MA 6.101
Probability and Statistics

Tejas Bodas

Assistant Professor, IIIT Hyderabad

Moment generating function

- The moment generating function (MGF) of a random variable X is a function $M_X : \mathbb{R} \to [0, \infty]$ defined by $M_X(t) = E[e^{tX}]$.
- If X is discrete, $M_X(t) = \sum_{x \in \Omega'} e^{tx} p_X(x)$.
- If X is continuous, $M_X(t) = \int_{-\infty}^{\infty} e^{tx} f_X(x) dx$.
- Define $D_X := \{t : M_X(t) < \infty\}$. D_X is called the region of convergence (ROC). $t = 0$ is always part of ROC.
- Find MGF of Z where Z is a Bernoulli(p) random variable.

MGF examples

- For $Exp(\lambda)$ variable, $M_X(t) = \frac{\lambda}{\lambda - t}$ for $\lambda < t$.
- For $Z \sim \mathcal{N}(\mu, \sigma^2)$, we have $M_Z(t) = e^{(\mu t + \frac{1}{2}\sigma^2 t^2)}$
- `https://proofwiki.org/wiki/Moment_Generating_Function_of_Gaussian_Distribution`
- HW: Find the MGF for a random variable X that has the following distributions: Binomial(n,p), Normal $\mathcal{N}(0, 1)$, Poisson(λ)

MGF

- If $M_X(t)$ is finite for all $|t| \leq \epsilon$ and for some $\epsilon > 0$ then $M_X(t)$ is infinitely differentiable on $(-\epsilon, \epsilon)$. (Property without proof)
- Let ${M_X}^{(r)}(t) := \frac{d^r}{dt^r} M_X(t)$ (r^{th}-derivative of $M_X(t)$)
- Intuitively, one can see that ${M_X}^{(r)}(t) = E[e^{tX} X^r]$ for all r.
- $E[X^r] = {M_X}^{(r)}(0)$
- HW: Work out these things for $Exp(\lambda)$
- HW: Find MGF for all random variables studied till now

MGF of Sums of independent random variable

- Consider $Z = X + Y$. What is the pdf of Z when X and Y?
- Let $M_X(t)$ and $M_Y(t)$ be their MGF's. What is $M_Z(t)$?
- $M_Z(t) = E[e^{Zt}] = E[e^{(X+Y)t}]$.
- $M_Z(t) = E[e^{Xt}.e^{Yt}]$.
- If X and Y are independent, $E[XY] = E[X]E[Y]$ and $E[g(X)h(Y)] = E[g(X)]E[h(Y)]$.
- $M_Z(t) = E[e^{Xt}].E[e^{Yt}]$.

$$\boxed{M_Z(t) = M_X(t)M_Y(t).}$$

MGF of Sums of independent random variable

- Consider $Z = X + Y$. What is the MGF of Z when X and Y?

$$M_Z(t) = M_X(t)M_Y(t).$$

- What about $M_Z(t)$ when $Z = X_1 + X_2 + \dots X_n$ and X_i are iid.?
- $M_Z(t) = (M_X(t))^n$.
- What about $M_Z(t)$ when $Z = X_1 + X_2 + \dots X_N$ where N is a positive discrete random variable?
- $M_Z(t) = E[e^{tZ}] = E_N[E[e^{tZ}|N]] = E_N((M_X(t))^N)$.
- $M_Z(t) = \sum_n p_N(n) M_X(t)^n$
- HW: Prove that $M_Z(t) = M_N(log M_X(t))$

Agenda for the next two lectures

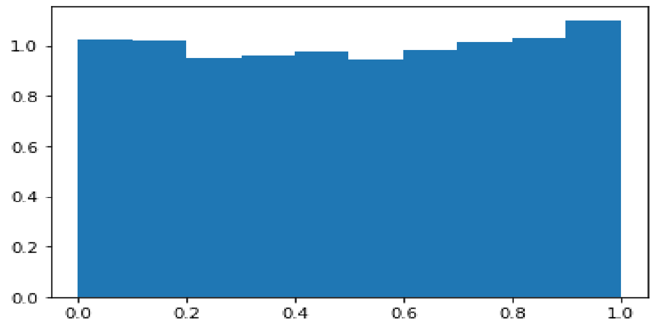
- Intro to Stochastic Simulation
  - We will generate samples from discrete or continuous r.v's using samples from uniform distribution.
- Limit theorems for Convergence of random variables
  - Sure convergence
  - Almost sure convergence & SLLN
  - Convergence in probability
  - Convergence in r^{th} mean
  - Weak Convergence or Convergence in distribution & CLT

Generate samples using uniform distribution

Our aim: Obtain samples from a discrete random variable

- Suppose you have access to samples from a uniform random variable U over support $[0, 1]$.

- ```
  import numpy as np
  import matplotlib.pyplot as plt
  uni_samples = np.random.uniform(0, 1, 5000)
  plt.hist(uni_samples, bins = 10, density = True)
  plt.show()
  ```



- *uni_samples* is a vector of 5000 realizations of uniform random variable U.

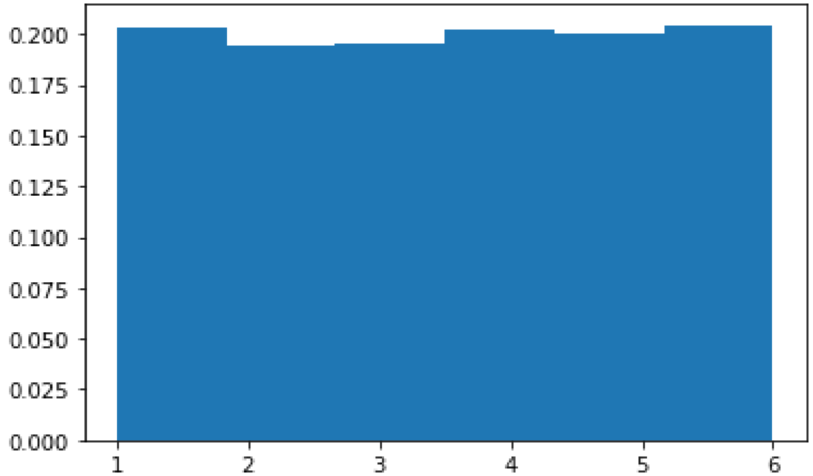
- You can also see it as a realization of $U_1, U_2, \ldots U_{5000}$ i.i.d uniform variables.
  ```

How to simulate a dice using these samples?

- Can you use these 5000 samples and convert them into outcomes of a dice ?

```
 t=0
dice_samples=np.zeros(5000)
for u in uni_samples:
  if u < 1/6:
    dice_sample = 1
  if 1/6 < u < 2/6:
    dice_sample = 2
  if 2/6 < u < 3/6:
    dice_sample = 3
  if 3/6 < u < 4/6:
    dice_sample = 4
  if 4/6 < u < 5/6:
    dice_sample = 5
  if 5/6 < u < 6/6:
    dice_sample = 6
  dice_samples[t] = dice_sample
  t = t+1
plt.hist(dice_samples, bins = 6, density = True)
```

- [0.02, 0.8, 0.6, 0.03]
- [1, 5, 4, 1]



Our aim: Obtain samples from a discrete random variable

- Consider a discrete random variable X with support set $\{x_0, x_1, \ldots\}$ and pmf $p_X(x_j) = p_j$ for $j = 0, 1, \ldots$ such that $\sum_j p_j = 1$.
- Cardinality of the support set of X could be finite or infinite.
- Our aim: Create i.i.d. samples of r.v. X using i.i.d. random samples of U.
- We shall now formally see the inverse transform method to do this.

The inverse transform method

- Aim: We wish to create i.i.d. samples of a discrete r.v. X with $p_X(x_j) = p_j$ using i.i.d. samples of a uniform r.v. U over $[0, 1]$.

- Let $u \in [0, 1]$ be a realization of r.v. U. Then the corresponding sample of X is generated as follows

$$X = \begin{cases} x_0 \text{ if } u < p_0 \\ x_1 \text{ if } p_0 \leq u < p_0 + p_1 \\ x_2 \text{ if } p_0 + p_1 \leq u < p_0 + p_1 + p_2 \\ \vdots \\ x_j \text{ if } \sum_{i=0}^{j-1} p_i \leq u < \sum_{i=0}^{j} p_i \\ \vdots \end{cases}$$

- Why is this method correct? Why call it inverse transform method?

The inverse transform method

- A sample of X is generated using the sample of U as follows

$$X = x_j \quad \text{if} \quad \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i$$

- Now $P(X = x_j) = p_j$ and hence the method is correct.
- Why the name "inverse transform method"?
- Recall that $F_X(x_j) = \sum_{i=0}^{j} p_i$. This implies that
- $$X = x_j \quad \text{if} \quad F_X(x_{j-1}) \leq U < F_X(x_j)$$
- After generating a random number U, we determine the value of X by finding the interval $\left[F_X(x_{j-1}), F_X(x_j)\right)$ in which u lies.
- At a high level, we are performing $X = F_X^{-1}(U)$ but note that F_X is discontinuous so its inverse has to be cleverly defined.

How to generate samples of a continuous random variable

(Using samples of a continuous uniform variable over $[0, 1]$)

Our aim: Obtain samples from a continuous random variable

- Suppose you have access to samples from a uniform random variable U over support $[0, 1]$.(We will not study how to generate such samples.)
- Consider a continuous random variable X with support set $\mathcal{X}$ and let $F_X(x)$ denotes its cdf.
- Support set of X could be arbitrary.
- <u>Our aim:</u> Create i.i.d. samples of r.v. X using i.i.d. samples of U.
- We shall again see the inverse transform method to do this.