Bellmanian Bandit Network

Antoine Bureau, Michèle Sebag

Bellmanian Bandit Network

Antoine Bureau
TAO, LRI - INRIA
Univ. Paris-Sud
bldg 650, Rue Noetzlin,
91190 Gif-sur-Yvette, France
antoine.bureau@lri.fr

Michèle Sebag
TAO, LRI - CNRS
Univ. Paris-Sud
bldg 650, Rue Noetzlin,
91190 Gif-sur-Yvette, France
sebag@lri.fr

Abstract

This paper presents a new reinforcement learning (RL) algorithm called Bellmanian Bandit Network (BBN), where action selection in each state is formalized as a multi-armed bandit problem. The first contribution lies in the definition of an exploratory reward inspired from the intrinsic motivation criterion [1], combined with the RL reward. The second contribution is to use a network of multi-armed bandits to achieve the convergence toward the optimal Q-value function. The BBN algorithm is validated in stationary and non-stationary grid-world environments, comparatively to [1].

1 Introduction

Reinforcement learning (RL) aims at finding optimal policies, maximizing the (discounted) expected cumulative reward gathered by the learning agent along its trajectory [2]. RL involves three interdependent tasks: 1. modelling the world (transition and reward function) on the basis of the available evidence; 2. building an optimal policy on the basis of the current models; 3. exploring the world and gathering further evidence to revise and improve these models. The trade-off between tasks 2 and 3, referred to as exploitation vs exploration dilemma (EvE), has been extensively studied in the RL literature (see e.g. [3, 4, 5]).

This paper tackles the EvE dilemma through the definition of an adaptive exploration-related reward; the approach is inspired from intrinsic motivation criteria [6, 1], originally designed to enforce autonomous behaviors in *in-situ* robotics [7, 8, 5]. The proposed algorithm, called Bellmanian Bandit Network (BBN), tackles action selection in each state as a multi-armed bandit problem [9, 10], where the instant reward aggregates the current exploration and exploitation rewards.

Section 2 introduces the formal background. Section 3 presents and discusses intrinsic motivation [1] for the sake of completeness. Section 4 gives an overview of BBN; its experimental validation, in particular considering the case of non-stationary environments, is discussed in section 5. Section 6 concludes with a discussion and some research perspectives.

2 Formal Background

Markov Decision Processes (MDP). A Markov decision process $\mathbb{M}$, modelling a sequential decision problem in discrete time, is a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $\mathcal{T}$ is the transition model with $\mathcal{T}(s, a, s')$ the probability of reaching state s' upon selecting action a in state s, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. The goal is to find a policy $\pi : S \to A$ such that the expected cumulative reward gathered following this policy is maximal. Let

us define the value function ($V^\pi : \mathcal{S} \mapsto \mathbb{R}$) associated to policy π as

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)), s_0 = s, s_{t+1} \sim T(s_t, \pi(s_t), \cdot)\right]$$

with s_t and $a_t = \pi(s_t)$ respectively the state and the selected action at time t, and $\gamma < 1$ a discounting factor. The quality function Q^π defined on state-action pairs ($Q^\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$) is likewise defined as:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)), s_0 = s, a_0 = a, a_t = \pi(s_t), s_{t+1} \sim T(s_t, \pi(s_t), \cdot)\right] \quad (1)$$

The optimal value function V^* is the max over all policies π of value functions V^π, and the optimal policy π^* is obtained by selecting greedily in each state s, the action a leading to the optimal expected value of the next state: $\pi^*(s) = \text{argmax}_a\, Q^*(s,a)$. RL algorithms most usually involve a learning phase, where the optimal value function and policy π^* are learned, followed by a production phase, using policy π^* ever after.

Multi-Armed Bandit (MAB). On the contrary, the Multi-Armed Bandit (MAB) setting achieves lifelong learning: it aims at maximizing the cumulative reward gathered while learning. A MAB problem can be viewed as a single state, K-action MDP, where the i-th action brings rewards according to stationary distribution $R_i, 1 \leq i \leq K$. Letting $n_{i,t}$ (respectively $\overline{r_{i,t}}$) denote the number of times the i-th action was selected (resp. the empirical average reward gathered when selecting the i-th action) up to time t, the Upper Confidence Bound (UCB) action criterion selects the action i_t^* s.t.:

$$i_t^* = argmax_{i \in 1 \ldots K} \left(\overline{r_{i,t}} + C\sqrt{\frac{\log t}{n_{i,t}}}\right) \quad (2)$$

with C a tradeoff parameter.

3 Intrinsic motivation criterion

For the sake of completeness, let us describe the intrinsic motivation criterion and how it is leveraged to enforce an adaptive exploration in the RL framework [1]. When wandering in an environment, an autonomous robot can evaluate for free the accuracy of any predictive model, a.k.a. forward model, predicting the next state visited depending on the current state and action. Letting h denote its current model, the robot can predict the state $\widehat{s_{t+1}} = h(s_t, a_t)$ and see in the next time step whether the prediction was accurate ($\widehat{s_{t+1}} = s_{t+1}$). The accuracy of the forward model cannot be taken directly as an exploratory reward, since an optimal opportunistic behavior would be to stay in the same state forever. The intrinsic motivation criterion thus uses the *reduction of the forward model error* as exploratory reward. This reward has been plugged in two model-based RL algorithms: R-Max [3] and Bayesian Exploration Bonus (BEB) [4], to enforce exploration.

In the initial $R-Max$ algorithm, reward $R(s,a)$ was set to the maximal reward R_{max} until state-action pair (s,a) has been visited a sufficient number m of times. Such an optimistic evaluation of unknown state-action pairs enforces a systematic exploration of the environment. Formally, the initial reward function in $R-Max$ is defined as:

$$R^{R-Max}(s,a) = \begin{cases} R(s,a) & \text{if } n(s,a) \geq m \\ R_{max} & \text{otherwise} \end{cases} \quad (3)$$

where $n(s,a)$ is the number of times the action a was selected in the state s and $R(s,a)$ is the standard MDP reward. As argued by Lopes et al. [1], the above scheme is sensitive to hyper-parameter m, and relies on the knowledge of the maximal reward. Additionally, it is ill-suited to non-stationary environments: after a state-action pair has been visited m times, there is no more exploration bonus. Accordingly, Lopes et al. proposed to modify the R-Max reward as follows:

$$R^{\zeta\text{-R-Max}}(s,a) = \begin{cases} R(s,a) & \text{if } \zeta(s,a) < m \\ R_{max} & \text{otherwise} \end{cases} \quad (4)$$

where function ζ measures the increase of the prediction model accuracy provided by the last k trajectory steps $(s_t, a_t, s_{t-1}, a_{t-1}, \ldots s_{t-k})$, plus the error variance measured after a leave-one-out validation procedure.

Discussion. Whereas the $R-Max$ exploratory bonus only depends on the number of visits (with a high sensitivity w.r.t. parameter m), the intrinsic exploratory bonus mostly depends on the transition model; typically, regions with a highly stochastic transition model (where the prediction model error tends to decrease slowly) will be more intensively visited than deterministic regions (where the prediction model error decreases much faster).
The intrinsic exploratory bonus might still be ill-suited to non-stationary environments, as the visited regions are increasingly restricted to the neighborhood of optimal paths as time goes on. Specifically, the exploration bonus is measured over a fixed time-window, involving a limited look-ahead of the exploration task. The problem of limited look-ahead in RL has been thoroughly investigated, as exemplified by e.g. the artificial RiverSwim benchmark [11].

4 Bellmanian Bandits Network

After a general overview of the Bellmanian Bandit Network framework, this section details the hybrid EvE reward used to control the action selection in each node state.

Principle. The BBN algorithm involves a network of multi-armed bandits, where a multi-armed bandit is attached to each state s. These bandits are structured along a directed graph, with an edge from state/bandit s to s' iff there is an action leading from s to s'. For each state-action pair (s,a), BBN maintains an internal and an external cumulative rewards, where the internal cumulative reward $\overline{cr^{(i)}(s,a)}$ measures the cumulative exploration effectiveness attached to (s,a) (see below), and the external cumulative reward $\overline{cr^{(e)}(s,a)}$ measures the MDP cumulative reward gathered for selecting action a in state s. The quality of each state-action pair is defined as:

$$Q_t(s,a) = \overline{cr^{(i)}(s,a)} + \overline{cr^{(e)}(s,a)} + \gamma \sum_{s'} p_{s,a,s'} V_t(s') \text{ with } V_t(s') = max_{a'} Q_{t-1}(s',a')$$

The action selection rule in each state is the UCB action selection rule, with:

$$\pi_t(s) = \operatorname*{argmax}_a \left(Q_t(s,a) + C\sqrt{\frac{\log n_t(s)}{n_t(s,a)}} \right) \tag{5}$$

where $n_t(s)$ and $n_t(s,a)$ respectively denote the number of times state s has been visited, and the number of times a has been selected in s, up to time t.
To account for non-stationary environments, quality function $Q_{t,ns}$ is defined as:

$$Q_{t,ns}(s,a) = r^{(i)}(s,a) + r^{(e)}(s,a) + Q_{t-\Delta t}(s,a) \times \delta^{\Delta t}$$

with $\delta < 1$ a discounting factor and Δt the number of time steps since state-action pair (s,a) has last been visited, and $r^{(i)}(s,a)$ and $r^{(e)}(s,a)$ the instant external and internal rewards gathered the last time (s,a) has last been visited. With $n_{t,ns}(s,a) = n_t(s,a) \times \delta^{\Delta t}$, the non-stationary action selection rule is defined as:

$$\pi_{t,ns}(s) = \operatorname*{argmax}_a \left(Q_{t,ns}(s,a) + C\sqrt{\frac{\log n_t(s)}{n_{t,ns}(s,a)}} \right) \tag{6}$$

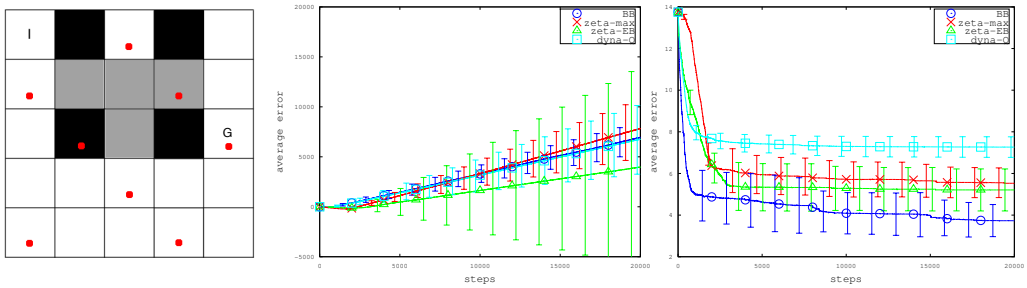
The rationale for the $Q_{t,ns}$ quality function is to allow for adaptively relaunching exploration.

Internal reward. The proposed internal reward is inspired from the intrinsic motivation [1] albeit with a lesser computational complexity. Formally, for each state-action pair (s,a) is maintained the list of states s' reached upon executing a in state s, and the number $n_t(s,a,s')$ state s' was reached upon executing a in state s, up to time t. Instead of considering the accuracy of the forward model, BBN thus computes the entropy attached to (s,a):

$$Entropy_t(s,a) = -\sum_{s'} p_{s,a,s'} \log(p_{s,a,s'}) \qquad \text{with } p(s,a,s') = \frac{n_t(s,a,s')}{n_t(s,a)} \tag{7}$$

The internal reward gathered upon visiting state-action (s,a) at time t is finally defined as:

$$r^{(i)}(s,a) = Entropy_t(s,a) - Entropy_{t-1}(s,a)$$



(a) Environment (b) cumulative reward (c) average error

Figure 1: Comparative performance of BBN in stationary environments (averaged on 20 runs). (a) The grid world. (b) Cumulative reward vs number of time steps. (c) Average transition error in norm L_1.

Discussion. The internal reward preserves the spirit of the intrinsic motivation as $r^{(i)}(s,a)$ goes to 0 when (s,a) has been "sufficiently visited" (in the sense of the convergence of $Entropy_t(s,a)$); the difference lies in the computational simplicity of the internal reward. The consistency of the UCB algorithm thus enforces the selection of the optimal action a^* in state s, according to the Q value function. The main weakness of BBN is the computational complexity, quadratic in the size of the state space in the worst case. However, under the realistic assumption that the branching factor (the number of states s' actually observed after selecting an action a in a state s) is bounded, the computational and memory complexity of the Bellmanian Bandit Network is linear in the size of the state and action space.

5 Empirical study

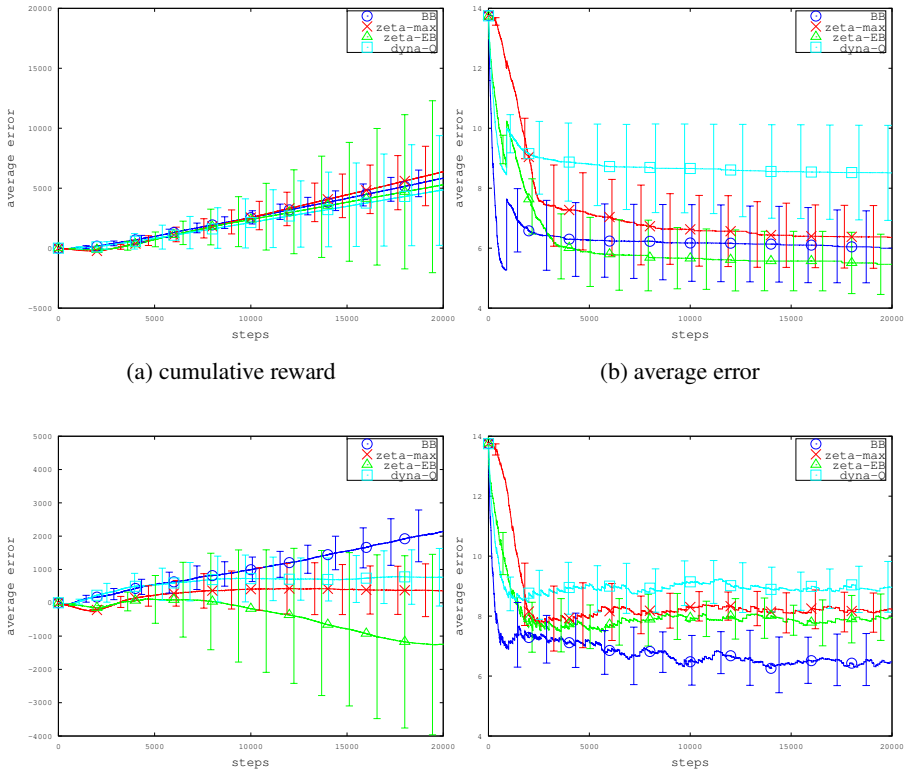
BBN is experimentally validated, comparatively to $\zeta R-max$ and ζEB policies [1] and the baseline ϵ-greedy policy. The performance indicators are the cumulative discounted reward, and the accuracy of the learned forward model. The goal of the experiments is to study the robustness of BBN w.r.t. the algorithm hyper-parameters and w.r.t. non-stationary environments.

Experimental setting. Same stochastic gridworld is considered as in [1] (Fig. 1a), with a 25-states space and a 5-actions space (left, right, up, down, stay). The transition probabilities for each state are drawn from a Dirichlet distribution with parameter α. For low-noise states $\alpha = 0.1$ (depicted in white on Fig 1a); for noisy states $\alpha = 1.0$ (depicted in black), corresponding to a uniform transition model. The instant reward is 1 in the single goal state (legend G), -1 in all gray states, and 0 otherwise. An episodic setting is considered, with time horizon 30, and the agent restarts from initial state I in each episode. The accuracy of the forward model is the distance between the true and the empirical transition matrices associated to each state-action pair. The BBN hyper-parameters are: $\delta = 0.98$, $\gamma = 0.95$, $C = 5.0$. Same hyper-parameter setting as in [1] is used for $\zeta R-max$ and ζEB: $K = 10$, $m = 0.9$.

The robustness with respect to non-stationary environments is investigated by considering: i) a stationary environment; ii) a single change of the transition model, uniformly swapping the transition distributions associated to the gray states at time $t = 900$; iii) repeated changes of the transition model every 200 time steps.

Stationary environment. As shown in Fig. 1, in terms of cumulative reward, BBN is slightly outperformed by only $\zeta R-max$ after 1,000 steps, which suggests that BBN is slightly biased toward exploration. In terms of precision of the forward model, BBN outperforms all other algorithms.

Non-stationary environments. In the first experiment, the transition probabilities of the states marked with a red dot are swapped after 900 time steps. As shown in Fig. 2 (top row), BBN outperforms ζEB and is outperformed by $\zeta R-max$ in terms of cumulative reward, while it recovers



(a) cumulative reward (b) average error

(c) cumulative reward (d) average error

Figure 2: Comparative performance of BBN in non-stationary environments, averaged over 20 runs. Top row: single change at time step 900. Bottom row: change every 200 time steps. Left: Cumulative reward *versus* number of time steps. Right: Average transition error in norm L_1.

slightly more slowly than ζEB, though faster than $\zeta R - max$ in terms of transition error. These results suggest that BBN and ζEB are biased toward exploration.

In the second experiment, every 200 time steps the transition probabilities of two uniformly selected red states (Fig. 1.a) are swapped. As shown in Fig. 2 (bottom row), BBN outperforms all other algorithms in terms of cumulative reward and transition error, with a significantly lower variance.

6 Discussion and perspectives

The Bellmanian Bandit Network algorithm presented in this paper proposes a new exploration-exploitation trade-off in RL, by modelling the sought policy as a network of Multi-Armed Bandits. The contribution is twofold. Firstly, the reward associated to each state-action pair captures i) the instant reward from the MDP setting; ii) the exploratory reward inspired from the intrinsic motivation [1]; iii) the expected value flowing from the destination nodes, inspired from the Bellman equation. Secondly, a proof of concept in simple stationary and non-stationary grid-worlds shows the robustness of BBN in terms of cumulative rewards and in terms of transition model estimation. Furthermore, BBN has linear complexity in the size of the action and state space, conditionally to a limited branching factor of the transition model.

This work opens several research perspectives. The first one concerns the theoretical analysis of the proposed scheme, building upon the consistency of the MAB setting. The use of the KL-UCB *in lieu* of the UCB criterion [12] will be considered, in a theoretical and experimental perspective. The automatic adjustment of the scale of the internal reward, compared to the external reward, will be studied. Additionally, the BBN extension to continuous state and action spaces, taking inspiration

from the double progressive widening [13], will be investigated and applied to model-free robotic settings.

References

[1] Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Neural Information Processing System*, 2012.

[2] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning, an introduction*. A Bradford Book, 1998.

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

[4] J. Zico Kolter and Andrew Ng. Near-bayesian exploration in polynomial time. In *Proceeding of the International Conference on Machine Learning (ICML)*, pages 513–520, 2009.

[5] Shiau Hong Lim and Peter Auer. Autonomous exploration for navigating in mdps. *JMLR: Workshop and Conference Proceedings*, 2012.

[6] Jürgen Schmidhuber. Curious model-building control systems. In *Proceeding of the International Joint Conference on Neural Networks*, pages 1458–1463, 1991.

[7] Pierre-Yves Oudeyer, Frédéric Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. In *IEEE Transactions On Evolutionary Computation*, 2007.

[8] Adrien Baranès and Pierre-Yves Oudeyer. R-iac : Robust intrinsically motivated exploration and active learning. In *IEEE Transactions on Autonomous Mental Development*, 2009.

[9] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. In *International Conference on Machine Learning*, 2002.

[10] Levente Kocsis and Csaba Szepesvari. Bandit based monte-carlo planning. In *European Conference on Machine Learning*, 2006.

[11] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision. In *Journal of Computer and System Sciences*, 2008.

[12] Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoutly. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergence. In *COLT*, 2011.

[13] David Auger, Adrien Couëtoux, and Olivier Teytaud. Continuous upper confidence trees with polynomial exploration - consistency. In *ECML/PKKD*, 2013.