Resampling Methods

重抽样方法

重抽样Resampling

- 重抽样方法在现代统计学与数据科学领域是不可或缺的重要工具
- 重复地从训练数据集中抽取数据，并基于这些数据拟合模型，以此来获得关于模型的额外信息
- 例如，为了估计一个线性回归模型的稳定性，我们可以从训练数据集中重复地抽取几个不同的子集，并测试得到的线性回归拟合的波动程度
- 这样的方法允许我们获取到一些基于单一的训练数据集拟合的模型得不到的信息
- Cross-Validation & Bootstrap
- 前者可以用来估计测试误差，以此来评估方法表现，或选择合适的超参数
- 后者可以用来提供参数估计准确度的度量

Outline

- Validation Set Approach
- Leave-One-Out Cross-Validation
- k-Fold Cross-Validation
- Bootstrap
- KNN

Cross-Validation

Motivation

- 假设我们正在尝试从几个不同的模型中进行选择。
- 例如，我们正在使用多项式回归模型：

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k$$

- 现在需要从 $k = 0,1,\ldots,10$ 中选择一个最合适的 k 值。
- 我们怎么才能自动地选出一个模型，很好的平衡偏差和方差？

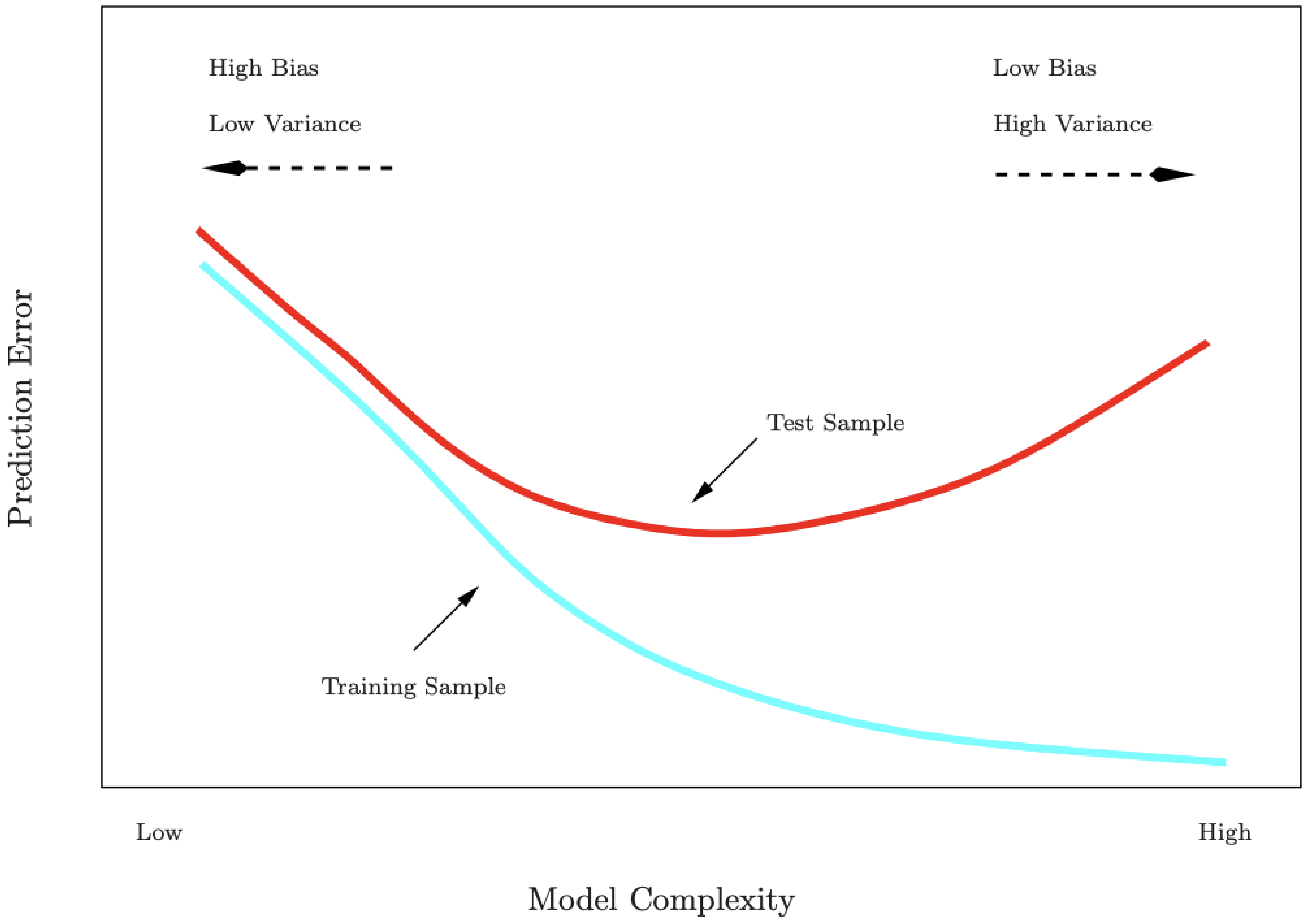
Motivation

- 不失一般性，假设我们有一组备选模型$\mathcal{M} = \{M_1, \dots M_k\}$和训练数据集 S
- 我们可以：
  - 在训练数据集S上训练备选模型M_i，获得映射函数f_i
  - 选择训练误差最小的映射函数
- 显然，这个方法并不可行。
- 考虑选择多项式的次数。 多项式的次数越高，对训练数据集拟合越好，因此训练误差越低。因此，这个方法总是会选择一个高方差的高阶多项式模型。
- 怎么办？Cross-Validation

训练误差与测试误差

- 回顾训练误差与测试误差的区别：
- 测试误差是使用机器学习模型预测新的（未出现过的）观测的响应变量时的平均误差，这是一个在训练该模型时没有用过的度量。
- 相反，通过将机器学习方法应用于训练数据，可以很容易地计算出训练误差。
- 训练误差通常与测试误差有很大不同，训练误差可能会大大低估测试误差。
- 给定一个数据集，如果一个机器学习方法的测试误差更低，则表现更好。

训练误差与测试误差


High Bias
Low Variance
Low Bias
High Variance
Test Sample
Training Sample
Prediction Error
Low
High
Model Complexity

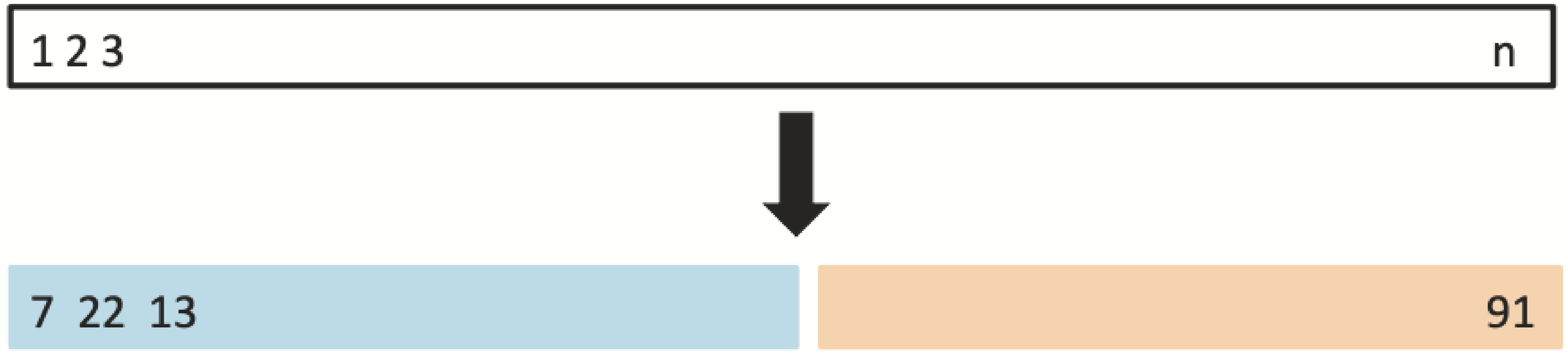
训练误差与测试误差

- 最优解决方案：一个很大的测试数据集
- 但这通常不现实

- 想法：保留一部分训练数据免于模型拟合。然后，我们将所拟合的机器学习模型应用到这些保留的数据当中，估计测试误差

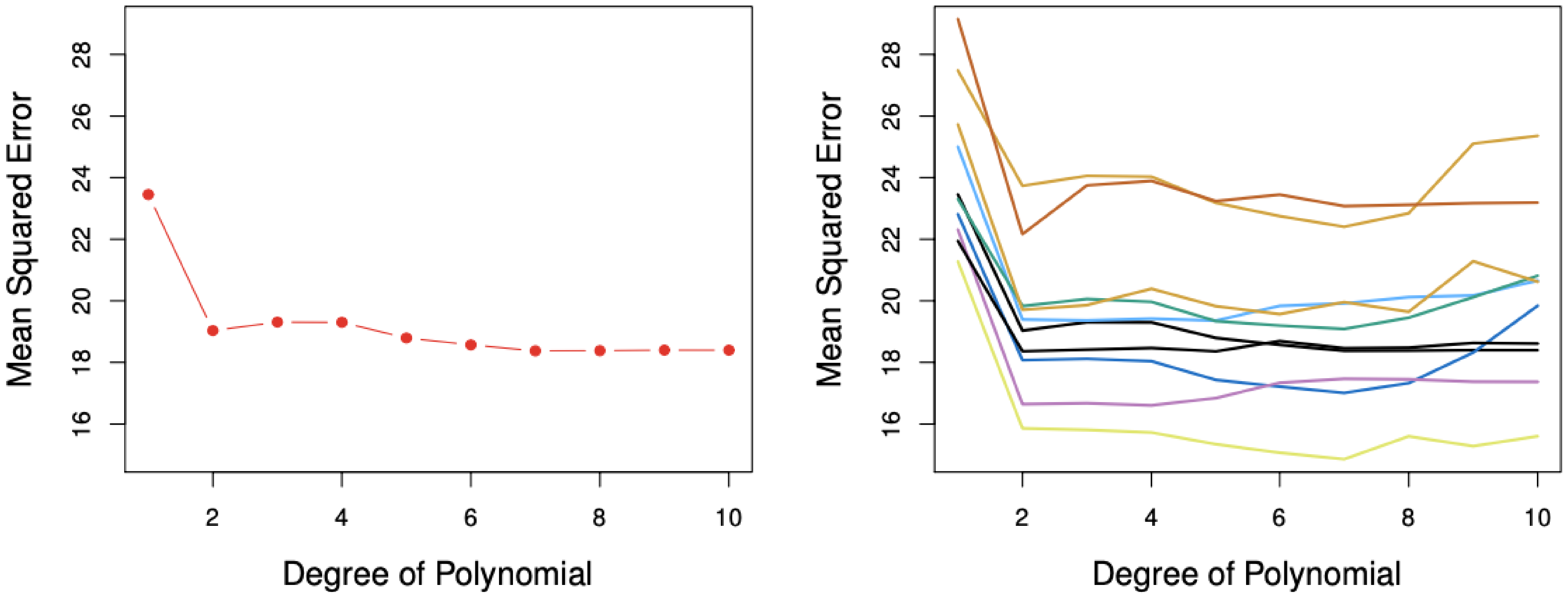
The Validation Set Approach （VSA）

- VSA是实现这一目标最简单的方法
- 它将所有数据随机地分成两部分：一个训练集，一个验证集(validation set)
- 我们在训练集上拟合模型，然后用拟合的模型来对验证集做预测
- 得到的验证误差（如MSE、分类错误率）对测试误差提供了一种估计方法



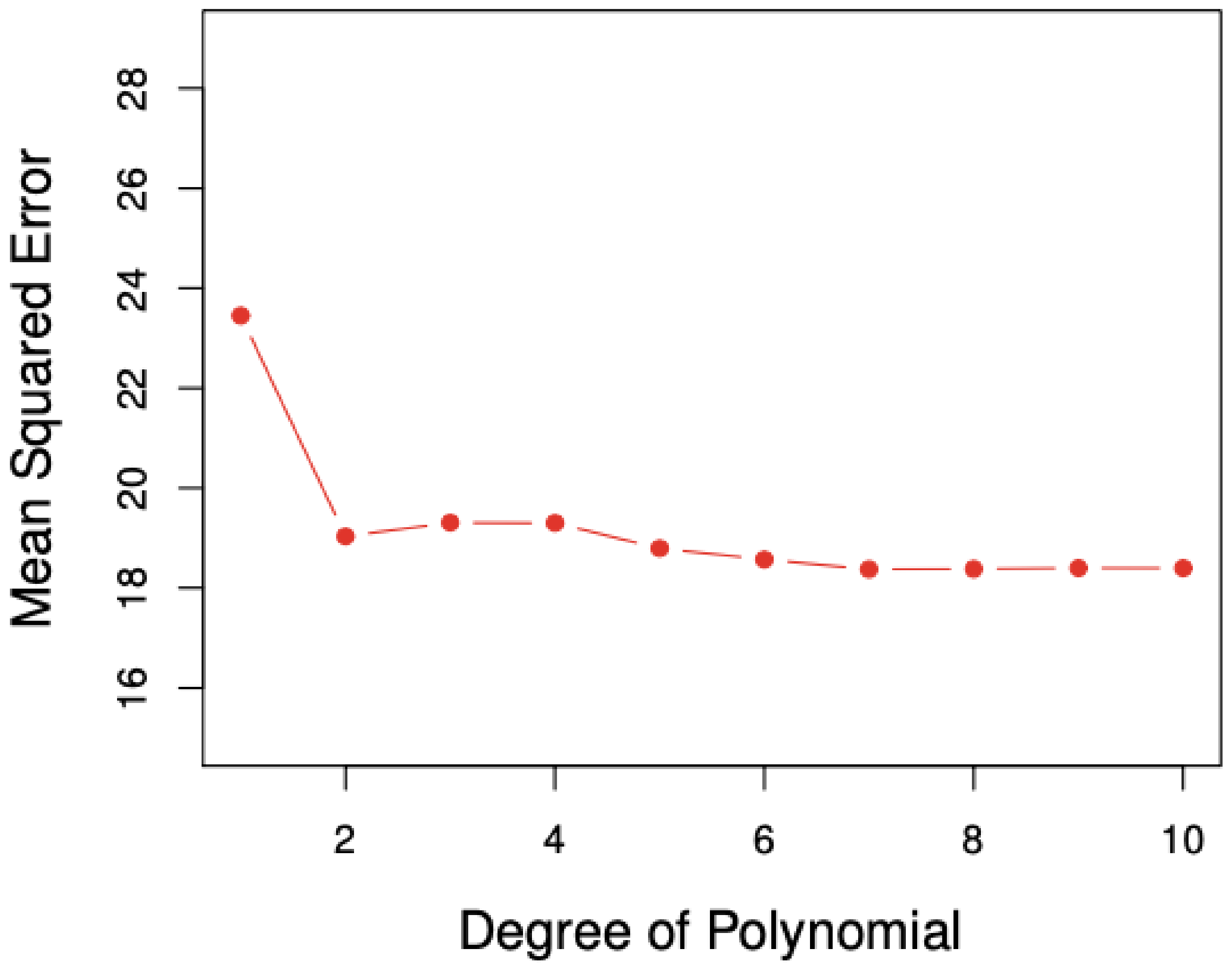
VSA案例

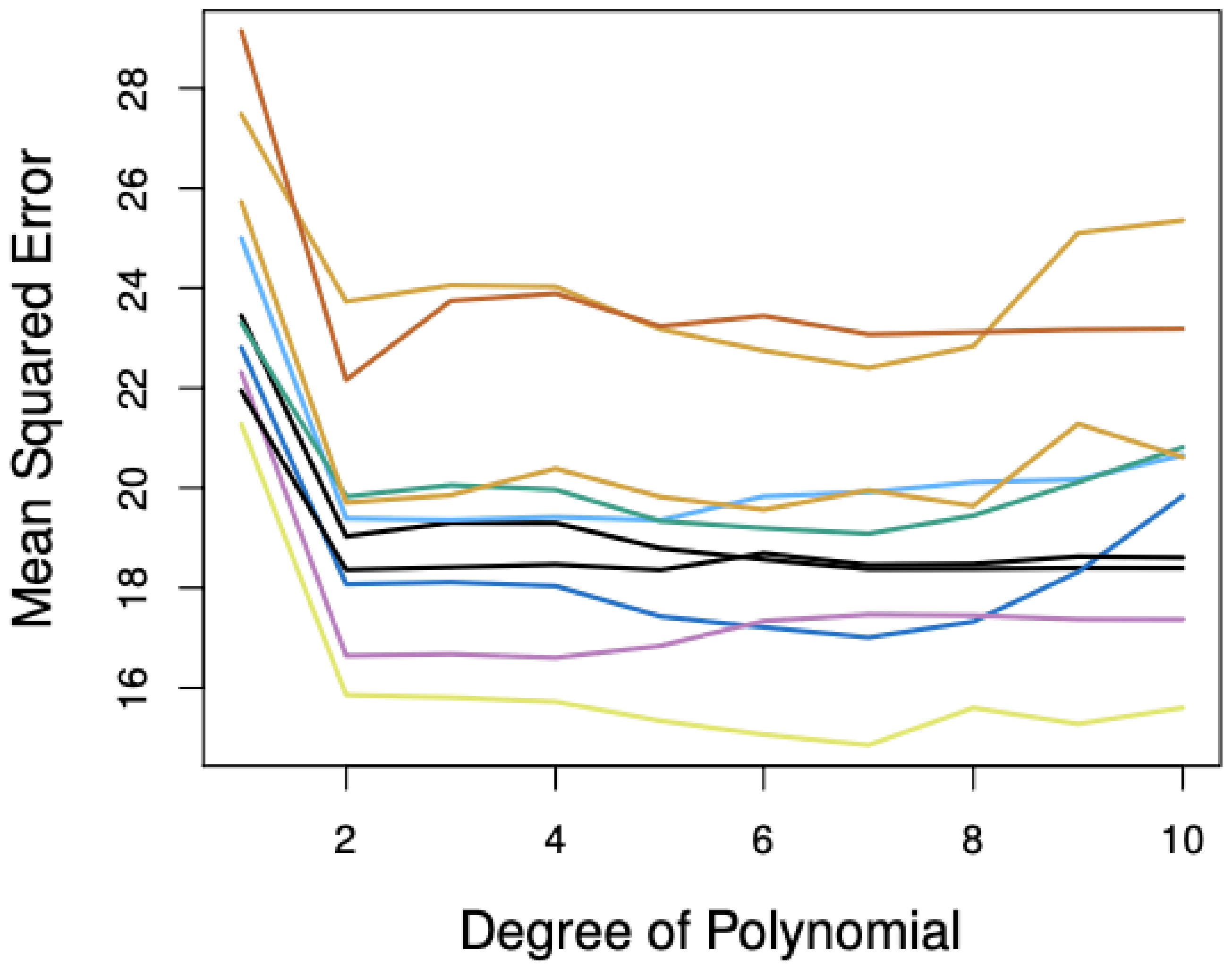
- 我们针对Auto数据集来示范VSA方法，通过horsepower的多项式来预测mpg
- 将392个观测随机分成两半，各包含196个观测
- 验证集MSE如左图所示，二次函数明显优于线性函数
- 左图展示了单次随机划分，右图展示了多次随机划分



VSA案例

- 多次随机划分的结果趋势类似，但波动很大，取决于具体哪些观测被分到了训练集和测试集
- 另外，每次运行VSA时，只有一部分观测被用来训练模型。
- 当训练数据变少时，方法的表现会变差，这也预示着验证误差会**高估**利用所有数据所得到的测试误差





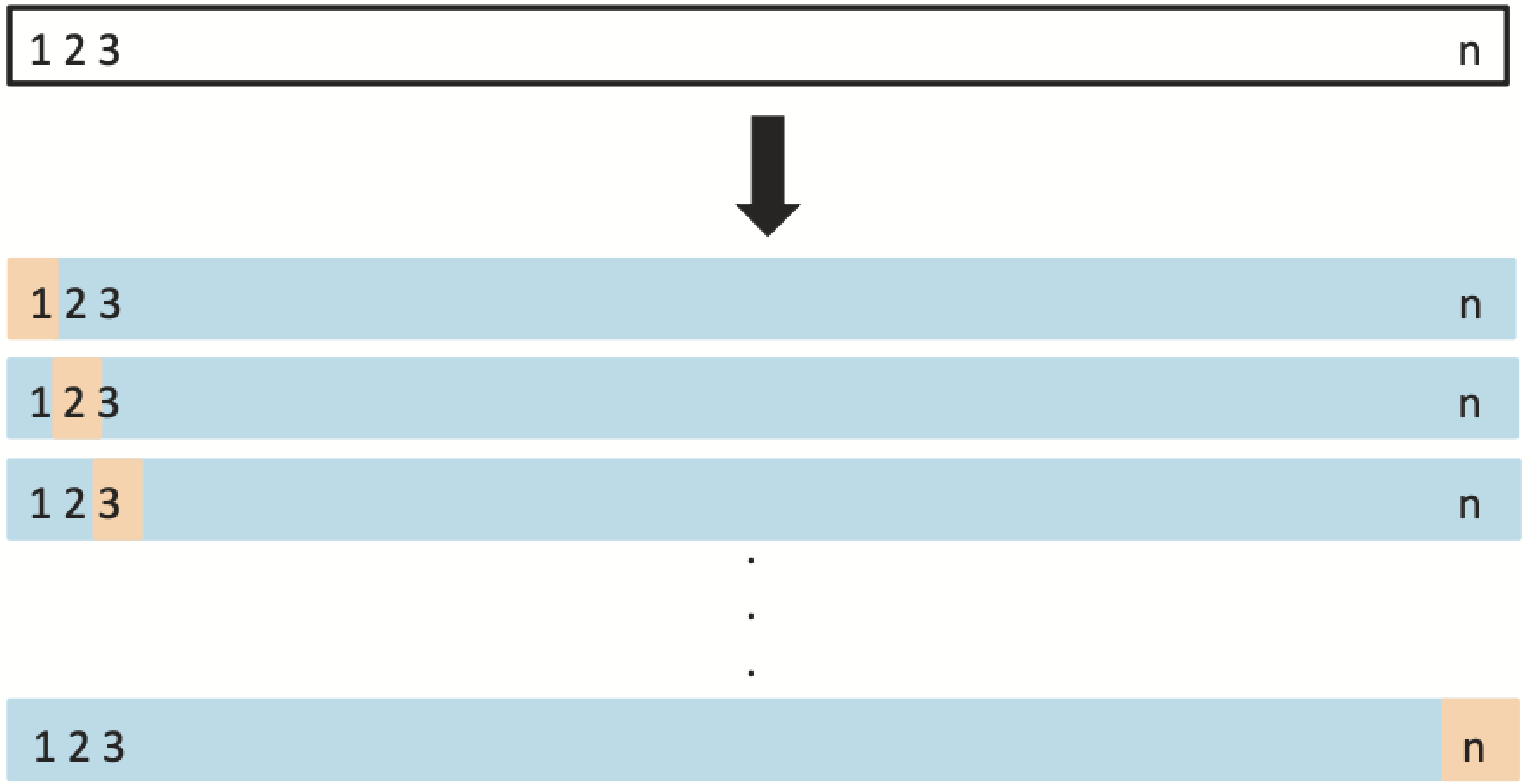
Leave-One-Out Cross-Validation (LOOCV)

- LOOCV和VSA类似，都是将数据分成两部分，训练集和验证集。
- 然而，验证集中只包含一个数据(x_1, y_1)，剩余数据 $\{(x_2, y_2), \ldots, (x_n, y_n)\}$ 构成了训练集
- 基于 $n-1$ 个数据构成的训练集，我们拟合机器学习方法，并基于 x_1 得到验证集的预测 $\hat{y}_1$
- 由于(x_1, y_1)在训练过程中没有使用，$MSE_1 = (y_1 - \hat{y}_1)^2$可以看作是测试误差的近似无偏估计
- 尽管如此，这个估计并不好，因为是基于一个观测得到的，所以波动巨大

LOOCV

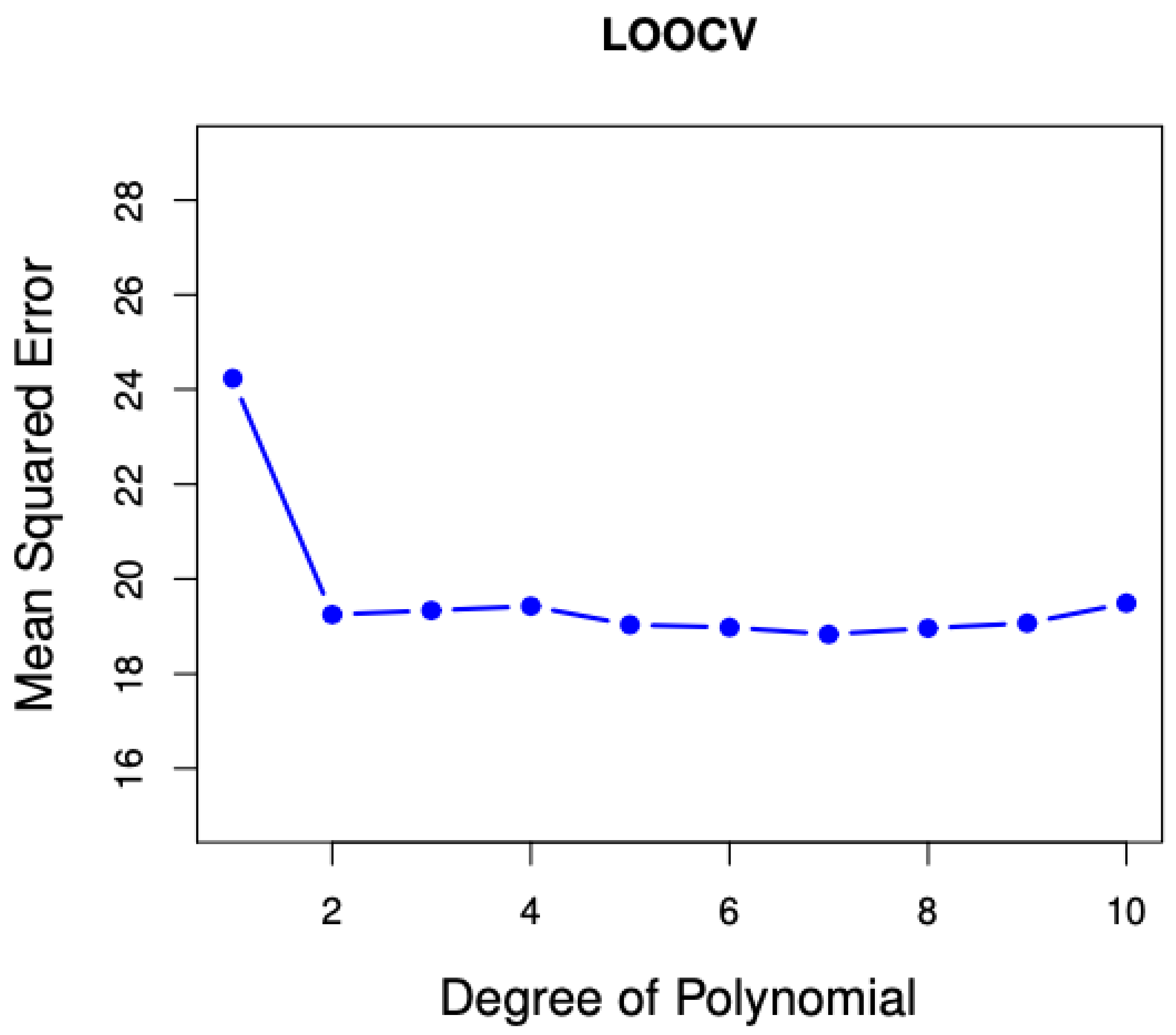
- 我们可以重复上述过程 n 次，得到$MSE_1, \ldots, MSE_n$
- 测试误差的LOOCV估计即为这 n 个测试误差的平均：

$$\mathrm{CV}_{LOOCV} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$



LOOCV优缺点

- 在LOOCV中，我们用$n-1$个训练数据来拟合模型，因此与VSA用 $n/2$ 相比，更不容易高估测试误差
- 因为其遍历性，在训练验证集的拆分上没有任何随机性，运行多次LOOCV只会得到一样的结果
- 可以用来评估任何方法，如逻辑回归、高斯判别分析等等
- 缺点：计算难度高，要拟合n次模型， n很大时计算复杂

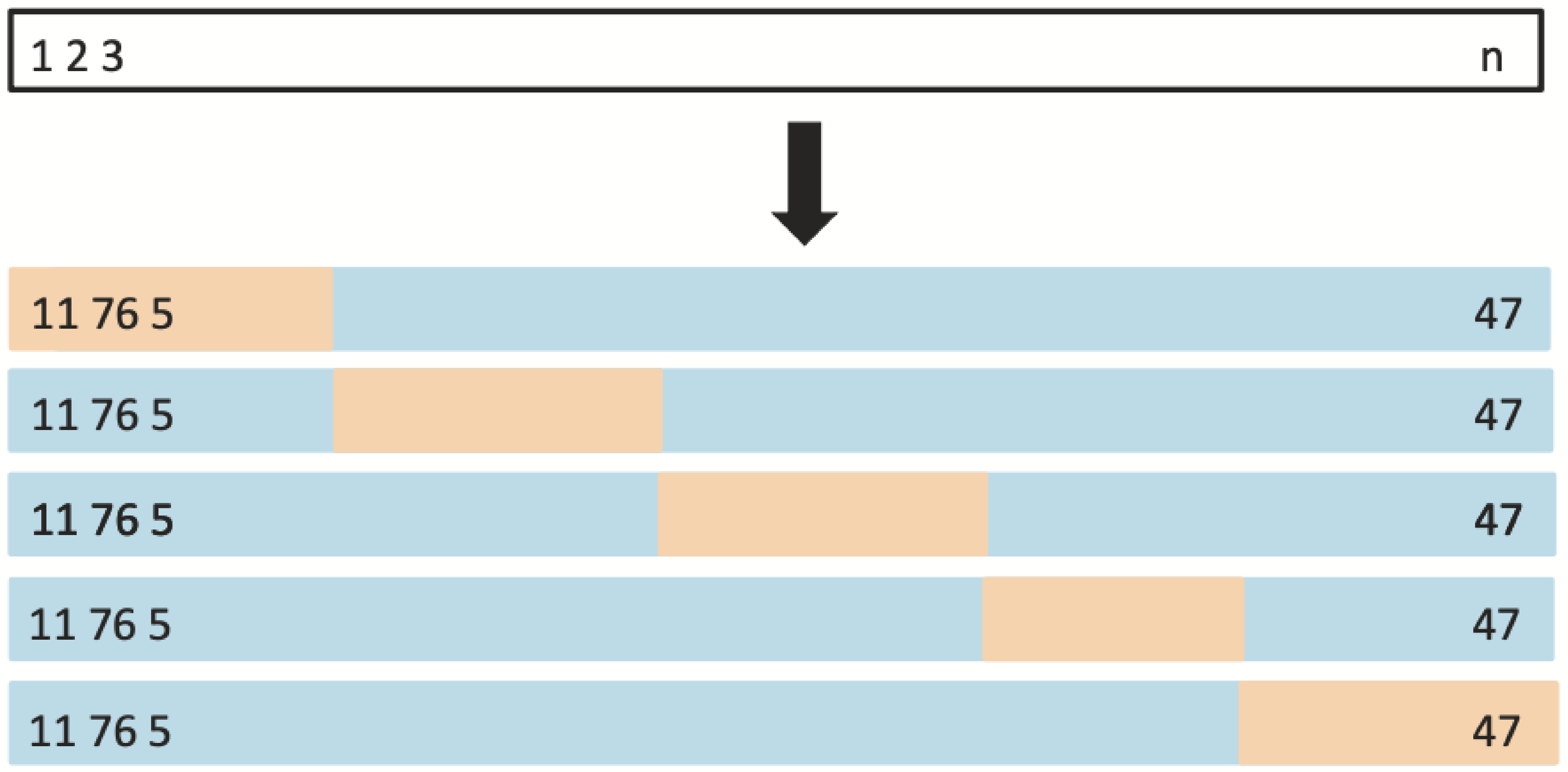


k-Fold Cross-Validation

- k折交叉验证是指将数据平均分为k份，其中 $k-1$份作为训练集，而另外的1份作为验证集。

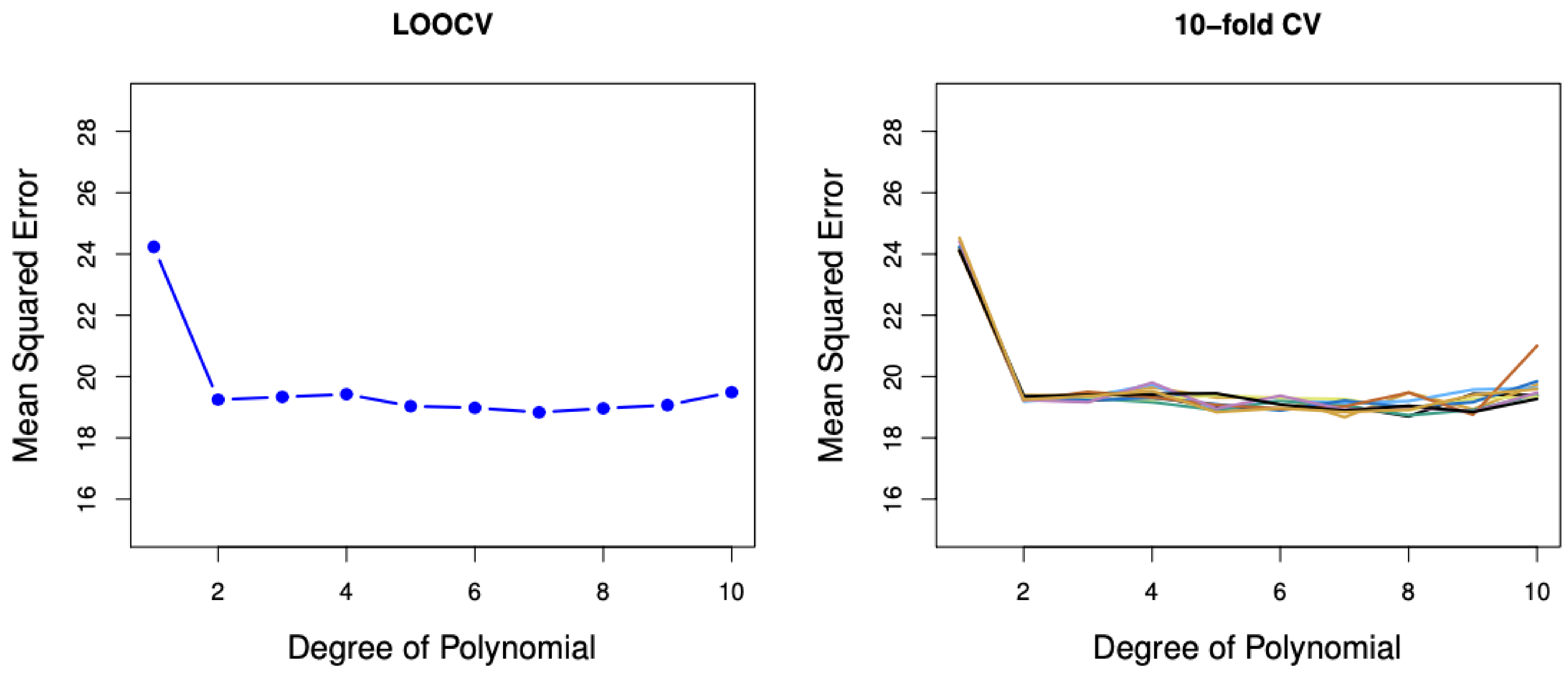
$$\mathrm{CV}_{\mathrm{KFCV}} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

- 5折交叉验证的示意图：



k-Fold Cross-Validation

- 不难发现，LOOCV是k折交叉验证的特例：令$k = n$， n折交叉验证
- 在实际中，我们常使用$k = 5$或$k = 10$
- 这样做最大的优势就是计算简便，尤其是对一些训练复杂的机器学习方法
- 右下图为多次10折交叉验证的结果展示，波动性已经非常小
- k折交叉验证的另一个优势是偏差方差的平衡



k-Fold Cross-Validation

- 除了计算优势以外， k折交叉验证的另一个优势是偏差方差的平衡。因此相对LOOCV而言，对测试误差的估计更加精准
  - VSA会高估测试误差，因为使用的训练数据少了很多；
  - LOOCV会得到测试误差的近似无偏估计，因为每次训练使用几乎全部训练数据；
  - k折交叉验证在两者之间，每次训练使用$\frac{(k-1)n}{k}$个数据。从降低偏差的角度，LOOCV优于k折交叉验证
- 然而，当$k < n$时，LOOCV比k折交叉验证有着更高的方差
- 当我们实施LOOCV时，我们对n个模型取平均，每个模型几乎都是基于同样的数据训练得到，导致结果高度正相关，测试误差的估计方差也随之增大。
- 而k折交叉验证很好地缓解了这个问题，每个模型的训练数据变得没有那么相似。

交叉验证其他用途

- 在实施交叉验证时，我们的目标可能是给定一个机器学习方法，看其表现好坏。即，估计其测试误差
- 有些时候，我们只在意估计的测试误差曲线的**最低点**，以此来选择最优的方法
- 这常常发生在我们比较**多个不同的机器学习方法**或者**同一个方法不同的超参数**的情况下，以此来选出最优的方法或超参数

Bootstrap

Motivation

- Bootstrap方法是一个广泛应用和非常有效的统计工具，来量化估计的不确定性
- 一个最简单的例子，bootstrap可以用来估计线性回归系数的标准误
- 这在线性回归中似乎并不稀奇，因为大多统计软件都可以自动输出系数的标准误
- 然而，Bootstrap可以轻易应用在很多方法，而其中很多方法的标准误很难输出

BOOTSTRAP：例子

- 假设我们希望将固定金额的资金投资于两种金融资产，分别产生X和Y的回报，其中X和Y是随机数。
- 我们会将一小部分α投资于X，并将剩余的 $1-\alpha$投资于Y 。 由于这两种资产的回报存在可变性，我们希望选择α以最小化我们投资的总风险或方差。
- 换言之，我们想要最小化 $Var(\alpha X+(1-\alpha)Y)$。
- 可以证明使风险最小化的值由下式给出：

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

- 实际中， σ_X^2， σ_Y^2， σ_{XY}都是未知的。我们可以使用X和Y的历史观测计算它们的估计，记为$\hat{\sigma}_X^2$，$\hat{\sigma}_Y^2$，$\hat{\sigma}_{XY}$。从而估计α：

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

BOOTSTRAP：例子

- 右图展示了估计方法在模拟数据集上的应用。
- 在每个图中，我们模拟了投资 X 和 Y 的 100对回报。 我们使用这些回报来估计 $\hat{\sigma}_X^2$，$\hat{\sigma}_Y^2$，$\hat{\sigma}_{XY}$，然后我们将其代入以获得$\hat{\alpha}$。
- 每个模拟数据集产生的$\hat{\alpha}$在 0.532 到 0.657 之间波动。
- 我们很自然的想要量化对α估计的准确度。

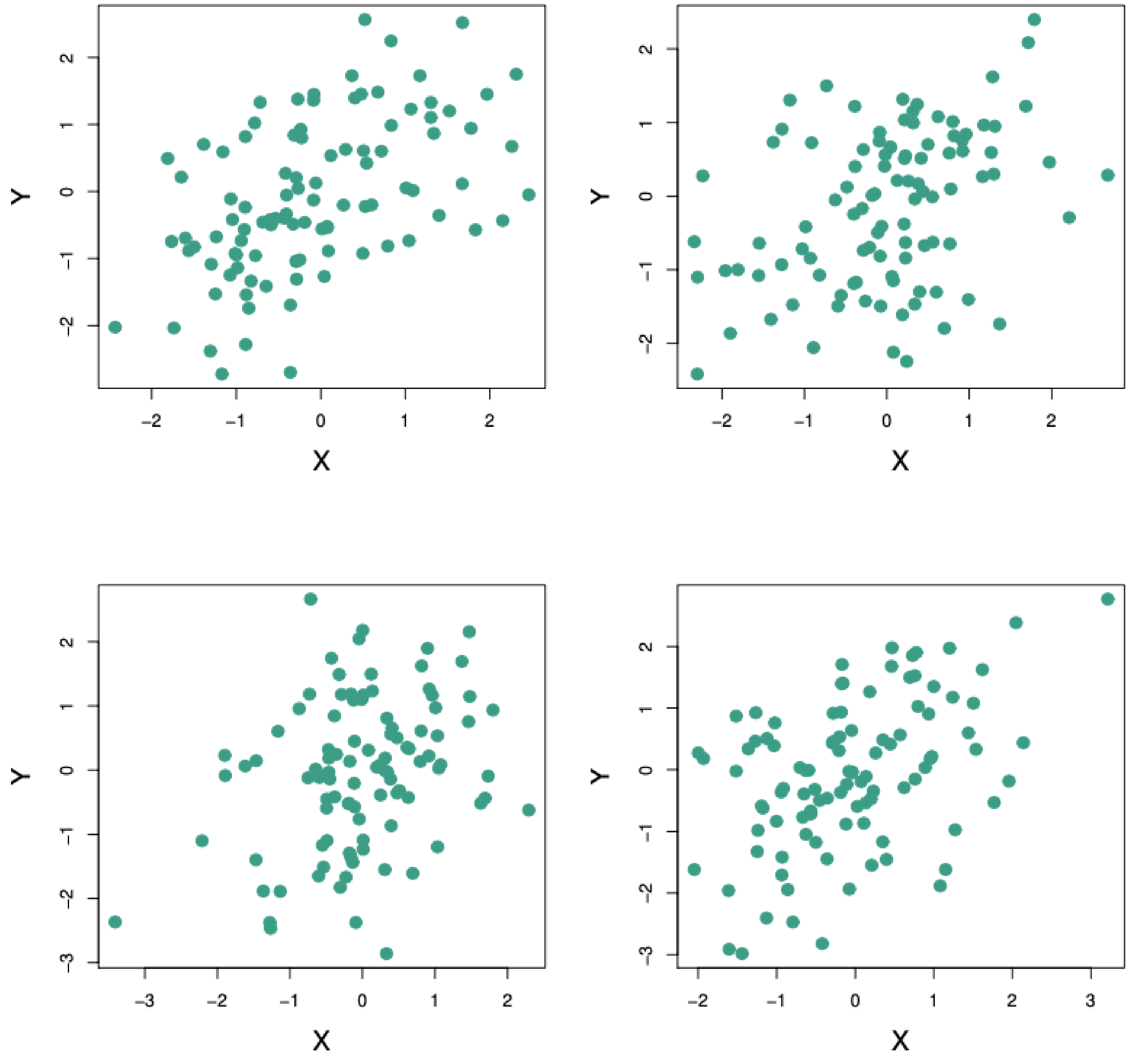


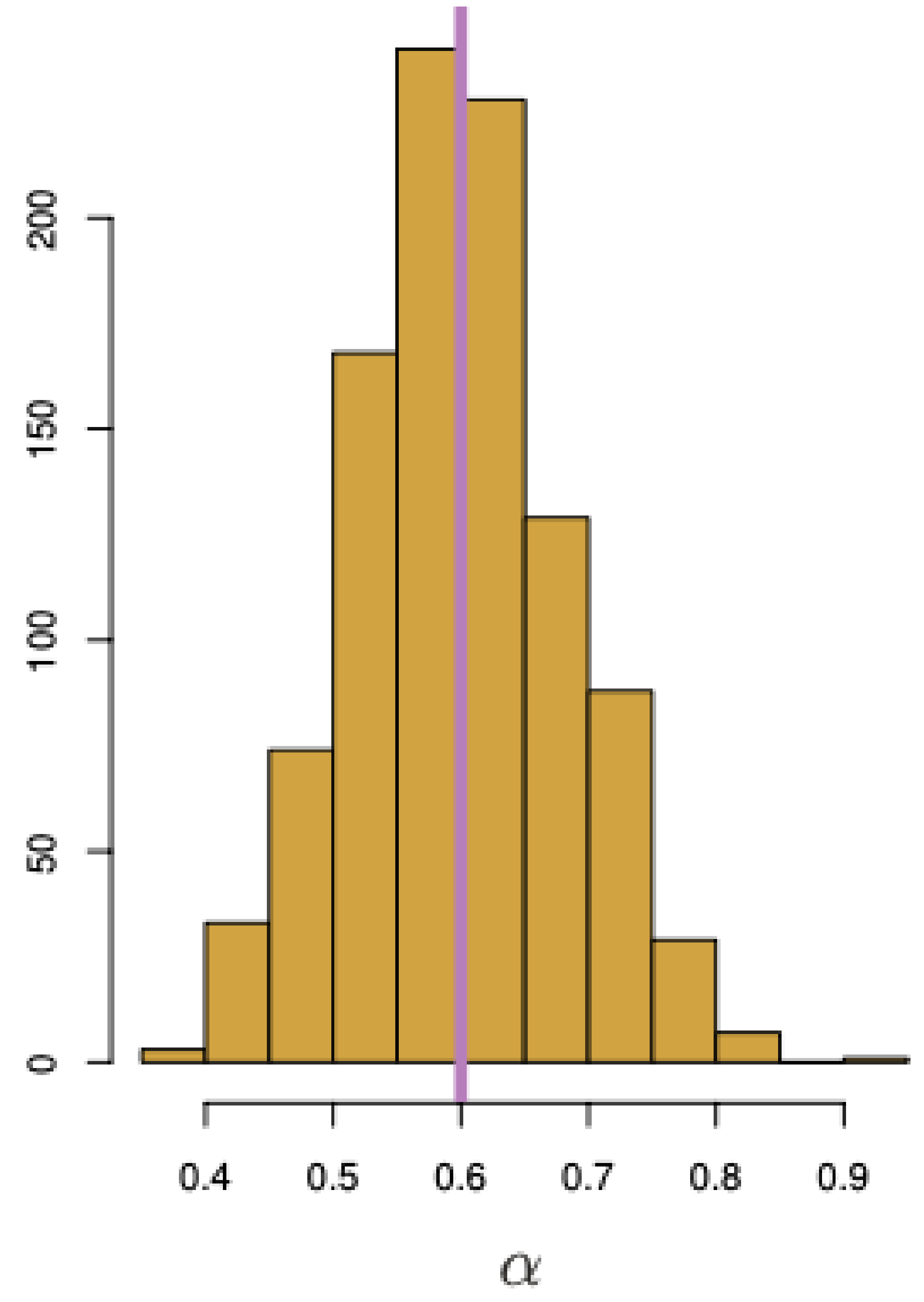
FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y*. From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

BOOTSTRAP：例子

- 我们很自然的想要量化对α估计的准确度。为了估计$\hat{\alpha}$的标准差，我们重复了1000次模拟 X 和 Y 的 100 对观察值的过程，并估计α，得到：$\hat{\alpha}_1, \ldots, \hat{\alpha}_{1000}$。
- α的真实值为0.6。1000个α估计的均值和标准差为：

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

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



- 这给了我们关于α估计的准确度的很好的感觉：$SE(\hat{\alpha}) \approx 0.083$。
- 这意味着对于一个总体中抽出的随机样本，我们可以粗略的认为$\hat{\alpha}$与α之间的差距平均在0.08。

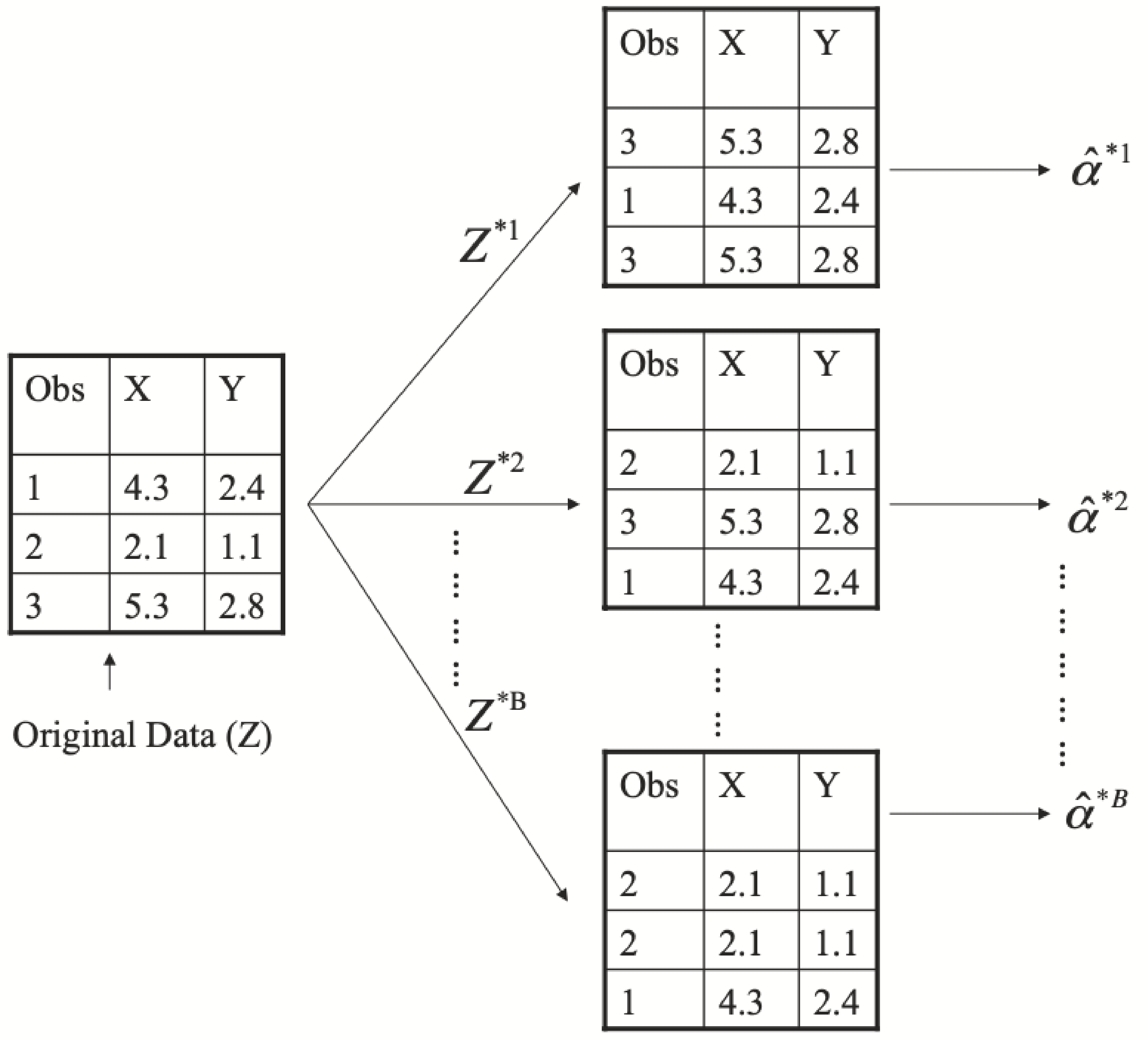
BOOTSTRAP

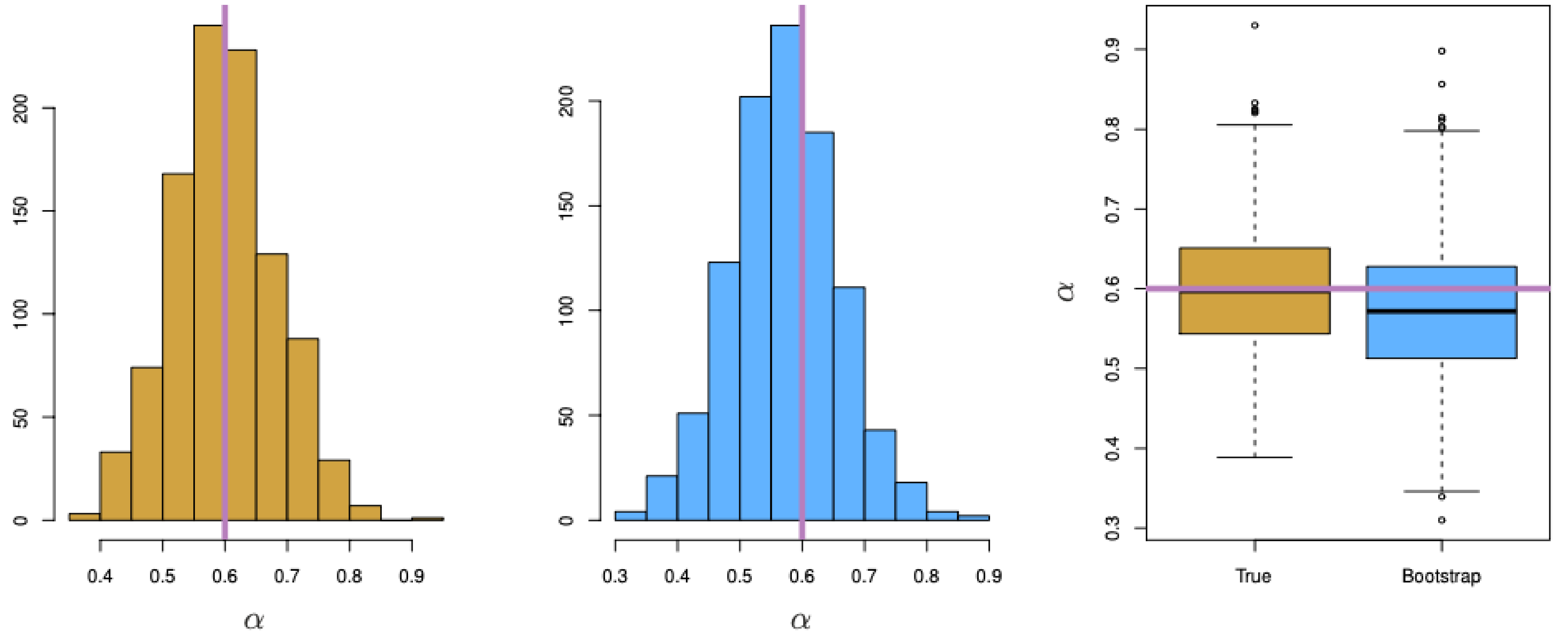
- 然而在实际中，上述方法不可用，因为我们不能从总体中不断生成新的样本。
- Bootstrap允许我们使用计算机来模拟获取新样本的过程，这样我们就可以在不生成额外样本的情况下估计$\hat{\alpha}$的可变性。
- 我们不是从总体中重复获取独立的数据集，而是通过从原始数据集中重复抽样来获取不同的数据集。
- 思想：把有限总体看作无限总体，从中有放回的抽取样本，来近似从无限总体中抽取样本的过程

BOOTSTRAP

- 右图展示了Bootstrap在一个只有3个样本的数据集Z上的应用
- 我们从数据集中有放回的抽取3个样本来产生一个Bootstrap数据集Z^{*1}，并得到α的估计$\hat{\alpha}^{*1}$。
- 整个过程重复B次，获得$\hat{\alpha}^{*1}, \ldots, \hat{\alpha}^{*B}$

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{*r} - \bar{\alpha})^2} = 0.087$$





KNN

Motivation

- 在局部回归的方法中，使用如下通用的回归模型

$$y = f(x) + \epsilon$$

- 不同于线性回归，非参数回归估计对 f 的形式没有做任何参数假设，而是使用局部数据对 f 进行局部建模

KNN

- K 邻近估计（k-nearest-neighbor，KNN）
- 使用离估计点最近的K 个样本的响应变量的均值当成该点的估计

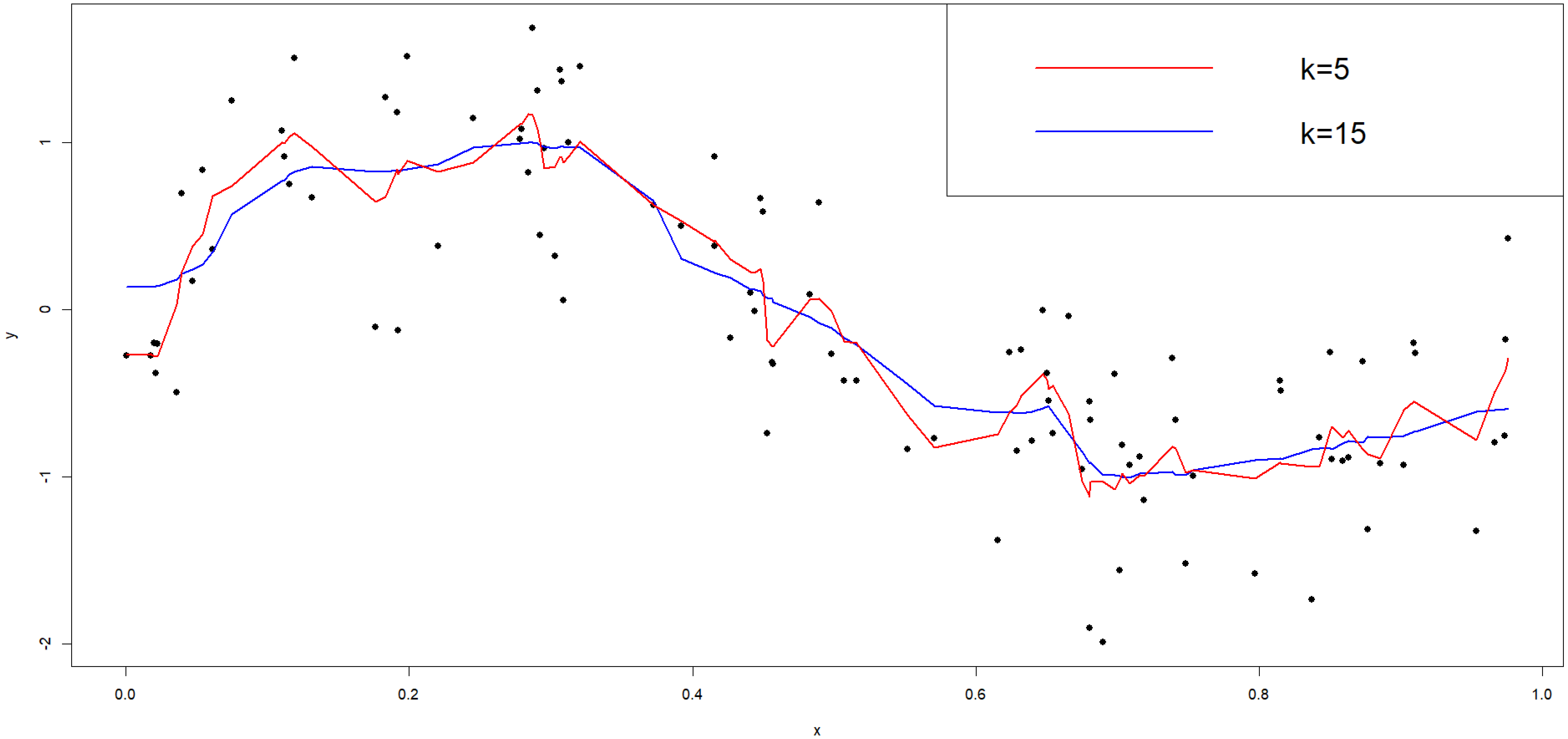
$$\hat{f}(x) = \frac{1}{K} \sum_{x_i \in N_{\mathrm{K}}(x)} y_i$$

其中$N_{\mathrm{K}}(x)$表示距离x最近的K个点的集合

- KNN 方法得到的条件期望函数不连续。在x 连续变化过程中，新观测点进入$N_K(x)$，距离最远的一个将离开，由此产生跳跃。

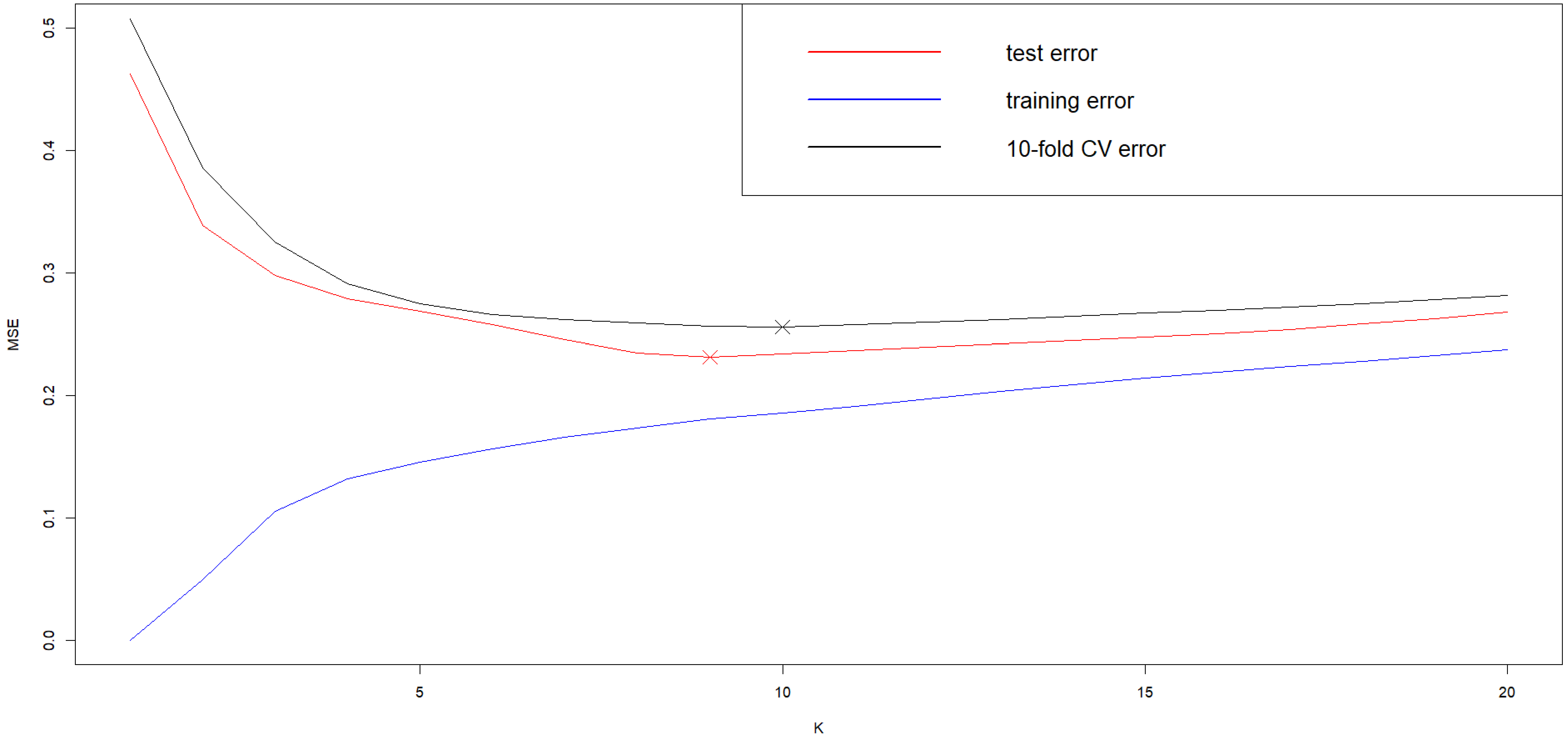
KNN

- KNN的表现取决于K的选择
- K小时可能过拟合，K大时可能欠拟合



KNN

- 10折CV提供了测试误差的良好近似



本章小节

- VSA
- LOOCV
- k折CV
- Bootstrap
- KNN