模型评估与选择

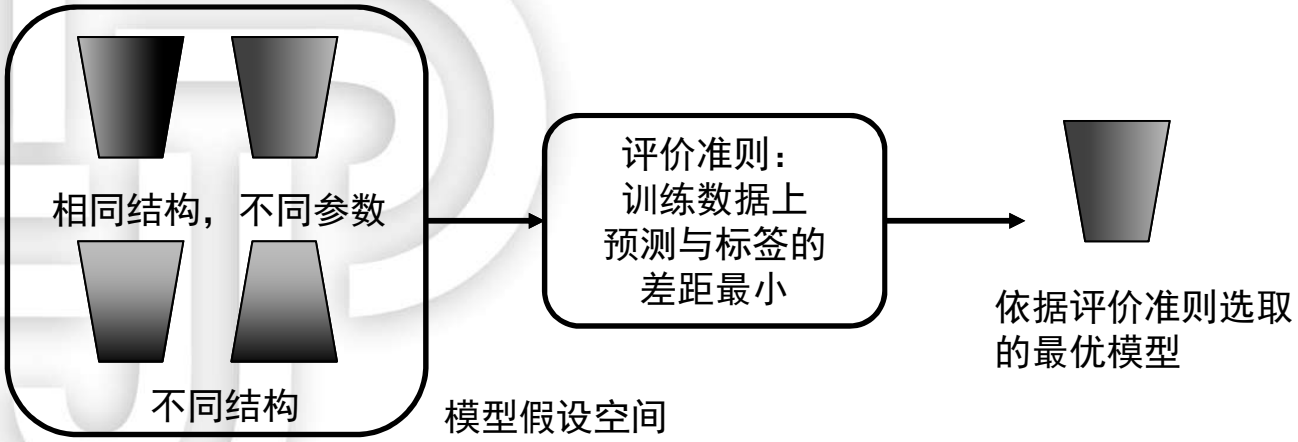
计算机学院并行与分布处理国家重点实验室

主要内容

- 讲授内容
  - 机器学习可行性
  - 模型评估定义与必要性
  - 选择途径
    - 对模型进行选择，需要对候选模型的泛化误差进行评估，选择泛化误差最小的那个模型
  - 实验方法与性能评价
- 要求
  - 理解模型评估的基本原理

机器学习是一个模型选择过程

- 机器学习：【李航：统计机器学习】
  - 机器学习：从训练数据，应用评价准则，从模型假设空间中选取一个最优的模型
  - 模型假设空间（对象）：形式，结构，超参数，参数
  - 评价准则（目标）：选择模型的标准



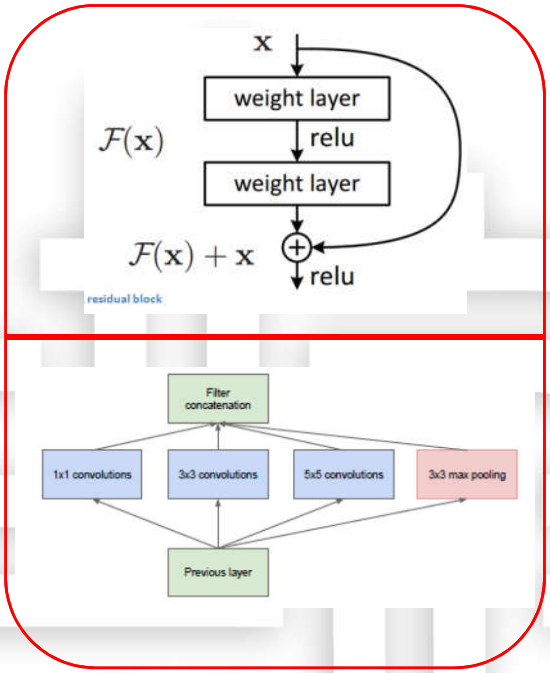
模型假设空间 Hypothesis space

- Hypothesis space: “defines the class of functions mapping the input space to the output space. ”
- 李铁岩 《Learning to Rank for Information Retrieval》P16

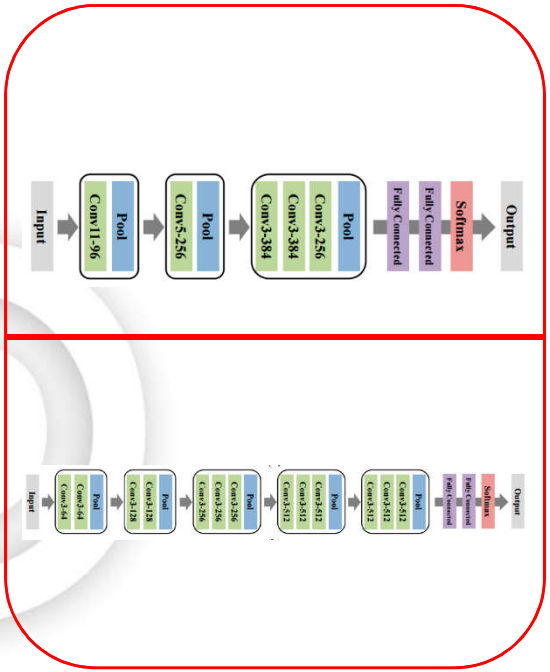
- 假设空间：监督学习的目的在于学习一个由输入到输出的映射，这一映射由模型来表示。换句话说，学习的目的就在于找到最好的这样的模型。模型属于由输入空间到输出空间的映射集合，这个集合就是假设空间（hypothesis space）。假设空间的确定意味着学习范围的确定。
- 李航 《统计学习方法》P5

模型假设空间

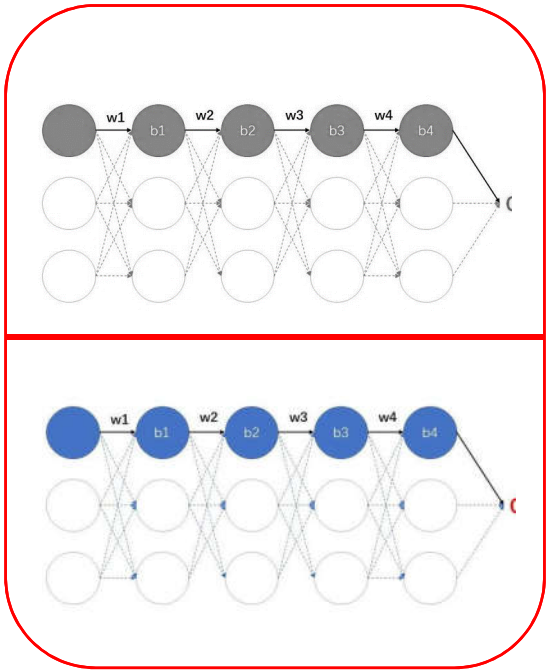
- 模型假设空间：模型，及其参数
- 例：一组不同的映射：映射+参数
- 确定的映射关系：参数　　$h(\theta;x)=\theta_0+\theta_1 x$



不同的网络类型

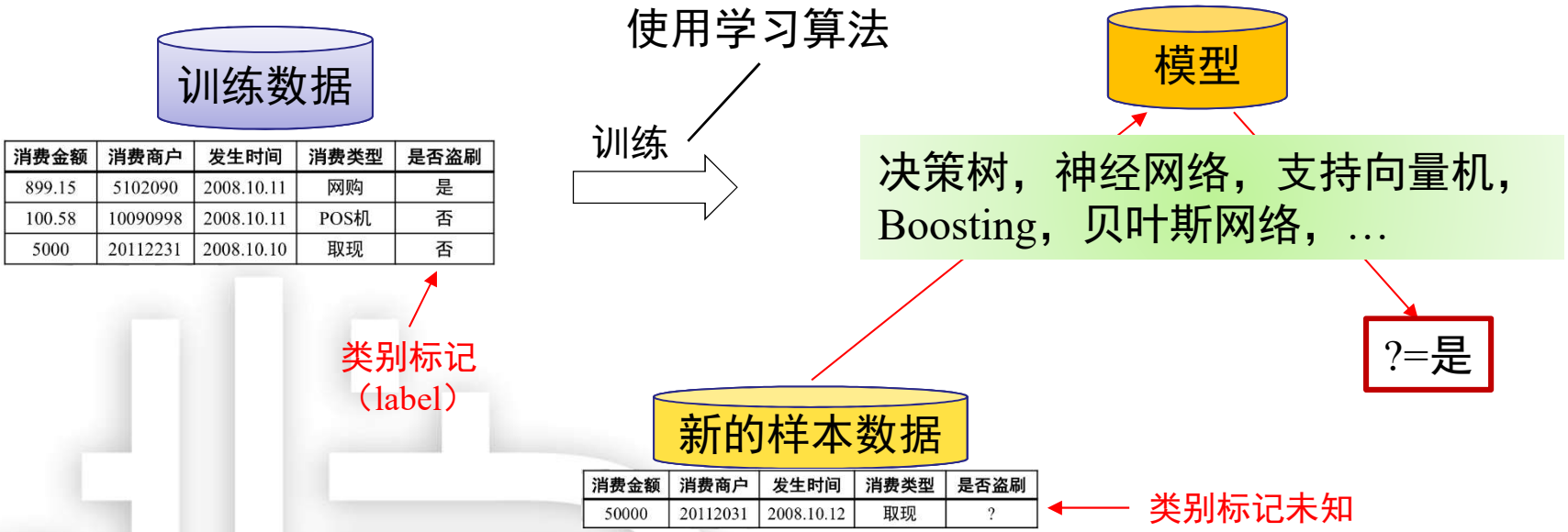


相同网络结构，不同的层数、节点数等超参数



相同网络结构及超参数，不同的网络权值等参数

典型的机器学习过程



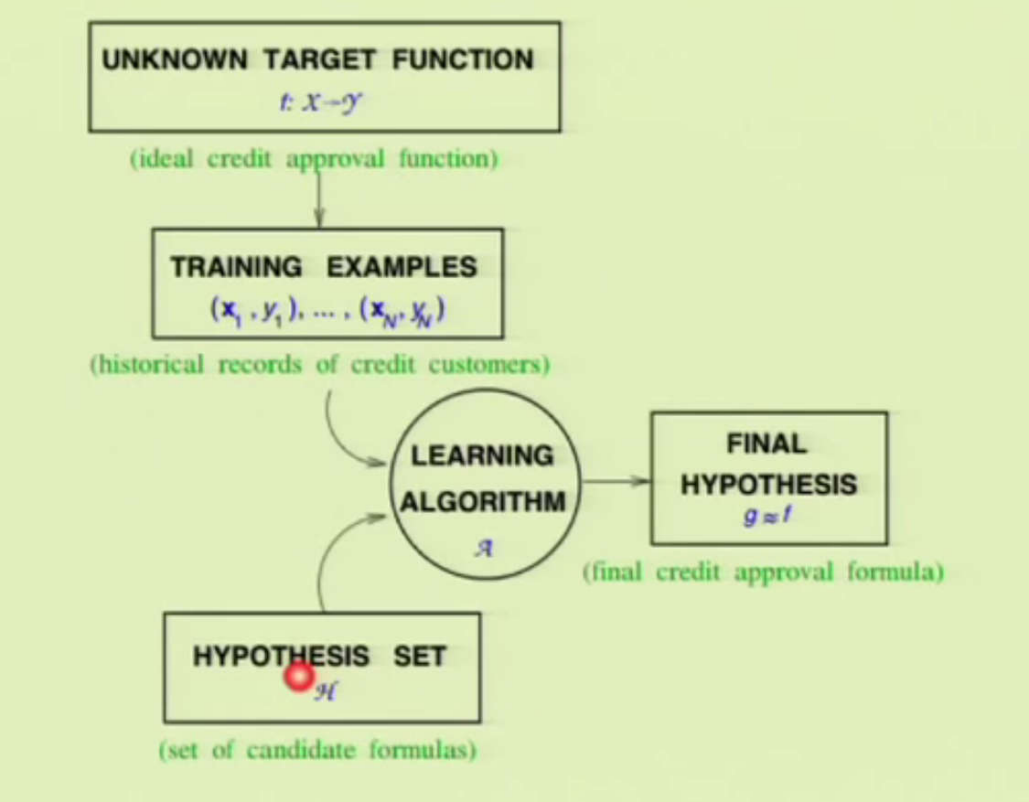
泛化能力强！

能很好的适用于未知样本 (unseen instance)
例如：错误率低，精度高

然而，我们手上没有unseen instance, …

机器学习是一个模型选择过程

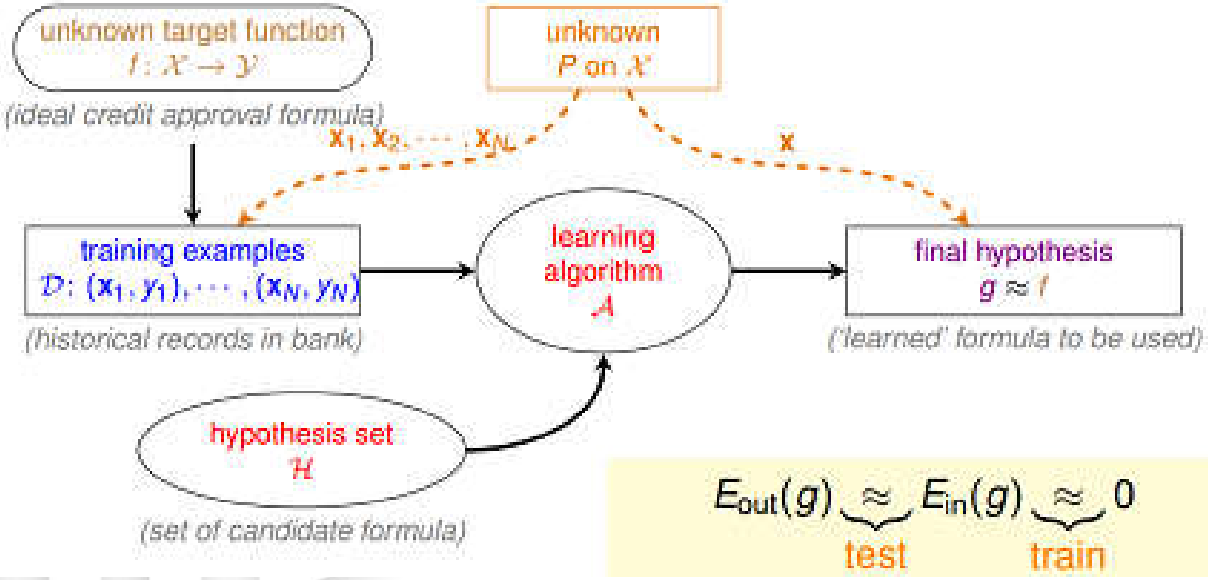
- 机器学习：找映射关系；从假设空间找最合适的映射



Courtesy of Prof. Yasser Abu Moustafa Caltech

机器学习是一个模型选择过程

- f：理想的方案(可以是一个函数，也可以是一个分布)
- H：我们机器学习方法的假设空间
- g：学习到的用来预测的假设，g属于H。



- 机器学习：通过算法A，在假设空间H中，根据样本集D，选择最优假设g。选择标准： g 近似于 f。

泛化误差 vs. 经验误差

- 如何判断g 近似于 f?
- 泛化误差 **Eout(h)**
  - 在“未来”样本上的误差
- 经验误差 **Ein(h)**
  - 在训练集上的误差，亦称“训练误差”
- 泛化误差越小越好，经验误差是否越小越好？
- 经验误差小，泛化误差会小？
- 经验误差和泛化误差有多大差距？

误差包含了哪些因素？

泛化误差分解

- 泛化误差分解：

$$E(f;D) = bias^2(x) + var(x) + \varepsilon^2$$

期望输出与真实输出的差别

$$bias^2(x) = (\bar{f}(x) - y)^2$$

同样大小数据集的变动，所导致的性能变化

$$var(x) = \mathrm{E}_D\left[(f(x;D) - \bar{f}(x))^2\right]$$

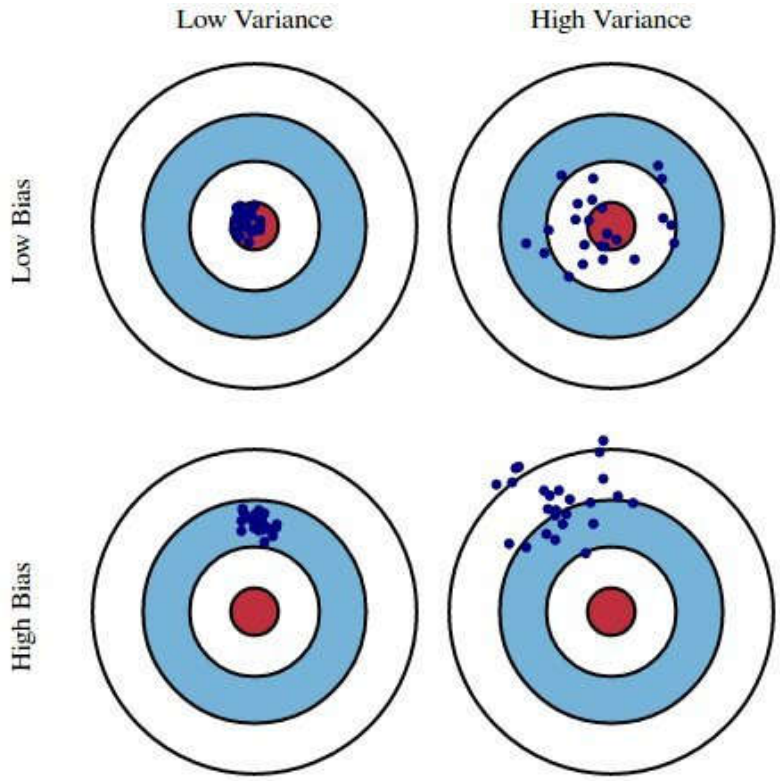
当前任务上任何学习算法所能达到的期望泛化误差下界

$$\varepsilon^2 = \mathrm{E}_D[(y_D - y)^2]$$

泛化性能由学习算法的能力、数据的充分性以及学习任务本身的难度共同决定

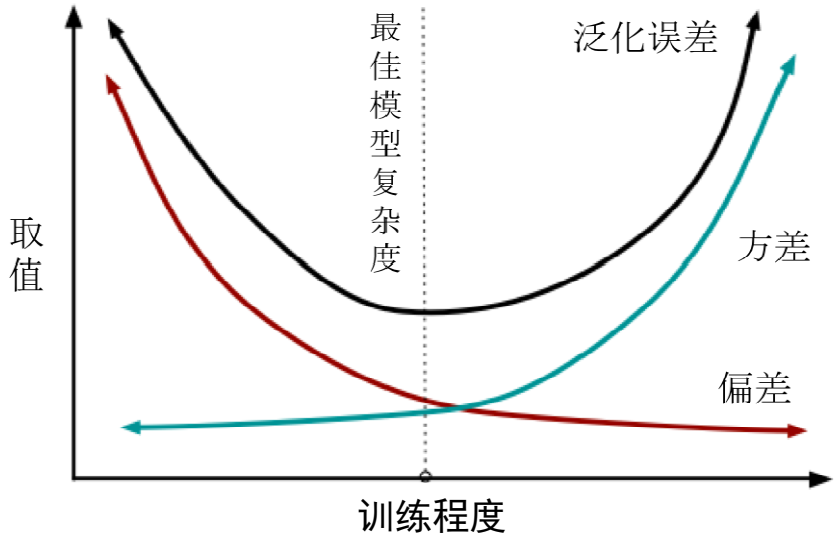
泛化误差分解

- Prediction Error=Bias+Variance+Noise



泛化误差、偏差、方差

- 一般而言，偏差与方差存在冲突
  - 训练不足时，学习器拟合能 力不强，偏差主导
  - 随着训练程度加深，学习器拟合能力逐渐增强，方差逐 渐主导
  - 训练充足后，学习器的拟合能力很强，方差主导



模型的过拟合与欠拟合

- 过拟合，overfitting
  - 学习器将训练样本自身的一些特点当作了所有潜在样本都具有的一般性质，导致泛化性能下降
- 欠拟合，under-fitting
  - 与过拟合相对应，是指对训练样本的一般性质尚未学好

过拟合 vs. 欠拟合

树叶训练样本

新样本

过拟合模型分类结果：

不是树叶

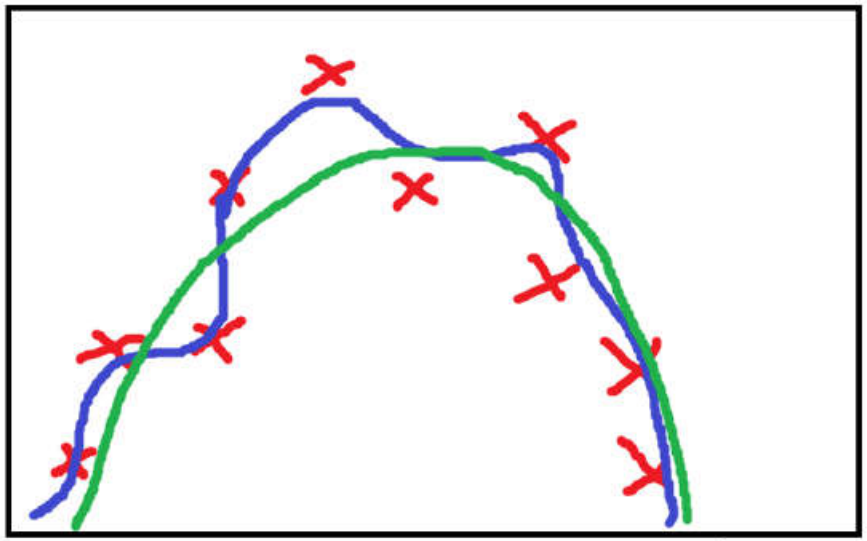
（误认为树叶必须有锯齿）

欠拟合模型分类结果：

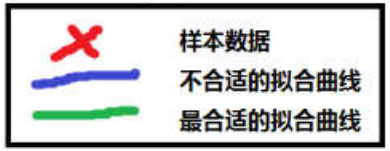
是树叶

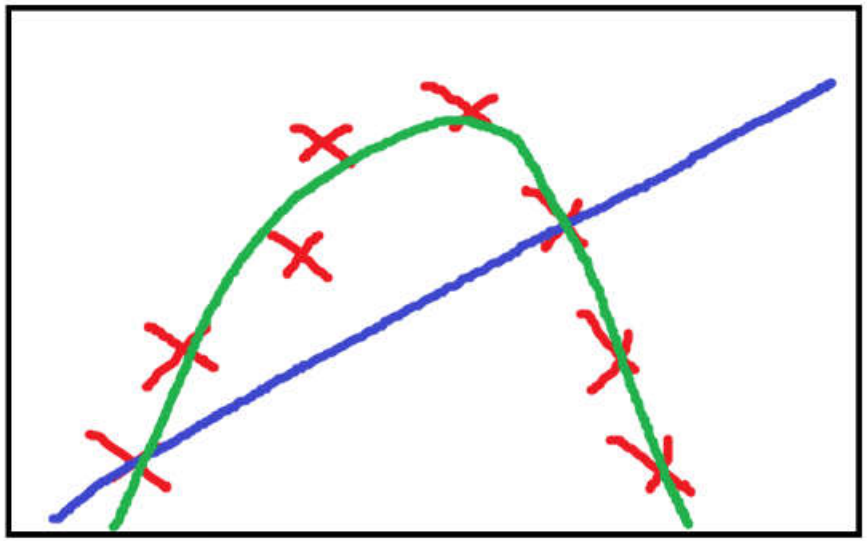
（误认为绿色的都是树叶）

过拟合 vs. 欠拟合



过拟合





欠拟合

过拟合、欠拟合的本质？

现有假设与理想假设间的差距

- **Eout(h)**，可以理解为在理想情况下(已知f)，总体(out-of-sample)的损失(这里是0–1 loss)的期望，称作expected loss。
- **Ein(h)**，可以理解为在训练样本上(in-of-sample)，损失的期望，称作expirical loss。
- 例：

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

for any fixed h, can probably infer

$$\text{unknown } E_{\text{out}}(h) = \mathop{\mathcal{E}}_{\mathbf{x}\sim P} [\![h(\mathbf{x}) \neq f(\mathbf{x})]\!]$$

$$\text{by known } E_{\text{in}}(h) = \frac{1}{N}\sum_{n=1}^{N} [\![h(\mathbf{x}_n) \neq y_n]\!].$$

机器学习问题

- 失败的学习：
  - 1.训练样本集（in-sample）中最佳假设g依然有很大误差
  - 2. 训练样本集（in-sample）中最佳假设g误差不大，但在训练样本集之外的其他样本（out-of-sample）中，g和目标函数f可能差别很远
- 例：通过少量的已知样本推论整个样本集

- 1. 机器学习的可能性（给定假设空间是否可学到一定误差下的模型）？
- 2. 怎样评价学习模型的优劣？（评价准则？）
- 3. 不同模型的选择？

Hoeffding 不等式 与 误差上界

- Hoeffding（霍夫丁）不等式
  - X1,X2,⋯,Xn：一组独立同分布的参数为p的伯努利分布随机变量（x:{0,1}）
  - 均值：

$$\bar{X} = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

  - 对于任意δ>0, Hoeffding不等式可以表示为：

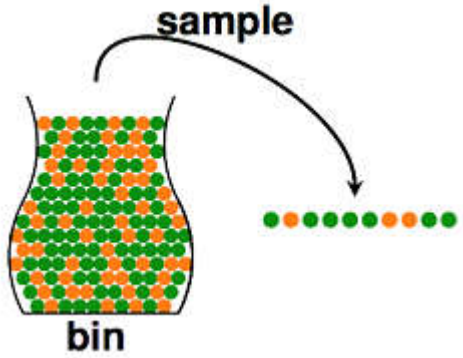
$$P\left(|\tilde{X} - E\left(\tilde{X}\right)| \geq \delta\right) \leq 2exp(-2N\delta^2)$$

  - N越大，UpperBound越接近于0

Hoeffding 不等式 与 误差上界

- 机器学习举例：
- 算法：罐中抽球，用样本统计量（样本均值）推断总体参数（总体期望）
- 假设空间：？
- 应用霍夫丁不等式：

$$P(|v-u| \geqslant \delta) \leqslant 2exp(-2N\delta^2)$$



现有假设与理想假设间的差距

- 1. 对某个特定假设h，经验误差和泛化误差的差距：

for any fixed h, in 'big' data (N large),

in-sample error $E_{in}(h)$ is probably close to

out-of-sample error $E_{out}(h)$ (within ϵ)

$$\mathbb{P}\left[\left|E_{in}(h) - E_{out}(h)\right| > \epsilon\right] \leq 2\exp\left(-2\epsilon^2 N\right)$$

- N足够大时，expected loss和expirical loss将非常接近
- 2. 对整个假设空间（M个假设函数）

$$P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon \cup |E_{in}(h_2) - E_{out}(h_2)| > \epsilon \ldots |E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

$$\leq P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon) + P(|E_{in}(h_2) - E_{out}(h_2)| > \epsilon) + \ldots + P(|E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

$$\leq 2M\exp(-2\epsilon^2 N)$$

- 上界取决于假设个数M与样本个数N

机器学习中的两个问题

- 1. 能否保证泛化误差**Eout(h)足够接近经验误差Ein(h)?**
- **2. 能否使Ein(h)尽量小**
- 如果假设空间H的size M有限，N足够大，对假设空间中任意g，Eout(g)约等于Ein(g)；
- 算法A从假设空间H中挑选出g，使得Ein(g)接近于0，则Eout(g)很大可能接近0；

假设空间大小M的影响

$$P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon \cup |E_{in}(h_2) - E_{out}(h_2)| > \epsilon \ldots |E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

$$\leq P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon) + P(|E_{in}(h_2) - E_{out}(h_2)| > \epsilon) + \ldots + P(|E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

$$\leq 2M \exp(-2\epsilon^2 N)$$

Trade-off on M

1. can we make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$?
2. can we make $E_{in}(g)$ small enough?

small M

1. Yes!, $\mathbb{P}[\text{BAD}] \leq 2 \cdot M \cdot \exp(\ldots)$
2. No!, too few choices

large M

1. No!, $\mathbb{P}[\text{BAD}] \leq 2 \cdot M \cdot \exp(\ldots)$
2. Yes!, many choices

- 很多情况下，M无穷大，此判别无意义

有效假设

- 可否建立包含有限个数的判别

establish **a finite quantity** that replaces M

$$\mathbb{P}\left[\left|E_{\text{in}}(g) - E_{\text{out}}(g)\right| > \epsilon\right] \overset{?}{\leq} 2 \cdot m_{\mathcal{H}} \cdot \exp\left(-2\epsilon^2 N\right)$$

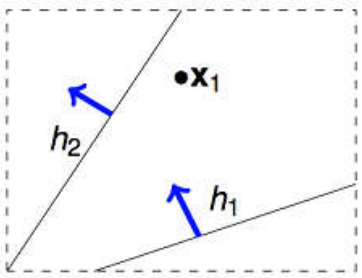
- 原式很宽松，很多条件其实可以合并

$$P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon \cup |E_{in}(h_2) - E_{out}(h_2)| > \epsilon \ldots |E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

$$\leq P(|E_{in}(h_1) - E_{out}(h_1)| > \epsilon) + P(|E_{in}(h_2) - E_{out}(h_2)| > \epsilon) + \ldots + P(|E_{in}(h_m) - E_{out}(h_m)| > \epsilon)$$

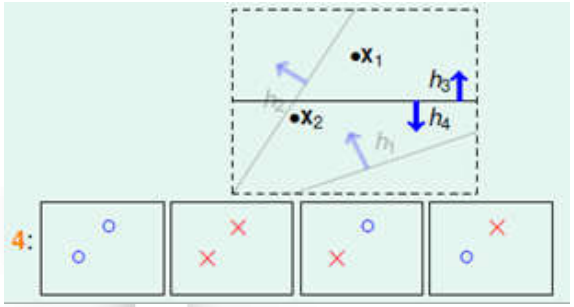
$$\leq 2M \exp(-2\epsilon^2 N)$$

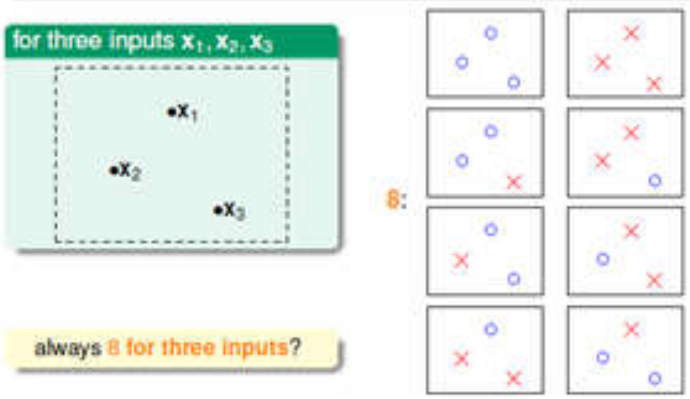
- 例：直线g划分点x

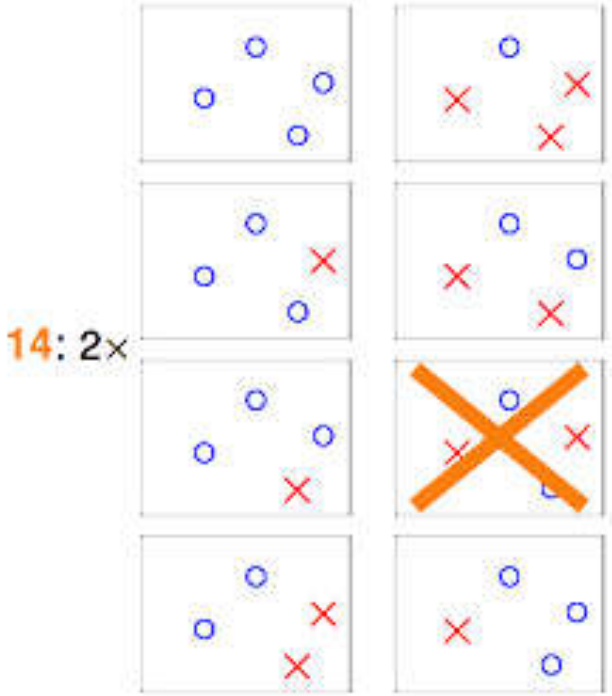


有效假设

- 例：直线(假设空间为直线)g划分2个点，H中最多有4类直线；3个点，8类；4个点，14类直线







- 假设空间size M虽然大，但在样本集D上，有效的假设函数数目是有限的

基于Dichotomy对分的有效假设

- H中任选一个方程h，让这个h对样本集合D进行二元分类
- 每一个二分的结果称为一个Dichotomy

- H可以无限，但Dichotomy个数有限（例4个点，只有14个Dichotomies ）

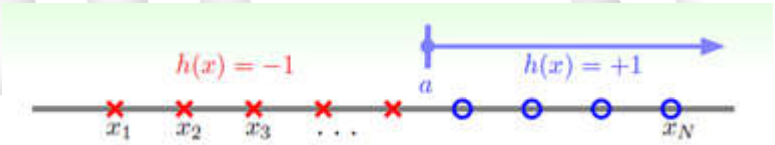
- 定义 effective(N)： H对于N个样本，“最多”能产生多少不同的dichotomy

基于Dichotomy对分的有效假设

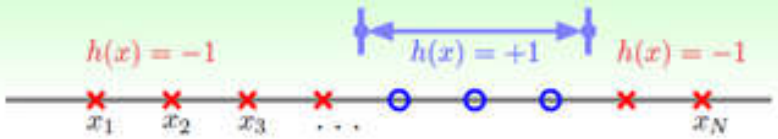
- effective(N)： H对于N个样本，“最多”能产生多少不同的dichotomy
  - 不同的H，面对同样的数据D的时候，产生的dichotomy会不同
  - 同样的H，面对不同的数据D的时候，产生的dichotomy会不同

The Four Growth Functions

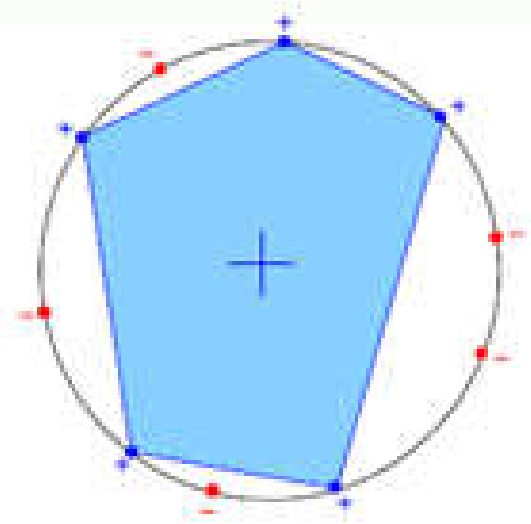
- positive rays: $m_{\mathcal{H}}(N) = N + 1$
- positive intervals: $m_{\mathcal{H}}(N) = \frac{1}{2}N^2 + \frac{1}{2}N + 1$
- convex sets: $m_{\mathcal{H}}(N) = 2^N$
- 2D perceptrons: $m_{\mathcal{H}}(N) < 2^N$ in some cases



Positive rays：最大情况，数据都在直线上



Positive intervals：最大情况，数据都在直线上



Convex sets：
最大情况，数据在圆上

基于Dichotomy 对分的有效假设

- Growth function 增长函数：
- H作用于D“最多”能产生多少种不同的dichotomies？
  - 与假设空间H有关，跟数据量N也有关
  - effective(N) 表示为增长函数：max_H(x1,x2,…,xN)
- H确定的情况下，growth function是一个与N相关的函数

$|\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$: depend on inputs $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$

growth function:
remove dependence by taking max of all possible $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$$

finite, upper-bounded by 2^N

- 基于增长函数的上界：

$$\mathbb{P}\left[\left|E_{\text{in}}(g) - E_{\text{out}}(g)\right| > \epsilon\right] \overset{?}{\leq} 2 \cdot m_{\mathcal{H}}(N) \cdot \exp\left(-2\epsilon^2 N\right)$$

Break Point与Shatter打散

- Shatter打散：假设空间H作用于N个样本，产生的dichotomies数量等于这N个点总的组合数2^N，称：这N个样本被H给shatter
- “N个点的所有(碎片般的)可能情形都被H产生”
- $m_H(N)= 2^N$的情形是即为“shatter”
- Break points: H不能打散的最小样本数k （显然更大的更不能被打散）

if no k inputs can be shattered by $\mathcal{H}$,
call k a **break point** for $\mathcal{H}$

- $m_{\mathcal{H}}(k) < 2^k$
- $k+1, k+2, k+3, \ldots$ also break points!
- will study minimum break point k

- 例：positive ray， $m_H(N)=N+1$
- N=2时， $m_H(2)<2^2$，所以它的break point就是2

增长函数的上界

- 增长函数$m_H(N)$ 上界是否还可以往更小的范围约束？
- 1. $m_H(N)$ 最多到2^N。为什么？
- 2. 2^N，对于N依然是指数级的，依然难以用于Hoeffding‘s Inequality，很多mH(N) 实际是比2^N小的（positive rays 等）
- 借助break point，可将$m_H(N)$ 进一步缩小
- 如果k是H的一个break point （H可以被shatter打散的最大样本量），可以证明：

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

$$\mathbb{P}\left[\left|E_{\text{in}}(g) - E_{\text{out}}(g)\right| > \epsilon\right] \overset{?}{\leq} 2 \cdot m_{\mathcal{H}}(N) \cdot \exp\left(-2\epsilon^2 N\right)$$

- 这是一个多项式polynomial，最高次项为k-1
- 结论：如果break point存在，则$m_H(N)$ 是一个多项式，Hoeffding's Inequality可以很好地收敛只要N足够大

基于增长函数的误差上界

- H作用于数据量为N的样本集D，可能的映射是无穷的，真正有效(effective)的映射是有限的，这个数量为$m_H(N)$。
- 回顾之前误差上界定义：

$$P(|E_{in}(h_1)-E_{out}(h_1)|>\epsilon \cup |E_{in}(h_2)-E_{out}(h_2)|>\epsilon \ldots |E_{in}(h_m)-E_{out}(h_m)|>\epsilon)$$

$$\leq P(|E_{in}(h_1)-E_{out}(h_1)|>\epsilon)+P(|E_{in}(h_2)-E_{out}(h_2)|>\epsilon)+\ldots+P(|E_{in}(h_m)-E_{out}(h_m)|>\epsilon)$$

$$\leq 2M\exp(-2\epsilon^2 N)$$

- 怎么替换？H中每一个h作用于D对应一个Ein，“最多”只有$m_H(N)$个不同的Ein，是不是可以这样？

$$\mathbb{P}[\exists h\in\mathcal{H}\ \text{s.t.}\ |E_{in}(h)-E_{out}(h)|>\epsilon]\leq 2m_H(N)exp(-2N\delta^2)$$

- 对比之前的上界（注意，“并”用“存在”表示；注意有效结果Ein“最多”只有$m_H(N)$）

基于增长函数的误差上界

- 这样直接替换问题在哪？

$$\mathbb{P}\left[\exists\, h \in \mathscr{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon\right] \leq 2m_H(N)exp(-2N\delta^2)$$

- 问题：因为（训练）样本D有限，经验误差Ein的可能取值是有限个的，但Eout的可能取值是无限的
- 方案：用有限验证集(verification set)D’（也有N个样本），用 Ein’ 来近似无限Eout，为此我们做一些变换：

Step 1: Replace E_{out} by E'_{in}

$$\frac{1}{2}\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon\right]$$

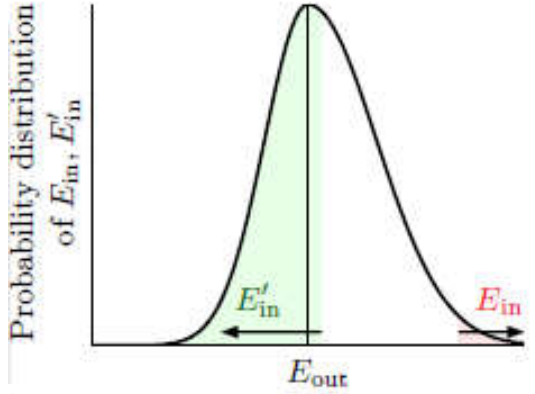
$$\leq \mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E'_{in}(h)| > \frac{\epsilon}{2}\right]$$

基于增长函数的误差上界

- Step1的简单证明:
- 如果存在h在D上使得
- 有很大概率使得h在D‘上$|E_{in}(h) - E_{out}(h)| > \epsilon$与E-out(h)的距离相比E-in(h)与E-out(h)应该更近：

$$\Pr(|E'_{in}(h) - E_{out}(h)| \leq \epsilon/2) \geq \Pr(|E'_{in}(h) - E_{in}(h)| \leq \epsilon/2) \quad \text{--- (1)}$$

- 由于 $|E_{in}(h) - E_{out}(h)| > \epsilon$
- 以上两事件没有交集（反证，如果有交
- 集：存在一个h，推出上式小于ϵ（三角
- 不等式））。两事件没交集，故：



$$\Pr(|E'_{in}(h) - E_{out}(h)| \leq \epsilon/2) + \Pr(|E'_{in}(h) - E_{in}(h) \leq \epsilon/2) \leq 1 \quad \text{---(2)}$$

- 代（1）入（2）有

$$\Pr(|E'_{in}(h) - E_{in}(h)| \leq \epsilon/2) \leq 1/2 \text{ 也就是:}$$
$$\Pr(|E'_{in}(h) - E_{in}(h)| > \epsilon/2) \geq 1/2$$

- 更有： $\Pr(\exists h \text{ s.t. } |E_{in}(h) - E'_{in}(h)| > \epsilon/2) \geq \frac{1}{2}\Pr(\exists h \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon)$

基于增长函数的误差上界

- Step 2：

$$\Pr(\exists h \text{ s.t. } |E_{in}(h) - E'_{in}(h)| > \epsilon/2) \geq \frac{1}{2}\Pr(\exists h \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon)$$

- 上式两边乘2， 右边定义为BAD事件，则

Step 2: Decompose $\mathcal{H}$ by Kind

$$\begin{aligned} \text{BAD} &\leq 2\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } |E_{\text{in}}(h) - E'_{\text{in}}(h)| > \frac{\epsilon}{2}\right] \\ &\leq 2m_{\mathcal{H}}(2N)\mathbb{P}\left[\text{fixed } h \text{ s.t. } |E_{\text{in}}(h) - E'_{\text{in}}(h)| > \frac{\epsilon}{2}\right] \end{aligned}$$

- 注意，现在Ein 从D上计算，Ein’从D’上计算，D和D’总共2N个数据，就算D和D’上的有效映射都不重复，最多 $m_H(2N)$ 个有效映射

基于增长函数的误差上界

- Step 3.
- 做等效变换：

$$|E_{in} - E'_{in}| > \frac{\epsilon}{2} \Leftrightarrow \left|E_{in} - \frac{E_{in}+E'_{in}}{2}\right| > \frac{\epsilon}{4}$$

- Step2右边等价于考虑 D上的误差期望与（D+D‘）上的误差期望的差距，应用Hoeffding不等式得：

$$P\left(|\tilde{X} - E\left(\tilde{X}\right)| \geq \delta\right) \leq 2exp\left(-2N\delta^2\right)$$

Step 3: Use Hoeffding without Replacement

$$\begin{aligned} \text{BAD} &\leq 2m_{\mathcal{H}}(2N)\mathbb{P}\left[\text{fixed } h \text{ s.t. } |E_{in}(h) - E'_{in}(h)| > \frac{\epsilon}{2}\right] \\ &\leq 2m_{\mathcal{H}}(2N) \cdot 2\exp\left(-2\left(\frac{\epsilon}{4}\right)^2 N\right) \end{aligned}$$

VC bound

- 在以上变换后，最终得到：

Vapnik-Chervonenkis (VC) bound:

$$\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } \left|E_{\text{in}}(h) - E_{\text{out}}(h)\right| > \epsilon\right]$$
$$\leq 4 m_{\mathcal{H}}(2N) \exp\left(-\frac{1}{8}\epsilon^2 N\right)$$

- 此上界提供了一个对机器学习结果可靠性的衡量
- 成长函数是N的多项式，故“学习结果差”事件发生的概率随着N的增大而显著下降
- N足够大，H中任意假设h，其经验误差都将接近泛化误差

VC Dimension

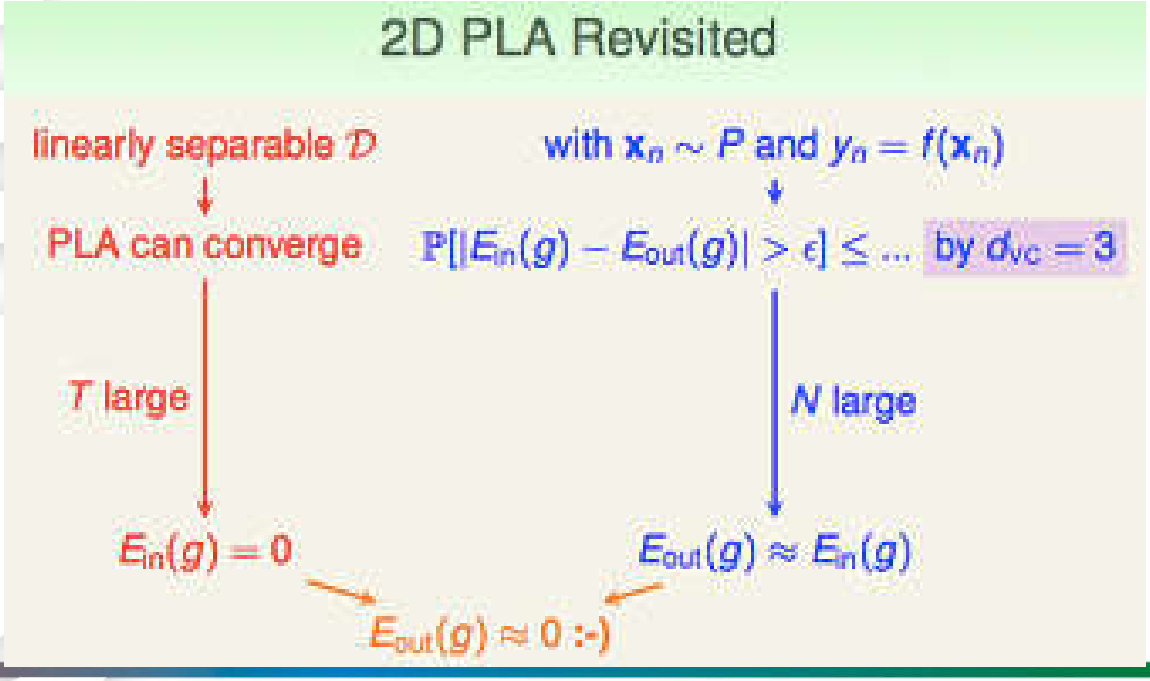
- 一个假设空间H的**VC dimension**，是这个H最多能够shatter掉的点的数量

- $d_{VC}(H)=\max\{N: m_H(N)=2^N\}$，$m_H(N)$为增长函数
- 和break point的关系:
- $k = d_{VC}(H)+ 1$

- VC维的大小：与学习算法A无关，与输入变量X的分布也无关，与我们求解的目标函数f 无关。它只与模型和假设空间**H**有关

VC dimension

- 对于2维的perceptron，它不能shatter 4个样本点，VC维是3
- 如果样本集是线性可分的，perceptron learning algorithm可以在假设空间里找到一条直线，使Ein(g)=0；
- 另外由于其VC维=3，当N足够大的时候，可以推断出：Eout(g)约等于Ein(g), 此时2维感知机可学



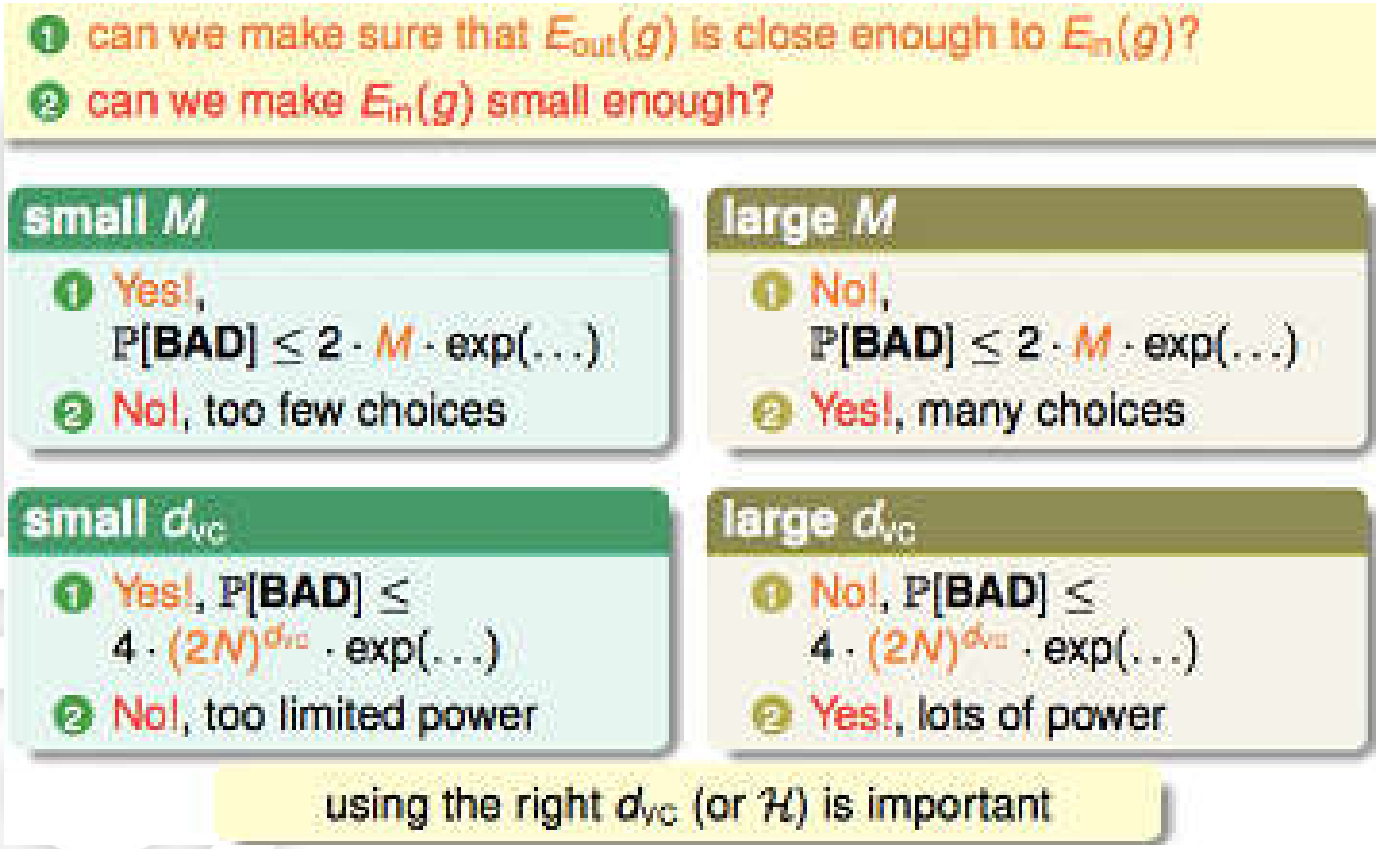
VC Dimension的意义

- VC维反映了假设空间H 的强大程度(powerfulness)，VC 维越大，H也越强，因为它可以打散(shatter)更多的点

- 基于$d_{VC}(H)$的机器学习可行判别：
  - $d_{VC}(H)$有限，VC bound才存在。 $d_{VC}(H)$小， $m_H(N)$才能尽量小（最高k-1次方）(good H 假设空间)；
  - N足够大(对于特定的$d_{VC}(H)$而言)，这样才能保证vc bound不等式的bound不会太大(good D 数据)
  - 算法A有办法在H中顺利的挑选一个使得Ein最小的g。(good A 优化方法)

VC Dimension

- 从假设空间H大小M和$d_{VC}(H)$两个角度比较机器学习可行性



VC dimension 与模型复杂度

- VC dimension 与模型复杂度
- 实践规律：VC 维与假设参数w 的自由变量数目大约相等。
  dVC = #free parameters
- 基于VC dimension的误差上界:

Vapnik-Chervonenkis (VC) bound:

$$\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon\right]$$
$$\leq 4 m_{\mathcal{H}}(2N) \exp\left(-\frac{1}{8}\epsilon^2 N\right)$$

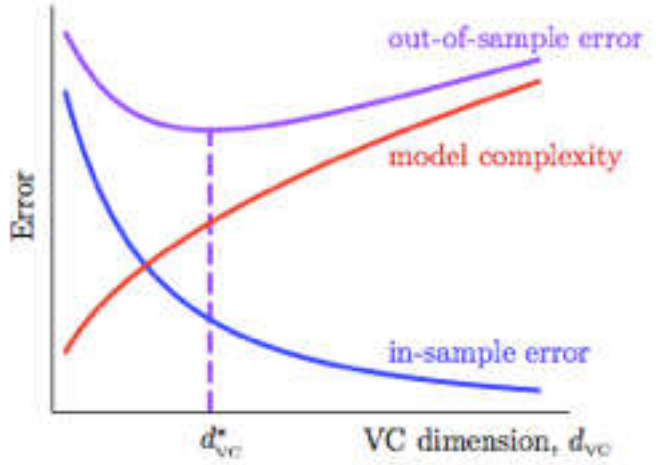
with a high probability,

$$E_{out}(g) \leq E_{in}(g) + \underbrace{\sqrt{\frac{8}{N}\ln\left(\frac{4(2N)^{d_{VC}}}{\delta}\right)}}_{\Omega(N,\mathcal{H},\delta)}$$

- 模型越复杂，VC维大，Eout 可能距离Ein 越远

VC dimension 与模型复杂度

- 如果VC Dimension太大，模型复杂度增加，E-in与E-out偏离
- 如果VC Dimension太小，虽然E-in≈E-out，但H不够给力，很难找到不犯错（或很少犯错）的h



- d_{VC} ↑: E_{in} ↓ but Ω ↑
- d_{VC} ↓: Ω ↓ but E_{in} ↑
- best d_{VC}^* in the middle

powerful $\mathcal{H}$ not always good!

VC bound与数据复杂度

- 例：用模型的VC Dimension = 3的模型做分类，要求E-in与E-out差距最大为ε=0.1；置信度为90%，即δ=0.1

given specs $\epsilon = 0.1, \delta = 0.1, d_{VC} = 3$, want $4(2N)^{d_{VC}} \exp\left(-\frac{1}{8}\epsilon^2 N\right) \leq \delta$

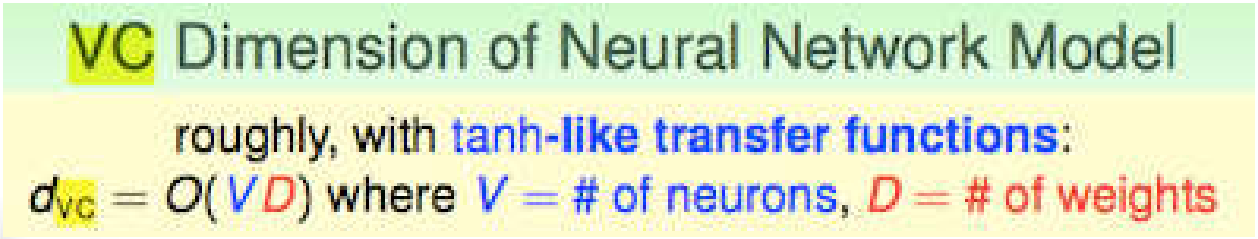
N	bound
100	2.82×10^{7}
1,000	9.17×10^{9}
10,000	1.19×10^{8}
100,000	1.65×10^{-38}
29,300	9.99×10^{-2}

sample complexity:
need $N \approx 10,000 d_{VC}$ in theory

- 理论上：need N ≈ 10000 * VC Dimension
- 实际应用： need N ≈ 10* VC Dimension
- 因为VC Bound 过于宽松，是一个比实际大得多的上界

深度学习与VC维

- 对于神经网络：



- 粗略估计: dVC = O(VD)，V:神经网络中神经元的个数，D:weight的个数，也就是神经元之间连接的数目。(注意：深度神经网络目前没有明确的vc bound)

- 举例：一个普通的三层全连接神经网络：input layer是1000维，hidden layer有1000个nodes，output layer为1个node，则它的VC维大约为O(1000*1000*1000)

深度学习与VC维

- 三层全连接神经网络，VC维大约为O(1000*1000*1000)
- 实际应用中至少需要10倍数据
- 为什么还效果好？
- （1）卷积网等网络（权值共享等大大降低参数）
- （2）一些适应网络的正则化技术
- （3）大量数据
- （4）预训练模型

- 启示：
- 1. 一定要划分训练、测试集，以测试集作为评价
- 2. 好的模型的标准：训练和测试效果相当，且测试误差很小
- 3. 模型太复杂，假设空间太大，关键是保证模型能力的情况下，减小假设空间：
- （1）更简单的模型假设
- （2）正则化技术

模型选择

- **奥卡姆剃刀原理：**
- **同样效果情况下，选简单的**

- **No Free Lunch Theorems**
- 没有哪个算法比其他算法高效。要根据具体问题，选择合适的假设。
- 某个模型在某方面有效，在另一方面可能就会差些

No Free Lunch Theorems

- f:希望学习到的映射， P(h|X,La)用算法La解出h映射的概率
- Eote(La|X,f)，算法La在X数据上的误差期望

$$E_{ote}(\mathfrak{L}_a|X,f)=\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\,\mathbb{I}\left(h\left(\boldsymbol{x}\right)\neq f\left(\boldsymbol{x}\right)\right)P\left(h\mid X,\mathfrak{L}_a\right)\ , \qquad (1.1)$$

- 考虑2分类问题：x->{0,1}函数空间为$\{0,1\}^{|x|}$

No Free Lunch Theorems

$$\begin{aligned}\sum_f E_{ote}(\mathfrak{L}_a|X,f) &= \sum_f \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x})\ \mathbb{I}(h(\boldsymbol{x})\neq f(\boldsymbol{x}))\ P(h\mid X,\mathfrak{L}_a)\\ &= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h\mid X,\mathfrak{L}_a) \sum_f \mathbb{I}(h(\boldsymbol{x})\neq f(\boldsymbol{x}))\\ &= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h\mid X,\mathfrak{L}_a) \frac{1}{2} 2^{|\mathcal{X}|}\\ &= \frac{1}{2} 2^{|\mathcal{X}|} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h\mid X,\mathfrak{L}_a)\\ &= 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x})\cdot 1\ . \end{aligned} \tag{1.2}$$

$$\sum_f E_{ote}(\mathfrak{L}_a|X,f) = \sum_f E_{ote}(\mathfrak{L}_b|X,f)\ , \tag{1.3}$$

- 算法La和算法Lb期望相同（注意前提条件：所有f均匀分布）

模型选择

- 两个关键问题
  - 如何获得测试结果？ ⇨ 评估方法
  - 如何评估性能优劣？ ⇨ 性能度量

评估方法

- 关键：如何获得“测试集”(test set)?

  测试集与训练集应该“互斥”

- 常见方法
  - 留出法（hold-out）
  - 交叉验证法（cross validation）
  - 自助法（bootstrap）

留出法

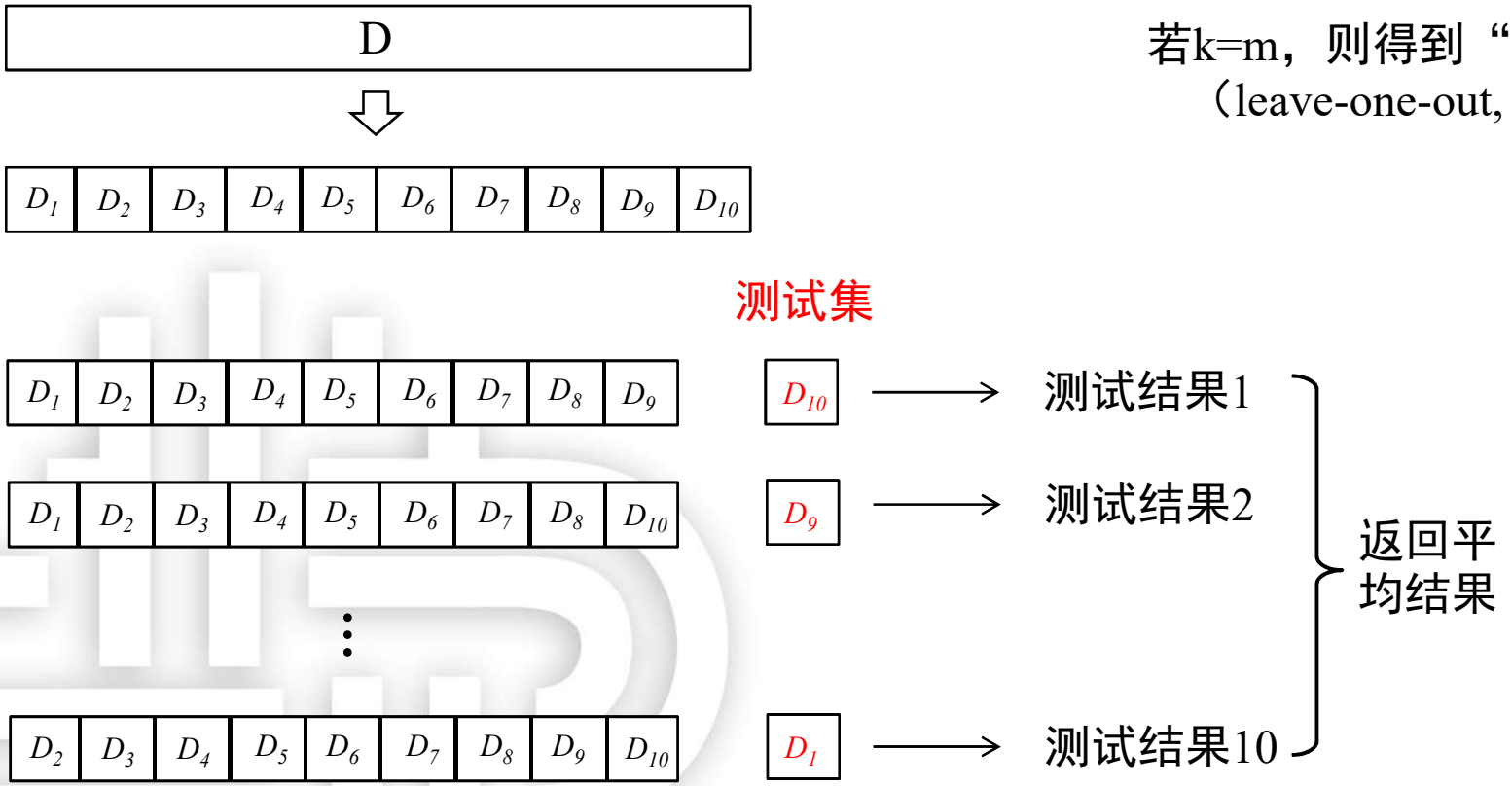
- “留出法” 直接将数据集D划分为两个互斥的集合，其中一个集合作为训练集，另一个作为测试集。

- 注意
  - 保持数据分布的一致性（例如：保留类别比例的分层采样）
  - 多次重复划分（例如：100次随机划分）
  - 测试集不能太大或太小（例如：1/5 ~ 1/3）

留出法的问题

- 若令训练集包含绝大多数样本，则训练出的模型可能更接近于用验证集训练出的模型，但由于测试集比较小，评估结果可能不够稳定准确;
- 若令测试集多包含一些样本，则训练集与验证集差别更大，被评估的模型与训练出的模型相比可能有较大差别，从而会降低了评估结果的保真性。

K-折交叉验证法

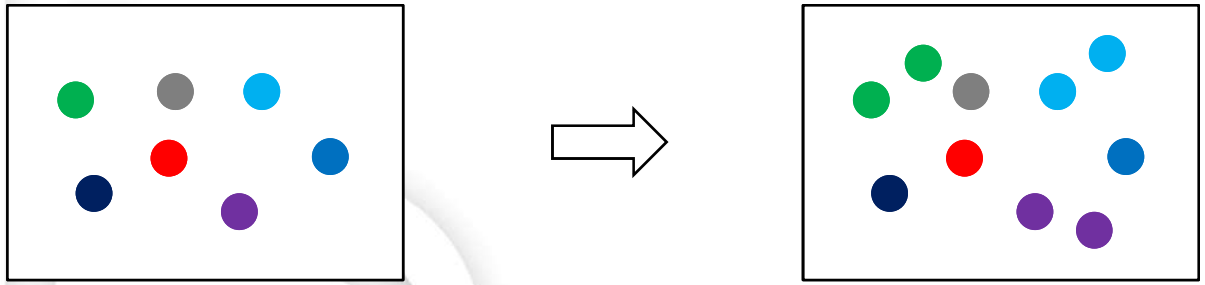


若k=m，则得到“留一法”
（leave-one-out, LOO）

自助法

- 基于“自助采样”(bootstrap sampling)

亦称“有放回采样”、“可重复采样”



- 训练集与原样本集规模相同
- 数据分布有所改变

约有36.8%的样本不出现

$$\lim_{m\to\infty}\left(1-\frac{1}{m}\right)^{m}\to\frac{1}{e}\approx 0.368$$

“包外估计”（out-of-bag estimation）

自助法

- 优点：
  - 在数据集较小、难以有效划分训练/测试集时很有用
  - 能从初始数据集中产生多个不同的训练集，这对集成学习等方法有很大的好处
- 缺点：
  - 自助法产生的数据集改变了初始数据集的分布，这会引入估计偏差

调参与最终模型

- 算法的参数：一般人工设定，亦称“超参数”
- 模型的参数：一般由学习确定

调参过程相似：先产生若干模型，然后基于某种评估方法进行选择

参数调试的好坏往往直接影响最终性能

区别：训练集 vs. 测试集 vs. 验证集（validation set）

算法参数选定后，要用“训练集+验证集”重新训练最终模型

模型选择

- 两个关键问题
  - 如何获得测试结果？ ⇨ 评估方法
  - 如何评估性能优劣？ ⇨ 性能度量

模型的性能度量

- 性能度量(performance measure):
  - 衡量模型泛化能力的评价标准

什么样的模型是“好”的，不仅取决于算法和数据，还取决于任务需求

- 常用的度量标准有：
  - 错误率与精度
  - 查准率，查全率，F1, P-R图
  - ROC与AUC
  - 代价敏感错误率与代价曲线

错误率 vs. 精度

- 错误率

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(x_i)\neq y_i)$$

- 精度

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(x_i)=y_i)$$
$$=1-\mathrm{E(f;D)}$$

精度和错误率的问题

- 当正负样例在数据集中严重失衡时，会使得这种度量方式缺乏说服力。

例如生物信息学中的基因剪辑点的识别任务中，正样例的占比可能只有万分之一，那么假设有一个无效模型，把所有的样例都分类为负样例，这样它的错误率会小于0.01%，而精度会高于99.99%，但这个模型显然是无效的。

混淆矩阵

- 对于二分类问题，可将样例根据其真实类别与学习器预测类别的组合划分为真正例(true positive)、假正例(false positive)、真反例(true negative)、假反例(false negative)四种情形。
- 分类结果的“混淆矩阵”(confusion matrix)如下

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

查准率（精度）、查全率（召回率）

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

查准率（精度）precision：

$$P = \frac{TP}{TP + FP}$$

查全率(召回率）recall：

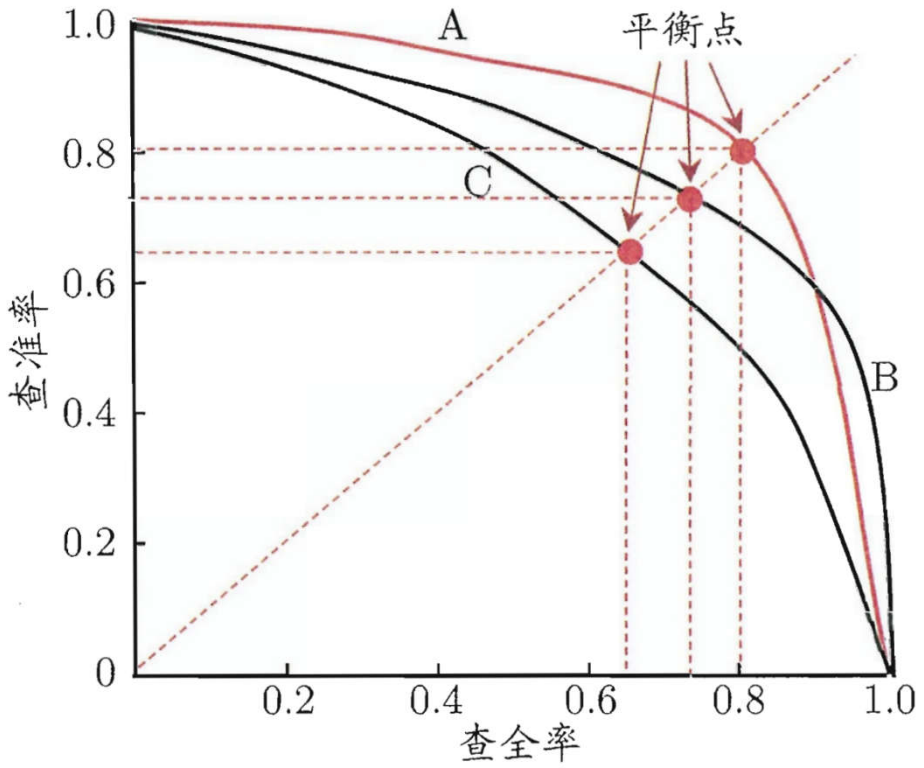
$$R = \frac{TP}{TP + FN}$$

PR图

根据学习器的预测结果按正例可能性大小对样例进行排序，并逐个把样本作为正例进行预测

PR图：
- 学习器A优于学习器C
- 学习器B优于学习器C
- 学习器A ?? 学习器B

BEP：
- 学习器A优于学习器B
- 学习器A优于学习器C
- 学习器B优于学习器C



F1

比BEP更常用的F1度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

若对查准率/查全率有不同偏好：

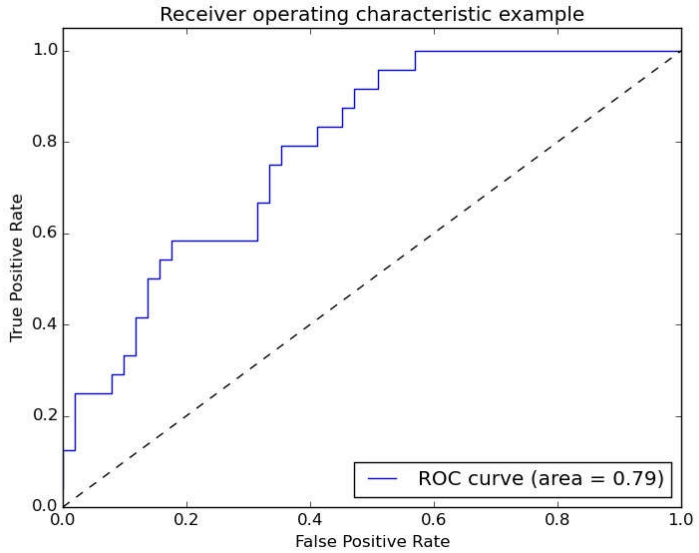
$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

β>1时查全率有更大影响； $\beta < 1$时查准率有更大影响

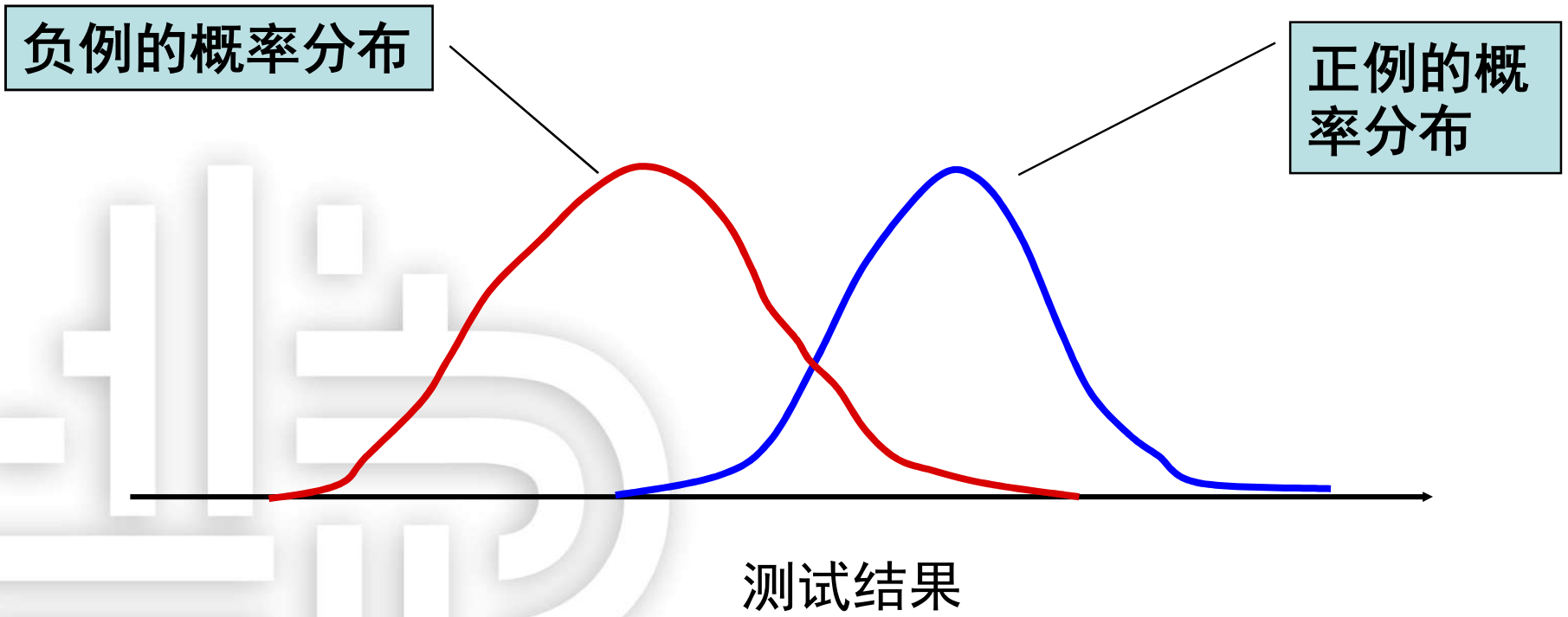
ROC图

- ROC (Receiver Operating Characteristic) 曲线，与P-R图类似，纵轴是“真正例率” (True Positive Rate，TPR)，横轴是“假正例率” (False Positive Rate，FPR)。其中：

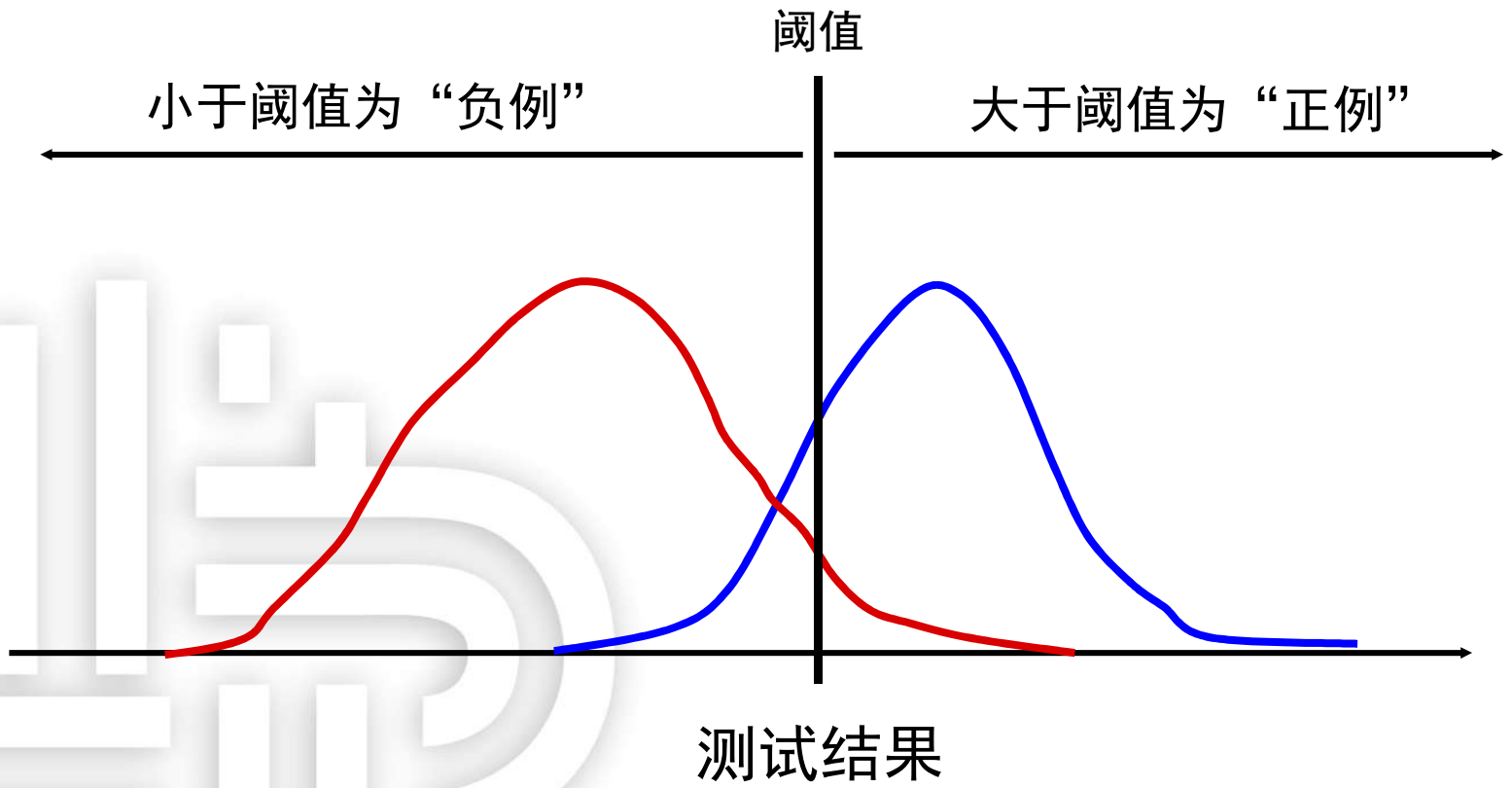
$$\mathrm{TPR} = \frac{TP}{TP + FN} \qquad \mathrm{FPR} = \frac{FP}{TN + FP}$$



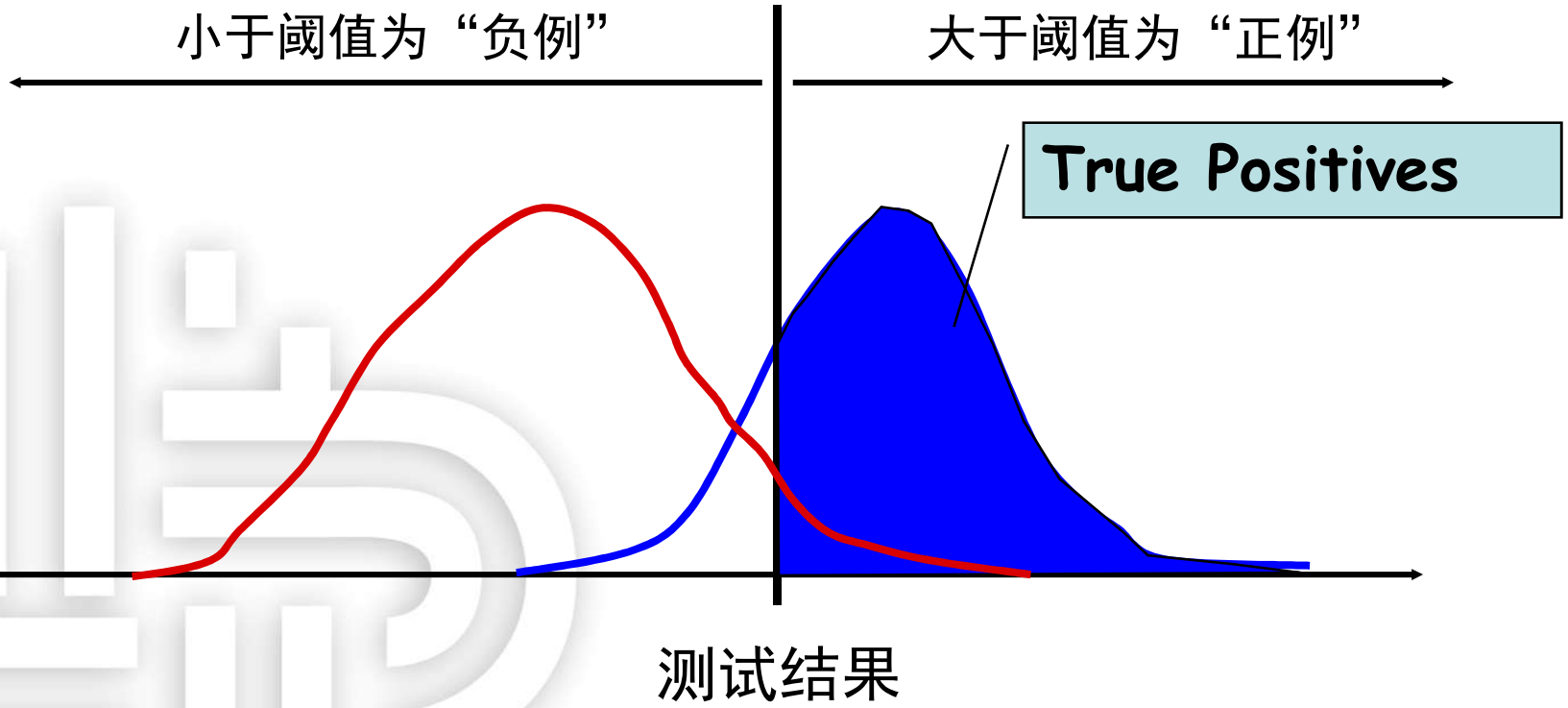
举例



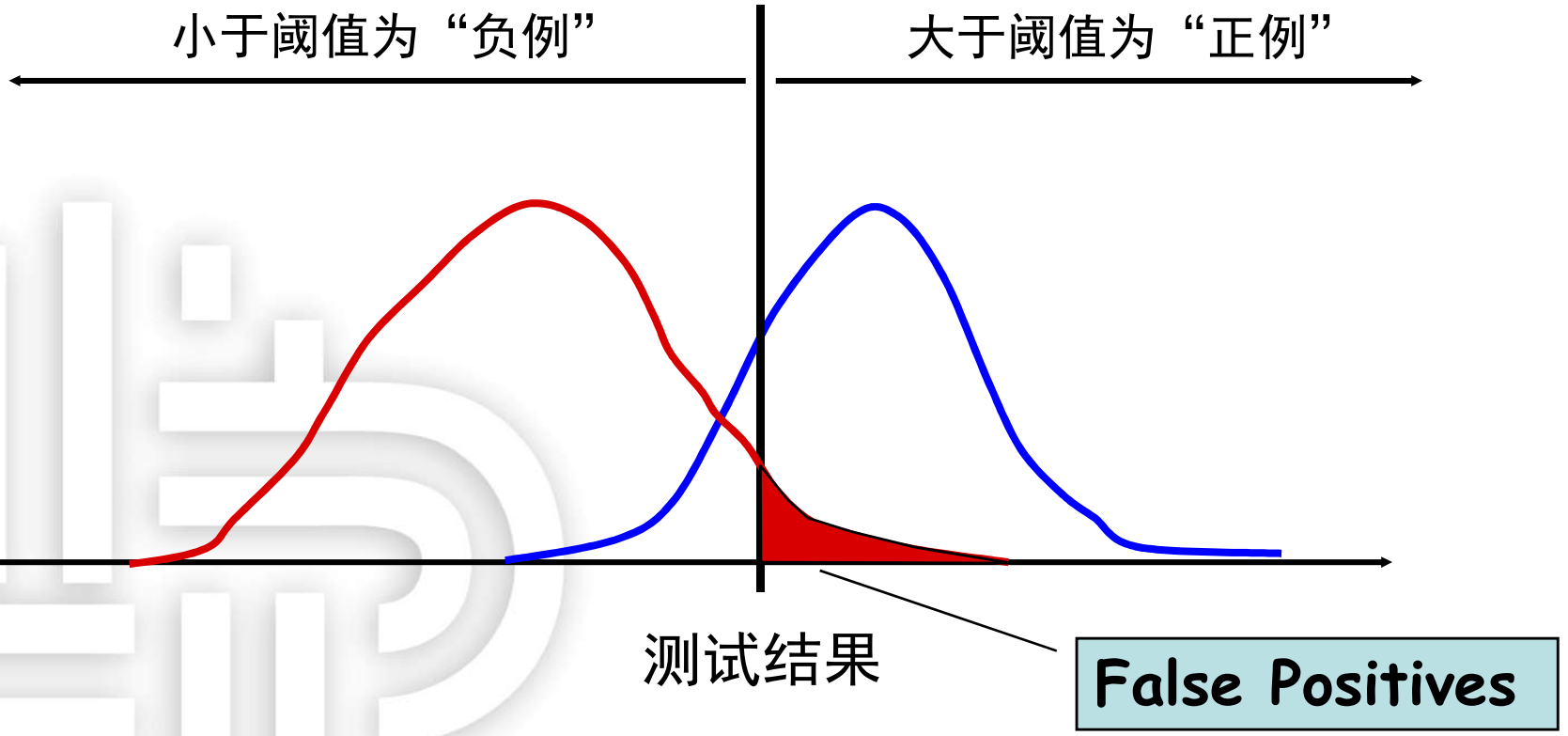
阈值判定

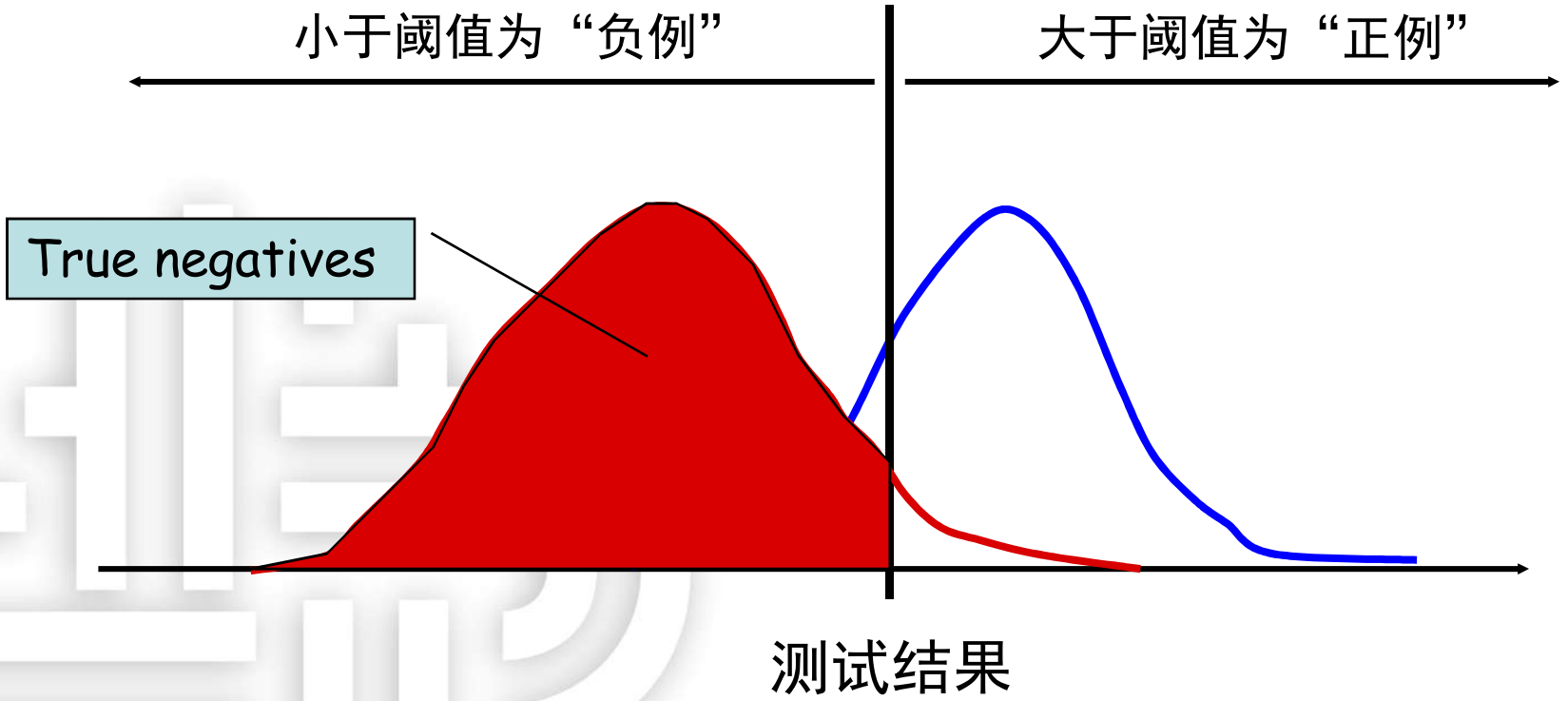


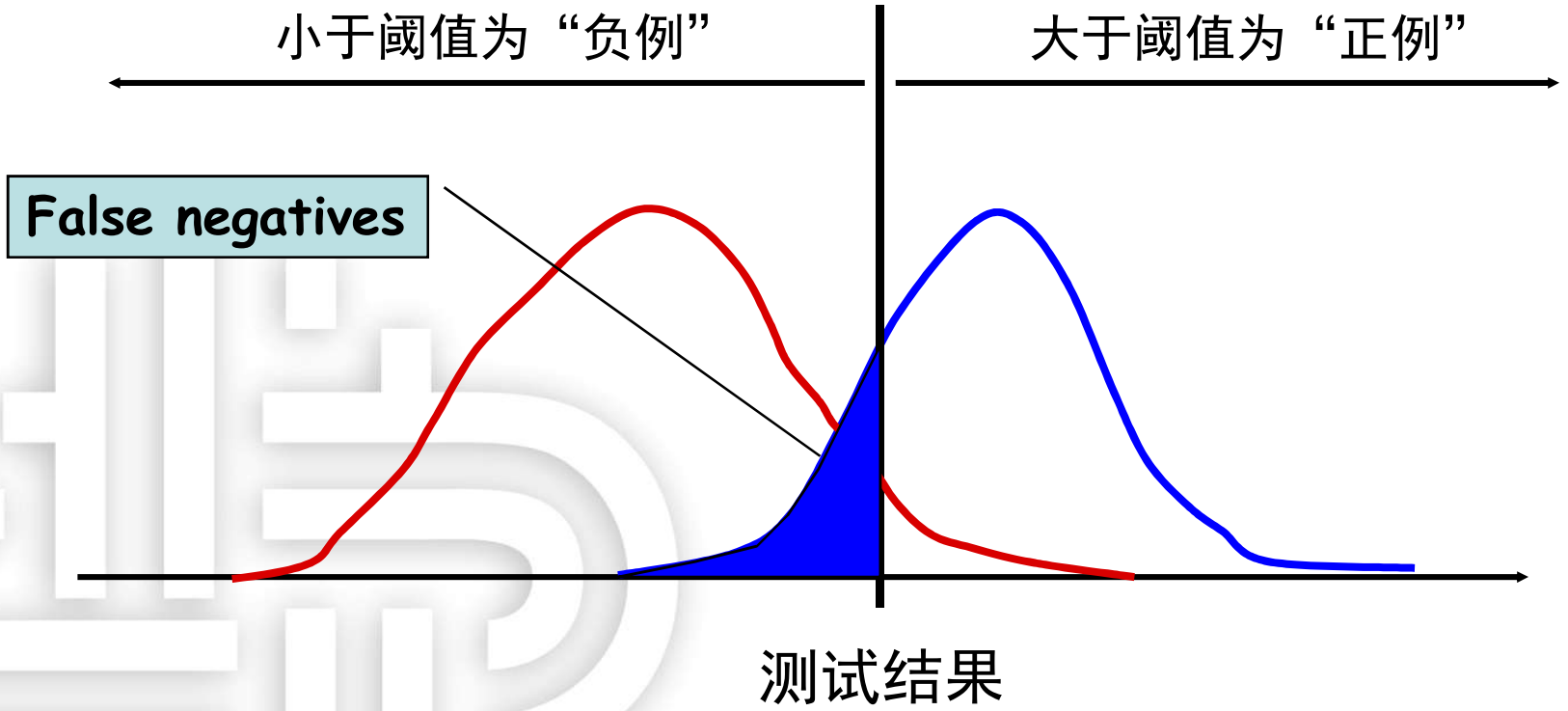
一些定义



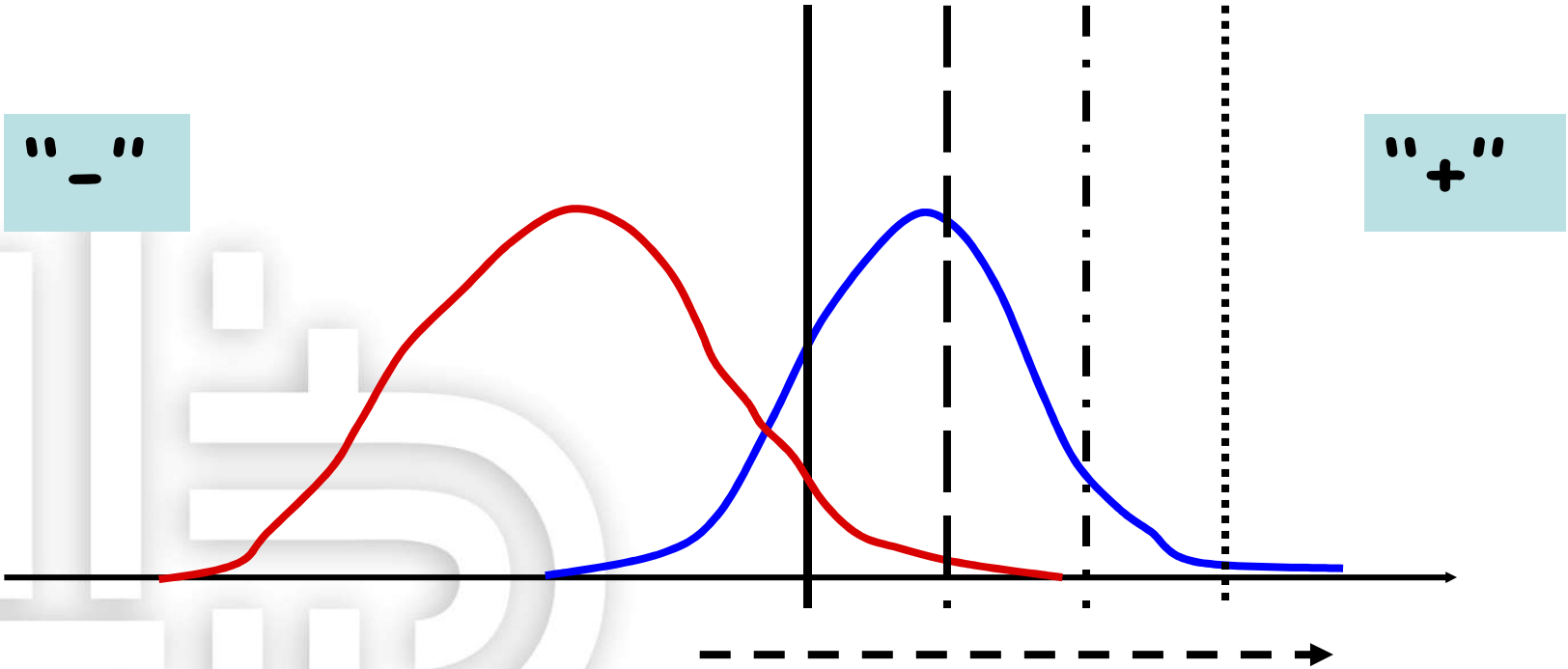
负样本　正样本


小于阈值为“负例”
大于阈值为“正例”
测试结果
False Positives
负样本
正样本

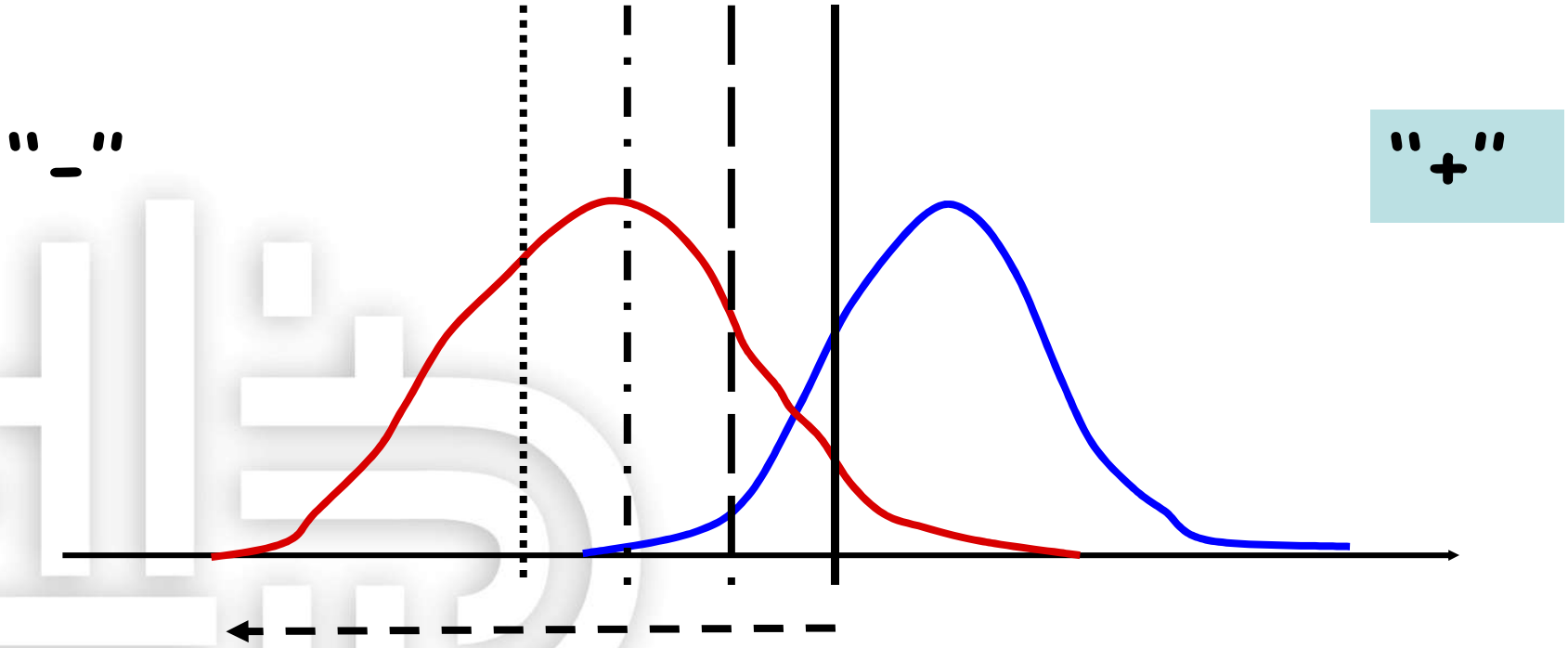

小于阈值为“负例”
大于阈值为“正例”
True negatives
测试结果
负样本
正样本


小于阈值为“负例”
大于阈值为“正例”
False negatives
测试结果
负样本
正样本

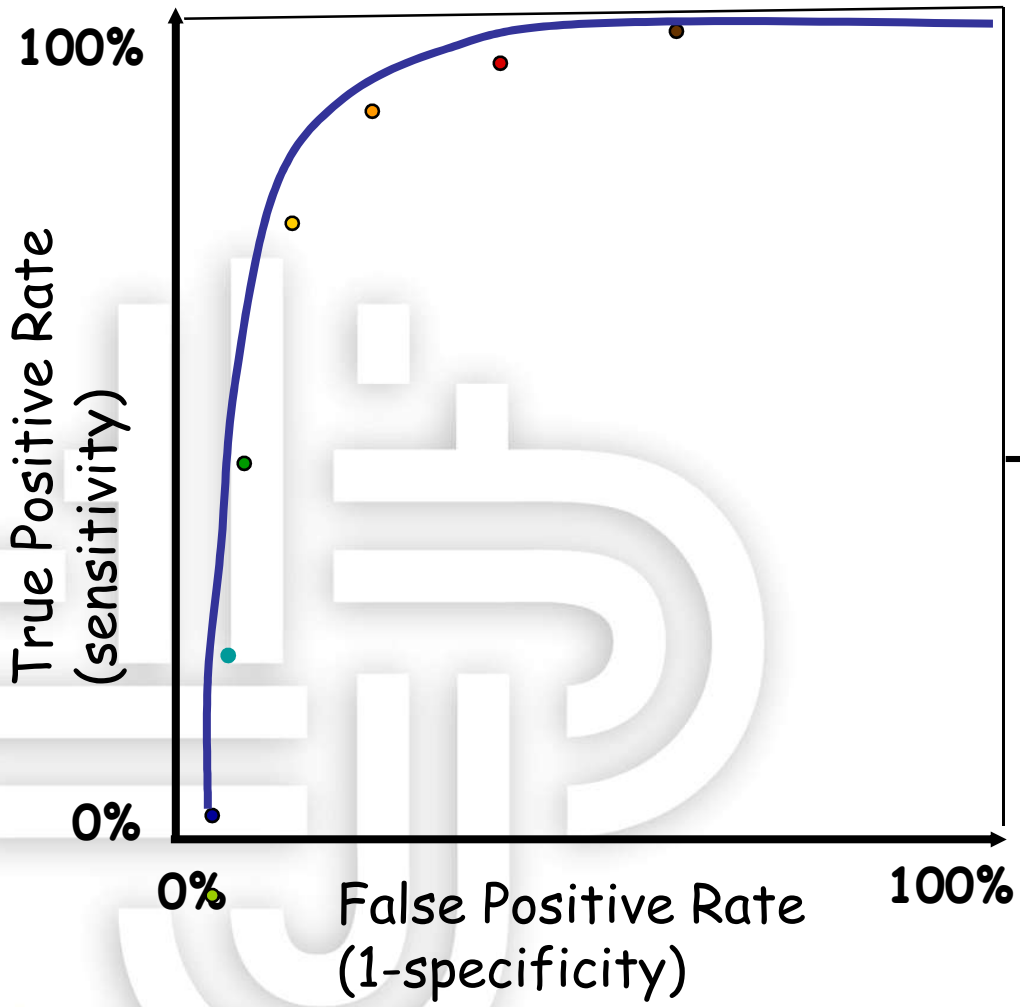
将“阈值”右移

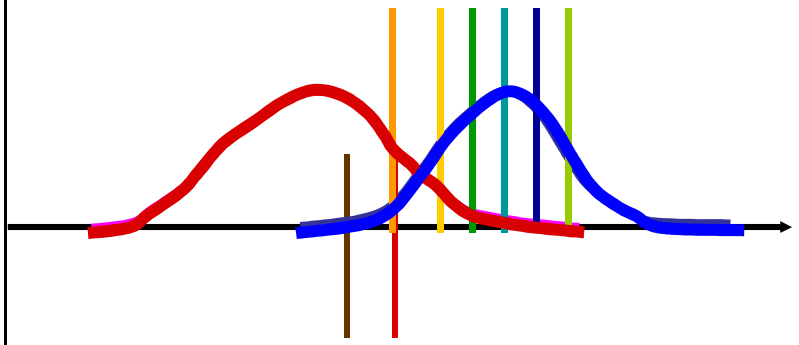


将“阈值”左移



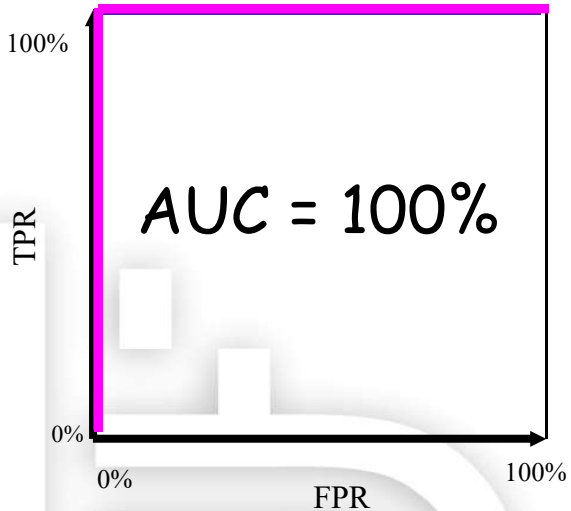
ROC 曲线

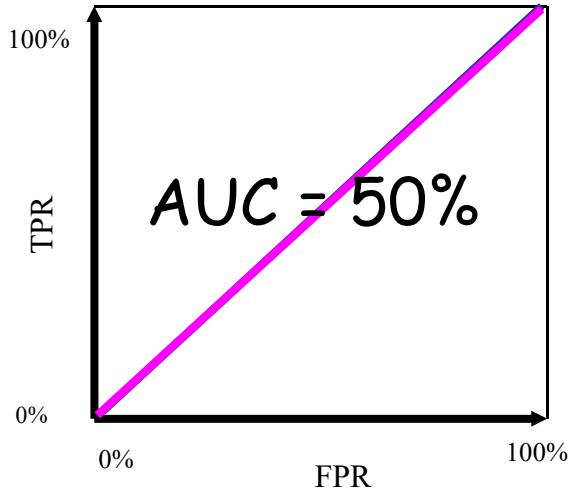

100%
True Positive Rate
(sensitivity)
0%
0%
False Positive Rate
(1-specificity)
100%

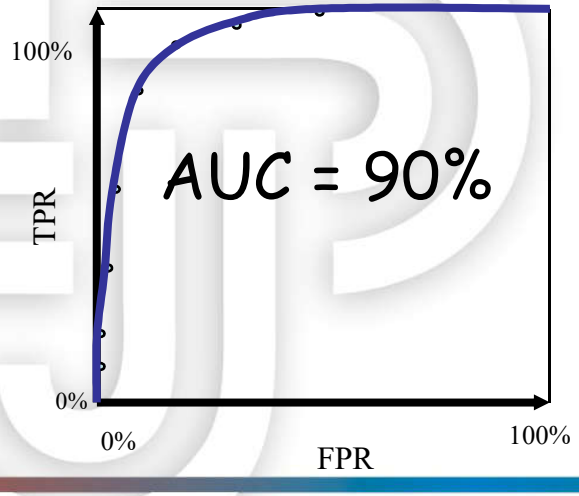


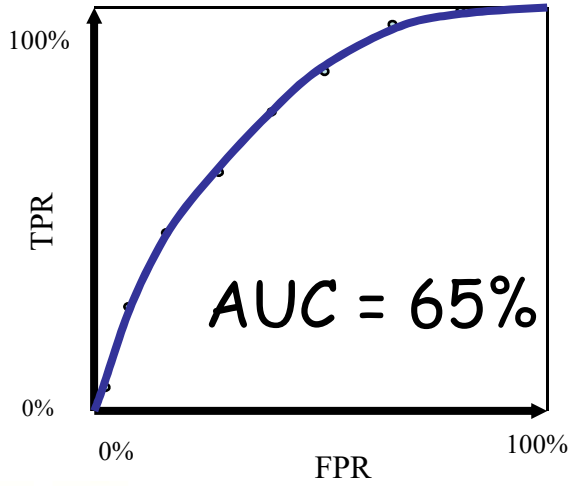
AUC for ROC 曲线

ROC曲线所包围的面积，便称为AUC(Area Under ROC Curve)



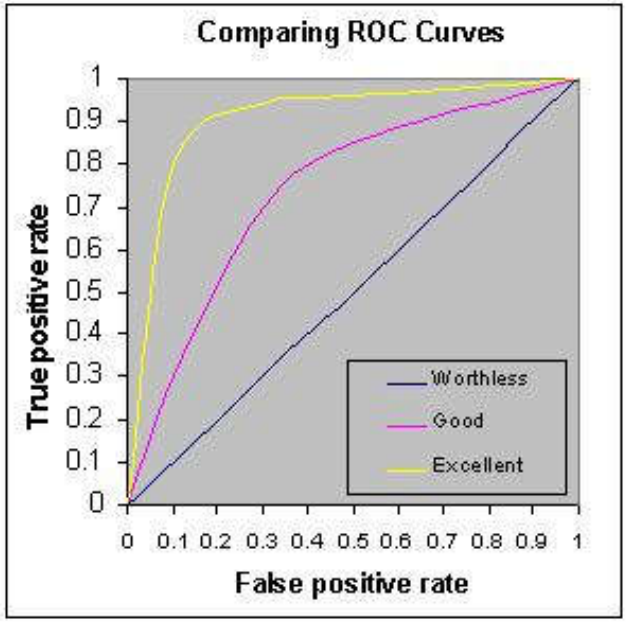


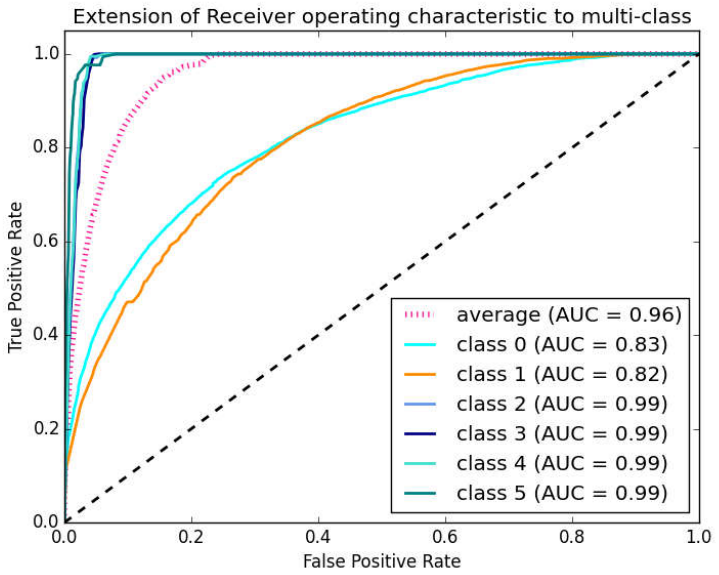




如何看ROC曲线？

- 若一个学习器的ROC 曲线被另一个学习器的曲线完全包住， 则可断言后者性能优于前者
- 若两条ROC 曲线交叉，则难以一般性断言孰优孰劣。此时如果一定要进行比较， 则可以用AUC来进行比较





小结

- 机器学习可行性
  - VC bound
  - VC dimension
- 评估方法
  - 过拟合与欠拟合
  - 留出法、交叉检验、bootstrap sampling
- 性能度量
  - 错误率与精度
  - F1、P-R图、ROC曲线、AUC
  - 误差分析