EE365: Probability and Monte Carlo

Notation

- in this course, random variables will take values in a *finite* set $\mathcal{X}$
- we will use multiple styles of notation
- *e.g.*, we switch between linear algebra notation and function notation

Abstract notation

- random variables: $x : \Omega \to \mathcal{X}$
- functions: $f(x)$ is the value of the function $f : \mathcal{X} \to \mathbb{R}$ at the element $x \in \mathcal{X}$
- distributions: $\pi : \mathcal{X} \to \mathbb{R}$ with $\pi(x)$ the probability of outcome $x \in \mathcal{X}$
- expected values: $\mathbf{E}\, f(x) = \sum_{x \in \mathcal{X}} \pi(x) f(x)$

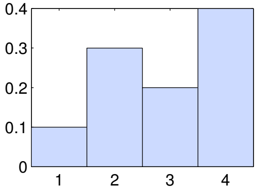
- useful in applications, where elements of $\mathcal{X}$ are *named variables*, possibly taking list, string, or other structured values
- fits with modern programming languages; *e.g.*, dictionaries in Python
- implement operations on $\mathcal{X}$ as iterators

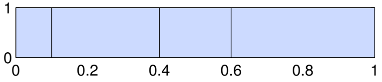
Concrete notation

- enumerate elements of $\mathcal{X}$, so $\mathcal{X} = \{1, 2, \ldots, n\}$
- functions: $f : \mathcal{X} \to \mathbb{R}$ is a *column vector* $f \in \mathbb{R}^n$
- distributions: $\pi : \mathcal{X} \to \mathbb{R}$ is a *row vector* $\pi \in \mathbb{R}^{1 \times n}$
- expected values: $\mathbf{E}\, f(x) = \pi f$
- probability of event $E \subset \mathcal{X}$: $\mathbf{Prob}(E) = \pi 1_E$ where 1_E is the indicator vector

$$(1_E)_i = \begin{cases} 1 & \text{if } i \in E \\ 0 & \text{otherwise} \end{cases}$$

Sampling from a distribution





- for distribution $\pi \in \mathbb{R}^{1\times n}$, *cumulative distribution* $c \in \mathbb{R}^{1\times n}$ is $c_i = \sum_{j=1}^{i} \pi_j$
- example: distribution $\pi = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \end{bmatrix}$
  gives cumulative distribution $c = \begin{bmatrix} 0.1 & 0.4 & 0.6 & 1.0 \end{bmatrix}$
- to generate $x \sim \pi$, pick $u \sim \mathcal{U}[0,1]$, and let $x = \min\{i \mid u \leq c_i\}$
- we denote a sample from distribution π as $\texttt{sample}(\pi)$

Monte Carlo

- a method to *estimate* $e = \mathbf{E}\, f(x)$
- useful when
  - n is too large to explicitly form sum in πf
  - but we have efficient method to generate samples from π and evaluate $f(x)$
- basic Monte Carlo method:
  - generate independent samples $x^{(1)}, \ldots, x^{(N)} \sim \pi$
  - estimate
$$\hat{e} = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)})$$
  - $\hat{e}$ is unbiased estimate of $e = \mathbf{E}\, f(x)$
  - $\mathbf{E}(\hat{e} - e)^2 = \mathbf{var}\, f(x)/N \quad$ ($\mathbf{var}(z)$ is variance of z)

Example

- $x = (b_1, \ldots, b_{50})$, $b_i \in \{0,1\}$ IID with $\mathbf{Prob}(b_i = 1) = 0.4$
- $|\mathcal{X}| = 2^{50}$; too big to enumerate all $\pi(x)$
- let's estimate $\mathbf{Prob}(\sum_{i=1}^{25} b_i \geq 0.6 \sum_{i=1}^{50} b_i)$
  (60% or more of ones appear in first half)
- plot shows MC estimate versus N