Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Introduction

Module Overview

Last Module: We studied ways to generate Uniform(0,1) pseudo-random numbers.   Who cares…?

This Module: We'll take those PRN's and use them to generate *everything* else! You name it, we'll generate it!

Idea: Find the proper trick or algorithm and off we go! This is how we drive simulations.

Module Overview

1. Introduction ← This lesson
2. Inverse Transform Method
3. Continuous Examples
4. Discrete Examples
5. Empirical Distribution Example
6. Convolution Method
7. Acceptance-Rejection Method
8. Proof
9. Continuous Examples
10. Poisson Example

Module Overview, II

11. Composition Method
12. Box-Muller Normals
13. Order Statistics and Other Stuff
14. Multivariate Normal Distribution
15. Baby Stochastic Processes
16. Nonhomogeneous Poisson
17. Time Series
18. Queueing Processes
19. Brownian Motion

Introduction

Goal: Use $\mathcal{U}(0,1)$ numbers to generate observations (variates) from other distributions, and even stochastic processes.

Try to be fast, reproducible.

- Discrete distributions, like Bernoulli, Binomial, Poisson, and empirical
- Continuous distributions like exponential, normal (many ways), and empirical

Intro (cont'd)

- Multivariate normal
- Nonhomogeneous Poisson process
- Autoregressive moving average time series
- Waiting times
- Brownian motion

Let's start with an old friend…

Inverse Transform Theorem: Let X be a continuous random variable with c.d.f. $F(x)$. Then $F(X) \sim \mathcal{U}(0, 1)$.

Summary

This Time: Discussed what's coming up in this module on random variate (and random process) generation.

Next Time: We'll look at the Inverse Transform method one last time!

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Inverse Transform Method

Lesson Overview

Last Lesson: We introduced the topic of RV generation.

This Lesson: Now it's time to get going and start putting together our bag of tricks.

Idea: Let's go into some additional detail with Inverse Transform…

One Last Time!

Inverse Transform Theorem: Let X be a continuous random variable with c.d.f. $F(x)$. Then $F(X) \sim \mathcal{U}(0,1)$.

Proof: Let $Y = F(X)$ and suppose that Y has c.d.f. $G(y)$. Then

$$
\begin{aligned}
G(y) &= \mathrm{P}(Y \le y) = \mathrm{P}(F(X) \le y) \\
&= \mathrm{P}(X \le F^{-1}(y)) = F(F^{-1}(y)) = y. \quad \Box
\end{aligned}
$$

How Do We Use This Result?

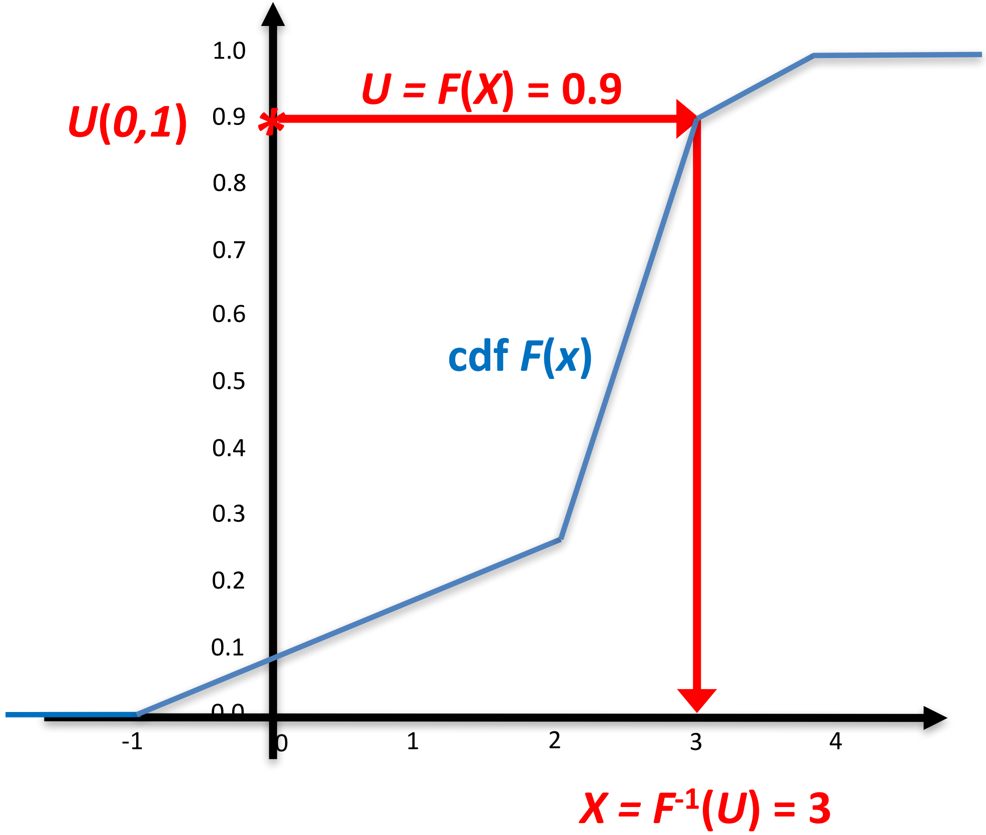
Let $U \sim \mathcal{U}(0,1)$. Then $F(X) = U$ means that the random variable $F^{-1}(U)$ has the same distribution as X.

So here is the *inverse transform method* for generating a RV X having c.d.f. $F(x)$:

1. Sample U from $\mathcal{U}(0,1)$.
2. Return $X = F^{-1}(U)$.

Inverse Transform Method

(generate *X* from *U*)



Example: The $\mathcal{U}(a, b)$ distribution, with $F(x) = \frac{x-a}{b-a}$, $a \leq x \leq b$.

Solving $(X - a)/(b - a) = U$ for X, we get $X = a + (b - a)U$. $\square$

Example: The Exp(λ) distribution, with $F(x) = 1 - e^{-\lambda x}$, $x > 0$.

Solving $F(X) = U$ for X,

$$X = -\frac{1}{\lambda}\ell\text{n}(1 - U) \quad \text{or} \quad X = -\frac{1}{\lambda}\ell\text{n}(U). \quad \square$$

Summary

This Time: Stated and finally *proved* the Inverse Transform Theorem. Then showed how to use it on a couple of easy examples.

Next Time: We'll do some more-interesting applications of Inverse Transform on trickier continuous examples.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Inverse Transform Method: Continuous Examples

Lesson Overview

Last Lesson: Stated and finally *proved* the Inverse Transform Theorem. Then showed how to use it on some trivial examples.

This Lesson: We'll apply the method on trickier continuous examples.

The method almost always works (well, sort of).

More Continuous Examples

Example: The Weibull distribution, $F(x) = 1 - e^{-(\lambda x)^\beta}, x > 0$.

Solving $F(X) = U$ for X,

$$X = \frac{1}{\lambda}[-\ell\mathrm{n}(1-U)]^{1/\beta} \quad \text{or} \quad X = \frac{1}{\lambda}[-\ell\mathrm{n}(U)]^{1/\beta}. \quad \square$$

Example: The triangular (0,1,2) distribution has p.d.f.

$$f(x) = \begin{cases} x & \text{if } 0 \le x < 1 \\ 2 - x & \text{if } 1 \le x \le 2. \end{cases}$$

The c.d.f. is

$$F(x) = \begin{cases} x^2/2 & \text{if } 0 \le x < 1 \\ 1 - (x-2)^2/2 & \text{if } 1 \le x \le 2. \end{cases}$$

Need to look at two cases separately!

If $U < 1/2$, we solve $X^2/2 = U$ to get $X = \sqrt{2U}$.

If $U \geq 1/2$, the only root of $1 - (X-2)^2/2 = U$ in $[1, 2]$ is

$$X = 2 - \sqrt{2(1-U)}.$$

Thus, for example, if $U = 0.4$, we take $X = \sqrt{0.8}$. $\square$

Remark: Do not replace U by $1 - U$ here!

Demo Time!!!

Example: The standard normal distribution. Unfortunately, the inverse c.d.f. $\Phi^{-1}(\cdot)$ does not have an analytical form. *This is often a problem with the inverse transform method.*

Easy solution: Do a table lookup. E.g., If $U = 0.975$, then $Z = \Phi^{-1}(U) = 1.96$. $\square$

NORMSINV(0.975) in Excel

Crude portable approximation (BCNN): The following approximation gives at least one decimal place of accuracy for $0.00134 \le U \le 0.98865$:

$$Z = \Phi^{-1}(U) \approx \frac{U^{0.135} - (1-U)^{0.135}}{0.1975}. \quad \square$$

Here's a better portable solution to generate Nor(0,1)'s: The following approximation has absolute error $\le 0.45 \times 10^{-3}$:

$$Z = \text{sign}(U - 1/2)\left(t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}\right),$$

where $\text{sign}(x) = 1, 0, -1$ if x is positive, zero, or negative, respectively,

$$t = \{-\ell\text{n}[\min(U, 1 - U)]^2\}^{1/2},$$

and

$$c_0 = 2.515517, \quad c_1 = 0.802853, \quad c_2 = 0.010328,$$
$$d_1 = 1.432788, \quad d_2 = 0.189269, \quad d_3 = 0.001308.$$

In any case, if $Z \sim \text{Nor}(0,1)$ and you want $X \sim \text{Nor}(\mu, \sigma^2)$, just take $X \leftarrow \mu + \sigma Z$.

Easy Example (Inverse Transform): Suppose you want to generate $X \sim \text{Nor}(3, 16)$, and you start with $U = 0.59$. Then

$$X = \mu + \sigma Z = 3 + 4\Phi^{-1}(0.59) = 3 + 4(0.2275) = 3.91. \quad \Box$$

Demo Time!!!

Summary

This Time: Discussed how to generate several interesting continuous RVs via the Inverse Transform method.

Next Time: We'll use Inverse Transform in a more-discrete way.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Inverse Transform Method: Discrete Examples

Lesson Overview

Last Lesson: Discussed how to generate several interesting continuous RVs via the Inverse Transform method.

This Lesson: We'll use Inverse Transform for discrete examples.

"What happens in simulation class stays in simulation class."

Discrete Examples

For discrete distributions, it's often best to construct a table.

Baby Discrete Example: The Bernoulli(p) distribution.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0,1)$'s
0	$1 - p$	$1 - p$	$[0,\ 1 - p]$
1	p	1	$(1 - p, 1]$

If $U \leq 1 - p$, then take $X = 0$; otherwise, $X = 1$. For instance, if $p = 0.75$ and we generate $U = 0.13$, we take $X = 0$. $\square$

Alternately, we can construct the following "backwards" table (which isn't strictly inverse transform, but it's the one that I usually use).

x	$P(X = x)$	$\mathcal{U}(0,1)$'s
1	p	$[0,\ p]$
0	$1 - p$	$(p, 1]$

if $U \leq p$, take $X = 1$; otherwise, $X = 0$. $\square$

Example: Suppose we have a slightly less-trivial discrete p.m.f.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0, 1)$'s
-1	0.6	0.6	[0.0,0.6]
2.5	0.3	0.9	(0.6,0.9]
4	0.1	1.0	(0.9,1.0]

Thus, if $U = 0.63$, we take $X = 2.5$. $\square$

Sometimes there's an easy way to avoid constructing a table.

Example: The discrete uniform distribution on $\{1, 2, \ldots, n\}$,

$$P(X = k) \;=\; \frac{1}{n}, \quad 1, 2, \ldots, n.$$

Clearly, $X = \lceil nU \rceil$, where $\lceil \cdot \rceil$ is the ceiling function.

So if $n = 10$ and $U = 0.376$, then $X = \lceil 3.76 \rceil = 4$. $\square$

Example: The geometric distribution with p.m.f. and c.d.f.

$$f(k) \;=\; q^{k-1}p \quad \text{and} \quad F(k) \;=\; 1-q^k,\; k=1,2,\ldots,$$

where $q = 1-p$. Thus, after some algebra,

$$X \;=\; \min[k : 1-q^k \geq U] \;=\; \left\lceil \frac{\ell\mathrm{n}(1-U)}{\ell\mathrm{n}(1-p)} \right\rceil \;\sim\; \left\lceil \frac{\ell\mathrm{n}(U)}{\ell\mathrm{n}(1-p)} \right\rceil.$$

For instance, if $p = 0.3$ and $U = 0.72$, we obtain

$$X \;=\; \left\lceil \frac{\ell\mathrm{n}(0.28)}{\ell\mathrm{n}(0.7)} \right\rceil \;=\; 4. \quad \Box$$

Remark: Can also generate Geom(p) by counting Bern(p) trials until you get a success.

Easy Example: Generate $X \sim$ Geom(1/6). This is the same thing as counting the number of dice tosses until a 3 (or any particular number) comes up, where the Bern(1/6) trials are the i.i.d. dice tosses. For instance, if you toss 6,2,1,4,3, then you stop on the Bernoulli trial $X = 5$, and that's your answer.

But life isn't always dice tosses. A general way to generate a Geom(p) is to count the number of trials until $U_i \leq p$. For example, if $p = 0.3$, then $U_1 = 0.71$, $U_2 = 0.96$, and $U_3 = 0.12$ implies that $X = 3$. $\square$

Remark: If you have a discrete distribution like Pois(λ) with an infinite number of values, you could write out table entries until the c.d.f. is nearly one, generate exactly one U, and then search until you find $X = F^{-1}(U)$, i.e., x_i such that $U \in \big(F(x_{i-1}), F(x_i)\big]$.

x	$P(X = x)$	$F(x)$	$\mathcal{U}(0,1)$'s
x_1	$f(x_1)$	$F(x_1)$	$[0, F(x_1)]$
x_2	$f(x_2)$	$F(x_2)$	$(F(x_1), F(x_2)]$
x_3	$f(x_3)$	$F(x_3)$	$(F(x_2), F(x_3)]$
$\vdots$			

Example: Suppose $X \sim \text{Pois}(2)$, so that $f(x) = \frac{e^{-2}2^x}{x!}$, $x = 0, 1, 2, \ldots$.

x	$f(x)$	$F(x)$	$\mathcal{U}(0,1)$'s
0	0.1353	0.1353	[0, 0.1353]
1	0.2706	0.4059	(0.1353, 0.4059]
2	0.2706	0.6765	(0.4059, 0.6765]
$\vdots$			

For instance, if $U = 0.313$, then $X = 1$. $\square$

Summary

This Time: Generated several discrete RVs by sort of using Inverse Transform.

Next Time: We'll generate RVs from continuous *empirical* (sample) distributions – very useful when we don't know beforehand the exact distribution of a RV.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Inverse Transform Method: Empirical Distributions

Lesson Overview

Last Time: We generated certain discrete RVs.

This Time: What happens when we have some data from an *unknown* continuous distribution?

Almost have to blend continuous and discrete ideas.

Continuous Empirical Distributions

If you can't find a good theoretical distribution to model a certain RV, you may want to use the *empirical c.d.f.* of the data, $X_1, X_2, \ldots, X_n$,

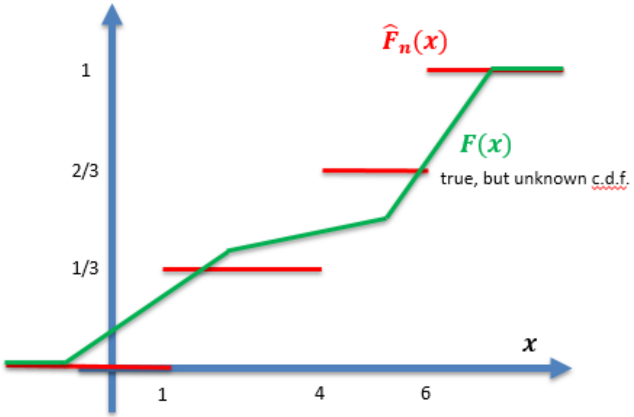
$$\hat{F}_n(x) \equiv \frac{\text{number of } X_i\text{'s} \leq x}{n}.$$

Note that $\hat{F}_n(x)$ is a step function with jumps of height $1/n$ (every time an observation occurs).

Good news: Even though X is continuous, the *Glivenko-Cantelli Lemma* says that $\hat{F}_n(x) \to F(x)$ for all x as $n \to \infty$. So $\hat{F}_n(x)$ is a good approximation to the true c.d.f., $F(x)$.

The ARENA functions `DISC` and `CONT` can be used to generate RV's from the empirical c.d.f.'s of discrete and continuous distributions, respectively.

To do so ourselves, we first define the *ordered* points, $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$. For example, if $X_1 = 4$, $X_2 = 1$, and $X_3 = 6$, then $X_{(1)} = 1$, $X_{(2)} = 4$, and $X_{(3)} = 6$.



Given that you only have a finite number n of data points, we can turn the empirical c.d.f. into a continuous RV by using linear interpolation between the $X_{(i)}$'s.

$$F(x) = \begin{cases} 0 & \text{if } x < X_{(1)} \\ \frac{i-1}{n-1} + \frac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})} & \text{if } X_{(i)} \le x < X_{(i+1)}, \forall i \\ 1 & \text{if } x \ge X_{(n)} \end{cases}$$

dice toss: 1,2,…,*n*-1

1. Set $F(X) = U \sim \mathcal{U}(0,1)$. Let $P = (n-1)U$ and $I = \lceil P \rceil$.
2. Solve to get $X = X_{(I)} + \boxed{(P - I + 1)(X_{(I+1)} - X_{(I)})}$.

random starting pt

turns out U(0,1)

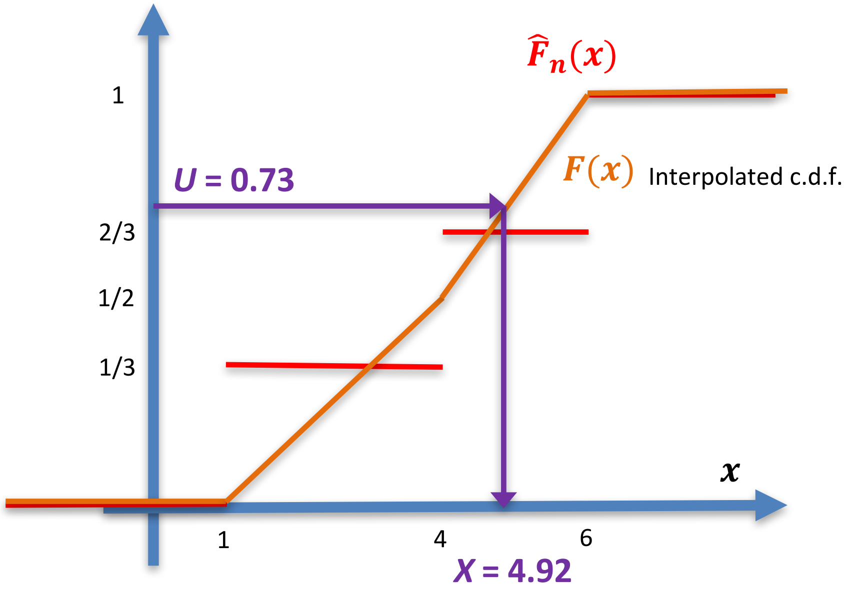
How far towards next pt?

GTx

Example: Suppose $X_{(1)} = 1$, $X_{(2)} = 4$, and $X_{(3)} = 6$. If $U = 0.73$, then $P = (n-1)U = 1.46$ and $I = \lceil P \rceil = 2$. So

$$
\begin{aligned}
X &= X_{(I)} + (P - I + 1)\big(X_{(I+1)} - X_{(I)}\big) \\
&= X_{(2)} + (1.46 - 2 + 1)\big(X_{(3)} - X_{(2)}\big) \\
&= 4 + (0.46)\big(6 - 4\big) \\
&= 4.92. \quad \Box
\end{aligned}
$$

Empirical vs. Interpolated c.d.f.'s



Check (slightly different way):

$$F(x) = \begin{cases} 0 + \frac{x-1}{2(4-1)} & \text{if } 1 \le x < 4 \quad (i = 1 \text{ case}) \\ \frac{1}{2} + \frac{x-4}{2(6-4)} & \text{if } 4 \le x < 6 \quad (i = 2 \text{ case}) \end{cases}$$

Setting $F(X) = U$ and solving for the two cases, we have

$$X = \begin{cases} 1 + 6U & \text{if } U < 1/2 \\ 2 + 4U & \text{if } U \ge 1/2 \end{cases}$$

Then $U = 0.73$ implies $X = 2 + 4(0.73) = 4.92$. $\square$

Summary

This Time: Showed how to generate RVs from a continuous *empirical* distribution.

Next Time: Things are about to get *convoluted* – literally!

Computer Simulation
Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Convolution Method

Lesson Overview

Last Time: Showed how to generate RVs from empirical (sample) distributions.

This Time: We'll discuss the *convolution* method.

Sum-thing's in the air!
And you know that it's right!

www.youtube.com/watch?v=RTZoJ01FpD8

Convolution

Convolution refers to adding things up.

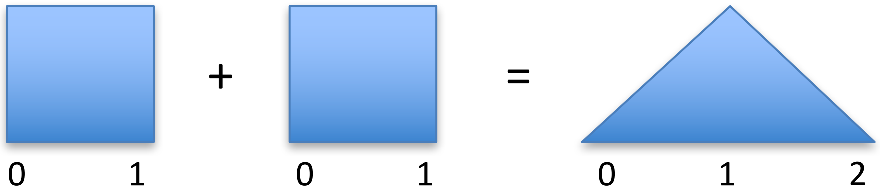
Example: Binomial(n, p). If $X_1, \ldots, X_n \sim$ i.i.d. Bern(p), then $Y = \sum_{i=1}^{n} X_i \sim \text{Bin}(n, p)$.

We already know how to get Bernoulli RVs via Inverse Transform: Suppose $U_1, \ldots, U_n$ are i.i.d. $\mathcal{U}(0,1)$. If $U_i \le p$, set $X_i = 1$; otherwise, set $X_i = 0$. Repeat for $i = 1, \ldots, n$. Add up to get Y.

For instance, if $Y \sim \text{Bin}(3, 0.4)$ and $U_1 = 0.63$, $U_2 = 0.17$, and $U_3 = 0.81$, then $Y = 0 + 1 + 0 = 1$. $\square$

Example: Triangular(0,1,2).

It can be shown that if U_1 and U_2 are i.i.d. $\mathcal{U}(0,1)$, then $U_1 + U_2$ is Tria(0,1,2). (This is easier — but maybe not faster — than our inverse transform method.) $\Box$



Example: $\text{Erlang}_n(\lambda)$. If $X_1, \ldots, X_n \sim$ i.i.d. $\text{Exp}(\lambda)$, then $Y = \sum_{i=1}^n X_i \sim \text{Erlang}_n(\lambda)$. By inverse transform,

$$Y = \sum_{i=1}^n X_i = \sum_{i=1}^n \left[\frac{-1}{\lambda}\ell\text{n}(U_i)\right] = \frac{-1}{\lambda}\ell\text{n}\Big(\prod_{i=1}^n U_i\Big).$$

This only takes one natural log evaluation, so it's pretty efficient. $\square$

Example: A *crude* "desert island" Nor(0,1) approximate generator (which I wouldn't use).

Suppose that $U_1, \ldots, U_n$ are i.i.d. $\mathcal{U}(0,1)$, and let $Y = \sum_{i=1}^{n} U_i$. For large n, the CLT implies that $Y \approx \text{Nor}(n/2, n/12)$.

In particular, let's choose $n = 12$, and assume that it's "large." Then

$$Y - 6 = \sum_{i=1}^{12} U_i - 6 \approx \text{Nor}(0,1). \quad \Box$$

Other convolution-related tidbits:

Did you know…?

If $X_1, \ldots, X_n$ are i.i.d. Geom(p), then $\sum_{i=1}^{n} X_i \sim \text{NegBin}(n, p)$.

If $Z_1, \ldots, Z_n$ are i.i.d. Nor(0,1), then $\sum_{i=1}^{n} Z_i^2 \sim \chi^2(n)$.

If $X_1, \ldots, X_n$ are i.i.d. Cauchy, then $\bar{X} \sim$ Cauchy (this is kind of like getting nowhere fast!).

Demo Time!

Summary

This Time: Used convolutions to generate various RVs. It's a nice trick that we can occasionally apply, and that about ***"sums"*** it up!

Next Time: Acceptance-Rejection. It's the most-useful RV generation technique, but it's a bit tricky at first.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Acceptance-Rejection Method

Lesson Overview

Last Visit: Convolution! Revolution!

www.youtube.com/watch?v=HBU2m8GdxuY

This Visit: Acceptance-Rejection

This is a tough topic, but also very useful, so we'll divide it up into palatable chunks.

Note: The last stage of grief is acceptance.

Acceptance-Rejection Method

Motivation: The majority of c.d.f.'s cannot be inverted efficiently. A-R samples from a distribution that is "almost" the one we want, and then adjusts by "accepting" only a certain proportion of those samples.

Baby Example: Generate a $\mathcal{U}(\frac{2}{3}, 1)$ RV. (You would usually do this via inverse transform, but what the heck!) Here's the A-R algorithm:

1. Generate $U \sim \mathcal{U}(0, 1)$.

2. If $U \geq \frac{2}{3}$, ACCEPT $X \leftarrow U$. O'wise, REJECT and go to Step 1.

Notation: Suppose we want to simulate a continuous RV X with p.d.f. $f(x)$, but that it's difficult to generate directly. Also suppose that we can easily generate a RV having p.d.f. $h(x) \equiv t(x)/c$, where $t(x)$ *majorizes* $f(x)$, i.e.,

$$t(x) \geq f(x), \quad x \in \mathbb{R},$$

and

$$c \equiv \int_{-\infty}^{\infty} t(x)\,dx \geq \int_{-\infty}^{\infty} f(x)\,dx = 1,$$

where we assume that $c < \infty$.

Theorem (von Neumann 1951): Define $g(x) \equiv f(x)/t(x)$ and note that $0 \le g(x) \le 1$ for all x. Let $U \sim \mathcal{U}(0,1)$, and let Y be a RV (independent of U) with p.d.f. $h(y) = t(y)/c$. If $U \le g(Y)$, then Y has (conditional) p.d.f. $f(y)$. ← *Y* has the right p.d.f.!

This suggests the following "acceptance-rejection" algorithm . . .

Algorithm A-R

Repeat

Generate U from $\mathcal{U}(0,1)$

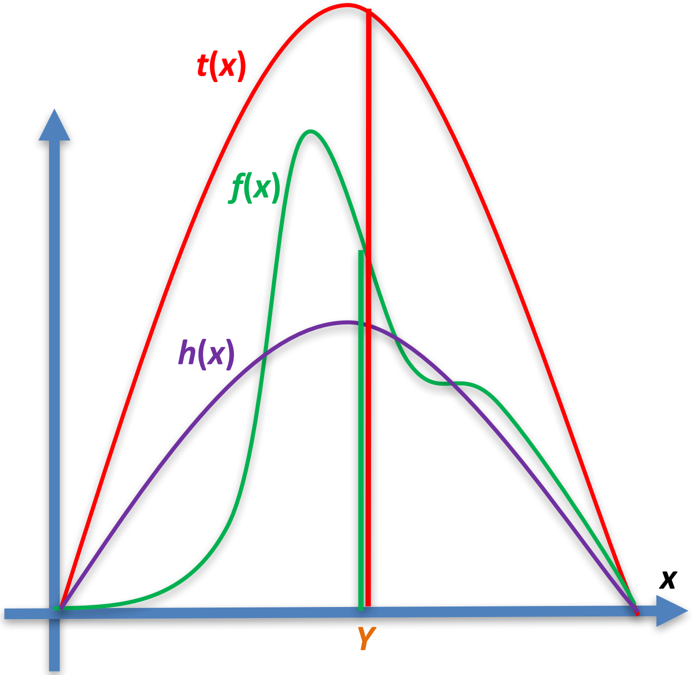
Generate Y from $h(y)$ (independent of U)

Acceptance event

until $U \leq g(Y) = \dfrac{f(Y)}{t(Y)} = \dfrac{f(Y)}{c\,h(Y)}$

Return $X \leftarrow Y$

It really works! Awful proof next lesson! Meanwhile…



Generate a point Y uniformly under $t(x)$ (equivalently, sample Y from p.d.f. $h(x)$).

Accept the point with probability $f(Y) / t(Y) = f(Y) / [c\, h(Y)]$.

If you accept, then set $X = Y$ and stop.

Summary

This Time: We started playing around with the Acceptance-Rejection method. We did a baby example, gave some motivation, and presented the underlying theorem.

Next Time: Proof of the theorem.
Let’s be careful out there… it’s nasty.

www.youtube.com/watch?v=Jmg86CRBBtw

Computer Simulation
Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Proof of the Acceptance-Rejection Method

Lesson Overview

Last Summit: Introduction to Acceptance-Rejection RV generation.

This Summit: Proof that it works.

Things may get a little painful, but you won’t really be expected to reproduce the proof yourself. ☺

Wise saying: “No pain, no pain.”

Proof of A-R Method

Theorem (von Neumann 1951): Define $g(x) \equiv f(x)/t(x)$ and note that $0 \leq g(x) \leq 1$ for all x. Let $U \sim \mathcal{U}(0,1)$, and let Y be a RV (independent of U) with p.d.f. $h(y) = t(y)/c$. If $\underbrace{U \leq g(Y)}_{\text{Acceptance event}}$, then Y has (conditional) p.d.f. $f(y)$.

And then you set
X = *Y* and stop.

Proof that X has p.d.f. $f(x)$.

Let A be the "Acceptance" event. The c.d.f. of X is

$$P(X \leq x) = P(Y \leq x | A) = \frac{P(A, Y \leq x)}{P(A)}. \qquad (1)$$

Then

$$
\begin{aligned}
P(A|Y=y) &= P\left(U \le g(Y)|Y=y\right) \\
&= P\left(U \le g(y)|Y=y\right) \\
&= P\left(U \le g(y)\right) \quad (U \text{ and } Y \text{ are independent}) \\
&= g(y) \quad (U \text{ is uniform}). \qquad (2)
\end{aligned}
$$

Let's keep (1) and (2) in the back of our minds for a wee bit…

By the law of total probability,

$$\begin{aligned}
P(A,\ Y \le x) &= \int_{-\infty}^{\infty} P(A,\ Y \le x | Y = y) h(y)\, dy \\
&= \int_{-\infty}^{x} P(A | Y = y) h(y)\, dy \\
&= \frac{1}{c} \int_{-\infty}^{x} P(A | Y = y) t(y)\, dy \\
&= \frac{1}{c} \int_{-\infty}^{x} g(y) t(y)\, dy \quad \text{(by (2))} \\
&= \frac{1}{c} \int_{-\infty}^{x} f(y)\, dy. \qquad (3)
\end{aligned}$$

Letting $x \to \infty$, we have

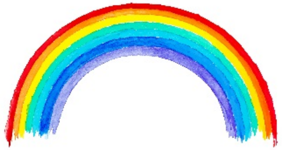
$$P(A) = \frac{1}{c}\int_{-\infty}^{\infty} f(y)\,dy = \frac{1}{c}. \tag{4}$$

Then (1), (3), and (4) imply

$$P(X \le x) = \frac{P(A,\, Y \le x)}{P(A)} = \int_{-\infty}^{x} f(y)\,dy,$$

so that the p.d.f. of X is $f(x)$. $\square$

www.youtube.com/watch?v=ZSnxYtFarNw



There are two main issues:

- The ability to quickly sample from $h(y)$.
- c must be small ($t(x)$ must be "close" to $f(x)$) since

$$P(U \leq g(Y)) = \frac{1}{c}$$

and the number of trials until "success" $[U \leq g(Y)]$ is Geom$(1/c)$, so that the mean number of trials is c.

Summary

This Time: Hearty huzzahs to all! We got through the hardest proof of the course!

Next Time: We'll do some examples to see how all of the magic works!

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

A-R Method: Continuous Examples

Lesson Overview

Last Meeting of the Minds:
Proved the major theorem behind Acceptance-Rejection.

This Meeting of the Minds: Some examples involving continuous distributions.

A-R is a general method that works when others may be difficult to apply.

Theorem (von Neumann 1951): Define $g(x) \equiv f(x)/t(x)$ and note that $0 \leq g(x) \leq 1$ for all x. Let $U \sim \mathcal{U}(0,1)$, and let Y be a RV (independent of U) with p.d.f. $h(y) = t(y)/c$. If $U \leq g(Y)$, then Y has (conditional) p.d.f. $f(y)$.

Algorithm A-R

Repeat

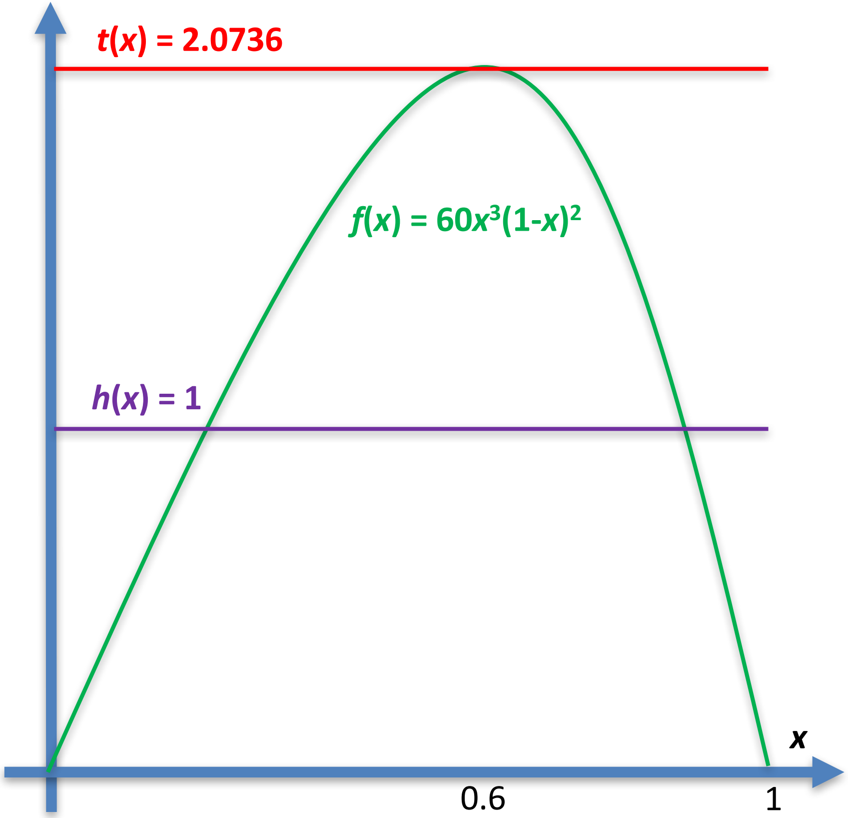
Generate U from $\mathcal{U}(0,1)$

Generate Y from $h(y)$ (independent of U)

until $U \leq g(Y) = \frac{f(Y)}{t(Y)} = \frac{f(Y)}{c\,h(Y)}$

Return $X \leftarrow Y$

Example (Law 2015): Generate a RV with p.d.f. $f(x) = 60x^3(1-x)^2$, $0 \le x \le 1$. Can't invert this analytically.



Max occurs at $x = 0.6$, and $f(0.6) = 2.0736$.

(Inefficient) majorizer $t(x) = 2.0736$.

Get $c = \int_0^1 t(x)dx = 2.0736$, so that $h(x) = t(x)/c = 1$, i.e., a Unif(0,1) p.d.f., and

$$g(x) = \frac{f(x)}{t(x)} = \frac{60x^3(1-x)^2}{2.0736}$$

E.g., if $U = 0.13$ and $Y = 0.25$, then turns out that $U \le g(Y)$, so we take $X \leftarrow 0.25$.

Demo Time!

Example (Ross): Generate a standard *half-normal* RV, with p.d.f.

$$f(x) \;=\; \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \geq 0.$$

Use the majorizing function

$$t(x) \;=\; \sqrt{\frac{2e}{\pi}} e^{-x} \;\geq\; f(x), \quad \text{with} \quad c \;=\; \int_0^\infty t(x)\,dx \;=\; \sqrt{\frac{2e}{\pi}}.$$

Then

$$h(x) \;=\; t(x)/c \;=\; e^{-x} \quad \text{(easy Exp(1) p.d.f.)},$$

and

$$g(x) \;=\; f(x)/t(x) \;=\; e^{-(x-1)^2/2}. \quad \square$$

We can use the half-normal result to generate a $\mathrm{Nor}(0,1)$ variate.

Generate U from $\mathcal{U}(0,1)$.

Generate X from the half-normal distribution.

Return

$$Z = \begin{cases} -X & \text{if } U \le 1/2 \\ X & \text{if } U > 1/2. \end{cases}$$

Reminder: We can then generate $\mathrm{Nor}(\mu, \sigma^2)$ RVs by using the obvious transformation $\mu + \sigma Z$.

Summary

This Time: We used Acceptance-Rejection to generate two non-trivial continuous RVs.

Next Time: We'll apply what looks like A-R on a discrete example.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

A-R Method: Poisson Distribution

Lesson Overview

Last Conclave: Used A-R on a couple of continuous examples: a crazy polynomial, and a half-normal.

This Conclave: Use a method similar to A-R to generate a *discrete* RV.

Something fishy is in the air!

Example: The Poisson distribution with probability mass function

$$P(X = n) = e^{-\lambda} \frac{\lambda^n}{n!}, \quad n = 0, 1, \ldots$$

We'll use a variation of A-R to generate a realization of X. The algorithm will go through a set of equivalent statements to arrive at a rule that gives $X = n$.

Recall that, by definition, $X = n$ if we observe exactly n arrivals from a Poisson(λ) process in one time unit.

Define A_i as the ith interarrival time from a Pois(λ) process.

$$
\begin{aligned}
X = n \quad &\Leftrightarrow \quad \text{See exactly } n \text{ Pois}(\lambda) \text{ arrivals by } t = 1 \\
&\Leftrightarrow \quad \sum_{i=1}^{n} A_i \le 1 < \sum_{i=1}^{n+1} A_i \\
&\Leftrightarrow \quad \sum_{i=1}^{n} \left[\frac{-1}{\lambda} \ell\text{n}(U_i) \right] \le 1 < \sum_{i=1}^{n+1} \left[\frac{-1}{\lambda} \ell\text{n}(U_i) \right] \\
&\Leftrightarrow \quad \frac{-1}{\lambda} \ell\text{n} \left(\prod_{i=1}^{n} U_i \right) \le 1 < \frac{-1}{\lambda} \ell\text{n} \left(\prod_{i=1}^{n+1} U_i \right) \\
&\Leftrightarrow \quad \prod_{i=1}^{n} U_i \ge e^{-\lambda} > \prod_{i=1}^{n+1} U_i. \qquad (5)
\end{aligned}
$$

n^{th} arrival occurs by time 1; and $(n+1)^{\text{st}}$ arrival occurs after time 1

The following A-R algorithm samples $\mathcal{U}(0,1)$'s until (5) becomes true, i.e., until the first time n such that $e^{-\lambda} > \prod_{i=1}^{n+1} U_i$.

Algorithm

$a \leftarrow e^{-\lambda};\, p \leftarrow 1;\, X \leftarrow -1$

Until $p < a$

  Generate U from $\mathcal{U}(0,1)$

  $p \leftarrow pU;\, X \leftarrow X + 1$

Return X

Example (BCNN): Obtain a Pois(2) RV.

Sample until $e^{-\lambda} = 0.1353 > \prod_{i=1}^{n+1} U_i$.

n	U_{n+1}	$\prod_{i=1}^{n+1} U_i$	Stop?
0	0.3911	0.3911	No
1	0.9451	0.3696	No
2	0.5033	0.1860	No
3	0.7003	0.1303	Yes

Thus, we take $X = 3$. $\square$

Remark: An easy argument says that the expected number of U's that are required to generate one realization of X is $\mathrm{E}[X+1] = \lambda + 1$.

Remark: If $\lambda \geq 20$, we can use the normal approximation

$$\frac{X - \lambda}{\sqrt{\lambda}} \approx \mathrm{Nor}(0,1).$$

Algorithm (for $\lambda \geq 20$)

Generate Z from $\mathrm{Nor}(0,1)$.

Return $X = \max(0, \lfloor \lambda + \sqrt{\lambda} Z + 0.5 \rfloor)$ ("continuity correction").

E.g., if $\lambda = 30$ and $Z = 1.46$, then $X = \lfloor 30.5 + \sqrt{30}(1.46) \rfloor = 38$.

Remark: Of course, another way to generate a Pois(λ) is simply to table the c.d.f. values like we did in an earlier discrete inverse transform example. This may be more efficient and accurate than the above methods — which is not to say that the A-R method isn't clever and pretty!

A Final Note: A-R is used for many other random variables, and even stochastic processes. We just don't have time to do any additional fellas right now, so try not to be too sad. ☹

www.youtube.com/watch?v=LTBqYEwl8mo

Summary

This Time: We used Acceptance-Rejection to generate Poisson RVs.

Next Time: Stay composed! We'll be generating RVs via the Composition method.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Composition

Lesson Overview

Last Rendez-Vous: Used A-R to generate Poisson RVs.

This Rendez-Vous: Learn about Composition, which is useful when you have “mixtures” of RVs.

A very nice technique for RVs that exhibit certain structures.

Composition

Idea: Suppose a RV actually comes from *two* RV's (sort of on top of each other). E.g., your plane can leave the airport gate late for two reasons — air traffic delays and maintenance delays, which compose the overall delay time.

What if there are *many* reasons?

In any case, how to generate?

The goal is to generate a RV with c.d.f.

$$F(x) = \sum_{j=1}^{\infty} p_j F_j(x),$$

← don't panic, this may be small

where $p_j > 0$ for all j, $\sum_j p_j = 1$, and the $F_j(x)$'s are "easy" c.d.f.'s to generate from.

- Generate a positive integer J such that $P(J = j) = p_j$ for all j.
- Return X from c.d.f. $F_J(x)$.

Proof that X has c.d.f. $F(x)$: By the law of total probability,

$$\begin{aligned} P(X \le x) &= \sum_{j=1}^{\infty} P(X \le x | J = j) P(J = j) \\ &= \sum_{j=1}^{\infty} F_j(x) p_j = F(x). \quad \square \end{aligned}$$

Example: Laplace distribution. Exponential distribution reflected off of y-axis.

$$f(x) \equiv \begin{cases} \frac{1}{2}e^{x}, & x < 0 \\ \frac{1}{2}e^{-x}, & x > 0 \end{cases} \quad \text{and} \quad F(x) \equiv \begin{cases} \frac{1}{2}e^{x}, & x < 0 \\ 1 - \frac{1}{2}e^{-x}, & x > 0 \end{cases}$$

Meanwhile, let's decompose X into "negative exponential" and regular exponential distributions:

$$F_1(x) \equiv \begin{cases} e^{x} & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases} \quad \text{and} \quad F_2(x) \equiv \begin{cases} 0 & \text{if } x < 0 \\ 1 - e^{-x} & \text{if } x > 0 \end{cases}$$

Then

$$F(x) = \frac{1}{2}F_1(x) + \frac{1}{2}F_2(x),$$

so that we generate from $F_1(x)$ half the time, and $F_2(x)$ half.

We'll use inverse transform to solve $F_1(X) = e^X = U$ for X half the time, and $F_2(x) = 1 - e^{-X} = U$ the other half. Then

$$X \leftarrow \begin{cases} \ell\text{n}(U) & \text{w.p. } 1/2 \\ -\ell\text{n}(U) & \text{w.p. } 1/2 \end{cases} \qquad \square$$

GTx

Summary

This Time: Learned about Composition – how to generate RVs that can themselves be decomposed into several easy-to-generate RVs.

And speaking of decomposing, how is Beethoven doing these days?

Next Time: We'll learn about a cool way to general normal RVs.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Box-Muller Normal RVs

Lesson Overview

Last Caucus: Talked about the Composition RV generation method.

This Caucus: Learn about the Box-Muller method to generate normals.

We'll turn you into a true believer!

Box–Muller Method: Here's a nice, easy way to generate standard normals.

Theorem: If U_1, U_2 are i.i.d. $\mathcal{U}(0,1)$, then

$$\begin{aligned} Z_1 &= \sqrt{-2\ell\mathrm{n}(U_1)}\cos(2\pi U_2) \\ Z_2 &= \sqrt{-2\ell\mathrm{n}(U_1)}\sin(2\pi U_2) \end{aligned}$$

are i.i.d. Nor(0,1).

Note that the trig calculations must be done in radians.

Proof Someday soon. $\square$

Some interesting corollaries follow directly from Box–Muller.

Example: Note that $Z_1^2 + Z_2^2 \sim \chi^2(1) + \chi^2(1) \sim \chi^2(2)$.

But

$$\begin{aligned} Z_1^2 + Z_2^2 &= -2\ell\mathrm{n}(U_1)(\cos^2(2\pi U_2) + \sin^2(2\pi U_2)) \\ &= -2\ell\mathrm{n}(U_1) \\ &\sim \mathrm{Exp}(1/2). \end{aligned}$$

Thus, we've just proven that

$$\chi^2(2) \sim \mathrm{Exp}(1/2). \quad \square$$

Example: $Z_2/Z_1 \sim \text{Nor}(0,1)/\text{Nor}(0,1) \sim \text{Cauchy} \sim t(1)$.

Moreover,

$$Z_2/Z_1 \;=\; \frac{\sqrt{-2\ell\text{n}(U_1)}\,\sin(2\pi U_2)}{\sqrt{-2\ell\text{n}(U_1)}\,\cos(2\pi U_2)} \;=\; \tan(2\pi U_2).$$

Thus, we've just proven that

$$\tan(2\pi U) \;\sim\; \text{Cauchy} \quad \text{(and, similarly, } \cot(2\pi U) \;\sim\; \text{Cauchy).}$$

Similarly, $Z_2^2/Z_1^2 = \tan^2(2\pi U) \sim t^2(1) \sim F(1,1)$.

Polar Method — a little faster than Box–Muller.

1. Generate U_1, U_2 i.i.d. $\mathcal{U}(0,1)$.

   Let $V_i = 2U_i - 1$, $i = 1, 2$, and $W = V_1^2 + V_2^2$.

2. If $W > 1$, reject and go back to Step 1.

   O'wise, let $Y = \sqrt{-2\ell\text{n}(W)/W}$, and accept $Z_i \leftarrow V_i Y$, $i = 1, 2$.

Then Z_1, Z_2 are i.i.d. Nor(0,1).

A Curious Misconception

It’s “Box-Muller”, not “Box-Müller”.
Surprising, considering that…

Umlauts are everywhere!

Many German and Turkish words:
Düsseldorf, Fahrvergnügen, köpek,
Bärkenpantzensniffersnatcher,…

Euro-Trash fake brand names:
Häagen-Dazs, Freshëns,…

More Umlauts!

Heavy-Metal Rock Groups:

- Motörhead
- Blue Öyster Cult
- Mötley Crüe (*two* umlauts!)

Talent-free teen stars looking to stay relevant

Summary

This Time: Looked at the Box-Muller method for generating normal, along with a couple of bonus corollaries.

Next Time: Some special-case tricks involving order statistics and some other distributions.

Computer Simulation

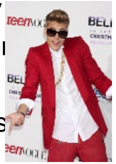
Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Order Statistics and Other Stuff

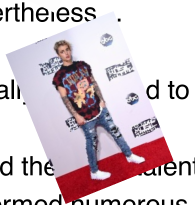
Apology to Mr. Ḃieber


Some ...s have unfortunately misinterpre... disparaging my com... ...ocent remarks from the last cl... ...out Mr. Ḃieber



Ge... ...ch denies any respons... ...ability for my remarks. Nevertheless...

Actuall... ...d to literary giant A... ...andre Dumas.



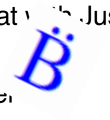
I used the ... ...ent-free" to signify that Mr. Ḃ... ...graciou... performed numerous free concerts so ... show... wonderful talent for the poor ...ldren of the ... e.

In fact, Mr. Bie... a wonderful s... um...aut.

https://www.youtube.com/watch?v=nntc...


Mr. Ḃieber ... his generation's great...

Selena rea... the boat w... Justin.



I am certainly a True Ḃelieber

Apology to Mr. Bieber

Some folks have unfortunately misinterpreted as disparaging my completely innocent remarks from the last class about Mr. Ḃieber.

Georgia Tech denies any responsibility or liability for my remarks. Nevertheless…

Actually compared to literary giant Alexandre Dumas.

I used the term “talent-free” to signify that Mr. Ḃieber has graciously performed numerous free concerts so as to showcase his wonderful talent for the poor children of the world to see.

In fact, Mr. Ḃieber has a wonderful song with an umlaut.

https://www.youtube.com/watch?v=nntGTK2Fhb0

Mr. Bieber is one of this generation’s great artists

Selena really missed the boat with Justin.

I am certainly a True Ḃelieḃer

Lesson Overview

Last Get-Together: Talked about the Box-Muller normal RV generation method.

This Get-Together: How to generate order statistics efficiently.

www.youtube.com/watch?v=53XyCbIJGKY

Order Statistics

Suppose that $X_1, X_2, \ldots, X_n$ are i.i.d. from some distribution with c.d.f. $F(x)$, and let $Y = \min\{X_1, \ldots, X_n\}$ with c.d.f. $G(y)$. (Y is called the first order stat.) Can we generate Y using just *one* $\mathcal{U}(0,1)$?

Yes! since the X_i's are i.i.d., we have

$$\begin{aligned} G(y) &= 1 - P(Y > y) = 1 - P(\min_i X_i > y) \\ &= 1 - P(\text{all } X_i\text{'s} > y) = 1 - [P(X_1 > y)]^n \\ &= 1 - [1 - F(y)]^n. \end{aligned}$$

GTx

Now do Inverse Transform: set $G(Y) = U$ and solve for Y. After a little algebra, get (don't be afraid)...

$$Y = F^{-1}\left(1 - (1-U)^{1/n}\right).$$

Example: Suppose $X_1, \ldots, X_n \sim \text{Exp}(\lambda)$. Then

$$G(y) = 1 - (e^{-\lambda y})^n = 1 - e^{-n\lambda y}.$$

Thus, $Y = \min_i\{X_i\} \sim \text{Exp}(n\lambda)$. So take $Y = -\frac{1}{n\lambda}\ell\text{n}(U)$. $\square$

We can do the same kind of thing for $Z = \max_i X_i$.

Other Quickies

$\chi^2(n)$ distribution: If $Z_1, Z_2, \ldots, Z_n$ are i.i.d. Nor(0,1), then $\sum_{i=1}^n Z_i^2 \sim \chi^2(n)$.

$t(n)$ distribution: If $Z \sim \text{Nor}(0,1)$ and $Y \sim \chi^2(n)$, and X and Y are independent, then

$$\frac{Z}{\sqrt{Y/n}} \sim t(n).$$

Note that $t(1)$ is the Cauchy distribution.

$F(n,m)$ distribution: If $X \sim \chi^2(n)$ and $Y \sim \chi^2(m)$ and X and Y are independent, then $(X/n)/(Y/m) \sim F(n,m)$.

Generating RV's from continuous empirical distributions — no time here. Can use the CONT function in Arena.

Summary

This Time: Showed how to efficiently generate certain order statistics.

Next Time: We'll finally be getting into *multivariate* generation! Example: heights and weights.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Multivariate Normal Distribution

Lesson Overview

Last Tête-à-Tête: Generated order statistics + some miscellaneous distributions.

This Tête-à-Tête: Multivariate normal!

We're about to enter a different dimension of sounds, sight, and mind.

www.youtube.com/watch?v=NzlG28B-R8Y

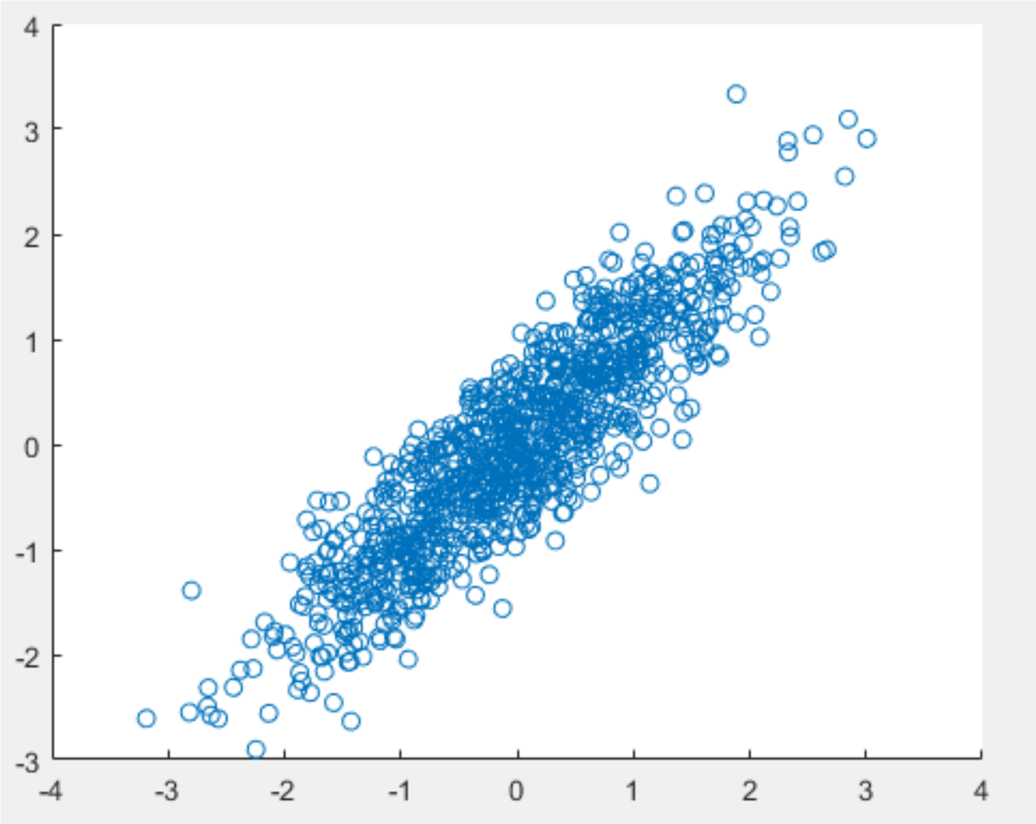
Bivariate Normal Distribution

The random vector (X, Y) has the *bivariate normal distribution* with means $\mu_X = \mathrm{E}[X]$ and $\mu_Y = \mathrm{E}[Y]$, variances $\sigma_X^2 = \mathrm{Var}(X)$ and $\sigma_Y^2 = \mathrm{Var}(Y)$, and correlation $\rho = \mathrm{Corr}(X, Y)$ if it has joint p.d.f.

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{\frac{-\left[z_X^2(x) + z_Y^2(y) - 2\rho z_X(x) z_Y(y)\right]}{2(1-\rho^2)}\right\},$$

where $z_X(x) \equiv (x - \mu_X)/\sigma_X$ and $z_Y(y) \equiv (y - \mu_Y)/\sigma_Y$.

For example, heights and weights of people can be modeled as bivariate normal.



MATLAB example: Bivariate normal means = 0, variances = 1, covariance = 0.9

Multivariate Normal Distribution

The random vector $\boldsymbol{X} = (X_1, \ldots, X_k)^{\mathrm{T}}$ has the *multivariate normal distribution* with mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^{\mathrm{T}}$ and $k \times k$ covariance matrix $\Sigma = (\sigma_{ij})$ if it has p.d.f.

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right\}, \quad \boldsymbol{x} \in \mathbb{R}^k.$$

$$\mathrm{E}[X_i] = \mu_i, \quad \mathrm{Var}(X_i) = \sigma_{ii}, \quad \mathrm{Cov}(X_i, X_j) = \sigma_{ij}.$$

Notation: $\boldsymbol{X} \sim \mathrm{Nor}_k(\boldsymbol{\mu}, \Sigma)$.

In order to generate $\boldsymbol{X}$, let's start with a vector $\boldsymbol{Z} = (Z_1, \ldots, Z_k)$ of i.i.d. Nor(0,1) RV's. That is, suppose $\boldsymbol{Z} \sim \text{Nor}_k(\boldsymbol{0}, I)$, where I is the $k \times k$ identity matrix, and $\boldsymbol{0}$ is simply a vector of 0's.

Suppose we can find the (lower triangular) Cholesky matrix C such that $\Sigma = CC^{\mathrm{T}}$.

Then it can be shown that $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$ is multivariate normal with mean $\boldsymbol{\mu}$ and covariance matrix

$$\Sigma \;\equiv\; \mathrm{E}[(C\boldsymbol{Z})(C\boldsymbol{Z})^T] \;=\; \mathrm{E}[C\boldsymbol{Z}\boldsymbol{Z}^T C^T] \;=\; C(\mathrm{E}[\boldsymbol{Z}\boldsymbol{Z}^T])C^T \;=\; CC^T$$

For $k = 2$, we can easily derive

$$C = \begin{pmatrix} \sqrt{\sigma_{11}} & 0 \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}} & \sqrt{\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}} \end{pmatrix}.$$

Since $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$, we have

$$\begin{aligned} X_1 &= \mu_1 + \sqrt{\sigma_{11}}\, Z_1 \\ X_2 &= \mu_2 + \frac{\sigma_{12}}{\sqrt{\sigma_{11}}}\, Z_1 + \sqrt{\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}}\, Z_2 \end{aligned}$$

The following algorithm computes C for general dimension k…

Algorithm

For $i = 1, \ldots, k$,

For $j = 1, \ldots, i - 1$,

$$c_{ij} \leftarrow \left(\sigma_{ij} - \textstyle\sum_{\ell=1}^{j-1} c_{i\ell}c_{j\ell}\right) / c_{jj}$$

$$c_{ji} \leftarrow 0$$

$$c_{ii} \leftarrow \left(\sigma_{ii} - \textstyle\sum_{\ell=1}^{i-1} c_{i\ell}^2\right)^{1/2}$$

Once C has been computed, the multivariate normal RV $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$ can easily be generated:

1. Generate $Z_1, Z_2, \ldots, Z_k \sim$ i.i.d. Nor$(0, 1)$.
2. Let $X_i \leftarrow \mu_i + \sum_{j=1}^{i} c_{ij} Z_j$, $i = 1, 2, \ldots, k$.
3. Return $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$.

Summary

This Time: Learned how to generate multivariate normal observations (correlated stuff like height vs. weight).

Next Time: We'll start playing around with a variety of useful stochastic processes.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Baby Stochastic Processes

Lesson Overview

Last Spiel: Multivariate normal distribution.

This Spiel: We'll start looking at the generation of some easy stochastic processes – *Markov chains* and *Poisson arrivals*.

www.youtube.com/watch?v=gGAiW5dOnKo

Markov Chains

Consider a time series having a certain number of states (e.g., sun / rain) that can transition from day to day.

Example: On Monday it's sunny, on Tues and Weds, it's rainy, etc.

Informally speaking, if tomorrow's weather only depends on today, then you have a *Markov chain*.

Markov Chains

Just do a simple example. Let $X_i = 0$ if it rains on day i; otherwise, $X_i = 1$. Denote the day-to-day transition probabilities by

$$P_{jk} \;=\; P(\text{state } k \text{ on day } i+1 \mid \text{state } j \text{ on day } i), \quad j,k = 0,1.$$

Suppose that the probability state transition matrix is

$$\mathbf{P} \;=\; \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}.$$

← e.g., $P_{01} = 0.3$

Suppose it rains on Monday. Simulate the rest of the work week.

	$P(\mathrm{R}\|X_{i-1})$	U_i	$U_i < P_{\cdot 0}$?	R/S
M	–	–	–	R
Tu	$P_{00} = 0.7$	0.62	Y	R
W	$P_{00} = 0.7$	0.03	Y	R
Th	$P_{00} = 0.7$	0.77	N	S
F	$P_{10} = 0.4$	0.91	N	S

www.youtube.com/watch?v=tIdIqbv7SPo

www.youtube.com/watch?v=FZmgGcZeayA

Poisson Arrivals

When the arrival rate is a *constant* λ, the interarrivals of a Poisson(λ) process are i.i.d. Exp(λ), and the arrival times are:

$$T_0 \leftarrow 0 \quad \text{and} \quad T_i \leftarrow T_{i-1} - \frac{1}{\lambda}\ell\text{n}(U_i), \quad i \geq 1.$$

Soooo easy!

Now suppose that we want to generate a *fixed number* n of PP(λ) arrivals in a *fixed time interval* $[a, b]$. To do so, we note a theorem stating that the joint distribution of the n arrivals is the same as the joint distribution of the order statistics of n i.i.d. $\mathcal{U}(a, b)$ RV's.

Generate i.i.d. $U_1, \ldots, U_n$ from $\mathcal{U}(0, 1)$

Sort the U_i's: $U_{(1)} < U_{(2)} < \cdots < U_{(n)}$

Set the arrival times to $T_i \leftarrow a + (b - a)U_{(i)}$

Still soooo easy!

Summary

This Time: Discussed some very simple Markov chain and Poisson arrival generation.

Next Time: What if the arrival rates *change over time*? We'll need to discuss *nonhomogeneous* Poisson processes!

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Nonhomogeneous Poisson Processes

Lesson Overview

Last Oration: MCs and PPs.

This Oration: Nonhomogeneous Poisson processes. What happens when the rate changes over time?

Careful! The "easy" algorithm may not work very well for NHPPs.

NHPPs – Nonstationary Arrivals

Same assumptions as regular Poisson process except the arrival rate λ isn't a constant, so stationary increments doesn't apply.

Let

$$\begin{aligned} \lambda(t) &= \text{rate (intensity) function at time } t, \\ N(t) &= \text{number of arrivals during } [0, t]. \end{aligned}$$

Then

$$N(s+t) - N(s) \sim \text{Poisson}\left(\int_s^{s+t} \lambda(u)\, du \right).$$

Example: Suppose that the arrival pattern to the Waffle House over a certain time period is a NHPP with $\lambda(t) = t^2$. Find the probability that there will be exactly 4 arrivals between times $t = 1$ and 2.

First of all, the number of arrivals in that time interval is

$$N(2) - N(1) \sim \text{Pois}\left(\int_1^2 t^2\,dt\right) \sim \text{Pois}(7/3).$$

Thus,

$$P\big(N(2) - N(1) = 4\big) = \frac{e^{-7/3}(7/3)^4}{4!} = 0.120. \quad \Box$$

Incorrect NHPP Algorithm [it can "skip" intervals with large $\lambda(t)$]

$T_0 \leftarrow 0$; $i \leftarrow 0$

Repeat

G

T

i

Don't use this algorithm!

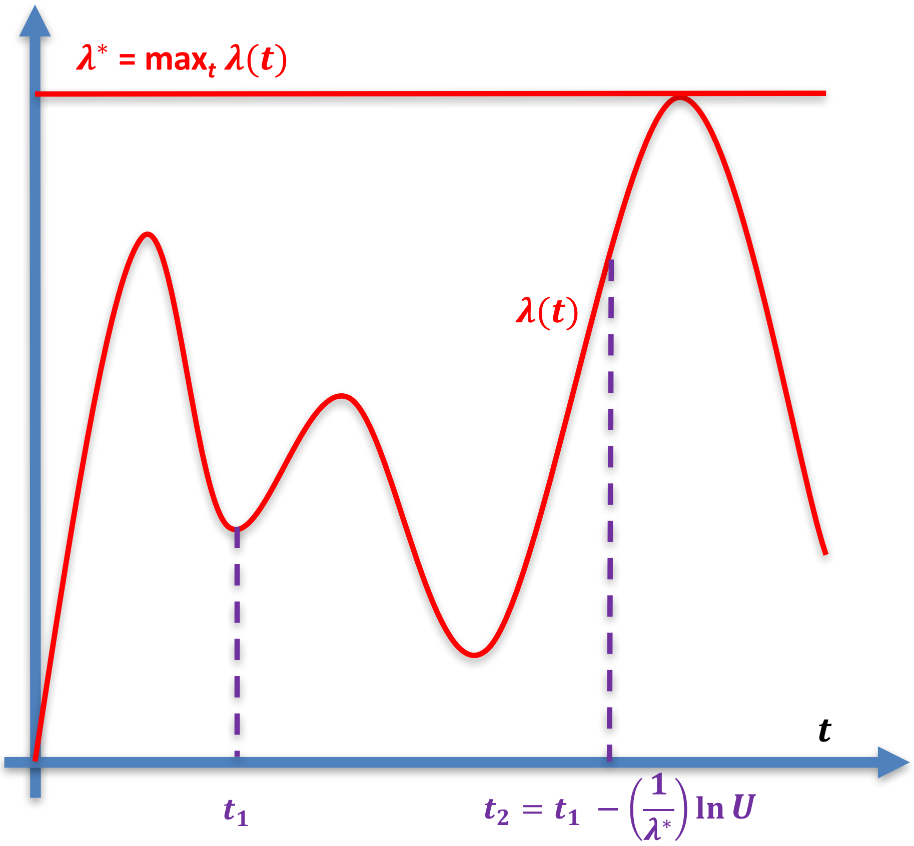
Whatever shall we do?

The ***Thinning Algorithm***

i. Assumes that $\lambda^* = \max_t \lambda(t)$ is finite,

ii. Generates *potential* arrivals at the max rate λ^*, and

iii. Accepts a potential arrival at time t w.p. $\lambda(t)/\lambda^*$.



Thinning Algorithm

$T_0 \leftarrow 0;\ i \leftarrow 0$

Repeat

$\quad t \leftarrow T_i$

$\quad$ Repeat

$\quad\quad$ Generate U, V from $\mathcal{U}(0,1)$

$\quad\quad t \leftarrow t - \frac{1}{\lambda^*}\ell\text{n}(U)$ Each t update represents a potential arrival (at rate λ^*)

$\quad$ until $V \le \lambda(t)/\lambda^*$ But we only keep the potential arrival w.p. $\lambda(t)/\lambda^*$

$\quad i \leftarrow i + 1$

$\quad T_i \leftarrow t$ These T_i's are the arrivals that we end up keeping.

Demo Time!

GTx

Summary

This Time: Talked about generating nonhomogeneous Poisson arrivals (where the arrival rate varies throughout the day).

Next Time: We'll look at various simple (and one not-so-simple) *time series* processes.

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Time Series

Lesson Overview

Last Soliloquy: NHPPs

This Soliloquy: Various time series processes.

We'll be doing standard normal-noise "ARMA" processes + a more-obscure "ARP" process with Pareto noise.

First-Order Moving Average Process

An MA(1) is a time series process is defined by

$$Y_i = \varepsilon_i + \theta\varepsilon_{i-1}, \quad \text{for } i = 1, 2, \ldots,$$

where θ is a constant and the ε_i's are i.i.d. $\text{Nor}(0, 1)$ RV's that are independent of Y_0.

The MA(1) is a popular tool for modeling and detecting trends.

The MA(1) has covariance function $\mathrm{Var}(Y_i) = 1 + \theta^2$,

$$\mathrm{Cov}(Y_i, Y_{i+1}) = \mathrm{Cov}(\varepsilon_i + \theta\varepsilon_{i-1}, \varepsilon_{i+1} + \theta\varepsilon_i) = \theta\mathrm{Var}(\varepsilon_i) = \theta,$$

and $\mathrm{Cov}(Y_i, Y_{i+k}) = 0$ for $k \geq 2$.

So the covariances die off pretty quickly.

How to generate? Start with $\varepsilon_0 \sim \mathrm{Nor}(0, 1)$. Then generate $\varepsilon_1 \sim \mathrm{Nor}(0, 1)$ to get Y_1, $\varepsilon_2 \sim \mathrm{Nor}(0, 1)$ to get Y_2, etc.

First-Order Autoregressive Process

An AR(1) process is defined by

$$Y_i = \phi Y_{i-1} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots,$$

where $-1 < \phi < 1$, $Y_0 \sim \text{Nor}(0, 1)$, and the ε_i's are i.i.d. $\text{Nor}(0, 1 - \phi^2)$ RV's that are independent of Y_0.

This is used to model lots of real-world stuff.

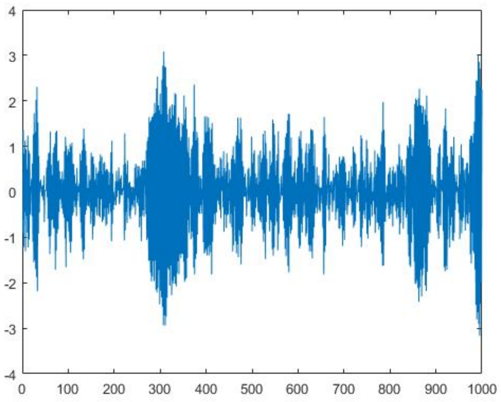
The AR(1) has covariance function $\mathrm{Cov}(Y_i, Y_{i+k}) = \phi^{|k|}$ for all $k = 0, \pm 1, \pm 2, \ldots$.

If ϕ is close to one, you get highly positively correlated Y_i's. If ϕ is close to zero, the Y_i's are nearly independent.

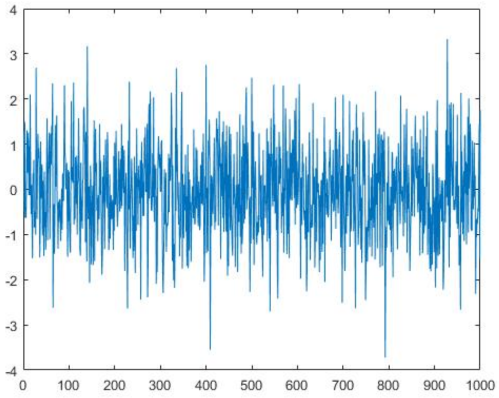
How to generate? Start with $Y_0 \sim \mathrm{Nor}(0, 1)$ and $\varepsilon_1 \sim \sqrt{1 - \phi^2}\,\mathrm{Nor}(0, 1)$ to get $Y_1 = \phi Y_0 + \varepsilon_1$.

Then generate $\varepsilon_2 \sim \sqrt{1 - \phi^2}\,\mathrm{Nor}(0, 1)$ to get $Y_2 = \phi Y_1 + \varepsilon_2$, etc.

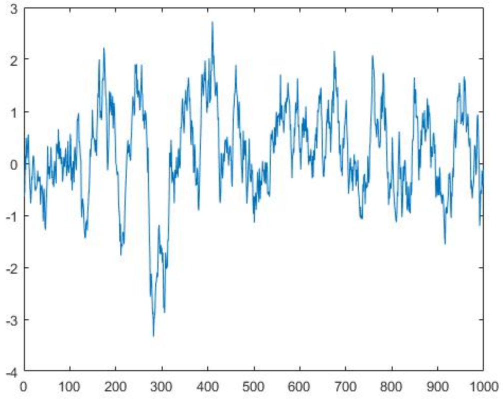
AR(1) pix



AR(1), ϕ = –0.95



AR(1), ϕ = 0



AR(1), ϕ = 0.95

ARMA(p,q) Process

An obvious generalization of the MA(1) and AR(1) processes is the ARMA(p,q), which consists of a pth order AR and a qth order MA, which we will simply define (without stating properties):

$$Y_i = \sum_{j=1}^{p} \phi_j Y_{i-j} + \epsilon_i + \sum_{j=1}^{q} \theta_j \epsilon_{i-j}, \quad i = 1, 2, \ldots,$$

where the ϕ_j's and θ_j's are chosen so as to assure that the process doesn't explode. Such processes are used in a variety of modeling and forecasting applications.

Exponential AR Process

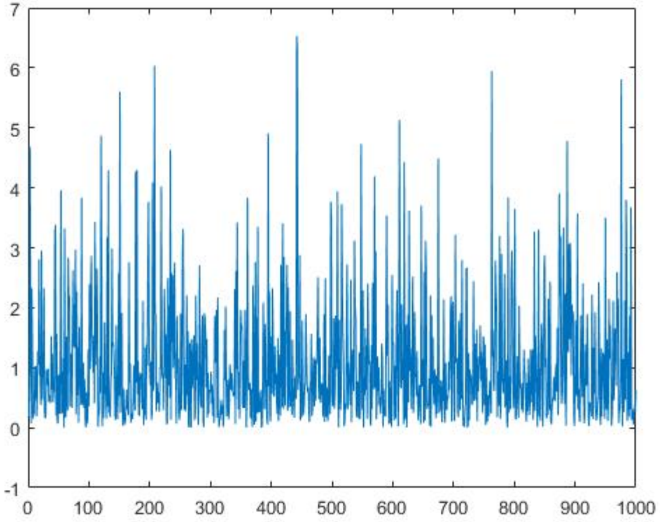
An EAR(1) process (Lewis 1980) is defined by

$$Y_i = \begin{cases} \phi Y_{i-1}, & \text{w.p. } \phi \\ \phi Y_{i-1} + \varepsilon_i, & \text{w.p. } 1-\phi \end{cases},$$

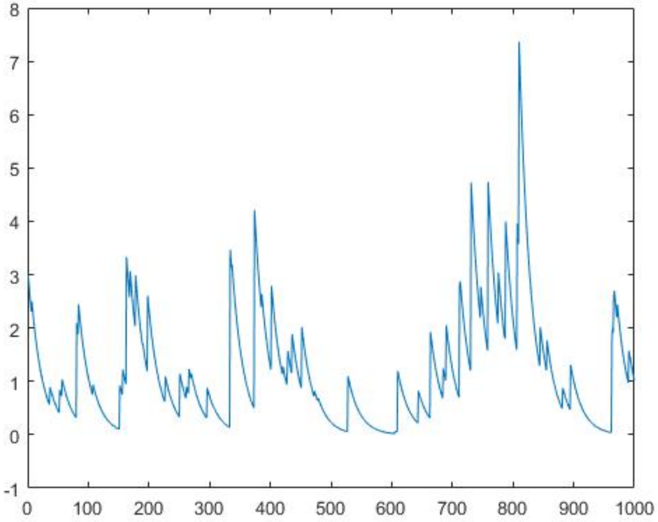
for $i = 1, 2, \ldots$, where $0 \le \phi < 1$, $Y_0 \sim \text{Exp}(1)$, and the ε_i's are i.i.d. Exp(1) RV's that are independent of Y_0.

The EAR(1) has the same covariance structure as the AR(1), except that $0 \le \phi < 1$, that is, $\text{Cov}(Y_i, Y_{i+k}) = \phi^{|k|}$.

EAR(1) pix



EAR(1), $\phi = 0.0$



EAR(1), $\phi = 0.95$

Autoregressive Pareto – ARP

Now let's see how to generate a series of correlated Pareto RV's. First of all, a RV X has the *Pareto distribution* with parameters $\lambda > 0$ and $\beta > 0$ if it has c.d.f.

$$F_X(x) \;=\; 1 - (\lambda/x)^{\beta}, \quad \text{for } x \geq \lambda.$$

The Pareto is a "heavy-tailed" distribution that has a variety of uses in statistical modeling.

In order to obtain the ARP process, let's start off with a regular AR(1) with normal noise,

$$Y_i = \rho Y_{i-1} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots,$$

where $-1 < \rho < 1$, $Y_0 \sim \text{Nor}(0, 1)$, and the ε_i's are i.i.d. $\text{Nor}(0, 1 - \rho^2)$ and independent of Y_0. Note that $Y_0, Y_1, Y_2, \ldots$ are marginally Nor(0,1) but correlated.

Feed this process into the Nor(0,1) c.d.f. $\Phi(\cdot)$ to obtain correlated Unif(0,1) RV's, $U_i = \Phi(Y_i)$, $k = 1, 2, \ldots$.

Now feed the correlated U_i's into the inverse of the Pareto c.d.f. to obtain correlated Pareto RV's:

$$X_i \;=\; F_X^{-1}(U_i) \;=\; F_X^{-1}(\Phi(Y_i)) \;=\; \frac{\lambda}{[1-\Phi(Y_i)]^{1/\beta}}, \quad i = 1, 2, \ldots.$$

Summary

This Time: Talked about a variety of time series models. AR(1), MA(1), ARMA, EAR, and the nasty ARP.

Next Time: We'll step back and take a quick look at a baby queueing process.

← Baby dreaming of queueing theory

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Queueing

Lesson Overview

Last Jam Session: Time series.

This Jam Session: An easy way to generate some queueing RVs.

This is actually a trivial lesson, since I want to take a breather before we end the module in the next lesson… You'll see…

M/M/1 Queue

Consider a single-server queue with customers arriving according to a Poisson(λ) process, standing in line with a FIFO discipline, and then getting served in an $\mathrm{Exp}(\mu)$ amount of time.

Let I_{i+1} denote the interarrival time between the ith and $(i+1)$st customers; let S_i be the ith customer's service time; and let W_i^Q denote the ith customer's wait before service.

Lindley gives a very nice way to generate a series of waiting times for this simple example (where you don't even need to worry about the exponential assumptions):

$$W_{i+1}^Q \;=\; \max\{W_i^Q + S_i - I_{i+1},\; 0\}.$$

And similarly, the total time in system, $W_i = W_i^Q + S_i$, is

$$W_{i+1} \;=\; \max\{W_i - I_{i+1},\; 0\} + S_{i+1}.$$

Summary

This Time: Some simple queueing results to enable easy generation of cycle and waiting times. Easy as pie!

Next Time: Brownian motion – one of my favorite topics!! And extremely important!

Computer Simulation

Module 7: Random Variate Generation

Dave Goldsman, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Brownian Motion

Lesson Overview

Last Board Meeting: Queueing results.

This Board Meeting: Generating Brownian motion.

Probably the most-important stochastic process out there. If you liked the Central Limit Theorem, you'll love this stuff!

Brownian Motion

Discovered by Brown; analyzed rigorously by Einstein; mathematical rigor established by Wiener (also called *Wiener* process).

Widely used in everything from financial analysis to queueing theory to statistics to other OR/IE application areas.

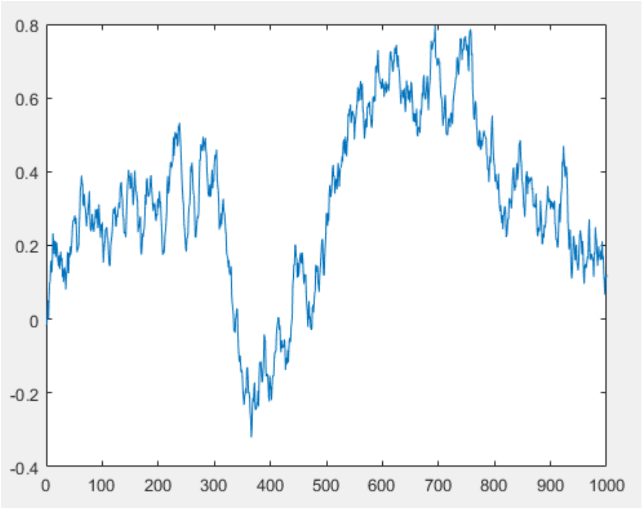
The stochastic process $\{\mathcal{W}(t), t \geq 0\}$ is *standard Brownian motion* if:

1. $\mathcal{W}(0) = 0$.
2. $\mathcal{W}(t) \sim \mathrm{Nor}(0, t)$.
3. $\{\mathcal{W}(t),\ t \geq 0\}$ has stationary and independent increments.

Increments: Anything like $\mathcal{W}(b) - \mathcal{W}(a)$.

Stationary increments: The distribution of $\mathcal{W}(t+h) - \mathcal{W}(t)$ only depends on h.

Independent increments: If $a < b < c < d$, then $\mathcal{W}(d) - \mathcal{W}(c)$ is indep of $\mathcal{W}(b) - \mathcal{W}(a)$.



How do you get BM? $Y_1, Y_2, \ldots$ is any sequence of i.i.d. RV's with mean zero and variance 1. (To some extent, the Y_i's don't even have to be indep!) *Donsker's* Central Limit Theorem says that

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} Y_i \xrightarrow{d} \mathcal{W}(t) \quad \text{as } n \to \infty,$$

where $\xrightarrow{d}$ denotes convergence in distribution as n gets big, and $\lfloor \cdot \rfloor$ is the floor function, e.g., $\lfloor 3.7 \rfloor = 3$.

The regular CLT is a very special case of this Big Boy!

Here's a way to construct BM:

One choice that works well is to take $Y_i = \pm 1$, each with probability 1/2. Take n at least 100, $t = 1/n, 2/n, \ldots, n/n$, and calculate $\mathcal{W}(1/n), \mathcal{W}(2/n), \ldots, \mathcal{W}(n/n)$.

Demo Time!

Another choice is simply to take $Y_i \sim \text{Nor}(0, 1)$.

Exercise: Let's construct some BM! First, pick some "large" value of n and start with $\mathcal{W}(0) = 0$. Then

$$\mathcal{W}\left(\frac{i}{n}\right) = \mathcal{W}\left(\frac{i-1}{n}\right) + \frac{Y_i}{\sqrt{n}}.$$

Here are some miscellaneous properties of Brownian Motion:

- BM is continuous everywhere, but has no derivatives! (Deep!)
  www.nbc.com/saturday-night-live/video/deep-thoughts-kryptonite/n10201 ⚠ (might be PG-13)
- $\text{Cov}(\mathcal{W}(s), \mathcal{W}(t)) = \min(s, t)$.
- Area under $\mathcal{W}(t)$ is normal: $\int_0^1 \mathcal{W}(t)\,dt \sim \text{Nor}(0, \frac{1}{3})$.
- A *Brownian bridge*, $\mathcal{B}(t)$, is conditioned BM such that $\mathcal{W}(0) = \mathcal{W}(1) = 0$.
- $\text{Cov}(\mathcal{B}(s), \mathcal{B}(t)) = \min(s, t) - st$.
- $\int_0^1 \mathcal{B}(t)\,dt \sim \text{Nor}(0, \frac{1}{12})$.

Geometric Brownian Motion

The process $S(t) = S(0)\exp\{(\mu - \frac{\sigma^2}{2})t + \sigma\mathcal{W}(t)\}$, $t \geq 0$, is often used to model stock prices, where μ is related to the "drift" of the stock price, σ is its volatility, and $S(0)$ is the initial price.

In addition, we can use GBM to estimate option prices. E.g., a European call option permits its owner, who pays an up-front fee for the privilege, to purchase the stock at a pre-agreed strike price k, at a pre-determined expiry date T. Its "value" is

$$e^{-rT}\mathrm{E}[(S(T) - k)^+],$$

where $x^+ = \max\{0, x\}$ and $\mu \leftarrow r$, the "risk-free" interest rate.

To estimate this expected value, we can run multiple simulation replications of $\mathcal{W}(T)$ and $(S(T) - k)^+$, and then take the sample average of the $e^{-rT}(S(T) - k)^+$ values.

Exercise: Let's estimate the value of a stock option. Pick your favorite values of r, σ, T, k, and off you go!

Lots of ways to actually do this. I would recommend that you directly simulate the BM many times as described on the last page.

But there are other ways: You can just simulate the distribution of $S(T)$ directly (it's lognormal), or you can actually look up the exact "Black–Scholes" answer (see below).

How to Win a Nobel Prize

Let $\phi(\cdot)$ and $\Phi(\cdot)$ denote the usual Nor(0,1) p.d.f. and c.d.f. Moreover, define the horrible-looking

$$b \equiv \frac{rT - \frac{\sigma^2 T}{2} - \ell n(k/S(0))}{\sigma\sqrt{T}}.$$

Now get your tickets to Norway or Sweden or wherever they give out the Nobel Prize…

The Black–Scholes European call option value is

$$
\begin{aligned}
& e^{-rT}\mathrm{E}[S(T)-k]^+ \\
&= e^{-rT}\mathrm{E}\left[S(0)\exp\left\{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\mathcal{W}(T)\right\}-k\right]^+ \\
&= e^{-rT}\int_{-\infty}^{\infty}\left[S(0)\exp\left\{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}z\right\}-k\right]^+\phi(z)\,dz \\
&= S(0)\Phi(b+\sigma\sqrt{T})-ke^{-rT}\Phi(b) \quad \text{(after lots of algebra).} \quad \square
\end{aligned}
$$

Summary

This Time: Brownian motion!
Study hard and make $ / **€ /** £ on Wall Street!

This completes Module 7, which was ginormous. It gets a little easier now (I hope).

Next on the agenda is Input Modeling – how do you decide what RVs to use to drive the simulation?