TP 1 : From planning to reinforcement learning

emilie.kaufmann@inria.fr

October 20th, 2015

The goal of this practical session is to learn how to solve a *planning problem* in an MDP, that is solving an MDP whose transition and reward functions are known using the VI (Value Iteration) and PI (Policy Iteration) algorithms. We also try a first *reinforcement learning* algorithm, Q-Learning, that aims at maximizing reward in an MDP whose reward and transition functions are unknown.

Material on http://chercheurs.lille.inria.fr/ekaufman/teaching.html

1 The retail store management problem

A store retailer who sells washing machines wants to optimize its profit by carefully choosing how to manage his stock, that is how many washing machine he should buy at the end of each week.

- X_t denotes the number of machines in his stock at the end of week t;
- he can store at most M machines. Each week, he has to pay a maintenance cost of h for each machine in his stock;
- at the end of week t, he chooses to order A_t new machines
- he gets the machines at the beginning of the next week and pays c/per machine ordered plus a fix cost of delivery of K (independent of the number of machines delivered, but of course not if $A_t = 0$);
- he sells the machines at price p;
- the number of machine bought by customers on week t is a random variable D_t, and we make the assumption that $(D_t)_{t\geq 1}$ is i.i.d.
- R_t denotes the profit made on week t;
- when $t = 1$, the stock is full : $X_1 = M$;
- let $\gamma \in]0, 1[$ be such that the inflation rate by week egals $\gamma^{-1} - 1$: the retailers wants to maximize its expected discounted profit:

$$\mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} R_t\Big]$$

For the simulation we can take for example $M = 15$, $K = 0.8$, $c = 0.5$, $h = 0.3$, $p = 1$, and D_t be a truncated geometric distribution with parameter 0.1.

Many useful functions can be found in the folder available online. The above model and the different stages of the TP are in the file 'mainTP1.m'.

Simulated trajectories and estimation of a value function

The problem described above is equivalent to solving an MDP with state space $\mathcal{X} = \{0, \dots, M\}$, action space $\mathcal{A} = \{0, \dots, M\}$ and transition

$$\begin{cases} X_{t+1} &= \Big((X_t + A_t) \wedge M - D_{t+1}\Big)_+ \\ R_{t+1} &= -K\mathbb{I}_{(A_t>0)} - c\Big((X_t + A_t) \wedge M - X_t\Big)_+ - hX_t + p\Big(D_{t+1} \wedge (X_t + A_t) \wedge M\Big) \end{cases}$$

Note that the transition is random (it depends on the number of customers that arrived in the current week). The *given* function

```
[xnew,reward] = simu_transition(x,a,M,K,h,c,p,D)
```

outputs the new state and reward obtained when action a is chosen in state x.

1. A policy π is represented by a line vector of size $M + 1$, where $\pi(1 + k)$ gives the number of machines the retailers choose to buy if the current stock is k. Write a function

   ```
   [X,R] = trajectory(n,x0,pi,...)
   ```

   simulating the evolution of the stock and profit over n weeks when the retailer buys following policy pi and starts with $x0$ machines.

2. Choose a policy π and visualize the discounted profit over one trajectory.

3. For the same policy π, and an initial state x_0 compute an estimate of

   $$V^\pi(x_0) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_0 = x_0\right].$$

   based on a lot of simulations (Monte-Carlo estimation).

Parameters of the MDP

The *given* function of the parameters of the problem

```
[P,R] = MDP(D,M,K,h,c,p)
```

outputs the transition matrix $P(x, y, a) = p(y|x, a)$ and the reward matrix $R(x, a) = E[R|x, a]$.

Policy evaluation

Propose three different methods to compute V^π for a given policy π and implement one of them:

```
[V] = pol_eval(pol,...)
```

What are the benefits of using Monte-Carlo estimation?

Value Iteration (VI), Policy Iteration (PI)

Recall that the Value Iteration and Policy Iteration algorithms can both be stated using the Q-value function, where the Q value associated to a value function V is given by

$$Q(x,a) = r(x,a) + \gamma \sum_{y \in \mathcal{X}} p(y|x,a)V(y).$$

1. Implement VI to find the optimal strategy for the retailer.

```
[V, pol] = VI(P,R,gamma,V0,...)
```

2. Implement PI with different methods for the evaluation step.

```
[V, pol] = PI(P,R,gamma,pi0,...)
```

Question 1: For a specified choice of the parameters, give the vector V^* describing the optimal policy, and comment on its form.

Question 2: For VI and your favorite version of PI, plot $||V^* - V_k||_\infty$ as a function of iteration k, to compare the speed of convergence. For a fixed number of iteration, which algorithm is the fastest to implement?

2 Q-Learning : a first RL algorithm

In real life, the retailer does not know the distribution D_t, so he cannot use the previous tools to act optimaly: we are in a reinforcement learning problem, where the MDP is unknown. Q-Learning is a stochastic algorithm based on simulated transitions that converges towards the optimal Q-value $Q^*(x,a)$.

Implement Q-learning. How fast does it converge?

Question 3: Write the code for Q-Learning.

Some details on Q-Learning Q-Learning builds a sequence $\hat{Q}_t \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{A}|}$ of estimates of Q^* based on trajectories (under a suitably exploratory policy), so that the estimate of the optimal policy at round t is

$$\hat{\pi}_t(x) = \text{argmax}_a \ \hat{Q}_t(x,a).$$

Entries

- policy π_t for choosing an action at round t
- sequence $\alpha_t(x,a)$ such that $\sum_t \alpha_t(x,a) = \infty$ and $\sum_t \alpha_t(x,a)^2 < \infty$
- initial state X_0

For $t \geq 0$

- select an action $A_t \sim \pi_t(X_t, .)$
- observe a transition : get reward R_t and new state X_{t+1}
- update (X_t, A_t):

$$\hat{Q}_{t+1}(X_t,A_t) = (1-\alpha_{N(X_t,A_t)}(X_t,A_t))\hat{Q}_t(X_t,A_t)+\alpha_{N(X_t,A_t)}(X_t,A_t)(R_t+\gamma \max_{a\in\mathcal{A}} \hat{Q}_t(X_{t+1},a)),$$

where $N(x,a)$ is the number of visits of the state-action pair (x,a).