Exploration Driven by an Optimistic Bellman Equation

Samuele Tosatto
Intelligent Autonomous Systems
Technische Universität Darmstadt
Darmstadt, Germany
samuele@robot-learning.de

Carlo D'Eramo
AIRLab
Politecnico di Milano
Milano, Italy
carlo.deramo@polimi.it

Joni Pajarinen
Intelligent Autonomous Systems
Technische Universität Darmstadt
Darmstadt, Germany
joni@robot-learning.de

Marcello Restelli
AIRLab
Politecnico di Milano
Milano, Italy
marcello.restelli@polimi.it

Jan Peters
Intelligent Autonomous Systems
Technische Universität Darmstadt
Darmstadt, Germany
jan@robot-learning.de

***Abstract*—Exploring high-dimensional state spaces and finding sparse rewards are central problems in reinforcement learning. Exploration strategies are frequently either naïve (e.g., simplistic ϵ-greedy or Boltzmann policies), intractable (i.e., full Bayesian treatment of reinforcement learning) or rely heavily on heuristics. The lack of a tractable but principled exploration approach unnecessarily complicates the application of reinforcement learning to a broader range of problems. Efficient exploration can be accomplished by relying on the uncertainty of the state-action value function. To obtain the uncertainty, we maintain an ensemble of value function estimates and present an optimistic Bellman equation (OBE) for such ensembles. This OBE is derived from a relative entropy maximization principle and yields an implicit exploration bonus resulting in improved exploration during action selection. The implied exploration bonus can be seen as a well-principled type of intrinsic motivation and exhibits favorable theoretical properties. OBE can be applied to a wide range of algorithms. We propose two algorithms as an application of the principle: Optimistic Q-learning and Optimistic DQN which outperform comparison methods on standard benchmarks.**

***Index Terms*—Reinforcement Learning; Exploration; Model Ensemble; Optimism**

I. Introduction

In recent years, RL has made enormous advances, especially on high-dimensional tasks, such as Atari games [21]. One of the open problems in such complex domains is how to explore the environment in order to uncover sparse valuable states. Before seeing these states the agent does not necessarily have much information to base its decisions on. For example, an agent may always perceive a null reward except for a terminal state that is particularly difficult to reach. Before the agent reaches the terminal state and observes the subsequent reward, it cannot connect its actions to rewards. In this particular case, the traditional quest for solving the exploration/exploitation trade-off (near)-optimally makes no sense since the agent has no information to reason about possible rewards that it has not yet observed. A classical way to solve this problem is to explore randomly. However, classical exploration approaches such as ϵ-greedy may fail as the probability of reaching the positive reward can be low. A more effective strategy should take into account the underlying uncertainty and try to minimize it, in order to maximize the information gain. Bayesian approaches consider the uncertainty in a principled way but are often computational demanding [8], [41]. Recently, computationally feasible algorithms inspired by Bayesian principles have been introduced [2], [25]. Bootstrapped DQN (BDQN) [25] uses an ensemble of value functions in order to have different estimates of the Q-value function approximating posterior sampling [35]. However, to the best of our knowledge, there is no algorithm among these approximate techniques that is particularly suited for very sparse rewards in high dimensional state space. Our hypothesis is that Bayesian methods are in general more focused on balancing between exploration and exploitation while they cannot achieve deep exploration. The broad category of algorithms based on *intrinsic motivation* (IM) [34], have less theoretical guarantees than Bayesian approaches, yet they have obtained impressive results for example in the challenging Montezuma's Revenge task [3]. IM algorithms define an additional *intrinsic* reward, which acts as an exploration bonus. Often, the additional reward is defined using heuristics, such as counting state visits and rewarding less visited states [27], or by "surprise", that is, the error in predicting future states [28]. The drawback of IM techniques is their lack of a principled definition of the intrinsic reward for exploration.

Another related class of techniques is based on optimism; which provides an optimistic estimation under uncertainty, encouraging in this way exploration of uncertain region. Optimism can be categorized in 1) *optimistic initial values*, where the main concept is to initialize the Q-value function with high values in order to ensure enough exploration [9], [37]; and 2) methods that require *confidence interval estimation*, such as IEQL+ [19] which directly estimates Q-value confidence intervals.

a) Contribution: We introduce a novel Optimistic Bellman Equation (OBE). The OBE results in an optimistic Q-value estimate from an ensemble of value functions where the optimistic estimate is obtained from a maximum-entropy principle. For the exploration bonus that OBE implicitly defines, we can prove that the bonus decreases consistently with the number of state visits. Our proposed algorithm can be seen as a mixture of different techniques: as an approximated Bayesian method, we estimate the uncertainty with an ensemble; like optimism-based methods, we select optimistic estimates, and like IM, we propagate an implicit exploration bonus. Nevertheless, OBE can be applied to a wide range of algorithms by introducing an ensemble for the estimation of the critic (as done in [25], [6]) and a softmax for updating the entries.

A. Problem Statement and Notation

An infinite-horizon discounted time-discrete Markov Decision Process (MDP) is defined by a tuple of $<\mathcal{S}, \mathcal{A}, R, P, \gamma, \mu_0>$ where $\mathcal{S}$ is the set of the states, $\mathcal{A}$ is a finite set of actions, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{M}(\mathbb{R})$ where $\mathcal{M}(\mathcal{Z})$ denotes the sets of probability measures over the space Z, $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{M}(\mathcal{S})$ is the transition distribution, $\gamma \in [0, 1)$ denotes the discount factor and μ_0 is the initial state distribution. We define the set of deterministic policies as $\Pi : \mathcal{S} \rightarrow \mathcal{A}$. Our goal is to find the optimal policy $\pi^* \in \Pi$ that maximizes the expected return J^π

$$J^\pi = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{(t-1)} r_t\right],$$

where $r_t \sim R(s_t, a_t)$, $s_t \sim P(\cdot|s_{t-1}, a_{t-1})$, $a_t = \pi(s_t)$, $s_1 \sim \mu_0$. A common approach to solve such a problem is to find the so-called optimal Q-value function Q^*, which is the solution to the optimal Bellman equation (BE):

$$Q^*(s, a) = \overline{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) \max_{a' \in \mathcal{A}} Q^*(s', a'), \quad (1)$$

where $\overline{R}(s, a) = \mathbb{E}[R(s, a)]$ and the summation could be replaced by an integral, depending on whether $\mathcal{S}$ is discrete or continuous. Once the optimal Q-value function Q^* is found, we know that the optimal policy is equivalent to $\pi^*(s) = \arg\max_a Q(s, a)$.

B. Related Work

The algorithm we propose could be considered as a mixture of different techniques: as an approximated Bayesian method [8], [41], we estimate the uncertainty with an ensemble [25]; like optimism [2], [4], [15], [16], we select optimistic estimates, and like intrinsic motivation (IM) approaches [31], [34], [43], we propagate an implicit exploration bonus. Unlike many IM techniques, our bonus is defined implicitly in a so-called *optimistic Bellman equation* (OBE), by selecting an optimistic estimate from an ensemble of value functions.

Intrinsic motivation algorithms define an additional *intrinsic* reward often using heuristics, such as counting state visits and rewarding less visited states [3], [27], or by "surprise", that is, the error in predicting future states [28]. Approaches based on the optimism in the face of uncertainty principle [4], [15], [16] add an additional reward term to state-action pairs proportional to the amount of uncertainty. The amount of uncertainty, and thus the additional reward, usually depends on the amount of information collected of a state-action pair. Due to explicit uncertainty modeling these methods are able to restrict exploration to regions where the policy is still far from the optimal solution. Often these methods provide performance guarantees. However, in practice the method may not always converge [13], [25]. Bayesian posterior sampling has shown performance improvements over optimism in the face of uncertainty methods [10], [26], [35].

There is a broad range of recent work on exploration in high dimensional state-spaces. UCB Exploration via Q-Ensembles [6], similarly to our approach, uses M different estimates of the Q value function in order to infer the uncertainty of the estimate. In particular, [6] use an optimistic bound for the policy:

$$a \in \arg\max_a \mu(s, a) + \lambda\sigma(s, a).$$

compute directly an estimate of the standard deviation over multiple estimations of the Q function, and use it to guide the exploration. However, such methods do not propagate the variance through the Bellman equation, and thus the agent is not able to be foresighted w.r.t. future exploration possibilities. Contrary to these approaches, the recent "uncertainty Bellman equation" (UBE) [23] propagates variance estimates of the Q-value with a Bellman recursion and uses the estimates for posterior sampling (our action selection is deterministic). For tabular policies, UBE estimates local uncertainty proportional to the inverse visitation count similarly to count based approaches and for neural network policies a linear uncertainty approximation is used. Instead, our implicit state/action specific uncertainty estimation is based on the diversity in the Q-function ensemble fwhile taking future Q-function uncertainty into account. Moreover, contrary to our approach, UBE assumes that the MDP is a directed acyclic graph.

Entropic regularization has been extensively used in reinforcement learning, either for ensuring stability [20], [29], [32] and sample efficiency or for providing risk awareness [11], [17], [30]. However, the entropic regularization is usually performed on the state-action distribution [22], and thus applied to the so-called "aleatoric" uncertainty. This kind of uncertainty is related to the intrinsic stochasticity of the MDP and to the state-action distribution induced by the policy. The Boltzmann policy [14], [38] or the soft Bellman equations are derived from this principle. In contrast, our entropic regularization is applied to the "epistemic" uncertainty, or, in other words, to the model uncertainty (in this specific case to the Q-function uncertainty). To the best of our knowledge this is the first attempt to use entropic regularization on the epistemic uncertainty to drive exploration.

II. Learning Value Function Ensembles with Optimistic Estimate Selection

Ensemble methods [24] are a prevalent machine learning technique where multiple models are used to learn the same target function. In addition to being commonly used to improve the generalization of the prediction, ensemble methods offer a simple way to estimate the uncertainty of the prediction. We consider the application of ensemble methods in the RL framework with the purpose of approximating the action-value functions. Indeed we want to obtain a cheap estimate of the uncertainty of action-values, in order to apply the *optimism in the face of uncertainty* (OFU) principle in action selection.

A. An Optimistic Bellman Equation for Action-Value Function Ensembles

The core of our work consists of a BE for action-value ensembles which incorporates the information about the uncertainty provided by a Q-function ensemble. In more detail, we want to overestimate the action-value functions with the result of encouraging exploration. Thus, we propose an optimistic Bellman equation (OBE) which propagates an optimistic estimate of the action-value function. We want to emphasize that when all the Q-functions of the ensemble are identical, we assume that there is no uncertainty, and under this condition the OBE will behave exactly equivalently to the classic BE. The solution Q^* of OBE is the same of the classic BE. In other words, the OBE differs from the classic BE when it is not satisfied, and more precisely when approximation is introduced either by limited availability of samples and/or functional approximation. This makes sense, since when the perfect solution is available there is no need for optimism and exploration. The optimistic Bellman operator derived from the OBE, enjoys the classical properties, like contractivity and the existence of a unique fixed point, enabling its usage in value-based or actor-critic reinforcement algorithms. The diversity in the Q-value ensemble should be ideally consistent with the uncertainty of the estimation; e.g. when the estimate is certain, all the values in the ensemble should agree on the same value, otherwise the ensemble should have discordant values.

Given an ensemble of Q-value functions $\{Q_m\}_{m=1}^M$, we want to work out an optimistic estimate from the diverse estimates provided by the ensemble. The simplest and most optimistic solution is to select the highest value

$$\max_m Q_m(s,a).$$

However selecting the highest estimate makes poor use of the information provided by the ensemble and can be sensible to noise. In order to mitigate this effect, we introduce a notion of *belief* over the estimates where $b_m(s,a)$ is the belief of $Q_m(s,a)$. The main idea is to add an entropic regularization term to the objective (i.e., $\max_{b(s,a)} \sum_m b_m(s,a)Q_m(s,a) - c\sum_m p_m \log p_m$); or to bound the information loss (i.e., -$\sum_m p_m \log p_m \geq \psi$). Hard constraint on the information loss is more appealing since the introduced hyper-parameter does not depend on the magnitude of the rewards but has no closed-form solution. In contrast, the penalization weighting constant introduced by the soft-constraint regularization term is sensitive to the magnitude of the rewards, but admits a closed-form solution. We define two different problems where we use an optimistic estimate of the Q-value function.

a) Entropy-Regularized Optimistic Q Selection: We define here a Bellman equation over the Q-function ensemble by introducing an optimistic estimate penalized by an entropic regularization term.

Problem 1 (Regularized version)**.**

$$Q_i(s,a) = \max_{b(s,a)\in\mathcal{P}^M} f\big(s,a;b(s,a)\big) - \tfrac{1}{\eta} D_{\mathrm{KL}}\big(p(s,a)\big\|u\big)$$

$$\textit{s.t.} \sum_{m=1}^M b_m(s,a) = 1$$

$$\forall s,a,i \in \mathcal{S}\times\mathcal{A}\times\{1,\dots,M\}$$

where $f(a,s;p) = R(s,a) + \gamma\sum_m b_m(s,a)V'_m(s,a)$, $V'_m(s,a) = \sum_{s'} P(s'|s,a)\max_{a'} Q_m(s',a')$,$u_m = 1/M$, $D_{KL}(b(s,a)\|u)$ *is the Kullback-Leibler divergence between the belief* $b(s,a)$ *and the uniform distribution* u*.*

The choice of using the relative entropy instead of the absolute one has two main advantages: it admits a solution for $\eta \to 0$ and provides a normalization factor. Since problem definition 1 is a convex constrained problem, it is solvable by dual optimization. Introducing λ as Lagrangian multiplier for the constraint, we write the Lagrangian

$$\begin{aligned} L_i(s,a) &= f(s,a;b(s,a)) - \frac{1}{\eta} D_{\mathrm{KL}}\big(b(s,a)\big\|u\big) \\ &\quad +\lambda\bigg(\sum_m b_m(s,a) - 1\bigg). \end{aligned} \tag{2}$$

Requiring the partial derivatives of L_i w.r.t p_m and λ to be zero yields

$$b_m(s,a) = \frac{e^{\eta\gamma V'_m(s,a)}}{\sum_{k=1}^M e^{\eta\gamma V'_k(s,a)}}. \tag{3}$$

By substituting b_m in (2), we obtain the solution to the problem (a detailed derivation is provided in the technical report [40])

Optimistic Bellman Equation 1. [1]

$$Q_i(s,a) = \begin{cases} \overline{R}(s,a) + \frac{1}{\eta}\log\frac{\sum_{m=1}^M e^{\eta\gamma V'_m(s,a)}}{M} & \text{if} \eta \neq 0 \\ \overline{R}(s,a) + \frac{\gamma}{M}\sum_{m=1}^M V'_m(s,a) & \text{otherwise} \end{cases}. \tag{4}$$

Notice that $\eta > 0$ leads to a positive (optimistic) biased estimation, while $\eta < 0$ will leads to a negative (pessimistic) estimate; in this work we will always assume $\eta > 0$ (and therefore we refer to the equation as optimistic). However, in general, the choice of η is difficult since it depends on the magnitude of the reward function. For this reason we introduce the constrained version of the proposed problem.

[1] We extend the solution for $\eta = 0$ by computing the limit.

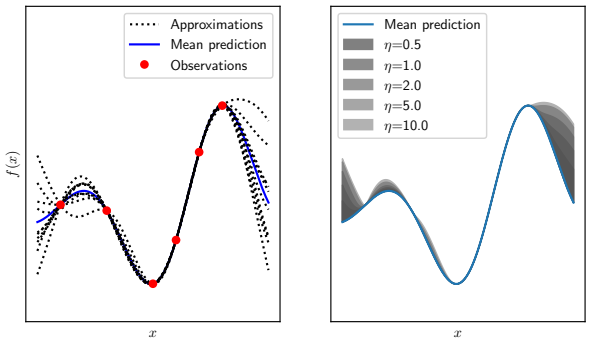

Fig. 1: **Left:** Different estimates of a function. **Right:** The entropic-map combines the function estimates to obtain an optimistic estimate where η controls the level of optimism.

b) Optimistic Q Selection Bounding the Information Loss: We bound the information loss between the distribution b_m and the uniform distribution to maintain compatibility with Problem 1. The information loss is bounded between $-\log M$ and 0 where $-\log M$ stands for complete information loss (i.e., only one model is selected) while 0 corresponds to no information loss (i.e., uniform belief distribution). Constraining the information loss has succeeded in prior work, for instance in policy search methods such as [29].

Problem 2 (Constrained version)**.**

$$\begin{aligned} Q_i(s,a) = \max_{b(s,a)\in\mathcal{P}^M} \quad & f(s,a;b(s,a)) \\ s.t. \quad & D_{\mathrm{KL}}\big(b(s,a)\big\|u\big) \leq \iota_{\max} \\ & \textstyle\sum_{m=1}^{M} b_m(s,a) = 1 \\ \forall s,a,i \in \mathcal{S}\times\mathcal{A}\times\{1,\dots,M\} & \end{aligned}$$

By letting β be the Lagrangian multiplier associated with the KL constraint, we obtain the Lagrangian

$$\begin{aligned} L_i = \; & f(s,a;b(s,a)) + \beta\Big(D_{\mathrm{KL}}\big(b(s,a)\big\|u\big) - \iota_{\max}\Big) \\ & +\lambda\Big(\sum_m b_m(s,a) - 1\Big). \end{aligned} \tag{5}$$

Substituting β with $-1/\eta$ we note that (5) becomes identical to (2) except for a constant factor. Since we can not solve η (or β) analytically, we obtain an approximate solution by iteratively optimizing η (or β) and b_m subsequently. OBE takes its name from the fact that when $\eta > 0$, the *log-sum-exp* acts as a *soft-max* operator. Such operator is also well known as an *entropic mapping*, as it can be derived from a maximum-entropy principle. Figure 1 shows how the entropic mapping works. The use of the entropic mapping is not new in reinforcement learning: [1] propose an interesting use of the entropic mapping as a soft-max over the action in the Bellman equation; [29] instead obtain it from an entropic regularization over the state-action distribution. However, as we discussed in Section I-B, differently from the mentioned approaches, our entropic regularization is applied to the epistemic uncertainty.

c) Relation to Intrinsic Motivation: In order to highlight the connection between OBE and IM, we reformulate OBE utilizing the unbiased average of the estimates instead of the log-sum-exp, and by introducing the resulting exploratory bonus U which includes the positive bias

$$Q_i(s,a) \;=\; \overline{R}(s,a) + U(s,a) + \gamma \sum_{m=1}^{M} \frac{V'_m(s,a)}{M} \tag{6}$$

with

$$U(s,a) = \frac{1}{\eta}\log \sum_{m=1}^{M} \frac{e^{\eta\gamma V'_m(s,a)}}{M} - \gamma \sum_{m=1}^{M} \frac{V'_m(s,a)}{M}. \tag{7}$$

Noticing that $\sum_{i=1}^{N} e^{\eta x_i}/N$ is the *sample moment generator* w.r.t. samples $\{x_i\}_{i=1}^{N}$ we can rephrase the exploration bonus as

$$\begin{aligned} U(s,a) \;&=\; \lim_{N\to+\infty} \frac{1}{\eta}\log\left[1 + \sum_{n=2}^{N} \frac{(\eta\gamma)^n}{n!}\mathcal{M}_n(s,a)\right] \\ &=\; \eta\gamma\mathcal{M}_2(s,a) + O(\eta^2) \end{aligned} \tag{8}$$

where $\mathcal{M}_n$ is the n^{th} central moment of the random variable V'_m (Proof in technical report [40])

$$\mathcal{M}_n(s,a) = M^{-1}\sum_{m=1}^{M}\left[\Big(V'_m(s,a) - \overline{V}(s,a)\Big)^n\right]$$

with

$$\overline{V}(s,a) = M^{-1}\sum_{m=1}^{M} V'_m(s,a).$$

Equation (6) shows that OBE is equivalent to BE with an additional bonus defined by Equation (8). The bonus U (for any positive η) is always positive, and provides a measure of the uncertainty w.r.t. Q. This is why OBE can be interpreted as a special principled form of IM.

d) Explicit Exploration: A general problem affecting intrinsically motivated algorithms, is that the policy greedy to the obtained Q-value function, is not optimized for the original problem. As a solution to this issue we approximate two functions: $\tilde{Q}$, which will be updated using the true reward and Q_E which will be updated using only the intrinsic reward [39]. In this way we obtain both the intrinsically motivated policy $\pi_o(s) = \arg\max_a \tilde{Q}(s,a) + Q_E(s,a)$ and the classic policy $\pi_u(s) = \arg\max_a \tilde{Q}(s,a)$. Define

$$\tilde{Q}_i(s,a) = R(s,a) + \gamma \sum_{m=1}^{M} \frac{\tilde{V}'_m(s,a)}{M} \quad \text{with} \tag{9}$$

$$\tilde{V}'_m(s,a) = \sum_{s'} P(s'|s,a)\max_{a'} \tilde{Q}_m(s',a') \tag{10}$$

to obtain an unbiased estimate of the Q-value function, yielding

$$\begin{aligned} Q_E(s,a) &= \sum_{t=0}^{T} \gamma^t U(s_t,a_t) \quad \text{where} \quad s_0 = s, a_0 = a \\ &= \eta^{-1} \log \frac{\sum_{k=1}^{M} e^{\eta\gamma \max_{a'} \tilde{Q}_k(s',a') + Q_E(s',a')}}{M} \\ &\quad - \frac{\sum_{k=1}^{M} \gamma \max_{a'} \tilde{Q}_k(s',a')}{M}. \end{aligned} \tag{11}$$

By a simple equation rearrangement, it is possible to show that $\tilde{Q}_i(s,a) + Q_E(s,a)$ is equivalent to $Q_i(s,a)$ as defined in the OBE (4).

B. Optimistic Value Function Estimators

The OBE offers a theoretical framework in which it is possible to develop optimistic value based algorithms. In fact, OBE enjoys all the desirable properties of the BE (e.g. max-norm contractivity), as shown in the technical report [40]. We present briefly two practical applications of the OBE, an optimistic variant of Q-learning (OQL) and deep Q-network (ODQN).

a) Optimistic Q-Learning: Motivated by the idea of employing an ensemble of regressors as is done in boostrapped DQN (BDQN) [25], we assume to have M randomly initialized Q-tables. Inspired by the well known Q-learning update rule, we derive an optimistic version which is consistent with the OBE.

Definition 1 (Optimistic Q-learning)**.** [2]

$$\begin{aligned} Q_{i,t+1}(s,a) &= (1-\alpha_t) Q_{i,t}(s,a) \\ &\quad + \alpha_t \Big(r_t + \frac{1}{\eta} \log M^{-1} \sum_{j=1}^{M} e^{\gamma \max_{a'} Qj,t(s',a')} \Big). \end{aligned}$$

We show that, with the update rule proposed, given infinite visits of each state-action pair, all the tables will converge to the same values, and more precisely, after each update, the n^{th} central moment of the updated cell is scaled exactly by $(1-\alpha_t)^n$:

$$\mathcal{M}_{n,t+1}(s,a) = (1-\alpha_t)^n \mathcal{M}_{n,t}(s,a) \tag{12}$$

where

$$\mathcal{M}_{n,t}(s,a) = M^{-1} \sum_{i=1}^{M} \left(Q_{i,t}(s,a) - \sum_{k=1}^{M} \frac{Q_{k,t}(s,a)}{M} \right)^n.$$

This implies that a cell updated N times, with learning rates $\{\alpha_i\}$, will have the n^{th} central moments scaled by $\Pi_{\alpha_i}(1-\alpha_i)^n$ w.r.t. the initial one. This leads us to some interesting considerations: 1) the bonus decrease accordingly to the number of state visits; 2) differing from several count-based approaches, our algorithm takes into account the impact of the learning rate; 3) in the limit of an infinite number of visits, the exploration bonus converges to zero. Further details, including a proof of convergence, are given in the technical report [40][3]. All the considerations done so far provide a deeper insight about how the algorithm works and its properties. However, in a more complex settings, (e.g., function approximation) the convergence to zero of the exploratory bonus is not guaranteed in general.

b) Optimistic DQN: In addition to the novel OQL algorithm described previously that can be used for limited discrete state spaces, we propose another algorithm for continuous state spaces based on our OBE. We take inspiration from the framework provided by BDQN [25] that uses an ensemble of neural networks as estimator for the Q value function. BDQN minimizes the loss

$$\mathcal{L}_B(s,a) = \sum_{k=1}^{M} \left(r + \gamma \max_{a'} Q_k^T(s',a') - Q_k(s,a) \right)^2,$$

where Q_k^T is the target network of the k^{th} approximator. To get an unbiased performance evaluation, we decided to update $M-1$ components of the ensemble with the update rule provided by BDQN. We make this choice in order to maintain diversity between the approximations of the ensemble as shown in [25]. We use the remaining single component of the ensemble to approximate Q_E. Using the first component to approximate Q_E, we get for our new algorithm optimistic DQN (ODQN) the loss

$$\begin{aligned} \mathcal{L}_O(s,a) = &\left(\eta^{-1} \log \frac{\sum_{k=2}^{M} e^{\eta\gamma \max_{a'} Q_k^T(s',a') + Q_1^T(s',a')}}{M} \right. \\ &\left. - \frac{\sum_{k=2}^{M} \gamma \max_{a'} Q_k^T(s',a')}{M} - Q_1(s,a) \right)^2 \\ &+ \sum_{k=2}^{M} \left(r + \gamma \max_{a'} Q_k^T(s',a') - Q_k(s,a) \right)^2. \end{aligned} \tag{13}$$

The exploratory bonus represented by $Q_E = Q_1$ in the proposed OQL and OQDN algorithms is needed to guide exploration during learning. During evaluation, we use majority voting on the remaining $M-1$ components $\{Q_k\}_{k=2}^{M}$. While we always select an optimistic policy in OQL during the training phase, in ODQN the neural network function approximator may have problems learning to approximate the optimal policy: if there are not enough unbiased samples the approximator may learn to model only the optimistic biased samples. Note that in the tabular case, this is not a problem since there is no Q-function approximation. In order to mitigate this problem, we introduce a hyper-parameter χ which denotes the probability to select an optimistic policy π_o in place of the unbiased one π_u. In this way, we can balance the number of unbiased and optimistic samples. Algorithm 1 shows the pseudocode of ODQN.

c) Automatic Hyper-parameter Adaptation: Recalling that the regularization coefficient η in the OBE is hard to tune, we want to focus our attention on Problem 2. Inspired by proximal policy optimization (PPO) [33], we propose an heuristic to optimize η. One of the optimization techniques

[2] We use α_t as a shortcut for $\alpha_t(s,a)$.

[3] We based our convergence proof for OQL on [18] and [12]

Algorithm 1 Optimistic DQN

Input: $\{Q_k\}_{k=1}^K$, $\iota_{\max}$, η_{init}, χ, N, C
Let B be a replay buffer storing the experience for training.
$\eta = \eta_{\text{init}}$.
Let $i \sim \text{Uniform}\{1 \dots M\}$ and $\psi = 1$ w.p. χ otherwise $\psi = 0$
for N epochs **do**
  for C steps **do**
    Observe s
    Choose $a = \arg\max_a Q_i(s,a) + \psi Q_1(s,a)$
    Observe reward r, next state s', end of episode t
    If t is terminal, $i \sim \text{Uniform}\{2 \dots M\}$ and
      $\psi = 1$ w.p. χ otherwise $\psi = 0$
    Store $< s, a, r, s', t >$ in buffer B
    Sample mini-batch B_{batch}
    Update $\{Q_k\}_{k=1}^K$ using equation (13)
    $V \leftarrow V +$ | violated constraints (14) in B_{batch}|
  end for
  Let $\rho = \frac{V}{C * \text{batch\_size}}$
  Update η by (15)
  Update target network
end for

proposed in [33] is to measure the "degree" of constraint violation and to update the Lagrangian multiplier accordingly. We have to adapt the technique to multiple constraints since the problem is defined for each state-action pair. The idea is to count the number of times the constraints have been violated and then update η. In more detail, suppose to have N state-action pairs and for each pair (s_i, a_i)

$$\sum_m b_m(s_i,a_i)(\log b_m(s_i,a_i) + \log M) \leq \iota_{\max}, \tag{14}$$

where $\iota_{\max}$ is defined in Problem 2, while $b_m(s_i,a_i)$ is defined by (3). We define ρ as the ratio of violated constraints. We update η according to the following rule

$$\eta_{T+1} = \frac{\eta_T}{(0.5 + 10\rho)}. \tag{15}$$

In ODQN, we decided to count the number of constraints violated every C time-steps (basically every update of the target network), using the samples of all the extracted mini-batches. See Algorithm 1 for further details.

d) Ensuring a Prior Distribution: As already discussed, it is important to maintain diversity in our ensemble, and this diversity should reflect the degree of uncertainty. For this reason, we should introduce a sort of prior distribution, as happens in the Bayesian framework. In the case of OQL, we observe that it is sufficient to randomly initialize each element of the ensemble, since diversity between estimates is a sufficient condition to obtain positive bonus. For ODQN, as is done in BDQN, we choose to maintain the diversity between approximation, by a random initialization of each component's parameters and by using the bootstrapping technique, so by adding a *mask* in the replay memory which is sampled by and use different data samples per regressor.

III. Experimental Evaluation

In the experiments, we compare in the tabular Q-function case our new optimistic Q-learning method (OQL) with bootstrapped Q-learning method (BQL) - which is the tabular version of BDQN, the well-known state-of-the-art Q-learning (QL) [42], and Q-learning with optimistic initialization (OIQL) [37] in the 50-Chain [25], Taxi (also known as Maze) [7] and Frozen Lake [5] environments. For neural-network based Q-functions, we compare our new optimistic deep Q-learning (ODQN) method with bootstrapped deep Q-learning (BDQN) and classical deep Q-learning (DQN) in the Taxi and Acrobot [36] environments. The environments are chosen to cover different types of dynamics, have sparse rewards, and include both discrete and continuous states. For Acrobot and Frozen Lake, we used the implementation provided by OpenAI Gym [5]. First, we will discuss the environments in more detail, then we will provide some details on the initialization of the methods, and finally finish with an analysis of the results.

The **N-Chain** environment [25] requires a long sequence of non-rewarding actions to achieve the optimal reward. The MDP consists of a chain with N states $\{s_i\}_{i=1}^N$, and two actions that move the agent to state s_{i+1} or s_{i-1}. The agent always starts in state s_2. In state s_1 the agent observes a small reward $r(s_1) = 1/1000$, while in the N^{th} state the agent observes $r(s_N) = 1$. The reward function is zero elsewhere. The agent needs to explore until reaching state s_N even if state s_1 looks promising.

The **Taxi** environment, also known as Maze [7], consists of a 8×8 grid-world where a taxi has to collect passengers and take them to the goal position where the only non-null reward is observed. For 0, 1, 2 and 3 passengers collected, the reward is 0, 1, 3 and 15, respectively. The agent must explore to 1) discover that reaching the goal position with less than 3 passengers is not optimal, 2) to find the optimal path.

The **Frozen Lake** environment is a 8×8 grid-world, in which the agent has to reach a goal position without falling into some holes. The stochastic perturbation of the agent's movement makes the environment challenging.

The **Acrobot** environment was firstly proposed by [36] and consists of two linked robotic arms that hang downward with only one actuated joint between the two arms. The agent swings the robotic arms until the end of the second link exceeds a certain height. At each step the agent perceives a negative reward of -1. The environment requires significant exploration to find a way to swing the robotic arms at defined height.

a) Initialization of the Q-functions: In the tabular setting, for optimistically initialized Q-learning (OIQL), we initialize the Q-function to 15 in Taxi and to 1 in N-Chain and Frozen Lake. For the other algorithms, we initialize $Q(s,a) \sim \mathcal{N}(\mu = 0, \sigma = 2)$, except Q_E of OQL is initialized to 0. In the taxi environment, for both ODQN and BDQN, we use a shared convolutional layer with multiple heads as described in [25]. For the Acrobot, each component of the ensemble

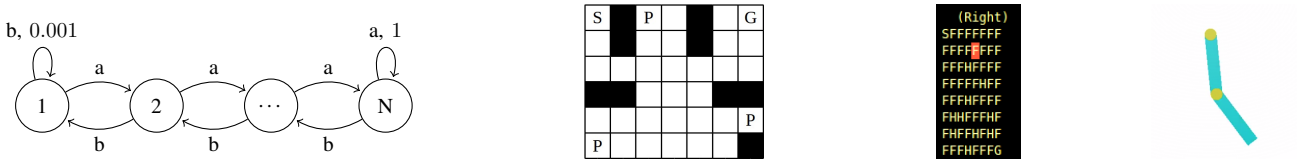

Fig. 2: Illustration of the environments, from the left to the right: N-Chain, Taxi, Frozen Lake and Acrobot.

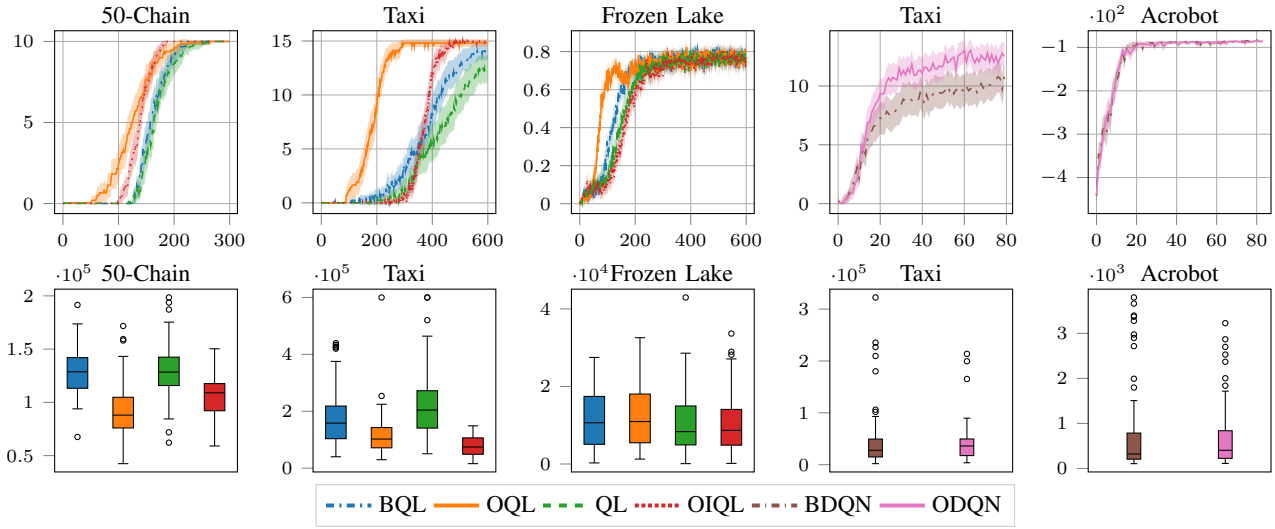

Fig. 3: The first row shows the average return for each tabular algorithm in the N-Chain, Taxi, and Frozen Lake environments and for each neural network based algorithm in the Taxi and Acrobot environments together with 95% confidence intervals. The second row shows the distribution over the number of time steps before observing the maximum reward in the MDP.

corresponds to a one-layer neural network. For ODQN, in both the environment, we initialize the output layer corresponding to Q_E to small values of the parameters, in order to obtain initially $Q_E \approx 0$.

b) Hyper-Parameter Tuning: For the tabular settings, we did not run any hyper-parameter optimization. With neural networks we performed a small grid search over the number of neurons of the network, and whether to use bootstrapping or not. More specifically, we selected hyper-parameters maximizing the mean return averaged over the whole learning curve, using 20 different seeds. In the plots; we compare ODQN and BDQN using the best hyper-parameter setup found for BDQN. In OQL we use $\eta = 10$ and for ODQN, we use $\chi = 0.25$ and $\iota_{\max} = 1$. For further details about the implementation of the algorithms, and the grid search, please see the technical report [40].

A. Results

Figure 3 summarizes the results obtained by averaging over 64 different seed the tabular alrogithms, and 100 seeds BDQN and ODQN (more details, e.g. results for different hyper-parameter settings, can be found in the technical report [40]). OQL learns faster than the other tabular algorithms. In the Chain environment, and in Taxi, our algorithm OQL finds the highest reward faster than BQL or QL. On the other hand, also OIQL seems to find high rewards fast, as shown in the box-plots. However, OIQL requires more training epochs to escape the high initial optimistic values of the value function. In contrast, OQL finds high values fast in all the problems (50-Chain, Taxi, Frozen Lake) using the optimistic Bellman equation while converging to a near-optimal solution as suggested by our convergence proofs.

With neural network approximation of the value function, ODQN outperforms BDQN in the Taxi environment demonstrating that the principles of OBE work even with function approximation. In the Acrobot environment, the learning curves of BDQN and ODQN are nearly identical, possibly, due to the simplicity of the environment.

IV. Conclusion

The main contribution of our work is the introduction of the optimistic Bellman equation (OBE) which provides an optimistic estimate of the value function over uncertainty. Our approach can be viewed as a principled IM technique where the agent is intrinsically rewarded by uncertainty and which, similar to approximated Bayesian methods, estimates the an ensemble. We propose two algorithms: OQL for the tabular case and ODQN for the neural network case. Given the usual assumptions on the learning rate and state visits, we show that OQL convergences to the optimal policy, analyze the implicitly defined exploration bonus in OQL and show the relationship to intrinsic motivation based approaches. In empirical evaluations on a variety of tasks where exploration is crucial, OQL and

OQDN outperform comparison methods due to being able to find high rewards earlier.

Acknowledgments: The research is financially supported from the Bosch-Forschungsstiftung program and from NVIDIA.

REFERENCES

[1] K. Asadi and M. L. Littman. An Alternative Softmax Operator for Reinforcement Learning. In *International Conference on Machine Learning*, pages 243–252, 2017.

[2] K. Azizzadenesheli, E. Brunskill, and A. Anandkumar. Efficient Exploration through Bayesian Deep Q-Networks. *Asteroids*, 2517(1516):108.

[3] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[4] R. I. Brafman and M. Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] R. Y. Chen, J. Schulman, P. Abbeel, and S. Sidor. UCB and InfoGain Exploration via Q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

[7] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998.

[8] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208. ACM, 2005.

[9] E. Even-Dar and Y. Mansour. Convergence of optimistic and incremental Q-learning. In *Advances in neural information processing systems*, pages 1499–1506, 2002.

[10] R. Grande, T. Walsh, and J. How. Sample efficient reinforcement learning with gaussian processes. In *International Conference on Machine Learning*, pages 1332–1340, 2014.

[11] R. A. Howard and J. E. Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

[12] T. Jaakkola, M. I. Jordan, and S. P. Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710, 1994.

[13] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

[14] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[15] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[16] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[17] S. I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernandez, S. Coraluppi, and P. Fard. Risk sensitive markov decision processes. In *Systems and control in the twenty-first century*, pages 263–279. Springer, 1997.

[18] F. S. Melo. Convergence of Q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[19] N. Meuleau and P. Bourgine. Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning*, 35(2):117–154, 1999.

[20] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[22] G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[23] B. O'Donoghue, I. Osband, R. Munos, and V. Mnih. The uncertainty Bellman equation and exploration. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3839–3848, 2018.

[24] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[25] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

[26] I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

[27] G. Ostrovski, M. G. Bellemare, A. v. d. Oord, and R. Munos. Count-based exploration with neural density models. *arXiv preprint arXiv:1703.01310*, 2017.

[28] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

[29] J. Peters, K. Mülling, and Y. Altun. Relative Entropy Policy Search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

[30] A. Ruszczyński. Risk-averse dynamic programming for markov decision processes. *Mathematical programming*, 125(2):235–261, 2010.

[31] J. Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In *Workshop on Anticipatory Behavior in Adaptive Learning Systems*, pages 48–76. Springer, 2008.

[32] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] S. P. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2004.

[35] M. Strens. A Bayesian framework for reinforcement learning. In *ICML*, pages 943–950, 2000.

[36] R. S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in neural information processing systems*, pages 1038–1044, 1996.

[37] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[38] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[39] I. Szita and A. Lőrincz. The many faces of optimism: a unifying approach. In *Proceedings of the 25th international conference on Machine learning*, pages 1048–1055. ACM, 2008.

[40] S. Tosatto, C. D'Eramo, J. Pajarinen, M. Restelli, and J. Peters. Technical Report: "Exploration Driven by an Optimistic Bellman Equation". https://www.ias.informatik.tu-darmstadt.de/uploads/Team/SamueleTosatto/tosatto1028tr.pdf, 2018.

[41] N. Vlassis, M. Ghavamzadeh, S. Mannor, and P. Poupart. Bayesian reinforcement learning. In *Reinforcement Learning*, pages 359–386. Springer, 2012.

[42] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[43] M. White and A. White. Interval estimation for reinforcement-learning algorithms in continuous-state domains. In *Advances in Neural Information Processing Systems*, pages 2433–2441, 2010.