Advanced Topics in Stochastic Modelling

40.305 Advanced Topics in Stochastic Modeling

Yustynn Panicker

April 30, 2019

Contents

1 Meta

1.1 Professor

Dr Karthyek Murthy

1.2 Office Hours

Thursday, 5-6pm

1.3 Grading

Item	Grading
HW	20%
2 Mini-Projects	20%
Class Participation	10%
Final Exam	50%

1.4 Mini-Projects

1.4.1 Regenerative Simulation Exercise

1.4.2 Choose 1, Submit Report by W13

1.4.2.1 Simulate Real-World System

- Must have regen structure
- Perform statistical analysis via regenerative method

1.4.2.2 Theoretical Stduy

- Compare performances of different service disciplines for single server queue

1.4.3 Problem Sets

- Submit on eDimension
- Acknowledge collaborators

2 L1: Introduction, Renewal Processes, Renewal Reward Theorem

2.1 Introduction

2.1.1 Motivation

- Analysis of systems over time
- Markov Chains rely on memoryless property, which is not always the case

2.1.1.1 When you can't use Markov Chains

- Not everything is exponentially distributed => not everything can be modelled with Markov chains

Possible Solution: Use simulations instead of Markov Chains

2.1.2 Syllabus

2.1.2.1 Renewal-Reward Theory and Applications

- Renewal-reward theorem and its application in characterizing long-run time averages
- Little's Law and PASTA property

- Inspection paradox, size-biased distributions
- Applications to M/G/1 queue analysis
- Regenerative processes
- Regenerative method of statistical analysis in simulation

2.1.2.2 Discrete-Time Martingales and its Applications

- Conditional expectation and martingales
- Gambling strategies and Martingale transforms
- Optional stopping and applications to gambler's ruin
- Application to arbitrage-free asset pricing
  - Binomial model (1 period and multi-period)
  - Extensions of binomial model to continuous time, Geometric Brownian motion and Black-Scholes formula (if we have time to do it)

2.2 Renewal Processes

2.2.1 What Are They?

Processes where interarrival times are iid

2.2.2 Non-Immediate Expiry

Often, we asume F(0) $<$ 1 (which codifies that each thing doesn't immediately expire, since they're iid)

2.2.3 Some Formulas

$$N(t) = \max\{n : S_n \leq t\}$$
$$F(t) = P(X_i \leq t)$$

2.2.4 Some Properties

$$\{N(t) \geq n\} \Leftrightarrow \{S_n \leq t\}$$
$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{\mathbb{E}[X_i]}, \text{ for any i}$$

2.2.5 Renewal Theorem

2.2.5.1 Intuition Average number of arrivals per time (which is $\frac{N(t)}{t}$) is obviously the rate of arrivals. Rate of arrivals is obviously the inverse of the time between arrivals (which is $\mathbb{E}[X]$)

2.2.5.2 Theorem For a renewal process, where $\mathbb{E}[X]$ is the mean time between renewals we have

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{\mathbb{E}[X]} \text{ with probability 1}$$

2.2.6 Features/Properties of Poisson Processes

2.2.6.1 Independent Increments Number of events in two disjoint intervals are independent

2.2.6.2 Stationary Increments Number of events in an interval is Poisson (rate × length of the interval)

2.2.6.3 Mergeable If N_i is Poisson with rate λ_i where i=1,2, then

$$N_1(t) + N_2(t) \sim \text{Poisson}(\lambda_1 t + \lambda_2 t)$$

2.2.6.4 Splittable Given a Poisson process with

- rate λ
- type A events occuring with probability p
- type B (where $B = \neg A$) events occuring with probability $1 - p$

$$N(t) \sim \text{Poisson}(\lambda) \quad \rightarrow \quad N_A(t) \sim \text{Poisson}(p\lambda t), \quad N_B(t) \sim \text{Poisson}((1-p)\lambda t)$$

2.3 Renewal Reward Theorem

2.3.1 Renewal Reward Theorem

2.3.1.1 Intuition You just scale the renewal theorem by expected reward.

2.3.1.2 Theorem

$$\frac{R(t)}{t} \rightarrow \frac{\mathbb{E}[R_i]}{\mathbb{E}[X_i]}$$

Note that if distribution is iid (which it generally is) then you can use any i, e.g. 1

$$\frac{R(t)}{t} \rightarrow \frac{\mathbb{E}[R_1]}{\mathbb{E}[X_1]}$$

2.3.2 Alternating Renewal Process

2.3.2.1 Set Up

- Let $Y_1, Y_2, \ldots$ be iid with:
  - distribution F
  - mean μ_F
- Let $Z_1, Z_2, \ldots$ be iid with:
  - distribution G
  - mean μ_G

2.3.2.2 Lightbulb Frame example

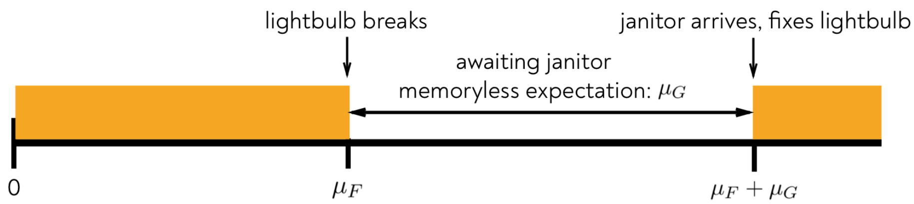
- Imagine Y is the time before a lightbulb conks out.
- Imagine Z is the interarrival time of a janitor who checks on the lightbulb
- If bulb is broken, janitor replaces it with a new one.

2.3.2.3 Property

Fraction of time in state 1 (in frame, that the lightbulb is on):

$$\frac{\mu_F}{\mu_F + \mu_G}$$

2.3.2.4 Intuition



Consider one cycle (light bulb works, then it breaks). What we want is $\frac{\text{time in state 1}}{\text{cycle time}}$

Numerator (μ_F) is the amount of time it works (state 1). This is obvious - it's the amount of time before the lightbulb breaks, meaning it's unbroken during this time.

Denominator is the total time of the cycle. The light bulb breaks, *then* the janitor arrives (assume memoryless, so you can take the expected janitor arrival as simply μ_G).

2.3.2.5 ???

I think this is only true when Z is poisson. Otherwise it really mkes no sense, you need the memoryless property for the intuition to work.

UPDATE: checked with Karthyek. Yup, the way he presented it was very misleading. Works for more than just poisson, but the distribution MUST depend on F. E.g. $G = X + (\text{some random variable})$

3 L2: Little's Theorem

3.1 Little's Law

3.1.1 Formula

$$\bar{N}^{time} = \lambda \bar{T}^{people}$$

3.1.2 Corollary 1: Number of people in queue

$$\bar{N}_Q^{time} = \lambda \bar{T}_Q^{people}$$

Number of people in the queue is number of people *not being served.*

3.1.3 Corollary 2: Utilization Law

$$\text{Server Utilization } = \rho_i = \frac{\lambda_i}{\mu_i}$$

- λ is arrival rate
- μ is service rate

4 L3: PASTA, Start of M/G/1

4.1 PASTA

Poisson Arrivals See Time Averages

4.1.1 Specific Example

$$I(t) = \begin{cases} 1 & \text{if the workstation is busy at time } t \\ 0 & \text{otherwise} \end{cases}$$
$$I_n = \begin{cases} 1 & \text{if the workstation is busy just prior to the } n \text{ th arrival,} \\ 0 & \text{otherwise.} \end{cases}$$

Then longrun fraction of arrivals that find the workstation busy = longrun fraction of time that the workstations are busy

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_n = \lim_{t\to\infty} \frac{1}{t} \int_0^t I(t)dt$$

4.1.2 General Theorem

$$I_B(t) = \begin{cases} 1 & \text{if the system state } X(t) \text{ is in } B \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$
$$I_n(B) = \begin{cases} 1 & \text{if the system state } X(t) \in B \text{ just prior to the } n \text{ th arrival,} \\ 0 & \text{otherwise.} \end{cases}$$

Then with probability 1,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_n(B) = \lim_{t\to\infty} \frac{1}{t} \int_0^t I_B(t)dt$$

4.1.3 Useful PASTA results for M/G/1

- Long-run fraction of arrivals that see empty queue = $1 - \frac{\lambda}{\mu} = 1 - \rho$
- Long-run fraction of arrivals that must wait before service = $\frac{\lambda}{\mu} = \rho$

4.2 M/G/1

- S_e is the remaining service time of the current job in queue,
- T_Q is the time in queue

$$\mathbb{E}[T_Q] = \frac{\rho}{1-\rho}\mathbb{E}[S_e]$$

5 L4: Excess, Age, Inspection Paradox, PK-formula for M/G/1

5.1 Age and Excess

5.1.1 Some Preliminary Notation

Term	Meaning
$N(t)$	Number of renewals up till time t
S_n	End time of cycle n ($S_n = \sum_{i=1}^{n} X_i$)
X_i	Length of cycle i (yes it's retardedly overloaded. Don't blame me, blame Karthyek)

5.1.2 Age

Age of cycle

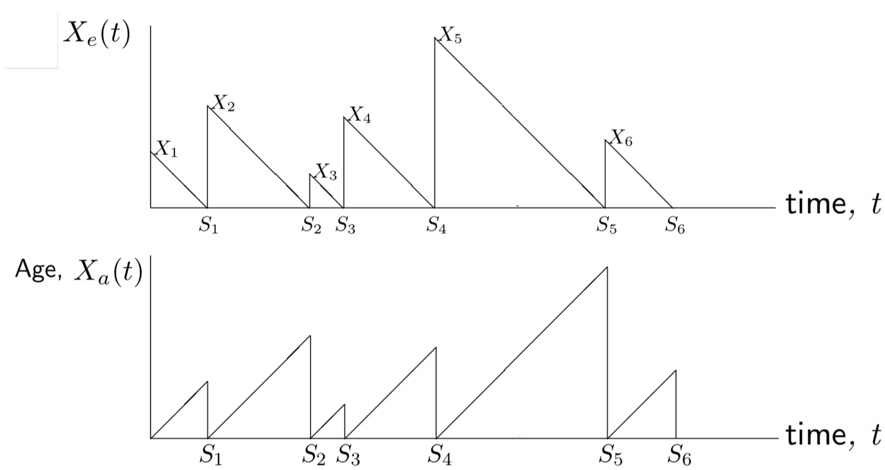
$$X_a(t) = t - S_{N(t)}$$

5.1.3 Excess

Time remaining until a cycle ends (yes, the naming is stupid)

$$X_e(t) = S_{N(t)+1} - t$$

5.1.4 Relationship



They're essentially the same thing, just from different directions

5.1.5 Coefficient of Variation

It's a relative measure of variability (variance over the squared mean)

Now, you may be asking yourself: why would you square the mean. Why not just square root the variance like a normal person, so you can talk in terms of measuring the standard deviation as a percentage of the mean? The answer to this, my friend, is yes. ???

But anyway, it lets us simplyify the formulas in the next section, so that's nice.

$$C_X^2 = \frac{Var[X]}{\mathbb{E}[X]^2} = \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2} - 1$$

5.1.6 Formulas

By renewal reward theorem,

$\mathrm{E}[\mathrm{T}_e] = \lim_{t\to\infty} \frac{1}{t} \int_0^t X_e(s)ds = \frac{\mathbb{E}[\int_0^X (X-s)ds]}{\mathbb{E}[X]} = \frac{\mathbb{E}[X^2]}{2\mathbb{E}[X]} = \frac{\mathbb{E}[X]}{2}(C_X^2 + 1)$

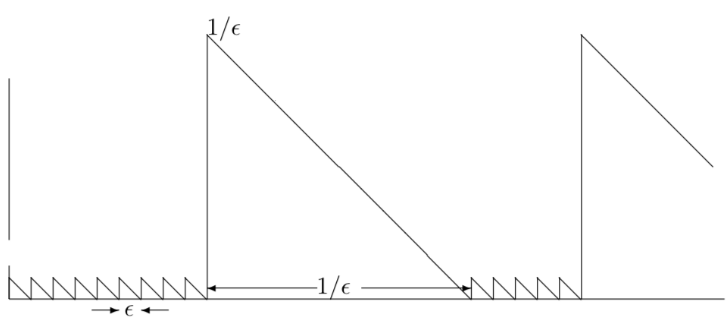
This is obviously the same as time-average age, as they're the same thing from different directions.

$$\mathbb{E}[T_a] = \lim_{t\to\infty} \frac{1}{t} \int_0^t X_a(s)ds = \frac{\mathbb{E}\left[\int_a^X (s)ds\right]}{\mathbb{E}[X]} = \frac{\mathbb{E}[X^2]}{2\mathbb{E}[X]} = \frac{\mathbb{E}[X]}{2}(C_X^2 + 1)$$

5.2 Inspection Paradox

Depending on variance of cycle lengths, a new job faced with busy workers may expect to wait for the next regeneration longer than the expected time to the next regeneration point in general.

The reason is strange: if there's a lot of cycle length variance, then odds are higher that this job landed in one of the cycles with a long length. Meaning that it'll take longer to reach the next regeneration (since the cycle the job is in in will take longer to end)



The image here could be better (I got it from the slides), but you get the idea.

5.3 PK-Formula for Mean Delay in M/G/1 (FCFS)

- Pollaczek-Khinchin
- Computes the long-run average waiting time (mean delay) in M/G/1 queues

Where:

- S: service time of job
- T_Q: time in queue (prior to service starting)

$$\begin{aligned}
\mathbb{E}[T_Q] &= \frac{\rho}{1-\rho} \cdot \frac{\mathbb{E}[S^2]}{2\mathbb{E}[S]} \\
&= \frac{\rho}{1-\rho} \cdot \frac{\mathbb{E}[S]}{2} \cdot (C_S^2 + 1) \\
&= \frac{\lambda \mathbb{E}[S^2]}{2(1-\rho)}
\end{aligned}$$

Note that increases in ρ (server utilization) non-linearly increase the mean delay

6 L5: Priority Queues, Setup Times

6.1 M/G/1 Priority Queues

6.1.1 Types

1. Non-Preemptive Priority Queues *(our main focus)*
   - E.g. hospital ER queues, airline boarding
2. Preemptive Priority Queues *(not covered)*
   - E.g. Computer resource scheduling

6.1.2 Terms

Note that by convention, priority 1 has higher priority than priority 2

Term	**Meaning**
S_k	(random) Size (time to process) of single priority k job
$\mathbb{E}[N_q(k)] = \lambda_k \cdot \mathbb{E}[T_q(k)]$	Avg num priority k jobs in queue. The RHS formula is because it's stable, so that many new jobs come in while processing the current jobs
$\mathbb{E}[T_q(k)]$	Avg queue time (mean delay) of all priority k jobs
$\mathbb{E}[T(k)]$	Avg total time of priority k jobs
$\lambda_k = \lambda \cdot p_k$	Average arrival rate of jobs of priority k; p_k is probability of job being priority k
$\rho_k = \lambda_k \cdot \mathbb{E}[S_k]$	Avg fraction of time the server spends on priority k jobs ($\rho = \sum_{k=1}^{n} \rho_k$)
$\rho = \sum_{k=1}^{n} \rho_k$	Server utilization (required to be less than 1)

??? Why require $\rho < 1$? What's wrong with exactly 1?

6.1.3 Average Time in Queue for NP Priority k

NP is non-preemption. In this course, we don't consider preemptive priority queues.

6.1.3.1 Formula

$$\mathbb{E}[T_Q(k)] = \frac{\rho\frac{\mathbb{E}[S^2]}{2\mathbb{E}[S]}}{\left(1 - \sum\limits_{i=1}^{k} \rho_i\right)\left(1 - \sum\limits_{i=1}^{k-1} \rho_i\right)}$$

6.1.3.2 Explanation Grepping this takes a little effort, so let's break it down.

There are 3 components:

Component	Meaning
$\rho \cdot \mathbb{E}[S_e]$	Expected time for current job that is being processed to finish. Interestingly, the ρ acts as probability here.
$\sum\limits_{i=1}^{k} \rho_i \cdot \mathbb{E}[N_Q(i)][S_i] = \sum\limits_{i=1}^{k} \rho_i \cdot \mathbb{E}[T_Q(i)]$	Expected time for all relevant-prioritied jobs currently in queue to finish
$\sum\limits_{i=1}^{k} \sum\limits_{j=1}^{i-1} \mathbb{E}[T_Q(i)] \cdot \lambda_j \mathbb{E}[S_j] = \sum\limits_{i=1}^{k} \sum\limits_{j=1}^{i-1} \rho_j \cdot \mathbb{E}[T_Q(i)]$	New relevant-prioritied jobs come in while the other jobs are being processed. This accounts for the time they take.

$$\begin{aligned}
\mathbb{E}[T_Q(k)] &= \rho \cdot \mathbb{E}[S_e] + \sum_{i=1}^{k} \rho_i \cdot \mathbb{E}[T_Q(i)] + \sum_{i=1}^{k} \sum_{j=1}^{i-1} \rho_j \cdot \mathbb{E}[T_Q(i)] \\
&= \rho \cdot \mathbb{E}[S_e] + \sum_{i=1}^{k} \sum_{j=1}^{i} \rho_j \cdot \mathbb{E}[T_Q(i)]
\end{aligned}$$

You math it a little (bring all the $T_Q(k)$ terms together, use expected excess formula to sub out $\mathbb{E}[S_e]$, etc), and eventually get this:

$$\mathbb{E}[T_Q(k)] = \frac{\rho\frac{\mathbb{E}[S^2]}{2\mathbb{E}[S]}}{\left(1 - \sum\limits_{i=1}^{k} \rho_i\right)\left(1 - \sum\limits_{i=1}^{k-1} \rho_i\right)}$$

Even these terms, you can kinda see as reminiscent of our original logic. Kind of. I don't fully get it tbh.

E.g.

- $(1 - \sum\limits_{i=1}^{k} \rho_i)$ in the denominator accounts for the jobs already in the queue. This represents the fraction of remaining time of the server, assuming it processed only the current jobs
- $(1 - \sum\limits_{i=1}^{k-1} \rho_i)$ in the denominator accounts for the jobs already in the queue. This represent the fraction of remaining time left to actually process the current jobs (because other jobs come in)

6.1.4 Intuition for why priority queues are faster than FCFS for low-k

First off, Karthyek's notes for this aren't great. In fact, I think it's wrong for several reasons. One is that he rests on the assumpiton that for low k, $\rho_k << \rho$. This obviously need not be the case. My explanation is more general.

Recall:

$$\mathbb{E}\left[T_Q(k)\right]^{\text{FCFS}} = \frac{1}{1-\rho} \cdot \frac{\rho\mathbb{E}\left[S^2\right]^{\text{FCFS}}}{2\mathbb{E}[S]^{\text{FCFS}}}$$

$$\mathbb{E}\left[T_Q(k)\right]^{\text{NP- Priority}} = \frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \cdot \frac{\rho\mathbb{E}\left[S^2\right]^{\text{NP-Priority}}}{2\mathbb{E}[S]^{\text{NP-Priority}}} =$$

Okay, so one the difference is clearly this scaling:

$$\text{FCFS: } \frac{1}{1-\rho}$$

$$\text{NP Priority: } \frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)}$$

Obviously, the denominator $\in [0,1]$. The lower it is, the higher the scaling is $\implies \mathbb{E}[T_Q(k)]$.

A given k-priority benefits from a priority queue so long as the FCFS denominator is lower than the priority denominator.

E.g. for k=1, NP Priority *always* dominates (minimally, at least as good as) FCFS.

Here's a quick proof by looking at the denominators:

$$\text{FCFS for k=1: } 1-\rho$$

$$\text{NP Priority for k=1: } 1-\rho_1$$

As $\rho_1 \leq \rho$,

$$\mathbb{E}\left[T_Q(k)\right]^{\text{NP- Priority}} \leq \mathbb{E}\left[T_Q(k)\right]^{\text{FCFS}}$$

6.1.5 Comparison with FCFS

$$\mathbb{E}\left[T_Q\right]^{\text{NP}-\text{Priority}} = \sum_{k=1}^{n}\mathbb{E}\left[T_Q(k)\right]\cdot p_k = \sum_{k=1}^{n}\mathbb{E}\left[T_Q(k)\right]\cdot\frac{\lambda_k}{\lambda}$$

This pre-supposes that you know $\mathbb{E}[T_Q(k)]$, of course. You have to go calculate that yourself.

6.2 M/G/1 with Setup Times

The idea is that you've got some setup time T (random variable).

From expected excess formula, we have:

$$\mathbb{E}[T_e] = \frac{\mathbb{E}[T^2]}{2\mathbb{E}[T]}$$

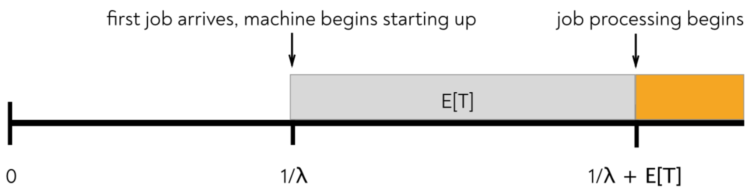
6.2.1 Mean Waiting Time

$$\frac{\rho}{1-\rho}\cdot\frac{\mathbb{E}\left[S^2\right]}{2\mathbb{E}[S]} + \left(\frac{1/\lambda}{1/\lambda+E[T]}E[T] + \frac{E[T]}{1/\lambda+E[T]}\frac{E\left[T^2\right]}{2E[T]}\right)$$

6.2.1.1 Contribution from the "restarting the machine" case

$$(1-\rho)\left(\frac{1/\lambda}{1/\lambda+E[T]}E[T]+\frac{E[T]}{1/\lambda+E[T]}\frac{E\left[T^2\right]}{2E[T]}\right)$$

This is probability of occurrence multiplied by expected time.



The probability of occurrence can be seen as meeting two consecutive conditions (condition `1` + the condition for either `Subcase A` or `B`):

1. Server not being used $(1-\rho)$
2.
   - `Subcase A`. Machine not yet starting up (no jobs) $(\frac{1/\lambda}{1/\lambda+\mathbb{E}[T]})$
   - `Subcase B`. Machine already in the process of starting up (no jobs) $(\frac{\mathbb{E}[t]}{1/\lambda+\mathbb{E}[T]})$

For `Subcase A`, expected time is $E[T]$. For `Subcase B`, expected time is $E[T_e]=\frac{E[T^2]}{2E[T]}$.

Adding up these subcases, we retrieve our original contribution:

$$(1-\rho)\left(\frac{1/\lambda}{1/\lambda+\mathbb{E}[T]}\mathbb{E}[T]+\frac{\mathbb{E}[T]}{1/\lambda+\mathbb{E}[T]}\frac{\mathbb{E}\left[T^2\right]}{2\mathbb{E}[T]}\right)$$

6.2.1.2 Contribution from processing jobs in queue

$$\frac{\rho}{1-\rho}\cdot\frac{\mathbb{E}\left[S^2\right]}{2\mathbb{E}[S]}$$

Literally just the PK formula. This is the basic formula for $\mathbb{E}[T_Q]$. We're just adding the extra stuff because of the start-up time.

7 L6: Regenerative Methods for Simulations

7.1 Other Methods

7.1.1 Replication/Deletion Method

The simulation starts in a non-steady state, unless you're crazy lucky. Discard the initial interval of non-steady state of the simulation. (called the `burn-in` period by Karthyek)

7.1.1.1 Problems

- Tough to determine burn-in period
- Have to do this every simulation iteration. The wasted computation time gets multiplied by number of simulations

7.1.2 Batch Means Method

- Have one really long simulation
- Discard the burn-in
- Split up remainder into batches, pretend they're independent

7.1.2.1 Problems

- Tough to determine burn-in period
- Different batches may be correlated

7.2 Regenerative

7.2.1 Regenerative Process Definition

A stochastic process $\{ X(t), t \in T \}$ with time-index set T is regenerative if $\exists$ a (random) epoch S_1 such that the following conditions are met:

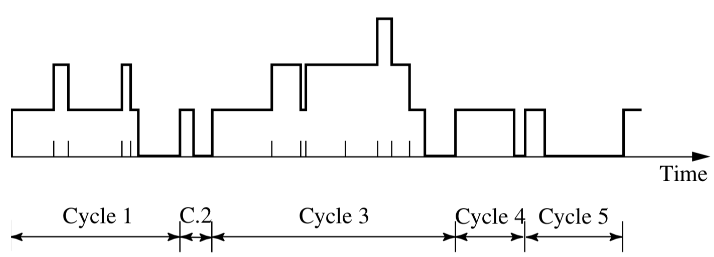
- $\{ X(t + S_1), t \in T \}$ is independent of $\{ X(t), 0 \leq t < S_1 \}$ *(memoryless)*
- $\{ X(t + S_1), t \in T \}$ is identically distributed as $\{ X(t), t \in T \}$ *(distribution is time-invariant)*

Note the generality here: we're doing this on purpose to extend regeneration beyond just Poisson processes.

7.2.2 Problems

- Not every simulation meets the regenerative process criteria (though the slides hint that a surprising amount do meet it)

7.2.3 Cycle



Defined as the time between two consecutive regeneration epochs. Note that of course, this may not be constant.

Generally, you set some state as the regeneration state. E.g. you choose state 0, then your cycle is the time from an occurrence of state 0 to the next occurrence of state 0.

7.2.4 Renewal Reward Theorem

Where R_i is the reward for cycle C_i,

$$\lim_{t \to \infty} \frac{R(t)}{t} = \frac{\mathbb{E}[R_1]}{\mathbb{E}[C_1]}$$

$$\lim_{t \to \infty} \frac{\mathbb{E}[R(t)]}{t} = \frac{\mathbb{E}[R_1]}{\mathbb{E}[C_1]}$$

7.2.5 General Method

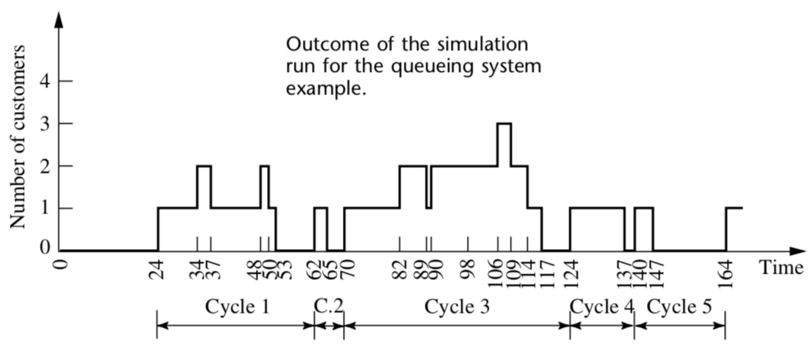
1. Choose number of cycles
2. Run simulation, noting `regeneration times` (cycle end times)
3. Compute reward per cycle
4. Compute steady-state average reward estimate $(\frac{\bar{R}}{\bar{C}}\}$
5. Compute confidence interval

7.2.5.1 3) Computing Reward Per Cycle Pretty obvious. Just do it.

S_i is regeneration time of cycle i, hence $C_i = S_i - Si - 1$

Personally, rather than leave S_0 undefined and have some special case for computing C_1, I'd define S_0 = steady state start = end of burn-in

$$R_n = \int_{S_{n-1}}^{S_n} f(X_s)ds$$



Cycle, i	Cycle length, C_i	Reward, $R_i = \int_{S_{i-1}}^{S_i} X(s)ds$
1	38	29 + 3 + 2 = 34
2	8	3
3	54	47 + 7 + 24 + 3 = 81
4	16	13

7.2.5.2 5) Computing Confidence Interval The challenge is really just computing the sample variances. This is dependent on both R and C.

$$s_{RR}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2$$
$$s_{CC}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(C_i - \bar{C})^2$$
$$s_{RC}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(C_i - \bar{C})(R_i - \bar{R})$$
$$s^2 = s_{RR}^2 - 2\frac{\bar{R}}{\bar{C}}s_{RC}^2 + \left(\frac{\bar{R}}{\bar{C}}\right)^2 s_{CC}^2$$

Note that the s^2 is derived using the Delta Method. What is the Delta Method, you ask? Yes. ???

8 L7,8: Conditional Expectation, Martingales

8.1 Conditional Expectation

Note to self: not fully done with this...

8.1.1 General Formula

This is the continuous case. Discrete case follows easily.

$$\mathbb{E}[X|Y] = \int_{-\infty}^{\infty} X \cdot P(X = x|Y)dx = \int_{-\infty}^{\infty} X \cdot \frac{P(X = x, Y)}{P(Y)}dx$$

Important to note:

- conditional expectation *usually* results in a random variable
- $E[X|Y] = E[X]$ if $X \perp Y$

8.1.2 Law of Iterated Expectations

$$\mathbb{E}\left[\mathbb{E}[X|Y_1, Y_2, \ldots, Y_n]\right] = \mathbb{E}[X]$$

This is actually obvious[2] when you note the linearity of expectation. The inner expectation results in some affine composition of random variables. So you take expectation of those, and obviously you get a constant.

8.1.2.1 Usefulness E.g. you want to find:

$$\mathbb{E}[X_1 + X_2 + \ldots + X_N]$$

Here, X and N are independent random variables

You can note this:

$$\mathbb{E}[X_1 + X_2 + \ldots + X_N] = \mathbb{E}[\mathbb{E}[X_1 + X_2 + \ldots + X_N|N]] = \mathbb{E}[X]\mathbb{E}[N] = N\mathbb{E}[X]$$

8.2 Martingales

8.2.1 Definition of Martingale

$M = (M_n : n \geq 0)$ is a martingale wrt $X = (X_n : n \geq 1)$ if for each n,

$$E\left[M_{n+1} | M_0, X_1, X_2, \ldots, X_n\right] = M_n$$

Note the dependence on the starting point, M_0.

For simplicity, we denote $\mathcal{H}_n = M_0, X_1, X_2, \ldots, X_n$

$$ $\therefore \mathrm{M_n} = \mathrm{E}[M_{\mathrm{n+1}} \mid \mathrm{H_n}$

8.2.2 Important Examples

8.2.2.1 Polya's Urn

Frame: urn containing red (R) and black (B) balls

- Init: 1R, 1B
- Step: Ball randomly drawn. Put back, +1 ball of same color added (e.g. 1R, 1B -> draw 1R -> 2R, 1B)

Define X_n as number of black balls after n draws

Fraction of black balls is Martingale (easily provable):

$$M_n = \frac{X_n}{n+2}$$

- $n+2$ because after n draws, the urn contains $n+2$ balls (draws + init)

$$X_{n+1} = \begin{cases} X_n + 1 & \text{with probability } M_n \\ X_n & \text{with probability } 1 - M_n \end{cases}$$

8.2.2.2 Branching Processes

View it as the random construction of a tree.

At each time step, the latest (leaf) generation reproducing, with each node birthing a random number of child nodes.

Define the number of child nodes from generation n's ith node as Z_{ni}

- $n \geq 0, i \geq 1$. Note that time and generation indexes start at 0, but child index starts at 1
- At timestep n, n is also the number of the current generation
- Z_{ni} are iid, positive and integer-valued

Define $m = \mathbb{E}[Z_{ni}]$

Define X_n as size of generation n

$$X_{n+1} = \begin{cases} \sum_{i=1}^{X_n} Z_{ni} & \text{if } X_n > 0 \\ 0 & \text{otherwise} \end{cases}$$

8.2.2.2.1 Martingale

It's the ratio of the size of generation n (X_n) over its expected size (m^n, which is the expected tree starting from $X_0 = 1$)

$$M_n = \frac{X_n}{m^n}$$

8.2.3 How to Prove that a Random Variable is Martingale

Just demonstrate that

$$E\left[M_{n+1}|\mathcal{H}_n\right] = M_n$$

8.2.4 Supermartingales and Submartingales

For annoying historical reasons, supermartingale is decreasing while submartingale is increasing.

For m > n,

$$\begin{aligned} \text{Martingale: } \mathbb{E}[M_n] &= \mathbb{E}[M_m] && \text{(constant)} \\ \text{Supermartingale: } \mathbb{E}[M_n] &\geq \mathbb{E}[M_m] && \text{(decreasing)} \\ \text{Submartingale: } \mathbb{E}[M_n] &\leq \mathbb{E}[M_m] && \text{(increasing)} \end{aligned}$$

8.2.5 Doubling Strategy (aka Martingale Betting Strategy)

When winning, bet $1. When losing, double the bet.

E.g. A coin-flip game where you win double the initial bet if you win, and nothing when you lose.

Outcome	T	T	T	T	H
Bet	1	2	4	8	16
Profit	-1	-3	-7	-15	1

The idea is that this strategy will always end up with a profit (although only $1) as long as you stop playing on a heads.

8.2.5.1 The Problem

- You win in the longrun, so this really only works if you have enough money to place for the longrun
- Winning amount doesn't seem like a lot (you win 1 unit only)

9 L9,10

9.1 Beating the System

For martingales and supermartingales, can you convert the expectation for any step $E[M_n]$ to be > 0?

(spoiler: no, proved by optional stopping theorem)

9.2 Martingale Transforms

Predictable process: $\Theta = (\Theta_n : n \geq 1)$ is a predictable process Θ_n can be determined from the history $\mathcal{H}_{n-1}$, $\forall n \geq 1$

A predictable process is also defined by Karthyek as an *admissible gambling strategy*.

Martingale transform is a scaling of the bet at each $\mathrm{M_k}$.

Winnings from game k = \$ $\sum_{\mathrm{k=1}}^{\mathrm{n}} \Theta_{\mathrm{k}}(M_{\mathrm{k}}$-$\mathrm{M_{k-1}}$\$

Note that $M_n := M_0 + \sum_{i=1}^{n} X_i$, NOT $M_n := M_0 + \sum_{i=1}^{n} \Theta_i X_i$

For convenience, denote the martingale transform of M by Θ with:

$$(\Theta \bullet M)_n = \begin{cases} \sum_{k=1}^{n} \Theta_k \left(M_k - M_{k-1}\right) & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases}$$

9.2.1 Theorem

$\Theta \bullet M$ is:

- Martingale if M is martingale
- Supermartingale if M is Supermartingale
- Submartingale if M is Submartingale

9.2.2 Martingale Doubling Strategy Example

$$\Theta_n = \begin{cases} 2\Theta_{n-1} & \text{if } X_{n-1} = -1 \\ 1 & \text{if } X_{n-1} = 1 \end{cases}$$

Assuming a binary game of outcome { -1, +1 }

9.3 TODO Stopping Time Based Strategies

The idea here is to give us the mathematical vocabulary to talk about strategies such as "I'll stop playing as soon as I hit \$1k"

Denoting the stopping time (a random variable btw) by T, we introduce the following notation:

$$M_n^T = M_{T \wedge n} = M_0 + \sum_{k=1}^{n} \mathbf{1}_{\{k \leq T\}} \left(M_k - M_{k-1}\right)$$

Here, we use Θ_k as an on/off switch. It's 1 when $k \leq T$ and 0 after T

9.4 Optional Stopping Theorem

9.4.1 Used For

- Disproving seemingly sure-win betting strategies
- Ensuring that arbitrage is not possible

9.4.2 Sufficient Conditions (each is sufficient on its own)

- Time is bounded ($\exists$ constant K$ such that $T \leq K$)
- Total money of gambler at each point is bounded ($\exists$ constant K such that $|M_{T \wedge n}| \leq K$)
- Expected time is finite ($\mathbb{E}[T] < \infty$) AND winnings at each stage are bounded ($\exists$ constant K such that $|M_n - M_{n-1}| \leq K$ for $n \leq T$)

9.4.3 Theorem

If any of the three conditions above hold (each is on its own sufficient), then:

- $\mathbb{E}[M_T] = \mathbb{E}[M_0]$ if M is martingale
- $\mathbb{E}[M_T] \leq \mathbb{E}[M_0]$ if M is supermartingale
- $\mathbb{E}[M_T] \geq \mathbb{E}[M_0]$ if M is submartingale

9.5 OST Application to Gambler's Ruin

9.5.1 Gambler's Ruin

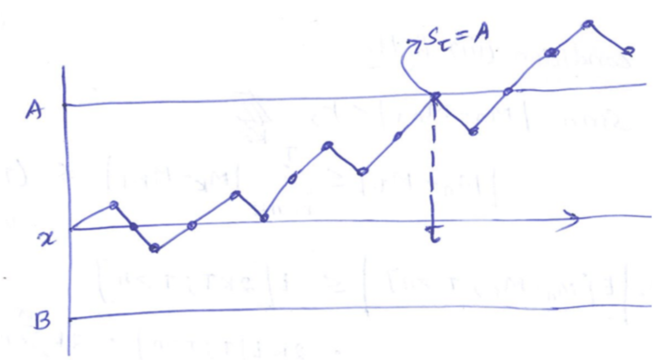
There are a few related definitions that each change what Gambler's Ruin means. The core idea is that in the scenario below, a gambler always goes broke (gets ruined).

Scenari

- Gambler is persistent
- Gambler has finite wealth
- Game is fair
- Opponent has infinite wealth

9.5.2 Our Setup

In our version of Gambler's Ruin, we consider a gambler with a stopping time strategy, and define his losing all his money as ruin. The game is binary, with outcome { -1, +1 }



- Wins if he hits A
- Ruined if he hits B

We denote the wealth of the gambler at time n by:

$$S_n = S_0 + \sum_{i=1}^{n} X_i$$

And denote his starting wealth

$$S_0 = x$$

We denote the probability of winning/losing arbitrary game i by

$$P(X_i = 1) = p \quad \text{and} \quad P(X_i = -1) = q$$

Allow the stopping time to be τ

9.5.3 Symmetric Case $p = q$, with retrieval of $\mathbb{E}[\tau]$

Fortunately, in this case, S and $S^2 - n$ are martingales.

9.5.3.1 Retrieving $P(S_\tau = A)$

$$P(S_\tau = A) = \frac{x - B}{A - B}$$

Shown below. Note: we use OST's result that $\mathbb{E}[S_\tau] = \mathbb{E}[S_0]$ since the problem is bounded.

$$\begin{aligned}
E[S_\tau] &= AP(S_\tau = A) + BP(S_\tau = B) \\
&= E[S_0] \\
x &= AP(S_\tau = A) + B(1 - P(S_\tau = A)) \\
&= (A - B)P(S_\tau = A) + B \\
\therefore P(S_\tau = A) &= \frac{x - B}{A - B}
\end{aligned}$$

9.5.3.2 Retrieving $\mathbb{E}[\tau]$ There are some errors with Karthyek's derivation. However, the general idea is to use $M_n := S_n^2 - n$. My derivation is listed below, and I've emailed him to verify

By OST,

$$
\begin{aligned}
\mathbb{E}[M_0] &= \mathbb{E}[M_\tau] \\
x &= \mathbb{E}[S_\tau^2] - \mathbb{E}[\tau] \\
&= A^2 P(S_\tau = A) + B^2 P(S_\tau = B) - \mathbb{E}[\tau] \\
&= A^2 P(S_\tau = A) + B^2 (1 - P(S_\tau = A)) - \mathbb{E}[\tau] \\
&= (A^2 - B^2) P(S_\tau = A) + B^2 - P(S_\tau = A)) - \mathbb{E}[\tau] \\
&= (A^2 - B^2)(\frac{x-B}{A-B}) + B^2 - \mathbb{E}[\tau] \\
\mathbb{E}[\tau] &= (A^2 - B^2)\left(\frac{x-B}{A-B}\right) + B^2 - x \\
&= (A^2 - B^2)\left(\frac{x-B}{A-B}\right) + B^2 - x \\
&= (A+B)(A-B)\left(\frac{x-B}{A-B}\right) + B^2 - x \\
&= (A+B)(x-B) + B^2 - x \\
&= Ax - AB + Bx - x
\end{aligned}
$$

Note: by symmetry using the x=0 case, Karthyek's shown that infact $\mathbb{E}[\tau] = -(A-x)(B-x)$. Idk how to reconcile it with the math above. ???

9.5.4 General Case, including $p \neq q$, without retrieval of $\mathbb{E}[\tau]$

9.5.4.1 Retrieving $P(S_\tau = A)$ Unfortunately, in this case S and $S^2 - n$ may not be a martingales (they only are when $p = q$

However, $q/p)^{S_n}$ is a martingale, so we define $M_n = q/p)^{S_n}$

$$
P\left(S_\tau = A\right) = \frac{(q/p)^x - (q/p)^B}{(q/p)^A - (q/p)^B}
$$

Shown below. Note: we use OST's result that $\mathbb{E}[S_\tau] = \mathbb{E}[S_0]$ since the problem is bounded.

Also note: $\mathbb{E}[M_0] = \left(\frac{q}{p}\right)^x$

$$
\begin{aligned}
&E\left[M_\tau\right] = E[M_0] \\
&\left(\frac{q}{p}\right)^x = \left(\frac{q}{p}\right)^A P\left(S_\tau = A\right) + \left(\frac{q}{p}\right)^B P\left(S_\tau = B\right) \\
&\qquad = \left(\frac{q}{p}\right)^A P\left(S_\tau = A\right) + \left(\frac{q}{p}\right)^B \left(1 - P\left(S_\tau = A\right)\right) \\
&\qquad = \left(\left(\frac{q}{p}\right)^A - \left(\frac{q}{p}\right)^B\right) P\left(S_\tau = A\right) + \left(\frac{q}{p}\right)^B
\end{aligned}
$$

10 Background to Know

- Random variables
- Joint distributions

- Markov chains
- Poisson processess