Reinforcement Learning

December 2, 2022

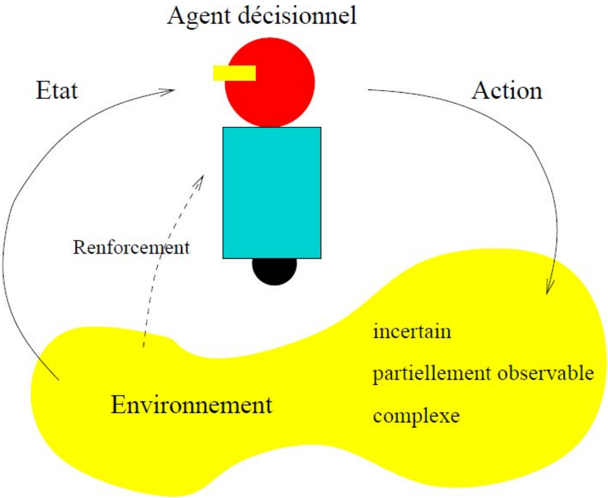
The different types of learning

Supervised learning : based on observed data $(X_t, Y_t)_t$ where $Y_t = f(X_t) + \varepsilon_t$ and f is the target function (unknown), build an empirical version of f in order to make predictions $\hat{f}(x)$

Unsupervised learning : based on observed data $(X_t)_t$, find structures, patterns,... clustering, abnormal regions, ...

Reinforcement : data are observed through time , as decisions are taken by the AI system

General framework for reinforcement learning



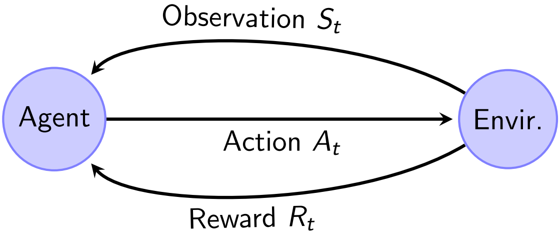
Objectives of Reinforcement Learning

Automated acquisition of skills for decision making (actions or control) in a complex and uncertain environment.

Learn through experience a behavioural strategy (called a policy) dependning on observed failures or successes (reinforcements or rewards).

Examples : sensori-motor learning, games (backgammon, chees, poker, go), robotics, portfolio management, operation research,. . .

RL : first formulation



dilemma
exploration
|
exploitation

- The agent is an actor, not a spectator [Sutton '92; Bertsekas '95]
- At each time t, she/he chooses an action $A_t \in A$ depending on the past observations and rewards $(S_s, R_s)_{s<t}$ in order to maximize the cumulated reward $\sum_{t=1}^{n} R_t$
- Examples: clinical trials, robotics, content recommendation, finance, on-line advertising, yield management, . . .

Historical milestones

Birth: Meeting in the late 70's of

- Computational neurosciences. Reinforcement of synaptic weights in neural transmissions (Hebbs's rule, models of Rescorla and Wagner in the 60's, 70's). Reinforcement = correlations between neural activities.
- and experimental psychology. Models of animal conditioning: reinforcement of the behaviour leading to satisfaction (research originally initiated in 1900 bu Pavlov, Skinner and the behaviorist wave). Reinforcement = satisfaction, pleasure or discomfort, pain.
- Appropriate mathematical frmework: dynamic programming introduced by Bellman (50's, 60's), in optimal control theory. Reinforcement = criterion to be maximized.

Experimental psychology

Law of effect (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 bond"*

A mutlidisciplinary domain

The environment

Deterministic or stochastic (ex: backgammon)

Hostile (ex: chess) or not (ex: Tetris video game)

Partially observable (ex: mobile robotics)

Known or unknown by the the agent

The reinforcement

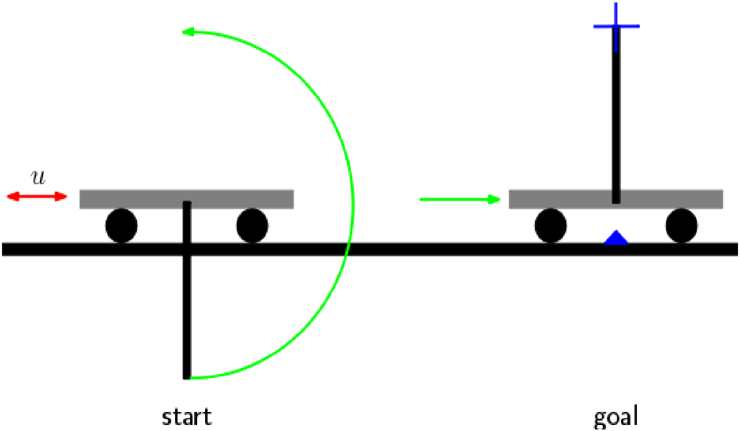
May reward a sequence of actions

$\Longrightarrow$ "credit-assignment" problem : which actions must be accredited to produce reinforcement after a sequence of decisions?

How to sacrifice a small gain in the short term to facilitate a larger gain in the long term?

$\Longrightarrow$ Dilemma exploration / exploitation

Example : inverted pendulum



The learning algorithm used here is Neural Fitted Q iteration, a version de fitted Q-iteration based on neural networks.

Examples 1/2

- TD-Gammon. [Tesauro 1992-1995]: backgammon game. Best player in the world!
- KnightCap [Baxter et al. 1998]: chess game ('2500 ELO)
- Computer poker (Nash equilibrium reached by means of adversarial bandits), [Alberta, 2008]
- Computer go (hierarchical bandits), [Mogo, 2006]
- Robotics: jugglers, balance poles, acrobots, ... [Schaal and Atkeson, 1994]
- Mobile robotics, navigation: robot guide in the Smithonian museum [Thrun et al., 1999]

Examples 2/2

- Managing elevators [Crites et Barto, 1996]
- Paquets routing [Boyan et Littman, 1993]
- Planification [Zhang et Dietterich, 1995]
- Maintenance of machines [Mahadevan et al., 1997]
- Social Networks [Acemoglu et Ozdaglar, 2010]
- Yield Management, pricing of airplane tickets [Gosavi 2010]
- Prediction of electricity consumption [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

Consistency

A strategy is said to be *consistent* if it permits to find, in a finite time, the optimal politicy whatever the problem.

Strength : requires to find *exactly* the solution (and not an approximate solution)

Weakness : one does not control at all what has been lost during the learning stage

PAC bounds

PAC = "Probably Approximately Correct"

The *complexity* of a strategy is, for a given $\varepsilon > 0$, the time required to identify a policy ε-optimal

A strategy is said to be PAC-MDP (Probably Approximately Correct in Markov Decision Processes) if, for all ε and δ, its complexity is bounded by a polynomial in $1/\varepsilon$ and in the parameters of the problem with probability at least $1 - \delta$.

Regret

The regret is defined as the difference between the sum of the rewards obtained by measn of a strategy and the *oracle* reward that would have been accumulated, at he same time, by an agent knowing the optimal policy

In other research works, different variants are considered to facilitate the analysis (average regret, conditional means, etc.)

This measure is more demanding : it takes into the performance of a strategy *starting from the first steps* (no burn-in)

The optimistic paradigm

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

Do as if you were in the most favorable environment among those making the observations likely enough

Introduced first in the context of bandits, and next widely generalised these last few years

Properties

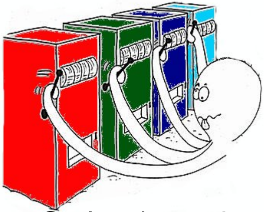
In a rather unexpected fashion, optimistic methods proved to be:

- relevant in many frameworks
- efficicent
- robust
- simple to implement

cf. examples below.

Formulation in Reinforcement Learning

- Static environment
- Conditionally upon the actions $(A_t)_{1\leq t\leq n}$, the rewards $(R_t)_{1\leq t\leq n}$ are i.i.d. with mean μ_{A_t}



- Goal : play action a^* that corresponds to the largest average reward:

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

- Performance measure : *cumulated regret* (in conditional mean)

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

Sequential clinical trials

One considers the following situation:

- patients suffering from a certain disease maladies are diagnosed progressively
- several treatments are a priori at disposal but their efficiency is poorly known for now
- one cures each patient with a treatment, and one observes the result (binary for simplicity)
- *objective :* cure as many patients as possible (and not : know precisely the efficiency of each treatment)

The ” multi-armed bandits” problem

Environment : ensemble of arms $\mathcal{A}$; the choice of arm $a \in \mathcal{A}$ at time t yields the reward

$$X_t = X_{a,t} \sim P_a \in \mathfrak{M}_1(\mathbb{R})$$

and the collection $\left(X_{a,t}\right)_{a\in\mathcal{A}, t\geq 1}$ is independent

Dynamic allocation rule : $\pi = (\pi_1, \pi_2, \dots)$ such that

$$A_t = \pi_t(X_1, \dots, X_{t-1})$$

Number of times one pulled the arm $a \in \mathcal{A}$ at time $t \in \mathbb{N}$:

$$N_a(t) = \sum_{s\leq t} \mathbb{1}\{A_s = b\}$$

Performance, regret

- Cumulated reward : $S_n = X_1 + \cdots + X_n, \quad n \geq 1$
- Objective: choose π so as to maximize

$$E\left[S_n\right] = \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \mathbb{E}\Big[\mathbb{E}\left[X_t \mathbb{1}\{A_t = a\} | X_1, \ldots, X_{t-1}\right]\Big]$$

$$= \sum_{a \in \mathcal{A}} \mu_a \mathbb{E}\left[N_a(n)\right]$$

where $\mu_a = E[P_a]$

- Equivalent objective: minimize the *regret*

$$R_n = n\mu^* - E\left[S_n\right] = \sum_{a : \mu_a < \mu^*} (\mu^* - \mu_a) \mathbb{E}\left[N_a(n)\right]$$

where $\mu^* = \max\left\{\mu_a : a \in \mathcal{A}\right\}$

The ε-Greedy Algorithms

Rationale : one plays the best arm with probability $1 - \varepsilon$, and an arm picked uniformly at random with probability ε

For a well-chosen value (or sequence of values) of ε, the algorithm is consistent and one may prove bounds for the regret

Meanwhile, this is frequently sub-optimal (in general this is a first approach to a problem)

The solution of Gittins

[Gittins '79] Bandit Processes and Dynamic Allocation Indices

Index policy : one associates to each arm an index of performance and one chooses that with largest index

Prefigures optimistic methods for MDP, cf. see later

Algorithm EXP3.P

Cf prediction of individual sequences with loss $M - R_t^a$. Estimation of (unobserved) :

$$\hat{\ell}_t(a) = \frac{M - R_t^a}{p_t(a)} \mathbb{1}_{\{A_t = a\}}$$

estimator *unbiased* of $M - R_t^a$

One then estimes the cumulated losses

$$\hat{L}_t(j, \mathbf{y}_t)) = \sum_{s=1}^{t} \hat{\ell}_t(a))$$

EXP3.P $=$ randomised strategies with choice for the weighting:

$$\hat{p}_t(j) = \frac{\exp(-\beta \hat{L}_{t-1}(j, \mathbf{y}_{t-1}))}{\sum_k \exp(-\beta \hat{L}_{t-1}(k, \mathbf{y}_{t-1}))}$$

Regret bound for EXP3.P

Theorem: By choosing

$$\beta = 1/M\sqrt{\frac{2\log(N)}{nM}}$$

the regret of algorithm EXP3.P compared to the best constant strategy satisfies :

$$\mathbb{E}\left[R_n(\hat{p})\right] \leq M\sqrt{2nN\log(N)}$$

In pratique, très robuste mais peu véloce à se concentrer sur le bon bras quand il y en a un.

Upper Confidence Bound (UCB)

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

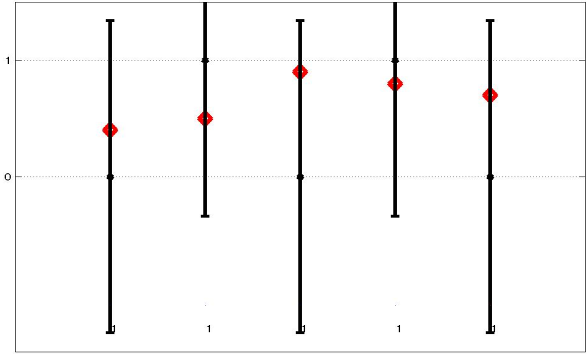
  Do as if you were in the most favourable environment among those making the observations likely enough

- Here : UCB (Upper Confidence Bound) = establish an upper bound for the interest of each action, and choose the most promising one [Auer&al '02; Audibert&al '07]
- Advantage : behaviour easily interpretable and “acceptable”

  ⇒ the regret grows as $C \log(n)$, where C depends on

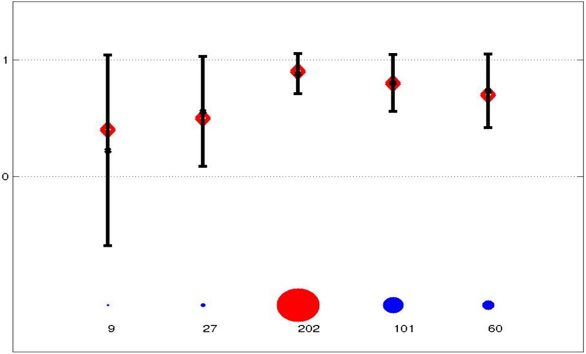
$$\Delta = \min_{\mu_a < \mu_{a^*}} \mu_{a^*} - \mu_a$$

UCB in action



Start

UCB in action



Start

Proof

UCB tends to align the upper bounds of confidence intervals

One bounds the number of times the sub-optimal arm j is pulled

One works conditionally to the confidence intervals

An arm of poor performance cannot have an upper confidence bound lastingly above that of the best arm

How to compute UCB ?

UCB The original version [Auer & al '02] uses the Hoeffding bound:

UCBV takes into account the variance by means of Bernstein inequality:
$\Longrightarrow$ not really satisfactory !

kl-UCB use the *true* Hoeffding bound:

- Be careful : the number of observations is random !

A problem with two bandits

[Lamberton & Pagès & Tarrès '04] When can the two-armed Bandit algorithm be trusted ?

Model : a fund is managed by two traders A and B, who are in charge of a fraction of the portfolio.

One wants to assign as quick as possible the management of the portfolio to the best

Each day, one of the two traders is evaluated (with a probability proportional to the fraction it manages) :

- if she/he performs well, one increases her/his fraction
- otherwise, she/he is not punished (the fraction is left unchanged)

Question : will be the whole portfolio assigned to the best trader ?

Mathematical formulation

Denote by X_n the fraction managed by trader A. One supposes $X_0 = x \in]0,1[$.

Denote by E_n (resp. F_n) the event “trader A (resp. B) outperforms day n”, and one suppose that the events $\{E_n, F_n\}$ are independent.

One denotes by γ_{n+1} the fraction that is managed by B which will be gained by trader A if she/he is evaluated and outperforms day n : one supposes $\gamma_n \in]0,1[$ and $\sum_n \gamma_n = \infty$.

$$X_{n+1} = X_n + \gamma_{n+1}\left((1-X_n)\mathbb{1}_{\{U_{n+1}\leq X_n\}\cap E_{n+1}} - X_n\mathbb{1}_{\{\{U_{n+1}\leq X_n\}\}\cap E_n}\right)$$

where U_n denotes a sequence of i.i.d. r.v.'s uniformly distributed on $[0,1]$.

Results

Theorem :

- X_n converges to $X_\infty \in \{0,1\}$ with probability one.
- if $0 < P(F_n) < P(E_n) \leq 1$, $P(X_\infty = 0)$ can be non zero if γ_n does not decrease fast enough towards 0. In particular, if $\gamma_n = \left(\frac{C}{n+C}\right)^\alpha$:
  - if $0 < \alpha < 1$ or if ($\alpha = 1$ and $C > 1/P(F_n)$), then $P(X_\infty = 0) > 0$
  - if $\gamma_n = \gamma$, $P(X_\infty = 0) \geq (1-x)^{1/\gamma P(F_n)}$
  - if $\alpha = 1$ and $C < 1/P(F_n)$, then $P(X_\infty = 0) = 0$
- if $0 < P(F_n) = P(E_n) \leq 1$, $P(X_\infty = 1) = x$

Sharp results obtained by means of the theory of *martingales*

The lower bound of Lai&Robbins

Denote by $KL(p_j|p^*)$ the Kullback-Leibler divergence between the distribution of the j-th arm and the optimal arm.

Theorem : for all strategy pulling always "sufficiently" the optimal arm, and for all sub-optimal arm j, the number of times the arm j is pulled is lower bounded in expectation :

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

Corollary : each strategy has a regret at least in $C\log(n)$, where C depends on the distributions of the arms;

Minimax lower bound

Theorem : For n and N large enough, one may build a bandit problem for which the regret of any strategy is at least

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

Remark : the analysis of UCB permits to bound the regret by

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

for a certain constant C *independent from* the problem.

Identification of failures

Detection of breakdowns in electrical networks

Number of possible circuits $\approx 10^{50}$

N random netwotk generators réseaux more or less focusing on failure configurations

One has at disposal a simulator capable of detecting a failure, but each job is computationally expensive

Problem : how to use efficiently our N simulators to find as fast as possible a large number of failure configurations

Simplified modeling

For $1 \leq j \leq N$, $(X_t^j)_t$ i.i.d. uniformly distributed on $\{1, \ldots, m\}$

The failure configurations are $\{1, \ldots, M_j\}$

At time t, one chooses the distribution J_t and one draws $X_t^{J_t}$

Goal : for a given budget n, maximize

$$\sum_{j=1}^{n} \# \{1, \ldots, M_j\} \cap \left\{ X_t^j : J_t = j \right\}$$

Dual problem : minimize the time required to pour find the failures, all or partly

A bandit problem ?

Analogies :

- problem of sequential decisions
- N arms
- rewards $R_t = 1$ if $X_t^{J_t} \leq M_{J_t}$ *and if it has not been seen yet*

But a huge difference : a “reward” does not necessarily encourage to redo the same action ! In contrast : once a simulator j ran through, one has to turn away from it !

$\Longrightarrow$ Parameter of a simulator $r_t^j = R_t^j / m$, where

$$R_t^j = \#\{1, \ldots, M_j\} \setminus \left\{ X_t^j : J_t = j \right\}$$

evolves through time (*bandit non stationary*)

An optimistic solution

Estimation of the missing mass : Good-Turing ['53]

$$\widehat{R_t^j} = \#\{i \in \{1, \dots, m\} : \sum_{t: J_t = j} \mathbb{1}_{X_j^j = 1} = 1\}$$

Bound of [McAllester-Schapire '97]

$$P\left(r_t^j > \frac{\widehat{R_t^j}}{m} + \left(2\sqrt{2} + \sqrt{3}\right) \sqrt{\frac{\log(3/\delta)}{m}} \right) \leq \delta$$

Quite comparable to usual concentration inequalities

Good-UCB

Optimistic algorithm : upper confidence boundfor the missing mass of each simulator
Si

$$N_t(j) = \sum_{s=1}^{t} \mathbb{1}_{J_s = j}$$

désigne le nombre de tirage avec le simulateurs j jusqu'à l'instant t, Good-UCB choisit :

$$J_{t+1} = \operatorname*{argmax}_j \frac{\widehat{R_t^j}}{m} + c\sqrt{\frac{\log(3t)}{N_t(j)}}$$

Performance, bornes de regret, améliorations, généralisations :

work in progress. . .