第5章 重抽样方法

01 交叉验证和自助法

- 在本章中，我们讨论两种 *重采样* 方法:交叉验证（Cross Validation，CV）和自助法（Bootstrap）。

01 交叉验证和自助法

- 在本章中，我们讨论两种 *重采样* 方法:交叉验证（Cross Validation，CV）和自助法（Bootstrap）。
- 这些方法将感兴趣的模型与训练集形成的样本进行拟合，以获得关于拟合模型的额外信息。

01 交叉验证和自助法

大连理工大学 Dalian University of Technology

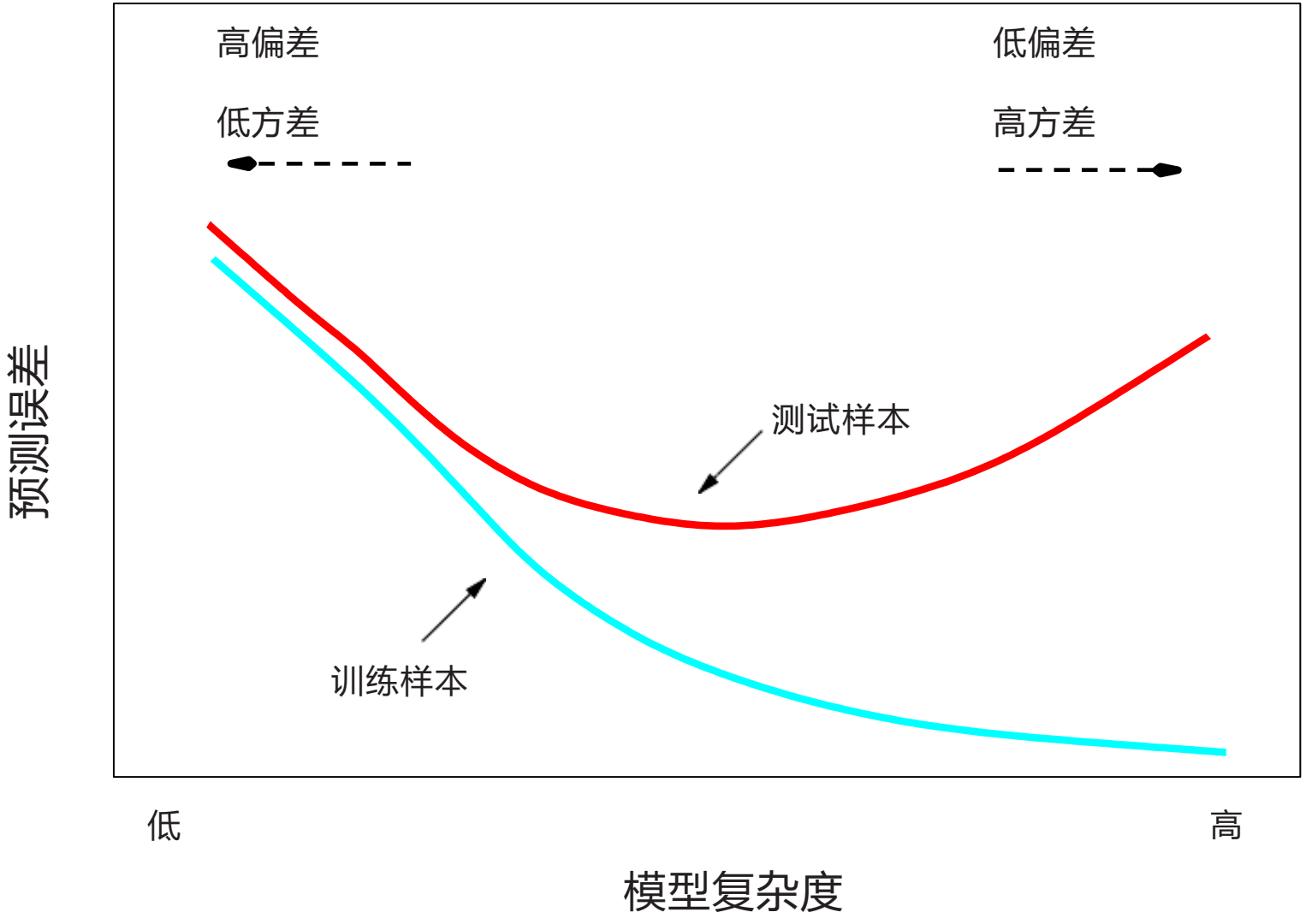
- 在本章中，我们讨论两种 *重采样* 方法:交叉验证（Cross Validation，CV）和自助法（Bootstrap）。
- 这些方法将感兴趣的模型与训练集形成的样本进行拟合，以获得关于拟合模型的额外信息。
- 例如，它们提供了测试集预测误差的估计值，以及我们的参数估计值的标准差和偏差。

03 训练误差 vs. 测试误差

大连理工大学 Dalian University of Technology

- 回想一下*测试误差*和*训练误差*之间的区别：
- *测试误差* 是使用统计学习方法来预测对新观测的反应所产生的平均误差，这个 观 测 在训练时没有使用。
- 相比之下，*训练误差* 可以很容易地通过将统计学习方法应用到训练观测来计算。
- 但训练错误率往往与测试错误率相差甚远，尤其是前者可以 *极大地低估* 后者。

训练集 vs. 测试集的表现



更多关于预测误差估计的内容

- 最佳解决方案:一个大型专门的测试集。通常没有。
- 有些方法对训练错误率进行数学调整，以估计测试错误率。这些方法包括C_p统计量，AIC和BIC。它们在本课程的其它地方也有讨论
- 在这里，我们转而考虑一类方法，它们通过从拟合过程中的训练数据集*拿出*子集来验证测试误差，将统计学习方法应用于这些拿出的观测数据

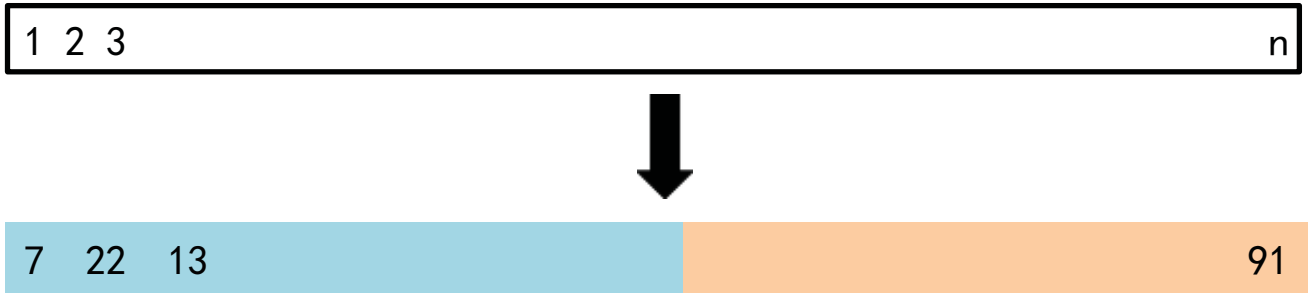
5.1 验证集的方法

5.1 验证集的方法

- 在这里，我们将可用的样本集随机分为两部分: 训练集 和 验证集（或保留集）。
- 模型在训练集上拟合，拟合后的模型用于预测验证集中的观察值的响应。
- 产生的验证集的误差提供了测试误差的估计值。通常，在量化响应情况下使用均方误差估计（Mean Square Error估计，MSE）；在定性（离散）响应情况下使用误分类率评估。

01 验证过程

大连理工大学
Dalian University of Technology



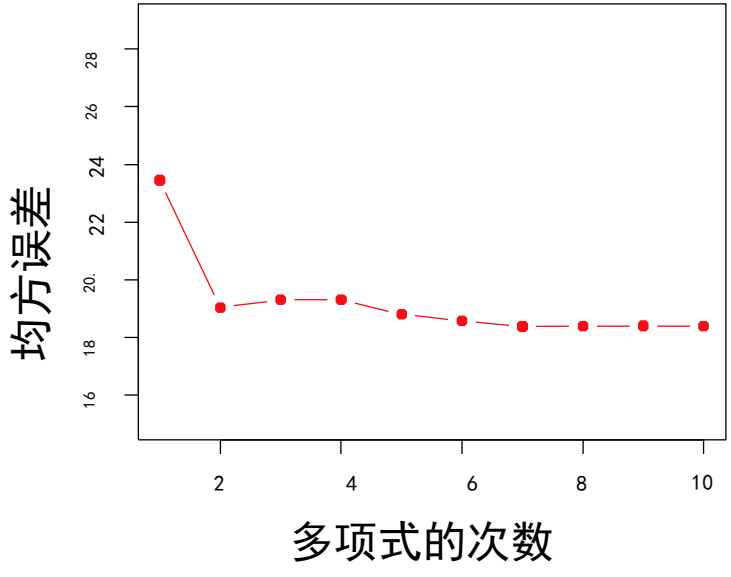
随机分成两半：左侧为训练集，右侧为验证集

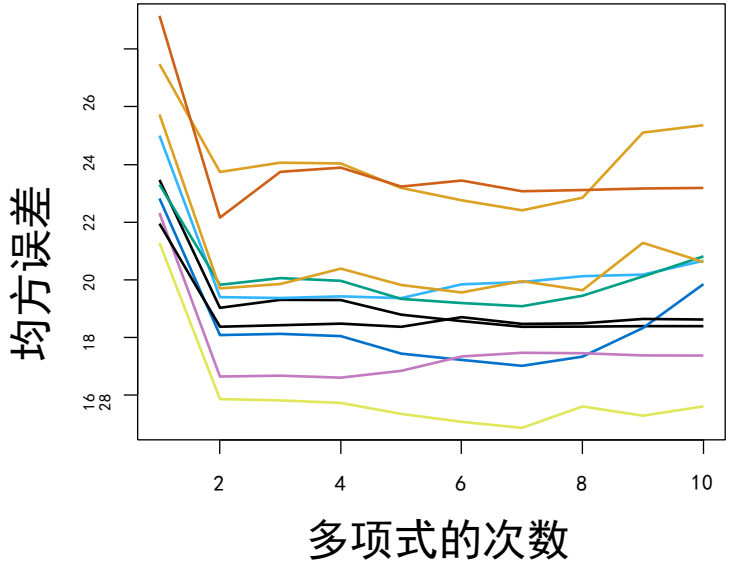
02

例子: 汽车的数据

大连理工大学
Dalian University of Technology

- 想要比较线性回归中的线性和高阶多项式项
- 我们将392个观测数据随机分成两个集合，一个是包含196个数据点的训练集，一个是包含剩余196个观测数据的验证集。





左图显示单划分

右图显示多划分

3 验证集方法的缺点

- 测试错误率的验证法估计的波动很大，取决于具体的哪些观测包含在训练集中，哪些包含在验证集中。
- 在验证方法中，只有观测数据的一个子集——那些被包括在训练集中而非验证集中的观测——被用来拟合模型。
- 这表明，验证集错误率可能会高估在整个数据集上拟合模型所得到的测试误差。

03 验证集方法的缺点

- 测试错误率的验证法估计的波动很大，取决于具体的哪些观测包含在训练集中，哪些包含在验证集中。
- 在验证方法中，只有观测数据的一个子集——那些被包括在训练集中而非验证集中的观测——被用来拟合模型。 为什么？
- 这表明，验证集错误率可能会高估在整个数据集上拟合模型所得到的测试误差。

04 验证集方法

首先用sample()函数把观测集分为两半，从原始的 392 个观测中随机地选取一个有 196个观测的子集，作为训练集。

```
>library(ISLR)
>set.seed(1)
train=sample(392,196)
```

然后用 lm() 函数中的 subset 选项，只用训练集中的观测来拟合一个线性回归模型。

```
>lm.fit=lm(mpg~horsepower,data=Auto,subset=train)
```

现在用 predict ()函数来估计余部 392 个观测的响应变量，再用 mean() 函数来计算验证集中196 个观测的均方误差。注意一下，下面的 -train 指标意味着只选取不在训练集中的观测。

```
>attach(Auto)
>mean((mpg-predict(lm.fit,Auto))[-train]^2)
[1] 26.14
```

因此，用线性回归拟合模型所产生的测试均方误差估计为26.14。下面用 poly ()函数来估计用二次和三次多项式回归所产生的测试误差。

```
>lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
>mean((mpg-predict(lm.fit2,Auto))[-train]^2)
[1] 19.82

>lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
>mean((mpg-predict(lm.fit3,Auto))[-train]^2)
[1] 19.78
```

验证集方法

这两个错误率分别为19. 82和19. 78。如果选择了一个不同的训练集的话，那就会在验证集上得到一个不同的误差。

```
>set.seed(2)
>train=sample(392,196)
>lm.fit=lm(mpg~horsepower,subset=train)
>mean((mpg-predict(lm.fit,Auto))[-train]^2)
[1] 23.30

>lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
>mean((mpg-predict(lm.fit2,Auto))[-train]^2)
[1] 18.90

>lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
>mean((mpg-predict(lm.fit3,Auto))[-train]^2)
[1] 19.26
```

用另一种分割把观测分为一个训练集和一个验证集，用线性、二次和三次项拟合的模型的验证集错误率分别为23.30, 18. 90，19. 26。
这些结果与之前的结论一致: 一个用horsepower的二次函数来拟合的模型预测mpg的效果比仅用horsepower的线性函数拟合模型的效果要好，而几乎没有证据表明用horsepower的三次函数拟合模型的效果更好。

5.2 K-折（K-fold）交叉验证

K-折（K-fold）交叉验证

- 估计测试误差的*广泛使用的方法*。
- 估计值可以用来选择最好的模型，并给出最终选择模型的测试误差的相关信息。
- 思路是：将数据随机分成 K 个大小（基本）相等的组。我们留出一组，如第 k 组，在剩下的 $K-1$ 组拟合出模型。然后用第 k 组测试模型得MSE。
- 对每一个 $k = 1, 2, \ldots, K$ 重复上述步骤，然后将结果进行组合。

K-折交叉验证细节举例

将数据分成 K 个大小大致相等的部分（这里 $K = 5$）

1	2	3	4	5
验证	训练	训练	训练	训练

- 令 K 个部分为C_1，C_2，…，C_K，其中 C_k 表示第 k 部分观测样本集合。第 k 部分有n_k个观测值：
- 计算加权平均

$$\mathrm{CV}_{(k)} = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

特别地，如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。其中，对于 C_k 中的观测样本 i，拟合值$\hat{y}_i$是使用将C_k 抠出的训练集训练所得模型预测出来的，样本 i 的实际标记值 y_i，

$$MSE_k = \sum_{\mathrm{i} \in C_k} \frac{(y_i - \hat{y}_i)^2}{n_k} \quad 。$$

- 令 K 个部分为C_1, C_2, …, C_K, 其中 C_k 表示第 k 部分观测样本集合。第 k 部分有n_k个观测值:
- 计算加权平均

$$\mathrm{CV}_{(k)} = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

特别地，如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。其中，对于 C_k 中的观测样本 i，拟合值$\hat{y}_i$是使用将C_k 抠出的训练集训练所得模型预测出来的，样本 i 的实际标记值 y_i ，

$$MSE_k = \sum_{i \in C_k} \frac{(y_i - \hat{y}_i)^2}{n_k}$$

- 设置“分组数K = 样本总数N”，称为N-折交叉验证或“留一法”交叉验证，即一个样本就是一组n_k =1。Leave One Out Cross Validation, LOOCV。

- 令 K 个部分为C_1, C_2, …, C_K, 其中 C_k 表示第 k 部分观测样本集合。第 k 部分有n_k个观测值:
- 计算加权平均

$$\mathrm{CV}_{(k)} = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

特别地，如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。其中，对于 C_k 中的观测样本 i，拟合值$\hat{y}_i$是使用将C_k 抠出的训练集训练所得模型预测出来的，样本 i 的实际标记值 y_i ，

$$MSE_k = \sum_{i \in C_k} \frac{(y_i - \hat{y}_i)^2}{n_k}$$

- 设置“分组数K = 样本总数N”，称为N-折交叉验证或“留一法”交叉验证，即一个样本就是一组n_k =1。Leave One Out Cross Validation, LOOCV。

缺点?

一个不错的特殊情况!

- 使用最小二乘线性或多项式回归，一个惊人的捷径让LOOCV的成本与单个模型拟合的成本相同!下面的公式成立:

$$\mathrm{CV}_{(\mathrm{N})} = \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

其中 $\hat{y}_i$ 是原始最小二乘拟合的第 i 个拟合值， h_i 是杠杆(“帽子”矩阵的对角线;详情见书)。这就像普通的MSE，除了第i个残差被一个系数 $1 - h_i$ 除。

一个不错的特殊情况!

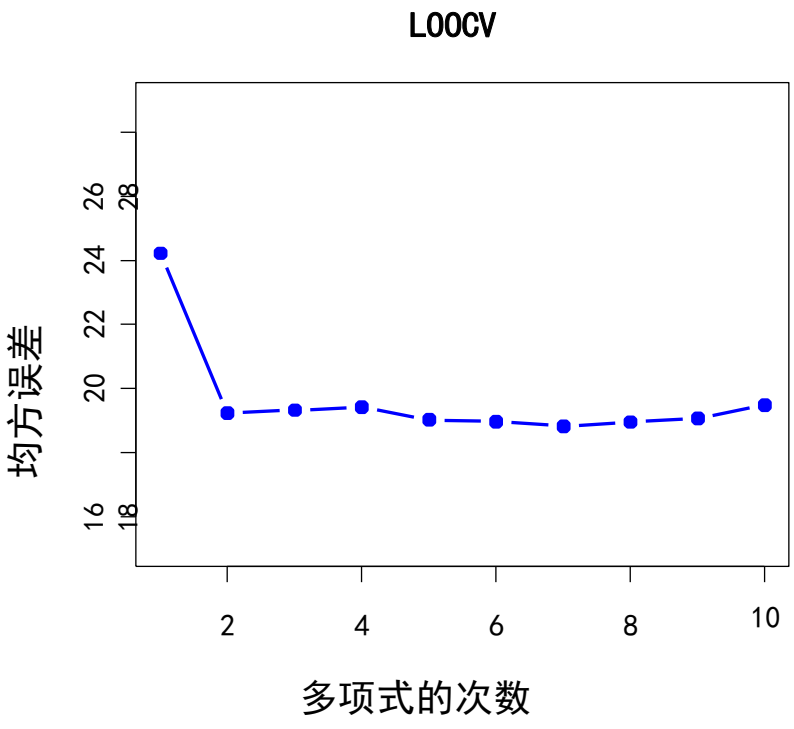
- 使用最小二乘线性或多项式回归，一个惊人的捷径让LOOCV的成本与单个模型拟合的成本相同!下面的公式成立:

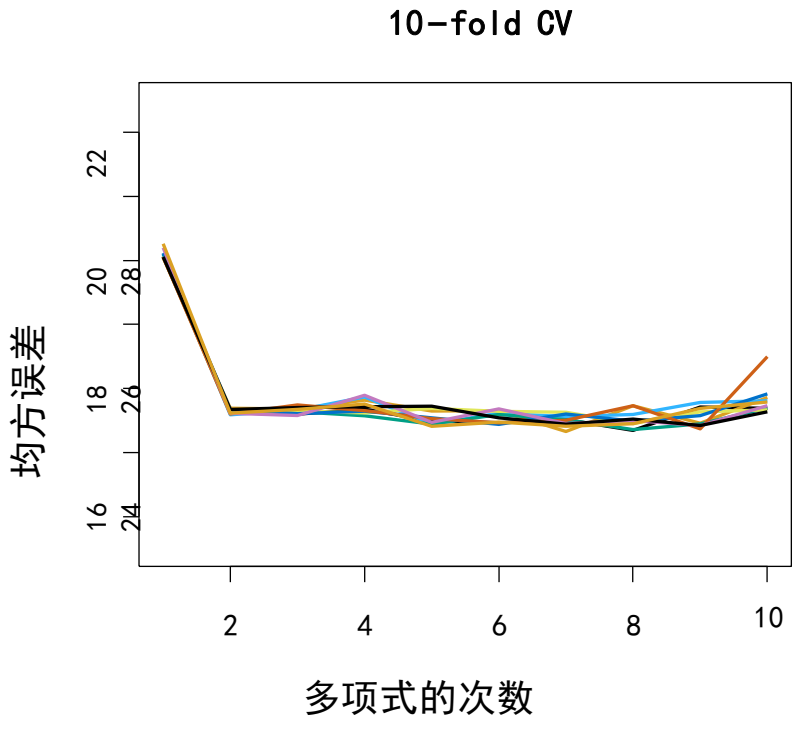
$$\mathrm{CV}_{(\mathrm{N})} = \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

其中 $\hat{y}_i$ 是原始最小二乘拟合的第 i 个拟合值， h_i 是杠杆(“帽子”矩阵的对角线;详情见书)。这就像普通的MSE，除了第i个残差被一个系数 $1 - h_i$ 除。

- LOOCV有时有用，但通常不会不会把数据充分“摇匀”。每一折的估计都高度相关，因此它们的平均值可能有很大的方差。
- 更好的选择是 $K = 5$或10。

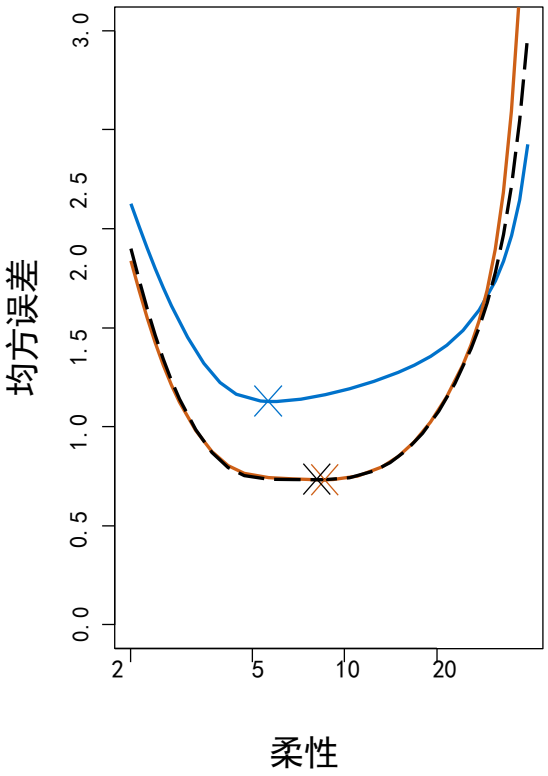
Auto数据回顾

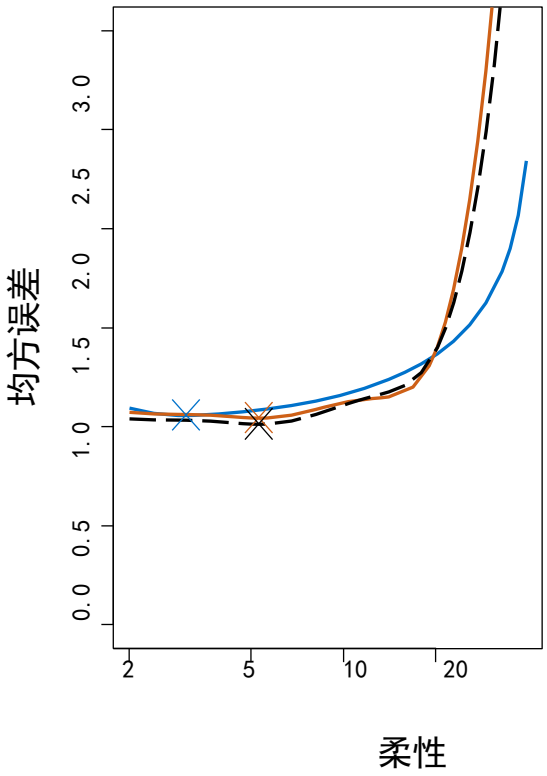

LOOCV
均方误差
16
18
20
22
24
26
28
2
4
6
8
10
多项式的次数

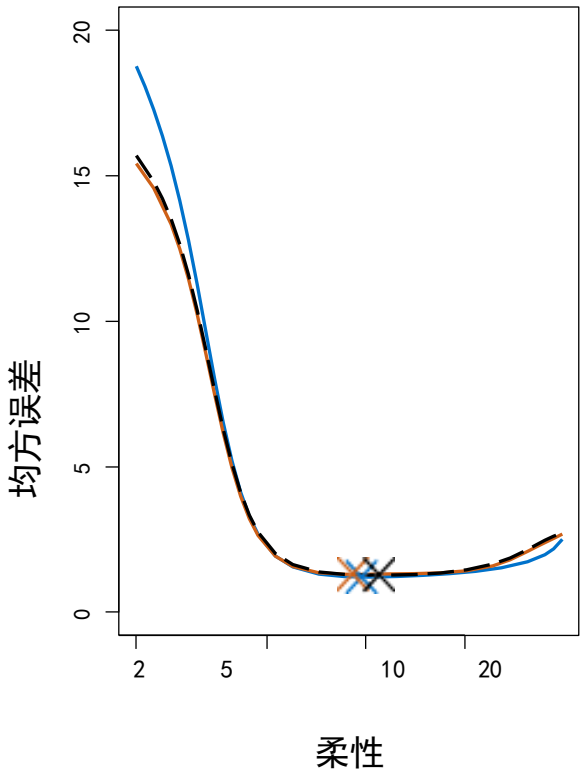

10-fold CV
均方误差
16
18
20
22
24
26
28
2
4
6
8
10
多项式的次数

05 对模拟数据的真实和估计检验MSE

大连理工大学
Dalian University of Technology







交叉验证的其它问题

- 由于每个训练集只有原始训练集的（K - 1）/ K 大，预测误差估计值通常会向上偏。

交叉验证的其它问题

- 由于每个训练集只有原始训练集的（K − 1）/ K 大，预测误差估计值通常会向上偏。**为什么？**

交叉验证的其它问题

- 由于每个训练集只有原始训练集的（K − 1）/ K大，预测误差估计值通常会向上偏。**为什么？**
- 当K = N（LOOCV）时，这一偏差最小，但如前所述，这一估计具有高方差。
- *K = 5或10为这种偏差-方差权衡提供了一个很好的折衷方案。*

分类问题的交叉验证

- 我们将数据分成K个大小大致相等的部分 C_1，C_2，…，C_K，其中 C_k 表示第 k 部分观测样本集合。第 k 部分有n_k个观测值：如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。
- K-折交叉验证错误率写作

$$\mathrm{CV_K} = \sum_{k=1}^{K} \frac{n_k}{N} \mathrm{Err}_k$$

其中，$\mathrm{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$。

- $\mathrm{CV_K}$的估计标准差为，

$$\widehat{\mathrm{SE}}(\mathrm{CV_K}) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\frac{(Err_k - \overline{Err_k})^2}{K-1}}$$

- 这是一个有用的估计，但严格地说，并不十分有效。

分类问题的交叉验证

- 我们将数据分成K个大小大致相等的部分 C_1，C_2，…，C_K，其中 C_k 表示第 k 部分观测样本集合。第 k 部分有n_k个观测值：如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。
- K-折交叉验证错误率写作

$$\mathrm{CV_K} = \sum_{k=1}^{K} \frac{n_k}{N} \mathrm{Err}_k$$

其中，$\mathrm{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$。

- $\mathrm{CV_K}$的估计标准差为，

$$\widehat{\mathrm{SE}}(\mathrm{CV_K}) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\frac{(Err_k - \overline{Err_k})^2}{K-1}}$$

- 这是一个有用的估计，但严格地说，并不十分有效。**为什么？**

交叉验证:对与错

- 考虑一个应用于一些二分类数据的简单分类器:
  1. 从5000个预测变量和50个样本开始，找到与类标签相关度最大的100个预测变量。
  2. 然后，我们应用一个分类器如逻辑斯谛回归，只使用这100个预测变量。

我们如何评估这个分类器的测试集表现?

交叉验证:对与错

- 考虑一个应用于一些二分类数据的简单分类器:
  1. 从5000个预测变量和50个样本开始，找到与类标签相关度最大的100个预测变量。
  2. 然后，我们应用一个分类器如逻辑斯谛回归，只使用这100个预测变量。

  我们如何评估这个分类器的测试集表现?

  我们是否可以在第2步中应用交叉验证，忘记第1步?

不!

- 这将忽略一个事实，即在步骤1中，过程中已经看到了训练数据的标签，并利用了它们。这是一种训练形式，必须包含在验证过程中。
- 很容易仿真真实数据，带类标签的，独立于输出的，使得真实的测试误差=50%，但忽略步骤1的交叉验证误差估计为零！

不!

- 这将忽略一个事实，即在步骤1中，过程中已经看到了训练数据的标签，并利用了它们。这是一种训练形式，必须包含在验证过程中。
- 很容易仿真真实数据，带类标签的，独立于输出的，使得真实的测试误差=50%，但忽略步骤1的交叉验证误差估计为零！

  试着自己做一下

不!

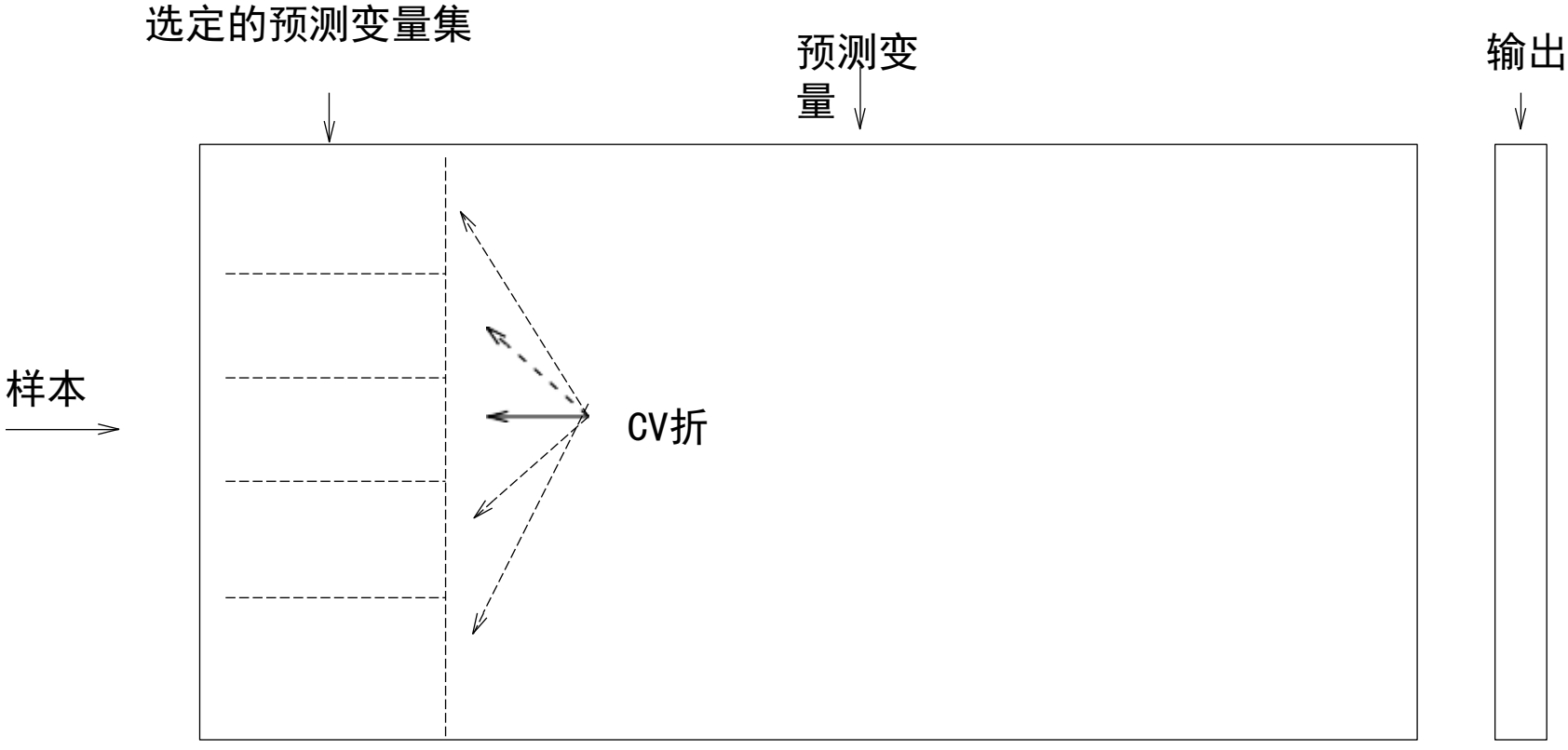
- 这将忽略一个事实，即在步骤1中，过程中已经看到了训练数据的标签，并利用了它们。这是一种训练形式，必须包含在验证过程中。
- 很容易仿真真实数据，带类标签的，独立于输出的，使得真实的测试误差=50%，但忽略步骤1的交叉验证误差估计为零！

  试着自己做一下
- 我们已经在许多备受瞩目的基因组学论文中看到了这个错误。

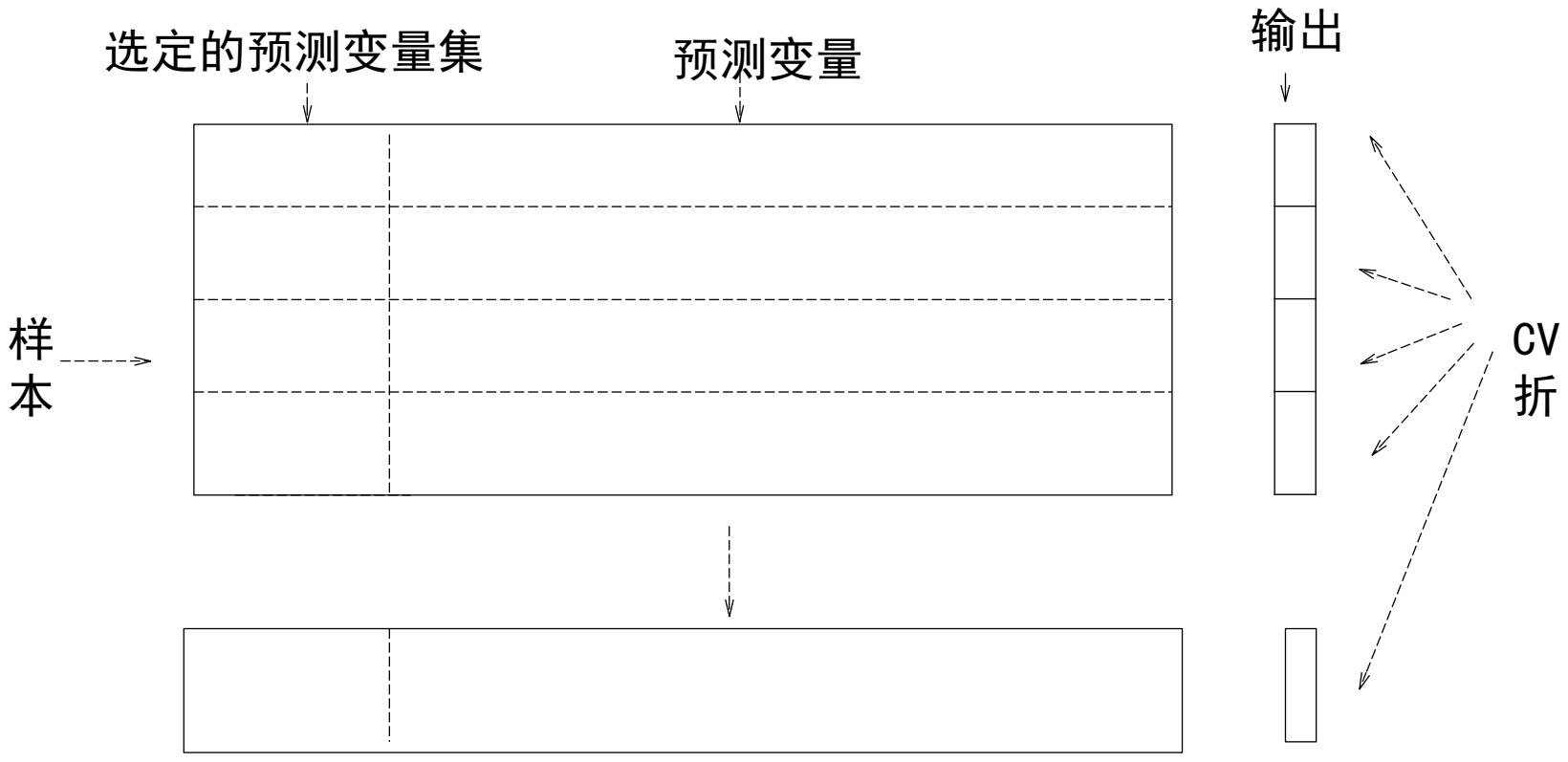
错误和正确的方法

- 错误:在步骤2中应用交叉验证。
- 正确:对步骤1、2应用交叉验证。

错误的方式



正确的方式



K-折（K-fold）交叉验证

CV.glm() 函数同样可以用于实现折CV。下面令=10这样一个通常的选择，然后在Auto数据集上使用折交叉验证。同样，下面设定一个随机种子，以及创建一个向量，把用一次到十次多项式拟合模型所产生的CV误差储存在这个向量中。

```
>set.seed(17)
>cv.error.10=rep(0,10)
>for (i in 1:10){
+ glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
+ cv.error.10[i]=cv.glm(Auto,glm.fit,K=10)$delta[1]
+ }
>cv.error.10
[1] 24.21 19.19 19.31 19.34 18.88 19.02 18.90 19.71 18.95 19.50
```

注意到K折交叉验证法的计算时间要比L0OCV的计算时间少得多。(理论上来说，由于有L00CV的公式(5.2) 的存在;用L00CV法拟合最小二乘线性模型的计算时间应该比k折CV法要短才对。但不幸的是，cv. glm ()函数并没有使用这个公式。)
同样，没有看到有证据表明用三次或者更高次的多项式拟合模型所产生的测试误差要比仅仅用二项式拟合模型的小。

5.3 自助法

5.3 自助法

- 自助法（Bootstrap）是一种灵活而强大的统计工具，可用于量化给定估计器或统计学习方法的不确定性。
- 例如，它可以提供一个系数的标准差的估计值，或者该系数的置信区间。

01 这个名字从何而来?

大连理工大学 Dalian University of Technology

- bootstrap一词源于短语“自力更生”(to pull yourself up by one 's bootstrap)，人们普遍认为这是基于18世纪鲁道夫·埃里希·拉斯佩(Rudolph Erich Raspe)的《蒙乔森男爵的惊人冒险》：

  男爵掉进了一个深湖的湖底。就在一切似乎都完了的时候，他想要自己爬起来。

- 这与计算机科学中使用的术语“bootstrap”(从一组内核指令“启动”一台计算机)并不相同，尽管两者的引申义类似。

02

一个简单的例子

大连理工大学
Dalian University of Technology

- 假设我们希望将一笔固定数额的钱投资于两种金融资产，它们的收益率分别为X和Y，其中X和Y是随机量。
- 我们将资金中的一小部分 α 投资于X，剩余的1 − α 投资于Y。
- 我们希望选择 α 来最小化我们投资的总风险(或方差)。换句话说，我们希望最小化Var(αX + (1 − α)Y)。

一个简单的例子

- 假设我们希望将一笔固定数额的钱投资于两种金融资产，它们的收益率分别为X和Y，其中X和Y是随机量。
- 我们将资金中的一小部分 α 投资于X，剩余的1 − α 投资于Y。
- 我们希望选择 α 来最小化我们投资的总风险(或方差)。换句话说，我们希望最小化Var(αX + (1 − α)Y)。
- 可以证明，使风险最小化的值是由

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y - 2\sigma_{XY}},$$

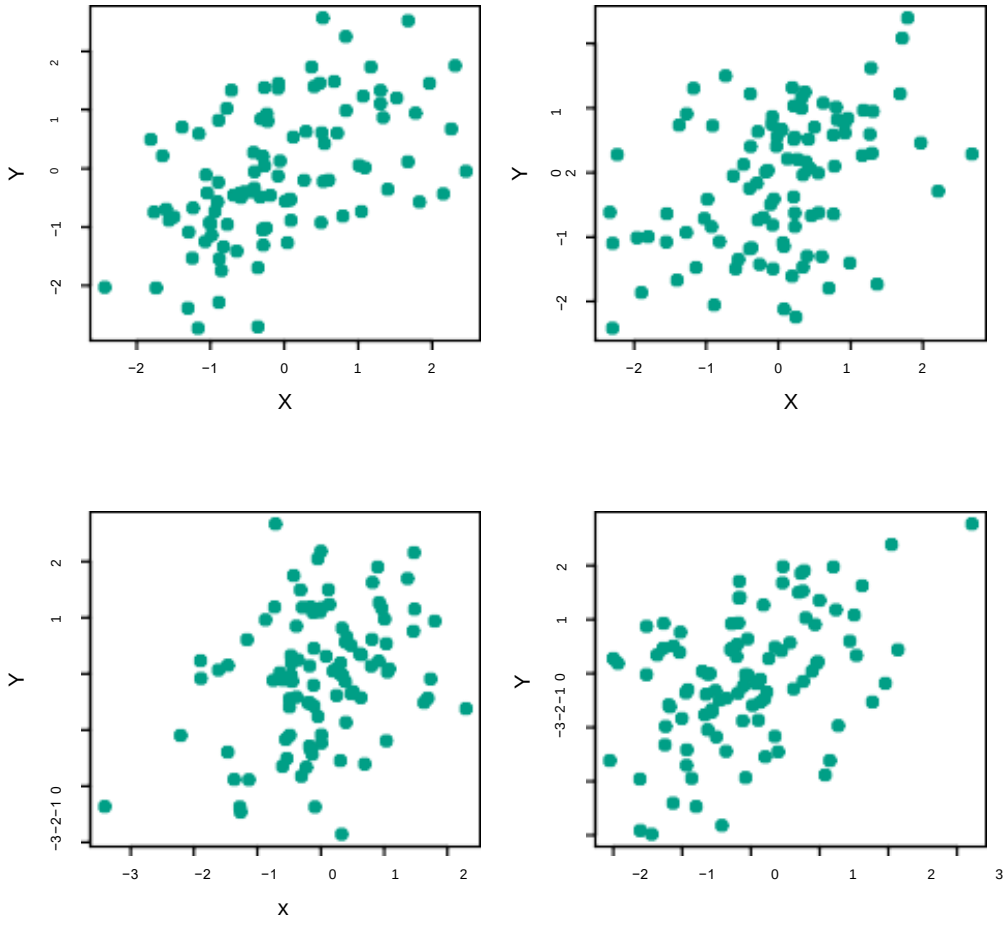
式中 σ^2 为方差， σ_{XY} 为协方差。

继续示例

- 但是这些方差、协方差值是未知的。$\alpha = \dfrac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y - 2\sigma_{XY}}$，
- 我们可以计算这些量的估计值，使用一个包含X和Y测量值的数据集。
- 然后我们可以估计 α 的值，其最小化我们投资的方差，使用下式

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

继续示例



每个面板显示投资 X 和 Y 的 100 个模拟回报。从左到右，从上到下，α 的结果估计为 0.576，0.532，0.657，0.651 。

继续示例

- 为了估计 α 的标准差ˆ，我们重复模拟100对X和Y的观测过程，并估计 α 1000次。
- 我们由此得到了 α 的1000个估计值，我们可以称之为$\hat{\alpha}_1$， $\hat{\alpha}_2$，…， $\hat{\alpha}_{1000}$。
- 后续幻灯片的图左侧面板显示了结果估计的直方图。
- 对于这些模拟，参数被设置为$\sigma_X^2 = 1$，$\sigma_Y^2 = 1.25$，$\sigma_{XY} = 0.5$，所以我们知道 α 的真实值是0.6（用红线表示）。

继续示例

- α 的1000个估计值的平均值是

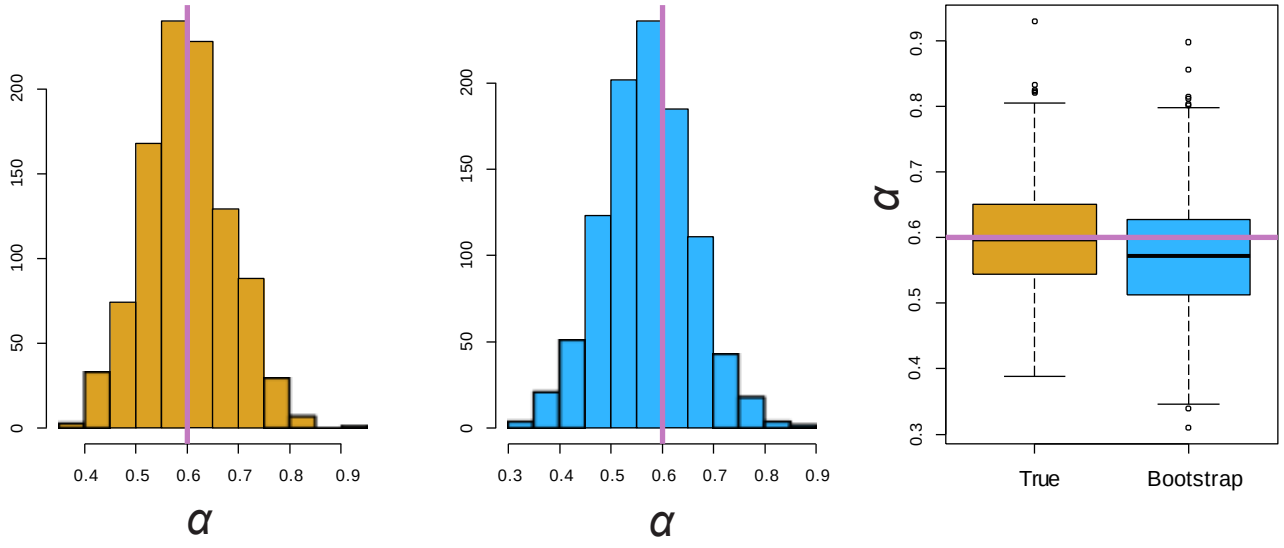
$$\bar{\alpha} = \frac{1}{1000}\sum_{r=1}^{1000}\hat{\alpha}_r = 0.5996\,,$$

非常接近 $\alpha = 0.6$，估计值的标准差为

$$\sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000}(\hat{\alpha}_r - \bar{\alpha})^2} = 0.083\,,$$

- 这让我们很好地了解了 $\hat{\alpha}$ 的准确性：$\mathrm{SE}(\hat{\alpha}) \approx 0.083$ 。
- 因此，粗略地说，对于从总体中随机抽取的样本，我们预计 $\hat{\alpha}$ 与 α 平均相差约为 0.08 。

结果



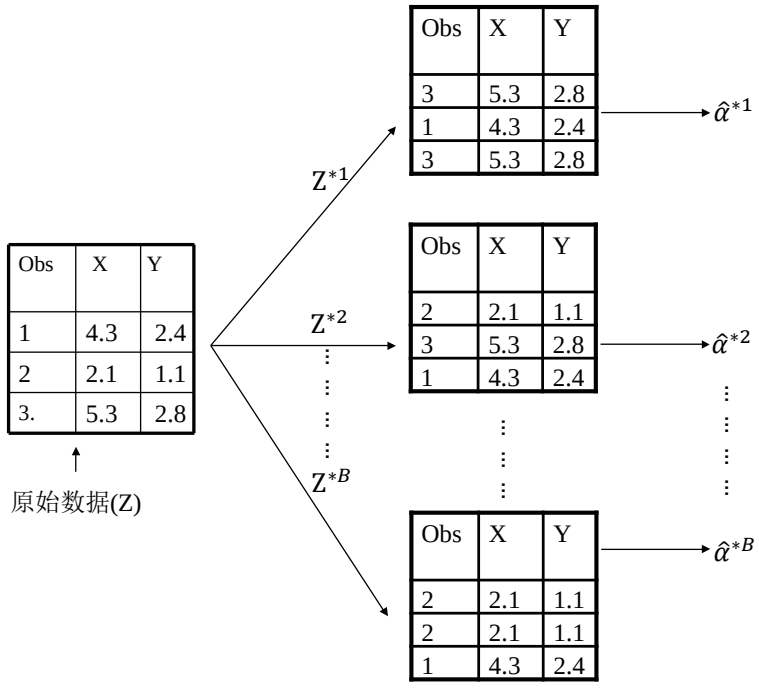
*左:*从真实集中生成1000个仿真数据集得到的 α 估计值直方图。

*中:*从单个数据集的1000个bootstrap样本中获得的 α 估计值直方图。*右:*左侧和中间面板中显示的 α 估计值以箱线图的形式显示。在每个面板中，粉色的线表示 α 的真实值。

现在回到现实世界

- 上述程序无法应用，因为对于真实数据，我们无法从原始总体中生成新样本。
- 然而，bootstrap方法允许我们使用计算机来模拟获得新数据集的过程，这样我们就可以在不产生额外样本的情况下评估我们估计的变化性。
- 我们不是反复地从总体中获得独立的数据集，而是通过对原始数据集有放回重复抽样来获得不同的数据集。
- 这些“bootstrap数据集”中的每一个都是通过有放回抽样创建的，大小与我们的原始数据集相同。因此，一些观察结果可能会在给定的bootstrap数据集中出现不止一次，而一些则根本不会出现。

只有3个观测的例子



包含n = 3个观察值的小样本上的bootstrap方法的图形说明。每个bootstrap数据集包含n个观察值，从原始数据集有放回抽样。每个bootstrap数据集被用来获得α的估计值。

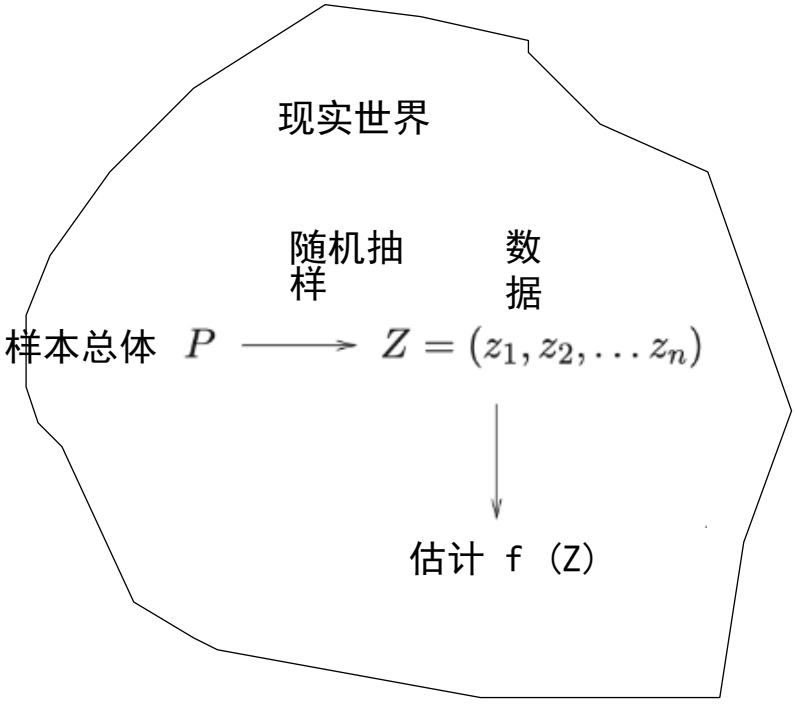
只有3个观测的例子

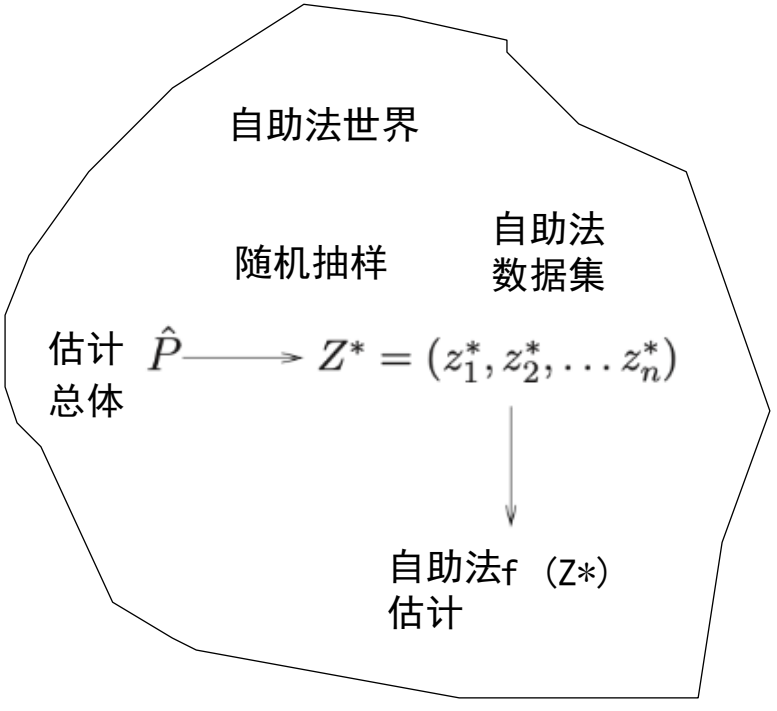
- 用Z^{*1}表示第一个bootstrap数据集，我们使用Z^{*1}产生一个$\hat{\alpha}$的新的bootstrap估计，记作$\hat{\alpha}^{*1}$。
- 这个过程重复 B 次（对一些大值 B 如100、1000），以得到 B 个不同的数据集，Z^{*1}，Z^{*2}，…，Z^{*B}，和 B 个α的估计值，$\hat{\alpha}^{*1}$，$\hat{\alpha}^{*2}$，…，$\hat{\alpha}^{*B}$。
- 我们利用该公式估计了这些bootstrap估计的标准误差

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^{*}\right)^2}\ .$$

- 这作为从原始数据集估计的 $\hat{\alpha}$ 的标准差。见幻灯片49页图的中间和右边面板。Bootstrap结果用蓝色表示。对于这个例子，$\mathrm{SE}_B(\hat{\alpha}) = 0.087$。

自助法概貌


现实世界
随机抽样
数据
样本总体 P $Z=(z_1,z_2,\ldots z_n)$
估计 f（Z）


自助法世界
随机抽样
自助法数据集
估计总体 $\hat{P}$ $Z^*=(z_1^*,z_2^*,\ldots z_n^*)$
自助法f（Z*）估计

自助法概貌

- 在更复杂的数据情况下，找出生成bootstrap样本的适当方法可能需要一些思考。
- 例如，如果数据是一个时间序列，我们不能简单地对观测数据进行有放回抽样(为什么不能?)。

自助法概貌

- 在更复杂的数据情况下，找出生成bootstrap样本的适当方法可能需要一些思考。
- 例如，如果数据是一个时间序列，我们不能简单地对观测数据进行有放回抽样(为什么不能?)。
- 相反，我们可以创建连续观察的区块，并对这些区块进行有放回抽取。然后我们将采样的块拼一起，以获得一个bootstrap数据集。

bootstrap的其它用途

- 主要用于得到估计的标准误差。
- 还提供了总体参数的近似置信区间。例如，查看幻灯片43中间面板中的直方图，1000个值的5%和95%分位数为(．43和．72)。
- 这代表了真实α的大约90%的置信区间。

bootstrap的其它用途

- 主要用于得到估计的标准误差。
- 还提供了总体参数的近似置信区间。例如，查看幻灯片43中间面板中的直方图，1000个值的5%和95%分位数为(．43和．72)。
- 这代表了真实α的大约90%的置信区间。我们如何解释这个置信区间?

bootstrap的其它用途

- 主要用于得到估计的标准误差。
- 还提供了总体参数的近似置信区间。例如，查看幻灯片43中间面板中的直方图，1000个值的5%和95%分位数为(．43和．72)。
- 这代表了真实α的大约90%的置信区间。我们如何解释这个置信区间?
- 上面的区间被称为bootstrap百分位数置信区间。这是从bootstrap获得置信区间的最简单的方法(对比许多方法)。

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？
- 为了使用bootstrap估计预测误差，我们可以考虑使用每个bootstrap数据集作为我们的训练样本，原始样本作为我们的验证样本。
- 但每个bootstrap样本都与原始数据有显著的重叠。大约三分之二的原始数据点出现在每个bootstrap样本中。

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？
- 为了使用bootstrap估计预测误差，我们可以考虑使用每个bootstrap数据集作为我们的训练样本，原始样本作为我们的验证样本。
- 但每个bootstrap样本都与原始数据有显著的重叠。大约三分之二的原始数据点出现在每个bootstrap样本中。 你能证明这一点吗？

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？
- 为了使用bootstrap估计预测误差，我们可以考虑使用每个bootstrap数据集作为我们的训练样本，原始样本作为我们的验证样本。
- 但每个bootstrap样本都与原始数据有显著的重叠。大约三分之二的原始数据点出现在每个bootstrap样本中。 你能证明这一点吗？
- 这将导致bootstrap严重低估真实的预测误差。

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？
- 为了使用bootstrap估计预测误差，我们可以考虑使用每个bootstrap数据集作为我们的训练样本，原始样本作为我们的验证样本。
- 但每个bootstrap样本都与原始数据有显著的重叠。大约三分之二的原始数据点出现在每个bootstrap样本中。 你能证明这一点吗？
- 这将导致bootstrap严重低估真实的预测误差。为什么？

bootstrap能估计预测误差吗?

- 在交叉验证中，K 折中的每一个都与其它用于训练的 $K-1$ 折不同：没有重叠。这对其成功至关重要。为什么？
- 为了使用bootstrap估计预测误差，我们可以考虑使用每个bootstrap数据集作为我们的训练样本，原始样本作为我们的验证样本。
- 但每个bootstrap样本都与原始数据有显著的重叠。大约三分之二的原始数据点出现在每个bootstrap样本中。 你能证明这一点吗？
- 这将导致bootstrap严重低估真实的预测误差。为什么？
- 反过来-原始样本=训练样本，bootstrap数据集=验证样本-更糟糕！

去除重叠

- 可以通过只对当前bootstrap样本中没有(偶然)出现的观测进行预测来部分解决这个问题。
- 但这种方法会变得复杂，最终，交叉验证为估计预测误差提供了一种更简单、更有吸引力的方法。

预验证

- 在微阵列和其它基因组学研究中，一个重要的问题是将从大量“生物标志物”衍生的疾病输出预测器与标准临床预测器进行比较。
- 在用于生成生物标志物预测的同一个数据集上比较它们，可能会导致结果严重偏向于生物标志物预测。

预验证

- 在微阵列和其它基因组学研究中，一个重要的问题是将从大量“生物标志物”衍生的疾病输出预测器与标准临床预测器进行比较。
- 在用于生成生物标志物预测的同一个数据集上比较它们，可能会导致结果严重偏向于生物标志物预测。
- 预验证可用于在两组预测之间进行更公平的比较。

激励的例子

这个问题的一个例子出现在van 't Veer等人的论文中。*Nature*（2002）。他们的微阵列数据有4918个基因，测量了78个病例，取自一项乳腺癌研究。预后良好组44例，预后不良组34例。构建“微阵列”预测器如下：

1. 选择了70个基因，与78个类别标签具有最大的绝对相关性。
2. 利用这70个基因，构建了一个最近质心分类器$C(x)$。
3. 将分类器应用于78个微阵列，每个病例i得到二分类器$z_i = C(x_i)$。

结果

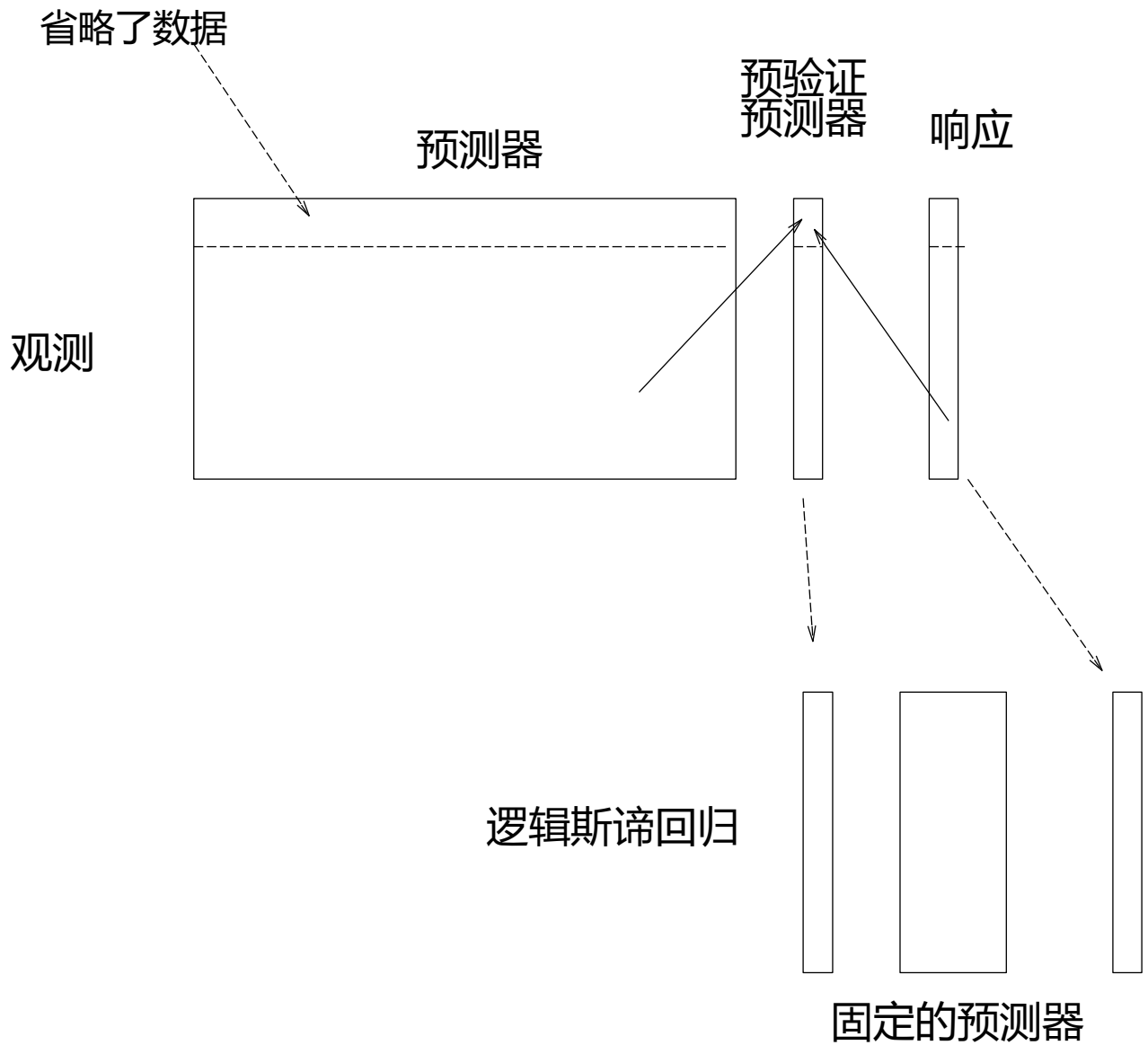
使用预后输出的logistic回归，比较微阵列预测与一些临床预测:

模型	系数	标准差	Z score	p-value
		重用		
微阵列	4.096	1.092	3.753	0.000
angio	1.208	0.816	1.482	0.069
er	-0.554	1.044	-0.530	0.298
grade	-0.697	1.003	-0.695	0.243
pr	1.214	1.057	1.149	0.125
年龄	-1.593	0.911	-1.748	0.040
大小	1.483	0.732	2.026	0.021
		预验证后		
微阵列	1.549	0.675	2.296	0.011
angio	1.589	0.682	2.329	0.010
er	-0.617	0.894	-0.690	0.245
grade	0.719	0.720	0.999	0.159
pr	0.537	0.863	0.622	0.267
年龄	-1.471	0.701	-2.099	0.018
大小	0.998	0.594	1.681	0.046

预验证背后的想法

- 被设计用来比较自适应生成的预测器与固定的、预先定义的预测器。
- 想法就是构成一个“预验证”版本的自适应预测器：具体来说，一个没有“看到”响应 y 的“更公平”的版本。

预验证过程



此例的预验证细节

1. 将病例分成 $K = 13$ 等量的部分，每部分6例。
2. 留出一份。仅使用来自其它12部分的数据，选择与类标签具有至少0.3绝对相关性的特征，并形成最近的质心分类规则。
3. 利用该规则预测第13部分的类标签
4. 对13份数据都做第2步和第3步，得到一个“预验证”的微阵列预测器 $\tilde{z}_i$，i 为78个病例的每一个。
5. 拟合逻辑斯谛回归模型到预验证的微阵列预测器和6个临床预测器。

自助法与置换检验

- 自助法从估计的总样本中抽取样本，并利用结果估计标准差和置信区间。
- 置换方法从数据的估计零分布中采样，并利用这一分布估计p值和假设检验的错误发现率。
- bootstrap可以用于在简单的情况下检验零假设。例如，如果θ = 0是零假设，我们检查θ的置信区间是否包含零。
- 也可以调整bootstrap从零分布中抽样(见Efron和Tibshirani的书《自助法导论》(1993)，第16章)，但与排列相比没有真正的优势。

自助法

估计一个感兴趣的统计量的精度

自助法的优点之一是它几乎可以被用于所有情形，而并不要求复杂的数学计算。在R中使用自助法只需要两个步骤。第一，创建一个计算感兴趣的统计量的函数。第二，用 boot 库中boot ()函数，通过反复地从数据集中有放回地抽取观测来执行自助法。

```
>alpha.fn=function(data,index){
+ X=data$X[index]
+ Y=data$Y[index]
+ return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
+ }
```

这个函数返回 (return) 或者说输出，对参数index选中的观测用公式 (5.7) 计算得到的α的一个估计。比如说，下面的命令让R用全部100个观测来估计α。

```
>alpha.fn(Portfolio,1:100)
[1] 0.576
```

下面的命令用sample() 函数来随机地从1到100中有放回地选取100个观测。这相当于创建了一个新的自助法数据集，然后在新的数据袋上重新计算 $\hat{\alpha}$ 。

```
>set.seed(1)
>alpha.fn(Portfolio,sample(100,100,replace=T))
[1] 0.596
```

自助法

可以通过多次运行这个命令，把所有相应的α估计记录下来，然后计算其标准差，来实现自助法分析。但是，boot ()函数可以让这个方法自动进行。下面产生 $R = 1000$ 个α的自助法估计。

```
>boot(Portfolio,alpha.fn,R=1000)
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = Portfolio.statistic = alpha.in, R = 1000)
Bootstrap Statistics :
      original          bias          std. error
t1*    0.5758          -7.315e-05     0.0886
```

最终的输出结果表明，对于原始数据， $\hat{\alpha} = 0.5758$ ，以及 $SE(\hat{\alpha})$ 的自助法估计为 0.0886。

自助法

估计线性回归模型的精度

自助法可以用来衡量一种统计学习方法的估计和预测的系数的波动性。下商用自助法来衡量β_0和β_1估计的波动性，这是在Auto数据集上用 horsepower来预测mpg的线性回归模型的截距和斜率项。而且将会比较用自助法和用3. 1.2节中$SE(\hat{\beta}_0)$ $SE(\hat{\beta}_1)$的公式得到的估计的区别。

首先创建一个简单的函数，boot.fn()，这个函数先输入 Auto数据集和观测序号的集合，然后返回线性回归模型的截距和斜率的估计。再将这个函数用于全部392个观测，对整个数据集用第3章的一般线线性回归系数估计的公式，来计算β_0和β_1的估计。注意一下，在函数 的开头和结尾并不需要{ 和 }，因为这函数只有一行。

```
>boot.fn=function(data,index)
+return(coef(lm(mpg~horsepower,data=data,subset=index)))
>boot.fn(Auto,1:392)
(Intercept)   horsepower
     39.936      -0.158
```

自助法

boot.fn() 函数还可以通过随机有放回地从观测里抽样，来产生对截距和斜率项的自助估计。下面给出两个例子。

```
>set.seed(1)
>boot.fn(Auto,sample(392,392,replace=T))
(Intercept)   horsepower
   38.739      -0.148


>boot.fn(Auto,sample(392,392,replace=T))
   40.038      -0.160
```

接下来，用boot ()函数来计算1000个截距和斜率项的自助法估计的标准误差。

```
>boot(Auto,boot.fn,1000)
ORDINARY NONPARAMETRIC BOQTSTRAP
Ca11 :
boot(data=Auto, statistic = boot.fn, R=1000)
Bootstrap Statistics :
       original          bias          std. error
t1*    39.936            0.0297        0.8600
t2*    -0.158           -0.0003        0.0074
```

自助法

这表明$SE(\hat{\beta}_0)$的自助法估计为0.86， $SE(\hat{\beta}_1)$ 的自助法估计为0.0074 正如在3.1.2节中讨论的那样，可以用标准公式来计算线性模型中回归系数的标准误差。这可以通过summary()函数借到。

```
>summary(lm(mpg~horsepower,data=Auto))$coef
              Estinate    Std.Error     t  value    Pr(>|t|)
(Intercept ) 39.936      0.71750     55.7        1.22e-187
horsepouer   -0.158      0.00645     -24.5       7.03e-81
```

自助法

下面计算对数据拟合二次模型所得到的标准线性回归系数的估计和标准误差的自助法估计。由于这个模型对数据的拟合效果很好(图3-8)，所以现在$SE(\hat{\beta}_0)$，$SE(\hat{\beta}_1)$和$SE(\hat{\beta}_2)$的自助法估计和标准估计更加接近了。

```
>boot.fn=function(data,index)
 +coefficients(lm(mpg~horsepower+I(horsepower^2),data=data,subset=index))
>set.seed(1)
>boot(Auto,boot.fn,1000)
ORDINARY NONPARAMETRIC BOQTSTRAP
Ca11 :
boot(data=Auto, statistic = boot.fn, R=1000)
Bootstrap Statistics :
       original          bias          std. error
t1*    56.900            6.098e-03     2.0945
t2*    -0.466            -1.777e-04    0.0334
t3*    0.001             1.324e-06     0.0001

>summary(lm(mpg~horsepower+I(horsepower^2),data=Auto))$coef
                 Estinate    Std.Error    t  value   Pr(>|t|)
(Intercept )     56.9001  |  1.80043           32   1.7e-109
horsepouer       -0.4662     0.03112          -15   2.3e-40
I(horsepower^2)  0.0012      0.00012           10   2.2e-21
```

问题？