{\mu^{\hat{\pi}_n}} ||\hat{V}^{\hat{\pi}_n} - V^*||_\infty = 0.$$

In plain English, the above theorem states that the mean difference between the optimal policy value function and the estimated policy value function using approximate policy iteration shrinks to 0 as the successive approximations of the value functions improve. This result lays down the basic foundations for analyzing convergence properties of algorithms using different kernel smoothing techniques that are discussed in later sections.

3 Least Squares Policy Iteration with Mercer Kernel

Temporal-difference (TD) learning algorithms (see Tsitsiklis & Van Roy (1997), Tadić (2001)) have been widely used for value function approximation in reinforcement learning, and least squares temporal difference (LSTD) learning algorithms (Bradtke & Barto (1996), Boyan (1999)) uses linear approximation and applies least squares method to improve data efficiency. Although LSTD learning has the merits of faster convergence rate and better performance than conventional TD algorithms, least squares approaches can not be applied directly to TD algorithm with nonlinear function approximation such as neural networks and it has been shown that TD algorithms with nonlinear approximation can actually diverge (see Tsitsiklis & Van Roy (1997)). To implement efficient and convergent TD learning with nonlinear function approximation, in this section we study the temporal difference learning algorithm that uses Mercer kernel functions to approximate value functions. Even though the value function approximation with Mercer kernels is nonlinear and nonparametric, the learning algorithms and the proof techniques for convergence analysis are all closely related to the recursive least squares policy iteration algorithm with linear function approximation owing to the famous kernel trick.

The trick known as Mercer's theorem basically states that any continuous, symmetric and positive semi-definite kernel function $K(x, y)$ can be expressed as a dot product in a high-dimensional feature space $\mathcal{F}$.

Theorem 3.1 (Mercer's theorem) *Let $\mathcal{S}$ a measurable space and the kernel K be a positive and semi-definite function i.e.*

$$\sum_{i,j} K(s_i, s_j) r_i r_j \geq 0$$

for any finite subset $\{s_1, ..., s_n\}$ of $\mathcal{S}$ and any real numbers $\{r_1, ..., r_n\}$. There exists a function $\phi : \mathcal{S} \to \mathcal{F}$, where $\mathcal{F}$ (feature space) is an inner product space of possibly high dimension, such that

$$K(x, y) = \langle \phi(x), \phi(y) \rangle.$$

To solve non-linear problems, the kernel trick can be applied to map the original non-linear space into a higher-dimensional inner product feature space $\mathcal{F}$ such that a linear algorithm such as least squares can be subsequently used. As a result, the kernel trick transforms a nonlinear algorithm to an equivalent linear one that operates on the feature space $\mathcal{F}$. This is only used to prove convergence of the algorithm. In the implementation of the algorithm, the kernel function is used to replace any dot product between two vectors of feature function ϕ. Hence, the inner products in feature space $\mathcal{F}$ do not make direct reference to feature vectors and as a result the feature vectors are never explicitly calculated. This is a desirable property for infinite-dimensional feature spaces such as the one associated with Gaussian kernels where direct computation of the inner products is not feasible.

Moreover, the kernel function K determines a Hilbert space H_K, which is often called reproducing kernel Hilbert space (RKHS). H_K is a vector space containing all linear combinations of the functions $K(\cdot, x)$, $f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$. Let $g(\cdot) = \sum_{j=1}^{n} \beta_j K(\cdot, y_j)$. Then the inner product is defined as:

$$\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j K(x_i, y_j),$$

and the norm is defined as $||f||_{H_K} = \sqrt{\langle f, f \rangle}$.

The Hilbert space L_2 is usually too big for analysis purposes because it contains too many non-smooth functions, while a RKHS is a restricted smooth function space smaller than the general Hilbert space. To do convergence analysis, we impose the following assumption on the policy value functions and the policy space Π.

Assumption 3.1 *Assume that the policy value function for any fixed policy $\pi \in \Pi$ is in the RKHS H_K.*

Xu et al. (2007) proposes a kernel version of the least squares policy iteration (LSPI see Lagoudakis & Parr (2003)) algorithm, which applies the kernel recursive least squares algorithm developed in Engel et al. (2004) to the LSTD-Q algorithm for approximating station-action Q-factor of a fixed policy. The algorithm is empirically shown to provide better

approximation accuracy for nonlinear value function approximation problems compared to conventional linear TD and LSTD algorithms.

Parallel to the KLSPI algorithm, we consider a kernelized version of the recursive least squares approximate policy algorithm (RLSAPI) developed in Ma & Powell (2010) that works around the post-decision state rather than the state-action pair. The algorithm (call it KRLSAPI) uses KLSTD to approximate policy value function around post-decision states for policy evaluation. As discussed at the beginning of the section, the basic concept of KLSTD is to use a Mercer kernel function to realize nonlinear LSTD learning in a high-dimensional feature space $\mathcal{F}$ produced by a kernel-based feature mapping. Let the feature mapping associated with $\mathcal{F}$ be $\phi : \mathcal{S}^\pi \to \mathcal{F}$ where $\mathcal{S}^\pi \subset \mathbb{R}^d$ is the original post-decision state space of following a fixed policy π, and $\phi(x)$ is the vector of feature functions. Recall that, by the kernel trick, all the inner products in feature spaces can be replaced by the computation of kernel functions $K(x, y) = \langle \phi(x), \phi(y) \rangle$, which will greatly simplify the computational difficulty caused by high-dimensional feature spaces in our algorithm.

Based on the above idea of the kernel trick, the KLSTD algorithm for policy evaluation approximates the value function of a Markov chain around post-decision states by using: $\hat{V}_m^\pi(x) = \phi(x)^T \theta_m$, where $\phi(x)$ and θ_m are both column vectors in H_K and m is the number of samples. According to the Kernel Representor theorem in Kimeldorf & Wahba (1971), the weight vector θ can be represented by the weighted sum of state feature vectors, i.e. $\theta_m = \sum_{i=0}^m \phi(x_i)\beta_i$ where $x_i (i = 1, 2, \cdots, m)$ are the observed post-decision states and $\beta_i (i = 1, 2, \cdots, m)$ are the corresponding coefficients. By the reproducing kernel property, we have

$$\hat{V}^\pi(x) = \phi^T(x)\theta = \sum_{i=1}^m K(x, x_i)\beta_i.$$

As a result, similar to the LSTD algorithm, the least-squares regression equation for KLSTD can be expressed as follows

$$\sum_{i=1}^m \phi(x_i)(\phi(x_i) - \gamma\phi(x_{i+1}))^T \sum_{i=1}^m \phi(x_i)\beta_i = \sum_{i=1}^m \phi(x_i)c_i + \epsilon_i,$$

where $c_i = C^\pi(s_i, s_{i+1})$ is the i-th observation of the contribution and ϵ_i is the noise term

for each time step.

The single-step regression function is

$$\phi(s_i)(\phi(s_i) - \gamma\phi(s_{i+1}))^T \sum_{i=1}^{m} \phi(s_i)\beta_i = \sum_{i=1}^{m} \phi(s_i)c_i + \epsilon_i.$$

Let $\Phi_m = [\phi(x_1), \cdots, \phi(x_m)]^T$ and $k_m(x_i) = [K(x_1, x_i), \cdots, K(x_m, x_i)]^T$. By multiplying Φ_m on both sides of the previous equation, due to the kernel trick, we have

$$k_m(x_i)(k_m(x_i) - \gamma k_m(x_{i+1}))\beta_m = k_m(x_i)c_i + \Phi_m\epsilon_i.$$

Then the new least squares regression function is

$$\sum_{i=1}^{m} k_m(x_i)(k_m(x_i) - \gamma k_m(x_{i+1}))\beta_m = \sum_{i=1}^{m} k_m(x_i)c_i.$$

Let $M_m = \sum_{i=1}^{m} k_m(x_i)(k_m(x_i) - \gamma k_m(x_{i+1}))$ and $b_m = \sum_{i=1}^{m} k_m(x_i)c_i$. Then, the recursive least squares solution to the kernel-based TD learning problem is $\beta_m = M_m^{-1}b_m$ with

$$M_{m+1} = M_m + k_{m+1}(x_{m+1})(k_{m+1}^T(x_{m+1}) - \gamma k_{m+1}^T(x_{m+2}))$$

and

$$b_{m+1} = b_m + k_{m+1}(x_{m+1})c_m.$$

Similar to the KLSPI in Xu et al. (2007), KRLSAPI uses KLSTD for policy evaluation and makes policy improvement until the optimal policy is reached. In KLSPI or KRLSAPI, the kernel-based feature vectors are automatically generated by the kernel function. This data-driven automatic feature selection provides an efficient solution to the fundamental difficulty of the LSPI algorithm. The details of the KLSPI algorithm is shown in figure 1. It is worth noting that the arg max function in step 7 is usually a multivariate and potentially non-convex optimization problem, which may require an external solver such as a nonlinear proximal point algorithm (Luque (1987)).

Moreover, the approximation and generalization ability of kernel methods greatly contribute to the convergence and performance of the approximate policy iteration algorithm. We start with the convergence analysis of the policy evaluation. The following theorem presents that the KLSTD algorithm converges in probability.

Figure 1: Kernel-based Recursive Least Squares Approximate Policy Iteration (KRLSAPI)

Step 0: Initialization:

Step 0a. Set the initial policy π_0.

Step 0b. Set the kernel function K.

Step 0c. Set the iteration counter $n = 0$.

Step 0d. Set the initial State S_0^0.

Step 1: Do for $n = 0, \ldots, N$,

Step 2: Do for $m = 1, \ldots, M$:

Step 3: Initialize $\hat{c}_m = 0$.

Step 4: Choose one step sample realization ω.

Step 5: Do the following:

Step 5a. Set $x_m^n = \pi_n$.

Step 5b. Compute $S_m^{n,x} = S^{M,x}(S_m^n, x_m^n)$ and $S_{m+1}^n = S^M(S_m^{n,x}, W_{m+1}(\omega))$.

Step 5c. Compute and store the corresponding kernel function value $k_m(S_m^{n,x})$ and $k_m(S_{m+1}^{n,x})$ in dictionary.

Step 6: Do the following:

Step 6a. Compute and store $\hat{c}_m = C(S_m^n, x_m^n)$ and M_m and b_m

Step 6b. Update parameters $\hat{\beta}^{n,m} = M_m^{-1} b_m$

Step 7: Update the parameter and the policy:

$$\hat{\beta}^{n+1} = \hat{\beta}^{n,M},$$

$$\pi_{n+1}(s) = \arg\max_{x \in \mathcal{X}} \{C(s,x) + \gamma k_{s^x}^T \hat{\beta}^{n+1}\}.$$

Step 8: Return the policy π_t^N and parameters $\hat{\beta}^N$.

Theorem 3.2 (Convergence of KLSTD in probability) *Suppose assumptions 2.1 and 2.2 hold and the Markov chain of the post-decision states (following a fixed policy π) follows a positive Harris chain having transition kernel $P(x, dy)$ and invariant probability measure μ. Further assume the policy value function V^π is in the reproducing kernel Hilbert space H_K with the kernel function K being C^∞ and $||K||_\infty \leq M$ for some constant M. Then, $\hat{V}_m^\pi \to V^\pi$ in probability where $\hat{V}_m^\pi(x) = \sum_{i=1}^m K(x, x_i)\beta_i$.*

We provide a sketch of the proof and omit details. We know that

$$\hat{V}_m^\pi(x) = \arg\min_{f \in H_K} \{R_{emp}(f) + \lambda ||f||_{H_K}\}$$

where R_{emp} is the empirical quadratic loss function and $\lambda \geq 0$ is the regularization term (in the original version of KLSTD, $\lambda = 0$). By the proof to corollary 6.2 of Wu et al. (2006) and theorem 6 of Smale & Zhou (2005), for any $0 < \epsilon < 1$ we have with probability $1 - \epsilon$

$$||\hat{V}_m^\pi - V^\pi||_{H_K} \leq C(\frac{\log(4/\epsilon)^2}{m})^{\frac{1}{6}}$$

where C is a constant. Therefore, $\hat{V}_m^\pi \to V^\pi$ in H_K norm in probability. By the reproducing property of the kernel and the Cauchy-Schwartz inequality, we have for any x,

$$f(x)^2 = \langle K(\cdot, x), f \rangle^2 \leq K(x, x)||f||_{H_K}^2.$$

As a result,

$$||\hat{V}_m^\pi - V^\pi||_\infty \leq \sqrt{M}||\hat{V}_m^\pi - V^\pi||_{H_K}.$$

Hence, $||\hat{V}_m^\pi \to V^\pi||_\infty \to 0$ in probability. Moreover, we have $\mathbb{E}_{\mu_\pi}||\hat{V}_m^\pi \to V^\pi||_\infty \to 0$ since $\hat{V}_m^\pi \to V^\pi$ are uniformly bounded owing to bounded kernel and compact post-decision space.

The convergence of KRLSAPI is fully determined by the convergence of the KLSTD algorithm and the approximate errors of the policy evaluation and policy updating in the approximate policy iteration algorithm. The following theorem illustrates the details. The proof is omitted since it is a direct application of theorems 3.2 and 2.1.

Corollary 3.1 (Mean convergence of KRLSAPI) *Suppose assumptions in theorem3.2 and assumption 3.1 hold. Theorem 2.1 applies to the kernel least squares approximate policy iteration algorithm in figure 1.*

To make the aforementioned Kernel algorithm practical, one key problem is to decrease the computational and memory costs of kernel vectors, whose dimension is originally equal

to the sample size. This is a common problem to almost all the kernel-based learning algorithms, so sparsification procedures are necessary to cope with the problem. There are several good candidates in the literature. Xu et al. (2007) uses approximate linear dependence developed in Engel et al. (2004) for online sparsification. Jung & Polani (2006) employs a subspace-based variant of least squares support vector machine for regularization and sparse approximation (in this case $\lambda > 0$) in the online learning setting. Loth et al. (2007) considers a reinforcement learning algorithm using LASSO for L_1 regularization for the purpose of kernel sparsification. Farahmand et al. (2009*a*,*b*) propose to use L_2-regularization to control the complexity of value function approximation including the kernel methods.

4 Approximate policy iteration with kernel smoothing

Following the idea in Rust (1997), the kernel-based value iteration algorithm in Ormoneit & Sen (2002) approximates solutions to continuous MDPs with discretization and kernel average. One limitation of the algorithm is the action space is assumed to be finite. Another limitation is the algorithm is off-policy and the sampling scheme is assumed to be uniform. Following the same line as Ormoneit & Sen (2002), we apply kernel smoothing to policy evaluation in our on-policy approximate policy iteration algorithm for continuous action space.

We first define the following random approximation of the fixed-policy Bellman operator $\hat{M}_m^\pi$ from a sample path of size $m+1$:

$$\hat{M}_m^\pi V(x) = \sum_{i=0}^{m-1} k(x_i, x) \left(C^\pi(x_i, x_{i+1}) + \gamma V(x_{i+1}) \right),$$

where

$$k(x_i, x) = \frac{K(\frac{||x_i - x||}{b})}{\sum_{i=0}^{m-1} K(\frac{||x_i - x||}{b})}$$

and $(x_i)_{i=0}^m$ is the sample path of historical transitions by following the policy π. The kernel weighting function k assigns equal weight to equidistant points and is determined by a univariate, non-negative kernel function K. By design, the weights are all positive and sum up to 1.

Since we have the fixed point property $V^\pi = M^\pi V^\pi$ for M^π, $\hat{M}_m^\pi V^\pi$ could be viewed as an approximation of V^π. First, we use the random operator $\hat{M}^\pi$ to estimate the true post-decision policy value function V^π on the finite sample of post-decision states $(x_i)_{i=1}^m$ by finding the fixed point solution the approximate Bellman equation

$$\hat{V}^\pi = \hat{M}_m^\pi \hat{V}^\pi = \hat{P}^\pi [c^\pi + \gamma \hat{V}^\pi], \tag{14}$$

where $\hat{V}^\pi$ is a vector of m for post-decision decision states $(x_i)_{i=1}^m$, P^π is a $m \times m$ stochastic matrix with i,j-th entry being $k(x_{i-1}, x_j)$ for $i, j \in \{1 \cdots m\}$ and c^π is the reward vector of dimension m with i-th entry being $C^\pi(x_{i-1}, x_i)$ for $i \in \{1 \cdots m\}$. Since $\hat{P}^\pi$ is a stochastic matrix, $I - \gamma P^\pi$ is nonsingular. This guarantees the existence and uniqueness of the solution, which is simply

$$\hat{V}^\pi = (I - \gamma \hat{P}^\pi)^{-1} c^\pi. \tag{15}$$

Then, the algorithm extrapolates the policy value function estimate to $x \in \mathcal{X}^\pi \setminus \{x_i\}_{i=1}^m$ with

$$\hat{V}^\pi(x) = \sum_{i=0}^{m-1} k(x_i, x) \left(C^\pi(x_i, x_{i+1}) + \gamma \hat{V}^\pi(x_{i+1}) \right). \tag{16}$$

Figure 2 illustrates details of the kernel-based approximate policy iteration algorithm.

In order to show convergence of the algorithm, we first introduce the following additional technical assumption on state space, contribution function, kernel function and the underlying Markov chain.

Assumption 4.1

a. For each policy $\pi \in \Pi$, $\mathcal{S}^\pi = [0,1]^d$.

b. The contribution function, $C^\pi(x, y)$ is a jointly Lipschitz continuous function of x and y i.e. there exists a $K_C > 0$ such that

$$|r(x', y') - r(x, y)| \leq K_C ||(x' - x, y' - y)||. \tag{17}$$

Figure 2: Approximate policy iteration with kernel smoothing (KSAPI)

Step 0: Initialization:

Step 0a. Set the initial policy π_0.

Step 0b. Set the kernel function K.

Step 0c. Set the iteration counter $n = 0$.

Step 1: Do for $n = 0, \ldots, N$,

Step 1a. Set the iteration counter $l = 0$.

Step 1b. Set the initial state x_0^n.

Step 1c: Do for $j = 0, \cdots, m$:

Step 1c.1: Set $u_j^n = \pi_n(x_j^n)$ and draw randomly or observe W_{j+1} from the stochastic process,

Step 1c.2: Compute $x_j^{n,\pi} = S^{M,\pi}(x_j^n, u_j^n)$ (store also) and $x_{j+1}^n = S^M(x_j^{n,\pi}, u_j^n, W_{j+1})$.

Step 1d. Let c^π be a vector of dimensionality m with i-th entry $C^\pi(x_{i-1}^{n,\pi}, x_i^{n,\pi})$ for $i = 1 \cdots m$.

Step 1e. Let $\hat{P}^\pi$ be a matrix of dimensionality $m \times m$ with i, j-th entry $k(x_{i-1}^{n,\pi}, x_j^{n,\pi})$ for $i, j \in \{1 \cdots m\}$.

Step 2: Solve for $\hat{v} = (I - \gamma \hat{P}^\pi)^{-1} c^\pi$ with $\hat{v}$ being a vector of dimensionality m with i-th element $\hat{v}(x_i^{n,\pi})$ for $i = 1 \cdots m$.

Step 3: Let $\hat{v}^n(x) = \sum_{i=0}^{m-1} k(x_i^{n,\pi}, x) \left(C^\pi(x_i^{n,\pi}, x_{i+1}^{n,\pi}) + \gamma \hat{v}(x_{i+1}^{n,\pi}) \right)$

Step 4: Update the policy:

$$\pi_{n+1}(x) = \arg\max_{u \in \mathcal{U}} \{C(x, u) + \gamma \hat{v}^n(x^u)\}.$$

Step 5: Return the policy π_{N+1}.

---

c. The kernel function $K^+ : [0, 1] \to \mathbb{R}^+$ is Lipschitz continuous, satisfying $\int_0^1 K^+(x)dx = 1$ and K is the completion of K^+ on $\mathbb{R}$.

d. For each policy $\pi \in \Pi$, the invariant probability measure μ^π is absolutely continuous with respect to the Lebesgue measure λ and $0 < \underline{K}_\pi \leq \frac{d\mu_\pi}{d\lambda} \leq \bar{K}_\pi$. In other words, the invariant probability measure μ^π has a continuous density function f^π such that f^π is bounded from

above and away from 0 on $\mathcal{S}^\pi$.

Remark: It looks like we place a strong restriction on the post-decision state space to a d-dimensional unit cube $[0,1]^d$ in part a of assumption 4.1. However, the assumption can be relaxed to the state space $\mathcal{X}^\pi = [a_1, b_1] \times \cdots \times [a_d, b_d]$. We know that $\mathcal{X}^\pi$ is isomorphic to $[0,1]^d$ with a bijective linear mapping $L : \mathcal{X}^\pi \to [0,1]^d$. Let P^π, μ_π and V^π be the transition kernel, invariant measure and policy value function on $\mathcal{X}^\pi$. Then, $P^\pi \circ L^{-1}$, $\mu_\pi \circ L^{-1}$ and $V^\pi \circ L^{-1}$ are the corresponding counterparts for $[0,1]^d$. Hence, we can assume the state space is $[0,1]^d$ without loss of generality since we can recover the policy value function V^π on $\mathcal{X}^\pi$ from $V^\pi \circ L^{-1}$ on $[0,1]^d$ using the identity $V^\pi = V^\pi \circ L^{-1} \circ L$.

We first present the following lemma on kernel volume bound, which provides crucial bounds in the main convergence result.

Lemma 4.1 (Kernel volume bound) *Let* $K_v = \int_{[0,1]^d} K(\frac{||u-y||}{b})\mu^\pi(du)$ *and* $\nu_n = \int_0^1 r^n K(r)dr$. $K_v \geq \frac{b^d 2\pi^{d/2}\nu_{d-1}\underline{K}_\pi}{\Gamma(d/2)}$ *where* $\Gamma(\cdot)$ *is Euler's gamma function.*

Proof:

By assumption 4.1.d,

$$\begin{aligned}
\int_{[0,1]^d} K(\frac{||u-y||}{b})\mu^\pi(du) &= \int_{[0,1]^d} K(||\frac{u-y}{b}||)f^\pi(u)du \\
&\geq \underline{K}_\pi \int_{[0,1]^d} K(||\frac{u-y}{b}||)du \\
&= \underline{K}_\pi b^d \int_{\mathbb{R}^d} K(||u||)du \\
&= \frac{b^d 2\pi^{d/2}\underline{K}_\pi}{\Gamma(d/2)} \int_0^1 r^{d-1}K(r)dr.
\end{aligned}$$

∎

We now present the convergence in mean of the policy evaluation step in figure 2.

Theorem 4.1 (Convergence in the mean of policy evaluation) *Suppose assumptions 2.1, 2.2 and 4.1 hold and the Markov chain of the post-decision states follows a positive Harris chain having transition kernel $P(x, dy)$ and invariant probability measure μ. Let the bandwidth $b(m)$ satisfy $b(m)^{d+1}\sqrt{m} \to \infty$ and $b(m) \to 0$ e.g. $b(m) = m^{-\frac{1}{2(d+2)}}$. Let $\hat{V}_m^\pi$ be defined as in (16). Then $\mathbb{E}_{\mu_\pi}||\hat{V}_m^\pi - V^\pi||_\infty \to 0$ as $m \to \infty$.*

Proof:

First define the asymptotic weighting kernel function $\tilde{k}$ be:

$$\tilde{k}(x, y) = \frac{K(\frac{||x-y||}{b})}{\int_{[0,1]^d} K(\frac{||u-y||}{b})\mu^\pi(du)}. \tag{18}$$

Since the chain is positive Harris, by lemma 2.1

$$mk(x, y) = \frac{K(\frac{||x-y||}{b})}{\frac{1}{m}\sum_{i=0}^{m-1} K(\frac{||X_i-y||}{b})} \to \tilde{k}(x, y) \tag{19}$$

almost surely as $m \to \infty$. For any Lipschitz continuous $V \in C([0, 1]^d)$, define

$$\tilde{M}^\pi V(x) = \tilde{k}(X, x)(C^\pi(X, Y) + \gamma V(Y)), \tag{20}$$

and we can write

$$\hat{M}_m^\pi V(x) = \tilde{M}_m V(x) = \sum_{i=0}^{m-1} \frac{\tilde{k}(X_i, x)}{\sum_{i=0}^{m-1} \tilde{k}(X_i, x)}(C^\pi(X_i, X_{i+1}) + \gamma V(X_{i+1})), \tag{21}$$

since $\tilde{k}$ is the kernel function K normalized by $\int_{[0,1]^d} K\left(\frac{||u-y||}{b}\right)\mu^\pi(du)$. Then, we have almost surely

$$\begin{aligned}
\lim_{m\to\infty} \hat{M}_m^\pi V(x) &= \lim_{m\to\infty} \frac{1}{m}\sum_{i=0}^{m-1} \frac{K(\frac{||x_i-x||}{b})}{\frac{1}{m}\sum_{i=0}^{m-1} K(\frac{||x_i-x||}{b})} (C^\pi(x_i, x_{i+1}) + \gamma V(x_{i+1})) \\
&= \int_{[0,1]^d}\int_{[0,1]^d} \frac{K(\frac{||z-x||}{b})}{\int_{[0,1]^d} K(\frac{||u-x||}{b})\mu^\pi(du)} (C^\pi(z, w) + \gamma V(w))\, P^\pi(z, dw)\mu^\pi(dz) \\
&= \int_{[0,1]^d}\int_{[0,1]^d} \tilde{k}(z, x)\, (C^\pi(z, w) + \gamma V(w))\, P^\pi(z, dw)\mu^\pi(dz) \\
&= \mathbb{E}_{\mu_\pi}\tilde{M}^\pi V(x).
\end{aligned}$$

Since $\hat{M}_m^\pi V$ are uniformly bounded, we have

$$\mathbb{E}_{\mu_\pi} \hat{M}_m^\pi V \to \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V(x). \tag{22}$$

By the triangle inequality, we have

$$||\hat{M}_m^\pi V - M^\pi V||_\infty \;\; \leq \;\; ||\hat{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V)||_\infty + ||\mathbb{E}_{\mu_\pi} \tilde{M}^\pi V - M^\pi V||_\infty.$$

We note that the first term $||\hat{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V||_\infty$ is random and the second term $||\mathbb{E}_{\mu_\pi} \tilde{M}^\pi V - M^\pi V||_\infty$ is deterministic. Therefore, it suffices to show the first term converges to 0 in the mean and the last term converges to 0 separately.

Define

$$\bar{M}_m^\pi V(x) = \sum_{i=0}^{m-1} \frac{\tilde{k}(X_i, x)}{m} (C^\pi(X_i, X_{i+1}) + \gamma V(X_{i+1})). \tag{23}$$

Again it is easy to see that $\bar{M}_m^\pi V(x) \to \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V(x)$ almost surely. In turn, $\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V(x) \to \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V(x)$ pointwise. Since $\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V(x)$ has uniform Lipschitz continuous constant, the sequence is equicontinuous. As a result, we have $\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V \to \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V$ uniformly. We write

$$\begin{aligned} ||\hat{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V||_\infty \;\; \leq \;\; & ||\hat{M}_m^\pi V - \bar{M}_m^\pi V||_\infty + ||\bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V||_\infty \\ & + ||\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V||_\infty. \end{aligned}$$

Note $||\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V||_\infty$ is deterministic and $||\mathbb{E}_{\mu_\pi} \bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi} \tilde{M}^\pi V||_\infty \to 0$.

By following a similar argument as in lemma 1 of Ormoneit & Sen (2002), we have for any fixed bandwidth $b > 0$, the sequence $\sqrt{m}(\bar{M}_m^\pi V(x) - \mathbb{E}_{\mu^\pi} \bar{M}_m^\pi V(x)])$ converges in distribution to a Gaussian process on $C([b, 1-b]^d)$. Then, by theorem 3.2 of Rust (1997),

$$\mathbb{E}_{\mu^\pi} \sqrt{m} ||\bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi} (\bar{M}_m^\pi V)||_\infty \leq \sqrt{\frac{\pi}{2}} (1 + d\sqrt{\pi} C) K_{\tilde{k}} ||C^\pi + \gamma V||_\infty, \tag{24}$$

where C is a constant and $K_{\tilde{k}}$ is the Lipschitz constant of $\tilde{k}$.

For fixed y, by lemma 4.1 we have

$$\begin{aligned}
|\tilde{k}(x,y) - \tilde{k}(x',y)| &= \frac{1}{\int_{[0,1]^d} K(\frac{||u-y||}{b})\mu^\pi(du)} |K(\frac{||x-y||}{b}) - K(\frac{||x'-y||}{b})| \\
&\leq \frac{\Gamma(d/2)K_k}{b^d 2\pi^{d/2}\nu_{d-1}\underline{K}_\pi} |\frac{||x-y||}{b} - \frac{||x'-y||}{b}| \\
&\leq \frac{\Gamma(d/2)K_k}{b^{d+1} 2\pi^{d/2}\nu_{d-1}\underline{K}_\pi} ||x-x'||.
\end{aligned}$$

Hence, we have

$$\mathbb{E}_{\mu^\pi} ||\bar{M}_m^\pi V - \mathbb{E}_{\mu_\pi}(\bar{M}_m^\pi V)||_\infty \leq \frac{(1+d\sqrt{\pi}C)\Gamma(d/2)K_k}{b^{d+1}\sqrt{m}2\sqrt{2}\pi^{(d-1)/2}\nu_{d-1}\underline{K}_\pi} ||C^\pi + \gamma V||_\infty.$$

We write

$$\begin{aligned}
|\hat{M}_m^\pi V(x) - \bar{M}_m^\pi V(x)| &= |\tilde{M}_m^\pi V(x) - \bar{M}_m^\pi V(x)| \\
&= \sum_{i=0}^{m-1} \frac{\tilde{k}(X_i,x)}{\sum_{i=0}^{m-1}\tilde{k}(X_i,x)} (C^\pi(X_i,X_{i+1}) + \gamma V(X_{i+1}))(1 - \frac{1}{m}\sum_{i=0}^{m-1}\tilde{k}(x_i,x))
\end{aligned}$$

Hence, by the proof of corollary to theorem 3.4 in Rust (1997),

$$||\hat{M}_m^\pi V - \bar{M}_m^\pi V||_\infty \leq ||C^\pi + \gamma V||_\infty \sup_{x\in[0,1]^d} |1 - \frac{1}{m}\sum_{i=0}^{m-1}\tilde{k}(x_i,x)|,$$

and

$$\mathbb{E}_{\mu^\pi} \sup_{x\in[0,1]^d} |1 - \frac{1}{m}\sum_{i=0}^{m-1}\tilde{k}(x_i,x)| \leq \frac{C' K_{\tilde{k}}}{\sqrt{m}} \leq \frac{C'\Gamma(d/2)K_k}{b^{d+1}\sqrt{m}2\pi^{d/2}\nu_{d-1}\underline{K}_\pi},$$

where C' is a constant. Therefore,

$$\mathbb{E}_{\mu^\pi} ||\hat{M}_m^\pi V - \mathbb{E}_{\mu_\pi}\tilde{M}^\pi V||_\infty \to 0$$

as $b^{d+1}\sqrt{m} \to \infty$.

Let $h(x) = M^\pi V(x) = \int_{[0,1]^d} P^\pi(x, dy)(C^\pi(x, y) + \gamma V(y))$ and K_h be the Lipschitz constant of h. Then,

$$\begin{aligned}
||\mathbb{E}_{\mu_\pi}\tilde{M}^\pi V - M^\pi V||_\infty &= \sup_{x\in[0,1]^d}\left|\int_{[0,1]^d}\tilde{k}(u,x)(h(u)-h(x))\mu^\pi(du)\right| \\
&\leq \sup_{x\in[0,1]^d}\int_{[0,1]^d}\tilde{k}(u,x)|h(u)-h(x)|\mu^\pi(du) \\
&\leq \sup_{x\in[0,1]^d} K_h\underline{K}_\pi\frac{\Gamma(d/2)}{b^d 2\pi^{d/2}\nu_{d-1}}\bar{K}^\pi\int_{[0,1]^d}K(\frac{||u-x||}{b})||u-x||du \\
&\leq K_h\underline{K}_\pi\frac{\Gamma(d/2)b^{d+1}}{b^d 2\pi^{d/2}\nu_{d-1}}\bar{K}^\pi\int_{\mathbb{R}^d}K(||u||)||u||du \\
&\leq K_h\underline{K}_\pi\frac{\Gamma(d/2)b}{2\pi^{d/2}\nu_{d-1}}\bar{K}^\pi\int_0^1 r^{d-1}K(r)rdr \\
&\leq \frac{bK_h\bar{K}^\pi\nu_d}{\underline{K}^\pi\nu_{d-1}}.
\end{aligned}$$

Hence, $||\mathbb{E}_{\mu_\pi}\tilde{M}^\pi V - M^\pi V||_\infty \to 0$ as $b(m) \to 0$.

Since it is easy to check that $\hat{M}_m^\pi$ defines a contraction and let $\hat{V}_m^\pi = \hat{M}_m^\pi\hat{V}_m^\pi$ (as defined in (16)), then

$$\begin{aligned}
||\hat{V}_m^\pi - V^\pi||_\infty &\leq ||\hat{M}_m^\pi\hat{V}_m^\pi - \hat{M}_m^\pi V^\pi||_\infty + ||\hat{M}_m^\pi V^\pi - V^\pi||_\infty \\
&\leq \gamma||\hat{V}_m^\pi - V^\pi||_\infty + ||\hat{M}_m^\pi V^\pi - M^\pi V^\pi||_\infty.
\end{aligned}$$

Hence, we have

$$\mathbb{E}_{\mu_\pi}||\hat{V}_m^\pi - V^\pi||_\infty \leq \frac{1}{1-\gamma}\mathbb{E}_{\mu_\pi}||\hat{M}_m^\pi V^\pi - M^\pi V^\pi||_\infty. \tag{25}$$

This implies that $\mathbb{E}_{\mu_\pi}||\hat{V}_m^\pi - V^\pi||_\infty \to 0$. ∎

Corollary 4.1 (Convergence in the mean of kernel-based API) *Suppose assumptions*

in theorem 4.1 hold for all policies $\pi \in \Pi$. Theorem 2.1 applies to the kernel-based approximate algorithm in figure 2.

The proof is omitted since it is a direct application of theorems 4.1 and 2.1. It is worth noting that the matrix inversion in step 2 of the algorithm in figure 2 demands $O(m^3)$ computational complexity, which can be burdensome for large m. Hence, we consider a value iteration type updating rule ($O(m^2)$ computational complexity) for policy value function estimates, which can be written compactly using matrix notation:

$$\hat{V}^{\pi}_{m,k+1} = \hat{P}^{\pi}(C^{\pi} + \gamma \hat{V}^{\pi}_{m,k}) \tag{26}$$

where $\hat{V}^{\pi}_{m,k}$ and $\hat{V}^{\pi}_{m,k+1}$ are the old and new policy value function estimates respectively. Since the random operator $\hat{M}^{\pi}$ is a contraction operator, by the Banach fixed point theorem the updating in (26) converges to a unique fixed point that satisfies equation (14), which is $\hat{V}^{\pi}_{m}$. The algorithm in figure 3 that uses the updating in equation (26) is a type of hybrid value/policy iteration algorithm.

We now analyze the convergence of the hybrid algorithm.

Corollary 4.2 (Convergence in the mean of the hybrid kernel-based hybrid API) *Suppose assumptions in theorem 4.1 hold for all policies $\pi \in \Pi$. Theorem 2.1 applies to the hybrid kernel-based approximate algorithm in figure 3 with policy evaluation stopping criterion that satisfies*

$$||\hat{V}^{\pi}_{m,k+1} - \hat{V}^{\pi}_{m,k}||_{\infty} \leq \frac{1-\gamma}{2\gamma}\epsilon_n. \tag{27}$$

Proof:

In each inner loop, first determine m such that $\mathbb{E}_{\mu_\pi}||\hat{V}^{\pi}_{m} - V^{\pi}||_{\infty} \leq \frac{\epsilon_n}{2}$. Since we have almost surely

$$\sup_{x \in \{x_i\}_{i=0}^{m-1}} |\hat{V}^{\pi}_{m,k}(x) - \hat{V}^{\pi}_{m}(x)| \to 0,$$

Figure 3: Hybrid value/policy iteration with kernel smoothing (KSHPI)

Step 0: Initialization:

Step 0a. Set the initial policy π_0.

Step 0b. Set the kernel function K.

Step 0c. Set the iteration counter $n = 0$.

Step 1: Do for $n = 0, \ldots, N$,

Step 1a. Set the iteration counter $l = 0$.

Step 1b. Set the initial state x_0^n.

Step 1c: Do for $j = 0, \cdots, m$:

Step 1c.1: Set $u_j^n = \pi_n(x_j^n)$ and draw randomly or observe W_{j+1} from the stochastic process,

Step 1c.2: Compute $x_j^{n,\pi} = S^{M,\pi}(x_j^n, u_j^n)$ (store also) and $x_{j+1}^n = S^M(x_j^{n,\pi}, u_j^n, W_{j+1})$.

Step 1d. Initialize $\hat{v}_0$ where $\hat{v}_0$ is a vector of dimensionality m with element $\hat{v}_0(x_i^{n,pi})$ for $i = 1 \cdots m$.

Step 1e. Let C^π be a vector of dimensionality m with i-th entry $C^\pi(x_{i-1}^{n,\pi}, x_i^{n,\pi})$ for $i = 1 \cdots m$.

Step 1f. Let $\hat{P}^\pi$ be a matrix of dimensionality $m \times m$ with i, j-th entry $k(x_{i-1}^{n,\pi}, x_j^{n,\pi})$ for $i, j \in \{1 \cdots m\}$.

Step 2: Do for $l = 0, \ldots, L-1$:

Step 2.1: $\hat{v}_{l+1} = c^\pi + \gamma \hat{P}^\pi \hat{v}_l$.

Step 3: Let $\hat{v}^n(x) = \sum_{i=0}^{m-1} k(x_i^{n,\pi}, x) \left(C^\pi(x_i^{n,\pi}, x_{i+1}^{n,\pi}) + \gamma \hat{v}_L(x_{i+1}^{n,\pi}) \right)$

Step 4: Update the policy:

$$\pi_{n+1}(x) = \arg\max_{u \in \mathcal{U}} \{C(x, u) + \gamma \hat{v}^n(x^u)\}.$$

Step 5: Return the policy π_{N+1}.

$||\hat{V}_{m,k}^\pi - \hat{V}_m^\pi||_\infty \to 0$ almost surely. With the stopping criterion (27), we have almost surely,

$$||\hat{V}_{m,K+1}^\pi - \hat{V}_m^\pi||_\infty \leq \frac{1}{1-\gamma} ||\hat{V}_{m,K+1}^\pi - \hat{V}_{m,K}^\pi||_\infty \leq \frac{\epsilon_n}{2}.$$

Taking expectations, we obtain $\mathbb{E}_{\mu_\pi} ||\hat{V}_{m,k+1}^\pi - V^\pi||_\infty \leq \epsilon_n$. Hence, theorem 2.1 applies. ∎

5 Finite horizon approximation using recursive kernel smoothing and local polynomials

In this section, we propose an algorithm that applies kernel smoothing to finite horizon rewards to approximate policy value function for the infinite horizon problems. The logic behind this idea is very simple. We approximate the infinite horizon post-decision policy value function V^π with its k-th finite horizon policy value function

$$V_k^\pi(x) = \mathbb{E}\left\{\sum_{t=0}^{k} \gamma^t C^\pi(x_t, x_{t+1}) | x_0 = x\right\}. \tag{28}$$

It is worth noting that x's are all post-decision states. Let $C_{\max}$ be the constant that bounds the contribution function C^π and $\epsilon > 0$. There exists $k \in \mathbb{N}$ such that for all $x \in \mathcal{X}^\pi$,

$$|V_k^\pi(x) - V^\pi(x)| = \left|\mathbb{E}\left\{\sum_{t=k+1}^{\infty} \gamma^t C^\pi(x_t, x_{t+1}) | x_0 = x\right\}\right| \leq \frac{\gamma^{k+1}}{1-\gamma} C_{\max} < \epsilon.$$

As a result, for each fixed policy π we can approximate the post-decision policy value function arbitrarily well with a finite horizon policy value function. Hence, we can apply kernel regression techniques such as Nadaraya-Watson estimate and local polynomial regression (recursive or non-recursive) to estimate the finite horizon policy value function with data (x_i, y_i) where

$$y_i = \sum_{t=i}^{i+k} \gamma^{t-i} C^\pi(x_t, x_{t+1}). \tag{29}$$

The details of the kernel smoothing approximate policy iteration algorithm with finite horizon approximation are shown in figure 4.

In order to apply theorem 2.1 to show the convergence of the algorithm, it suffices to use the kernel regression estimation techniques that satisfy strong uniform convergence to the finite horizon policy value function. We consider the following generic form of the kernel smoothing estimate

$$\hat{m}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} Y_i K(\frac{x - X_i}{h}) \tag{30}$$

Figure 4: Approximate policy iteration with kernel smoothing and finite horizon rewards (KSFHRAPI)

Step 0: Initialization:

Step 0.1: Set the initial policy π_0 and the iteration counter $n = 0$.

Step 0.2: Set the kernel function K and fixed horizon number $k \geq 1$.

Step 1: Do for $n = 0, \ldots, N$,

Step 1.1: Set the iteration counter $m = 0$ and the initial state x_0^n.

Step 2: Do for $m = 0, \ldots, M$:

Step 2.1: If $m = 0$, do the following:

Step 2.1.1: Set the initial state x_0^n, initial kernel estimate $\hat{f}_{-1}^n = 0$ and $\hat{v}_m = 0$

Step 2.1.2: Draw randomly or observe $W_1, \cdots, W_{k+1}$ from the stochastic process.

Step 2.1.3: Do for $j = 0, \cdots, k$:

Step 2.1.3.a: Set $u_j^n = \pi_n(x_j^n)$.

Step 2.1.3.b: Compute $x_j^{n,\pi} = S^{M,\pi}(x_j^n, u_j^n)$ and $x_{j+1}^n = S^M(x_j^{n,\pi}, u_j^n, W_{j+1})$.

Step 2.2: If $m = 1, \ldots, M$, do the following:

Step 2.2.1: Draw randomly or observe W_{m+k+1} from the process.

Step 2.2.2 Set $u_{m+k}^n = \pi_n(x_{m+k_n}^n)$.

Step 2.2.3 Compute $x_{m+k}^{n,\pi} = S^{M,\pi}(x_{m+k}^n, u_{m+k}^n)$ and $x_{m+k+1}^n = S^M(x_{m+k}^n, u_{m+k}^n, W_{m+k+1})$.

Step 2.2.4 Compute $u_{m+k+1}^n = \pi_n(x_{m+k+1}^n)$ and $x_{m+k+1}^{n,\pi} = S^{M,\pi}(x_{m+k+1}^n, u_{m+k+1}^n)$.

Step 2.3: Compute $\hat{v}_m = \sum_{j=0}^{k-1} \gamma^j C^\pi(x_{m+j}^{n,\pi}, x_{m+j+1}^{n,\pi})$.

Step 2.4: Apply kernel methods with sparsification and compute kernel estimate $\hat{f}_m^n$ with $(x_j^{n,\pi})_{j=0}^m$ and $(\hat{v}_j)_{j=0}^m$

or recursively with $\hat{f}_{m-1}^n$, $x_m^{n,\pi}$ and $\hat{v}_m$.

Step 3: Update the policy:

$$\pi_{n+1}(x) = \arg\max_{u \in \mathcal{U}} \{C(x,u) + \gamma \hat{f}_M^n(x^u)\}.$$

Step 4: Return the policy π_{N+1}.

---

where h is a bandwidth and $K : \mathbb{R}^d \to \mathbb{R}$ is a kernel function. With proper choice of the kernel function K, the generic form includes most kernel-based nonparametric estimators such as kernel density estimators of density functions, Nadaraya-Watson estimators of the regression function, and local polynomial estimators.

There are numerous strong uniform convergence results under various technical conditions for the generic form of kernel regression estimates. The most common convergence results can be found for the case when the data are sampled with identical and independent distribution (e.g. Mack & Silverman (1982)). However, these results are less relevant because dependence in data structure arises naturally in Markov decision process problems. Peligrad (1992), Masry (1996), Fan & Yao (2003) provide strong uniform convergence results under dependent data structure assumptions. Hansen (2008) generalizes previous results and proves strong uniform convergence for kernel regression estimation under milder conditions such as stationary strong mixing data with infinite support and kernels with unbounded support. Therefore, to prove convergence of the policy iteration algorithm, we consider the same set of assumptions and check whether they are satisfied in our setting of the problem class.

The following assumption stipulates that the kernel function K is bounded, integrable and smooth e.g. being Lipschitz continuous on a truncated support or having a bounded derivative with an integrable tail. This assumption admits a lot of commonly used kernels, such as the (higher order) polynomial kernels and the (higher order) Gaussian kernels (see Hansen (2008)).

Assumption 5.1 (Kernel) *Let K be a kernel function. $|K(x)| \leq K_1$ for all x and $\int_{\mathbb{R}^d} |K(x)|dx \leq K_2 < \infty$. Furthermore suppose for some $K_3, C < \infty$, either $|K(x)| = 0$ for all $|x| \geq K_3$ and $\forall x, x' \in \mathbb{R}^d$*

$$|K(x) - K(x')| \leq C||x - x'||,$$

or $\frac{\partial}{\partial x}K(x) \leq K_3$ and for some $\nu > 1$, $\frac{\partial}{\partial x}K(x) \leq K_3||x||^\nu$ for $||x|| > C$.

Before presenting the assumption on the chain, we first present the definition of strong

mixing (α mixing) condition.

Definition 5.1 (Strong (α) mixing) *Let $(X_t)_{t=0}^{\infty}$ be a sequence of random variables. The strong mixing coefficient is defined as*

$$\alpha(n) = \sup\{|\mathbb{P}(A \cap B) - \mathbb{P}(A)\mathbb{P}(B)| : k \geq 0, A \in \mathcal{F}_k, B \in \mathcal{F}_{k+n}^{\infty}\}, \tag{31}$$

where $\mathcal{F}_k = \sigma(X_0, \cdots, X_k)$ and $\mathcal{F}_{k+n}^{\infty} = \sigma(X_{k+n}, \cdots)$. The process X is strong mixing if $\alpha(n) \to 0$ as $n \to \infty$.

The following assumption specifies the technical dependence structures of the data such as stationarity and the strong mixing property with a certain decay rate. It also requires boundedness of the marginal density function, joint density and conditional expectations of Y.

Assumption 5.2 (Data structure)

a. *The data sequence (X_n, Y_n) is a strictly stationary and strong process mixing with mixing coefficients $\alpha(n)$ that satisfy $\alpha(n) \leq An^{-\beta}$ where $A < \infty$ and for some $s > 2$ $\mathbb{E}|Y_0|^s < \infty$ and $\beta > \frac{2s-2}{s-2}$.*

b. *X_n has marginal density f such that $\sup_x f(x) \leq B_1 < \infty$ $\sup_x \mathbb{E}[|Y_0|^s | X_0 = x] f(x) \leq B_2 < \infty$.*

c. *There is some $j^* < \infty$ such that for all $j > j^*$ $\sup_{x_0, x_j} \mathbb{E}[|Y_0 Y_j| | X_0 = x_0, X_j = x_j] f_j(x_0, x_j) \leq B_3 < \infty$ where $f_j(x_0, x_j)$ denotes the joint density of X_0, X_j.*

Now we check if the above assumption on data structure is satisfied in our problem setting of MDP. Suppose we initialize the chain according to its invariant measure μ_π. Then the data sequence is strictly stationary. The following lemma 5.1 exhibits that the data is strong mixing.

Lemma 5.1 (Strong mixing of the data) *The data in figure 4 for policy evaluation is a strong mixing process given the post decision states for a fixed policy π evolves according to a positive Harris chain.*

Proof:

Let $Z_i = (X_i, X_{i+1}, \cdots, X_{i+k})$. Since X is a Markov chain, Z is a Markov chain with transition kernel

$$P^Z(z, A) = \prod_{i=1}^{k} 1_{x_i \in A_{i-1}} P^X(x_k, A_k),$$

where $z = (x_0, x_1, \cdots, x_k)$ and $A = A_0 \times A_1 \cdots \times A_k$. Since X is positive Harris by assumption, it has an invariant measure π^X. We claim Z has an invariant measure

$$\mu^Z(dz) = \mu^X(dx_0)P^X(x_0, dx_1) \cdots P^X(x_{k-1}, dx_k)$$

where $z = (x_0, \cdots, x_k)$. For simplicity of presentation, consider the case $k = 1$. Then we have

$$\mu^Z(A \times B) = \int_{A \times B} \mu^X(dx_0)P^X(x_0, dx_1) = \int_A \mu^X(dx_0)P^X(x_0, B) \tag{32}$$

and

$$\begin{aligned} \int \mu^Z(dz)P^Z(z, A \times B) &= \int \mu^X(dx_0)P^X(x_0, dx_1)1_{x_1 \in A}P(x_1, B) \\ &= \int \mu^X(dx_1)1_{x_1 \in A}P(x_1, B) \\ &= \int_A \mu^X(dx_1)P^X(x_1, B). \end{aligned}$$

Hence, μ^Z is the invariant measure for Z. Consider $||(P^Z)^n(z, \cdot) - \mu^Z(\cdot)||_{TV}$ where $||\cdot||_{TV}$ is the total variation norm. We have

$$\begin{aligned} (P^Z)^n(z, \prod_{i=0}^{k} A_k) &= \mathbb{P}(X_n \in A_0, \cdots, X_{n+k} \in A_k | X_0 = x_0, \cdots, X_k = x_k) \\ &= \mathbb{P}(X_n \in A_0, \cdots, X_{n+k} \in A_k | X_k = x_k) \end{aligned}$$

where $z = (x_0, \cdots, x_k)$. Since (X) is positive Harris, as $n \to \infty$

$$\begin{aligned} ||(P^Z)^n(z,\cdot) - \mu^Z(\cdot)||_{TV} &= \sup_{\prod_{i=0}^k A_k} |\mathbb{P}(X_n \in A_0, \cdots, X_{n+k} \in A_k | X_k = x_k) \\ &\quad - \int_{A_0 \times \cdots A_{k-1}} \mu^X(dx_0) P^X(x_0, dx_1) \cdots P^X(x_{k-2}, dx_{k-1}) P^X(x_{k-1}, A_k)| \\ &\quad \to 0 \end{aligned}$$

for all $x_k \in \mathcal{X}^\pi$. Therefore, Z is also positive Harris. As a result, by theorem A of Athreya & Pantula (1986), Z is strong mixing, meaning $\alpha^Z(n) \to 0$ as $n \to \infty$. We also note that there is a measurable function $g : \prod_{j=0}^k \mathcal{X} \to \mathcal{X} \times \mathbb{R}$ such that $g(Z_i) = (X_i, Y_i)$. We claim $g(Z)$ is strong mixing. First, we note that $\mathcal{F}^{g(Z)} \subset \mathcal{F}^Z$ since g is measurable. Then,

$$\begin{aligned} \alpha^{g(Z)}(n) &= \sup\{|\mathbb{P}(A \cap B) - \mathbb{P}(A)\mathbb{P}(B)| : k \geq 0, A \in \mathcal{F}_k^{g(Z)}, B \in (\mathcal{F}^{g(Z)})_{k+n}^\infty\} \\ &\leq \sup\{|\mathbb{P}(A \cap B) - \mathbb{P}(A)\mathbb{P}(B)| : k \geq 0, A \in \mathcal{F}_k^{Z}, B \in (\mathcal{F}^{Z})_{k+n}^\infty\} \\ &= \alpha^Z(n). \end{aligned}$$

∎

Since Y_n's are bounded in discounted problems, we can take $s = \infty$ and let $\beta > 2$. In order to have uniform convergence, further restriction has to be put on β i.e. $\beta > 1 + d + \frac{d}{q}$ for some $q > 0$ (see Hansen (2008)). In addition, the boundedness condition on conditional expectations are reduced to the boundedness of density and joint density functions. We just need to assume the invariant density of the chain exists and is bounded. As a result, the joint density of X_0, X_j is bounded due to previous assumptions of compact state space and continuous transition probability function. Therefore, the assumption 5.2 is reduced to the following for our problem class.

Assumption 5.3 *The Markov chain X^π is positive Harris that admits a bounded invariant density. Assume the chain is initialized according to its invariant measure and its mixing coefficients $\alpha(n)$ satisfy $\alpha(n) \leq An^{-\beta}$ where $A < \infty$ and $\beta > 1 + d + \frac{d}{q}$ for some $q > 0$.*

In many applications, the chains are initialized at some fixed state or some arbitrary distribution but not at their invariant distribution. In this case, the data is nonstationary. Kristensen (2009) extends the results of Hansen (2008) to heterogeneously dependent (non-stationary) data. It is not hard to believe that a positive Harris chain is not sufficient to guarantee uniform convergence and stronger assumption on the chain has to be imposed. As a result, assumption 5.3 is modified to the following:

Assumption 5.4 *The Markov chain X^π satisfies the strong Doeblin condition: there exist $n > 1$ and $\rho \in (0,1)$ such that $p_n(y|x) \geq \rho f(y)$ where f is the invariant density of the chain and $p_n(y|x)$ is the n-th transition density defined as*

$$p_n(y|x) = \int_{\mathcal{X}^\pi} p(y|z)p_{n-1}(z|x)dz \tag{33}$$

for $n = 1, 2, \cdots$. The transition density $p(y|x)$ is $r \geq 1$ times differentiable with $\frac{\partial^r}{\partial y^r}p(y|x)$ being uniformly continuous for all x. $||x||^q f(y)$ is bounded for some $q \geq d$.

Lemma 5.2 *Suppose assumptions 5.1 and either 5.3 or 5.4 hold. Let the data (X, Y) be collected as in equation (29), $V_k^\pi(x)$ be defined in equation (28) and $\hat{V}_k^\pi$ defined in the same way as $\hat{m}$ in equation (30). Then, we have $\sup_x |\hat{V}_k^\pi(x) - V_k^\pi(x)| \to 0$ almost surely.*

We provide a sketch of the proof. If assumption 5.3 holds, then the chain is stationary. The proof of the lemma follows the same line as the proof of theorem 3 of Hansen (2008) with proper bandwidth selection. Let Z be defined as in the proof of lemma 5.1. If assumption 5.4 holds, Z also satisfies the strong Doeblin condition since X satisfies the Doeblin condition. Therefore, (X, Y) is embedded in a Markov chain satisfying the Doeblin criterion. Then, theorem 3 in Kristensen (2009) can be applied to obtain strong uniform convergence rates for the kernel density estimator of the joint invariant density of (X, Y) $(f(x, y))$ and invariant density of X $(f(x))$ respectively. This can in turn be used to obtain convergence rates for the kernel estimator of

$$V_k^\pi(x) = \int_{\mathcal{X}^\pi} yp(y|x)dx = \int_{\mathcal{X}^\pi} yf(y, x)/f(x)dx$$

by following a similar argument as in the proof of theorem 3 of Kristensen (2009).

Corollary 5.1 (Convergence in mean of API with finite horizon approximation) *Suppose assumptions in lemma 5.2 hold for all policies $\pi \in \Pi$. Theorem 2.1 applies to the kernel-based approximate policy iteration algorithm with finite horizon approximation in figure 4.*

The proof is omitted since it is a direct application of lemma 5.2. Compared to the non-recursive techniques, we are more interested in the recursive regression estimation where estimates are updated as additional sample observations are obtained over time, because this situation happens naturally in Markov decision processes. Moreover, from a practical point of view, recursive estimates have the tremendous advantage of saving in computational time and memory, because the updating of the estimates is independent of the previous sample size. Several potential candidates of recursive kernel estimates have been proposed to be used with our algorithm. Révész (1973) applies Robbins-Monro procedure to construct a stochastic approximation algorithm with recursive kernel smoothing updates:

$$\hat{m}_n(x) = m_{n-1}(x) + \frac{1}{n} W_n(x)$$

and

$$W_n(x) = \frac{1}{h_n} Y_n K(\frac{X_n - x}{h_n}) - \frac{1}{h_n} K(\frac{X_n - x}{h_n}) \hat{m}_{n-1}(x)$$

where K is a kernel function and h_n is the bandwidth converging to 0. Revesz (1977), Mokkadem et al. (2008) prove uniform convergence (weak and strong) of the algorithm under i.i.d. sampling scheme.

Vilar-Fernández & Vilar-Fernández (1998) proposes recursive local polynomial fitting for regression function estimation by minimizing the following kernel weighted loss function (for simplicity of presentation, we consider the univariate case for input variable):

$$\hat{L}_n(\beta) = \sum_{i=1}^{n} (Y_i - \sum_{j=0}^{p} \beta_j (X_i - x)^j)^2 w_{n,i}$$

where $\beta = (\beta_0, \cdots, \beta_p)^T$, $w_{n,i} = \frac{1}{nh_n} K_n(\frac{X_i - x}{h_n})$. Recall that local polynomial fitting is a special case of the weighted least squares method. K is a kernel function and h_n the bandwidth. Then, the recursive updating for the linear parameter β is

$$\hat{\beta}_{n+1} = \hat{\beta}_n + w_{n+1,n+1}(Y_{n+1} - X_{n+1}\hat{\beta}_n) S_{n+1}^{-1} X_{n+1}$$

and

$$S_{n+1}^{-1} = (1 + \frac{1}{n}) \left(S_n^{-1} - \frac{\frac{1}{h_{n+1}} K(\frac{X_{n+1} - x}{h_{n+1}}) S_n^{-1} x_{n+1} x_{n+1}^T S_n^{-1}}{n + \frac{1}{h_{n+1}} K(\frac{X_{n+1} - x}{h_{n+1}}) x_{n+1} S_n^{-1} x_{n+1}^T} \right)$$

where $x_{n+1} = (1, X_{n+1} - x, \cdots, (X_{n+1} - x)^p)^T$. Furthermore, Vilar-Fernández & Vilar-Fernández (2000) provides strong uniform convergence results of the algorithm under dependent data assumption.

6 Conclusion

In this paper, we use different kernel smoothing techniques to propose three different online, on-policy approximate policy iteration algorithms for infinite-horizon Markov decision process problems with continuous state and action spaces. We provide rigorous convergence analysis for the algorithms under a variety of technical assumptions. However, kernel smoothing suffers from the curse of dimensionality of input variables by nature and is often slow for problems of more than 7 dimensions when applied naively. One line of future research is to incorporate methods that can handle the curse of dimensionality into our kernel algorithms. One example of such methods can be found in Goutte & Larsen (2000), which automatically adjusts the importance of different dimensions by adapting the input metric used in multivariate regression and minimizing a cross-validation estimate of the generalization error.

Acknowledgement

The first author would like to thank Professor Erhan Cinlar, Professor Philippe Rigollet, Professor Ramon Van Handel, Professor Dennis Kristensen, Yang Feng and Ke Wan for many inspiring discussions. This work was supported by Castle Lab at Princeton University and partially supported by AFOSR grant FA9550-06-1-0496.

References

Athreya, K. & Pantula, S. (1986), 'Mixing properties of Harris chains and autoregressive processes', *Journal of Applied Probability* **23**(4), 880–892.

Bertsekas, D. & Shreve, S. (1978), *Stochastic Optimal Control: The Discrete-Time Case*, Academic Press, Inc. Orlando, FL, USA.

Boyan, J. (1999), Least-squares temporal difference learning, *in* 'Proceedings of the Sixteenth International Conference on Machine Learning', pp. 49–56.

Bradtke, S. & Barto, A. (1996), 'Linear Least-Squares algorithms for temporal difference learning', *Machine Learning* **22**(1), 33–57.

Engel, Y., Mannor, S. & Meir, R. (2004), 'The kernel recursive least-squares algorithm', *IEEE Transactions on Signal Processing* **52**(8), 2275–2285.

Fan, J. & Yao, Q. (2003), *Nonlinear time series: nonparametric and parametric methods*, Springer Verlag.

Farahmand, A., Ghavamzadeh, M., Szepesvári, C. & Mannor, S. (2009*a*), Regularized fitted Q-iteration for planning in continuous-space Markovian decision problems, *in* 'Proceedings of the 2009 conference on American Control Conference', Institute of Electrical and Electronics Engineers Inc., The, pp. 725–730.

Farahmand, A., Ghavamzadeh, M., Szepesvári, C. & Mannor, S. (2009*b*), 'Regularized policy iteration', *Advances in Neural Information Processing Systems* (21), 441–448.

Goutte, C. & Larsen, J. (2000), 'Adaptive metric kernel regression', *The Journal of VLSI Signal Processing* **26**(1), 155–167.

Hansen, B. (2008), 'Uniform convergence rates for kernel estimation with dependent data', *Econometric Theory* **24**(03), 726–748.

Judd, K. (1998), *Numerical Methods in Economics*, MIT Press Cambridge, MA.

Jung, T. & Polani, D. (2006), 'Least squares SVM for least squares TD learning', *Frontials in Artificial Intellegence and Applications* **141**, 499.

Kimeldorf, G. & Wahba, G. (1971), 'Some results on Tchebycheffian spline functions', *Journal of Mathematical Analysis and Applications* **33**(1), 82–95.

Kristensen, D. (2009), 'Uniform Convergence Rates Of Kernel Estimators With Heterogeneous Dependent Data', *Econometric Theory* **25**(05), 1433–1445.

Lagoudakis, M. & Parr, R. (2003), 'Least-Squares Policy Iteration', *Journal of Machine Learning Research* **4**(6), 1107–1149.

Loth, M., Davy, M. & Preux, P. (2007), 'Sparse temporal difference learning using LASSO', *In IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* .

Luque, J. (1987), A nonlinear proximal point algorithm, *in* '26th IEEE Conference on Decision and Control', Vol. 26, pp. 816–817.

Ma, J. & Powell, W. (2010), 'Convergence Analysis of On-Policy LSPI for Multi-Dimensional Continuous State and Action-Space MDPs and Extension with Orthogonal Polynomial Approximation', *Submitted to SIAM Journal of Control and Optimization* .

Mack, Y. & Silverman, B. (1982), 'Weak and strong uniform consistency of kernel regression estimates', *Probability Theory and Related Fields* **61**(3), 405–415.

Masry, E. (1996), 'Multivariate local polynomial regression for time series: uniform strong consistency and rates', *Journal of Time Series Analysis* **17**(6), 571–600.

Meyn, S. & Tweedie, R. (1993), *Markov chains and stochastic stability*, Springer, New York.

Mokkadem, A., Pelletier, M. & Slaoui, Y. (2008), 'Revisiting R\'ev\'esz's stochastic approximation method for the estimation of a regression function', *working paper* .

Ormoneit, D. & Sen, Ś. (2002), 'Kernel-Based Reinforcement Learning', *Machine Learning* **49**(2), 161–178.

Peligrad, M. (1992), 'Properties of uniform consistency of the kernel estimators of density and regression functions under dependence assumptions', *Stochastics An International Journal of Probability and Stochastic Processes* **40**(3), 147–168.

Powell, W. B. (2007), *Approximate Dynamic Programming: Solving the curses of dimensionality*, John Wiley and Sons, New York.

Révész, P. (1973), 'Robbins-Monro procedure in a Hilbert space and its application in the theory of learning processes I', *I., Studia Sci. Math. Hungar* **8**, 391–398.

Revesz, P. (1977), 'How to apply the method of stochastic approximation in the non-parametric estimation of a regression function', *Statistics* **8**(1), 119–126.

Rust, J. (1997), 'Using randomization to break the curse of dimensionality', *Econometrica: Journal of the Econometric Society* pp. 487–516.

Smale, S. & Zhou, D. (2005), 'Shannon sampling II: Connections to learning theory', *Applied and Computational Harmonic Analysis* **19**(3), 285–302.

Sutton, R. & Barto, A. (1998), *Reinforcement Learning: An Introduction*, MIT Press Cambridge, MA.

Tadić, V. (2001), 'On the convergence of temporal-difference learning with linear function approximation', *Machine learning* **42**(3), 241–267.

Tsitsiklis, J. & Van Roy, B. (1996), 'Feature-based methods for large scale dynamic programming', *Machine Learning* **22**(1), 59–94.

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

Van Roy, B., Bertsekas, D., Lee, Y. & Tsitsiklis, J. (1997), A Neuro-Dynamic Programming Approach to Retailer Inventory Management, *in* 'Proceedings of the 36th IEEE Conference on Decision and Control, 1997', Vol. 4.

Vilar-Fernández, J. & Vilar-Fernández, J. (1998), 'Recursive estimation of regression functions by local polynomial fitting', *Annals of the Institute of Statistical Mathematics* **50**(4), 729–754.

Vilar-Fernández, J. & Vilar-Fernández, J. (2000), 'Recursive local polynomial regression under dependence conditions', *Test* **9**(1), 209–232.

Wu, Q., Ying, Y. & Zhou, D. (2006), 'Learning rates of least-square regularized regression', *Foundations of Computational Mathematics* **6**(2), 171–192.

Xu, X., Hu, D. & Lu, X. (2007), 'Kernel-based least squares policy iteration for reinforcement learning', *IEEE Transactions on Neural Networks* **18**(4), 973–992.