



STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

Markov Models

SYS 611: Systems Modeling and Simulation

Paul T. Grogan, Ph.D.
Assistant Professor
School of Systems and Enterprises





Agenda

1. Discrete Time Markov Chains
2. Continuous Time Markov Processes

Reading: R.C. Larson and A.R. Odoni, “Introduction to Queuing Theory and Its Applications,” Ch. 4 in *Urban Operations Research*, 2007, pp. 182-211.
([Web Version Available](#))



Discrete Time Markov Chains



Discrete Time Markov Chains

- Discrete space and time

- State $q(t) \in \{0,1, \dots\}$
- State transition function:

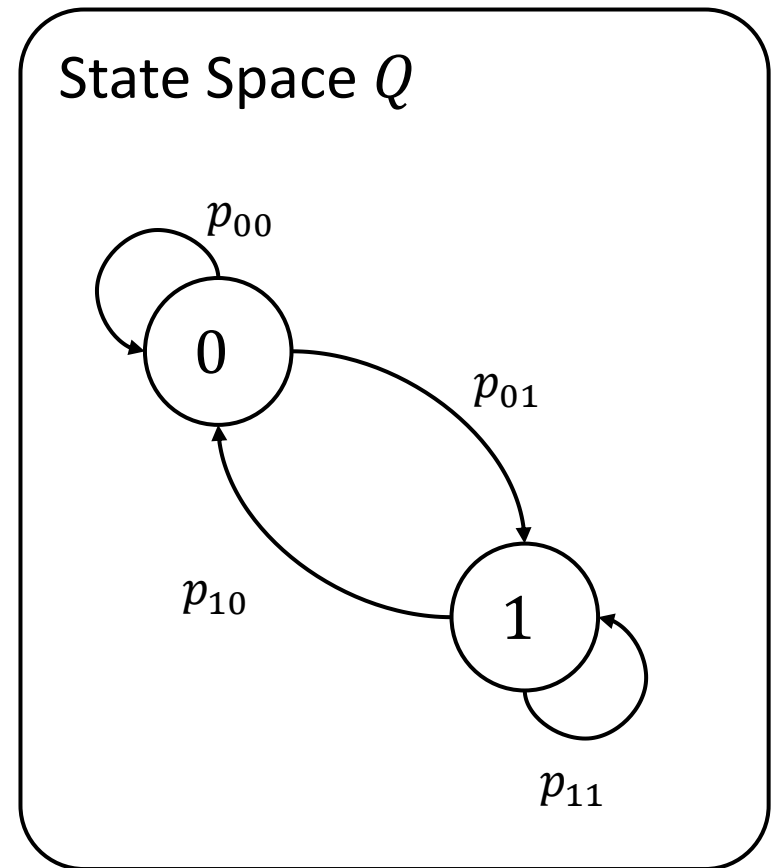
$$\delta(q) = q(t + 1)$$

- Stochastic transitions:

$$\delta(i) = j \text{ with prob. } p_{ij}$$

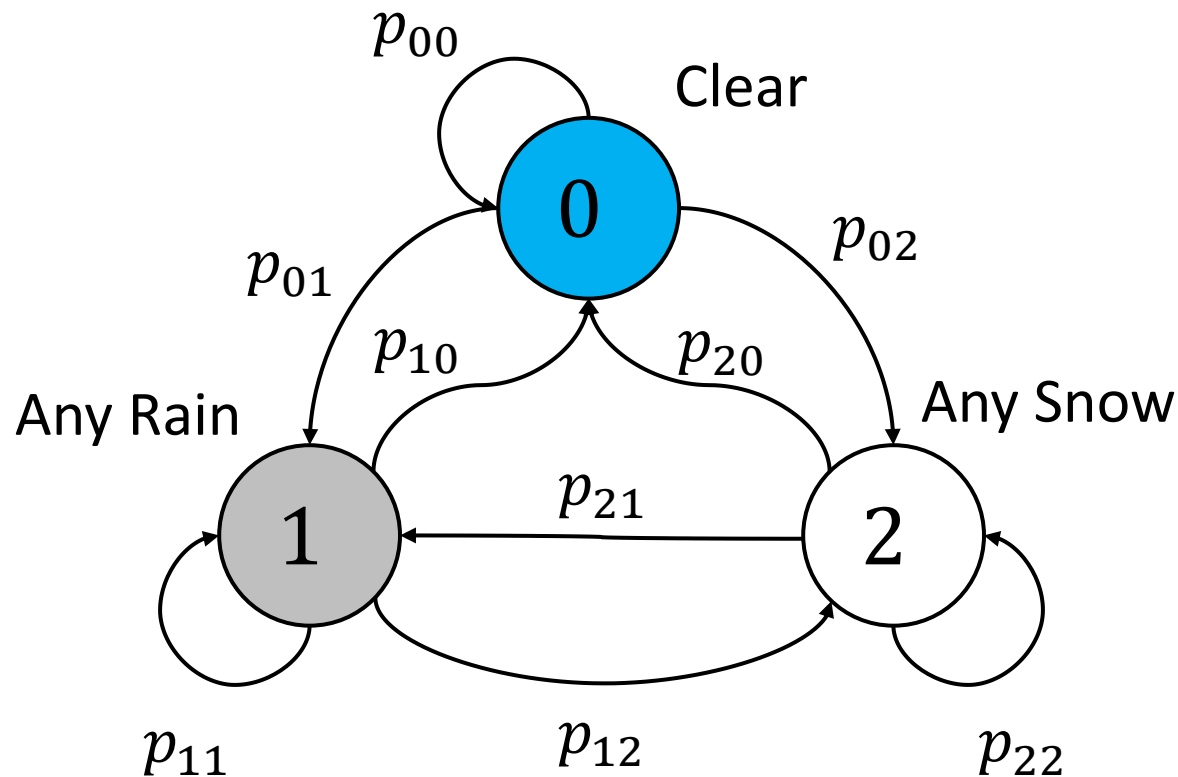
- Markov property:
memoryless

- State transitions *only* based on current state



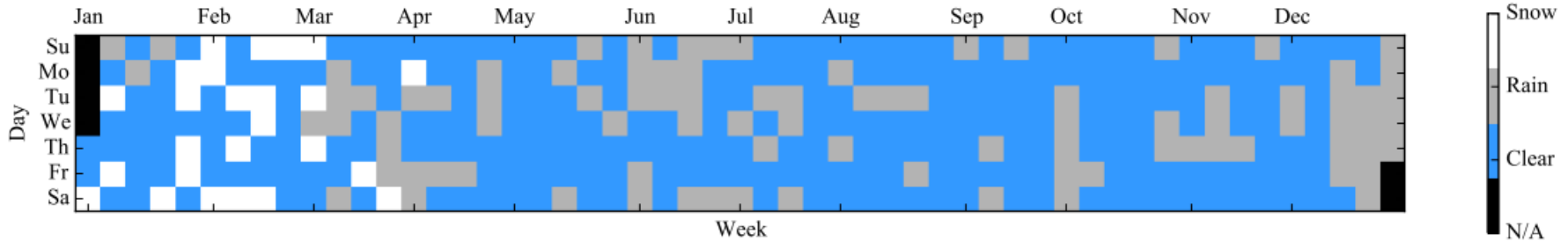
Example: Weather Model

Estimate the probabilities:

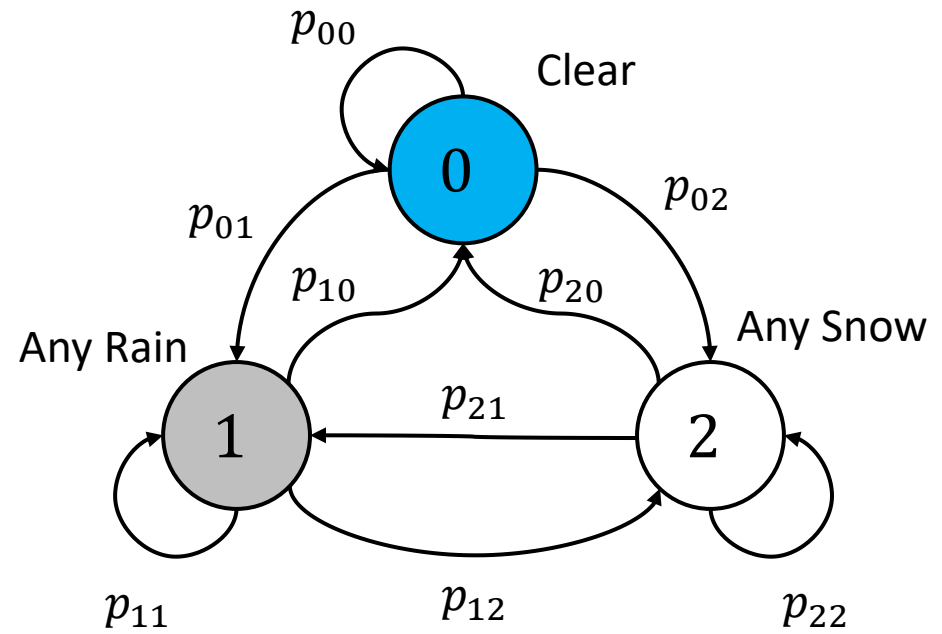




Weather in Central Park

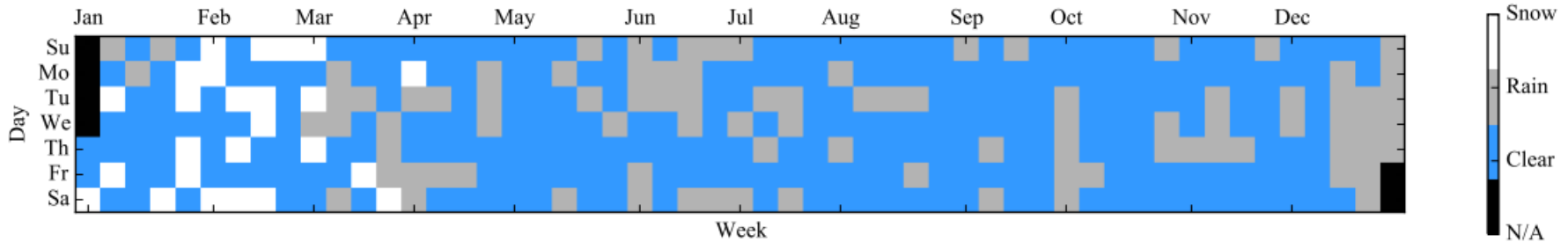


- Data from 2015:
 - 250 clear days
 - 89 rainy days
 - 25 snowy days
- Build Markov model with observed PMF





Weather in Central Park

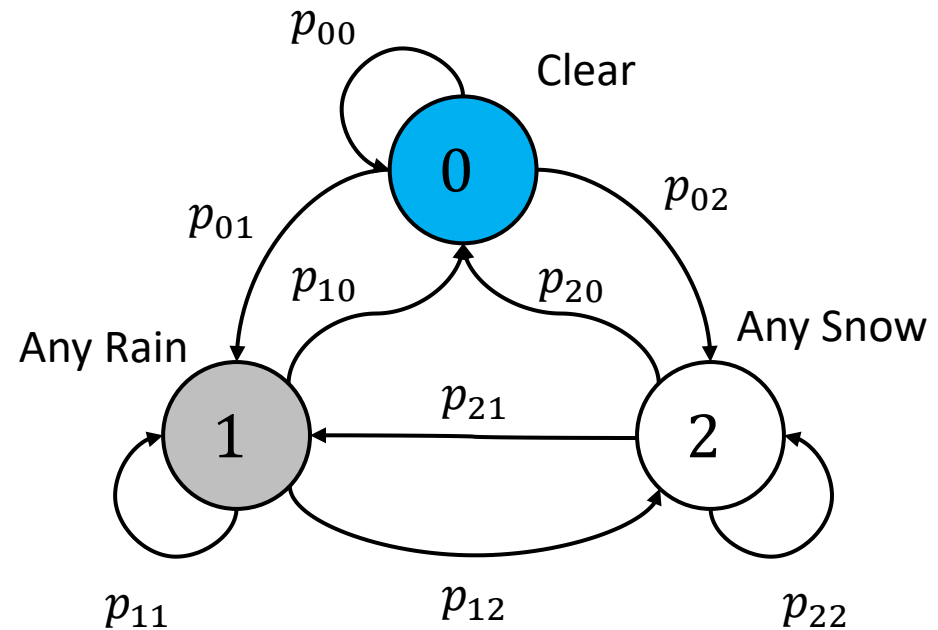


- Data from 2015:

- F_{ij} : frequency state i is followed by state j

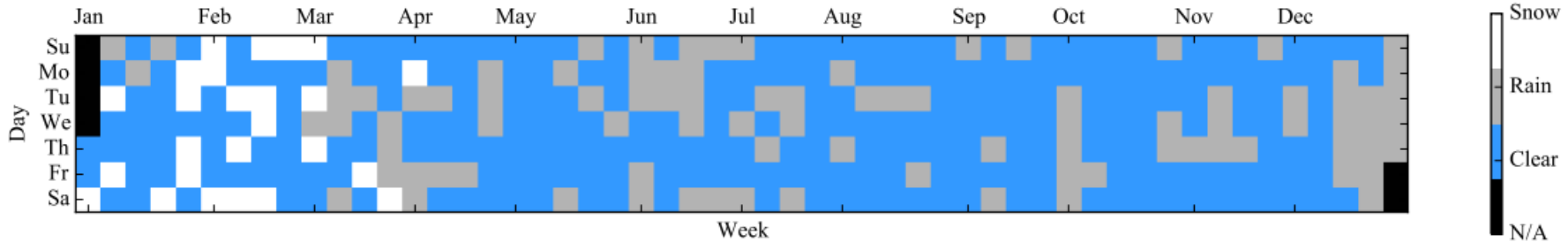
$$F = [F_{ij}] = \begin{bmatrix} 186 & 47 & 17 \\ 47 & 40 & 2 \\ 16 & 3 & 6 \end{bmatrix}$$

$$p_{ij} = \frac{F_{ij}}{\sum_{k=0}^2 F_{ik}}$$



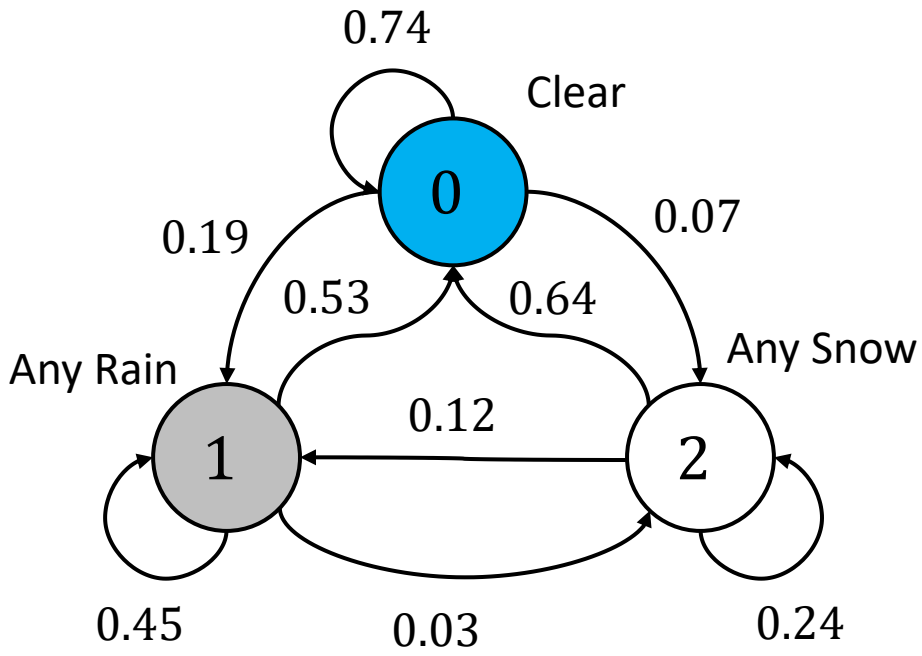


Weather in Central Park



$$P = \begin{bmatrix} \overline{186} & \overline{47} & \overline{17} \\ \overline{250} & \overline{250} & \overline{250} \\ \overline{47} & \overline{40} & \overline{2} \\ \overline{89} & \overline{89} & \overline{89} \\ \overline{16} & \overline{3} & \overline{6} \\ \overline{25} & \overline{25} & \overline{25} \end{bmatrix}$$

$$P = \begin{bmatrix} 0.74 & 0.19 & 0.07 \\ 0.53 & 0.45 & 0.03 \\ 0.64 & 0.12 & 0.24 \end{bmatrix}$$



Process Generator (IVT)

$$P = \begin{bmatrix} 0.74 & 0.19 & 0.07 \\ 0.53 & 0.45 & 0.03 \\ 0.64 & 0.12 & 0.24 \end{bmatrix}$$



If $q(t) = 0$:

$$q(t+1) = \begin{cases} 0 & \text{with prob. } p_{00} \\ 1 & \text{with prob. } p_{01} \\ 2 & \text{with prob. } p_{02} \end{cases}$$

If $q(t) = 1$:

$$q(t+1) = \begin{cases} 0 & \text{with prob. } p_{10} \\ 1 & \text{with prob. } p_{11} \\ 2 & \text{with prob. } p_{12} \end{cases}$$

If $q(t) = 2$:

$$q(t+1) = \begin{cases} 0 & \text{with prob. } p_{20} \\ 1 & \text{with prob. } p_{21} \\ 2 & \text{with prob. } p_{22} \end{cases}$$

Random (0,1) sample r

If $q(t) = 0$:

$$q(t+1) = \begin{cases} 0 & \text{if } 0 \leq r \leq 0.74 \\ 1 & \text{if } 0.74 < r \leq 0.74 + 0.19 \\ 2 & \text{if } 0.74 + 0.19 < r \leq 1 \end{cases}$$

If $q(t) = 1$:

$$q(t+1) = \begin{cases} 0 & \text{if } 0 \leq r \leq 0.53 \\ 1 & \text{if } 0.53 < r \leq 0.53 + 0.45 \\ 2 & \text{if } 0.53 + 0.45 < r \leq 1 \end{cases}$$

If $q(t) = 2$:

$$q(t+1) = \begin{cases} 0 & \text{if } 0 \leq r \leq 0.64 \\ 1 & \text{if } 0.64 < r \leq 0.64 + 0.12 \\ 2 & \text{if } 0.64 + 0.12 < r \leq 1 \end{cases}$$



Process Generator (Excel)

- Build multiple VLOOKUP tables using transition probability values p_{ij}
- Chain together multiple IF functions to switch between VLOOKUP tables based on state

	A	B	C	D	E	F	G	H	I	J	K	L	M				
1	t	r	q	q(t+1)		vlookup table for q=0			vlookup table for q=1			vlookup table for q=2					
2		0	0.78353	0	=IF(C2=0,VLOOKUP(B2,\$F\$3:\$G\$5,2),IF(C2=1,VLOOKUP(B2,\$I\$3:\$J\$5,2),VLOOKUP(B2,\$L\$3:\$M\$5,2)))												
3		1	0.537566	1		0	0		0	0		0	0				
4		2	0.60236	1		0.744	1		0.52809	1		0.64	1				
5		3	0.814086	1		0.932	2		0.977528	2		0.76	2				
6		4	0.862817	1													
7		5	0.04518	1													

- Can also use IF functions instead of VLOOKUP:

IF (B2<0.744 , 0 , IF (B2<0.932 , 1 , 2))



Process Generator (Python)

- Define state transition function `next_state`
- Use two layers of if/elif/else blocks:
 - Layer 1: switch based on current state `q`
 - Layer 2: switch based on sampled `r` (IVT)

```
def next_state(q):  
    r = np.random.rand()  
    if q == 0:  
        if r < p_00:  
            return 0  
        elif r < p_00 + p_01:  
            return 1  
        else:  
            return 2  
    elif q == 1:  
        if r < p_10:  
            return 0  
        elif r < p_10 + p_11:  
            return 1  
        else:  
            return 2  
    ...
```



Stochastic Markov State

- Can also view state as stochastic vector:

$$\mathbf{q}^{(t)} = [q_0 \quad q_1 \quad q_2], \quad \text{where } q_i = P\{q(t) = i\}$$

- Deterministic initial state (clear day, $q(0) = 0$):

$$\mathbf{q}^{(0)} = [1 \quad 0 \quad 0]$$

- Simpler state transitions:

$$\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)} \times P,$$

$$\mathbf{q}^{(t+1)} = [q_0 \quad q_1 \quad q_2] \times \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix}$$



Stochastic Markov State

What is the PMF for tomorrow's weather if it is clear today?

$$\mathbf{q}^{(0)} = [1 \quad 0 \quad 0]$$

$$\mathbf{q}^{(1)} = \mathbf{q}^{(0)}P$$

$$\mathbf{q}^{(1)} = [1 \quad 0 \quad 0] \cdot \begin{bmatrix} 0.74 & 0.19 & 0.07 \\ 0.53 & 0.45 & 0.03 \\ 0.64 & 0.12 & 0.24 \end{bmatrix}$$

$$\mathbf{q}^{(1)} = [0.74 \quad 0.19 \quad 0.07]$$

What is the PMF for the next day's weather?

$$\mathbf{q}^{(0)} = [1 \quad 0 \quad 0]$$

$$\mathbf{q}^{(2)} = \mathbf{q}^{(1)}P = \mathbf{q}^{(0)}P^2$$

$$\mathbf{q}^{(2)} = [1 \quad 0 \quad 0] \cdot \begin{bmatrix} 0.74 & 0.19 & 0.07 \\ 0.53 & 0.45 & 0.03 \\ 0.64 & 0.12 & 0.24 \end{bmatrix}^2$$

$$\mathbf{q}^{(2)} = [0.70 \quad 0.23 \quad 0.07]$$



Steady-state Analysis

- Over long durations, a Markov chain will converge to either:
 - Terminal state/states
 - Stationary stochastic distribution π
- Limiting distribution π exists mathematically if every state is:
 - Positive recurrent (return-able)
 - Aperiodic (reachable from any state)

$$\pi = \lim_{n \rightarrow \infty} q^{(n)}$$

$$\pi P = \pi$$

$$\pi(P - I) = \mathbf{0}$$

$$\pi \cdot \begin{bmatrix} 0.74 - 1 & 0.19 & 0.07 \\ 0.53 & 0.45 - 1 & 0.03 \\ 0.64 & 0.12 & 0.24 - 1 \end{bmatrix} = 0$$

(note: π is the normalized eigenvector of P corresponding to the eigenvalue 1)

$$\pi = [0.68 \quad 0.25 \quad 0.07]$$



Markov Chain Monte Carlo

- Can sample from complex distributions by building a Markov Chain with the appropriate stationary stochastic distribution π
 - Only sample after sufficient “burn-in” period
 - Does *not* provide independent samples, need to skip every N samples
- Applications to numerical integration
 - Can estimate non-integrable functions
 - Bayesian inference to compute posterior distributions

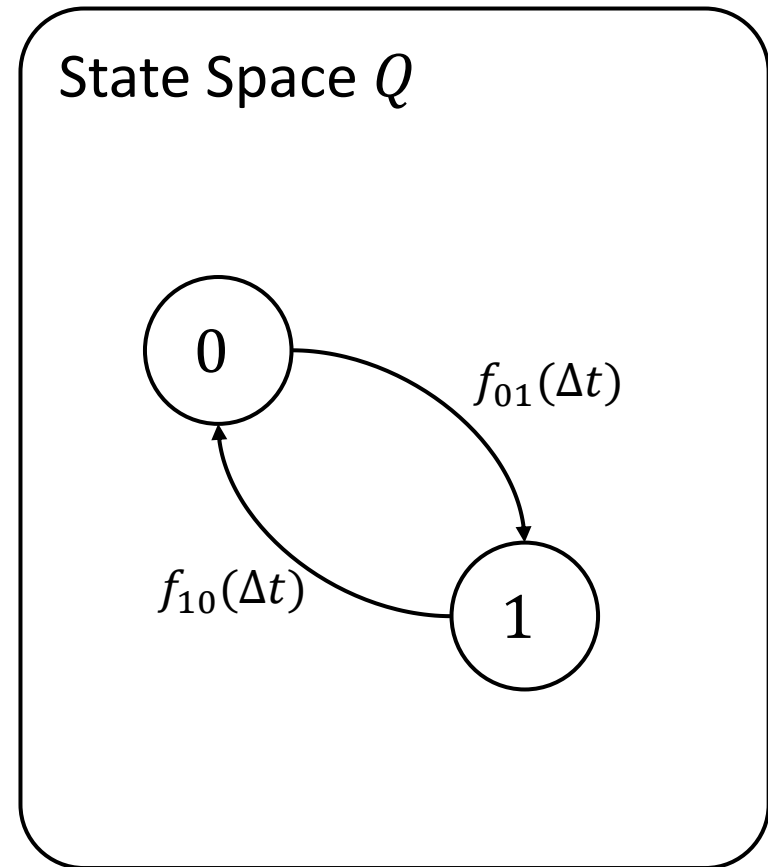


Continuous Time Markov Processes



Continuous Time Markov Processes

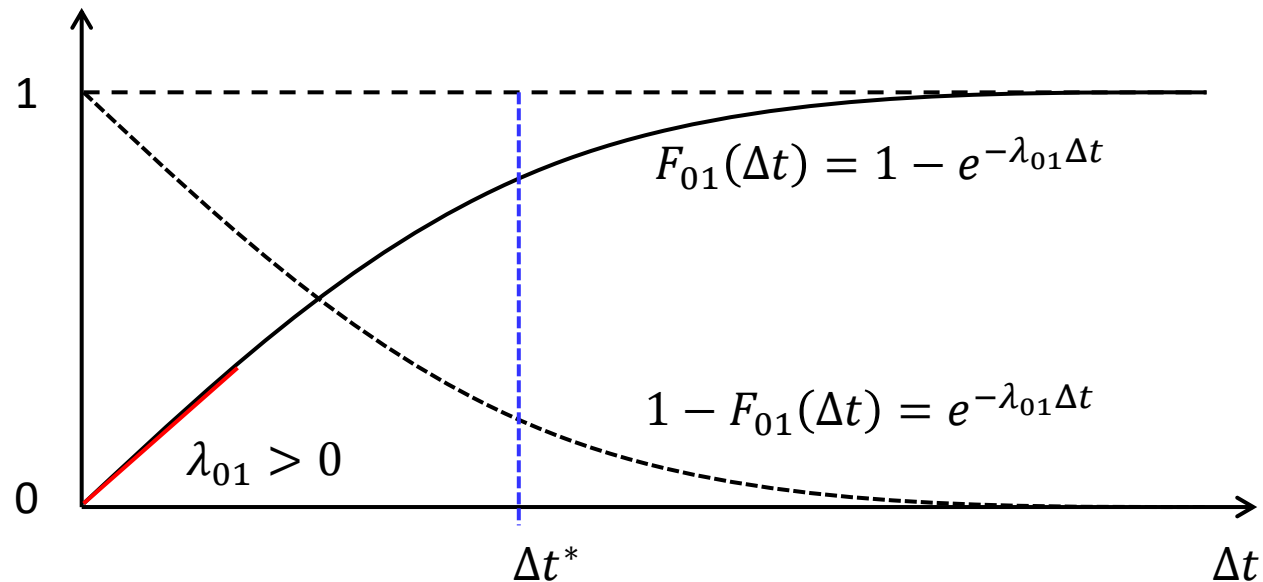
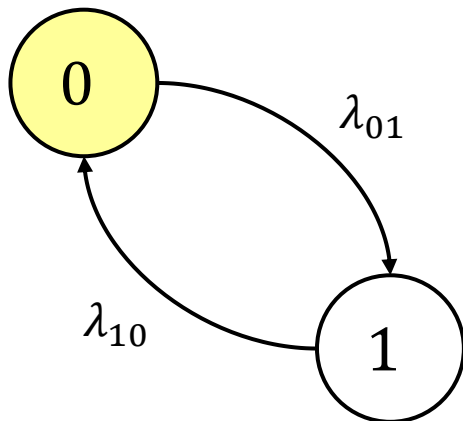
- Discrete space, continuous time
 - State $q(t) \in \{0,1, \dots\}$
 - State transition function:
$$\delta(q) = q(t + \Delta t)$$
 - Stochastic transitions:
$$\delta(i) = j \text{ with prob. } f_{ij}(\Delta t)$$
- Markov property: **memoryless**
 - State transitions *only* based on current state: $f_{ij}(\Delta t) \rightarrow \lambda_{ij}$



Markov Property in Continuous Time

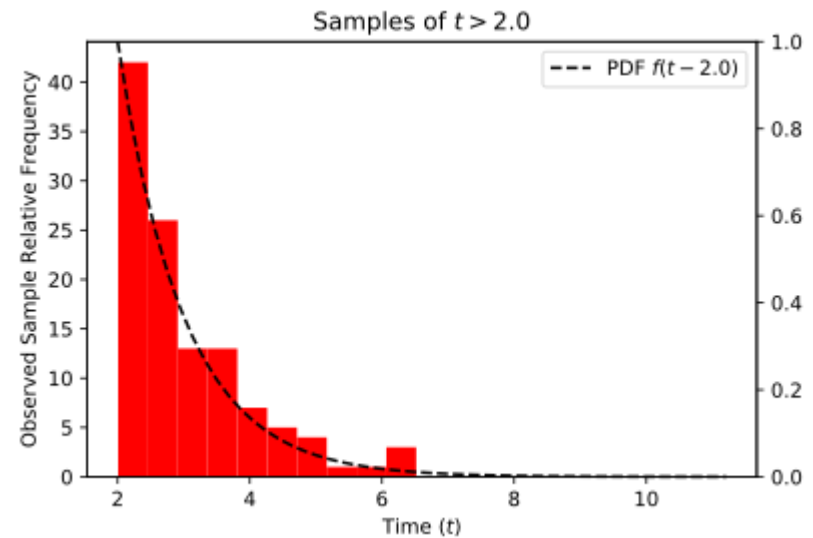
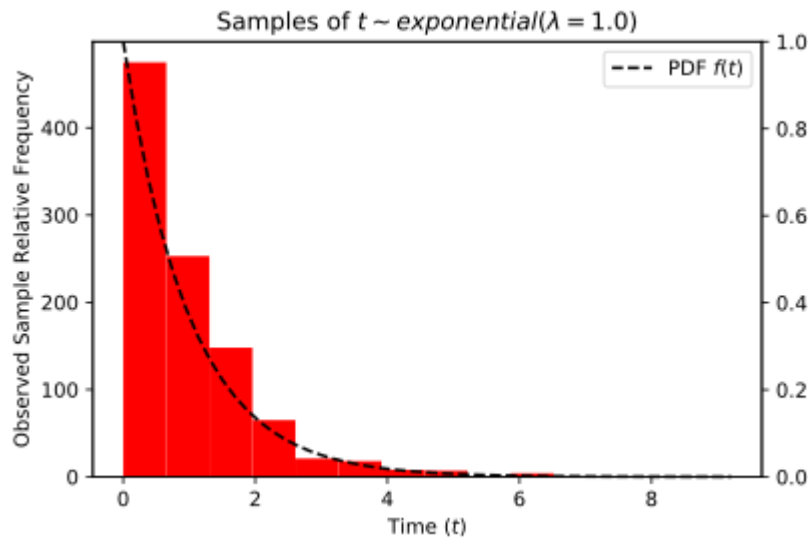
- Markov property requires state changes to be memoryless... only one possible distribution:

$$\Delta t \sim \text{exponential}(\lambda_{ij}) \Rightarrow F_{ij}(\Delta t) = 1 - e^{-\lambda_{ij}\Delta t} \quad (i \neq j)$$



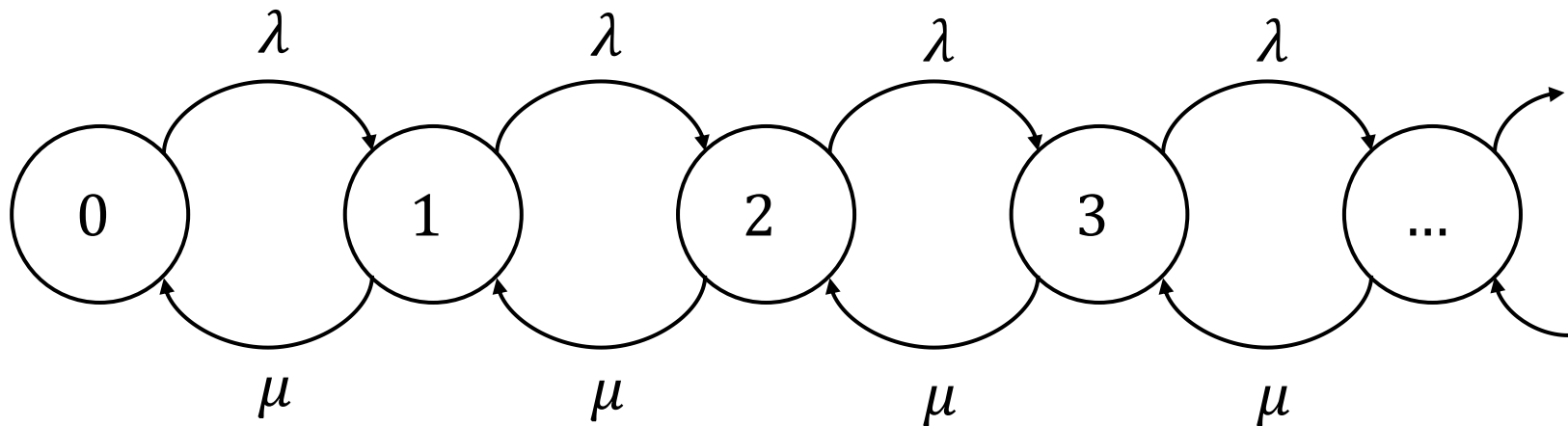
Example: Conditional Samples

- Sample transition time with 1.0 transitions/minute:
 $\Delta t \sim \text{exponential}(\lambda = 1.0)$
- Generate 1000 samples and plot histogram
- Waited 2.0 minutes without a transition... what is the conditional distribution?
- Total time: $\Delta t = 2.0 + \Delta t^*$
- $\Delta t^* \sim \text{exponential}(\lambda = 1.0)$



Example: Queuing Model

- State represents number of customers in queue
- Exponential inter-arrival (X) with rate λ cust./min.
- Exponential service time (Y) with rate μ cust./min.
- (Also called a "birth and death" process)





Exponential Process Generators

- $X \sim \text{exponential}(\lambda)$:
inter-arrival period

$$f(x) = \lambda e^{-\lambda x}$$

$$F(x) = 1 - e^{-\lambda x}$$

- X is the time until next customer arrival
- IVT process generator

$$x = \frac{-\ln(1 - r_x)}{\lambda}$$

- $Y \sim \text{exponential}(\mu)$:
service time

$$f(y) = \mu e^{-\mu y}$$

$$F(y) = 1 - e^{-\mu y}$$

- Y is the time until next service completion
- IVT process generator

$$y = \frac{-\ln(1 - r_y)}{\mu}$$

Markov Process Generator (IVT)

If $q(t) = 0$:

$$q(t + x) = 1$$

If $q(t) > 0$:

$$q(t + \Delta t) = \begin{cases} q(t) + 1 & \text{if } x < y \\ q(t) - 1 & \text{if } y < x \end{cases}$$

$$\Delta t = \min(x, y)$$

$$x = \frac{-\ln(1-r_x)}{\lambda}$$

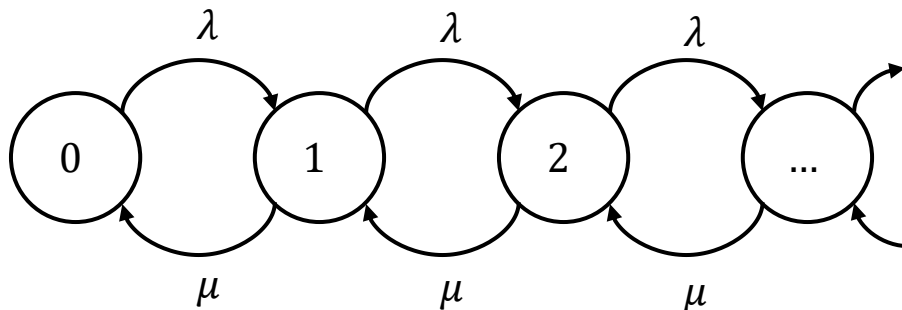
$$y = \frac{-\ln(1-r_y)}{\mu}$$

If $q(t) = 0$ or $x < y$:

$$q(t + x) = q(t) + 1$$

If $q(t) > 0$ and $y < x$:

$$q(t + y) = q(t) - 1$$





Replacing Time with Events

- Easier to index states with event i instead of time t

$$\delta(q) = q(i + 1) = \begin{cases} q(i) + 1, & \text{if } q(i) = 0 \text{ or } x < y \\ q(i) - 1, & \text{if } q(i) > 0 \text{ and } y < x \end{cases}$$

- Define event duration as a derived state variable

$$\Delta t(i) = \begin{cases} x, & \text{if } q(i) = 0 \text{ or } x < y \\ y, & \text{if } q(i) > 0 \text{ and } y < x \end{cases}$$

- Time becomes a new state variable to be updated

$$\delta(t) = t(i + 1) = t(i) + \Delta t(i)$$



Café Java (Class Problem 4-2*)

Café Java's manager wants to model the queuing system to see if it can handle the evening coffee rush hour. Assume:

- Inter-arrival times: $X \sim \text{exponential}(\lambda = 1/1.5)$
 - 1.5 min. per customer \rightarrow 0.67 customers per minute
- Service times: $Y \sim \text{exponential}(\mu = 1/0.75)$
 - 0.75 min. per customer \rightarrow 1.33 customers per minute



Café Java State Transitions

$$q(0) = 0, \quad t(0) = 0.00$$

$$x = 0.43 < y = 0.90$$

$$q(0 + 1) = q(0) + 1, \quad \Delta t(0) = 0.43$$

$$\Rightarrow q(1) = 1, \quad t(1) = 0.43$$

$$x = 4.49 > y = 1.66$$

$$q(1 + 1) = q(1) - 1, \quad \Delta t(1) = 1.66$$

$$\Rightarrow q(2) = 0, \quad t(2) = 2.09$$

$$x = 0.95 > y = 0.02$$

$$q(2 + 1) = q(2) + 1, \quad \Delta t(2) = 0.95$$

$$\Rightarrow q(3) = 1, \quad t(3) = 3.04$$

$$x = \frac{-\ln(1 - r_x)}{1/1.5}, \quad y = \frac{-\ln(1 - r_y)}{1/0.75}$$

If $q(t) = 0$ or $x < y$:

$$q(t + x) = q(t) + 1$$

If $q(t) > 0$ and $y < x$:

$$q(t + y) = q(t) - 1$$

r_x	x	r_y	y
0.25	0.43	0.70	0.90
0.95	4.49	0.89	1.66
0.47	0.95	0.02	0.02

Markov Simulator (Excel)

- Include 2 continuous process generators
- Apply IF conditions to update time, state

	A	B	C	D	E	F	G	H	I	J
1	lambda=	0.666667		mu=	1.333333					
2										
3	i	t	q(i)	r_arrival	t_arrival	r_service	t_service	delta t	q(i+1)	
4	0	0	0	0.318991	0.576271	0.239259	0.205097	=IF(OR(C4=0,E4<G4),E4,G4)		
5	1	0.576271	1	0.173624	0.286059	0.046538	0.035742	0.035742	0	
6	2	0.612013	0	0.9715	5.336789	0.111304	0.0885	5.336789	1	

	A	B	C	D	E	F	G	H	I	J	K
1	lambda=	0.666667		mu=	1.333333						
2											
3	i	t	q(i)	r_arrival	t_arrival	r_service	t_service	delta_t	q(i+1)		
4	0	0	0	0.702319	1.817597	0.639585	0.765374	1.817597	=IF(OR(C4=0,E4<G4),C4+1,C4-1)		
5	1	1.817597	1	0.350154	0.64653	0.877957	1.577538	0.64653	2		
6	2	2.464127	2	0.633314	1.504875	0.082646	0.064697	0.064697	1		
7	3	2.528824	1	0.248823	0.429172	0.615099	0.716078	0.429172	2		



Markov Simulator (Python)

- Define state transition function `next_state`
- Include 2 continuous process generators
- Use if statements to return new time, state

```
N = 100
q = np.zeros(N)
t = np.zeros(N)
x = np.zeros(N)
y = np.zeros(N)
delta_t[i] = np.zeros(N)

for i in range(N):
    x[i] = gen_x()
    y[i] = gen_y()
    if q[i]==0 or x[i]<y[i]:
        delta_t[i] = x[i]
        q[i+1] = q[i] + 1
    else:
        delta_t[i] = y[i]
        q[i+1] = q[i] - 1
    t[i+1] = t[i] + delta_t[i]
```



Other Caveats/Limitations

- This simulation approach is **only valid** for Markov models with exponential inter-arrival/service periods
 - Need to be able to “re-sample” at every time step
 - Memoryless property → forgets prior samples
 - Will overcome next week with discrete event simulation
- Statistics of interest need derived state variables:
 - Average waiting time: $\bar{W} = \frac{\sum w(i)}{\sum c(i)} = \frac{\text{total waiting time}}{\text{number customers}}$
 - Utilization ratio: $\rho = \frac{\sum b(i)}{\sum \Delta t(i)} = \frac{\text{total busy time}}{\text{total duration}}$