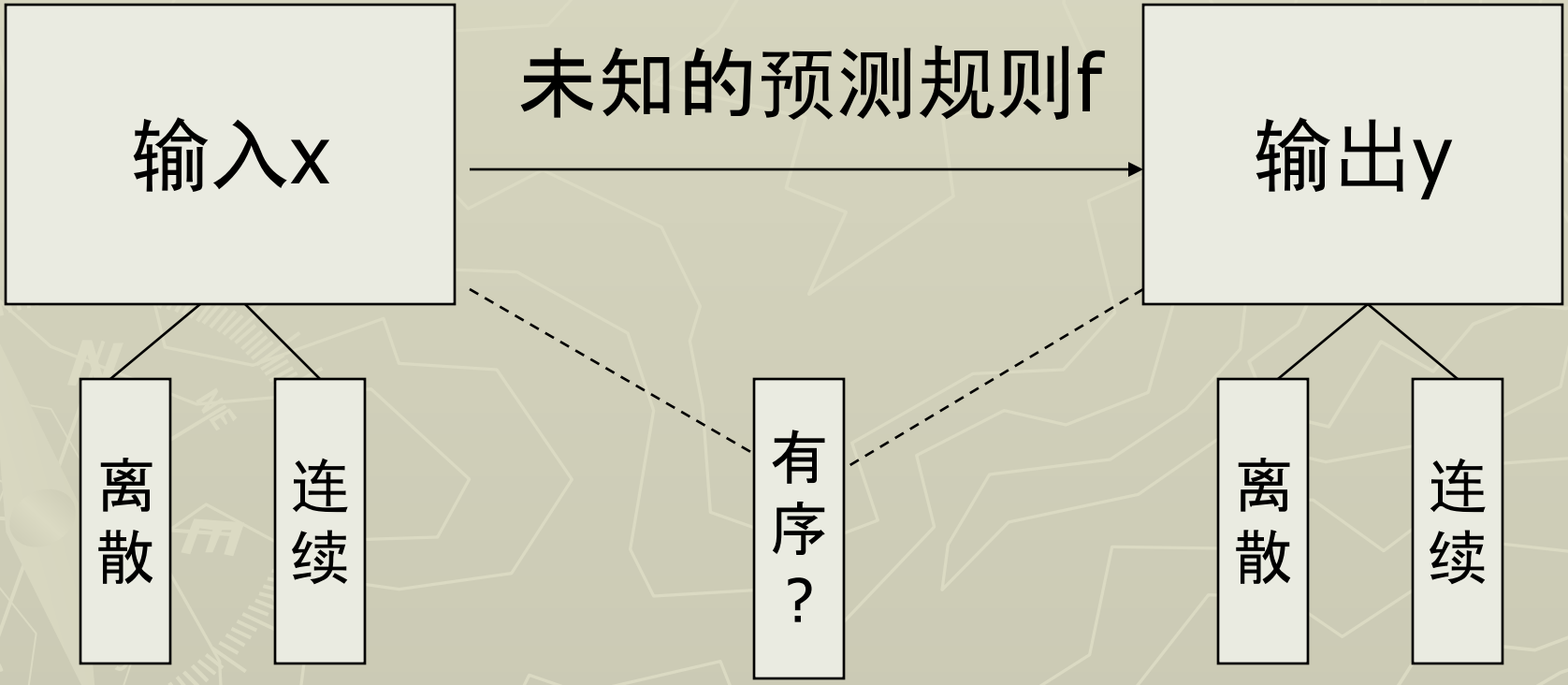
Overview of Supervised Learning

吴高巍

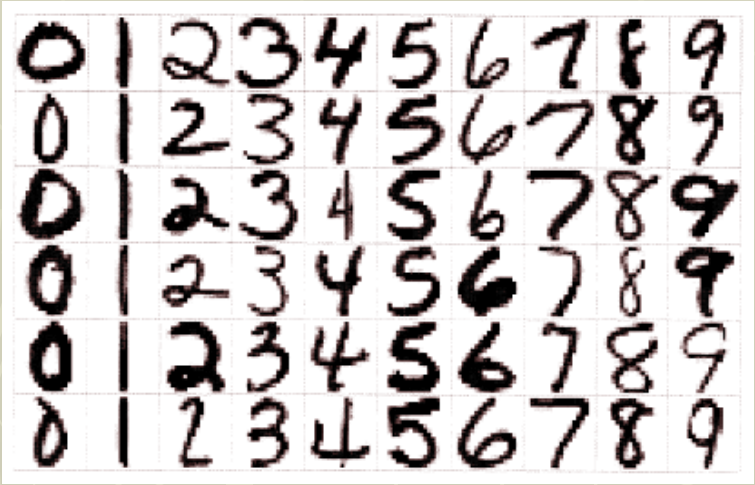
目录

函数估计



各种函数估计的例子

- 手写数字识别:给定数字图像的灰度信息,判断数字是属于0-9中的哪一类.



- 天气预报:给定大气云层的信息,判断明天的气温和降水概率.

各种函数估计的例子

- 通过蘑菇的各种属性（如伞和杆的形状,颜色等来判断蘑菇是否有毒）



记号说明

- 以下我们总是用N来代表训练数据的个数，用p来代表训练数据的维数。用$\hat{x}$ 表示对x的估计。
- 为了不和p混淆，用Pr(x)来表示概率密度函数。
- 用小写字母表示训练数据的值，不加粗体的大写字母表示p维向量，加粗体的大写字母表示N维向量或矩阵。

监督学习基础

- 在监督学习中，我们的目标是要通过训练来完成函数估计的任务.
- 通过观察，构造一个训练集$\mathbf{T}=(x_i,y_i)$, $i=1,\ldots,N$
- 训练集中的数据称为训练数据，也称为训练样本,单个数据称为样本点。
- 有一个学习算法，把训练集交给这个学习算法，它产生对预测规则的一个估计。
- 学习算法可以根据估计的规则和真实的规则之间的误差（通过样本点上输出的y_i来衡量）修改它的估计。

数学表示

- 我们把训练集**T**中的$(x_j,\ldots,x_p)$看作是p维欧氏空间中的点集。要求预测的规则看作是定义在p维欧氏空间中的一个函数$Y=f(X)$。
- 我们的目标是在整个p维的欧氏空间上,使用训练集**T**来逼近这个函数$f(X)$。
- 通过这样的转换，我们就可以利用欧氏空间的几何性质和概率推理的工具。

考虑的问题

- 我们考虑两种问题，分别为输出取值于有限离散集合和连续域的问题。
- 有限离散集合的问题总可以转换为分类$Y \in \{0,1,\ldots,k\}$，所以针对离散集合，我们只讨论这样的分类问题。
- 对于连续域的问题，我们讨论$y \in \mathbf{R}$的问题，称之为回归问题。

Return to TOC

线性模型

- 线性模型是统计学习中最基础的模型.
- 给定输入向量X=(X_1,X_2,...,X_p,1), 通过模型

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j = X^T \hat{\beta}$$

  预测Y.
- 整个线性模型共有p+1个参数, 所以大概不需要很多的数据, 就可以拟合一个线性模型。

最小二乘法

- 训练线性（回归）模型的最常用方式是最小二乘法。
- 最小二乘法即最小化以下的平方误差准则：

$$RSS(\hat{\beta}) = \sum_{i=1}^{N} (y_i - x_i^T \hat{\beta})^2$$

- 对于参数$\hat{\beta}$，平方误差准则是一个二次函数，所以它的最小值一定存在。这是选用平方误差准则的一个重要原因。

最小二乘法

- 简单求导就可得到最小二乘法的解，在整个训练集上，最小二乘法的解为：

$$\hat{\beta} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

- 把$\hat{\beta}$ 代回线性模型，就可以得到一个线性的预测或叫估计。

线性模型示例

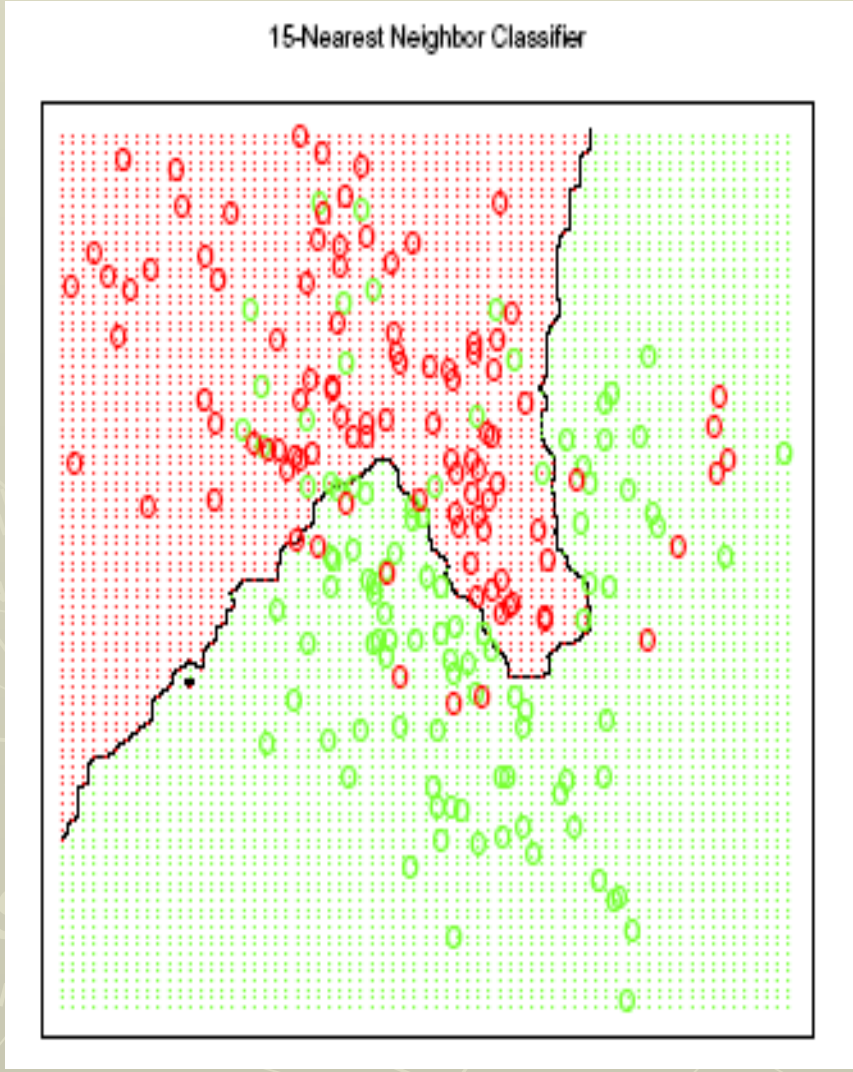
k近邻法

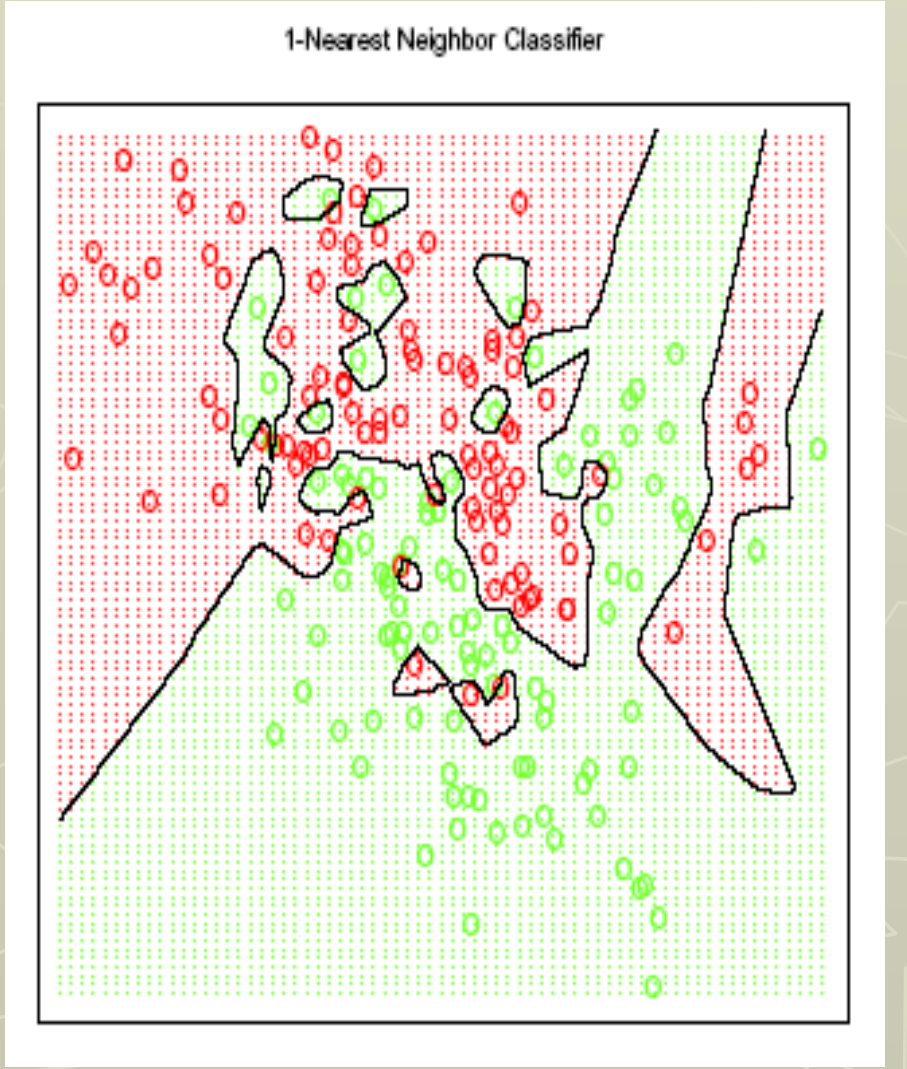
- k近邻法是另一种最基础的学习方法。
- k近邻回归如下定义：

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- k近邻回归就是对一个样本点x，取训练集中离它最近的k个近邻样本点，求它们的输出y的均值，以之来估计x对应的输出。

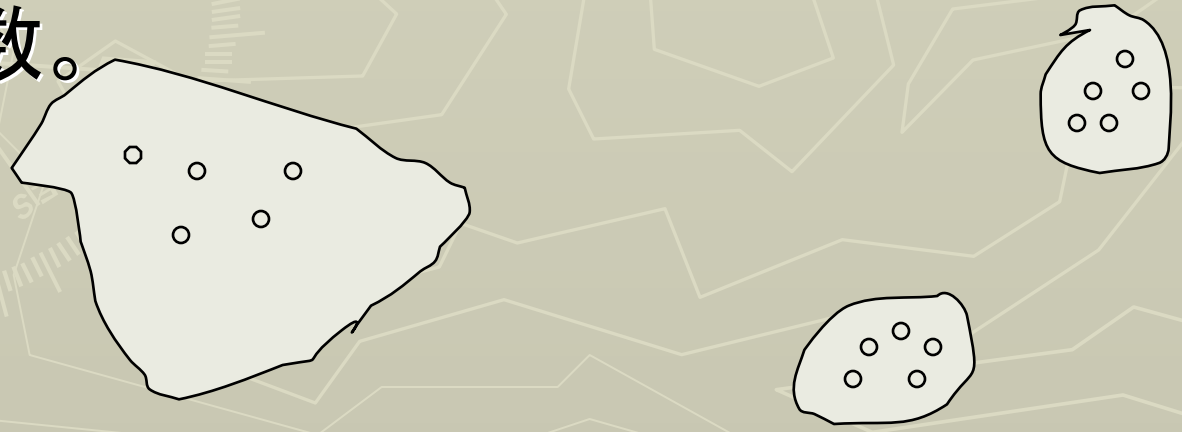
k近邻示例


15-Nearest Neighbor Classifier


1-Nearest Neighbor Classifier

k近邻的讨论

- 表面上看，k近邻只有一个参数k，但实际上，k近邻的参数是很多的。
- 直观上说，如果假设样本被划分成N/k个不重叠的部分，每个部分之间都有比较大的距离，那么每个部分就定义了一个k近邻模型的参数（其均值），则此时k近邻规则有N/k个参数。



k近邻和线性模型

- k近邻和线性模型代表了统计学习中的两个极端：最松泛的指定模型和最严格的限制模型。
- 直观的讲，假设数据服从的分布比较规则，比如每类数据服从一个正态分布，那么线性模型可能要比k近邻更好。
- 反过来，如果数据所服从的分布非常不规则，那么k近邻模型就会工作得更出色。
- 很大一部分统计学习算法都是对线性模型和k近邻的扩展。

Return to TOC

统计决策理论

- 下面我们用统计的方式来定义学习问题。
- 设$X \in \mathbf{R}^p$为实值随机输入向量，$Y \in \mathbf{R}$为随机输出值。X, Y有联合概率密度Pr(X,Y)，寻找函数f(X)，通过输入的X预测Y。
- 于是我们需要定义损失函数L(Y,f(X))。为了解决不同的问题，可以定义各种损失函数。在回归中最常用的损失函数是平方误差准则

$$L(Y,f(X)) = (Y-f(X))^2$$

条件期望

- 使用条件概率公式Pr(X,Y)=Pr(Y|X)Pr(X)来考察数学期望:

$$E(f(X,Y)) = \iint f(x,y)\Pr(x,y)dxdy$$

$$= \iint f(x,y)\frac{\Pr(x,y)}{\Pr(x)}\Pr(x)dydx$$

$$= \int \Pr(x)\int f(x,y)\Pr(y\mid x)dydx$$

$$= E_X[E_{Y|X}(f(x,y)\mid X=x)]$$

条件期望

统计决策理论：回归

- 我们在整个分布上检查平方误差准则

$$EPE(f) = E(Y - f(X))^2$$

$$= E_X E_{Y|X}([Y - f(X)]^2 \mid X = x)$$

- 可以在每一点上求最优的f(x)：

$$f(x) = \arg\min_c E_{Y|X}([Y - c]^2 \mid X = x)$$

- 这样得到的解就是条件期望

$$f(x) = E(Y \mid X = x)$$

统计决策理论：回归

- 得到的结果是条件期望,也叫做回归函数 .
- $f(x) = E(Y \mid X = x)$ ，这也就是我们的目标。
- 有了这个回归函数,我们再来看
- k近邻：$\hat{f}(x) = Ave(y_i \mid x_i \in N_k(x))$
- 在k近邻中，用均值逼近期望，用点的邻域逼近点。当k/N->0且N,k->∞时，可以证明

  $\hat{f}(x) -> E(Y \mid X = x)$
- 尽管k近邻有好的渐进性质，但是它的收敛速度 是随着维数的增高而下降的。

统计决策理论：回归

- 我们来看如何把线性模型放进统计决策理论的框架：
- 假设回归函数 $f(x) = E(Y \mid X = x) = x^T \beta$
- 代回平方误差准则 $EPE(f) = E(Y - f(X))^2$

$$= \int (y - f(x))^2 \Pr(dx, dy)$$

- 对 β 求导，可以得到 $\beta = [E(XX^T)]^{-1} E(XY)$
- 用样本均值替代数学期望，我们就回到

$$\hat{\beta} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

统计决策理论：回归

- k近邻和线性模型都可以放进我们的统计决策理论，它们的相同之处是都用样本均值来逼近数学期望。但是它们对模型的假设天差地别。
- 线性模型假设的是回归函数f(x)近似线性。
- k近邻假设的是f(x)可由一个局部恒定的函数逼近(考虑定积分中的微元法）。
- 看起来k近邻似乎可以逼近更多的函数类。

统计决策理论：分类

- 更换一个损失函数就可以讨论分类问题，最常用的是0-1损失函数，即错分的样本损失为1，正确分类的样本损失为0。

$$EPE(f) = E_X[\sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)] \Pr(\mathcal{G}_k \mid X)]$$

- 同样的，我们在每一点上取最小值

$$\hat{G}(X) = \arg\min_{g \in \mathcal{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k \mid X = x)$$

$$= \arg\min_{g \in \mathcal{G}} [1 - \Pr(g \mid X = x)]$$

$$= \arg\max_{g \in \mathcal{G}} \Pr(g \mid X = x)$$

统计决策理论：分类

- 这样得到的

$$\hat{G}(X) = \arg\max_{g \in \mathsf{G}} \Pr(g \mid X = x)$$

称为Bayes分类器，在0-1损失函数的意义下，Bayes分类器是最优的。Bayes分类器的错误率称为Bayes错误率。

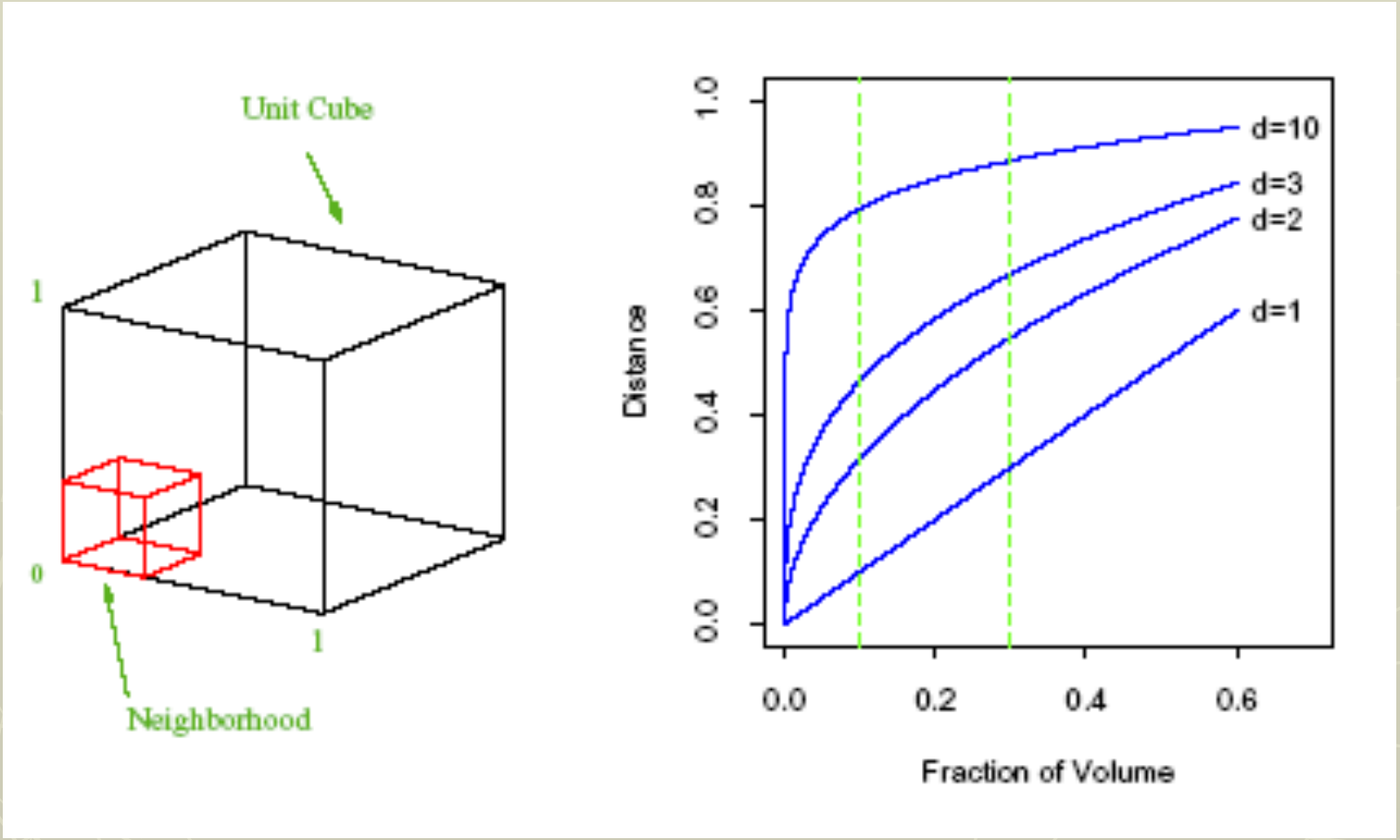
- 要注意的是机器学习中有很多算法并不考虑Bayes错误率。

Return to TOC

高维空间

- 直观上，由于k近邻的逼近能力，给定很多的数据，它应该可以很好的逼近输出Y。
- 但实际上，在高维空间内，事情并不像我们想象的那样。
- Bellman在1961年的书中，就详细的描述了这种维数灾难（Curse of Dimensionality）。

维数灾难



在p维超立方体中，想要取到体积为r的子立方体，需要取边长为$r^{1/p}$

维数灾难

- 这就意味着在高维空间内,数据几乎永远是稀疏的.
- 如果一个1维的“稠密”样本需要N=10个样本点的话,一个10维的“稠密”样本就需要$N=10^{10}$个样本点.
- 因此想要取得“局部邻域”几乎是不可能的.
- 这也就是不能用微积分的方法来处理学习问题的原因.

Return to TOC

回归估计问题的统计模型

- 假设一个数据是由具有加性噪声的统计模型生成。

  $Y=f(X)+\varepsilon$，ε是均值为0，方差为σ^2 的与X独立的随机变量（条件可放松）。
- 加性噪声模型在现实中是很常见的。假定$Y=f(X)$通常过于严格，经常情况下，会有除了X以外的未知因素影响着Y的值，通过这种加性模型，我们可以捕捉这些未知的影响。
- 令方差$\sigma^2=0$，我们就得到无噪声的模型。

分类问题的统计模型

- 对于分类问题，通常不使用上页中定义的加性噪声模型。
- 分类问题的目标函数p(X)就是条件概率密度Pr(G|X)，这就是分类问题的统计模型。
- 对于两类问题，通常假设出现某一结果的概率为p(X)，而出现另一结果的概率为1-p(X)。

偏差-方差分解

- 我们考虑回归的统计模型Y=f(X)+ε
- 代入我们的期望预测错误函数

$$EPE(x_0) = E[(Y - \hat{f}(x_0))^2]$$

$$= E[(\varepsilon^2 + 2\varepsilon(f(x_0) - \hat{f}(x_0)) + (f(x_0) - \hat{f}(x_0))^2)]$$

$$= \sigma^2 + E[(f(x_0) - \hat{f}(x_0))^2]$$

这里用了不相关的条件

$$= \sigma^2 + E[\hat{f}(x_0) - E(\hat{f}(x_0))]^2 + [E(\hat{f}(x_0)) - f(x_0)]^2$$

$$= \sigma^2 + Var(\hat{f}(x_0)) + Bias^2(\hat{f}(x_0))$$

偏差-方差分解

$$EPE(x_0) = \sigma^2 + Var(\hat{f}(x_0)) + Bias^2(\hat{f}(x_0))$$

称为偏差-方差分解。

- 第一项σ^2只和我们面对的问题有关，即使我们知道真正的$f(x_0)$，也无法控制σ^2。
- 后两项是可以控制的，它们构成了$\hat{f}(x_0)$的均方误差。均方误差分为偏差和方差两部分。
- 偏差是真正的均值和预测值的差。
- 方差是这个预测值作为随机变量的方差（在所有可能的训练集上平均）。

偏差-方差分解:k近邻

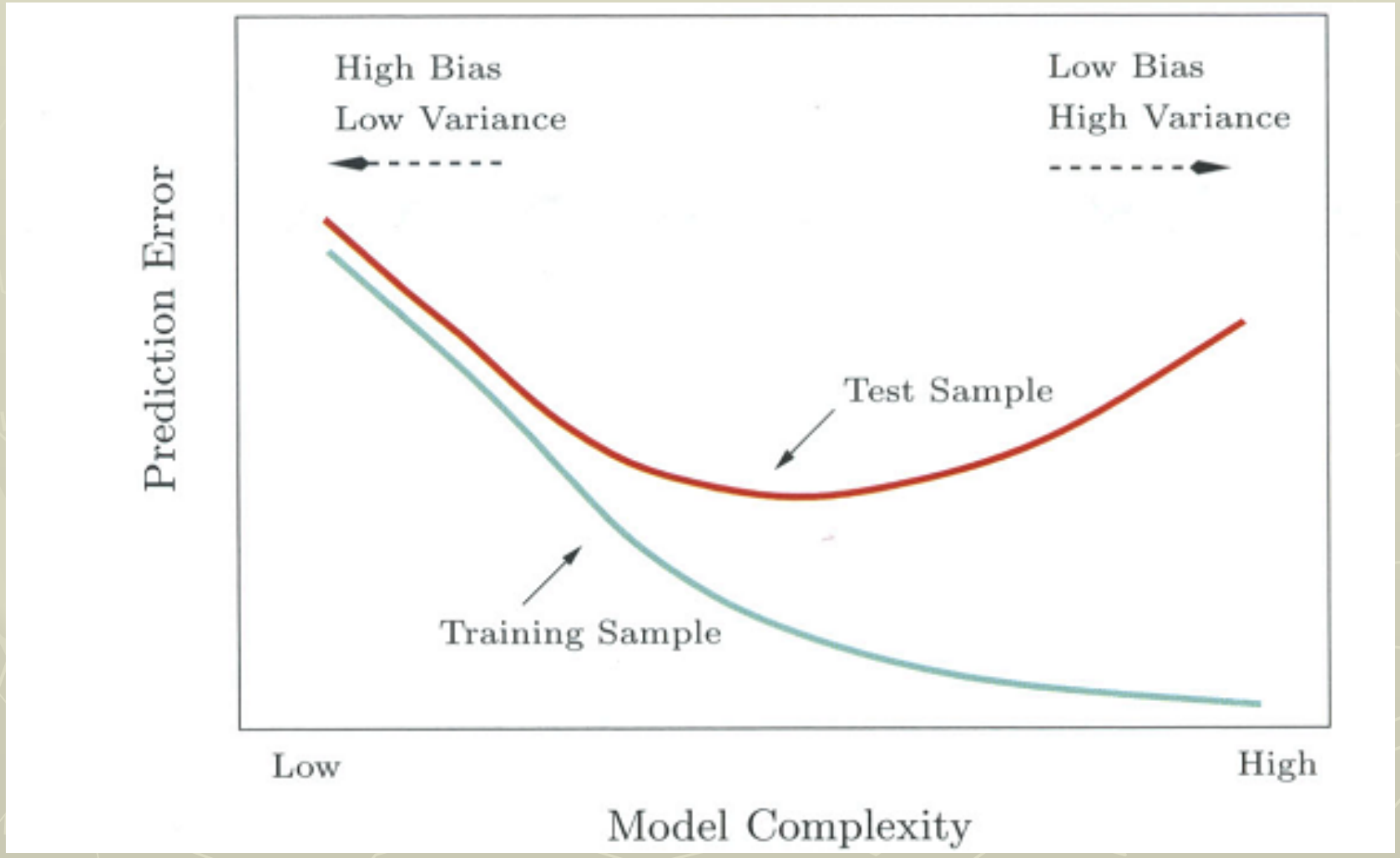
- 对于k近邻，我们可以求出偏差和方差

$$Bias^2(\hat{f}(x_0)) = [E(\hat{f}(x_0)) - f(x_0)]^2 = [f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})]^2$$

$$Var(\hat{f}(x_0)) = E[\hat{f}(x_0) - E(\hat{f}(x_0))]^2 = \frac{\sigma^2}{k}$$

- 可以看到，k近邻的方差随着k的上升而下降。这表示了k近邻估计的“稳定性”随着k的上升而提高。
- 而k越高，取的邻域就越大，用这个大邻域中的均值去估计$f(x_0)$，偏差就会增大。

The Bias-Variance Tradeoff



偏差-方差折中:更多的讨论

- 有些方法似乎可以同时减少偏差和方差(至少在有些时候),比如boosting.

- 偏差和方差比起SVM中使用的经验风险+置信范围来,最大的好处就是它通常可以计算(给定一个模型),而SVM理论中的VC维一般是不能计算甚至很难估计的,margin界则几乎不能用来讨论回归问题.

偏差-方差折中与维数灾难

- k近邻的偏差项

$$Bias^2(\hat{f}(x_0)) = [E(\hat{f}(x_0)) - f(x_0)]^2 = [f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})]^2$$

显然是受到维数灾难影响巨大的。

- 而对于线性模型来说，如果目标函数本身是线性的话，最小二乘法是无偏的。其在样本集上的平均方差在N很大的情况下有

$$E_{x_0} Var(x_0) \approx \sigma^2(p/N) + \sigma^2$$

- 所以，最小二乘法在这种情况下，不太受维数灾难的影响（但是这里有很强的假设）。

Return to TOC

约束模型

- 从有限的样本估计无限的函数，解是无穷多的。
- 要获得一个固定的解，必须要加约束。
- 通常，学习方法中的约束大都是关于模型复杂性的。假设在输入空间的一个邻域内，有一些特殊的性质（不变，线性或者满足某个低阶多项式），这样就可以求得估计。
- 邻域越大，约束越强（微元法和全局线性模型）。
- 考虑到维数灾难，不能要求邻域很小。

模型，模型

- 在经验风险后面加一个惩罚项（比如关于光滑性）。

  Smoothing Splines: $\int [f''(x)]^2 dx$

- 核方法（不同于SVM的Kernel trick！）：定义一个核函数（窗函数）K(x_0,x)，决定x_0周围的一个区域内的数据对x_0的影响，最后使用如加权平均一类的方法求得 $\hat{f}(x_0)$

  高斯核：

$$K_\lambda(x_0, x) = \frac{1}{\lambda}\exp[-\frac{\|x - x_0\|^2}{2\lambda}]$$

模型，模型。。。

- 基函数：选择一组基函数，使得 $\hat{f}$ 的模型是基函数的线性组合：

$$\hat{f}_\theta(x) = \sum_{m=1}^{M} \theta_m h_m(x)$$

- 目前机器学习中的很多方法都可以归结为基函数法，如SVM，神经网络，Boosting，小波，样条等。
- 一个非常有趣的问题是：如何根据问题，选择基函数？

Thanks!