Machine Learning manuscript No.
(will be inserted by the editor)

Model Selection in Reinforcement Learning

Amir-massoud Farahmand[1], **Csaba Szepesvári**[1]

Department of Computing Science
University of Alberta
Edmonton, Canada, T6G 2E8
e-mail: {amirf,szepesva}@ualberta.ca

The date of receipt and acceptance will be inserted by the editor

Abstract We consider the problem of model selection in the batch (offline, non-interactive) reinforcement learning setting when the goal is to find an action-value function with the smallest Bellman error among a countable set of candidate functions. We propose a complexity regularization-based model selection algorithm, BErMin, and prove that it enjoys an oracle-like property: the estimator's error differs from that of an oracle, who selects the candidate with the minimum Bellman error, by only a constant factor and a small remainder term that vanishes at a parametric rate as the number of samples increases. As an application, we consider a problem when the true action-value function belongs to an unknown member of a nested sequence of function spaces. We show that under some additional technical conditions BErMin leads to a procedure whose rate of convergence, up to a constant factor, matches that of an oracle who knows which of the nested function spaces the true action-value function belongs to, i.e., the procedure achieves *adaptivity*.

Key words Reinforcement learning, model selection, complexity regularization, adaptivity, offline learning, off-policy learning, finite-sample bounds

1 Introduction

Most reinforcement learning algorithms rely on the use of some function approximation method. In general, their performance will be largely influenced by what function approximation method is being used and how it is configured. Current practice is that the user of the algorithm decides about both the method and its configuration. For example, the user may opt for using linear function approximation (cf. Chapter 8 of the book of Sutton and Barto 1998). To configure the linear function approximation method, the user must decide about the number and the nature of basis functions. As another example, the user may also decide to use a neural network-based function approximation method (e.g., Riedmiller 2005). In this case, the user should determine the architecture of the network. If the user elects to use a nonparametric regularization-based method (e.g., Engel et al. 2005; Jung and Polani 2006; Loth et al. 2007; Farahmand et al. 2009b; Taylor and Parr 2009; Kolter and Ng 2009), the regularization coefficient and kernel (or other) parameters should be selected. From a general viewpoint, the decision of which method (linear vs. non-linear, parametric vs. non-parametric) to use is not different from that of how to tune a particular method.

Although good rules of thumb may exist of how to tune a particular method, or which method to use in a particular situation, there is no guarantee that a rule of thumb will give good results on the problem that the user wants to solve. A superior approach is to choose and configure the method based on the data. To address the issue of data-based tuning in a unified framework we assume that the user enumerates the list of possible configurations, so the problem becomes that of automating the choice between a large number of solution candidates. The advantage of this approach is that it abstracts away the details of how the solution candidates are generated and is thus generally applicable. Another advantage is that, as our results show, strong theoretical results can be proven if an appropriate selection procedure is used.

In this paper the above plan is carried out in the following context: we assume the batch learning scenario when we are given a representative dataset $\mathcal{D}_n$ of sampled transitions from a Markovian Decision Process (MDP), the goal being to find a good policy of the MDP (Szepesvári, 2010). Following previous works (e.g., Ernst et al., 2005; Riedmiller, 2005; Lagoudakis and Parr, 2003; Antos et al., 2007, 2008b; Xu et al., 2007; Antos et al., 2008a; Farahmand et al., 2009b), instead of directly working with policies, we consider the problem of finding an action-value function with a small (integrated, squared) *Bellman error*, which is supposed to facilitate the search for a good policy: When the Bellman error of an action-value function is zero (or very small) an optimal (respectively, good) policy can be obtained from the action-value function with minimal effort (some alternatives to this approach will be discussed at the end of this work). As suggested beforehand, to abstract away the details of the learning algorithms, we assume that we are given a list of action-value functions $Q_1, Q_2, \ldots$ and reduce the problem to that of selecting the function from this list with the smallest Bellman error with the help of the dataset $\mathcal{D}_n$.

In supervised learning, the classical method to find the candidate with the smallest risk amongst some functions given a finite amount of data is *complexity regularization* (Barron, 1991; Bartlett et al., 2002; Wegkamp, 2003; Lugosi and Wegkamp, 2004). A straightforward adoption of complexity regularization to our problem suggests the following procedure: First, assume that data $\mathcal{D}_n$ is independent of the candidates $Q_1, Q_2, \ldots$. Further, assume that data-based estimates $\mathrm{BE}_n(Q_k)$ of the respective Bellman errors of the candidates are available. Then choose

$$\hat{k} = \operatorname*{argmin}_{k \geq 1} \left[C_1 \, \mathrm{BE}_n(Q_k) + C_2 \, \frac{\mathrm{pen}(k)}{n} \right],$$

where $C_1 \geq 1$ and $C_2 > 0$ are appropriate constants and $\mathrm{pen}(k)$ is a suitable complexity penalty, such as $\mathrm{pen}(k) = \ln(k)$.

In the regression context, $\mathrm{BE}_n(Q_k)$ would stand for the estimate of the prediction loss of Q_k. For example, if loss is measured by the expected squared prediction error, one straightforward possibility is to estimate the loss of Q_k by averaging the squared prediction errors as measured according to $\mathcal{D}_n$. These estimates will have zero bias and a small variance that scales with $1/n$, from which it follows easily that the price paid for not having access to the true losses is negligible (as follows immediately from our umbrella result, Theorem 1). The penalties are included to prevent overfitting: Without the penalties, given P candidates, one would suffer an optimistic selection bias of order $\sqrt{\log(P)/n}$. Thus, in the limit of a very large number of models, the penalty is necessary to control the selection bias (but it also holds that for "small" P the penalties are not needed).

In a reinforcement learning context, the main issue is the construction of appropriate estimates of the Bellman error. To explain the main idea of our procedure, remember that the (integrated squared) Bellman error of Q_k is defined as $\|Q_k - T^* Q_k\|_\nu^2$, where T^* is the Bellman optimality operator underlying the unknown MDP and ν is the distribution of the state-action pairs in the sample. The main idea of our procedure is to estimate $T^* Q_k$ using a regression method and then estimate the Bellman error of Q_k based on the new estimate. In order to preclude overly optimistic estimates, the error of estimating $T^* Q_k$ is also incorporated into the estimate of the Bellman error. Plugging the so-constructed estimate into the above general method leads to our new method, BErMin, (Section 4).

Our main theoretical result (Section 5.2, Theorem 2) shows that BErMin has an oracle-like property in the sense that it selects the model with the minimum Bellman error up to a multiplicative constant and some additional terms that converge to zero. One particular application of this result is presented in Section 5.3, where we assume that as k increases the candidate generation process searches in function spaces of increasing complexity. Then, for k small, the candidate Q_k is expected to underfit (i.e., its approximation error will be large), while for k large, the candidate Q_k is expected to overfit (i.e., fit to the noise in the training data). The question is if in this case BErMin leads to a method which automatically finds the best value of k. In particular, a procedure is called *adaptive*, if the price paid for not knowing the best value of k is negligible as compared to the loss suffered by a procedure with oracle knowledge. Our result in Section 5.3 shows that the procedure built on BErMin posses this property.

In addition to the above results, we provide some auxiliary results that might be of independent interest. In particular, Theorem 1, which was mentioned previously, is an umbrella result for complexity-regularization-based model selection, and its application leads to Theorem 2. Theorem 1 is inspired by Theorem 3 of Bartlett et al. (2002): our result is an abstract reformulation of their result, the purpose being to broaden the applicability of their result beyond supervised learning. In the appendix we provide noncentral tail inequalities for Hidden Markov Processes that help us to obtain fast rates (Lemma 2 in Appendix C). Finally, in Section D we provide a procedure to estimate the excess risk of a regression problem and prove its correctness. Interesting on its own, this procedure is needed by BErMin.

In the next two sections we start with a brief review of the necessary background (Section 2), followed by a formal definition of the learning problem (Section 3).

2 Background

In the first part of this section, we provide a very brief summary of some of the concepts and definitions from the theory of Markov Decision Processes (MDP, Section 2.1) and reinforcement learning (RL, Section 2.2). However, we assume that the reader is familiar with these concepts and so the purpose of the section is merely to introduce the notation used. For further information about MDPs and reinforcement learning the reader is referred to the books by Bertsekas and Shreve (1978); Bertsekas and Tsitsiklis (1996); Sutton and Barto (1998); Szepesvári (2010). In addition to the background on MDPs, in Section 2.2 we introduce our assumptions on the learning scenario considered, as well as some less standard notations. Thus, readers are advised to pay some extra attention to the second half of this section.

2.1 Background on Markov Decision Processes

We start with the definition of Markovian Decision Processes:

Definition 1 *A* finite-action discounted MDP *is a 4-tuple $(\mathcal{X}, \mathcal{A}, P, \gamma)$, where $\mathcal{X}$ is a measurable space of states, $\mathcal{A}$ is a finite set of actions, P is a mapping with domain $\mathcal{X} \times \mathcal{A}$, and $0 \le \gamma < 1$ is a discount factor. Mapping P evaluated at $(x, a) \in \mathcal{X} \times \mathcal{A}$ gives a distribution over $\mathbb{R} \times \mathcal{X}$, which we shall denote by $P(\cdot, \cdot|x, a)$.*

An MDP together with an initial distribution P_1 of states encode the laws governing the temporal evolution of a discrete-time stochastic process controlled by an agent as follows: The controlled process starts at time $t = 1$ with random initial state $X_1 \sim P_1$ (here and in what follows $X \sim Q$ denotes that X is drawn from distribution Q). At stage t, action $A_t \in \mathcal{A}$ is selected by the agent controlling the process. In response, the pair (R_t, X_{t+1}) is drawn from $P(\cdot, \cdot|X_t, A_t)$, i.e., $(R_t, X_{t+1}) \sim P(\cdot, \cdot|X_t, A_t)$, where, R_t is the reward that the agent receives at time t and X_{t+1} is the state at time $t + 1$. The process then repeats with the agent selecting action A_{t+1}, etc. The *return* underlying the process is the discounted sum of the rewards, $\mathcal{R} = \sum_{t=1}^{\infty} \gamma^{t-1} R_t$.

In general, the agent can use all past states and rewards in deciding about its action. However, for our purposes it will suffice to consider action-selection procedures, or policies, that select an action deterministically in a time-invariant manner, solely on the basis of the last state:

Definition 2 (Deterministic, stationary Markov policy) *A measurable mapping $\pi : \mathcal{X} \to \mathcal{A}$ is called a* deterministic Markov stationary policy*, or just* policy *in short. Following a policy π in an MDP means that at each time step t it holds that $A_t = \pi(X_t)$.*

Given some policy π, the *value* of the policy in a state determines the expected return the policy achieves from that state. The function V^π mapping states to reals collects these values. If $(X_1, A_1, R_1, X_2, A_2, R_2, \ldots)$ is the process generated from π with $X_1 \sim P_1$ and the support of P_1 includes $x \in \mathcal{X}$,

$$V^\pi(x) \overset{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x\right] .$$

Similarly, the *action-value function* of policy π given a state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$ determines the expected return of the process whose first state is x, first action is a, and subsequent actions are taken from π. These values can be collected into the action-value function underlying policy π, which we shall denote by Q^π. If $(X_1, A_1, R_1, X_2, A_2, R_2, \ldots)$ is the process generated from π with $(X_1, A_1) \sim P'_1$ and the support of P'_1 includes $(x, a) \in \mathcal{X} \times \mathcal{A}$,

$$Q^\pi(x, a) \overset{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x, A_1 = a\right] .$$

It is easy to see that if the absolute value of the immediate expected reward $r(x, a) = \int r\, P(dr, dy|x, a)$ is uniformly bounded by $R_{\max}$, then the functions V^π and Q^π are bounded by $V_{\max} = Q_{\max} = R_{\max}/(1 - \gamma)$, independent of the choice of π.

For a discounted MDP, we define the *optimal value* and the *optimal action-value* functions by the respective equations

$$\begin{aligned} V^*(x) &\overset{\text{def}}{=} \sup_\pi V^\pi(x) , && x \in \mathcal{X}, \\ Q^*(x, a) &\overset{\text{def}}{=} \sup_\pi Q^\pi(x, a) , && x \in \mathcal{X}, a \in \mathcal{A} . \end{aligned}$$

We say that a policy π is *optimal* if it achieves the optimal values in every state, i.e., if $V^\pi = V^*$.

We say that a policy π is *greedy* with respect to (w.r.t.) an action-value function Q and write $\pi = \hat{\pi}(\cdot; Q)$, if $\pi(x) \in \arg\max_{a \in \mathcal{A}} Q(x, a)$ holds for all $x \in \mathcal{X}$ (if there exist multiple maximizers, some maximizer is chosen in an arbitrary deterministic manner). Greedy policies are important because a greedy policy w.r.t. Q^* is an optimal policy. Hence, knowing Q^* is sufficient for behaving optimally (cf. Proposition 4.3 of Bertsekas and Shreve 1978).[1]

For some measurable space $\mathcal{S}$, let $B(\mathcal{S})$ denote the space of bounded, measurable real function with domain $\mathcal{S}$. The so-called Bellman operators are $B(\mathcal{X}) \to B(\mathcal{X})$ (resp., $B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$) mappings, defined as follows:

Definition 3 (Bellman Operators) *Fix a policy* π. *The Bellman operators* $T^\pi : B(\mathcal{X}) \to B(\mathcal{X})$ *and* $T^\pi : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ *are defined as*

$$\begin{aligned} (T^\pi V)(x) &\overset{\text{def}}{=} r(x, \pi(x)) + \gamma \int_{\mathbb{R} \times \mathcal{X}} V(y) P(dr, dy|x, \pi(x)) , && x \in \mathcal{X} , \\ (T^\pi Q)(x, a) &\overset{\text{def}}{=} r(x, a) + \gamma \int_{\mathbb{R} \times \mathcal{X}} Q(y, \pi(y)) P(dr, dy|x, a) , && x \in \mathcal{X}, a \in \mathcal{A} . \end{aligned}$$

To avoid unnecessary clutter we use the same symbol to denote both operators. However, this should not introduce any ambiguity: Given some expression involving T^π one can always determine which operator T^π is meant by looking at the type of function T^π is applied to.

It is known that the fixed point of $T^\pi : B(\mathcal{X}) \to B(\mathcal{X})$ is the value function of π: $T^\pi V^\pi = V^\pi$ and the fixed point of $T^\pi : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ is the action-value function of π: $T^\pi Q^\pi = Q^\pi$, e.g., Proposition 4.2(b) of Bertsekas and Shreve (1978).

We will also need the so-called Bellman *optimality* operators:

[1] Measurability issues are dealt with in Section 9.5 of the same book. In the case of finitely many actions, no additional condition is needed besides the obvious measurability assumptions on the immediate reward function and the transition kernel (Bertsekas and Shreve, 1978, Corollary 9.17.1), which we will assume from now on.

Definition 4 (Bellman Optimality Operators) *The* Bellman optimality operators $T^* : B(\mathcal{X}) \to B(\mathcal{X})$*,* $T^* : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ *are defined by*

$$(T^*V)(x) \overset{\text{def}}{=} \max_a \left\{ r(x,a) + \gamma \int_{\mathbb{R}\times\mathcal{X}} V(y) P(dr, dy|x,a) \right\}, \qquad x \in \mathcal{X},$$
$$(T^*Q)(x,a) \overset{\text{def}}{=} r(x,a) + \gamma \int_{\mathbb{R}\times\mathcal{X}} \max_{a'} Q(y,a') P(dr, dy|x,a), \qquad x \in \mathcal{X}, a \in \mathcal{A}.$$

Again, we use the same symbol to denote both operators: the previous comment that no ambiguity should arise because of this still applies. The Bellman optimality operators enjoy a fixed-point property similar to that of the Bellman operators. In particular, $T^*V^* = V^*$ and $T^*Q^* = Q^*$, see e.g., Proposition 4.2(a) of Bertsekas and Shreve (1978). The Bellman optimality operator thus provides a vehicle to compute the optimal action-value function and therefore to compute an optimal policy.

2.2 Offline Learning Problem and Empirical Bellman Operators

In the learning scenario, the Bellman (optimality) operators are not accessible. In the *offline* learning scenario, all that is known about the MDP is in the form of a batch of data[2]

$$\mathcal{D}_n = \{(X_1, A_1, R_1, X'_1), \ldots, (X_n, A_n, R_n, X'_n)\}.$$

Here $(R_i, X'_i) \sim P(\cdot,\cdot|X_i, A_i)$, $A_i \sim \pi_b(\cdot|X_i)$, and $X_i \sim \nu_{\mathcal{X}}$ $(i = 1, \ldots, n)$, where $\nu_{\mathcal{X}}$ is some fixed distribution over the states and π_b is some stochastic, stationary Markov policy, the so-called behavior policy.[3] We shall denote by ν the common distribution underlying (X_i, A_i). Samples X_i and X_{i+1} may be sampled independently, or may be coupled through $X_{i+1} = X'_i$. In the latter case the data forms a single, long trajectory. In either of these two cases, we say that the data meets the *standard offline sampling assumption*. The assumption that the states $\{X_i\}$ are identically distributed and that a fixed stationary policy is used to generate the data can be relaxed, but would complicate the analysis and henceforth is not considered. Similarly, we do not consider other cases such as when the data consists of independently sampled trajectories, though the analysis would extend to such cases without much change.
The data $\mathcal{D}_n$ allows us to define the so-called empirical Bellman operators, which can be thought of as empirical approximations to the true Bellman operators:

Definition 5 (Empirical Bellman Operators) *Let* $\mathcal{D}_n$ *be a dataset as above. Define the ordered multiset* $S_n = \{(X_1, A_1), \ldots, (X_n, A_n)\}$*. For a given fixed policy* π*, the empirical Bellman operator* $\hat{T}^\pi : \mathbb{R}^{S_n} \to \mathbb{R}^n$ *is defined as*

$$(\hat{T}^\pi Q)(X_i, A_i) \overset{\text{def}}{=} R_i + \gamma Q(X'_i, \pi(X'_i)), \quad 1 \le i \le n.$$

Similarly, the empirical Bellman optimality operator $\hat{T}^* : \mathbb{R}^{S_n} \to \mathbb{R}^n$ *is defined as*

$$(\hat{T}^* Q)(X_i, A_i) \overset{\text{def}}{=} R_i + \gamma \max_{a'} Q(X'_i, a'), \quad 1 \le i \le n.$$

[2] In what follows, when $\{\cdot\}$ is used in connection to a dataset, we treat the set as an ordered multiset, where the ordering is given by the time indices of the data points. In particular, given such an ordered multiset $\mathcal{D}_n$, we can both condition on $\mathcal{D}_n$ without losing information about the order of the elements of $\mathcal{D}_n$ and write, e.g., $f : \mathcal{D}_n \to Y$, by which we mean a function specified by giving n values in Y, if the dataset $\mathcal{D}_n$ was created from n points, one value for each datapoint.

[3] Being a stochastic, stationary Markov policy π_b determines a probability distribution over $\mathcal{A}$ given any state $x \in \mathcal{X}$.

In words, the empirical Bellman operators get an n-element list S_n and return an n-dimensional real-valued vector of the single-sample estimate of the Bellman operators applied to the value function Q at the selected points.

The following proposition, which follows immediately from the definitions, shows that the empirical Bellman operators provide an unbiased estimate to the respective Bellman operators (note that $\hat{T}^\pi$ and $\hat{T}^*$ depend on the data, and hence they are random. The dependence is suppressed to simplify the notation).

Proposition 1 *For any fixed, bounded, measurable, deterministic function $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, policy π and index $1 \le i \le n$, it holds that*

$$\mathbb{E}\left[\hat{T}^\pi Q(X_i, A_i) \,\middle|\, X_i, A_i \right] = T^\pi Q(X_i, A_i)\,,$$
$$\mathbb{E}\left[\hat{T}^* Q(X_i, A_i) \,\middle|\, X_i, A_i \right] = T^* Q(X_i, A_i)\,,$$

In what follows we shall use $\|Q\|_\nu$ to denote the $L^2(\nu)$-norm of a measurable function $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$:

$$\|Q\|_\nu^2 \overset{\text{def}}{=} \int_{\mathcal{X}\times\mathcal{A}} |Q(x,a)|^2 d\nu(x,a)\,,$$

whereas its empirical counterpart will be denoted by

$$\|Q\|_n^2 \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} |Q(X_i, A_i)|^2\,.$$

Since by assumption, $(X_i, A_i) \sim \nu$, it follows that for any fixed Q, we have $\mathbb{E}\left[\|Q\|_n^2\right] = \|Q\|_\nu^2$.

3 Problem Definition

Suppose that we are given a list of action-value functions $Q_1, Q_2, \ldots, Q_P$ (with the possibility of $P > n$, or even $P = \infty$) and a dataset $\mathcal{D}_n$, the latter satisfying the standard offline sampling assumption. Our goal is to devise a procedure that selects the action-value function amongst $\{Q_1, \ldots, Q_P\}$ that has the smallest (integrated, squared) Bellman (optimality) error. Thus, the ideal procedure would return $Q_{\hat{k}}$, where

$$\hat{k} = \underset{1 \le k \le P}{\operatorname{argmin}} \|Q_k - T^* Q_k\|_\nu^2\,.$$

The idea of using the Bellman error as a criterion of optimization is not new. The algorithms implementing generalized policy iteration can be viewed as working towards minimizing it, e.g., Lagoudakis and Parr (2003); Antos et al. (2008b). There are also some basis generation/adaptation methods that use the Bellman error to guide their search, e.g., Menache et al. (2005); Keller et al. (2006); Parr et al. (2007). For a justification of minimizing the Bellman error see the discussion in the paper by Antos et al. (2008b) following their Theorem 4, or Lemma 7 of Antos et al. (2007).

Unfortunately, the Bellman error is not easy to work with. This is because neither T^* nor T^π is available in the learning setting. Moreover, even though $\hat{T}^*$ ($\hat{T}^\pi$) provides an unbiased estimate to T^* (respectively, T^π) in the sense of Proposition 1, these operators cannot be used in a simple manner to estimate the Bellman error. One might think that given any fixed function Q, the mean-squared empirical Bellman residual, $\|Q - \hat{T}^* Q\|_n^2$, is a reasonable estimate to the Bellman error. However, it follows from a standard bias-variance decomposition that

$$\mathbb{E}\left[\|Q - \hat{T}^* Q\|_n^2\right] = \|Q - T^* Q\|_\nu^2 + \mathbb{E}\left[\|\hat{T}^* Q - T^* Q\|_n^2\right] \neq \|Q - T^* Q\|_\nu^2\,,$$

which shows that $\|Q - \hat{T}^* Q\|_n^2$ is a biased estimate. In fact, from the above decomposition, we see that selecting the policies based on the mean-squared empirical Bellman residual leads to favoring

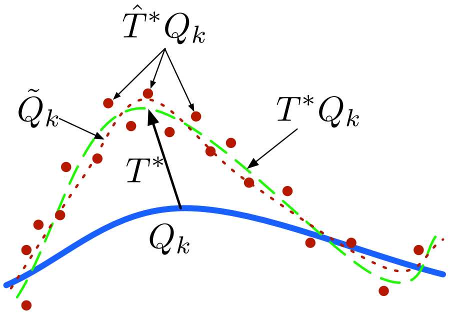

Fig. 1 When the problem is to estimate the difference between T^*Q_k (solid green line) and Q_k (bold, solid blue line) and the function T^*Q_k is unknown, one may use samples from $\hat{T}^*Q_k$ (red dots) and solve a regression problem to get $\tilde{Q}_k$ (dashed red line). This estimate can be used in place of T^*Q_k to construct an estimate of $T^*Q_k - Q_k$.

policies whose underlying variance-like term $\mathbb{E}\left[\|T^*Q - \hat{T}^*Q\|_n^2\right]$ is small, as noted previously by, e.g., Menache et al. (2005) or Antos et al. (2008b).

The main contribution of this work is a procedure, BErMin, and its analysis that shows that BErMin finds a candidate whose Bellman error is not much larger than that of the best candidate.

Remark 1 In the analysis below, for the sake of simplicity, we assume that $Q_1, \ldots, Q_P$ are fixed deterministic functions. In practice, these functions would be estimated based on some data, in which case, they would become random (data-dependent) functions. Our results, however, still continue to hold provided that the sample $\mathcal{D}_n$ used to evaluate the candidates is independent of $Q_1, \ldots, Q_P$. In particular, in this case the results can be stated and proven on the probability space obtained by conditioning on the data that generated $Q_1, \ldots, Q_P$ (the proofs would work word-by-word with no further changes). The study of the case when the same data is used to generate $Q_1, \ldots, Q_P$ is left for future work. One possible starting point for such a study could be the work by Antos et al. (2008b), who have analyzed the theoretical properties of approximate policy iteration when the same data is used in all iterations, with the main message of their result being that the correlations arising from reusing the same data are not necessarily catastrophic.

4 Model Selection Algorithm for Bellman Error Minimization (BErMin)

The purpose of this section is to introduce BErMin, a complexity regularization-based model selection algorithm for the problem of finding the Bellman error minimizer among the action-value function candidates $\{Q_k\}_{k=1}^P$. The setup is as described in Section 3. We start by describing the main idea behind the algorithm in Section 4.1, while the algorithm itself is presented in Section 4.2.

4.1 The Idea Behind the Algorithm

The basic idea behind our approach is that while the Bellman operator T^* itself is not accessible, one still may approximately learn T^*Q and use it to estimate the Bellman error. Thanks to the definition of the empirical Bellman operator $\hat{T}^*$ (Definition 5), the regression function underlying

$$\mathcal{D}_{n,k} = \left\{ \Big((X_1, A_1), (\hat{T}^*Q_k)(X_1, A_1) \Big), \ldots, \Big((X_n, A_n), (\hat{T}^*Q_k)(X_n, A_n) \Big) \right\} \tag{1}$$

is T^*Q_k (cf. Proposition 1). Thus, we can feed $\mathcal{D}_{n,k}$ to a regression procedure which, ideally, returns a “good” approximation to T^*Q_k. As the regression method one can use any of the large number of state-of-the-art techniques (cf., the books by Hastie et al. 2001; Györfi et al. 2002; Wasserman 2007; Rasmussen and Williams 2006; Bishop 2006). Although the discussion of the relative merits

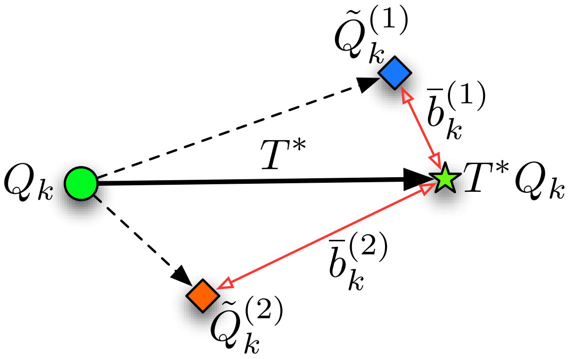

Fig. 2 Consider the problem of estimating the Bellman error $\|Q_k - T^*Q_k\|_\nu^2$. If T^*Q_k is replaced by a surrogate $\tilde{Q}_k^{(1)}$, $\|Q_k - \tilde{Q}_k^{(1)}\|_\nu^2$ gives a relatively good estimate of this quantity because $\tilde{Q}_k^{(1)}$ is close to T^*Q_k. However, when $\tilde{Q}_k^{(2)}$ replaces T^*Q_k, the resulting estimate of the Bellman error becomes poor and $\|Q_k - \tilde{Q}_k^{(2)}\|_\nu^2$ would be an *underestimate* of the true Bellman error. This might lead to the unjust selection of the candidate Q_k. One way to protect oneself against such mistakes is to take into account how well the surrogate $\tilde{Q}_k$ approximates T^*Q.

of the available methods is beyond the scope of this paper, we will shortly be more specific about the desired properties of the method.

Let the action-value function returned by the chosen regression algorithm be denoted by $\tilde{Q}_k$. If $\tilde{Q}_k$ is close to T^*Q_k, then by calculating $\|Q_k - \tilde{Q}_k\|_n^2 \approx \|Q_k - \tilde{Q}_k\|_\nu^2 \approx \|Q_k - T^*Q_k\|_\nu^2$ one can select the action-value function with the smallest Bellman error based on computing

$$\operatorname*{argmin}_{1\le k\le P} \|Q_k - \tilde{Q}_k\|_n^2 .$$

Figure 1 depicts function $\tilde{Q}_k$ and its relation to Q_k and T^*Q_k.

The problem with this procedure is that it might be overly optimistic and thus it may result in an uncontrolled error. To see why, imagine that for some index k_0 whose associated Bellman error $\|Q_{k_0} - T^*Q_{k_0}\|_\nu^2$ is “large”, the regression procedure returns an estimate such that $\|Q_{k_0} - \tilde{Q}_{k_0}\|_\nu^2 \ll \|Q_{k_0} - T^*Q_{k_0}\|_\nu^2$ (for example, because the regression procedure might be biased towards action-values close to zero, Q_{k_0} might be close to zero, while $T^*Q_{k_0}$ might be far from zero, cf. also Figure 2). As a result, the above procedure will likely select k_0, and thus might miss some other index with a lower Bellman error. To avoid this problem, we must guard the procedure against the underestimation of the Bellman error.

BErMin achieves this by correcting $\|Q_k - \tilde{Q}_k\|_\nu^2$ with $\|T^*Q_k - \tilde{Q}_k\|_\nu^2$. Since

$$\|Q_k - T^*Q_k\|_\nu^2 \le 2\left[\|Q_k - \tilde{Q}_k\|_\nu^2 + \|T^*Q_k - \tilde{Q}_k\|_\nu^2\right] ,$$

the correction indeed prevents the choice of an overly optimistic estimate (the sum in the brackets cannot be less than half of the estimated quantity). The first term of the right-hand side can be estimated by $\|Q_k - \tilde{Q}_k\|_n^2$. We further assume that we are provided with a (tight) high-probability upper bound, $\bar{b}_k$, on $\|T^*Q_k - \tilde{Q}_k\|_\nu^2$, i.e., $\|T^*Q_k - \tilde{Q}_k\|_\nu^2 \le \bar{b}_k$ with high probability. We propose to select the action-value function corresponding to the minimum of $\|Q_k - \tilde{Q}_k\|_n^2 + \bar{b}_k$. If $\bar{b}_k$ is a sufficiently tight bound, we expect that using $\bar{b}_k$ in place of $\|T^*Q_k - \tilde{Q}_k\|_\nu^2$ will not introduce any significant further bias.

We want to take care of one more detail. We would like our procedure to handle situations where the number of candidate action-value functions, P, is very large, or even potentially infinite. The latter situation arises when one transforms the algorithm into an anytime method, whose computation budget may or may not be limited, which keeps generating candidates if given more time. As a consequence of this, we add another penalty term that prevents optimistic selection bias and we will let $P = \infty$. If P is finite and small compared to n, this penalty term can safely be ignored.

Algorithm 1 BErMin($\{Q_k\}_{k=1,2,\ldots}, \mathcal{D}_{(m,n)}, \text{Regress}(\cdot), \delta, a, B, \tau$)

```
1: Split $\mathcal{D}_{(m,n)}$ into two disjoint parts: $\mathcal{D}_{(m,n)} = \mathcal{D}'_m \cup \mathcal{D}''_n$.
2: Choose $(C_k)$ such that $S = \sum_{k\geq 1} \exp(-\frac{(1-a)^2 a\, n}{16B^2\tau(1+a)} C_k) < \infty$.
3: Choose $(\delta'_k)$ such that $\sum_{k\geq 1} \delta'_k = \delta/2$.
4: for $k = 1, 2, \ldots$ do
5:     $(\tilde{Q}_k, \bar{b}_k) \leftarrow \text{Regress}(\mathcal{D}'_{m,k}, \delta'_k)$
6:     $e_k \leftarrow \frac{1}{|\mathcal{D}''_n|} \sum_{(X,A)\in\mathcal{D}''_n} (Q_k(X,A) - \tilde{Q}_k(X,A))^2$
7:     $\mathcal{R}^{\text{RL}}_k \leftarrow \frac{1}{(1-a)^2}\, e_k + \bar{b}_k$
8: end for
9: $\hat{k} \leftarrow \text{argmin}_{k\geq 1} \left[\mathcal{R}^{\text{RL}}_k + C_k\right]$
10: return $\hat{k}$
```

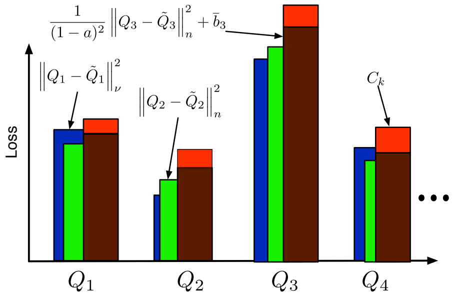

Fig. 3 A graphical illustration of the BErMin algorithm. The error $\|Q_k - \tilde{Q}_k\|_\nu$ (blue, leftmost bar) is estimated by $\left\|Q_k - \tilde{Q}_k\right\|_n^2$ (green, second bar from left), this is topped by $\bar{b}_k$, an upper bound on $\|\tilde{Q}_k - T^*Q_k\|_\nu^2$. This is followed by inflating this result by a factor of $\frac{1}{(1-a)^2}$ (brown, third bar, dark segment). Finally, the algorithm adds a complexity regularization term C_k (e.g., $C_k = \frac{32B^2\tau(1+a)}{a(1-a)^2 n}\ln(k)$) (red, third bar), and the minimum of all these values will be selected. In this figure, BErMin would select the function Q_2.

4.2 BErMin *Algorithm*

BErMin, shown as Algorithm 1, implements the ideas described in the previous section. A graphical illustration of the procedure is given on Figure 3.

The algorithm's inputs are the candidate action-value functions, the dataset $\mathcal{D}_{(m,n)}$, a regression procedure Regress, a desired error probability δ, and three constants: $0 < a < 1$, B, and τ. Here a is a tuning parameter, the constant B is the bound on all functions involved (that is Q_k, $\tilde{Q}_k$, T^*Q_k, and $\bar{b}_k$), and τ is the forgetting time of the Markov chain (cf. Definition 6 in Appendix B). The effect of these values on the quality of the solution is quantified in Theorem 2.

The algorithm initializes its data structures in three steps. In the first line the dataset is split into two disjoint parts, the first having m points, the second having n points. In Line 2, the values of the constants (C_k) are chosen such that they satisfy a Kraft-McMillan-like inequality

$$\sum_{k\geq 1} \exp\left(-\frac{(1-a)^2 a\, n}{16B^2\tau(1+a)} C_k\right) < \infty.$$

One feasible choice is $C_k = \frac{32B^2\tau(1+a)}{(1-a)^2 a n}\ln(k)$, but any other choice is possible as long as it satisfies the required condition. The choice of these values should reflect one's prior beliefs about the suitability of the candidate functions. The default choice above (which increases with k) reflects the prior belief that functions with higher indices are less suitable. Such a choice can be justified, e.g., if Q_k is expected to become more susceptible to overfitting as the value of k increases. When one has a finite number of models (i.e., $P < \infty$) and no good prior knowledge about the suitability of Q_k, one can use $C_k \equiv const$. In Line 3, we choose the confidence parameters (δ'_k) such that their sum is $\delta/2$. One possible choice is $\delta'_k = \frac{3}{\pi^2}\frac{\delta}{k^2}$ (when P is finite, one can simply use $\delta'_K = \delta_k/P$). For consistency, it might be a good idea to make δ_k and C_k behave "similarly" as a function k.

In Line 5 the regression procedure REGRESS is called with the dataset $\mathcal{D}'_{m,k}$ derived from $\mathcal{D}'_m$ using (1) (i.e., $\mathcal{D}'_{m,k}$ depends on Q_k) and δ'_k as the confidence parameter. The requirement on REGRESS is that it returns $\tilde{Q}_k$, an estimate of T^*Q_k, and $\bar{b}_k$, a high-probability upper bound on the *excess risk* $\|\tilde{Q}_k - T^*Q_k\|_\nu^2$. The upper bound on the excess risk is required to hold with probability at least $1 - \delta'_k$ (cf. Assumption 1). One possible approach to estimate the excess risk is proposed in Section D.

In Line 6, the dataset $\mathcal{D}''_n$ is used to empirically estimate $\|Q_k - \tilde{Q}_k\|_\nu^2$, i.e., the blue bars in Figure 3 are estimated by the green bars. The error of this is expected to be well controlled (and "small"). In the next line the two error estimates are combined to yield $\mathcal{R}_k^{\mathrm{RL}}$ (brown bars in Figure 3). In Line 9 this estimate is further biased upwards (red portion of bars in the graph) by the amount of C_k and then the minimizer of $\mathcal{R}_k^{\mathrm{RL}} + C_k$ is selected, where $k = 1, 2, \ldots$, giving rise to the value returned by the procedure.

Remark 2 (Computational Complexity) The complexity of BERMIN is expected to be dominated by the cost of running REGRESS. Let us assume that BERMIN selects the candidate returned amongst P candidates. If the computational complexity of REGRESS is $O(\mathbf{r}(m))$, the computational complexity of BERMIN becomes $O((n + \mathbf{r}(m))P)$. Thus, knowing the amount of time available, one could come up with an estimate of how many models can be evaluated. However, we think that a better approach is to run the algorithm in an anytime fashion until the computational budget is exhausted. Although BERMIN is not expected to be cheap, overall it might still be cheaper than an ad-hoc tuning method with a human in the loop, though admittedly, this would be hard to measure in practice.

Remark 3 (Candidate Models: An Example) An important question is what candidate functions one should feed to BERMIN and how these are found. In general, this will depend on what *a priori* information one has about the unknown MDP. Even though this is not the focus of this work, we give an example when we assume *a priori* that the optimal action-value function belongs to a Sobolev space, but the identity of the Sobolev space to which the function belongs is unknown.

First, let us define what we mean by Sobolev spaces. Assume that $\mathcal{X} = \mathbb{R}^d$ and let $k \geq \lceil \frac{d}{2} \rceil$. An order k Sobolev space over the domain $\mathcal{X} \times \mathcal{A} = \mathbb{R}^d \times \mathcal{A}$ consists of all real-valued functions whose domain is $\mathcal{X} \times \mathcal{A}$ and whose squared *Sobolev norm*, $\|f\|^2_{\mathbb{W}^k(\mathbb{R}^d \times \mathcal{A})} = \sum_{a \in \mathcal{A}} \|f(\cdot, a)\|^2_{\mathbb{W}^k(\mathbb{R}^d)}$, is finite. Here, $\|f(\cdot, a)\|^2_{\mathbb{W}^k(\mathbb{R}^d)}$ is the sum of the squared $L_2(\mathcal{X})$-norms of the mixed (weak) at most k-order partial derivatives of f. Let us denote by $\mathbb{W}^k(\mathbb{R}^d \times \mathcal{A})$ the set of these functions.[4]

For a pair $(k, J) \in \mathbb{N} \times \mathbb{R}_+$, define

$$\mathcal{F}(k, J) = \left\{ f \in \mathbb{W}^k(\mathbb{R}^d \times \mathcal{A}) \,:\, \|f\|_{\mathbb{W}^k(\mathbb{R}^d \times \mathcal{A})} \leq J \right\}.$$

Note that $\cup_{k \in \mathbb{N}, J \in \mathbb{R}_+} \mathcal{F}(k, J)$ is a huge space. For regression problems, it is known that the minimax optimal rate of estimating functions belonging to $\mathcal{F}(k, J)$ is $O(J^{2d/(2k+d)} m^{-2k/(2k+d)})$ (Györfi et al., 2002). Here, m is the number of samples used in the learning procedure and although we use the same letter to denote the number of samples as in $\mathcal{D}'_m$, this should be considered as a coincidence.

Assume now that the true action-value function belongs to $\mathcal{F}(k^*, J^*)$ for some unknown $(k^*, J^*) \in \mathbb{N} \times \mathbb{R}_+$. Define the set of candidate function spaces as $(\mathcal{F}(k, J))_{(k,J) \in \mathcal{P}_m}$, where

$$\mathcal{P}_m = \left\{ (k, J) \in \mathbb{N} \times \mathbb{N} \,:\, \lceil \tfrac{d}{2} \rceil \leq k \leq m,\ J \in \{2^0, 2^1 \ldots, 2^{\lceil \log_2 m \rceil}\} \right\}.$$

This set defines a grid on both the smoothness order k and the size of the smoothness term J. As we see shortly, the resolution of this grid is set such that $\mathcal{F}(k^*, J^*)$ is contained within a member of $(\mathcal{F}(k, J))_{(k,J) \in \mathcal{P}_m}$ that is not much larger than $\mathcal{F}(k^*, J^*)$ itself.

[4] A more precise notation would be $\mathbb{W}^{k,2}(\mathbb{R}^d \times \mathcal{A})$ because of the use of the L_2-norm in the definition of the Sobolev norms.

Suppose that we have a learning algorithm A that can be configured to seek the estimate of the action-value function in $\mathcal{F}(k, J)$ and has the convergence rate of $O(J^{2d/(2k+d)} m^{-2k/(2k+d)})$, provided that the true optimal action-value function indeed belongs to $\mathcal{F}(k, J)$ (for some results in this direction see, e.g., Farahmand et al. 2009b,a). Construct $Q_{(k,J)} = \mathsf{A}(\mathcal{D}_m, \mathcal{F}(k, J))$ for all $(k, J) \in \mathcal{P}_m$. Note that for m large enough there is a pair (k', J') in $\mathcal{P}_m$, close to (k^*, J^*), such that $\mathcal{F}(k^*, J^*)$ is contained within $\mathcal{F}(k', J')$. In particular if $m \geq \max\{k^*, J^*\}$, then there exists $(k', J') \in \mathcal{P}_m$ such that $k' = k^*$, $J' \leq 2J^*$, and $\mathcal{F}(k^*, J^*) \subset \mathcal{F}(k', J')$. The convergence rate of the estimator based on $(k', J')[= (k^*, J')]$ is $O(J'^{2d/(2k^*+d)} m^{-2k^*/(2k^*+d)})$, which is to be compared with the optimal rate, $O({J^*}^{2d/(2k^*+d)} m^{-2k^*/(2k^*+d)})$. We see that asymptotically, the rate associated with the model (k', J') is within at most a factor of 2 of the optimal rate. Thus, even when the set of models is restricted to a set with less than $m(\log_2(m) + 1)$ elements, by selecting an appropriate model amongst them, one can match the asymptotic rate of the true model, up to a constant factor. Thus, if we can prove that the model selected by BErMin is almost as good as (k', J') in terms of its Bellman error, we get that BErMin also comes within a constant factor of the Bellman error of the best model. This is the subject of Theorem 2, which will be stated in the next section.

5 Theoretical Analysis

The goal of this section is to provide a theoretical justification for the BErMin procedure. We start with a rather abstract complexity regularization-based model selection algorithm and its analysis in Section 5.1. The main result proven there (Theorem 1), which goes beyond the setting of reinforcement learning, will be the basis of our main result, Theorem 2, which is presented in Section 5.2. Theorem 2 shows that BErMin has an oracle-like behavior, in the sense that with high probability it selects the model with the minimum Bellman error up to a multiplicative constant and some additional terms that converge to zero. Finally, in Section 5.3, we introduce the concept of adaptivity and prove that the oracle-like behavior of BErMin leads to its adaptivity (Theorem 3).

5.1 A Generic Model-Selection Theorem

The theorem presented in this section concerns a generic complexity regularization-based model selection procedure. The theorem and its proof technique are similar to Theorem 3 of Bartlett et al. (2002). The main difference to this previous work is that our result is stated for an abstract setting where we are concerned with selecting the minimum amongst a set of values measured in noise, whereas Bartlett et al. (2002) developed their result in a specific supervised learning setting. Further, we make the role of non-central tail inequalities needed for the risk estimators explicit. Finally, we prove another related result, which will be useful for our later developments. Nevertheless, the main proof technique is essentially the same as used in the proof of Theorem 3 of Bartlett et al. (2002). For further similar results on complexity regularization, see Barron (1991); Lugosi and Wegkamp (2004).

Theorem 1 (Key Technical Model Selection Theorem) *Consider two sequences of random variables, $L_k, \mathcal{R}_k$, $k = 1, 2, \ldots$. Assume that there exist positive constants c_1, c_2, c_3, c_4 and $0 < a < 1$, such that for any $0 < \delta \leq 1$ and $k = 1, 2, \ldots$, the random variables L_k, $\mathcal{R}_k$ satisfy*

$$\mathbb{P}\left((1-a)\mathcal{R}_k \geq L_k - \frac{1}{c_2} \ln \frac{c_1}{\delta} \right) \geq 1 - \delta\,, \tag{2}$$

$$\mathbb{P}\left(\frac{1}{1+a}\mathcal{R}_k \leq \mathbb{E}\left[\mathcal{R}_k\right] + \frac{1}{c_4} \ln \frac{c_3}{\delta} \right) \geq 1 - \delta\,. \tag{3}$$

Let C_k $(k = 1, 2, \dots)$ be a deterministic sequence that satisfies

$$c_5 \overset{\text{def}}{=} \sum_{k \geq 1} \exp\left(-c_2(1-a)C_k\right) < \infty\,, \tag{4}$$

$$c_6 \overset{\text{def}}{=} \sum_{k \geq 1} \exp\left(-c_4 \frac{1+2a}{1+a} C_k\right) < \infty, \tag{5}$$

and define $\hat{k}$ by

$$\hat{k} \leftarrow \underset{k \geq 1}{\operatorname{argmin}} [\mathcal{R}_k + C_k]\,.$$

Then, the following hold true:
(A) For any $0 < \delta < 1$, with probability at least $1 - \delta$, it holds that

$$L_{\hat{k}} < (1-a^2) \min_{k \geq 1} \{\mathbb{E}\left[\mathcal{R}_k\right] + 2C_k\} + \frac{\ln(\frac{2c_1c_5}{\delta})}{c_2} + \frac{(1-a^2)\ln(\frac{2c_3c_6}{\delta})}{c_4}\,.$$

(B) For any $\alpha > 0$,

$$L_{\hat{k}} \leq (1-a^2) \min_{k \geq 1} \{\mathbb{E}\left[\mathcal{R}_k\right] + 2C_k\} + \alpha$$

holds with probability at least $1 - \left\{c_1c_5 \, \exp\left(-\frac{c_2\alpha}{2}\right) + c_3c_6 \, \exp\left(-\frac{c_4\alpha}{2(1-a^2)}\right)\right\}$.

In a typical application of this theorem, L_k would be the loss associated to some candidate k (from a set of at most countable candidates) and the random variable $\mathcal{R}_k$ would be a tightly concentrated, inflated estimate of L_k so that $(1-a)\mathcal{R}_k$ is still an overestimate of L_k, as required by condition (2). The theorem then yields that the loss associated with the selected candidate is not much larger than constant times the minimum of the losses biased by the "small' quantities C_k. In the appendix we show that conditions (2)-(3) are always satisfied for a slightly inflated estimate of L_k that tightly concentrates around its mean.

Proof Fix $0 < \delta_1, \delta_2 \leq 1$. We start by bounding the deviation $\Delta = L_{\hat{k}} - (1-a^2) \min_k \{\mathbb{E}\left[\mathcal{R}_k\right] + 2C_k\}$. By adding and subtracting $(1-a)\min_k(\mathcal{R}_k + C_k)$, we can decompose Δ into two terms as follows:

$$\Delta = \underbrace{\left(L_{\hat{k}} - (1-a)\min_k(\mathcal{R}_k + C_k)\right)}_{\Delta_1} + (1-a)\underbrace{\left(\min_k(\mathcal{R}_k + C_k) - (1+a)\min_k(\mathbb{E}\left[\mathcal{R}_k\right] + 2C_k)\right)}_{\Delta_2}\,.$$

To bound the first term of this sum, we use that $\min_k(\mathcal{R}_k + C_k) = \mathcal{R}_{\hat{k}} + C_{\hat{k}}$, which holds thanks to the definition $\hat{k}$. Thus, we have

$$\Delta_1 = L_{\hat{k}} - (1-a)(\mathcal{R}_{\hat{k}} + C_{\hat{k}}) \leq \max_k \{L_k - (1-a)(\mathcal{R}_k + C_k)\}\,.$$

Choose any $0 < \delta'_k \leq 1$ such that $\sum_k \delta'_k = \delta_1$. By condition (2), with probability $1 - \delta_1$, the quantity on the right-hand side of the last inequality is upper bounded by

$$\max_k \left\{\frac{1}{c_2} \ln \frac{c_1}{\delta'_k} - (1-a)C_k\right\}.$$

In particular, if we choose $\delta'_k = \delta_1/c_5 \, \exp(-c_2(1-a)C_k)$, the argument of the maximum becomes $\frac{1}{c_2} \ln \frac{c_1}{\delta'_k} - (1-a)C_k = \frac{1}{c_2} \ln \frac{c_1c_5}{\delta_1}$ and thus we get that

$$\Delta_1 \leq \frac{1}{c_2} \ln \frac{c_1c_5}{\delta_1}$$

holds with probability $1 - \delta_1$.

Now, using $\min_\theta f(\theta) - \min_\theta g(\theta) \le \max_\theta(f(\theta) - g(\theta))$, $\varDelta_2$ can be bounded by

$$\varDelta_2 \le (1+a) \max_k \left(\frac{\mathcal{R}_k}{1+a} - \mathbb{E}\left[\mathcal{R}_k\right] - \frac{1+2a}{1+a} C_k \right) .$$

By condition (3), for any $0 < \delta''_k \le 1$ such that $\sum_k \delta''_k = \delta_2$, it holds with probability $1 - \delta_2$ that the quantity on the right-hand side of the above inequality is upper bounded by

$$(1+a) \max_k \left(\frac{1}{c_4} \ln \frac{c_3}{\delta''_k} - \frac{1+2a}{1+a} C_k \right) .$$

Choosing $\delta''_k = \delta_2 / c_6 \exp(-c_4 \frac{1+2a}{1+a} C_k)$, we get that $\frac{1}{c_4} \ln \frac{c_3}{\delta''_k} - \frac{1+2a}{1+a} C_k = \frac{1}{c_4} \ln \frac{c_3 c_6}{\delta_2}$, therefore, with probability $1 - \delta_2$,

$$\varDelta_2 \le \frac{1+a}{c_4} \ln \frac{c_3 c_6}{\delta_2} .$$

Combining the inequalities obtained for $\varDelta_1$ and $\varDelta_2$, we get that with probability $1 - (\delta_1 + \delta_2)$,

$$\varDelta \le \frac{1}{c_2} \ln \frac{c_1 c_5}{\delta_1} + \frac{1-a^2}{c_4} \ln \frac{c_3 c_6}{\delta_2} . \tag{6}$$

To show Part (A), fix $0 < \delta \le 1$. Using the definition of $\varDelta$ and (6), by choosing $\delta_1 = \delta_2 = \delta/2$ we get Part (A). To prove Part (B), fix some $\alpha > 0$. Choosing $\delta_1 = c_1 c_5 \exp(-c_2 \alpha/2)$, $\delta_2 = c_3 c_6 \exp(-c_4 \alpha / (2(1-a^2)))$, from (6) we get that with probability $1 - (\delta_1 + \delta_2)$ the inequality $\varDelta \le \alpha$ holds, thus finishing the proof.

5.2 Model Selection for Reinforcement Learning and Planning

In this section we state and prove our main result which shows that BErMin has an oracle-like behavior. We prove the result under the following assumption.

Assumption 1 *Assume that the following hold:*

1. *The standard offline sampling assumption is satisfied by the data set*

$$\mathcal{D}''_n = \{(X_1, A_1, R_1, X'_1), \ldots, (X_n, A_n, R_n, X'_n)\}$$

and the time-homogeneous Markov chain $X_1, X_2, \ldots, X_n$ uniformly quickly forgets its past with a forgetting time τ (cf. Definition 6 in Appendix B).

2. *The functions Q_k, $\tilde{Q}_k$, $T^* Q_k$ $(k \ge 1)$ are bounded by a deterministic quantity $B > 0$.*
3. *The functions Q_k $(k \ge 1)$ are deterministic.*
4. *For each k and for any $0 < \delta'_k < 1$, $(\tilde{Q}_k, \bar{b}_k) = \text{Regress}(\mathcal{D}'_{m,k}, \delta'_k)$ are $\sigma(\mathcal{D}'_m)$-measurable, $\bar{b}_k \in [0, 4B^2]$ and $\|\tilde{Q}_k - T^* Q_k\|_\nu^2 \le \bar{b}_k$ holds with probability at least $1 - \delta'_k$.*
5. *For $(X_i, A_i, R_i, X'_i) \in \mathcal{D}''_n$, the distribution of (X_i, A_i) given $\mathcal{D}'_m$ is ν: $\mathbb{P}\left((X_i, A_i) \in U | \mathcal{D}'_m\right) = \nu(U)$ for any measurable set $U \subset \mathcal{X} \times \mathcal{A}$.*

A couple of remarks on these assumptions are in order.

Remark 4 The standard offline sampling assumption was discussed in Section 2.2. The additional assumption here demands that the Markov chain should “forget its past” uniformly fast. The actual definition, which we think is often satisfied, is somewhat technical and is given in the appendix. Here we note that this condition is satisfied if the Markov chain is uniformly ergodic (or, in other words, if the so-called Doeblin condition holds for the Markov chain (Meyn and Tweedie, 2009)). Note that if the chain mixes but the “mixing rate” is slow, a result similar to the one presented below would still hold, but possibly with a worse rate. On another note, although we have not made any specific distributional assumptions about $\mathcal{D}'_m$, it is expected that $\mathcal{D}'_m$ should satisfy similar assumptions to $\mathcal{D}''_n$ to make $\bar{b}_k$ small.

Remark 5 If the immediate rewards are bounded with probability one, most algorithms would return deterministically bounded value functions. If this is not known to hold for some algorithm, but a bound $r_{\max}$ on the immediate reward function is known, then boundedness can be achieved by truncating the value functions Q_k and $\tilde{Q}_k$ so that they take values in the interval $[-B, B] = [-r_{\max}/(1-\gamma), r_{\max}/(1-\gamma)]$ (i.e., instead of $Q_k(x, a)$, use $\min(\max(Q_k(x, a), -B), B)$). Since the target of learning in both cases is a function with range contained in $[-B, B]$, truncating the action-values this way introduces no loss of quality.

Remark 6 That the functions (Q_k) are deterministic is not an essential requirement, as already noted in Remark 1.

Remark 7 In Line 5 of Algorithm 1, we call $(\tilde{Q}_k, \bar{b}_k) \leftarrow \textsc{Regress}(\mathcal{D}'_{m,k}, \delta'_k)$. The condition that $\sum_{k\geq 1} \delta'_k = \delta/2$ ensures that simultaneously, for all $k \geq 1$, $\|\tilde{Q}_k - T^* Q_k\|_\nu^2 \leq \bar{b}_k(\delta'_k)$ holds with probability at least $1 - \delta/2$.

Remark 8 One approach to get the required high probability estimates $\bar{b}_k$ is described in Section D.

Remark 9 The success of BErMin will depend critically on the quality of the regression procedure, Regress, that it calls. If the value-function estimation procedure A used to calculate the candidate action-value functions is available, one appealing idea is to reuse this procedure for the purpose of computing the functions $(\tilde{Q}_k)$. This can be done when A also accepts the value of the discount factor as input γ. In this case, one could feed A with $\gamma = 0$ and the data

$$\mathcal{D}'_{m,k} = \left\{ \Big(X, A, (\hat{T}^* Q_k)(X, A), X'\Big) \,:\, (X, A, R, X') \in \mathcal{D}'_m \right\}$$

to produce $\tilde{Q}_k$, where we have replaced the immediate rewards in the data with the estimates of $T^* Q_k$.[5] This works because with $\gamma = 0$ the problem of finding the optimal value function becomes equivalent to estimating the immediate reward function based on the available sample. When producing the estimate $\tilde{Q}_k$ it would make sense to use the same tuning of A as the one used to produce Q_k. This will be further explored in Section 5.3. Nevertheless, one is not limited to this choice and, in fact, it makes perfect sense to use an adaptive regression procedure. This can be done based on Theorem 1 or in many other ways (for some recent works on adaptive regression estimation, refer to e.g., Wegkamp 2003; van der Vaart et al. 2006 or Arlot and Celisse 2009).

We are ready to present the main result of this work:

Theorem 2 (Model Selection for RL/Planning) *Let Assumption 1 hold. Consider the* BErMin *algorithm defined in Section 4 used with some* $0 < a < 1$, $0 < \delta \leq 1$, *and* $(C_k)_{k\geq 1}$ *such that*

$$S \stackrel{\text{def}}{=} \sum_{k\geq 1} \exp\left(-\frac{(1-a)^2 a\, n}{16 B^2 \tau\, (1+a)} C_k \right) < \infty \tag{7}$$

holds. Let $\hat{k}$ *be the index selected by* BErMin. *Then, with probability at least* $1 - \delta$,

$$\|Q_{\hat{k}} - T^* Q_{\hat{k}}\|_\nu^2 \leq \\ 4(1+a) \min_{k\geq 1} \left\{ \frac{2}{(1-a)^2} \|Q_k - T^* Q_k\|_\nu^2 + \frac{3}{(1-a)^2} \bar{b}_k + 2 C_k \right\} + \frac{96 B^2 \tau\, (1+a)}{(1-a)^2 a\, n} \ln\left(\frac{4S}{\delta}\right).$$

Note that $C_k = \frac{32 B^2 \tau (1+a)}{(1-a)^2 a n} \ln(k)$ satisfies $S < \infty$ (in particular, with this choice we get $S = \pi^2/6$). A detailed discussion of the result is given after its proof.

[5] Note that here and in what follows we use the notation $\hat{T}^*$ liberally to be interpreted based on the local context as the empirical Bellman operator underlying the dataset whose samples $\hat{T}^*$ interacts with in the given expression. Thus, in the above case, $(\hat{T}^* Q)(X, A)$ is meant to be computed based on $\mathcal{D}'_m$.

Proof By the triangle inequality and $(|x|+|y|)^2 \le 2(x^2+y^2)$, we get

$$\|Q_{\hat{k}} - T^* Q_{\hat{k}}\|_\nu^2 \le 2\left(\|Q_{\hat{k}} - \tilde{Q}_{\hat{k}}\|_\nu^2 + \|\tilde{Q}_{\hat{k}} - T^* Q_{\hat{k}}\|_\nu^2\right).$$

Define $L_k = \|\tilde{Q}_k - Q_k\|_\nu^2 + (1-a)\bar{b}_k$. The first term on the right-hand side of the last inequality can be upper bounded by $L_{\hat{k}}$, while, outside of an error event $\mathcal{E}_1$ of probability mass at most $\delta/2$, the second term can be upper bounded by $\bar{b}_{\hat{k}}$. Using the definition of L_k, we can further upper bound this term by $L_{\hat{k}}/(1-a)$, thus obtaining that on $\mathcal{E}_1^c$

$$\|Q_{\hat{k}} - T^* Q_{\hat{k}}\|_\nu^2 \le \frac{2(2-a)}{1-a} L_{\hat{k}} \le \frac{4}{1-a} L_{\hat{k}}\,.$$

Thus, the problem is reduced to that of bounding $L_{\hat{k}}$. For this, we will use Theorem 1. Let

$$\|\tilde{Q}_k - Q_k\|_n^2 = \frac{1}{n} \sum_{(x,a,r,x') \in \mathcal{D}_n''} (\tilde{Q}_k(x,a) - Q_k(x,a))^2.$$

Note that by our assumptions and conventions for multisets, this sum has n terms. Define

$$\mathcal{R}_k = \frac{1}{(1-a)^2} \|\tilde{Q}_k - Q_k\|_n^2 + \bar{b}_k\,.$$

With these definitions, the index $\hat{k}$ returned by BErMin can be given as

$$\hat{k} = \operatorname*{argmin}_{k \ge 1} \left[\mathcal{R}_k + C_k\right].$$

Thus, provided that $(\mathcal{R}_k)$, (L_k) satisfy (2)–(3) and (C_k) satisfies (4)–(5), we will be able to conclude from Theorem 1 a bound on $L_{\hat{k}}$ and thus also on the Bellman error of the selected action-value function. Since $\tilde{Q}_k$, $\bar{b}_k$ are themselves a function of $\mathcal{D}_m'$, we will use Theorem 1 on the probability space $\Omega_m = (\Omega, \Sigma, \mathbb{P}_m)$ with $\mathbb{P}_m(\cdot) = \mathbb{P}(\cdot|\mathcal{D}_m')$, i.e., we will apply the theorem on the probability space obtained by conditioning on $\mathcal{D}_m'$. Since a bound on a conditional probability gives a bound on the unconditioned probability, this will be sufficient to conclude a high probability bound on $L_{\hat{k}}$.

Let us consider (2). This condition requires that for some $c_1, c_2 > 0$, for any $0 < \delta' \le 1$, $\mathbb{P}_m(L_k - (1-a)\mathcal{R}_k \le \frac{1}{c_2} \ln \frac{c_1}{\delta'}) \ge 1 - \delta'$. By the definition of L_k and $\mathcal{R}_k$,

$$\begin{aligned} L_k - (1-a)\mathcal{R}_k &= \|\tilde{Q}_k - Q_k\|_\nu^2 + (1-a)\bar{b}_k - \left(\frac{1}{1-a}\|\tilde{Q}_k - Q_k\|_n^2 + (1-a)\bar{b}_k\right) \\ &= \|\tilde{Q}_k - Q_k\|_\nu^2 - \frac{1}{1-a}\|\tilde{Q}_k - Q_k\|_n^2\,. \end{aligned}$$

Our plan is to use Lemma 2 of Appendix C to provide the required bound. For this notice that $\mathbb{E}\left[\|\tilde{Q}_k - Q_k\|_n^2 | \mathcal{D}_m'\right] = \|\tilde{Q}_k - Q_k\|_\nu^2$ and that $\|\tilde{Q}_k - Q_k\|_n^2$ can be written as an average of the values taken by the function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, $(x,a) \mapsto (\tilde{Q}_k(x,a) - Q_k(x,a))^2$ over a Markov chain taking values in $\mathcal{X} \times \mathcal{A}$. By Assumption 1.1, the forgetting time of the underlying $\mathcal{X}$-valued chain is bounded by τ. It follows from the definition of forgetting times and that the actions are sampled from a fixed behavior policy that the forgetting time of the $\mathcal{X} \times \mathcal{A}$-valued chain is also bounded by τ. Further, by Assumption 1.2, the range of f is in $[0, 4B^2]$. Thus, by the first part of Lemma 2, $\mathbb{P}_m(\|\tilde{Q}_k - Q_k\|_\nu^2 - \frac{1}{1-a}\|\tilde{Q}_k - Q_k\|_n^2 \le \frac{8B^2(1+a)\tau}{(1-a)an} \ln \frac{1}{\delta'}) \ge 1 - \delta'$. Thus, condition (2) holds with $c_1 = 1$ and $c_2 = \frac{(1-a)an}{8B^2(1+a)\tau}$.

Now, let us consider (3). This condition requires that for some $c_3, c_4 > 0$, for each $0 < \delta' \le 1$, $\mathbb{P}_m(\frac{1}{1+a}\mathcal{R}_k - \mathbb{E}\left[\mathcal{R}_k | \mathcal{D}_m'\right] \le \frac{1}{c_4} \ln \frac{c_3}{\delta'}) \ge 1 - \delta'$. Again, $\mathcal{R}_k$ is an average of the function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, $(x,a) \mapsto \frac{1}{(1-a)^2}(\tilde{Q}_k(x,a) - Q_k(x,a))^2 + \bar{b}_k$ over an $\mathcal{X} \times \mathcal{A}$-valued Markov chain with forgetting time bounded by τ. The range of function f is contained in $[0, 4B^2(1+\frac{1}{(1-a)^2})]$. Therefore, the second part of Lemma 2 gives that the required inequality holds with $c_3 = 1$, $c_4 = \frac{(1-a)^2 a n}{8B^2(1+(1-a)^2)\tau}$.

It remains to check (4) and (5). A simple calculation gives that condition (7) ensures that both $c_5 = \sum_{k\ge 1} \exp(-c_2(1-a)C_k)$ and $c_6 = \sum_{k\ge 1}\exp(-c_4\frac{1+2a}{1+a}C_k)$ are finite and upper bounded by S. Therefore, by Part (A) of Theorem 1,

$$L_{\hat{k}} \le (1-a^2)\min_{k\ge 1}\left[\frac{1}{(1-a)^2}\|\tilde{Q}_k - Q_k\|_\nu^2 + \bar{b}_k + 2C_k\right] + \varDelta_1\,, \tag{8}$$

holds outside of an error event $\mathcal{E}_2$ of probability mass at most $\delta/2$, where

$$\varDelta_1 = \frac{\ln(\frac{2c_5}{\delta/2})}{c_2} + \frac{(1-a^2)\ln(\frac{2c_6}{\delta/2})}{c_4} \le \frac{8B^2\tau(1+a)(2+(1-a)^2)}{(1-a)a\,n}\ln\left(\frac{4S}{\delta}\right).$$

It remains to upper bound $\|\tilde{Q}_k - Q_k\|_\nu^2$. For this note that on $\mathcal{E}_1^c$ the inequalities $\|\tilde{Q}_k - T^*Q_k\|_\nu^2 \le \bar{b}_k$ hold simultaneously for all $k \ge 1$. Hence, on this event, $\|\tilde{Q}_k - Q_k\|_\nu^2 \le 2(\|Q_k - T^*Q_k\|_\nu^2 + \bar{b}_k)$. Thus, on $(\mathcal{E}_1 \cup \mathcal{E}_2)^c$,

$$\|Q_{\hat{k}} - T^*Q_{\hat{k}}\|_\nu^2 \le 4(1+a)\min_{k\ge 1}\left[\frac{2}{(1-a)^2}\|Q_k - T^*Q_k\|_\nu^2 + \frac{2+(1-a)^2}{(1-a)^2}\bar{b}_k + 2C_k\right] + \frac{4\varDelta_1}{1-a}$$

Bounding $2+(1-a)^2$ by 3 gives the final result.

To gain a better understanding of the bound of Theorem 2, we discuss the contribution of each of its right-hand side terms.

The term $\|Q_k - T^*Q_k\|_\nu^2$ is the true Bellman error of each candidate action-value function Q_k, and is a measure of the approximation error. This is the main quantity of interest and the ultimate goal of the minimization, which is not accessible to us. An oracle, having access to T^*Q_k, would select $\hat{k} = \mathrm{argmin}_{k\ge 1}\|Q_k - T^*Q_k\|_\nu^2$.

By definition, the term $\bar{b}_k$ is a bound on how well $\tilde{Q}_k$ approximates T^*Q_k. We need two conditions to hold true to make this term small: The regression procedure Regress should return a good estimate of T^*Q_k, while the bound returned on the excess risk by the same procedure should also be a tight bound on the excess-risk of the returned regressor. In Section D of the Appendix we show how these goals can be achieved by building on Theorem 1 in a quite general situation. To make the whole procedure competitive with an oracle, one should ensure that $\bar{b}_k$ is comparable to the size of Bellman-error $\|Q_k - T^*Q_k\|_\nu^2$. How to achieve this is further discussed in Section 5.3.

The third term of the bound is the complexity regularizer C_k and shows the price we pay to have an algorithm that works with a very large (or even infinite) number of models. As discussed earlier, the choice of C_k should reflect our prior belief about the suitability of the candidates. Note that if one has a finite number of models, then one can use $C_k = 0$. In the general case, C_k will depend on k, but it is still expected to be small compared to the other terms. The complexity regularizer has an information theoretic interpretation, which is discussed by Barron (1991); Barron et al. (2008).

The term outside the minimizer comes from the randomness of the sample $\mathcal{D}_n''$ used to estimate one component of the Bellman error. This term, just like C_k, converges to zero at a parametric rate, and it is thus expected to be small compared to the other terms. Note the tradeoff between the terms (C_k) and this last term.

Another tradeoff exists between the first two and the last two terms. This tradeoff is governed by a: as a approaches zero, the constant in front of the first two terms become smaller, but the last two terms diverge to infinity (see the specific form of C_k after the statement of the theorem). Moreover, as a approaches 1, the multipliers of all these terms blow up. As the first two terms often go to zero slower than the last one as the number of samples grows, one expects that a value of a close to zero will give the best tradeoff and in fact letting a go to zero like $a \sim n^{-\frac{1}{2}}$ might be the best choice. However, when the first two terms are fast (i.e., they converge to zero at the $O(1/n)$ rate) then one should keep a bounded away from zero to get the best asymptotic rate.

Remark 10 The result also holds true for policy evaluation, when given some policy π, the goal is to select a function Q_k that minimizes the Bellman error $\|T^\pi Q_k - Q_k\|_\nu$. In order to use BErMin for this problem, in the definition of the dataset, $\hat{T}^\pi$ should be used in place of $\hat{T}^*$. In fact, the only property of $\hat{T}^*$ that we used in the proof was the property stated in Proposition 1, which holds for both T^* and T^π.

Remark 11 If the forgetting time τ or an upper bound thereof is not known, one may use $\hat{\tau}(n) = \tau_0\, f(n)$ in the BErMin procedure for some $\tau_0 > 0$, and a positive-valued function f that diverges. Then, as soon as $\hat{\tau} > \tau$, the conclusion of Theorem 2 will hold with τ in the bound replaced by $\hat{\tau}$. In order to get the asymptotically best rate, one should choose a function f that grows slowly and a small value of τ_0. For example, when $f(n) = \ln(n)$, the asymptotic bound is increased only by a logarithmic factor. However, a slowly growing f with a small τ_0 can lead to a poor transient performance. On the other hand, if f grows faster (e.g., $f(n) = n^r$ for some $0 < r < 1$) or when τ_0 is larger, the transient performance is expected to improve at the price of a worse asymptotic performance.

5.3 Adaptivity

The purpose of this section is to show that BErMin can be made an adaptive procedure in a well-defined sense. We start with explaining what we mean by adaptivity.

5.3.1 The concept of adaptivity We consider the special case when the algorithm A used to compute Q_k, in addition to a dataset, takes as input a function space $\mathcal{F}(p_k)$, the discount factor $0 < \gamma < 1$ and the confidence parameter $0 < \delta \le 1$. The idea is that when A is run with this input, it will output an action-value function belonging to $\mathcal{F}(p_k)$. For a given k, $\mathcal{F}(p_k)$ may or may not hold the optimal action-value function. As a result, $\mathcal{F}(p_k)$ will impact the quality of Q_k returned by A in two ways: First, if $\mathcal{F}(p_k)$ is large, the limiting Bellman error of Q_k (as the number of samples converges to infinity) is expected to be smaller. Let us denote this quantity by $a_k(T^*)$ (the parameters signifying that the limiting error depends on k and on the MDP through T^*). The second effect is that if $\mathcal{F}(p_k)$ is large, the algorithm A will be more susceptible to overfitting. Overall, we expect that for any k, T^*, $0 < \delta \le 1$, $n \ge 1$, a high-probability bound of the form

$$\|Q_k - T^* Q_k\|_\nu^2 \le a_k(T^*) + c_{T^*}\, b_k(n, \ln(1/\delta)), \tag{9}$$

which holds with probability at least $1 - \frac{2}{\pi^2}\delta$, will hold for A.[6] Here, the second term bounds the error that results from using a finite number of samples. In this term, c_{T^*} is a constant that depends on T^* only (i.e., on the MDP), but is independent of $\mathcal{F}(p_k)$, n and δ. On the other hand, b_k does not depend on T^*. If in the limit of an infinite sample, $\|Q_k - T^* Q_k\|_\nu^2$ converges to $\inf_{Q\in\mathcal{F}(p_k)} \|Q - T^* Q\|_\nu$, then $a_k(T^*) = \inf_{Q\in\mathcal{F}(p_k)} \|Q - T^* Q\|_\nu^2$. Thus, in this case $a_k(T^*)$ becomes equal to the (squared) *approximation error* underlying $\mathcal{F}(p_k)$ and the second term is said to bound the *estimation error*. Typically, b_k is a polynomial of the ratio of its arguments and scales with how "large" $\mathcal{F}(p_k)$ is and it is expected that $b_k \to \infty$ as $k \to \infty$. It is assumed that (9) is a tight bound of this form (at this stage, the particular form of the above bound is unimportant). Note that being a tight bound, in general one cannot compute this bound as this would require *a priori* knowledge of quantities which, in general, are *a priori* unknown. For example, $a_k(T^*)$ is typically unknown. Thus, only an oracle could evaluate these bounds.

By (9), it follows that the inequalities

$$\|Q_k - T^* Q_k\|_\nu^2 \le a_k(T^*) + c_{T^*}\, b_k(n, \ln(k^2/\delta)) \tag{10}$$

[6] The purpose of constant $\frac{2}{\pi^2}$ is to simplify subsequent developments, but is otherwise unimportant due to the logarithmic dependence of b_k on $1/\delta$.

hold *simultaneously* for all $k \geq 1$, with probability at least $1 - \delta/3$. Thus, an oracle, having access to the bounds on the right-hand side could select the index k^* such that $\|Q_{k^*} - T^* Q_{k^*}\|_\nu^2 = \beta_n$, where

$$\beta_n \overset{\text{def}}{=} \min_{k \geq 1} \left\{ a_k(T^*) + c_{T^*}\, b_k(n, \ln(k^2/\delta)) \right\}.$$

We call a procedure *adaptive* if it only uses data set $\mathcal{D}_n$ but still matches the error of the candidate k^* up to a constant factor. Formally, if $\hat{k}$ is the index selected by a procedure then we call the procedure adaptive, if for some $C, c \geq 1$ it holds that for each MDP of interest[7], $n \geq 1$, $0 < \delta < 1/c$, we have

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq C \min_{k \geq 1} \left\{ a_k(T^*) + c_{T^*} b_k(n, \ln(k^2/\delta)) \right\} ,$$

with probability $1 - c\,\delta$.

5.3.2 The adaptivity of BErMin In this section we assume that $m = n$, i.e., the initial data has an even length which is split into two equal halves. The purpose of this section is to show that BErMin can be used as the basis of an adaptive procedure. For this, we propose to use A as the regression procedure Regress used in BErMin. To make our proposal formal, assume that A takes four parameters: the function space, the dataset, the discount factor, and the confidence parameter, and it returns both an action-value estimate and a confidence bound. We propose that BErMin should use

$$(\tilde{Q}_k, \bar{b}_{k,n}(\delta)) = \mathsf{A}\left(\mathcal{F}(p_k), \mathcal{D}'_{n,k}, 0, \tfrac{2}{\pi^2}\tfrac{\delta}{k^2}\right)$$

with

$$\mathcal{D}'_{n,k} = \mathcal{D}'_n(Q_k) = \left\{ \left(X_1, A_1, (\hat{T}^* Q_k)(X_1, A_1), X'_1\right), \ldots, \left(X_n, A_n, (\hat{T}^* Q_k)(X_n, A_n), X'_n\right) \right\} .$$

Since $\gamma = 0$, algorithm A acts as a regression procedure that works in the function space $\mathcal{F}(p_k)$ (and will in fact disregard the next states $X'_1, \ldots, X'_n$).

We make the following assumption on $\bar{b}_{k,n}$ returned by A:

Assumption 2 (Tightness of $\bar{b}_{k,n}$) *There exists some $C \geq 1$ such that for each MDP of interest, sample-size n, model index k, action-value function Q bounded by B and confidence parameter $0 < \delta < 1$, when A is fed with $\mathcal{F}(p_k)$, $\mathcal{D}'_n(Q)$, $\gamma = 0$, and δ then $\bar{b}_{k,n}(\delta)$ returned by A satisfies*

$$\bar{b}_{k,n}(\delta) \leq C \left[\inf_{Q' \in \mathcal{F}(p)} \|Q' - T^* Q\|_\nu^2 + b_k\left(n, \ln(\tfrac{2}{\pi^2 \delta})\right) \right] \tag{11}$$

with probability at least $1 - \delta$.[8]

Note that we make no assumption on how A behaves when its input is different from the above. In particular, we make no assumption about whether $\bar{b}_{k,n}(\delta)$ will be tight when A is fed with $\gamma > 0$. A crucial point about the above assumption is that it uses the same b_k functions which are used in the definition of adaptivity.

Since $\bar{b}_{k,n}(\delta)$ is an upper bound on the error of the action-value function returned by A, the above assumption implies two things about A when used as a regression procedure. First, in the limit of infinite samples the function returned should become close (up to a positive constant) to the theoretically best approximation error. In fact, many regression algorithms (such as the ones mentioned earlier) satisfy this condition (and can in fact achieve the approximation error). Second, the term bounding estimation error underlying A when used as a regression procedure, apart from a constant factor, should be the same as the corresponding term when A is used to approximate the fixed point of some non-constant operator. This is again reasonable, since regression in general is expected to be easier than fixed point estimation.

Now, we are ready to state the main result of this section:

[7] The class of MDPs can be restricted. Then the procedure is called adaptive within the chosen class.

[8] As before, the constant $\pi^2/2$ is included only to simplify some further results.

Theorem 3 *Let Assumptions 1 and 2 hold and assume that $m = n$. In addition, assume that* (i) *for each $k \geq 1$,* (9) *holds with probability at least $1 - \delta$ where $c_{T^*} \geq C^*$ for some positive constant C^* that is independent of T^*; and* (ii) *for any index $k \geq 1$, $L > 0$, we have $b_k(n, L) = \Omega(L/n)$. Then, when* BErMin *is used with* Regress $= \mathcal{A}$ *with $\gamma = 0$ and $C_k = \frac{32B^2\tau(1+a)}{(1-a)^2 an}\ln(k)$, the resulting procedure is adaptive: there exists a positive constant C'' such that for each MDP, $n \geq 1$, and $0 < \delta < 1$, the Bellman-error of the action-value function selected by* BErMin *is bounded by*

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq C''\beta_n = C'' \min_{k\geq 1}\left[a_k(T^*) + c_{T^*}\, b_k\left(n, \ln(\tfrac{k^2}{\delta})\right)\right],$$

with probability at least $1 - \frac{5}{3}\delta$.

Proof From Theorem 2, with the choice of $C_k = \frac{32B^2\tau(1+a)}{(1-a)^2 an}\ln(k)$, we have that with probability at least $1 - \delta$,

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq \min_{k\geq 1}\left[c_1\, \left\|Q_k - T^* Q_k\right\|_\nu^2 + c_2\, \bar{b}_{k,n}\left(\tfrac{2}{\pi^2}\tfrac{\delta}{k^2}\right) + c_3\,\frac{\ln(k)}{n}\right] + c_4\,\frac{\ln(1/\delta)}{n}$$

holds for some constants $c_1, c_2, c_3, c_4 > 0$ which do not depend on the MDP, δ and n. From Assumption 2, we get that the inequalities

$$\begin{aligned}\bar{b}_{k,n}\left(\tfrac{2}{\pi^2}\tfrac{\delta}{k^2}\right) &\leq C\left[\inf_{Q'\in\mathcal{F}(p)}\left\|Q' - T^* Q_k\right\|_\nu^2 + b_k(n, \ln(k^2/\delta))\right]\\ &\leq C\left[\left\|Q_k - T^* Q_k\right\|_\nu^2 + b_k(n, \ln(k^2/\delta))\right] \qquad (12)\end{aligned}$$

hold simultaneously for all $k \geq 1$ with probability at least $1 - \delta/3$. Thus, for some $c_1', c_2' > 0$,

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq \min_{k\geq 1}\left[c_1'\, \left\|Q_k - T^* Q_k\right\|_\nu^2 + c_2'\, b_k\left(n, \ln(\tfrac{k^2}{\delta})\right) + c_3\,\frac{\ln(k)}{n}\right] + c_4\,\frac{\ln(1/\delta)}{n}$$

holds with probability at least $1 - \frac{4}{3}\delta$. Now, by *(ii)*, $\frac{\ln(k)}{n} = \mathcal{O}\left(b_k\left(n, \ln\frac{k^2}{\delta}\right)\right)$ and $\frac{\ln(1/\delta)}{n} = \mathcal{O}\left(b_k\left(n, \ln\frac{k^2}{\delta}\right)\right)$. Hence, with some $C' > 0$, on the event where the previous inequality holds,

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq C' \min_{k\geq 1}\left[\left\|Q_k - T^* Q_k\right\|_\nu^2 + b_k\left(n, \ln\tfrac{k^2}{\delta}\right)\right]$$

holds, too. By (9), the inequalities

$$\left\|Q_k - T^* Q_k\right\|_\nu^2 \leq a_k(T^*) + c_{T^*}\, b_k(n, \ln(k^2/\delta))$$

hold simultaneously for all $k \geq 1$ with probability $1 - \delta/3$. Hence, with probability $1 - \frac{5}{3}\delta$, with some $C'' > 0$,

$$\left\|Q_{\hat{k}} - T^* Q_{\hat{k}}\right\|_\nu^2 \leq C'' \min_{k\geq 1}\left[a_k(T^*) + c_{T^*}\, b_k\left(n, \ln(\tfrac{k^2}{\delta})\right)\right] = C''\beta_n\,,$$

where we used that, by assumption, c_{T^*} is bounded away from zero.

6 Conclusion

In this work we suggested a principled approach for the tuning of reinforcement learning algorithms in the offline and non-interactive scenario. The problem was formulated as that of finding an action-value function with a small Bellman error among a set of candidate functions. BErMin, a complexity regularization-based algorithm, was introduced for this purpose.

Our main theoretical result, Theorem 2, is a finite-sample high-probability upper bound that shows that the Bellman error of the action-value function selected by BErMin is almost as small as that of an oracle who has access to the true Bellman errors. This result was further elaborated in Section 5.3, where we have shown that BErMin can be made adaptive in the sense that it can compete with an oracle who selects the model with the smallest error bounds (Theorem 3). As far as we know, this is the first work that considers adaptivity in a reinforcement learning scenario. The main message of our results is that just like in supervised learning, it is possible to learn almost as fast as if one had extra *a priori* information.

In this paper we focused on the goal of finding an action-value function with a small Bellman error. However, the primary goal in reinforcement learning is to find good policies. Is it possible to derive results similar to ours for this alternative problem? In what follows we consider two possible approaches.

First, still sticking to the action-value based approach, one might be tempted to consider the *projected Bellman error*, instead of the Bellman error. To recap, for some function space $\mathcal{F}^{|\mathcal{A}|}$, the projected Bellman error of $Q \in \mathcal{F}^{|\mathcal{A}|}$ is defined as $\|Q - \mathit{\Pi}_{\mathcal{F}^{|\mathcal{A}|}} T^* Q\|$, where $\mathit{\Pi}_{\mathcal{F}^{|\mathcal{A}|}}$ is the projection operator that maps its argument to the closest point on $\mathcal{F}^{|\mathcal{A}|}$ w.r.t. an appropriate norm. The projected Bellman error is typically defined for linear function spaces $\mathcal{F}^{|\mathcal{A}|}$, therefore we also restrict our discussion to such spaces. The advantage of the projected Bellman error then is that its magnitude can be readily estimated based on a sample (see, e.g., Antos et al. 2008b; Szepesvári 2010). However promising this is, unfortunately, the projected Bellman error is unsuitable for model selection purposes as it eliminates the component of the error that is orthogonal to $\mathcal{F}^{|\mathcal{A}|}$. Thus, even if one could calculate the exact values of the projected Bellman error, this knowledge would be useless for model selection purposes. This limitation of the projected Bellman error is also apparent if we note that under the so-called on-policy sampling condition and when $\mathcal{F}^{|\mathcal{A}|}$ is a nontrivial space, the projected Bellman error is always zero, independently of the choice of $\mathcal{F}^{|\mathcal{A}|}$. Therefore, the projected Bellman-error alone contains no information about the suitability of $\mathcal{F}^{|\mathcal{A}|}$.

Let us consider the next alternative, which we might call model-(or simulation-)based policy selection. Assume as before that the problem is already reduced to that of selecting the best policy from a list of policy candidates $\pi_1, \dots, \pi_P$. Let the performance be measured as the expected total discounted reward with respect to some known initial distribution ρ. For an MDP M and policy π, let this measure be $V^{\pi}(M, \rho)$.

One way to avoid using value functions is to use part of the data to build an approximate model $\hat{M} = (\mathcal{X}, \mathcal{A}, \hat{P}, \gamma)$ of the MDP of interest. Assume that for any learned model $\hat{P}$, one can efficiently generate *virtual* trajectories for the initial distribution ρ and any policy of interest π. For $1 \le i \le P$, let $V^{\pi_i}(\hat{M}, \rho)$ be the average of the returns obtained by following policy π_i in $\hat{M}$. If $\hat{P}$ is close enough to P, in an appropriate norm, and enough virtual trajectories are used, the estimates of $V^{\pi_i}(\hat{M}, \rho)$ will be close to $V^{\pi_i}(M, \rho)$ and thus it makes sense to select the policy with the maximum estimated expected return. The quality of this procedure will ultimately depend on how well $\hat{M}$ approximates M (since generating virtual trajectories is cheap), i.e., the problem of designing an effective policy selection method is reduced to that of learning a good generative model. Model learning based on sampled transitions falls into the realm of supervised learning. Hence, having an adaptive procedure for policy-selection will hinge upon if we have an adaptive model-learning procedure. Even though this idea looks more straightforward than the idea studied in this paper, it has a major weakness: it suggests learning an accurate model of the environment regardless of whether the fine-detail of the model is relevant for evaluating the performance of a policy or not. In a "large" environment many details of the environment might be hard to learn, but some of these details might be unnecessary to know about when it comes to searching for a good policy.

Future Work

Although in this paper we made some progress toward reinforcement learning algorithms that require minimum human supervision, the problem is far from being solved. In particular, the following questions require further investigation:

A) How to generate the list of candidate action-value functions $(Q_1, Q_2, \ldots)$? In what order should we run the methods available? We briefly discussed this issue in Remark 3 in an abstract setting. However, a more thorough, systematic approach would be desired and much remains to be done in this respect.

B) How can one construct data-dependent estimates of the forgetting time parameter τ? Both Meir (2000) and Modha and Masry (1998) face a similar situation; their respective procedures require the knowledge of the β-mixing coefficients of the dependent stochastic process. As far as we know, there is yet no rigorous procedure to estimate such parameters in the general case. Nevertheless, McDonald (2010) has recently proposed to use a mutual information-based estimator to upper bound the β-mixing coefficients, but the sample-efficiency of the method is yet to be shown. Meanwhile, one may use the procedure described in Remark 11 at the cost of a marginally slower than $1/n$ extra loss.

C) What is the relation between the quality of the solution of the fixed point of the Bellman optimality operator and the performance of the corresponding greedy policy? Antos et al. (2008b) and Antos et al. (2007) made some initial steps towards answering this question. However, their methods are rather crude and it seems possible to improve the bounds derived in these works.

D) We derived some data-dependent bounds on the excess-risk of a regression procedure that operates in a large function space which suited our immediate needs. However, the bound is asymptotic in nature and is potentially suboptimal. Can this bound be improved?

E) Finally, we briefly touched upon alternatives to value-function estimation methods. We have identified a model-based approach as one possible alternative. The model-based approach, however, should be tailored so that the irrelevant aspects of the world are not paid attention to while learning the model. How to do this remains another very intriguing open problem.

Appendices

In the following appendices, we provide some auxiliary technical results that are omitted from the main body of the text. We start with a noncentral tail inequality (Appendix A, Lemma 1), followed by a Bernstein-like concentration inequality for Hidden Markov Processes (Appendix B, Theorem 4). We put these two results together to obtain a noncentral tail inequality for the considered class of dependent sequences (Appendix C, Lemma 2). Finally in Appendix D, we consider the problem of deriving high-probability excess-risk bounds in a regression setting.

A Noncentral Tail Inequalities

The following result shows that if a random variable X satisfies a Bernstein-like inequality, the probability distribution of X being ε-smaller than $(1-a)\mathbb{E}[X]$ or ε-larger than $(1+a)\mathbb{E}[X]$ (for $0 < a < 1$) decays with the rate $\exp(-c\,\varepsilon)$ for some c independent of ε. This should be contrasted with the slower $\exp(-c'\,\varepsilon^2)$ concentration rate of X around its expectation $\mathbb{E}[X]$ (for ε "small").

Lemma 1 (Noncentral Tail Inequality) *Let X be a random variable whose expected value is nonnegative. Assume that for some $V > 0$ and for all $\varepsilon > 0$, X satisfies the following Bernstein-like tail inequality*

$$\mathbb{P}\left(\mathbb{E}[X] - X \geq \varepsilon\right) \leq \exp\left(-\frac{V\varepsilon^2}{\mathbb{E}[X] + \varepsilon}\right). \tag{13}$$

Then, for any $0 < a < 1$, $\varepsilon > 0$,

$$\mathbb{P}\left(\mathbb{E}\left[X\right] - \frac{1}{1-a}X \geq \varepsilon\right) \leq \exp\left(-\frac{V(1-a)a\varepsilon}{(1+a)}\right).$$

Similarly, if for some $V > 0$ *and for all* $\varepsilon > 0$ *it holds that*

$$\mathbb{P}\left(X - \mathbb{E}\left[X\right] \geq \varepsilon\right) \leq \exp\left(-\frac{V\varepsilon^2}{\mathbb{E}\left[X\right] + \varepsilon}\right) \tag{14}$$

then for all $0 < a < 1$ *and* $\varepsilon > 0$, *it also holds that*

$$\mathbb{P}\left(\frac{1}{1+a}X - \mathbb{E}\left[X\right] \geq \varepsilon\right) \leq \exp\left(-Va\varepsilon\right).$$

Proof We have

$$\begin{aligned}
\mathbb{P}\left(\mathbb{E}\left[X\right] - (1-a)^{-1}X \geq \varepsilon\right) &= \mathbb{P}\left(\mathbb{E}\left[X\right] - X \geq \varepsilon(1-a) + a\mathbb{E}\left[X\right]\right) \\
&\leq \exp\left(-\frac{V\left((1-a)\varepsilon + a\mathbb{E}\left[X\right]\right)^2}{(1+a)\mathbb{E}\left[X\right] + (1-a)\varepsilon}\right) \\
&\leq \exp\left(-\frac{V\left((1-a)\varepsilon + a\mathbb{E}\left[X\right]\right)^2}{\left((1-a)\varepsilon + a\mathbb{E}\left[X\right]\right)\left(\frac{1+a}{a}\right)}\right) \\
&= \exp\left(-\frac{Va\left((1-a)\varepsilon + a\mathbb{E}\left[X\right]\right)}{1+a}\right) \\
&\leq \exp\left(-\frac{V(1-a)a\varepsilon}{1+a}\right),
\end{aligned}$$

where we used (13) to get the first inequality, added a positive value to upper bound the denominator in the second inequality, and used the fact that $\mathbb{E}\left[X\right] \geq 0$ to derive the last inequality.

Similarly, (14) leads to

$$\begin{aligned}
\mathbb{P}\left((1+a)^{-1}X - \mathbb{E}\left[X\right] > \varepsilon\right) &= \mathbb{P}\left(X - \mathbb{E}\left[X\right] > \varepsilon(1+a) + a\mathbb{E}\left[X\right]\right) \\
&\leq \exp\left(-\frac{V\left((1+a)\varepsilon + a\mathbb{E}\left[X\right]\right)^2}{(1+a)\mathbb{E}\left[X\right] + (1+a)\varepsilon}\right) \\
&\leq \exp\left(-\frac{V\left((1+a)\varepsilon + a\mathbb{E}\left[X\right]\right)^2}{\left((1+a)\varepsilon + a\mathbb{E}\left[X\right]\right)\left(\frac{1+a}{a}\right)}\right) \\
&= \exp\left(-\frac{Va\left((1+a)\varepsilon + a\mathbb{E}\left[X\right]\right)}{1+a}\right) \\
&\leq \exp\left(-Va\varepsilon\right).
\end{aligned}$$

B Concentration Inequality for Hidden Markov Processes (HMPs)

The classical Bernstein inequality for independent and identically distributed sequences (e.g., Györfi et al. (2002, Appendix A)) can be shown to hold for the sequences of dependent random variables under various conditions. Such extensions are very useful when studying reinforcement learning algorithms when the standard assumption is that the data comes from some Markov chain. In this section we give such an extension based on Samson (2000).

Let $X_1, \ldots, X_n$ be a time-homogeneous Markov chain with transition kernel $P(\cdot|\cdot)$ taking values in some measurable space $\mathcal{X}$. We shall consider the concentration of the average of the Hidden-Markov Process

$$(X_1, f(X_1)), \ldots, (X_n, f(X_n)),$$

where $f : \mathcal{X} \to [0, B]$ is a fixed measurable function. To arrive at such an inequality, we need a characterization of how fast (X_i) forgets its past.

For $i > 0$, let $P^i(\cdot|x)$ be the i-step transition probability kernel: $P^i(A|x) = \mathbb{P}(X_{i+1} \in A \mid X_1 = x)$ (for all $A \subset \mathcal{X}$ measurable). Define the upper-triangular matrix $\Gamma_n = (\gamma_{ij}) \in \mathbb{R}^{n\times n}$ as follows:

$$\gamma_{ij}^2 = \sup_{(x,y)\in\mathcal{X}^2} \left\| P^{j-i}(\cdot|x) - P^{j-i}(\cdot|y) \right\|_{\mathrm{TV}}. \tag{15}$$

for $1 \le i < j \le n$ and let $\gamma_{ii} = 1$ $(1 \le i \le n)$.

Matrix Γ_n, and its operator norm $\|\Gamma_n\|$ w.r.t. the 2-norm, are measures of dependence for the random sequence $X_1, X_2, \ldots, X_n$. For example if the X_is are independent, $\Gamma_n = \mathbf{I}$ and $\|\Gamma_n\| = 1$. In general $\|\Gamma_n\|$, which appears in the forthcoming concentration inequalities for dependent sequences, can grow with n. Since the concentration bounds are homogeneous in $n/\|\Gamma_n\|^2$, a larger value $\|\Gamma_n\|^2$ means a smaller "effective" sample size. This motivates the following definition.

Definition 6 *We say that a time-homogeneous Markov chain* uniformly quickly forgets its past *if $\tau = \sup_{n\ge1} \|\Gamma_n\|^2 < +\infty$. Further, τ is called the* forgetting time *of the chain.*

Conditions under which a Markov chain uniformly quickly forgets its past are of major interest. The following proposition, extracted from the discussion on pages 421–422 of the paper by Samson (2000), gives such a condition.

Proposition 2 *Let μ be some nonnegative measure on $\mathcal{X}$ with nonzero mass μ_0. Let P^i be the i-step transition kernel as defined above. Assume that there exists some integer r such that for all $x \in \mathcal{X}$ and all measurable sets A,*

$$P^r(A|x) \le \mu(A). \tag{16}$$

Then,

$$\|\Gamma_n\| \le \frac{\sqrt{2}}{1 - \rho^{\frac{1}{2r}}},$$

where $\rho = 1 - \mu_0$.

Meyn and Tweedie (2009) calls homogeneous Markov chains that satisfy the majorization condition (16) *uniformly ergodic*. We note in passing that there are other cases when $\sup_{n\ge1} \|\Gamma_n\|$ is finite. Most notable, this holds when the Markov chain is contracting. The matrix Γ_n can also be defined for more general dependent processes and such that the theorem below remains valid. With such a definition, $\|\Gamma_n\|$ can be shown to be bounded for general Φ-dependent processes.

The following result is a trivial corollary of Theorem 2 of Samson (2000) (Theorem 2 is stated for empirical processes and can be considered as a generalization of Talagrand's inequality to dependent random variables):

Theorem 4 *Let f be a measurable function on $\mathcal{X}$ whose values lie in $[0, B]$, $X_1, \ldots, X_n$ be a homogeneous Markov chain taking values in $\mathcal{X}$ and let Γ_n be the matrix with elements defined by (15). Let*

$$Z = \frac{1}{n} \sum_{i=1}^{n} f(X_i).$$

Then, for every $\varepsilon \ge 0$,

$$\mathbb{P}(Z - \mathbb{E}[Z] \ge \varepsilon) \le \exp\left(-\frac{\varepsilon^2 n}{2B \|\Gamma_n\|^2 (\mathbb{E}[Z] + \varepsilon)} \right),$$

$$\mathbb{P}(\mathbb{E}[Z] - Z \ge \varepsilon) \le \exp\left(-\frac{\varepsilon^2 n}{2B \|\Gamma_n\|^2 \mathbb{E}[Z]} \right).$$

C Noncentral Tail Inequality for HMPs

By putting together the results of the last two sections we obtain the following noncentrail tail inequality for HMPs.

Lemma 2 *Let $X_1, X_2, \ldots, X_n$ be a time-homogenous Markov chain taking values in some measurable space $\mathcal{X}$, and f be a measurable function with $0 \le f \le B$. Let $Z = \frac{1}{n}\sum_{i=1}^n f(X_i)$. Let $\varGamma_n$ be the matrix with elements defined by (15). Then, for any $0 < a < 1$,*

$$\mathbb{P}\left(\mathbb{E}\left[Z\right] - \frac{1}{1-a} Z \ge \varepsilon\right) \le \exp\left(-\frac{(1-a)an\varepsilon}{2B\left\|\varGamma_n\right\|^2(1+a)}\right),$$
$$\mathbb{P}\left(\frac{1}{1+a} Z - \mathbb{E}\left[Z\right] \ge \varepsilon\right) \le \exp\left(-\frac{an\varepsilon}{2B\left\|\varGamma_n\right\|^2}\right).$$

Proof According to Theorem 4,

$$\mathbb{P}\left(Z - \mathbb{E}\left[Z\right] \ge \varepsilon\right) \le \exp\left(-\frac{\varepsilon^2 n}{2B\left\|\varGamma_n\right\|^2\left(\mathbb{E}\left[Z\right]+\varepsilon\right)}\right)$$

and

$$\mathbb{P}\left(\mathbb{E}\left[Z\right] - Z \ge \varepsilon\right) \le \exp\left(-\frac{\varepsilon^2 n}{2B\left\|\varGamma_n\right\|^2 \mathbb{E}\left[Z\right]}\right)$$
$$\le \exp\left(-\frac{\varepsilon^2 n}{2B\left\|\varGamma_n\right\|^2\left(\mathbb{E}\left[Z\right]+\varepsilon\right)}\right).$$

These inequalities have the same form as the Bernstein-like inequality in Lemma 1 with the choice of $V = \frac{n}{2B\|\varGamma_n\|^2}$, and therefore imply the result.

D Excess-Risk Estimation

Assumption 1 requires that BErMin has access to a function $\bar{b}$ such that the excess risk $\|\tilde{Q}_k - T^* Q_k\|_\nu^2$ is below $\bar{b}(\delta)$ with probability at least $1-\delta$. In this section, we provide a general approach to come up with such a function. To avoid clutter, the notation of this section is not specialized to the reinforcement learning setup. The conversion, however, is straightforward: the function f^* here is the same as $T^* Q_k$ $(k = 1, \ldots, P)$ and the estimate $\hat{f}$ is the same as $\tilde{Q}_k$ that is returned by the Regress module in Algorithm 1. The random variables $X_i \in \mathcal{X}$ should be "read as" $(X_i, A_i) \in \mathcal{X} \times \mathcal{A}$ and $Y_i = \hat{T}^* Q_k(X_i, A_i)$.

The task of estimating the excess risk is difficult because what can directly be estimated based on the sample is the loss, and the expected loss of a predictor is larger than the excess risk by the loss of the best regressor, which is an unknown quantity. In this section we attack this problem under the assumption that the best regressor belongs to a known function space $\mathcal{F}$. We target the problem of simultaneously estimating a regressor and returning a high-probability risk bound for the excess risk of the computed regressor. If $\mathcal{F}$ was a "small" function space (e.g., it had a finite pseudo-dimension) then any procedure (such as empirical risk minimization) with known bounds on its estimation error would directly give a solution: The estimation error bound would provide a bound on the excess risk. To increase generality, here we consider the case when $\mathcal{F}$ is too large for such a simple approach to succeed, but $\mathcal{F}$ can be decomposed into an infinite sequence of "small" function spaces, $\mathcal{F}_k$: $\mathcal{F} = \cup_k \mathcal{F}_k$. Under this assumption the natural approach is to perform model selection and return the estimation error of the selected model. The reason this can be successful is because model selection will ultimately select a sufficiently complex model. We develop this idea in the rest of this section.

Algorithm 2 $\textsc{Regress}(\{\mathcal{D}_n, \mathcal{D}'_n\}, \{\mathcal{F}_1, \mathcal{F}_2, \dots\}, a_n, \tau, (C_k))$

```
// Let $\{(X'_t, Y'_t)\}$ be the input-output pairs in $\mathcal{D}'_n$: $\mathcal{D}'_n = \{(X'_1, Y'_1), \dots, (X'_n, Y'_n)\}$.
for $k = 1, 2, \dots$ do
    $\hat{f}_k \leftarrow \mathsf{A}(\mathcal{D}_n, \mathcal{F}_k)$.
    $\bar{\mathcal{R}}_k = \frac{1}{(1-a_n)^2} \frac{1}{n} \sum_{i=1}^{n} (\hat{f}_k(X'_i) - Y'_i)^2$.
end for
$\hat{k} \leftarrow \mathrm{argmin}_{k \ge 1} \left[ \bar{\mathcal{R}}_k + C_k \right]$.
Choose $\beta_1, \beta_2, \dots$ such that $\beta_k \ge 0$ and $\sum_{k \ge 1} \beta_k = 2/3$.
return $\hat{f}_{\hat{k}}$ and $\mathfrak{B}_{\hat{k}}(n, \cdot\beta_{\hat{k}}, \tau)$
```

D.1 The Excess-Risk Estimation Algorithm

Let $(X_1, Y_1), \dots, (X_n, Y_n)$ be a stationary, time-homogeneous Markov chain taking values in $\mathcal{X} \times [-B, B]$ for $\mathcal{X} \subset \mathbb{R}^d$ and let the regression function f^* be defined by $f^*(x) = \mathbb{E}\left[Y_i | X_i = x\right]$. Let τ be an upper bound on the forgetting time of (X_i, Y_i) (cf. Appendix B). Denote the stationary distribution underlying (X_i) by ν. Given $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$, the goal is to provide a good estimate $\hat{f}$ of f^* and a high confidence upper bound on the excess-risk

$$\|\hat{f} - f^*\|^2 \stackrel{\text{def}}{=} \|\hat{f} - f^*\|_{2,\nu}^2.$$

We assume that we are given a sequence of nested function spaces $(\mathcal{F}_k)$ and f^* is known to belong to their union $\cup_{k \ge 1} \mathcal{F}_k$. We further assume that we are given an algorithm A, which, given $\mathcal{F}_k$, δ, and a dataset of n points, returns an estimate $\hat{f}_k$ of f^* that belongs to $\mathcal{F}_k$. We further assume that for any $k \ge 1$ there exist functions $\mathfrak{A}_k$ and $\mathfrak{B}_k$ such that for any $0 < \delta \le 1$,

$$L_k \stackrel{\text{def}}{=} \|\hat{f}_k - f^*\|^2 \le \mathfrak{A}_k(f^*) + \mathfrak{B}_k(n, \delta, \tau) \tag{17}$$

holds with probability $1 - \delta$ and that the value $\mathfrak{B}_k(n, \delta, \tau)$, which possibly depends on the data, can be computed at any arguments (n, δ, τ) and hence is available to our algorithm. No similar assumption is made about function $\mathfrak{A}_k$.

The algorithm that we propose works with the data split in half: The first half, $\mathcal{D}_n$, is used to find the candidates $\hat{f}_k$ (by calling A), while the second half is used to run the model-selection algorithm to approximately select the candidate with the smallest excess risk. Finally, the algorithm returns the function $\mathfrak{B}_k(n, \cdot\beta_k, \tau)$ for the selected value of k as the high-probability bound on the excess-risk returns. Here, $\beta_k \ge 0$, $\sum_{k \ge 1} \beta_k = 2/3$ determines the *prior* distribution of the error probability δ. The algorithm is given as Algorithm 2. For simplicity, we assume that the full dataset, $\mathcal{D}_n \cup \mathcal{D}'_n$ holds $2n$ data points.

Bounds of the type (17) are of major interest in the theory of regression estimation. The first term, which depends only on k and f^* and is independent of n and δ corresponds to the so-called *approximation error* and shows how well one can approximate f^* with elements of $\mathcal{F}_k$. The second term is a bound on the error resulting from using a finite sample, i.e., it bounds the *estimation error*. When the sample is made of a sequence of independent, identically distributed random variables, there are many results in the literature that can provide bounds of the type (17), e.g., Györfi et al. 2002; van de Geer 2000; Lugosi and Wegkamp 2004; Bartlett et al. 2005. The case of dependent sample is much less explored. However, since at the heart of most result are exponential tail inequalities and most exponential tail inequalities available for the independent case have been extended to the dependent case, one expects that with some work existing bounds can be readily extended to the dependent case (see Farahmand and Szepesvári (2011) for some recent results along this direction and a discussion of some prior work).

D.2 Theoretical Analysis of the Excess Error Estimator

The purpose of this section is to prove that under some technical conditions the regression estimate returned by Algorithm 2 satisfies an oracle-like property and the returned bound is a proper high-probability bound on the excess risk of the resulting estimator. The first part of the statement

follows easily from Theorem 1 and Lemma 2. The proof of this part is included mainly for the sake of completeness. The main novelty is the second part. The main idea underlying the proof of the second part is that for n large enough, with high probability $\hat{k}$ will be such that $\mathfrak{A}_{\hat{k}}(f^*) = 0$ and thus, by inequality (17), $\mathfrak{B}_{\hat{k}}(n, \delta\beta_{\hat{k}}, \tau)$ will bound the excess risk $L_{\hat{k}}$.
The assumptions under which we prove our result are as follows:

Assumption 3 *Assumptions on the data:*

1. $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, $\mathcal{D}'_n = \{(X'_1, Y'_1), \ldots, (X'_n, Y'_n)\}$, $X_i, X'_i \in \mathcal{X}$, $|Y_i|, |Y'_i| \leq B$ *for some* $B > 0$.
2. $\mathcal{D}_n$ *and* $\mathcal{D}'_n$ *are independent.*
3. (X'_i, Y'_i) *is a time-homogenous, stationary Markov chain and its forgetting time is upper bounded by* τ. *We denote by* ν *the stationary distribution underlying* (X'_i) *and we let* $\|\cdot\| = \|\cdot\|_\nu$.

Assumptions on $(\mathcal{F}_k)$ *and the regressor function* f^**:*

1. *The function spaces* $\mathcal{F}_1, \mathcal{F}_2, \ldots$ *hold measurable, real-valued functions with domain* $\mathcal{X}$ *bounded by* $B > 0$.
2. *The function* $f^*(x) = \mathbb{E}\left[Y'_t | X'_t = x\right]$ *belongs to* $\cup_{k\geq 1}\mathcal{F}_k$.

Assumptions on algorithm A and functions $\mathfrak{A}_k$, $\mathfrak{B}_k$*:*

1. *For any* $n \geq 1$, $k \geq 1$, *A returns a* $\sigma(\mathcal{D}_n)$*-measurable function* $\hat{f}_k$ *that belongs to* $\mathcal{F}_k$ *and the error bound (17) holds for this function with probability* $1 - \delta$.
2. *The functions* $\mathfrak{A}_k$ *are such that for some* $C > 1$, $\mathfrak{A}_k(f^*) \leq C \inf_{f\in\mathcal{F}_k} \|f - f^*\|^2$ *holds for all* $k \geq 1$ *and* $\mathfrak{A}_k(\cdot) \geq \mathfrak{A}_{k+1}(\cdot)$ *holds for any* $k \geq 1$.
3. *The function* $\mathfrak{B}_k(n, \delta, \tau) \xrightarrow{n\to\infty} 0$ *is a decreasing function of* n *and an increasing function of* τ.

Note that we did not need to assume that the function spaces are nested, because in the proof all we need is that the functions $(\mathfrak{A}_k)$ satisfy $\mathfrak{A}_{k+1} \leq \mathfrak{A}_k$. If $\mathfrak{A}_k(f^*) = C \inf_{f\in\mathcal{F}_k} \|f - f^*\|^2$, then the nestedness of $(\mathcal{F}_k)$ implies that $(\mathfrak{A}_k)$ is a pointwise decreasing sequence of functions.

The following theorem is the main result of this section.

Theorem 5 *Assume that the conditions listed in Assumption 3 hold and the value of* a_n *given to the algorithm depends on* n *and in particular* $a_n = cn^{-1/2}$ *with some* $c > 0$. *Assume that the penalty factors,* $C_k = C_k(n)$, *passed to Algorithm 2 are such that for any fixed* k, $C_k(n)$ *is a strictly decreasing function of* n *and for any fixed* n,

$$S_n = \sum_{k\geq 1} \exp\left(-\frac{(1-a_n)^2 a_n n}{8B^2(1+a_n)\tau} C_k(n)\right) < \infty. \tag{18}$$

Let $\hat{f}$ *and* $\hat{b}$ *be the pair returned by Algorithm 2. Then, the following hold:*
(A) For any $0 < \delta \leq 1$,

$$\left\|\hat{f} - f^*\right\|^2 \leq (1 - a_n^2) \min_{k\geq 1} \left[\frac{\left\|\hat{f}_k - f^*\right\|^2}{(1-a_n)^2} + 2C_k(n)\right] + \frac{2a_n}{1-a_n} L(f^*) + \frac{16B^2(1+a_n)\tau \ln(\frac{2S_n}{\delta})}{(1-a_n)a_n n}$$

holds with probability at least $1 - \delta$, *where* $L(f) = \mathbb{E}\left[(f(X'_1) - Y'_1)^2\right]$.
(B) Fix $0 < \delta \leq 1$. *Then, there exists* $n_0 = n_0(f^*, \delta) \geq 1$ *such that for any* $n \geq n_0$, *the inequality* $\left\|\hat{f} - f^*\right\|^2 \leq \hat{b}(\delta)$ *holds with probability at least* $1 - \delta$.

Note that by selecting $a_n \propto n^{-1/2}$, Part (A) shows that the procedure's excess error above the oracle's performance is $O(n^{-1/2})$.

Proof Let $\hat{k}$ be the index selected by Algorithm 2. A standard calculation shows that $\mathbb{E}\left[\bar{\mathcal{R}}_k | \mathcal{D}_n\right] = \frac{1}{(1-a_n)^2} L(\hat{f}_k)$, where for any fixed function f, $L(f) = \mathbb{E}\left[(f(X'_1) - Y'_1)^2\right]$ denotes the squared prediction loss of f. Our goal is to apply Theorem 1 to derive a bound on $L(f_{\hat{k}})$ and then relate

$L(f_{\hat{k}})$ to the excess risk $L_{\hat{k}}$. We verify the conditions of Theorem 1. As before, the theorem is applied to the probability space obtained by conditioning w.r.t. $\mathcal{D}_n$. Let us first verify conditions (2)-(3) of Theorem 1, which connect $L(\hat{f}_k)$ and $\mathcal{R}_k$. In order to verify these conditions, we use Lemma 2. Let $g : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$ be defined by $x \mapsto (\hat{f}_k(x) - y)^2$. By assumption, the range of g is a subset of $[0, 4B^2]$. Hence, applying Lemma 2 to $Z = 1/n \sum_{i=1}^n g(X_i', Y_i')$, exploiting that $(1 - a_n)^2 \bar{\mathcal{R}}_k = Z$, after some algebra we get that for all $\varepsilon > 0$, the following inequalities are satisfied:

$$\mathbb{P}\left(L(\hat{f}_k) - (1 - a_n)\bar{\mathcal{R}}_k > \varepsilon \middle| \mathcal{D}_n \right) \le \exp\left(-\frac{(1 - a_n)a_n}{8B^2\tau(1 + a_n)}\varepsilon \right),$$
$$\mathbb{P}\left(\frac{1}{1 + a_n}\bar{\mathcal{R}}_k - \mathbb{E}\left[\bar{\mathcal{R}}_k | \mathcal{D}_n\right] > \varepsilon \middle| \mathcal{D}_n \right) \le \exp\left(-\frac{(1 - a_n)^2 a_n n}{8B^2\tau}\varepsilon \right).$$

Choosing $c_1, c_3 = 1$, $c_2 = \frac{(1-a_n)a_n n}{8B^2\tau(1+a_n)}$, and $c_4 = \frac{(1-a_n)^2 a_n n}{8B^2\tau}$, we see that conditions (2) and (3) of Theorem 1 are satisfied. Further, let c_5 (c_6) of Theorem 1 be defined as in (4) (respectively, as in (5)). Then, if $(C_k(n))$ is chosen such that (18) is satisfied, we also have $c_6 \le c_5 = S_n < +\infty$, as required. Therefore, Part (B) of Theorem 1 with the choice of $\alpha = \alpha(n, a_n, \delta)$, where

$$\alpha(n, a_n, \delta) = \frac{16B^2(1 + a_n)\tau \ln(\frac{2S_n}{\delta})}{(1 - a_n)a_n n}$$

implies that with probability $1 - \delta$,

$$L(\hat{f}_{\hat{k}}) \le (1 - a_n^2) \min_{k \ge 1} \left[\frac{1}{(1 - a_n)^2} L(\hat{f}_k) + 2C_k(n) \right] + \alpha(n, a_n, \delta).$$

Subtract $L(f^*)$ from both sides and use that $L_k = L(\hat{f}_k) - L(f^*)$ to get

$$L_{\hat{k}} \le (1 - a_n^2) \min_{k \ge 1} \left[\frac{1}{(1 - a_n)^2} L_k + 2C_k(n) \right] + \frac{2a_n}{1 - a_n} L(f^*) + \alpha(n, a_n, \delta).$$

This finishes the proof of Part (A).

Let us now prove Part (B). Fix some $0 < \delta \le 1$. Let $\mathcal{E}_1$ be the error event where

$$\left\| \hat{f}_{\hat{k}} - f^* \right\|^2 \le (1 - a_n^2) \min_{k \ge 1} \left[\frac{\left\| \hat{f}_k - f^* \right\|^2}{(1 - a_n)^2} + 2C_k(n) \right] + \frac{2a_n}{1 - a_n} L(f^*) + \alpha(n, a_n, \delta/3) \tag{19}$$

fails to hold. By Part (A), $\mathbb{P}(\mathcal{E}_1) \le \delta/3$. Let $\mathcal{E}_2$ be the error event where one of the inequalities

$$\left\| \hat{f}_k - f^* \right\|^2 \le \mathfrak{A}_k(f^*) + \mathfrak{B}_k(n, \beta_k\delta, \tau), \quad k = 1, 2, \ldots \tag{20}$$

fails to hold. By assumption and the choice of (β_k), $\mathbb{P}(\mathcal{E}_2) \le 2\delta/3$. Our goal is to show that for n large enough, outside of $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$, $\mathfrak{A}_{\hat{k}}(f^*) = 0$. Indeed, if this holds then outside of $\mathcal{E}$, $\left\| \hat{f}_{\hat{k}} - f^* \right\|^2 \le \mathfrak{A}_{\hat{k}}(f^*) + \mathfrak{B}_{\hat{k}}(n, \beta_{\hat{k}}\delta, \tau) = \mathfrak{B}_{\hat{k}}(n, \beta_{\hat{k}}\delta, \tau)$, which implies the desired statement.

In the rest of the proof, all of our derivations will be done on the event $\mathcal{E}^c$. Let k^* be the first index where $\mathfrak{A}_k(f^*) = 0$. Note that k^* is well-defined by our assumption that relates $\mathfrak{A}_k(f^*)$ to the approximation errors, $\inf_{f \in \mathcal{F}_k} \|f - f^*\|^2$, and because $f^* \in \cup_{k \ge 1} \mathcal{F}_k$. If $k^* = 1$, then $\hat{k} \ge k^*$ and thus $\mathfrak{A}_{\hat{k}}(f^*) = 0$ holds, too. Therefore, from now on assume that $k^* > 1$. From (19), it follows that

$$\left\| \hat{f}_{\hat{k}} - f^* \right\|^2 \le (1 - a_n^2) \left[\frac{\left\| \hat{f}_{k^*} - f^* \right\|^2}{(1 - a_n)^2} + 2C_{k^*}(n) \right] + \frac{2a_n}{1 - a_n} L(f^*) + \alpha(n, a_n, \delta/3).$$

By (20), we also have $\|\hat{f}_{k^*} - f^*\|^2 \le \mathfrak{A}_{k^*}(f^*) + \mathfrak{B}_{k^*}(n, \beta_{k^*}\delta, \tau) = \mathfrak{B}_{k^*}(n, \beta_{k^*}\delta, \tau)$. Chaining these inequalities gives

$$\left\| \hat{f}_{\hat{k}} - f^* \right\|^2 \le (1 - a_n^2) \left[\frac{\mathfrak{B}_{k^*}(n, \beta_{k^*}\delta, \tau)}{(1 - a_n)^2} + 2C_{k^*}(n) \right] + \frac{2a_n}{1 - a_n} L(f^*) + \alpha(n, a_n, \tfrac{\delta}{3}). \qquad (21)$$

Let n_0 be the first integer such that the right-hand side of (21) is strictly below $0 < \mathfrak{A}_{k^*-1}(f^*)/C$. Such an index exists because the right-hand side of (21) converges to zero as $n \to \infty$. Since $\hat{f}_{\hat{k}} \in \mathcal{F}_{\hat{k}}$, we have $\inf_{f \in \mathcal{F}_{\hat{k}}} \|f - f^*\|^2 \le \|\hat{f}_{\hat{k}} - f^*\|^2$. Therefore, if $n \ge n_0$, $\hat{k} = \hat{k}_n$ is such that $\mathfrak{A}_{\hat{k}}(f^*) \le C \inf_{f \in \mathcal{F}_{\hat{k}}} \|f - f^*\|^2 \le C\|\hat{f}_{\hat{k}} - f^*\|^2 < \mathfrak{A}_{k^*-1}(f^*)$ and thus, by the definition of k^*, $\mathfrak{A}_{\hat{k}}(f^*) = 0$, thus finishing the proof.

References

Antos, A., R. Munos, and Cs. Szepesvári: 2008a, 'Fitted Q-iteration in continuous action-space MDPs'. In: J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.): *Advances in Neural Information Processing Systems (NIPS - 20).* Cambridge, MA, pp. 9–16, MIT Press.

Antos, A., Cs. Szepesvári, and R. Munos: 2007, 'Value-iteration Based Fitted Policy Iteration: Learning with a Single Trajectory'. In: *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL).* pp. 330–337, IEEE.

Antos, A., Cs. Szepesvári, and R. Munos: 2008b, 'Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path'. *Machine Learning Journal* **71**, 89–129.

Arlot, S. and A. Celisse: 2009, 'A survey of cross-validation procedures for model selection'. *Statistics Surveys* **4**, 40–79.

Barron, A. R.: 1991, 'Complexity Regularization with Application to Artificial Neural Networks'. In: G. Roussas (ed.): *Nonparametric Function Estimation and Related Topics.* Kluwer Academic Publishers, pp. 561–576.

Barron, A. R., C. Huang, J. Q. Li, and X. Luo: 2008, 'The MDL Principle, Maximum Likelihoods, and Statistical Risk'. In: P. Grünwald, P. Myllymäki, I. Tabus, M. Weinberger, and B. Yu (eds.): *Festschrift in Honor of Jorma Rissanen on the Occasion of his 75th Birthday*, TICSP Series #38. Tampere International Center for Signal Processing.

Bartlett, P. L., S. Boucheron, and G. Lugosi: 2002, 'Model Selection and Error Estimation'. *Machine Learning* **48**(1-3), 85–113.

Bartlett, P. L., O. Bousquet, and S. Mendelson: 2005, 'Local Rademacher complexities'. *The Annals of Statistics* **33**(4), 1497–1537.

Bertsekas, D. P. and S. E. Shreve: 1978, *Stochastic Optimal Control: The Discrete-Time Case.* Academic Press.

Bertsekas, D. P. and J. N. Tsitsiklis: 1996, *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3).* Athena Scientific.

Bishop, C. M.: 2006, *Pattern Recognition and Machine Learning.* Springer.

Engel, Y., S. Mannor, and R. Meir: 2005, 'Reinforcement Learning with Gaussian processes'. In: *ICML '05: Proceedings of the 22nd international conference on Machine learning.* pp. 201–208, ACM.

Ernst, D., P. Geurts, and L. Wehenkel: 2005, 'Tree-based batch mode reinforcement learning'. *Journal of Machine Learning Research* **6**, 503–556.

Farahmand, A.-m., M. Ghavamzadeh, Cs. Szepesvári, and S. Mannor: 2009a, 'Regularized Fitted Q-Iteration for Planning in Continuous-Space Markovian Decision Problems'. In: *Proceedings of American Control Conference (ACC).* pp. 725–730.

Farahmand, A.-m., M. Ghavamzadeh, Cs. Szepesvári, and S. Mannor: 2009b, 'Regularized Policy Iteration'. In: D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.): *Advances in Neural Information Processing Systems (NIPS - 21).* MIT Press, pp. 441–448.

Farahmand, A.-m. and Cs. Szepesvári: 2012, 'Regularized Least-Squares Regression: Learning from a β-mixing Sequence'. *Journal of Statistical Planning and Inference* **142**(2), 493 – 505.

Györfi, L., M. Kohler, A. Krzyżak, and H. Walk: 2002, *A Distribution-Free Theory of Nonparametric Regression*. Springer Verlag, New York.

Hastie, T., R. Tibshirani, and J. Friedman: 2001, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Jung, T. and D. Polani: 2006, 'Least Squares SVM for Least Squares TD Learning'. In: *In Proc. 17th European Conference on Artificial Intelligence*. pp. 499–503.

Keller, P. W., S. Mannor, and D. Precup: 2006, 'Automatic basis function construction for approximate dynamic programming and reinforcement learning'. In: *ICML '06: Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA, pp. 449–456, ACM.

Kolter, J. Z. and A. Y. Ng: 2009, 'Regularization and feature selection in least-squares temporal difference learning'. In: *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. pp. 521–528, ACM.

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

Loth, M., M. Davy, and P. Preux: 2007, 'Sparse Temporal Difference Learning using LASSO'. In: *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. pp. 352–359.

Lugosi, G. and M. Wegkamp: 2004, 'Complexity regularization via localized random penalties'. *The Annals of Statistics* **32**, 1679–1697.

McDonald, D.: 2010, 'Generalization Error Bounds for State Space Models with an Application to Economic Forecasting'. Technical report, Department of Statistics, Carnegie Mellon University.

Meir, R.: 2000, 'Nonparametric Time Series Prediction Through Adaptive Model Selection'. *Machine Learning* **39**(1), 5–34.

Menache, I., S. Mannor, and N. Shimkin: 2005, 'Basis Function Adaptation in Temporal Difference Reinforcement Learning'. *Annals of Operations Research* **134**(1), 215–238.

Meyn, S. and R. L. Tweedie: 2009, *Markov Chains and Stochastic Stability*. New York, NY, USA: Cambridge University Press.

Modha, D. S. and E. Masry: 1998, 'Memory-Universal Prediction of Stationary Random Processes'. *IEEE Transactions on Information Theory* **44**(1), 117–133.

Parr, R., C. Painter-Wakefield, L. Li, and M. Littman: 2007, 'Analyzing Feature Generation for Value-Function Approximation'. In: *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA, pp. 737 – 744, ACM.

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

Riedmiller, M.: 2005, 'Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method'. In: *16th European Conference on Machine Learning*. pp. 317–328.

Samson, P.-M.: 2000, 'Concentration of Measure Inequalities for Markov Chains and Φ-Mixing Processes'. *The Annals of Probability* **28**(1), 416–461.

Sutton, R. S. and A. G. Barto: 1998, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press.

Szepesvári, Cs.: 2010, *Algorithms for Reinforcement Learning*. Morgan Claypool Publishers.

Taylor, G. and R. Parr: 2009, 'Kernelized value function approximation for reinforcement learning'. In: *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA, pp. 1017–1024, ACM.

van de Geer, S. A.: 2000, *Empirical Processes in M-Estimation*. Cambridge University Press.

van der Vaart, A. W., S. Dudoit, and M. J. van der Laan: 2006, 'Oracle Inequalities for Multi-fold Cross Validation'. *Statistics and Decisions* **24**, 351–372.

Wasserman, L.: 2007, *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer.

Wegkamp, M.: 2003, 'Model Selection in Nonparametric Regression'. *The Annals of Statistics* **31**(1), 252–273.

Xu, X., D. Hu, and X. Lu: 2007, 'Kernel-Based Least Squares Policy Iteration for Reinforcement Learning'. *IEEE Trans. on Neural Networks* **18**, 973–992.