统计计算 张楠 2019秋: Variance Reduction: Antithetic Variables

(返回 统计计算 张楠 2019秋)

我们已经知道Monte Carlo积分方法可以用来估计$E[g(X)]$这种类型的积分. 下面我们讨论如何减少$\theta = E[g(X)]$ 的样本均值估计量.

若$\hat{\theta}_1$和$\hat{\theta}_2$为参数θ的两个估计量, 且$Var(\hat{\theta}_2) < Var(\hat{\theta}_1)$, 则使用$\hat{\theta}_2$比使用$\hat{\theta}_1$方差的减少百分比为$100\left(\frac{Var(\hat{\theta}_1)-Var(\hat{\theta}_2)}{Var(\hat{\theta}_1)}\right)$. 尽管可以增加样本量$m$来减少方差, 但是计算成本会很高. 比如标准差要至多为e, 而$Var(g(X)) = \sigma^2$, 则需要样本量$m \geq \frac{\sigma^2}{e^2}$. 如要把标准差从0.01减少到0.0001, 则样本量至少要10000.

Antithetic Variables

注意都对两个随机变量U_1, U_2,

$$Var((U_1+U_2)/2) = \frac{1}{4}(Var(U_1)+Var(U_2)+2Cov(U_1,U_2).$$

因此, 当U_1, U_2负相关时$(U_1+U_2)/2$的方差比其相互独立时要小. 这个事实促使我们考虑使用负相关的随机变量 来减少方差.

比如, 假设$X_1, \ldots, X_n$为通过逆变换方法生产的, 对此m个随机向量$(X = (X_1, \ldots, X_n))$的每一个, 我们已经产生$U_j \sim U(0,1)$以及计算$X^{(j)} = F_X^{-1}(U_j), j = 1, \ldots, n$. 注意到若$U \sim U(0,1)$,则 $1-U \sim U(0,1)$, 而U与$1-U$负相关. 则

$$Y_j = g(X^{(j)}) = g(F_X^{-1}(U_1^{(j)}), \ldots, F_X^{-1}(U_n^{(j)}))$$

与

$$Y_j' = g(X^{(j)}) = g(F_X^{-1}(1-U_1^{(j)}), \ldots, F_X^{-1}(1-U_n^{(j)}))$$

同分布.

那么什么情形下Y_j与Y_j'是负相关的? 下面的定理说明如果g是单调的, 则Y_j和Y_j'是负相关的.

定义$(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$ 如果$x_j \leq y_j, j = 1, \ldots, n$. 一个$n$元函数 $g = g(x_1, \ldots, x_n)$称为是增的, 如果其对每一个变量都是增的, 即如果$(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$, 则$g(x_1, \ldots, x_n) \leq g(y_1, \ldots, y_n)$. 类似的称$g$为递减的,如果其对每个变量都是减的. 最后, 称$g$是单调的,如果其是增的或者减的.

定理：设$X = (X_1, \cdots, X_n)$各分量相互独立, f和g为同方向单调函数(同增或者同减), 则

$$E[f(X)g(X)] \geq E[f(X)]E[g(X)].$$

证: [折叠]

不妨设f, g同增. 证明方法是对n归纳. 假设$n = 1$, 则对所有的$x, y \in \mathcal{R}$, $(f(x)-f(y))(g(x)-g(y)) \geq 0$. 因此$E[(f(X)-f(Y))(g(X)-g(Y))] \geq 0$, 即有

$$E[f(X)g(X)]+E[f(Y)g(Y)] \geq E[f(X)g(Y)]+E[f(Y)g(X)].$$

此处$X,Y\ i.i.d$, 因此

$$\begin{aligned}&2E[f(X)g(X)]=E[f(X)g(X)]+E[f(Y)g(Y)]\\&\geq E[f(X)g(Y)]+E[f(Y)g(X)]=2E[f(X)]E[g(X)],\end{aligned}$$

即结论对$n=1$成立. 现在假设对$n-1$成立, 则由于

$$\begin{aligned}&E[f(X)g(X)|X_n=x_n] \geq E[f(X_1,\ldots,X_{n-1},x_n)]E[g(X_1,\ldots,X_{n-1},x_n)]\\&=E[f(X|X_n=x_n)]E[g(X|X_n=x_n)],\end{aligned}$$

最后一式每个为X_n的增函数, 因此根据$n=1$的结论有

$$\begin{aligned}&E[f(X)g(X)]=E\{E[f(X)g(X)|X_n]\}\\&\geq E\{E[f(X|X_n=x_n)]\}E\{E[g(X|X_n=x_n)]\}\\&=E[f(X)]E[g(X)].\end{aligned}$$

□

推论：设$g=g(X_1,\ldots,X_n)$ 是单调的, 则

$$Y=g(F_X^{-1}(U_1),\ldots,F_X^{-1}(U_n))$$

与

$$Y'=g(F_X^{-1}(1-U_1),\ldots,F_X^{-1}(1-U_n))$$

是负相关的.

证: [折叠]

不失一般性, 假设g是递增的. 则

$$Y=g(F_X^{-1}(U_1),\ldots,F_X^{-1}(U_n))$$

与

$$-Y'=f=-g(F_X^{-1}(1-U_1),\ldots,F_X^{-1}(1-U_n))$$

是两个递增的函数. 因此$E[g(U)f(U)] \geq E[g(U)]E[f(U)]$, 即 $E[YY'] \leq E[Y]E[Y']$, 此即意味着

$$Cov(Y,Y')=E[YY']-E[Y]E[Y'] \leq 0.$$

从而Y和Y'负相关.

□

对偶变量(Antithetic Variables)方法很容易应用. 如果需要m个Monte Carlo重复, 则产生$m/2$个重复

$$Y_j=g(F_X^{-1}(U_1^{(j)}),\ldots,F_X^{-1}(U_n^{(j)}))$$

以及剩下$m/2$个重复

$$Y_j' = g(F_X^{-1}(1-U_1^{(j)}), \ldots, F_X^{-1}(1-U_n^{(j)})),$$

其中$U_i^{(j)}, i = 1, \ldots, n; j = 1, \ldots, m/2.\ i.i.d\ U(0,1)$. 则对偶估计量为

$$\begin{aligned}\hat{\theta} &= \frac{1}{m}\{Y_1 + Y_1' + Y_2 + Y_2' + \cdots + Y_{m/2} + Y_{m/2}'\} \\ &= \frac{2}{m}\sum_{j=1}^{m/2}\frac{Y_j + Y_j'}{2}.\end{aligned}$$

通过使用对偶变量使得得到的Monte Carlo 估计的方差减少.

例6: 对偶变量方法使用之前的例子, 比如要估计积分 $\Phi(x) = \int_{-\infty}^{x} \frac{1}{2\pi} e^{-t^2/2} dt.$

使用对偶变量方法估计此积分, 并找出标准差的减少量.

作变量代换后, 目标积分是$\theta = E_U[xe^{-(xU)^2/2}]$, 其中$U \sim U(0,1)$. 显然此处$g$函数是单调的, 因此前面推论的条件满足. 产生随机数$u_1, \cdots, u_{m/2} \sim U(0,1)$, 计算

$$Y_j = g^{(j)}(u) = xe^{-(xu_j)^2/2}, j = 1, \cdots, m/2.$$

以及

$$Y_j' = xe^{-(x(1-u_j))^2/2}, j = 1, \cdots, m/2.$$

则样本均值

$$\hat{\theta} = \overline{g_m(u)} = \frac{1}{m/2}\sum_{j=1}^{m/2}\frac{xe^{-(u_jx)^2} + xe^{-(x(1-u_j))^2/2}}{2}$$

收敛到$E\hat{\theta}$.

从而若$x > 0$,则$\Phi(x)$的估计为$0.5 + \hat{\theta}/\sqrt{2\pi}$; 若$x < 0$, 则$\Phi(x) = 1 - \Phi(-x)$.

```
MC.Phi <- function(x, R = 10000, antithetic = TRUE) {
        u <- runif(R/2)
        if (!antithetic) v <- runif(R/2) else
            v <- 1 - u
        u <- c(u, v)
        cdf <- numeric(length(x))
        for (i in 1:length(x)) {
            g <- x[i] * exp(-(u * x[i])^2 / 2)
            cdf[i] <- mean(g) / sqrt(2 * pi) + 0.5
        }
        cdf
    }
```

和简单的Monte Carlo积分方法相比较:

```
x <- seq(.1, 2.5, length=5)
Phi <- pnorm(x)
set.seed(123)
MC1 <- MC.Phi(x, anti = FALSE)
```

```
set.seed(123)
MC2 <- MC.Phi(x)
print(round(rbind(x, MC1, MC2, Phi), 5))
```

方差的减少量可以模拟来比较:

```
m <- 1000
MC1 <- MC2 <- numeric(m)
x <- 1.95
for (i in 1:m) {
    MC1[i] <- MC.Phi(x, R = 1000, anti = FALSE)
    MC2[i] <- MC.Phi(x, R = 1000)
}
print(sd(MC1))
print(sd(MC2))
print((var(MC1) - var(MC2))/var(MC1))
```

对$x = 1.95$, 对偶变量方法相比于简单Monte Carlo方法估计的方差大约减少99.5%.