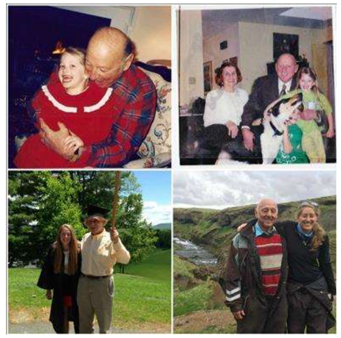
Markov Decision Processes

Thursday, March 25, 2021 5:02 PM

Markov Chains

- In the early 20th century, the mathematician Andrey Markov studied stochastic processes with **no memory**, called **Markov chains (MC)**.

- With **MC** as inspiration Leonard Baum and colleagues described the **Hidden Markov Models** (HMM) in several statistical papers ending the 60s.



- HMM were heavily used for speech processing thanks to Lawrence Rabiner in the 80-90s.

- **MC** process has a fixed number of states, and it randomly evolves from one state to another at each step.
- MC processes are memoryless processes because the probability to evolve from a state s to a state s' is fixed, and it depends only

on the pair (s, s'), not on past states:

- The set of all states is $\mathcal{S}$ and it is finite $M = |\mathcal{S}|$ and the transition probabilities hold:

$$p(t_{k+1} = s_j | t_k = s_i, t_{k-1} = s_l, \ldots, t_0 = s_m) = p(t_{k+1} = s_j | t_k = s_i)$$

- So, an $M \times M$ matrix holds all transition probabilities:

$$A \in [0,1]^{M \times M}$$

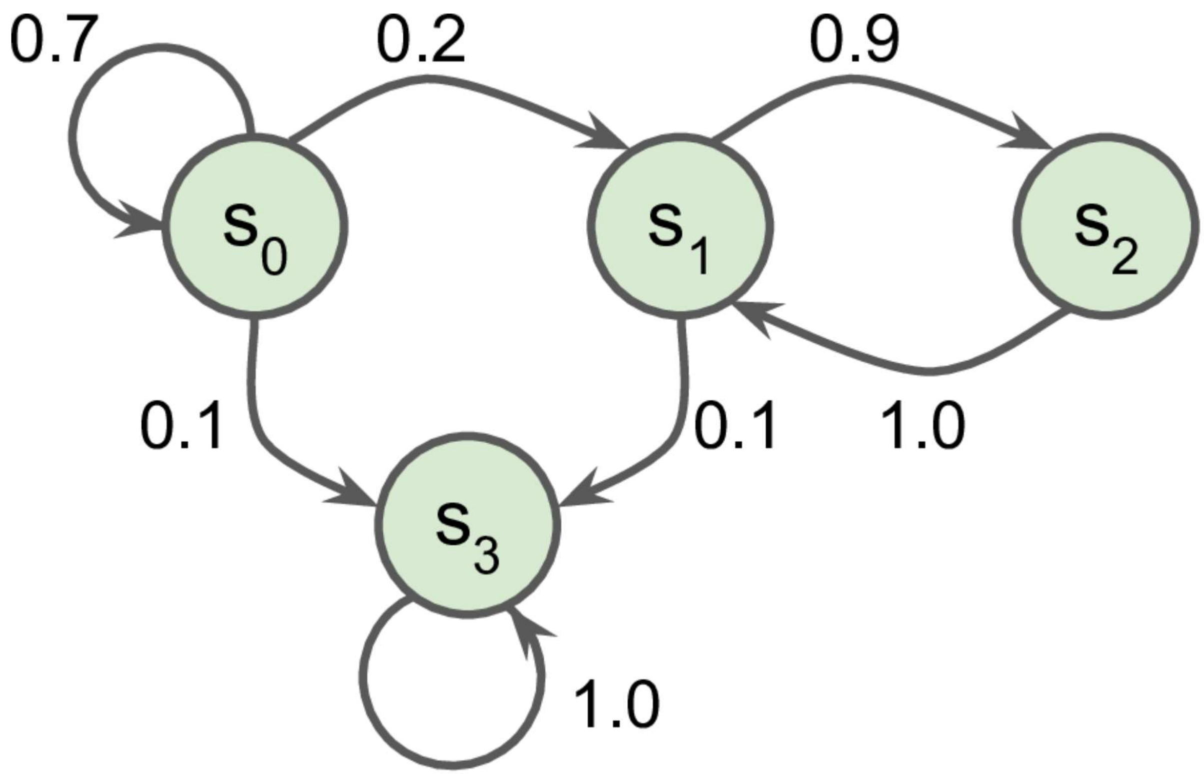
$$a_{ij} = p(t_{k+1} = s_j | t_k = s_i) \quad \forall s_i, s_j \in \mathcal{S}$$

- As any distribution, it meets the sum rule:

$$\sum_j p(t_{k+1} = s_j | t_k = s_i) = \sum_j a_{ij} = 1$$

- Let's explore the following MC with four states:

$$A = \begin{bmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{bmatrix}$$



- **Example 1:** Simulate a sequence of states of fixed length given the transition matrix.

- Markov chains are heavily used in thermodynamics, chemistry, statistics, and much more…

- Manuele Bicego (et al.) modeled shapes as with HMMs of the contour trajectories and classified objects using the likelihoods. The sky is the limit.

Markov Decision Processes

- **Markov decision processes (MDP)** were first described in the 1950s by Richard Bellman.

- MDP are like MC but with a twist:

- At each step, an agent can choose one of several possible actions $\mathcal{A} = \{a_k\}_{k=1}^{K}$,

- Then, the transition probabilities depend on the chosen action:

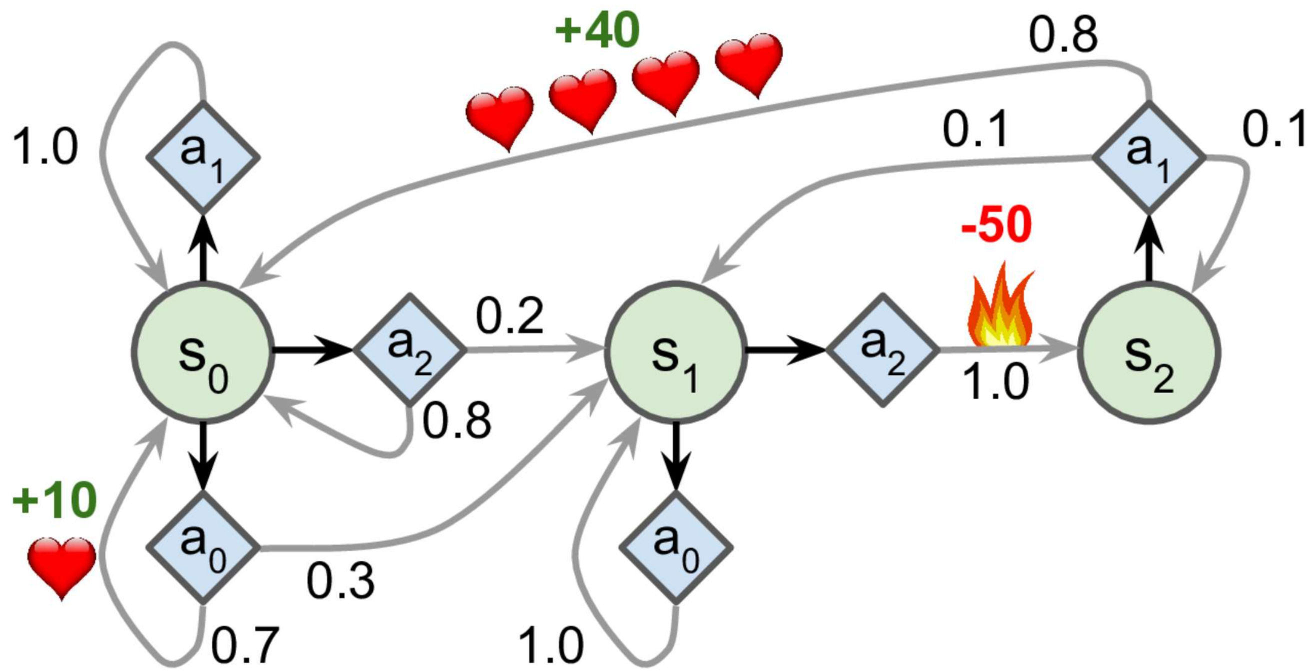
$$a_{ijk} = p(s_j | s_i, a_k) \quad \forall s_i, s_j \in \mathcal{S}, a_k \in \mathcal{A}$$

- Moreover, some state transitions return some reward (positive or negative).

- **Note that the reward is not mandatory.**

- The agent goal is: Given an MDP, find a policy that maximizes reward over time.

- Let's explore the following MDP with three states and up to three actions:



- **Example 2:** Simulate a sequence of state-action of fixed length given the MDP, assuming that all actions are equally probable (No policy bias).

- Do you imagine a strategy that make you gain the most reward over time?

Optimal State Value and Quality Value

Thursday, March 25, 2021 5:02 PM

- The **optimal state value** of any state s, noted $V^*(s)$ is the sum of all discounted future rewards the agent can expect on average after it reaches a state s, assuming **it acts optimally**.

- If the **agent acts optimally**, then the optimal value of the current state is equal to the reward it will get on average after taking one optimal action, plus the expected optimal value of all possible next states that this action can lead to (Bellman Optimality Equation):

$$V^*(s) = \max_a \sum_{s \in \mathcal{S}} T(s,a,s')[R(s,a,s') + \gamma V^*(s')] \; \forall \, s \in \mathcal{S}$$

$T(s,a,s')$ is the transition probability from state s to state s', given that the agent chose action a.

$R(s,a,s')$ is the reward that the agent gets when it goes from state s to state s', given that the agent chose action a.

γ is a discount factor. The larger the factor, the more the agent values the future reward.

- Since the Bellman Optimality Equation is recursive, the solution is the **Value Iteration** algorithm:
    - Initialize all $V(s) = 0$
    - $V_{k+1}(s) = \max_{a} \sum_{s \in \mathcal{S}} T(s, a, s')[R(s, a, s') + \gamma V_k(s')]$
- Note that the BOE does not provide a policy.
- Fortunately, Bellman developed the algorithm to estimate the optimal **state-action** values, aka Q-Values (Quality Values).
- The optimal Q-Value of the state-action pair (s, a), noted $Q^*(s, a)$, is the sum of discounted future rewards the agent can expect on average after it reaches the state s and chooses action a, but before it sees the outcome of this action, assuming **it acts optimally** after that action:

$$Q_{k+1}(s,a) = \sum_{s' \in \mathcal{S}} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

- Given the optimal Q-Values, defining the optimal policy when the agent is in state s is trivial:

$$\pi^*(s) = \max_a Q^*(s,a)$$

- **Example 3:** Apply Q-Value Iteration algorithm