Computational Statistics, M1 MAS DS, Aix-Marseille University

Houssam BOUKHECHAM

December 15, 2024

Contents

1 Random variable simulation

[1, 2, 3]

1.1 Transformation methods

Lemma 1. *Let $F : \mathbb{R} \to [0,1]$ be an non-decreasing function. If a random variable X has F as its cumulative distribution function (CDF), then the random variable $U = F(X) \sim U(0,1)$.*

Proof. □

Example 1 (Normal variable generation)**.** *The cumulative distribution function (CDF) of a Gaussian random variable with mean μ and standard deviation σ is given by:*

$$F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \, dt. \tag{1}$$

*The function F is a diffeomorphism. Assuming $\mu = 0$ and $\sigma = 1$, an approximation F_a^{-1} of the inverse function F^{-1} can be computed to arbitrary precision ([**?**], Example 2.6). To generate a random sample of size n from the standard Gaussian distribution, we first generate n random samples from the uniform distribution, $\{u_1, u_2, \ldots, u_n\}$. Then, we map this sample to the Gaussian distribution using the inverse approximation :*

$$\left\{F_a^{-1}(u_1), F_a^{-1}(u_2), \ldots, F_a^{-1}(u_n)\right\}.$$

Exercise 1.

1. *Generate a sample $\mathcal{S} = \{u_1, u_2, \cdots, u_n\}$ of size $n = 500$ from the uniform distribution.*

2. *Implement the function*

$$F_a^{-1}(u) = t - \frac{a_0 + a_1 t}{1 + b_1 t + b_2 t^2}, \quad u \in (0,1),$$

   where $t^2 = \log\left(u^{-2}\right)$ and $a_0 = 2.30753, a_1 = 0.27061, b_1 = 0.99229, b_2 = 0.04481$

3. *Plot the histogram of the set $F_a^{-1}(\mathcal{S})$ and comment the results.*

4. ♣ *Repeat the process for a larger value of n (e.g., $n = 50000$). Compare the generated sample with a standard Gaussian random variable generator and comment the results.*

1.2 Accept-reject method

In the previous subsection, we discussed how to generate Gaussian and exponential distributions from a uniform distribution, but in doing so, we needed an approximation of the inverse of the CDF. In general, the analytic form of the inverse of the CDF is not available, and even if the analytic form exists, approximation of this function may be computationally expensive. An alternative to transformation methods, is the *Accept-Reject* method. Roughly speaking, this method is a technique for generating a sample from a target distribution with density f, when direct sampling from it is not possible. Instead, we use a proposal distribution with density g, from which we generate x, and depending on the values of $g(x)$ and $f(x)$, we either accept x as a sample of f or reject it. More precisely if f has a compact support and is bounded then we have

Theorem 1 (Fundamental theorem of simulation)**.** *Let f be a target density. Then simulating $X \sim f$ is equivalent to simulating*

$$(X, U) \sim \mathcal{U}\left\{(x, u) \ : \ 0 < u < f(x)\right\}. \tag{2}$$

To simulate $X \sim f$, we first determine the maximum value m of f. Next, we generate a pair (x, u) from a uniform distribution over the rectangle $[a, b] \times [0, m]$, where $[a, b]$ is the support of f. The value x is accepted if $u \leq f(x)$; otherwise, it is rejected.

Example 2. ...

Corollary 1. ...

Using Theorem 1, the Accept-Reject method can be done as follows

Algorithm 1 Accept-Reject

```
1: Choose $M \geq 1$ such that $f(x) \leq Mg(x)$ for all $x$.
2:  Generate $X \sim g$ and $u \sim \mathcal{U}[0, 1]$.
3: If $u \leq \frac{f(X)}{Mg(X)}$, accept $Y = X$ and **return** $Y$.
4: Otherwise, go back to step 2.
```

2 Importance sampling

```r
set.seed(12)

# Define the target distribution (Gamma distribution)
target_dist <- function(x, shape = 2, rate = 1) {
  ifelse(x > 0, x^(shape - 1) * exp(-rate * x), 0)
}

# Define the proposal distribution (Normal distribution)
proposal_dist <- function(x) {
  dnorm(x, mean = 2, sd = 2)  # Normal(2, 2) PDF
}
```

```r
# Generate samples from the proposal distribution
n_samples <- 2000

# proposal sample
ps <- rnorm(n_samples, mean = 2, sd = 2)

# Compute the weights for each sample
weights <- target_dist(ps) / proposal_dist(ps)

# Estimate the mean of the target distribution using IS
importance_sampling_mean <- sum(weights * ps) / sum(weights)
print(importance_sampling_mean)

## [1] 2.011539
```

3 Bootstrap

Exercise 2. *Let $S = \{x_1, x_2, \ldots, x_n\}$ be a random sample from the uniform distribution on the interval $(0, \theta)$. Assume we want to estimate the unknown parameter θ, so we use the estimator $X_{(n)} = \max x_i$.*

1. *If $X_1, X_2, \cdots, X_n$ are iid with uniform distribution on $(0, \theta)$, what is the distribution of the random variable $X_{(n)} = \max X_i$?*

   (Hint: Determine the cumulative distribution function of $X_{(n)}$)

2. *Is the estimator $X_{(n)}$ biased?*

3. *How would you use the bootstrap to estimate the bias in $X_{(n)}$ for θ?*

4 Monte Carlo methods

5 Gibbs algorithms for bayesian statistics

5.1 Metropolis Hastings algorithm

5.2 MCMC

test

References

[1] Robert Casella and Roger L. Berger. *Monte Carlo Statistical Methods.* Springer, New York, 2nd edition, 1999.

[2] James E Gentle. *Computational statistics*, volume 308. Springer, 2009.

[3] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.