Multi-Arm Bandit

Alessandro Tenaglia

Machine and Reinforcement Learning in Control Applications

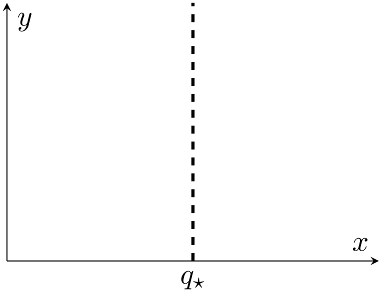
March 9, 2022

Problem

- Choose repetitively which arm to pull from those available
- Each arm returns a reward
- The objective is to maximize the expected total reward

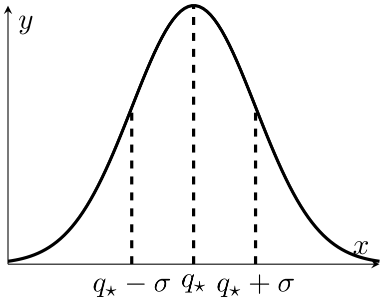
Bandit

1. **Deterministic** and **stationary**
   - Rewards are equal to $q_\star(a)$
   - $q_\star(a)$'s don't change over time



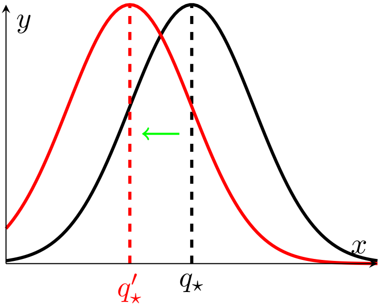
Bandit

1. **Deterministic** and **stationary**
   - Rewards are equal to $q_\star(a)$
   - $q_\star(a)$'s don't change over time
2. **Stochastic** and **stationary**
   - Normally distributed rewards with mean $q_\star(a)$
   - $q_\star(a)$'s don't change over time



Bandit

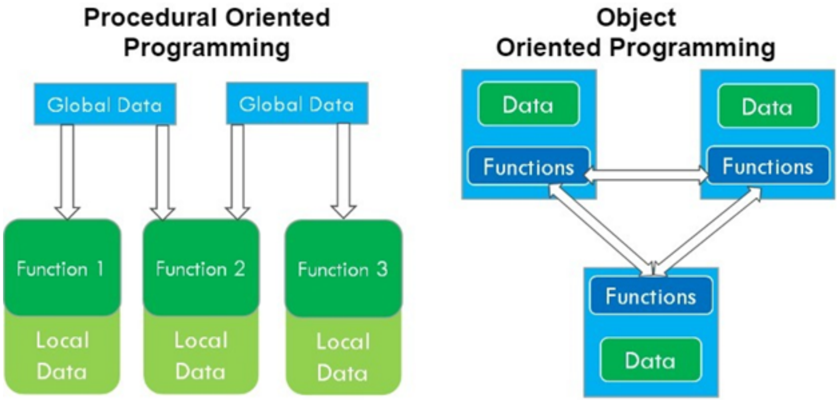
1. **Deterministic** and **stationary**
   - Rewards are equal to $q_\star(a)$
   - $q_\star(a)$'s don't change over time
2. **Stochastic** and **stationary**
   - Normally distributed rewards with mean $q_\star(a)$
   - $q_\star(a)$'s don't change over time
3. **Stochastic** and **non-stationary**
   - Normally distributed rewards with mean $q_\star(a)$
   - $q_\star(a)$'s change over time



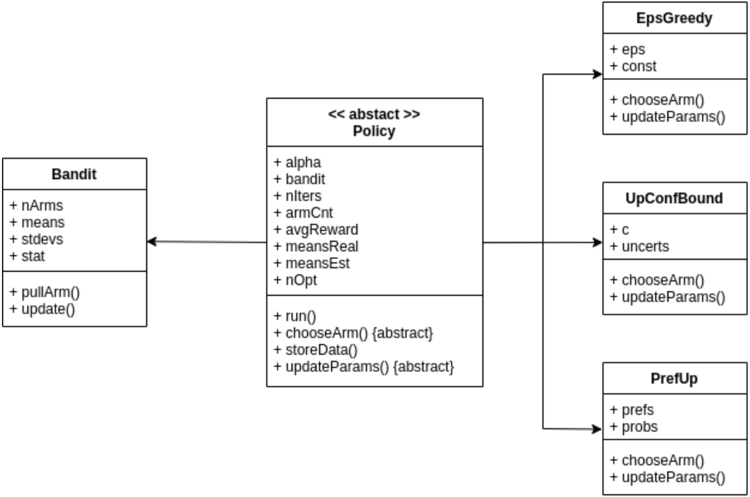
Policy

- The arm to pull is chosen according to the following policies:
  - **ϵ-greedy sample-average**
  - **Upper confidence bound**
  - **Preference updates**
- Comparison of the results obtained with the various policies and parameters

POP vs OOP

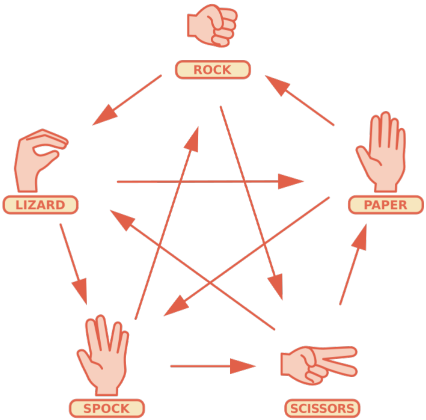

Procedural Oriented Programming
Global Data
Global Data
Function 1
Local Data
Function 2
Local Data
Function 3
Local Data
Object Oriented Programming
Data
Functions
Data
Functions
Functions
Data

Diagram


EpsGreedy
+ eps
+ const
+ chooseArm()
+ updateParams()
<< abstact >>
Policy
+ alpha
+ bandit
+ nIters
+ armCnt
+ avgReward
+ meansReal
+ meansEst
+ nOpt
+ run()
+ chooseArm() {abstract}
+ storeData()
+ updateParams() {abstract}
Bandit
+ nArms
+ means
+ stdevs
+ stat
+ pullArm()
+ update()
UpConfBound
+ c
+ uncerts
+ chooseArm()
+ updateParams()
PrefUp
+ prefs
+ probs
+ chooseArm()
+ updateParams()

Assignment #1

- Model the **Rock, Paper, Scissors, Lizard, Spock** game



Assignment #1

- Model the **Rock, Paper, Scissors, Lizard, Spock** game
  - See the episode 5x17 of "The Big Bang Theory" for rules
  - Choose repetitively which action to play between: Rock, Paper, Scissors, Lizard, Spock
  - The opponent plays randomly
  - Reward $+1$ for winning and -1 for losing
  - The objective is to maximize the rewards
  - Analyze the trends of the expected rewards for each action