Kernel-Based Reinforcement Learning in Average–Cost Problems

Dirk Ormoneit and Peter Glynn

***Abstract*—Reinforcement learning (RL) is concerned with the identification of optimal controls in Markov decision processes (MDPs) where no explicit model of the transition probabilities is available. Many existing approaches to RL, including "temporal-difference learning", employ simulation-based approximations of the value function for this purpose. This procedure frequently leads to numerical instabilities of the resulting learning algorithm, especially if the function approximators used are parametric, such as linear combinations of basis functions or neural networks. In this paper, we propose an alternative class of RL algorithms which always produces stable estimates of the value function. In detail, we use "local averaging" methods to construct an approximate dynamic programming (ADP) algorithm. Nearest-neighbor regression, grid-based approximations, and trees can all be used as the basis of this approximation. We provide a thorough theoretical analysis of this approach and we demonstrate that ADP converges to a unique approximation in continuous-state average–cost MDPs. In addition, we prove that our method is consistent in the sense that an optimal approximate strategy is identified asymptotically. With regard to a practical implementation, we suggest a reduction of ADP to standard dynamic programming in an artificial finite-state MDP.**

***Index Terms*—Average–cost problem, dynamic programming, kernel smoothing, local averaging, Markov decision process (MDP), perturbation theory, policy iteration, reinforcement learning, temporal-difference learning.**

I. Introduction

We consider optimal control in Markov decision processes (MDPs) with continuous state-spaces and unknown transition probabilities. To approach this problem, one typically estimates the parameters of an explicit transition model from sample trajectories. The feasibility of this approach depends on the complexity of the system and on the amount of available training data. For continuous state-spaces, convergence may be very slow unless prior information is available to restrict the number of model parameters. An interesting alternative is to approximate the value function *directly* from the data, without explicitly modeling the transition probabilities. Instead, this approach relies on an *implicit* transition model in the form of simulated data, and it is referred to as "reinforcement learning" in the machine learning literature. The potential benefits of reinforcement learning are several. First, the value function may be better suited for estimation than the transition kernel. Second, even if the transition probabilities are known, the computational effort to derive the optimal policy from it via dynamic programming may become prohibitive in continuous state-spaces. An approximation of the value function, on the other hand, can sometimes be estimated much more efficiently from the training data. This approximate value function may then serve to construct an approximation of the optimal strategy.

A standard approach to reinforcement learning is "temporal-difference learning" [1], [2]. This method has been applied successfully to many discrete state-space problems using an explicit representation of the value function as a lookup table. However, a lookup table representation may be unsatisfactory in continuous or very large discrete state-spaces because of its poor ability to "generalize" to previously unseen data. To improve the generalization performance, neural networks have been suggested as approximators of the value function. A serious drawback of this approach is that stability and convergence properties of approximate temporal-difference methods are available only in special cases and examples are known where temporal-difference learning fails to converge [3]. To circumvent these shortcomings, a nonparametric approach to reinforcement learning has been recently proposed by Ormoneit and Sen [4]. Specifically, they suggest the use of kernel smoothers, a form of local averaging, to approximate the value function in finite-horizon and infinite-horizon discounted-cost problems. In this work, we establish stability and convergence results for a general class of reinforcement learning algorithms based on local averaging. Emphasis is on the application to average–cost MDPs.

The mathematical analysis of average–cost problems is typically more involved than the analysis of discounted problems, both in the case of known transition probabilities and for reinforcement learning [5]. For known transition probabilities, the optimal policy μ^* can be derived from solutions η^* and h^* to the *average cost optimality equation (ACOE)*

$$\eta^* + h^*(x) = \min_a \{c(x, a) + (\Gamma_a h^*)(x)\} \tag{1}$$

under rather weak conditions on the underlying MDP [6]. Here, $c(x, a)$ is the one-step cost using action a and Γ_a is the conditional expectation operator given a. In reinforcement learning, on the other hand, the transition probabilities are unknown so that Γ_a cannot be evaluated and consequently (1) cannot be employed to determine μ^*. In this work, we suggest using instead an approximate expectation operator $\hat{\Gamma}_{m,a}$ that can be

Manuscript received May 22, 2001; revised January 4, 2002. Recommended by Associate Editor L. Dai. The work of D. Ormoneit was carried out while he was at the Departments of Statistics and Computer Science, Stanford University, Stanford, CA 94305 USA, and was supported by a Grant from the Deutsche Forschungsgemeinschaft (DFG) as part of its Postdoctoral program, and by Grant ONR MURI N00014-00-1-0637.

D. Ormoneit is with Marshall Wace Asset Management, SW1H 9BU London, U.K. (e-mail: d.ormoneit@mwam.com).

P. Glynn is with the Department of Management Science, Stanford University, Stanford, CA 94305 USA.

Digital Object Identifier 10.1109/TAC.2002.803530.

estimated from a set of m sample transitions, $\mathcal{S}$, using local averaging. Special cases of local averaging include methods based on kernel-smoothers, nearest neighbor regression, grid-based approximations, and trees. Replacing Γ_a with $\hat{\Gamma}_{m,a}$ in (1), we obtain the *approximate average cost optimality equation (AACOE)*

$$\hat{\eta}_m + \hat{h}_m(x) = \min_a \left\{ c(x, a) + \left(\hat{\Gamma}_{m,a} \hat{h}_m \right)(x) \right\}. \quad (2)$$

A straightforward approach to reinforcement learning is, thus, to compute solutions $\hat{\eta}_m$ and $\hat{h}_m$ of (2) using the familiar policy iteration algorithm and to derive an approximate policy $\hat{\mu}_m$ based on these estimates. The convergence of policy iteration or alternative dynamic programming rules is a consequence of the special properties of local averaging. Moreover, due to the "self-approximating property" of the proposed method, this computation can be carried out in a finite-state framework which greatly simplifies the practical implementation.

Besides algorithmic considerations, the proposed method raises interesting questions from a statistical perspective. In detail, it seems desirable to characterize the approximation error of the quantities $\hat{\eta}_m$, $\hat{h}_m$, and $\hat{\mu}_m$. For this purpose, we interpret the approximate operator $\hat{\Gamma}_{m,a}$ as the true expectation operator in a "perturbed" Markov chain, and we relate differences in the average–costs to the perturbation $\hat{\Gamma}_{m,a} - \Gamma_a$. This argument reduces the approximation of η^* and h^* essentially to a nonlinear regression problem so that we can generalize existing asymptotic theory for local averaging. As a result, we obtain consistency of local averaging under general assumptions.

II. Previous Work

Existing literature on approximate dynamic programming (ADP) can be divided broadly into work in the areas of optimal control and reinforcement learning. In *optimal control*, where the transition probabilities are known, ADP serves as a computational tool that makes large-scale problems admissible for an approximate solution. One influential paper by Rust [7] describes an ADP algorithm that uses the transition density to weigh the samples generated from a simulator. Rust concludes that under suitable conditions it is possible to "break" the curse of dimensionality in the sense that the number of observations needed to achieve a prespecified accuracy depends subexponentially on the problem dimension. Gordon [8] follows a closely related approach, summarizing methods that lead to numerically stable approximations of the value function.

In *reinforcement learning*, where the transition probabilities are unknown, attention has focused on the asymptotic properties of the TD- and Q-learning algorithms. For example, in [9], Gordon notes that Q learning can be shown to be convergent to a unique fixed point under suitable conditions on the update operator. Baker [10] and Borkar [11] both employ a weighting function to identify the value function asymptotically based on a stochastic approximation approach. Munos and Moore [12] emphasize the continuous time and space aspects of solving MDPs and propose convergent numerical schemes based on variable resolution discretization. With regard to the mathematical techniques used in the proofs, the regeneration based on a pseudoatom was previously used in reinforcement learning in [13].

This paper builds on the idea of using of a weighting kernel to approximate the value function in a continuous state space. However, by contrast to Rust we consider this problem in a reinforcement rather than in an optimal control setting which is considerably more complicated because the unknown transition information must be determined implicitly during learning. This algorithm is fundamentally different from those proposed by Baker and Borkar because it is based on ADP rather than on stochastic approximation. In particular, we present the first continuous state reinforcement learning algorithm which exploits the benefits of statistical kernel smoothers to identify a unique and provably statistically accurate approximation of the value function for finite sample sizes and based on finite computational resources. By iterating until convergence for each finite sample, ADP uses data more efficiently than stochastic approximation. In addition, the solution of the stochastic programming approach is only asymptotically unique. We also present a new methodology to prove the convergence of our algorithm in rigorous mathematical detail for a broad variety of local averaging operators. For example, the variable resolution discretization techniques of Munos and Moore can be viewed as one application of the ADP methodology where the weighting function represents a dynamic grid. Hence, our results could be used to extend the theoretical evidence these authors put forward in support of their methodology.

The remainder of this paper is organized as follows. In Section III, we state our assumptions and we characterize the average–cost reinforcement learning problem formally. In Section IV, we introduce the local averaging operator. Sections V and VI describe the main theoretical results of this paper, including theorems establishing the admissibility of the policy iteration algorithm to compute solutions to the AACOE (2) and asymptotic bounds on the approximation error, respectively. In Section VII, we present conclusions.

III. Preliminaries and Problem Formulation

Consider an MDP defined by a sequence of states X_t taking values in $\mathbb{R}^d$ and a sequence of actions a_t taking values in the action space $A \equiv \{1, 2, \ldots, M\}$. The transition probabilities of the MDP are described by a family of kernels, $\{P_a(x, B) | a \in A\}$, characterizing the (time-homogeneous) conditional probability of the event $X_t \in B$ given $X_{t-1} = x$ and $a_{t-1} = a$. Here, B is a set in $\mathcal{B}(\mathbb{R}^d)$, the class of Borel sets on the state-space $\mathbb{R}^d$, and the sequences $\mathcal{X} \equiv \{X_0, \ldots, X_\infty\}$ and $A \equiv \{a_0, \ldots, a_\infty\}$ are stochastic processes on a probability space $(\Omega, \mathcal{F}, P)$. For reinforcement learning, it is convenient to define the strategy space $\mathcal{M}$ as the set of all stationary randomized policies of the form $\mu: \mathbb{R}^d \to \Delta(A)$, where $\Delta(A)$ is the set of all probability distributions over the elements of A. Obviously, $\mathcal{M}$ contains all stationary deterministic policies as a subset. We also define $\mathcal{M}_\varrho \subseteq \mathcal{M}$ as the set of all (stationary) ϱ-perturbed strategies, $\mu: \mathbb{R}^d \to \Delta_\varrho(A)$, where $\mu(x) \in \Delta_\varrho(A)$ implies that each $a \in A$ is chosen with a probability of at least ϱ by $\mu(x)$. Note that the application of a fixed $\mu \in \mathcal{M}$ transforms the process $\mathcal{X}$ into a Markov chain governed by the transition kernel $P_{\mu(x)}(x, B)$. Frequently, we will be interested in the long-run behavior of this chain starting from a

given initial condition $X_0 = x$, and we let $P_{x,\mu}$ denote the "induced" probability measure. Similarly, $E_{x,\mu}$ denotes the conditional expectation operator associated with $P_{x,\mu}$, and $P_{x,a}$ and $E_{x,a}$ denote the corresponding measure and expectation under a trivial strategy that always applies action a, respectively. Finally, each state–action pair has associated with it a cost, $c(x, a)$, representing an immediate penalty for visiting x and applying a. We define an induced cost function according to $c_\mu(x) \equiv \sum_{a=1}^{M} c(x, a)\mu(x, a)$, for all $\mu \in \mathcal{M}$.

A. Assumptions

A.1) For each $a \in A$, $x \in \mathbb{R}^d$, $P_a(x, \cdot)$ possesses a strictly positive Radon–Nikodym derivative, $p_a(x, y)$, with respect to the Lebesgue measure λ on $\mathbb{R}^d$.

A.2) For each $a \in A$, the conditional probability density $p_a(x, y)$ is uniformly continuous in $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$.

A.3) The cost function $c(\cdot, a)$ is continuous and "norm-like," i.e., the sublevel sets $\{x\colon c(x, a) \leq \gamma\}$ are pre-compact. There exists a norm-like function $\underline{c}\colon \mathbb{R}^d \to [1, \infty)$ such that $c(x, a) \geq \underline{c}(x)$.

A.4) There exists a probability measure ν on $\mathbb{R}^d$, $\varphi > 0$, and an integer $q \geq 1$ such that for all $x \in \mathbb{R}^d$, $\mu \in \mathcal{M}$:

$$P_{x,\mu}(X_q \in \cdot) \geq \varphi\nu(\cdot). \tag{3}$$

Additional assumptions regarding details of the learning algorithm are listed throughout the paper. For more details, see [14].

B. The Average–Cost Optimality Equation

We focus on MDPs where policies are assessed via their long-run average costs and refer the reader interested in discounted-costs to [4]. The average–cost of a policy μ is defined formally according to

$$\eta_\mu \equiv \lim_{T\to\infty} E_{x_0,\mu}\left[\frac{1}{T}\sum_{t=0}^{T-1} c_\mu(X_t)\right]. \tag{4}$$

η_μ is defined uniquely and independently of the starting position x_0 under the assumptions in Section III-A (see, for example, [14]). We will be interested in those policies $\mu^* \in \mathcal{M}$ that minimize the average–cost (4), i.e., in policies that satisfy $\eta_{\mu^*} \leq \eta_\mu$ for all $\mu \in \mathcal{M}$. A standard approach to determine an optimal policy is to solve the ACOE (1). Here, Γ_a is a "conditional expectation operator," defined according to $(\Gamma_a h)(x) \equiv E_{x,a}[h(X_1)]$.[1] Under the assumptions of this paper, if finite solutions η^* and h^* to (1) exist, a policy μ^* is optimal if and only if

$$\mu^*(x, a) \equiv \mathbb{1}(a = a') \quad \text{for some } a' \in \arg\min_a\{c(x, a) + (\Gamma_a h^*)(x)\}. \tag{5}$$

For a detailed discussion of this relationship, see [5], [14], [2], and [6].

[1]Because the transition probabilities $P_a(x, \cdot)$ are time-homogeneous, $(\Gamma_a h)(x)$ may be thought of alternatively as the conditional expectation of $h(X_t)$ given $X_{t-1} = x$ for arbitrary t.

Meyn [6] provides a detailed account of the applicability of the policy iteration algorithm to compute solutions to (1). Specifically, he relies on an alternative interpretation of (1) as a special case of *Poisson's equation*

$$\eta_\mu + h_\mu = c_\mu + \Gamma_\mu h_\mu \tag{6}$$

where $\mu = \mu^*$, and he demonstrates that the sequences η_μ and h_μ generated by policy iteration are convergent under the circumstances of Section III-A. Furthermore, the limiting values of these sequences are solutions to the ACOE (1) and hence the algorithm produces an optimal strategy. Here, η_μ is the average–cost defined in (4) and h_μ is the so-called *relative value function* or *differential cost function* associated with μ. Intuitively, h_μ is the relative disadvantage of starting the chain in the state x as opposed to drawing a random initial state from the stationary distribution π_μ.

Both Meyn's and our results hinge on the existence of regenerative events, R, that can be used to "split" the Markov chain $\mathcal{X}$. Roughly, R decouples the trajectory after its occurrence from the previous history of the chain. Because the differential cost after the first hitting time of R, $\tau \equiv \inf\{t \in \{1, 2, \ldots, \infty\} | R \text{ occurs}\}$, is independent of the initial position x, the splitting construction gives an alternative expression for the relative value function h_μ in terms of the history of $\mathcal{X}$ up to τ

$$h_\mu(x) \equiv E_{x,\mu}\left[\sum_{t=0}^{\tau-1}(c_\mu(X_t) - \eta_\mu)\right]. \tag{7}$$

Then, (7) together with the average–cost (4) define a solution to Poisson's equation. Details of the formal construction of the splitting procedure are discussed in [14].

IV. Averaging Operators

In this section, we discuss several approaches to local averaging and we draw a connection between regression and approximate dynamic programming. Consider first a typical regression task where we wish to approximate the conditional expectation $f(z) \equiv E[\mathrm{Y}|\mathrm{Z} = z]$ based on realizations of the continuous random variables Y and Z. Then, we can alternatively interpret the conditional expectation operation $\Gamma_a h$, defined in the previous section, as a regression by choosing $\mathrm{Y} = h(X_1)$ and $\mathrm{Z} = X_0$. That is, given realizations of $h(X_1)$ and X_0, any regression method is transformed easily into an approximate expectation operator $\hat{\Gamma}_{m,a}$ in principle. However, many parametric or semiparametric regression approaches such as linear estimators or neural networks fail to generate stable learning algorithms in practice (for a discussion, see [3]). By a *stable* algorithm, we mean a procedure based on finite computational resources, in particular, a finite set of sample data and a finite representation of the value function, that converges to a unique solution independently of any starting conditions. A class of regression methods particularly suited for stable ADP are methods based on "local averaging." These models have been studied extensively in the statistics literature both from a practical and a theoretical viewpoint (e.g., [15]–[18]).

We assume that a training data set, $\mathcal{S}$, is generated by simulating the MDP for m steps using a fixed initial policy, $\overline{\mu}$, and a

fixed initial state, $x_0 \in \mathbb{R}^d$. Here, it is important that $\overline{\mu}$ chooses each action with positive probability to guarantee sufficient "exploration" of the state-space, i.e., $\overline{\mu} \in \mathcal{M}_{\overline{\varrho}}$ for a fixed $\overline{\varrho} > 0$. Formally, we let $\{Z_0, \ldots, Z_m\}$ be a collection of random variables distributed according to $P_{x_0,\overline{\mu}}(X_0, \ldots, X_m)$, and let $\mathcal{S} = \{z_0, \ldots, z_m\}$ be the m-step sample trajectory of the MDP.[2] Besides the state variables, we also have the actions $\{a_0, \ldots, a_{m-1} | a_s \sim \overline{\mu}(z_s, \cdot)\}$ and the costs $\{c(z_s, a_s) | 0 \leq s < m\}$ generated during the simulation, and we define a partition of $\mathcal{S}$ into subsets according to $\mathcal{S}_a \equiv \{Z_s \in \mathcal{S} | a_s = a\}$. The cardinality of each of these subsets is m_a, i.e., $m_a \equiv \#\mathcal{S}_a$. Furthermore, let $\#_a(D)$ be the number of samples in S that are located within a generic set D, $\#_a(D) \equiv \sum_{s'=0}^{m-1} \mathbf{1}(a_{s'} = a, Z_s \in D)$.

We consider nonparametric regression operators that can be written as local averages in terms of the data in $\mathcal{S}$ of the form

$$\left(\hat{\Gamma}_{m,\mu} h\right)(x) \equiv \sum_{s=0}^{m-1} k_{m,\mu}(z_s, x) h(z_{s+1}). \tag{8}$$

Here, h is a generic function and $k_{m,\mu}(z_s, x) \equiv \sum_{a=1}^{M} \mu(x, a) k_{m,a}(z_s, x)$ is a so-called "weighting function" or "weighting kernel" on $\mathbb{R}^d \times \mathbb{R}^d$. As previously mentioned, the fundamental idea underlying approximation (8) is simple. In order to estimate the conditional expectation of $h(X_1)$ given $X_0 = x$, it suffices to consider a large number of sample transitions starting from a state z_s in the neighborhood of x to some successor state z_{s+1}. Then the average of the function values at the successors $h(z_{s+1})$ is a natural estimate of the conditional expectation. The weighting function, $k_{m,a}(z_s, x)$, serves to assess the vicinity of z_s to x. That is, $h(z_{s+1})$ obtains a substantial weight only if z_s is close to x. The weighting function is constrained to be positive, decreasing with $\|z_s - x\|$, and to satisfy $\sum_{s=0}^{m-1} k_{m,a}(z_s, x) = 1$. These properties ensure proper averaging of the successor values $h(z_{s+1})$ in (8). Also, notice that the transition from z_s to z_{s+1} only reveals information about that action a_s used during simulation at time s. Hence, only those transitions should be used for averaging that were generated using the proper action; that is, $k_{m,a}(z_s, x) = 0$ if $a_s \neq a$. Otherwise, we assign a *strictly* positive weight to z_s. Next, we discuss various possibilities to define $k_{m,a}(z_s, x)$ in practice. For further reference on these weighting kernels, see [18].

1) Grid-Based Methods: An intuitive way to approximate an MDP is to partition the state-space $\mathbb{R}^d$ into a collection of mutually exclusive subsets. These sets may be simple rectangles as in the case of a regular grid, or they may be polytopes resulting from a Voronoi tessellation based on a lattice rule (e.g., [19]). We formally characterize this partition using a mapping U: $\mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ that assigns each $x \in \mathbb{R}^d$ to a subset of $\mathbb{R}^d$, and define a corresponding weighting function according to

$$k_{m,a}(z_s, x) \equiv \begin{cases} (1-\vartheta)/\#_a U(x) & \text{if } a_s = a,\ z_s \in U(x), \\ \vartheta/\#_a U^c(x) & \text{if } a_s = a,\ z_s \in U^c(x), \\ 0 & \text{if } a_s \neq a. \end{cases} \tag{9}$$

[2]For simplicity, we sometimes ignore the difference between $\{Z_0, \ldots, Z_m\}$ and $\{z_0, \ldots, z_m\}$ below.

That is, we assign uniform weights to the observations within the neighborhood of x, and we also assign some small weight to the remaining observations satisfying $a_s = a$. The "perturbation constant" ϑ in (9) satisfies $0 < \vartheta < 1$.

2) Averaging Using Trees: Another possibility to partition the state-space is by recursive splitting. For example, k–d-trees partition the state-space into rectangles in such a way that each rectangle contains an approximately equal numbers of observations. Hence, a k–d-tree implies a partition function U by analogy to the grid-based approach and its weighting function, $k_{m,a}(z_s, x)$, can be defined formally as in (9). Note, however, that U is dependent on the training data set $\mathcal{S}$ in this case so that U is random and $k_{m,a}(z_s, x)$ is adaptive with respect to the local data design. In what follows, we assume that a separate tree is available for each action $a \in A$ and that the leaves of these trees contain exactly l samples for simplicity. The main advantages of tree-based averaging are computational speed, interpretability, and the capability to deal with relatively high-dimensional data [20].

3) Nearest Neighbor Weights: While approaches 1) and 2) are computationally very efficient, they can sometimes be suboptimal because the weights implied by (9) are based on a "hard" decision boundary between the neighborhood $U(x)$ and its complement. A natural idea is to determine weights as a more gradually decaying function of the distance $\|z_s - x\|$, corresponding to "soft" neighborhood boundaries. Formally, we let $(z_{(0)}, \ldots, z_{(m_a)})$ denote a permutation of the elements in $\mathcal{S}_a$, ordered according to increasing values of $\|z_s - x\|$, and we assign fixed weights to these samples according to their position in the ordered sequence:

$$k_{m,a}(z_s, x) \equiv \begin{cases} (1-\vartheta)\upsilon_{l,j}, & \text{if } a_s = a,\ s = (j),\ j \leq l \\ \vartheta/(m_{\mu(x)} - l), & \text{if } a_s = a,\ s = (j),\ j > l, \\ 0, & \text{if } a_s \neq a. \end{cases} \tag{10}$$

Here, $\upsilon_{l,j}$ is a decreasing sequence of scalars satisfying $\overline{\upsilon}/l > \upsilon_{l,j} \geq \underline{\upsilon}/l > 0$ and $\sum_{j=0}^{l-1} \upsilon_{l,j} = 1$. For example, $\upsilon_{l,j}$ can be defined as a function of j using a uniform, a triangular, a quadratic, or a Gaussian kernel (for details, see [21]). Intuitively, the l nearest neighbors are weighted according to their distance from x by (10), while the remaining observations again obtain a small uniform weight. We assume for simplicity that $m_a > l$ for this approach, i.e., at least l observations must be available in each of the sets $\mathcal{S}_a$ (see also Proof of Theorem 1).

Note that the list of averaging approaches 1)–3) is intended as a set of illustrative examples rather than an exhaustive enumeration of viable approaches. Specifically, below we aim at presenting our theoretical results in a general framework that covers 1)–3) as special cases. Alternative averaging methods of practical interest that could be described in this formal framework include weighting using a Gaussian "mother kernel" [4], locally weighted regression, discrete kernel-based averaging across multiple cells of a partition, as well as combinations of the ideas above. With regard to our theoretical analysis, the main difference between these various approaches is the choice of the neighborhood used for averaging. In particular, the neighboring region $U_{m,a}(x)$ can be defined explicitly as in 1) or 2)

or implicitly as in 3). (We give details of the implicit definition in the proof of Theorem 2). Also, $U_{m,a}(x)$ can be either a fixed subset of the state-space as in 1) or it can be data-dependent as in 2) or 3), which requires a separate mathematical treatment. We denote these distinct approaches as *fixed neighborhood* and *adaptive neighborhood* methods, respectively. For notational convenience, we will also introduce a formal representation of the samples in the neighborhood $U_{m,a}(x)$ using the symbol $N_{m,a}(x) \equiv \{z_s \in \mathcal{S}_a | z_s \in U_{m,a}(x)\}$. That is, $N_{m,a}(x)$ is the subset of samples in $\mathcal{S}_a$ that are the nearest neighbors of x. For the nearest neighbor approach, we adopt the convention $N_{m,a}(x) = \{z_s \in \mathcal{S}_a: \|z_s - x\| \leq \|z_{(l)} - x\|\}$, and we define $U_{m,a}(x)$ via the relationship $U_{m,a}(x) \equiv \{y \in \mathbb{R}^d | \exists z_s \in N_{m,a}(x): \|y - x\| \leq \|z_s - x\|\}$. To prove the consistency of the averaging approaches 1)–3), it is important that the size of $U_{m,a}(x)$ [and, hence, also of $N_{m,a}(x)$] shrinks to zero with increasing sample size at a rate that warrants a suitable bias-variance tradeoff. Depending on the chosen method, this condition must be expressed as an assumption on $U_{m,a}(x)$ or $N_{m,a}(x)$.

A.5) The neighborhood size in the grid-based averaging approach 1) satisfies for all $x \in \mathbb{R}^d$, $a \in A$

$$\lambda(U_{m,a}(x)) \overset{m\to\infty}{\longrightarrow} 0 \tag{11}$$

$$m\lambda(U_{m,a}(x))^2 - \log(m+1) \overset{m\to\infty}{\longrightarrow} \infty. \tag{12}$$

The number of neighbors $\#N_{m,a}(x)$ in the tree-based approach 2) and in the nearest neighbor approach 3) satisfies for all $x \in \mathbb{R}^d$, $a \in A$

$$\#N_{m,a}(x) \overset{m\to\infty}{\longrightarrow} \infty \tag{13}$$

$$\#N_{m,a}(x)/m \overset{m\to\infty}{\longrightarrow} 0. \tag{14}$$

Conditions similar to (11)–(14) are standard in the theory of pattern recognition and are easy to satisfy in practice [18]. In addition, the magnitude of the perturbations must decay asymptotically.

A.6) The perturbation constant ϑ approaches zero as m goes to infinity.

Next, we consider the implications of applying approaches 1)–3) for approximate dynamic programming. We first investigate the solubility of (2) as a means to compute an approximate strategy $\hat{\mu}_m$ in Section V, and then we derive theoretical properties of this approximation in Section VI.

V. Approximate Dynamic Programming

As previously shown, we suggested substituting the unknown Γ_a in (1) with one of the approximate expectation operators $\hat{\Gamma}_{m,a}$ defined in the previous section. The approximate average–cost optimality equation takes the form of (2), and an approximately optimal strategy can be found by analogy to (5)

$$\hat{\mu}_m(x, a) \equiv \mathbb{1}(a = a') \qquad \text{for some } a' \in \arg\min_a \cdot \left\{c(x, a) + \left(\hat{\Gamma}_{m,a}\hat{h}_m\right)(x)\right\}. \tag{15}$$

In this section, we derive numerical methods for the solution of (2). In more detail, to derive an algorithm for the solution of (2) we reinterpret the approximate conditional expectation operator $\hat{\Gamma}_{m,a}$ as the *true* conditional expectation operator in a new, artificial Markov chain in Section V-A and we introduce an additional condition for the solubility of this artificial MDP in Section V-B. In Section V-C, we suggest an implementation of kernel-based reinforcement learning using finite-state dynamic programming. Note that we describe our algorithm using mathematical notation rather than pseudocode for brevity and consistency with the rest of this paper. A treatment providing even more algorithmic detail can be found in [4] and [22]. Moreover, in [22], we describe the practical application of the algorithm to the financial problem of optimal portfolio choice and provide detailed experimental results.

A. An "Artificial" MDP

As previosuly shown, we defined the weighting function $k_{m,a}(z_s, x)$ so as to satisfy the conditions $k_{m,a}(z_s, x) \geq 0$ and $\sum_{s=0}^{m-1} k_{m,a}(z_s, x) = 1$. Hence, we can think of these weights as conditional probabilities and define an artificial transition kernel according to

$$K_{m,\mu}(x, A) \equiv \sum_{s=0}^{m-1} \sum_{a=1}^{M} \mu(a, x) k_{m,a}(z_s, x) \mathbb{1}(z_{s+1} \in A). \tag{16}$$

Let $E_{m,\mu}$ be the expectation operator associated with $K_{m,\mu}(x, A)$, so that $(\hat{\Gamma}_{m,\mu}h)(x) = E_{m,\mu}[h(X_1)|X_0 = x]$. Then, we can interpret the AACOE (2) as the true ACOE of the artificial MDP implied by (16) and we can apply dynamic programming for its solution. Note that this is a major simplification of the estimation problem because it allows us to treat equation (2) in the familiar MDP framework. Specifically, the ability to analyze approximate dynamic programming using an artificial MDP is one of the main reasons for using local averaging to approximate the conditional expectation operator in the first place: For most alternative approaches, including linear or nonlinear least-squares regression, locally weighted regression, smoothing splines, wavelets, etc., a probabilistic interpretation of (16) is generally not applicable and, therefore, some of these methods fail to converge.[3]

Given our new interpretation of $\hat{\Gamma}_{m,a}$ as the conditional expectation under $K_{m,\mu}(x, A)$, it will be convenient to define average–costs and relative value functions of the artificial MDP by analogy to (4) and (7)

$$\eta_{m,\mu} \equiv \lim_{T\to\infty} E_{m,\mu}\left[\frac{1}{T}\sum_{t=0}^{T-1} c_\mu(X_t)\middle| X_0 = x_0\right] \tag{17}$$

$$h_{m,\mu}(x) \equiv E_{m,\mu}\left[\sum_{t=0}^{\tau-1}(c_\mu(X_t) - \eta_{m,\mu})\middle| X_0 = x\right]. \tag{18}$$

In particular, an approximately optimal strategy $\hat{\mu}_m$ achieves the minimum of $\eta_{m,\mu}$ over all μ, and the magnitudes $\hat{\eta}_m$ and

[3]The majority of the mentioned approximations can be interpreted as a special case of local averaging using the notion of "equivalent kernels" [23]. However, the equivalent kernel typically violates the positivity and normalization constraints on $k_{m,a}(z_s, x)$.

$\hat{h}_m$ in (2) are the optimal average–cost and the optimal relative value function associated with $\hat{\mu}_m$. Hence, it is straightforward to compute $\hat{\eta}_m$ and $\hat{h}_m$ by applying standard algorithms such as policy iteration or value iteration to the artificial MDP. For (17) and (18) to be well defined, however, we must first show that the artificial kernel $K_{m,\mu}(x, A)$ inherits some regularity conditions from the original transition kernel $P_\mu(x, A)$. Next, we introduce an additional condition that will be needed for this purpose.

B. "Trembling Hand Policies"

In this section, we consider the stability of the artificial MDP implied by the transition kernel (16). Our goal is to recover the same regeneration structure that was established for the original MDP in Section III-B. In particular, we would like to use the same stopping time, τ, in the definitions of h_μ and $h_{m,\mu}$ in (7) and (18). This important connection between the original and the artificial MDP is sufficient to guarantee the convergence of policy and value iteration for the artificial MDP [6].

As in Assumption A.4 of Section III-A, focus is on the existence of a suitable minorizing measure ν suitable for spitting the artificial chain by analogy to Section III-B. A simple way to define this measure is by requiring that $\mu(a, x) \geq \varrho$ for some positive constant ϱ (for details, see [24]). A game-theoretic interpretation of this condition is that of a player with a "trembling hand" that causes him to commit errors sporadically. Mathematically, trembling hand policies simply correspond to elements of the perturbed action space $\mathcal{M}_\varrho$ defined in Section III. That is, we approximate the optimal policy, $\mu^* \in \mathcal{M}$, by using the policy within $\mathcal{M}_\varrho \subset \mathcal{M}$ that is optimal with respect to the artificial MDP, $\hat{\mu}_m$. This proceeding introduces an additional approximation error which needs to decay asymptotically to achieve consistency. A necessary condition is that the magnitude of the perturbation itself vanishes at an appropriate rate with growing sample size

A.7) The perturbation constant ϱ approaches zero as m goes to infinity.

The following policies are meant to be elements of $\mathcal{M}_\varrho$ where ϱ is contingent on m in accordance with Assumption A.7). It remains to define the minorizing measure $\nu^K(B)$ by analogy to (3). For this purpose, note that every state in $\mathcal{S}$ can be reached from any other state with a probability of at least $\varrho\vartheta > 0$ under $K_{m,\mu}(x, A)$ and let $\nu^K(B) \equiv (1/m)\sum_{s=0}^{m-1} \mathbb{1}(Z_{s+1} \in A)$. $\nu^K(B)$ has the property that $K_{m,\mu}(x, B) \geq \varrho\vartheta\nu^K(B)$ for all $x \in S$, $B \in \mathcal{B}(\mathbb{R}^d)$, and $\mu \in \mathcal{M}_\varrho$. Hence, it can be used for regeneration and we interpret τ as the corresponding renewal time.

C. "Self-Approximating Property"

We saw that the artificial MDP defined in Section V-A reduces the solution of the AACOE (2) to the solution of an ordinary dynamic programming problem in the new MDP. Computationally, a severe problem for dynamic programming in both the original and the artificial chain is the representation of the continuous value function: It must be approximated for practical implementation. On the other hand, in kernel-based reinforcement learning the AACOE is itself an approximation. We will show that the AACOE can be solved exactly using the so-called "self-approximating property" of local averaging that will be described next. The relevance of this concept for approximate dynamic programming has been emphasized in [7].

To illustrate the concept of self approximation, note first that the only information needed to apply the approximate expectation operator $\hat{\Gamma}_{m,\mu}$ defined in (8) to a generic function h are the values of h at the states in the training sample $\mathcal{S}$. With regard to the solution of the AACOE (2), this insight suggests the following procedure. In a first step, we compute the values of $\hat{h}_m$ at the locations z_s that are consistent with (2)

$$\hat{\eta}_m + \hat{h}_m(z_s) = \min_a \left\{ c(z_s, a) + \left(\hat{\Gamma}_{m,a}\hat{h}_m\right)(z_s) \right\} \quad (19)$$

for $s = 0, \ldots, m-1$. Second, we derive the values of $\hat{h}_m$ at new locations x using

$$\hat{h}_m(x) \equiv \min_a \left\{ c(x, a) + \left(\hat{\Gamma}_{m,a}\hat{h}_m\right)(x) \right\} - \hat{\eta}_m. \quad (20)$$

The reader may wish to verify that the magnitudes determined in this manner constitute solutions to (2). The advantage of this two-step procedure is that equation (19) can be thought of alternatively as the AACOE of yet another artificial MDP with a *finite* state-space consisting of the elements of $\mathcal{S}$. To make this precise, we identify the samples in $\mathcal{S}$ with the elements of the set $\tilde{\mathcal{X}} \equiv \{1, \ldots, m\}$, and we define the vectors and matrices

$$\begin{aligned} \hbar_{m,\mu}(i) &\equiv h_{m,\mu}(z_i) && \text{for } i = 1, \ldots, m, \\ \mathbf{c}_{m,\mu}(i) &\equiv c_\mu(z_i) && \text{for } i = 1, \ldots, m, \\ \Phi_{m,\mu}(i, j) &\equiv k_{m,\mu}(z_j, z_i) && \text{for } i = 1, \ldots, m \text{ and } j = 1, \ldots, m. \end{aligned}$$

The m vectors $\hbar_{m,\mu}$ and $\mathbf{c}_{m,a}$ summarize the values of $h_{m,\mu}$ and c_μ at the sample states, and the $m \times m$ matrix $\Phi_{m,\mu}$ contains the weight assigned to the sample z_j at state z_i in location (i, j). Note that $\Phi_{m,\mu}$ is strictly positive and stochastic given the conventions of Sections IV and V-B; that is, it satisfies $\Phi_{m,\mu}(i, j) > 0$ and $\sum_{j=1}^m \Phi_{m,\mu}(i, j) = 1$. Hence, we interpret $\Phi_{m,\mu}$ as a transition matrix that defines a new MDP with the state space $\tilde{\mathcal{X}}$, and $\mathbf{c}_{m,\mu}$ and $\hbar_{m,\mu}$ are the cost and the relative value function of the new MDP, respectively. Due to the strict positivity of $\Phi_{m,\mu}$, the transition structure of the discrete MDP is unichain for all $\mu \in \mathcal{M}_\varrho$, and its ACOE is a discrete version of the first step for the solution of the AACOE (19)

$$\hat{\eta}_m + \hbar_m(i) = \min_a \{\mathbf{c}_{m,a}(i) + (\Phi_{m,a}\hbar_m)(i)\} \quad (21)$$

for $i = 1, \ldots, m$. The correspondence between (21) and (19) is extremely helpful to determine $\hat{\eta}_m$ and $\hbar_m$ in practice, because it reduces the solution of (19) to a dynamic programming problem in a unichain finite-state MDP. This dynamic programming problem can then be attacked using standard algorithms such as policy iteration, value iteration, or linear programming (for example, see [5], [25], and [2]). Specifically, it can be shown that all of these algorithms converge to the optimal solution if the MDP is unichain as in the case of (21). For example, in [22], we use value iteration in combination with a Gaussian weighting kernel to solve a real-world optimal portfolio choice problem.

One of the findings of this experiment is that the number of iterations until convergence depends on the *kernel bandwidth* in an interesting manner. In [4], we highlight further algorithmic details including the optimal choice of the bandwidth parameter. We shall not repeat these details here because they are not directly relevant to the theoretical results presented in this work. Instead we assume simply that an efficient algorithm for the solution of (21) is available in the following.

As previously indicated, we obtain the approximate average cost $\hat{\eta}_m$ and the values $\hat{h}_m(z_i) = \hbar_m(i)$ at the locations in $\mathcal{S}$ as the output of our algorithm. The values of $\hat{h}_m(x)$ at new locations is then computed using (20) so that $\hat{\eta}_m$ and $\hat{h}_m(x)$ constitute solutions to (2). Statistical properties of these solutions are topic of Section VI.

VI. Consistency

In Section V, we outlined an iterative algorithm for the solution of the AACOE (8) and for the determination of an approximate optimal strategy $\hat{\mu}_m$. A crucial issue is the asymptotic behavior of this approximation. As a minimum statistical requirement, we demand that $\hat{\mu}_m$ should converge to the *true* optimal policy, μ^*, in an appropriate sense as the sample size m grows to infinity. In other words, the algorithm should produce a consistent estimate of μ^*.

The derivation of this consistency result is complicated by the fact that we consider average–cost problems in this work. Convergence results for discounted-cost problems can be derived using the contraction property of the approximate Bellman operator [4]. For average–cost problems, we demonstrate first in Proposition 1 that, given any *fixed* strategy μ, the approximation error of $\eta_{m,\mu}$ with respect to the true cost, η_μ, can be related to the approximation error of the approximate expectation operator $\hat{\Gamma}_{m,\mu}$ with respect to Γ_μ. Second, we use the result of the first step to demonstrate also that $\eta_{m,\mu}$ converges to η_μ under suitable conditions in Proposition 2 and Theorem 1. Here, it is crucial that the convergence occurs *uniformly* for all strategies. Finally, we argue that, because $\hat{\eta}_m$ and η^* constitute the minima of $\eta_{m,\mu}$ and η_μ with respect to μ, this convergence property also carries over to the approximate optimal costs in Theorem 2.

As a minimum prerequisite to obtain the first result, i.e., that $\eta_{m,\mu}$ converges to η_μ, it seems intuitively clear that the approximate expectation operator $\hat{\Gamma}_{m,\mu}$ should converge to the true Γ_μ. We summarize this result in the following proposition, which can be interpreted as a continuous-state version of a result in [26].

Proposition 1: For any $\mu \in \mathcal{M}_\varrho$, the approximation errors of the magnitudes $\eta_{m,\mu}$ and $h_{m,\mu}$ can be written as

$$\begin{aligned}\eta_{m,\mu} - \eta_\mu &= \int \left(\hat{\Gamma}_{m,\mu} h_\mu - \Gamma_\mu h_\mu\right)(x)\pi_{m,\mu}(dx) \qquad (22)\\ h_{m,\mu}(x) - h_\mu(x) &= -(h_{m,\mu}(x) + C_h)(\eta_{m,\mu} - \eta_\mu)\\ &\quad + E_{m,x}\left[\sum_{j=0}^{\tau-1}\left(\hat{\Gamma}_{m,\mu} h_\mu - \Gamma_\mu h_\mu\right)(X_j)\right] \qquad (23)\end{aligned}$$

where $\pi_{m,\mu}$ is the unique invariant measure associated with the artificial transition kernel $K_{m,\mu}(x, \cdot)$ and where C_h is a scalar depending on m, ϱ, and ϑ.

For brevity, we provide short versions of proofs in the Appendix, and refer the reader interested in details to a longer technical report [24]. Equation (22) relates the approximation error $E_{x_0,\mu}|\eta_{m,\mu} - \eta_\mu|$ to the expected integral over an expression of the form $(\hat{\Gamma}_{m,\mu} h_\mu - \Gamma_\mu h_\mu)(x)$. Recollecting our discussion at the beginning of Section IV, this expression can be interpreted as the residual of a nonlinear regression of $h_\mu(X_1)$ onto $X_0 = x$. Roughly, $\eta_{m,\mu}$ is a consistent estimate of η_μ whenever the regression method underlying $\hat{\Gamma}_{m,\mu}$ is consistent in an appropriate sense. Conditions for the consistency of local averaging have been studied extensively in the Statistics literature (e.g., [15]–[18]). Our next result summarizes these conditions in a form that is suitable for the weighting functions described in Section IV. For its proof, we need a Markov chain version of Hoeffding's inequality described in [27] as well as an additional "approximatibility" assumption.

A.8) The sequence of balls $\mathcal{B}_{\overline{x},u} \equiv \{z \in \mathbb{R}^d : \|z - \overline{x}\| \le u\}$ satisfies the condition

$$\lim_{T\to\infty} E_{x,\mu}\left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{1}(X_t \notin \mathcal{B}_{\overline{x},u}) c_\mu(X_t)\right] \overset{u\to\infty}{\longrightarrow} 0$$

uniformly for all $x \in \mathbb{R}^d$, $\mu \in \mathcal{M}$ and for some $\overline{x} \in \mathbb{R}^d$.

Assumption A.8) affirms that any Markov chain that is induced by a strategy μ can be approximated by considering the costs incurred within finite distance from the "center" of the state-space, $\overline{x}$. This assumption is necessary to restrict the region of the state-space on which $\hat{\Gamma}_{m,\mu} h_\mu$ needs to approximate $\Gamma_{m,\mu} h_\mu$, because the training sample $\mathcal{S}$ cannot cover every neighborhood in $\mathbb{R}^d$ simultaneously. Intuitively, Assumption A.8) precludes from our investigation cases in which a substantial contribution to the average cost $\eta_{m,\mu}$ "escapes to infinity" with growing sample size for some sequence of strategies. Under this additional assumption we obtain the key result of this section:

Proposition 2: Assume there exists two constants $0 < \underline{C}_k \le \overline{C}_k$ such that for all $\delta > 0$, $\varrho > 0$, $u > 0$, $C_1 > 0$, $C_2 > 0$ and all closed balls $\mathcal{B} \subseteq \mathbb{R}^d$ the weighting function $k_{m,\mu}(Z_s, x)$ satisfies the following conditions for all $x \in \mathcal{B}$:

$$\begin{aligned}\frac{\underline{C}_k}{\#N_{m,\mu}(x)} &\le k_{m,\mu}(Z_s, x)\\ &\le \frac{\overline{C}_k}{\#N_{m,\mu}(x)}, \qquad Z_s \in N_{m,\mu}(x) \qquad (24)\end{aligned}$$

$$\sum_{Z_s \in N^c_{m,\mu}(x)} k_{m,\mu}(Z_s, x) \overset{m\to\infty}{\longrightarrow} 0 \quad \text{uniformly.} \qquad (25)$$

Assume also that the set of neighboring samples $N_{m,a}(x)$ has the properties

$$v(\mathcal{E}, m) \sup_{a\in A, x\in B} P_{x_0,\overline{\mu}}\left(\#_a \mathcal{B}_{x,u/2} < C_1 \#N_{m,a}(x)\right) \overset{m\to\infty}{\longrightarrow} 0 \qquad (26)$$

$$v(\mathcal{E}, m) E_{x_0,\overline{\mu}}\left[\sup_{a\in A, x\in B} e^{-C_2 \#N_{m,a}(x)}\right] \overset{m\to\infty}{\longrightarrow} 0 \qquad (27)$$

uniformly for all $x_0 \in \mathbb{R}^d$, $\overline{\mu} \in \mathcal{M}_\varrho$. Then, we have

$$E_{x_0,\overline{\mu}}|\eta_{m,\mu} - \eta_\mu| \stackrel{m\to\infty}{\longrightarrow} 0 \tag{28}$$

uniformly for all $x_0 \in \mathbb{R}^d$, $\mu \in \mathcal{M}_{\overline{\varrho}}$.

Remember from Section IV that x_0 and $\overline{\mu}$ are a fixed initial state and a fixed exploration strategy, respectively. Assumption (24) guarantees that the weights are essentially uniform in the interior of the neighborhood $N_{m,a}(x)$, and Assumption (25) confirms that any weight outside $N_{m,a}(x)$ becomes unimportant asymptotically. Assumptions (26) and (27) concern the growth rate of the number of samples in $N_{m,a}(x)$. They are based on the *shattering coefficient*, $v(\mathcal{E}, m)$, which is implied by the weighting function. Intuitively, $v(\mathcal{E}, m)$ denotes the maximum number of different subsets of m arbitrary points in $\mathbb{R}^d$ that can be "picked out" by a collection of subsets of $\mathbb{R}^d$, $\mathcal{E}$. Depending on the type of local averaging used, these sets may be spheres, rectangles, or polygons of a specific form. Shattering coefficients are a standard tool to establish uniform convergence in statistical learning theory and pattern recognition. For example, Vapnik [28] and Devroye *et al.* [18] provide excellent surveys of this method and they also discuss the important connection between the shattering coefficient and the VC-dimension, d_{VC}. For the averaging methods suggested in Section IV, the shattering coefficients are bounded by $(m+1)^{d_{\mathrm{VC}}}$ which simplifies the application of Proposition 2.

Theorem 1: The local averaging approaches 1)–3) are uniformly consistent.

Hence, by gathering sufficient information from the trajectory of a single policy, $\overline{\mu}$, it is possible to infer the value of any other policy μ arbitrarily well using local averaging. We have thus proven that local averaging is suitable to approximate the average–costs of a wide class of MDPs for any fixed strategy μ. From here it is a relatively easy step to demonstrate also that local averaging can be used in combination with approximate dynamic programming in order to approximate the *optimal* strategy, μ^*. In detail, because the approximation error of $\eta_{m,\mu}$ can be made arbitrarily small for all $\mu \in \mathcal{M}_\varrho$ according to Proposition 2, it must be small specifically for the choices $\hat{\mu}_m$ and μ^*. We use this fact to prove our next theorem:

Theorem 2: Under the conditions of Proposition 2, the approximate optimal cost $\hat{\eta}_m$ converges to the true optimal cost in the sense that

$$E_{x_0,\overline{\mu}}|\hat{\eta}_m - \eta^*| \stackrel{m\to\infty}{\longrightarrow} 0.$$

Furthermore, the cost of $\hat{\mu}_m$ converges to the optimal cost in the sense that

$$E_{x_0,\overline{\mu}}|\eta_{\hat{\mu}_m} - \eta^*| \stackrel{m\to\infty}{\longrightarrow} 0.$$

In other words, the approximate optimal strategy $\hat{\mu}_m$ performs as well as μ^* asymptotically and we can predict the optimal costs, η^*, using the estimate $\hat{\eta}_m$. In principle, we can thus solve any average–cost MDP satisfying our assumptions using the approximate dynamic programming algorithm of Section V. From a practical standpoint, Theorem 2 asserts that the performance of approximate dynamic programming can be improved arbitrarily by increasing the amount of training data.

VII. Conclusion

We presented a new learning algorithm to approximate the value function and the optimal policies of an continuous-state average–cost MDPs using simulation. This approximation uses finite-state dynamic programming, where we replace the conditional expectation operator in the average–cost optimality equation with an approximate operator. The approximate operator is based on one of various forms of local averaging such as grids, nearest-neighbor regression, and trees. In Section VI, we proved the consistency of this approach by relating reinforcement learning to nonlinear regression. In principle, the average–cost of the approximate strategy is hence arbitrarily close to the average–cost of the optimal control for a sufficiently large sample.

Practically, the performance of our approximation (and of any other method) is dictated by the amount of available computational resources. In particular, the computational complexity of kernel-based reinforcement learning is $O(Mm^2)$ for each approximate value iteration step and the storage requirements are of the complexity $O(m)$ due to the self-approximating property. Hence the computational effort grows with the sample size which prevents exact online operation. However, efficient online approximations can be constructed easily based on discarding old observations or summarizing them by "sufficient statistics." For details on the approximation issue and on the computational complexity of our algorithm, see [24].

The fact that the amount of training data needed to achieve a given accuracy depends exponentially on the dimensionality of the state-space, d, can be interpreted as evidence of the "curse of dimensionality" in reinforcement learning. In particular, *any* method for approximating the value function of an MDP from data is subject to this curse. Otherwise, an approximation method that "breaks" the curse of dimensionality could be used alternatively as a nonlinear regression method with the same property by constructing a trivial one-step MDP. This is clearly inconsistent with theoretical lower complexity bounds for nonlinear regression derived in [17].[4] The fact that the curse of dimensionality cannot be broken implies that the computational effort necessary to obtain a statistically satisfactory approximation of the value function must eventually become prohibitive in high dimensions. In many real-world situations this problem is alleviated by prior knowledge which may be used to define a low-dimensional approximation of the original state space. For example, Tsitsiklis and Van Roy select special "features" summarizing the dynamics of an MDP [3], and Ormoneit and Hastie describe an approach designed for local averaging where an optimal linear projection of the system state onto a low-dimensional subspace is learned automatically from training data [29]. Nonetheless, the statistical and computational problems of reinforcement learning in high-dimensional spaces remain a serious obstacle in many applications and should be addressed in more detail in future work.

[4]In [7] it is shown that the curse-of-dimensionality can indeed be broken under special circumstances if the transition dynamics of the MDP are known. Note that this is different from breaking the curse of dimensionality in reinforcement learning.

APPENDIX I
PROOF OF PROPOSITION 1

Consider Poisson's equation (6) for a fixed policy $\mu \in \mathcal{M}_{\varrho}$ and let $\xi_m(x) \equiv (\hat{\Gamma}_{m,\mu} h_\mu - \Gamma_\mu h_\mu)(x)$. We can rewrite (6) in terms of the "drift equation"

$$\left(\hat{\Gamma}_{m,\mu} h_\mu\right)(x) = h_\mu(x) + \eta_\mu - c_\mu(x) + \xi_m(x). \tag{29}$$

Using (29), it is straightforward to construct the martingale (under $K_{m,\mu}$): $M_k \equiv h_\mu(X_k) - \sum_{j=0}^{k-1}(\eta_\mu - c_\mu(X_j) + \xi_m(X_j))$. Using $E_{m,x}[M_T] = M_0$, we obtain

$$\frac{1}{T} h_\mu(x) = \frac{1}{T} E_{m,x} \cdot \left[h_\mu(X_T) - \sum_{j=0}^{T-1} (\eta_\mu - c_\mu(X_j) + \xi_m(X_j)) \right]. \tag{30}$$

Equation (22) follows upon observing that all terms on the right-hand side of (30) are convergent as T goes to infinity and that the h_μ terms vanish in the limit (for more details, see [24]). Similarly, in order to derive (23), we apply optional sampling with regard to the stopping time τ to M_k

$$\begin{aligned} E_{m,x}[M_\tau] &= E_{m,x}[h_\mu(X_\tau)] \\ &\quad - E_{m,x}\left[\sum_{j=0}^{\tau-1} (\eta_\mu - c_\mu(X_j) + \xi_m(X_j))\right] \\ &= 0 + E_{m,x}[\tau](\eta_{m,\mu} - \eta_\mu) + h_{m,\mu}(x) \\ &\quad - E_{m,x}\left[\sum_{j=0}^{\tau-1} \xi_m(X_j)\right]. \end{aligned}$$

Using $E_{m,x}[M_\tau] = M_0$ and $E_{m,x}[\tau] \leq h_{m,\mu}(x) + C_h$ (see [14]) we obtain (23). ∎

APPENDIX II
PROOF OF PROPOSITION 2

We describe a proof extending results of [21], [30], and [18]. We begin by defining several auxiliary magnitudes. First, consider a fixed ball $B \equiv \mathcal{B}_{\overline{x},\overline{u}}$ centered at $\overline{x}$ with radius $\overline{u}$ and let a modified average cost function, $\tilde{\eta}_\mu^B$, and a modified relative value function, $\tilde{h}_\mu^B$, be defined according to

$$\tilde{\eta}_\mu^B \equiv \lim_{T\to\infty} E_{x,\mu}\left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{1}(X_t \in B) c_\mu(X_t)\right] \tag{31}$$

$$\tilde{h}_\mu^B(x) \equiv E_{x,\mu}\left[\sum_{t=0}^{\tau-1} \left(\mathbb{1}(X_t \in B) c_\mu(X_t) - \tilde{\eta}_\mu^B\right)\right]. \tag{32}$$

That is, $\tilde{\eta}_\mu^B$ and $\tilde{h}_\mu^B$ are the average–cost and to the relative value of an MDP that only incurs costs in the interior of B. Both of these terms can be shown to be uniformly bounded using the compactness of B and Assumption A.4) (for details, see [24]). Hence, $\tilde{h}_\mu^B(x)$ is bounded in absolute value by a constant C_B. Below we assume that B is fixed and we write $\tilde{h}_\mu(x)$ for the modified relative value function $\tilde{h}_\mu^B(x)$ for simplicity; similarly, we let $\tilde{\eta}_\mu \equiv \tilde{\eta}_\mu^B$.

Next, we define the "error term" $e_{s,\mu} \equiv \tilde{h}_\mu(Z_{s+1}) - (\Gamma_\mu \tilde{h}_\mu)(Z_s)$. Note that $e_{s,\mu}$ has zero mean and it is independent of any other $e_{s',\mu}$ unless $s' = s$. Because Γ_μ is a conditional expectation operator and because $\tilde{h}_\mu$ is bounded in absolute value by C_B, $e_{s,\mu}$ is bounded in absolute value by $2C_B$.

Finally, we observe that for all $\mu \in \mathcal{M}_{\varrho}$ the function $\Gamma_\mu \tilde{h}_\mu$ is uniformly continuous on B in the sense that for any $\varepsilon > 0$ there exists a $\delta > 0$ such that $\|x - z\| < \delta$ for all $x, z \in B$ implies $|(\Gamma_\mu \tilde{h}_\mu)(z) - (\Gamma_\mu \tilde{h}_\mu)(x)| < \varepsilon$. To see this, note that

$$\begin{aligned} &\left|\left(\Gamma_\mu \tilde{h}_\mu\right)(z) - \left(\Gamma_\mu \tilde{h}_\mu\right)(x)\right| \\ &\qquad \leq \int \left|\tilde{h}_\mu(y)\right| |p_{\mu(x)}(x, y) - p_{\mu(z)}(z, y)| \lambda(dy) \end{aligned} \tag{33}$$

where $p_a(x, y)$ is uniformly continuous according to Assumption A.2) and $|\tilde{h}_\mu(y)|$ is uniformly bounded as previously shown.

Proof of Proposition 2: We assume that a fixed ball $B \subset \mathbb{R}^d$ is given in (31) and (32), and that kernel-based reinforcement learning approximates the unknown MDP on B. That is, by a slight abuse of notation, we redefine the approximate average cost $\eta_{m,\mu}$ and the approximate relative value function $h_{m,\mu}$ in terms of the modified cost function $\mathbb{1}(x \in B) c_\mu(x)$. Then, we consider the decomposition

$$E_{x_0,\overline{\mu}} |\eta_{m,\mu} - \eta_\mu| \leq E_{x_0,\overline{\mu}} |\eta_{m,\mu} - \tilde{\eta}_\mu| + |\tilde{\eta}_\mu - \eta_\mu|. \tag{34}$$

Because of Assumption A.8), it is always possible to choose B so that the second term on the right-hand side of (34) is smaller than a given ε. The first term in (34) can be rewritten using Proposition 1 as

$$\begin{aligned} &E_{x_0,\overline{\mu}} |\eta_{m,\mu} - \tilde{\eta}_\mu| \\ &\quad \leq E_{x_0,\overline{\mu}} \left| \int_B \left(\hat{\Gamma}_{m,\mu} \tilde{h}_\mu - \Gamma_\mu \tilde{h}_\mu\right)(x) \pi_{m,\mu}(dx) \right| \\ &\qquad + E_{x_0,\overline{\mu}} \left[\int_B \left(\left|\hat{\Gamma}_{m,\mu} \tilde{h}_\mu\right| + \left|\Gamma_\mu \tilde{h}_\mu\right| \right)(x) \pi_{m,\mu}(dx) \right] \\ &\quad \leq E_{x_0,\overline{\mu}} \left| \sup_{x\in B} \left(\hat{\Gamma}_{m,\mu} \tilde{h}_\mu - \Gamma_\mu \tilde{h}_\mu\right)(x) \right| \\ &\qquad + 2C_B E_{x_0,\overline{\mu}}[\pi_{m,\mu}(B)]. \end{aligned}$$

Without loss of generality, we choose B such that $2C_B E_{x_0,\overline{\mu}}[\pi_{m,\mu}(B)] < \varepsilon$. With regard to the first term of the last inequality, we consider yet another decomposition

$$\begin{aligned} &E_{x_0,\overline{\mu}} \left| \sup_{x\in B} \left(\hat{\Gamma}_{m,\mu} \tilde{h}_\mu - \Gamma_\mu \tilde{h}_\mu\right)(x) \right| \\ &\quad \leq E_{x_0,\overline{\mu}} \left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \right. \\ &\qquad \left. \cdot \left(\left(\Gamma_\mu \tilde{h}_\mu\right)(Z_s) - \left(\Gamma_\mu \tilde{h}_\mu\right)(x) \right) \right| \qquad (35) \\ &\qquad + E_{x_0,\overline{\mu}} \left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) e_{s,\mu} \right|. \qquad (36) \end{aligned}$$

First, we investigate the "bias term" (35). We derived previously that $|\tilde{h}_\mu(x)| \leq C_B$ so that $|(\Gamma_\mu \tilde{h}_\mu)(Z_s) - (\Gamma_\mu \tilde{h}_\mu)(x)| \leq 2C_B$. Next, because of the uniform continuity of $\Gamma_\mu \tilde{h}_\mu$, we can

always guarantee that $|(\Gamma_\mu \tilde{h}_\mu)(Z_s) - (\Gamma_\mu \tilde{h}_\mu)(x)| \leq \varepsilon$ within a sufficiently small neighborhood of x, say, for all $||Z_s - x|| \leq u$. Then

$$\begin{aligned} E_{x_0,\overline{\mu}} &\left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \left(\left(\Gamma_\mu \tilde{h}_\mu\right)(Z_s) - \left(\Gamma_\mu \tilde{h}_\mu\right)(x) \right) \right| \\ &\leq E_{x_0,\overline{\mu}} \left[\sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) (2C_B \mathbb{1}(||Z_s - x|| > u) \right. \\ &\qquad \left. + \varepsilon \mathbb{1}(||Z_s - x|| \leq u)) \right] \\ &\leq 2C_B E_{x_0,\overline{\mu}} \left[\sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| > u) \right] + \varepsilon. \end{aligned} \tag{37}$$

The term in the expectation operator is bounded by one so that we can split the left term in (37) another time with respect to the condition $\sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| > u) > \delta$. Hence, it suffices to show that there exists $\delta > 0$ such that

$$\begin{aligned} 2C_B \left(\delta + P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \right. \right. \\ \left. \left. \cdot \mathbb{1}(||Z_s - x|| > u) > \delta \right) \right) \leq \varepsilon \end{aligned} \tag{38}$$

for all $\mu \in \mathcal{M}_\varrho$ to obtain that (37) is bounded by 2ε. Using again that $\sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) = 1$ and that $\mathcal{B}_{x,u} \subseteq N_{m,\mu}(x)$, without loss of generality

$$\begin{aligned} &P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| > u) > \delta \right) \\ &= P_{x_0,\overline{\mu}} \left(\inf_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| \leq u) < 1 - \delta \right) \\ &\leq P_{x_0,\overline{\mu}} \left(\inf_{x\in B} \frac{\underline{C}_k}{\#N_{m,\mu}(x)} \#_{\mu(x)} \mathcal{B}_{x,u} < 1 - \delta \right) \\ &\leq P_{x_0,\overline{\mu}} \left(\exists\, a \in A,\, 1 \leq i \leq C_v, \right. \\ &\qquad \left. D \in \mathcal{D}_{m,a} \colon \#_a \mathcal{B}_{v_i,u/2} < \frac{(1-\delta)\#D}{\underline{C}_k} \right) \\ &\leq P_{x_0,\overline{\mu}} \left(\bigcup_{a\in A, 1\leq i\leq C_v, D\in\mathcal{D}_{m,a}} \left\{ \#_a \mathcal{B}_{v_i,u/2} < \frac{(1-\delta)\#D}{\underline{C}_k} \right\} \right) \\ &\leq M C_v v(\mathcal{E}, m) \sup_{a\in A, x\in B} \\ &\qquad \cdot P_{x_0,\overline{\mu}} \left(\#_a \mathcal{B}_{x,u/2} < \frac{(1-\delta)\#N_{m,a}(x)}{\underline{C}_k} \right). \end{aligned}$$

We used the value $\epsilon > 0$ to bound the probability in (24) which appears also as the second term of the first inequality. Note that ϵ is arbitrary and it is different from the bound ε in (38). Here, $\mathcal{B}_{x,u}$ is a ball centered at x with radius u as previously described. We also used a covering of B in terms of C_v spheres of the form $\mathcal{B}_{v_i,u/2}$ with the property that any $\mathcal{B}_{x,u}$, $x \in B$ contains some $\mathcal{B}_{v_i,u/2}$ as a subset. $\mathcal{D}_{m,\mu}$ is the set of all sets $N_{m,\mu}(x)$ for some $x \in B$. Clearly, $\#\mathcal{D}_{m,\mu}$ is smaller than the mth shattering coefficient, $v(\mathcal{E}, m)$. The last term in this derivation can be chosen so as to satisfy (38) by choosing $\delta = \varepsilon/(4C_B)$ and using Assumption (26). Hence, altogether (35) is bounded by 2ε.

With respect to the "variance term" (36), because $e_{s,\mu}$ is bounded in absolute value by $2C_B$, it suffices to demonstrate that for any $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$2\delta + 2C_B P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \left| \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) e_{s,\mu} \right| > 2\delta \right) \leq \varepsilon.$$

This is by analogy to (38). The probability term can be bounded as follows:

$$\begin{aligned} &P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \left| \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) e_{s,\mu} \right| > 2\delta \right) \\ &\leq P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \left| \sum_{Z_s\in N_{m,\mu}(x)} k_{m,\mu}(Z_s, x) e_{s,\mu} \right| > \delta \right) \\ &\quad + P \left(\sup_{x\in B} \sum_{Z_s\in N^c_{m,\mu}(x)} k_{m,\mu}(Z_s, x) > \frac{\delta}{2C_B} \right) \\ &\leq P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \left| \sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)} \right| > \frac{\delta}{\overline{C}_k} \right) + \epsilon. \end{aligned} \tag{39}$$

We used (24) to derive the last inequality. In detail, the second probability term equals zero by (25) for a sufficiently large m. In our next step, we apply the law of iterated expectations in order to condition on the samples in $\mathcal{S}$

$$\begin{aligned} &P_{x_0,\overline{\mu}} \left(\sup_{x\in B} \left| \sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)} \right| > \frac{\delta}{\overline{C}_k} \right) \\ &= E_{x_0,\overline{\mu}} \left[P \left(\sup_{x\in B} \left| \sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)} \right| \right. \right. \\ &\qquad \left. \left. > \frac{\delta}{\overline{C}_k} \right| Z_0, \ldots, Z_m \right) \right] \\ &\leq E_{x_0,\overline{\mu}} \left[P \left(\bigcup_{a\in A,\, D\in\mathcal{D}_{m,a}} \left\{ \left| \sum_{Z_s\in D} \frac{e_{s,a}}{\#D} \right| > \frac{\delta}{\overline{C}_k} \right\} \right| \right. \\ &\qquad \left. \cdot\, Z_0, \ldots, Z_m \right) \Bigg] \\ &\leq M v(\mathcal{E}, m) E_{x_0,\overline{\mu}} \left[\sup_{a\in A,\, D\in\mathcal{D}_{m,a}} P \left(\left| \sum_{Z_s\in D} \frac{e_{s,a}}{\#D} \right| \right. \right. \\ &\qquad \left. \left. > \frac{\delta}{\overline{C}_k} \right| Z_0, \ldots, Z_m \right) \Bigg] \\ &\leq M v(\mathcal{E}, m) E_{x_0,\overline{\mu}} \\ &\quad \cdot \left[\sup_{a\in A, x\in B} \exp\left(-\frac{\varphi^2 (\#N_{m,a}(x)\delta/\overline{C}_k - 4C_B/\varphi)^2}{8\#N_{m,a}(x) C_B^2} \right) \right] \end{aligned}$$

for sufficiently large m. Here, we applied [27, Th. 2] using $|e_{s,a}|$ as a cost function; in particular, note that $g(Z_s)$ takes on values in the domain $[0,\ 2C_B]$. In addition, we used again the set $\mathcal{D}_{m,\mu}$ of neighborhoods of points in B. Furthermore, we bounded (39) by 3ϵ using (27), and we set $\delta = \varepsilon/4$ and $\epsilon = \varepsilon/(8C_B)$.

Collecting terms, we find that $E_{x_0,\overline{\mu}}|\eta_{m,\mu} - \eta_\mu|$ is bounded by 3ε for sufficiently large m. Because the size of this bound for m is independent of x_0 and $\overline{\mu}$, the convergence occurs furthermore uniformly which completes the proof. ■

APPENDIX III
PROOF OF THEOREM 1

Prior to the proof of the theorem we establish two auxiliary conditions. First, we investigate the limiting behavior of the number of samples in each of the subsets $\mathcal{S}_a$, defined as m_a in Section IV, as m goes to infinity. The following properties are proven in [24] using the splitting idea together with a law of large numbers

$$m_a = \Omega_P(m) \tag{40}$$

$$l/m_a \overset{m\to\infty}{\longrightarrow} 0 \quad \text{a.s.} \tag{41}$$

Hence, m_a grows at least proportionally to m and, for adaptive neighborhoods, the proportion of the m_a samples located in $N_{m,a}(x)$ goes to zero ($l = \#N_{m,a}(x)$).

Second, note that, because B is compact and the transition density $p_a(x,\, y)$ is continuous (A.1), $p_a(x,\, y)$ is bounded away from zero by a constant $\underline{C}_p$ and it is bounded above by another constant $\overline{C}_p$. Thus, the transition kernel satisfies the conditions $\underline{C}_p\lambda(D) \le P_\mu(x,\, D) \le \overline{C}_p\lambda(D)$ for all $D \subseteq B$, $\mu \in \mathcal{M}_\varrho$. This condition is furthermore inherited by the invariant measure, π_μ:

$$\underline{C}_p\lambda(D) \le \pi_\mu(D) = \int \pi_\mu(dy)P_\mu(y,\, D) \le \overline{C}_p\lambda(D). \tag{42}$$

Proof of Theorem 1: We verify the conditions (24)–(27) of Proposition 2. For approaches 1) and 2), (24) is obvious using $\underline{C}_k = \overline{C}_k = 1 - \vartheta$. For the nearest neighbor approach 3), we choose $\underline{C}_k = (1-\vartheta)\underline{v}$ and $\overline{C}_k = (1-\vartheta)\overline{v}$, respectively. Condition (25) is obvious given Assumption A.6).

With regard to (26), we distinguish two cases depending on the event

$$E \equiv \left\{ \frac{C_1 \#N_{m,a}(x)}{m_a} \le \frac{\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})}{2} \right\}.$$

Conditioning on E, we obtain that

$$\begin{aligned}
&\sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}\big(\#_a\mathcal{B}_{x,\,u/2} < C_1\#N_{m,a}(x)\big) \\
&\quad\le \sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}\left(\frac{\#_a\mathcal{B}_{x,\,u/2}}{m_a} - \pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2}) < -\frac{\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})}{2}\right) \\
&\qquad + \sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}\left(\frac{C_1\#N_{m,a}(x)}{m_a} > \frac{\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})}{2}\right).
\end{aligned} \tag{43}$$

We analyze the first of these two terms by using the large deviation result of [27, Th. 2]. This gives

$$\begin{aligned}
&\sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}\left(\frac{\#_a\mathcal{B}_{x,\,u/2}}{m_a} - \pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2}) < -\frac{\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})}{2}\right) \\
&\quad\le \frac{1}{2}\sup_{a\in A,\, x\in B}\left\{\exp\left(-\frac{\varphi^2(m_a\pi_{\overline{\mu}}(\mathcal{B}_{x,u/2})/2 - 2/\varphi)^2}{2m_a}\right)\right\}.
\end{aligned} \tag{44}$$

Note that we applied the Markov chain version of Hoeffding's inequality to the subchain associated with the set $\mathcal{S}_a$. Because u is fixed and $\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})$ is uniformly bounded by (42), expression (44) converges to zero as m_a goes to infinity using (40).

In order to deal with the second term in (43), we treat separately adaptive neighborhood approaches, where $l = \#N_{m,a}(x)$ is a fixed function of m_a, and fixed neighborhood approaches, where $N_{m,a}(x)$ contains all samples in the fixed region $U_{m,a}(x)$ and is therefore random: For adaptive neighborhoods, condition (41) ensures that E is eventually true for sufficiently large m_a, so that the second term in (43) equals zero. For fixed neighborhoods, we reformulate the second term in (43) using the neighboring region $U_{m,a}(x)$ instead of $N_{m,a}(x)$. As previously shown, we consider the limiting value as m goes to infinity, applying [27, Th. 2]:

$$\begin{aligned}
&\sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}\left(\frac{C_1\#_aU_{m,a}(x)}{m_a} > \frac{\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2})}{2}\right) \\
&\quad\le \frac{1}{2}\sup_{a\in A, x\in B}\left\{\exp\left(-\frac{\varphi^2(m_a\pi_{\overline{\mu}}(\mathcal{B}_{x,u/2})/4 - 2C_1/\varphi)^2}{2m_aC_1^2}\right)\right\}.
\end{aligned} \tag{45}$$

We used the fact that $\pi_{\overline{\mu}}(U_{m,a}(x)) < (\pi_{\overline{\mu}}(\mathcal{B}_{x,\,u/2}))/4$ for large m by (42) and (11). As previously shown, (45) and, hence, also (43) converge to zero by (40).

We deal with (27) by analogy to (26). In detail, we use a decomposition of the exponential term in (27) based on the scalar w

$$\begin{aligned}
&e^{-C_2\#N_{m,a}(x)} \\
&\qquad\le e^{-C_2w}\mathbb{1}(\#N_{m,a}(x) \ge w) + \mathbb{1}(\#N_{m,a}(x) < w).
\end{aligned}$$

Using this decomposition, the left-hand side of (27) can be bounded by the term

$$\begin{aligned}
v(\mathcal{E},\, m)\Bigg(&\sup_{a\in A,\, x\in B} e^{-C_2w} \\
&+ \sup_{a\in A,\, x\in B} P_{x_0,\overline{\mu}}(\#N_{m,a}(x) < w)\Bigg).
\end{aligned} \tag{46}$$

As in the case of (26), we choose $w = l$ for adaptive neighborhoods so that $P_{x_0,\overline{\mu}}(\#N_{m,a}(x) < w)$ equals zero for suffi-

ciently large m using (41). For fixed neighborhoods, we choose $w = m_a \pi_{\overline{\mu}}(U_{m,a}(x)) - \zeta$. Then, by analogy to (44)

$$P_{x_0,\overline{\mu}}(\#N_{m,a}(x) < w) \leq \frac{1}{2} \sup_{a \in A,\, x \in B} \left\{ \exp\left(-\frac{\varphi^2(\zeta - 2/\varphi)^2}{2m_a} \right) \right\}.$$

Note that in order for (46) to converge to zero we require both $w \to \infty$ and $\zeta^2/m_a \to \infty$ quickly by comparison to the growth of the shattering coefficient $v(\mathcal{E}, m)$. For example, consider the definition $\zeta = m_a \pi_{\overline{\mu}}(U_{m,a}(x))^{\sqrt{2}}$. In this case we have $w = m_a \pi_{\overline{\mu}}(U_{m,a}(x)) - m_a \pi_{\overline{\mu}}(U_{m,a}(x))^{\sqrt{2}}$ which goes to infinity provided condition (11) and $m_a \pi_{\overline{\mu}}(U_{m,a}(x)) \to \infty$ as desired. On the other hand, we have $\zeta^2/m_a = m_a \pi_{\overline{\mu}}(U_{m,a}(x))^2$. We mentioned that, for the averaging approaches 1)–3) proposed in Section IV, the shattering coefficient $v(\mathcal{E}, m)$ is bounded by $(m+1)^{d_{\mathrm{VC}}}$. We obtain the following sufficient condition for the convergence of (46):

$$E_{x_0,\overline{\mu}}[m_a \pi_{\overline{\mu}}(U_{m,a}(x))^2] - \log(m+1) \to \infty. \tag{47}$$

Using (40) and (42), (12) is sufficient to guarantee that (47) holds. Altogether, (24)–(27) hold for approaches 1)–3) from which we conclude that these approaches are consistent by Proposition 2. Again, for a more detailed derivation, see [24]. ■

APPENDIX IV
PROOF OF THEOREM 2

An important aspect of this proof is that the perturbation constant ϱ is decreasing in Theorem 2 according to Assumption A.7); in contrast, ϱ was fixed above.

First, consider the case $\hat{\eta}_m < \eta^*$. Because convergence is uniform for all $\mu \in \mathcal{M}_\varrho$ in Theorem 1, the result of the theorem holds specifically for the choice $\mu = \hat{\mu}_m$. That is, we have $E_{x_0,\overline{\mu}}|\hat{\eta}_m - \eta_{\hat{\mu}_m}| < \varepsilon$ for arbitrary $\varepsilon > 0$ and sufficiently large m. However, we also have $|\hat{\eta}_m - \eta^*| \leq |\hat{\eta}_m - \eta_{\hat{\mu}_m}|$ because η^* attains the minimum costs and hence $\eta_{\hat{\mu}_m} \geq \eta^*$. Taking expectations on both sides gives $E_{x_0,\overline{\mu}}|\hat{\eta}_m - \eta^*| < \varepsilon$.

In the case where $\hat{\eta}_m > \eta^*$, let μ^*_ϱ denote the "projection" of μ^* onto $\mathcal{M}_\varrho$, that is, the strategy obtained by setting the minimal probability of each action to ϱ and by renormalizing the remaining probabilities appropriately. Also, let η^*_ϱ denote the average–costs associated with μ^*_ϱ so that $\hat{\eta}_m - \eta^* = (\hat{\eta}_m - \eta^*_\varrho) + (\eta^*_\varrho - \eta^*)$. Using Lemma 1, the last term can be written as $\eta^*_\varrho - \eta^* = \int (\Gamma_{\mu^*_\varrho} h^* - \Gamma_{\mu^*} h^*)(x) \pi_{\mu^*_\varrho}(dx)$.

This expression is made small by choosing ϱ small or, equivalently, choosing m sufficiently large. Because $\hat{\eta}_m - \eta^* > 0$ and $|\eta^*_\varrho - \eta^*|$ is small, it is without loss of generality to assume also that $\hat{\eta}_m > \eta^*_\varrho$. Next, recall that $\hat{\eta}_m$ and $\hat{h}_m$ achieve the minimum and the pointwise minimum of $\eta_{m,\mu}$ and $h_{m,\mu}$, respectively. Therefore, we have $\eta_{m,\mu^*_\varrho} \geq \hat{\eta}_m$ and thus $|\hat{\eta}_m - \eta^*_\varrho| \leq |\eta_{m,\mu^*_\varrho} - \eta^*_\varrho|$, where the last term can again be bounded using Theorem 1. ■

ACKNOWLEDGMENT

The authors would like to thank S. Meyn for helpful suggestions. The scientific integrity of this paper rests with its authors.

REFERENCES

[1] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, pp. 9–44, 1988.
[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.
[3] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large-scale dynamic programming," *Mach. Learn.*, vol. 22, pp. 59–94, 1996.
[4] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, pp. 161–178, 2002.
[5] A. Arapostathis, V. S. Borkhar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus, "Discrete-time controlled Markov processes with average cost criterion: A survey," *SIAM J. Control Optim.*, vol. 31, no. 2, pp. 282–344, 1993.
[6] S. P. Meyn, "The policy iteration algorithm for average reward Markov decision processes with general state space," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 1382–1393, Oct. 1997.
[7] J. Rust, "Using randomization to break the curse of dimensionality," *Econometrica*, vol. 65, no. 3, pp. 487–516, 1997.
[8] G. Gordon, "Approximate solutions to Markov decision processes," Ph.D. dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, 1999.
[9] G. J. Gordon, "Stable fitted reinforcement learning," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds. Cambridge, MA: MIT Press, 1996, vol. 8.
[10] W. L. Baker, "Learning via stochastic approximation in function space," Ph.D. dissertation, Div. Appl. Sci., Harvard Univ., Cambridge, MA, 1997.
[11] V. S. Borkar, "A learning algorithm for discrete-time stochastic control," *Probab. Eng. Inform. Sci.*, vol. 14, pp. 243–258, 2000.
[12] R. Munos and A. Moore, "Variable resolution discretization in optimal control," *Mach. Learn.*, vol. 49, pp. 291–324, 2002.
[13] J. N. Tsitsiklis and V. R. Konda, "Actor-critic algorithms," Tech. Rep., Lab. Inform. Decision Systems., Mass. Inst. Technol.., Cambridge, MA, 2001, Preprint.
[14] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. New York: Springer-Verlag, 1993.
[15] E. A. Nadaraya, "On estimating regression," *Theor. Probab. Appl.*, vol. 9, pp. 141–142, 1964.
[16] G. S. Watson, "Smooth regression analysis," *Sankhyã Series A*, vol. 26, pp. 359–372, 1964.
[17] C. J. Stone, "Optimal global rates of convergence for nonparametric regression," *Ann. Stat.*, vol. 10, no. 4, pp. 1040–1053, 1982.
[18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.
[19] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: SIAM, 1992.
[20] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1983.
[21] C. J. Stone, "Consistent nonparametric regression," *Ann. Stat.*, vol. 5, no. 4, pp. 595–645, 1977.
[22] D. Ormoneit and P. W. Glynn, "Kernel-based reinforcement learning in average–cost problems: An application to optimal portfolio choice," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001.
[23] J. Fan and I. Gijbels, *Local Polynomial Modeling and Its Applications*. London, U.K.: Chapman & Hall, 1996.
[24] D. Ormoneit and P. W. Glynn, "Kernel-based reinforcement learning in average–cost problems," Tech. Rep., Dept. Comput. Sci., Stanford Univ., Stanford, CA, 2001.
[25] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.
[26] X-R. Cao, "Perturbation realization, potentials, and sensitivity analysis of Markov processes," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 1382–1393, Oct. 1997.
[27] P. W. Glynn and D. Ormoneit, "Hoeffding's inequality for uniformly ergodic Markov chains," *Stat. Probab. Lett.*, vol. 56, pp. 143–146, 2002.
[28] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[29] D. Ormoneit and T. Hastie, "Optimal kernel shapes for local linear regression," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 540–546.

[30] L. Devroye, "The uniform convergence of nearest neighbor regression function estimators and their application in optimization," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 142–151, Feb. 1978.

Dirk Ormoneit received the Ph.D. degree in computer science from the Technische Universität München, Germany, in 1998.

He then joined Stanford University, Stanford, CA as a Research Associate at the Departments of Statistics and Computer Science. In 2001, he moved into the financial industry and is now Fund Manager for Statistical Arbitrage at Marshall Wace Asset, London, U.K. His research focus is on stochastic control and the application of statistical models for trading.

Peter Glynn received the Ph.D. degree in operations research from Stanford University, Stanford, CA, in 1982.

He then joined the faculty of the University of Wisconsin at Madison, where he held a joint appointment between the Industrial Engineering Department and Mathematics Research Center, and courtesy appointments in Computer Science and Mathematics. In 1987, he returned to Stanford University, where he is now the Thomas Ford Professor of Engineering in the Department of Management Science and Engineering. He also has a courtesy appointment in the Department of Electrical Engineering. He has research interests in computational probability, queueing theory, statistical inference for stochastic processes, and stochastic modeling.

Dr. Glynn is a Fellow of the Institute of Mathematical Statistics