LECTURE 16

Markov Processes – I

- Readings: Sections 7.1–7.2

Lecture outline

- Checkout counter example
- Markov process definition
- $n$-step transition probabilities
- Classification of states

Checkout counter model

- Discrete time $n = 0, 1, \ldots$
- Customer arrivals: Bernoulli($p$)
  - geometric interarrival times
- Customer service times: geometric($q$)
- “State” $X_n$: number of customers at time $n$

Finite state Markov chains

- $X_n$: state after $n$ transitions
  - belongs to a finite set, e.g., $\{1, \ldots, m\}$
  - $X_0$ is either given or random
- Markov property/assumption:
  (given current state, the past does not matter)
  \[
  p_{ij} = P(X_{n+1} = j \mid X_n = i) = P(X_{n+1} = j \mid X_n = i, X_{n-1}, \ldots, X_0)
  \]
- Model specification:
  - identify the possible states
  - identify the possible transitions
  - identify the transition probabilities

$n$-step transition probabilities

- State occupancy probabilities, given initial state $i$:
  \[
  r_{ij}(n) = P(X_n = j \mid X_0 = i)
  \]
- Key recursion:
  \[
  r_{ij}(n) = \sum_{k=1}^{m} r_{ik}(n-1) p_{kj}
  \]
- With random initial state:
  \[
  P(X_n = j) = \sum_{i=1}^{m} P(X_0 = i) r_{ij}(n)
  \]
Example

```
0.5 0.8
0.5 0.2
```

<table>
<thead>
<tr>
<th>$r_{11}(n)$</th>
<th>$n = 0$</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 100$</th>
<th>$n = 101$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{12}(n)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{21}(n)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{22}(n)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Generic convergence questions:
- Does $r_{ij}(n)$ converge to something?

```
0.5 0.5
1 2
```

- Does the limit depend on initial state?

```
0.4
0.3
```

Recurrent and transient states

- State $i$ is **recurrent** if:
  - starting from $i$,
  - and from wherever you can go,
  - there is a way of returning to $i$
- If not recurrent, called **transient**

- $i$ transient:
  - $P(X_n = i) \rightarrow 0$,
  - $i$ visited finite number of times

- **Recurrent class**:
  - collection of recurrent states that
    "communicate" with each other and with no other state