Learning Automata: A Short Course

Ole-Christoffer Granmo[1]

[1]Centre for Artificial Intelligence Research (CAIR)
University of Agder, Norway

Outline

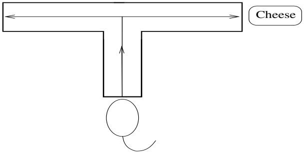
- Reinforcement Learning
- Learning Automata
- Game Theory and Automata Games
- Assignment

Part I: Reinforcement Learning

Reinforcement Learning

- **Reinforcement Learning (RL):** An agent that *explores* an *environment*
- **The Agent:** *Perceives* its current state and takes *actions*
- **The Environment:** provides a *reward* or a *penalty*
- **RL algorithms:** Attempt to find a policy for maximizing the agent's cumulative reward over the course of the problem

The T-maze Problem

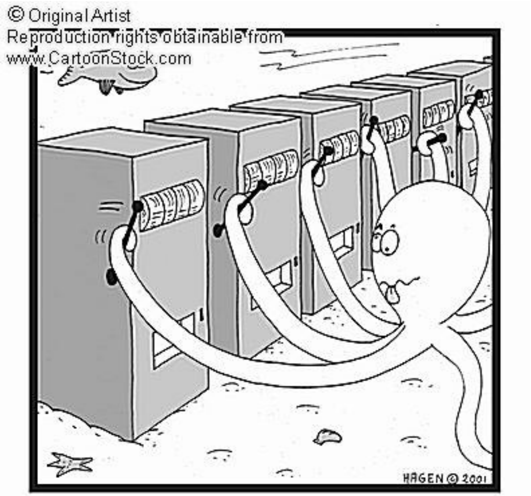


- Hungry rat is placed at the lower end of the middle limb of a T-maze
- Rat can move along the limb and turn to the right or left
- Food: Kept at the end of the right arm with probability 0.7 and at the end of the left arm with probability 0.3
  - How does the rat behave over successive trials?
  - Can the rat behavior be modeled mathematically?

The Multi-Armed Bandit Problem

- **Exploration vs. Exploitation:** The conflict between Exploration vs. Exploitation is well-known in RL and other areas of AI
- **The Multi-Armed Bandit Problem:** Captures the essence of this conflict
- Has thus occupied researchers for over fifty years

The Multi-Armed Bandit Problem (Contd...)



Compulsive gambling

The Multi-Armed Bandit Problem (Contd...)

- **The Multi-Armed Bandit Problem** is a classical optimization problem where an agent sequentially pulls one of n arms attached to a gambling machine
  - Each pull results either in a *reward* or a *penalty*
  - The reward probabilities, $(r_1, \ldots, r_n)$, of arms are unknown
- **Challenge:** One must balance between exploiting existing knowledge and obtaining new information in order to maximize the amount of rewards received

Part II: Learning Automata

Learning Automata

The concept of learning automaton grew out of the work of:

- Psychologists in modeling observed behavior
- Statisticians to model the choice of experiments based on past observations
- Operation researchers to implement optimal strategies in the context of the two-armed bandit problem
- System theorists to make rational decisions in random environments.

Learning Automata Characteristics

- Learning Automata (LA) are adaptive decision making devices that can operate in:
  - **Unknown Environments:** They do not need information about the effect of their actions at the start of the operation
  - **Random Environments:** An action does not necessarily produce the same response each time it is performed
- A powerful property of LA is that they progressively improve their performance
  - By the means of a “learning” process
  - Combine rapid and accurate convergence
  - Low computational complexity

Learning Automata Make Decisions in an Environment

- Example decisions:
  - Which of two alternate routes to choose in network routing
  - Choosing between air travel and car travel to a neighbor city
- Outcome of choice random:
  - The time to reach a destination using one route is assumed to be a random variable depending on traffic conditions

Numerous Applications

- Call Routing in Telephone Networks
- Playing Board Games Like *Go*
- Resource Allocation in Web Polling
- Pattern Recognition (The Tsetlin Machine)
- ...

Application I - Call Routing in Telephone Networks

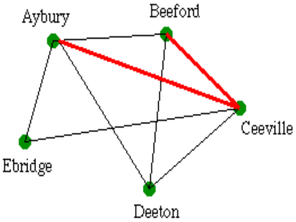
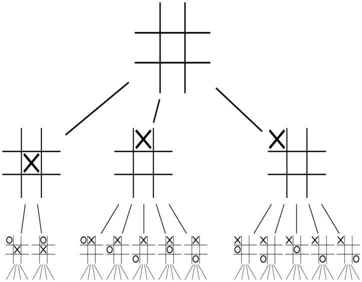


Figure: Telephone Routing using LA[1]

- **Routing Problem:** Exploring possible outgoing links from a network node

[1] Figure from "Call routing in telephone networks" by Richard Gibbens and Stephen Turner, 1997

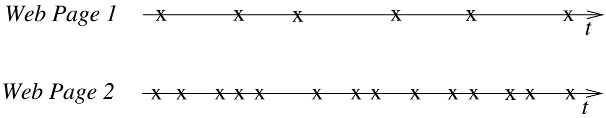
Application II - Game Tree Exploration



- **Bandit Problem:** Exploring the possible moves from a board position of a game
- *MoGo:* UCB-Tuned (Auer *et al*, 2002) is used for move exploration in *MoGo:*, a top-level Computer-*Go* program on 9 $\times$ 9 *Go* boards (Gelly and Wang, 2006)

Application III - Polling of Web Pages

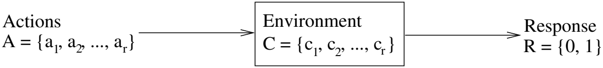
*Given a limited monitoring capacity, how often should each monitored web page be polled in order to **maximize** the amount of new information discovered?*



Bandit Problem: Exploring how large a fraction of the limited monitoring capacity that should be assigned to each web page that is monitored

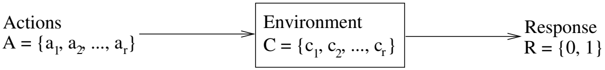
Learning Automata - The Environment

The Environment



- Environment $\longrightarrow$ Large class of general unknown media in which an LA or a group of LA can operate
  - **Input:** Action from Set of Actions
  - **Output:** *Reward* ($\beta =$ '0$'$) or *Penalty* ($\beta =$ '1$'$)
  - **Penalty Probabilities:** For action i, there is a certain probability that the *Environment* responds with a *Penalty*
    - $P(\text{Penalty}|\text{Action} = a_i) = c_i, 1 \leq i \leq r$
  - **Remark:** If the Environment does not respond with a *Penalty*, it responds with a *Reward* instead

Static and Dynamic Environments



- An environment can be
  - **Static:** Penalty probabilities do not change
  - **Dynamic:** Penalty probabilities change

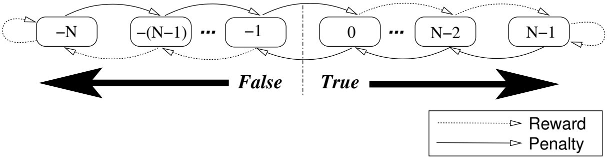
Simulation of Environment

Simulation of Two Action Environment:

```
class Environment:
  def __init__(self, c_1, c_2):
      self.c_1 = c_1
      self.c_2 = c_2

  def penalty(self, action):
     if action == 1:
         if random.random() <= self.c_1:
             return True
         else:
             return False
     elif action == 2:
         if random.random() <= self.c_2:
             return True
         else:
             return False
```

Learning Automata - The Automaton

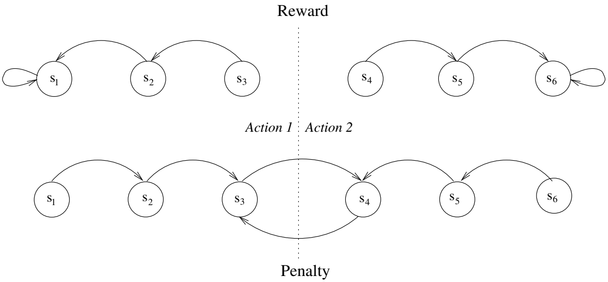
The Operation of an Automaton



- An automaton remembers which actions are "good" by maintaining a state $s_t \in \{s_1, \ldots, s_n\}$
- **Operation** – an automaton:
  1. Selects and *outputs* an action based on its present state
  2. Takes a response from the environment as *input*
  3. Changes its *state* based on (a) the *response* and (b) the *action* performed
- An automaton can be said to *learn* if it reduces the number of penalties received as a result of interacting with the environment

Two-Action Tsetlin Automaton

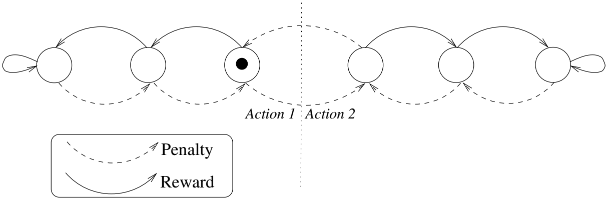
Tsetlin Automaton with 3 states per action:



- A Tsetlin-automaton can learn the optimal action if the lowest penalty probability is less than 0.5
- Number of automaton states determines “learning accuracy” and “learning speed”

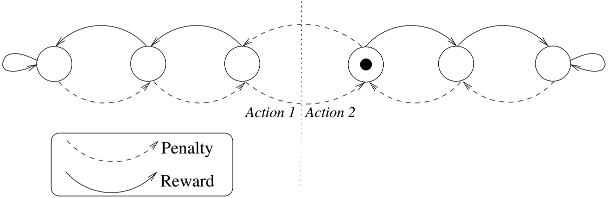
Two-Action Tsetlin Automaton – Example Run I

Initial state of Tsetlin Automaton with 3 states per action:



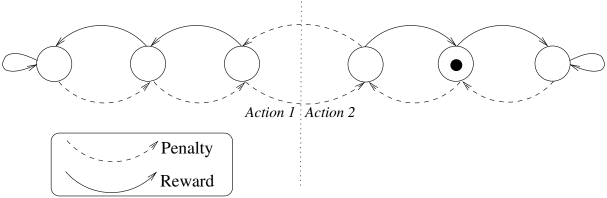
1. **Selected Action:** *Action 1*
2. **Response from Environment:** *Penalty*

Two-Action Tsetlin Automaton – Example Run II



1. **Selected Action:** *Action 2*
2. **Response from Environment:** *Reward*

Two-Action Tsetlin Automaton – Example Run III



1. **Selected Action:** *Action 2*
2. **Response from Environment:** *Penalty*

Two-Action Tsetlin Automaton – Implementation

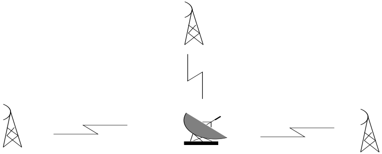
```
class Tsetlin:
    def __init__(self, n):
        # 'n' is the number of states per action
        self.n = n
        # Initial state selected randomly
        self.state = random.choice([self.n, self.n+1])

    def reward(self):
        if self.state <= self.n and self.state > 1:
            self.state -= 1
        elif self.state > self.n and self.state < 2*self.n:
            self.state += 1

    def penalize(self):
        if self.state <= self.n:
            self.state += 1
        elif self.state > self.n:
            self.state -= 1

    def makeDecision(self):
        if self.state <= self.n:
            return 1
        else
            return 2
```

Part III: Game Theory and Automata Games

Decentralized Decision Making

- **Decentralization** is a common and often necessary feature of complex natural and man-made systems
- Arises from the reality that the complete information exchange needed for centralized decision making may not be feasible
- **Formidable Problem:** Coordination of decentralized decision makers

Example: QoS Control in Sensor Networks



- Consider a basic sensor network that consists of an *unknown* number of *sensors* and a single *base station*
  - Each sensor can be *powered-down* or *powered-up*
  - The base station receives packets from powered-up sensors
- **Problem:** How can the base station control the sensors so that only Q sensors are powered up when the base station:
  1. Is only able to broadcast information
  2. Cannot address the sensors individually
  3. Does not know the number of sensors in the network

Learning Automata and Games

- Decentralized decision making can be formulated as a game
- **Game:** Metaphor for a much wider range of human interactions
    - Outcomes depend on the *interactive strategies* of two or more persons
    - Persons have *opposed* or at best *mixed motives*

Normal Form Representation of Games

	B Strategy 1	**B Strategy 2**
A Strategy 1	*3*, **3**	*0*, **5**
A Strategy 2	*5*, **0**	*1*, **1**

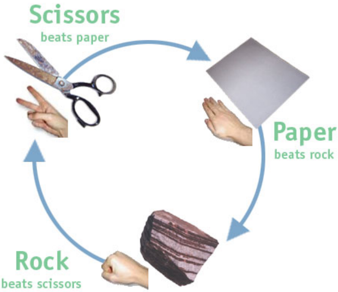
- The **normal form representation of a game** is a matrix which shows *players*, *strategies*, and *payoffs*
- In the example, there are two players:
  - One chooses the *row* and the other chooses the *column*
  - Each player has two strategies, which are specified by the number of rows and the number of columns
- The payoffs are provided in the interior
  - The first number is the payoff received by the row player
  - The second is the payoff for the column player

Normal Form Representation of Games

	B Strategy 1	**B Strategy 2**
A Strategy 1	*3*, **3**	*0*, **5**
A Strategy 2	*5*, **0**	*1*, **1**

Example: Suppose that *Player A* plays *Strategy 2* and that *Player B* plays *Strategy 1*. Then *Player A* gets 5, and *Player B* gets 0.

Example — Rock, Paper, Scissors



- Rock, Paper, Scissors is a so-called **zero-sum** game
- In zero-sum games the total benefit to all players in the game, for every combination of strategies, always adds to zero
- **Remark:** A player benefits only at the expense of others

Example — Prisoner's Dilemma

	B Remains Silent	**B** Betrays
A Remains Silent	Both six months	**A** ten years; **B** free
A Betrays	**A** free; **B** ten years	Both two years

- Two suspects, A and B, are arrested by the police
- The police have insufficient evidence for a conviction, and, having separated both prisoners, visit each of them to offer the same deal:
  1. If one testifies for the prosecution against the other and the other remains silent, the betrayer goes free and the silent accomplice receives the full 10-year sentence
  2. If both stay silent, the police can sentence both prisoners to only six months in jail for a minor charge
  3. If each betrays the other, each will receive a two-year sentence
- Each prisoner must make the choice of whether to betray

Dominant Strategy

Let an individual player in a game evaluate separately each of the strategy combinations he may face, and, for each combination, choose from his own strategies the one that gives the best payoff. If the same strategy is chosen for each of the different combinations of strategies the player might face, that strategy is called a "dominant strategy" for that player in that game

Nash Equilibrium

	B Remains Silent	**B** Betrays
A Remains Silent	Both six months	**A** ten years; **B** free
A Betrays	**A** free; **B** ten years	Both two years

If there is a set of strategies with the property that no player can benefit by changing her strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute the Nash Equilibrium

Pareto Optimality

	B Remains Silent	**B** Betrays
A Remains Silent	Both six months	**A** ten years; **B** free
A Betrays	**A** free; **B** ten years	Both two years

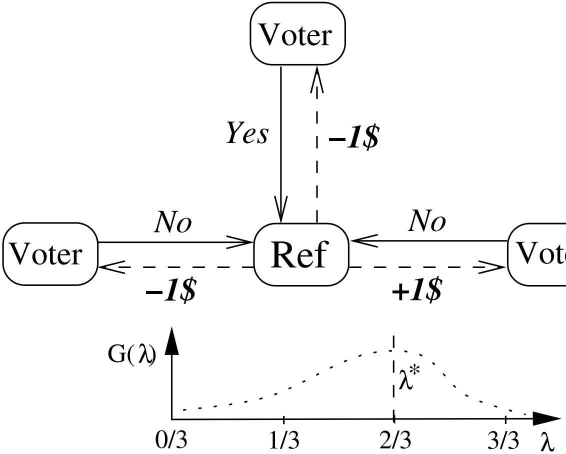
An outcome of a game is Pareto optimal if there is no other outcome that makes every player at least as well off and at least one player strictly better off.
Remark: A Pareto Optimal outcome cannot be improved upon without hurting at least one player

Assignment

The Goore Game

- Imagine a large room containing N cubicles and a raised platform
- One person (voter) sits in each cubicle and a Referee stands on the platform
- The Referee conducts a series of voting rounds as follows:
  1. On each round the voters vote “Yes” or “No”
  2. Votes doen simultaneously and independently
  3. The Referee counts the number λ of “Yes” votes
  4. The Referee has a uni-modal performance criterion $G(\lambda)$
  5. Optimized when the number of “Yes” votes is exactly λ^*
  6. The current voting round ends: Referee awards a dollar with probability $G(\lambda)$ and charges a dollar with probability $1 - G(\lambda)$ to every voter
  7. This is done independently
  8. On the basis of their individual gains and losses, the voters then decide, again independently, how to cast their votes on the next round

The Goore Game



Assignment

Download and install **Python** from *http://www.python.org*

Implement the following program and justify your results:

1. Create 5 Tsetlin Automata with actions “No” and “Yes”
2. Count the number of Tsetlin Automata that outputs a “Yes”-action
3. If the number of “Yes”-actions is M Then:
   - If $M = 0$ *OR* 1 *OR* 2 *OR* 3: Give each Automaton a reward with probability $M * 0.2$, otherwise a penalty
   - If $M = 4$ *OR* 5: Give each Automaton a reward with probability $0.6 - (M - 3) * 0.2$, otherwise a penalty
4. Goto 2

Remark: Generate the rewards independently for each automaton

Assignment

Deliverables:

- Oral presentation and demo
- Calculate average performance on 100 separate runs
- Calculate average performance with different number of states (e.g., 1, 2, 3, 5, 10)