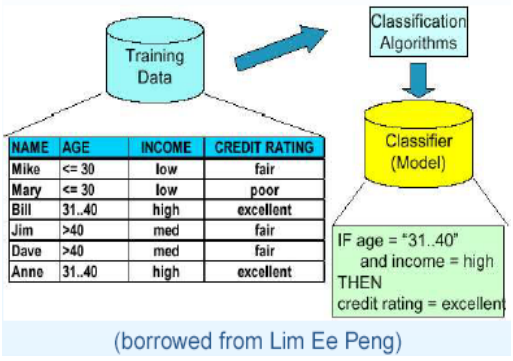
第一章 数据科学与机器学习基本概念

1. **指导学习是什么？**
   1.1 指导学习的类型三种观点：搜索、生成、模型
   1.2 概念和概念学习
2. **基本术语：**
   2.1 数据集、属性、样本空间
   2.2 假设、假设空间、偏序，归纳学习
   2.3 版本空间和泛化学习

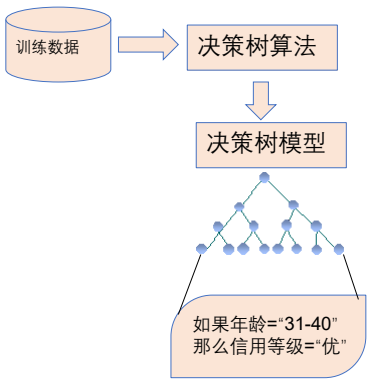
1. 指导学习(Supervised Learning):

用一组输入变量 (predictors, inputs, features, independents)
对输出变量 (responses, outputs , dependent) 产生预测



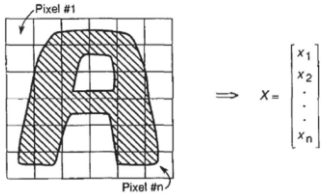
NAME	AGE	INCOME	CREDIT RATING
Mike	<= 30	low	fair
Mary	<= 30	low	poor
Bill	31..40	high	excellent
Jim	>40	med	fair
Dave	>40	med	fair
Anne	31..40	high	excellent

(borrowed from Lim Ee Peng)

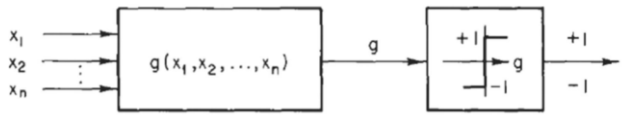


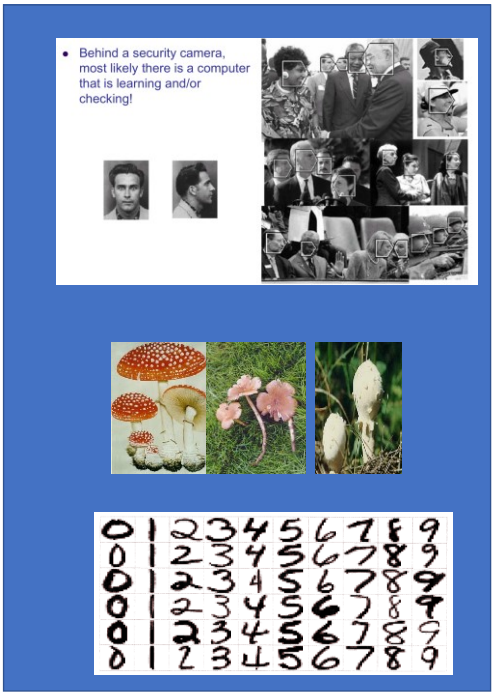
最常见的有指导的学习：Classification:分类问题
Learn a method for predicting the instance class from pre-labeled (classified) instances
人员定位问题，类别识别，数字识别

- Representing data:



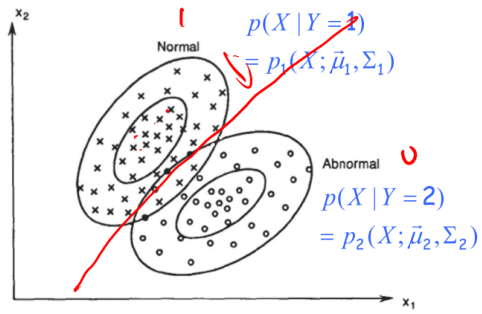
- Hypothesis (classifier)





Decision-making as dividing a high-dimensional space高维空间的决策问题

- Classification-specific Dist.: P(X|Y)



- Class prior (i.e., "weight"): P(Y)

学习问题的一般表示

- $\mathcal{X}$ 输入空间（ $\mathcal{X} \subseteq \mathbb{R}^d$ ），每个元样例 $x_i = \{x_{i1}, \cdots, x_{id}\}$.
- $\mathcal{Y}$ 输出空间每个元样例 y_i.

  分类问题（分类的输出）：$\mathcal{Y} = \{c_1, \cdots, c_k\}$;

  回归问题（连续的输出）：$\mathcal{Y} \subseteq \mathbb{R}$
- $S = \{(x_i, y_i)\}_{i=1}^m$：训练样本

有监督的学习模型问题就是要计算出一个最优的函数，该函数可以恰当地描述输入和输出之间的关系。

5

1.1指导学习的类型

- **定义：指导**学习的目标是学习输入到输出的映射关系，其中正确值已部分地由指导者通过训练数据给出。
- **类型**:
  - 概念学习：0-1学习
    - 特点1：将学习问题转化为一个搜索问题；
    - 特点2：强调假设空间的性质、搜索算法和评价准则；
  - 生成式学习：
  - 模型学习（统计学习）：回归
    - 特点1：将学习问题转化为一个估计问题，特别是分布的特征估计问题；
    - 特点2：强调分布选择，估计的性质和模型的解释；

6

2.1 数据集、属性到样本空间

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

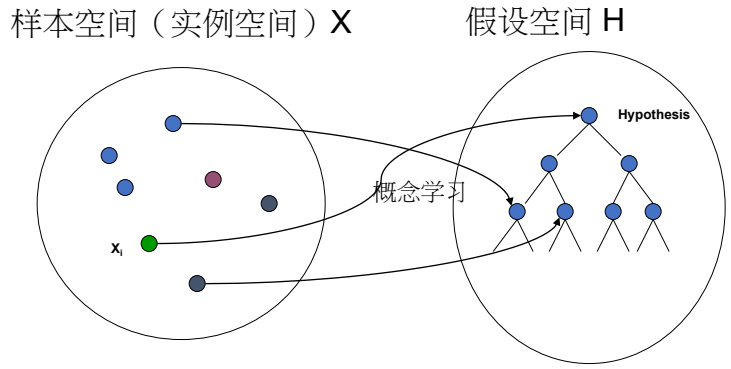
- 以样本的属性为坐标轴张成的多维空间，也叫属性空间、输入空间。
- 上例中，每行样本包含三个属性：色泽、根蒂、敲声，则可以以这三个属性为坐标轴，生成一个三维空间，每个西瓜（只要用这三种属性描述）都能在该空间中找到其对应的坐标位置。

泛化、假设空间和版本空间

- **泛化**：通过对训练集中“好瓜”的经验归纳出对没有见过的瓜进行判断的能力。
- 上例中，采用属性合取式描述假设空间假设空间由形如“(色泽=？) ∧ (根蒂=？) ∧ (敲声=？)” 的所有假设组成。
- 如果属性色泽、根蒂、敲声分别有3 、2 、3种可能取值，还要考虑到一种属性可能无论取什么值都合适（用通配符* 表示），另外有一种情况就是好瓜这个概念根本不成立（用∅表示），则不同语义的假设空间大小为 $(3+1)\times(2+1)\times(3+1)+1=49$

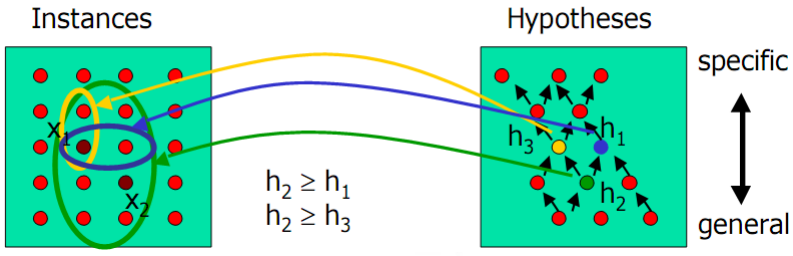
Learning

学习问题经常归结为搜索问题，即对一个假设空间进行搜索，以确定最佳拟和观察到的数据和学习器中已有的假设。



假设的一般到特殊序：偏序,很多假设空间的假设存在序结构

x1=<青绿,蜷缩,浊响>
x2=<乌黑,蜷缩,浊响>



h3=<？ ,蜷缩,浊响>
h1=< 青绿,?,浊响 >
h2=< ？ ,？ ,浊响 >

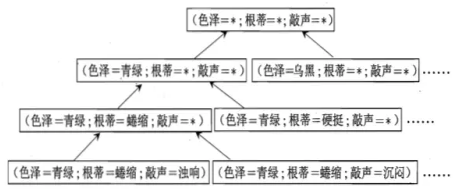
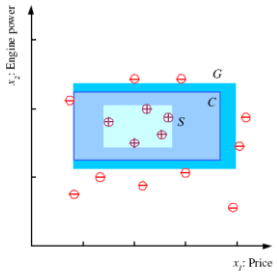


图 1.1 西瓜问题的假设空间

最特殊的假设，最一般的假设



涵盖所有正例不包括任何负例的最小的假设
<∅, ∅, ∅, ∅,…, ∅ >

most general hypothesis, G
<?,?,?,?,…,?>涵盖所有正例不包括任何负例的最大的假设

h ∈ $\mathcal{H}$, between S and G is consistent

and make up the version space
(Mitchell, 1997)

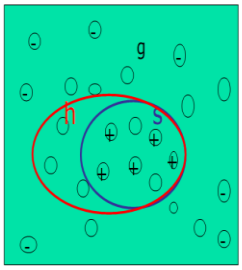
一致性定义:一个假设h和训练样本集称为一致,当且仅当对D中每个样例(x,C(x)),有C(x)=h(x)

$$\text{consistent}(h, D) = \{\forall(x, c(x)) \in D, h(x) = C(x)\}$$

FIND-S特点

- 属性合取式的假设空间
- 保证输出为H中与正例一致的最大的特殊假设
- 只要正确的目标概念在H中，训练数据正确，也可能得到最好的假设

Find-S的不足：



h is consistent with D, then h>s;

- Can't tell if the learner has converged to the target concept, in the sense that it is unable to determine whether it has found the only hypothesis consistent with the training examples. (more examples get better approximation)

无法评价所得假设与目标的接近程度

- Can't tell when training data is inconsistent, as it ignores negative training examples. (prefer to detect and tolerate errors or noise)

忽略负例

- Why prefer the most specific hypothesis? Why not the most general, or some other hypothesis? (more specific less likely coincident)

忽略一般性

- What if there are multiple maximally specific hypothesis? (all of them are equally likely)

不允许多个假设共存

13

2.2.版本空间基本概念

- Any $h \in H$ between S and G
- Consisting of valid hypotheses with no error (consistent with the training set)

1.**版本空间定义**: 假设空间H和训练数据集D的变型空间是H中每个与训练样本D一致的假设构成的子集

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

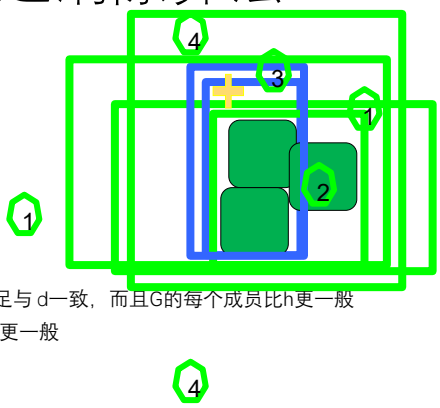
2.关于假设空间H和训练数据集D的**一般边界**(General Boundary)

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

3.关于假设空间H和训练数据集D的**特殊边界**(Specific Boundary)

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

使用版本空间的候选消除算法



- 将G中初始化为H中极大一般假设
- 将S中初始化为H中极小特殊假设
- 对每个样例d,进行以下操作:

如果d是正例

（1）从G中移出所有与d不一致的假设

（2）对S中每个与d不一致的假设s

从S中移除s

把s中所有极小泛化h加入到S中，其中h满足与 d一致，而且G的每个成员比h更一般

（3）从S中移除所有的假设：它比S中另一假设更一般

如果d是负例

（1）从S中移出所有与d不一致的假设

（2）对G中每个与d不一致的假设g

从G中移除g

把g中所有极小特殊h加入到G中，其中h满足与 d一致，而且S的每个成员比h更特殊

（3） 从G中移除所有的假设：它比G中另一假设更特殊

15

在西瓜问题中，如何根据训练集求所对应的版本空间

- ①写出假设空间：先列出所有可能的样本点（即特征向量）（即每个属性都取到所有的属性值）
- ②对应着给出已知数据集，将与正样本不一致的、与负样本一致的假设删除。
- 即可得出与训练集一致的假设集合，也就是版本空间了。

表 1.1 西瓜数据集

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

表1.1的训练数据集对应的假设空间应该如下：

1 色泽 = * , 根蒂 = * , 敲声 = *
2 色泽 = 青绿 , 根蒂 = * , 敲声 = *
3 色泽 = 乌黑 , 根蒂 = * , 敲声 = *
4 色泽 = * , 根蒂 = 蜷缩 , 敲声 = *
5 色泽 = * , 根蒂 = 硬挺 , 敲声 = *
6 色泽 = * , 根蒂 = 稍蜷 , 敲声 = *
7 色泽 = * , 根蒂 = * , 敲声 = 浊响
8 色泽 = * , 根蒂 = * , 敲声 = 清脆
9 色泽 = * , 根蒂 = * , 敲声 = 沉闷
10 色泽 = 青绿 , 根蒂 = 蜷缩 , 敲声 = *
11 色泽 = 青绿 , 根蒂 = 硬挺 , 敲声 = *
12 色泽 = 青绿 , 根蒂 = 稍蜷 , 敲声 = *
13 色泽 = 乌黑 , 根蒂 = 蜷缩 , 敲声 = *
14 色泽 = 乌黑 , 根蒂 = 硬挺 , 敲声 = *
15 色泽 = 乌黑 , 根蒂 = 稍蜷 , 敲声 = *
16 色泽 = 青绿 , 根蒂 = * , 敲声 = 浊响
17 色泽 = 青绿 , 根蒂 = * , 敲声 = 清脆
18 色泽 = 青绿 , 根蒂 = * , 敲声 = 沉闷
19 色泽 = 乌黑 , 根蒂 = * , 敲声 = 浊响
20 色泽 = 乌黑 , 根蒂 = * , 敲声 = 清脆
21 色泽 = 乌黑 , 根蒂 = * , 敲声 = 沉闷
22 色泽 = * , 根蒂 = 蜷缩 , 敲声 = 浊响
23 色泽 = * , 根蒂 = 蜷缩 , 敲声 = 清脆
24 色泽 = * , 根蒂 = 蜷缩 , 敲声 = 沉闷
25 色泽 = * , 根蒂 = 硬挺 , 敲声 = 浊响
26 色泽 = * , 根蒂 = 硬挺 , 敲声 = 清脆
27 色泽 = * , 根蒂 = 硬挺 , 敲声 = 沉闷
28 色泽 = * , 根蒂 = 稍蜷 , 敲声 = 浊响
29 色泽 = * , 根蒂 = 稍蜷 , 敲声 = 清脆
30 色泽 = * , 根蒂 = 稍蜷 , 敲声 = 沉闷
31 色泽 = 青绿 , 根蒂 = 蜷缩 , 敲声 = 浊响
32 色泽 = 青绿 , 根蒂 = 蜷缩 , 敲声 = 清脆
33 色泽 = 青绿 , 根蒂 = 蜷缩 , 敲声 = 沉闷
34 色泽 = 青绿 , 根蒂 = 硬挺 , 敲声 = 浊响
35 色泽 = 青绿 , 根蒂 = 硬挺 , 敲声 = 清脆
36 色泽 = 青绿 , 根蒂 = 硬挺 , 敲声 = 沉闷
37 色泽 = 青绿 , 根蒂 = 稍蜷 , 敲声 = 浊响
38 色泽 = 青绿 , 根蒂 = 稍蜷 , 敲声 = 清脆
39 色泽 = 青绿 , 根蒂 = 稍蜷 , 敲声 = 沉闷
40 色泽 = 乌黑 , 根蒂 = 蜷缩 , 敲声 = 浊响
41 色泽 = 乌黑 , 根蒂 = 蜷缩 , 敲声 = 清脆
42 色泽 = 乌黑 , 根蒂 = 蜷缩 , 敲声 = 沉闷
43 色泽 = 乌黑 , 根蒂 = 硬挺 , 敲声 = 浊响
44 色泽 = 乌黑 , 根蒂 = 硬挺 , 敲声 = 清脆
45 色泽 = 乌黑 , 根蒂 = 硬挺 , 敲声 = 沉闷
46 色泽 = 乌黑 , 根蒂 = 稍蜷 , 敲声 = 浊响
47 色泽 = 乌黑 , 根蒂 = 稍蜷 , 敲声 = 清脆
48 色泽 = 乌黑 , 根蒂 = 稍蜷 , 敲声 = 沉闷
49 Ø

Candidate-Elimination Algorithm

S_0 = {<∅, ∅, ∅>}
G_0 = {<?, ?, ?>}

S_1 = {<青绿, 蜷缩,浊响 >}
G_1 = {<?, ?, ? >}

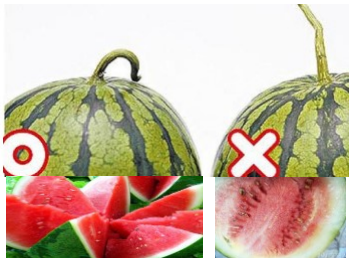
S_2 = {<?, 蜷缩,浊响>}
G_2 = {<?, ?, ? >}

S_3 = {<?, 蜷缩,浊响>}
G_3 = { <?,蜷缩, ? >, <?, ?,浊响>}

S_4 = {<?, 蜷缩,浊响>}
G_4 = { <?,蜷缩, ? >, <?, ?,浊响>}

表 1.1 西瓜数据集

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



Candidate-EliminationAlgorithm(收敛性)

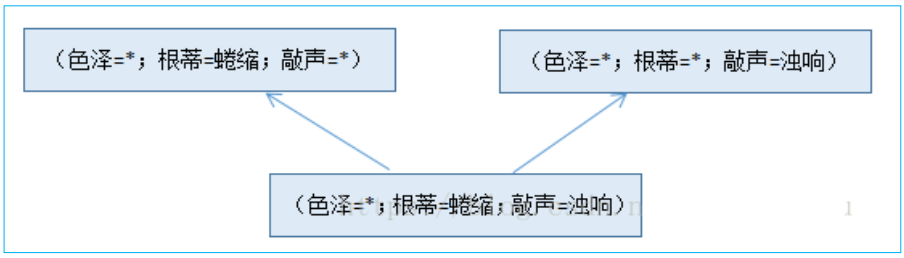
- 候选消去算法的特点：寻找与训练样例一致的假设；
- 原理：通过S泛化和G的特殊化不断缩小版本空间，实现对一致假设的搜索。
- The version space will converge toward the correct target concepts if:
  - H contains the correct target concept
  - H中包含了描述目标概念的正确假设(可知学习)
  - There are no errors in the training examples
    在训练样本中没有错误（完全学习）
- A training instance to be requested next should discriminate among the alternative hypotheses in the current version space。理想的训练样例是对S和G都有作用，于是可以使边界单调移动，从而有效地推动搜索进程。
- Partially learned concept can be used to classify new instances using the majority rule.不完全学习仍然可以用于预测

19

学习过后剩余的假设为：

- 4 色泽 = *，根蒂 = 蜷缩，敲声 = *
- 7 色泽 = *，根蒂 = *，敲声 = 浊响
- 22 色泽 = *，根蒂 = 蜷缩，敲声 = 浊响

这就是最后的“假设集合”，也就是“版本空间”。



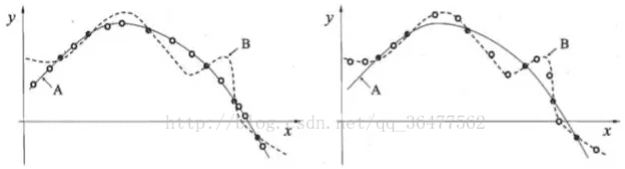
在同一个训练样本集上进行匹配，可能匹配出多个假设，那么机器学习选择算法的依据是什么呢?

归纳偏好：什么是一个好模型?

- **归纳偏好：**机器学习算法在学习过程中对某种类型假设的偏好。
- 任何一个有效的机器学习算法必有其归纳偏好。
- **“奥卡姆剃刀”原则：**“若有多个假设与观察一致，则选最简单的那一个。”

  注意：奥卡姆剃刀并非唯一可行的原则;

什么是最佳拟合



(a) A 优于 B　　(b) B 优于 A

所有“问题”出现的机会相同、或所有问题同等重要。

但实际情况并不是这样的，很多时候，我们只关注自己正在试图解决的问题。比如，要找到快速从A地到B地的算法，如果我们考虑A地是人民大学东门、B地是北京大学数学楼，那么“骑自行车”是很好的解决方案；但是这个方案对A地是人民大学东门、B地是山东大学的情形显然很糟糕，但研究算法的人对此并不关心。

所以，NFL定理最重要的寓意，是让我们清楚意识到，**脱离具体场景，空泛地谈论“什么学习算法更好”毫无意义。**

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

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

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

考虑二分类问题, 且真实目标函数可以是任何函数 $\mathcal{X}\mapsto\{0,1\}$, 函数空间为 $\{0,1\}^{|\mathcal{X}|}$. 对所有可能的 f 按均匀分布对误差求和, 有

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

式(1.2)显示出, 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 $\mathfrak{L}_b$, 我们都有

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

也就是说, 无论学习算法 $\mathfrak{L}_a$ 多聪明、学习算法 $\mathfrak{L}_b$ 多笨拙, 它们的期望性能竟然相同! 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].

机器学习试验律

- **试验律（NFL律：No Free Lunch）：对于数据分析者来说，天下没有免费的午餐，一个正确的模型只有通过试验（experiment）才能被发现。** 这也成为机器学习一项基本原则：如果我们充分了解一个问题空间（problem space），我们可以选择或设计一个找到最优方案的最有效的算法。**一个卓越算法的参数依赖于数据挖掘的问题空间一组特定的属性集，这些属性可以通过分析发现或者算法创建**。但是，这种观点事实上来自于一个错误的设想，在数据挖掘过程中数据挖掘者将问题公式化，然后利用算法找到解决方法。事实上，**数据挖掘者将问题形式化和寻找解决方法是同时进行的-----算法仅仅是帮助数据挖掘者理解数据和问题之间桥梁的一个工具。**