Markov Chains IV
Dmitri Loguinov
Texas A&M University
March 7, 2017
**Agenda**

- Continuous chains
  - Definitions
  - M/M/1 queue
  - Loss probability, queue size distribution
- More examples
  - Google PageRank
  - Resilience of P2P networks
- Wrap-up
Continuous Chains

• Second type of Markov chains that we study
  – Recall that continuous-time chains (CTMC) make jumps after spending random amounts of time in each state
  – Each such duration is called sojourn time $\tau_i$

• Note that sojourn times are independent; however, their distribution may vary with state $j$
  – Thus, the chain may stay longer in some states than in others

• One classical example of CTMC is the Poisson arrival process $M(t)$ where $\tau_i$ are iid exponential variables
Continuous Chains 2

• Here is again an illustration of the differences between discrete and continuous chains

• One important class of CTMC are router queues
  - There are many types of queue models
  - They typically have 3 parameters as in $M/M/1$, which stand for the arrival process, departure process, and the number of routers serving the packets
Continuous Chains 3

- $M$ stands for Markov (exponential delay), $D$ for deterministic (constant inter-renewal delay), $G$ for general (an arbitrary distribution)
  - $M/G/1$, $G/M/1$, $G/G/1$ are all examples with a single router
  - $M/M/c$ or $M/M/\infty$ allow more than one server to work on the packets in parallel

- Consider an $M/M/1$ queue
  - Packets arrive and depart as two Poisson processes with rates $\lambda$ and $\mu$, respectively
  - State of the chain $N(t)$ at time $t$ is the number of packets in the queue
Continuous Chains 4

- Queue size over time

- Once the process is in state $j$, the delay to the next packet (i.e., transition to state $j+1$) is an exponential random variable $X$ with CDF $1 - e^{-\lambda x}$
  - We say that the transition rate $j \rightarrow j+1$ is $\lambda$
Similarly, since packet service times are also random, the delay before finishing transmission of the current packet (i.e., going from state $j$ to $j-1$) is an exponential variable $Y$ with distribution $1 - e^{-\mu x}$

- We call this transition rate $\mu$

Why are service times random?
- Could be many reasons, but one of them is packet size is assumed to be exponentially distributed
- Not always a realistic assumption for the Internet

In ATM networks, for example, packet size is fixed and one may use $M/D/1$ queues instead
• What is the distribution of sojourn times $\tau_i$?
• Notice that $\tau_i$ is the minimum between the delay to receive a new packet and serve the next packet
  - Which means $\tau_i = \min(X,Y) \sim \exp(\lambda+\mu)$
• Also straightforward to get transition probabilities
  - Except boundary cases when $i = 0$
Thus, the one-step probability matrix of $M/M/1$ queues has the following shape

$$P = \begin{pmatrix} 0 & 1 & 0 & \ldots \\ q & 0 & p & \ldots \\ 0 & q & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

For all states to be positive, we must have $p < q$ (or equivalently, $\lambda < \mu$)
- This makes the arrival rate smaller than the service rate
- In such cases, the queue is said to be “stable”
Continuous Chains 8

- While generally continuous-time chains are more complex than discrete chains, $M/M/1$ queues allow a simple derivation of $\pi$

- Recall that transition rates across any boundary of a stable queue must be equal
  - Thus, we can apply similar principles to continuous chains:

![Diagram](https://via.placeholder.com/150)

- Rate of transitions from state $i$ to $i+1$ equals $\pi_i$ (i.e., fraction of time in state $i$) times the departure rate $\lambda$ along the link
Therefore, we can write our balance equations:

\[ \lambda \pi_i = \mu \pi_{i+1} \]

Recursively expanding:

\[ \pi_i = \left( \frac{\lambda}{\mu} \right)^i \pi_0 = \rho^i \pi_0, \quad \rho = \frac{\lambda}{\mu} \]

Next we need to determine \( \pi_0 \)

\[ 1 = \sum_{i=0}^{\infty} \pi_i = \pi_0 \sum_{i=0}^{\infty} \rho^i = \frac{\pi_0}{1 - \rho} \]
Continuous Chains 10

• Thus, the fraction of time queue size equals \( i \) packets is simply:

\[
\pi_i = (1 - \rho) \rho^i
\]

• Metric \( \rho < 1 \) is called *traffic intensity* and solely determines the distribution of queue size
  - For example, given \( \rho = 0.8 \), the probability to find the queue empty is 20% and with 1 packet 16%
  - Also notice that we can view \( \rho \) as the *link-utilization factor* since \( P(\text{queue is busy}) = 1 - \pi_0 = \rho \)

• For \( \rho = 0.8 \), we can conclude that 8.6% of the time queue size exceeds 10 packets
Continuous Chains 11

- For 95% link utilization, the buffer exceeds 20 packets at least 34% of the time.

- Next assume finite buffers of capacity $L$ packets.
  - Thus, all transitions from state $L$ occur into state $L - 1$.

- The only difference is in normalizing $\pi_0$.

\[
1 = \sum_{i=0}^{L} \pi_i = \pi_0 \sum_{i=0}^{L} \rho^i = \frac{\pi_0(1 - \rho^{L+1})}{1 - \rho}
\]

\[
\pi_i = \frac{(1 - \rho)\rho^i}{1 - \rho^{L+1}}
\]
Continuous Chains 12

• How many arrivals are now lost?
  – Applying PASTA, the number of customers who observe the chain in state $L$ equals the probability to find the chain in that state!
  – Since all such customers (packets) are lost, we have that the packet-loss rate is equal to $\pi_L$:

$$P(\text{packet lost}) = \pi_L = \frac{(1 - \rho)\rho^L}{1 - \rho^{L+1}}$$

• For example, $\rho = 0.8$ and $L = 10$, we have 2.35% of packets are dropped
  – If $\rho = 0.95$, we will lose 6.9% of arriving packets
Continuous Chains 13

• From all this data, we may want to design a buffer such that the probability of an overflow is below a certain threshold
  - For 1% packet loss and 80% link utilization, buffer size must be at least 14 packets
  - For the same loss and 99% link utilization, we need at least 69 packets

• Practical limitations of $M/M/1$
  - Delays between arrival are usually not 1) iid or 2) exponential
  - Service times are not exponential either

• Bursty traffic requires much larger queues
  - Often ISPs run at 20% or less link utilization on backbone
Example: Google PageRank

- Search engines need a method to rank the importance of pages on the web

- PageRank is one such technique proposed in 1998 under the following intuition:
  - Assumption 1: links pointing to a page indicate endorsement of its content
  - Assumption 2: users randomly surf the web, clicking each outgoing link from a page with equal probability
  - Assumption 3: users may randomly decide to teleport to another page using bookmarks or search-engine results

- **Goal**: design an algorithm that takes the webgraph $G$ and outputs the ranks of all pages
Google PageRank 2

• Denote by $u^+$ the set of page $u$’s out-degree neighbors and by $u^-$ the set of its in-degree neighbors

• A “random surfer” keeps clicking on out-going links
  - The probability that the surfer jumps from page $u$ to $v$ is:
    $$p_{uv} = \frac{1}{\deg(u^+)}$$, if $u \rightarrow v$

    ![u’s out-degree]

  - The random walk defines a discrete-time MC with transition probability matrix $P = (p_{uv})$, where each state $u$ is a webpage
  
  - Is this good enough? What if the surfer visits a page with zero out-degree, or enters a loop, or gets bored
Google PageRank 3

• Define a damping factor $\alpha$ to be the probability that the surfer will continue their random walk
  - With probability $1 - \alpha$, the surfer jumps to some other page

• The transition probability from page $u$ to $v$ is then:

$$p_{uv} = \begin{cases} 
\alpha \frac{1}{|u^+|} + (1 - \alpha) \frac{1}{N} & u \rightarrow v \\
(1 - \alpha) \frac{1}{N} & \text{otherwise}
\end{cases}$$

• There exists a stationary distribution $\pi$ for this MC, which satisfies $\pi = \pi P$

• It is then easy to see that

$$\pi_u = \alpha \sum_{v \in u} \frac{\pi_v}{|v^+|} + \frac{1 - \alpha}{N}$$

PageRank of page $u$'s in-neighbors
Google PageRank 4

• PageRank $\pi$ can be computed via matrix operations
  - However, the matrix size $N \times N$ is huge

• In practice, $\pi$ is obtained by iterating the computation until it converges
  - Assign any random initial vector to $\pi^0$
  - While $|\pi^{i+1} - \pi^i| > \epsilon : \pi^{i+1} = \pi^i P$
  - After $K$ iterations, obtain the final PageRank $\pi^K$

• Dealing with zero out-degree (dangling) nodes
  - Eliminate from the graph, possibly in several iterations
  - Add edges back to in-degree neighbors
  - Teleport with probability 1 (common case in practice)
Example: Resilience of P2P Networks

- Recall from HW3 that each joining user $v$ selects $k$ neighbors from among the existing users
  - Denote by $R_i$ the residual lifetime of neighbor $i$
  - If neighbor replacement is not allowed, time to isolation $T = \max(R_1, R_2, \ldots, R_k)$
  - Then, the probability that $v$ is isolated is $P(T < L)$, where $L$ is the lifetime of $v$

- Consider that once a neighbor fails, a replacement is found after some random search delay $S$
  - The $i$-th neighbor ON/OFF process:
• Denote by $W(t)$ the number of neighbors at time $t$
  − Clearly, this is a continuous-time stochastic process

• Assume that user lifetimes are exponential with rate $\mu$
  and search delays are exponential with rate $\lambda$
  − Neighbor residual lifetimes $R \sim \exp(\mu)$
  − Search delays $S \sim \exp(\lambda)$
  − Then, process $W(t)$ is a Markov chain

• By manipulating this CTMC, we examine the delay $T_{00}$
  between visits to state 0
  − Variable $T_{00}$ is a good approximation for isolation time $T$
Resilience of P2P Networks 2

- Given that $W(t) = i$, there are $i$ live neighbors and $k - i$ dead in search of a replacement.

- Transition from $i$ to $i + 1$ is triggered by finding a replacement.
  - Transition delay from $i$ to $i + 1$ is $\min(S_1, \ldots, S_{k-i})$, which is exponential with rate $(k - i)\lambda$, for $i < k$.

- Transition from $i$ to $i - 1$ is caused by neighbor failure.
  - Transition delay from $i$ to $i - 1$ is $\min(R_1, \ldots, R_i)$, which is exponential with rate $i\mu$, for $i > 0$.
Resilience of P2P Networks

• Compute the stationary distribution $\pi$ of process $W(t)$

• Recall that the transition rates across any boundary must be equal:

\[ \pi_i q_{i,i-1} = \pi_{i-1} q_{i-1,i} \]

The above are called balance equations. It follows that

\[ \pi_i = \pi_0 \rho^i \frac{k!}{i!(k-i)!}, \quad \rho = \frac{\lambda}{\mu} \]

• After normalization, we have:

\[ \pi_i = \binom{k}{i} \frac{\rho^i}{(1 + \rho)^k}, \quad i = 0, 1, ..., k \]
Resilience of P2P Networks

• For continuous-time Markov chains, the expectation of delay $T_{jj}$ between visits to state $j$ is given by:

$$E[T_{jj}] = \frac{1}{\pi_j q_j}$$

The transition rate out of state $j$ is $q_0 = k\lambda$

• The mean delay $T_{00}$ between visits to state 0 is $1/\pi_0 q_0$

  - Node isolation probability $\phi = P(L < T)$ is close to $E[L]/E[T_{00}]$, where $E[L]$ is the mean user lifetime
  - Compare isolation probability $\phi$ for exponential lifetimes with $E[L] = 0.5$ hours and $k = 8$

<table>
<thead>
<tr>
<th>$E[S]$</th>
<th>$E[L]/E[T_{00}]$</th>
<th>$\phi$ with replacement</th>
<th>$\phi$ without replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>$1.5 \times 10^{-5}$</td>
<td>$2.4 \times 10^{-5}$</td>
<td>$1/(k+1) = 0.111$</td>
</tr>
<tr>
<td>0.01</td>
<td>$8.3 \times 10^{-12}$</td>
<td>$8.7 \times 10^{-12}$</td>
<td>$1/(k+1) = 0.111$</td>
</tr>
</tbody>
</table>

Significant improvement!
Wrap-up

• We skipped a lot of technical definitions and associated derivations in chapters 3-4

• Even the simplest stochastic queues require non-trivial modeling efforts and half of a semester of derivations

• Midterm includes today’s lecture
  – Practice computing $\pi$ for continuous chains