TAM 598 Lecture 7:

UNCERTAINTY PROPAGATION – BASIC SAMPLING

i.e. how to generate random numbers

Announcements:

- HW 2 covers lectures 4-8; due on Feb 26

Why generate random numbers?

(1) Monte Carlo - to estimate difficult expectations, to solve high-dimensional integrals

(2) Randomized Algorithms - use random numbers to influence their behavior or decisions, to do things faster

eg) Quicksort - random choice of pivot element, reduces from $O(n^2)$ worst case to $O(n \log n)$

(3) cryptography

(4) games & graphics

Computers cannot generate true random numbers

We usually don't need true random numbers, but need sequences that have the same statistics as true random numbers:

- distributed according to the pmf or pdf
- statisically uncorrelated (no detectable patterns)

Random numbers that we generate deterministically on a computer are called pseudo random numbers

- initialize sequence with a seed, and iterate

Today: three pseudo random number generators (PRNGs)

1) Middle Square Algorithm
2) Linear Congruential Generator
3) Mersenne Twister

Algorithms for random number generation follow a typical recipe.

$\mathcal{U}$ = output set, eg uniform, unsigned 32 bit integers

$\mathcal{S}$ = state space

We need two functions

$$\begin{cases} f : \mathcal{S} \to \mathcal{S} \\ g : \mathcal{S} \to \mathcal{U} \end{cases}$$

f ← evolves state forward

g ← takes state and produces integer u in desired distribution

and an initial state, called the SEED

$$s_0 \xrightarrow{} f(s_0) = s_1 \xrightarrow{} f(s_1) = s_2 \xrightarrow{} \cdots$$

$$g \downarrow u_0 \qquad g \downarrow u_1 \qquad g \downarrow u_2$$

Middle Square Algorithm - John von Neumann

1. state space $S = \{0, 1, \ldots, 10^{m+1} - 1\}$
2. $f(s) = \text{pad}(s^2, m)$
3. $g(s) = $ take the middle digits of $f(s)$

say $m = 8$
$S_{max} = 100,000,000 - 1$
pad result with zeroes to get desired # of digits

generates a sequence of numbers

$$S = \{s_0, s_1, s_2, \ldots\}$$

where s_{i+1} is entirely determined by s_i

eg $s_0 = 1234$
digits = 8

$1234^2 = 01522756$ → s_1

$5227^2 = 27321529$ → s_2

$3215^2 = \ldots$

issue: Necessarily, this and other PRNGs have some period. Eventually, the sequence will land on some previous state, and then start to repeat the same sequence.

Linear Congruential Generator

- state space $S = \{0, 1, \ldots, m-1\}$
- $f(s) = (as + b) \bmod m$ for chosen integers a, b, m
  - a: multiplier
  - b: increment
  - m: modulus
- $g(s) = s$

eg) $m = 2^{31} - 1$, $a = 2^7$, $b = 0$ $\Rightarrow$ period of $2^{31} - 2$

can get poor behavior for bad choices of a, b, m

eg) $a = 2$, $b = 0$, $m = 10$

$S_0 = 1234$

$S_1 = 2468 \bmod 10 = 8$

$S_2 = 16 \bmod 10 = 6$

$S_3 = 12 \bmod 10 = 2$

$S_4 = 4 \bmod 10 = 4$

$S_5 = 8 \bmod 10 = 8$

Mersenne Twister PRNG

- used in numpy and most languages

- period length is given by a Mersenne prime $2^n - 1$

$$2^{19937} - 1$$

If $2^n - 1$ is prime, then n is prime

- maintains an internal state of 624 integers, generates random numbers by performing bitwise operations on this state to produce a new sequence of bits with each iteration

  → generate 624 numbers

  → apply a "twisting" transformation to mix bits from adjacent integers

  → range is from 0 to $2^{31} - 1 = 2,147,483,647$

  so a 31 bit integer

Say we have a PRNG that yields a uniform distribution on $\mathcal{U} = \{0, 1, 2, \ldots, m-1\}$. How do we go to something more interesting / useful?

eg) uniform on $[0, 1)$ given by u/m

To test this:

- Empirical histograms should be uniform (pdf)
  - → can also evaluate the empirical cdf
- are consecutive numbers independent? ie uncorrelated

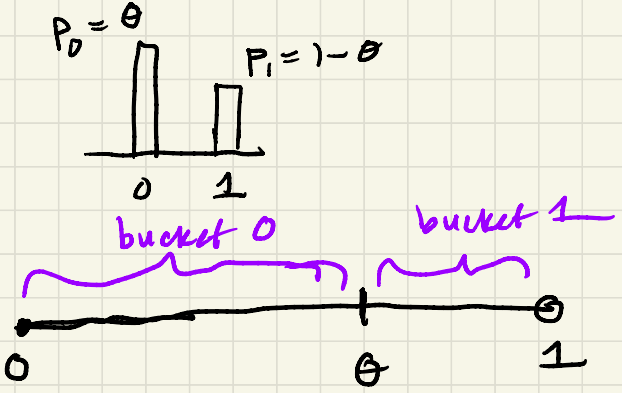
SAMPLING THE CATEGORICAL

(discrete distributions)

given we can draw from $U[0,1]$

$X \sim \text{Bernoulli}(\theta)$

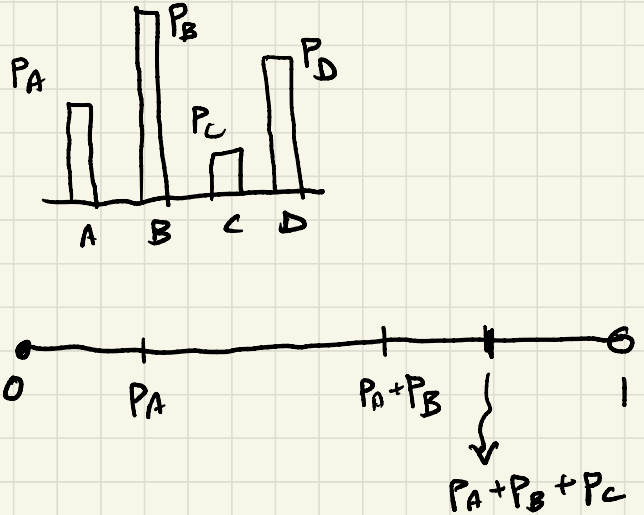
$$X = \begin{cases} 1 & \text{with prob } \theta \\ 0 & \text{otherwise} \end{cases}$$



$X \sim \text{Categorical } K$

$$X = \{0, 1, \ldots, K-1\}$$

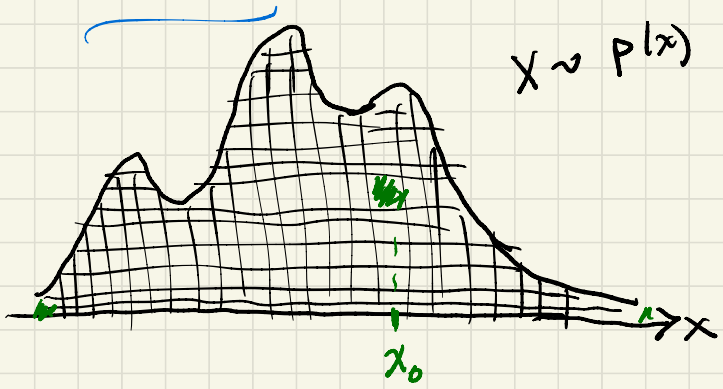
$$P(X = k) = P_k \quad \text{for } k = 0 \ldots K-1$$

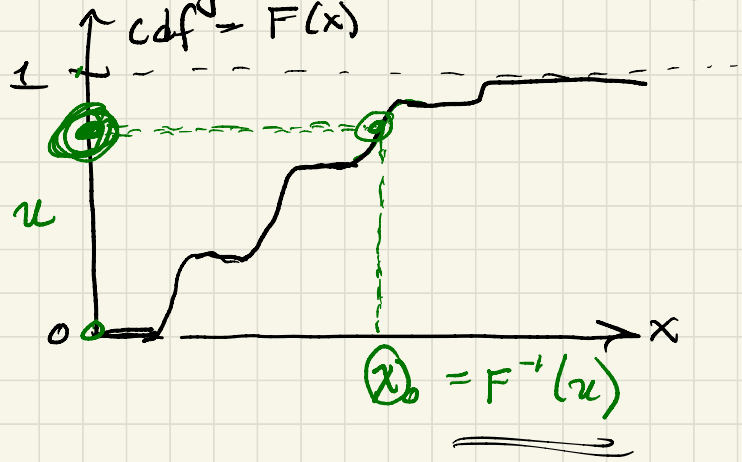


SAMPLING FROM CONTINUOUS DISTRIBUTIONS:

INVERSE TRANSFORM SAMPLING

given draws from $U[0,1]$



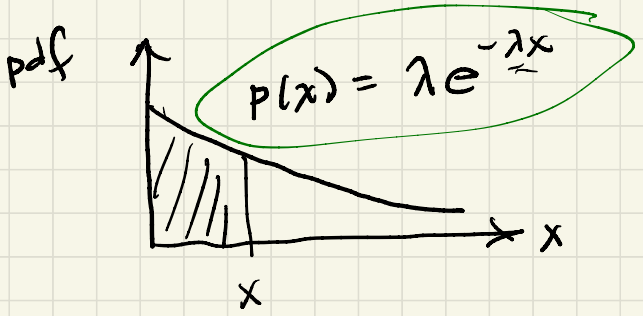


Say X has CDF $F(x)$, then

$$Y = F^{-1}(u)$$

has the same distribution as X.

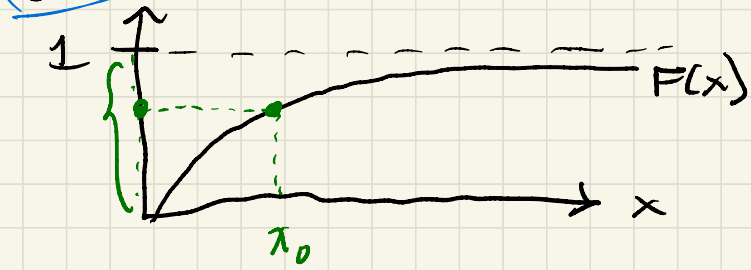
Example: Poisson distribution



$$F(x) = \int_0^x \lambda e^{-\lambda z}\, dz$$

$$= 1 - e^{-\lambda x}$$

CDF



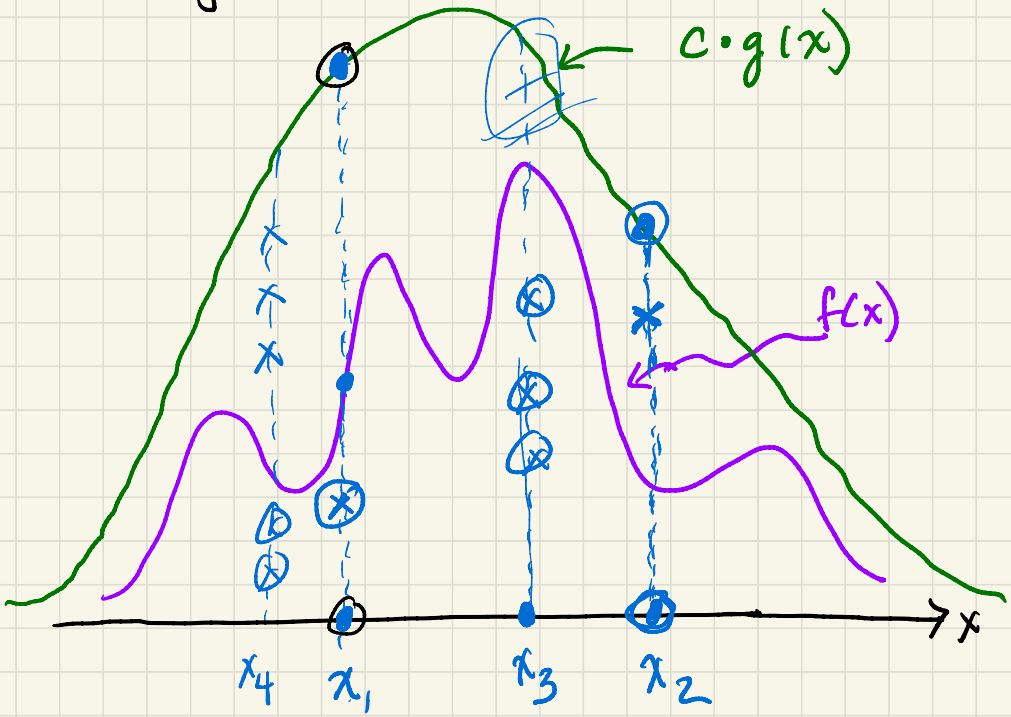
$$F(x) = 1 - e^{-\lambda x} = u$$

$$x = -\frac{1}{\lambda} \ln(1-u)$$

$$x = -\frac{\ln(u)}{\lambda}$$

Rejection Sampling

– use randomness on simple distributions, and make decisions about whether to reject or keep the quantity sampled. The set of things we keep is from the right distribution.



want to sample from $f(x)$ but don't have access to its CDF

But we have access to CDF of $g(x)$, and we know that $c \cdot g(x)$ is an upperbound to $f(x)$

two step process:

(1) draw from $c \cdot g(x)$ using inv. transf. sampling x_i

(2) draw u $[0, c \cdot g(x_i)]$ and accept if $\leq f(x_i)$ reject if $> f(x_i)$