Generating discrete RVs

Wednesday, February 19, 2014 10:32 AM

Announcements

1. Three lectures today: 10.40 - 12:00
   13:00 - 14:20
   14:40 - 16:00

2. For the lectures in the afternoon,
   prepare some blank sheets of paper for taking note and working on in-class exercise.

3. Next HW .... HW 6 will be posted tomorrow.

4. Don't forget about the second project.

5. New set of slides today ... part of this overlap with slides from previous lecture(s).

Recall · last time we start discussion on generation of random variables.

First step : Talked about how to implement the "rand" command.

↳ generate uniform discrete RV on the interval (0,1).

Classical Technique : MCG

congruential

⇓

mod

$$x \bmod y = \mathrm{mod}(x,y) = x - ny \quad \text{where} \quad n = \left\lfloor \frac{x}{y} \right\rfloor$$

in MATLAB

floor function

For example, $13 \bmod 5 = 13 - 2 \times 5 = 3$ $\quad \frac{13}{5} = 2....$

$\lfloor \quad \rfloor = 2$

$\lfloor \ \rfloor = 2$

Alternatively, we can get the result as the remainder from the division.

$$\begin{array}{r} 2 \\ 5 \overline{)\,13} \\ \underline{10} \\ 3 \end{array}$$

Another example: $129 \bmod 7 = 3$

$$\begin{array}{r} 18 \\ 7 \overline{)\,129} \\ \underline{7} \\ 59 \\ \underline{56} \\ 3 \end{array}$$

Next step: Now that we know how to generate $\mathcal{U}(0,1)$ RV, we want to generate RVs with other distributions.

Bernoulli, binomial, Poisson, expo., Gaussian, etc.

Generating discrete RVs

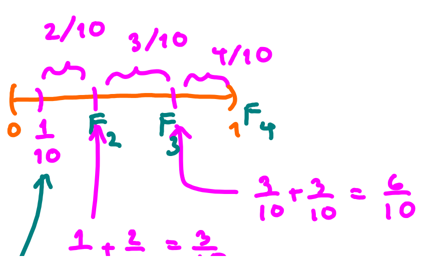
Ex. Suppose we want to generate X where

$$p_X(x) = \begin{cases} \underbrace{1/10}_{p_1}, & x = 1 \;(x_1) \\ \underbrace{2/10}_{p_2}, & x = 2 \;(x_2) \\ \underbrace{3/10}_{p_3}, & x = 3 \;(x_3) \\ \underbrace{4/10}_{p_4}, & x = 4 \;(x_4) \\ 0, & \text{otherwise} \end{cases}$$

First, we generate U.

Then we can generate X by determining the interval (bin) that U falls into.

The intervals (bins) are constructed such that their widths correspond to the probabilities in the pmf.



$F_1 = p_1 = F_X(x_1)$

$F_2 = F_1 + p_2 = p_1 + p_2 = F_X(x_2)$

$F_3 = F_2 + p_3 = p_1 + p_2 + p_3 = F_X(x_3)$

$\vdots$

$\frac{1}{10}+\frac{2}{10}=\frac{3}{10}$ $\quad \frac{3}{10}+\frac{3}{10}=\frac{6}{10}$

F_1

$F_3 = F_2 + p_3 = p_1 + p_2 + p_3 = F_X(x_3)$

$\vdots$

Boundaries of the intervals (bins) can be found from the cumulative sum of the probabilities in the pmf.

This can be formally considered via the cdf.

cumulative distribution function (cdf)

$$F_X(x) \equiv P[X \leq x]$$

Ex. $p_X(x) = \begin{cases} 1/3, & x=0, \\ 2/3, & x=1, \\ 0, & \text{otherwise.} \end{cases}$

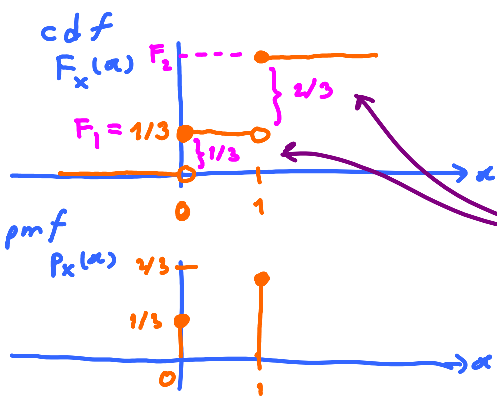
Try some values ....

$F_X(-1) = P[X \leq -1] = 0$

$F_X(0) = P[X \leq 0] = \frac{1}{3}$

$F_X(0.5) = P[X \leq 0.5] = \frac{1}{3}$

$F_X(1) = P[X \leq 1] = \frac{1}{3} + \frac{2}{3} = 1$



The cdf is piecewise constant.
Jumps occurs at the x values with positive pmf.
Amount of jump is the same as the corresponding pmf at that point.

Three characterizing properties of cdf

1. nondecreasing
2. $\lim_{x \to -\infty} F_X(x) = 0$, $\lim_{x \to \infty} F_X(x) = 1$
3. right-continuous

These three properties are very important because they hold regardless of the RV types.

The cdf grap could also be used to categorize RVs.

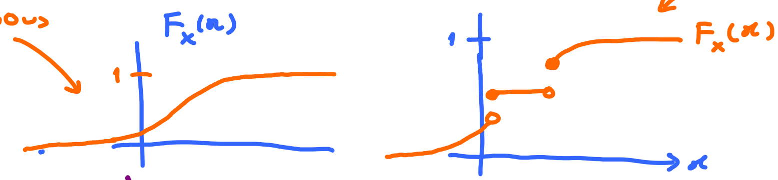
Discrete RV : cdf looks like staircase (piecewise constant)
Continuous RV : cdf is continuous

cdf of a

Continuous RV : cdf is continuous

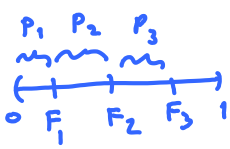
cdf of a continuous RV

cdf of a mixed RV



Back to generating discrete RV...

To work with RV with infinite support...



0 Generate a random number $U = \text{rand}$

1 $F_1 = p_1$

If $U < F_1$, set $X = x_1$ and stop.

2 $F_2 = F_1 + p_2$

If $U < F_2$, set $X = x_2$ and stop.

$F_1 \leq U < F_2$

3 $F_3 = F_2 + p_3$

If $U < F_3$, set $X = x_3$ and stop.

$\vdots$

$X = x_k$ iff $F_{k-1} < U < F_k$

More efficient way to generate geometric, RV :

$$X = \left\lceil \frac{\ln \text{rand}}{\ln(1-p)} \right\rceil$$

ceiling function

To see this...

$$p_X(x) = p(1-p)^{x-1}$$

$p_X(1) = p$ $\qquad$ $F_1 = p$

$p_X(2) = p(1-p)$ $\qquad$ $F_2 = p + p(1-p)$

$p_X(3) = p(1-p)^2$ $\qquad$ $F_3 = p + p(1-p) + p(1-p)^2$

$\vdots$ $\qquad$ $\vdots$

$$F_k = \sum_{x=1}^{k} p(1-p)^{x-1} = 1-(1-p)^k$$

$$= 1 - q^k, \quad q = 1-p$$

Recall that

Recall that

$X = k$ iff $F_{k-1} < U < F_k$

$$1 - q^{k-1} < U < 1 - q^k$$

$$1 - q^{k-1} < U \qquad\qquad U < 1 - q^k$$

$$-q^{k-1} < U - 1$$

$$q^{k-1} > 1 - U$$

$$(k-1)\ln(q) > \ln(1-U)$$

$$k < \frac{\ln(1-U)}{\ln(q)} + 1 \qquad\qquad k > \frac{\ln(1-U)}{\ln q}$$

$$\frac{\ln(1-U)}{\ln(q)} < k < \frac{\ln(1-U)}{\ln q} + 1$$

$$k = \left\lceil \frac{\ln(1-U)}{\ln(1-p)} \right\rceil$$

can be replaced by U because they have the same distribution (pdf)

Now that we have an efficient way to generate geometric RV, we also have an efficient way to generate Bernoulli trials.

Previously, we first generate a sequence of $\mathcal{U}(0,1)$ RVs and then set the result of the Bernoulli trials to be 0 or 1 by thresholding the generated sequence.
This requires generating many $\mathcal{U}(0,1)$ RVs.

A different approach is to recall that the numbers of slots between adjacent success events are i.i.d. Geometric RVs.
One realization of a geometric RV corresponds to ≥ 1 $\mathcal{U}(0,1)$ RVs.
For example, we with the geometric RV value of 5, the corresponding Bernoulli trials results are $\underbrace{00001}$.

↑
~~x~~ slots until the next success is 5

If we need to consider n trials, then we can use the while loop to generate geometric RVs so long as their cumulative sum still does not exceed n.

The same idea can be used to generate a Poisson process from a sequence of exponential RVs because these RVs describes the times between adjacent events.