Stats 102C - Lecture 3-2: More Inverse CDF; Convolutions

Miles Chen, acknowledgements Michael Tsiang

Week 3 Wednesday

Section 1

More Inverse CDF

Inverse CDF Method

Goal: generate samples $X \sim F(x)$

1. Derive the inverse CDF $F^{-1}(u)$
2. Generate $U \sim \text{Unif}(0, 1)$
3. Let $X = F^{-1}(U)$

The inverse CDF transform is defined as:

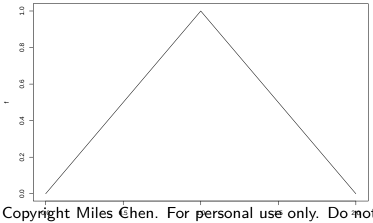
$$F^{-1}(u) := \min \{t : F(t) \geq u\}, \text{for } 0 \leq u \leq 1$$

Example: Triangular distribution

The following triangular distribution is the distribution of a sum of two random values from a uniform distribution.

It has the following PDF:

$$f(x) = \begin{cases} x & \text{for } 0 \leq x \leq 1 \\ 2 - x & \text{for } 1 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases}$$



Example: Triangular distribution

The CDF is found piece-wise.

$$
\begin{aligned}
F(x) &= \int_0^x f(t)dt \\
&= \int_0^x t dt \\
&= \left[\frac{t^2}{2}\right]_0^x \\
&= \frac{x^2}{2}, \qquad \text{for } 0 \leq x \leq 1
\end{aligned}
$$

Example: Triangular distribution

The other piece of the CDF:

$$
\begin{aligned}
F(x) &= \int_0^x f(t)dt \\
&= \int_0^1 t dt + \int_1^x 2 - t dt \\
&= \left[\frac{t^2}{2}\right]_0^1 + \left[2t - \frac{t^2}{2}\right]_1^x \\
&= \frac{1}{2} + \left(2x - \frac{x^2}{2}\right) - \left(2 - \frac{1}{2}\right) \\
&= -1 + 2x - \frac{x^2}{2}, \qquad \text{for } 1 \leq x \leq 2
\end{aligned}
$$

Inverse CDF of Triangular Distribution

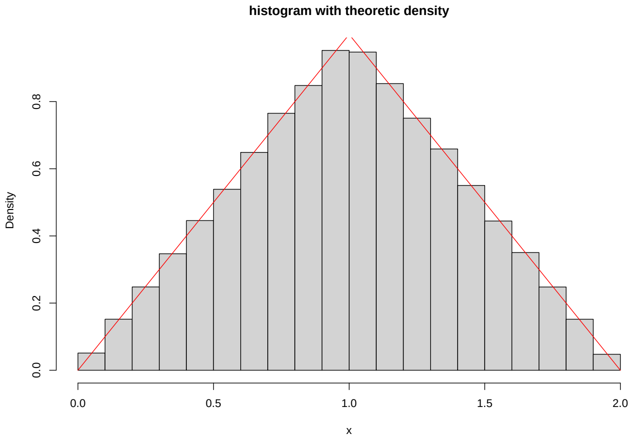
Find the inverse of F. Set $F(x) = u$ and solve for x. This too is done piecewise.

$$
\begin{aligned}
F(x) &= u \\
\frac{x^2}{2} &= u, \qquad \text{for } 0 \leq x \leq 1 \\
x &= \pm\sqrt{2u} \qquad \text{because } 0 \leq x \text{ we keep only the } + \text{ side} \\
x &= \sqrt{2u}, \qquad \text{for } 0 \leq u \leq \frac{1}{2}
\end{aligned}
$$

Inverse CDF of Triangular Distribution

$$
\begin{aligned}
F(x) &= u \\
-1 + 2x - \frac{x^2}{2} &= u, \qquad \text{for } 1 \leq x \leq 2 \text{ plug in x=1 and x=2 to get bounds of } u \\
2 - 4x + x^2 &= -2u \\
(2 - x)^2 &= -2u + 2 \qquad \text{complete the square} \\
2 - x &= \pm\sqrt{-2u + 2}, \qquad \text{because } 1 \leq x \leq 2, \text{ we keep only the + side} \\
2 - x &= \sqrt{-2u + 2} \\
2 - \sqrt{-2u + 2} &= x \\
2 - \sqrt{2(1 - u)} &= x, \qquad \text{for } \frac{1}{2} \leq u \leq 1
\end{aligned}
$$

```r
u <- runif(10 ^ 5) # generate U
x <- sqrt(2 * u) # create X based on first component
indicator <- u > 0.5
x[indicator] <- 2 - sqrt(2 * (1 - u[indicator])) # replace x based on 2nd component
```



Section 2

Inverse CDF for Discrete Distributions

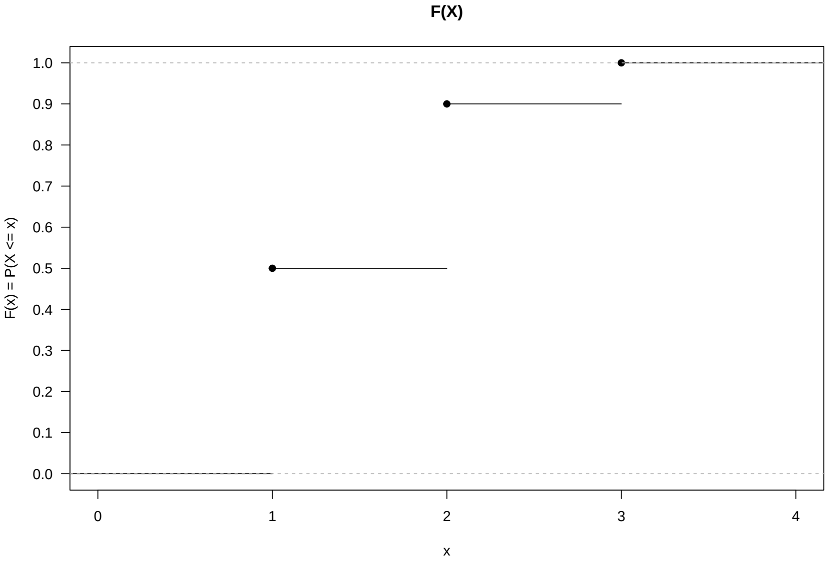
PMF and CDF of Discrete Distribution

Let's say you have a discrete distribution with the following PMF:

Outcome x_i	Probability $\Pr(X = x_i)$
1	0.5
2	0.4
3	0.1

The CDF $F(x) = \Pr(X \leq x)$ is a discontinuous step function.

X	Cumulative Prob. $\Pr(X \leq x_i)$
$X < 1$	0
$1 \leq X < 2$	$\Pr(X = 1) = 0.5$
$2 \leq X < 3$	$\Pr(X = 1) + \Pr(X = 2) = 0.9$
$3 \leq X$	$\Pr(X = 1) + \Pr(X = 2) + \Pr(X = 3) = 1$



F(X)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
F(x) = P(X <= x)
0
1
2
3
4
x

Inverse CDF - Discrete Distributions

The inverse CDF is:

$F^{-1}(u) = \min\{t : F(t) \geq u\}$

For example:

- If $u = 0.326$, the smallest t that has $F(t)$ greater than $u = 0.326$ is $t = 1$.
- If $u = 0.786$, the smallest t that has $F(t)$ greater than $u = 0.786$ is $t = 2$.

The complete Inverse CDF can then be used to generate X

$$F^{-1}(u) = \begin{cases} 1 & \text{for } 0 < u \leq 0.5 \\ 2 & \text{for } 0.5 < u \leq 0.9 \\ 3 & \text{for } 0.9 < u < 1 \end{cases}$$

Inverse CDF - Discrete Distributions

- Sample $U \sim \text{Unif}(0, 1)$
- Generate X as follows

$$X = \begin{cases} 1 & \text{if } 0 < U \leq 0.5 \\ 2 & \text{if } 0.5 < U \leq 0.9 \\ 3 & \text{if } 0.9 < U < 1 \end{cases}$$

PMF Discrete Distributions

In general, if a discrete distribution has the following PMF:

Outcome x_i	Probability $\Pr(X = x_i)$
x_1	p_1
x_2	p_2
$\vdots$	$\vdots$
x_K	p_K

Where $\sum_k p_k = 1$

CDF Discrete Distributions

Then the CDF will be:

X	Cumulative Prob. $\Pr(X \leq x_i)$
$X < x_1$	0
$x_1 \leq X < x_2$	p_1
$x_2 \leq X < x_3$	$p_1 + p_2$
$\vdots$	$\vdots$
$x_j \leq X < x_{j+1}$	$\sum_{k=1}^{j} p_k$
$\vdots$	$\vdots$
$x_K \leq X$	$\sum_{k=1}^{K} p_k = 1$

Inverse CDF Discrete Distributions

The inverse CDF can then be used to generate values of X by sampling $U \sim \text{Unif}(0,1)$

$$X = F^{-1}(U) = \begin{cases} x_1 & \text{if } 0 < U \leq p_1 \\ x_2 & \text{if } p_1 < U \leq p_1 + p_2 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{k=1}^{j-1} p_k < U \leq \sum_{k=1}^{j} p_k \\ \vdots & \vdots \\ x_K & \text{if } \sum_{k=1}^{K-1} p_k < U \leq 1 \end{cases}$$

Inverse CDF Algorithm in R

```r
X <- c(1, 2, 3)
prob <- c(0.5, 0.4, 0.1)
rand_discrete <- function(X, prob, info = FALSE) {
  U <- runif(1)
  position <- 1
  while( U > sum(prob[1:position]) ) {
    position <- position + 1
  }
  if(info){ print(U) } # only for informational purposes
  X[position] # returned value
}
```

Inverse CDF Algorithm in R

```r
X <- c(1, 2, 3)
prob <- c(0.5, 0.4, 0.1)
set.seed(5)
rand_discrete(X, prob, info = TRUE)
```

```
## [1] 0.2002145

## [1] 1
```

```r
rand_discrete(X, prob, info = TRUE)
```

```
## [1] 0.6852186

## [1] 2
```

```r
rand_discrete(X, prob, info = TRUE)
```

```
## [1] 0.9168758

## [1] 3
```

Inverse CDF Algorithm in R

R's `sample()` function can generate values from any discrete distribution. You just need to provide the vector of possible `X` values and the vector of corresponding `probabilities`.

```
X <- c(1, 2, 3)
prob <- c(0.5, 0.4, 0.1)
set.seed(500)
sample(x = X, size = 30, prob = prob, replace = TRUE)
```

```
##  [1] 2 2 3 1 2 1 2 3 2 2 1 2 2 1 2 1 2 2 2 1 1 2 1 2 1 2 2 2 1 1
```

The `sample()` function in R will produce the same results as our own random discrete distribution code if provided the same seed.

```
set.seed(500)
replicate(30, rand_discrete(X, prob, info = FALSE))
```

```
##  [1] 2 2 3 1 2 1 2 3 2 2 1 2 2 1 2 1 2 2 2 1 1 2 1 2 1 2 2 2 1 1
```

Section 3

Convolutions

Convolutions

Let $X_1, X_2, \ldots, X_m$ be independent random variables.

The **convolution** of these variables is the sum.

$$S = \sum_{i=1}^{m} X_i = X_1 + X_2 + \ldots + X_m$$

Many random variable distributions are convolutions.

Convolution Example: Binomial

$X_1, X_2, \ldots, X_m \sim$ Bernoulli(p) are Bernoulli random variables with parameter p.

$$X_i = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$

Then the convolution

$$S = \sum_{i=1}^{m} X_i = X_1 + X_2 + \ldots + X_m$$

has a binomial distribution with parameters m and p.

Binomial generation

Generate m values of $U \sim \text{Unif}(0,1)$. Let $X_i = 1$ if $U < p$ and $X_i = 0$ otherwise. Sum X_i

Binomial with m = 10, and p = 0.6

```
rand_binom <- function(m, p) {
  U <- runif(m)
  X <- rep(0, m) # set all X to be 0
  X[U < p] <- 1  # set X to 1 where U < p
  sum(X) # returned value
}
m <- 10
p <- 0.6
rand_binom(m, p)
```

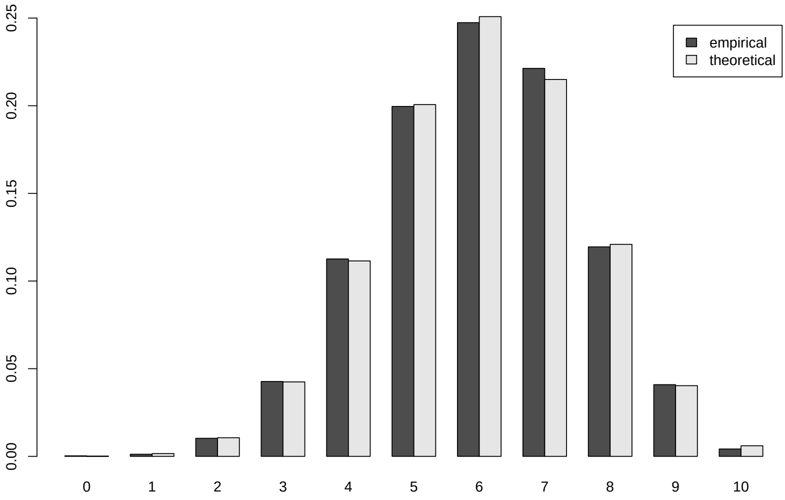
```
## [1] 6
```

Binomial generation

```r
set.seed(101)
n <- 10^4
samp <- replicate(n, rand_binom(m, p))
empirical <- table(samp) / n # empirical probabilities
theoretical <- dbinom(0:10, m, p) #  theoretic probabilities
comparison <- rbind(empirical, theoretical)
round(comparison, 4)
```

```
##                  0      1      2      3      4      5      6      7      8
## empirical   3e-04 0.0012 0.0103 0.0427 0.1126 0.1996 0.2474 0.2213 0.1195
## theoretical 1e-04 0.0016 0.0106 0.0425 0.1115 0.2007 0.2508 0.2150 0.1209
##                  9     10
## empirical   0.0409 0.0042
## theoretical 0.0403 0.0060
```

```r
barplot(comparison, beside = TRUE, legend.text = row.names(comparison))
```



Chi-squared test

We can run a chi-squared test to see if the generated values match the expected probabilties.

```r
# x is a vector of counts, which can be found using table
chisq.test(x = table(samp), p = theoretical)
```

```
## Warning in chisq.test(x = table(samp), p = theoretical): Chi-squared
## approximation may be incorrect

##
##  Chi-squared test for given probabilities
##
## data:  table(samp)
## X-squared = 13.006, df = 10, p-value = 0.2234
```

We get a warning because some of the cells (like 0 and 1) have very low counts and very low expected probabilities. The chi-squared test is not recommended when the expected counts of any cell is less than 5.

Chi-squared test

To fix the issue, I will combine the observed counts of the 0 and 1 cells. This technique of combining categories with small probabilities is called "collapsing categories".

```
xcounts <- table(samp)
xcounts
```

```
## samp
##     0    1    2    3    4    5    6    7    8    9   10
##     3   12  103  427 1126 1996 2474 2213 1195  409   42
```

```
xcounts["1"] <- xcounts["0"] + xcounts["1"] # add the 0 counts to the 1 cell
xcounts <- xcounts[-1] # delete the first cell (the 0 cell)
xcounts
```

```
##    1    2    3    4    5    6    7    8    9   10
##   15  103  427 1126 1996 2474 2213 1195  409   42
```

Chi-squared test

I make the same adjustments to the vector of theoretic probabilities so the two vectors are of the same length and I rerun the chi-squared test.

```
# collapse theoretic probabilities for 0 and 1
theoretical <- dbinom(0:10, m, p)
theoretical[2] <- theoretical[1] + theoretical[2]
theoretical <- theoretical[-1]
chisq.test(x = xcounts, p = theoretical)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  xcounts
## X-squared = 8.6784, df = 9, p-value = 0.4675
```

The resulting p-value is greater than 0.05 indicating that we have no reason to believe that the observed values do not match the theoretic probabilities.

Convolution Example: Poisson

The Poisson distribution is used to model the number of occurrences of a random event in a set time frame (examples: number of pieces of mail received in a day, number of injuries in a month, etc.). The parameter λ is the rate.

If $X_i \sim \text{Pois}(\lambda_i), i = 1, 2, \ldots, m$ are independent Poisson random variables for $\lambda_i > 0$, then

$$S = \sum_{i=1}^{m} X_i = X_1 + X_2 + \ldots + X_m \sim \text{Pois}\left(\sum_{i=1}^{m} \lambda_i\right)$$

has a Poisson distribution with mean parameters $\sum_{i=1}^{m} \lambda_i$

Convolution Example: Negative Binomial

The geometric distribution is used to model the number X of failures in a sequence of Bernoulli trials before the first success. For example, what is the probability we must roll X non-sixes before we see a 6.

The negative binomial distribution is used to model the number of failures observed in a sequence of Bernoulli trials before the m^{th} success occurs. For example, what is the probability we must roll X non-sixes before we see a 6 for the third time.

The negative binomial distribution can be expressed as a convolution of geometric random variables.

$X_1, X_2, \ldots, X_m \overset{iid}{\sim} \text{Geom}(p)$ are Bernoulli random variables with parameter p.

$$S = \sum_{i=1}^{m} X_i = X_1 + X_2 + \ldots + X_m \sim \text{NegBin}(m, p)$$

has a negative binomial distribution with parameters m and p.

Convolution Example: Chi-square

Let $Z_1, Z_2, \ldots, Z_m \overset{iid}{\sim} \mathcal{N}(0,1)$ be independent random Normal variables.

The convolution of the squares

$$S = \sum_{i=1}^{m} Z_i^2 = Z_1^2 + Z_2^2 + \ldots + Z_m^2 \sim \chi^2(m)$$

has a chi-square distribution with m degrees of freedom.

Convolution Example: Gamma

The exponential distribution is used to model the amount of time between random events in a Poisson point process: a process where random events occur continuously and independently at a constant average rate.

The gamma distribution can be used to model the amount of time that must pass before the m^{th} occurrence of a random event.

The gamma distribution can be expressed as a convolution of exponential random variables.

$X_1, X_2, \ldots, X_m \overset{iid}{\sim} \text{Exp}(\lambda)$ are independent Exponential random variables with rate parameter λ.

$$S = \sum_{i=1}^{m} X_i = X_1 + X_2 + \ldots + X_m \sim \text{Gamma}(m, \lambda)$$

has a negative binomial distribution with shape parameter m and rate parameter λ.

Convolution Example: Beta

The beta distribution can be expressed in a couple ways.

$X_1, X_2, \ldots, X_a, X_{a+1} \ldots X_{a+b} \overset{iid}{\sim}$ Exp$(\lambda = 1)$ are independent Exponential random variables with rate parameter $\lambda = 1$.

Then

$$Y = \frac{\sum_{i=1}^{a} X_i}{\sum_{i=1}^{a+b} X_i} \sim \text{Beta}(a, b)$$

has a beta distribution with parameters a, b for natural numbers a and b.

Note the denominator is the sum of all X_i values and the numerator is the sum of the first a X_i values.

Convolution Example: Beta

The beta distribution can also be expressed as:

Let $U \sim \text{Gamma}(a, \lambda)$ and $V \sim \text{Gamma}(b, \lambda)$ be independent gamma random variables with the same rate parameter λ.

$$X = \frac{U}{U+V} \sim \text{Beta}(a, b)$$

has a beta distribution with parameters a, b.

Another example: t-distribution

A random variable from the t-distribution can be expressed as

$$T = \frac{Z}{\sqrt{V/v}}$$

Where

- Z is a random value from the standard normal distribution.
- V is a random value from a chi-square distribution with v degrees of freedom
- Z and V are independent