Statistical Inference (MA862)

Lecture Slides

Topic 1: Monte Carlo Simulation

Example

- Let we want to estimate the average distance between two randomly selected points in a region.
- Let $\boldsymbol{X} = (X_1,\ X_2)$ and $\boldsymbol{Y} = (Y_1,\ Y_2)$ be independent and uniformly distributed two points drawn from a finite rectangle $R = [0,\ a] \times [0,\ b]$.
- The Euclidean distance between these two points is

$$Z = d(\boldsymbol{X},\ \boldsymbol{Y}) = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2}.$$

- We need $E(Z)$.

$$E(Z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \times f_{X_1,X_2}(x_1,\ x_2) f_{Y_1,\ Y_2}(y_1,\ y_2) dx_1 dx_2 dy_1 dy_2$$

Example

- We can approximate $E(Z)$ by sampling pair points $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i = 1, 2, \ldots, n$, form R and then calculating the average

$$\frac{1}{n}\sum_{i=1}^{n} d(\boldsymbol{X}_i, \boldsymbol{Y}_i).$$

Simple Monte Carlo

- In a simple Monte Carlo problem, we express the quantity we want to know as the expected value of a random variable Y, such as $\mu = E(Y)$.
- Generate values $Y_1, \ldots, Y_n$ independently from the distribution of Y.
- Take their average

$$\widehat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

as an estimate of μ.

Simple Monte Carlo

- In many examples, $Y = h(\boldsymbol{X})$.
- $\boldsymbol{X}$ has a PMF/PDF $p(x)$ (known).
- f is a real-valued function defined over the support of $\boldsymbol{X}$.

Justification of Simple MC

- The strong law of large numbers:

$$\mathbb{P}\left(\lim_{n\to\infty} |\widehat{\mu}_n - \mu| = 0\right) = 1.$$

- Loosely speaking, the SLLNs says that the error in approximation will be very small if we increase n.

Random Number Generation

- Our aim is to generate random number from appropriate distribution.
- Basic step of a random number generation from a distribution is to generate random numbers from Uniform distribution $U(0, 1)$.
- The PDF of a random variable having $U(0, 1)$ distribution is

$$f(x) = \begin{cases} 1 & \text{if } 0 < x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Uniform Random Numbers Generation

- To generate random numbers from a process that according to well established understanding of physics is truly random.
- Radioactive particle emission, that are thought to be truly random.
- Such process has it's own draw back.
- Therefore, people use Pseudo-random numbers (computer generated).
- We will not discuss it in detail, as almost all software has a routine to generate pseudo-random numbers from $U(0, 1)$.
- Now-onward, the pseudo-random number will be referred as random number.

Uniform Random Numbers Generation

- A linear congruence generator (LCG):

$$x_{i+1} = ax_i \bmod m, \quad u_{i+1} = \frac{x_{i+1}}{m}, \quad i = 0, 1, 2, \ldots.$$

- The multiplier a and the modulus m are integer constants.
- The initial value (*seed*) x_0 is called seed.
- x_0 is an integer between 1 and $m - 1$.
- LCG is a deterministic recurrence relation.

Uniform Random Numbers Generation

- The general linear congruence generator (GLCG):

$$x_{i+1} = (ax_i + c) \bmod m, \quad u_{i+1} = \frac{x_{i+1}}{m}, \quad i = 0, 1, 2, \ldots .$$

- a, m and c are appropriate integers.

Non-uniform Random Number Generation

- Method for transforming random numbers from $U(0, 1)$ distribution to samples from other required distributions.
- The two most widely used general techniques are:
  1. Inverse Transform Method.
  2. Acceptance Rejection Method.

Inverse Transform Method

- The inverse transform method is based on the following theorem.

  Theorem 1: Let F be a CDF. Define the quasi-inverse of F by

  $$F^{-1}(u) = \inf\{x \in \mathbb{R} : F(x) \geq u\} \quad \text{for } 0 < u < 1.$$

  Let $U \sim U(0, 1)$ and $X = F^{-1}(U)$. Then, the CDF of X is F.
- $\{x \in \mathbb{R} : F(x) \geq u\}$ is non-empty and has a lower bound for all $u \in (0, 1)$.

Inverse Transform Method

- Want a sample from the CDF $F(x)$. That means that we want to generate a random variable X with the property that $P(X \leq x) = F(x)$ for all $x \in \mathbb{R}$.
- Using above theorem, we have the following algorithm.

Algorithm 1 Inverse Transform Method

1: Generate U from $U(0, 1)$ distribution.
2: Set $X = F^{-1}(U)$.
3: Return X.

- In principle, this algorithm can be used to generate random number from any distribution.
- However, there are computational aspects. We generally use this algorithm if F^{-1} is in closed form and easy to compute.

Inverse Transform Method

Example 1: (Generation from exponential distribution) The PDF is

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Example 2: (Generation from Arc Sin Law) Consider the CDF

$$F(x) = \frac{2}{\pi} \text{ arc sin } \sqrt{x}\ ,\ 0 \leq x \leq 1.$$

Example 3: (Generation from Rayleigh Distribution) The CDF is

$$F(x) = 1 - e^{-2x(x-b)}\ ,\ x \geq b.$$

Inverse Transform Method

Lemma 1: F and F^{-1} both are non-decreasing.
Lemma 2: $F\,F^{-1}(u) \geq u$ for all $u \in (0,\,1)$.
Lemma 3: $F^{-1}\,F(x) \leq x$ for all $x \in \mathbb{R}$.
Lemma 4: For $x \in \mathbb{R}$ and $0 < u < 1$, $F(x) \geq u$ if and only if $F^{-1}(u) \leq x$.

Inverse Transform Method

Example 4: Generation from a Bernoulli distribution with probability of success p $(q = 1 - p)$.

$$F_X(x) = \begin{cases} 0 & \text{if } x < 0 \\ q & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x \geq 1. \end{cases} \quad \text{and} \quad F^{-1}(u) = \begin{cases} 0 & \text{if } 0 < u < q \\ 1 & \text{if } q \leq u < 1. \end{cases}$$

Algorithm 2 Generation from *Bernoulli*(p)

generate U from $U(0,1)$.
if $U < 1 - p$ **then**
    Set $X \leftarrow 0$.
else
    Set $X \leftarrow 1$.
end if
return X

Inverse Transform Method

Example 5: Generation from a discrete distribution with finite support.

- Consider a DRV X whose support is $c_1 < c_2 < c_3 < \cdots < c_N$.
- Let $p_i = P(X = c_i)$, $i = 1, 2, 3 \ldots, N$.
- Set $q_0 = 0$ and $q_i = \sum_{j=1}^{i} p_j$, $i = 1, 2, 3 \ldots, N$.

Algorithm 3 Inversion Transformation Method for Discrete Random Variable with Finite Support

1: Generate a uniform $U \sim U(0, 1)$.
2: Find $K \in \{1, 2, \ldots, N\}$ such that $q_{K-1} < U \leq q_K$.
3: Return c_K.

Acceptance-Rejection Method

- We want to generate random number from a PDF f (target distribution).
- Let g (candidate distribution) be a PDF such that for all $x \in \mathbb{R}$ and for some $c \geq 1$

$$f(x) \leq cg(x).$$

- The technique to generate random number from g is known.
- Then we can use the following algorithm to generate random number from f.

Algorithm 4 Acceptance Rejection Method

```
repeat
    generate X from distribution g.
    generate U from U(0,1).
until U ≤ f(X)/(cg(X))
return X
```

Generation from Gamma Distribution

Example 6: Generation random number from *Gamma*(α, β).

- The PDF of the distribution is

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \quad \text{for } x > 0.$$

- We will consider generation from *Gamma*$(\alpha, 1)$ for $\alpha > 0$.

Generation from Gamma Distribution

Case I : $0 < \alpha < 1$.

- Take
$$g(x) = \begin{cases} \frac{x^{\alpha-1}}{A} & \text{if } 0 < x < 1 \\ \frac{e^{-x}}{A} & \text{if } x \geq 1, \end{cases}$$
where $A = \frac{1}{\alpha} + \frac{1}{e}$.
- Then, $f(x) \leq cg(x)$, where $c = \frac{A}{\Gamma(\alpha)}$.
- The CDF corresponding to g is
$$G(x) = \begin{cases} \frac{x^{\alpha}}{\alpha A} & \text{if } 0 < x < 1 \\ 1 - \frac{e^{-x}}{A} & \text{if } x \geq 1. \end{cases}$$
- Now,
$$G^{-1}(u) = \begin{cases} (\alpha A u)^{\frac{1}{\alpha}} & \text{if } 0 < u < \frac{1}{\alpha A} \\ -\ln A - \ln(1-u) & \text{if } \frac{1}{\alpha A} \leq u < 1. \end{cases}$$

Generation from Gamma Distribution

Algorithm 5 Generation from *Gamma*$(\alpha, 1)$ for $0 < \alpha < 1$

repeat
  generate U_1 from $U(0, 1)$
  if $U < \frac{1}{\alpha A}$ **then**
    Set $X \leftarrow (\alpha A U)^{\frac{1}{\alpha}}$
  else
    Set $X \leftarrow -\ln A - \ln(1 - U)$
  end if
  generate U_2 from $U(0, 1)$
until $cg(X)U_2 \leq f(X)$
return X

Generation from Gamma Distribution

Case II: α is a positive integer.

- Let $X_i \overset{i.i.d.}{\sim} Exp(1)$ for $i = 1, 2, \ldots, n$.
- Then $\sum_{i=1}^{n} X_i \sim Gamma(n, 1)$.

Algorithm 6 Generation from $Gamma(\alpha, 1)$, α is a positive integer

```
Set n ← α and Y ← 0
while n ≠ 0 do
    generate U from U(0, 1)
    Set X ← −ln(U)
    Y ← Y + X
    n ← n − 1
end while
return Y
```

Generation from Gamma Distribution

Case III: $\alpha > 1$ and not an integer.

- Let $X \sim Gamma(\alpha_1, \beta)$ and $Y \sim Gamma(\alpha_2, \beta)$.
- Suppose that X and Y are independent.
- Then $X + Y \sim Gamma(\alpha_1 + \alpha_2, \beta)$.
- $\lfloor x \rfloor$: the integer part of the positive real number x.
- $\{x\}$: the fractional part of the positive real number x.
- $\alpha = \lfloor \alpha \rfloor + \{\alpha\}$.

Algorithm 7 Generation from $Gamma(\alpha, 1)$ when $\alpha > 1$ and α is not an integer

1: generate Y from $Gamma(\{\alpha\}, 1)$ using Algorithm 5
2: generate X from $Gamma(\lfloor \alpha \rfloor, 1)$ using Algorithm 6
3: $Z = X + Y$
4: **return** Z

Acceptance-Rejection Method

Theorem 2: Let f and g be two PDFs such that

$$f(x) \leq cg(x) \quad \text{for all } x \in \mathbb{R} \text{ and for some } c \geq 1.$$

Then X generated by Algorithm 4 has PDF f.

Technique based on Transformation

Example 7: Generation of random number from $Beta(\alpha, \beta)$.

- $X \sim Gamma(\alpha_1, \beta)$
- $Y \sim Gamma(\alpha_2, \beta)$
- X and Y are independent
- Then $\dfrac{X}{X+Y} \sim Beta(\alpha_1, \alpha_2)$.

Algorithm 8 Generation from $Beta(\alpha_1, \alpha_2)$ distribution

1: generate X from $Gamma(\alpha_1, \beta)$
2: generate Y from $Gamma(\alpha_2, \beta)$
3: $Z = \dfrac{X}{X+Y}$
4: **return** Z

Technique based on Transformation

Example 8: Generation of random number from $N(\mu, \sigma^2)$

- $U_1, U_2 \overset{i.i.d.}{\sim} U(0, 1)$
- Define

$$Z_1 = \sqrt{-2\ln U_1}\cos(2\pi U_2) \quad \text{and} \quad Z_2 = \sqrt{-2\ln U_1}\sin(2\pi U_2).$$

- Then $Z_1, Z_2 \overset{i.i.d.}{\sim} N(0, 1)$
- This transformation is called Box-Muller transformation.

Algorithm 9 Box-Muller Method to generate from $N(0, 1)$

```
1: generate U1 and U2 from U(0, 1)
2: R ← sqrt(−2 ln U1)
3: θ ← 2πU2
4: Z1 ← R cos(θ)
5: Z2 ← R sin(θ)
6: return (Z1, Z2).
```

Technique based on Transformation

Example 9: Generation from *Geometric*(p).

- The PMF is given by

$$P(X = i) = p(1-p)^i \quad \text{for } i = 0, 1, 2, \ldots.$$

- Let Y be an exponential random variable with mean $\frac{1}{\lambda}$.
- Take $W = \lfloor Y \rfloor$. Then

$$P(W = i) = e^{-i\lambda}\left(1 - e^{-\lambda}\right) \quad \text{for } i = 0, 1, 2, \ldots.$$

- $W \sim Geometric(1 - e^{-\lambda})$.

Algorithm 10 Generation from *Geometric*(p)

1: generate U from $U(0, 1)$
2: $X \leftarrow \left\lfloor \frac{\ln U}{\ln(1-p)} \right\rfloor$
3: **return** X.

Error in Monte Carlo Integration

- Both laws of large numbers tell us that Monte Carlo will eventually produce an error as small as we like.
- They do not tell us how large n has to be for this to happen.
- They also do not say for a given sample $Y_1, \ldots, Y_n$ whether the error is likely to be small.
- The situation improves markedly when Y has a finite variance.
- Suppose that $Var(Y) = \sigma^2 < \infty$.
- The mean of $\widehat{\mu}_n$ is $\mathbb{E}(\widehat{\mu}_n) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(Y_i) = \mu$.
- The variance of $\widehat{\mu}_n$ is $Var(\widehat{\mu}_n) = \mathbb{E}\left((\widehat{\mu}_n - \mu)^2\right) = \frac{\sigma^2}{n}$.
- The variance of $\widehat{\mu}$ may be considered as (average) error in $\widehat{\mu}$ for approximating or estimating μ.
- For fixed n, the variance increases as σ increases.
- The variance decreases as n increases for fixed σ.
- The root mean squared error of $\widehat{\mu}_n$ is $\sqrt{Var(\widehat{\mu}_n)} = \frac{\sigma}{\sqrt{n}}$.