Introduction to Optimization
Exercise 7

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

July 10, 2013

1 Multi-armed bandits & UCB

Assume there are $n = 10$ bandits. Each bandit is binary (i.e., $y_t \in \{0, 1\}$) with $P(y_t = 1|a_t = i) = p_i$. The agent has $T = 100$ rounds to play the machines and aimes to maximize $\sum_{t=1}^{T} y_t$.

For simplicify, in the following assume that $p_i = i/10$ for $i = 1, .., 10$. But the agent does not know this, of course.

a) Implement this bandit scenario using a proper (clock) random seed. (Write a method that receives a a_t and returns a $y_t \in \{0, 1\}$.) Simulate a random agent that chooses actions $a_t \sim \mathcal{U}(\{1, .., 10\})$ uniformly. Let the agent play 10 games (each with $T = 100$ rounds). What is the random agent's average reward?

b) Implement a UCB agent. For this, the agent needs to keep track how often he has played a machine (n_i) and how often this machine returned $y = 1$ (let's call this β_i) or $y = 0$ (let's call this α_i). What is the agent's average reward? (Averaged over 10 games, as above.)

c) (Bonus.) Assume the agent knows that the bandits are binary. He can exploit this knowledge: His belief can be

$$b_t = P((p_1, .., p_n)|h_t) = \prod_i \text{Beta}(p_i|\alpha_i, \beta_i)$$

where Beta is the so-called Beta-distribution over the Bernoulli parameter $p_i \in [0, 1]$. At Wikipedia you can find information on the mean and variance (and also the cumulative distribution function, called regularized incomplete beta function) of a Beta distribution. How exactly could an agent use this to perhaps become better than the agent in b)?

2 Global optimization on the Rosenbrock function

On the webpage you'll find octave code for GP regression from Carl Rasmussen (`gp01pred.m`). The `test.m` demonstrates how to use it.

Use this code to implement a global optimization method for 2D problems. Test the method

a) on the 2D Rosenbrock function defined in exercise e06, and

b) on the Rastrigin function as defined in exercise e04 with $a = 6$.

Note that in `test.m` I've chosen hyperparameters that correspond to assuming: smoothness is given by a kernel width $\sqrt{1/10}$; initial value uncertainty (range) is given by $\sqrt{10}$. How does the performance of the method change with these hyperparameters?

3 Constrained global optimization?

On slide 6:2 it is speculated that one could consider a constrained blackbox optimization problem as well. How could one approach this in the UCB manner?