Lecture 6: 统计推断中的Monte Carlo 方法

张伟平

Monday 12^{th} October, 2009

Contents

Chapter 1
Monte Carlo Methods in Inference

Monte Carlo方法可以指统计推断或者数值分析中任何使用随机数模拟的方法. 本章 我们讨论一些统计推断中的Monte Carlo方法. Monte Carlo方法可以用于估计统计量分布 中的参数, 均方误差, 分位数等, 也可以用于估计置信区间覆盖真实参数的概率(置信水平), 假设检验中检验的一型错误率, 功效等等.

1.1 Monte Carlo Methods for Estimation

假设$X_1, \cdots, X_n$为从总体X中抽取的随机样本, 参数θ的估计量记为

$$\hat{\theta} = \hat{\theta}(X_1, \cdots, X_n)$$

记$x=(x_1,\cdots,x_n)^T\in\mathcal{R}^n$, 以及 $x^{(1)},x^{(2)},\cdots$为从总体X中抽取的一列独立的随机样本观测值. 则 有关于$\hat\theta$的性质, 可以通过估计值序列$\hat\theta(x_1^{(j)},\cdots,x_n^{(j)})$, $j=1,2,\cdots$ 来研究.

1.1.1 Monte Carlo Estimation and Standard Error

例 1: 假设X_1,X_2 $i.i.d\ \sim N(0,1)$, 估计$E|X_1-X_2|$.

显然, $\theta=E|X_1-X_2|$的Monte Carlo估计可用通过从标准正态分布中 产生m个样本 $x^{(j)}=(x_1^{(j)},x_2^{(j)})$, $j=1,\cdots,m$. 然后计算 $\hat\theta^{(j)}=|x_1^{(j)}-x_2^{(j)}|,j=1,\cdots,m$. 以及$\theta$的估计

$$\hat\theta=\frac{1}{m}\sum_{j=1}^{m}\hat\theta^{(j)}=\frac{1}{m}\sum_{j=1}^{m}|x_1^{(j)}-x_2^{(j)}|.$$

在R中很容易实现:

↑Code

```
m <- 1000
```

```
g <- numeric(m)
for (i in 1:m) {
    x <- rnorm(2)
    g[i] <- abs(x[1] - x[2])
}
est <- mean(g)
est
```

↓Code

我们也可以计算出$E|X_1 - X_2| = 2/\sqrt{\pi} \doteq 1.128379$ 以及方差$Var(|X_1 - X_2|) = 2 - 4/\pi$. 因此$E\hat{\theta} = 2/\sqrt{\pi}, Var(\hat{\theta}) = [2 - 4/\pi]/m$.

对标准差的Monte Carlo 估计我们可以从一般场合出发讨论. 由于样本量为n的样本均值$\bar{X}$ 的标准差为$\sqrt{Var(X)/m}$, 当X的分布未知时, 可以使用"Plug-in"法估计: 由

$$\hat{Var}(X) = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})^2.$$

因此$\bar{X}$标准差的估计为

$$\hat{se}(\bar{x}) = \frac{1}{\sqrt{m}}\left[\frac{1}{m}(x_i-\bar{x})^2\right]^{1/2} = \frac{1}{m}\left[(x_i-\bar{x})^2\right]^{1/2}.$$

或者也可以使用无偏估计量

$$\hat{se}(\bar{x}) = \frac{1}{\sqrt{m}}\left[\frac{1}{m-1}(x_i-\bar{x})^2\right]^{1/2}.$$

因此, 前例中的标准差估计为

$$\hat{se}(\hat{\theta}) = \frac{1}{m}\left[\sum_{j=1}^{m}(\hat{\theta}^{(j)}-\hat{\theta})^2\right].$$

计算程序如下

↑Code

```
sqrt(sum((g-mean(g))^2))/m
#sd(g)/sqrt(m)  #for unbiased estimator
```

↓Code

1.1.2 Estimation of MSE

Monte Carlo 方法可以用于计算一个估计量的MSE. 一个估计量的MSE定义为$MSE(\hat{\theta}) = E[\hat{\theta} - \theta]^2$. 如果从总体$X$中产生了$m$个样本$x^{(1)}, \cdots, x^{(m)}$, 则$\hat{\theta}$的MSE的Monte Carlo估计为

$$M\hat{S}E = \frac{1}{m}\sum_{j=1}^{m}(\hat{\theta}^{(j)} - \theta)^2,$$

其中$\hat{\theta}^{(j)} = \hat{\theta}(x^{(j)})$.

例 2: 使用Monte Carlo方法估计标准正态分布的截尾均值$\bar{X}_{[-1]}$的MSE.

当样本存在异常点时, 截尾的样本均值常常被用来估计总体的中心. 假设$X_1, \cdots, X_n$为一个随机样本, $X_{(1)}, \cdots, X_{(n)}$为相应的次序统计量, 则 一个k水平的截尾样本均值为

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

本例中，目标参数为$\theta = E\bar{X} = E\bar{X}_{[-1]} = 0$. 记$T = \bar{X}_{[-1]}$, 则其MSE的一个 Monte Carlo 估计算法如下

1. 通过如下步骤产生m个重复$T^{(j)}, j = 1, \cdots, m$:
   - 产生总体X的样本: $x_1^{(j)}, \cdots, x_n^{(j)}$.
   - 从小到大排序 $x_{(1)}^{(j)} \le \cdots \le x_{(n)}^{(j)}$
   - 计算 $T^{(j)} = \frac{1}{n-2} \sum_{i=2}^{n-1} x_{(i)}^{(j)}$.
2. 计算MSE $\hat{MSE}(T) = \frac{1}{m} \sum_{j=1}^{m} (T^{(j)} - \theta)^2 = \frac{1}{m} (T^{(j)})^2$.

实现程序如下

↑Code

```
n <- 20
m <- 1000
tmean <- numeric(m)
```

```
for (i in 1:m) {
    x <- sort(rnorm(n))
    tmean[i] <- sum(x[2:(n-1)]) / (n-2)
    }
mse <- mean(tmean^2)
mse
sqrt(sum((tmean - mean(tmean))^2)) / m    #se
```

↓Code

截尾均值的MSE的估计为0.0504531($\hat{se} \doteq 0.007$). 样本均值的 MSE为$Var(X)/n = 1/20 = 0.05$. 另一方面, 中位数本质上也是一种截尾均值, 其截掉了除中间的一个或两个点外的所有点. 样本中位数的MSE估计如下

↑Code

```
n <- 20
m <- 1000
tmean <- numeric(m)
for (i in 1:m) {
    x <- sort(rnorm(n))
    tmean[i] <- median(x)
    }
```

```
mse <- mean(tmean^2)
mse
sqrt(sum((tmean - mean(tmean))^2)) / m    #se
```

↓Code

从而样本中位数的MSE估计为0.075($\hat{se} = 0.0086$).

例 3: 比较标准正态分布与如下混合("污染")正态分布下的k水平截尾均值估计的MSE.

$$pN(0, \sigma^2 = 1) + (1-p)N(0, \sigma^2 = 100)$$

我们写一个函数来对不同的k和p计算截尾均值$\bar{X}_{[-k]}$的MSE. 从混合正态中产生 样本时, 要根据$P(\sigma = 1) = p; P(\sigma = 10) = 1 - p$来随机选择$\sigma$. 注意正态随机数产生 函数**rnorm**可以使用参数向量作为标准偏差. 考虑$p = 1.0, 0.95, 0.9$ 以及$k = 0, 1, \cdots, n/2$. 因此程序如下

↑Code

```
n <- 20; K <- n/2 - 1;  m <- 1000;
mse <- matrix(0, n/2, 6)
```

```
 trimmed.mse <- function(n, m, k, p) {
     #MC est of mse for k-level trimmed mean of
     #contaminated normal pN(0,1) + (1-p)N(0,100)
     tmean <- numeric(m)
     for (i in 1:m) {
         sigma <- sample(c(1, 10), size = n,
             replace = TRUE, prob = c(p, 1-p))
         x <- sort(rnorm(n, 0, sigma))
         tmean[i] <- sum(x[(k+1):(n-k)]) / (n-2*k)
         }
     mse.est <- mean(tmean^2)
     se.mse <- sqrt(mean((tmean-mean(tmean))^2)) / sqrt(m)
     return(c(mse.est, se.mse))
 }
for (k in 0:K) {
    mse[k+1, 1:2] <- trimmed.mse(n=n, m=m, k=k, p=1.0)
    mse[k+1, 3:4] <- trimmed.mse(n=n, m=m, k=k, p=.95)
    mse[k+1, 5:6] <- trimmed.mse(n=n, m=m, k=k, p=.9)
}
round(n*mse,3)
```

↓Code

结果表明, 均值的稳健估计(截尾均值估计)在总体分布被污染时能降低 MSE.

1.2 Estimating a confidence level

在应用统计中经常遇到的一个问题是估计总体的分布. 比如, 许多常用的统计推断 方法和工具都是基于正态性假设下的. 而在实际中, 总体分布非正态是经常的, 估计 量的分布可能不知道或者没有显示表示, 此时, Monte Carlo 方法则可以用来进行 统计推断.

假设(U, V)是未知参数θ的置信区间, 则统计量U, V的分布都依赖于抽样 分布X的分布F_X. 置信水平就是区间(U, V)能够覆盖θ真值的概率. 因此 估计置信水平就是一个积分估计问题.

比如考虑方差的置信区间估计问题. 标准的方法是对正态性假设很敏感的, 在数据(样本) 偏离正态分布时, 我们来使用Monte Carlo方法估计真实的置信水平. 首先看正态假定 下方差的置信区间估计(标准方法):

例 4: 方差的置信区间 假设$X_1, \cdots, X_n$, $i.i.d \sim N(\mu, \sigma^2)$, $n \geq 2$, S^2为样本方差, 则 由

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1}$$

知道σ^2的一个$100(1-\alpha)$置信上界为

$$(0, (n-1)S^2/\chi^2_{\alpha}(n-1))$$

从而从正态总体$N(0, \sigma^2 = 4)$中随机抽取$n = 20$个样本, 则计算σ^2的%95置信上界的 程序如下

↑Code

```
n <- 20
alpha <- .05
x <- rnorm(n, mean=0, sd=2)
UCL <- (n-1) * var(x) / qchisq(alpha, df=n-1)
```

↓Code

我们可以使用图来直观上判断经验的置信水平:

↑Code

```
m<-100000
ucls<-numeric(m)
for(i in 1:m){
x <- rnorm(n, mean=0, sd=2)
ucls[i] <- (n-1) * var(x) / qchisq(alpha, df=n-1)
}
ind<-ucls>4
cov.rate<-cumsum(ind)/1:m
plot(2:m,cov.rate[-1],type="l")
abline(h=0.95)
```

↓Code

经验的置信水平是通过模拟, 对理论的置信水平进行估计. 其一般做法如下

假设$X \sim F_X$, 感兴趣的参数为θ. 则对$j = 1, \cdots, m$:

1. 产生第j个随机样本$x_1^{(j)}, \cdots, x_n^{(j)}$.
2. 计算基于第j个样本的置信区间C_j
3. 计算$y_j = I(\theta \in C_j)$
4. 计算经验的置信水平$\bar{y} = \frac{1}{m}\sum_{j=1}^{m} y_j$.

例 5: 置信水平的Monte Carlo估计 在上一个例子中, 我们使用了**for** 循环来实现计算σ^2的m个置信上界. 我们也可以使用**replicate**函数:

↑Code

```
n <- 20
alpha <- .05
UCL <- replicate(1000, expr = {
    x <- rnorm(n, mean = 0, sd = 2)
    (n-1) * var(x) / qchisq(alpha, df = n-1)
    } )
```

```
#计算包含sigma^2=4的区间数
sum(UCL > 4)
#计算经验的覆盖率(置信水平)
mean(UCL > 4)
```

↓Code

replicate函数要重复执行的代码放在**{}**中, 参数**expr**可以调用一个函数:

↑Code

```
calCI <- function(n,alpha){
    x <- rnorm(n, mean = 0, sd = 2)
    return((n-1) * var(x) / qchisq(alpha, df = n-1))
    }
UCL<-replicate(1000,expr=calCI(n=20,alpha=.05))
mean(UCL>4)
```

↓Code

以上所说的置信区间构造方法是建立在正态性假设之上的，如果数据(样本)不服从正态分布，则 真正的置信水平为

$$P(\frac{(n-1)S^2}{\chi^2_\alpha(n-1)} > \sigma^2) = P(S^2 > \frac{\sigma^2\chi^2_\alpha(n-1)}{n-1}) = 1 - G(\frac{\sigma^2\chi^2_\alpha(n-1)}{n-1}).$$

这里G为统计量S^2的分布. 因此估计置信水平等价于要估计$G(t) = P(S^2 < t) = \int_0^t g(x)dx$, 从而Monte Carlo积分估计方法可以被用来估计此积分.

例 6: 经验的置信水平 在例5中，假设样本服从χ^2_2，则经验的置信水平是多少？此时 方差仍然是4.

↑Code

```
n <- 20; alpha <- .05
UCL <- replicate(1000, expr = {
    x <- rchisq(n, df = 2)  # 从chi^2(2)中抽样
    (n-1) * var(x) / qchisq(alpha, df = n-1) } )
sum(UCL > 4)
mean(UCL > 4)
```

↓Code

前面的例子中考虑的问题都是在总体分布已知的情形下, 对参数进行Monte Carlo估计. 因此这种 情况下的Monte Carlo方法也称为是参数*Bootstrop* 方法. 有关于Bootstrap 方法我们将在后面 学习.

1.3 Monte Carlo Methods for Hypothesis Tests

假定我们要考虑如下形式的假设检验问题:

$$H_0 : \theta \in \Theta_0 \quad vs \quad H_1 : \theta_1 \in \Theta_1$$

这里Θ_0, Θ_1为参数空间Θ的划分.

统计假设检验中会出现两种错误:

- Type I error: 零假设被拒绝, 但实际上零假设是正确的;
- Type II error: 零假设被接受, 但实际上零假设是错误的.

检验的**显著性水平**, α, 是一型错误率的上界. 拒绝零假设的概率 依赖于参数θ的真值, 记$\pi(\theta)$为拒绝零假设的概率, 则

$$\alpha = \sup_{\theta \in \Theta_0} \pi(\theta).$$

一型错误率是当零假设正确时, 零假设被拒绝的概率. 因此当一个检验在零假设条件件下 被重复很多次后, 观察到的一型错误率就应该逼近α.

若T为检验统计量, T^*为检验统计量的观测值, 则称T^*是**显著的**, 如果基于T^*的检验结论 是拒绝零假设H_0. **显著概率或p值**是使得检验统计量显著的最小的可能α值.

1.4 Empirical Type I error rate

Monte Carlo模拟可以用来计算一个检验方法的经验一型错误率. 检验过程在零假设条件件下大量重复, 则 经验一型错误率为检验统计量在重复中是显著的比例.

Monte Carlo模拟来计算经验的一型错误率:

1. 对每个重复j, $j = 1, \cdots, m$.
   (a) 从零分布产生第j个随机样本 $x_1^{(j)}, \cdots, x_n^{(j)}$;
   (b) 基于第j个样本计算检验统计量T_j;
   (c) 记录决策结果$I_j = 1$,若H_0在显著性水平α下被拒绝; 否则 $I_j = 0$.
2. 计算显著的检验比例$\frac{1}{m}\sum_{j=1}^{m} I_j$, 此比例即为观测到的一型错误率.

用$\hat{p}$表示估计的一型错误率, 则其标准方差的估计为

$$\hat{se}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \leq \frac{0.5}{\sqrt{m}}.$$

例7: 经验的一型错误率 假设$X_1, \cdots, X_{20}$ $i.i.d$ $N(\mu, \sigma^2)$, 假设$H_0: \mu = 500$ $H_1: \mu > 500$. 在零假设下,

$$T^* = \frac{\bar{X} - 500}{S/\sqrt{20}} \sim t_{19}.$$

大的T^*值是支持对立假设. 我们使用Monte Carlo方法来计算在$\sigma = 100$时的一型错误率, 来检测其是否逼近0.05.

↑Code

```
n <- 20; alpha <- .05;   mu0 <- 500;   sigma <- 100;
m <- 10000            #number of replicates
p <- numeric(m)       #storage for p-values
for (j in 1:m) {
    x <- rnorm(n, mu0, sigma)
    ttest <- t.test(x, alternative = "greater", mu = mu0)
    p[j] <- ttest$p.value
    }
p.hat <- mean(p < alpha)
se.hat <- sqrt(p.hat * (1 - p.hat) / m)
print(c(p.hat, se.hat))
```

```
plot(1:m,cumsum(p<alpha)/1:m,type="l")
abline(h=0.05)
```

↓Code

例8: 正态分布的偏度检验 一个随机变量X的偏度系数定义为

$$\beta_1 = \frac{E[(X-\mu)^3]}{\sigma^2},$$

其中$\mu = EX, \sigma^2 = Var(X)$. 一个分布称为是对称的, 如果$\beta_1 = 0$; 称为是正偏的,如果$\beta_1 > 0$;称为是负偏的,如果$\beta_1 < 0$. 偏度 系数的估计为

$$b_1 = \frac{\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^3}{(\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^2)^{3/2}}.$$

可以证明, $\sqrt{n}(b_1 - \beta_1) \overset{Asy}{\Longrightarrow} N(0,6)$. 正态分布是对称的分布, 因此偏度系数可以用来检验一个分布是否为对称分布. 假设为

$$H_0 : \beta_1 = 0 \quad vs \quad H_1 : \beta_1 \neq 0.$$

从而可以基于检验统计量b_1构建一个检验法则.

$$|b_1| \geq z_\alpha\sqrt{6/n}, Reject\ H_0;\ Otherwise,\ Accept\ H_0.$$

对大样本来说, 检验的显著性 水平达到指定的显著性水平, 比如$\alpha = 0.05$; 对小样本来说, 由于b_1的理论分布 不可得到, 从而确切的显著性水平不可得到, 但其经验的一型错误率可以通过Monte Carlo方法来估计.

↑Code

```
n <- c(10, 20, 30, 50, 100, 500) #sample sizes
cv <- qnorm(.975, 0, sqrt(6/n))  #asymptotical crit. values for each n

sk <- function(x) {
    #computes the sample skewness coeff.
    xbar <- mean(x)
    m3 <- mean((x - xbar)^3)
    m2 <- mean((x - xbar)^2)
    return( m3 / m2^1.5 )
}
```

```
#n is a vector of sample sizes
#we are doing length(n) different simulations
p.reject <- numeric(length(n)) #to store sim. results
m <- 10000                     #num. repl. each sim.
for (i in 1:length(n)) {
    sktests <- numeric(m)       #test decisions
    for (j in 1:m) {
        x <- rnorm(n[i])
        #test decision is 1 (reject) or 0 (Accept)
        sktests[j] <- as.integer(abs(sk(x)) >= cv[i] )
        }
    p.reject[i] <- mean(sktests) #proportion rejected
}
names(p.reject)<-as.character(n)
p.reject
```

↓Code

结果表明对较少的样本量时, 这种检验的显著性水平偏低,没有达到理论值0.05. 在正态总体下, 可以求出

$$Var(b_1) = \frac{6(n-2)}{(n+1)(n+3)}.$$

因此, 对小样本, 应该使用此方差来计算检验的临界值:

↑Code

```
cv<-qnorm(.975, 0, sqrt(6*(n-2)/((n+1)*(n+3))))  # crit. values for each n
p.reject <- numeric(length(n)) #to store sim. results
    m <- 10000                          #num. repl. each sim.
    for (i in 1:length(n)) {
        sktests <- numeric(m)          #test decisions
        for (j in 1:m) {
            x <- rnorm(n[i])
            #test decision is 1 (reject) or 0 (Accept)
            sktests[j] <- as.integer(abs(sk(x)) >= cv[i] )
            }
        p.reject[i] <- mean(sktests) #proportion rejected
    }
    p.reject
```

现在得到的估计就比较靠近名义值0.05了.

1.4.1 Power of a Test

一个检验的功效定义为

$$\pi(\theta) = P_\theta(Reject\ H_0)$$

当有一个$\theta_1 \in \Theta_1$时, 犯二型错误的概率为$1-\pi(\theta_1)$. 理想中, 我们 希望一个检验的错误率越低越好. 一型错误率通过显著性水平α的选择而控制, 较低的二型错误率对应于较高的功效. 因此, 在比较两个具有同样显著性水平的检验时, 我们感兴趣 的是比较它们的功效. 一般来说这种比较不是简单的一个问题, 比如一个检验的功效$\pi(\theta_1)$ 依赖于在对立假设下的θ_1值. 当一个检验的功效函数没有分析的表达式时, 功效函数 在特定的$\theta_1 \in \Theta_1$处的值可以通过Monte Carlo方法得到.

Monte Carlo 方法计算检验的功效:

1. 选择一个特定的$\theta_1 \in \Theta$.
2. 对每个重复j, $j = 1, \cdots, m$.
   (a) 在对立假设空间$\theta = \theta_1$的情况下产生第j个随机样本 $x_1^{(j)}, \cdots, x_n^{(j)}$;
   (b) 基于第j个样本计算检验统计量T_j;
   (c) 记录决策结果$I_j = 1$,若H_0在显著性水平α下被拒绝; 否则 $I_j = 0$.
3. 计算显著的检验比例$\frac{1}{m}\sum_{j=1}^{m} I_j$, 此比例即为观测到的一型错误率.

例9: 使用模拟方法估计例7中t检验的功效并画出经验功效函数曲线

↑Code

```
n <- 20
m <- 1000
```

```
mu0 <- 500
sigma <- 100
mu <- c(seq(450, 650, 10))  #alternatives
M <- length(mu)
power <- numeric(M)
for (i in 1:M) {
    mu1 <- mu[i]
    pvalues <- replicate(m, expr = {
        #simulate under alternative mu1
        x <- rnorm(n, mean = mu1, sd = sigma)
        ttest <- t.test(x,
                 alternative = "greater", mu = mu0)
        ttest$p.value  } )
    power[i] <- mean(pvalues <= .05)
}
```

↓Code

估计的功效$\hat{\pi}(\theta)$存在向量$power$中, 下面画出经验功效函数的图像, 我们这里使用**Hmisc**包里的**errbar**函数.

↑Code

```
par(ask = TRUE)
library(Hmisc)  #for errbar
plot(mu, power)
abline(v = mu0, lty = 1)
abline(h = .05, lty = 1)

#add standard errors
se <- sqrt(power * (1-power) / m)
errbar(mu, power, yplus = power+se, yminus = power-se,
    xlab = bquote(theta))
lines(mu, power, lty=3)
detach(package:Hmisc)
par(ask = FALSE)
```

↓Code

注意对t检验, 对假设$H_0: \mu = \mu_0 \quad vs \quad H_1: \mu \neq \mu_0$而言, 检验统计量$T = (\bar{X} - \mu)/(S/\sqrt{n})$在零假设下服从中心的$t$分布, 而在对立假设下, T服从非中心的t分布, 非中心参数为$\delta = (\mu - \mu_0)\sqrt{n}/\sigma$. 在R中,可以使用函数**power.t.test**来计算t检验的功效或者在给定功效下确定各参数的值.

例10: 混合正态分布下正态性偏度检验的功效 在例8中, 我们考虑了正态性偏度检验的一型错误率. 对混合正态分布

$$pN(0,\sigma^2=1)+(1-p)N(0,\sigma^2=100), 0\le p\le 1.$$

当$p=0,1$, 为正态分布; 而当$0<p<1$, 不再为正态分布. 因此, 我们可以计算一下偏度检验用于检验正态性的功效. 这里考虑显著性水平$\alpha=0.1$, 样本量$n=30$. 我们重复此过程$m=2500$次来计算功效.

↑Code

```
alpha <- .1
n <- 30
m <- 2500
epsilon <- c(seq(0, .15, .01), seq(.15, 1, .05))
N <- length(epsilon)
pwr <- numeric(N)
#critical value for the skewness test
cv <- qnorm(1-alpha/2, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))

for (j in 1:N) {            #for each epsilon
```

```
    e <- epsilon[j]
    sktests <- numeric(m)
    for (i in 1:m) {       #for each replicate
        sigma <- sample(c(1, 10), replace = TRUE,
            size = n, prob = c(1-e, e))
        x <- rnorm(n, 0, sigma)
        sktests[i] <- as.integer(abs(sk(x)) >= cv)
        }
    pwr[j] <- mean(sktests)
    }
#plot power vs epsilon
plot(epsilon, pwr, type = "b",
     xlab = bquote(epsilon), ylim = c(0,1))
abline(h = .1, lty = 3)
se <- sqrt(pwr * (1-pwr) / m)  #add standard errors
lines(epsilon, pwr+se, lty = 3)
lines(epsilon, pwr-se, lty = 3)
```

↓Code

经验的功效函数曲线在$p=0$和$p=1$两点接近$\alpha=0.1$, 对$0<p<1$, 功效函数大于0.1, 最大值大约在$p=0.15$处达到.

1.4.2 Power Comparisons

Monte Carlo方法经验用于比较不同检验的功效. 本节我们讨论对于正态性偏度检验问题几种检验 的功效差异. 文献中对于正态性检验已有很多不同的方法. 我们这里考虑三种检验方法.

例11: 正态性检验的功效比较 对一元正态性检验问题, 比较Shapiro-Wilk检验, Energy检验 和偏度检验三种检验的功效差异.

假设$\mathcal{N}$为一个一元正态分布类, 检验假设

$$H_0 : F_X \in \mathcal{N} \quad vs \quad H_1 : F_X \notin \mathcal{N}.$$

Shapiro-Wilk 检验是基于样本次序统计量对它们在正态性成立下的期望作回归, 因此它属于一般基于回归 和相关的类别检验方法. 对样本量$7 \leq n \leq 2000$, 近似的检验统计量临界值通过将统计量W变换到 正态分布随机变量而得到. 参考阅读材料了解更多Shapiro-Wilk检验方法. 在R中, 可以通过**shapiro.test** 函数作Shapiro-Wilk检验.

Energy检验是基于抽样分布和正态分布之间的"energy"距离进行检验, 因此大的检验统计量值意味着显著性. energy检验是用来检验多元正态性的一种方法, 这里我们考虑的是其特例$d=1$, 在一元正态检验下, energy检验类似于Anderson-Darling检验. energy检验统计量为

$$Q_n = n\left[\frac{2}{n}\sum_{i=1}^{n} E\|x_i - X| - E\|X - X'\| - \frac{1}{n^2}\sum_{i,j=1}^{n}\|x_i - x_j\|\right],$$

其中X, X'为$i.i.d$的正态分布随机变量. 大的Q_n值表明显著性. 在一元情形下, Q_n可以表示为

$$Q_n = n\left[\frac{2}{n}\sum_{i=1}^{n}(2Y_i\Phi(Y_i) + 2\phi(Y_i)) - \frac{2}{\sqrt{\pi}} - \frac{1}{n^2}\sum_{k=1}^{n}(2k-1-n)Y_{(k)}\right],$$

其中$Y_i = \frac{X_i - \mu_X}{\sigma_X}$, $Y_{(k)}$为$Y_1, \cdots, Y_n$的第k个次序统计量. 若参数μ_X, σ_X未知, 则用样本均值和样本方差. 在多元情形也有类似的计算公式, 在R中energy检验包含在**energy**包中, 名称为**mvnorm.etest**.

第三种检验就是我们前面例子中的偏度检验. 我们取显著性水平为$\alpha = 0.1$, 对立假设为

$$(1-p)N(\mu = 0, \sigma^2 = 1) + pN(\mu = 0, \sigma^2 = 100),\ \ 0 \le p \le 1$$

当$p = 0$或者1时, 分布为正态分布, 此时经验的一型错误率应该被名义的错误率$\alpha = 0.1$控制住, 我们 感兴趣的是在$0 < p < 1$时三种检验的功效比较.

↑Code

```
library(energy)
alpha <- .1
n <- 30
m <- 500        #try small m for a trial run
test1 <- test2 <- test3 <- numeric(m)

#critical value for the skewness test
cv <- qnorm(1-alpha/2, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))
sim <- matrix(0, 11, 4)

# estimate power
```

```
for (i in 0:10) {
    epsilon <- i * .1
    for (j in 1:m) {
        e <- epsilon
        sigma <- sample(c(1, 10), replace = TRUE,
            size = n, prob = c(1-e, e))
        x <- rnorm(n, 0, sigma)
        test1[j] <- as.integer(abs(sk(x)) >= cv)
        test2[j] <- as.integer(
                    shapiro.test(x)$p.value <= alpha)
        test3[j] <- as.integer(
                    mvnorm.etest(x, R=200)$p.value <= alpha)
        }
    print(c(epsilon, mean(test1), mean(test2), mean(test3)))
    sim[i+1, ] <- c(epsilon, mean(test1), mean(test2), mean(test3))
}
detach(package:energy)

# plot the empirical estimates of power
plot(sim[,1], sim[,2], ylim = c(0, 1), type = "l",
    xlab = bquote(epsilon), ylab = "power")
```

```
lines(sim[,1], sim[,3], lty = 2)
lines(sim[,1], sim[,4], lty = 4)
abline(h = alpha, lty = 3)
legend("topright", 1, c("skewness", "S-W", "energy"),
    lty = c(1,2,4), inset = .02)
```

↓Code

结果表明Shapiro-Wilks检验和energy检验在检验此假设中有着差不多的功效, 模拟比较中设定$n = 30$ 和$p < 0.5$. 此两个检验的功效都比偏度检验的功效大, energy检验看起来在$0.5 \leq p \leq 0.8$时 功效最大.

1.5 Application: "Count Five" Test for Equal Variance

本节中我们介绍一个简单两样本方差齐性检验方法. "Count five" 检验方差齐性方法由McGrath 和 Yeh 引入, 通过对一组样本相对于另外一组样本的样本范围的极值点个数. 假设两组样本的均值和样本量都相同. 一组样本下的一个观测值如果不在另一组样本的样本范围中, 则就是一个极值点. 如果每组样本

都有至少 5个极值点, 则拒绝等方差的假设.

例 12: Count Five test statistic 通过如下数值例子来说明此检验.

↑Code

```
x1 <- rnorm(20, 0, sd = 1)
x2 <- rnorm(20, 0, sd = 1.5)
y <- c(x1, x2)

group <- rep(1:2, each = length(x1))
boxplot(y ~ group, boxwex = .3, xlim = c(.5, 2.5), main = "")
points(group, y)
```

↓Code

对此两组样本, 通过Box图就可以直观上计算出相应的极值点个数. 也可以计算如下

↑Code

```
# now identify the extreme points
range(x1)
range(x2)
```

```
i <- which(x1 < min(x2))
j <- which(x2 > max(x1))

x1[i]
x2[j]

out1 <- sum(x1 > max(x2)) + sum(x1 < min(x2))
out2 <- sum(x2 > max(x1)) + sum(x2 < min(x1))
max(c(out1, out2))
```

↓Code

此时我们不能拒绝方差齐性的假设.

例 13: 考虑从同一正态分布中独立抽样的两组样本, 估计Count Five检验中极值点个数最大值的分布, 并 找出0.80,0.90和0.95分位数.

下面的函数**maxout**计算极值点的最大值.其分布可以通过Monte Carlo方法估计.

↑Code

```
maxout <- function(x, y) {
    X <- x - mean(x)
    Y <- y - mean(y)
    outx <- sum(X > max(Y)) + sum(X < min(Y))
    outy <- sum(Y > max(X)) + sum(Y < min(X))
    return(max(c(outx, outy)))
}

n1 <- n2 <- 20;   mu1 <- mu2 <- 0;  sigma1 <- sigma2 <- 1;  m <- 1000

# generate samples under H0
stat <- replicate(m, expr={
    x <- rnorm(n1, mu1, sigma1)
    y <- rnorm(n2, mu2, sigma2)
    maxout(x, y)
    })
print(cumsum(table(stat)) / m)
print(quantile(stat, c(.8, .9, .95)))
```

↓Code

注意**quantile**函数给出的0.95分位数是6, 但是如果$\alpha = 0.05$, 则5作为临界值更合适. quantile函数 给出的分位数并不是总是最好的, 因此一般要注意和经验分布函数比较.

McGrath 和 Yeh证明了 Count Five 检验用于检验中心化数据方差齐性时显著性水平最多为0.0625. 实际中, 总体均值一般是不知道的, 每组样本可以通过减去各自的样本均值来中心化.

例 14: 当总体均值未知, 每组样本减去各自的样本均值来中心化, 使用Monte Carlo方法估计 此时Count Five检验的显著性水平.

↑Code

```
count5test <- function(x, y) {
    X <- x - mean(x)
    Y <- y - mean(y)
    outx <- sum(X > max(Y)) + sum(X < min(Y))
    outy <- sum(Y > max(X)) + sum(Y < min(X))
    # return 1 (reject) or 0 (do not reject H0)
    return(as.integer(max(c(outx, outy)) > 5))
}
```

```
n1 <- n2 <- 20;  mu1 <- mu2 <- 0;  sigma1 <-  sigma2 <- 1;
m <- 10000
tests <- replicate(m, expr = {
    x <- rnorm(n1, mu1, sigma1)
    y <- rnorm(n2, mu2, sigma2)
    x <- x - mean(x)  #centered by sample mean
    y <- y - mean(y)
    count5test(x, y)
    } )

alphahat <- mean(tests)
print(alphahat)
```

↓Code

如果总体均值是已知的, 则经验的一型错误率大约为0.055(前面的例子中估计出的). 本例中 使用样本均值代替总体均值, 一型错误率大约为0.0565(se$\doteq$ 0.0022).

例 15: 重复例 13, 当两组样本的样本量不相同时, 计算经验的一型错误率.

↑Code

```
n1 <- 20
n2 <- 30
mu1 <- mu2 <- 0
sigma1 <- sigma2 <- 1
m <- 10000

alphahat <- mean(replicate(m, expr={
    x <- rnorm(n1, mu1, sigma1)
    y <- rnorm(n2, mu2, sigma2)
    x <- x - mean(x)  #centered by sample mean
    y <- y - mean(y)
    count5test(x, y)
    }))

print(alphahat)
```

↓Code

结果表明当两组样本的样本量不同时, Count Five检验并没有用0.0625控制住一型错误率.

例 16: 使用Monte Carlo方法估计Count Five检验的功效. 抽样分布为$N(\mu_1=0,\sigma_1^2=1)$ 和$N(\mu_2=0,\sigma_2^2=1.5^2)$, 样本量为$n_1=n_2=20$.

↑Code

```
# generate samples under H1 to estimate power
 sigma1 <- 1
 sigma2 <- 1.5

 power <- mean(replicate(m, expr={
     x <- rnorm(20, 0, sigma1)
     y <- rnorm(20, 0, sigma2)
     count5test(x, y)
     }))

 print(power)
```

↓Code

在对立假设$\sigma_1=1,\sigma_2=1.5$下检验的经验功效约为$0.3025(se\le 0.005)$.