TP2: The stochastic multi-armed bandit

emilie.kaufmann@inria.fr

November 3rd, 2015

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

1 Building a MAB problem

In a stochastic multi-armed bandit model, each arm is a probability distribution and drawing an arm means observing a sample from this distribution. We will consider only distributions supported in $[0, 1]$. Arms can be implemented as objects, and we give the following classes (you can create others):

`armBernoulli.m`   `armBeta.m`   `armFinite.m`   `armExp.m`

For each object *Arm* belonging to one of these classes, we have the following commands (methods):

- *Arm.mean* returns the mean of the arm
- *Arm.play* gives a sample from the arm

A multi-armed bandit model is a collection of arms:

```
MAB = {Arm1, Arm2, ...,ArmK}
```

Start by completing the begining of the file *'mainTP2.m'* with your own multi-armed bandit problem (i.e. choose the number of arms and the distribution of each arm).

2 The UCB algorithm

We denote by :

- $N_a(t)$ the number of draws of arm a up to time t
- $S_a(t)$ the sum of rewards gathered up to time t

$\hat{\mu}_a(t) = \frac{S_a(t)}{N_a(t)}$ is therefore the empirical mean of the rewards gathered from arm a up to time t. When $N_t(a) > 0$, the UCB index associated to arm a is

$$B_t(a) = \hat{\mu}_a(t) + \sqrt{\frac{\alpha \log(t)}{N_a(t)}}$$

The α-UCB algorithm starts with an initialization phase that draws each arm once, and for $t \geq K$, chooses at time $t+1$ arm

$$\boxed{A_{t+1} = \underset{a \in \mathcal{A}}{\text{argmax}}\ B_t(a)}$$

1. Write a function

   ```
   [rew,draws]=UCB(T,alpha,MAB)
   ```

   simulating a bandit game of length T with the UCB-α strategy: *rew* and *draws* are respectively the sequence of the T rewards obtained and the arms drawn.

2. Compare with the naive strategy that chooses at time t the arm with highest empirical mean $\hat{\mu}_a(t)$.

3. Based on many simulations, estimate the expected regret of the naive strategy and of the UCB algorithm with several values of α.

   $$\mathbb{E}[R_T] = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} x_t\right]$$

Question 1 For a bandit problem of your choice (to specify), draw (on the same plot) regret curves for the naive strategy and the UCB algorithm for several values of α. Which value of α seems to be the best?

3 Complexity of a bandit problem

Lai and Robbins proved in 1985 that in a bandit problem with arms $\nu_1, \dots, \nu_K$ that are parametric distributions, the regret is lower bounded, for large values of T as

$$\frac{\mathbb{E}[R_T]}{\log T} \geq \sum_{a \neq a^*} \frac{(\mu^* - \mu_a)}{\text{KL}(\nu_a, \nu^*)},$$

where $\text{KL}(\nu, \nu') = \int \log(d\nu(x)/d\nu'(x))d\nu(x)$ is the Kullback-Leibler (KL) divergence between distributions ν and ν'. The Kullback-Leibler divergence between two Bernoulli distribution $\mathcal{B}(x)$ and $\mathcal{B}(y)$ is given by

$$\text{KL}(\mathcal{B}(x), \mathcal{B}(y)) = x \log \frac{x}{y} + (1-x) \log \frac{1-x}{1-y}.$$

Write a function that returns the complexity term of a multi-armed bandit problem with Bernoulli arms. The complexity term C is such that the optimal regret at time T should be $C \times \log(T)$.

```
[c]=complexity(MAB)
```

4 A Bayesian idea for Bernoulli bandit problems

Consider a bandit problem with K arms that are Bernoulli distributions with means $\theta_1, \dots, \theta_K$. The UCB algorithm uses confidence intervals on the unknown mean of each arm to make its decision.

In a *Bayesian* view on the MAB, the θ_a are no longer seen as unknown parameters but as (independent) random variables following a uniform distribution. The *posterior distribution* on the arm a at time t of the bandit game is the distribution of θ_a conditional to the observations from arm a gathered up to time t. Each sample from arm a leads to an update of this posterior distribution.

$$\begin{array}{rl} \textbf{Prior distribution} & \theta_a \sim \mathcal{U}([0,1]) \\ \textbf{Posterior distribution} & \theta_a | X_1, ..., X_{N_a(t)} \sim \text{Beta}\left(S_a(t)+1, N_a(t)-S_a(t)+1\right) \end{array}$$

where $X_1, ..., X_{N_a(t)}$ are the rewards from arm a gathered up to time t.
Bayesian bandit algorithms choose an action based on the current posterior distributions over the parameters of the arms,

$$\pi_a(t) = \text{Beta}(S_a(t)+1, N_a(t)-S_a(t)+1).$$

Thompson Sampling is a simple, randomized, Bayesian algorithm.
At time $t+1$,

- for each arm a, draw a sample $\theta_a(t)$ from $\pi_a(t)$
- choose
$$A_{t+1} = \underset{a \in Arms}{\text{argmax}}\ \theta_a(t)$$
- update the posterior on arm A_t (posterior distributions on the other arms are unchanged)

1. Write a function

   ```
   [rew,draws]=Thompson(T,MAB)
   ```

   simulating a bandit game of length T with Thompson Sampling (the Matlab command *betarnd* helps drawing samples from a Beta distribution)

2. Can Thompson Sampling also be called an ‘optimistic algorithm’?

Question 2: For two different bandit problems with Bernoulli arms (that you specify), one ’easy’ (small complexity term), and one ’difficult’ (large complexity term), compare the regret of Thompson Sampling with that of UCB. Add Lai and Robbins’ lower bound on your plots.