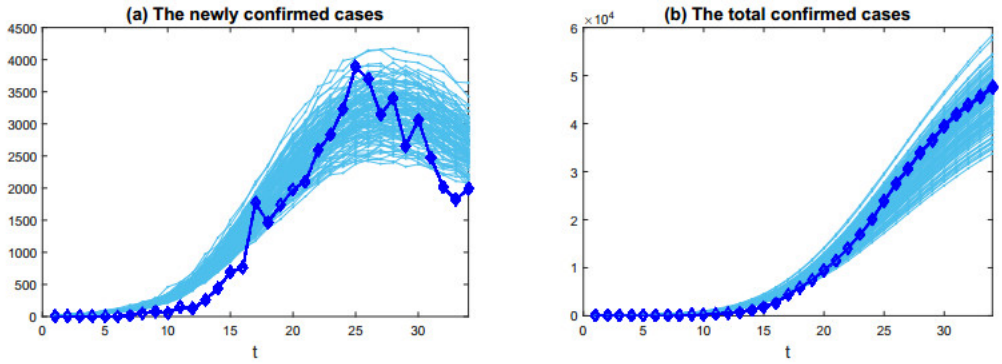


LINMA2470 Stochastic Modelling

Simon Desmidt

Academic year 2024-2025 - Q2

UCLouvain

Contents

Reminders

1.1 General properties of probability

- $P[A \cup B] = P[A] + P[B] - P[1 \cap B]$;
- $P[A|B] = \frac{P[A \cap B]}{P[B]} = \frac{P[AB]}{P[B]}$;
- A and B are independent iff $P[AB] = P[A]P[B] \implies P[A|B] = P[A]$;
- $P[X \leq x] = F_X(x)$ is the distribution function, i.e. a monotone increasing function of x going from 0 to 1 when x goes from $-\infty$ to $+\infty$.
- Its derivative is the density function $f_X(x)$ such that $f_X(x)\delta \approx P[x \leq X \leq x + \delta]$ for an infinitesimal δ.
- A random variable X is said to be memoryless if $\forall t, x > 0$, $P[X > t + x | X > t] = P[X > x]$.
- Markov inequality (for a nonnegative random variable): $P[Y \geq y] \leq \frac{\mathbb{E}[Y]}{y}$;
- Chebyshev inequality: $P[|Z - \mathbb{E}[Z]| \geq \varepsilon] \leq \frac{\sigma_Z^2}{\varepsilon^2}$;

1.2 Expectation and variance

- For a discrete random variable, $\mathbb{E}[X] = \sum_{n=-\infty}^{\infty} nP[X = n]$;
- For a continuous random variable, $\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx$;
- $\mathbb{E}[X] = \int_0^{\infty} (1 - F_X(x)) dx$.
- $Var[X] = \sigma_X^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$;

1.3 Law of large numbers

Let $X_1, \ldots, X_n$ be a series of independent and uniformly distributed (IID) random variables with expectation $\bar{X}$ and finite variance σ_X^2. Let $S_n = X_1 + \cdots + X_n$. Then,

- Weak version:

$$\lim_{n\to\infty} P\left[|\frac{S_n}{n} - \bar{X}| \geq \varepsilon\right] = 0 \tag{1.1}$$

- Strong version:

$$\lim_{n\to\infty} P\left[\sup_{m\geq n}\left(\frac{S_m}{m} - \bar{X}\right) > \varepsilon\right] = 0 \Longleftrightarrow \lim_{n\to\infty} \frac{S_n}{n} = X \qquad \text{with probability 1} \tag{1.2}$$

1.4 Central limit theorem

$$\lim_{n\to\infty} P\left[\frac{S_n - n\bar{X}}{\sqrt{n}\sigma} \leq y\right] = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \tag{1.3}$$

1.5 Exponential distribution

- $f_X(x) = \lambda e^{-\lambda x}$, for $x \geq 0$;
- $F_X(x) = 1 - e^{-\lambda x}$, for $x \geq 0$;
- $\mathbb{E}[X] = 1/\lambda$.

$\rightarrow$ Note: the exponential distribution is memoryless.

Poisson Processes

A Poisson process $N(t)$ counts the number of arrivals with exponentially distributed inter-arrival times.

$$S_n = \sum_{i=1}^{n} X_i \qquad X_i \sim \exp(\lambda) \tag{2.1}$$

$\forall n, t$, we have the relation $\{S_n \leq t\} = \{N(t) \geq n\}$, where S_n is a random variable telling at which time the n-th occurence appears.

$\rightarrow$ Note: a Poisson process is memoryless: $P[Z_1 > x] = e^{-\lambda x}$, with Z_1 be the duration of the time interval from t until the first arrival after t.

For a Poisson process of rate λ, and any given $t > 0$, the length of the interval from t until the first arrival after t is an exponentially distributed random variable. This random variable is idenpendent of both $N(t)$ and of the $N(t)$ arrival epochs before time t. It is also independent of $N(\tau)$, $\forall \tau \leq t$.

Let us consider the process after Z_1, Z_m, the time until the m-th arrival after time t. It is independent of $N(t)$ and of the entier previous history of the process.
Let us denote $\tilde{N}(t, t') = N(t') - N(t)$.

- Stationary increments property: It has the same distribution as $N(t' - t)$, $\forall t' \geq t$ (stationary increments property);
- Independent increments property: For any sequence of times $0 < t_1 < \cdots < t_k$, the set $\{N(t_1), \tilde{N}(t_1, t_2), \ldots, \tilde{N}(t_{k-1}, t_k)\}$ is a set of independent random variables.

From the memoryless property, here is another definition of a Poisson process:

- A Poisson process is a counting process that has the stationay and independent increment properties and such that

$$\begin{aligned} P[\tilde{N}(t, t+\delta) = 0] &= 1 - \lambda\delta + o(\delta) \\ P[\tilde{N}(t, t+\delta) = 1] &= \lambda\delta + o(\delta) \\ P[\tilde{N}(t, t+\delta) \geq 2] &= o(\delta) \end{aligned} \tag{2.2}$$

2.1 Distribution of $N(t)$

S_n is the sum n IID random variables and f_{S_n} is the convolution of n times f_X:

$$f_{S_n}(t) = \frac{\lambda^n t^n e^{-\lambda t}}{(n-1)!} \tag{2.3}$$

From this,

$$P[N(t) = n-1] = \frac{(\lambda t)^n e^{-\lambda t}}{(n)!} \tag{2.4}$$

and finally,

$$\mathbb{E}[N(t)] = \lambda t \qquad Var[N(t)] = \lambda t \tag{2.5}$$

From equation (2.4), the Poisson process verifies the following probability conditions:

- $P[\tilde{N}(t, t+\delta) = 0] = 1 - \lambda\delta + o(\delta)$;
- $P[\tilde{N}(t, t+\delta) = 1] = \lambda\delta + o(\delta)$;
- $P[\tilde{N}(t, t+\delta) \geq 2] = o(\delta)$;

where we use a first-order approximation of the exponential term, with $o(\delta)$ its residual. As $o(\delta)$ is negligible, we can approximate the Poisson process as a Bernoulli process.

2.1.1 Combining Poisson processes

Let $N_1(t)$ and $N_2(t)$ be tow independent Poisson processes. Let the process $N(t) = N_1(t) + N_2(t)$. We can show using the three properties above that $N(t)$ is a Poisson process with rate $\lambda_1 + \lambda_2$.

2.1.2 Subdividing a Poisson process

Let $N(t)$ be a Poisson process with rate λ. We split the arrivals in 2 subprocesses $N_1(t)$ and $N_2(t)$. Each arrival of $N(t)$ is sent to $N_1(t)$ with probability p and to $N_2(t)$ with probability $(1-p)$, each split being independent from all others.
Then, the resulting processes $N_1(t)$ and $N_2(t)$ are two independent Poisson processes with respective rate $p\lambda$ and $(1-p)\lambda$.

2.1.3 Conditional arrival distribution

The density probability function when we have n Poisson processes, under the condition that $N(t) = n$, is

$$f(s_1, \ldots, s_n | N(t) = n) = \frac{n!}{t^n} \tag{2.6}$$

From the previous results, we can compute that

$$P[S_1 > \tau | N(t) = n] = \left(\frac{t-\tau}{t}\right)^n \tag{2.7}$$

and the expectation is

$$E[S_1|N(t) = n] = \frac{t}{n+1} \tag{2.8}$$

And from this, we derive that

$$P[X_i > \tau|N(t) = n] = \left(\frac{t-\tau}{t}\right)^n \tag{2.9}$$

with expectation

$$E[X_i] = \frac{t}{n+1} \tag{2.10}$$

And thus the density function is

$$f_{S_i}(x|N(t) = n) = \frac{x^{i-1}(t-x)^{n-i}n!}{t^n(n-i)!(i-1)!} \tag{2.11}$$

2.2 Non-homogenous Poisson processes

A non-homogenous Poisson rocess $N(t)$ is a counting process with increments that are independent but not stationary, with

- $P[\tilde{N}(t, t+\delta) = 0] = 1 - \lambda(t)\delta + o(\delta)$;
- $P[\tilde{N}(t, t+\delta) = 1] = \lambda(t)\delta + o(\delta)$;
- $P[\tilde{N}(t, t+\delta) \geq 2] = o(\delta)$;

where $\tilde{N}(t, t+\delta) = N(t+\delta) - N(t)$. The time-varying arrival rate $\lambda(t)$ is assumed to be continuous and stricly positive.

2.3 Bernoulli process approximation

We can approximate the non-homogenous Poisson process with a Bernoulli process where the time is partitioned into increments of lengths inversely proportional to $\lambda(t)$ (i.e. using a nonlinear time scale).

- $P[\tilde{N}(t, t+\epsilon/\lambda(t)) = 0] = 1 - \epsilon + o(\epsilon)$;
- $P[\tilde{N}(t, t+\epsilon/\lambda(t)) = 1] = \epsilon + o(\epsilon)$;
- $P[\tilde{N}(t, t+\epsilon/\lambda(t)) \geq 2] = o(\epsilon)$;

Letting ϵ tend to zero, we obtain

$$P[N(t) = n] = \frac{(m(t))^n e^{-m(t)}}{n!} \qquad P[\tilde{N}(t, t') = n] = \frac{(m(t, t'))^n e^{-m(t,t')}}{n!} \tag{2.12}$$

with

$$m(t) = \int_0^t \lambda(\tau)d\tau \qquad m(t, t') = \int_t^{t'} \lambda(\tau)d\tau \tag{2.13}$$

2.4 Classification of queueing systems

- We note $A/B/k$ where A is the type of distribution for the arrival process, B for the service time and k the number of servers.

We suppose that the arrivals wait in a single queue. Commonly used letters are

- M: exponential distribution (for A) or Poisson process (for B);
- D: deterministic time intervals;
- E: Erlang distribution;
- G: general distribution.

Renewal Processes

A renewal process is a counting process with IID interarrival intervals. We note X_i the interval between arrivals, $\bar{X} = \mathbb{E}[X]$ is supposed to be finite with probability $P[X_i] > 0 = 1$[1], σ can be finite, and we denote $S_n = \sum_{i=1} X_i$ the time of the n-th arrival.

3.1 Strong law of large numbers

Let $\{N(t); t \geq 0\}$ be a renewal process, then

$$\lim_{t\to\infty} N(t) = \infty \qquad \lim_{t\to\infty} \mathbb{E}[N(t)] = \infty \tag{3.1}$$

This implies that

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{\bar{X}} \text{ with probability 1} \tag{3.2}$$

3.2 Central limit theorem

If the interarrival intervals of the renewal process $N(t)$ have a finite standard deviation, then from the CLT for IID random variables, we have

$$\lim_{t\to\infty} P\left[\frac{S_n - n\bar{X}}{\sqrt{n}\sigma} \leq \alpha\right] = \Phi(\alpha) \tag{3.3}$$

What is $\Phi(\alpha)$?
and

$$\lim_{t\to\infty} P\left[\frac{N(t) - t/\bar{X}}{\sigma\bar{X}^{-3/2}\sqrt{t}} < \alpha\right] = \Phi(\alpha) \tag{3.4}$$

→ Note: The reliability of the observed mean of successive results that are supposed to be IID depends a lot on the rule used to decide when we stop repeating the experiment.

[1] A probability of 1 means that the opposite can happen, but is so rare that the probability is 0.

3.3 Stopping time

Let N be the rv corresponding to the total number of experiments observed. Let I_n be a series of rv being the indicator fucnction of $\{N \geq n\}$:

$$I_n = \begin{cases} 1 & \text{if the } n\text{-th experiment is observed} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

N is a stopping time if I_n depends only on $X_1, \ldots, X_{n-1}$. This means that stopping at 3pm, for example, is not a stopping time, because it can depend on X_n, depending if the n-th arrival is before or after 3pm.

3.3.1 Wald's inequality

Let N be a stopping time for $\{X_n; n \geq 1\}$. Then, $\mathbb{E}[S_N] = \mathbb{E}[N]\bar{X}$.

3.4 Blackwell's renewal theorem

3.4.1 Arithmetic distribution

If interarrival intervals can only have a length that is a multiple of some real number d, the interarrival distribution will be called an arithmetic distribution, and d the span of the distribution.

3.4.2 Blackwell's inequality

If the interarrival distribution of a renewal process $N(t)$ is not arithmetic, then

$$\lim_{t\to\infty} (m(t+\delta) - m(t)) = \frac{\delta}{\bar{X}} \qquad \forall \delta \tag{3.6}$$

If the interarrival distribution is arithmetic with span d, then

$$\lim_{t\to\infty} (m(t+nd) - m(t)) = \frac{nd}{\bar{X}} \qquad \forall n \geq 1 \tag{3.7}$$

3.4.3 Relationship with a Poisson process

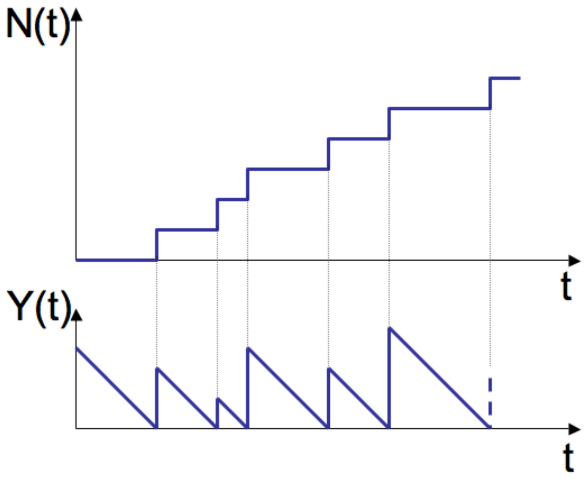
The sum of many renewal processes tends to a Poisson process: for a non-arithmetic renewal process with $P[X_i = 0] = 0$, we have

$$\begin{aligned} &\lim_{t\to\infty} P[N(t+\delta) - N(t) = 0] = 1 - \delta/\bar{X} + o(\delta) \\ &\lim_{t\to\infty} P[N(t+\delta) - N(t) = 1] = \delta/\bar{X} + o(\delta) \\ &\lim_{t\to\infty} P[N(t+\delta) - N(t) \geq 2] = o(\delta) \end{aligned} \tag{3.8}$$

The increments are asymptotically stationary, but not independent. |sectionRenewal reward process Along to the renewal process $N(t)$, we can add a reward function $R(t)$.

It models the rate at which the process is accumulating a reward or cost. It can however only depend on the current renewal but not the previous ones.
Let $Y(y)$ be the residual life at time t for the current renewal:

$$R(t) = Y(t) = S_{N(t)+1} - t \tag{3.9}$$

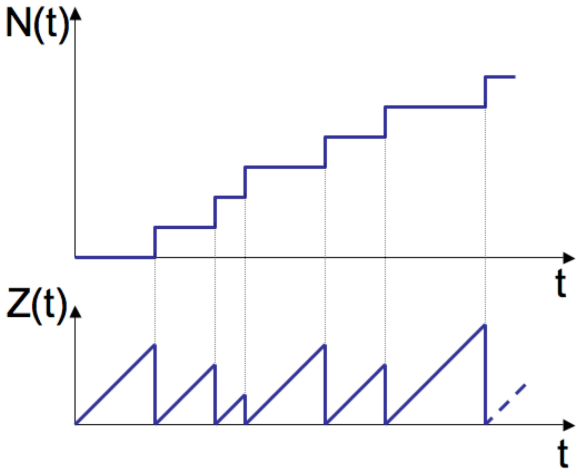


The time average residual life is $\frac{1}{t}\int_0^t Y(\tau)d\tau$.
From the definition of $Y(t)$, we can calculate that

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t Y(\tau)d\tau = \frac{\mathbb{E}[X^2]}{2\mathbb{E}[X]} = \frac{1}{2}\mathbb{E}[X] + \frac{Var(X)}{\mathbb{E}[X]} > \frac{1}{2}\mathbb{E}[X] \text{ with probability 1} \tag{3.10}$$

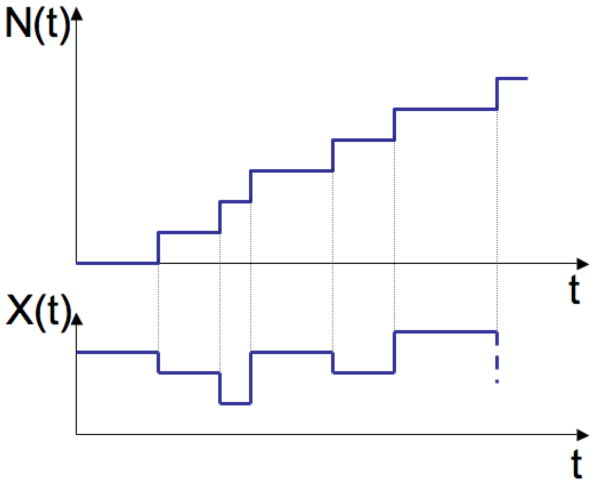
3.4.4 Time average age

Let $Z(t)$ be the age of the current renewal at time t: $R(t) = Z(t) = t - S_{N(t)}$. The time average age is $\lim_{t\to\infty}\frac{1}{t}\int_0^t Z(\tau)d\tau = \frac{\mathbb{E}[X^2]}{2\mathbb{E}[X]}$.



3.4.5 Time average duration

Let $X(t)$ be the duration of the renewal containing time t: $R(t) = X(t) = S_{N(t)+1} - S_{N(t)}$. The time average duration is $\lim_{t\to\infty}\frac{1}{t}\int_0^t X(\tau)d\tau = \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]}$.



lsubsectionGeneral renewal reward functions Let $R(t)$ be a reward function for a renewal process with expected inter-renewal times $\bar{X} < \infty$, then with probability 1,

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t R(\tau)d\tau = \frac{\mathbb{E}[R_n]}{\mathbb{E}[X]} \tag{3.11}$$

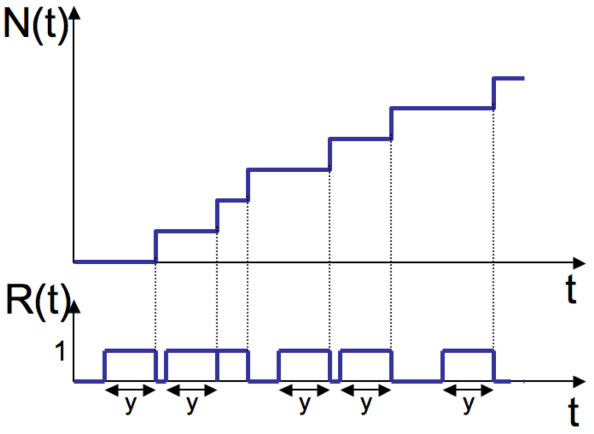
where R_n is defined as

$$R_n = \int_{S_n}^{S_{n+1}} R(\tau)d\tau \tag{3.12}$$

3.4.6 Distribution of residual life

We are interested in the fraction of time that $Y(t) \leq y$:

$$R(t) = I\{Y(t) \leq y\} \qquad R_n = \min\{y, X_n\} \tag{3.13}$$



And we can calculate that

$$\begin{aligned} \mathbb{E}[R_n] &= \int_0^y P[X > x]dx \\ F_Y(y) &= \frac{1}{\mathbb{E}[X]} \int_0^y P[X > x]dx \end{aligned} \tag{3.14}$$

3.4.7 Key theorem

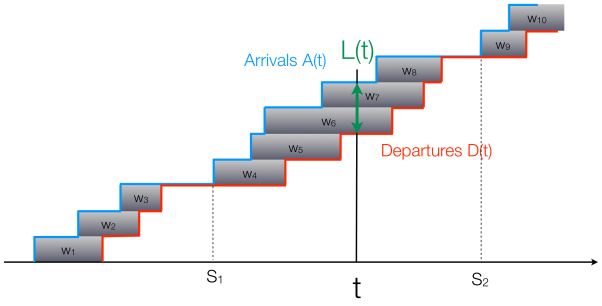
Let $N(t)$ be a non-arithmetic renewal process, let $R(z,x) \geq 0$ be such that $r(z) = \int_{x=z}^{\infty} R(z,x) dF_X(x)$ is directly Rieman integrable. Then,

$$\lim_{t\to\infty} \mathbb{E}[R(t)] = \frac{\mathbb{E}[R_n]}{\bar{X}} \tag{3.15}$$

3.5 Little's Law

Let a queueing system be such that

- $A(t)$ is the number of arrivals between 0 and t;
- $D(t)$ is the number of departures between 0 and t;
- $L(t) = A(t) - D(t)$ is the number of customers in the system at time t;
- w_i the time the i^{th} customer spends in the system;
- $N(t)$ is the renewal process counting the number of busy periods of the system (each time a customer arrives when the system is empty).



A general queueing system

Let us use $L(t)$ as a reward function for the renewal process $N(t)$. This implies

$$\begin{gathered} \sum_{n=1}^{N(t)} R_n \leq \int_0^t L(\tau) d\tau \leq \sum_{i=1}^{A(t)} w_i \leq \sum_{n=1}^{N(t)+1} R_n \\ \lim_{t\to\infty} \frac{1}{t} \int_0^t L(\tau) d\tau = \frac{\mathbb{E}[R_n]}{\mathbb{E}[X]} \end{gathered} \tag{3.16}$$

Putting all this together, we can show that $\bar{L} = \lim_{t\to\infty} \frac{1}{t} \int_0^t L(\tau) d\tau = \bar{W}\lambda$.

3.5.1 M/G/1 queue

Let $R(t)$ be the remaining time for the customer being served. Let $U(t)$ be the time an arrival at time t would have to wait before being served. Let $L_q(t)$ be the number of

customers in queue at time t, independent of the Z_i. We define

$$U(t) = \sum_{i=1}^{L_q(t)} Z_i + R(t) \Longrightarrow \mathbb{E}[U(t)] = \mathbb{E}[L_q(t)]\mathbb{E}[Z] + \mathbb{E}[R(t)] \tag{3.17}$$

We can show that

$$\int_0^{S_N(t)} R(\tau)d\tau \leq \int_0^{S_N(t)+1} R(\tau)d\tau \tag{3.18}$$

And from Little's Law,

$$\lim_{t\to\infty} \mathbb{E}[L_q(t)] = \lambda \bar{W}_q \Longrightarrow \lim_{t\to\infty} \mathbb{E}[U(t)] = \lambda \bar{W}_q \mathbb{E}[Z] + \lambda \frac{\mathbb{E}[Z^2]}{2} \tag{3.19}$$

Poisson arrival process implies that arrivals occur with identical probability at any moment, this implies independence with $U(t)$. Hence $\mathbb{E}[W_q(t)] = \mathbb{E}[U(t)]$. Hence $\bar{W}_q = \lambda \bar{W}_q \mathbb{E}[Z] + \lambda \frac{\mathbb{E}[Z^2]}{2}$. And we can isolate $\bar{W}_q$:

$$\bar{W}_q = \frac{\lambda(\mathbb{E}[Z]^2 + \sigma^2)}{2(1 - \lambda\mathbb{E}[Z])} \tag{3.20}$$

And we remember $\bar{W} = \bar{W}_q + \mathbb{E}[Z]$.

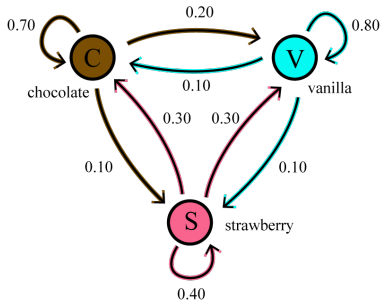
Finite State Markov Chains

4.1 Definitions

A Markov chain is a stochastic process with fixed intervals $\{X_n, n \geq 0\}$ such that each random variable $X_n, n \geq 1$ depends on the past only through the most recent random variable X_{n-1}:

$$P[X_n = j | X_{n-1} = i, X_{n-2} = k, \ldots, X_0 = m] = P[X_n = j | X_{n-1} = i] = P_{ij} \tag{4.1}$$

The rv X_n is called the state of the Markov chain, and the set of possible sample values for the states lie in a countable set. A Markov chain can be represented under a graph or matrix form:



$$P = \begin{pmatrix} 0.70 & 0.20 & 0.10 \\ 0.10 & 0.80 & 0.10 \\ 0.30 & 0.30 & 0.40 \end{pmatrix} \tag{4.2}$$

- We say that a state j is accessible from i ($i \to j$) if there exists a xalk in the graph from i to j: $i \to j$ iff $P_{ij}^n = P[X_n = j | X_0 = i] > 0$ for some n.
- Two distinct states i, j communicate ($i \leftrightarrow j$) if i is accessible from j and vice versa.
- A class C of states is a non-empty set of states such that for each $i \in C$, each state $j \neq i$ satisfies $j \in C$ if $i \leftrightarrow j$ and $j \notin C$ if $i \not\leftrightarrow j$.
- A state i is recurrent if it is accessible from all states that are accessible from i. A transient state is a state that is not recurrent.

$\to$ Note: all states of a same class are of the same type.

- A finite-state Markov chain has at least one recurrent class.
- The period of a state i, denoted $d(i)$, is the greatest common divisor of all n such that $P_{ii}^n > 0$. A state is aperiodic if $d(i) = 1$.

$\to$ Note: All states of a class have the same periodicity.

- If a class has a period $d > 1$, then there existsa partition $\{C_i\}_{i=1}^{d}$ of the states of the class such that all the transitions from a state of C_n go to a state of class C_{n+1} and all transitions from C_d go to a state of C_1, i.e. we make a cycle of subclasses.
- A class is called ergodic if it is aperiodic and recurrent.
- A matrix is stochastic iff it is square, non negative ad each row sums to 1, i.e. $P\mathbb{1}_n = \mathbb{1}_n$.

4.2 Transition probabilities

We can calculate that

$$P[X_{n+2} = j | X_n = i] = P_{ij}^2 = \sum_{k=1}^{J} P_{ik}P_{kj} \Longrightarrow P^2 = P \cdot P \Longrightarrow P^n = P \cdot ... \cdot P \tag{4.3}$$

More generally, $P_{ij}^{n+m} = \sum_{k=1}^{J} P_{ik}^n P_{kj}^m$.
Because of ergodicity, all rows converge to the same value, and we store those values in a row vector called π. Then, $\pi = \pi P$ and the sum of all values of π is 1.

From this, we induce that π is a left eigenvector of P for the eigenvalue 1, and the number of linearly independent solutions corresponds to the multiplicity of the eigenvalue 1. There will be one independent solution for each recurrent class of P. Moreover, if P is ergodic, then $\lim_{n\to\infty} P^n = \mathbb{1}_m \pi$, else π will be the average over the different subclasses.

4.3 Markov chains with rewards

Let r_i be the reward associated with state i. In the case where the reward is on the edge from node i to j and not the states, the reward is r_{ij} and we have $r_i = \sum_j P_{ij} r_{ij}$. For an ergodic chain, we observe that the average reward per period will be

$$g = \sum_i r_i \pi_i \tag{4.4}$$

where π_i is the steady state probability.

4.3.1 Expected reward over multiple transitions

Let X_m be the state at time m and let $R_m = R(X_m)$ be the reward at time m. Under the condition $X_m = i$ (starting point), the aggregate expected reward $v_i(n)$ over n periods from X_m to X_{m+n-1} is

$$v_i(n) = \mathbb{E}[R(X_m) + \cdots + R(X_{m+n-1}) | X_m = i] = r_i + \sum_j P_{ij} r_j + \cdots + \sum_j P_{ij}^{n-1} r_j \tag{4.5}$$

And in vector notation

$$v(n) = r + [P]r + \cdots + [P^{n-1}]r = \sum_{h=0}^{n-1} [P^h]r \tag{4.6}$$

4.3.2 Relative gain vector

Assuming the Markov chain is an ergodic unichain, i.e. it has a single ergodic class with possibly some transient classes, we know that $\lim_{n\to\infty}[P^n] = \mathbb{1}_m\pi$ and thus

$$\lim_{n\to\infty}[P^n]r = \mathbb{1}_m\pi r = g\mathbb{1}_m \tag{4.7}$$

This means that the expected reward per period converges to g. From this, we can evaluate the transient effect and define the relative-gain vector, denoted by w:

$$w = \lim_{n\to\infty}(v(n) - ng\mathbb{1}_m) = \lim_{n\to\infty}\sum_{h=0}^{n-1}[P^h - \mathbb{1}_m\pi]r \tag{4.8}$$

It can also be computed by solving the following equations instead of calculating the limit:

$$w + g\mathbb{1}_m = [P]w + r \qquad \pi w = 0 \tag{4.9}$$

The second equation means that the sum (weighted by the probabilities) of all gains is 0.

4.4 Markov decision processes

Suppose that in each state i, we can choose between K_i different possibilities with rewards $r_i^{(1)}, \ldots, r_i^{(K_i)}$. This means that we choose between different Markov chains that have the same states but not necessarily the same edges. We want to find the optimal (stationary or dynamic) policy for this problem.

4.4.1 Dynamic programming algorithm for the dynamic optimal policy

This section aims to find a dynamic programming algorithm for the dynamic optimal policy. We assume here that for any policy A, the resulting Markov chain with matrix $[P^A]$ is an ergodic unichain.

Let $v(0)$ be the final reward vector. Through a recurrence method, we can show that

$$v_i^*(n) = \max_k\{r_i^{(k)} + \sum_j P_{ij}^{(k)}v_j^*(n-1)\} \tag{4.10}$$

or in vector form:

$$v^*(n) = \max_A\{r^A + [P^A]v^*(n-1)\} \tag{4.11}$$

for a policy A, i.e. a decision k_i for each state i. From equations (4.9), we know that

$$w^A = r^A - g^A\mathbb{1}_m + [P^A]w^A \tag{4.12}$$

and thus

$$v^A(n) = ng^A\mathbb{1}_m + w^A + [P^A]^n(v(0) - w^A) \tag{4.13}$$

If $v(0) = w^B$ for a policy B such that

$$r^B + [P^B]w^B \geq r^A + [P^A]w^A \quad \forall A \tag{4.14}$$

then B is the dynamic optimal policy for each time period, and

$$v^*(n) = w^B + ng^B \mathbb{1}_m \tag{4.15}$$

This relation is an equivalence.

4.4.2 Policy improvement algorithm

Algorithm 1 Policy improvement algorithm

1: **Step 1:** Choose an arbitrary policy B;
2: **while** $\exists A \ : \ r^B + [P^B]w^B \overset{\leq}{\neq} r^A + [P^A]w^B$ **do**
3: **Step 2:** Compute w^B;
4: **Step 3:** Find A such that $r^A + [P^A]w^B \overset{\neq}{\geq} r^B + [P^B]w^B$;
5: **Step 4:** $B \leftarrow A$;
6: **end while**

Theorem 4.1. Assuming for any policy A that the Markov chain $[P^A]$ is an ergodic unichain, if B is an optimal stationary policy, then

$$\lim_{n\to\infty} v^*(n) - ng^B \mathbb{1}_m = w^B + (\beta - \pi^B w^B)\mathbb{1}_m \tag{4.16}$$

What is β?

4.5 Dynamic programming algorithm for the stationary optimal policy

Algorithm 2 Dynamic programming algorithm for the stationary optimal policy

1: **Step 1:** Fix an arbitrary vector $v(0)$;
2: **while** l < u **do**
3: **Step 2:** Compute

$$l = \min_i [v_i^*(n) - v_i^*(n-1)] \qquad u = \max_i [v_i^*(n) - v_i^*(n-1)] \tag{4.17}$$

4: **end while**
5: $l = u = g^A$ and A is the optimal stationary policy.

Markov Decision Processes and Reinforcement Learning

5.1 MDP and Policies

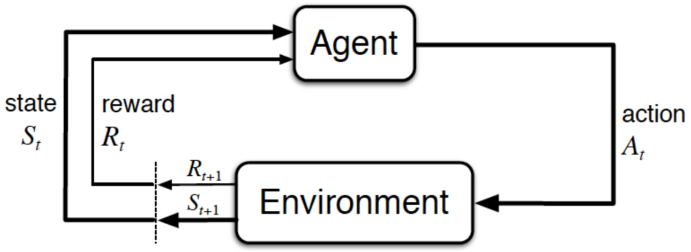
5.1.1 Markov Decision Process

A Markov Decision Process models a sequential decision-making process under uncertainty, where moving to the next stage only depends on the current action-state pair.

Definition 5.1. A Markov Decision Process (MDP) is defined by

- a set of system states S;
- a set of actions $\mathcal{A}$;
- a set of rewards R;
- probabilities $p(s', r|s, a)$ of getting a reward r and moving to state s' if action a is taken in state s.

The MDP is finite if the three sets are finite.



From the definition, we can derive 3 quantities:

- Transition probabilities: $p(s'|s, a) = \sum_r p(s', r|s, a)$;
- Reward probabilities: $p(r|s, a) = \sum_{s'} p(s', r|s, a)$;
- Expected reward knowing s, a: $r(s, a) = \mathbb{E}[R|s, a] = \sum_r r \sum_{s'} p(s', r|s, a)$.

5.1.2 Policies

A policy defines the decision making in each state;

Definition 5.2. A policy is a mapping $\pi : s \in S \to \pi(a|s)$, where $\pi(a|s)$ represents the probability of taking action a in state s.

5.1.3 Policy values and state-action values

The policy value $v_\pi(s)$ at a given state s corresponds to the expected rewards collected over time by applying policy π starting from state s.

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \tag{5.1}$$

where $\gamma \in [0, 1[$ is a discounted factor. The state-action value function is the same idea, but has a dependance on the action we intend to take.

$$q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right] \tag{5.2}$$

We can show that

$$v_\pi(s) = \mathbb{E}_{A \sim \pi(a|s)}[q_\pi(s, A)] \tag{5.3}$$

Those definitions induce the Bellman equations, a linear system in $v_\pi(s)$:

$$\forall s \in S, \quad v_\pi(s) = \sum_a \pi(a|s) \sum_{s'|r} p(s', r|s, a)[r + \gamma v_\pi(s')] \tag{5.4}$$

5.1.4 Policy evaluation

Given the policy π and the MDP probabilites p, we can rewrite the Bellman equations as

$$V = R + \gamma PV \iff (I - \gamma P)V = R \tag{5.5}$$

where $I - \gamma P$ is invertible as $\|P\|_\infty \leq 1$ and $\gamma < 1$.
Another way to solve this is by an iterative process: introducing the linear operator $L : V \to \gamma PV + R$, we want to find V such that $V \approx L(V)$. This appraoch is guaranteed to converge since L is γ-Lipschitz (affine application). Defining $V^* = \lim_{t \to \infty} L^t(V_0)$, with V_0 the first iterate, we can show that

$$\|V^* - V_{n+1}\|_\infty \leq \frac{\gamma}{1 - \gamma} \|V_{n+1} - V_n\|_\infty \tag{5.6}$$

Algorithm 3 Policy Evaluation Algorithm

Input: π the policy to be evaluated, and θ the guaranteed accuracy of estimation
Initialization: $V(s)$ the arbitrary initial value for all s
while $\Delta \geq \theta$ **do**
    $\Delta = 0$
    for each state s **do**
        $v = V(s)$
        $V(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
        $\Delta = \max(\Delta, |v - V(s)|)$
    end for
end while

5.2 Optimizing Policies

5.2.1 Policy Improvement Theorem

Theorem 5.3.

$$[\forall s \in S,\ q_\pi(s, \pi^{new}(s)) \geq q_\pi(s, \pi(s))] \Longrightarrow v_{\pi^{new}}(s) \geq v_\pi(s) \tag{5.7}$$

A strict inequality on the left implies a strict one on the right too. This theorem means that a new policy will be at least as good as a given policy π if changing any action of π by the corresponding action of π^{new} yields a better total gain at the end.

5.2.2 Bellman optimality conditions

Definition 5.4. A policy π^* is optimal if for any state $s \in S$ and any other policy π, $v_{\pi^*}(s) \geq v_\pi(s)$.

Theorem 5.5. A policy π is optimal iff for any state action pair (s, a) with a positive probability to be selected by the policy, i.e. $\pi(a|s) > 0$, we have

$$a \in \arg\max_{a' \in A} q_\pi(s, a') \tag{5.8}$$

Meaning that any action with a nonzero probability to be taken maximizes the gain of the state at which it is taken.
It can be shown that this implies that any finite MDP admits an optimal policy which is deterministic.

This theorem yields the two following equations for optimal policies:

$$\begin{aligned} v_*(s) &= \max_a q_{\pi^*}(s, a) = \max_{a \in A(s)} \sum_{s',r} p(s', r|s, a)(r + \gamma v_*(s')) \\ q^*(s, a) &= \sum_{s',r} p(s', r|s, a)(r + \gamma \max_{a'} q^*(s', a')) \end{aligned} \tag{5.9}$$

5.3 Value Iteration Algorithm

In the same idea as we did in the evaluation section, let us define the operator $\Phi : V \to \Phi(V)$ such that

$$\Phi(V)(s) := \max_{a \in A(s)} \sum_{s',r} p(s', r|s, a)(r + \gamma V(s')) \tag{5.10}$$

And thus the Bellman optimality equation ((5.9)) is equivalent to $V = \Phi(V)$, which can be solved iteratively from the starting point V_0. This method is guaranteed because Φ is a contracting operator ($\sim$ L-Lipschitz function).

Algorithm 4 Value Iteration Algorithm

Input: the guaranteed accuracy of estimation θ;
Initialization: $V(s) :=$ an arbitrary initial value for all s;
$\Delta \geq \theta$;
while $\Delta \geq \theta$ **do**
  $\Delta = 0$;
  for each state s **do**
    $v = V(s)$
    $V(s) = \max_a \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$;
    $\Delta = \max(\Delta, |v - V(s)|)$;
  end for
end while

5.4 Linear programming approach

The problem consists in computing the values of $v(s)$ for some abritrary weights $\alpha(s) > 0$.

$$\begin{aligned} &\min_{v(s)} \sum_s \alpha(s) v(s) \\ &v(s) \geq \sum_{s',r} p(s', r|s, a)[r + \gamma v(s')] \qquad \forall s \in S, a \in A \end{aligned} \tag{5.11}$$

And the dual is

$$\begin{aligned} &\max_{x(s,a) \geq 0} \left[\sum_{s' \in S, r \in R} p(s', r|s, a) r \right] x(s, a) \\ &\sum_{a \in A} x(s, a) = \alpha(s) + \gamma \sum_{s' \in S, a \in A, r \in R} p(s, r|s', a) x(s', a) \qquad \forall s \in S \end{aligned} \tag{5.12}$$

The variables $x(s, a)$ represent the total discounted probabilities of being in state s and take action a.

Algorithm 5 Policy Iteration Algorithm

Input: the guaranteed accuracy of estimation θ;
Initialization: $A(s) :=$ arbitrary action for all s;
$V(s) :=$ an arbitrary initial value for all s;
$\Delta \geq \theta$;
while $\Delta \geq \theta$ **do**
  $\Delta = 0$;
  for each state s **do**
    $v = V(s)$;
    OldAction:$= A(s)$;
    $V(s) = \max_a \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$;
    $A(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$;
    $\Delta = \max(\Delta, |v - V(s)|)$;
  end for
  Update the $V(s)$ by evaluating the new policy.
end while

5.5 Policy iteration algorithm

5.6 Reinforcement Learning

5.6.1 Temporal Difference Learning

Temporal Difference Learning is used when we do not the probabilities $p(s', r|s, a = \pi(s))$. In that case, we rather use a temporal difference error, measuring the difference between the current estimate $V(s)$ and $r + \gamma V(s')$ (see blue term).

Algorithm 6 TD(0) for estimating v_π

Input: π the policy to be evaluated;
  α: algorithm step size with $0 < \alpha \leq 1$;
Initialization: $V(s) :=$ an arbitrary initial value for all s;

for do each episode
  $s \leftarrow$ initial state;
  while s is not terminal **do**
    for do each time step t in the episode
      $a \leftarrow \pi(s)$;
      Take action q and observe r, s';
      $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$;
      $s \leftarrow s'$;
    end for
  end while
end for

5.6.2 Q-learning

Q-Learning works juste like TD Learning, but using $Q(s,a)$ instead of $V(s)$. The policy is then computed from the final iterate using the usual formula. The colored lines are

Algorithm 7 Q-learning for estimating π^*

Input: π: policy to be optimized;
  α: algorithm step size with $0 < \alpha \leq 1$;
Initialization: $Q(s,a) :=$ an arbitrary initial value for all (s,a);
for **do** each episode
  $s \leftarrow$ arbitrary initial state;
  while s is not terminal **do**
    for **do** each time step t in the episode
      Choose action a according to the learning policy, e.g. ϵ-greedy;
      Take action a and observe r, s';
      $Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$;
      $s \leftarrow s'$;
    end for
  end while
end for

for the comparison with the next algorithm.

5.6.3 SARSA

Algorithm 8 SARSA for estimating π^*

Input: π: policy to be optimized;
  α: algorithm step size with $0 < \alpha \leq 1$;
Initialization: $Q(s,a) :=$ an arbitrary initial value for all (s,a);
for **do** each episode
  $s \leftarrow$ arbitrary initial state;
  Choose a according to the learning policy, e.g. ϵ-greedy;
  while s is not terminal **do**
    for **do** each time step t in the episode
      Take action a and observe r, s';
      Choose a' from s' according to the learning policy derived from Q, e.g. ϵ-greedy;
      $Q(s,a) = Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$;
      $s \leftarrow s'$;
      $a \leftarrow a'$;
    end for
  end while
end for

Countable-state Markov Chains

A countable-state Markov chain contains an infinite but countable number of states, e.g. $\mathbb{N}$.

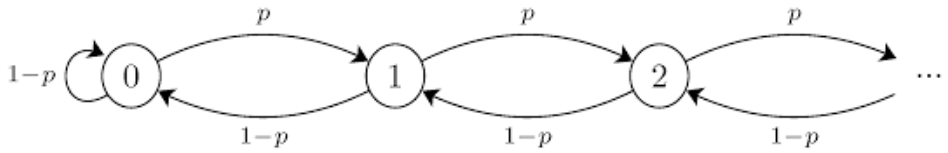


Figure 6.1: Example of countable-state Markov chain

In the example in figure 6.1, the first state is transient if $p < 0.5$ and reccurent otherwise.

6.1 Definitions

- $f_{ij}(n)$ is the probability that the first passage of the Markov chain by state j happens at time n, given that $X_0 = i$. Mathematically,

$$f_{ij}(n) = P[X_n = j, X_k \neq j \quad \forall k < n | X_0 = i] \tag{6.1}$$

  There exists a recurrence relation for $f_{ij}(n)$: $f_{ij}(n) = \sum_{k \neq j} p_{ik} f_{kj}(n-1)$, and $f_{ij}(1) = p_{ij}$.

- We denote $F_{ij}(n)$ the probability that the first passage of the Markov chain by state j happens at time n or before, given that $X_0 = i$. Mathematically,

$$F_{ij}(n) = \sum_{m=1}^{n} f_{ij}(m) = P[X_k = j \quad \text{for some } k \leq n | X_0 = i] \tag{6.2}$$

- A state is recurrent if $F_{jj}(\infty) = 1$;
- A state is transient if $F_{jj}(\infty) < 1$;
- We define the random variable T_{ij} as the time until the first passage from i to j. Hence T_{jj} is the time needed to come back to state j after leaving it.
- The mean of T_{ij} is

$$\bar{T}_{ij} = 1 + \sum_{n=1}^{\infty} (1 - F_{ij}(n)) \tag{6.3}$$

▲ Theorem: all states from a same class are of the same type.

▲ Theorem: Let j be recurrent and aperiodic, i be a state in the same class, then

$$\lim_n P[X_n = j | X_0 = i] = \frac{1}{\bar{T}_{jj}} \tag{6.4}$$

- A Markov chain is said to be irreducible if all states communicate.

▲ Theorem: For an irreducible Markov chain, if the system

$$\begin{cases} \pi_j = \sum_i \pi_i p_{ij} \quad \forall j \\ \sum_j \pi_j = 1 \\ \pi_j \geq 0 \quad \forall j \end{cases} \tag{6.5}$$

has a solution, then $\pi_i = 1/\bar{T}_{ii}$ and all states are positive-recurrent, i.e. $f_{ii}(\infty) = 1$ and $\bar{T}_{ii} < \infty$. Moreover, if all states are positive-recurrent, then the system has a solution.

6.2 Birth-death Markov chains

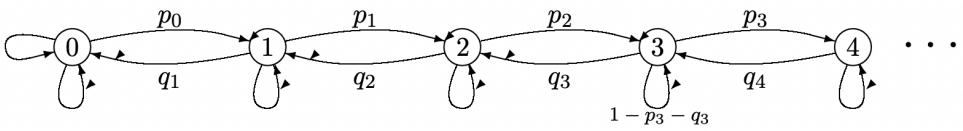


Figure 6.2: Example of birth-death Markov chain

This kind of Markov chain is used to model the evolution of a population or a queueing system. If the states are positive recurrent, we must have that

$$\pi_i p_i = \pi_{i+1} q_{i+1} \tag{6.6}$$

And so by recursion,

$$\begin{aligned} \pi_0 &= \frac{1}{1 + \sum_{i=1}^{\infty} \prod_{j=0}^{i-1} \rho_j} \qquad \rho_j = \frac{p_i}{q_{i+1}} \\ \pi_i &= \pi_0 \prod_{j=0}^{i-1} \rho_j \end{aligned} \tag{6.7}$$

→ Note: if $\rho_i = \rho < 1$ for all i, then a solution exists and all states are positive recurrent.

6.2.1 Reversibility

If the Markov chain is in steady state, then the backward chain is homogeneous, i.e.

$$P[X_{n-1} = j | X_n = i] = P[X_n = i | X_{n-1} = j] \frac{\pi_j}{\pi_i} \tag{6.8}$$

and the backward transition probabilities p_{ij}^* are defined as

$$p_{ij}{}^* = p_{ji}\frac{\pi_j}{\pi_i} \iff \pi_i p_{ij}^* = \pi_j p_{ji} \tag{6.9}$$

A Markov chain is said to be reversible if $p_{ij}^* = p_{ij}$, or equivalently, $\pi_i p_{ij} = \pi_j p_{ji}$.

Theorem 6.1. Every birth-death chain with a steady-state probability distribution is reversible, due to the way the steady-state probabilities are computed (see equation (6.6)).

Theorem 6.2. Assume that an irreducible Markov chain has transition probabilities p_{ij}. Suppose $\{\pi_i\}$ is a set of positive numbers summing to 1 and satisfying $\pi_i p_{ij} = \pi_j p_{ji}$ for all i, j. Then, $\{\pi_i\}$ is the steady-state distribution for the chain and the chain is reversible.

6.3 Processor sharing

The system considered here consists in m customers $c_1, \ldots, c_m$ where they are each served for an increment of time δ. After c_m has been served, the server returns and start again. It is a cyclic serving order. When the service of c_i is completed, the client leaves and m is reduced, and when a client arrives, m increases.

$\rightarrow$ Note: processor sharing is the limit of round-robin service as the increment δ goes to 0.

6.3.1 Model

- Assume a Bernoulli arrival process in which the probability of an arrival in an interval δ is $\lambda\delta$;
- Assume the service time of customers to be IID arithmetic (i.e. mutliple of δ) random variables Z_i with span δ.

Let $f(j) = P[Z_i = j\delta]$ and $\bar{F}(j) = P[Z > j\delta]$. Let $g(j)$ be the probability that the customer that was served during j increments departs after one more increment of service: $g(j) = \frac{f(j+1)}{\bar{F}(j)}$.
Here is the dynamical system:

- A new customer arrives with probability $\lambda\delta$ and goes to first position;
- The customer in first position is served for an increment δ;
- The customer that was served departs if service is complete;
- Otherwise, the customer goes to the back of the queue.

The states of the system are $s = (m, z_1, \ldots, z_m)$, and we can show that the transition probabilities are given by

$$\begin{aligned} p_{s,r(s)} &= (1 - \lambda\delta)(1 - g(z_1)) \\ p_{s,d(s)} &= (1 - \lambda\delta)g(z_1) \\ p_{s,a(s)} &= \lambda\delta(1 - f(1)) \\ p_{s,s} &= \lambda\delta f(1) \end{aligned} \tag{6.10}$$

where $r(s) = (m, z_2, \ldots, z_m, z_1 + 1)$ (=rotation), $d(s) = (m - 1, z_2, \ldots, z_m)$ (=departure), $a(s) = (m + 1, 1, z_1, \ldots, z_m, 1)$ (=arrival).

Markov processses

7.1 Semi-Markov processes

7.1.1 Definitions

- A semi-Markov process is a Markov chain with holding times between transitions that are random variables that depend only from the starting and ending state of the transition. We denote $S_1 < S_2 < \dots$ the times of the successive transitions, and $X_0, X_1, \dots$ the successives states. Then, $X(t) = X_n$ for all $S_n \leq t < S_{n+1}$.
- The embedded Markov chain is formed by the successive states of the semi-Markov process: $P[X_n = j | X_{n-1} = i] = P_{ij}$.
- $U_n \coloneqq S_n - S_{n-1}$ is a random variable depending only on X_{n-1} and X_n:

$$P[U_n \leq u | X_{n-1} = i, X_n = j] =: G_{ij}(u) \qquad \mathbb{E}[U_n | X_{n-1} = i, X_n = j] =: \bar{U}(i,j) \tag{7.1}$$

7.1.2 Stationary probabilities

Theorem 7.1. If the embedded Markov chain is irreducible and positive recurrent, then the stationary probability p_i of being in state i is given by

$$p_i = \frac{\pi_i \bar{U}(i)}{\sum_j \pi_j \bar{U}(j)} \tag{7.2}$$

where $\bar{U}_i = \sum_j P_{ij} \bar{U}(i,j)$.

7.2 Markov processes

7.2.1 Definitions

- A Markov process is a semi-Markov process where the holding times between transitions are exponentially distributed. Hence the holding time depends only on the current state:

$$P[U_n \leq x | X_{n-1} = i, X_n = j] = 1 - e^{-\nu_i x} \tag{7.3}$$

  where ν_i is the exponential parameter [transition/hour].

7.2.2 Transition rates

→ Note: the transition rates are not transition probabilities and can thus be bigger than 1.

Let $Y(t)$ be the time until the next transition.

$$P[Y(t) \leq x, X(t + Y(t)) = j | X(t) = i, \{X(\tau); \tau < t\}] = p_{ij} e^{-\nu_i x} \tag{7.4}$$

where the term p_{ij} is the transition probability of the embedded Markov chain and the second term is the exponential distribution. Taking x infinitesimal, we find the transition rate between state i and j:

$$q_{ij} = \nu_i p_{ij} \qquad \nu_i = \sum_j q_{ij} \tag{7.5}$$

7.2.3 Equivalences

As a summary of what has been seen until now, here is an equivalence of different stochastic processes:

- A semi-Markov process with exponentially distributed holding times.
- N Poisson processes with rates ν_i. If $X(t) = i$, the next transition will occur when an arrival occurs for the Poisson process i and the choice of the next state j is made with probability p_{ij}.
- N^2 Poisson processes with rates q_{ij}.

7.2.4 Stationary probabilities

The steady state probabilities $\{\pi_i\}$ are found solving the system $\pi_j = \sum_i \pi_i p_{ij}$, and so we can find

$$p_i = \frac{\pi_i / \nu_i}{\sum_k \pi_k / \nu_k} \tag{7.6}$$

And if the sum at the denominator is finite,

$$\pi_i = \frac{p_i \nu_i}{\sum_k p_k \nu_k} \tag{7.7}$$

→ Note: A Markov process is irreducible iff the embedded Markov chain is irreducible.

7.2.5 Chapman-Kolmogorov differential equation (what for?)

Let $P_{ij}(t) = P[X(t) = j | X(0) = i]$. Then

$$\frac{dP_{ij}(t)}{dt} = \sum_{k \neq i} q_{ik} P_{kj}(t) - \nu_i P_{ij}(t) \tag{7.8}$$

7.2.6 Transient equation

Let us define the matrix Q:

$$Q_{ij} = q_{ij} \qquad Q_{ii} = -\sum_{j \neq i} q_{ij} = -\nu_i \tag{7.9}$$

Then, Chapman-Kolmogorov differential equation can be written as

$$\frac{dP(t)}{dt} = QP(t) \tag{7.10}$$

And, with $P(0) = I$,

$$P(t) = e^{Qt} = \sum_{n=0}^{\infty} \frac{Q^n}{n!} t^n \tag{7.11}$$

Theorem 7.2. Let Q be the transition rate matrix associated to an irreducible Markov process with n states. Then, Q has an equivalue equal to 0 with the corresponding right eigenvector being $\mathbb{1}_n^T$ and left eigenvector $p = (p_1, \ldots, p_n) > 0$. All other eigenvalues have strictly negative real part.

Defining λ_i as the eigenvalues of Q,

$$P(t) = \sum_{i=0}^{n} \nu_i e^{\lambda_i t} \pi_i \tag{7.12}$$

where ν_i and π_i are respectively the right and left eigenvector associated to λ_i.

7.3 Markovian queueing models

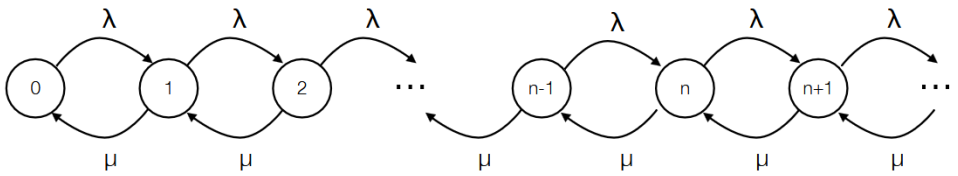


Figure 7.1: M/M/1 queueing model

7.3.1 M/M/1 queue

In a M/M/1 queue, the arrival process is a Poisson(λ) and the service time follows an Exponential(μ). Therefore, the transition rate matrix is

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \ldots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \ldots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \ldots \\ 0 & 0 & \mu & -(\lambda+\mu) & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{7.13}$$

We make the sensible assumption that $\lambda < \mu$ (otherwise the queue would grow indefinitely). The steady-state probabilities are calculated as usual and give

$$p_0 = \frac{1}{\sum_{i=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^i} = 1 - \frac{\lambda}{\mu} \qquad p_i = \left(\frac{\lambda}{\mu}\right)^i \left(1 - \frac{\lambda}{\mu}\right) \qquad \forall i > 0 \tag{7.14}$$

And the average number of jobs in the system is

$$L = \sum_{i=0}^{\infty} i p_i = \left(1 - \frac{\lambda}{\mu}\right) \sum_{i=0}^{\infty} n \left(\frac{\lambda}{\mu}\right)^n = \frac{\lambda/\mu}{1 - \lambda/\mu} = \frac{\lambda}{\mu - \lambda} \tag{7.15}$$

We can also calculate the following parameters:

- Average time spent in the system: $W = L/\lambda = 1/(\mu - \lambda)$;
- Average time spent in the queue: $W_q = W - 1/\mu = \frac{\lambda}{\mu(\mu-\lambda)}$;
- Average number of clients waiting in the queue: $L_q = \lambda W_q = \frac{\lambda^2}{\mu(\mu-\lambda)}$;

Let us now calculate the probability to wait less than t minutes in the queue. By definition,

$$P[W < t | L = n] = \int_0^t \mu e^{-\mu\tau} \frac{(\mu\tau)^n}{n!} d\tau \tag{7.16}$$

And so

$$P[W < t] = \sum_{n=0}^{\infty} P[W < t | L = n] P[L = n] = 1 - e^{-(\mu-\lambda)t} \tag{7.17}$$

7.3.2 M/M/s queue

The M/M/s queue is a generalization of the M/M/1 queue to s servers instead of a single one.

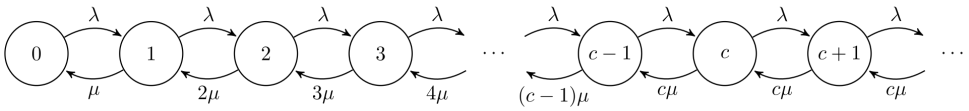


Figure 7.2: M/M/s queueing model

→ Note: the service time is not identical between every node: when there are less than s people in the system, not all servers are working.

→ Note: once again, we need an assumption for the convergence. Here, it is $\lambda/s\mu < 1$.

The steady state probabilities are

$$p_0 = \left(\left[\sum_{n=0}^{s-1} \frac{\lambda^n}{n!\mu^n} \right] + \frac{\lambda^s}{s!\mu^s} \frac{1}{1 - \frac{\lambda}{s\mu}} \right)^{-1} \tag{7.18}$$

and

$$p_n = \frac{\lambda^n}{n!\mu^n} p_0 \quad \forall n \leq s \qquad\qquad p_n = \frac{\lambda^s}{s!\mu^s} \left(\frac{\lambda}{s\mu}\right)^{n-s} p_0 \quad \forall n > s \tag{7.19}$$

And once again, we have some parameters:

- Averaeg number of clients waiting in the queue: $L_q = p_0 \frac{\lambda^s}{s!\mu^s} \frac{\lambda/s\mu}{(1-\lambda/s\mu)^2}$;