Little's Result

For a time period $[0,t)$, let $\alpha(t)$ be the number of arrivals $\delta(t)$ be the number of departures, $N(t)=\alpha(t)-\delta(t)$ the number in the system at t and $\gamma(t)$ be the difference in area between $\alpha(t)$ and $\delta(t)$. Now $\lambda(t)=\frac{\alpha(t)}{t}$ is the average arrival rate, $T(t)=\frac{\gamma(t)}{\alpha(t)}$ the average system time per customer and $N(t)=\frac{\gamma(t)}{t}$ the average number of customers in the system. Now $N(t)\equiv\lambda(t)T(t)$. If $\lambda=\lim_{t\to\infty}\lambda(t)$ and $T=\lim_{t\to\infty}T(t)$ then $N=\lim_{t\to\infty}N(t)=\lambda T$.

Probability

The *nth central moment* is $\mathbb{E}((X-\mathbb{E}(X))^n)=\int_{-\infty}^{\infty}(x-\mathbb{E}(X))^n f_X(x)dx$.

The *coefficient of variation* is $C_x=\frac{\sigma_x}{\mathbb{E}(X)}$.

The *central limit theorem* states that for a sequence $X_i$of independent, identically distributed random variables with mean μ and variance σ^2, $\lim_{n\to\infty}\mathbb{P}\left(\frac{\sum(X_i-\mu)}{\sqrt{n}\sigma}<x\right)=\Phi(x)$

Statistics

The *sample mean* is $\bar{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$.

The *mean squared error* is $\mathbb{E}((\bar{X}-\mu)^2)=\frac{\sigma^2}{n}$.

The *sample variance* is $S^2=\frac{\sum_{i=1}^{n}(X_i-\bar{X})^2}{n-1}$.

If we write z_α for the value such that $\mathbb{P}(Z>z_\alpha)=\alpha$ where Z is distributed $N(0,1)$. It follows that $\mathbb{P}(-z_{\alpha/2}<Z<z_{\alpha/2})=1-\alpha$ so by the CLT $\mathbb{P}\left(\bar{X}-z_{\alpha/2}\frac{S}{n}<\mu<\bar{X}+z_{\alpha/2}\frac{S}{\sqrt{n}}\right)\approx 1-\alpha$ i.e. $\bar{X}\pm\frac{z_{\alpha/2}S}{\sqrt{n}}$ is a $100(1-\alpha)$ percent confidence interval for μ.

If the common distribution of the X_i are themselves $N(0,1)$ then $\sqrt{n}\frac{(\bar{X}-\mu)}{S}$ has Student's t-distribution with $n-1$ degrees of freedom. Unlike confidence intervals this does not require n to be large.

If we have n independent random variables Y_i and we wish to test that for some p_i, $\mathbb{P}(Y_j=i)=p_i$ for $N_i$being the number of Y_j equal to i we would expect that $\mathbb{E}(N_i)=np_i$ (since $N_i\sim Binom(n,p_i)$) and we would reject the null hypothesis when $t=\sum_{i=1}^{k}\frac{(N_i-np_i)^2}{np_i}$, a normalized distance of N from expectation, is too large. For $T\sim\chi^2(k-1)$, we reject the hypothesis when $\mathbb{P}(T>t)<0.05$.

The Kolmogorov-Smirnov test lets us see whether n independent random variables Y_i arise from a common continuous distribution $F(x)$. We construct $F_e(x)=\frac{\text{No of } i \text{ such that } Y_i\leq x}{n}$ and expect $D=max_x|F_e(x)-F(x)|$ to be small.

For iid X_1 and X_2, we obtain a reduced variance for $\frac{X_1+X_2}{2}$ when $Cov(X_1,X_2)<0$. Such variables are called *antithetic*.

If we wish to estimate $\mu_X=\mathbb{E}(X)$ but know $\mu_Y=\mathbb{E}(Y)$ from the same output, $Z=X+c(Y-\mu_Y)$ is also an estimator for μ_X and for $c*=-\frac{Cov(X,Y)}{Var(Y)}$ $Var(Z)=Var(X)-\frac{Cov(X,Y)^2}{Var(Y)}\leq Var(X)$. Such a Y is called a *control variate*.

Distributions

For $Exp(\lambda)$, $f_X(x)=\begin{cases}\lambda e^{-\lambda x} & x>0\\ 0 & x\leq 0\end{cases}$, $\mu_X=\frac{1}{\lambda}$, $\sigma_X^2=\frac{1}{\lambda^2}$. This is the only continuous distribution with the *memoryless property*, $\mathbb{P}(X>t+s|X>t)=P(X>s)$.

For $\Gamma(n,\lambda)$, $f_X(x)=\begin{cases}\lambda e^{-\lambda x\frac{(\lambda x)^{n-1}}{(n-1)!}} & x>0\\ 0 & x\leq 0\end{cases}$, $\mu_X=\frac{n}{\lambda}$,$\sigma_X^2=\frac{n}{\lambda^2}$. The sum of n independent $Exp(\lambda)$ variables has a $\Gamma(n,\lambda)$ distribution.

For $N(\mu,\sigma^2)$, $f_X(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

For $Binom(n,p)$, $\mathbb{P}(X=x)=\binom{n}{x}p^x(1-p)^{n-x}$ for $0\leq x\leq n$, $\mu_X=np$, $\sigma^2=np(1-p)$.

For $Poisson(\lambda)$, $\mathbb{P}(X=i)=e^{-\lambda}\frac{\lambda^i}{i!}$ for $i\geq 0$, $\mu_x=\sigma_X^2=\lambda$. This is a good approximation to the binomial distribution for large n and small p where $\lambda=np$.

For $Geometric(p)$, $\mathbb{P}(X=n)=p(1-p)^{n-1}$ for $n\geq 1$, $\mu_X=\frac{1}{p}$, $\sigma_X^2=\frac{1-p}{p^2}$. In this definition the number of trials includes the first successful trial.

Poisson Process

We say $g(h)=o(h)$ if $\lim_{h\to 0}\frac{g(h)}{h}=0$.

Let $N(t)$ be the number of events that occur in the interval $[0,t]$. For a Poisson process with rate λ $N(0)=0$, the number of events in disjoint time intervals are independent and the distribution of the number of events in an interval depends only on its length and $P(N(h)=i)=\begin{cases}1-\lambda h+o(h) & i=0\\ \lambda h+o(h) & i=1\\ o(h) & i\geq 2\end{cases}$.

If we consider dividing the interval of length t into n intervals of length $h=\frac{n}{t}$. A sub-interval contains a single event with probability approximately $\lambda\frac{t}{n}$ so the number of such sub-intervals is $Binom(n,\lambda\frac{t}{n})$. Letting $n\to\infty$ shows that $\mathbb{P}(N(t)=i)=\mathbb{P}(Poisson(\lambda t)=i)$.

Consider a sequence X_i of inter-arrival times between events in a Poisson process of rate λ. Since $\mathbb{P}(X_1>t)=\mathbb{P}(N(t)=0)=e^{-\lambda t}$ so X_1 has an $Exp(\lambda)$ distribution. Now:

$$\begin{aligned}\mathbb{P}(X_{i+1}>t|X_i=s) &= \mathbb{P}(0\text{ events in }(s,s+t]|X_i=s)\\ &= \mathbb{P}(0\text{ events in }(s,s+t])\\ &= e^{-\lambda t}\end{aligned}$$

So inductively inter-arrival times are distributed $Exp(\lambda)$. Furthermore $S_n = \sum_{j=1}^{n} X_i$ has a $\Gamma(n, \lambda)$ distribution.

Simulation Techniques

Multiplicative congruential: $X_n = (aX_{n-1}) \, mod \, m$ for $a \in \mathbb{N}$, $m \in \mathbb{N}$. Choose the numbers to maximise the period of X_i.

Mixed congruential/linear congruential: $X_n = (aX_{n-1} + c) \, mod \, m$ with a, m as above.

Discrete inverse transform: given U distributed $U(0,1)$ and target distribution x_i, $\mathbb{P}(X = x_i) = \mathbb{P}\left(\sum_{j=0}^{i-1} p_j \le U < \sum_{j=0}^{i} p_j\right) = p_i$.

Geometric inverse transform: since $\sum_{j=1}^{i-1} p_j = 1 - \mathbb{P}(X > i - 1) = 1 - (1-p)^{i-1}$, $X = \left\lfloor \frac{log(U)}{log(1-p)} \right\rfloor + 1$.

Poisson inverse transform: since $p_{i+1} = \frac{\lambda}{i+1} p_i$, $p_0 = e^{-\lambda}$ can iteratively find an i which satisfies the criteria.

Continuous inverse transform: given U distributed $U(0,1)$ and target distribution X, $X = F_X^{-1}(U)$ and $U = F_X(X)$ since:

$$\begin{aligned} \mathbb{P}(X \le x) &= \mathbb{P}(F_X^{-1}(U) \le x) \\ &= \mathbb{P}(F_X(F_X^{-1}(U)) \le F_X(x)) \\ &= \mathbb{P}(U \le F_X(x)) \\ &= F_X(x) \end{aligned}$$

Uniform inverse transform: $X = (x-a)U + a$.

Exponential inverse transform: $X = -\frac{1}{\lambda} log(U)$.

Poisson inverse transform alternative: $X = argmin_n(\prod_{i=1}^{n} U_i < e^{-\lambda}) - 1$.

Queueing Systems

Kendall notation: $A/B/m/k/l$ where A is an inter-arrival time distribution, B is a service time distribution, m is the number of parallel servers, k is the limit of customers in the system (NOT the queues) and l is the population size.

Queuing networks are queues connected together. They can be *closed* (fixed set of jobs circulate) and *open* (jobs may enter and leave). Open networks are *feed-forward* if they visit each server at most once.

Can have discrete state/time or continuous state/time simulations. Simulations are very general but they can be time consuming to design, code and debug, may be complex and obscure understanding, could be computationally intensive and can hinder statistical analysis of the output (e.g. how long should we run the simulation before averaging?). Events are time-labelled. The simulator picks the event with the lowest available timestamp to execute next. We can attempt to reduce error by running the simulation for longer and by running the same simulation with a number of different pseudo-random number sequences.

Utilization is the proportion of time that a server is busy.

Queue length can be obtained by estimating the queue length distribution and finding the mean and by viewing queue length as a function of time and finding its average.

Queueing time can be obtained by using Little's law or by finding the average observed queue times.

Multiple simulation runs are called *replications*. Each replication requires re-stabilising the simulation so we can break a simulation run up into large blocks (with low inter-block correlation) to obtain more samples. We can do subsequent runs with the antithetic variables U and $1 - U$ to reduce variance.

Stochastic processes are collections of random variables $X(t)$ that take values in a state space S indexed by times T. A *sample path* is an observed set of values $X(t)$. Processes may be *discrete-state* (when S is countable) or *discrete-time* (when T is countable).

Markov processes are stochastic processes that obey the Markov property $\mathbb{P}(X(t) \in A_{n+1} | \forall i \le n. X(t_i) \in A_i) = \mathbb{P}(X(t) \in A_{n+1} | X(t_n) \in A_n)$.

Birth-death processes are Markov processes in which transitions are only allowed between neighbouring states. Typically, if $X_n = i$ then $X_{n+1} = \begin{cases} i+1 & \text{birth} \\ i-1 & \text{death} \end{cases}$. The birth and deaths rate in i respectively are λ_i and μ_i. The Chapman-Kolmogorov equations capture their behaviour:

$$\frac{dP_i(t)}{dt} = \begin{cases} -(\lambda_i + \mu_i)P_i(t) + \mu_{i+1}P_{i+1}(t) + \lambda_{i-1}P_{i-1}(t) & i \ne 0 \\ -\lambda_0 P_0(t) + \mu_1 P_1(t) & i = 0 \end{cases}$$

Systems reach equilibrium iff $\forall i. \lim_{t \to \infty} P_i(t) = p_i$ exists. The stationary solution occurs when $\frac{dP_i(t)}{dt} = 0$. The *global balance* equation states that for $i \ge 1$, $p_{i-1}\lambda_{i-1} + p_{i+1}\mu_{i+1} = p_i\lambda_i + p_i\mu_i$. The *detailed balance equations* state that for $i \ge 0$, $p_i\lambda_i = p_{i+1}\mu_{i+1}$ and for $i \ge 1$, $p_i\mu_i = p_{i-1}\lambda_{i-1}$. Hence for $k \ge 1$ $p_k = p_i \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}$.

For the $M/M/1$ queue we can derive that if $\rho = \frac{\lambda}{\mu}$, $p_k = (1 - \rho)\rho^k$. Hence $\mathbb{E}(N) = \frac{\rho}{1-\rho}$ and $\mathbb{E}(T) = \frac{1}{\mu - \lambda}$.

For the $M/M/m$ queue we have that where k is the queue length, $\mu_k = \begin{cases} k\mu & 0 \le k \le m \\ m\mu & k > m \end{cases}$. For an equilibrium we require $\frac{\lambda}{m\mu} < 1$.

For the $M/M/1/K$ queue we have $p_0 = \frac{1-\rho}{1-\rho^{K+1}}$ and for $k \le K$ $p_k = p_0\rho^k$.

For the $M/M/1//N$ queue we have for $0 \leq k \leq N$, $\lambda_K = (N-k)\lambda$.

For the $M/M/m/m$ queue (aka the m server loss system) we have $p_m = \left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!} \left(\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}\right)^{-1}$ which is known as *Erlang's formula.*

We can have multistage birth processes, where r stages each have exponentially distributed residence times with rate $r\lambda$: the average time through the stages is $\frac{1}{\lambda}$ and the coefficient of variation is $\frac{1}{\sqrt{r}}$.

For the $M/G/1$ queue service times are given by $B(x) = \mathbb{P}(\text{service time} \leq x)$. It does not have the Markov property, but it is possible to pick out a discrete set of times where the Markov property holds, such as the times t_i where departures occur. The mean queuing time before receiving service is $\mathbb{E}(T_q) = \mathbb{E}(N_q)\frac{1}{\mu} + \rho\mathbb{E}(R)$ where $\mathbb{E}(N_q)$ is the average number of customers enqueued and $\mathbb{E}(R)$ is the average remaining service time of the customer in service upon arrival. Renewal theory shows that $\mathbb{E}(R) = \frac{\mu\mathbb{E}(S^2)}{2}$ where S is the service time distribution. By Little's law, $\mathbb{E}(N_q) = \lambda\mathbb{E}(T_q)$ so since $C_S^2 = \frac{\mathbb{E}(S^2)}{\mathbb{E}(S)^2} - 1$ for the distribution where $\mathbb{E}(S) = \frac{1}{\mu}$ it is true that $\mathbb{E}(T_q) = \frac{\rho(1+C_S^2)}{2\mu(1-\rho)}$.

Open Queueing Networks

We consider a class of open queuing networks called *Jackson networks.* Customers arrive as a Poisson stream at server i with rate γ_i. Each of the N servers has service time distributed $Exp(\mu_i)$. Customers completing at node i move to node j with probability q_{ij}. A job leaves the network with probability $q_{i0} = 1 - \sum_{j=1}^{N} q_{ij}$: $Q = (q_{ij})$ is called the routing matrix. The system state is $(k_1, k_2, \ldots, k_N)$ where k_i is the number of jobs at node i.

The *traffic equations* are $\lambda_i = \gamma_i + \sum_{j=1}^{N} \lambda_j q_{ji}$. An equilibrium distribution exists iff $\rho_i = \frac{\lambda_i}{\mu_i} < 1$. The distribution is $p(k_1, \ldots k_N) = \prod_{i=1}^{N} p_i(k_i)$ where $p_i(k_i)$ is the equilibrium distribution for when there are k_i jobs in an $M/M/1$ queue with traffic intensity ρ_i.

Closed Queuing Networks

We can also consider closed queuing networks of this form with a constant K jobs in the system. There are $\binom{K+N-1}{N-1}$ states in such a system: consider $K+N-1$ boxes aligned in a row and select $N-1$ of those boxes (which can be done in $\binom{K+N-1}{N-1}$ ways). Place a "/" symbol in each of the boxes and a "1" in all others. The boxes now represent an ordered partition of K into N groups of "1" which are added together to give the k_i summands.

The traffic equations become $\lambda_i = \sum_{j=1}^{N} \lambda_j q_{ji}$. Analogously, $p(k_1, \ldots k_N) = \frac{1}{G} \prod_{i=1}^{N} r_i(k_i)$ where $G = \sum_{(k_1, \ldots, k_N) \in S} \prod_{i=1}^{N} r_i(k_i)$.

The Pollaczek-Khintchine formula applies to queuing systems with a service time distribution $\mathbb{E}(S) = \frac{1}{\mu}$ (i.e. containing $M/G/1$ queues). Since $\mathbb{E}(T) = \mathbb{E}(T_q) + \frac{1}{\mu}$, $\mathbb{E}(N) = \rho + \frac{\rho^2(1+C_S^2)}{2(1-\rho)}$.

Building Models

Load testing is accurate but time consuming and expensive, and a system must exist. Performance modelling is quick and cheap, and can be applied at the design stage, but its accuracy depends entirely on model representativeness.

Capacity planning methodology:

1. Characterise IT infrastructure to get infrastructure model
2. Characterise and test workload to get workload model
   (a) Identify request classes
   (b) Identify resources used by each class
   (c) Measure service demand for each request class at each resource
   (d) Specify number of requests of each class that the system will be exposed to
3. Model, validate and calibrate performance to obtain a performance model and prediction
   (a) Typically model resources and clients using queues, nets, delay resources etc

Queuing networks are powerful for modelling contention and scheduling strategies. Many efficient analysis techniques are available. However, they are not suitable for modelling blocking, synchronization, simultaneous resource possession or software contention.

Petri nets are suitable for qualitative and quantitative analysis. They lend themselves to modelling blocking, synchronization, simultaneous resource possession and software contention. However, there are no direct means for modelling scheduling strategies and fewer algorithms and tools available for analysing them.

Queuing Petri nets can model everything mentioned and allow the integration of hardware and software aspects. However, analysis suffers from state-space explosion!