第5章 重抽样方法

5.1 交叉验证法
5.2 自助法

01 重采样方法

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）。
- 通过反复的从训练集种抽取样本，然后对每一个样本重新拟合一个感兴趣的模型，来获取关于拟合模型的附加信息（主要是测试集的预测误差，以及参数估计的误差）；
- 它们提供了测试误差的估计值（主要指5.1交叉验证方法），以及我们的参数估计值的误差（主要指5.2自助法）。

5.1 交叉验证法

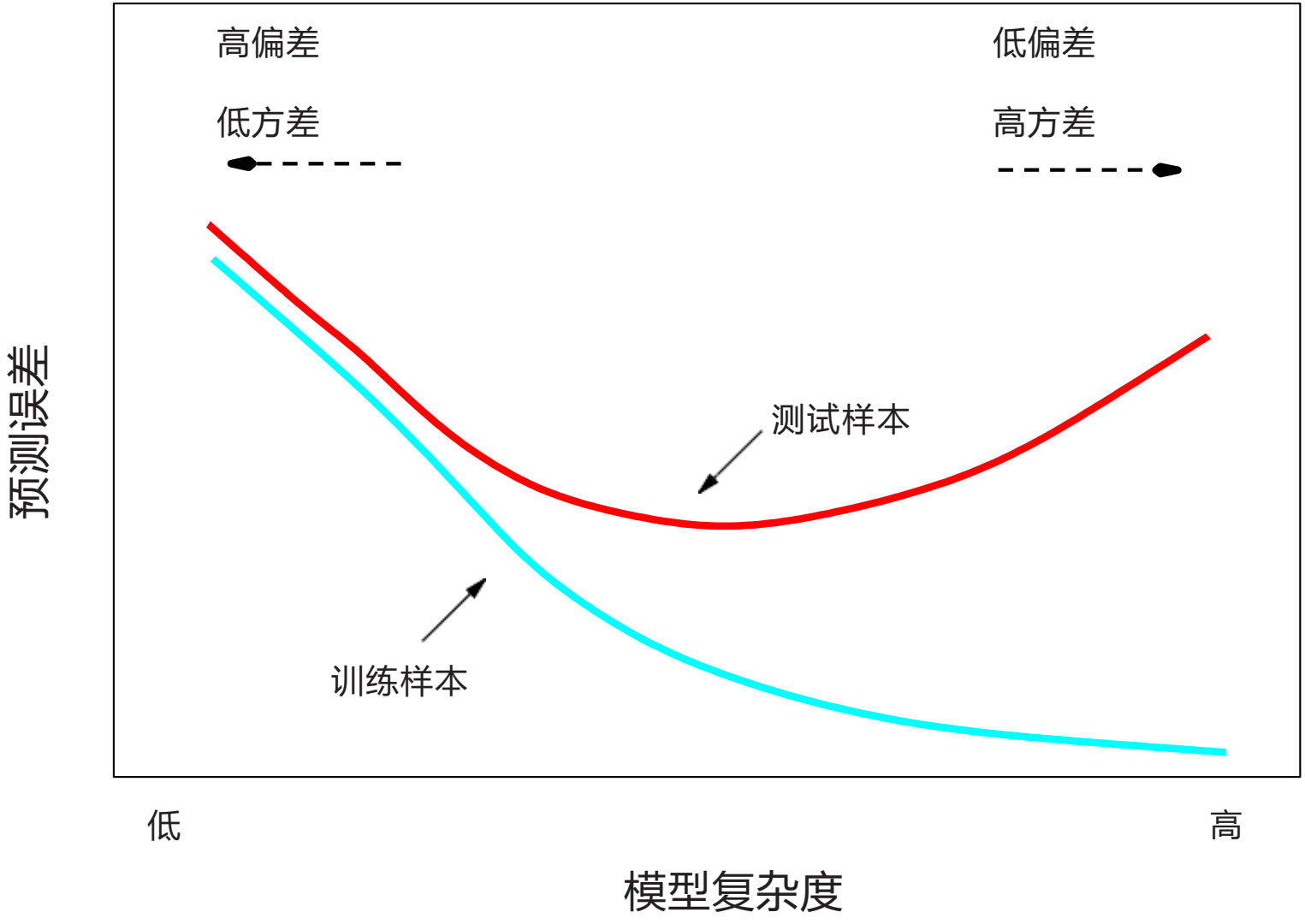
5.1.1 验证集方法
5.1.2 K折交叉验证
5.1.3 留一交叉验证法
5.1.4 交叉验证法应用与讨论
5.1.5 分类问题的交叉验证

01 回顾：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 是使用已训练得到的统计学习模型 来预测新观测的响应/输出的平均误差，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，尤其是前者可能会 极大地低估 后者。

回顾：训练集 vs. 测试集的表现



01 回顾：关于测试误差估计

大连理工大学 Dalian University of Technology

- 如何获取测试误差：一个大型专门的测试集，但，**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；
- 这里考虑另一类方法，它们通过**从训练数据集 拿出 子集**来估计测试误差，将统计学习方法应用于这些拿出的观测数据；
- **所以，测试误差的估计**，是本小节的**直接目标**。但是：检验真实数据时（模型实际使用时），测试误差的真值是没有的，因此很难衡量哪种估计方法是有效的。所以，只能通过模拟数据的方法找到哪种方法（留一法、验证法等）对测试误差的估计更准确。
- **更重要、更进一步的目标是**：使用本节介绍的测试误差的估计方法，进一步找到哪种统计学习方法的测试误差可能最小。

5.1.1 验证集方法

5.1.1 验证集的方法

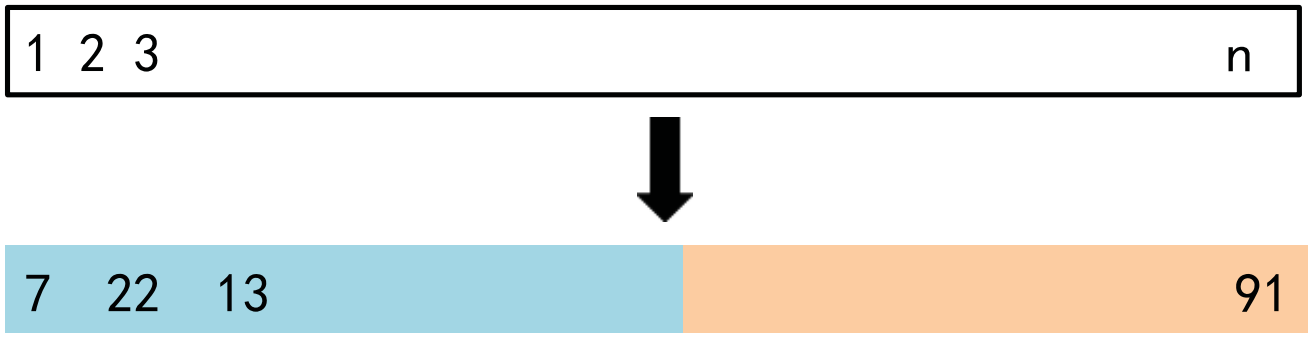
- 在这里，我们将可用的样本集随机分为两部分（大小相当）：

  训练集 和 验证集（或称：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。

- **验证集的误差**提供了**测试误差的估计值**。通常，回归问题时使用均方误差估计（Mean Square Error估计，MSE）；在分类问题时使用误分类率评估。

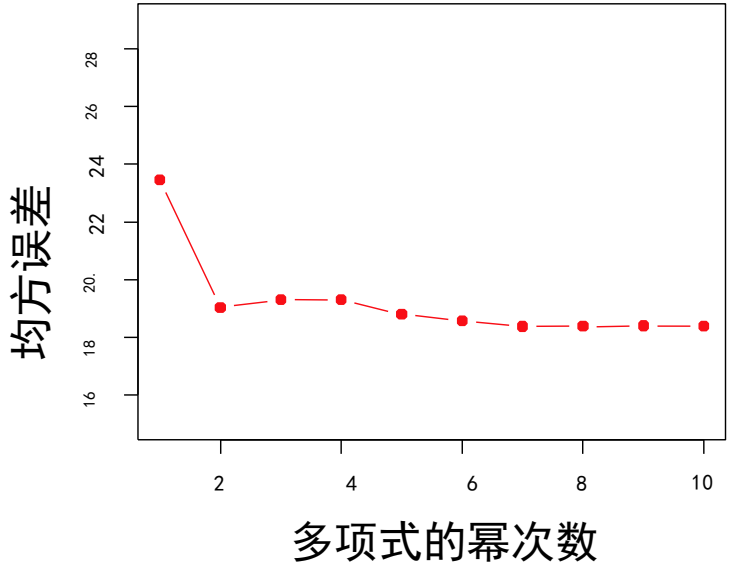
5.1.1 验证集的方法（分割示例）



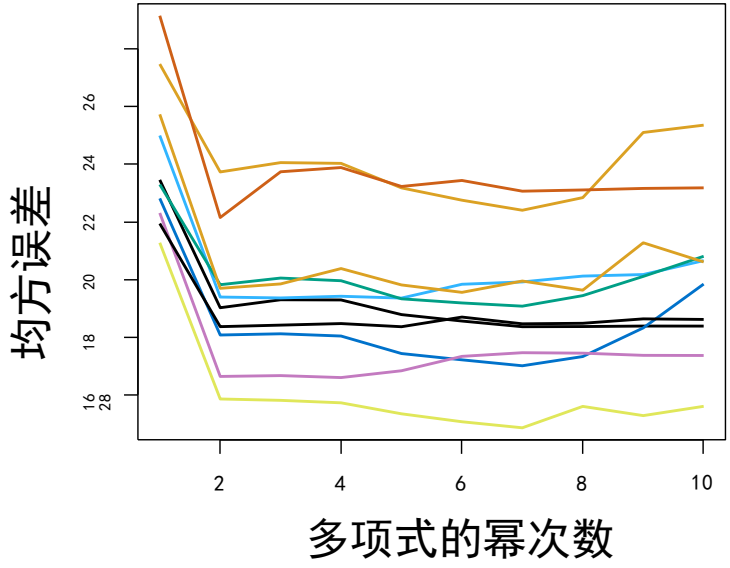
随机分成两部分：左侧为训练集，右侧为验证集

5.1.1 验证集的方法（汽车数据集）

- 目标：使用验证集方法验证：线性回归和高阶多项式哪个更好，以及如果是高阶多项式，包含到多少次幂的多项式的拟合较好；
- 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。分别测试加上不同多项式时的模型拟合后测试的结果，结果如左图。
- 重复10次（分割+测试），结果如右图。



左图：一种分割方法



右图：十种分割方法重复十次

5.1.1 验证集的方法（优缺点）

- 验证集方法估计的测试错误率的**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。
- 主要原因：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。

为什么?

5.1.1 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。

5.1.1 验证集方法

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
```

这两个错误率分别为19. 82和19. 78。
即，线性模型、 +二次项、 +二三次项的错误率分别为： 26.14、 19. 82和19. 78

结论： 用horsepower的二次函数来拟合的模型果比只用horsepower的线性函数拟合模型的效果更好；但包含三次函数后的区别不大；

5.1.1 验证集方法

用**另一种分割方法**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。

```
>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
```

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更好，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。

5.1.2 K-折（K-fold）交叉验证

01 K-折（K-fold）交叉验证

大连理工大学 Dalian University of Technology

- 估计测试误差的**广泛使用的方法**。
- 估计结果可以**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**。然后，**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_k。
- 对**每一个**$k = 1, 2, \ldots, K$ **重复**上述步骤，然后，将**结果**进行**组合**。

01 K-折（K-fold）交叉验证

大连理工大学 Dalian University of Technology

将数据分成 K 个**大小基本相等**的部分（这里 $K = 5$）

1	2	3	4	5
验证	训练	训练	训练	训练
训练	验证	训练	训练	训练
训练	训练	验证	训练	训练
训练	训练	训练	验证	训练
训练	训练	训练	训练	验证

K-折交叉验证的误差估计 （1）

- 令 K 个部分为$C_1, C_2, \ldots, C_K$，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；
- 依次保留每一折用作验证，其他折用作训练，得到各折的MSE 。
- **具体的，**对于 C_k 中的**观测样本*i***(共n_k个样本)，拟合值$\hat{y}_i$是使用**不包含C_k**（即：预留的第 k 部分观测样本集）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，可得各折的MSE为：

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

- K折交叉验证的均方误差，由**各折误差的加权平均**得到：

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

K-折交叉验证的误差估计 （1）

特别地，如果样本总数 N 是 k 的倍数，可得$n_k = N / K$。

可以得到

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

5.1.3 留一交叉验证法

01 留一交叉验证法

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**。

- 也即：**LOOCV**留出1个观测用于验证，剩下的所有观测用于训练模型；用得到的模型，在预留的那1个观测上进行验证。
- **重复*N*次**，也就是逐一预留各个观测，然后用其他观测训练模型并求预留观测的误差，取平均后，即可LOOCV下的测试误差估计；
- 模型训练**用到了**绝大多数的数据，但**没有用到**预留的那唯一一个观测，所以，提供了对“**这个观测的误差**”的**渐进无偏估计**；

LOOCV快速计算

- LOOCV原本需要进行***N*次模型拟合**。但是，如果使用最小二乘线性或多项式回归，数学上，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为杠杆统计量(教材P68，公式3.37)，反应了一个观测对他自己拟合值的影响。

由计算方式可见，类似普通的MSE计算方法，仅系数1-h_i有区别。

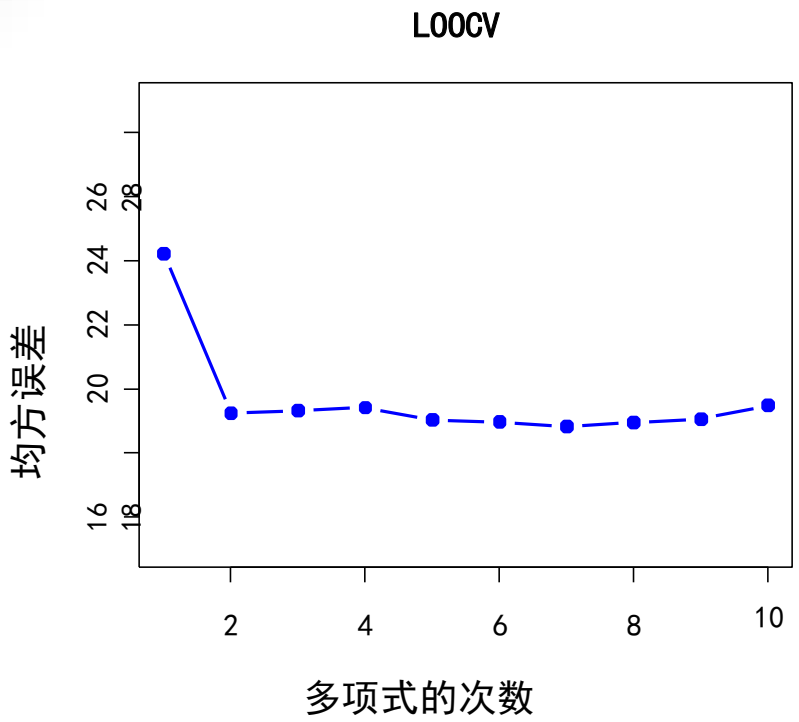
所以：LOOCV的误差估计成本与单个模型拟合的成本相同

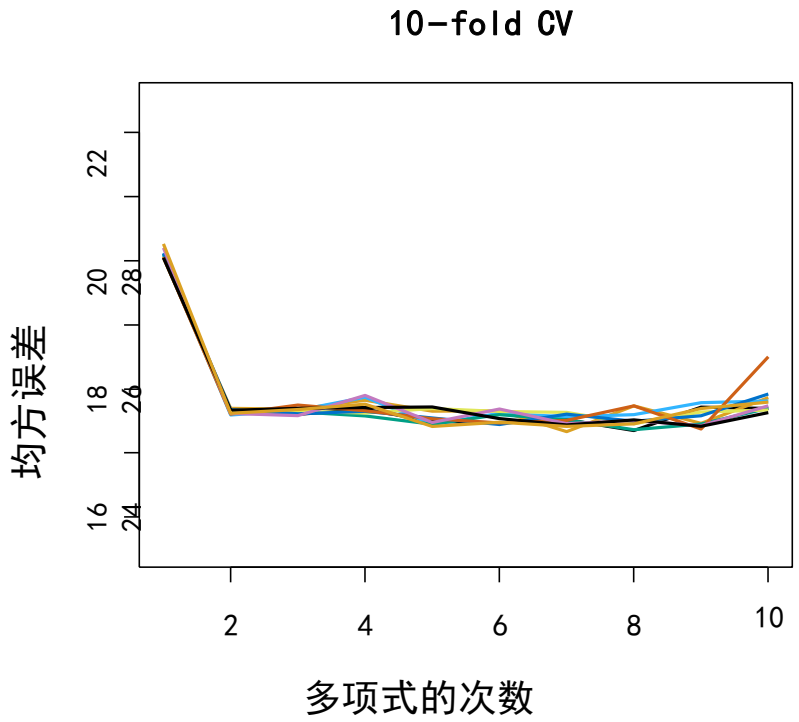
- 由于在拟合过程中没有用到预留的那个观测，所以$(y_i - \hat{y}_i)^2$提供了对测试误差的一个渐进无偏估计；但，虽然是无偏的，却是对**预留的那仅一个的观测**计算得出，具有**高度的波动性**。

5.1.4 交叉验证法应用与讨论

01 K-折交叉验证（AUTO数据集）

大连理工大学 Dalian University of Technology





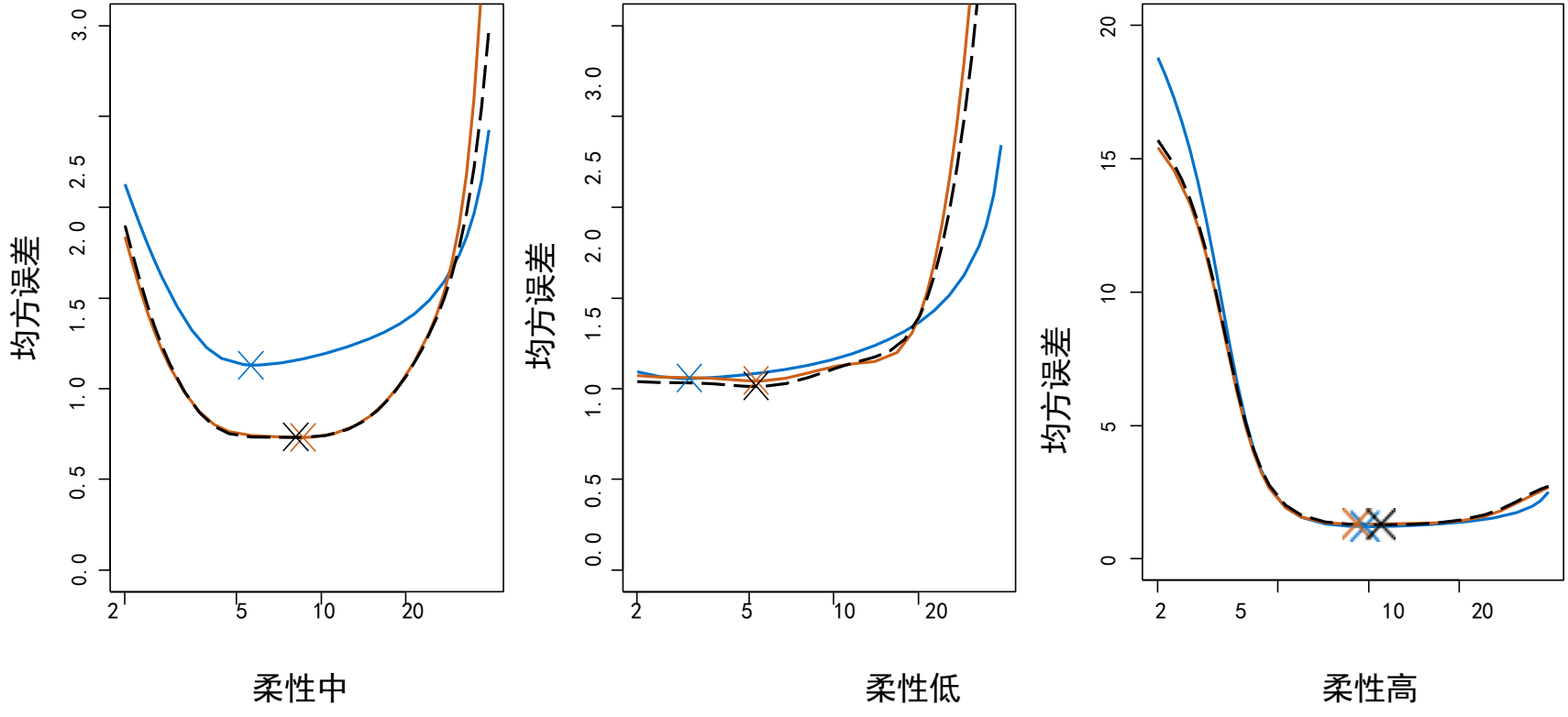
用CV法估计“用horsepower的多项式函数来预测mpg”所产生的测试误差。左图LOOCV；右图为**9次运用10折CV**，可见结果略微不同，但这个波动通常比验证集方法（差不多对半分）的波动小得多；

要注意：LOOCV的结果不随机，只有一个值；而验证集方法、K折交叉验证可以重复多次（采用不同的分割方法），从而得到很多个值

02 K-折交叉验证（不同柔性数据集测试）

大连理工大学 Dalian University of Technology

下图左中右分别是第2章图2-9，2-10，和2-11，其真实的曲线柔性水平为中-低-高。



在不同的数据集上，不同模型的真实预测误差（蓝色）与利用LOOCV和10折CV估计的预测误差（橙色、黑色），可见LOOCV和10折CV非常接近。

03 交叉验证的其它问题

大连理工大学 Dalian University of Technology

- 验证集方法中一部分数据用于训练、一部分用于测试；LOOCV方法，仅一条用于测试，其余用于训练；*K* 折交叉验证位于二者之间；
- 错误率的偏差：实际的错误率应是建立在所有数据用于训练所得到的模型上，基于此：LOOCV方法使用最多的数据用于训练，所以，能提供近似无偏的测试误差估计；而验证集方法往往会高估测试的错误率，因为约50%的数据没有用于训练；*K*折交叉验证居于二者之间；
- 错误率的方差：验证集方法中，因为用于训练的观测的随机性很大，所以估计的错误率的波动很大；而，LOOCV的每个模型都几乎是在相同观测集上进行（总观测-1），这些结果之间高度相关，且每次用仅一个样本进行测试，波动性也很大；而*K*折交叉验证各模型之间的训练样本重叠部分相对较小，其对测试率方差的估计的波动较小；
- 综合来看：从经验上看，选择$K=5$或$K=10$，使得测试错误率不会有过大的偏差或方差。

5.1.5 分类问题的交叉验证

01 分类问题的交叉验证

大连理工大学 Dalian University of Technology

- 我们将数据分成 K 个大小大致相等的部分 $C_1, C_2, \ldots, C_K$，其中 C_k 表示第 k 部分观测样本集合，并设其有n_k个观测，如果样本总数 N 是 k 的倍数，可令$n_k = N / K$。
- 依次保留各折作为测试集，其他折作为训练集，得到各折的错误率

$$Err_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k。$$

- K-折交叉验证错误率为：

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

- $\mathrm{CV_K}$的估计标准差为

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

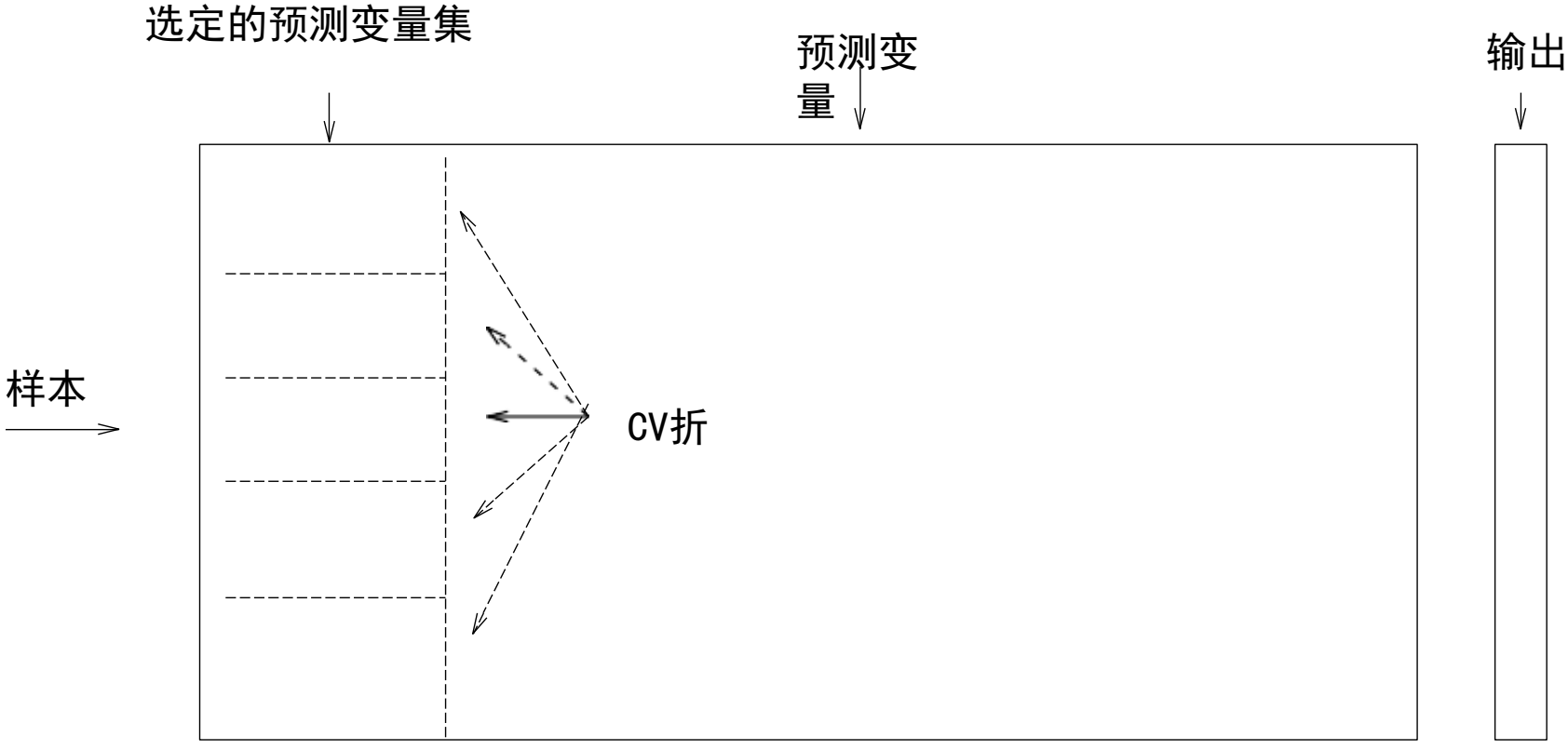
交叉验证: 对与错 (1)

- 考虑一个应用于50个样本的二分类数据的简单分类器:
  1. 从**5000个预测变量**和50个样本（全部的样本）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。

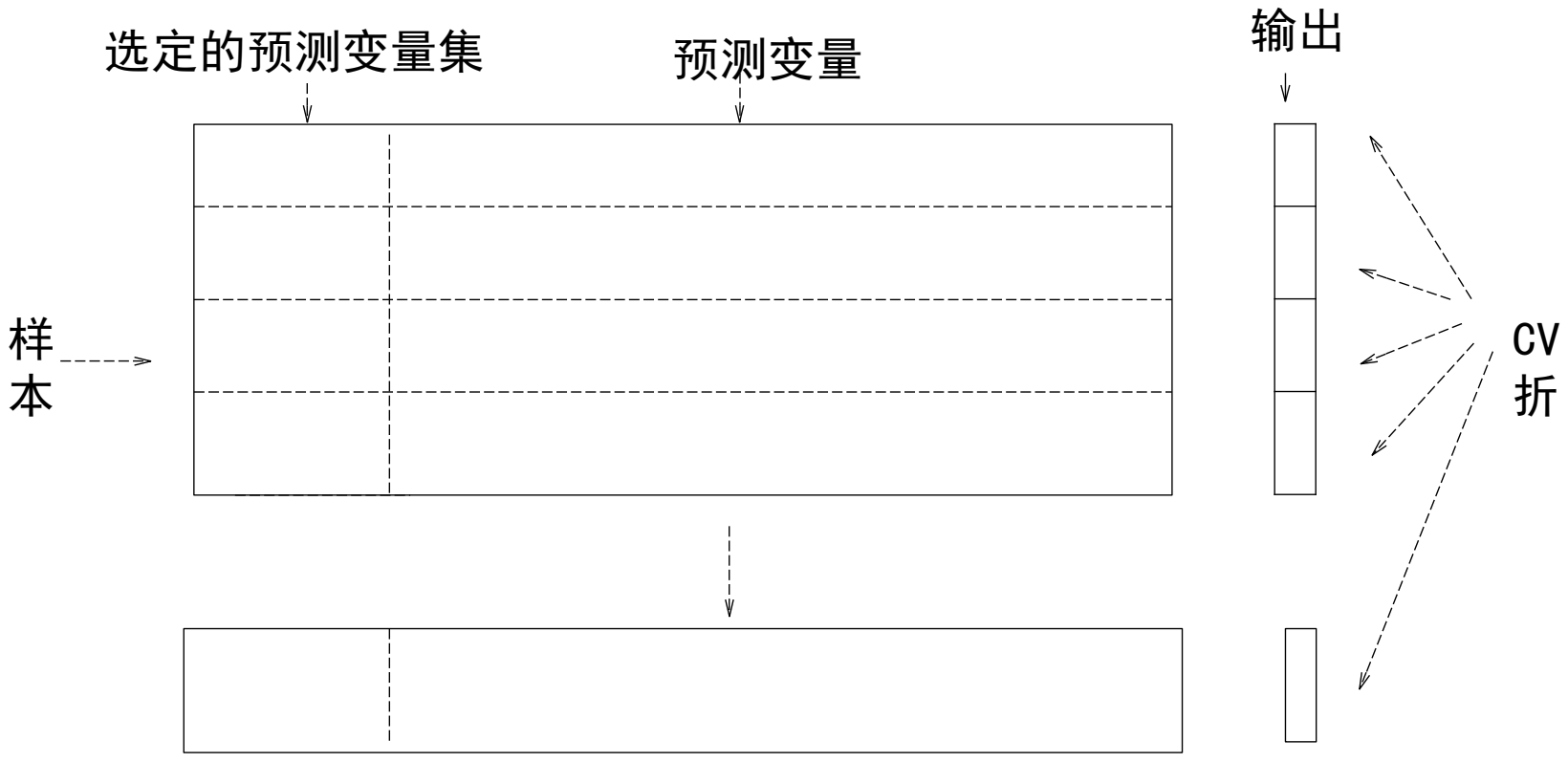
我们如何评估这个分类器的测试集表现?

我们是否可以在第2步中应用交叉验证，忘记第1步?

错误的方式



正确的方式



06 K-折（K-fold）交叉验证

大连理工大学 Dalian University of Technology

CV.glm() 函数同样可以用于实现K折CV。令K=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
```

K折交叉验证法的计算时间要比LOOCV的计算时间少得多。(理论上来说，由于有 LOOCV 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。

5.2 自助法

5.2 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量或统计学习方法的不确定性。

- 例如，使用该方法，可以衡量一个估计量的不确定性（标准差或置信区间）。（相对应的，交叉验证主要是用于估计测试误差）

02 一个简单的例子（投资组合）

大连理工大学 Dalian University of Technology

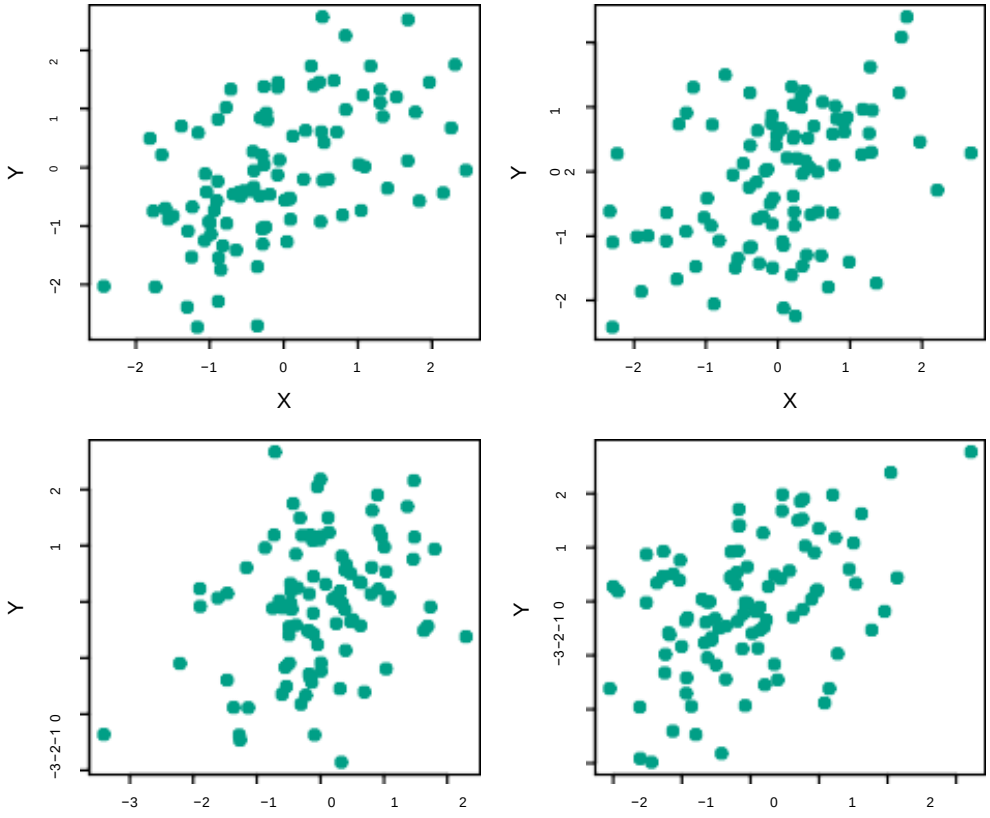
- 假设我们希望将一笔固定数额的钱投资于两种金融资产，它们的收益率分别为X和Y，其中X和Y是随机量。
- 我们将资金中的一小部分 α 投资于X，剩余的1-α 投资于Y。
- 我们希望选择 α 来最小化我们投资的总风险 (或方差)。换句话说，我们希望最小化Var(αX + (1 - α)Y)。
- 可以证明，使风险最小化的值是

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

式中σ^2 为方差， σ_{XY} 为协方差。

- **但是这些方差、协方差值是未知的；需要估计他们。**

03 一个简单的例子（投资组合）



- **假设我们**知道总体的参数为：$\sigma_X^2 = 1, \sigma_Y^2 = 1.25, \sigma_{XY} = 0.5$；所以我们知道 α 的**真实值**是0.6
- 为了分析，我们**模拟**生成了四套数据，每套都是由上述总体中生成的，包含100个(x, y)对，如图。

- 上图中的每一张图包含了**模拟**生成的X 和 Y 的 100 个回报，即：包含100个点，每个点可视为某一天X和Y的回报率。
- 从左到右，从上到下，α 的结果估计为 0.576, 0.532, 0.657, 0.651 。

一个简单的例子（投资组合）

- 为了估计α的标准差 $\hat{\alpha}$ ，我们重复上述模拟100对X和Y的过程，重复进行1000次，所以可估计 α 也是1000次。
- 我们由此得到了α的1000个估计值，我们可以称之为$\hat{\alpha}_1$ ，$\hat{\alpha}_2$ ，…， $\hat{\alpha}_{1000}$ 。

一个简单的例子（投资组合）

- 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**，我们预计的 $\hat{\alpha}$ 与 α 平均相差约为 0.08 。

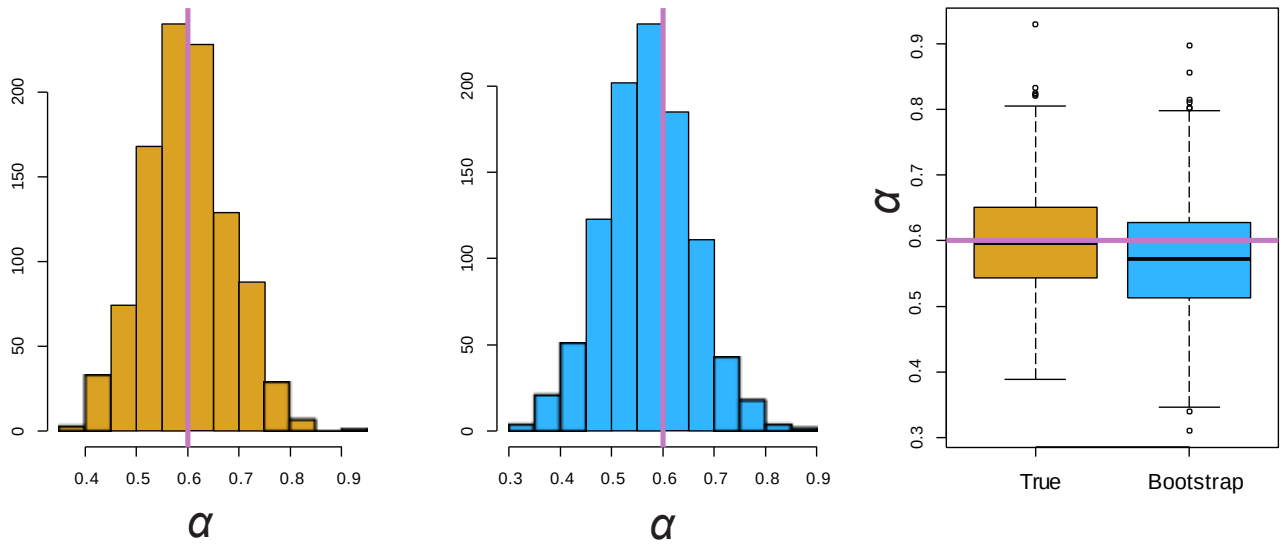
现实的真实情况

- 对于真实情况，上述方法不能使用，因为我们**无法从原始总体中生成新样本**。（我们不知道总体的公式，而且如果我们知道总体的公式，也就不需要执行上述过程了）
- bootstrap方法允许我们**模拟获得新数据集的过程**，从而可以在不产生额外样本的情况下评估我们估计的不确定性（误差）。
- **具体的**：bootstrap不是反复地从总体中获得独立的数据集（因为不知道总体的公式），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” 的大小原始数据集**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。

04 一个简单的例子（投资组合）

大连理工大学 Dalian University of Technology

- 对于投资组合的例子，使用Bootstrap方法估计$\hat{\alpha}$得到的结果（执行1000次Bootstrap方法）用蓝色柱状图表示，**此时，$SE_B(\hat{\alpha}) = 0.087$。**

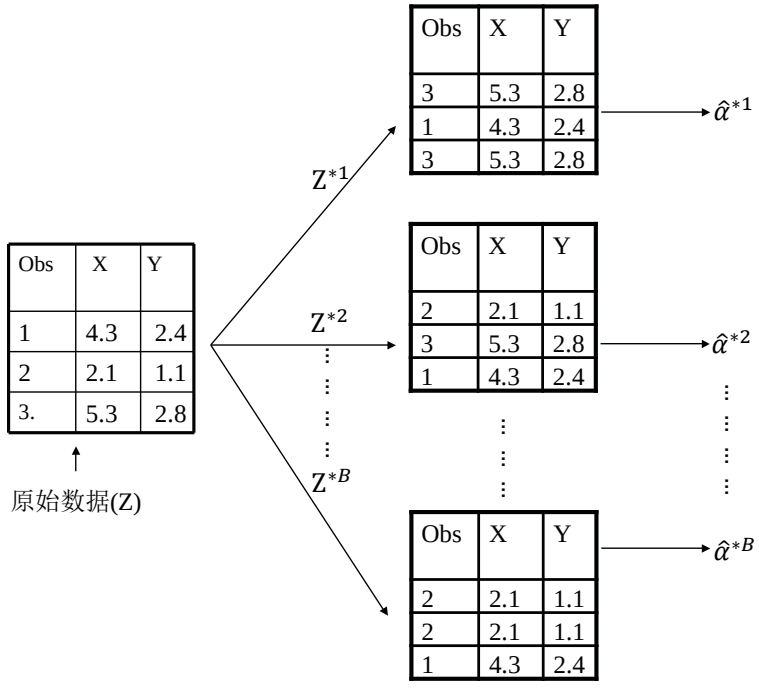


左: 从真实集中生成1000个仿真数据集得到的α估计值直方图。

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。

Bootstrap方法 (只有3个样本的例子)



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。

Bootstrap方法

- 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

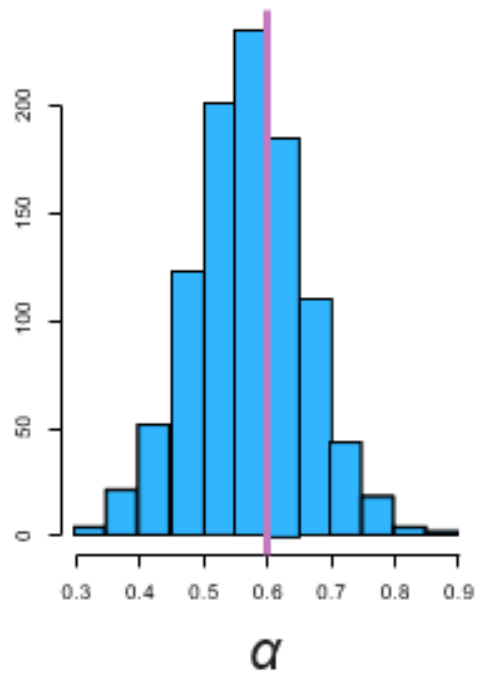
$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^{*}\right)^2}\ .$$

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

- 主要用途：是用于得到**估计值的标准误差**。
- 还提供了总体参数的近似置信区间。例如，查看幻灯片中间面板中的直方图，1000个值的5%和95%分位数为 (0.43和0.72)。
- 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)。



09 bootstrap能估计测试误差吗?

大连理工大学 Dalian University of Technology

- 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!（因为验证样本全部出现在训练样本中）

注：对数据集D采样 m 次生成训练集 D',没被采到的概率是(1-1/m)^m，lim m->infinity! 为1/e=0.368，D/D'作为测试集!

10 bootstrap能估计测试误差吗?

大连理工大学
Dalian University of Technology

- 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。

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)))
+ }
```

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

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。

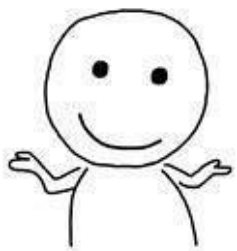
```
>boot.fn=function(data,index)
 +coefficients(lm(mpg~horsepower+I(horsepower^2),data=data,subset=index))
>set.seed(1)
>boot(Auto,boot.fn,1000)
ORDINARY NONPARAMETRIC BOQTSTRAP
Ca11 :
boot(data=Auto, statistic = boot.fn, R=1000)
Bootstrap Statistics :
       original          bias          std. error
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
```

本周作业（第六周）

教材5.4习题3、4、6、8、9

教材6.8习题1



今天你对作业爱理不理
明天它就让你补的飞起