CSCI E-82a
Probabilistic Programming and AI
Lecture 10
Bandit Models

Steve Elston

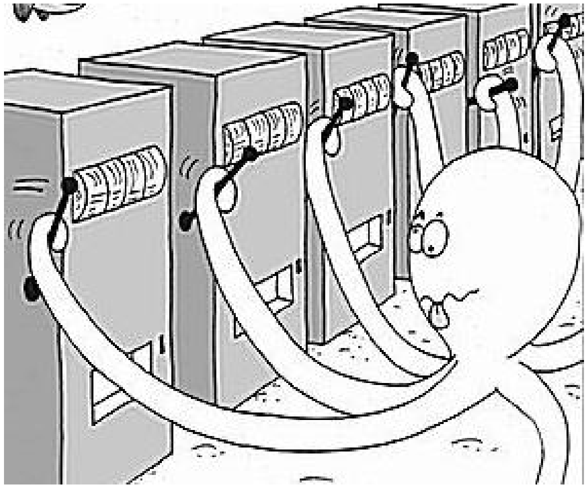
HARVARD
Extension School

Introductions to Bandit Models

- What is a bandit agent?
- How do bandit agents learn?
- What is a policy?
- Exploration vs. exploitation

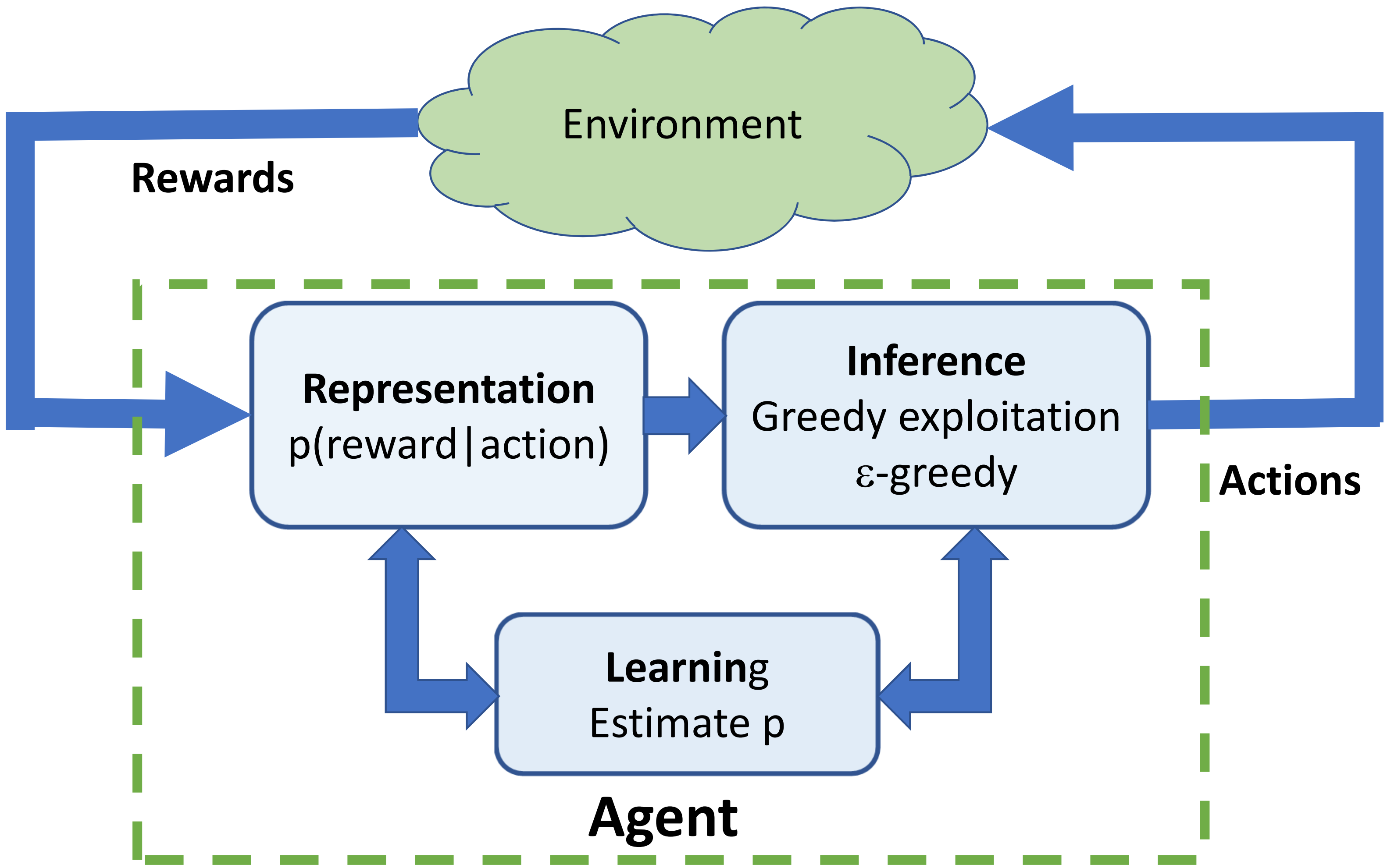
The Bandit Agent

- Bandit model is based on one-arm bandit gambling machine
- Bandit simplified RL model
  - **No State**
- Bandit agent **learns policy** to maximize reward
- Learning by experience
  - Pull lever
  - Receive reward



Attribution: Microsoft Research

The Bandit Agent



The Bandit Agent

- Agent has no knowledge of the environment
- Agent is **model-free**
- Agent **learns by experience** in the environment
  - Takes **actions** in the environment
  - Receives **rewards** from the environment
- Agent learns a **policy**
  - The **actions** of the agent **follow the policy**

Bandit Agent Learning

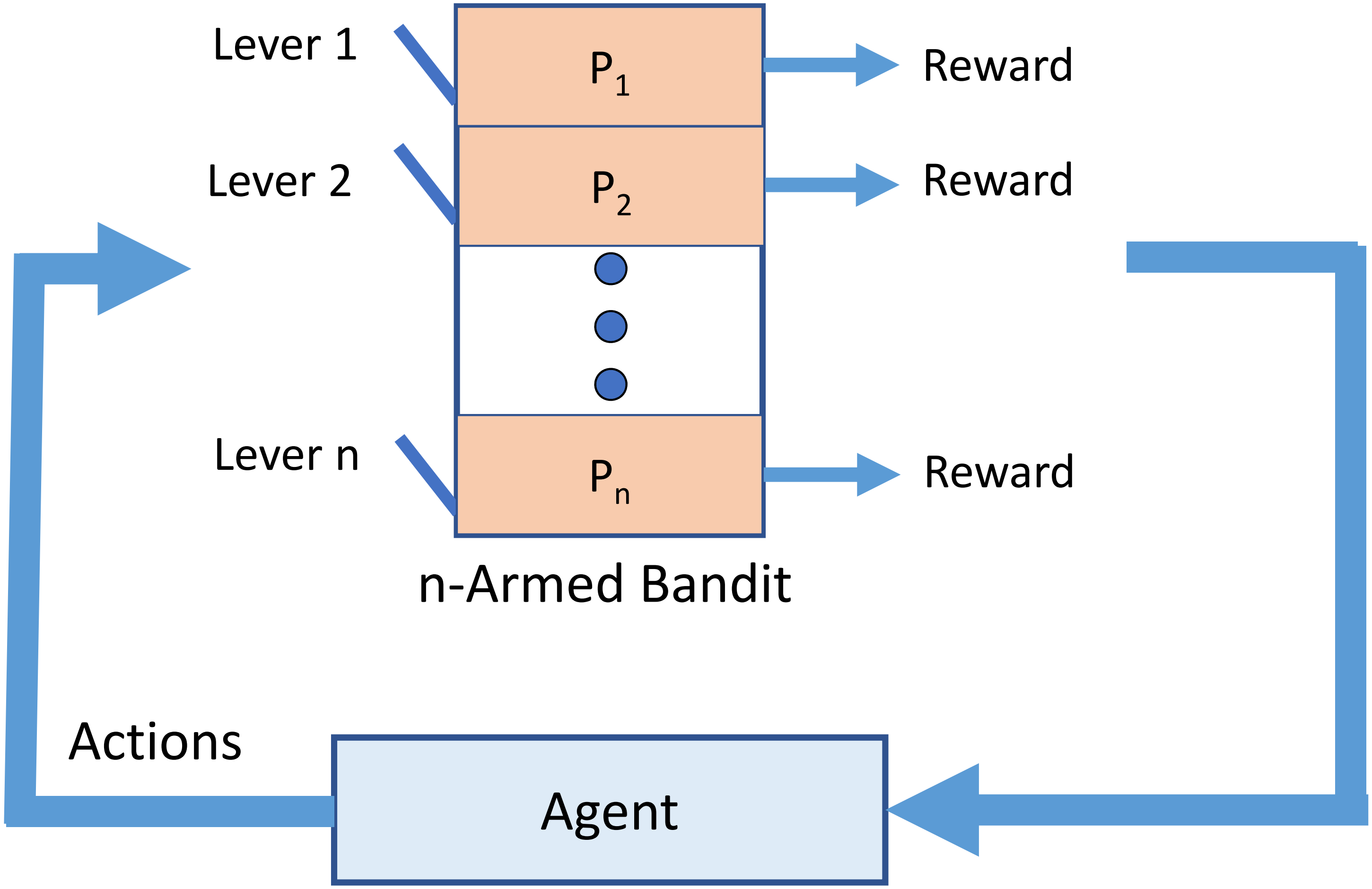
- Each arm of the bandit undergoes a series of Bernoulli trials
- The outcomes are in set { 1, 0 }, where 1 is win, 0 is loose
- The probability of win or loss:

$$P(x \mid p) = \begin{cases} p \textit{ if } x = 1 \\ (1 - p) \textit{ if } x = 0 \end{cases}$$

Or

$$P(x \mid p) = p^x (1 - p)^{(1-x)} \; x \in 0, 1$$

Bandit Learning Model



Bandit Agent Learning

- Estimate of p_i, given action, a_i, is the representation
- Can estimate p_i by counts; fraction of success
- Or, step-wise learning:

$$p_{t+1} = p_t + \alpha\,(p_t - \text{reward})$$

where,

α = learning rate

p_t – reward = error

Policy

- The actions of the bandit agent are determined by a **policy, π**
- The **expected reward** the policy determines the **action value**

$$q_\pi(a) = \mathbb{E}_\pi[R_t \mid A_t = a]$$

- Our goal is to **learn** an **optimal policy**

$$q_{\pi^*}(a) = \mathbb{E}_{\pi^*}[R_t \mid A_t = a]$$

- The optimal policy has an expected action value greater than or equal to all possible policies:

$$q_{\pi^*}(a) \geq q_\pi(a) \; \forall \; \pi$$

Policy

- Agent learns a **policy**
  - The **actions** of the agent **follow the policy**
- An **optimal policy, π*,** has maximum expected value

$$q_{\pi^*}(a) = \mathbb{E}_{\pi^*}[R_t \mid A_t = a] = \max_{a^*} \mathbb{E}[R_t \mid A_t = a^*]$$

- There is **no state** in the above relation
- The agent samples levers and uses result to estimate p_i for each lever

Exploitation vs. Exploration

- The agent following a **greedy policy** maximizes short-term reward
- But, the greedy policy may not be optimal
  - Learning is stochastic
  - There is always uncertainty in learned parameters
  - May be a better policy
- Improve policy by mixing **greedy exploitation** with **random exploration**

Exploitation vs. Exploration

- A **greedy policy** never improves once set
- Must mix **exploitation** with **exploration**
  - At each step determine if exploit with greedy policy or explore
  - Explore with **probability** ε; e.g. take a **random action**
  - Exploit with greedy policy with **probability** (1 - ε)
- Result is an ε**-greedy policy**
  - ε is small number; 0.05, 0.01, 0.001…..
  - Decrease ε as learning progresses: policy becomes greedier

Exploitation vs. Exploration

- Update policy with ε-greedy improvement to find improved policy π_{k+1} at *k*th step of algorithm

$$q_{\pi_{k+1}}(a) = \begin{cases} Greedy\ improvement\ with\ p = 1 - \epsilon \\ Random\ action\ with\ p = \epsilon \end{cases}$$

$$= \begin{cases} \max\limits_{a}\ q_{\pi_k}(a)\ with\ p = 1 - \epsilon \\ a \sim\ Bernoulli\ with\ p = \epsilon \end{cases}$$

- Iterate until convergence – small change in probability of success