Theory Biosci. (2012) 131:139–148
DOI 10.1007/s12064-011-0142-z

ORIGINAL PAPER

An information-theoretic approach to curiosity-driven reinforcement learning

Susanne Still · Doina Precup

Received: 19 June 2010 / Accepted: 5 April 2011 / Published online: 12 July 2012

Abstract We provide a fresh look at the problem of exploration in reinforcement learning, drawing on ideas from information theory. First, we show that Boltzmann-style exploration, one of the main exploration methods used in reinforcement learning, is optimal from an information-theoretic point of view, in that it optimally trades expected return for the coding cost of the policy. Second, we address the problem of curiosity-driven learning. We propose that, in addition to maximizing the expected return, a learner should choose a policy that also maximizes the learner's predictive power. This makes the world both interesting and exploitable. Optimal policies then have the form of Boltzmann-style exploration with a bonus, containing a novel exploration–exploitation trade-off which emerges naturally from the proposed optimization principle. Importantly, this exploration–exploitation trade-off persists in the optimal deterministic policy, i.e., when there is no exploration due to randomness. As a result, exploration is understood as an emerging behavior that optimizes information gain, rather than being modeled as pure randomization of action choices.

Keywords Reinforcement learning · Exploration–exploitation trade-off · Information theory · Rate distortion theory · Curiosity · Adaptive behavior

S. Still (✉)
Information and Computer Sciences,
University of Hawaii at Mānoa, Honolulu, HI 96822, USA
e-mail: sstill@hawaii.edu

D. Precup
School of Computer Science, McGill University,
Montreal, QC, Canada
e-mail: dprecup@cs.mcgill.ca

Motivation

The problem of optimal decision making under uncertainty is crucial to both animals and artificial intelligent agents. Reinforcement learning (RL) addresses this problem by proposing that agents should choose actions that maximize an expected long-term return provided by the environment (Sutton and Barto 1998). To achieve this goal, an agent has to *explore* its environment, while at the same time *exploiting* the knowledge it currently has in order to achieve good returns. In many existing algorithms, this trade-off is achieved mainly through simple randomization of the action choices. Practical implementations rely heavily on heuristics, only few theoretically principled approaches exist (see Sect. 5 for a more detailed discussion). In this article, we look at the exploration–exploitation trade-off from a fresh perspective: we use information-theoretic methods to analyze an existing exploration method, and to propose a new one.

Recently, an information-theoretic framework for behavioral learning has been introduced by one of us (Still 2009), which we use here to tackle reward-driven behavioral learning. First, we propose an intuitive optimality criterion for exploration policies which includes both the reward received, as well as the complexity of the policy. Having a simple policy is not usually a stated goal in reinforcement learning, but it is desirable for bounded-rationality agents, and it is especially useful in the context of developmental agents, which should evolve increasingly complex strategies as they get more experience, and as their knowledge of the environment becomes more sophisticated. We show in Sect. 2 that the general solution of the proposed optimization problem is a Boltzmann-style exploration algorithm. This approach is closely related to rate distortion theory (Shannon 1948), which is based on the fact that approximating a true signal using a compressed representation will cause a loss, computed as the expected value of a distortion function. The choice of the distortion function implicitly provides the distinction of relevant and irrelevant features of the data. Here, the trade-off is between the return, on the one hand, and the average bit cost of the policy, on the other hand. This trade-off is controlled by a "temperature"-like parameter[1]. At high temperatures, simplicity is more important than return. As the temperature decreases, return becomes increasingly important and the policy goes to the optimal-return policy as the temperature goes to zero.

Animals often explore their environment not only to gather rewards, but also just for the sake of learning about it. Continuous learning is also useful in the context of reinforcement learning, because the reward structure of the environment may change over time, and the agent may need to adapt to this change. Hence, it is advantageous to know more about the environment than what is strictly necessary in order to maximize the long-term return under the current conditions. Similar arguments have been presented in Singh et al. (2005) and Still (2009) as well as in many papers on transfer of knowledge in reinforcement learning (see Taylor and Stone 2009, for a survey). In Sect. 3, we formalize this idea, building on previous work (Still 2009). We seek behavioral policies that maximize future return, while at the same time maximizing predictive power, which we measure by the information captured about the future. Our objective function also contains a term which ensures that the agent continues to prefer simple policies. This term penalizes behaviors for retaining more memory about the past than is necessary to predict the future and to maximize the expected reward. As a consequence, it ensures that undesirable repetitive behaviors are avoided. We show that the resulting optimal policy contains a trade-off between exploration and exploitation which emerges naturally from the optimization principle.

The article is structured as follows. In Sect. 2, we lay the information-theoretic foundation of exploration for a reinforcement learning agent, whose main goal is to optimize long-term returns. Next, we formulate the problem of curiosity-driven reinforcement learning and solve it using a similar principle, including the maximization of predictive power (Sect. 3). Finally, we discuss algorithmic implementation issues in Sect. 4, and close with a discussion of the relationship of our approach to classical and current work in RL in Sect. 5.

Information-theoretic approach to exploration in reinforcement learning

We consider the standard RL scenario (Sutton and Barto 1998) in which an agent is interacting with an environment on a discrete time scale. At each time step t, the agent observes the state of the environment, $x_t \in \mathbf{X}$ and takes an action $a_t \in \mathbf{A}$. In response to its action, the agent receives an immediate (extrinsic) reward, r_{t+1} and the environment changes to a new state x_{t+1}. We assume that the environment is Markovian. The reward is expressed as $r_{t+1} = R(x_t, a_t)$, where $R : \mathbf{X} \times \mathbf{A} \to \mathbb{R}$ is the reward function. The next state x_{t+1} is drawn from the distribution $p(X_{t+1}|x_t, a_t)$[2]. Together, the reward function and the state transition distributions constitute the *model of the environment*. A way of behaving, called a *policy*, $\pi : \mathbf{X} \times \mathbf{A} \to [0, 1]$ is a probability distribution over actions, given the state. Each policy has an *action-value function* associated with it:

$$Q^{\pi}(x,a) = E_{\pi}\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | X_t = x, A_t = a\right], \tag{1}$$

where $\gamma \in (0, 1)$ is a discount factor expressing the fact that later rewards should be emphasized less. The interpretation of this value function is that the agent starts in state x, chooses a as its first action and thereafter chooses actions according to π. The goal of a reinforcement learning agent is to find a policy that maximizes the value function for all state-action pairs. In a finite Markov decision process (MDP), there is always at least one deterministic policy that achieves this goal, and many methods can be used to find such a policy, e.g. Q-learning (Watkins 1989). A comprehensive review of methods can be found in Sutton and Barto (1998). In some situations, e.g. when the state space $\mathbf{X}$ is too large and value functions cannot be represented exactly, policies are compared with respect to a starting state distribution, $p_0(X)$. Then, the goal is to maximize the expected return:

$$V^{\pi} = \sum_{x \in \mathbf{X}} \sum_{a \in \mathbf{A}} p_0(x) \pi(a|x) Q^{\pi}(x,a) \tag{2}$$

The advantage of using this criterion is that it allows a policy to be characterized by a single number, and offers a clear ordering of policies. Then, the optimal policy for the MDP maximizes V^{π}, for example, for the uniform starting distribution.

Suppose that we had a set of policies that all produce the same expected return. Which policy should be preferred? If one is to implement the policy on a real system, e.g. a

[1] We will refer to this parameter as the temperature in the rest of the article. One has to keep in mind that this is a metaphor, not a physical temperature.

[2] Here and throughout, we use capital letters to denote random variables, and small letters to denote particular realizations of these variables.

robot, then it is reasonable to prefer the simplest policy, i.e. the policy that can be specified with the smallest number of bits. To make this precise, let us reinterpret the meaning of a policy. The action can be viewed as a *summary* of the state of the environment. Therefore, mapping states onto actions can be viewed as lossy compression. If a large group of states share the same optimal action, then that action can be viewed as a *compressed representation* for this "class" of states, a representation which is sufficient from the point of view of attaining a desired level of return.

In order to formalize this intuition, we revisit rate distortion theory, introduced by Shannon (1948). Rate distortion theory measures the cost of approximating a signal Z by a signal Y, using the expected value of some distortion function, $d(Z, Y)$. This distortion measure can, but need not, be a metric. Lossy compression is achieved by assigning Z to Y via the probabilistic map $P(Y|z)$, such that the mutual information:

$$\begin{aligned} I(Z,Y) &= \sum_{z\in\Omega_Z}\sum_{y\in\Omega_Y} P(z,y)\log_2\left[\frac{P(z,y)}{P(z)P(y)}\right] \\ &= \sum_{z\in\Omega_Z}\sum_{y\in\Omega_Y} P(y|z)P(z)\log_2\left[\frac{P(y|z)}{P(y)}\right] \end{aligned} \tag{3}$$

is minimized. The minimization is constrained by fixing the expected distortion $\sum_{z\in\Omega_Z}\sum_{y\in\Omega_Y} P(z,y)d(z,y)$. In other words, recalling the meaning of information in terms of the rate of bit-flow, among the representations with the same quality, the most compact one will be preferred.

Now, let us interpret return as a function that measures quality, rather than distortion. The action is interpreded as a lossy summary of the state; hence, among the policies with the same return, we want to find the most compact one. Considering a set of policies that achieve a fixed average return V^π, we can express this idea through the following optimization problem:

$$\begin{aligned} \min_{\pi} \quad & I^\pi(A,X) \\ \text{subject to:} \quad & V^\pi = \text{const.} \\ & \sum_{a\in\mathbf{A}} \pi(a|x) = 1, \forall x \ \in \mathbf{X} \\ & \pi(a|x) \geq 0, \forall x \in \mathbf{X}, \forall a \in \mathbf{A} \end{aligned} \tag{4}$$

Here, π is the policy we seek, which can be viewed as a probabilistic assignment of states to actions. The second constraint ensures normalization, and the third ensures positivity of the probability function. The average return related to policy π, V^π, is defined in Eq. (2). The term $I^\pi(A, X)$ denotes the information that the action A carries about the state X under policy π, where the joint distribution is given by $p(X, A) = \pi(A|X)p^\pi(X)$:

$$I^\pi(A,X) = \sum_{x\in\mathbf{X}}\sum_{a\in\mathbf{A}} \pi(a|x)p^\pi(x)\log\left[\frac{\pi(a|x)}{p^\pi(a)}\right]. \tag{5}$$

Note that $I^\pi(A, X)$ depends also on the stationary distribution of states under policy π, $p^\pi(X)$ (which we assume exists, as is standard in RL), and on the average action probability, $p^\pi(A) = \sum_{x\in\mathbf{X}} p^\pi(x)\pi(A|x)$.

This optimization problem is complex, because of the dependence on the stationary distribution, which in general is unknown (though computable) and which changes as the policy π evolves during learning. A standard approach for changing the policy in reinforcement learning is to assume that we fix the policy π, compute its return, but then we consider a small perturbation around it at a given time step t. Let $V_t^\pi(q)$ be the expected return if the agent acts according to policy π on all time steps, except on time step t, when it chooses its action according to a different action distribution q:

$$\begin{aligned} V_t^\pi(q) = &\sum_{x_0 a_0 \ldots x_t, a_t} p_0(x_0)\left(\prod_{j=0}^{t-1}\pi(a_j|x_j)p(x_{j+1}|x_j,a_j)\right)q(a_t|x_t) \\ &\times\left[\sum_{i=0}^{t-1}\gamma^i R(x_i,a_i) + \gamma^t Q^\pi(x_t,a_t)\right] \end{aligned} \tag{6}$$

where $q(a_t|x_t)$ is the new probability of choosing action a_t from the state x_t, which we seek. Let $I_q^\pi(A_t, X_t)$ denote the information that the action A_t carries about the state X_t:

$$I_q^\pi(A_t,X_t) = \sum_{x\in\mathbf{X}}\sum_{a\in\mathbf{A}} q(a|x)p_t^\pi(x)\log\left[\frac{q(a|x)}{p_t^\pi(a)}\right], \tag{7}$$

where x and a range over the possible values of random variables X_t and A_t, $p_t^\pi(x)$ is the probability of arriving at state x on time step t if the agent starts with a state drawn from p_0 and chooses actions according to π:

$$\begin{aligned} p_t^\pi(x) &= p^\pi(X_t = x) \\ &= \sum_{x_0 a_0 \ldots x_{t-1} a_{t-1}} p_0(x_0)\pi(a_0|x_0)p(x_1|x_0,a_0)\ldots\pi(a_{t-1}|x_{t-1}) \\ &\qquad \times p(X_t = x|x_{t-1},a_{t-1}), \end{aligned} \tag{8}$$

and $p_t^\pi(a) = p^\pi(A_t = a) = \sum_{x\in\mathbf{X}} p_t^\pi(x)q(A_t = a|x)$.

Now, the optimization problem can be written as:

$$\begin{aligned} \min_{q} \quad & I_q^\pi(A_t,X_t) \\ \text{subject to:} \quad & V_t^\pi(q) = \text{const.} \\ & \sum_{a\in\mathbf{A}} q(a|x) = 1, \forall x \in \mathbf{X} \\ & q(a|x) \geq 0, \forall x \in \mathbf{X}, \forall a \in \mathbf{A}. \end{aligned} \tag{9}$$

This optimization problem has a dual form, where we maximize the average return under the constraint that the "size" of the policy is kept constant. Note that this is mathematically equivalent, but constitutes a potentially useful way to think about agents with limited computational capacity (e.g., robots with limited on-board computation). In this case, one may just want to find the best policy which still fits on the available physical system. Similar capacity constraints may apply to animals. The dual form is the following:

$$\begin{aligned} &\max_{q} \quad V_t^{\pi}(q) \\ &\text{subject to:} \quad I_q^{\pi}(A_t, X_t) = \text{const.} \\ &\qquad\qquad\quad \sum_{a \in \mathbf{A}} q(a|x) = 1; \forall x \in \mathbf{X}, \\ &\qquad\qquad\quad q(a|x) \geq 0, \forall x \in \mathbf{X}, \forall a \in \mathbf{A}. \end{aligned} \tag{10}$$

A similar cost function was given by Bagnell and Schneider (2003), as well as Peters and Schaal (2008); they used a linearization to compute a better type of policy gradient update.

We can now rewrite the constrained optimization principle, using the Lagrange multipliers λ and $\mu(x)$:

$$\max_{q} F[q], \tag{11}$$

$$\text{with: } F[q] = V_t^{\pi}(q) - \lambda I_q^{\pi}(A_t, X_t) + \sum_{x \in \mathbf{X}} \mu(x) \sum_{a \in \mathbf{A}} q(a|x), \tag{12}$$

where we have dropped irrelevant constants. The objective function, $F[q]$, is a functional of the policy q. The solution is obtained by setting the variation of F to zero which leads to the optimal policy

$$\begin{aligned} q_{\text{opt}}(A_t = a|X_t = x) &= \frac{p_t^{\pi}(a)}{Z(x)} e^{\frac{1}{\lambda}Q^{\pi}(x,a)} \\ &= \frac{1}{Z(x)} e^{\frac{1}{\lambda}Q^{\pi}(x,a) + \log\left[p_t^{\pi}(a)\right]}, \ \forall x \in \mathbf{X}, a \in \mathbf{A} \end{aligned} \tag{13}$$

which has to be solved self-consistently, together with:

$$p_t^{\pi}(a) = \sum_{x \in \mathbf{X}} q_{\text{opt}}(a|x) p_t^{\pi}(x), \ \forall a \in \mathbf{A} \tag{14}$$

The partition function $Z(x) = \sum_{a \in \mathbf{A}} p_t^{\pi}(a) e^{\frac{1}{\lambda}Q^{\pi}(x,a)}$ ensures normalization. This solution, Eq. (13) is similar to Boltzmann exploration, also known as softmax action selection (Sutton and Barto 1998). The difference is that here, we have an additional “complexity penalty,” $\log[p_t^{\pi}(a)]$. We note that by a similar calculation, if one tries to optimize the return at a fixed level of (Shannon) entropy, then one recovers exactly Boltzmann exploration. This follows immediately from the results in Rose (1998), and the arguments presented in Jaynes (1957). In contrast, here we penalize explicitly for the complexity of the policy, measured by the coding cost. The result is that there is a penalty for using more actions than necessary. This is useful not only when the agent has limited computational capacity, but also when the action space is very large (for example, in combinatorial optimization or inventory control problems). In this case, Eq. (13) may force the agent to use only a subset of the entire action space, which makes the learning task easier. The policy update in Eq. (13) appears also related to the ones suggested in references (Azar and Kappen 2010; Peters et al. 2010) despite their different roots.

This Boltzmann-style softening of the policy optimally trades the complexity of the policy for average return. The trade-off is governed by the temperature-like parameter λ, and exploration takes place due to fluctuations only at non-zero temperature, when emphasis is put on the compactness of the policy. As λ tends to zero, the information minimization constraint in Eq. (12) becomes less relevant, and in the limit, the optimal policy becomes deterministic if there are no degeneracies.[3] The optimal action then becomes:

$$a_{\text{opt}}(x) = \arg\max_{a} [Q^{\pi}(x,a)], \tag{15}$$

i.e. the action is chosen to maximize return. In this framework, explorative behavior is driven by randomness, the level of which is controlled by λ.

Curiosity-driven reinforcement learning

Intuitively, exploration should be driven by the curiosity to visit unknown areas of the state space of the coupled world-agent system. However, the theory we have laid out thus far is lacking any notion of curiosity. Apart from the coding rate constraint, the agent is just maximizing the return, as defined based on external rewards received from the environment.

In this section, we present a formalization of curiosity based on information-theoretic principles, drawing on ideas from Still (2009), where one of us has postulated that the main goal of a curious agent is to create for itself a world that is interesting. This is quantified by means of the predictive power that the agent’s behavior carries, as defined by Still (2009). In the present context of a fully observable Markovian environment, the agent’s sensations directly correspond to the state of the world, and predictive power simplifies to the mutual information carried by the action and the current state of the environment about the future state of the environment, $I[\{A_t, X_t\}; X_{t+1}]$.

Our goal then becomes to find the policy that maximizes predictive power together with expected return. We note that this is also important in extensions to partially observable MDPs (POMDPs), in which the exact state of the environment is unknown. Maximizing predictive power is

[3] If there are N actions that maximize $Q^{\pi}(x, a)$, then those occur with probability $1/N$, while all other actions occur with probability 0.

highly desirable in this setting, because it means that the agent is able to predict well its future sensation given the past data. An extension of the work in this section to the POMDP setting is possible, because the general framework (Still 2009) does not assume that the environment is fully observable.

Predictive information, defined as the mutual information shared between the past and the future of a time series, measures temporal structure and is related to other measures of complexity (Bialek et al. 2001; Crutchfield and Feldman 2001). This measure has shown up under other names in the literature, such as "stored information" in Shaw (1984); see also (Crutchfield and Feldman 2003) and references therein. It provides a measure of how complex, surprising, or how "interesting," a time series is. Intuitively, if a time series has high predictive information, there will be data available for learning about a variety of situations, and also about different ways of behaving. Formally, by taking predictive power into account, our objective becomes

$$\max_q \tilde{F}[q], \tag{16}$$

$$\text{with: } \tilde{F}[q] = I_q^\pi(\{X_t, A_t\}, X_{t+1}) + \ \alpha V_t^\pi(q) \\ -\lambda I_q^\pi(A_t, X_t) + \sum_{x \in \mathbf{X}} \mu(x) \sum_{a \in \mathbf{A}} q(a|x). \tag{17}$$

where we have dropped irrelevant constants. Variation of $I_q^\pi(\{X_t, A_t\}, X_{t+1})$ w.r.t. q, results in the additional contribution

$$\mathcal{D}^\pi(x,a) := D_{\mathrm{KL}}[p(X_{t+1}|x,a) \| p^\pi(X_{t+1})] \tag{18}$$

to the exponent in the solution. The Kullback-Leibler divergence is defined as

$$D_{\mathrm{KL}}[p_1(X) \| p_2(X))] = \sum_x p_1(x) \log\left[\frac{p_1(x)}{p_2(x)}\right], \tag{19}$$

and

$$p^\pi(X_{t+1} = x') = \sum_{a \in \mathbf{A}} \sum_{x \in \mathbf{X}} p(X_{t+1} = x'|x,a) q(a|x) p_t^\pi(x), \forall x' \in \mathbf{X} \tag{20}$$

With the extra contribution (18), the optimal solution is given by

$$q_{\mathrm{opt}}(A_t = a|X_t = x) = \frac{p_t^\pi(a)}{Z(x)} e^{\frac{1}{\lambda}(\mathcal{D}^\pi(x,a) + \alpha Q^\pi(x,a))}, \ \forall x \in \mathbf{X}, \forall a \in \mathbf{A}. \tag{21}$$

The first term in the exponent of Eq. (21) drives the agent toward exploration: the optimal action will maximize the divergence between the distribution over the next state, given the current state x and the action a, to the average distribution over the next state. This means that the optimal action policy will result in producing a next state with a conditional probability distribution far from the average distribution. The second term is the value maximization, as before. The exponent in Eq. (21) thus represents a trade-off between exploration and exploitation.

As $\lambda \to 0$, the chosen action becomes the one that maximizes the functional in the exponent of Eq. (21):[4]

$$a_{\mathrm{opt}}(x) = \arg\max_a [\mathcal{D}^\pi(x,a) + \alpha Q^\pi(x,a)] \tag{22}$$

Importantly, the optimal action policy includes a trade-off between exploration and exploitation, even when the policy is deterministic. Recall that this is not the case for Boltzmann exploration, as we can see from comparison with Eq. (15). There, the optimal action under a deterministic policy maximizes only the return, subject possibly to size constraints, and exploration arises only at nonzero temperature, due to randomization.

The parameter α can be viewed as a measure of how interested the agent is in obtaining a reward. For example, if the reward is energy intake, then α could be set by measuring the charge of a robot's batteries, and would represent how "hungry" the agent is.

Illustrations

To build some intuition about what this approach to curiosity-driven RL does, we consider two simple examples of a world in which there are two states, $x \in \{0, 1\}$, and assume that $\alpha = 0$, so the optimal action becomes the one that maximizes only the predictive power. All calculations for this section can be found in the Appendix.

First, consider a continuous range of actions $a \in [0, 1]$. Let the value of the action express how strongly the agent tries to stay in the same state or leave it, such that $a = 0$ means that the agent wants to remain in the same state, $a = 1/2$ means that the agent is ambivalent about staying or leaving, and $a = 1$ means that the agent tries to switch states. Let the Markovian transitions of the environment be given by $p(\bar{x}|x,a) = a$. Then, the optimal policy, Eq. (21), chooses only those two actions which result in the largest predictability, namely $a = 0$ and $a = 1$, and it chooses between these two actions with equal probability. This "clever random" policy is an example of the balance between control and exploration that was mentioned by Still (2009). Note the difference to a random policy, which would assign equal probability to all possible values of a.

As a second case, consider a two-state world in which there are only two actions, STAY or FLIP, $a \in \{s, f\}$, and the transition probabilities are such that one state is completely reliable: $p(0|0, s) = p(1|0, f) = 1$, while the other

[4] The assignment becomes deterministic if there are no degeneracies, otherwise all those actions occur with equal probability, as in Sect. 2.

state is completely unreliable $p(0|1, s) = p(0|1, f) = 1/2$. This is a test for our information-theoretic objective: if we are doing the right thing, then we should find that in the absence of a reward (or the absence of an interest in a reward, $\alpha = 0$), the optimal curious policy will enable exploration of the combined state-action space, which means that the optimal policy should *not* stay in the more reliable state with probability one, i.e., we should *not* find $\pi(A = s|0) = 1$. Indeed, maximizing $I[\{X_t, A_t\}; X_{t+1}]$ results asymptotically in the optimal policy $\pi(A = s|0) = 3/4$, which balances between exploration and choosing a reliable state, i.e. control.

Algorithmic issues

The optimal solution consists of Eq. (21), which has to be solved self-consistently, together with Eq. (20). Furthermore, the action-value function Q has to be estimated. In this section, we discuss how this can be implemented in practice.

We propose an implementation that is inspired by the usual Boltzmann exploration algorithm. The algorithm proceeds as follows:

1. Initialize $t \leftarrow 0$ and get initial state x_0. Initialize $\pi(a|x), \forall x \in \mathbf{X}, \forall a \in \mathbf{A}$ (e.g., uniformly randomly) and initialize the action-value function Q.
2. Repeat at every time step t

   (a) Update $p_t(x), \forall x \in \mathbf{X}$ (the current estimate of the state visitation distribution)

   (b) Initialize $q^{(0)}(a|x), \forall a \in \mathbf{A}, \forall x \in \mathbf{X}$

   (c) Repeat the following updates, until the difference between $q^{(j)}$ and $q^{(j+1)}$ is small:

   $$p^{(j)}(a) \leftarrow \sum_x q^{(j)}(a|x)p_t(x), \forall a \in \mathbf{A} \tag{23}$$

   $$p^{(j)}(x') \leftarrow \sum_x \sum_a p(x'|a,x)q^{(j)}(a|x)p_t(x), \quad \forall x' \in \mathbf{X} \tag{24}$$

   $$q^{(j+1)}(a|x) \leftarrow \frac{p^{(j)}(a)}{Z^{(j)}(x)} \exp\left[\frac{1}{\lambda}(\mathcal{D}^{\pi}(x,a) + \alpha Q(x,a))\right], \quad \forall x \in \mathbf{X}, \forall a \in \mathbf{A} \tag{25}$$

   Update $\pi \leftarrow q^{(j+1)}$

   (d) Choose action $a_t \sim \pi(\cdot|x_t)$ and obtain reward r_{t+1} and next state x_{t+1}

   (e) Update the action-value function estimates Q.

   (f) $t \leftarrow t + 1$

In this algorithm, step 2 (a) can be performed exactly by using the true model and all the previous policies; the update of the model is similar to the one in Eq. (24); we note that this is exactly the same type of update used in the forward algorithm in a hidden Markov model (HMM). However, this computation can be expensive if the number of states is large. As a result, in this case, we would use the state samples x_k, $k \leq t$, to estimate $p_t(x)$ approximately.

The initial value $q^{(0)}(a|x)$ is important, as it will influence the point to which iteration 2(c) converges (convergence is guaranteed to a locally optimal solution). One can start with the result of the previous iteration, under the assumption that the policy will change fairly smoothly from one time step to the next.

In principle, the action-value function Q should be recomputed exactly at every step, using the known model and the computed policy. This involves solving a system of linear equations with $|\mathbf{X}| \times |\mathbf{A}|$ unknowns. While this may be feasible for small environments, it is computationally expensive for larger problems. In this case, the value $Q(x_t, a_t)$ can instead be updated incrementally, using the standard temporal-difference learning approach (i.e. a learning rule like Sarsa or Q-learning; for details see Sutton and Barto 1998). Intuitively, this approach should work well if the policy changes slowly, because the action-value function will only change around the current state x_t. Similarly, in order to save computation, the policy may be recomputed only at x_t, rather than at all states $x \in \mathbf{X}$, as indicated in Eq. (25).

If the agent has no knowledge of the environment, then it can use the samples received to fit an approximate model, $\hat{p}(X_{t+1}|x_t, a_t)$, and then use this model in the computation above. The model, action values, and distributions of interest can all be updated incrementally from samples. If a batch of samples is gathered first and then we run the algorithm above, we obtain an approach fairly close to batch model-based reinforcement learning. If on every time step t we update the model estimate $\hat{p}(X_{t+1}|X_t, A_t)$ and immediately use it in the policy computation, we obtain an algorithm very close to incremental, model-free reinforcement learning. In general, the optimal policy can be re-computed every T time steps, and the approximate model can be improved using the T samples from the intervening period.

The temperature-like parameter λ determines how deterministic the resulting policy is. There are different possibilities for choosing this parameter. In the simplest case, the parameter is fixed to a pre-specified value, for example dictated by the capacity/memory constraints of a robot. This selects a fixed trade-off between complexity and utility. More generally, a process known as deterministic annealing (Rose 1998) can be employed at every time step. It consists of starting with a large temperature, running the iterative algorithm until convergence, then lowering the temperature by a factor (the annealing rate), and continuing this process, until the policy is deterministic, always using the current result as initial conditions for the iterations at the next (lower) temperature. This method obtains, at each time step, the deterministic, optimal policy, according to the criterion. The procedure is computationally intensive, but guarantees that actions are always chosen in a way that maximizes the optimization criterion, given that the annealing rate is sufficiently slow. Finally, the temperature can be fixed during each time step, but lowered as a function of time, $\lambda(t)$, until it approaches zero. This approach is preferable when the agent's knowledge about the world increases with time. Methods such as the ones discussed by Still and Bialek (2004) can be used to find (a bound on) $\lambda(t)$. Finally, if a complexity constraint is given by the design of the agent, then this scheme can be modified to include a $\lambda_{\min} = \lim_{t\to\infty} \lambda(t)$.

If the algorithm is implemented using only exact computations (i.e., the most computationally expensive version, outlined above), it is guaranteed to converge to a locally optimal solution for the proposed optimization criterion. Convergence analysis for the case in which samples are used incrementally is quite tricky and we leave it for future work.

Related work

The textbook by Sutton and Barto (1998) summarizes several randomization-based exploration approaches used in reinforcement learning, such as Boltzmann exploration and ϵ-greedy (in which there is simply a fixed, small probability of trying out actions which appear sub-optimal). Many heuristic variations have been proposed, in which bonuses are added to the value function to encourage more efficient exploration (e.g. Thrun and Moeller 1992; Ratitch and Precup 2003)

A different strategy, which yields interesting theoretical results, is that of optimism in the face of uncertainty: if a state has not been visited sufficiently for the agent to be familiar with it, it is automatically considered good, so the agent will be driven toward it. This ensures that an agent will explore new areas of the state space. The first sample-complexity results for reinforcement learning using this idea were provided by Kearns and Singh (1998). The authors assumed that a state is "known" if it has been visited a sufficiently large number of times. The RMAX algorithm proposed by Brafman and Tennenholtz (2002) is a practical implementation of this idea. An extensive theoretical analysis of this approach was given by Strehl et al. (2006), showing sample-complexity results both for reinforcement learning methods that learn a model and ones that learn directly a value function. Those PAC-style bounds are not directly related to our work.

Previous work on curiosity-driven reinforcement learning is centered around the idea that agents are motivated by an internal reward signal, and in the process of maximizing this reward, they learn a collection of skills. The work we presented in Sect. 3 could be interpreted as defining implicitly an intrinsic reward, based on the idea of maximizing how interesting is the time series experienced by the agent. In early work Schmidhuber (1991) proposed different kinds of internal reward signals. More recently, a hierarchical learning approach was put forth by Singh et al. (2005). In this case, both an external and an internal reward signal are used to learn a behavior policy. At the same time, the extrinsic reward is used to learn multiple temporally extended behaviors. The particular setting proposed for the intrinsic reward is attempting to provide a novelty bonus. We note that the intrinsic reward is only used to generate behavior. The paper assumes that there are certain events in the world that are "salient" and which the agent will be motivated to seek.We furthermore note that the intrinsic rewards proposed by Singh et al. (2005) also involve the probability of the next state given the current state, under different extended behaviors. However, this is proposed as a heuristic. The relationship to our results remains to be explored. Oudeyer and colleagues implemented these ideas in robotics tasks (see e.g. Oudeyer et al. 2007). Recently, Schmidhuber (2009) proposed a novel approach to creativity and exploration which is related to information theory.

Information-theoretic approaches inspired by some form of rate distortion theory have been used in machine learning, for example for clustering and dimensionality reduction (Rose et al. 1990; Pereira et al. 1993; Rose 1998; Tishby et al. 1999; Chigirev and Bialek 2004; Still and Bialek 2004; Still et al. 2004; Chechnik et al. 2005). Ay et al. (2008) explore maximization of $I[X_t, X_{t+1}]$ for a specific class of models. In contrast, the work we have presented in Sect. 3 maximizes predictive power while keeping the coding rate, or model complexity, fixed, and thereby penalizing policies with more memory than is needed for prediction. This is in line with Still (2009). Tishby and Polanyi (2010) proposed an MDP formulation in which rewards are traded off against information. The authors observe that information also obeys Bellman-like equations, and use this observation to set up dynamic programming algorithms for solving such MDPs. Our work is different in a few important aspects. First, the development in Tishby and Polani (2010) is with respect to a single state distribution, while we account for the state distributions induced by different policies. Second, in their formulation, the information value ends up mixed with the value function. In our case, information influences the exploration policy, but ultimately, one can still obtain a value function, and a policy, that reflect only reward optimization. Another important distinction is that in their formulation, deterministic policies are more "complex" than randomized ones, whereas in our case, a deterministic policy that is constant everywhere would still be considered "simple." We anticipate that such a treatment will be important in the generalization of these ideas to continuous states and actions (where simple policies will share the same choices across large subsets of states). Interestingly, Little and Sommer (2011) considered several measures for estimating (or approximating) the information gain of an action in the context of past data. They found that learning efficiency is strongly dependent on temporal integration of information gain but less dependent on the particular measure used to quantify information gain.

The recent work on differential dynamic programming (e.g. Todorov 2009; Azar and Kappen 2010) addresses the problem of finding closed-form solutions to reinforcement learning problems, by reformulating the optimization objective. More specifically, the system is considered to have "passive" dynamics (induced by a default policy). The optimization criterion then includes both the value function and a term that penalizes deviations form the "passive" dynamics (using the KL-divergence between the state distributions induced by the sought policy and the default policy). This line of work comes from the perspective of continuous control. The results obtained for the optimal policy bear some similarity with the updates we obtain, but the motivation behind the approach is very different. A similarly defined policy update is also obtained by Peters et al. (2010), coming from yet another different angle. They formulate an optimization problem in which the goal is to utilize the existing samples as well as possible. They give a policy search algorithm in which new policies are penalized if they induce a state distribution that is different from the empirical distribution observed in the past data. The fact that very different points of view lead to syntactically similar policy updates is intriguing and we plan to study it further in future work.

Conclusion and future work

In this article, we introduced a new information-theoretic perspective on the problem of optimal exploration in reinforcement learning. We focused, for simplicity, on Markovian environments, in which the state of the environment is observable and does not have to be learned.

First, we showed that a soft policy similar to Boltzmann exploration optimally trades return for the coding cost (or complexity) of the policy. Second, by postulating that an agent should, in addition to maximizing the expected return, also maximize its predictive power, at a fixed policy complexity, we derived a trade-off between exploration and exploitation that does not rely on randomness in the action policy. In this view, exploration is an emergent behavior that arises to maximize predictive power. This may be a more adequate way of modeling explorative behavior than previous schemes, such as Boltzmann exploration, where exploration hinges upon randomization of the action policy.

Our results can be extended easily to POMDPs, using the framework of Still (2009). In that case, the additional goal is to build a good, predictive internal representation of the environment. Our theoretical framework can also be extended to continuous states and actions; very little work has been done so far in this direction (Wingate and Singh 2007). A third important direction for future work is empirical: we are currently evaluating the proposed method in comparison to existing exploration techniques. Experience in large domains will be especially useful in the future.

Acknowledgment This research was funded in part by NSERC and ONR.

Appendix

Clever random policy

There are two world states, $x \in \{0, 1\}$ and a continuous action set, $a \in [0, 1]$. The value of the action sets how strongly the agent tries to stay in or leave a state, and $p(\bar{x}|x, a) = a$. The interest in reward is switched off ($\alpha = 0$), so that the optimal action becomes the one that maximizes only the predictive power.

Policies that maximize $I[X_{t+1}, \{X_t, A_t\}]$

For brevity of notation, we drop the index t for the current state and action.

$$I[X_{t+1}, \{X, A\}] = H[X_{t+1}] - H[X_{t+1}|X, A] \quad (26)$$

The second term in (24) is minimized and equal to zero for all policies that result in deterministic world transitions. Those are all policies for which $\pi(\tilde{a}|x) = 0$ for all $\tilde{a} \notin \{0, 1\}$. This limits the agent to using only two (the most extreme) actions: $a \in \{0, 1\}$. Since we have only two states, policies in this class are determined by two probabilities, for example the flip probabilities $\pi(A = 0|X = 1)$ and $\pi(A = 1|X = 0)$.

The first term in Eq. (26) is maximized for $p(X_{t+1} = 1) = p(X_{t+1} = 0) = 1/2$. Setting $p(X_{t+1} = 1)$ to 1/2 yields

$$\pi(A=0|X=1)p(X=1) + \pi(A=1|X=0)p(X=0) = \frac{1}{2}. \quad (27)$$

We assume that $p(X = 0)$ is estimated by the learner. Equation 27 is true *for all* values of $p(X = 0)$, if $\pi(A = 0|X = 1) = \pi(A = 1|X = 0) = 1/2$. We call this the "clever random" policy (π_R). The agent uses only those actions that make the world transitions deterministic, and uses them at random, i.e. it explores within the subspace of actions that make the world deterministic. This policy maximizes $I[X_{t+1}, \{X, A\}]$, independent of the estimated value of $p(X = 0)$.

However, when stationarity holds, $p(X = 0) = p(X = 1) = 1/2$, then all policies for which

$$\pi(A = 0|X = 1) = \pi(A = 0|X = 0) \tag{28}$$

maximize $I[X_{t+1}, \{X, A\}]$. Those include "STAY-STAY", and "FLIP-FLIP".

Self-consistent policies.

Since $\alpha = 0$, the term in the exponent of Eq. (21), for a given state x and action a, is:

$$\mathcal{D}^{\pi}(x,a) = -H[a] + a \log\left[\frac{p(X_{t+1} = x)}{p(X_{t+1} = \bar{x})}\right] - \log[p(X_{t+1} = x)] \tag{29}$$

with $\bar{x}$ being the opposite state, and $H[a] = -(a\log(a) + (1-a)\log(1-a))$. Note that $H\ [0] = H\ [1] = 0$. The clever random policy π_R is self-consistent, because under this policy, for all x, both actions, STAY ($a = 0$) and FLIP ($a = 1$) are equally likely. This is due to the fact that $p(X_{t+1} = x) = p(X_{t+1} = \bar{x}) = 1/2$, hence $D^{\pi_R}(x, 0) = D^{\pi_R}(x, 1), \forall x$. If stationarity holds, $p(X = 0) = 1/2$, and no policy which uses only actions $a \in \{0, 1\}$ other than policy π_R is self consistent. This is because under other such policies we also have that $p(X_{t+1} = x) = p(X_{t+1} = \bar{x}) = 1/2$, and we have $H[0] = H[1] = 0$, and therefore $D^{\pi}(x, 0) - D^{\pi}(x, 1) = 0$. This means that the algorithm gets to π_R after one iteration. We can conclude that π_R is the unique optimal self-consistent solution.

A reliable and an unreliable state

There are two possible actions, STAY (s) or FLIP (f), and two world states, $x \in \{0, 1\}$, distinguished by the transitions: $p(X_{t+1} = 0|X_t = 0, A_t = s) = p(X_{t+1} = 1|X_t = 0, A_t = f) = 1$, while $p(X_{t+1} = x|X_t = 1, a) = 1/2, \forall x, \forall a$. In other words, state 0 is fully reliable, and state 1 is fully unreliable, in terms of the action effects. There is no uncertainty when we start in the reliable state, and the uncertainty when starting in the unreliable state is exactly one bit. The predictive power is then given by

$$\begin{aligned} I[X_{t+1}, \{X, A\}] = &- \sum_{x \in \{0,1\}} p(X_{t+1} = x) \log_2[p(X_{t+1} = x)] \\ &- p(X_t = 1) \end{aligned} \tag{30}$$

Starting with a fixed value for $p(X_t = 1)$ which is estimated from past experiences, the maximum is reached by a policy that results in equiprobable futures, i.e., $p(X_{t+1} = 1) = 1/2$. We have $p(X_{t+1} = 0) = \pi(A = s|X = 0)p(X = 0) + \frac{1}{2}p(X = 1)$. Therefore, this implies that $\pi(A = s|X = 0) = 1/2$, which, in turn, implies that after some time $p(X_t = 1) = 1/2$, and thus $I[X_{t+1}, \{X, A\}] = 1/2$. However, asymptotically, $p(X_t = 0) = p(X_{t+1} = 0)$, and the information is given by $-(p(X = 0)\log_2[p(X = 0)/(1 - p(X = 0))] - \log_2[1 - p(X = 0)]) + p(X = 0) - 1$. Setting the first derivative, $1 - \log_2[p(X = 0)/(1 - p(X = 0))]$, to zero implies that the extremum lies at $p(X = 0) = 2/3$, where the information reaches $\log_2(3) - 1/3 \simeq 5/4$ bits. Now, $p(X_{t+1} = 0) = 2/3$ implies that $\pi(A = s|X = 0) = 3/4$. Asymptotically, the optimal strategy is to stay in the reliable state with probability 3/4. We conclude that the agent starts with the random strategy in state 0, i.e., $\pi(A = s|X = 0) = 1/2$, and asymptotically finds the strategy $\pi(A = s|X = 0) = 3/4$. This asymptotic strategy still allows for exploration, but it results in a more controlled environment than the purely random strategy. Note that the optimal policy in state 1 is obviously random, i.e. $\pi(A|1) = 1/2$, because $D_{\mathrm{KL}}[p(X_{t+1}|X_t = 1, A_t = s)||p(X_{t+1})] = D_{\mathrm{KL}}[p(X_{t+1}|X_t = 1, A_t = f)||p(X_{t+1})]$.

References

Ay N, Bertschinger N, Der R, Guttler F, Olbrich E (2008) Predictive information and explorative behavior of autonomous robots. Eur Phys J B 63:329–339

Azar MG, Kappen HJ (2010) Dynamic policy programming. J Mach Learn Res arXiv:1004.2027:1–26

Bagnell JA, Schneider J (2003) Covariant policy search. In: International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico

Bialek W, Nemenman I, Tishby N (2001) Predictability, complexity and learning. Neural Comput 13:2409–2463

Brafman RI, Tennenholtz M (2002) R-max—a general polynomial time algorithm for near-optimal reinforcement learning. J Mach Learn Res 3:213–231

Chechnik G, Globerson A, Tishby N, Weiss Y (2005) Information bottleneck for gaussian variables. J Mach Learn Res 6:165–188

Chigirev DV, Bialek W (2004) Optimal manifold representation of data: an information theoretic perspective. In: Thrun S, Saul L, Schölkopf B (eds) Advances in neural information processing systems 16. MIT Press, Cambridge, MA

Crutchfield JP, Feldman DP (2001) Synchronizing to the environment: information theoretic limits on agent learning. Adv Complex Syst 4(2):251–264

Crutchfield JP, Feldman DP (2003) Regularities unseen, randomness observed: levels of entropy convergence. Chaos 13(1):25–54

Jaynes ET (1957) Information theory and statistical mechanics. Phys Rev 106(4):620–630

Kearns M, Singh S (Eds) (1998) Near-optimal reinforcement learning in polynomial time. In: Proceedings of the 15th International Conference on Machine Learning, pp 260–268

Little DY, Sommer FT (2011) Learning in embodied action-perception loops through exploration. arXiv:1112.1125v2

Oudeyer P-Y, Kaplan F, Hafner V (2007) Intrinsic motivation systems for autonomous mental development. IEEE Trans Evol Comput 11(2):265–286

Pereira F, Tishby N, Lee L (1993) Distributional clustering of english words. In 30th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 183–190. http://xxx.lanl.gov/pdf/cmp-lg/9408011

Peters J, Muelling K, Altun Y (2010) Relative entropy policy search. In: Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence (AAAI). AAAI Press, Menlo Park

Peters J, Schaal S (2008) Reinforcement learning of motor skills with policy gradients. Neural Netw 21(4):682–697

Ratitch B, Precup D (2003) Using MDP characteristics to guide exploration in reinforcement learning. In: Proceedings of ECML, pp 313–324

Rose K (1998) Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proc IEEE 86(11):2210–2239

Rose K, Gurewitz E, Fox GC (1990) Statistical mechanics and phase transitions in clustering. Phys Rev Lett 65(8):945–948

Schmidhuber J (1991) Curious model-building control systems. In Proceedings of IJCNN, pp 1458–1463

Schmidhuber J (2009) Art and science as by-products of the search for novel patterns, or data compressible in unknown yet learnable ways. In: Multiple ways to design research. Research cases that reshape the design discipline. Swiss Design Network—et al. Edizioni, 2009, pp 98–112

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423, 623–656

Shaw R (1984) The dripping faucet as a model chaotic system. Aerial Press, Santa Cruz, California

Singh S, Barto AG, Chentanez N (2005) Intrinsically motivated reinforcement learning. In Proceedings of NIPS, pp 1281–1288

Still S (2009) Information-theoretic approach to interactive learning. EPL 85 28005. doi:10.1209/0295-5075/85/28005

Still S, Bialek W (2004) How many clusters? An information theoretic perspective. Neural Computation 16(12):2483–2506

Still S, Bialek W, Bottou L (2004) Geometric clustering using the information bottleneck method. In: Thrun S, Saul LK, Schölkopf B (eds) Advances in neural information processing systems 16. MIT Press, Cambridge, MA

Strehl AL, Li L, Littman ML (2006) Incremental model-based learners with formal learning-time guarantees. In: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, Cambridge, MA

Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge

Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res 10(1):1633–1685

Thrun S, Moeller K (1992) Active exploration in dynamic environments. In: Advances in Neural Information Processing Systems (NIPS) 4, San Mateo, CA, pp 531–538

Tishby N, Pereira F, Bialek W (1999) The information bottleneck method. In: Proceedings of the 37th Annual Allerton Conference, pp 363–377

Tishby N, Polani D (2010) Information theory of decisions and actions. In: Perception-reason-action cycle: models, algorithms and systems. Springer, New York

Todorov E (2009) Efficient computation of optimal actions. Proc Nat Acad Sci USA 106(28):11478–11483

Watkins CJCH (1989) Learning from delayed rewards. PhD thesis, Cambridge University

Wingate D, Singh S (2007) On discovery and learning of models with predictive representations of state for agents with continuous actions and observations. In Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp 1128–1135