Die Monte Carlo (MC) Methode

9. März 2004

1 Die Idee

Sei $F(x)$ eine beliebige Verteilungsfunktion und es existiere der Erwartungswert einer Funktion $g(X)$, d.h. $\mathrm{E}(g(X)) = \int g(x)dF(x) < \infty$. Dann gilt für $X^{(1)}, \ldots, X^{(R)} \overset{iid}{\sim} F(x)$ (Starkes Gesetz der großen Zahlen)

$$\widehat{\mathrm{E}}_{\mathrm{MC}}(g(X)) = \frac{1}{R}\sum_{r=1}^{R} g(X^{(r)}) \overset{f.s.}{\to} \mathrm{E}(g(X)).$$

Die große Zahl R nennt man hierbei *Replikationszahl*. Es sind daher ausreichend viele (künstlich erzeugte) Zufallszahlen $X^{(r)}$, $r = 1, \ldots, R$, aus der Verteilung von X zu generieren, darauf jeweils die Funktion $g(X^{(r)})$ anzuwenden, und von all diesen das arithmetische Mittel zu berechnen. Dieser Monte-Carlo Schätzer $\widehat{\mathrm{E}}_{\mathrm{MC}}(g(X))$ ist selbst eine Zufallsvariable, die für $R \to \infty$ fast sicher gegen den gesuchten wahren Erwartungswert $\mathrm{E}(g(X))$ strebt.

Allgemeiner MC Algorithmus in `R`:

```
 n <- 1     # sample size, dim(X)=1
 R <- 1000  # number of replications
 z <- 1:R   # initialize z as a list of R elements
 for (r in 1:R) {
   x <- rF(n, par)
   z[r] <- g(x)
}
mean(z)
```

Verfügbare Zufallszahlengeneratoren `rF` in `R` mit Defaultwerten für die Parameter:

- $\mathrm{N}(\mu, \sigma^2)$: `rnorm(n, mean=0, sd=1)`
- $\mathrm{Uniform}(\min, \max)$: `runif(n, min=0, max=1)`
- $\mathrm{Beta}(a, b)$: `rbeta(n, a, b)`
- $\mathrm{Binom}(s, p)$: `rbinom(n, size, prob)`
- $\mathrm{Cauchy}(\alpha, \sigma)$: `rcauchy(n, loc=0, scale=1)`
- $\chi^2(df, ncp)$: `rchisq(n, df, ncp = 0)`, (entspricht $\mathrm{Gamma}(df/2, 1/2)$)
- $\mathrm{Exp}(rate)$: `rexp(n, rate=1)`
- $\mathrm{F}(n_1, n_2)$: `rf(n, df1, df2)`
- $\mathrm{Gamma}(a, s)$: `rgamma(n, shape, rate=1, scale=1/rate)`
- $\mathrm{Geom}(p)$: `rgeom(n, prob)`

- Hyper(m, n, k): `rhyper(nn, m, n, k)`
- LogN(μ, σ^2): `rlnorm(n, meanl=0, sdl=1)`
- Logistic(μ, σ^2): `rlogis(n, loc=0, scale=1)`
- NegBinom(s, p): `rnbinom(n, size, prob, mu)`
- Poisson(λ): `rpois(n, lambda)`
- t(df): `rt(n, df)`
- Weibull(a, b): `rweibull(n, shape, scale=1)`

Darüberhinaus bietet `R` auch Funktionen zur Berechnung der Dichte (`dF`), Verteilungsfunktion (`pF`) und Quantilsfunktion (`qF`), wobei `F` so wie zuvor bei den Generatoren definiert ist.

Anwendung: Sei X_n eine n-elementige Zufalls-Stichprobe aus F. Untersucht werden sollen die Varianzen von $\overline{X}_n$ und $\tilde{X}_n$ für endliche (speziell für kleine) $n < \infty$. Zu berechnen sind also der MC Schätzer für $\mathrm{var}(\tilde{X}_n)$, $\mathrm{var}(\overline{X}_n)$, und zusätzlich noch für die Asymptotische Relative Effizienz

$$\mathrm{are}(\overline{X}_n, \tilde{X}_n) = \frac{\mathrm{var}(\tilde{X}_n)}{\mathrm{var}(\overline{X}_n)}\,.$$

Z.B. anhand von $\tilde{X}_n$:

$$\mathrm{var}(\tilde{X}_n) = \int (x - \mathrm{E}(\tilde{X}_n))^2 dF_{\tilde{X}_n}(x)\,.$$

Was auch immer die exakte Verteilungsfunktion $F_{\tilde{X}_n}$ des empirischen Medians einer n-elementigen Zufalls-Stichprobe aus F sein mag, wir benötigen *nur* recht viele Replikationen von $\tilde{X}_n$ aus $F_{\tilde{X}_n}$. Seien diese $\tilde{X}_n^{(1)}, \ldots, \tilde{X}_n^{(R)} \overset{iid}{\sim} F_{\tilde{X}_n}$, dann gilt

$$\widehat{\mathrm{var}}_{\mathrm{MC}}(\tilde{X}_n) = \frac{1}{R}\sum_{r=1}^{R}\left(\tilde{X}_n^{(r)} - \widehat{\mathrm{E}}_{\mathrm{MC}}(\tilde{X}_n)\right)^2 \overset{f.s.}{\to} \mathrm{var}(\tilde{X}_n)$$

mit

$$\widehat{\mathrm{E}}_{\mathrm{MC}}(\tilde{X}_n) = \frac{1}{R}\sum_{r=1}^{R}\tilde{X}_n^{(r)} \overset{f.s.}{\to} \mathrm{E}(\tilde{X}_n)\,.$$

Die Replikationszahl R soll dabei so groß gewählt sein, dass der MC Schätzer (der von R abhängt) stabil ist. Als Faustregel verwendet man zumindest $100 < R < 1000$ für Momente und $R > 1000$ für Quantile x_α. Je größer oder kleiner das Niveau des Quantils α ist, d.h. je näher α bei 0 oder 1 liegt, desto größer muss R gewählt werden.

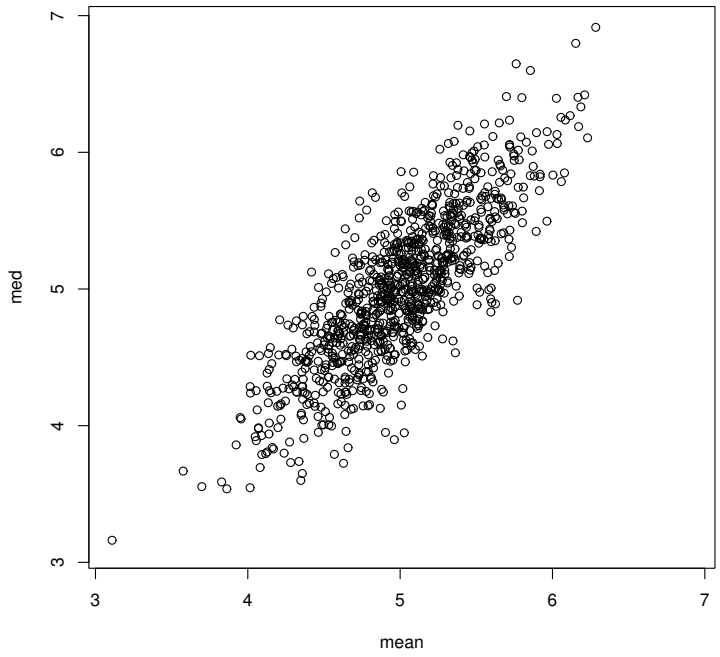
MC Algorithmus in `R`:

```
n <- 20      # sample size, dim(X_n)=n
R <- 1000    # number of replications
med <- 1:R   # initializations
mean <- 1:R
for (r in 1:R) {
  x <- rF(n, par)
  med[r] <- median(x)
  mean[r] <- mean(x)
}
areMC <- var(med) / var(mean)
```

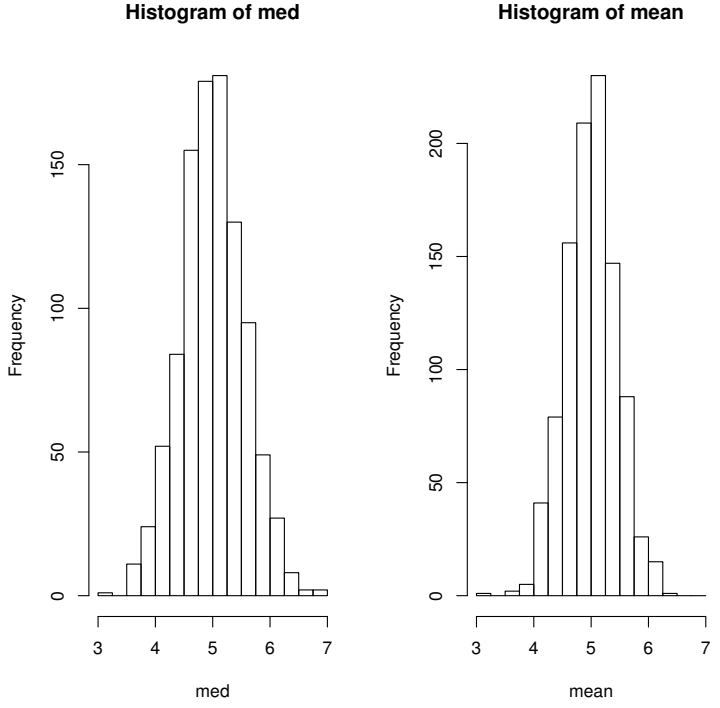
Zur Erinnerung gilt asymptotisch für beliebig normalverteilte Zufalls-Stichproben $\lim_{n\to\infty} \mathrm{are}(\overline{X}_n, \tilde{X}_n) = \pi/2$. Der MC Schätzer erlaubt auch speziell für kleine Werte von n eine Aussage.

Verwendet man für `rF(n, par)` speziell `rnorm(n, 5, 2)`, so führt dies (zufälligerweise) zu Monte-Carlo Realisationen $(\tilde{x}_n^{(r)}, \overline{x}_n^{(r)})$ mit folgender Struktur:

```
plot(mean, med)
```




```
par(mfrow=c(1, 2)) # 2 plots in 1 row
hist(med, xlim=c(3, 7)); hist(mean, xlim=c(3, 7))
```



Die MC Methode kann auch zur **Überprüfung der Überdeckungswahrscheinlichkeit** $1-\alpha$ eines Konfidenzintervalls für den Parameter θ genutzt werden. Sei dazu $(L^{(r)}, U^{(r)})$, $r = 1, \ldots, R$, eine Folge unabhängig ident-verteilter Konfidenzintervalle für den Parameter θ zum Niveau $1-\alpha$. Diese können generiert werden, indem R mal eine Zufalls-Stichprobe mit Umfang n aus $F(\theta_0)$ erzeugt wird (θ_0 ist dabei der wahre Parameter θ), und darauf basierend die r-te Realisation des Konfidenzintervalls berechnet wird. Dann gilt

$$1 - \widehat{\alpha}_{\mathrm{MC}} = \frac{1}{R} \sum_{r=1}^{R} I_{[L^{(r)}, U^{(r)}]}(\theta_0) \overset{f.s.}{\to} 1 - \alpha \,.$$

Für eine Zufalls-Stichprobe vom Umfang n aus der $N(\mu, \sigma^2)$-Verteilung (mit σ^2 bekannt) liefert das zweiseitige Konfidenzintervall für μ

$$\overline{X}_n \pm z_{1-\alpha}\sigma/\sqrt{n}$$

bekanterweise eine Überdeckungswahrscheinlichkeit von $1-\alpha$.

```
n <- 20      # sample size
R <- 1000    # number of replications
mu <- 5; sigma <- 2 # true parameter(s)
alpha <- 0.05       # 1 - coverage probability
L <- U <- 1:R       # initializations
a <- sigma/sqrt(n) * qnorm(1 - alpha/2)
for (r in 1:R) {
  m <- mean(rnorm(n, mu, sigma))
  L[r] <- m - a;   U[r] <- m + a
}
left  <- as.numeric(mu < L); sum(left)
[1] 27
right <- as.numeric(U < mu); sum(right)
[1] 25
```

Es liegt hier der wahre Parameter ($\mu_0 = 5$) 27 mal unter den unteren Intervallsgrenzen, sowie 25 mal über den oberen Grenzen, d.h. in 52 (von 1000) Fällen wird der wahre Parameter μ_0 nicht von den Monte-Carlo Konfidenzintervallen überdeckt, was einem MC Schätzer $\widehat{\alpha}_{\mathrm{MC}} = 0.052$ (bei vorgegebenem $\alpha = 0.05$) entspricht.

2 Der Bootstrap

Bis jetzt wurde immer ein vollständig spezifiziertes Verteilungsmodell $F(\theta)$ für die Monte-Carlo Simulationen angenommen und kein Bezug zu einer konkreten Datensituation gemacht. Sei nun $X_1, \ldots, X_n$ eine Stichprobe aus einer uns unbekannten Verteilungsfunktion F. Wir kennen F nicht, haben aber daraus gerade eine Stichprobe vom Umfang n vorliegen. Beim Bootstrap wird nun die Stichprobeninformation auf zweierlei Art verwendet.
Beim **parametrischen Bootstrap** wird wie zuvor eine Verteilung $F(\theta)$ für die Stichprobe angenommen. Die Parameter selbst werden hierbei aber durch die entsprechenden *Schätzer* $\hat{\theta}$ aus der Stichprobe ersetzt. Nimmt man beispielsweise an, dass $X_1, \ldots, X_n \sim N(\mu, \sigma^2)$ gilt, so basiert der parametrische Bootstrap auf die generierte Stichprobe (eine Replikation) $X_1^*, \ldots, X_n^*$ mit $X_i^* \sim N(\overline{x}, s^2)$.
Der **nicht-parametrische Bootstrap** verzichtet gänzlich auf eine derartige Verteilungs-Annahme und verwendet die *empirische Verteilungsfunktion* der Stichprobe als Schätzer für die unbekannte Verteilung der Population. Die Replikation kommt somit aus $\hat{F}_n$. Realisiert wird dieses Verfahren, indem n mal mit Zurücklegen $X_1^*, \ldots, X_n^*$ aus der Realisierung $x_1, \ldots, x_n$ gezogen wird.
Beide Bootstrap-Ansätze basieren auf eine Stichprobe $X_1^*, \ldots, X_n^*$ aus der geschätzten Verteilungsfunktion. Ist man zum Beispiel an der Schätzung der Varianz des Medians der originalen Stichprobe interessiert, also an $\text{var}(\tilde{X}|X_i \sim F)$, so liefert der entsprechende Bootstrap-Schätzer entweder $\text{var}(\tilde{X}^*|X_i^* \sim F(\hat{\theta}))$ oder $\text{var}(\tilde{X}^*|X_i^* \sim \hat{F}_n)$. Nur in seltenen Fällen sind diese Bootstrap-Momente analytisch berechenbar. Daher wird wiederum die Monte Carlo Methode dafür verwendet.

Allgemeiner MC-Bootstrap Algorithmus in `R`:

```
n <- length(x)
R <- 1000
med.star <- 1:R
for (r in 1:R) {
  x.star <- rF(n, par.estimate)          # parametric Bootstrap
  x.star <- sample(x, size=n, replace=T) # non-parametric Bootstrap
  med.star[r] <- median(x.star)
}
EMC.median <- mean(med.star)
varMC.median <- var(med.star)
```

Als Monte-Carlo Approximation der Bootstrap-Schätzung für die asymptotische relative Effizienz des Medians im Vergleich zum Mittel erhält man unter Normalverteilungsannahme für die Variable `fvc` aus dem Datensatz `aimu`

```
aimu <- read.table("aimu.dat",
        col.names=c("nr","jahr","alter","gr","gew","fvc","fev","fvcfev.ratio","ort"))
attach(aimu)
n <- length(fvc)
R <- 1000
med.star.p <- mean.star.p <- med.star.np <- mean.star.np <- 1:R
for (r in 1:R) {
  x.star.p <- rnorm(n, mean(fvc), sd(fvc))    # parametric Bootstrap
  x.star.np <- sample(fvc, size=n, replace=T) # non-parametric Bootstrap
  mean.star.p[r] <- mean(x.star.p)
  med.star.p[r] <- median(x.star.p)
  mean.star.np[r] <- mean(x.star.np)
  med.star.np[r] <- median(x.star.np)
}
are.MCB.p <- var(med.star.p)/var(mean.star.p); are.MCB.p
[1] 1.523219
are.MCB.np <- var(med.star.np)/var(mean.star.np); are.MCB.np
[1] 1.369213
```

```
breaks <- seq(from=500, to=600, by=10)
hist(med.star.p, breaks, xlim=c(500, 600), ylim=c(0,350))
hist(med.star.np, breaks, xlim=c(500, 600), ylim=c(0,350))
```