Monte Carlo Methods for Inference

Max Turgeon

STAT 3150–Statistical Computing

Lecture Objectives

- Recall the definition of a statistic and its sampling distribution
- Learn how simulations can be used for estimation
- Learn how simulations can be used for hypothesis testing

Motivation

- Over the last few weeks, we talked about generating random variables and use this to estimate integrals.
- This week, we will investigate how these ideas can be used for data analysis.
    - How to use simulations to estimate population parameters.
    - How to use simulations to perform hypothesis testing. (See interactive tutorial)

Definitions i

First, recall the following definitions:

- A **statistic** is a function of a sample, i.e. from a sample $X_1, \ldots, X_n$ compute an output.
    - Sample mean, sample variance, etc.
    - Histogram, empirical CDF
- An **estimator** is a statistic $\hat{\theta}$ is a statistic used to estimate (or "approximate") a population parameter θ.
    - The sample mean estimates the population mean
    - The empirical CDF approximates the population CDF.

Definitions ii

- A statistic is a random variable, because it is a function of the sample. Therefore it has a distribution: the **sampling distribution**.
  - If $X_1, \dots, X_n$ are $N(\mu, \sigma^2)$, then the sampling distribution for the sample mean is $N(\mu, \sigma^2/n)$

Remark

- The sampling distribution is often a function of unknown population parameters.
    - Or even the type of distribution may be unknown.
- Monte Carlo methods can be used to estimate the sampling distribution and derive quantities of interest.
    - E.g. Mean Squared Error, percentiles.

Example: 538's The Riddler i

- Refer to this post: `https://fivethirtyeight.com/features/can-you-parallel-park-your-car/`
- The population parameter we want to estimate is $P(\text{Have to parallel park})$.
- A sample is an arrangement of four cars in six parking spots, with each arrangement equally likely.
- From a sample, we can determine if the Riddler will have to parallel park or not.
    - Our statistic T is binary: *Yes* or *No*.

Example: 538's The Riddler ii

- This can be modeled using a Bernoulli distribution with parameter $p = P(T = \text{Yes})$.
    - Recall, this is the **sampling distribution**.
- To estimate p, we can simulate $B = 1000$ samples, compute T for each sample, and count the proportion $\hat{p}$ of samples for which $T = \text{Yes}$.
    - This is Monte Carlo integration!
- The estimate of the variance of T is $\hat{p}(1 - \hat{p})$, and therefore our standard error for our estimate $\hat{p}$ is

$$se(\hat{p}) = \sqrt{\frac{\hat{p}(1 - \hat{p})}{B}}.$$

Example i

- Assume we have a sample of size 2 from a standard normal distribution: X_1, X_2.
- We want to estimate the expected value of their absolute difference:

$$g(X_1, X_2) = |X_1 - X_2|.$$

- **How can we do this?** Monte Carlo integration!

Example ii

```r
B <- 1989
norm_vars1 <- rnorm(B)
norm_vars2 <- rnorm(B)
# Compute statistic
gvars <- abs(norm_vars1 - norm_vars2)
mean(gvars)
```

```
## [1] 1.124522
```

```r
sd(gvars)/sqrt(B)
```

```
## [1] 0.01919893
```

- Suppose we want to use an estimator $\hat{\theta}$ to estimate a parameter θ.
- Recall $\hat{\theta}$ is a random variable with a distribution. We say the estimator $\hat{\theta}$ is **unbiased** if its expected value is θ:

$$E(\hat{\theta}) = \theta.$$

- We can study the (un)biasedness of $\hat{\theta}$ by using the **mean squared error** (MSE):

$$MSE(\hat{\theta}) = E\left[\left(\hat{\theta} - \theta\right)^2\right].$$

Mean squared error ii

- **Why?** The MSE is related to the variance and the bias of $\hat{\theta}$:

$$MSE(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + \left(E(\hat{\theta}) - \theta\right)^2.$$

- This relates to what is called the **variance-bias tradeoff**:
    - For a fixed MSE, lower bias implies higher variance and vice-versa.

Example i

- The sample mean is an unbiased estimate of the population mean.
- However, it can be sensitive to outliers.

```r
mean(c(1,5,2,8, 4))
```

```
## [1] 4
```

```r
mean(c(1,5,2,8, 100))
```

```
## [1] 23.2
```

Example ii

- An estimator of the mean that is *less* sensitive to outliers is the **trimmed mean**.
- The idea is to remove the extreme values from the sample before taking the mean.
- More precisely: let $X_1, \dots, X_n$ be a random sample, and let $k < 0.5n$ be a positive integer.
- The k-th level trimmed mean is defined as:

$$\bar{X}_{[k]} = \frac{1}{n - 2k} \sum_{i=k+1}^{n-k} X_{(i)},$$

where $X_{(i)}$ is the i-th order statistic.

Example iii

```r
# Generate a standard normal
# sample of size 4
(norm_vars <- rnorm(4))
```

```
## [1] 0.8435305 0.5441357 0.8464267 0.2005197
```

```r
# Sort it
(norm_vars <- sort(norm_vars))
```

```
## [1] 0.2005197 0.5441357 0.8435305 0.8464267
```

Example iv

```r
# Compute 1st level trimmed mean
mean(norm_vars[c(-1, -4)])
```

```
## [1] 0.6938331
```

```r
# Compare to sample mean
mean(norm_vars)
```

```
## [1] 0.6086532
```

- We can generate a sample of size $n = 20$ and compare the MSE of the sample mean with the 1st-level trimmed mean.

Example v

```r
n <- 20
results <- replicate(3150, {
  norm_vars <- sort(rnorm(n))

c("TM" = mean(norm_vars[c(-1, -n)]),
  "SM" = mean(norm_vars))
})
```

```r
# Bias
rowMeans(results) - 0
```

Example vi

```
##          TM           SM
## -0.004406591 -0.004092315
```

```r
# MSE
rowMeans((results - 0)^2)
```

```
##         TM         SM
## 0.05247055 0.05130009
```

- There isn’t any outliers, so we get similar results for both types of means.

Example vii

- Let's introduce outliers through a *contaminated normal* distribution:

$$X \sim pN(0,1) + (1-p)N(0,100).$$

- In other words, X follows a mixture distribution.
    - The second component, $N(0,100)$, is responsible for the outliers in the sample.

Example viii

```
p <- 0.9
n <- 20; B <- 2209

results <- replicate(B, {
  sigmas <- sample(c(1, 10), n, replace = TRUE,
                   prob = c(p, 1 - p))
  contnorm_vars <- rnorm(n, sd = sigmas)
  contnorm_vars <- sort(contnorm_vars)
  c("TM" = mean(contnorm_vars[c(-1, -n)]),
    "SM" = mean(contnorm_vars))
})
```

Example ix

```r
# Bias
rowMeans(results) - 0
```

```
##         TM         SM
## 0.01658554 0.02571665
```

```r
# MSE
rowMeans((results - 0)^2)
```

```
##        TM        SM
## 0.2107086 0.5694208
```

Example x

- As we can see, the trimmed mean has lower bias.
- It also has a lower MSE than the sample mean.
- **Conclusion**: With finite samples, we can sometimes find more efficient estimates of the mean.