Composition

- Is applied when the distribution F can be expressed as a combination of other distributions F_1, F_2,…. F_n

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

$$\sum_{j=1}^{\infty} p_j = 1, \; p_j \geq 0$$

- Equivalent to say if X has density function f such that $f(x) = \sum_{j=1}^{\infty} p_j f_j(x),$

That corresponds to decomposing f into its convex combination representation

Composition

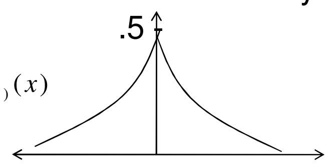
- The decomposition can also be seen as dividing the are under ***f*** into regions of areas p_1,p_2,…. p_n then determine F_j for each j then apply the inverse method on each one.

- Algorithm
  1. Generate a positive random integer, such that $P(J=j)=p_j$
  2. Return X with distribution F_j

Composition Example1

- For ***0<a<1***, the right trapezoidal distribution has density

$$f(x) = 0.5e^{|x|} \qquad \forall x \;\; real$$



Decompose $f(x) = .5e^{x} I_{(-\infty,0)}(x) + .5e^{-x} I_{[0,\infty)}(x)$

$$I_A = \begin{cases} 1 & x \in A \\ 0 & O.w. \end{cases}$$

$$f_1(x) = e^{x} I_{[0,1]}(x)$$

$$f_2(x) = e^{-x} I_{[0,1]}(x)$$

$$U_1 = F_1(x) = e^{x}, \quad U_2 = F_2(x) = e^{-x}$$

Use the inverse method

$$x = \ln U_1 \quad or \quad x = \ln U_2$$

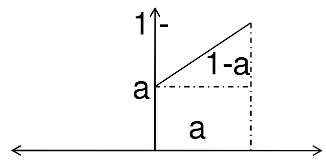
Composition Example1

- Algorithm:
- 1) Generate $U_1 \sim U(0,1), U_2 \sim U(0,1)$
- If U_1 <.5 return $x = \ln U_2$
- U_1 >=.5 $x = \ln U_2$

Composition Example2

- For ***0<a<1***, the right trapezoidal distribution has density

$$f(x) = \begin{Bmatrix} a + 2(1-a)x, 0 \le x \le 1 \\ 0, otherwise \end{Bmatrix},$$



- We may divide the are as shown
- ***f(x)*** can be decomposed as

$$f(x) = aI_{[0,1]}(x) + (1-a)2xI_{[0,1]}(x)$$

$f_1(x) = I_{[0,1]}(x)$ is U(0,1) density and

$f_2(x) = 2xI_{[0,1]}(x)$ is a right triangular density

$$p_1 = a, p_2 = 1-a, p_1 + p_2 = 1$$

Composition Example2

- ***Algorithm: Generate*** $U_1 \sim U(0,1), U_2 \sim U(0,1)$
- ***If***

$U_1 < a \ return \ x = U_1$

$U_2 \ge a \quad f_2 = 2x$

$$U = F_2(x) = x^2$$

$$x = \sqrt{U_2}$$

$$return \ \sqrt{U_2}$$

- ***Yet, in some applications, we find computing the square root is expensive So we may use another random number instead U_3 and return x=max(U_2, U_3)***

Convolution

- Assume that $X = Y_1 + Y_2 + ... + Y_m$ (where the ***Y***s are IID with CDF), called m-fold convolution of the distributions of Y's
- Algorithm

  1. Generate $Y_1, Y_2, ..., Y_m$ IID each with CDF

  2. Return $X = Y_1 + Y_2 + ... + Y_m$

Remark:

- Composition: Expressed the distribution function (or density or mass) as a (weighted) sum of other distribution functions (or densities or masses)
- Convolution: Express the random variable itself as the sum of other random variables

Normal Distribution

- Note that if $X \sim N(0,1)$

$$\Rightarrow \mu + \sigma X \sim N(\mu, \sigma)$$

- If we can generate unit normal, then we can generate any normal

$$f_Z(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \quad -\infty < x < \infty$$

$$\phi(x) = F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-x^2}{2}} dx$$

- Neither the distribution function nor the density function is invertible
- Use indirect method

Normal: Box-Muller

- Algorithm

$$1.\text{ Generate independent } U_1, U_2 \sim U(0,1)$$

$$2.\text{ Set } X_1 = \sqrt{-2\ln U_1}\cos 2\pi U_2,$$

$$X_2 = \sqrt{-2\ln U_1}\sin 2\pi U_2$$

$$3.\text{Return } X_1$$

- Each one X_1 or X_2 may be used. Each one of them is an I.I.D N(0,1)
- Technically, independent N(0,1), but serious problem if used with LCGs

Normal: Box-Muller

- For any normal distribution $N(\mu, \sigma^2)$
- Generate $S_i = \sigma X_i + \mu$