Optimism in Reinforcement Learning and Kullback-Leibler Divergence

Sarah Filippi, Olivier Cappé, Aurélien Garivier

Optimism in Reinforcement Learning and Kullback-Leibler Divergence

Sarah Filippi, Olivier Cappé, and Aurélien Garivier *

LTCI, TELECOM ParisTech and CNRS
46 rue Barrault, 75013 Paris, France
(filippi,cappe,garivier)@telecom-paristech.fr,

Abstract. We consider model-based reinforcement learning in finite Markov Decision Processes (MDPs), focussing on so-called optimistic strategies. In MDPs, optimism can be implemented by carrying out extended value iterations under a constraint of consistency with the estimated model transition probabilities. The UCRL2 algorithm by Auer, Jaksch and Ortner (2009), which follows this strategy, has recently been shown to guarantee near-optimal regret bounds. In this paper, we strongly argue in favor of using the Kullback-Leibler (KL) divergence for this purpose. By studying the linear maximization problem under KL constraints, we provide an efficient algorithm, termed KL-UCRL, for solving KL-optimistic extended value iteration. Using recent deviation bounds on the KL divergence, we prove that KL-UCRL provides the same guarantees as UCRL2 in terms of regret. However, numerical experiments on classical benchmarks show a significantly improved behavior, particularly when the MDP has reduced connectivity. To support this observation, we provide elements of comparison between the two algorithms based on geometric considerations.

Key words: Reinforcement learning; Markov decision processes; Model-based approaches; Optimism; Kullback-Leibler divergence; Regret bounds

1 Introduction

In reinforcement learning, an agent interacts with an unknown environment, aiming to maximize its long-term payoff [17]. This interaction is commonly modelled by a Markov Decision Process (MDP) and it is assumed that the agent does not know the parameters of the process and needs to learn directly from observations. The agent thus faces a fundamental trade-off between gathering experimental data about the consequences of the actions (exploration) and acting consistently with past experience in order to maximize the rewards (exploitation).

We consider in this article an MDP with finite state and action spaces for which we propose a *model-based* reinforcement learning algorithm, i.e., an algorithm that maintains a running estimate of the model parameters (transitions

* This publication is partially supported by Orange Labs under contract n°289365.

probabilities and expected rewards)[8,11,16,18]. A well-known approach to balance exploration and exploitation, followed for example by the well-know algorithm R-MAX [6], is the so-called *optimism in the face of uncertainty* principle. It was first proposed in the multi-armed bandit context by [12], and has been extended since then to several frameworks: instead of acting optimally according to the estimated model, the agent follows the optimal policy for a surrogate model, named *optimistic model*, which is close enough to the former but leads to a higher long-term reward. The performance of such an algorithm can be analyzed in term of *regret*, which consists in comparing the rewards collected by the algorithm with the rewards obtained when following an optimal policy. The study of the asymptotic regret due to [12] in the multi-armed context has been extended to MDPs by [7], proving that an optimistic algorithm can achieve logarithmic regret. The subsequent works of [3,2,4] introduced algorithms that guarantee non-asymptotic logarithmic regret in a large class of MDPs. In these latter works, the optimistic model is computed using the L^1 (or total variation) norm as a measure of proximity between the estimated and optimistic transition probabilities.

In addition to logarithmic regret bounds, the UCRL2 algorithm of [2] is also attractive due to the simplicity of each L^1 extended value iteration step. In this case, optimism simply results in adding a bonus to the most promising transition (i.e., the transition that leads to the state with current highest value) while removing the corresponding probability mass from less promising transitions. This process is both elementary and easily interpretable, which is desirable in some applications.

However, the L^1 extended value iteration leads to undesirable pitfalls, which may compromise the practical performance of the algorithm. First, the optimistic model is not continuous with respect to the estimated parameters – small changes in the estimates may result in very different optimistic models. More importantly, the L^1 optimistic model can become incompatible with the observations by assigning a probability of zero to a transition that has actually been observed. Moreover, in MDPs with reduced connectivity, L^1 optimism results in a persistent bonus for all transitions heading towards the most valuable state even when significant evidence has been accumulated that these transitions are indeed not sufficiently likely.

In this paper, we propose an improved optimistic algorithm, called KL-URCL, that avoids these pitfalls altogether. The key is the use of the Kullback-Leibler (KL) pseudo-distance instead of the L^1 metric, as in [7]. Indeed, the smoothness of the KL metric largely alleviates the first issue. The second issue is completely avoided thanks to the strong relationship between the geometry of the probability simplex induced by the KL pseudo-metric and the theory of large deviations. For the third issue, we show that the KL-optimistic model stems from a trade-off between the relative value of the most promising state and the statistical evidence accumulated so far regarding its reachability.

We provide an efficient procedure, based on one-dimensional line searches, to solve the linear maximization problem under KL constraints. As a conse-

quence, the numerical complexity of the KL-URCL algorithm is comparable to that of UCRL2. Building on the analysis of [2,4], we also obtain logarithmic regret bounds for the KL-UCRL algorithm. The proof of this result is based on novel concentration inequalities for the KL-divergence, which have interesting properties when compared with those traditionally used for the L^1 norm. Although the obtained regret bounds are comparable to earlier results in term of rate and dependence in the number of states and actions, we observed in practice significant performance improvements. This observation is illustrated using a classic benchmark example (the *river swim* environment of [16]) and through a thorough discussion of the geometric properties of KL neighborhoods.

The paper is organized as follows. The model and a brief survey of the value iteration algorithm in undiscounted MDPs are presented in Section 2. Section 3 and 4 are devoted, respectively, to the description and the analysis of the KL-UCRL algorithm, with corresponding proofs in appendix. Section 5 contains our numerical results and Section 6 concludes the paper by discussing the advantages of using KL rather than L^1 confidence neighborhoods.

2 Markov Decision Process

Consider a Markov decision process (MDP) $\mathbf{M} = (\mathcal{X}, \mathcal{A}, P, r)$ with finite state space $\mathcal{X}$, and action space $\mathcal{A}$. Let $X_t \in \mathcal{X}$ and $A_t \in \mathcal{A}$ denote respectively the state of the system and the action chosen by the agent at time t. The probability to jump from state X_t to state X_{t+1} is denoted by $P(X_{t+1}; X_t, A_t)$. Besides, the agent receives at time t a random reward $R_t \in [0,1]$ with mean $r(X_t, A_t)$. The aim of the agent is to choose the sequence of actions so as to maximize the cumulated reward. His choices are summarized in a *stationary policy* $\pi : \mathcal{X} \to \mathcal{A}$.

In this paper, we consider *weakly communicating* MDPs, i.e., MDPs satisfying the *weak accessibility* condition [5]: the set of states can be partitioned into two subsets $\mathcal{X}_t$ and $\mathcal{X}_c$ such that all states in $\mathcal{X}_t$ are transient under every stationary policy and, for any states $x, x' \in \mathcal{X}_c$, there exists a policy $\pi_{x,x'}$ that takes one from x to x'. For those MDPs, it is known that the *average reward* following a stationary policy π, denoted by $\rho^\pi(\mathbf{M})$ and defined as

$$\rho^\pi(\mathbf{M}) = \lim_{n\to\infty} \frac{1}{n} \mathbb{E}_{\mathbf{M},\pi} \left(\sum_{t=0}^{n} R_t \right) ,$$

is state-independent [15]. Let $\pi^*(\mathbf{M}) : \mathcal{X} \to \mathcal{A}$ and $\rho^*(\mathbf{M})$ denote respectively the optimal policy and the optimal average reward: $\rho^*(\mathbf{M}) = \sup_\pi \rho^\pi(\mathbf{M}) = \rho^{\pi^*(\mathbf{M})}(\mathbf{M})$. The notations $\rho^*(\mathbf{M})$ and $\pi^*(\mathbf{M})$ are meant to highlight the fact that both the optimal average reward and the optimal policy depend on the model $\mathbf{M}$. The optimal average reward satisfies the so-called *optimality equation*

$$\forall x \in \mathcal{X} , \qquad h^*(\mathbf{M}, x) + \rho^*(\mathbf{M}) = \max_{a\in\mathcal{A}} \left(r(x,a) + \sum_{x'\in\mathcal{X}} P(x'; x, a) h^*(\mathbf{M}, x') \right) .$$

where the $|\mathcal{X}|$-dimensional vector $h^*(\mathbf{M})$ is called a *bias* vector. Note that it is only defined up to an additive constant. For a fixed MDP $\mathbf{M}$, the optimal policy $\pi^*(\mathbf{M})$ can be derived by solving the optimality equation and by defining, for all $x \in \mathcal{X}$,

$$\pi^*(\mathbf{M}, x) \in \operatorname*{argmax}_{a \in \mathcal{A}} \left(r(x,a) + \sum_{x' \in \mathcal{X}} P(x'; x, a) h^*(\mathbf{M}, x) \right) .$$

In practice, the optimal average reward and the optimal policy may be computed using the value iteration algorithm [15].

3 The KL-UCRL algorithm

In this paper, we focus on the reinforcement learning problem in which the agent does not know the model $\mathbf{M}$ beforehand, i.e. the transition probabilities and the distribution of the rewards are unknown. More specifically, we consider model-based reinforcement learning algorithms which estimate the model through observations and act accordingly. Denote by $\hat{P}_t(x'; x, a)$ the estimate at time t of the transition probability from state x to state x' conditionally to the action a, and, by $\hat{r}_t(x,a)$ the mean reward received in state x when action a has been chosen. We have:

$$\hat{P}_t(x'; x, a) = \frac{N_t(x,a,x')}{\max(N_t(x,a),1)} \qquad \text{and} \qquad \hat{r}_t(x,a) = \frac{\sum_{k=1}^{t-1} R_k \mathbb{1}_{\{X_k = x, A_k = a\}}}{\max(N_t(x,a),1)} , \tag{1}$$

where $N_t(x,a,x') = \sum_{k=0}^{t-1} \mathbb{1}_{\{X_k = x, A_k = a, X_{k+1} = x'\}}$ is the number of visits, up to time t, to the state x followed by a visit to x' if the action a has been chosen, and similarly, $N_t(x,a) = \sum_{k=0}^{t-1} \mathbb{1}_{\{X_k = x, A_k = a\}}$. The optimal policy in the estimated model $\hat{\mathbf{M}}_t = (\mathcal{X}, \mathcal{A}, \hat{P}_t, \hat{r}_t)$ may be misleading due to estimation errors: pure exploitation policies are commonly known to fail with positive probability. To avoid this problem, *optimistic model-based approaches* consider a set $\mathcal{M}_t$ of potential MDPs including $\hat{\mathbf{M}}_t$ and choose the MDP from this set that leads to the largest average reward. In the following, the set $\mathcal{M}_t$ is defined as follows:

$$\begin{aligned} \mathcal{M}_t = \{ \mathbf{M} = (\mathcal{X}, \mathcal{A}, P, r) : \forall x \in \mathcal{X}, \forall a \in \mathcal{A}, \;\; & |\hat{r}_t(x,a) - r(x,a)| \le C_R \\ & \text{and } d(\hat{P}_t(.; x, a), P(.; x, a)) \le C_P \} , \end{aligned}$$

where C_P and C_R are fixed constants and d measures the difference between the transition probabilities.

In contrast to UCRL2, which uses the L^1-distance for d, we propose to rely on the Kullback-Leibler divergence, as in the seminal article [7]; however, contrary to the approach of [7], no prior knowledge on the state structure of the MDP is needed. Recall that the Kullback-Leibler divergence is defined for all n-dimensional probability vectors p and q by $KL(p,q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$ (with the convention that $0 \log 0 = 0$). In the sequel, we will show that this choice dramatically alters the behavior of the algorithm and leads to significantly better

performance, while causing a limited increase of complexity; in Section 6, the advantages of using a KL-divergence instead of the L^1-norm are illustrated and argumented.

3.1 The KL-UCRL algorithm

The KL-UCRL, described below, is a variant of the efficient model-based algorithm UCRL2, introduced by [2] and extended to more general MDPs by [4]. The key step of the algorithm, the search for the optimistic model (Step 12), is detailed below as Algorithm 2.

Algorithm 1 KL-UCRL

1: Initialization: $j = 0$; $\forall a \in \mathcal{A}, \forall x \in \mathcal{X}, n_j(x,a) = 0$; initial policy π_0.
2: **for** all $t \geq 1$ **do**
3: Observe X_t.
4: **if** $n_j(X_t, \pi_j(X_t)) < \max(N_{t_j}(X_t, \pi_j(X_t)), 1)$ **then**
5: *Follow the current policy:*
6: Choose action $A_t = \pi_j(X_t)$ and receive reward R_t.
7: Update the count: $n_j(X_t, A_t) = n_j(X_t, A_t) + 1$.
8: **else**
9: *Begin a new episode:* $j = j + 1$
10: Reinitialize: $\forall a \in \mathcal{A}, \forall x \in \mathcal{X}\ ,\ n_j(x,a) = 0$
11: Estimate the model $\hat{\mathbf{M}}_t = (\mathcal{X}, \mathcal{A}, \hat{P}_t, \hat{r}_t)$ according to (1).
12: Find the optimistic model $\mathbf{M}_j \in \mathcal{M}_t$ and the related policy π_j.
13: Choose action $A_t = \pi_j(X_t)$ and receive reward R_t.
14: Update the count: $n_j(X_t, A_t) = n_j(X_t, A_t) + 1$.
15: **end if**
16: **end for**

The KL-UCRL algorithm proceeds in episodes. Let t_j be the starting time of episode j; the length of the j-th episode depends on the number of visits $N_{t_j}(x,a)$ to the state-action pair (x,a) before t_j compared to the number of visits $n_j(x,a)$ to the same pair during the j-th episode. More precisely, an episode finishes as soon as $n_j(x,a) < N_{t_j}(x,a)$ for some state-action pair (x,a). The policy π_j, followed during the j-th episode, is a near optimal policy related to the optimistic MDP $\mathbf{M}_j = (\mathcal{X}, \mathcal{A}, P_j, r_j) \in \mathcal{M}_{t_j}$ which is computed solving the *extended optimality equations*:

$$\forall x \in \mathcal{X}\ ,\ h^*(x) + \rho^* = \max_{P,r} \max_{a \in \mathcal{A}} \left(r(x,a) + \sum_{x' \in \mathcal{X}} P(x';x,a) h^*(x') \right) \tag{2}$$

$$\text{such that } \forall x \in \mathcal{X}, \forall a \in \mathcal{A}, \quad KL(\hat{P}_{t_j}(.;x,a), P(.;x,a)) \leq C_P(x,a,t_j)$$

$$\forall x \in \mathcal{X}, \forall a \in \mathcal{A}, \quad |\hat{r}_{t_j}(x,a), r(x,a))| \leq C_R(x,a,t_j)\ .$$

The transition matrix P_j and the mean reward r_j of the optimistic MDP $\mathbf{M}_j$ maximizes those equations. Remark that the diameter C_P (resp. C_R) of the

neighborhood around the estimated transition probability $\hat{P}_{t_j}(.;x,a)$ (resp. the mean reward $\hat{r}_{t_j}(x,a)$) depends on the state action pair (x,a) and on t_j. The *extended value iteration* algorithm may be used to approximately solve the fixed point equation (2) [15,2].

3.2 Maximization of a linear function on a KL-ball

At each step of the extended value iteration algorithm, the maximization problem (2) has to be solved. Remark that, for every action a, the maximization in $r(x,a)$ is obviously solved taking $r(x,a) = \hat{r}_{t_j}(x,a) + C_R(x,a,t_j)$, so that the main difficulty lies in maximizing the dot product between the probability vector $q = P(.;x,a)$ and the *value vector* $V = h^*$ over a KL-ball around the fixed probability vector $p = \hat{P}_{t_j}(.;x,a)$:

$$\max_{q \in \mathbb{S}^{|\mathcal{X}|}} V'q \quad \text{s.t.} \quad KL(p,q) \leq \epsilon \,, \tag{3}$$

where the constant $0 < \epsilon < 1$ controls the size of the confidence ball[1] and $\mathbb{S}^n$ denotes the set of n-dimensional probability vectors. This maximization of a linear function under convex constraints is solved explicitly in Appendix A; the resulting algorithm is presented below. It relies on the function f (that depends on the parameter V), defined for all $\nu \geq \max_{i \in \bar{Z}} V_i$ with $\bar{Z} = \{i, p_i > 0\}$, by

$$f(\nu) = \sum_{i \in \bar{Z}} p_i \log(\nu - V_i) + \log\left(\sum_{i \in \bar{Z}} \frac{p_i}{\nu - V_i}\right) \,. \tag{4}$$

Algorithm 2 Function MaxKL

Require: A value function V, a probability vector p, a constant ϵ
Ensure: A probability vector q that maximizes (3)
Let $Z = \{i, p_i = 0\}$ and $\bar{Z} = \{i, p_i > 0\}$. Let $I^* = Z \cap \operatorname{argmax}_i V_i$
if $f(V_i) < \epsilon$ for $i \in I^*$ **then**
  Let $\nu = V_j$ and $r = 1 - \exp(f(\nu) - \epsilon)$.
  For all $i \in I^*$, assign values of q_i such that $\sum_{i \in I^*} q_i = r$.
  For all $i \in Z/I^*$, let $q_i = 0$.
else
  For all $i \in Z$, let $q_i = 0$. Let $r = 0$.
  Find ν solution of the equation $f(\nu) = \epsilon$ using Newton's method.
end if
For all $i \in \bar{Z}$, let $q_i = \frac{\tilde{q}_i}{r + \sum_{i \in \bar{Z}} \tilde{q}_i}$ where $\tilde{q}_i = \frac{p_i}{\nu - V_i}$.

If the estimated transition to a state with a potentially highest value V_{i^*} is equal to 0, there are two options: first, one may improve $V'q$ by taking $q_{i^*} > 0$;

[1] Here and in the sequel, V' denotes the transpose of V.

second, one may rather choose to add probability to more likely transitions. The dilemma's solution depends both on the relative value of V_{i^*} compared to other components of V and on the exploration bonus: namely, $f(V_{i^*})$ is compared to the diameter of the neighborhood ϵ, and the decision to abandon this transition or not is taken accordingly.

In practice, f being a convex positive decreasing function (see Appendix B), Newton's method can be applied to find ν such that $f(\nu) = \epsilon$ (in Step 10 of the algorithm), so that numerically solving (3) is a matter of a few iterations. Appendix B contains a discussion on the initialization of Newton's algorithm based on asymptotic arguments.

4 Regret bounds

To analyze the performance of KL-UCRL, we compare the rewards obtained following the algorithm with the rewards that would be obtained, on average, by an agent who knows an optimal policy. The *regret* of the algorithm after T steps is defined as [1]:

$$\mathrm{Regret}_T = \sum_{t=1}^{T} \rho^*(\mathbf{M}) - R_t \ .$$

We adapt the regret bound analysis of the UCRL2 algorithm to the use of KL-neighborhoods, and obtain similar theorems. Let

$$D(\mathbf{M}) = \max_{x,x'} \min_{\pi} \mathbb{E}_{\mathbf{M},\pi}(\tau(x,x')) \ ,$$

where $\tau(x,x')$ is the first random time step in which state x' is reached from the initial state x. The $D(\mathbf{M})$ constant will appear in the regret bounds. For all communicating MDPs $\mathbf{M}$, $D(\mathbf{M})$ is finite. Theorem 1 establishes an upper bound on the regret of the KL-UCRL algorithm with C_P and C_R in (2) defined as

$$C_P(x,a,t,\delta,T) = \frac{|\mathcal{X}|\left(B + \log(B + 1/\log(T))\left[1 + 1/(B + 1/\log(T))\right]\right)}{\max(N_{t_k}(x,a),1)}$$

where $B = \log\left(\frac{2e|\mathcal{X}|^2|\mathcal{A}|\log(T)}{\delta}\right)$ and $C_R(x,a,t,\delta,T) = \sqrt{\frac{\log(4|\mathcal{X}||\mathcal{A}|\log(T)/\delta)}{1.99\max(N_t(x,a),1)}}$.

Theorem 1. *With probability $1-\delta$, it holds that for a large enough $T > 1$, the regret of KL-UCRL is bounded by*

$$\mathit{Regret}_T \leq CD(\mathbf{M})|\mathcal{X}|\sqrt{|\mathcal{A}|T\log(\log(T)/\delta)} \ ,$$

for a constant C that does not dependent on the model.

It is also possible to prove a logarithmic upper bound for the expected regret. This bound, presented in Theorem 2, depends on the model through another constant $\Delta(\mathbf{M})$ defined as $\Delta(\mathbf{M}) = \rho^*(\mathbf{M}) - \max_{\pi,\rho^\pi(\mathbf{M})<\rho^*(\mathbf{M})} \rho^\pi(\mathbf{M})$.

Theorem 2. *For a large enough horizon $T > 1$, the expected regret of KL-UCRL is bounded by*

$$\mathbb{E}(Regret_T) \leq CD(\mathbf{M})^2 \frac{|\mathcal{X}|^2|\mathcal{A}| \log(T)}{\Delta(\mathbf{M})} ,$$

for a constant C independent of the model.

Elements of proof The proof of Theorem 1 is analogous to the ones in [2,4]. Due to the lack of space, we refer to [2] for the notations and we focus only on the original steps. First, the following proposition enables us to ensure that, with high probability, the true model $\mathbf{M} = (|\mathcal{X}|, |\mathcal{A}|, P, r)$ belongs to the set of models $\mathcal{M}_t$ at each time step.

Proposition 1. *For every $T \geq 1$ and $\delta > 0$, $\mathbb{P}\left(\forall t \leq T\ ,\ \mathbf{M} \in \mathcal{M}_t\right) \geq 1 - 2\delta$.*

The proof relies on the two following concentration inequalities due to [10,9]: for all $x \in \mathcal{X}$, $a \in \mathcal{A}$, any $\epsilon_P > 0$, and $\epsilon_R > 0$, it holds that

$$\mathbb{P}\left(\forall t \leq T,\ KL(\hat{P}_t(.;x,a), P(.;x,a)) > \frac{\epsilon_P}{N_t(x,a)}\right) \leq 2e(\epsilon_P \log(T) + |\mathcal{X}|)e^{-\frac{\epsilon_P}{|\mathcal{X}|}}$$

$$\mathbb{P}\left(\forall t \leq T, \quad |\hat{r}_t(x,a) - r(x,a)| \leq \frac{\epsilon_R}{\sqrt{N_t(x,a)}}\right) \leq 4\log(T)e^{-1.99\epsilon_R} . \tag{5}$$

Then, taking $\epsilon_P = N_t(x,a)C_P(x,a,t,\delta,T)$ and $\epsilon_R = \sqrt{N_t(x,a)}C_R(x,a,t,\delta,T)$ and summing over all state-action pairs, Proposition 1 follows.

As shown in [2], with high probability the regret at time T can be written as the sum of the regret in each of the $m(T)$ episodes plus an additional term $C_e(T,\delta)$. Let P_k and π_k denote, respectively, the transition probability matrix of the optimistic model and the optimal policy in the k-th episode ($1 \leq k \leq m(T)$). It is easy to show that (see [4] for details), with probability $1-\delta$,

$$\begin{aligned}\text{Regret}_T \leq \sum_{k=1}^{m(T)} \sum_{x\in\mathcal{X}} n_k(x,\pi_k(x)) \big[&(P_k(.;x,\pi_k(x)) - P(.;x,\pi_k(x)))' h_k \\ &+ (P(.;x,\pi_k(x)) - \mathbf{e}_x)' h_k + (r_k(x,\pi_k(x)) - r(x,\pi_k(x)))\big] + C_e(T,\delta) ,\end{aligned}$$

where h_k is a bias vector, $\mathbf{e}_x(y) = 1$ if $x = y$ and $\mathbf{e}_x(y) = 0$ otherwise. The geometry of the neighborhood around the estimated transition probabilities only plays a role to bound the first term of the above sum: using Cauchy-Schwartz

and Pinsker inequalities, we have

$$\begin{aligned}
&\sum_{k=1}^{m(T)} \sum_{x\in\mathcal{X}} n_k(x,\pi_k(x))(P_k(.;x,\pi_k(x)) - P(.;x,\pi_k(x)))' h_k \\
&\quad \leq 2D\sqrt{2} \sum_{k=1}^{m(T)} \sum_{x\in\mathcal{X}} n_k(x,\pi_k(x)) \sqrt{C_P(x,\pi_k(x),t_k,\delta,T)} \\
&\quad \leq CD\sqrt{|\mathcal{X}|\log(\log(T)/\delta)} \sum_{k=1}^{m(T)} \sum_{x\in\mathcal{X}} \frac{n_k(x,\pi_k(x))}{\sqrt{N_{t_k}(x,\pi_k(x))}} \,, \qquad (6)
\end{aligned}$$

for some constant C. The other terms are upper-bounded as in [2]. Remarking that, for T large enough, (6) dominates the remaining terms in the upper-bound of the regret and using the fact that $\sum_{k=1}^{m(T)} \sum_x \frac{n_k(x,\pi_k(x))}{\sqrt{N_{t_k}(x,\pi_k(x))}} \leq \sqrt{|\mathcal{X}||\mathcal{A}|T}$ (see Appendix B.1 of [2]), Theorem 1 follows. The proof of Theorem 2 follows from Theorem 1 using the same arguments as in the proof of Theorem 4 in [2].

5 Simulations

To compare the behaviors of algorithms KL-UCRL and UCRL2, we consider the benchmark environment *RiverSwim* proposed by [16]. It consists of six states.

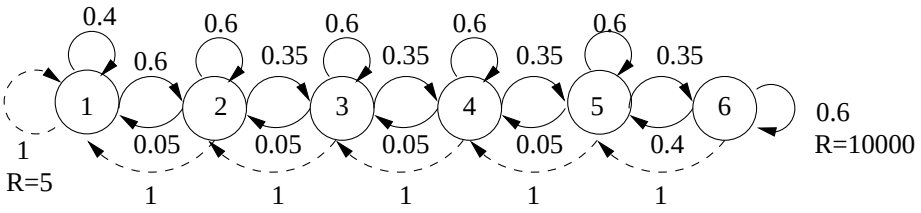

Fig. 1. *RiverSwim* Transition Model: the continuous (resp. dotted) arrows represent the transitions if action 1 (resp. 2) has been chosen.

The agent starts from the left side of the row and, in each state, can either swim left or right. Swimming to the right (against the current of the river) is successful with probability 0.35; it leaves the agent in the same state with a high probability equal to 0.6, and leads him to the left with probability 0.05 (see Figure 1). On the contrary, swimming to the left (with the current) is always successful. The agent receives a small reward when he reaches the leftmost state, and a much larger reward when reaching the rightmost state – the other states offer no reward. This MDP requires efficient exploration procedures, since the agent, having no prior idea of the rewards, has to reach the right side to discover which is the most valuable state-action pair.

We compare the performance of the KL-UCRL algorithm to UCRL2 using 20 Monte-Carlo replications. For both algorithms, the constants C_P and C_R are settled to ensure that the upper bounds of the regret of Theorem 1 and Theorem 2 in [1] hold with probability 0.95. We observe in Figure 2 that the KL-UCRL algorithm accomplishes a smaller average regret than the UCRL2 algorithm. Indeed, in this environment, it is crucial for the agent to quickly learn that there is no possible transition between one of the first four states and the sixth state whose reward is the highest.

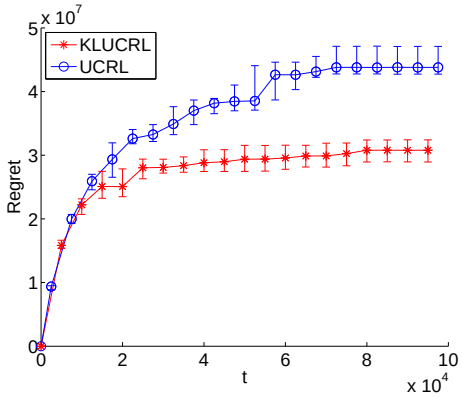

Fig. 2. Comparison of the regret of the UCRL2 and KL-UCRL algorithms.

6 Discussion

In this section, we expose the advantages of using a confidence ball based on the Kullback-Leibler divergence rather than an L^1-ball, as proposed for instance in [2,18], in the computation of the optimistic policy that solves Equation (2). This discussion aims at explaining and interpreting the difference of performance that can be observed in simulations. In KL-UCRL, optimism reduces to maximizing the linear function $V'q$ over a KL-ball (see (3)), whereas the other algorithms make use of a L^1-ball:

$$\max_{q \in \mathbb{S}^{|\mathcal{X}|}} V'q \quad \text{s.t.} \quad \|p - q\|_1 \leq \epsilon' \,. \tag{7}$$

Continuity

Consider an estimated transition probability vector p, and denote by q^{KL} (resp. p^1) the probability vector which maximizes Equation (3) (resp. Equation (7)). It is easily seen that q^{KL} and q^1 lie respectively on the border of the convex set $\{q \in \mathbb{S}^{|X|} : KL(p,q) \leq \epsilon\}$ and at one of the vertices of the polytope $\{q \in$

$\mathbb{S}^{|X|} : \|p - q\|_1 \leq \epsilon'\}$. A first noteworthy difference between those neighborhoods is that, due to the smoothness of the KL-neighborhood, q^{KL} is continuous with respect to the vector V, which is not the case for q_1.

To illustrate this, Figure 3 displays L^1- and KL-balls around 3-dimensional probability vectors. The set of 3-dimensional probability vectors is represented by a triangle whose vertices are the vectors $(1, 0, 0)'$, $(0, 1, 0)'$ and $(0, 0, 1)'$, the probability vector p by a white star, and the vectors q^{KL} and q^1 by a white point. The arrow represents the direction of V's projection on the simplex and indicates the gradient of the linear function to maximize. The maximizer q^1 can vary significantly for small changes of the value function, while q^{KL} varies continuously.

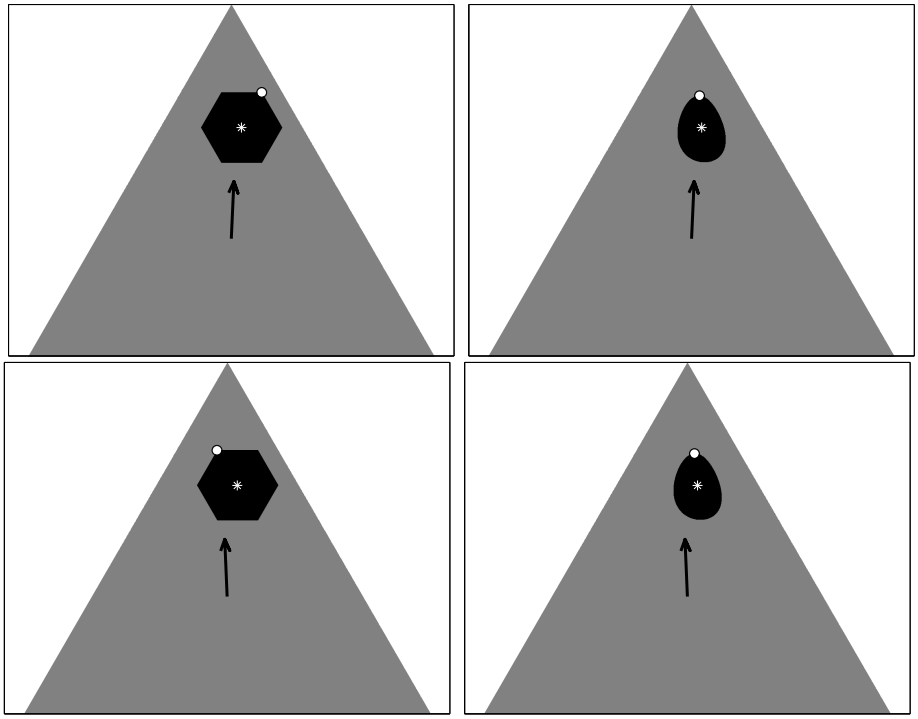

Fig. 3. The L^1-neighborhood $\{q \in \mathbb{S}^3 : \|p - q\|_1 \leq 0.2\}$ (left) and KL-neighborhood $\{q \in \mathbb{S}^3 : KL(p, q) \leq 0.02\}$ (right) around the probability vector $p = (0.15, 0.2, 0.65)'$ (white star). The white points are the maximizers of equations (3) and (7) with $V = (0, 0.05, 1)'$ (up) and $V = (0, -0.05, 1)'$ (down).

Unlikely transitions

Denote $i_m = \operatorname{argmin}_j V_j$ and $i_M = \operatorname{argmax}_j V_j$. As underlined by [2], $q^1(i_m) = \max(p(i_m) - \epsilon'/2, 0)$ and $q^1(i_M) = \min(p^1(i_M) + \epsilon'/2, 1)$. This has two consequences:

1. if p is such that $0 < p(i_m) < \epsilon'/2$, then the vector $q^1(i_m) = 0$; so the optimistic model may assign a probability equal to zero to a transition that has actually been observed, which makes it hardly compatible with the optimism principle. Indeed, an optimistic MDP should not forbid transitions that really exists, even if they lead to states with small values;
2. if p is such that $p(i_M) = 0$, then $q^1(i_M)$ never equals 0; therefore, an optimistic algorithm that uses L^1-balls will always assign positive probability to transitions to i_M even if this transition is impossible under the true MDP and if much evidence has been accumulated against the existence of such a transition. Thus, the exploration bonus of the optimistic procedure is wasted, whereas it could be used more efficiently to favor some other transitions. This observation extends the criticism by R. Ortner that optimistic models should be *refutable*, see [14], in a slightly different context.

This explains a great part of the experimental advantage of KL-UCRL observed in the simulations. Indeed, q^{KL} always assigns strictly positive probability on observed transitions, and eventually renounces to unobserved transitions even if the target states have a potentially large value. Indeed, Algorithm 2 works as follows: for all i such that $p(i) \neq 0$, $q(i) \neq 0$; for all i such that $p(i) = 0$, $q(i) = 0$ except if $p(i_M) = 0$ and if $f(V_{i_M}) < \epsilon$, in which case $q(i_M) = 1 - \exp(f(V_{i_M}) - \epsilon)$. But this is no longer the case when ϵ becomes small enough, that is, when sufficiently many observations are available. We illustrate those two important differences in Figure 4, by representing the L^1 and KL neighborhoods together with the maximizers q^{KL} and q^1, first if $p(i_m)$ is positive by very small, and second if $p(i_M)$ is equal to 0. Figure 5 also illustrates the latter case, by representing the evolution of the probability vector q that maximizes both (7) and (3) for an example with $p = (0.3, 0.7, 0)'$, $V = (1, 2, 3)'$ and ϵ decreasing from 1/2 to 1/200.

A Appendix: Linear optimization over a KL-ball

This section explains how to solve the optimization problem of Equation (3). In [13], a very similar problem arises in a different context, and a somewhat different solution is proposed for the case when the p_i are all positive. As a problem of maximizing a linear function under convex constraints, it is sufficient to consider the Lagrangian function

$$L(q, \lambda, \nu, \mu_1, \ldots, \mu_N) = \sum_{i=1}^{N} q_i V_i - \lambda \left(\sum_{i=1}^{N} p_i \log \frac{p_i}{q_i} - \epsilon \right) - \nu \left(\sum_{i=1}^{N} q_i - 1 \right) + \sum_{i=1}^{N} \mu_i q_i \ .$$

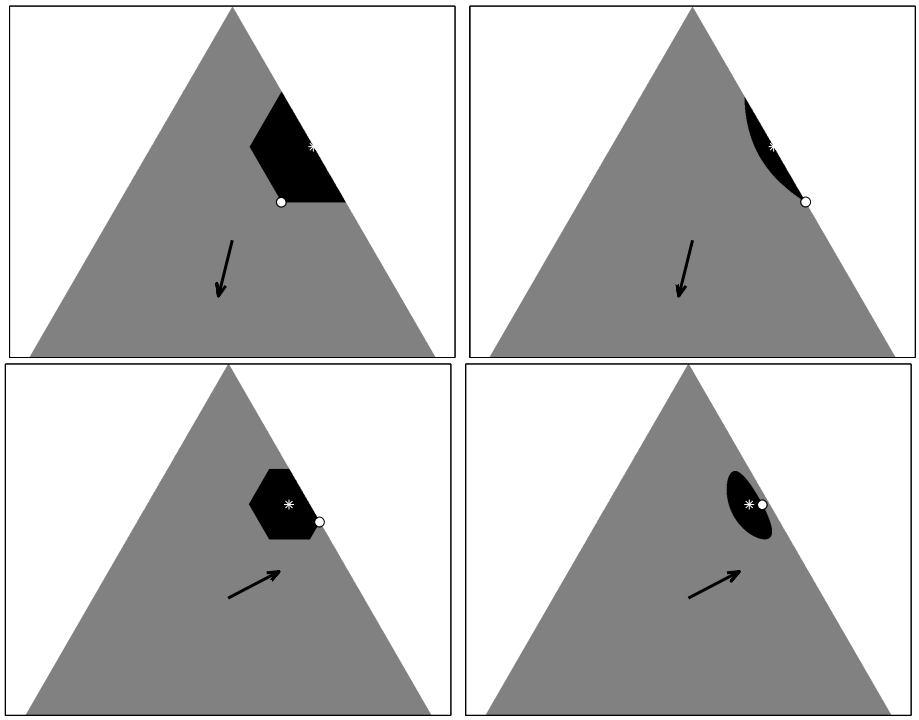

Fig. 4. The L^1 (left) and KL-neighborhoods (right) around the probability vector $p = (0, 0.4, 0.6)'$ (up) and $p = (0.05, 0.35, 0.6)'$ (down). The white point is the maximizer of the equations (3) and (7) with $V = (-1, -2, -5)'$ (up) and $V = (-1, 0.05, 0)'$ (down). We took, $\epsilon = 0.05$ (up), $\epsilon = 0.02$(down) and $\epsilon' = \sqrt{2\epsilon}$.

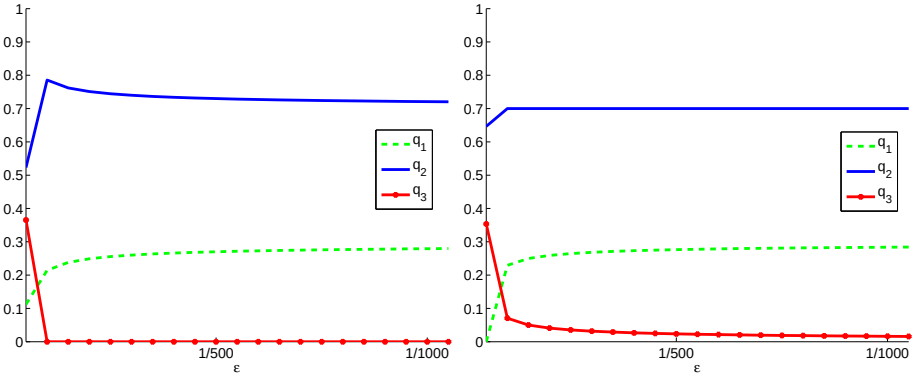

Fig. 5. Evolution of the probability vector q that maximizes both (3) (left) and (7) (right) with $p = (0.3, 0.7, 0)'$, $V = (1, 2, 3)'$ and ϵ decreasing from $1/2$ to $1/200$

If q is a maximizer, there exist $\lambda \in \mathbb{R}$, $\nu, \mu_i \geq 0$ $(i = 1 \dots N)$ such that the following conditions are simultaneously satisfied:

$$\begin{cases} V_i + \lambda \dfrac{p_i}{q_i} - \nu + \mu_i = 0 & (8) \\ \lambda \left(\displaystyle\sum_{i=1}^{N} p_i \log \frac{p_i}{q_i} - \epsilon \right) = 0 & (9) \\ \nu \left(\displaystyle\sum_{i=1}^{N} q_i - 1 \right) = 0 & (10) \\ \mu_i q_i = 0 & (11) \end{cases}$$

Let $Z = \{i, p_i = 0\}$. Conditions (8) to (11) imply that $\lambda \neq 0$ and $\nu \neq 0$. For $i \in \bar{Z}$, Equation (8) implies that $q_i = \lambda \frac{p_i}{\nu - \mu_i - V_i}$. Since $\lambda \neq 0$, $q_i > 0$ and then, according to (11), $\mu_i = 0$. Therefore,

$$\forall i \in \bar{Z} \ , \quad q_i = \lambda \frac{p_i}{\nu - V_i} \ . \tag{12}$$

Let $r = \sum_{i \in Z} q_i$. Summing on $i \in \bar{Z}$ and using equations (12) and (10), we have

$$\lambda \sum_{i \in \bar{Z}} \frac{p_i}{\nu - V_i} = \sum_{i \in \bar{Z}} q_i = 1 - r \ . \tag{13}$$

Using (12) and (13), we can write $\sum_{i \in \bar{Z}} p_i \log \frac{p_i}{q_i} = f(\nu) - \log(1 - r)$ where f is defined in (4). Then, q satisfies condition (9) if and only if

$$f(\nu) = \epsilon + \log(1 - r) \ . \tag{14}$$

Consider now the case where $i \in Z$. Let $I^* = Z \cap \operatorname{argmax}_i V_i$. Note that, for all $i \in Z / \{I^*\}$, $q_i = 0$. Indeed, otherwise, μ_i should be zero, and then $\nu = V_i$ according to (8), which involves a possible negative denominator in (12). According to (11), for all $i \in I^*$, either $q_i = 0$ or $\mu_i = 0$. The second case implies that $\nu = V_i$ and $r > 0$ which requires that $f(\nu) < \epsilon$ so that (14) can be checked with $r > 0$. Therefore,

- if $f(V_i) < \epsilon$ for $i \in I^*$, then $\nu = V_i$ and the constant r can be computed solving equation $f(\nu) = \epsilon - \log(1 - r)$; the values of q_i for $i \in I^*$ may be chosen in any way such that $\sum_{i \in I^*} q_i = r$;
- if for all $i \in I^*$ $f(V_i) \geq \epsilon$, then $r = 0$, $q_i = 0$ for all $i \in Z$ and ν is the solution of the equation $f(\nu) = \epsilon$.

Once ν and r have been determined, the other components of q can be computed, according to (12): we have that for $i \in \bar{Z}$, $q_i = \frac{\tilde{q}_i}{r + \sum_{i \in \bar{Z}} \tilde{q}_i}$ where $\tilde{q}_i = \frac{p_i}{\nu - V_i}$.

B Appendix: Properties of the f function

In this section, a few properties of function f defined in Equation (4) are stated, as it plays a key role in the maximizing procedure of Section 3.2.

Proposition 2. *f is a convex, decreasing mapping from* $]\max_{i\in\bar{Z}} V_i;\infty[$ *onto* $]0;\infty[$.

Proof. Using Jensen's inequality, it is easily shown that function the f decreases from $+\infty$ to 0. The second derivative of f with respect to ν satisfies

$$f''(\nu) = -\sum_i \frac{p_i}{(\nu - V_i)^2} + \frac{2\sum_i \frac{p_i}{(\nu-V_i)^3}\sum_i \frac{p_i}{\nu-V_i} - \left(\sum_i \frac{p_i}{(\nu-V_i)^2}\right)^2}{\left(\sum_i \frac{p_i}{\nu-V_i}\right)^2} .$$

If Z denotes a positive random value such that $\mathbb{P}\left(Z = \frac{1}{\nu-V_i}\right) = p_i$, then

$$f''(\nu) = \frac{2\mathbb{E}(Z^3)\mathbb{E}(Z) - \mathbb{E}(Z^2)\mathbb{E}(Z)^2 - \mathbb{E}(Z^2)^2}{\mathbb{E}(Z)^2} .$$

Using Cauchy-Schwartz inequality, we have $\mathbb{E}(Z^2)^2 = \mathbb{E}(Z^{3/2}Z^{1/2})^2 \leq \mathbb{E}(Z^3)\mathbb{E}(Z)$. In addition $\mathbb{E}(Z^2)^2 \geq \mathbb{E}(Z^2)\mathbb{E}(Z)^2$. These two inequalities show that $f''(\nu) \geq 0$.

As mentioned in Section 3.2, Newton's method can be applied to solve the equation $f(\nu) = \epsilon$ for a fixed value of ϵ. When ϵ is close to 0, the solution of this equation is quite large and an appropriate initialization accelerates convergence. Using a second-order Taylor's-series approximation of the function f, it can be seen that, for ν near ∞, $f(\nu) = \frac{\sigma_{p,V}}{2\nu^2} + o(\frac{1}{\nu^2})$, where $\sigma_{p,V} = \sum_i p_i V_i^2 - (\sum_i p_i V_i)^2$. The Newton iterations can thus be initialized by taking $\nu_0 = \sqrt{\sigma_{p,V}/(2\epsilon)}$.

References

1. Auer, P., Jaksch, T., Ortner, R.: Near-optimal regret bounds for rein- forcement learning (full version). Tech. rep., URL : http://institute.unileoben.ac.at/infotech/publications/ ucrl2.pdf. (2009)
2. Auer, P., Jaksch, T., Ortner, R.: Near-optimal regret bounds for reinforcement learning. In: Advances in Neural Information Processing Systems. vol. 21 (2009)
3. Auer, P., Ortner, R.: Logarithmic online regret bounds for undiscounted reinforcement learning. Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference p. 49 (2007)
4. Bartlett, P., Tewari, A.: REGAL: A Regularization based Algorithm for Reinforcement Learning in Weakly Communicating MDPs. In: Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (2009)
5. Bertsekas, D.: Dynamic Programming and Optimal Control, Two Volume Set. Athena Scientific (1995)
6. Brafman, R., Tennenholtz, M.: R-max-a general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research 3, 213–231 (2003)
7. Burnetas, A., Katehakis, M.: Optimal adaptive policies for Markov decision processes. Mathematics of Operations Research pp. 222–255 (1997)
8. Even-Dar, E., Mannor, S., Mansour, Y.: Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research 7, 1079–1105 (2006)

9. Garivier, A., Moulines, E.: On upper-confidence bound policies for non-stationary bandit problems. Arxiv preprint arXiv:0805.3415 (2008)
10. Garvier, A., Leonardi, F.: Context tree selection: A unifying view (2010), preprint
11. Kearns, M., Singh, S.: Near-optimal reinforcement learning in polynomial time. Mach. Learn. 49(2-3), 209–232 (2002)
12. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics 6(1), 4–22 (1985)
13. Nilim, A., El Ghaoui, L.: Robust control of markov decision processes with uncertain transition matrices. Operations Research 53(5), 780–798 (2005)
14. Ortner, R.: Optimism in the face of uncertainty should be refutable. Minds and Machines 18(4), 521–526 (2008)
15. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc. New York, NY, USA (1994)
16. Strehl, A., Littman, M.: An analysis of model-based interval estimation for Markov decision processes. Journal of Computer and System Sciences 74(8), 1309–1331 (2008)
17. Sutton, R.: Reinforcement Learning. Springer (1992)
18. Tewari, A., Bartlett, P.: Optimistic linear programming gives logarithmic regret for irreducible MDPs. Advances in Neural Information Processing Systems 20, 1505–1512 (2008)