Kernel-Based Reinforcement Learning in Average-Cost Problems

Dirk Ormoneit* **and Peter Glynn**†

*Department of Computer Science

†Department of Management Science

Stanford University, Stanford, CA 94305

ormoneit@cs.stanford.edu, http://robotics.stanford.edu/

Abstract

Reinforcement learning (RL) is concerned with the identification of optimal controls in Markov Decision Processes (MDP) 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 [1], [2]. This proceeding 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 work, 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 and we discuss methods to extend this algorithm to allow for online operation.

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], [3]. 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 [4], [5]. To circumvent these shortcomings, a nonparametric approach to reinforcement learning has been recently proposed by Ormoneit and Sen [6]. Specifically, Ormoneit and Sen 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. Our emphasis is on the application of these methods 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 [7], [2]. 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 [8]. 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 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)*:

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

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. Here 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 non-linear 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.

The remainder of this work is organized as follows: In Section II we state our assumptions and we characterize the average-cost reinforcement learning problem formally. In Section III we introduce the local averaging operator. Sections IV and V 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 VI we discuss practical issues for online operation and in Section VII we present conclusions.

II. Preliminaries and Problem Formulation

Consider an MDP defined by a sequence of states X_t taking values in $I\!R^d$ 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 $(\Omega, \mathcal{F}, P)$. The collection of σ-algebras $\mathcal{F}_t \equiv \sigma(X_1, a_1, \ldots, X_t, a_t)$ generated by the random variables in $\mathcal{X}$ and A form a filtration of $\mathcal{F}$ characterizing the way information is gathered over time. Strategies, policies, or controls are understood as fixed rules to choose an action at time t based on the information represented by the sets in $\mathcal{F}_t$. Even though arbitrary A-valued processes adapted with respect to the σ-algebras $\mathcal{F}_t$ can be used to define strategies in principle, it suffices to consider stationary Markov strategies that depend on the present state of the system, X_t, alone for the purpose of optimal control [9]. 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. By analogy to the transition kernel, we define an induced cost function according to $c_\mu(x) \equiv \sum_{a=1}^{M} c(x, a)\mu(x, a)$ for all $\mu \in \mathcal{M}$. While in practice costs frequently depend also on the successor state X_{t+1}, this additional dependency is usually suppressed in dynamic programming by considering expected costs. Note, however, that this definition is strictly speaking inappropriate in reinforcement learning because the conditional expectation is unknown in this case. Regardless, we ignore possible dependencies of c on X_{t+1} for simplicity in this work and refer the interested reader to an alternative problem formulation in [6].

A. *Assumptions*

We impose the following assumptions to guarantee sufficient regularity of the MDP to allow for learning:

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

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

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

For details on the implications of these assumptions, see [10].

The following assumption ensures that long-run average costs and unique invariant distributions are well-defined under arbitrary policies:

A.4: There exists a probability measure ν on $I\!R^d$, $\varphi > 0$, and an integer $q \geq 1$ such that

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

for all $x \in I\!R^d$.

In detail, Assumption A.4 is easily enough to guarantee the existence of *small* sets, $S \in \mathcal{B}(I\!R^d)$, that are *accessible* in the sense that the expected hitting time $\sup_{\mu \in \mathcal{M}} E_{x,\mu}[\tau_S]$ is finite for all $x \in I\!R^d$, where $\tau_S \equiv \inf \{t \in \{1, 2, \ldots, \infty\} | X_t \in S\}$. As usual we turn a designated small set S into an "artificial atom", ϖ, by using Nummelin splitting [11]. This splitting construction provides the necessary regenerative structure of the Markov chain and will be used repeatedly throughout this work (see also Section II-C). It follows that any chain that is induced by an arbitrary policy $\mu \in \mathcal{M}$ is automatically c_μ-regular in the sense that its long-run average costs are finite [10]. Furthermore, the induced chain is positive Harris and possesses a unique invariant measure, π_μ. Additional assumptions regarding details of the learning algorithm are listed throughout the paper.

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 [6]. The average-cost of a policy μ is defined formally according to

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

The average-cost η_μ is defined uniquely and independently of the starting position x_0 under the assumptions in Section II-A (see, for example [10]). Specifically, 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

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

Here Γ_a is a "conditional expectation operator", defined according to $(\Gamma_a h)(x) \equiv E_{x,a}[h(X_1)]$.* Under the assumptions of this paper – if finite solutions η^* and h^* to (1) exist – a policy μ^* is optimal if and only if

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

For a detailed discussion of this relationship, see [7], [10], [3], [8].

*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 [8] 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 II-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 can be interpreted as 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.

C. Regeneration Structure

Both [8] and this paper rely heavily on the existence of regenerative events that can be used to "split" a given Markov chain. Roughly, we define a recurring event, R, that decouples the trajectory after its occurrence from the previous history of the chain. This construction is particularly useful to obtain an alternative expression for the relative value function h_μ in (6). Specifically, 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, it suffices to consider the history of the chain up to τ to express h_μ:

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

Then (7) together with the average-cost (4) define a solution to Poisson's equation.

Techniques for the formal construction of the regenerative event R have been suggested by Nummelin [11] and by Athreya and Ney [12]. The idea is to use a decomposition of the q-step transition kernel $P_\mu^q(x,B)$ of the form:

$$P_\mu^q(x,B) \;=\; \varphi\nu(B) + (1-\varphi)\frac{P_\mu^q(x,B)-\varphi\nu(B)}{1-\varphi}.$$

Here δ and λ are the constant and the minorizing measure of equation (3), respectively. Note that the last expression can be interpreted as a mixture distribution. We can imagine that the next state of the Markov chain is generated according to the measure ν with probability δ and according to $\frac{P_\mu^q(x,B)-\varphi\nu(B)}{1-\varphi}$ with probability $1-\varphi$. If ν is chosen, the distribution of the successor state is furthermore independent of x, implying the desired regeneration property. Hence the first multiple of q at which ν is chosen is a suitable definition of the regenerating event R. For a more detailed discussion of Nummelin splitting, we refer the interested reader again to [10].

D. Generation of Training Data Using Simulation

In reinforcement learning, we compute an approximately optimal strategy, $\hat{\mu}$, using historical observations. These observations will then serve as a set of training data to construct an approximate conditional expectation operator, $\hat{\Gamma}_{m,a}$, in Section III. In detail, we assume that the training data set, $\mathcal{S}$, is generated by simulating the MDP for m steps using a fixed initial policy, $\bar{\mu}$, and a fixed initial state, $x_0 \in I\!R^d$. Here it is important that $\bar{\mu}$ chooses each action with positive probability to guarantee sufficient "exploration" of the state-space. Formally, we let $\bar{\mu} \in \mathcal{M}_{\bar{\varrho}}$ for a positive constant $\bar{\varrho} > 0$. It will be formally convenient to distinguish the states generated during the simulation, $\mathcal{S} = \{z_0, \ldots, z_m\}$, from the states of the MDP in a more general context, $\{X_1, \ldots, X_\infty\}$. That is, we will assume that $\{Z_0, \ldots, Z_m\}$ is a new collection of random variables, adapted with respect to the filtration of $\mathcal{F}$ defined above and distributed according to $P_{x_0,\bar{\mu}}(X_0, \ldots, X_m)$. Then $\mathcal{S}$ is a sequence of realizations of these random variables with an intuitive interpretation as a m-step sample trajectory of the MDP. To ease notation, we will not be rigorous below in our distinction between random variables and samples. Depending on the context, we may use the symbol $Z_s, s = 1, \ldots, m$ to emphasize the random nature of the training samples in mathematical derivations, while we use z_s to emphasize their computational role. $\mathcal{S}$ may sometimes be used to denote the set $\{Z_0, \ldots, Z_m\}$ rather than a collection of its realizations. Besides the state variables, we also have the actions $\{a_0, \ldots, a_{m-1} | a_s \sim \bar{\mu}(z_s)\}$ and the costs $\{c(z_s, a_s) | 0 \leq s < m\}$ that were generated during the simulation at our disposal for learning. Because a specific action is associated with each state, A partitions $\mathcal{S}$ into subsets of the form $\mathcal{S}_a \equiv \{Z_s \in \mathcal{S} | a_s = a\}$. We denote the cardinality of each of these subsets as m_a, i.e. $m_a \equiv \#\mathcal{S}_a$, and we also define a counting function, $\#_a$, for the number of samples in S that are located within a generic set D according to $\#_a(D) \equiv \sum_{s'=0}^{m-1} \mathbb{1}(a_{s'} = a, Z_s \in D)$. Next, we propose an approximate expectation operator based local averaging with respect to the data in $\mathcal{S}_a$.

III. 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 semi-parametric regression approaches such as linear estimators or neural networks fail to generate stable learning algorithms in practice (for a discussion, see [5]). Below we demonstrate that a class of regression methods particularly suited for approximate dynamic programming are methods based on "local averaging". These models have been studied extensively in the statistics literature both from a practical and a theoretical viewpoint [13], [14], [15], [16], [17],

[18], [19].

We consider nonparametric regression operators that can be written as local averages of the form

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

Here h is a generic relative value 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 $\mathbb{R}^d \times \mathbb{R}^d$. Recall from Section II that $\mu(\cdot, x) \in \Delta(A)$ is a distribution over all actions. The fundamental ideal 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, because Γ_μ depends on μ, we only use those states Z_s 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 [20].

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 more complicated lattice rules, for instance [21], [22]. We formally characterize such a partition in terms of a mapping, $U : \mathbb{R}^d \rightarrow \mathcal{P}(\mathbb{R}^d)$, that assigns each $x \in \mathbb{R}^d$ to a subset of $\mathbb{R}^d$. Based on U, we define the weighting function $k_{m,a}(z_s, x)$ for all $z_s \in \mathcal{S}$ based on the simulated actions

$\{a_0, \ldots, a_{m-1} | a_s \sim \bar{\mu}(z_s)\}$:

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

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 that correspond to the "leaves" of the tree. This partition is typically chosen such that each rectangle contains an approximately equal numbers of observations and hence the neighboring regions are adaptive with respect to the local data design in this case. More formally, a k-d-tree implies a partition function U by analogy to the grid-based approach, yet U is dependent on the training data set $\mathcal{S}$. Based on this random function, the weighting function can be defined formally as in (9). Below 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 [23].

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:

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

Here $v_{l,j}$ is a decreasing sequence of scalars satisfying $\overline{v}/l > v_{l,j} \geq \underline{v}/l > 0$ and $\sum_{j=0}^{l-1} v_{l,j} = 1$. For example, $v_{l,j}$ can be defined as a function of j using a uniform, a triangular, a quadratic, or a Gaussian kernel (for details, see [15]). 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 to 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 through 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" [6], 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 makes a considerable difference in our 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 $U_{m,a}(x) \equiv \{y \in I\!R^d | \exists z_s \in N_{m,a}(x) : ||y - x|| \leq ||z_s - x||\}$. Here the subscripts make explicit the possible dependency of the neighborhood on the sample size and on the chosen action. To prove the consistency of the averaging approaches 1 to 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 the conditions:

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

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

for all $x \in I\!R^d, a \in A$.

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

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

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

for all $x \in I\!R^d, a \in A$.

Conditions similar to (11) through (14) are standard in the theory of pattern recognition and are easy to satisfy in practice [20]. 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 to 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 IV, and then we derive theoretical properties of this approximation in Section V.

IV. Approximate Dynamic Programming

Above we suggested to substitute the unknown Γ_a in (1) with one of the the approximate expectation operators $\hat{\Gamma}_{m,a}$ defined in the previous section. The approximate average cost optimality equation takes the form (2):

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

In this section, we derive numerical methods for the solution of (2). Given a pair of solutions $\hat{\eta}_m$ and $\hat{h}_m$, it is then straightforward to define an approximately optimal strategy by analogy to (5):

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

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 IV-A and we introduce an additional condition for the solubility of this artificial MDP in Section IV-B. In Section IV-C, we suggest an implementation of kernel-based reinforcement learning using finite-state dynamic programming.

A. An "Artificial" MDP

Above 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 we can 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 non-linear 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.† In a sense, local averaging has therefore a special status in the class of nonlinear regression methods that makes it particularly useful for reinforcement learning.

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 $\hat{h}_m$ 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.

†The majority of the mentioned approximations can be interpreted as a special case of local averaging using the notion of "equivalent kernels" [19]. However, the equivalent kernel typically violates the positivity and normalization constraints on $k_{m,a}(z_s, x)$.

B. "Trembling Hand Policies"

In this section, we consider the stability of the artificial MDP implied by the transition kernel (16). Essentially we would like to recover the same regeneration structure that was established for the original MDP in Section II-C. 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 [8].

As in Section II-A, focus is on the existence of a suitable minorizing measure ν for Nummelin spitting. A fundamental requirement is that certain sets S need to be uniformly "accessible", defining the support of ν. We deal first with the issue of accessibility with respect to the artificial kernel $K_{m,\mu}(x,A)$ defined in (16). Note that only the sets $S \in \mathcal{B}(I\!R^d)$ with $z_{s+1} \in S$ for some s can potentially be accessible. Not all of these sets need to be accessible, however, for $\mu(a,x)$ may well be zero for deterministic policies. In the extreme situation where $\mu(a, z_{s+1}) = 1\!\!1(a_s = a)$ transitions occur only within the subsets $\mathcal{S}_a$ for $a = 1, \ldots, M$ so that no single point can be reached with positive probability in any number of transitions from points outside its own subset. A simple way to prevent such inaccessible cases is to require that $\mu(a,x) \geq \varrho$ for some positive constant ϱ. This assumption is fairly standard in the reinforcement learning community where it is usually quoted in the context of the "exploitation-exploration dilemma" [24], [25]. A game-theoretic interpretation is that of a player with a "trembling hand" that causes him to commit errors sporadically. Another interpretation is that the trembling protects the player against superior strategies of his opponents via randomization. Mathematically, trembling hand policies simply correspond to elements of the perturbed action space $\mathcal{M}_\varrho$ defined in Section II. 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.

Below policies are meant to be elements of $\mathcal{M}_\varrho$ where ϱ is contingent on m in accordance with Assumption A.7. For finite m every state in $\mathcal{S}$ can be reached from any other state with a probability of at least $\varrho\vartheta > 0$. It remains to define the minorizing measure $\nu^K(B)$ by analogy to (3). An obvious candidate is the counting measure

$$\nu^K(B) \equiv \frac{1}{m} \sum_{s=0}^{m-1} \mathbb{1}(Z_{s+1} \in A). \tag{19}$$

$\nu^K(B)$ 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 the regenerative decomposition of Section II-C. As described above, we interpret τ as the corresponding renewal time.

C. "Self-Approximating Property"

We saw above that the artificial MDP defined in Section IV-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 in order to implement this algorithm on a computer system which significantly complicates the analysis of dynamic programming [21]. In kernel-based reinforcement learning, the AACOE is itself an approximation. However, the AACOE can 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 by Rust [26], [27].

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 $\hat{\eta}_m$ at the locations z_s that are consistent with equation (2):

$$\hat{\eta}_m + \hat{h}_m(z_s) = \min_a \{c(z_s, a) + (\hat{\Gamma}_{m,a}\hat{h}_m)(z_s)\} \tag{20}$$

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

$$\hat{h}_m(x) \equiv \min_a \{c(x, a) + (\hat{\Gamma}_{m,a}\hat{h}_m)(x)\} - \hat{\eta}_m. \tag{21}$$

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 (20) 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 III and IV-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-space version of the first step for the solution of the AACOE, (20):

$$\hat{\eta}_m + \hbar_m(i) = \min_a \{\mathbf{c}_{m,a}(i) + (\Phi_{m,a}\hbar_m)(i)\} \tag{22}$$

for $i = 1, \ldots, m$. The correspondence between (22) and (20) is extremely helpful to determine $\hat{\eta}_m$ and $\hbar_m$ in practice, because it reduces the solution of (20) 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 [7], [9], [3]). 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 (22). Because algorithmic details are widely known we will not repeat them here but assume that an efficient algorithm for the solution of (22) is available below. As indicated above 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 this algorithm. The values of $\hat{h}_m(x)$ at new locations is then computed using (21) so that $\hat{\eta}_m$ and $\hat{h}_m(x)$ constitute solutions to (2). The statistical properties of these solutions are the topic of the next section.

V. Consistency

In Section IV 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 relatively easily using the contraction property of the approximate Bellman operator, as demonstrated by Ormoneit and Sen in [6]. For average-cost problems, a promising link between the approximate and the true MDP is that both can be decomposed using a common small set S as shown in Section IV-B. Below we use this decomposition to prove consistency in several steps: First, we demonstrate 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.

In the first two steps of the proof, we establish conditions under which the approximation error — measured in terms of $E_{x_0,\bar{\mu}}|\eta_{m,\mu} - \eta_\mu|$ — converges to zero uniformly for all strategies. We denote an approximation method satisfying this condition as a *uniformly consistent* method. As a minimum prerequisite to obtain uniform consistency, it seems intuitively clear that the approximate expectation operator $\hat{\Gamma}_{m,\mu}$ should converge to the true Γ_μ, for $\hat{\Gamma}_{m,\mu}$ is an essential building block for the construction of $\eta_{m,\mu}$. More formally, our goal is to relate the approximation error $E_{x_0,\bar{\mu}}|\eta_{m,\mu} - \eta_\mu|$ to the approximation error of a single application of the approximate expectation operator $\hat{\Gamma}_{m,\mu}$. This task can be reinterpreted as a perturbation problem: How do small perturbations in the transition kernel affect the average cost of the chain? Perturbations have been studied in the control literature with some focus on denumerable state-spaces (e.g. [28]). Because we are concerned with continuous state-spaces in this work, we first derive a proposition adapting these results to our current framework:

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

$$\begin{aligned} \eta_{m,\mu} - \eta_\mu &= \int (\hat{\Gamma}_{m,\mu} h_\mu - \Gamma_\mu h_\mu)(x) \pi_{m,\mu}(dx), \qquad (23) \\ h_{m,\mu}(x) - h_\mu(x) &= -(h_{m,\mu}(x) + C_h)(\eta_{m,\mu} - \eta_\mu) \end{aligned}$$

$$+E_{m,x}\left[\sum_{j=0}^{\tau-1}(\hat{\Gamma}_{m,\mu}h_\mu - \Gamma_\mu h_\mu)(X_j)\right], \tag{24}$$

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.

Equation (23) 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 III, 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. [13], [14], [15], [16], [17], [18]). Our next result summarizes these conditions in a form that is suitable for the weighting functions described in Section III. For its proof we need a Markov chain version of Hoeffding's inequality described in Appendix A as well as an additional "approximatibility" assumption:

A.8: The sequence of balls $\mathcal{B}_{\bar{x},u} \equiv \left\{z \in I\!R^d : ||z-\bar{x}|| \leq u\right\}$ satisfies the condition

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

uniformly for all $x \in I\!R^d, \mu \in \mathcal{M}$ and for some $\bar{x} \in I\!R^d$.

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, $\bar{x}$. 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 $I\!R^d$ 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 $0 < \underline{C}_k \leq \overline{C}_k$ such that for all $\delta > 0$, $\varrho > 0$, $u > 0$, $C_1 > 0$, $C_2 > 0$ and all closed balls $\mathcal{B} \subseteq I\!R^d$ the weighting function $k_{m,\mu}(Z_s, x)$ satisfies the following conditions for all $x \in \mathcal{B}$:

$$\frac{\underline{C}_k}{\#N_{m,\mu}(x)} \leq k_{m,\mu}(Z_s, x) \leq \frac{\overline{C}_k}{\#N_{m,\mu}(x)}, \qquad Z_s \in N_{m,\mu}(x), \tag{25}$$

$$\sum_{Z_s \in N^c_{m,\mu}(x)} k_{m,\mu}(Z_s, x) \overset{m \to \infty}{\longrightarrow} 0 \qquad \text{uniformly.} \tag{26}$$

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

$$v(\mathcal{E}, m) \sup_{a \in A, x \in B} P_{x_0, \bar{\mu}} \left(\#_a \mathcal{B}_{x,u/2} < C_1 \# N_{m,a}(x) \right) \quad \overset{m \to \infty}{\longrightarrow} \quad 0, \tag{27}$$

$$v(\mathcal{E}, m) E_{x_0, \bar{\mu}} \left[\sup_{a \in A, x \in B} e^{-C_2 \# N_{m,a}(x)} \right] \quad \overset{m \to \infty}{\longrightarrow} \quad 0 \tag{28}$$

uniformly for all $x_0 \in I\!R^d$, $\bar{\mu} \in \mathcal{M}_\varrho$.

Then

$$E_{x_0, \bar{\mu}} |\eta_{m,\mu} - \eta_\mu| \overset{m \to \infty}{\longrightarrow} 0 \tag{29}$$

uniformly for all $x_0 \in I\!R^d$ and all $\mu \in \mathcal{M}_{\bar{\varrho}}$.

Remember from Section II-D that x_0 and $\bar{\mu}$ are a fixed initial state and a fixed exploration strategy, respectively. The Assumption (25) guarantees that the weights are essentially uniform in the interior of the neighborhood $N_{m,a}(x)$, and Assumption (26) confirms that any weight outside $N_{m,a}(x)$ becomes unimportant asymptotically. The Assumptions (27) and (28) 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 $I\!R^d$ that can be "picked out" by a collection of subsets of $I\!R^d$, $\mathcal{E}$. 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 [29] and Devroye et al. [20] 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 III, the shattering coefficients are bounded by $(m+1)^{d_{VC}}$ which simplifies the application of Proposition 2.

Theorem 1: The local averaging approaches 1 through 3 are uniformly consistent. Hence, by gathering sufficient information from the trajectory of the MDP generated by a single policy, $\bar{\mu}$, it is possible to infer the value of any other policy μ arbitrarily well using local averaging. We have 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

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

Furthermore, $\hat{\mu}_m$ converges to the optimal strategy μ^* in the sense that

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

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 IV. 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. On the other hand, a drawback of using large datasets is that also the computation time needed to determine $\hat{\mu}_m$ increases. In the next section we discuss this complexity issue and we also hint at the design of algorithms that can be executed in an online fashion.

VI. Computational Complexity and Approximations

In Section V we demonstrated that kernel-based reinforcement learning can be used in principle to approximate the optimal control in an MDP arbitrarily well by using a sufficiently large amount of training data. In this section, we investigate the computational effort of this approximation in terms of computer memory and computation time. We find that there is a tradeoff between statistical accuracy and computational efficiency that can make the use of very large data sets infeasible in practice unless approximations are used. A particularly important case is that of "online" operation where the computations for each new observation must be carried out in constant time.

First, we consider the time-complexity of kernel-based reinforcement learning. If value or policy iteration is used to solve the discrete AACOE (22) in Section IV-C, for example, the complexity is dominated by the matrix multiplication $\Phi_{m,a}\hbar_m$. This multiplication costs Mm^2 operations, since $\Phi_{m,a}$ is of size $m \times m$ and $\Phi_{m,a}\hbar_m$ must be evaluated for $a = 1, \ldots, M$. If we assume that a constant number of value iteration steps must be carried out to find a solution to (20), the overall complexity is hence still $O(Mm^2)$. Second, the storage requirements of kernel-based reinforcement learning are of the complexity $O(m)$ due to the self-approximating property. Note that, even though these bounds are polynomial, they may still be problematic for large data sets. Hence it may be necessary to compute an approximate solution at the cost of sacrificing the consistency property of Section V. More specifically, we frequently want to carry out reinforcement learning "online" in the sense that the training sample and the control strategy are updated continuously over time as new data arrive. Each time step must be of complexity $O(1)$ for this purpose to prevent a slowdown of the computation. Thus a time-complexity of $O(Mm^2)$ is as bad as a complexity of $O(m)$, for example, from the perspective of online learning. Parametric approximation methods, on the other hand, achieve a constant time-complexity simply by bounding the number of parameters, e. g. in a linear model. Similar strategies are clearly also feasible for non-parametric methods. We consider two alternative approximations:

- **Discarding of "Old" Observations**

  One possibility to save computation is simply to discard observations based on one of various criteria. For example, we may eliminate the oldest observations from the data set or we may delete those observations that are least influential to the value function estimate in an appropriate sense. In either case, the quality of the controller may vary considerably over time.

- **Summarizing Observations Using "Sufficient Statistics"**

  A more sophisticated alternative is to replace groups of "similar" observations using sufficient statistics rather than to eliminate them completely.[‡] For example, we might replace a group of transitions with similar starting positions and similar endpoints using a new, "artificial" transition. Here each artificial transition should be weighted so as to reflect the number of contributing observations.

The effect of these approximations is very similar to that of a parametric model. Just as a parametric model has a fixed number of parameters and can hence only approximate a limited class of functions, we consider only a fixed number of (possibly artificial) data points in the approaches above. However, in contrast to most parametric models, the convergence property of the approximate dynamic programming algorithm remains unchanged by this simplification.

VII. Conclusions

We presented a new learning algorithm to approximate the value function and the optimal policies of an continuous-state average-cost Markov Decision Processes using

[‡]We use the term "sufficient" in a colloquial sense.

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 V, 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. For example, in Section VI we sketched an online version of kernel-based reinforcement learning that eliminates old observations and replaces them with sufficient statistics. Here the amount of training data needed depends crucially on the dimensionality of the state-space, d. In particular, *any* method for approximating the value function of an MDP from data is subject to the so-called "curse-of-dimensionality", i.e. the fact that the amount of training data and computational resources to achieve comparable accuracy increases exponentially with d. 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 by Stone [16].[§] 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

[§]In [26], [27] 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.

original state-space. For example, Tsitsiklis and Van Roy select special "features" summarizing the dynamics of an MDP [4], 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 [30]. 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.

Acknowledgments

The work of Dirk Ormoneit was supported by a grant of the Deutsche Forschungsgemeinschaft (DFG) as part of its post-doctoral program. We are particularly grateful to Sean Meyn for helpful suggestions. Any remaining mistakes are, of course, our own responsibility.

Appendix

I. A Large Deviation Result for Uniformly Ergodic Markov Chains

Let $g : I\!\!R^d \to I\!\!R$, set $Y_i \equiv g(X_i)$, and let $S_n \equiv \sum_{i=1}^m Y_i$. Furthermore, let the norm and the long-term expected value of g be defined according to $||g|| \equiv \sup\{|g(x)| : x \in S\}$ and $\alpha \equiv \int_S g(x)\pi(dx)$, respectively. The following Markovian extension of Hoeffding's inequality is proven in [31]:

Theorem 3: Suppose that $||g|| < \infty$. Then under assumption A.4 we have that

$$P_x\left(S_m - E[S_m] \geq m\varepsilon\right) \leq \exp\left(-\frac{\varphi^2(m\varepsilon - 2||g||q/\varphi)^2}{2m||g||^2q^2}\right)$$

for $m > 2||g||q/(\varphi\varepsilon)$.

This theorem is immediately applicable to the Markov chains in this work.

II. 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"

$$(\hat{\Gamma}_{m,\mu} h_\mu)(x) = h_\mu(x) + \eta_\mu - c_\mu(x) + \xi_m(x). \tag{30}$$

Using (30) it is straightforward to construct the martingale (under $K_{m,\mu}$):

$$M_k \equiv h_\mu(X_k) - \sum_{j=0}^{k-1} \left(\eta_\mu - c_\mu(X_j) + \xi_m(X_j)\right).$$

Using $E_{m,x}[M_T] = M_0$ and dividing by T we obtain the equality

$$\frac{1}{T} h_\mu(x) = \frac{1}{T} E_{m,x}\left[h_\mu(X_T) - \sum_{j=0}^{T-1} \left(\eta_\mu - c_\mu(X_j) + \xi_m(X_j)\right)\right]. \tag{31}$$

Next, we consider the limiting behavior of (31) as T goes to infinity. In particular, focus is on the perturbation term

$$E_{m,x}\left[\frac{1}{T}\sum_{j=0}^{T-1} \xi_m(X_j)\right] = E_{m,x}\left[\frac{1}{T}\sum_{j=0}^{T-1}(\hat{\Gamma}_{m,\mu} h_\mu)(X_j)\right] - E_{m,x}\left[\frac{1}{T}\sum_{j=0}^{T-1}(\Gamma_\mu h_\mu)(X_j)\right]. \tag{32}$$

Note that the first term on the right-hand side of (32) is convergent as T goes to infinity due to the finiteness of $h_\mu(x)$ and the fact that $\hat{\Gamma}_{m,\mu} h_\mu$ averages $h_\mu(x)$ at finitely many locations (see equation (8)). The second term in (32) also converges using again Poisson's equation and Assumption A.**??**, provided that $E_{m,x}[1/T \sum_{j=0}^{T-1} h_\mu(X_j)]$ converges. But this must be the case, for $E_{m,x}$ restricts the values of X_j to the finitely many elements of $\mathcal{S}$. It follows that also (32) converges and hence the artificial MDP is ξ_m-regular and the limiting value of (32) can be expressed in terms of the unique stationary distribution of $K_{m,\mu}(x, A)$ [10]. Equation (23) follows upon observing that in the limit of equation (31) the h_μ-terms vanish.

Similarly, in order to derive equation (24) we apply optional sampling with regard to the stopping time τ to M_k:

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

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

Using $E_{m,x}[M_\tau] = M_0$ and $E_{m,x}[\tau] \leq h_{m,\mu}(x) + C_h$ (see [10]) we obtain (24). ■

III. Proof of Proposition 2

We describe a proof extending results of Stone [15] and Devroye et al. [32], [20]. We begin by defining several auxiliary magnitudes and by establishing some of their properties.

First, consider a fixed ball $B \equiv \mathcal{B}_{\bar{x},\bar{u}}$ centered at $\bar{x}$ with radius $\bar{u}$ 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

$$\tilde{\eta}_\mu^B \equiv \lim_{T\to\infty} E_{x,\mu}\left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{1}(X_t \in B) c_\mu(X_t)\right] \tag{33}$$

$$\tilde{h}_\mu^B(x) \equiv E_{x,\mu}\left[\sum_{t=0}^{\tau-1}(\mathbb{1}(X_t \in B) c_\mu(X_t) - \tilde{\eta}_\mu^B)\right]. \tag{34}$$

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. We want to show that $\tilde{\eta}_\mu^B$ and $\tilde{h}_\mu^B(x)$ are uniformly bounded for all $\mu \in \mathcal{M}_\varrho$. For this purpose, let τ_B be the first hitting time of B and note that $\tilde{h}_\mu^B$ satisfies

$$\begin{aligned}
\tilde{h}_\mu^B(x) &\leq E_{x,\mu}\left[\sum_{t=0}^{\tau_B-1} \mathbb{1}(X_t \in B)(c_\mu(X_t) - \tilde{\eta}_\mu^B)\right] + \\
&\quad + \sup_{x\in B} E_{x,\mu}\left[\sum_{t=0}^{\tau-1} \mathbb{1}(X_t \in B)(c_\mu(X_t) - \tilde{\eta}_\mu^B)\right] \\
&\leq 0 + \sup_{x\in B} E_{x,\mu}\left[\sum_{t=0}^{\tau-1}(c_\mu(X_t) - \tilde{\eta}_\mu^B)\right] \\
&\leq \sup_{x\in B, a\in A} c(x,a) \sup_{x\in B,\mu\in\mathcal{M}} E_{x,\mu}[\tau].
\end{aligned}$$

Both suprema on the right-hand side of the last inequality are finite because B is compact and because there are continuous bounds for $c(x,a)$ and $E_{x,\mu}[\tau]$. In detail, we use the facts that A is finite and that $E_{x,\mu}[\tau] \leq V(x) + C_S$ (see Theorem 11.3.11 in [10]). Similarly, it is easy to derive a uniform lower bound on $\tilde{h}^B_\mu(x)$ so that $\tilde{h}^B_\mu(x)$ 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 $\tilde{h}^B_\mu(x)$ for simplicity; similarly, we let $\tilde{\eta}_\mu \equiv \tilde{\eta}^B_\mu$.

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

$$|(\Gamma_\mu \tilde{h}_\mu)(z) - (\Gamma_\mu \tilde{h}_\mu)(x)| \leq \int |\tilde{h}_\mu(y)||p_{\mu(x)}(x, y) - p_{\mu(z)}(z, y)|\lambda(dy). \tag{35}$$

Because $p_a(x, y)$ is uniformly continuous for all $a \in A$ according to Assumption A.2 and because A is finite, for all $\epsilon > 0$ we can choose a δ such that $|p_{\mu(x)}(x, y) - p_{\mu(z)}(z, y)| < \epsilon$ provided that $||x - z|| < \delta$. Because $|\tilde{h}_\mu(y)|$ is bounded by C_B we obtain that (35) is bounded by $\epsilon C_B \lambda(B)$, which is in turn smaller than ε without loss of generality.

Proof of Proposition 2:

We assume that a fixed ball $B \subset \mathbb{R}^d$ is given in equations (33) and (34) and that kernel-based reinforcement learning approximates the unknown MDP on B. That is, in 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

$$E_{x_0,\bar{\mu}}|\eta_{m,\mu} - \eta_\mu| \leq E_{x_0,\bar{\mu}}|\eta_{m,\mu} - \tilde{\eta}_\mu| + |\tilde{\eta}_\mu - \eta_\mu|. \tag{36}$$

Because of Assumption A.8 it is always possible to choose B so that the second term on the right-hand side of (36) is smaller than a given ε. The first term in (36) can be rewritten using Proposition 1 as

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

It is clear from the definition of $\pi_{m,\mu}$ that $E_{x_0,\bar{\mu}}[\pi_{m,\mu}(\cdot)]$ denotes a probability measure. Hence it is without loss of generality to assume that B is chosen so that $2C_B E_{x_0,\bar{\mu}}[\pi_{m,\mu}(B)] < \varepsilon$. With regard to the first term of the last inequality, we consider yet another decomposition:

$$
\begin{aligned}
&E_{x_0,\bar{\mu}} \left| \sup_{x\in B} (\hat{\Gamma}_{m,\mu}\tilde{h}_\mu - \Gamma_\mu \tilde{h}_\mu)(x) \right| = \\
&\qquad = E_{x_0,\bar{\mu}} \left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \left(\tilde{h}_\mu(Z_{s+1}) - (\Gamma_\mu \tilde{h}_\mu)(x) \right) \right| \\
&\qquad \leq E_{x_0,\bar{\mu}} \left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \left((\Gamma_\mu \tilde{h}_\mu)(Z_s) - (\Gamma_\mu \tilde{h}_\mu)(x) \right) \right| \qquad (37) \\
&\qquad\quad + E_{x_0,\bar{\mu}} \left| \sup_{x\in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) e_{s,\mu} \right|. \qquad (38)
\end{aligned}
$$

First, we investigate the "bias term" (37). 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

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

$$\leq 2C_B E_{x_0,\bar{\mu}} \left[\sup_{x \in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| > u) \right] + \varepsilon. \tag{39}$$

The term in the expectation operator is bounded by one so that we can split the left term in (39) 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

$$2C_B \left(\delta + P_{x_0,\bar{\mu}} \left(\sup_{x \in B} \sum_{s=0}^{m-1} k_{m,\mu}(Z_s, x) \mathbb{1}(||Z_s - x|| > u) > \delta \right) \right) \leq \varepsilon \tag{40}$$

for all $\mu \in \mathcal{M}_\varrho$ to obtain that (39) 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:

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

We used the value $\epsilon > 0$ to bound the probability in (25) 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 (40). Here $\mathcal{B}_{x,u}$ is a ball centered at x with radius u as described above. 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 m-th shattering coefficient, $v(\mathcal{E}, m)$. The last term in this derivation can be chosen so as to satisfy (40) by choosing $\delta = \varepsilon/(4C_B)$ and using Assumption (27). Hence, altogether (37) is bounded by 2ε.

With respect to the "variance term" (38), 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

$$2\delta + 2C_B P_{x_0,\bar{\mu}}\left(\sup_{x\in B}\left|\sum_{s=0}^{m-1} k_{m,\mu}(Z_s,x)e_{s,\mu}\right| > 2\delta\right) \le \varepsilon. \tag{41}$$

This is by analogy to (40). The probability term in (41) can be bounded as follows:

$$\begin{aligned}
P_{x_0,\bar{\mu}}&\left(\sup_{x\in B}\left|\sum_{s=0}^{m-1} k_{m,\mu}(Z_s,x)e_{s,\mu}\right| > 2\delta\right) \le \\
&\le P_{x_0,\bar{\mu}}\left(\sup_{x\in B}\left|\sum_{Z_s\in N_{m,\mu}(x)} k_{m,\mu}(Z_s,x)e_{s,\mu}\right| > \delta\right) + \\
&\qquad + \mathbb{1}\left(\sup_{x\in B}\sum_{Z_s\in N^c_{m,\mu}(x)} k_{m,\mu}(Z_s,x) > \frac{\delta}{2C_B}\right) \\
&\le P_{x_0,\bar{\mu}}\left(\sup_{x\in B}\left|\sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)}\right| > \frac{\delta}{\overline{C}_k}\right).
\end{aligned} \tag{42}$$

We used Assumption (25) to derive the last inequality. In detail, the indicator term equals zero by (26) for a sufficiently large m. In our next step, we apply the law of iterated expectations in order to condition on the samples in $\mathcal{S}$.

$$\begin{aligned}
P_{x_0,\bar{\mu}}&\left(\sup_{x\in B}\left|\sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)}\right| > \frac{\delta}{\overline{C}_k}\right) \\
&= E_{x_0,\bar{\mu}}\left[P\left(\sup_{x\in B}\left|\sum_{Z_s\in N_{m,\mu}(x)} \frac{e_{s,\mu}}{\#N_{m,\mu}(x)}\right| > \frac{\delta}{\overline{C}_k}\,\middle|\, Z_0,\ldots,Z_m\right)\right] \\
&\le E_{x_0,\bar{\mu}}\left[P\left(\bigcup_{a\in A, D\in\mathcal{D}_{m,a}}\left\{\left|\sum_{Z_s\in D} \frac{e_{s,a}}{\#D}\right| > \frac{\delta}{\overline{C}_k}\right\}\,\middle|\, Z_0,\ldots,Z_m\right)\right]
\end{aligned}$$

$$\leq Mv(\mathcal{E},m)E_{x_0,\bar{\mu}}\left[\sup_{a\in A, D\in\mathcal{D}_{m,a}} P\left(\left|\sum_{Z_s\in D}\frac{e_{s,a}}{\#D}\right| > \frac{\delta}{\overline{C}_k}\middle| Z_0,\ldots,Z_m\right)\right]$$
$$\leq Mv(\mathcal{E},m)E_{x_0,\bar{\mu}}\left[\sup_{a\in A, x\in B}\exp\left(-\frac{\varphi^2(\#N_{m,a}(x)\delta/\overline{C}_k - 4C_B/\varphi)^2}{8\#N_{m,a}(x)C_B^2}\right)\right]$$

for sufficiently large m. Here we applied Theorem 3 of Appendix A using the definition $g(Z_s) \equiv |e_{s,a}|$; 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 (42) by 2ϵ using Assumption (28), and we choose $\delta = \varepsilon/4$ and $\epsilon = \varepsilon/(8C_B)$ in order to satisfy condition (41).

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

■

IV. 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 II-D, as m goes to infinity. More specifically, we establish the conditions

$$m_a = \Omega_P(m) \tag{43}$$
$$l/m_a \overset{m\to\infty}{\longrightarrow} 0. \qquad \text{a.s.} \tag{44}$$

For this purpose note that, according to Sections II and IV-B, the random draw of the action a_s at time s occurs independently from the previous history using the probability distribution $\bar{\mu}(x_s) \in \mathcal{M}_\varrho$. Hence each action $a \in A$ is chosen with a probability of at least ϱ at each time step during the simulation, and the measure $\bar{\mu}(x_s) \in \mathcal{M}_{\bar{\varrho}}$ can be "split" as in Section II-C into a random draw from a uniform distribution over A with probability $\bar{\varrho}$ and from a distribution proportional to $\bar{\mu}(x_s) - \bar{\varrho}$ with probability $1 - \bar{\varrho}$. Then the number of draws of action a, m_a, in a simulation of length m is larger than the number of draws that results from the uniform draws, $\bar{m}_a$. On the other hand, we know that $\bar{m}_a/m$ converges to $\bar{\varrho}/M$ almost surely using the strong law of large numbers because these draws are independent. Hence the probability of the event $m_a/m < \bar{\varrho}/M - \epsilon$ converges to zero for arbitrary $\epsilon > 0$ which gives (43). This condition ensures that m_a grows at least as quickly as $m\underline{\varrho}$ asymptotically. In the case of an adaptive neighborhood, where $l = \#N_{m,a}(x)$ is a fixed function of m_a, (43) and condition (14) in Assumption A.5 furthermore imply the result (44).

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 $\underline{C}_p\lambda(D) \leq P_\mu(x,D) \leq \overline{C}_p\lambda(D)$ for all $D \subseteq B$, $\mu \in \mathcal{M}_\varrho$. This condition is furthermore inherited by the invariant measure, π_μ:

$$\underline{C}_p\lambda(D) \leq \pi_\mu(D) = \int \pi_\mu(dy)P_\mu(y,D) \leq \overline{C}_p\lambda(D). \tag{45}$$

Proof of Theorem 1:

We verify the conditions (25) through (28) of Proposition 2. For approaches 1 and 2, condition (25) 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 (26) is obvious given Assumption A.6.

With regard to the condition (27), we distinguish two cases depending on the event

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

Conditioning on E we obtain that

$$\begin{aligned}&\sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\#_a\mathcal{B}_{x,u/2} < C_1\#N_{m,a}(x)\right)\\ &\qquad \leq \sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\#_a\mathcal{B}_{x,u/2} < C_1\#N_{m,a}(x)\middle| E\right) + P_{x_0,\bar{\mu}}\left(E^c\right)\end{aligned}$$

$$\leq \sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\frac{\#_a\mathcal{B}_{x,u/2}}{m_a} - \pi_{\bar{\mu}}(\mathcal{B}_{x,u/2}) < -\frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{2}\right) +$$
$$+ \sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\frac{C_1\#N_{m,a}(x)}{m_a} > \frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{2}\right). \quad (46)$$

We analyze the first of these two terms by using the large deviation result of Theorem 3 in Appendix I.

$$\sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\frac{\#_a\mathcal{B}_{x,u/2}}{m_a} - \pi_{\bar{\mu}}(\mathcal{B}_{x,u/2}) < -\frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{2}\right)$$
$$= \sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\frac{\#_a\mathcal{B}_{x,u/2}}{m_a} - \pi_{\bar{\mu}}(\mathcal{B}_{x,u/2}) < -\frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{2}\right)$$
$$\leq \frac{1}{2}\sup_{a\in A, x\in B}\left\{\exp\left(-\frac{\varphi^2(m_a\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})/2 - 2/\varphi)^2}{2m_a}\right)\right\}. \quad (47)$$

Note that we applied Theorem 3 to the subchain associated with the set $\mathcal{S}_a$. Because u is fixed and $\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})$ is uniformly bounded for all $x \in B$ by (45), the expression (47) converges to zero as m_a goes to infinity using (43).

In order to deal with the second term in (46), 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 (44) ensures that E is eventually true for sufficiently large m_a, so that the second term in (46) equals zero. For fixed neighborhoods, we reformulate the second term in (46) using the neighboring region $U_{m,a}(x)$ instead of $N_{m,a}(x)$. As above, we consider the limiting value as m goes to infinity, applying Theorem 3:

$$\sup_{a\in A, x\in B} P_{x_0,\bar{\mu}}\left(\frac{C_1\#_aU_{m,a}(x)}{m_a} > \frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{2}\right)$$
$$\leq \sup_{a\in A, x\in B}\left\{\lim_{m\to\infty} P_{x_0,\bar{\mu}}\left(\frac{C_1\#_aU_{m,a}(x)}{m_a} - \pi_{\bar{\mu}}(U_{m,a}(x)) > \frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{4}\right)\right\}$$
$$\leq \frac{1}{2}\sup_{a\in A, x\in B}\left\{\exp\left(-\frac{\varphi^2(m_a\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})/4 - 2C_1/\varphi)^2}{2m_aC_1^2}\right)\right\}. \quad (48)$$

We used the fact that $\pi_{\bar{\mu}}(U_{m,a}(x)) < \frac{\pi_{\bar{\mu}}(\mathcal{B}_{x,u/2})}{4}$ for large m by (45) and (11). As above, the expression (48) and hence also (46) converge to zero by (43).

We deal with condition (28) by analogy to (27). In detail, we use a decomposition of the exponential term in (28) based on the scalar w:

$$e^{-C_2 \# N_{m,a}(x)} \leq e^{-C_2 w} \mathbb{1}(\# N_{m,a}(x) \geq w) + \mathbb{1}(\# N_{m,a}(x) < w).$$

Using this decomposition, the left-hand side of (28) can be bounded by the term

$$v(\mathcal{E}, m) \left(\sup_{a \in A, x \in B} e^{-C_2 w} + \sup_{a \in A, x \in B} P_{x_0, \bar{\mu}}(\# N_{m,a}(x) < w) \right). \tag{49}$$

As in the case of condition (27), we choose $w = l$ for adaptive neighborhoods so that $P_{x_0,\bar{\mu}}(\# N_{m,a}(x) < w)$ equals zero for sufficiently large m using (44). For fixed neighborhoods, we choose $w = m_a \pi_{\bar{\mu}}(U_{m,a}(x)) - \zeta$. Then we obtain by analogy to (47) that

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

Note that in order for (49) 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 $\zeta = m_a \pi_{\bar{\mu}}(U_{m,a}(x))^{\sqrt{2}}$. In this case we have $w = m_a \pi_{\bar{\mu}}(U_{m,a}(x)) - m_a \pi_{\bar{\mu}}(U_{m,a}(x))^{\sqrt{2}}$ which goes to infinity provided condition (11) and $m_a \pi_{\bar{\mu}}(U_{m,a}(x)) \to \infty$ as desired. On the other hand, we have $\zeta^2/m_a = m_a \pi_{\bar{\mu}}(U_{m,a}(x))^2$. We mentioned above that, for the averaging approaches 1, 2, and 3 proposed in Section III, the shattering coefficient $v(\mathcal{E}, m)$ is bounded by $(m+1)^{d_{VC}}$. Therefore, we obtain the following sufficient condition for the convergence of (49):

$$E_{x_0,\bar{\mu}}[m_a \pi_{\bar{\mu}}(U_{m,a}(x))^2] - \log(m+1) \to \infty. \tag{50}$$

Using the properties (43) and (45), assumption (12) is sufficient to guarantee that (50) holds. Altogether, (25) through (28) hold for approaches 1, 2, and 3 from which we conclude that these approaches are consistent by Proposition 2.

■

V. 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 in the proofs above.

First, consider the case where $\hat{\eta}_m < \eta^*$. 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 $E_{x_0,\bar{\mu}}|\hat{\eta}_m - \eta_{\hat{\mu}_m}| < \varepsilon$ for an arbitrary $\varepsilon > 0$ and for all sufficiently large m. On the other hand, because η^* attains the minimum costs, we have $\eta_{\hat{\mu}_m} \geq \eta^*$ and consequently $|\hat{\eta}_m - \eta^*| \leq |\hat{\eta}_m - \eta_{\hat{\mu}_m}|$. Because the expectation of the last term is bounded by ε, so is $E_{x_0,\bar{\mu}}|\hat{\eta}_m - \eta^*|$.

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 can be made arbitrarily small in absolute value by choosing a correspondingly small ϱ, i.e. by using a sufficiently large m. Because $\hat{\eta}_m - \eta^* > 0$ and because $|\eta^*_\varrho - \eta^*|$ is arbitrarily small, it is without loss of generality to assume also that $\hat{\eta}_m > \eta^*_\varrho$. Next, recall from Section IV that $\hat{\eta}_m$ achieves the minimum of $\eta_{m,\mu}$ and that $\hat{h}_m$ achieves the pointwise minimum of $h_{m,\mu}$ within $\mathcal{M}_\varrho$. 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.

■

References

[1] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, pp. 9–44, 1988.

[2] J. N. Tsitsiklis and B. Van Roy, "Average cost temporal-difference learning," *Automatica*, vol. 35, no. 11, pp. 1799–1808, 1999.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.

[4] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large-scale dynamic programming," *Machine Learning*, vol. 22, pp. 59–94, 1996.

[5] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.

[6] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Machine Learning*, 2001.

[7] 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 Journal on Control and Optimization*, vol. 31, no. 2, pp. 282–344, 1993.

[8] S. P. Meyn, "The policy iteration algorithm for average reward Markov decision processes with general state space," *IEEE Trans. on Automatic Control*, vol. 42, no. 10, pp. 1382–1393, 1997.

[9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, 1994.

[10] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Springer-Verlag, 1993.

[11] E. Nummelin, "A splitting technique for Harris recurrent chains," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 43, pp. 309–318, 1978.

[12] K. B. Athreya and P. Ney, "Limit theorems for semi-Markov processes," *Bulletin of the Australian Mathematical Society*, vol. 19, pp. 283–294, 1978.

[13] E. A. Nadaraya, "On estimating regression," *Theor. Prob. Appl.*, vol. 9, pp. 141–142, 1964.

[14] G. S. Watson, "Smooth regression analysis," *Sankhyã Series A*, vol. 26, pp. 359–372, 1964.

[15] C. J. Stone, "Consistent nonparametric regression," *Annals of Statistics*, vol. 5, no. 4, pp. 595–645, 1977.

[16] C. J. Stone, "Optimal global rates of convergence for nonparametric regression," *Annals of Statistics*, vol. 10, no. 4, pp. 1040–1053, 1982.

[17] L. Devroye and T. J. Wagner, "The strong uniform consistency of kernel density estimates," *Multivariate Analysis*, vol. 5, pp. 59–77, 1980.

[18] L. Devroye, L. Györfi, A. Krzyzak, and G. Lugosi, "On the strong unversal consistency of nearest neighbor regression function estimates," *Annals of Statistics*, vol. 22, no. 3, pp. 1371–1385, 1994.

[19] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applic.*, Chapman & Hall, 1996.

[20] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, 1996.

[21] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, 1992.

[22] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Capital City Press, 1992.

[23] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1983.

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

[25] D. Pucci de Farias and B. Van Roy, "On the existence of fixed points for approximate value iteration and temporal difference learning," Tech. Rep., Department of Engineering-Economic Systems and Operation Research, Stanford University, 1999.

[26] J. Rust, "Using randomization to break the curse of dimensionality," *Econometrica*, vol. 65, no. 3, pp. 487–516, 1997.

[27] J. Rust, "A comparison of policy iteration methods for solving continuous-state, infinite-horizon markovian decision problems using random, quasi-random, and deterministic discretizations," Tech. Rep., Department of Economics, Yale University, 1997.

[28] X-R. Cao, "Perturbation realization, potentials, and sensitivity analysis of Markov processes," *IEEE Transactions on Automatic Control*, vol. 42, no. 10, pp. 1382–1393, 1997.

[29] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[30] 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. 2000, pp. 540–546, The MIT Press.

[31] P. W. Glynn and D. Ormoneit, "Hoeffding's inequality for uniformly ergodic Markov chains," Tech. Rep., Department of Management Science and Engineering, Stanford University, 2001.

[32] L. Devroye, "The uniform convergence of nearest neighbor regression function estimators and their application in optimization," *IEEE Transactions on Information Theory*, vol. 24, no. 2, pp. 142–151, 1978.