Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Estimating the Maximum Expected Value in Continuous Reinforcement Learning Problems

Carlo D'Eramo, Alessandro Nuara, Matteo Pirotta, Marcello Restelli
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Piazza Leonardo da Vinci, 32, 20133, Milano, Italy
carlo.deramo@polimi.it, alessandro.nuara@mail.polimi.it, matteo.pirotta@polimi.it, marcello.restelli@polimi.it

Abstract

This paper is about the estimation of the maximum expected value of an infinite set of random variables. This estimation problem is relevant in many fields, like the Reinforcement Learning (RL) one. In RL it is well known that, in some stochastic environments, a bias in the estimation error can increase step-by-step the approximation error leading to large overestimates of the true action values. Recently, some approaches have been proposed to reduce such bias in order to get better action-value estimates, but are limited to finite problems. In this paper, we leverage on the recently proposed weighted estimator and on Gaussian process regression to derive a new method that is able to natively handle infinitely many random variables. We show how these techniques can be used to face both continuous state and continuous actions RL problems. To evaluate the effectiveness of the proposed approach we perform empirical comparisons with related approaches.

Introduction

The computation of the maximum expected value is fundamental in several applications. For example, almost any process of acting involves the optimization of an expected utility function. While sometimes only the ordering of these alternatives matters, many applications require an explicit computation of the maximal utility. This value is often required to calibrate other variables. For examples, in a financial trading system we are not only interested in detecting which is the best strategy, but we may also be interested in how much such strategy is profitable in order to evaluate the risk of a market operation or to define a proper business plan. Another example is the medical treatment strategy where it is important to compute the expected efficacy of the best therapy in order to provide accurate prognosis or to manage the hospitalization.

The most widespread approach to this estimation problem is the *Maximum Estimator* (ME) that simply takes the maximum estimated utility. As proved in (Smith and Winkler 2006), this estimate is positively biased and, if used in iterative algorithms, can increase the approximation error step-by-step (Van Hasselt 2010). More effective estimators have been proposed in the recent years. The *Double Estimator* (DE) (Van Hasselt 2013) approximates the maximum by splitting the sample set into two disjoint sample sets. One of this set is used to pick the element with the maximum approximate value and its value is picked from the other set. This has to be done the opposite way switching the role of the two sets. Eventually, the average (or a convex combination) of the two values is considered. This approach has been proven to have negative bias (Van Hasselt 2013) which, in some applications, allows to overcome the problem of ME. Finally, the recently proposed Weighted Estimator (WE) (D'Eramo, Restelli, and Nuara 2016) approximates the maximum value by a sum of different values weighted by their probability of being the maximum. WE can have both negative and positive bias, but its bias always stays in the range between the ME and DE biases.

All the mentioned approaches are limited to finite random variables and, as far as we know, the continuous case is unexplored in literature. In this paper, we leverage on the WE formalism to provide the first approach for the estimation of the maximum expected value with infinitely many random variables. WE needs to estimate the probability distribution of each random variable. By exploiting the central limit theorem, such distribution are approximated as normal distributions whose means and variance are estimated from sample observations. When moving to an infinite number of random variables, we cannot observe samples for each random variable, but we need to generalize the information collected from samples to similar variables using regression techniques. Differently from ME and DE, WE requires to measure the uncertainty of the mean estimate. Despite many regression techniques have such capability, the natural choice is to use Gaussian Process (GP) regression (Rasmussen and Williams 2005). Using GPs the distribution of each random variable is modeled as a normal distribution whose mean and variance are computed from training samples by leveraging on the spatial correlation. Finally, we leverage on the product integral (Grossman and Katz 1972) to compute the joint probability of an infinite set of events (it is required for the weights in WE).

As mentioned above, a standard applicative domain of these techniques is represented by sequential decision problems. Reinforcement Learning (RL) approaches have made wide use of these techniques. RL leverages on Markov Decision Processes (MDPs) to provide a mathematical framework for modeling sequential decision making in environments with stochastic dynamics (Puterman 1994). In MDPs, an optimal policy is the one that applies in each state the action that attains the maximum expected cumulative reward. The objective is thus to compute the maximum expected cumulative reward for each state-action pair (named optimal action-value function $Q^*(s,a)$). When the reward function and the state transition model are known it can be done by means of the Bellman optimality equation (Bellman 1957). However, in RL these two elements are unknown and the optimal action-value function must be computed in an iterative way that involves the computation of the maximum value of a partial estimate of Q^*. For instance, the update rule of the Q-Learning algorithm (Watkins 1989) is:

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha_t(s_t,a_t) \cdot \left(r_t + \gamma \max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t)\right), \tag{1}$$

where $\alpha_t(s_t,a_t)$ is a learning rate, γ is a discount factor and r_t is the immediate reward obtained by taking action a_t in state s_t. Under mild assumptions, it can be shown that the above update rule converges to Q^* in the limit. Nonetheless, at the beginning of the learning process, the estimates of the action values have usually a very low accuracy. As a consequence, the ME may introduce a large positive bias (Smith and Winkler 2006) that, given the recursive nature of the update, propagates to the other state-action pairs resulting in very bad results (Van Hasselt 2010). This issue is shared by all the off-policy value-based approaches like Q-learning and its variants (e.g., Delayed Q-Learning (Strehl et al. 2006), Phased Q-Learning (Kearns and Singh 1999), Fitted Q-Iteration (Ernst, Geurts, and Wehenkel 2005)) that need to estimate the maximum action values for different states.

Recently both DE and WE have found application in RL. DE has been exploited within Double Q-Learning (Van Hasselt 2010) and Double DQN (Van Hasselt, Guez, and Silver 2015), where they both achieve better results than, respectively, Q-Learning and DQN, in particular when the reward is noisy. WE has been applied in the Weighted Q-Learning algorithm (D'Eramo, Restelli, and Nuara 2016) attaining good performance in different problems where either Q-Learning or Double Q-Learning performed poorly, showing superior robustness. Other ad-hoc techniques (hardly generalizable to non RL domains) have been designed to overcome such estimation issue. Under the assumption of Gaussian rewards, in (Lee, Defourny, and Powell 2013) the authors propose a Q-learning variant that corrects the positive bias of ME by subtracting a term that depends on the number of actions and the variance of the rewards. Since the positive bias of the maximum operator increases when there are multiple actions with an expected value that is close to the maximum one, a modified Bellman operator that reduces the bias by increasing the action gap (that is the difference between the best action value and the second best) was proposed in (Bellemare et al. 2016).

The *primary contribute* of this paper is a generic approach to the estimate of the maximum expected value with infinitely many random variables. Although this contribute can be exploited in several applications, we focus on the reinforcement learning scenario where we provide two additional developments. The *RL contribute* is twofold: I) we extend WE (designed for finite domains) in order to handle continuous state space; II) we provide a novel value-based method that is able to natively handle continuous actions in the approximation of the optimal action-value function.

The rest of the paper is organized as follows. In the next section we provide an overview of the main approaches to estimate the maximum expected value of a set of finite random variable. Then we introduce the first contribute of the paper: *how to face the case of an infinite number of random variables*. The second contribute is the application of the newly defined estimator to RL algorithms with both a finite or infinite set of action variables (but continuous states). Finally, we evaluate the proposed approaches through empirical comparison with other estimators.

Estimating the Maximum Expected Value

Statistics has widely focused on the analysis of the maximum expected value of a set of i.i.d. random variables obtaining several results (Shaked 1975; Chow and Teugels 1978). A more complicated and less studied scenario is the one where the random variables are not identically distributed. Formally, given a set of $M \geq 2$ independent random variables $X = \{X_1, \dots, X_M\}$, where each variable X_i has unknown mean μ_i and variance σ_i^2, we are interested in finding the maximum expected value defined as

$$\mu_*(X) = \max_i \mu_i = \max_i \int_{-\infty}^{\infty} x f_i(x)\,\mathrm{d}x, \tag{2}$$

where $f_i : \mathbb{R} \to \mathbb{R}$ is the probability density function (PDF) of variable X_i. In most of the cases f_i is unknown and μ_* cannot be found analytically. The maximum expected value can be approximated with $\hat{\mu}_*(S) \approx \mu_*(X)$ using a set of noisy samples $S = \{S_1, \dots, S_M\}$ retrieved from the unknown distribution of each X_i. The means $\hat{\mu}_1, \dots, \hat{\mu}_M$ of these samples are unbiased estimators of the true means μ_i. The sample means are used by some methods to approximate the true maximum expected value that, unfortunately, does not have an unbiased estimator as shown by (D., Shabma, and Krishnamoorthy 1985).

Maximum Estimator. The *Maximum Estimator* (ME) method approximates the maximum expected value with the maximum of the sample means:

$$\hat{\mu}_*^{\mathrm{ME}}(S) = \max_i \hat{\mu}_i(S) \approx \mu_*(X). \tag{3}$$

This estimator has a positive bias (Smith and Winkler 2006) that can be explained as follows. Since $\hat{\mu}_i(S)$ is an unbiased estimate of μ_i, it follows that $\max_i \hat{\mu}_i(S)$ is an unbiased estimate for $\max_i \mu_i$. Then, considering the cumulative density function (CDF) $F_{\hat{\mu}_*}(x)$ that represents the probability that the ME is less than or equal to x and, also, the probability that all $\hat{\mu}_i$ are less than or equal to x: $F_{\hat{\mu}_*}(x) = \prod_{i=1}^{M} P(\hat{\mu}_i \leq x) = \prod_{i=1}^{M} F_{\hat{\mu}_i}(x)$ and considering the PDF $f_{\hat{\mu}_*}$, the expected value of the ME is:

$$\mathbb{E}\left[\hat{\mu}_*^{\mathrm{ME}}(S)\right] = \mathbb{E}\left[\max_i \hat{\mu}_i\right] = \int_{-\infty}^{\infty} x f_{\hat{\mu}_*}(x)\mathrm{d}x \tag{4}$$

$$= \int_{-\infty}^{\infty} x \frac{\mathrm{d}}{\mathrm{d}x} \prod_{j=1}^{M} F_{\hat{\mu}_j}(x)\mathrm{d}x = \sum_{i=1}^{M} \int_{-\infty}^{\infty} x f_{\hat{\mu}_i}(x) \prod_{j\neq i}^{M} F_{\hat{\mu}_j}(x)\mathrm{d}x,$$

where, everywhere, the expectation is taken w.r.t. all the possible sample sets S. This formula shows that the expected value of the ME is not the maximum expected value in equation (2) and the positive bias can be explained by the presence of the x in the integral which correlates with the monotonically increasing product $\prod_{j\neq i}^{M} F_{\hat{\mu}_j}(x)$.

Double Estimator. To overcome the issues due to the positive bias introduced by ME, the *Double Estimator* (DE) was proposed in (Van Hasselt 2010; 2013). DE is a cross-validation approach that splits the set of samples S into two disjoint subsets $S^A = \{S_1^A, \dots, S_M^A\}$ and $S^B = \{S_1^B, \dots, S_M^B\}$ and uses the sample means $\hat{\mu}_i^A$ and $\hat{\mu}_i^B$ (computed over S_i^A and S_i^B respectively) that are unbiased estimates of the true means if the sets are split in a proper way (e.g., randomly). Then, the best action a^* according to the samples means computed using S^A ($\hat{\mu}_{a^*}^A = \max_i \hat{\mu}_i^A$) is used to pick the sample mean $\hat{\mu}_{a^*}^B$ that estimates $\max_i \mathbb{E}\left[\hat{\mu}_i^B\right] \approx \max_i \mu_i$. This has to be done the opposite way considering the estimator b^* over the sample S^B and picking the sample mean $\hat{\mu}_{b^*}^A$. The DE finally uses the average (or a convex combination) of the two picked sample means. The expected value of DE is a weighted sum of the expected values of the sample means in one set weighted by the probability of each sample mean to be the maximum in the other set:

$$\sum_{i=1}^{M} \mathbb{E}\left[\hat{\mu}_i^B\right] P(i = a^*) = \sum_{i=1}^{M} \mathbb{E}\left[\hat{\mu}_i^B\right] \int_{-\infty}^{\infty} f_{\hat{\mu}_i}^A(x) \prod_{i\neq j}^{M} F_{\hat{\mu}_j}^A(x)\mathrm{d}x.$$

As a consequence, the DE has some negative bias since it may give some weight also to variables whose expected value is less than the maximum.

Weighted Estimator. The *Weighted Estimator* (WE) (D'Eramo, Restelli, and Nuara 2016) estimates the maximum expected value with a weighted mean of the sample averages:

$$\hat{\mu}_*^{\mathrm{WE}}(S) = \sum_{i=1}^{M} \hat{\mu}_i(S) w_i^S, \tag{5}$$

where w_i^S represents the probability of $\hat{\mu}_i(S)$ being the maximum among all the means. The distributions of the sample means $\hat{\mu}_i(S)$ are approximated considering that, as stated by the central limit theorem (CLT), the distribution $f_{\hat{\mu}_i}^S$ approaches the normal distribution $\mathcal{N}\left(\mu_i, \frac{\sigma_i^2}{|S_i|}\right)$ as the sample number $|S_i|$ increases. Thus, the weights in (5) are computed replacing the distribution of the sample means $f_{\hat{\mu}_i}^S$ with a normal distribution $\tilde{f}_{\hat{\mu}_i}^S = \mathcal{N}\left(\hat{\mu}_i(S), \frac{\widehat{\sigma}_i^2}{|S_i|}\right)$ resulting in:

$$\hat{\mu}_*^{\mathrm{WE}}(S) = \sum_{i=1}^{M} \hat{\mu}_i(S) \int_{-\infty}^{\infty} \tilde{f}_{\hat{\mu}_i}^S(x) \prod_{j\neq i} \tilde{F}_{\hat{\mu}_j}^S(x)\mathrm{d}x. \tag{6}$$

As shown in (D'Eramo, Restelli, and Nuara 2016) the bias of WE can be either positive or negative and is always within the range of biases of ME and DE.

The Weighted Estimator with Infinitely many Random Variables

As far as we know, previous literature has focused only on the finite case and no approaches that natively handle continuous sets of random variables (e.g., without discretization) are available. Let us consider a continuous space of random variables $\mathcal{Z}$ equipped with some metric (e.g., a Polish space) and assume that variables in $\mathcal{Z}$ have some spatial correlation. Here, we consider $\mathcal{Z}$ to be a closed interval in $\mathbb{R}$ and that each variable $z \in \mathcal{Z}$ has unknown mean μ_z and variance σ_z^2. Given a set of samples S we assume to have an estimate $\hat{\mu}_z(S)$ of the expected value μ_z for any variable $z \in \mathcal{Z}$ (in the next section we will discuss the spatial assumption and we will explain how to obtain this estimate). As a result, the weighted sum of equation (5) generalizes to an integral over the space $\mathcal{Z}$:

$$\hat{\mu}_*^{\mathrm{WE}}(S) = \int_{\mathcal{Z}} \hat{\mu}_z(S)\, \mathfrak{f}_z^*(S)\mathrm{d}z, \tag{7}$$

where $\mathfrak{f}_z^*(S)$ is the probability density for z of *being the variable with the largest mean*, that plays the same role of the weights used in (5). Given the distribution $f_{\hat{\mu}_z}^S$ of $\hat{\mu}_z(S)$, the computation of such density is similar to what is done in (6) for the computation of the weights w_i^S, with the major difference that in the continuous case we have to (ideally) consider a product of an infinite number of cumulative distributions. Let us provide a tractable formulation of such density function:

$$\mathfrak{f}_z^*(S) = f\left(\hat{\mu}_z(S) = \sup_{y\in\mathcal{Z}} \hat{\mu}_y(S)\right)$$

$$= \int_{-\infty}^{\infty} f(\hat{\mu}_z(S) = x)\; P\Big(\hat{\mu}_y(S) \leq x,\ \forall y \in \mathcal{Z} \setminus \{z\}\Big)\mathrm{d}x$$

$$= \int_{-\infty}^{\infty} f_{\hat{\mu}_z}^S(x)\; P\left(\bigwedge_{y\in\mathcal{Z}\setminus\{z\}} \hat{\mu}_y(S) \leq x\right)\mathrm{d}x \tag{8}$$

$$= \int_{-\infty}^{\infty} f_{\hat{\mu}_z}^S(x)\; \frac{P\left(\bigwedge_{y\in\mathcal{Z}} \hat{\mu}_y(S) \leq x\right)}{P\left(\hat{\mu}_z(S) \leq x\right)}\mathrm{d}x \tag{9}$$

$$= \int_{-\infty}^{\infty} f_{\hat{\mu}_z}^S(x)\; \frac{\prod_{\mathcal{Z}} F_{\hat{\mu}_y}^S(x)^{dy}}{F_{\hat{\mu}_z}^S(x)}\mathrm{d}x$$

where (8)-(9) follow from the independence assumption. The term $\prod_{\mathcal{Z}} F_{\hat{\mu}_y}^S(x)^{dy} = P\left(\bigwedge_{y\in\mathcal{Z}} \hat{\mu}_y(S) \leq x\right)$ is the product integral defined in the geometric calculus (that is the generalization of the product operator to continuous supports) and can be related to the classical calculus through the following relation: $\prod_{\mathcal{Z}} F^S_{\hat{\mu}_y}(x)^{dy} = \exp\left(\int_{\mathcal{Z}} \ln F^S_{\hat{\mu}_y}(x) dy\right)$ (Grossman and Katz 1972, Sec. 2.6).

Spatially Correlated Variables

The issues that remain to be addressed are I) the computation of the empirical mean $\hat{\mu}_z(S)$ and II) the computation of the density function $f^S_{\hat{\mu}_z}$ (for each random variable $z \in \mathcal{Z}$). In order to face the former issue we have assumed the random variables to be spatially correlated. In this way we can use any regression technique to approximate the empirical means and generalize over poorly or unobserved regions.

In order to face the second issue, we need to restrict the regression class to methods for which it is possible to evaluate the uncertainty of the outcome. Let g be a generic regressor whose predictions are the mean of a variable z and the *confidence (variance) of the predicted mean* $\left(\text{i.e., } \hat{\mu}_z, \hat{\sigma}^2_{\hat{\mu}_z} \leftarrow g(z)\right)$. As done in the discrete case, we exploit the CLT to approximate the distribution of the sample mean $f^S_{\hat{\mu}_z}$ with a normal distribution $\tilde{f}^S_{\hat{\mu}_z} = \mathcal{N}\left(\hat{\mu}_z, \hat{\sigma}^2_{\hat{\mu}_z}\right)$.

As a result, the *weighted estimator for the continuous case* can be computed as follows:

$$\hat{\mu}^{\mathrm{WE}}_*(S) = \int_{\mathcal{Z}} \int_{-\infty}^{\infty} \frac{\hat{\mu}_z(S)\tilde{f}^S_{\hat{\mu}_z}(x)}{\tilde{F}^S_{\hat{\mu}_z}(x)} e^{\int_{\mathcal{Z}} \ln \tilde{F}^S_{\hat{\mu}_y}(x)\mathrm{d}y} \mathrm{d}x\mathrm{d}z. \tag{10}$$

Since in the general case no closed-form solution exists for the above integrals, as in the finite case, the WE can be computed through numerical integration (e.g., trapezoidal rule or Romberg integration).

Gaussian Process Regression. While several regression techniques can be exploited (e.g., linear regression), the natural choice in this case is the Gaussian Process (GP) regression since it provides both an estimate of the process mean and variance. Consider to have a GP trained on a dataset of N samples $\mathcal{D} = \{z_i, q_i\}_{i=1}^N$, where q_i is a sample drawn from the distribution of z_i. Our objective is to predict the target q_* of an input variable z_* such that $q_* = f(z_*) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Given a kernel function k used to measure the covariance between two points (z_i, z_j) and an estimate of the noise variance σ_n^2, the GP approximation for a certain variable z^* is $q_* \sim \mathcal{N}\left(\hat{\mu}_{z^*}, \hat{\sigma}^2_{\hat{\mu}_{z_*}} + \sigma_n^2 I\right)$ where:

$$\hat{\mu}_{z_*} = \mathbf{k}_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{q}, \tag{11}$$
$$\hat{\sigma}^2_{\hat{\mu}_{z_*}} = \mathrm{Cov}\left(\mu_{z_*}\right) = k(z_*, z_*) - \mathbf{k}_*^{\mathrm{T}} \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_*,$$

and $\mathbf{k}_*$ is the column vector of covariances between z_* and all the input points in $\mathcal{D}$ ($\mathbf{k}_*^{(i)} = K(z_i, z_*)$), K is the covariance matrix computed over the training inputs ($K^{(ij)} = k(z_i, z_j)$), and $\mathbf{q}$ is the vector of training targets. Given the mean estimate in (11), the application of ME and DE is straightforward, while using WE requires to estimate also the *variance of the mean estimates*. The variance of the GP target q_* is composed by the variance of the mean ($\hat{\sigma}^2_{\hat{\mu}_{z_*}}$) and the variance of the noise (σ_n^2) (Rasmussen and Williams 2005). As a result, by only considering the mean contribute, we approximate the distribution of the sample mean by $\tilde{f}^S_{\hat{\mu}_z} = \mathcal{N}\left(\hat{\mu}_z, \hat{\sigma}^2_{\hat{\mu}_z}\right)$ as defined in equations (11).

The Reinforcement Learning Scenario

In this section, we first introduce the basic notions about Markov Decision Processes (MDPs) and how they can be solved with the model-free value-based approaches. Since the presented methods require to estimate the maximum action-value function, in the second part of the section we extend an approach with the new techniques to face continuous state problem with both a finite and infinite number of actions.

Markov Decision Processes. A *continuous* MDP is defined as a 5-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ where $\mathcal{S}$ is the state space of the process, $\mathcal{A}$ is the action space, $\mathcal{P}$ is a Markovian transition model with $\mathcal{P}(s'|s,a)$ being the probability density of reaching state s' when taking action a in state s, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function and $\gamma \in [0,1]$ is the discount factor of future rewards. A policy π defines, in each state, a distribution on the action space ($\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$). Following a policy π, the value of an action a in a state s is the expected discounted cumulative reward that is obtained performing action a in state s and following the policy π thereafter. It can be computed as $Q^\pi(s,a) = \mathbb{E}\left[\sum_{k=0}^\infty \gamma^k r_{t+k+1} | s_t = s, a_t = a, \pi\right]$, where r_{t+1} is the reward received after the t-th transition. An *optimal* policy π^* is the one that maximizes the expected discounted cumulative reward. Optimal policies can be found computing the optimal action-values that are the values of actions when following the optimal policy. These values satisfy the Bellman optimality equation (Bellman 1957):

$$Q^*(s,a) = \int_{\mathcal{S}} P(s'|s,a) \left[R(s,a,s') + \gamma \max_{a'} Q^*\left(s', a'\right)\right] \mathrm{d}s'.$$

Among value-based methods, Q-learning (Watkins 1989) and Fitted Q-Iteration (FQI) (Ernst, Geurts, and Wehenkel 2005) are iterative online and offline algorithms that approximates the optimal action-values without the need of a model of the environment. The idea of FQI is to reformulate the RL problem as a sequence of supervised learning problems. Given a set of samples $\mathcal{D} = \{\langle s_i, a_i, s'_i, r_i \rangle\}_{1 \le i \le N}$ previously collected by the agent according to a given sampling strategy, at each iteration t, FQI builds an approximation of the optimal value-function by fitting a regression model on a bootstrapped sample set:

$$\mathcal{D}_t = \left\{\langle (s_i, a_i), r_i + \gamma \max_{a'} Q_{t-1}\left(s'_i, a'\right) \rangle\right\}_{1 \le i \le N}. \tag{12}$$

The FQI update, similarly to the Q-Learning update (see equation (1)), requires the computation of the maximum expected value of an approximation of the optimal action-value

Algorithm 1 Double FQI

```
Inputs: dataset $\mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^K$, $M$ GPs $\widehat{Q}^{a_m}$, horizon $T \in \mathbb{N}$, discrete action space $\mathcal{A} = \{a_1, \dots, a_M\}$
$\mathcal{T}_{A_0} \leftarrow \{s_i, a_i, r_i, s'_i\}_{i=1}^{\frac{K}{2}}$
$\mathcal{T}_{B_0} \leftarrow \{s_i, a_i, r_i, s'_i\}_{i=\frac{K}{2}+1}^{K}$
Train $\widehat{Q}_{A_0}^{\bar{a}}$ on $\mathcal{T}_{A_0}^{\bar{a}} = \{\langle s_i^A, r_i^A \rangle$ s.t. $a_i^A = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
Train $\widehat{Q}_{B_0}^{\bar{a}}$ on $\mathcal{T}_{B_0}^{\bar{a}} = \{\langle s_i^B, r_i^B \rangle$ s.t. $a_i^B = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
for t=1 to T do
  for i=1 to $\frac{K}{2}$ do
    for m=1 to M do
      $\hat{\mu}_m^A \leftarrow \widehat{Q}_{A_{t-1}}^{a_m}({s'_i}^A)$ (evaluate GP)
      $\hat{\mu}_m^B \leftarrow \widehat{Q}_{B_{t-1}}^{a_m}({s'_i}^B)$ (evaluate GP)
    end for
    $a^* \leftarrow \arg\max_m \hat{\mu}_m^A$
    $b^* \leftarrow \arg\max_m \hat{\mu}_m^B$
    $\mathcal{T}_{A_t} \leftarrow \mathcal{T}_{A_t} \cup \{\langle (s_i^A, a_i^A), q_i^A \leftarrow r_i^A + \gamma \hat{\mu}_{a^*}^B \rangle\}$
    $\mathcal{T}_{B_t} \leftarrow \mathcal{T}_{B_t} \cup \{\langle (s_i^B, a_i^B), q_i^B \leftarrow r_i^B + \gamma \hat{\mu}_{b^*}^A \rangle\}$
  end for
  Train $\widehat{Q}_{A_t}^{\bar{a}}$ on $\mathcal{T}_{A_t}^{\bar{a}} = \{\langle s_i^A, q_i^A \rangle$ s.t. $a_i^A = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
  Train $\widehat{Q}_{B_t}^{\bar{a}}$ on $\mathcal{T}_{B_t}^{\bar{a}} = \{\langle s_i^B, q_i^B \rangle$ s.t. $a_i^B = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
end for
```

function. As mentioned above, a well-known problem of such approaches is the positive bias introduced by the maximum operator. Secondly, the maximum on a continuous action space is usually solved using a problem dependent discretization of such space.

Algorithm Extensions. RL literature has focused on solving the bias issue in the finite scenario by exploiting DE and WE. In this paper we focus on the second issue by defining algorithms that are able to handle continuous state problems with both a finite or infinite number of actions. For simplicity we will only focus on the extension of the FQI algorithm although it is simple to apply the proposed approach to several value-based approaches (e.g., Q-Learning).

The extension of FQI with DE, that we call *Double FQI*, consists in splitting the dataset into two halves and training a regressor for each action on one half and another regressor for each action on the other half (refer to Algorithm 1).

The *Weighted FQI* for continuous states and discrete actions is reported in Algorithm 2. For each discrete action, a GP is trained on the samples in order to obtain an estimate of the action-value function. Such GPs are used to compute the weights required for the estimation of the maximum value in (5). This process is repeated for each sample in order to obtain the bootstrapped dataset.

In the case of continuous state and action spaces, a single regressor is defined over the joint space and is used to approximate the distributions involved in equation (10). The structure of the algorithm is reported in Algorithm 3. Note that the integral associated to v_i can be solved using any numerical integration method.

Algorithm 2 Weighted FQI (finite actions)

```
Inputs: dataset $\mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^K$, $M$ GPs $\widehat{Q}^{a_m}$, horizon $T \in \mathbb{N}$, discrete action space $\mathcal{A} = \{a_1, \dots, a_M\}$
Train $\widehat{Q}_0^{\bar{a}}$ on $\mathcal{T}_0 = \{\langle s_i, r_i \rangle$ s.t. $a_i = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
for t=1 to T do
  for j=1 to K do
    for m=1 to M do
      $\hat{\mu}_m, \hat{\sigma}_{\hat{\mu}_m}^2 \leftarrow \widehat{Q}_{t-1}^{a_m}(s'_j)$ (evaluate GP)
      $\tilde{f}_{\hat{\mu}_m} \leftarrow \mathcal{N}(\hat{\mu}_m, \hat{\sigma}_{\hat{\mu}_m}^2)$ ($\tilde{F}_{\hat{\mu}_m}$ is the associated CDF)
      $w_{a_m} \leftarrow \int_{-\infty}^{+\infty} \tilde{f}_{\hat{\mu}_m}(x) \prod_{k \neq m} \tilde{F}_{\hat{\mu}_m}(x) \mathrm{d}x$
    end for
    $\mathcal{T}_t \leftarrow \mathcal{T}_t \cup \{\langle (s_j, a_j), q_j \leftarrow r_j + \gamma \sum_{a_m \in \mathcal{A}} w_{a_m} \mu_{a_m} \rangle\}$
  end for
  Train $\widehat{Q}_t^{\bar{a}}$ on $\mathcal{T}_t^{\bar{a}} = \{\langle s_i, q_i \rangle$ s.t. $a_i = \bar{a}\}$ $(\forall \bar{a} \in \mathcal{A})$
end for
```

Algorithm 3 Weighted FQI$_\infty$ (continuous actions)

```
Inputs: dataset $\mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^K$, GP regressor $\widehat{Q}$, horizon $T \in \mathbb{N}$, continuous action space $\mathcal{A}$
Train $\widehat{Q}_0$ on $\mathcal{T}_0 = \{\langle (s_i, a_i), r_i \rangle\}$
for t=1 to T do
  for i=1 to K do
    $\hat{\mu}_z, \hat{\sigma}_{\hat{\mu}_z}^2 := \widehat{Q}_{t-1}(s'_i, z)\ \forall z \in \mathcal{A}$ (evaluate GP)
    $\tilde{f}_{\hat{\mu}_z} := \mathcal{N}(\hat{\mu}_z, \hat{\sigma}_{\hat{\mu}_z}^2)\ \forall z \in \mathcal{A}$ ($\tilde{F}_{\hat{\mu}_z}$ is the associated CDF)
    $v_i \leftarrow \int_{-\infty}^{\infty} \exp\left(\int_{\mathcal{A}} \ln \tilde{F}_{\hat{\mu}_y}(x) \mathrm{d}y\right) \int_{\mathcal{A}} \frac{\hat{\mu}_z \tilde{f}_{\hat{\mu}_z}(x)}{\tilde{F}_{\hat{\mu}_z}(x)} \mathrm{d}z \mathrm{d}x$
    $\mathcal{T}_t \leftarrow \mathcal{T}_t \cup \{\langle (s_i, a_i), r_i + \gamma v_i \rangle\}$
  end for
  Train $\widehat{Q}_t$ on $\mathcal{T}_t$
end for
```

Experiments

In this section we evaluate the performance of ME, DE and WE on three sequential decision-making problems: one Multi-Armed Bandit (MAB) problem and an MDP with both finite and continuous actions.

Pricing Problem In the MAB problem we validate the proposed weighted estimator with an infinite set of random variables (WE$_\infty$) and we compare its performance against ME and DE whose support (actions) has been discretized. The problem consists in estimating the maximum expected value of the gross profit in a pricing problem. An accurate estimation of this value can be crucial in order to evaluate, for example, an investment decision or to analyze products profitability. The support (action) space is bounded but continuous, and represents the price p to be shown to the user ($p \in [0, 10]$). The reserve price τ , which is the highest price that a buyer is willing to pay, is modeled as a mixture of 3 Gaussian distributions with mean $\mu = \{2, 4, 8\}$, covariances $\sigma^2 = \{0.01, 0.01, 0.09\}$ and weights $w = \{0.6, 0.1, 0.3\}$. The revenue function $r_\tau(p)$ is p when $\tau \geq p$ and 0 otherwise. The maximum revenue is about 2.17.

In each test the algorithms are fed with a set of samples $\mathcal{D} = \{\langle p_i, r_i \rangle\}_{i=1}^{n_s}$. Each sample is obtained by sampling a reserve price τ_i from the Gaussian mixture, a price p_i from a uniform distribution over the price range, and by evaluat-

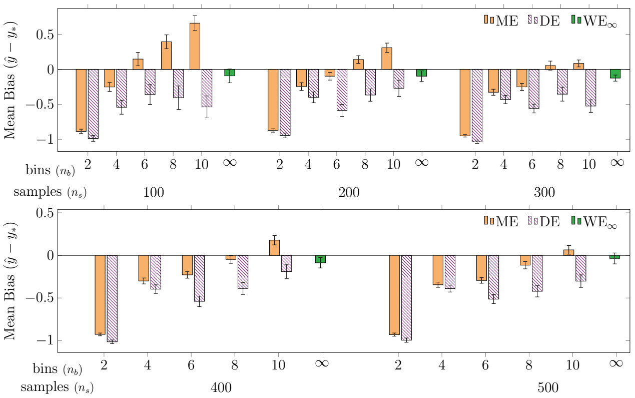

Figure 1: Mean bias obtained by ME, DE and WE$_\infty$ with different sample sizes and bins (only for ME and DE).

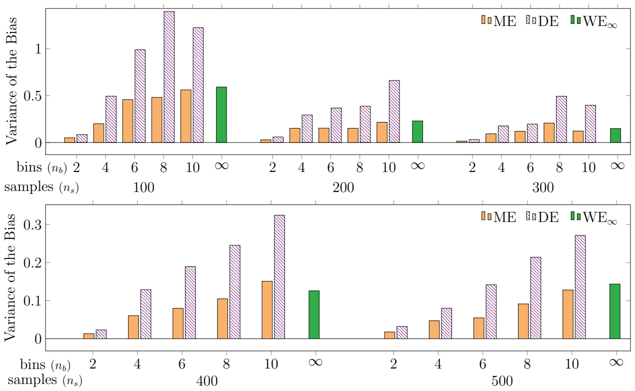

Figure 2: Variance of the bias obtained by ME, DE and WE$_\infty$ with different sample sizes and bins.

ing the revenue function ($r_i = r_{\tau_i}(p_i)$). Clearly, the reserve price is unknown to the algorithm. Results are averaged on 50 runs in order to show confidence intervals at 95%. WE exploits a Gaussian process with squared exponential kernel to generalize over the continuous price (GP parameters are learned from $\mathcal{D}$), while ME and DE discretize the price space into n_b uniformly spaced bins. As shown in Figure 1, the number n_b of optimal bins varies with the number n_s of available samples. This means that, once the samples have been collected, ME and DE need an optimization phase for selecting the appropriate number of bins (not required by WE). WE is able to achieve the lowest or a comparable level of bias with every batch dimension even through it exploits a sensibly wider action space (infinite). In fact, as shown by the experiments, the performance of ME and DE may degrade as the number of bins increases, i.e., the action space increases. This means that, if you want to be accurate, you cannot increase the number of bins arbitrarily (it is somehow counterintuitive). Additionally, Figure 2 shows that the higher complexity of WE has practically no impact on the variance of the estimate. The variance is always comparable to the one of the best configuration of WE and DE.

Finally, several applications do not consider positive and negative bias to be the same, in particular, in iterative application positive bias can lead to large overestimates that have proven to be critical (e.g., in RL). This is not the case because this pricing problem is not iterated. From Figure 1 we can see that ME is prone to provide positive bias, while WE bias is almost always the smaller or stays between ME and DE. The reason for which the ME bias is not always positive, as stated by its theoretical property (for finite case), is due to the use of binning for the discretization of the continuous MAB. This discrete approximation introduces an additional (here negative) term to the bias.

Swing-up Pendulum A more complex scenario is represented by the continuous control problem analyzed in this section: the swing-up pendulum with limited torque (Doya 2000). The aim of these experiments is to compare the newly proposed extensions of FQI (Double FQI and Weighted FQI) in a continuous state domain with both discrete and continuous actions. The peculiarity of this domain resides in the fact that the control with a limited torque ($u \in [-5, 5]$) makes the policy learning non-trivial. The continuous state space is $x = (\theta, \omega)$, where θ is the angle and ω is the angular velocity. An episode starts with $x_0 = (\theta_0, 0)$ where $\theta_0 \sim \mathcal{U}(-\pi, \pi)$, evolves according to the the dynamic system $\dot{\theta} = \omega$ and $ml^2\dot{\omega} = -\mu\omega + mgl\sin(\theta) + u$, and terminates after 100 steps. The physical parameters are mass $m = 1$, length $l = 1$, $g = 9.8$, step time $\tau_0 = 0.01$. The reward depends on the height of the pendulum: $r(x) = \cos(\theta)$. The problem is discounted with $\gamma = 0.9$. The GP uses a squared exponential kernel with independent length scale for each input dimension (ARD SE). The hyperparameters are fitted on the samples and the input values are normalized between $[-1, 1]$. We collected training sets of different sizes using a random policy. The FQI horizon is 10 iterations. The final performance of the algorithm is the *average reward*, calculated starting from 36 different initial angles $\theta_0 = \{\frac{2\pi k}{36} | k = \{0, 1, \dots, 35\}\}$.

In the first experiment we compare Double FQI, Weighted FQI (Algorithm 2) and FQI on a continuous state problem with discrete actions using a different GP for each action. The actions are the 11 integer torque values in $[-5, 5]$. Results show that Weighted FQI and FQI are robust with respect to the number of episodes and Weighted FQI reaches the highest average reward in each case (with statistical confidence obtained over 100 runs and level 95%). Double FQI performance is reasonably poor with few examples since it uses a half of the training set to train each regressor.

The second experiment is designed to show the behavior of the algorithms in a continuous action MDP. The only algorithm that is able to directly handle continuous space is the Weighted FQI defined in Algorithm 3. The other algorithms use a linear discretization of the infinite action space picking 100 actions. As shown in Table 1, the behavior observed in the finite domain is preserved. The limited number of repetitions (20) allows to derive results with statistical significance only for DE, that is the algorithm that suffers the most these settings.

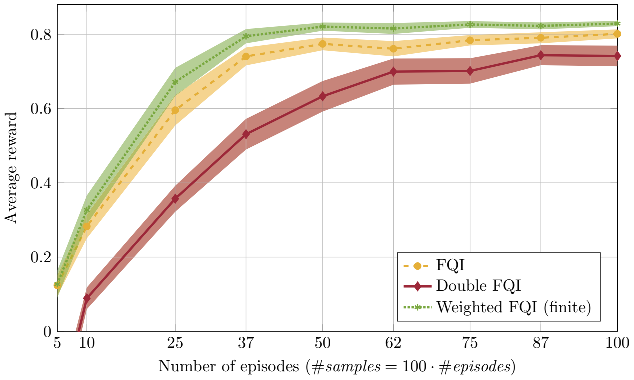

Figure 3: Average reward of the policies found by the three algorithms on different dataset sizes.

Table 1: Average reward in continuous action MDP.

Episodes	FQI	Double FQI	Weighted FQI$_\infty$
5	0.412 ± 0.131	0.120 ± 0.086	0.426 ± 0.119
10	0.650 ± 0.121	0.465 ± 0.138	0.695 ± 0.097
15	0.713 ± 0.095	0.587 ± 0.144	0.762 ± 0.062
20	0.793 ± 0.044	0.767 ± 0.091	0.823 ± 0.034

Conclusion

We have proposed an extension of WE that is able to estimate the expected maximum with infinitely many random variables. We have successfully tested such approach on a pricing problem with continuous price support. Despite ME and DE which employ a discretization of the continuous space, our WE estimator is able to directly deal with an infinite number of variables. We also discussed how ME, DE and classical WE can be used in continuous MDPs. We leveraged on GPs in order to generalize over the continuous state space. Empirical results confirm the expected behavior showing that WE estimation of the maximum is more robust. Finally, we have presented an off-line value-based algorithm able to natively handle continuous action space in the computation of $\max$ operator involved in the optimal Bellman equation. Although the continuous WE estimator has proved to be effective in the experiments, we think that a theoretical analysis is worth in order to formally explain the experienced behavior.

References

Bellemare, M. G.; Ostrovski, G.; Guez, A.; Thomas, P. S.; and Munos, R. 2016. Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the thirtieth AAAI Conference on Artificial Intelligence*.

Bellman, R. 1957. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press.

Chow, T. L., and Teugels, J. L. 1978. The sum and the maximum of iid random variables. In *Proceedings of the 2nd Prague Symposium on Asymptotic Statistics*, 81–92.

D., B. I.; Shabma, D.; and Krishnamoorthy, K. 1985. Non-existence of unbiased estimators of ordered parameters. *Statistics* 16(1):89–95.

D'Eramo, C.; Restelli, M.; and Nuara, A. 2016. Estimating maximum expected value through gaussian approximation. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1032–1040. JMLR.org.

Doya, K. 2000. Reinforcement learning in continuous time and space. *Neural computation* 12(1):219–245.

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

Grossman, M., and Katz, R. 1972. *Non-Newtonian Calculus: A Self-contained, Elementary Exposition of the Authors' Investigations...* Non-Newtonian Calculus.

Kearns, M., and Singh, S. 1999. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems* 996–1002.

Lee, D.; Defourny, B.; and Powell, W. B. 2013. Bias-corrected q-learning to control max-operator bias in q-learning. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*, 93–99. IEEE.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: Wiley-Interscience.

Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Shaked, M. 1975. On the distribution of the minimum and of the maximum of a random number of iid random variables. In *A Modern Course on Statistical Distributions in Scientific Work*. Springer. 363–380.

Smith, J. E., and Winkler, R. L. 2006. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science* 52(3):311–322.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, 881–888. ACM.

Van Hasselt, H.; Guez, A.; and Silver, D. 2015. Deep reinforcement learning with double q-learning. *CoRR* abs/1509.06461.

Van Hasselt, H. 2010. Double q-learning. In *Advances in Neural Information Processing Systems*, 2613–2621.

Van Hasselt, H. 2013. Estimating the maximum expected value: an analysis of (nested) cross-validation and the maximum sample average. *arXiv preprint arXiv:1302.7175*.

Watkins, C. J. C. H. 1989. *Learning from delayed rewards*. Ph.D. Dissertation, University of Cambridge England.