数据挖掘与机器学习

潘斌

panbin@nankai.edu.cn

范孙楼227

上节回顾

- 数据预处理的方法
  - 降维（特征提取、特征选择）
  - 汇总统计
  - 可视化
- 概念学习
  - 根据训练数据，推测布尔函数
  - more-general-than-or-equal-to，偏序
  - 求解假设h

x_1= <Sunny, Warm, High, Strong, Cool, Same>
x_2= <Sunny, Warm, High, Light, Warm, Same>

h_1= <Sunny, ?, ?, Strong, ?, ?>
h_2= <Sunny, ?, ?, ?, ?, ?>
h_3= <Sunny, ?, ?, ?, Cool, ?>

- 求解h的算法
  - **Find-S Algorithm**
    - 基本思想：
      - 使用more_general_than偏序的搜索算法
      - 沿着偏序链，从较特殊的假设逐渐转移到较一般的假设。
      - 在每一步，假设只在需要覆盖新的正例时被泛化。
      - 每一步得到的假设，都是在那一点上与训练样例一致的最特殊的假设。

- Find-S Algorithm
  - 将h初始化为H中最特殊假设
  - 对每个正例x
    - 对h的每个属性约束ai
    - 如果x满足ai
    - 那么不做任何事
    - 否则，将h中ai替换为x满足的紧邻的更一般约束
  - 输出假设h

- **Find-S算法实例** $h_0 \leftarrow <\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset>$

$x_1 = <Sunny\ Warm\ Normal\ Strong\ Warm\ Same>,\ +$

$h_1 \leftarrow <Sunny, Warm, Normal, Strong, Warm, Same>$

$x_2 = <Sunny\ Warm\ High\ Strong\ Warm\ Same>,\ +$

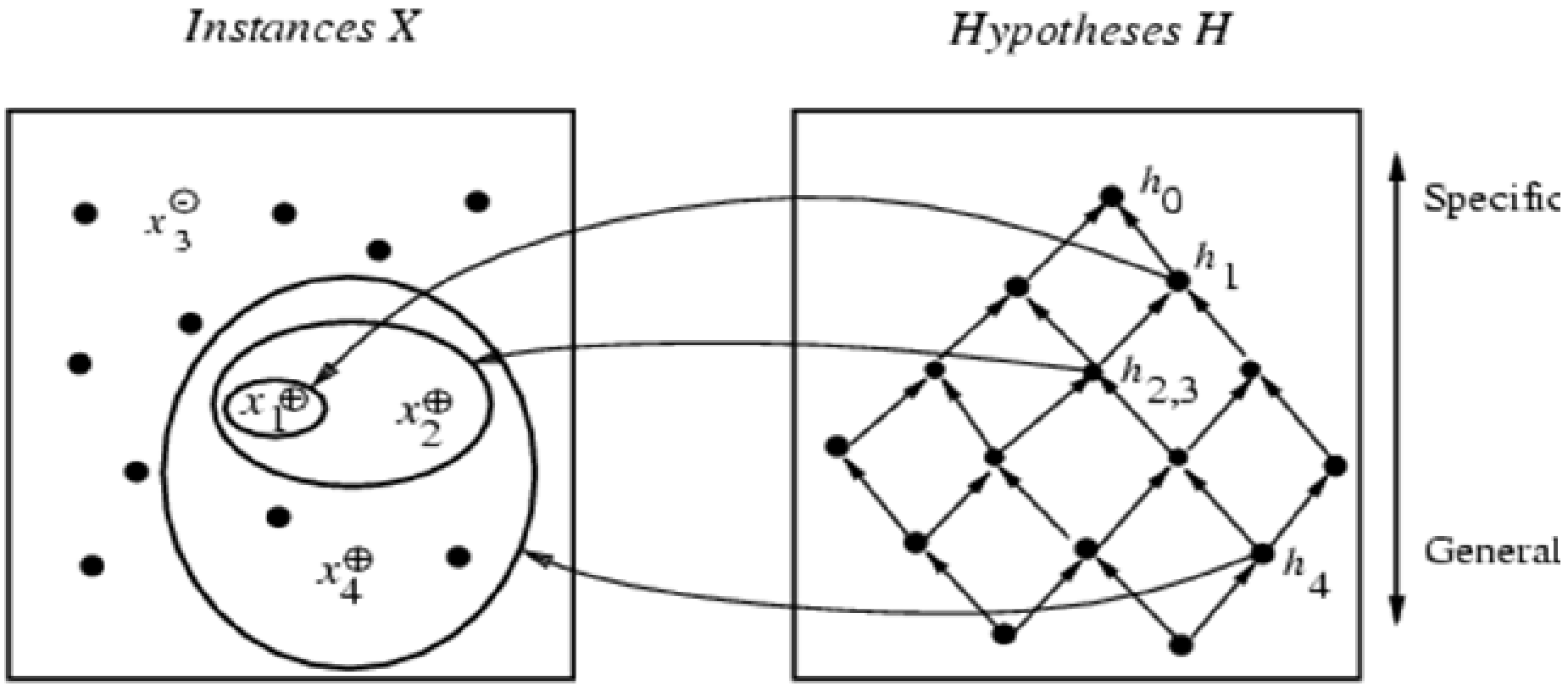
$h_2 \leftarrow <Sunny, Warm, ?, Strong, Warm, Same>$

$x_3 = <Rainy\ Cold\ High\ Strong\ Warm\ Change>,\ -$

$h_3 \leftarrow <Sunny, Warm, ?, Strong, Warm, Same>$

$x_4 = <Sunny\ Warm\ High\ Strong\ Cool\ Change>,\ +$

$h_4 \leftarrow <Sunny, Warm, ?, Strong, ?, ?>$


Instances X
Hypotheses H
$x_3^{\ominus}$
$x_1^{\oplus}$
$x_2^{\oplus}$
$x_4^{\oplus}$
h_0
h_1
$h_{2,3}$
h_4
Specific
General

练习

Outlook	Temperature	Humidity	Wind	PlayTennis
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Sunny	Cool	Normal	Weak	Yes
Overcast	Hot	Normal	Weak	Yes

- Find-S学习算法的特点及不足
  - Find-S的重要特点：对以属性约束的合取式（conjunction）描述的假设空间H，保证输出为H中与正例一致的最特殊的假设。
  - 存在的问题（反例；容错性）

- 变型空间（Version space，版本空间）
  - 定义：关于假设空间H和训练样例集D的变型空间，标记为$VS_{H,D}$，是H中与训练样例集D一致的所有假设构成的子集。

$$VS_{H,D} = \{h \in H \mid Consistent(h,D)\}$$

  - 与训练样例集一致的所有假设组成的集合
  - 包含的是目标概念的所有合理的变型

- 求解h的算法
  - 列表后消除算法（The List-Then-Eliminate Algorithm）
    - 变型空间VersionSpace←包含H中所有假设的列表
    - 对每个训练样例<x, c(x)> 从变型空间中移除所有h(x)≠c(x)的假设h
    - 输出VersionSpace中的假设列表
      - 只要假设空间是有限的，就可使用
      - 保证得到所有与训练数据一致的假设
      - 非常繁琐地列出H中的所有假设，大多数实际的假设空间无法做到

- 变型空间举例
  - EnjoySport

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

结果如→

每来一个样本，压缩一次假设空间

h1: <*Sunny, Warm, ?, Strong, ?, ?*>

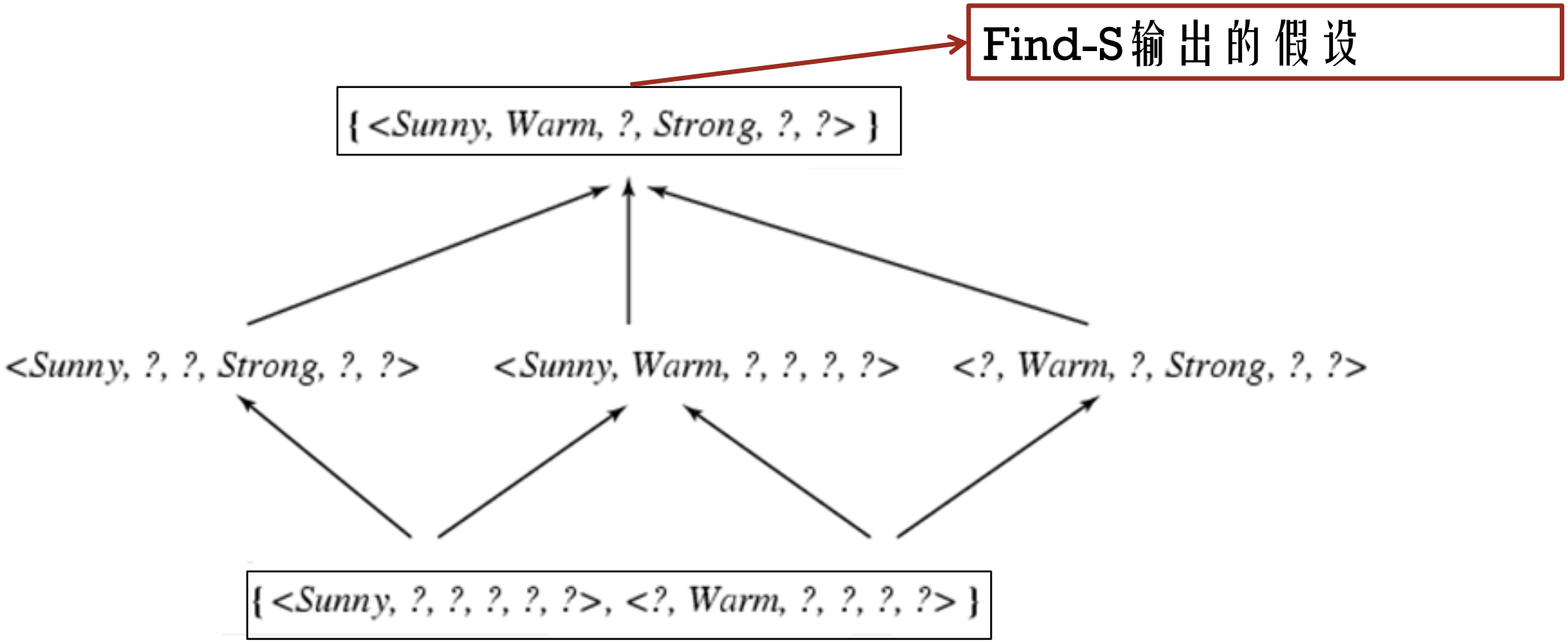
h2: <*Sunny, ?, ?, Strong, ?, ?*>

h3: <*Sunny, Warm, ?, ?, ?, ?*>

h4: <*?, Warm, ?, Strong, ?, ?*>

h5: <*Sunny, ?, ?, ?, ?, ?*>

h6: <*?, Warm, ?, ?, ?, ?*>


Find-S输出的假设
{<Sunny, Warm, ?, Strong, ?, ?>}
<Sunny, ?, ?, Strong, ?, ?>
<Sunny, Warm, ?, ?, ?, ?>
<?, Warm, ?, Strong, ?, ?>
{<Sunny, ?, ?, ?, ?, ?>, <?, Warm, ?, ?, ?, ?>}

- 求解h的算法：候选消除算法
  - 变型空间的两个边界定义
    - 一般边界（General Boundary）G：
      - H中与D相一致的极大一般成员的集合。

$$G \equiv \{ g \in H \mid Consistent(g, D) \wedge (\neg \exists g' \in H)[(g' >_g g) \wedge Consistent(g', D)]\}$$

    - 特殊边界（Specific Boundary）S:
      - 在H中与D相一致的极大特殊成员的集合。

$$S \equiv \{ s \in H \mid Consistent(s, D) \wedge (\neg \exists s' \in H)[(s >_g s') \wedge Consistent(s', D)]\}$$

求解h的算法

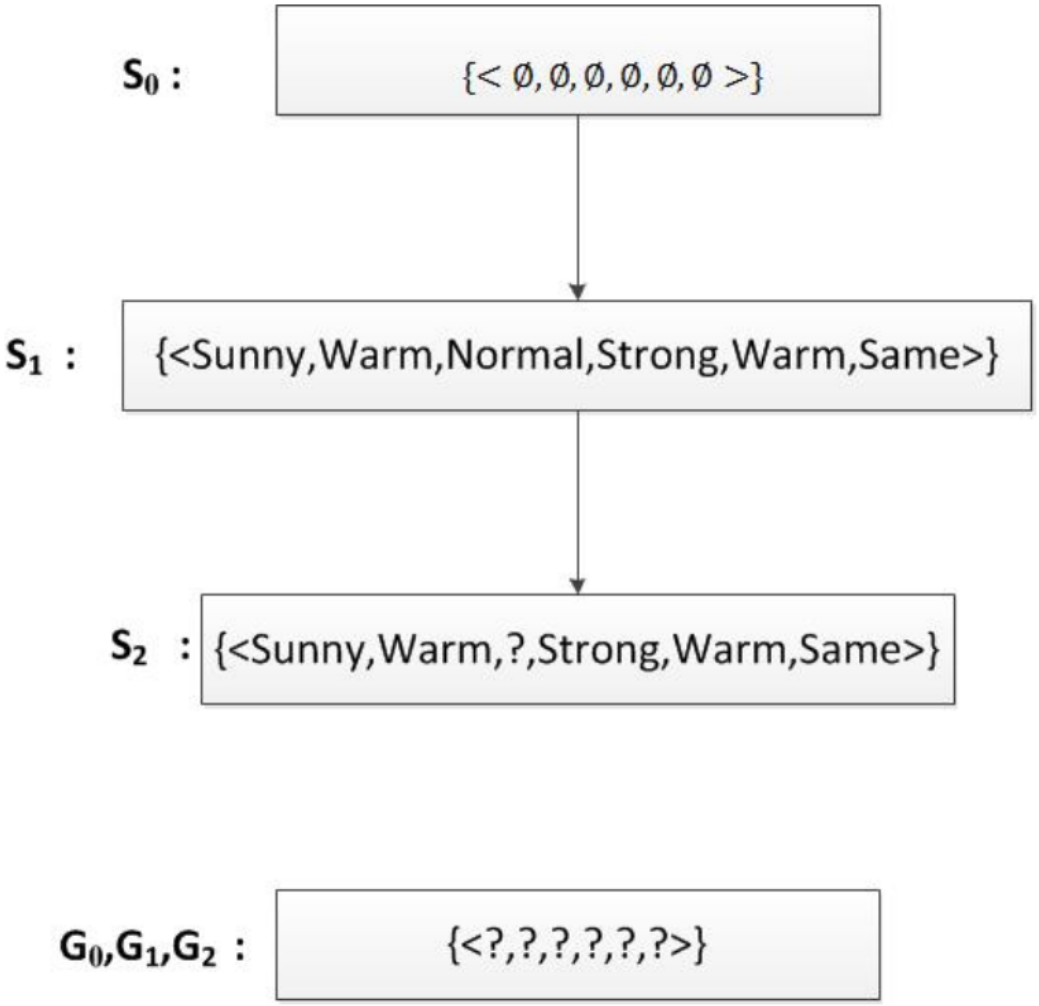
候选消除算法

将G集合初始化为H中极大一般假设

将S集合初始化为H中极大特殊假设

对每个训练例d，进行以下操作：

- 如果d是一正例
  - 从G中移去所有与d不一致的假设
  - 对S中每个与d不一致的假设s
    - 从S中移去s
  - 把s的所有的极小一般化式h加入到S中，其中h满足
    - h与d一致，而且G的某个成员比h更一般
  - 从S中移去所有这样的假设：它比S中另一假设更一般
- 如果d是一个反例
  - 从S中移去所有d不一致的假设
  - 对G中每个与d不一致的假设g
    - 从G中移去g
  - 把g的所有的极小特殊化式h加入到G中，其中h满足
    - h与d一致，而且S的某个成员比h更特殊
  - 从G中移去所有这样的假设：它比G中另一假设更特殊

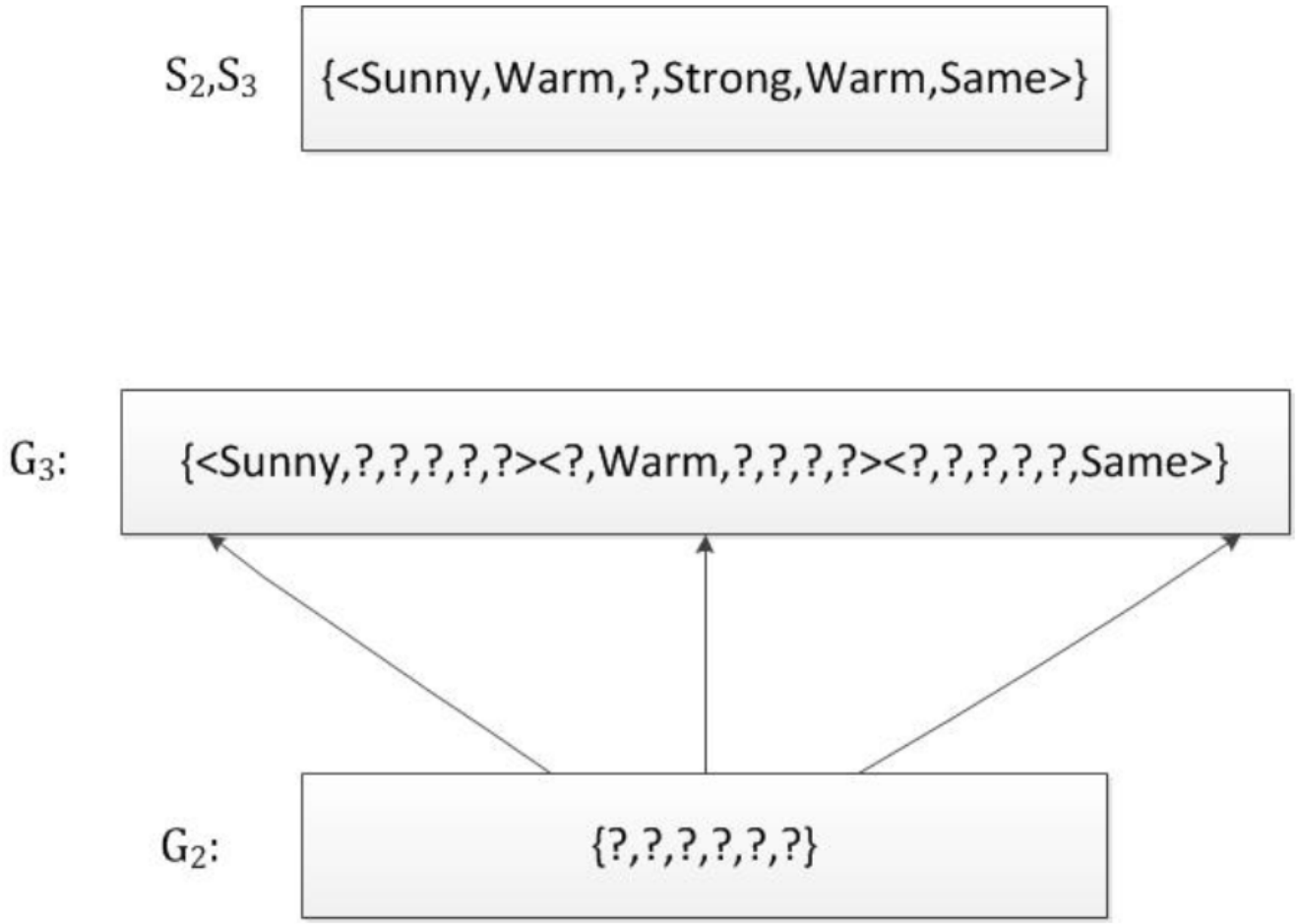


- 分别看是否满足S，G

训练样例：

1.<Sunny,Warm,Normal,Strong,Warm,Same>,EnjoySport=Yes

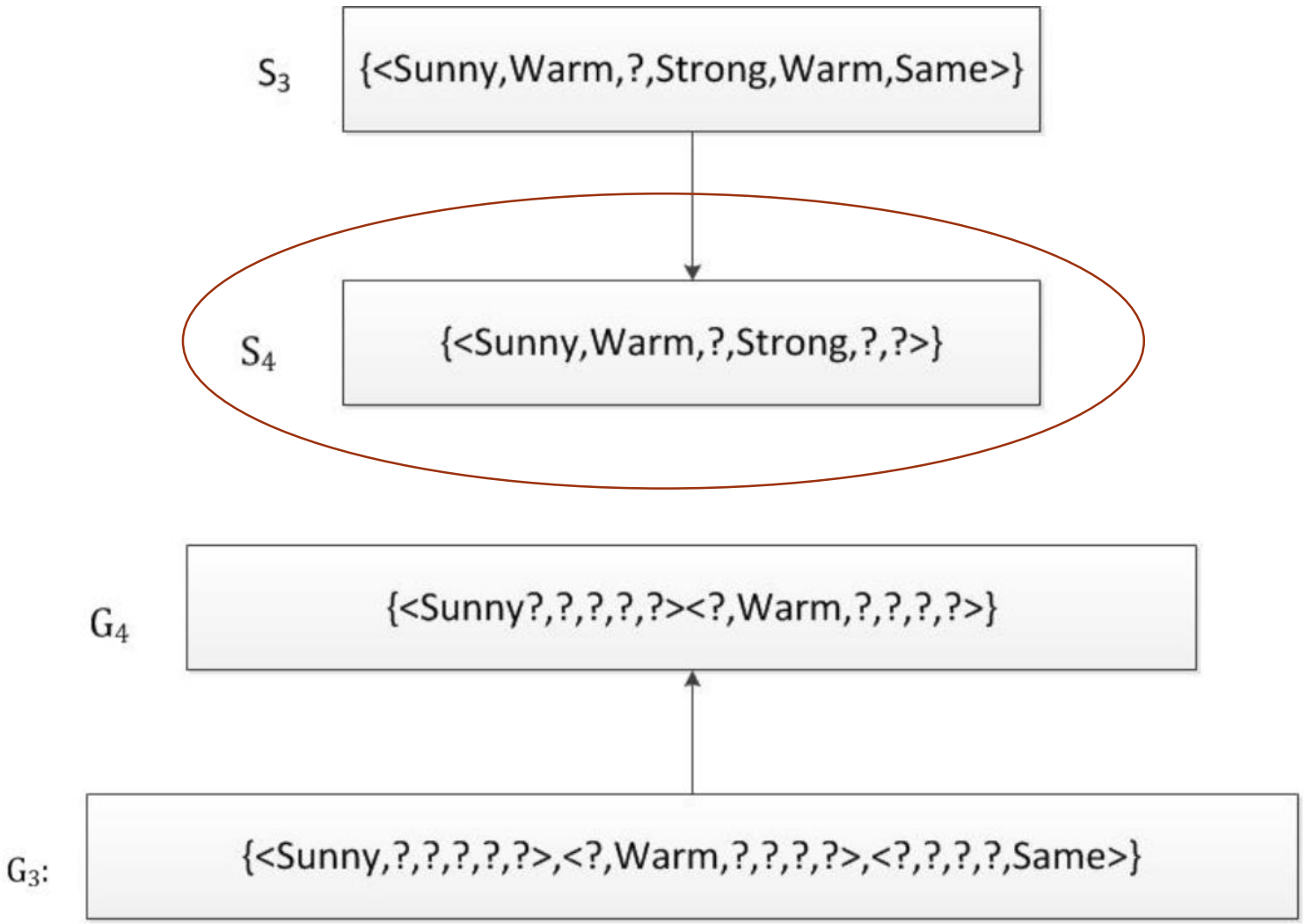
2.<Sunny,Warm,High,Strong,Warm,Same>,EnjoySport=Yes



- 分别看是否满足S，G
- G3中6选3
- G3中有多个可选的极大一般假设

训练样例:

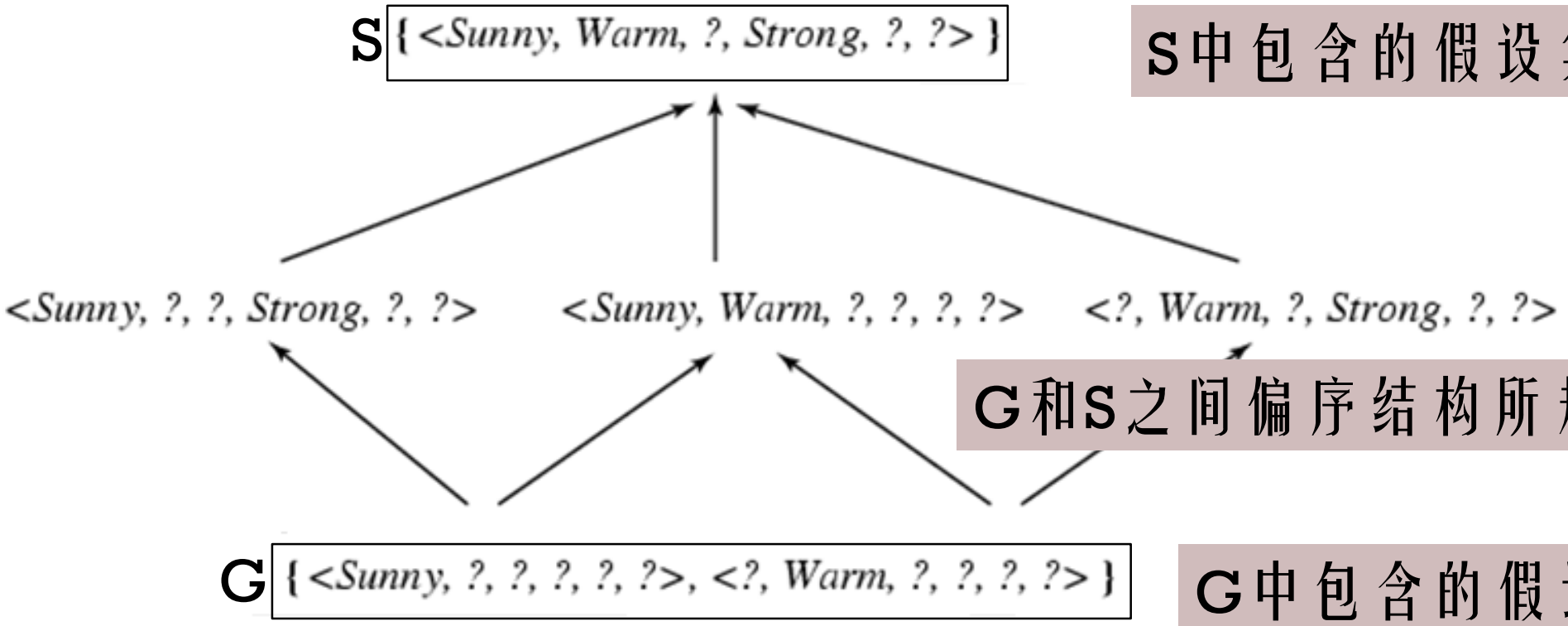
3.<Rainy,Cold,High,Strong,Warm,Change>,EnjoySport=No



训练样例：

4.<Sunny,Warm,High,Storage,Cool,Change>,EnjoySport=Yes

- 分别看是否满足S，G
- 红框为最终结果



S中包含的假设集

G和S之间偏序结构所规定的假设

G中包含的假设集

- 变型空间表示定理
- 令X为一任意的实例集合，H为X上定义的布尔假设的集合。令c: X→{0,1}为X上定义的任一目标概念，并令D为任一训练样例集合{<x,c(x)>}。对所有的X, H, c, D以及定义好的S和G，变型空间表示如下：

$$VS_{H,D}=\{h\in H\mid(\exists s\in S)(\exists g\in G)(g\geq_g h\geq_g s)\}$$

  - 上述定理表明：
    - 变型空间由G、S、G和S之间的偏序结构所规定的假设h组成

- 候选消除算法讨论
  - 候选消除算法输出与训练样例一致的所有假设的集合
  - 候选消除算法在描述这一集合时不需要明确列举所有成员
  - 利用more_general_than偏序结构，可以得到一个一致假设集合的简洁表示
  - 候选消除算法的缺点：同Find-S一样，容错性能差

- 候选消除算法收敛到正确目标概念的条件
  - 训练样例中没有错误
  - H中确实包含描述目标概念的正确假设

- 一个有偏的假设空间
  - 在EnjoySport这个例子中，假设空间限制为只包含属性值的合取（同时发生，交集）。（有偏）
  - 这一限制，导致假设空间不能够表示最简单的析取（发生一件，并集）形式的目标概念。

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

“Sky=Sunny或Sky=Cloudy”

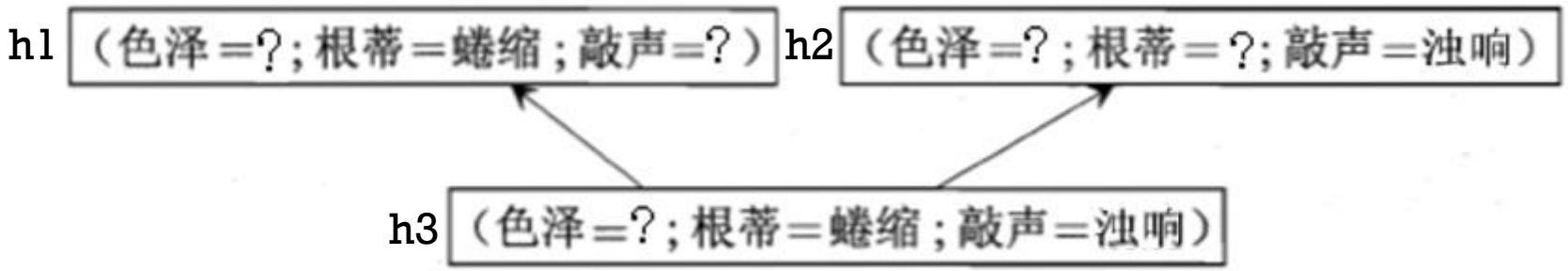
- 归纳推理的一个重要的基本属性：
  - 学习器如果不对目标概念的形式做预先的假定，那么它从根本上无法对未见的样本（实例）进行分类。
- 归纳偏置（**Inductive Bias**）：
  - 对归纳学习进行的某种形式的预先假定

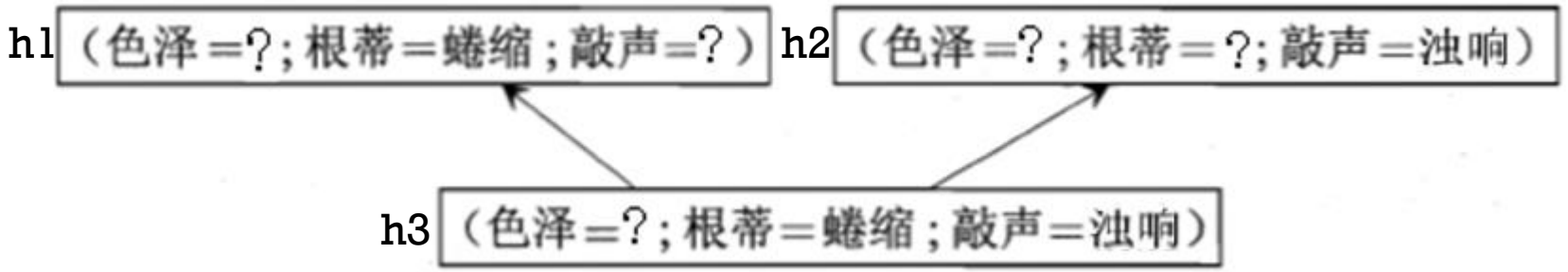
- 归纳偏好：
  - 假如通过不同学习算法得到三个与训练集一致的假设，但是他们对应的模型在遇到相同的问题时，会产生不同的预测结果。那么，应该选择哪种模型？我们无法通过训练模型得知哪个模型“更好”。这时，学习算法本身的“偏好”就会起到决定性作用。机器学习算法在学习过程中对某种类型假设的偏好，称为：“归纳偏好”。

- 西瓜数据集：学习概念“好瓜”

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

- 变型空间：

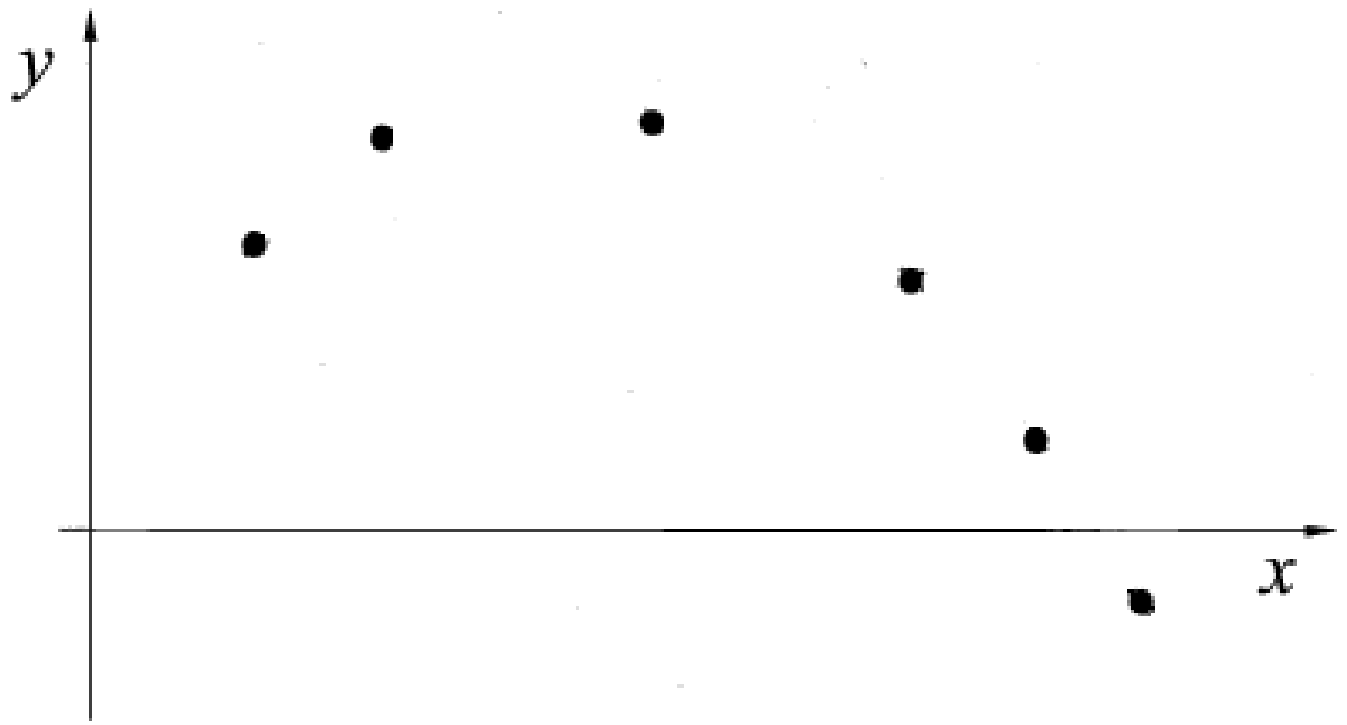


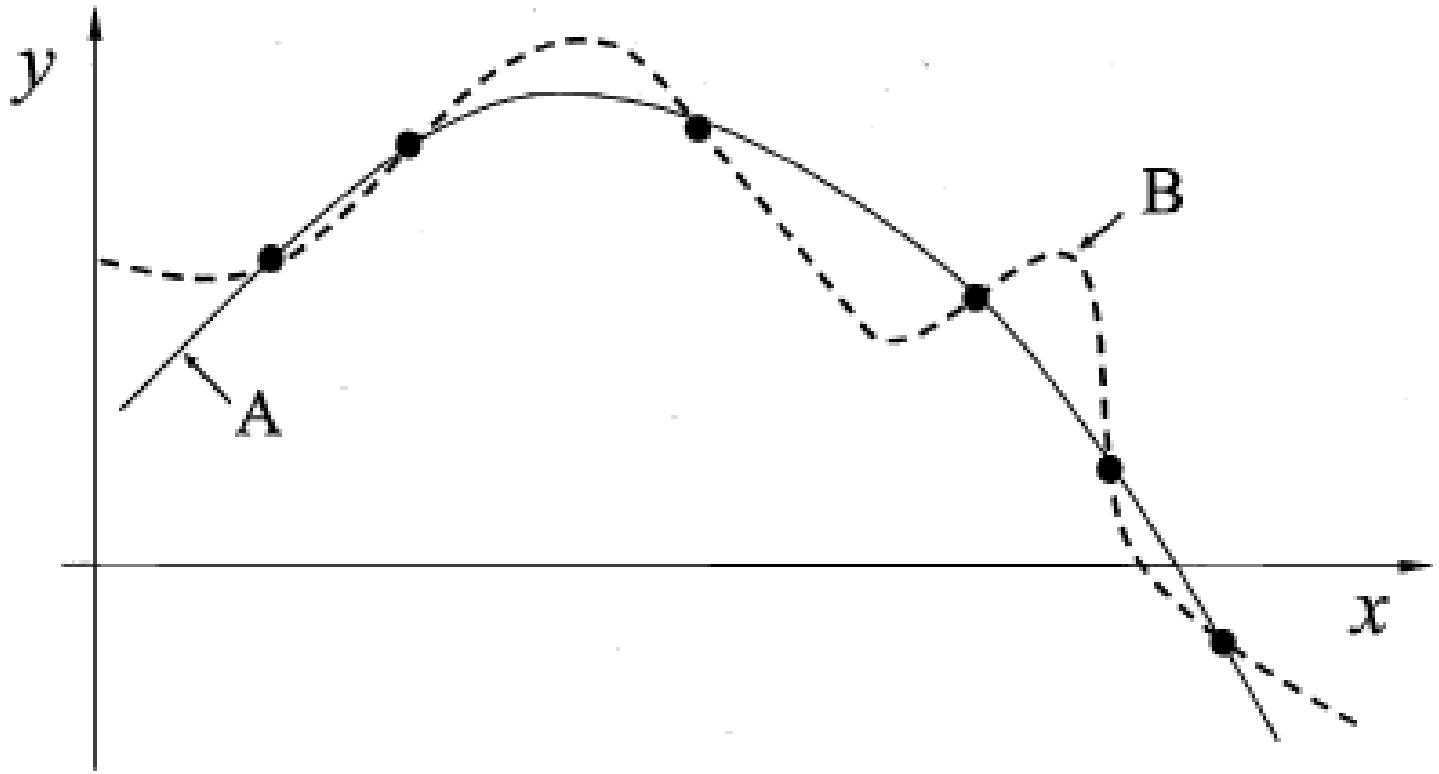


- 新瓜（好瓜）：色泽=青绿；根蒂=蜷缩；敲声=清脆
  - 算法偏好尽可能特殊的模型：否
  - 算法偏好尽可能一般的模型，且由于某种原因它更“相信”根蒂：是
  - 算法偏好多数表决：否

- 归纳偏好常表现为正则化项
- 可依据“奥卡姆剃刀”原理进行选择：若有多个假设与观察一致，则选择最简单的那个

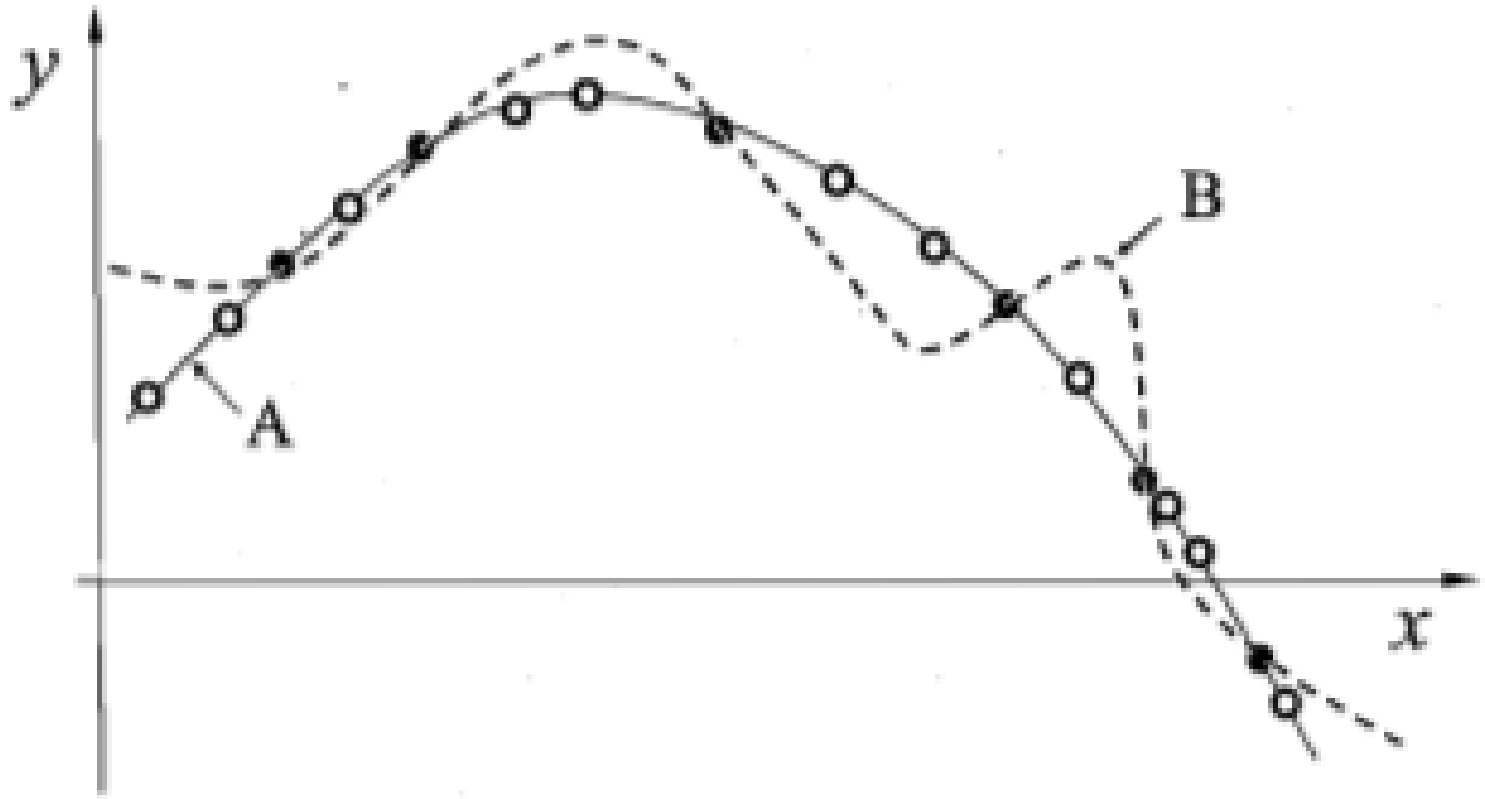
- 考察一个回归学习



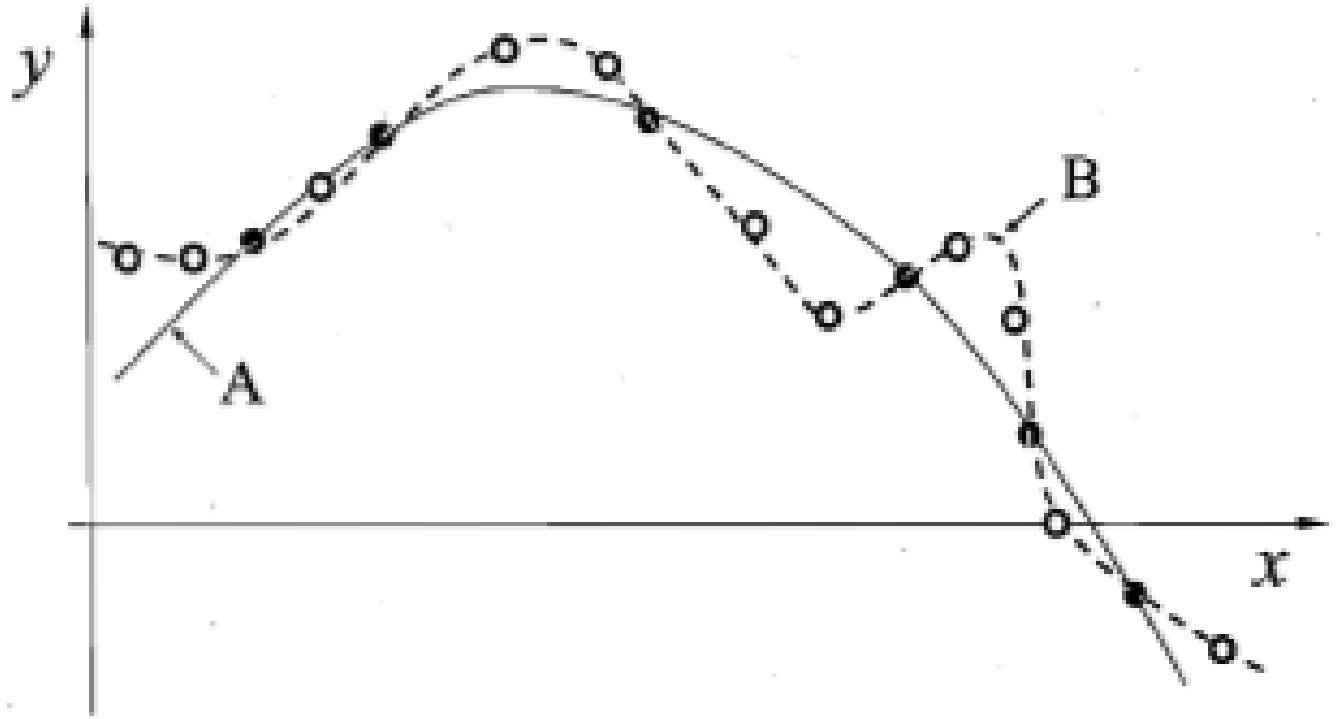


存在多条曲线与有限样本训练集一致

- 通过学习算法得到A、B两条拟合曲线



- 根据“奥卡姆剃刀”原理，A优于B



- 但B优于A的情况也是完全可能存在的

对于一个学习算法fa，若它在某些问题上比学习算法fb好，则必然存在在另一些问题上，fb比fa好。这个结论对任何算法均成立。“没有免费的午餐”定理证实，无论学习算法fa多聪明、学习算法fb多笨拙，它们的期望性能竟然相同（训练集外误差）。

为简单起见，假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 都是离散的. 令 $P(h|X,\mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 X 产生假设 h 的概率，再令 f 代表我们希望学习的真实目标函数. $\mathfrak{L}_a$ 的“训练集外误差”，即 $\mathfrak{L}_a$ 在训练集之外的所有样本上的误差为

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

其中 $\mathbb{I}(\cdot)$ 是指示函数，若 $\cdot$ 为真则取值 1，否则取值 0.

若 f 均匀分布，则有一半的 f 对 $\boldsymbol{x}$ 的预测与 $h(\boldsymbol{x})$ 不一致.

对所有可能的 f 按均匀分布对误差求和，有

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

总误差竟然与学习算法无关!

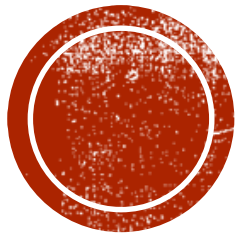
对于任意两个学习算法 $\mathfrak{L}_a$ 和 $\mathfrak{L}_b$,

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

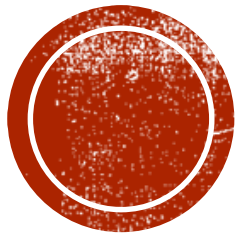
这就是“没有免费的午餐”定理 (No Free Lunch Theorem, 简称 NFL定理) [Wolpert, 1996; Wolpert and Macready, 1995].

- 优化算法的等价性
- 任何优化算法都不比穷举法好
- 为什么还要研究最优化和机器学习算法呢?

- NFL定理有一个重要前提：所有“问题”出现的机会相同、或所有问题同等重要。但实际情形并不是这样的。很多时候，我们只关注自己正在试图解决的问题，希望为它找到一个解决方案，至于这个解决方案在别的问题上是否为好方案，我们并不关心。
- 脱离具体问题，空泛的谈论“什么学习算法更好”毫无意义
- 收敛速度



THE END!



模型评估与选择

Model Assessment and Selection

损失函数（LOSS FUCTION）

（1）0-1 损失函数（0-1 loss function）

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

（2）平方损失函数（quadratic loss function）

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

（3）绝对损失函数（absolute loss function）

$$L(Y, f(X)) = |Y - f(X)|$$

（4）对数损失函数（logarithmic loss function）或对数似然损失函数（log-likelihood loss function）

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

风险函数（RISK FUNCTION）

- 理论上f(X)基于联合分布P(X,Y)的平均意义下的损失，即期望损失（Expected Loss）

$$R_{\exp}(f) = E_P[L(Y, f(X))] = \int_{x \times y} L(y, f(x)) P(x, y) \mathrm{d}x\mathrm{d}y$$

未知

经验风险（EMPIRICAL RISK）

- f(x)关于训练数据集T=$\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$的平均损失

$$R_{emp}(f)=\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i))$$

模型选择策略

- 经验风险最小化（ Empirical Risk Minimization, ERM ） 求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))$$

- 结构风险最小化（ Structural Risk Minimization, SRM ） 求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f)$$

模型性能度量方法

- 错误率（error rate）和准确度（accuracy）
  - 错误率：分类问题中，误分类样本的比例
  - 准确度：正确分类样本的比例，即正确率
  - 对样本集D

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

$$\begin{aligned}\mathrm{acc}(f;D) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)=y_i)\\ &= 1-E(f;D)\ .\end{aligned}$$

$$E(f;\mathcal{D}) = \int_{\boldsymbol{x}\sim\mathcal{D}} \mathbb{I}\left(f\left(\boldsymbol{x}\right) \neq y\right) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \ ,$$

$$\begin{aligned} \mathrm{acc}(f;\mathcal{D}) &= \int_{\boldsymbol{x}\sim\mathcal{D}} \mathbb{I}\left(f\left(\boldsymbol{x}\right) = y\right) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= 1 - E(f;\mathcal{D}) \ . \end{aligned}$$

模型：锯齿∩绿色→树叶

- 训练误差（training error）和泛化误差（generalization error）
  - 训练误差：关于训练数据集的平均损失，又称经验误差、再代入（resubstitution）误差或表现（apparent）误差

$$R_{emp}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

0-1损失

  - 泛化误差：对未知数据预测的误差

$$R_{exp}(\hat{f}) = E_P[L(Y, \hat{f}(X))] = \int_{x\times y} L(y, \hat{f}(x))P(x,y)\mathrm{d}x\mathrm{d}y$$

二分类问题的泛化误差上界

训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$ 从联合概率分布 $P(X,Y)$ 独立同分布产生的，$X\in\mathbf{R}^n$，$Y\in\{-1,+1\}$．设 f 是从 $\mathcal{F}=\{f_1,f_2,\cdots,f_d\}$ 中选取的函数．损失函数是 0-1 损失．

关于 f 的期望风险和经验风险分别是

$$R(f)=E[L(Y,f(X))]$$

$$\hat{R}(f)=\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i))$$

经验风险最小化函数是

$$f_N = \arg\min_{f \in \mathcal{F}} \hat{R}(f)$$

f_N 的泛化能力

$$R(f_N) = E[L(Y, f_N(X))]$$

定理 （泛化误差上界） 对二类分类问题，当假设空间是有限个函数的集合 $\mathcal{F}=\{f_1,f_2,\cdots,f_d\}$ 时，对任意一个函数 $f\in\mathcal{F}$，至少以概率 $1-\delta$，以下不等式成立：

$$R(f)\leqslant\hat{R}(f)+\varepsilon(d,N,\delta)$$

$\mathcal{F}$中包含的函数越多，泛化误差上界越大。

训练误差小的模型泛化误差也会小。

样本容量N越大，训练误差与泛化误差越接近。

其中，

$$\varepsilon(d,N,\delta)=\sqrt{\frac{1}{2N}\left(\log d+\log\frac{1}{\delta}\right)}$$

Hoeffding 不等式

设 $S_n=\sum_{i=1}^{n}X_i$ 是独立随机变量 $X_1,X_2,\cdots,X_n$ 之和，$X_i\in[a_i,b_i]$，则对任意 $t>0$，以下不等式成立：

$$P(ES_n-S_n\geqslant t)\leqslant\exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right)\qquad(1)$$

- 泛化误差的估计
  - 用训练误差（training error）估计
  - 其他方法?
- 测试集（testing set）与测试误差（testing error）
  - 假定从真实分布中独立采样得到
  - 与训练集互斥

ID	X1	X2	Y
1	A	7	T
2	A	7	T
3	B	5	T
4	A	3	F
5	A	2	F
6	A	6	F
7	A	7	F
8	A	7	T
9	B	2	T
10	A	3	F

ID	X1	X2	Y
1	A	7	T
2	A	7	T
4	A	3	F
6	A	6	F
9	B	2	T
10	A	3	F

IF X1 = B OR X2> 6
THEN Y = T

ID	X1	X2	Y
3	B	5	T
5	A	2	F
7	A	7	F
8	A	7	T

ID	X1	X2	Y
3	B	5	T
5	A	2	F
7	A	7	T
8	A	7	T

测试误差：0.25

准确率的局限性

- 考虑一个分类问题：
  - 类0中的样本数为9990
  - 类1中的样本数为10
  - 若模型将所有样例类别预测为类0，则分类准确率为9990/10000 = 99.9 %
    - 模型没有发现任何类1的样例，因而准确率具有误导性。

混淆矩阵（CONFUSION MATRIX）

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- a: TP (true positive) 真正，对应于被分类模型正确预测的正样本数
- b: FN (false negative) 假负，对应于被分类模型错误预测为负类的正样本数
- c: FP (false positive) 假正，对应于被分类模型错误预测为正类的负样本数
- d: TN (true negative) 真负，对应于被分类模型正确预测的负样本数

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- 真正率（TPR）或称灵敏度（Sensitivity）
  - 模型正确预测的正样本的比例
  - TPR=TP/(TP+FN)= a/(a+b)
- 真负率（TNR）或称特指度（Specificity）
  - 模型正确预测的负样本的比例
  - TNR=TN/(TN+FP) = d/(c+d)
- 假正率（FPR）
  - 被预测为正类的负样本的比例
  - FPR=FP/(TN+FP) = c/(c+d)
- 假负率（FNR）
  - 被预测为负类的正样本的比例
  - FNR=FN/(TP+FN) = b/(a+b)

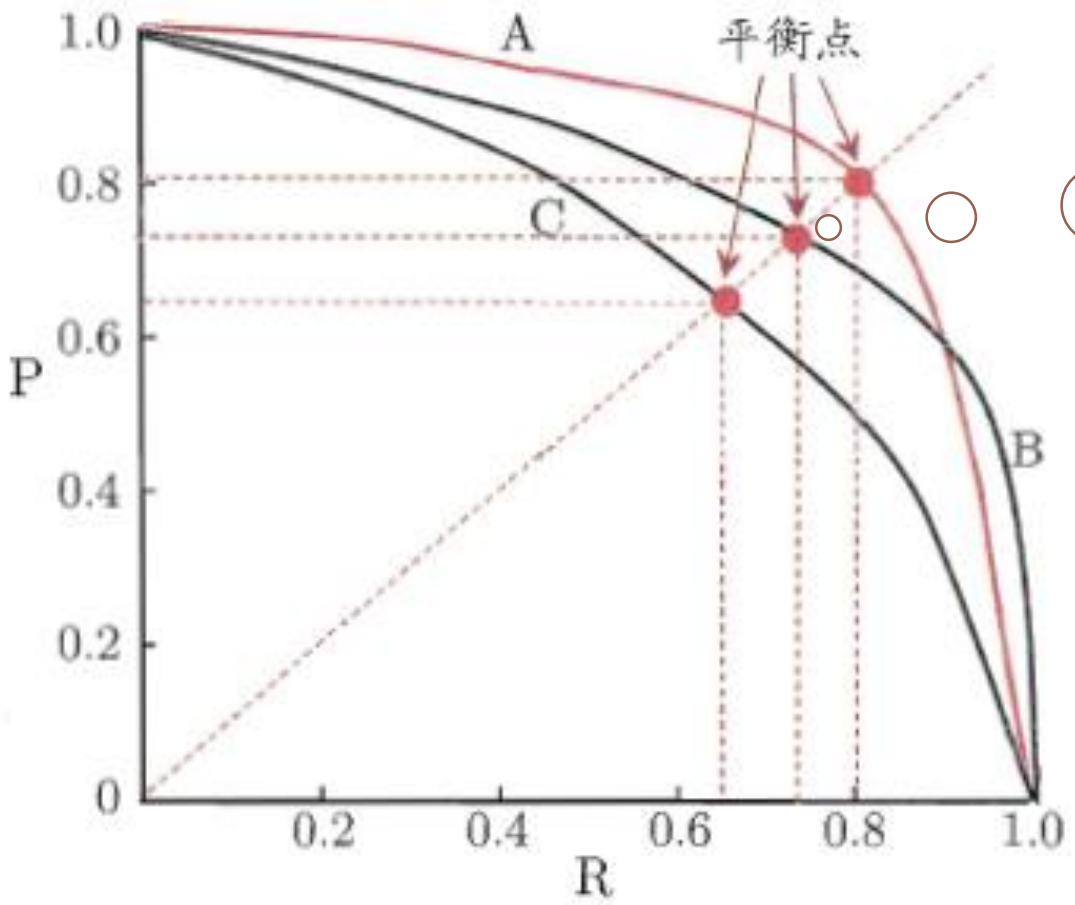
		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- 精度p: $\text{Precision (p)} = \frac{a}{a+c}$
  - 也称查准率，确定在分类器断言为正类的那部分记录中实际为正类的记录所占的比率。
  - 精度越高，分类器的假正错误率就越低
- 召回率r: $\text{Recall (r)} = \frac{a}{a+b}$
  - 度量被分类器正确预测的正样本的比例，亦称查全率。
  - 具有高召回率的分类器很少将正样本误分为负样本
- F_1度量: $F_1\text{-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$
  - 精度和召回率的调和平均

P-R曲线和P-R图

- 根据学习器的预测结果对样例进行排序
  - “最可能”是正例的样本排在最前面；
  - “最不可能”是正例的样本排在最后面；
- 按此顺序逐个将样本作为正例进行预测，每次计算p、r值
- 以p为纵轴，r为横轴作图，即得“P-R曲线”
- 显示该曲线的图称为“P-R图”

P-R曲线和P-R图



P-R曲线与平衡点示意图

学习器A优于学习器C

F_β度量考察召回率和精度之间的折中：

$$F_\beta = \frac{(\beta^2+1)rp}{r+\beta^2 p} = \frac{(\beta^2+1)\times TP}{(\beta^2+1)TP+\beta^2 FP+FN}$$

加权准确率： Weighted Accuracy $= \dfrac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$

加权准确率和其他的性能度量值之间的关系

度量	w_1	w_2	w_3	w_4
召回率	1	1	0	0
精度	1	0	1	0
F_β值	β^2+1	β^2	1	0
准确率	1	1	1	1

多个二分类混淆矩阵

- 各混淆矩阵上度量的平均值

$$\text{macro-}P = \frac{1}{n}\sum_{i=1}^{n} P_i \ ,$$

$$\text{macro-}R = \frac{1}{n}\sum_{i=1}^{n} R_i \ ,$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R} \ .$$

- 将混淆矩阵对应元素平均后再求指标值

$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}},$$

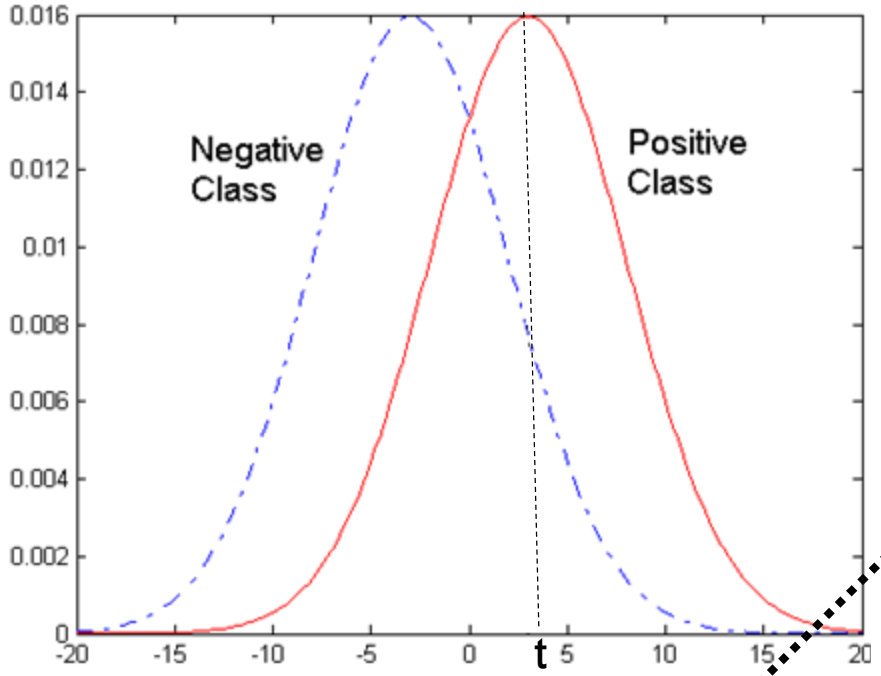
$$\text{micro-}R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}},$$

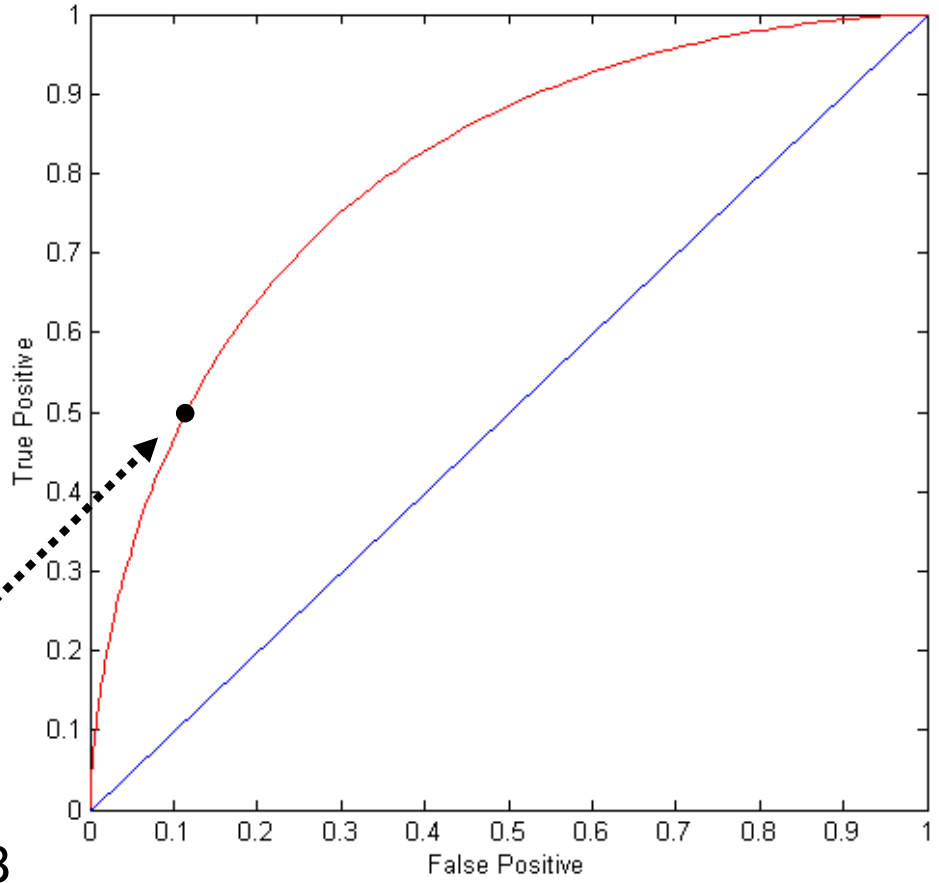
$$\text{micro-}F1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R}.$$

接受者操作特征（RECEIVER OPERATING CHARACTERISTIC，ROC）曲线

- 显示分类器真正率和假正率之间折中的一种图形化方法
- 真正率（TPR）沿y轴绘制（召回率）
- 假正率（FPR）沿x轴绘制
- 沿着曲线的每个点对应于一个分类器归纳的模型

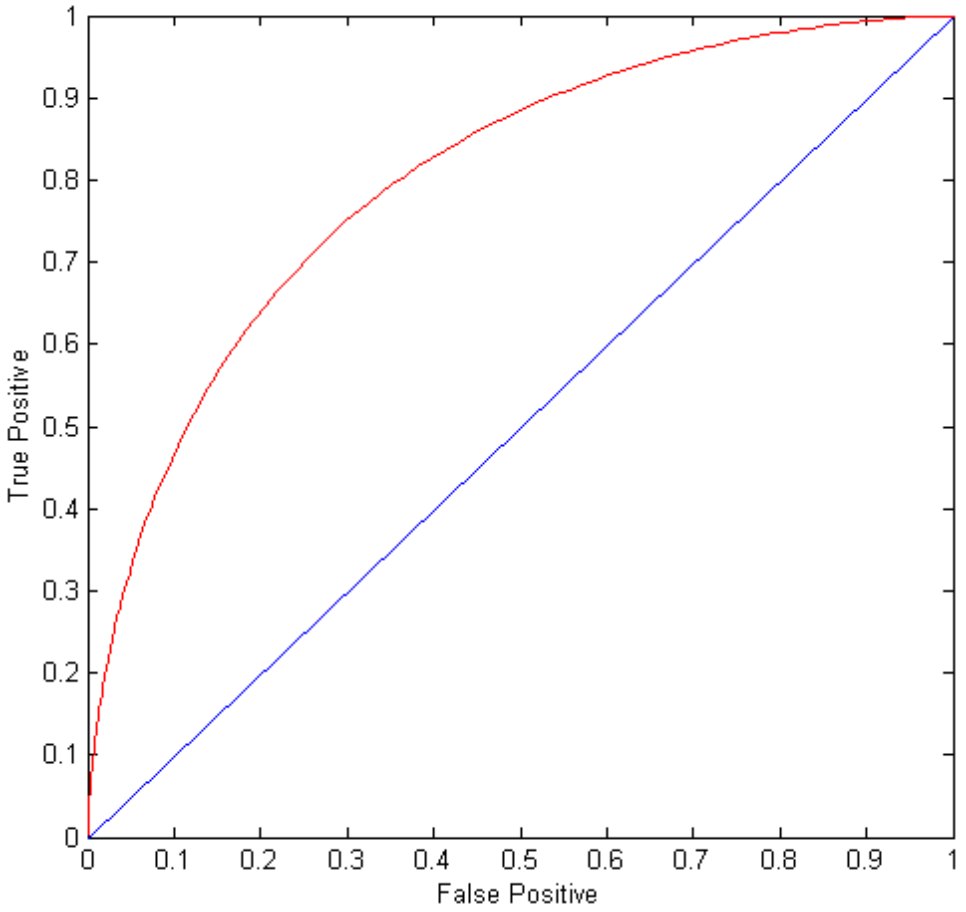
- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at x > t is classified as positive





At threshold t:

TP=0.5, FN=0.5, FP=0.12, TN=0.88



- ROC曲线
  - (TPR=0, FPR=0): 把每个实例都预测为负类的模型
  - (TPR=1, FPR=1): 把每个实例都预测为正类的模型
  - (TPR=1, FPR=0): 理想模型
  - 对角线：随机猜测的模型
  - 好的模型应尽可能靠近图的左上角

如何绘制ROC曲线

(1) 假定为正类定义了连续值输出，对检验记录按它们的输出值递增排序。

(2) 选择秩最低的检验记录（即输出值最低的记录），把选择的记录以及那些秩高于它的记录指派为正类。这种方法等价于把所有的检验实例都分为正类。因为所有的正检验实例都被正确分类，而所有的负测试实例都被误分，因此 $TPR=FPR=1$。

(3) 从排序列表中选择下一个检验记录，把选择的记录以及那些秩高于它的记录指派为正类，而把那些秩低于它的记录指派为负类。通过考察前面选择的记录的实际类标号来更新 *TP* 和 *FP* 计数。如果前面选择的记录为正类，则 *TP* 计数减少而 *FP* 计数不变。如果前面选择的记录为负类，则 *FP* 计数减少而 *TP* 计数不变。

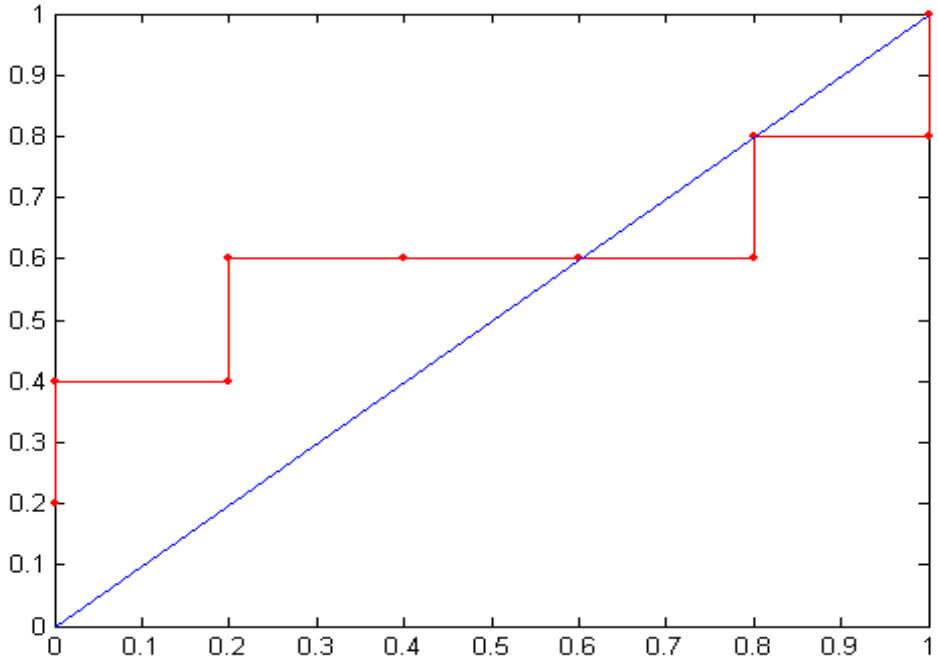
(4) 重复步骤 3 并相应地更新 *TP* 和 *FP* 计数，直到最高秩的记录被选择。

(5) 根据分类器的 *FPR* 画出 *TPR* 曲线。

- 对十个样本用不同阈值进行分类，5正5负

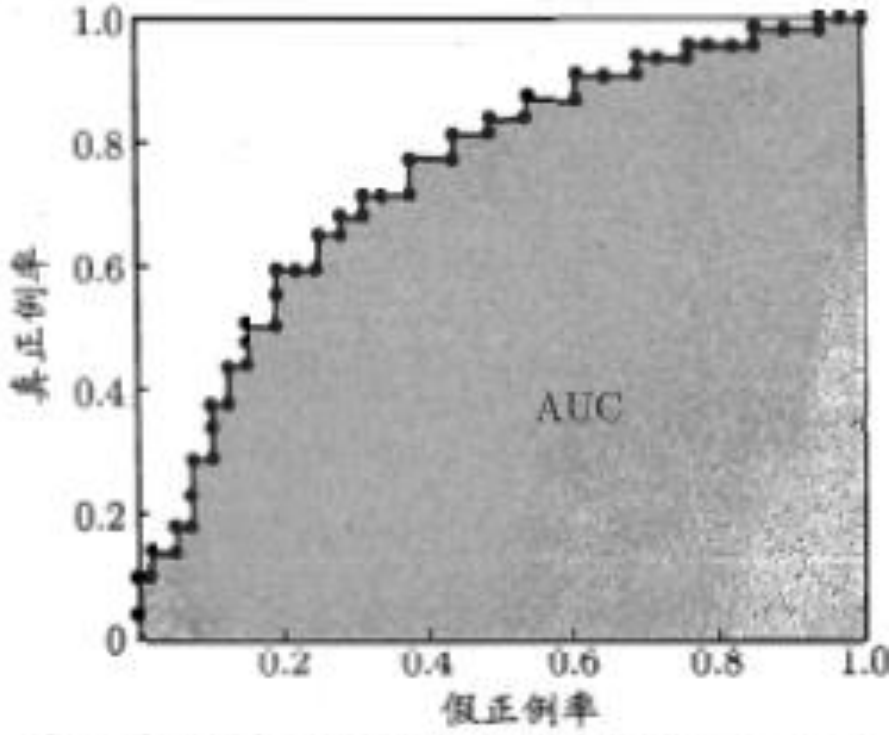
Class	+	-	+	-	-	-	+	-	+	+	
阈值	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



AUC（AREA UNDER ROC CURVE）

- 即ROC曲线下的面积
- 可以评价模型的性能

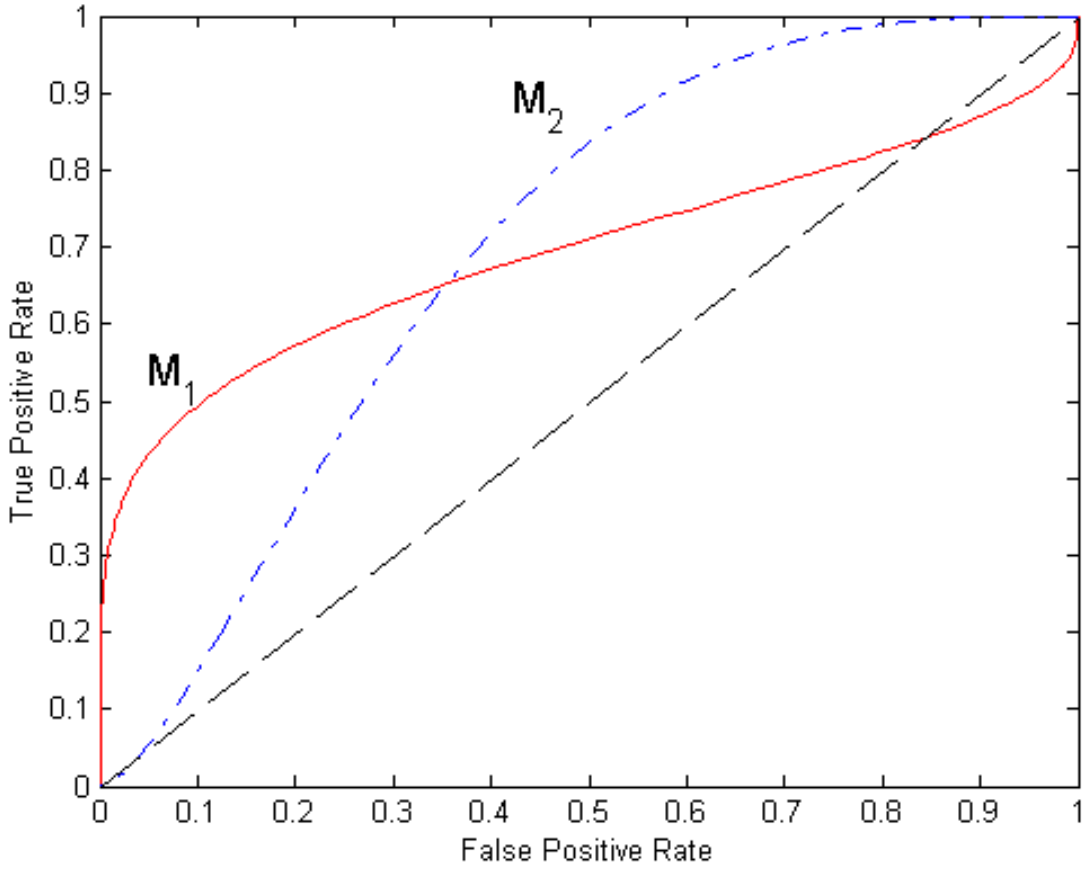


基于有限样例绘制的 ROC 曲线与 AUC

假定 ROC 曲线是由坐标为 $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ 的点按序连接而形成$(x_1 = 0, x_m = 1)$

$$\mathrm{AUC} = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \cdot (y_i + y_{i+1}) \ .$$

Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- AUC

PR曲线与ROC曲线

- 例：计算下述二分类器的TPR，FPR，P，R

M1	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	90	10
	Class=No	10	1,999,890

M2	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	90	10
	Class=No	910	1,998,990

M1: TPR=0.9,
FPR=0.00000500025,
P=0.9
R=0.9

M2: TPR=0.9,
FPR=0.00045502275,
P=0.09
R=0.9

代价矩阵

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j):预测（误分类）一个j类记录为i类的代价（医院诊断）

计算分类代价

犯假负错误的代价是犯假正错误的100倍

尽管模型M_2改善了准确率，但仍然较差。因这些改善是建立在增加代价更高的假负错误之上的。

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

标准的准确率度量趋向于M_2优于M_1

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	220	30
	-	70	180

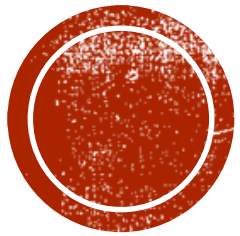
Accuracy = 80%

Cost = 2850

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	210	40
	-	10	240

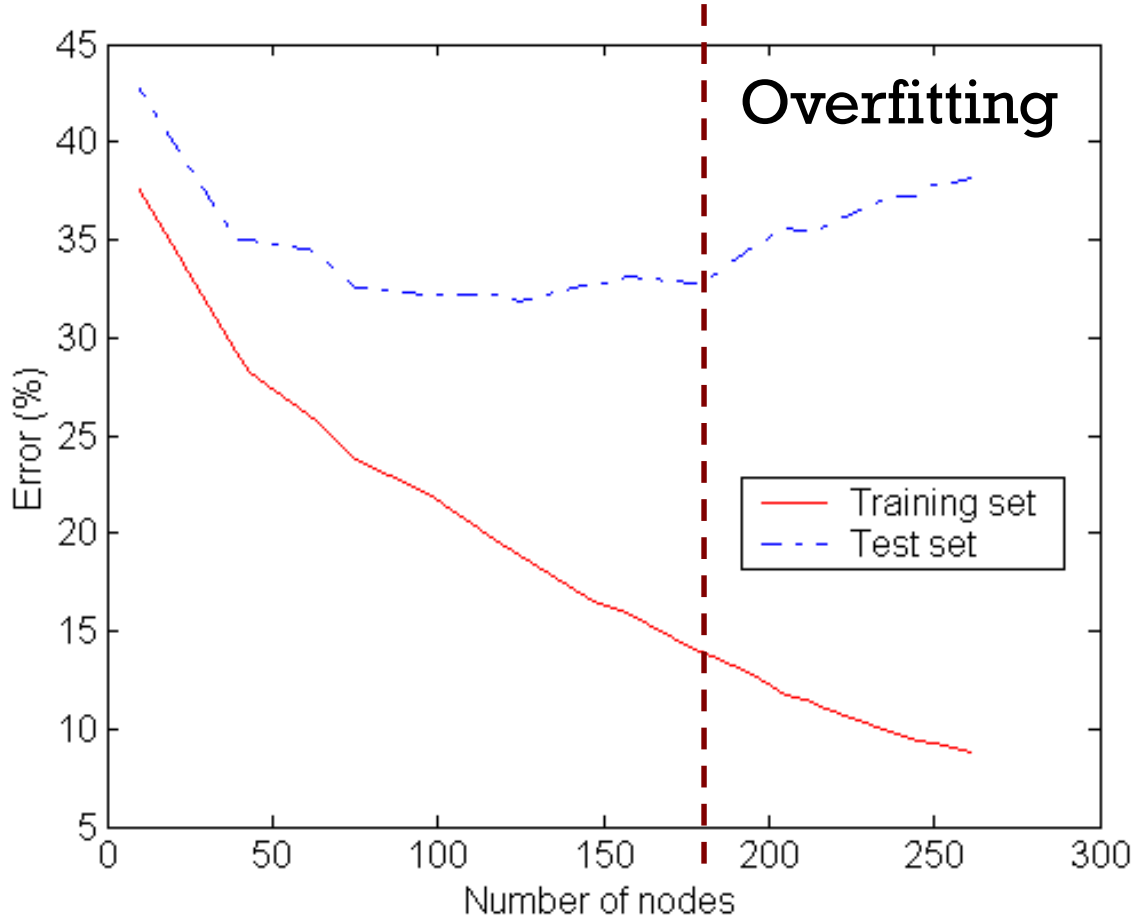
Accuracy = 90%

Cost = 3800



THE END!

过拟合与欠拟合



Underfitting: when model is too simple, both training and test errors are large

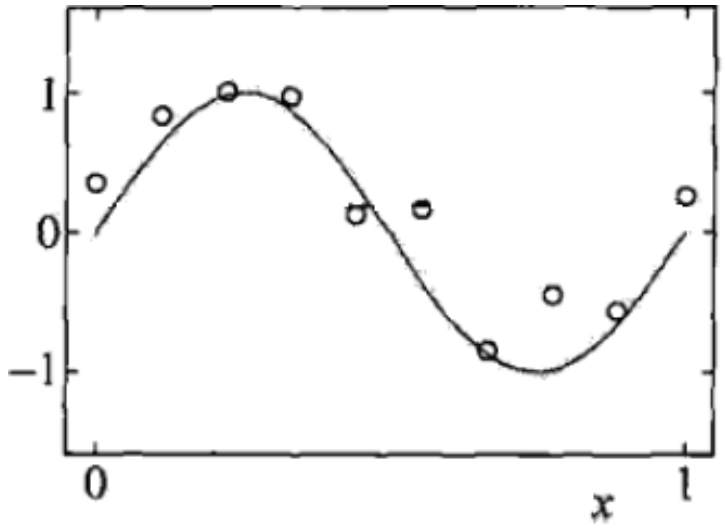
模 型M1：锯齿∩绿色→树叶

模 型M2：绿色→树叶

例 假设给定一个训练数据集：

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$$

在 M 次多项式函数中选择一个对已知数据以及未知数据都有很好预测能力的函数.



设 M 次多项式为

$$f_M(x,w)=w_0+w_1x+w_2x^2+\cdots+w_Mx^M=\sum_{j=0}^{M}w_jx^j$$

求以下经验风险最小化：

$$L(w)=\frac{1}{2}\sum_{i=1}^{N}(f(x_i,w)-y_i)^2=\frac{1}{2}\sum_{i=1}^{N}\left(\sum_{j=0}^{M}w_jx_i^j-y_i\right)^2$$

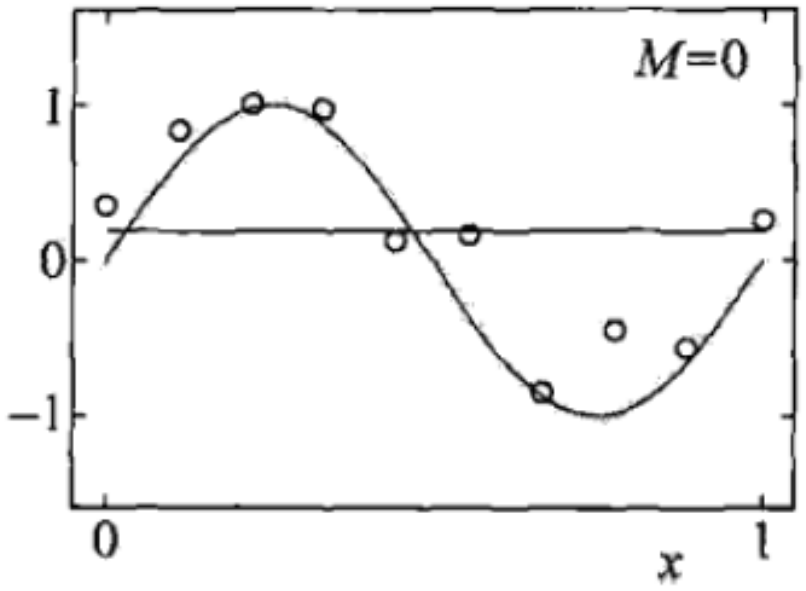
平方损失

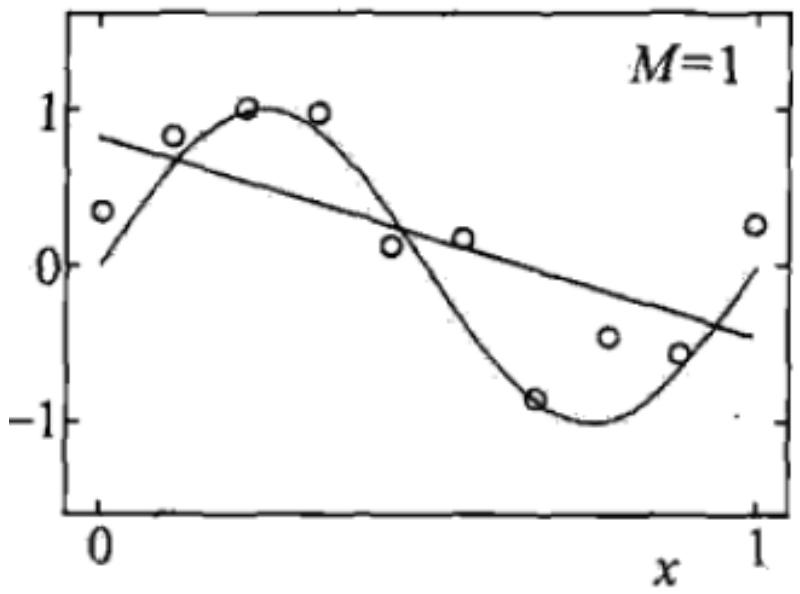
对 w_j 求偏导数并令其为 0，可得

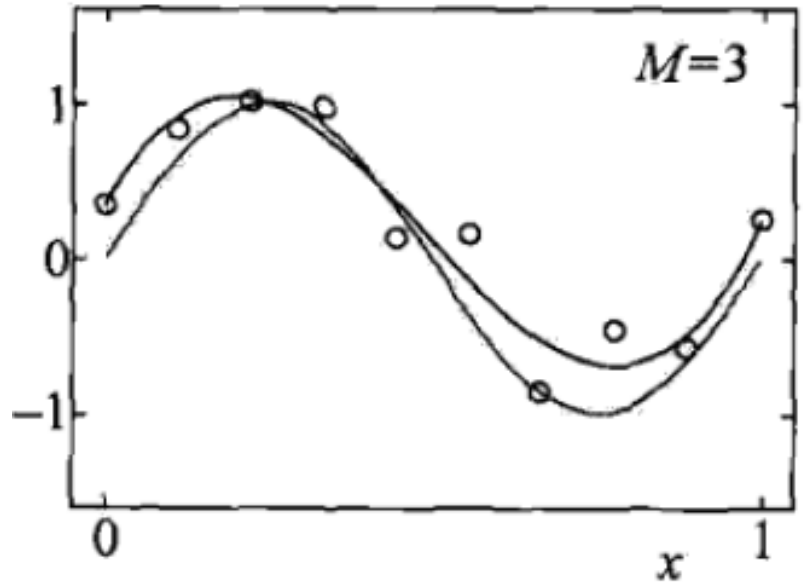
$$w_j = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^{j+1}}, \quad j = 0,1,2,\cdots,M$$

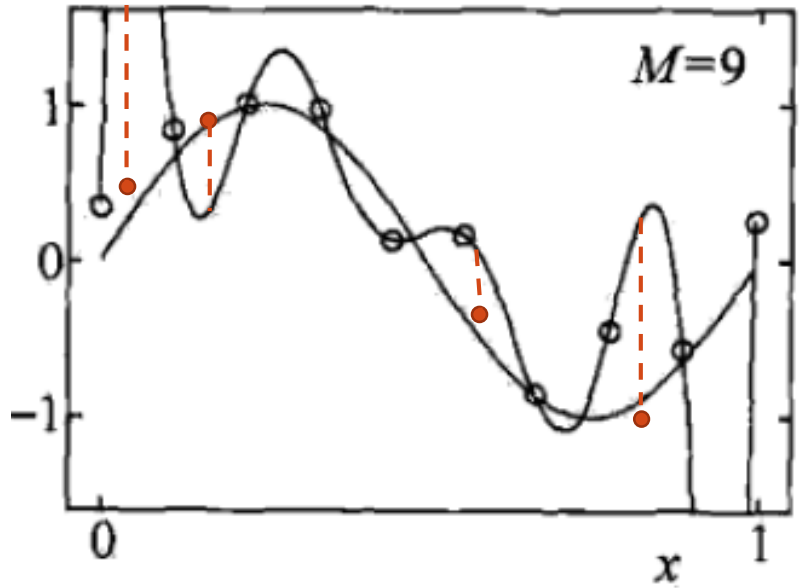
于是求得拟合多项式系数 $w_0^*, w_1^*, \cdots, w_M^*$.

$$f_M(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$









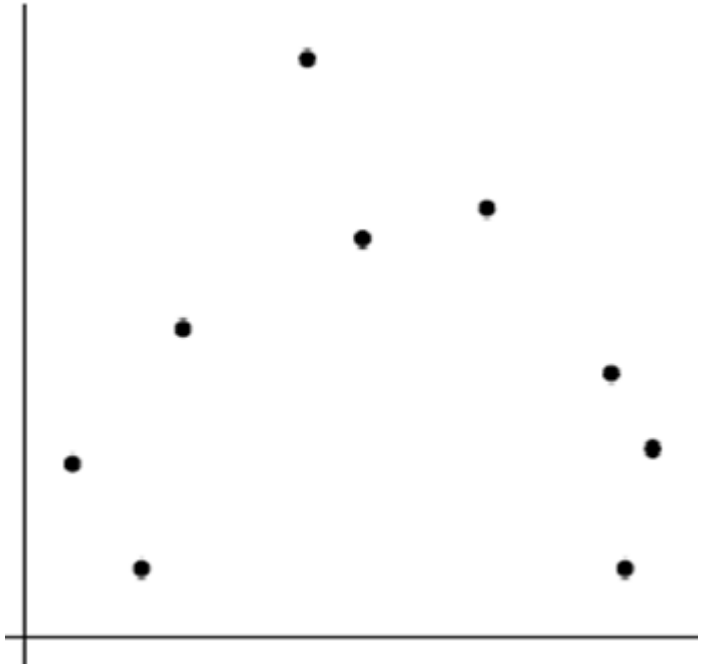
模型的检验方法——留出法（HOLDOUT）

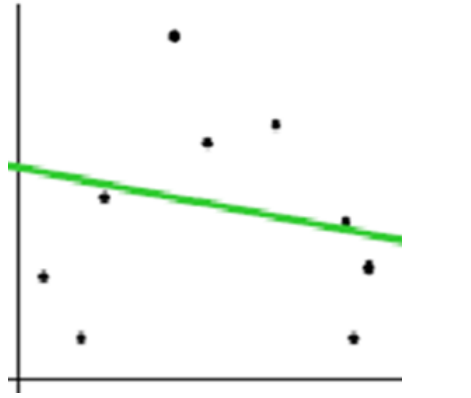
- 将原始数据集D划分为二不相交集合，一个作为训练集S，一个作为测试集T
- 在训练集上训练出模型
- 在测试集上评估模型的性能
- 二者的划分常依分析者的判断
  - 50-50
  - 2/3作为训练集，1/3作为检验集

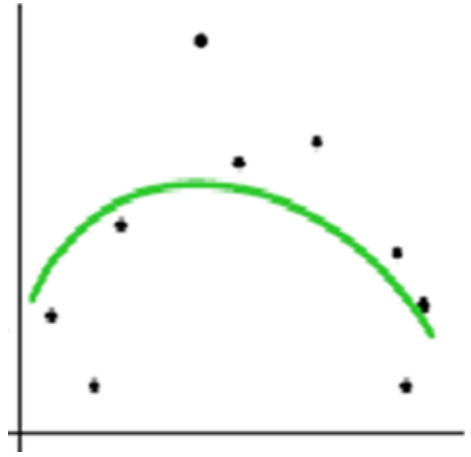
以二分类任务为例, 假定 D 包含 1000 个样本, 将其划分为 S 包含 700 个样本, T 包含 300 个样本, 用 S 进行训练后, 如果模型在 T 上有 90 个样本分类错误, 那么其错误率为 $(90/300) \times 100\% = 30\%$, 相应的, 精度为 $1 - 30\% = 70\%$.

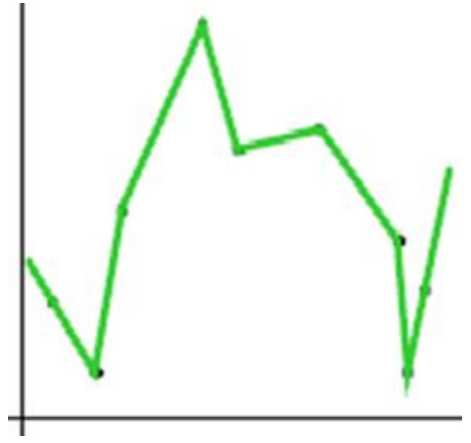
- 划分要尽可能保持原始数据分布的一致性
- 一般采用多次随机划分、重复评估后取平均值

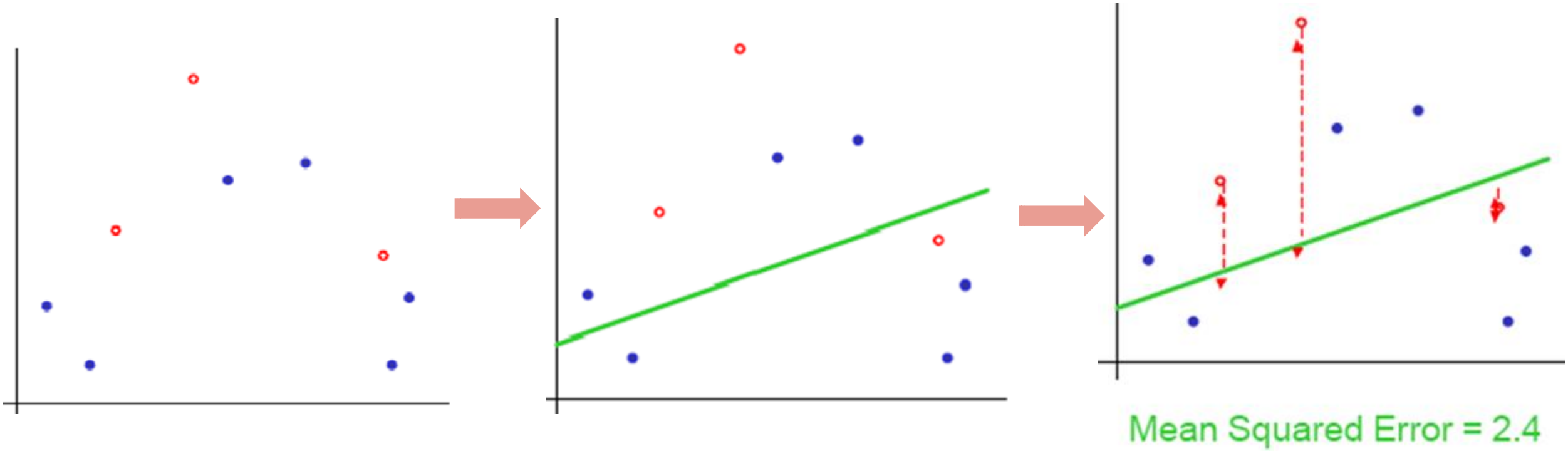
例：回归问题

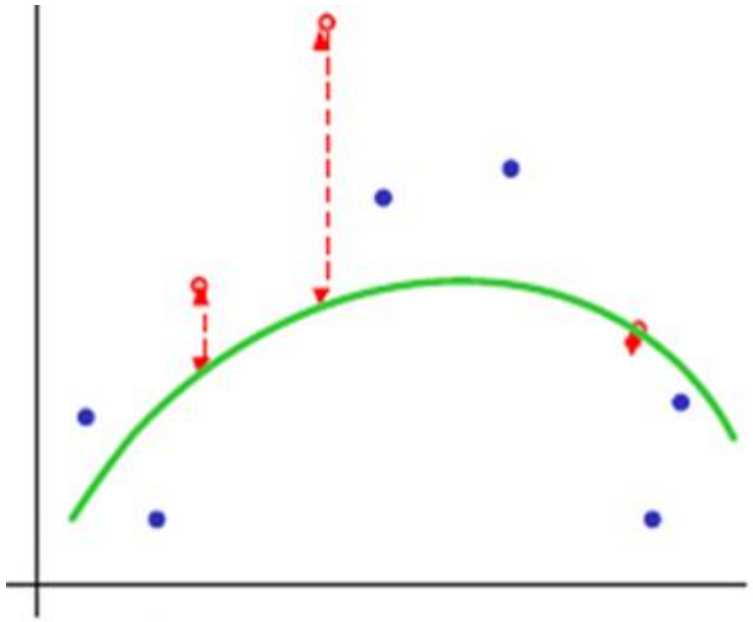




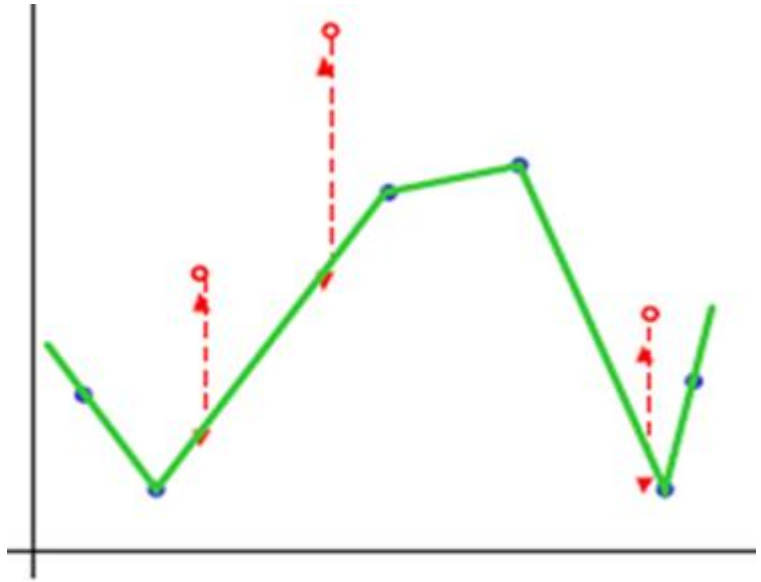





Mean Squared Error = 2.4

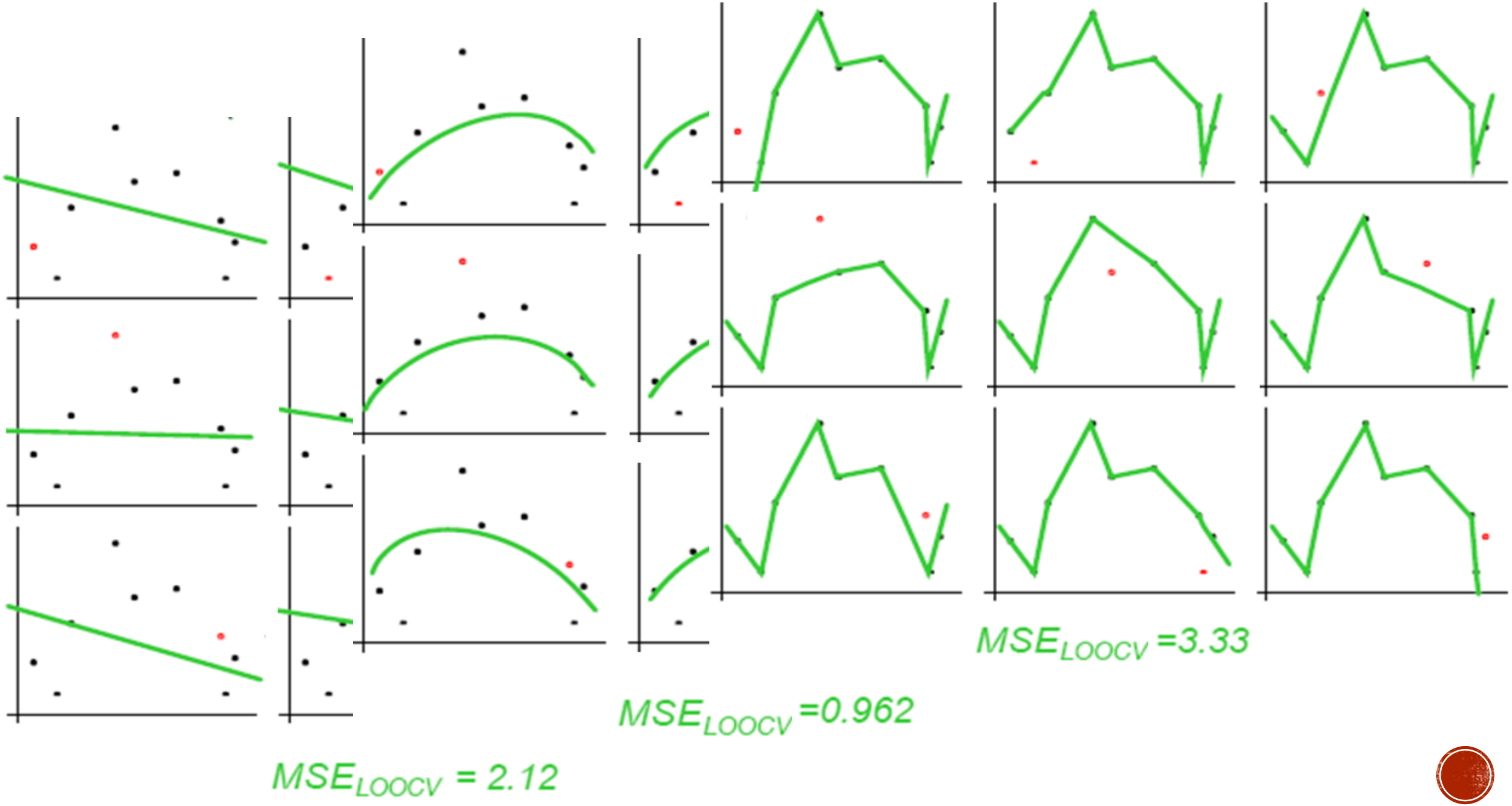


Mean Squared Error = 0.9

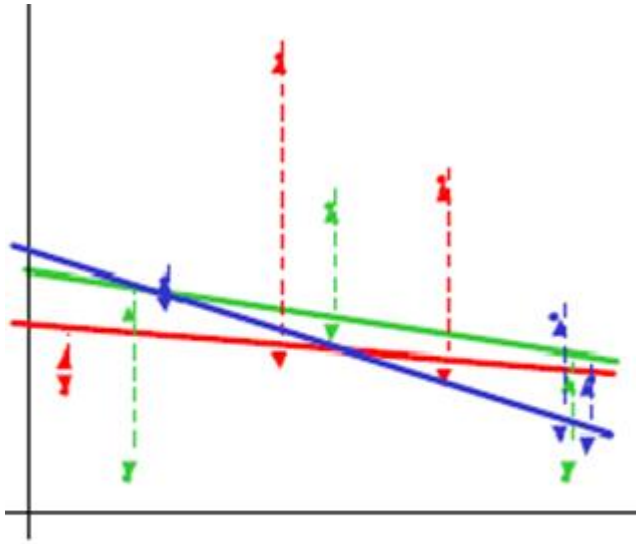


Mean Squared Error = 2.2

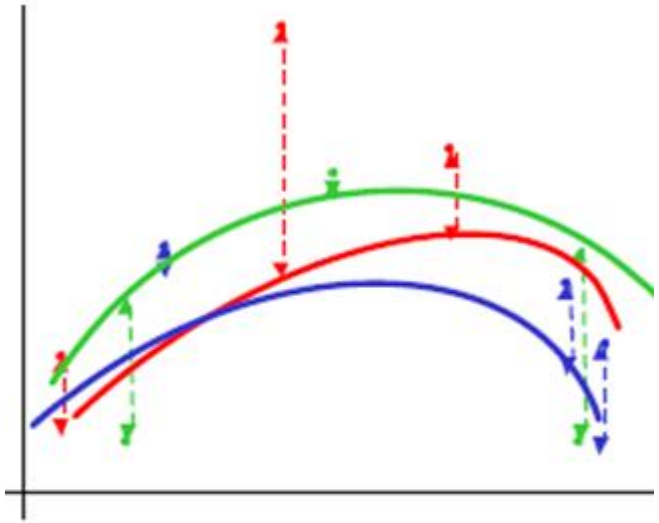
- 留一法（Leave-One-Out Cross Validation）
  - 记L为整个训练样本集的大小。
  - 对于第i个训练样本，将其取出，对剩下L-1个样本进行训练，得到模型，并用第i个训练样本对该模型的性能进行测试
  - 该过程重复L次
  - 用这L个模型的准确率的平均值作为评价模型性能的指标


MSE_{LOOCV} = 2.12
MSE_{LOOCV} =0.962
MSE_{LOOCV} =3.33

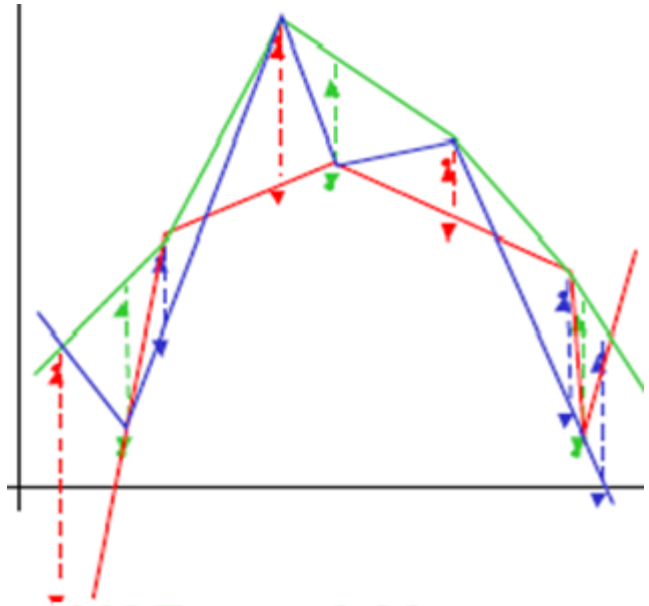
- K折交叉验证法（K-fold Cross Validation ）
  - 将训练样本集随机分为K个集合，通常分为K等份
  - 对其中的K-1个集合进行训练，得到模型，并用剩下的一个集合对该模型的性能进行测试。
  - 该过程重复K次，取K次过程中的测试错误的平均值作为评价模型性能的指标。



$MSE_{3FOLD}=2.05$



$MSE_{3FOLD}=1.11$



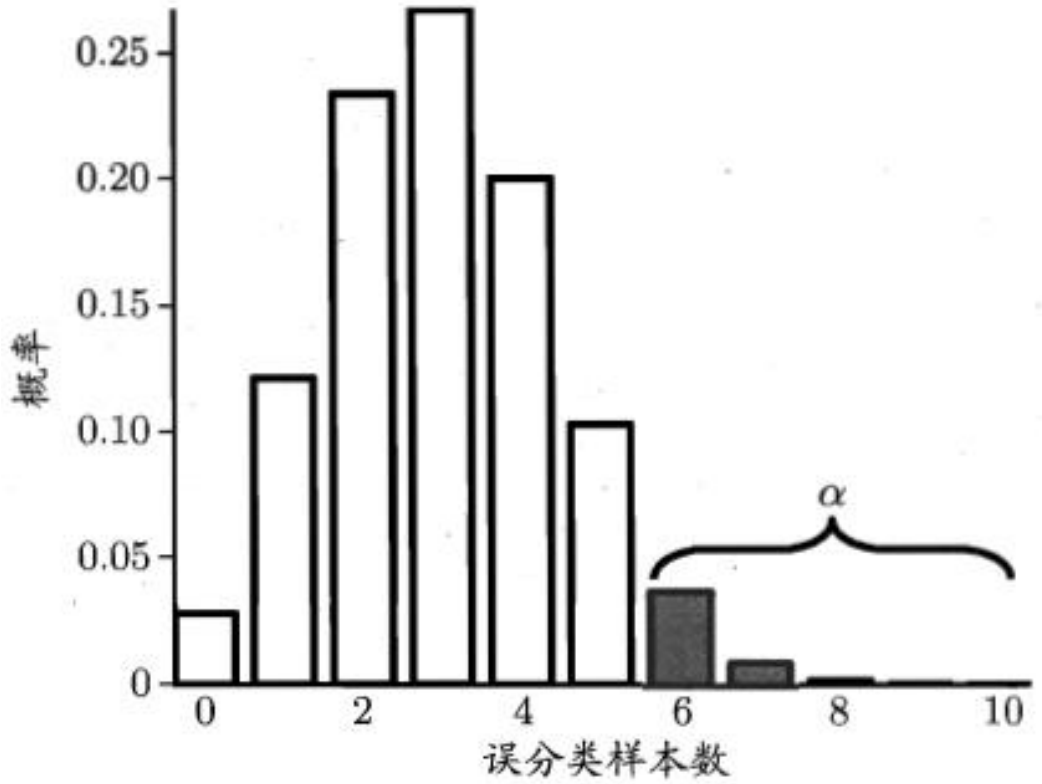
$MSE_{3FOLD}=2.93$

自助法（BOOTSTRAPPING）

- 以自助采样法为基础
  - 数据集D中包含m个样本
  - 每次随机从D中挑选一个样本到D’
  - 该样本被放回初始数据集D中
  - 重复m次得到包含m个样本的数据集D’

模 型 检 验

- H_0： $\varepsilon <= \varepsilon_0$
- 二 项 检 验



二项分布示意图($m = 10, \epsilon_0 = 0.3$)

模 型 检 验

- H_0：$\varepsilon <= \varepsilon_0$
- 近 似 正 态 检 验

$$\frac{\frac{x}{m} - \varepsilon_0}{\sqrt{\varepsilon_0 (1-\varepsilon_0)/m}} \sim AN(0,1)$$

模型检验

- H_0：$\varepsilon <= \varepsilon_0$
- t检验（交叉验证法）

k 个测试错误率，$\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_k$，

$$\mu = \frac{1}{k}\sum_{i=1}^{k}\hat{\epsilon}_i \ ,$$

$$\sigma^2 = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{\epsilon}_i - \mu)^2 \ .$$

$\tau_t = \dfrac{\sqrt{k}(\mu - \epsilon_0)}{\sigma}$ 服从自由度为 $k-1$ 的 t 分布

不同模型（学习器）性能的比较

- 考虑以下二个模型：
  - 模型M1: accuracy = 85%, tested on 30 instances
  - 模型M2: accuracy = 75%, tested on 5000 instances
- 模型M1是否优于模型M2?

- 准确率的置信区间
  - X：模型正确预测的记录数
  - p：模型真正的准确率
- 当N充分大时

$$P\left(\left|\frac{acc-p}{\sqrt{p(1-p)/N}}\right| \le Z_{1-\alpha/2}\right) = 1-\alpha$$

例　考虑一个模型，它在 100 个检验记录上具有 80%的准确率。在 95%的置信水平下，模型的真实准确率的置信区间是什么？

随着记录数 N 的增大所产生的置信区间：

N	20	50	100	500	1000	5000
置信区间	0.584 –0.919	0.670 –0.888	0.711 –0.867	0.763 –0.833	0.774 –0.824	0.789 –0.811

- Given two models, say M1 and M2, which is better?
  - M1 is tested on D1 (size=n1), found error rate = e_1
  - M2 is tested on D2 (size=n2), found error rate = e_2
  - Assume D1 and D2 are independent
  - If n1 and n2 are sufficiently large, then

$$e_1 \sim N\left(\mu_1, \sigma_1^2\right)$$

$$e_2 \sim N\left(\mu_2, \sigma_2^2\right)$$

  - Approximate: $\hat{\sigma}_i^2 = \frac{e_i(1-e_i)}{n_i}$

- To test if performance difference is statistically significant: d = e1 – e2
  - d ~ $N(d_t, \sigma^2_t)$ where d_t is the true difference
  - Since D1 and D2 are independent, their variance adds up:

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  - At (1-α) confidence level, $d_t = d \pm Z_{1-\alpha/2}\hat{\sigma}_t$

- An Illustrative Example
  - Given: M1: n1 = 30, e1 = 0.15
    M2: n2 = 5000, e2 = 0.25
  - d = |e2 – e1| = 0.1 (2-sided test)

$$\hat{\sigma}_d^2 = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

  - At 95% confidence level, $Z_{1-\alpha/2}$=1.96

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

  - => Interval contains 0 => difference may not be statistically significant

- 交叉验证配对t检验
  - 两个学习器A、B
    - K折交叉验证法得测试集上的误差为 $\epsilon_1^A, \epsilon_2^A, \ldots, \epsilon_k^A$ 和 $\epsilon_1^B, \epsilon_2^B, \ldots, \epsilon_k^B$
    - 若A、B性能相同，则在相同训练/测试集上的误差应一样，即 $\Delta_i = \epsilon_i^A - \epsilon_i^B = 0$
    - 计算差值的均值和方差 $\bar{\Delta} = \frac{\sum_{i=1}^{K} \Delta_i}{K}$ $S^2 = \frac{\sum_{i=1}^{K} \left(\Delta_i - \bar{\Delta}\right)^2}{K-1}$
    - $H_0 : \varepsilon^A - \varepsilon^B = 0$

$$\frac{\sqrt{K}\bar{\Delta}}{S} \sim t(K-1)$$

- 5×2交叉验证配对t检验
  - 缓解测试误差不独立的问题（高估假设成立的概率）
  - 做五次2折交叉验证，每次随机将数据打乱
  - 只计算第一次2折验证的差值平均值（缓解非独立性）
  - 每次都计算方差

$$\mu = \frac{\Delta_1^1 + \Delta_1^2}{2} \qquad \sigma_i^2 = \left(\Delta_i^1 - \frac{\Delta_i^1 + \Delta_i^2}{2}\right)^2 + \left(\Delta_i^2 - \frac{\Delta_i^1 + \Delta_i^2}{2}\right)^2$$

$$\tau_t = \frac{\mu}{\sqrt{0.2\sum_{i=1}^{5}\sigma_i^2}} \text{ 服从自由度为 5 的 } t \text{ 分布}$$

- McNemar检验
  - 考察二学习器

两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	e_{00}	e_{01}
错误	e_{10}	e_{11}

  - 若二学习器性能相同，则$e_{01}=e_{10}$

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \sim \chi^2(1)$$

- Friedman检验和Nemenyi检验
  - 用于多算法的比较（H_0：所有算法性能相同）
  - 基于算法排序
    - 使用交叉验证法得到每个算法在每个数据集上的测试结果
    - 在每个数据集上根据测试性能好坏排序，并赋序值1，2，……。若算法性能相同，则平分序值
    - 计算平均序值

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

- 若算法性能相同，则平均序值应相同。

$$\tau_{\chi^2} = \frac{k-1}{k} \cdot \frac{12N}{k^2-1} \sum_{i=1}^{k} \left(r_i - \frac{k+1}{2} \right)^2$$

$$= \frac{12N}{k(k+1)} \left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4} \right)$$

在 k 和 N 都较大时，服从自由度为 $k-1$ 的 χ^2 分布.

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(k-1) - \tau_{\chi^2}} \sim F(k-1, (k-1)(N-1))$$

- 若H0被拒绝，则算法性能显著不同。需要进一步区分各算法。

Nemenyi 检验计算出平均序值差别的临界值域

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}},$$

若两个算法的平均序值之差超出了临界值域 CD, 则以相应的置信度拒绝“两个算法性能相同”这一假设.

- 例（续）
  - K = 3，N = 4，$\alpha = 0.05$

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

Nemenyi 检验中常用的 q_α 值

α	算法个数 k		
	2	3	4
0.05	1.960	2.344	2.569
0.1	1.645	2.052	2.291

$$\tau_F = 24.429$$

F 检验的常用临界值

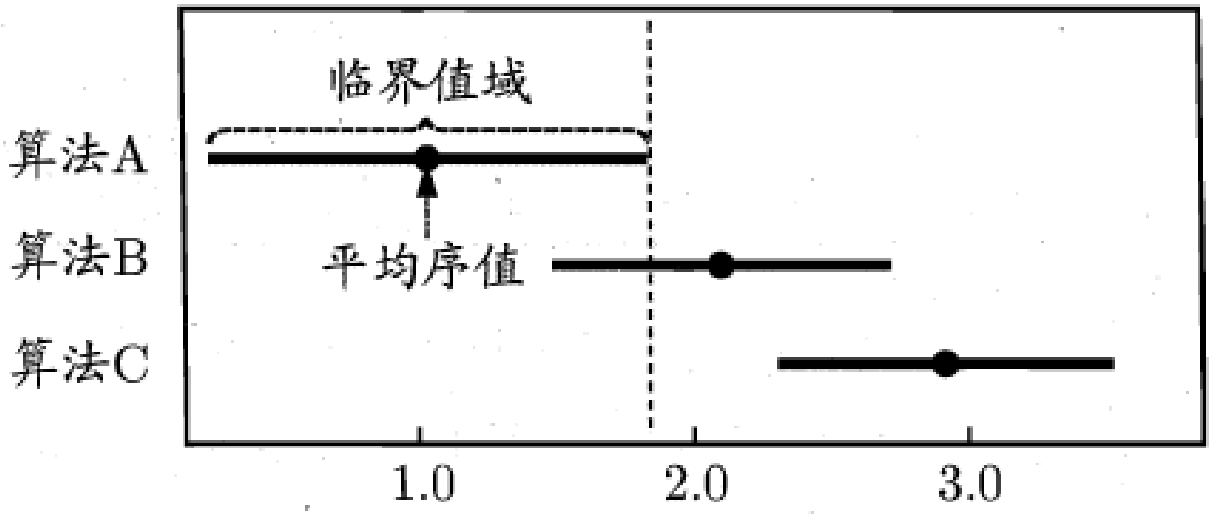
$\alpha = 0.05$		
数据集	算法个数 k	
个数 N	2	3
4	10.128	5.143
5	7.709	4.459
8	5.591	3.739
10	5.117	3.555
15	4.600	3.340
20	4.381	3.245

拒绝原假设！

$$CD = 1.657$$

A、C性能显著不同。

- Friedman检验图
  - 横轴：平均序值；纵轴：各算法
  - 点：每个算法的平均序值；横线段：临界值域
  - 若两个算法的横线段有交叠，则算法无显著差别；否则说明有显著差别。

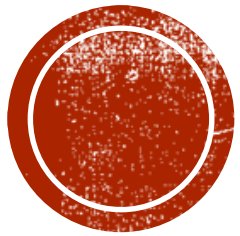


Friedman 检验图

- 偏差-方差分解

假定$\mathbb{E}_D[y_D - y] = 0$.　　$\bar{f}(\boldsymbol{x}) = \mathbb{E}_D[f(\boldsymbol{x}; D)]$

$$
\begin{aligned}
E(f;D) &= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)+\bar{f}\left(\boldsymbol{x}\right)-y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y_D\right)^2\right] \\
&\quad +\mathbb{E}_D\left[2\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)\left(\bar{f}\left(\boldsymbol{x}\right)-y_D\right)\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y+y-y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y\right)^2\right]+\mathbb{E}_D\left[\left(y-y_D\right)^2\right] \\
&\quad +2\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y\right)\left(y-y_D\right)\right]
\end{aligned}
$$



THE END!