Reinforcement Learning and Self-Normalized Deviation Bounds

Aurélien Garivier

NUMEC - USP

Machine Learning

Supervised Learning : given observations $(X_t, Y_t)_t$ where $Y_t = f(X_t) + \varepsilon_t$ and f is the unknown target function, estimate f in order to make predictions

Un-supervised Learning : given observations $(X_t)_t$, find structure in this data (e.g., classes), estimate density, ...

Reinforcement Learning : the data arrives progressively, and decisions must be taken at the same time

General framework of RL

$$\begin{tikzcd}
\text{Agent} \arrow[rr, "\text{Action } A_t"'] & & \text{Envir.} \arrow[ll, bend right, "\text{Observation } S_t"'] \arrow[ll, bend left, "\text{Reward } R_t"]
\end{tikzcd}$$

exploration
|
exploitation
dilemma

- Agent = Actor, not spectator [Sutton '92; Bertsekas '95]
- At each timestep t, he chooses an action $A_t \in A$ depending on past actions and rewards $(S_s, R_s)_{s<t}$ in order to maximize the cumulative reward $\sum_{t=1}^{n} R_t$
- Examples: clinical trial, robotic, advertisement, finance, mobile networks, . . .

History

Emerged at the end of the 1970's from the meeting of

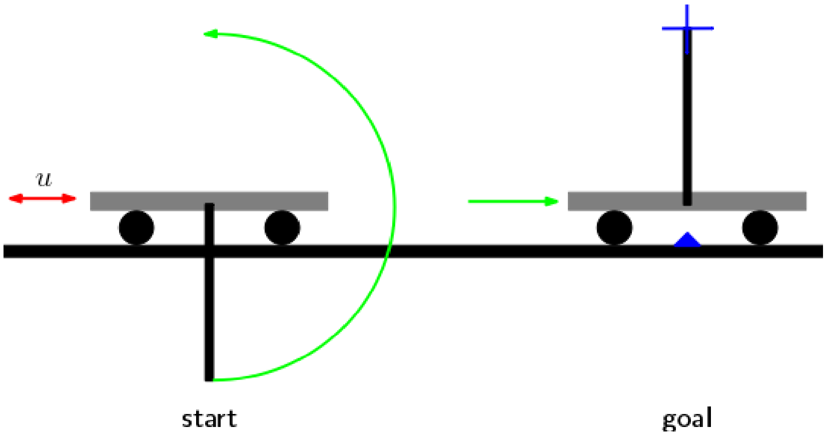
- Computational neuroscience (Hebbs' rule, Rescorla and Wagner models)
- Experimental psychology, models of animal behavior (reinforcement of behaviors leading to satisfaction), behaviorists
- Adequate mathematical framework: Dynamic Programming by Bellman (50', 60'), optimal control

Experimental Psychology

Law of effects (Thorndike, 1911)

> *Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the connection*

Example: inverted pendulum



The learing algorithm used by Martin is Neural Fitted Q iteration

Applications 1/2

- TD-Gammon. [Tesauro 1992-1995]: backgammon player, best in the world
- KnightCap [Baxter et al. 1998]: chess player ('2500 ELO)
- Computer poker (computates the Nash Equilibrium with adversarial bandits), [Alberta, 2008]
- Computer go [Mogo, 2006]
- Robotic: acrobots, ... [Schaal et Atkeson, 1994]

Applications 2/2

- Command of a lift bunch [Crites et Barto, 1996]
- Internet packets routing [Boyan et Littman, 1993]
- Task ordering [Zhang et Dietterich, 1995]
- Machine management [Mahadevan et al., 1997]
- Social Networks [Acemoglu et Ozdaglar, 2010]
- Yield Management, pricing of plane tickets [Gosavi 2010]
- Load prevision for electrical prevision [S. Meynn, 2010]

References

[Puterman ’94] Markov Decision Processes, Discrete Stochastic Dynamic Programming

[Bertsekas ’95] Dynamic Programming and Optimal Control

[Sutton & Barto ’98] Reinforcement Learning

[Sigaud & Buffet ’08] Processus Décisionnels de Markov en Intelligence Artificielle

[Cesa-Bianchi & Lugosi ’06] Prediction, Learning, and Games

Wishful Thinking

Optimistic algorithms: [Lai&Robins '85; Agrawal '95]

Make as if you were in the environment that is most favorable to you, among all those for which the past observations are sufficiently likely

First introduced for bandit problems, recently generalized to general RL problems

Properties

Somewhat unexpectedly, optimistic methods are:

- relevant in very different frameworks
- efficient
- robust
- simple to implement

We show two examples now

Outline

What is Reinforcement Learning?

Bandits problems
- The model
- An Optimistic Method : UCB
- Optimality

MDP : Markov Decision Processes

RL Formulations

- Constant Environnement
- Conditionnally on the actions $(A_t)_{1\leq t\leq n}$, the rewards $(R_t)_{1\leq t\leq n}$ are i.i.d. with expected values μ_{A_t}
- Goal : choose action a^* with highest expected reward :

$$\mu_{a^*} = \max_{a\in A} \mu_a$$

- Measure of performance : *cumulated regret*

$$\text{Regret}(n) = \sum_{t=1}^{n} \mu_{a^*} - \mu_{A_t}$$

Outline

What is Reinforcement Learning?

Bandits problems

The model

An Optimistic Method : UCB

Optimality

MDP : Markov Decision Processes

Upper Confidence Bound (UCB)

- Optimistic algorithm : [Lai&Robins ’85; Agrawal ’95]

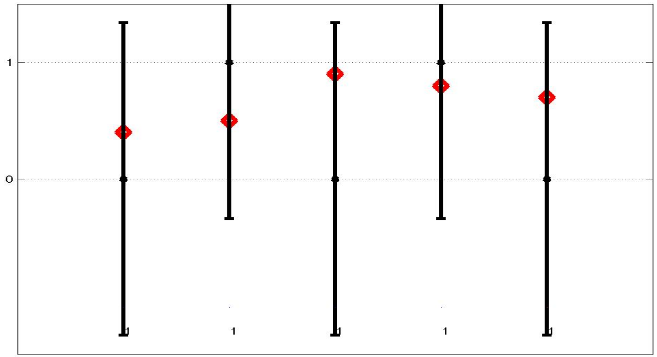
  Make as if you were in the environment that is most favorable to you, among all those for which the past observations are sufficiently likely

- Here : UCB (Upper Confidence Bound) = compute an upper-bound of the expected reward for each action, and choose the most promizing one [Auer&al ’02; Audibert&al ’07]
- Advantage : easily interpretable and acceptable behavior

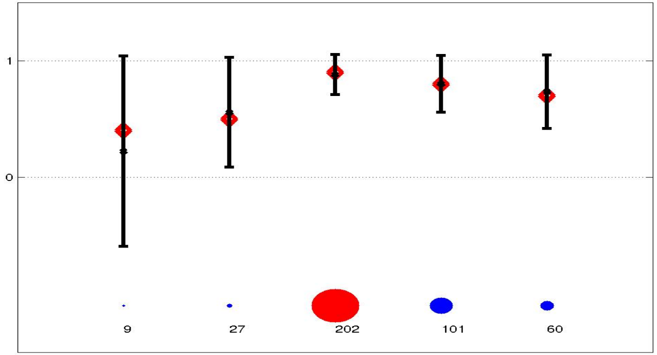
  $\Rightarrow$ the regret grows only a $C \log(n)$, where C depends on

$$\Delta = \min_{\mu_a < \mu_{a^*}} \mu_{a^*} - \mu_a$$

UCB in action



UCB in action



Proof

UCB tends to equalize the upper bounds of the confidence intervals

We upper-bound the number of times a sub-optimal arm is chosen

We work conditionnally on the confidence intervals

A weak arm cannot have an UCB higher than a good arm very often

Outline

What is Reinforcement Learning?

Bandits problems

- The model
- An Optimistic Method : UCB
- Optimality

MDP : Markov Decision Processes

Lai&Robbins Lower Bound

Denote $KL(p_j|p^*)$ the Kullback-Leibler between the reward distributions of the j-th arm and the optimal arm

Theorem: whatever strategy that always plays ”sufficiently often” the best arm, the number of times a sub-optimal arm j is played is bounded in expectation as:

$$\mathbb{E}[T^j(n)] \geq \frac{\log(n)}{KL(p_j|p^*)}$$

Corollary : any strategy as a regret at least $C\log(n)$, where C depends on the arms distributions

Minimax Lower Bound

Theorem: For n and N sufficiently large, there is a bandit problem for which the regret of any strategy is at least

$$\frac{1}{20}\sqrt{Nn}$$

Remark : the maximal regret of UCB can be upper-bounded as

$$C\sqrt{n\log(n)}$$

for a constant C that does *not* depend on the problem.

Outline

What is Reinforcement Learning?

Bandits problems

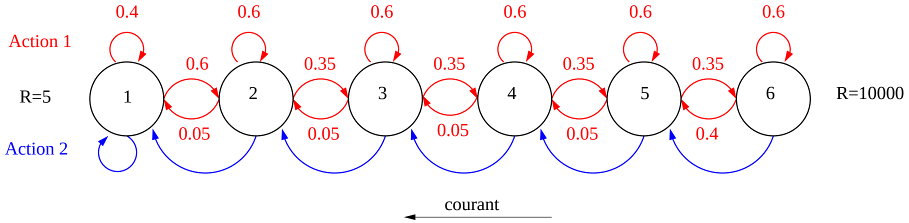
MDP : Markov Decision Processes
- Markov Decision Processes
- Upper Confidence Reinforcement Learning (UCRL)
- Kullback-Leibler URCL (KL-UCRL)
- Discussion

The model 1/2

The state of the system evolves as a Markov Chain controlled by the actions:

$$S_{t+1} \sim P(\cdot; S_t, A_t) \text{ et } R_t = r(S_t, A_t) + \varepsilon_t$$

Exemple / Benchmark : RiverSwim [Strehl&Littman'08]



The model 2/2

[Bellman 1957, Howard 1960, Dubins et Savage 1965, Fleming et Rishel 1975, Bertsekas 1987, Puterman 1994]

MDP = $(\mathcal{S}, \mathcal{A}, p, r)$, where:

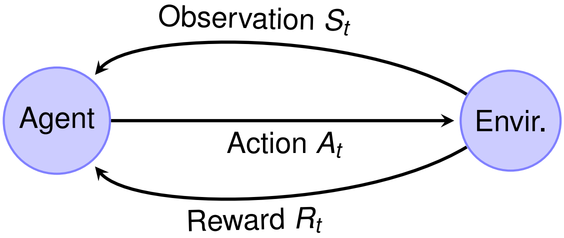
- $\mathcal{S}$ = state space (*finite*, countable, uncountable)
- $\mathcal{A}$ action space (*finite*, countable, uncountable)
- $p(y|x,a)$: transition probability from state $x \in \mathcal{S}$ to $y \in \mathcal{S}$ when action a is chosen:

$$p(y|x,a) = P(X_t + 1 = y | X_t = x, A_t = a)$$

- $r(x,a,y)$: reward obtained during the transition from state x to y, as action a has been chosen

Our Goal: Model-Based Online Reinforcement Learning

Assuming a finite state-space finite action-space Markov Decision Process (MDP)



with unknown

Transition $P(s';s,a) = P(S_{t+1} = s' | S_t = s, A_t = a)$

Reward $r(s,a) = \mathbb{E}(R_t | S_t = s, A_t = a)$

- Implement an on-policy strategy for controlling the agent
- Doing "almost as good" (in terms of cumulated rewards) as an oracle agent that knows the optimal policy

Policy

Politique π = decision rule, determines which action is chosen

Two types of policies:

- *deterministic* : $\pi : \mathcal{S} \to A$
  $\pi(x)$ = action chosen in state x.
- *stochastic* : $\pi : \mathcal{S} \to \mathfrak{M}_1(A)$
  $\pi(a|x)$ = probability of choosing action a in state x.

Policy Value

Finite horizon:

$$V^\pi(x,t) = E_\pi\left[\sum_{s=t}^{T} r(X_s, \pi(X_s)) | X_t = x;\right]$$

Infinite horizon with discounted criterion:

$$V^\pi(x) = E_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(X_s, \pi(X_s)) | X_0 = x;\right]$$

where $\gamma \in]0,1[$ is a discount factor.

Infinite horizon with average criterion:

$$V^\pi(x) = \lim_{T\to\infty} \frac{1}{T} E_\pi\left[\sum_{t=0}^{T-1} r(X_s, \pi(X_s)) | X_0 = x;\right]$$

Optimal policy

- Goal: find policy $\pi : \mathbb{S} \to \mathsf{A}$ with highest *expected cumulated reward* :

$$\rho^\pi = \lim_{n\to\infty} \frac{1}{n} \mathbb{E}^\pi \left[\sum_{t=0}^{n} R_t \right]$$

- Even if the parameters of the MDP are known, finding an optimal policy is not completely obvious: it is called the *planning* problem $\Rightarrow$ Dynamic Programming

Outline

What is Reinforcement Learning?

Bandits problems

MDP : Markov Decision Processes

- Markov Decision Processes
- Upper Confidence Reinforcement Learning (UCRL)
- Kullback-Leibler URCL (KL-UCRL)
- Discussion

Upper Confidence Reinforcement learning

In MDPs, [Auer et al, 07–10; Tewari & Bartlett, 07–08] propose to replace the upper confidence bound of UCB by an optimistic MDP (P^*, r^*) whose average reward $\rho^* = \lim n^{-1} \sum_{t=0}^{n-1} \mathbb{E}_{\pi^*}(R_t)$ and bias vector $h^*(s)$ satisfy an extended version of Bellman's optimality equations

$$\forall s,\ h^*(s) + \rho^* = \max_{P,r \in \mathcal{C}_t^P \times \mathcal{C}_t^r} \max_{a \in A} \left(r(s,a) + \sum_{s' \in \mathbb{S}} P(s'; s, a) h^*(s') \right)$$

$$\forall s,\ \pi^*(s) = \operatorname*{argmax}_{a \in A} \left(r^*(s,a) + \sum_{s' \in \mathbb{S}} P^*(s'; s, a) h^*(s') \right)$$

where $\mathcal{C}_t^R$ and $\mathcal{C}_t^r$ are confidence sets for P and r, respectively

Extended Value Iteration

The bias vector h^* is determined (up to a constant) as the limit of extended value iterations

- While $\operatorname{span}(V_{k+1} - V_k) > \varepsilon$,

$$\forall s \; , \; V_{k+1}(s) = \max_{a \in \mathcal{A}} \left(\max_{r \in \mathcal{C}_t^r} r(s,a) + \max_{P \in \mathcal{C}_t^P} \sum_{s' \in \mathbb{S}} P(s'; s,a) V_k(s') \right)$$

Issues Not Discussed Here

Influence of the termination tolerance ε Ignored in our work, analyzed in [Auer et al, 07–10]

Convergence of extended value iterations Considered (for L^1 neighborhoods) by [Auer et al, 07–10; Tewari & Bartlett, 07–08] and for KL neighborhoods in the discounted case by [Nilim & EL Ghaoui, 05]

Persistence of policies In MDPs it is not possible to continuously change the policy as in MABs. We used the episodic construction of [Auer et al, 07–10] in which the optimistic policy is recomputed at times that approximately follow a geometric progression with ratio 2

Definition of the Confidence Set

[Auer et al, 07–10; Tewari & Bartlett, 07–08] consider rectangular confidence sets of the form

$$\forall (s,a),\ \left\| \hat{P}_t(.;s,a) - P(.;s,a) \right\|_1 \leq \delta_P$$
$$\forall (s,a),\ |\hat{r}_t(s,a) - r(s,a)| \leq \delta_R$$

where $\hat{P}_t(s';s,a) = N_t(s,a)^{-1} \sum_{i=0}^{t-1} \mathbb{1}\{S_{i+1} = s', S_i = s, A_i = a\}$ and $\hat{r}_t(s,a) = N_t(s,a)^{-1} \sum_{i=0}^{t-1} R_i \mathbb{1}\{A_i = a\}$ are the empirical estimates of P and r at time t

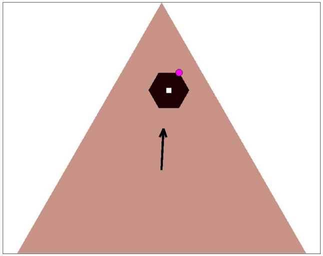
The probabilities of violating the confidence sets are controlled by the Hoeffding inequality for $\hat{r}_t(s,a)$ and by the bound of [Weissman *et al*, 03] for $\hat{P}(.;s,a)$

How Does L^1 Extended Value Iteration Operates?

For each state and action pair, one must solve a problem of the form

$$q^* = \operatorname*{argmax}_{q:\|p-q\|_1\leq\delta} q'V$$

where p is the empirical estimate of the transition probabilities and V is the current estimate of the bias vector



- inflate p_i (if possible) by a total amount of δ for indices i that maximize V_i
- reduce p_i (as much as needed) for indices i where V_i is the smallest

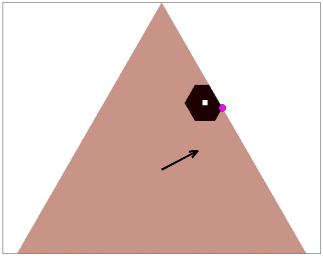
$\Rightarrow$ easy both to implement an interpret, but...

How Does L^1 Extended Value Iteration Operates?

For each state and action pair, one must solve a problem of the form

$$q^* = \operatorname*{argmax}_{q:\|p-q\|_1\le\delta} q'V$$

where p is the empirical estimate of the transition probabilities and V is the current estimate of the bias vector



- inflate p_i (if possible) by a total amount of δ for indices i that maximize V_i
- reduce p_i (as much as needed) for indices i where V_i is the smallest

$\Rightarrow$ easy both to implement an interpret, but...

Outline

What is Reinforcement Learning?

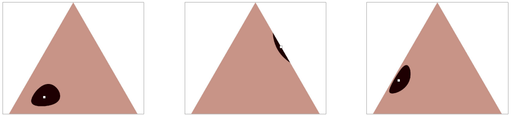
Bandits problems

MDP : Markov Decision Processes

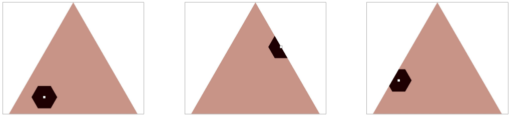
- Markov Decision Processes
- Upper Confidence Reinforcement Learning (UCRL)
- Kullback-Leibler URCL (KL-UCRL)
- Discussion

Our Proposal: Kullback-Leibler URCL

The role played by the KL divergence in large deviations of multinomial experiments suggests that the proper confidence neighborhoods are



rather than

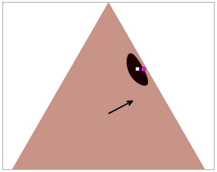


$\Rightarrow$ Use KL rather than L^1 constraints!

Solving KL-Extended Value Maximization

For each state-action pair, one must solve a linear program under KL constraint

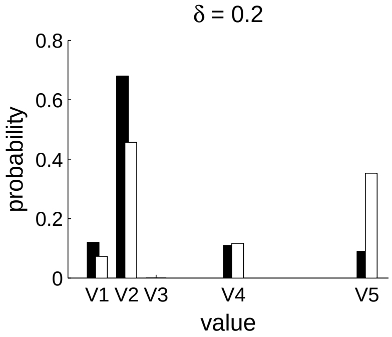
$$q^* = \underset{q:KL(p;q)\leq\delta}{\operatorname{argmax}} \; q'V$$

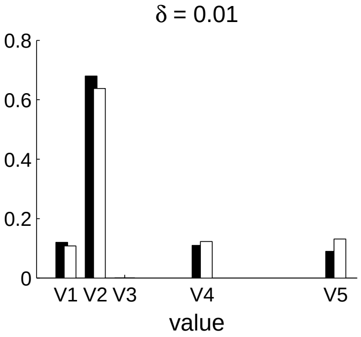


The solution is given by an explicit non-linear transformation of p which is fully controlled by the solution ν to the equation $f(\nu) = \delta$, where f is the one-dimensional decreasing stricly convex function on $(\max_{i:p_i>0} V_i, \infty)$ defined by

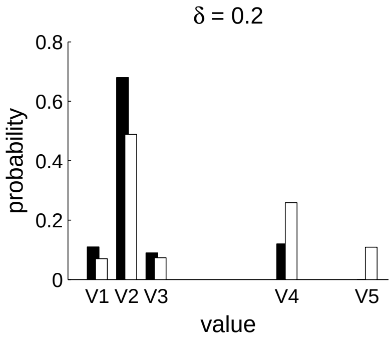
$$f(\nu) = \sum_i p_i \log(\nu - V_i) + \log\left(\sum_i \frac{p_i}{\nu - V_i}\right)$$

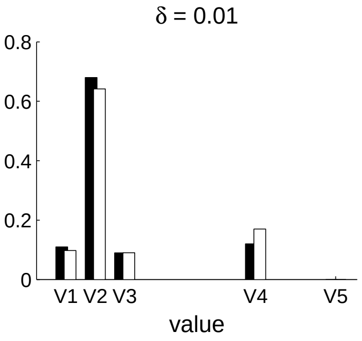
KL-LP's Rule I
"Bigger rewards gets more likely"





KL-LP's Rule II
"You can't get to heaven when δ is too small"





Regret Bound

Adapting the proof of [Auer et al, 07–10] it is possible to show that KL-UCRL achieves logarithmic regret in communicating MDPs (as does UCRL)

Main arguments of the proof

- Pinsker's inequality $\|p - q\|_1 \leq \sqrt{2KL(p; q)}$
- Bound of [Garivier & Leonardi, 10]

$$P\left(\forall t \leq n,\ KL(\hat{p}_t; p) > \frac{\delta}{t}\right) \leq 2e(\delta \log(n) + |\mathbb{S}|)e^{-\delta/|\mathbb{S}|}$$

$\square$

In simulations however (benchmark and random sparsely connected environments), KL-UCRL performs significantly better than UCRL

Outline

What is Reinforcement Learning?

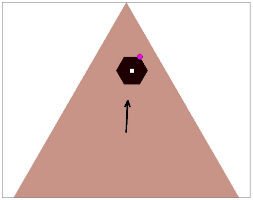
Bandits problems

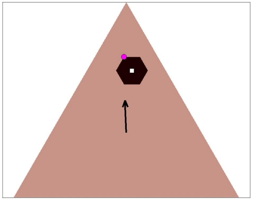
MDP : Markov Decision Processes

- Markov Decision Processes
- Upper Confidence Reinforcement Learning (UCRL)
- Kullback-Leibler URCL (KL-UCRL)
- Discussion

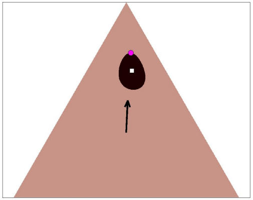
Discussion: Continuity of the optimistic MDP

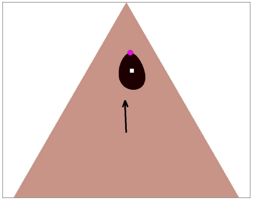
L^1 Neighborhoods





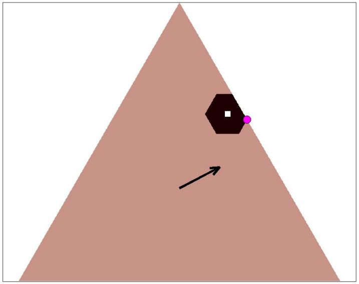
KL Neighborhoods



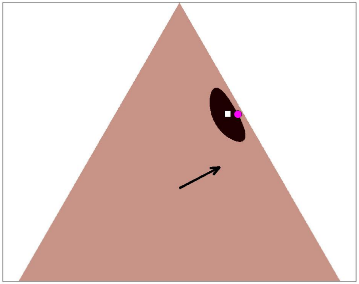


Discussion: Compatibility with observed transitions

L^1 Neighborhoods

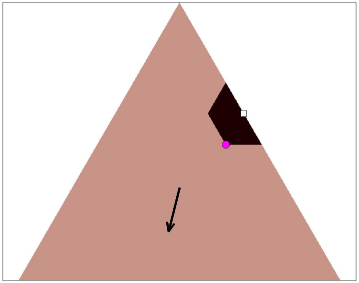


KL Neighborhoods

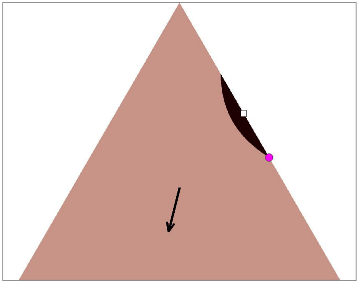


Discussion: Tradeoff between the attraction towards the best state and the statistical evidence that it may not be reachable from all states

L^1 Neighborhoods



KL Neighborhoods



Simulations : RiverSwim

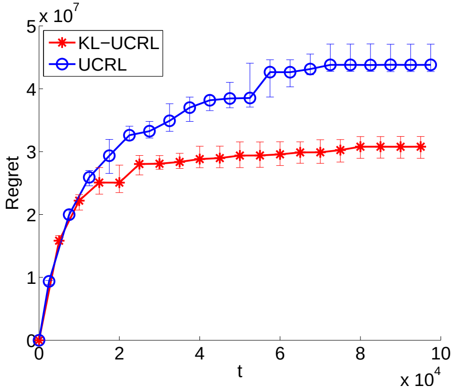


Figure: Regrets for UCRL-2 et KL-UCRL.

Thank you!