Study guide

The focus of this session is the converse of the previous session.

- In the previous session, we used simple, uniformly distributed random numbers to power our simulations. In this session, we use simulations to generate more complex (and useful) random numbers.
- Some of these simulations lead to well-known probability distributions, while others lead to distributions that are difficult to characterize analytically.
- The words “simulation” and “sampling” are often conflated in the literature since generating a sample from a probability distribution is often effectively a small simulation.

Shonkwiler & Mendivil, Section 2.1

This section covers some cases you have seen before in this course.

- There are non-simulation techniques for drawing samples from some probability distributions. CDF inversion is the most straightforward, but you need to be able to compute the inverse of the CDF efficiently.
- The Bernoulli trial (a biased coin toss) uses the check

  ```
  if uniform(0,1) < p
  ```

  to determine whether the trial results in a success or a failure.
- Choosing one of multiple discrete outcomes (the spinner / roulette wheel from Sayama) uses the cumulative sum of the probabilities of the possible outcomes and a uniformly distributed random number. This is a generalization of the Bernoulli trial
  .

  ```
  probabilities = [0.1, 0.2, 0.3, 0.4]  # four possible outcomes
  cumulative_prob = scipy.cumsum(probabilities)
  sample = cumulative_prob.searchsorted(scipy.random.uniform(0, 1))
  ```

Shonkwiler & Mendivil, Section 2.3

- A sample from the binomial distribution can be seen as a simulation of a number of Bernoulli trials — do n flips of a coin with bias p and count the number of heads.

Shonkwiler & Mendivil, Sections 2.4 and 2.5

- The Poisson and exponential distributions are related.
  - The exponential distribution models waiting times — the duration between two events, for example between one bus arriving and the next bus arriving at a bus stop.
  - The Poisson distribution models rates — the number of events per unit of time, for example the number of buses arriving at a bus stop per hour.
- Section 2.5.1 shows how to generate samples from the exponential distribution using CDF inversion. The CDF of the exponential distribution is

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

  where λ is a parameter of the exponential distribution over waiting time x. The inverse of the CDF is simply

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

  So if we draw a uniformly distributed random number, F, and plug it into the inverse CDF above, x will be exponentially distributed.
- A Poisson sample can be simulated by generating samples from the exponential distribution (waiting times between events) and counting how many events occur per unit interval.
- Section 2.5.2: If you are interested in modeling rates and queues, there is a whole field called queueing theory. You might model the length of a queue at your favorite coffee shop depending on how popular the coffee shop is (arrival rate of customers) and how well they are staffed (rate at which customers get served). It turns out that this type of problem is difficult to analyze in complex situations and simulation is often used to derive results. Queueing theory has application in telecommunications networks (routing and queueing of packets in IP networks), distributed software systems, scheduling, traffic modeling, etc.

Connection to *CS146: Computational Statistics*

- The simulations you have seen so far in this course are essentially generating samples from some really complex probability distributions.
- There is a deep connection with probabilistic inference here, which is why Monte Carlo methods are covered in some detail in CS146.
- The key idea is that inference is the process of estimating model parameters from data, while simulations generate data when given specific parameter settings.
- This idea is formalized in Bayes’ equation

  P(model parameters | simulation results)
  = (1/Z) P(simulation results | model parameters) P(model parameters)

  where
  - P(simulation results | model parameters) is the output from a simulation,
  - P(model parameters | simulation results) is the inference result (known as the posterior distribution),
  - P(model parameters) is the prior distribution,
  - Z is constant with respect to the simulation parameters.