



Machine Learning 410

Lesson 8

Introduction to Markov Processes

Steve Elston

Copyright 2019, Stephen F Elston. All rights reserved.

Introduction to Markov Processes

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

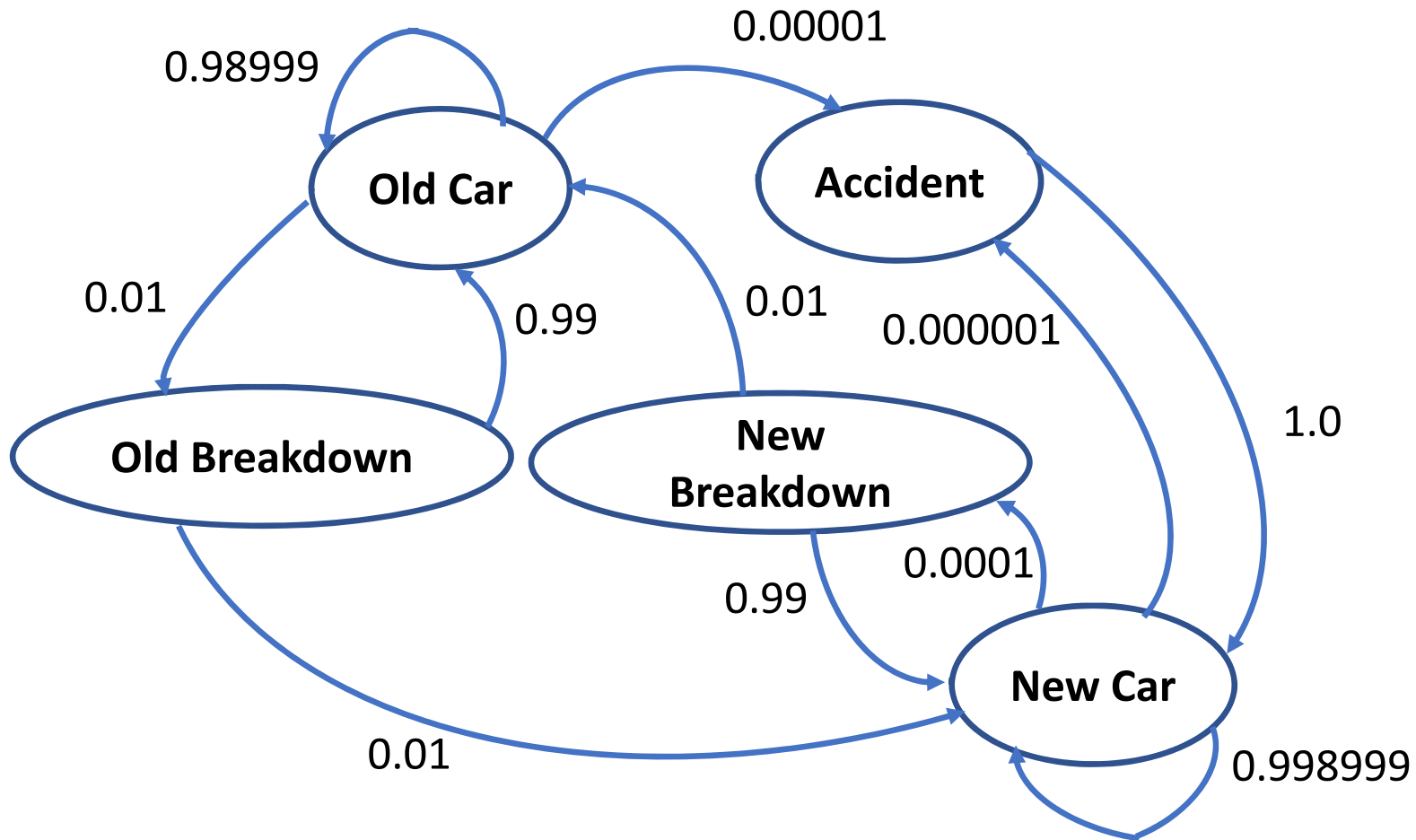
Introduction to Markov Processes

- 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**:

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

- Where, the history of states is, S_1, \dots, 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:

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

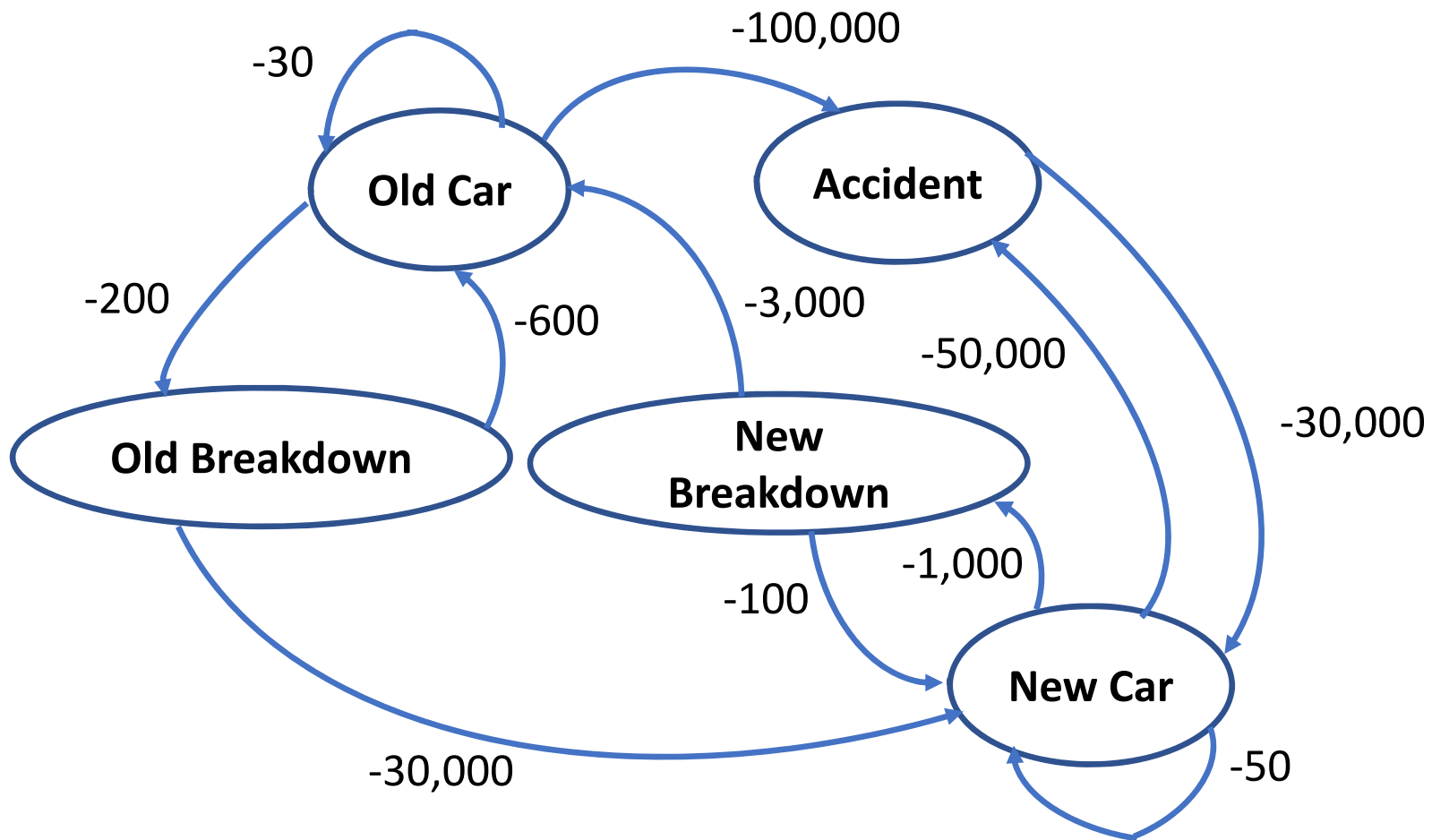
- Or,

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

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

Example of Markov Reward Process



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:

$$\mathcal{R}_{ss'} = E[R_{t+1} \mid S_t = s]$$

Introduction to Markov Reward

How can you compute expectation for a probabilistic process?

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

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

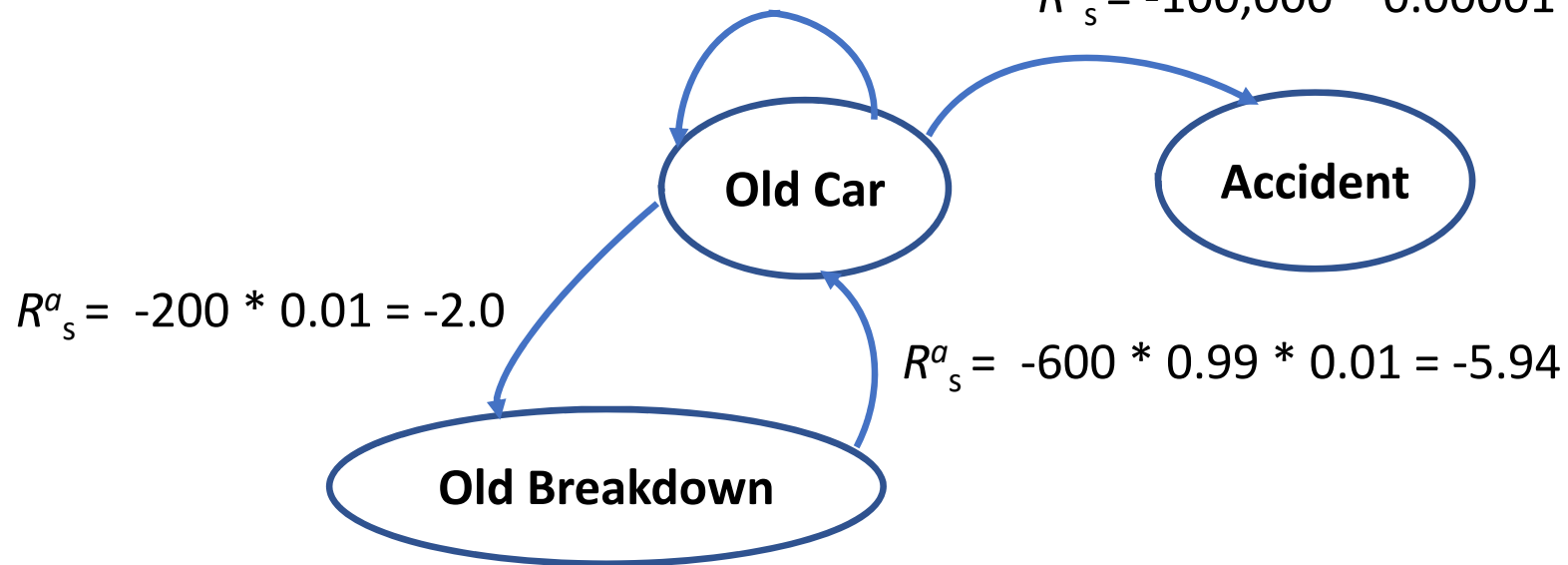
- The normalization of the expectation is correct since:

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

Example of Markov Reward Process

$$R_s^a = -30 * 0.98999 = -29.6997$$

$$R_s^a = -100,000 * 0.00001 = -1.0$$



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

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**

$$\text{As } T \rightarrow \infty \quad U(s_t) \rightarrow \infty$$

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

$$U([s_0, s_1, s_2, s_3 \dots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \gamma^3 R(s_3) \dots = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- Properties of discounted returns

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

As $\gamma \rightarrow 1$, 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

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

- S is a finite **set of states**
- A is a finite **set of actions**
- P is the **state transition probability matrix**

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

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

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

$$R_s^a = \sum_a \text{reward for state transition} * \text{probability of transition}$$

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 $\langle S, A, P, R, \gamma \rangle$ and policy π :

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

$$P_{ss'}^\pi = \sum_A \pi(a|s) P_{ss'}^a$$

$$R_s^\pi = \sum_A \pi(a|s) R_{ss'}^a$$

- 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:

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