CSCI E-82a
Probabilistic Programming and AI
Lecture 6
Markov Processes

Steve Elston

Introduction to Markov Processes

- What is a Markov process?
- Markov reward processes
- Utility and Gain
- Markov decision processes

Introduction to Markov Processes

- A **Markov process** has **states**
- A Markov process **transitions between states** at discrete time steps
- The probability of transition from one state to another for a **first order Markov process** is determined only by the **current state**:

$$p[S_{t+1} \mid S_1, \ldots, S_t] = p[S_{t+1} \mid S_t]$$

- Where, the history of states is, $S_1, \ldots, S_t$
- And, the current state is S_t
- A first order Markov process has **no memory**

Example of Markov Processes
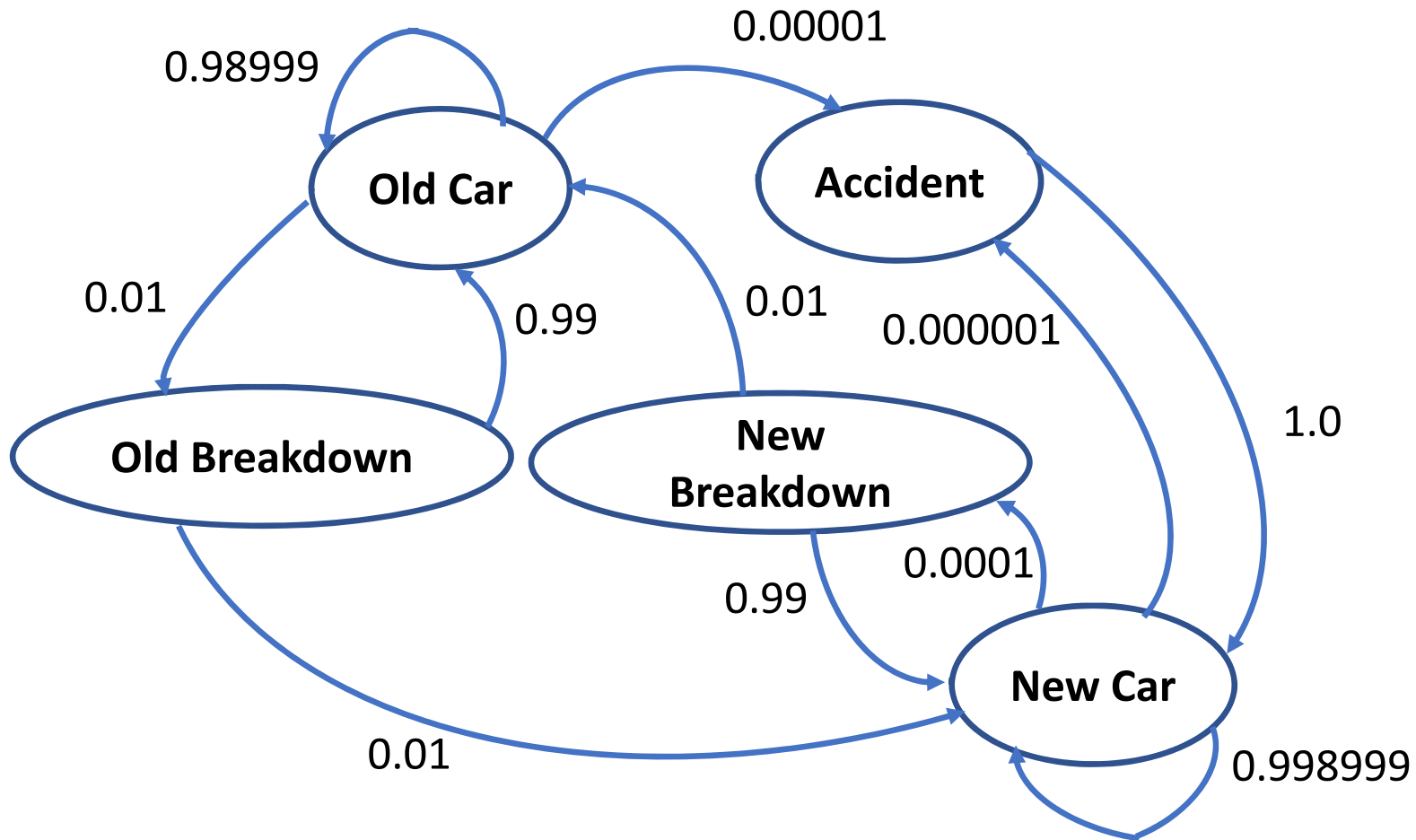
Introduction to Markov Processes

- A Markov process is characterized by a **state probability transition matrix**:

$$\mathcal{P}_{ss'} = \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \dots & P_{nn} \end{bmatrix}$$

- Where, $\mathcal{P}_{ij}$ = probability of transition from state s_i to s_j

Introduction to Markov Processes

- The probability of transition from one state to the next state is computed with the state transition probability matrix:

$$S' = \mathcal{P}_{ss'}S$$

- Or,

$$
\begin{bmatrix} s'_1 \\ \vdots \\ s'_n \end{bmatrix} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}
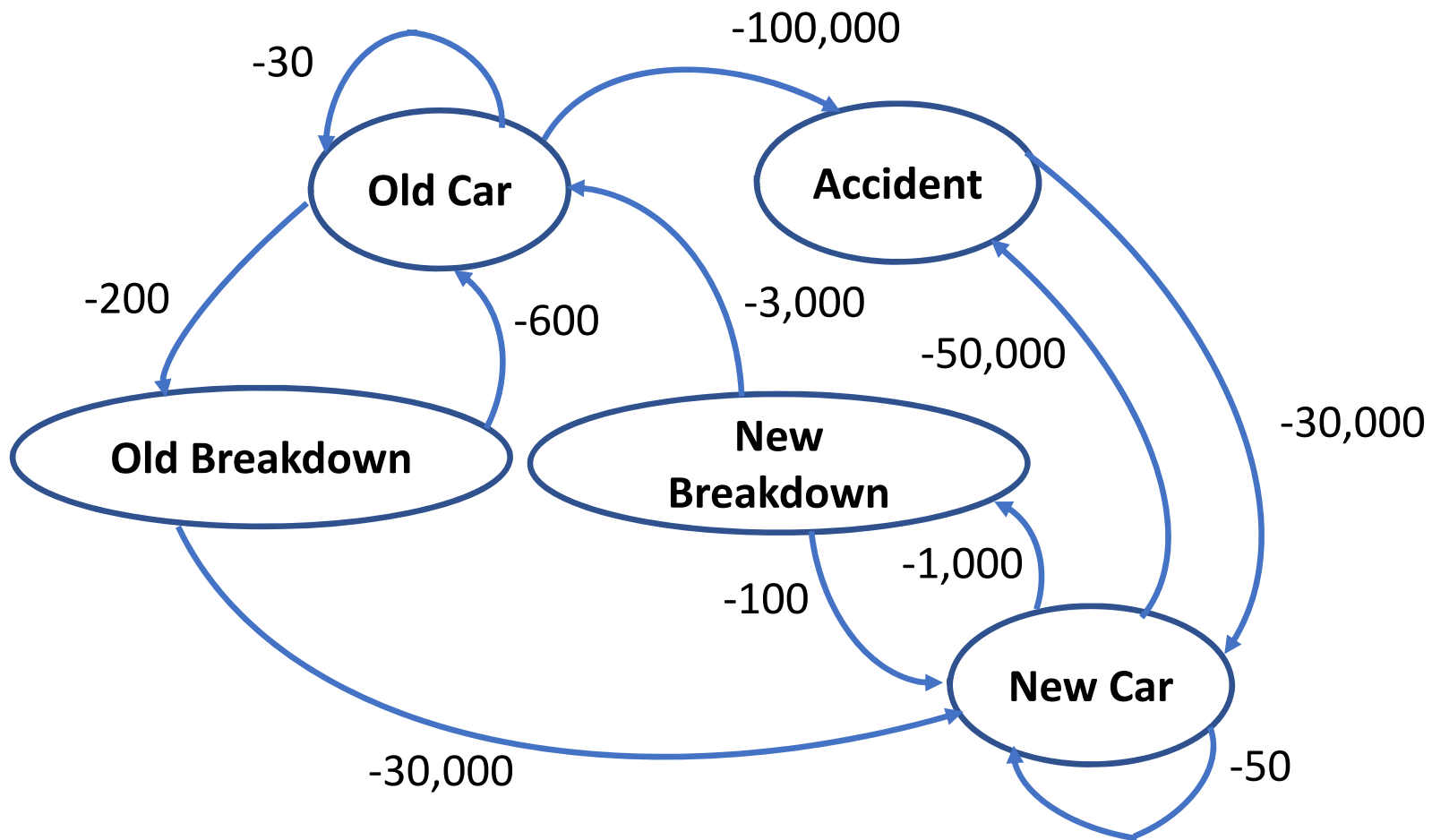$$

Introduction to Markov Processes

- A **Markov chain** is a sequence of Markov state transition processes
  - E.g. running a Markov process over several time steps creates a Markov chain
- If the state transition probability matrix, $\mathcal{P}_{ss'}$, does not change with time, the Markov chain is **stationary**
- Stationary Markov chains **converge to a steady state**
  - At steady state the state probabilities are unchanged

Introduction to Markov Reward Processes

- A **Markov reward process** generates a reward or change in utility for each state transition

- Reward can be positive or negative

- Reward may not follow economic value
  - The inconvenience of a car breakdown may exceed the cost of repair
  - A piece of art has aesthetic value

- The **reward function** for a transition from state S_t to state S_{t+1} is defined:
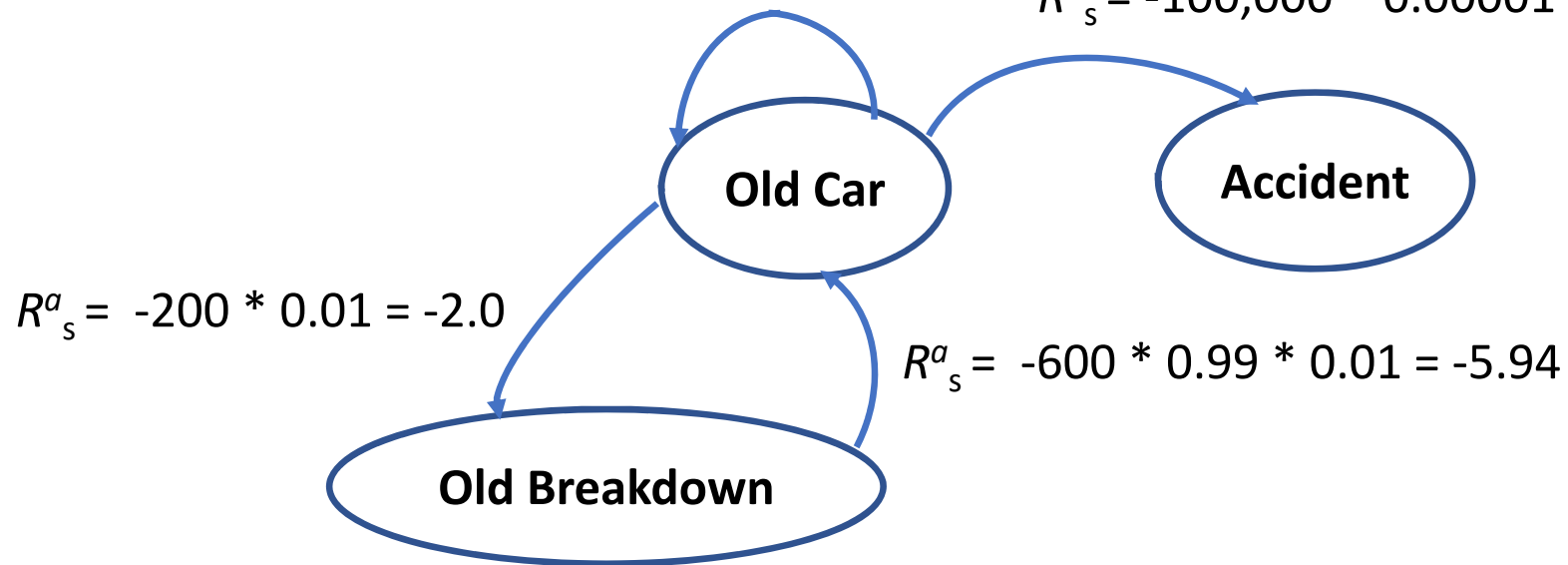
$$\mathcal{R}_{ss'} = E\left[R_{t+1} \mid S_t = s\right]$$

Example of Markov Reward Process

Example of Markov Reward Process

$R^a_s = -30 * 0.98999 = -29.6997$

$R^a_s = -100,000 * 0.00001 = -1.0$

Old Car

Accident

$R^a_s = -200 * 0.01 = -2.0$

$R^a_s = -600 * 0.99 * 0.01 = -5.94$

Old Breakdown

$E[R(Old\ Car)] = -29.6997 - 1.0 - 2.0 - 5.94 = -38.6397$

Utility and Gain

- **Utility** is the sum of rewards in a Markov chain
- Since the rewards are **additive**:

$$U([s_o, s_1, \dots, s_T]) = R(s_o) + R(s_1) + \dots + R(s_T) = \sum_{t=0}^{T} R(s_t)$$

- The above formulation works for an **episodic process**
- An episodic process has a **terminal state**

Utility and Gain

- What happens to the **Utility** for a **continuous process**

- A continuous process has **no termination state** and the **utility is unbounded**

  As $\quad T \to \infty \quad U(s_t) \to \infty$

- So, apply a **discount factor** at each time step:

$$U([s_o, s_1, s_2, s_3 \ldots]) = R(s_o) + \gamma R(s_1) + \gamma^2 R(s_2) + \gamma^3 R(s_3) \ldots = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- Properties of discounted returns

  As $\gamma \to 0$, the reward process becomes myopic, only counting near term rewards

  As $\gamma \to 1$, the reward process becomes far sighted, valuing distant rewards highly.

Utility and Gain

- The **Gain** at time t is the sum of future rewards in a Markov chain:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- We can define the **state-value function**:

$$v(s) = E[G_t \mid S_t = s]$$

Introduction to Markov Decision Processes

How can you compute expectation for a probabilistic process?

- The expectation is the sum of the probability weighted values of the states

$$\mathbb{E}(states) = \sum_{states} p(state)\, value(state)$$

- The normalization of the expectation is correct since:

$$\sum_{states} p(state) = 1.0$$

Introduction to Markov Decision Processes

A **Markov decision process** is a tuple $<S, A, P, R, \gamma>$:

- S is a finite **set of states**

- A is a finite **set of actions**

- P is the **state transition probability matrix**

$$P^a_{ss'} = P[S_{t+1} = s' \mid S_t = s]$$

- $R(s,a)$ is a **reward function** with **expected reward given the state and action**

$$R^a_s = E[R_t \mid S_t = s, A_t = a]$$

$$R^a_s = \sum\nolimits_a \text{ reward for state transition * probability of transition}$$

Introduction to Markov Decision Processes

A **policy, π,** is probability distribution over actions given states:

$$\pi(a|s) = P[A_t = a \mid S_t = s]$$

- The policy **fully defines agent behavior**
- The MDP **depends only on current state**, not history
- A given policy is **stationary**; does not change in time

Introduction to Markov Decision Processes

Given a **Markov decision process** tuple $<S, A, P, R, \gamma>$ and policy π:

- *Let $S_1, S_2, S_3, ..., S_t$* be a **state sequence** determined by a **Markov process**
- Then, **probability of state transitions** and **expected rewards** are:

$$P^{\pi}_{ss'} = \sum_A \pi(a|s) \, P^a_{ss'}$$

$$R^{\pi}_s = \sum_A \pi(a|s) \, R^a_{ss'}$$

- In other words, the expected reward is the **sum over all actions of the policy** (probability of action) **times the reward** for the action

Optimal Policy for MDP

- Actions of the agent are determined by a **policy, π**
- The expectation of the policy determines the **action-value**

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- Goal is to **learn** an **optimal policy**

$$q_{\pi^*}(s, a) = \mathbb{E}_{\pi^*}[G_t \mid S_t = s, A_t = a]$$

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

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