



第8章 分类理论

北京交通大学《机器学习》课程组





可乎可，不可乎不可。

——《庄子·齐物论》

能认可吗？一定有可以加以肯定的东西方才可以认可；

不可以认可吗？一定也有不可以加以肯定的东西方才不能认可。



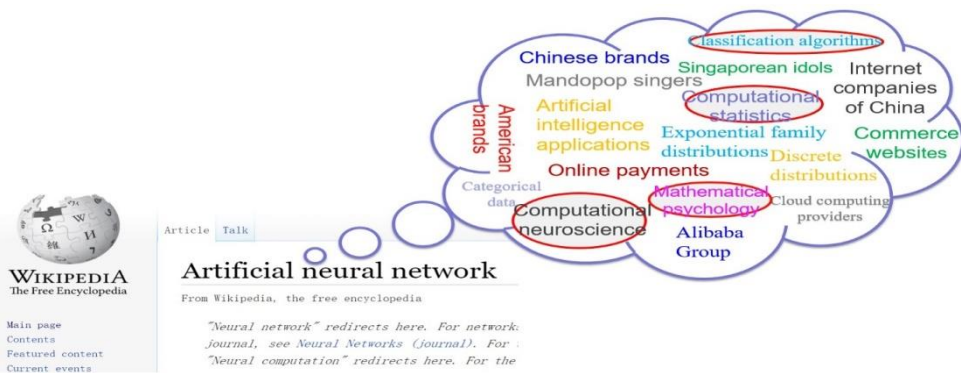
目录

- **8.1 分类及相关定义**
- **8.2 从归类理论到经典分类理论**
 - 8.2.1 概率近似正确(Probably Approximately Correct, PAC)理论
 - 8.2.2 统计机器学习理论
- **8.3 分类测试公理**
- **8.4 分类结果评价**
- **8.5 拓展应用**
- **8.6 作业**



什么是分类?

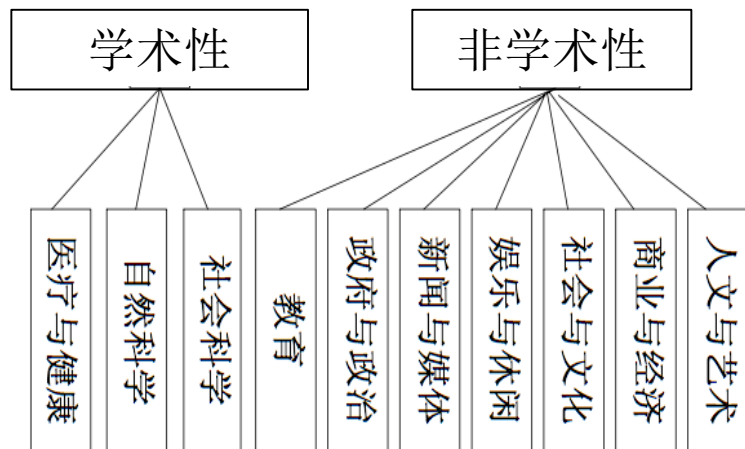
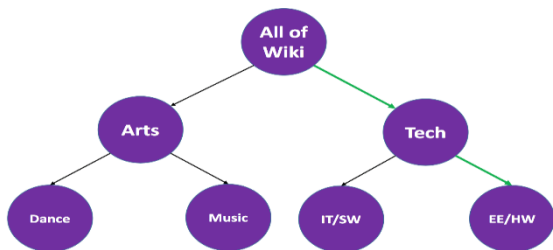
- **分类**: 给定一个**对象**, 从一个**事先定好的分类体系**中挑出一个 (或者多个) **最适合该对象的类别**.
 - **对象**: 可以是任何东西
 - **事先定好的分类体系**: 可能有结构
 - **最合适的**: 判断标准
- **最直接最普遍的应用是便于今后查找。**





分类体系

- 分类体系一般由人工构造
 - 政治、体育、军事、娱乐
- 分类体系可能是层次结构





分类模式

■ 分类模式

- 两类问题：属于或不属于
- 多类问题：多个类别，可拆分成多个两类问题
- 多标签问题：一个对象可以属于多类

Pick one

Label 1	✓
Label 2	

Binary

Pick one

Label 1	
Label 2	
Label 3	
Label 4	✓
...	
...	
Label L	

Multi-class

Pick all applicable

Label 1	
Label 2	✓
Label 3	
Label 4	✓
...	
...	
Label L	✓

Multi-label



应用举例



{Fall foliage, Field}




{Beach, Urban}

scene dataset consists of 2407 images assigned to 6 labels



应用举例



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools
[What links here](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Bharata Natyam


From Wikipedia, the free encyclopedia

Bharathanatyam (Tamil: பரதநாட்டியம்) is a form of [Indian classical dance](#) that originated in the temples of [Tamil Nadu](#).^{[1][2][3][4][5]} It was described in the treatise *Natya Shastra* by [Bharata](#) around the beginning of the common era. Bharata Natyam is known for its grace, purity, tenderness, expression and sculptural poses. [Lord Shiva](#) is considered the God of this dance form. Today, it is one of the most popular and widely performed dance styles and is practiced by male and female dancers all over the world, although it is more commonly danced by women.^[6]

Contents [\[hide\]](#)

- 1 Etymology
- 2 Dance tradition
- 3 Essential ideas
 - 3.1 Spiritual symbolism
- 4 Medieval decline
- 5 Modern rebirth

Bharathanatyam



Dances by name, Indian culture, Performing arts in India, South India, Tamil culture



分类方法

■ 人工方法

- 知识工程的方法建立专家系统（80年代末期）
- 结果容易理解
- 费时费力
- 难以保证一致性和准确性（40%左右的准确率）
- 专家有时凭空想象

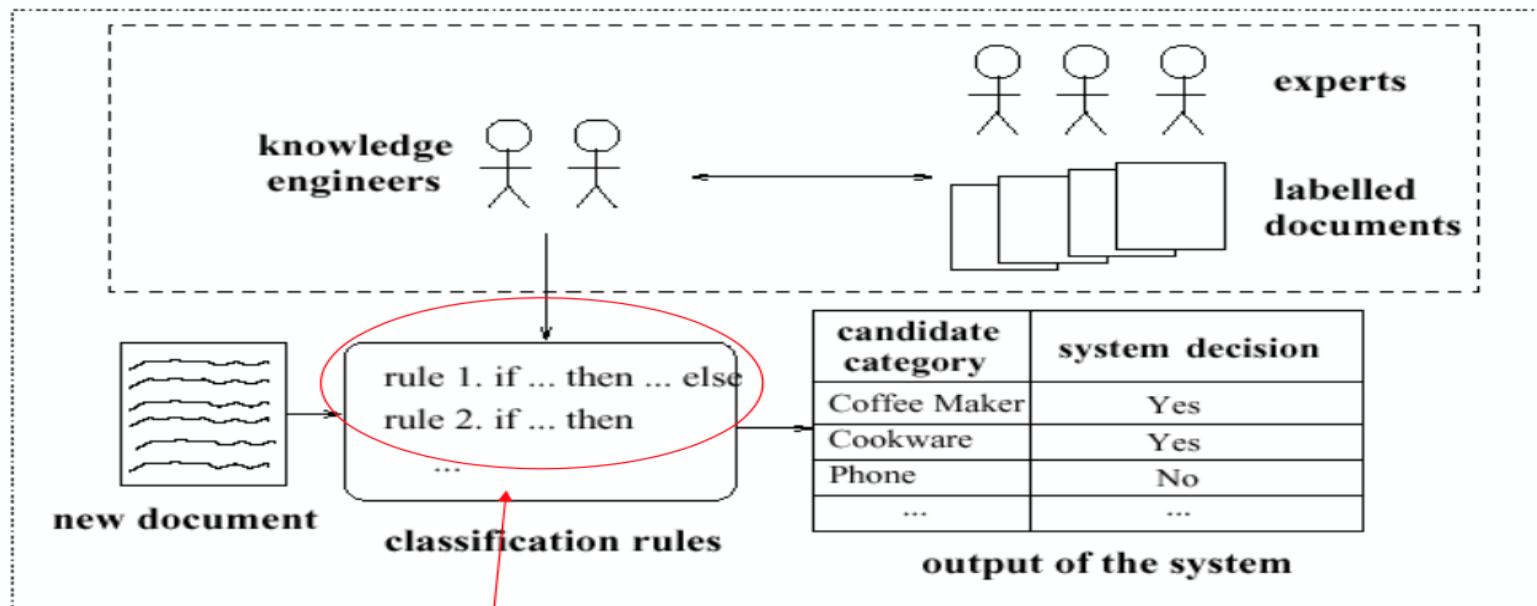
■ 自动方法（学习）

- 减少人工分类的繁杂工作，但结果可能不易解释
- 提高信息处理的效率，且准确率相对较高（85%或者更高）
- 来源于真实数据，减少人工分类的主观性，可信度高



专家系统

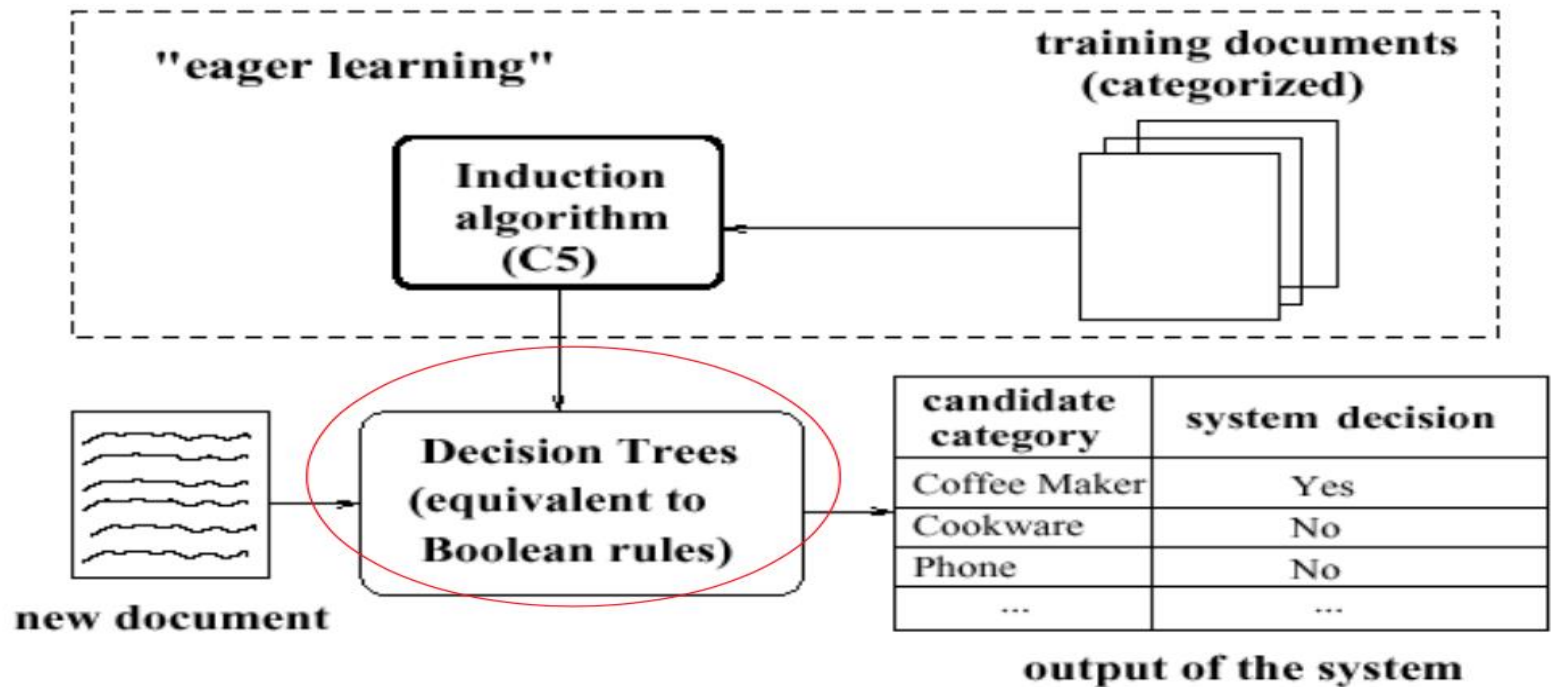
Expert system for text categorization (late 1980s)



人工定义规则

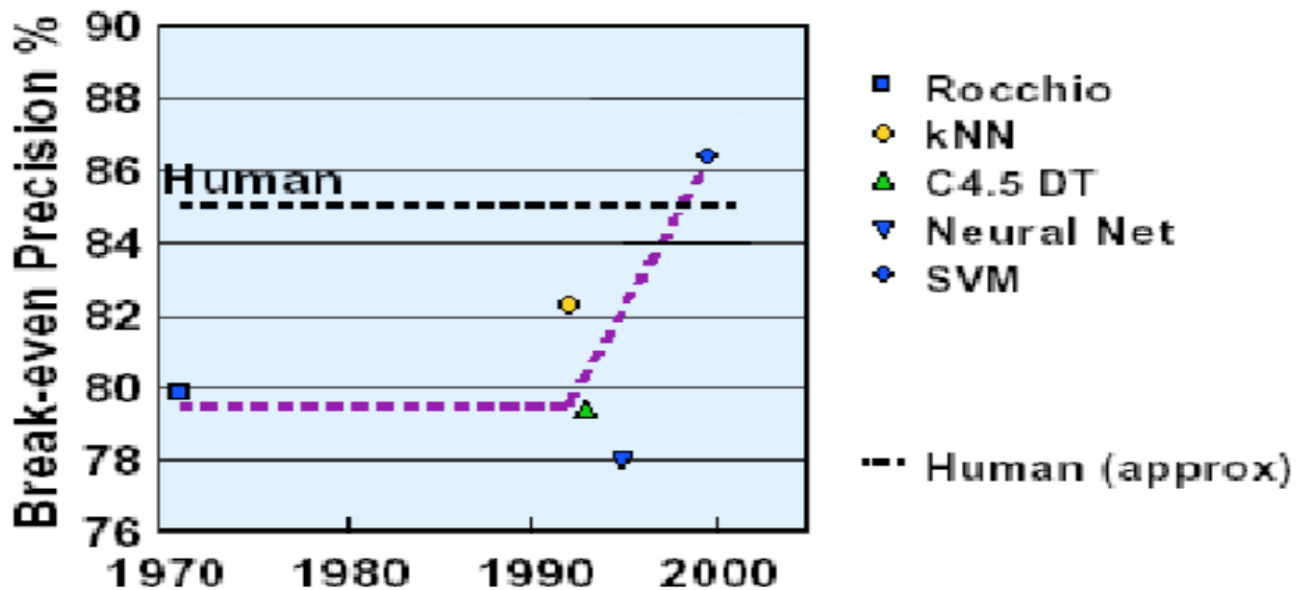


DTree induction for text categorization (since 1994)



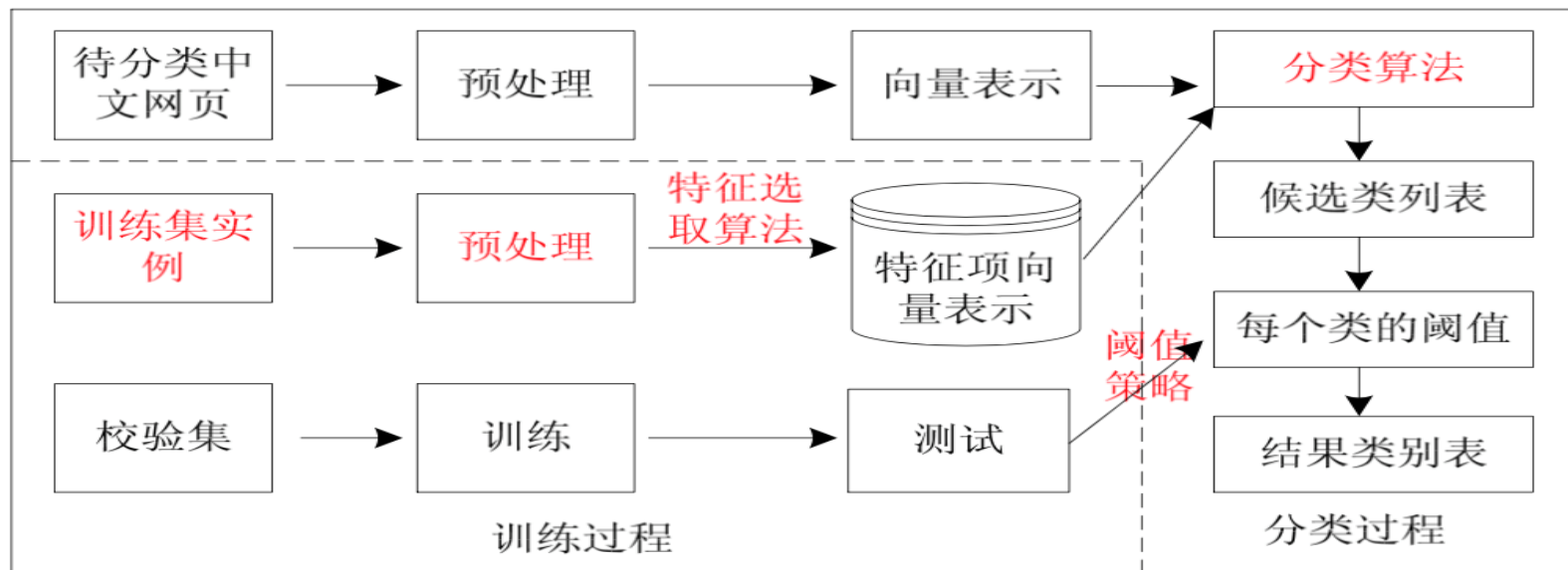


统计学习





分类基本过程





新闻类别预测

政府事务

企业个人事务

Amatil Proposes
Two-for-Five
Bonus Share Issue

Citibank Norway
Unit Loses Six
Mln Crowns in
1986

Japan Ministry
Says Open Farm
Trade Would Hit
U.S.

Vieille Montagne
Says 1986
Conditions
Unfavourable

Jardine Matheson
Said It Sets Two-
for-Five Bonus
Issue Replacing "B"
Shares

Anheuser-Busch
Joins Bid for San
Miguel

Italy's La
Fondiarria to Report
Higher 1986
Profits

Isuzu Plans No
Interim Dividend

Senator Defends
U.S. Mandatory
Farm Control Bill

Bowater Industries
Profit Exceed
Expectations

?

Senate
Panel
Studies
Loan Rate,
Set Aside
Plans

Senate
Panel
Studies
Loan Rate,
Set Aside
Plans



分类问题

- 如果 $c > 1$ 并且已知 (X, U) ，则对应的归类学习问题就是分类问题，是**多类问题**。
- 如果每个对象只有唯一一个类与其对应，则为**标准分类问题**。
- 对**标准分类问题**，类表示存在公理和归类公理成立，类表示唯一公理一般不成立。
 - 如果成立，分类错误率为零，实际应用中过于苛求；
 - 一般能达到工程要求的分类错误率即可；
 - 一个性能良好的分类算法应该使类表示唯一公理尽可能成立，此时类一致性准则至关重要。



分类表示

- 在分类问题中，输入表示为 $(X, U, \underline{X}, \text{Sim}_X)$ ，输出表示为 $(Y, V, \underline{Y}, \text{Sim}_Y)$ 。
- 已知： (X, U) 是训练集
- 待学习： $(\underline{X}, \text{Sim}_X)$ 是期望的分类器， (Y, V) 为训练结果， $(\underline{Y}, \text{Sim}_Y)$ 为实际学到的分类器。
- 通过对训练集 (X, U) 的学习得到分类器 $(\underline{Y}, \text{Sim}_Y)$ 后，对新的测试样本 x_T ，可以通过学到的分类器 $(\underline{Y}, \text{Sim}_Y)$ 预测 x_T 所属的类别。



分类表示

- 输入划分矩阵 $U = [u_{ik}]_{C \times N}$, u_{ik} 不是0就是1。
- **标准分类问题:** U 中的每一列只有一个元素的值为1。本书主要讨论**标准分类问题**。
- **多标记分类问题:** U 是重叠划分, 即 U 中的每一列多于一个元素的值为1。



- 8.1 分类及相关定义
- 8.2 从归类理论到经典分类理论
 - 8.2.1 概率近似正确(PAC)理论
 - 8.2.2 统计机器学习理论
- 8.3 分类测试公理
- 8.4 分类结果评价
- 8.5 拓展应用
- 8.6 作业



机器学习两大理论

➤ 概率近似正确(Probably Approximately Correct, PAC)理论

- 最成熟最古老的计算学习理论
- 最早由Valiant在1984年提出，由此产生了“计算学习理论”这个机器学习的分支领域，提出者获2010年图灵奖。

➤ 统计机器学习理论

- 20世纪90年代由Vapnik建立的一套机器学习理论，一种研究训练样本有限情况下的机器学习规律，它可以看作是基于数据的机器学习问题的一个特例，即有限样本情况下的特例。
- 由这套理论所引出的支持向量机（SVM）对机器学习的理论界以及各个应用领域都有极大的贡献。



相似算子 (像)

输入相似指称: $\tilde{X} = \{\widetilde{x_1}, \widetilde{x_2}, \dots, \widetilde{x_n}\}$

其中 $\widetilde{x_k} = \arg \max_i \text{Sim}_X(x_k, \underline{X_i})$

输出相似指称: $\tilde{Y} = \{\widetilde{y_1}, \widetilde{y_2}, \dots, \widetilde{y_n}\}$

其中 $\widetilde{y_k} = \arg \max_i \text{Sim}_Y(y_k, \underline{Y_i})$

指派算子 (归)

输入归类: $\vec{X} = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \dots, \overrightarrow{x_n}\}$

其中 $\overrightarrow{x_k} = \arg \max_i u_{ik}$

输出归类: $\vec{Y} = \{\overrightarrow{y_1}, \overrightarrow{y_2}, \dots, \overrightarrow{y_n}\},$

其中 $\overrightarrow{y_k} = \arg \max_i v_{ik}$



类表示唯一公理

对一个归类算法，其输入输出对应的类表示（语义）应该相同

如果归类算法的归类输入为 $(X, U, \underline{X}, \text{Sim}_X)$,

其对应归类结果为 $(Y, V, \underline{Y}, \text{Sim}_Y)$,

则有 $\vec{X} = \vec{Y}$, $\underline{X} = \underline{Y}$, $\tilde{X} = \tilde{Y}$ 。

指派算子

内部表示

相似算子

类认知表示



分类问题

- 分类算法希望学到的类输入认知表示为 $(\underline{X}, \text{Sim}_X)$ ，由于是期望学到的，只可能推测其形式。
- 算法真正通过学习得到的只能是输出类认知表示 $(\underline{Y}, \text{Sim}_Y)$ 。
- 由于**类表示唯一性公理**对分类问题不再严格成立，类一致性准则成为 $(\underline{Y}, \text{Sim}_Y)$ 近似逼近 $(\underline{X}, \text{Sim}_X)$ 的保证。
- 分类问题属于多类问题。理论上，多类问题比单类问题研究困难很多。



分类问题

- 分类中 (X, U) 已知

经典分类理论将分类问题转化为了回归问题

- 回归问题可以看作是单类归类问题，便于研究。



分类问题的回归函数

- 当 U 是正则化划分时,
有 $\forall k \in \{1, 2, \dots, N\}$, $\vec{x}_k \in \{1, 2, \dots, C\}$,
故可以定义分类问题的**期望回归函数**为 $\rho(x_k) = \vec{x}_k$ 。
- 由于**归类等价公理**成立, 因此必有 $\forall k(\rho(x_k) = \tilde{x}_k)$ 。
- 当 V 是正则化划分时, 则**学到的回归函数**为 $\hat{h}(y_k) = \vec{y}_k$ 。同样, **归类等价公理**保证 $\forall k(\hat{h}(y_k) = \tilde{y}_k)$



类标函数 与 类标预测函数

对于同一个对象 o ，如果 x 为其输入表示， y 为其输出表示，并假设 $y = \theta(x)$

■ 类标函数：

分类问题的期望回归函数为 $\rho(x) = \vec{x}$ ，

$\rho(x)$ 所组成的集合称为目标空间 T_s

■ 类标预测函数：

学到的回归函数为 $h(x) = \hat{h}(\theta(x)) = \hat{h}(y) = \tilde{y}$ ，

$h(x)$ 所组成的集合称为假设空间记为 H

$$\text{令 } X = \begin{bmatrix} \mathbf{x}_1 & \rho(\mathbf{x}_1) \\ \mathbf{x}_2 & \rho(\mathbf{x}_2) \\ \dots & \dots \\ \mathbf{x}_N & \rho(\mathbf{x}_N) \end{bmatrix}, Y = \begin{bmatrix} \mathbf{x}_1 & h(\mathbf{x}_1) \\ \mathbf{x}_2 & h(\mathbf{x}_2) \\ \dots & \dots \\ \mathbf{x}_N & h(\mathbf{x}_N) \end{bmatrix}.$$



类标函数 与 类标预测函数

$$X = \begin{bmatrix} \mathbf{x}_1 & \rho(\mathbf{x}_1) \\ \mathbf{x}_2 & \rho(\mathbf{x}_2) \\ \dots & \dots \\ \mathbf{x}_N & \rho(\mathbf{x}_N) \end{bmatrix}, Y = \begin{bmatrix} \mathbf{x}_1 & h(\mathbf{x}_1) \\ \mathbf{x}_2 & h(\mathbf{x}_2) \\ \dots & \dots \\ \mathbf{x}_N & h(\mathbf{x}_N) \end{bmatrix}$$

则 $\underline{X} = (\mathbf{x}, \rho(\mathbf{x})), \underline{Y} = (\mathbf{x}, h(\mathbf{x}))$.

类表示唯一公理

如果归类算法的归类输入为
 $(X, U, \underline{X}, \text{Sim}_X)$,

其对应归类结果为 $(Y, V, \underline{Y}, \text{Sim}_Y)$, 则

$\vec{X} = \vec{Y}, \underline{X} = \underline{Y}, \tilde{X} = \tilde{Y}$.

- 分类问题转化为回归问题，所谓的机器学习就是学习一个函数。
- 是单类问题，不需要输入输出指派算子相等和输入输出相似算子相等。
- 只需要输入输出类认知表示相等即可。



8.2.1 PAC理论

- 对单类问题，类表示唯一公理要求 $X = Y$ 成立，也就是要求 $\forall x(\rho(x) = h(x))$ ，要求过高。
 - 实际应用中，类标函数 $\rho(x)$ 未知，只有其有限个值 $\rho(x_k)$ ，其中 $k \in \{1, 2, \dots, N\}$
 - 即使 $\rho(x) = h(x)$ 在 x_1, x_2, \dots, x_N 这有限个对象上成立，也远远不能保证 $\forall x(\rho(x) = h(x))$
 - 函数有无穷多个点，在概率上尽可能成立这是很自然的要求。



8.2.1 PAC理论

■ 归类问题的可分和不可分

➤ 归类问题对学习算法可分：

假设 $H \cap T_S \neq \emptyset$ ，即 $\rho(x) \in H$ 成立

➤ 归类问题对学习算法不可分：

假设 $H \cap T_S = \emptyset$ ，即 $\rho(x) \notin H$ 成立



泛化能力

- 给定训练集 (X, U) ，不管分类问题可分还是不可分，类表示唯一公理一般不再成立。如果成立，则其分类错误率为 0%。
- 类表示唯一公理是分类的理想情形。类一致性准则要求 $\rho(x)$ 与 $h(x)$ 尽可能一致，因此学到的错误率不可能为零，需要估计学到的类标预测函数 $h(x)$ 的错误率。
- 分类算法可以保证学到的类标预测函数在训练集上效果较好，但一般不能保证其在测试数据上的预测效果。而人们期望在测试数据上也能获得良好的预测效果，即泛化能力好。
- 泛化能力：类标预测函数对未知数据的预测能力。理想的分类算法应该具有良好的泛化能力。



泛化错误率

- 训练集 (X, U) 反映类的一个有限抽样，学到的 $h(x)$ 的错误率可能随着抽样的变化而不同，因此 $h(x)$ 的错误率只在抽样分布下有意义。

类一致性准则要求 $\forall x(\rho(x) = h(x))$ 尽可能成立也只能在概率上尽可能成立

- 计算 $\Pr(h(x) = \rho(x))$ 的理论意义更大



泛化错误率

- 已知训练集 (X, U) , 计算 $\Pr(h(x) = \rho(x))$ 非常困难。假设数据集 X 服从抽样分布 P , 那么可以用抽样分布 P 代替 X , 所有的 x_k 都独立服从同一个隐含的概率分布 P 。
- **泛化错误率**: 类标函数与类标预测函数不同的概率, 即所学到类表示的期望风险, 反映学习方法的泛化能力。泛化错误率越小, 说明类表示越有效。

$$\begin{aligned} R(h) &= \Pr_{x \sim P}[h(x) \neq \rho(x)] = E_{x \sim P}[1_{h(x) \neq \rho(x)}] \\ &= 1 - \Pr(\rho(x) = h(x)) \end{aligned}$$

- 类一致性准则要求泛化错误率不能太大, 最好在实际应用中可以容忍的范围内。



PAC辨识

对 $0 < \epsilon, \delta < 1$, 所有类标函数 $\rho(x) \in T_s$ 和抽样分布 P , 如果存在学习算法 \mathfrak{A} , 其输出类标预测函数 $h(x) \in H$ 满足

$$P_r(R(h) \leq \epsilon) \geq 1 - \delta,$$

则称学习算法 \mathfrak{A} 能够从假设空间 H 中**辨识**目标空间 T_s 中的类标函数。

PAC可辨识是类表示唯一公理的弱化

满足PAC辨识的学习算法 \mathfrak{A} 可以以很大的置信度 ($\geq 1 - \delta$) 学到目标空间 T_s 中的某个类标函数 $\rho(x)$ 的近似 (误差最多为 ϵ) 。



PAC可学习

令 N 为根据抽样分布 P 独立同分布得到的数据集 X 中的样例数目, 如果存在学习算法 \mathfrak{A} 和一个 **多项式函数** $poly()$, 对 $0 < \epsilon, \delta < 1$, 所有类标函数 $\rho(x) \in T_S$ 和抽样分布 P , 其在数据集 (X, U) 中输出的类标预测函数 $h_X(x) \in H$ 满足

$$\Pr_{X \sim P^N}(R(h_X) \leq \epsilon) \geq 1 - \delta,$$

其中 $N \geq poly(\frac{1}{\epsilon}, \frac{1}{\delta}, size(x), size(\rho(x)))$

则称目标空间 T_S 对于假设空间 H 是 **PAC可学习的** (有时也简称目标空间 T_S 是PAC可学习的)。

$size(x)$ 表示 x 的最大计算开销, $size(\rho(x))$ 表示 $\rho(x)$ 的最大计算开销
要推导需要概率统计的知识, 可以推导出来。



PAC学习算法和样本复杂度

PAC Learning Algorithm:

如果学习算法 \mathcal{A} 使目标空间 T_S 是 **PAC 可学习** 的，且 \mathcal{A} 的运行时间也是 $poly(\frac{1}{\epsilon}, \frac{1}{\delta}, size(x), size(\rho(x)))$

则称目标空间 T_S 是**高效PAC可学习**的， \mathcal{A} 为目标空间 T_S 的**PAC学习算法**。

如果学习算法 \mathcal{A} 处理每个样本的时间为一个常数，则 \mathcal{A} 的时间复杂度等价于**样本复杂度**。

Sample Complexity:

满足学习算法 \mathcal{A} 所需的样本个数

$N \geq poly(\frac{1}{\epsilon}, \frac{1}{\delta}, size(x), size(\rho(x)))$ 中最小的 N ，称为学习算法 \mathcal{A} 的样本复杂度。。



■ PAC学习的意义

- 是类表示唯一公理在分类问题上的推广框架, 是符合类一致性准则的一个分类问题理论描述框架。
- 给出了一个抽象地刻画分类能力的框架, 基于这个框架可以对很多重要问题进行理论探讨。
 - ✓ 研究某任务在什么样的条件下可学得较好的模型?
 - ✓ 某算法在什么样的条件下可进行有效的学习?
 - ✓ 需要多少训练样例才能获得较好的模型?
- 把对复杂算法的**时间复杂度**的分析转为对**样本复杂度**的分析

需要考虑样本的分布, 针对现有假设空间



8.2.2 统计机器学习理论

- PAC学习理论的泛化错误率的计算假设过于理论化，需要考虑样本的抽样分布。
- 但在学习过程中能够利用的数据集只有训练集，抽样分布并不知道。
- 使用训练数据集的平均损失来代替泛化能力。
- 用经验风险或经验损失 $l(\rho(x), h(x))$ 来代替泛化能力作为设计分类算法的依据。
- 根据类一致性准则，泛化错误率要小，即经验风险越小越好。



经验风险和测试错误率

- 为了评估学习方法的泛化能力，数据分成两部分：训练数据集和测试数据集。
- 经验风险: 模型在训练数据集上的平均误差, 也称为训练误差。

$$D(h, \rho) = \frac{1}{N} \sum_{k=1}^N l(\rho(x_k), h(x_k))$$

$l(\rho(x), h(x))$ 是损失函数, N 为训练集中的样本个数

- 测试错误率: 指测试样本集中误分类的数据所占的比例。

$$\hat{R}(h) = \frac{1}{N_T} \sum_{k=1}^{N_T} 1_{h(x_k) \neq \rho(x_k)}$$



经验风险最小

- 在测试集和训练集服从独立同分布假设的前提下，可以证明测试错误率的期望等于泛化错误率。

$$E(\hat{R}(h)) = R(h)$$

- 用测试错误率来估计分类算法的泛化能力是可行的。
- 在假设空间、损失函数和训练集已知的情况下，经验风险就可以确定类标预测函数的形式。
- 类一致性准则要求经验风险最小，则分类问题变为最优化问题

$$\min_{h \in H} D(h, \rho)$$

如果样本代表性充分且假设空间与目标空间匹配时，经验风险最小能够保证有很好的学习效果，即在测试集上也有很好的泛化性能。



欠拟合和过拟合

- **欠拟合 (under-fitting)** : 学习到的类标预测函数其表示能力比期望的类标函数表示能力简单, 训练误差较大, 测试误差也较大。**解决: 增加类标预测函数的复杂度。**
- **过拟合 (over-fitting)** : 对于本身类标预测函数表示能力很强的学习算法, 如果过度减少训练误差, 测试误差也较大, 过分地拟合了训练数据。特别是样本数比较少的情况下。

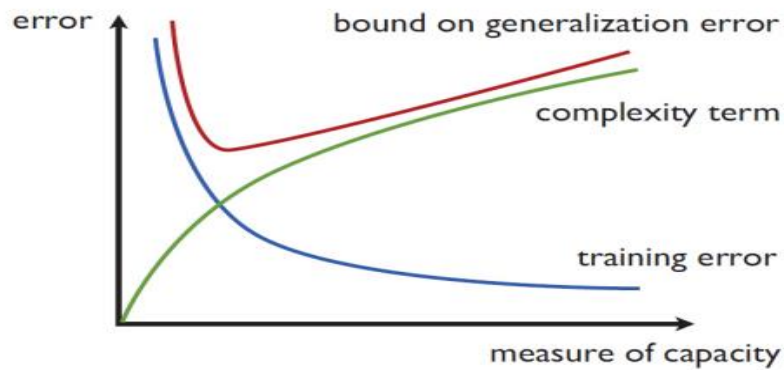
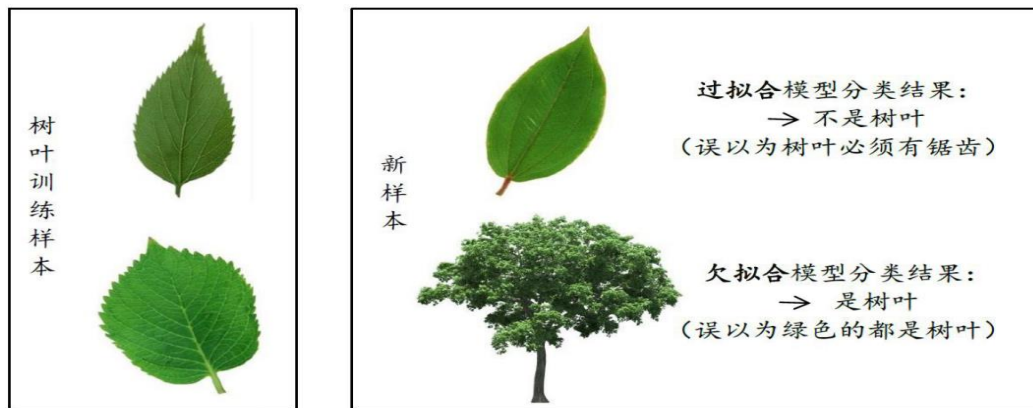


图 8.1 结构风险最小化示意图。



欠拟合和过拟合



过拟合、欠拟合的直观类比

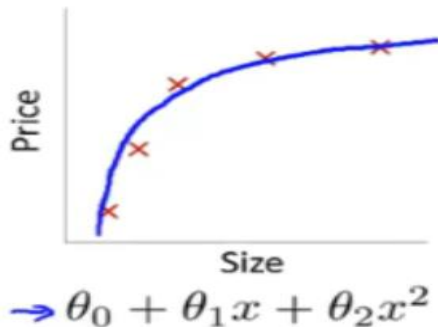
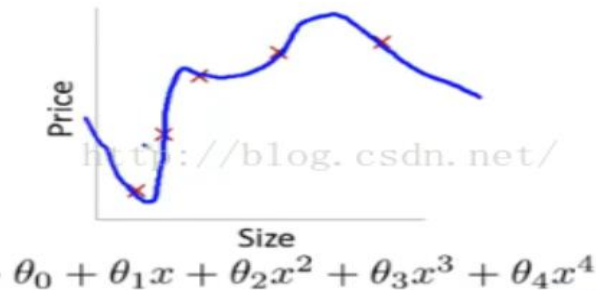
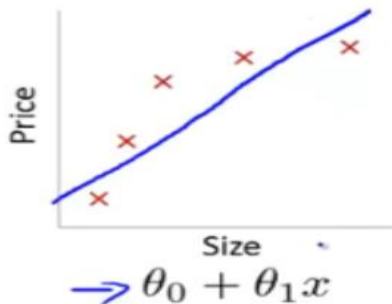
过拟合：学习器把训练样本本身特点当做所有潜在样本都会具有的一般性质。

欠拟合：训练样本的一般性质尚未被学习器学好。



欠拟合和过拟合

- **欠拟合**：模型在训练和预测时表现都不好的情况。
- **过拟合**：模型对于训练数据拟合程度过当的情况。





欠拟合和过拟合

■ 过拟合:

学习器把训练样本学习的“太好”，将训练样本本身的特点当做所有样本的一般性质，导致泛化性能下降

- 优化目标加正则项
- early stop

■ 欠拟合:

对训练样本的一般性质尚未学好

- 决策树:拓展分支
- 神经网络:增加训练轮数

(奥卡姆剃刀准则)



奥卡姆剃刀准则

- Occam's Razor: 14世纪逻辑学家、圣方济各会修士奥卡姆的威廉 (William of Occam, 约1285年至1349年) 提出

“
如无必要，勿增实体”
即 “简单有效原理”

- 威廉使用这个原理证明了许多结论：包括
 - “通过思辨不能得出上帝存在的结论”；
 - 如果你有2个类似的解决方案，选择最简单的；
 - 多出来的东西未必是有益的，相反更容易使我们为自己制造的麻烦而烦恼
 -
 - 应用于企业管理、网络应用、股市分析、理论人物等



模型结构风险最小化

- 模型结构风险最小化：为了防止过拟合现象，在经验风险最小化的基础上同时考虑奥卡姆剃刀准则，选择简单的类标预测函数。
- 同时考虑类一致性准则与奥卡姆剃刀准则，得到模型结构风险最小化准则。在经验风险的基础上再加上模型复杂度的正则项或者惩罚项，获得学习模型：

$$\min_{h \in H} \frac{1}{N} \sum_{k=1}^N l(\rho(x_k), h(x_k)) + \lambda J(h)$$

$J(h)$ 被称为正则化项或惩罚项，表示模型 h 的结构复杂度； λ 越大表示惩罚力度越大； N 为所有样本的数量。



模型结构风险最小化

$$\min_{h \in H} \frac{1}{N} \sum_{k=1}^N l(\rho(x_k), h(x_k)) + \lambda J(h)$$

$J(h)$ 被称为正则化项或惩罚项，表示模型 h 的结构复杂度; λ 越大表示惩罚力度越大; N 为所有样本的数量。

- ✓ 一般来说，模型越复杂，正则化值就越大。
- ✓ 因为越复杂的模型，在训练集上的误差就越小，就越容易发生过拟合现象，所以要增加一项比较大的正则化项来调整模型，来避免过拟合。
- ✓ 正则化模型选择方法在设定分类判据时，平衡考虑了类一致性准则和奥卡姆剃刀准则两方面。



模型选择问题

- 模型结构风险最小化是一个模型选择问题。
- 没有免费的午餐定理：说明学习模型是问题依赖的，没有任何一个普适的模型适用于所有问题。
- 在模型选择中最重要的是适用性选择，即以完成任务的性能好坏为模型（或者算法）选择的首要因素。在泛化性能满足需要的前提下，下一步的问题才是选择简单的模型。
- 如果泛化性能不能满足需要，单纯追求简单的模型也是违反奥卡姆剃刀准则的。
- 一般来说，泛化性能与可解释性是机器学习算法设计者设计学习算法的两个追求。面对具体的学习任务，最理性的选择是选出泛化性能和解释能力都好的学习算法。



白箱算法&黑箱算法

■ 白箱算法:

$$\underline{Y} = (\mathbf{x}, h(\mathbf{x}))$$

- \underline{Y} 被分类算法显式输出，故 \underline{Y} 对于使用者和设计者都是可见的。
- 该类算法偏重可解释性，如最近邻、SVM、概率图等。

■ 黑箱算法:

- \underline{Y} 不被分类算法显式输出，故 \underline{Y} 尽管对于设计者可见的，但对于使用者是不可见的。
- 该类算法偏重泛化性能，如神经网络、随机决策树、集成学习等。



目录

- 8.1 分类及相关定义
- 8.2 从归类理论到经典分类理论
 - 8.2.1 PAC理论
 - 8.2.2 统计机器学习理论
- 8.3 分类测试公理
- 8.4 分类结果评价
- 8.5 拓展应用
- 8.6 作业



■ 测试类表示一致公理：

➤ 对于一个分类问题来说，如果其训练集是 (X, U) ，其测试集为 (X_T, U_T) ，则 $(\underline{X}, Sim_X) = (\underline{X_T}, Sim_{X_T})$ 。

- 提供了分类算法对未知样本具有泛化能力的先决条件。
- 如果类认知表示差别巨大，就不能认为是同一个分类问题。
- 测试结果完全不可信，泛化能力就不能由测试结果来推测。



分类测试公理

■ 测试抽样一致公理（独立同分布假设）：

- 对于一个分类问题来说，如果其训练集是 (X, U) 与测试集为 (X_T, U_T) 的样本彼此独立且服从统一的抽样分布。

训练集与测试集的抽样分布不同，泛化能力也难以估计。

学习样本密度分布时，测试类表示一致公理与测试抽样一致公理两者等价。

类表示与样本密度分布独立时，两者要求不同，前者成立不能保证后者成立。同样，后者成立不能保证前者成立。一般总是假设两者成立。



补充内容

- 评估方法
- 调参和模型参数
- 性能评估



评估方法

- 通过实验测试对学习器的泛化误差进行评估并进而做出选择。
- 使用测试集来测试学习器对新样本的判别能力，以测试集上的测试误差作为泛化误差的近似。
- 测试样本尽量不在训练集中出现、没有在训练过程中使用过。
- 以考试为例。
- 希望泛化性能强的模型，如果测试样本也被用于训练了，将得到过于“乐观”的估计结果。



留出法(hold-out)

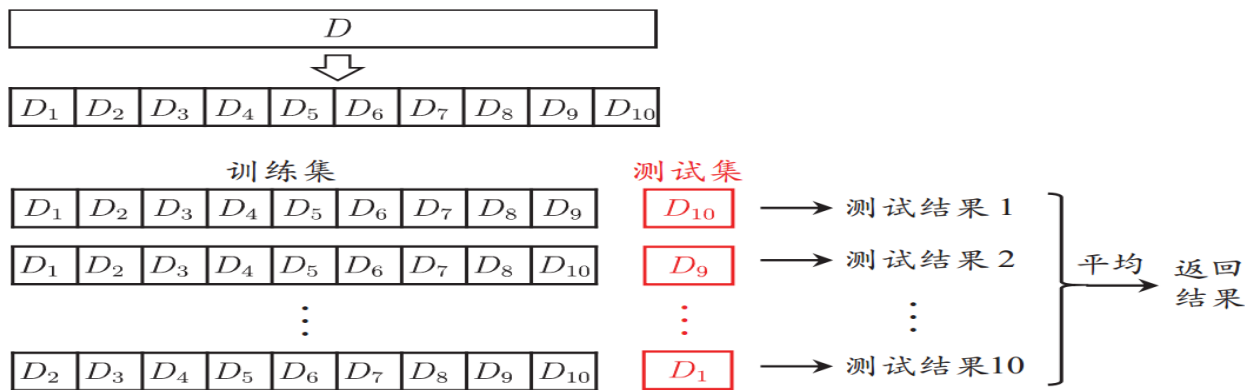
通常将包含 m 个样本的数据集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ 拆分成训练集 S 和测试集 T 。

- 直接将数据集划分为两个互斥集合
- 训练/测试集划分要尽可能保持数据分布的一致性
(如, 类别比例相似)
- 一般若干次随机划分、重复实验取平均值
- 训练/测试样本比例通常为1:1~9:1



交叉验证法

将数据集分层采样划分为 k 个大小相似的互斥子集，每次用 $k-1$ 个子集的并集作为训练集，余下的子集作为测试集，最终返回 k 个测试结果的均值， k 最常用的取值是10.



10 折交叉验证示意图



交叉验证法

与留出法类似，将数据集D划分为k个子集同样存在多种划分方式，为了减小因样本划分不同而引入的差别，k折交叉验证通常随机使用不同的划分重复p次，最终的评估结果是这p次k折交叉验证结果的均值，例如常见的“10次10折交叉验证”

- 假设数据集D包含m个样本，若令 $k = m$ ，则得到留一法LOO：
 - 不受随机样本划分方式的影响
 - 结果往往比较准确
 - 当数据集比较大时，计算开销难以忍受



自助法

- 在留出法和交叉验证法中，由于保留了一部分样本用于测试，因此实际评估模型使用的训练集比 D 小，训练样本规模不同会导致估计偏差。留一法受样本训练规模变化的影响较小，但计算复杂度太高。
- “自助法” (bootstrapping) 可以解决此问题，直接以Efron和Tibshirani于1993年提出的自助采样法(bootstrap sampling)为基础。



自助法

- 给定包含 m 个样本的数据集 D ，采样产生数据集 D' ：每次随机从 D 中挑选一个样本，将其拷贝放入 D' ，然后再将该样本放回初始数据集 D 中，使得该样本在下次采样时仍有可能被采到；这个过程重复执行 m 次，可以得到包含 m 个样本的数据集，这就是自助采样的结果。
- D 中会有多个样本在 D' 中多次出现，而另一部分样本不出现。样本在 m 次采样中始终不被采到的概率是 $(1 - \frac{1}{m})^m$ ，取极限得到

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



自助法

- 通过**自助采样**，初始数据集 D 中约有36.8%的样本未出现在采样数据集 D' 中。可以将 D' 用作训练集， $D-D'$ 用作测试集；实际评估的模型与期望评估的模型都使用 m 个训练样本，而我们仍有数据总量约1/3的、没在训练集中出现的样本用于测试。
- **优点：**自助法在数据集较小、难以划分训练/测试集时很有用；可以从初始数据集产生多个不同的训练集，有利于集成学习。
- **缺点：**产生的数据集改变了初始数据集的分布，会引入估计偏差。数据量足够时，留出法和交叉验证法更常用。



调参和模型参数

- 很多学习算法都有参数需要设定，参数不同，模型的性能往往会有显著差别。因此，需要对算法的参数进行设定，也就是“参数调节”，简称“调参”。
- 对每种参数配置都训练出模型，将对应最好模型的参数作为结果。
 - 不太可行，因为很多参数在实数范围内取值，通常对每个参数选定范围和变化步长。
 - 这样选定的参数值往往不是“最佳”值，是在计算开销和性能估计之间的进行折中的结果。



调参和模型参数

调参是很困难的

- 假定算法有3个参数，每个参数仅考虑5个候选值，这样对每一组训练/测试集就有125个模型需要评估。
- 很多强大的学习算法有很多参数需要设定，这导致调参工作量极大，甚至参数调得不好对最终模型性能有关键影响。



调参和模型参数

- 有两类参数，一类是算法的参数，也称“超参数”，另一类是模型的参数。
- 两者调参方式类似，都是在产生多个模型之后基于某种评估方法进行选择。
- 算法的参数（超参数）
 - 数目常在10以内
 - 通常由人工设定多个参数候选值后产生模型
- 模型的参数
 - 数目很多，如大型“深度学习”模型有上百亿个参数
 - 通过学习产生多个候选模型，例如神经网络在不同轮数停止训练



调参和最终模型

- 给定包含 m 个样本的数据集 D ，在模型评估与选择过程中由于需要流出一部分数据进行评估测试，只采用了部分数据训练模型。
- 在模型选择完成后，学习算法和参数配置已选定，此时应该用数据集 D 重新训练这个模型。这个模型使用了 m 个数据进行训练，这是提交给用户的最终模型，用来对未知样本进行预测。



测试集与验证集

- 把学习得到的模型在实际应用中遇到的数据称为测试数据。模型评估与选择中用于评估测试的数据集常称为“验证集” (validation set).
- 在研究对比不同算法的泛化性能时，用测试数据集上的判别效果来估计模型在实际使用时的泛化能力。
- 把训练数据再划分为训练集和验证集，基于验证集上的性能来进行模型选择和调参。



性能度量

- 性能度量是衡量模型泛化能力的评价标准。
- 不同的性能度量往往会导致不同的评判结果，这说明“性能好坏”是相对的，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**。
- 错误率：分类错误的样本数占样本总数的比例。
- 精度（准确率）：分类正确的样本数占样本总数的比例。



Confusion Matrix

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)
	Predicted condition negative	False negative (Type II error)	True negative

根据分类混淆矩阵计算出各种性能指标



Confusion Matrix

■ Precision, Recall, FOR, FPR

		True condition		
Total population		Condition positive	Condition negative	
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	



Confusion Matrix

■ FNR, TNR, FDR, NPV

		True condition		
Total population		Condition positive	Condition negative	
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	False negative (Type II error)	True negative	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	



Confusion Matrix

■ACC, LR+, LR-, DOR, Fscore, Prevalence

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)
	Predicted condition negative	False negative (Type II error)	True negative

$$\text{Accuracy (ACC)} = \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$$

$$\text{Prevalence} = \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$$

$$\text{Positive likelihood ratio (LR+)} = \frac{\text{TPR}}{\text{FPR}}$$

$$\text{Negative likelihood ratio (LR-)} = \frac{\text{FNR}}{\text{TNR}}$$

$$\text{Diagnostic odds ratio (DOR)} = \frac{\text{LR+}}{\text{LR-}}$$

$$\text{F}_1 \text{ score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$$

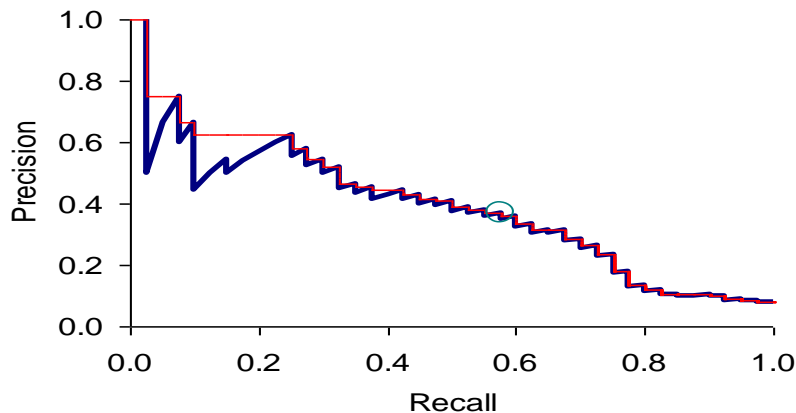


Precision \ Recall

- You can get high recall but low precision
- Recall is a non-decreasing function
- In a good system, precision decreases as either the number of docs retrieved or recall increases
 - This is not a theorem, but a result with strong empirical confirmation

查准率 $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$

查全率 $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$





MIT-BIH心拍分类数据集

心拍类型	DS1	DS2	合计
N	45868	44258	89723
S	942	1837	2773
V	3787	3221	6986
F	415	388	801
Q	8	7	15
合计	51020	49711	100731



心拍分类结果

		N	S	V	F	Q	合计
	N	43910	146	155	24	23	44258
标	S	1693	76	64	4	0	1837
注	V	171	10	3025	15	0	3221
类	F	338	1	49	0	0	388
别	Q	3	1	3	0	0	7
	合计	46115	234	3296	43	23	49711

室性异搏心拍（V）的灵敏度为93.91%，正预测率为91.78%。



A combined measure: F

- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

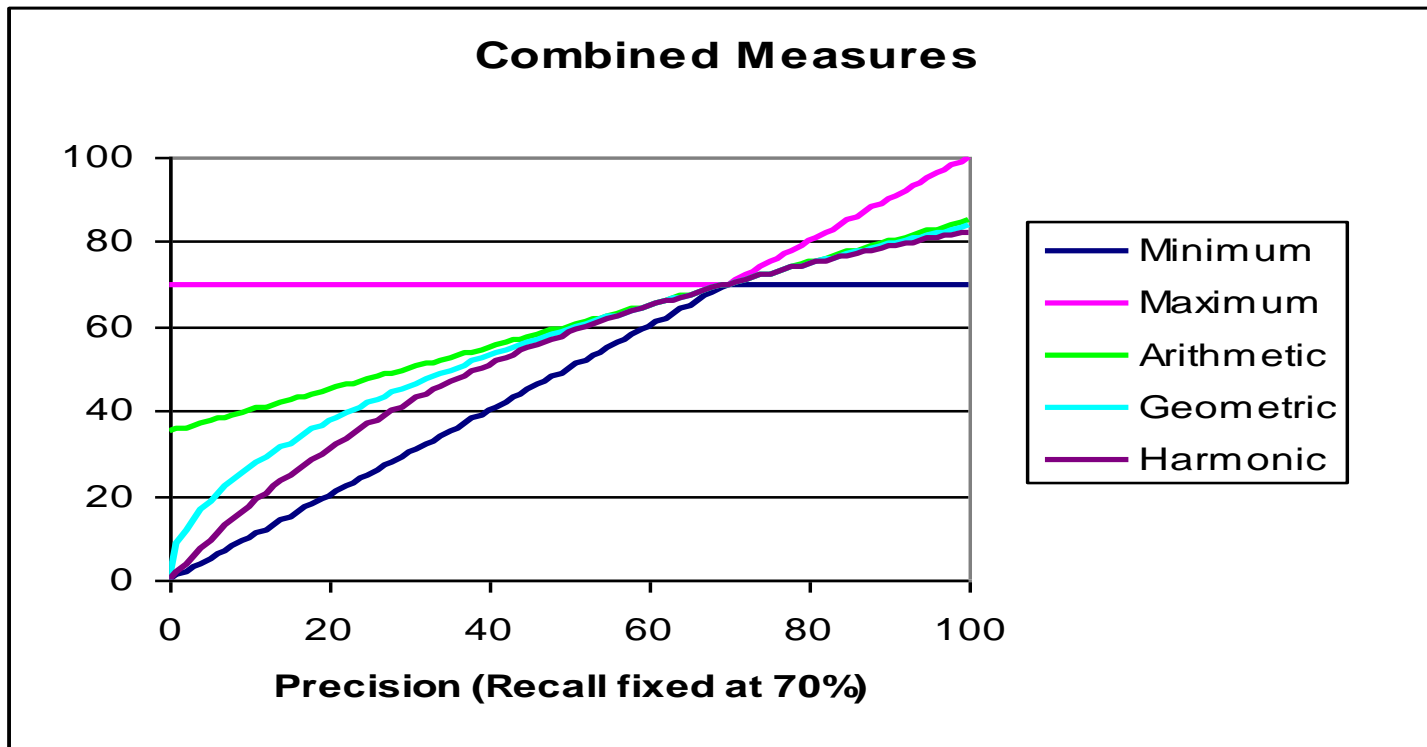
$$F\text{-score}(\beta) = \frac{(1 + \beta^2) PR}{\beta^2 P + R}$$

$$F\text{-score}(\beta) = \frac{(1 + \beta^2) tp}{(1 + \beta^2) tp + \beta^2 fp + fn}$$

- People usually use balanced F_1 measure
 - i.e., with $\beta = 1$ or $\alpha = 1/2$
- Harmonic mean is a conservative average
 - See CJ van Rijsbergen, *Information Retrieval*



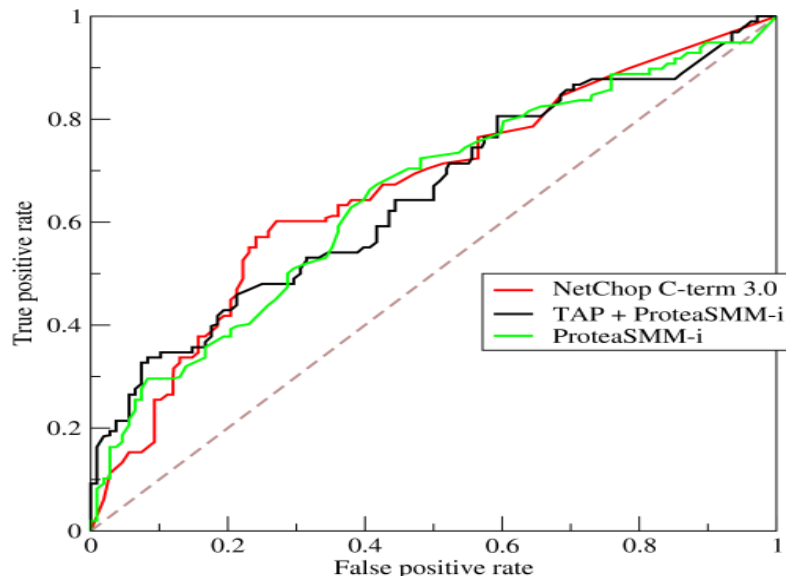
F_1 and other averages





ROC和AUC

- ROC全称是“受试者工作特征” (Receiver Operating Characteristic)。ROC曲线的面积就是AUC (Area Under the Curve)。AUC用于衡量“**二分类问题**”机器学习算法性能 (泛化能力)。
- ROC曲线横轴为FPR，纵轴为TPR，直观的展示FPR与TPR之间的对应关系。
- 当测试集中的正负样本的分布变化的时候，ROC曲线能够保持不变。





Micro-average vs. Macro-average

Let $tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda, f_\lambda$ be the number of true positives, false positives, true negatives, false negatives and relative frequency for label λ , then

- Macro-averaged result is defined as

$$B_{macro} = \frac{1}{q} \sum_{\lambda=1}^q B(tp_\lambda, fp_\lambda, tp_\lambda, fn_\lambda)$$

多分类

Macro-average precision = (P1+P2)/2

- Macro-averaged result is defined as

$$B_{micro} = B\left(\sum_{\lambda=1}^q tp_\lambda, \sum_{\lambda=1}^q fp_\lambda, \sum_{\lambda=1}^q fn_\lambda, \sum_{\lambda=1}^q tn_\lambda\right)$$

Micro-average of precision = (TP1+TP2)/(TP1+TP2+FP1+FP2)



Micro-average vs. Macro-average

Let $tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda, f_\lambda$ be the number of true positives, false positives, true negatives, false negatives and relative frequency for label λ , then

- Label-frequency-based micro-averaged

$$B_{lfb} = \sum_{\lambda=1}^q f_\lambda B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda)$$

c_λ = number of instances in training with class label λ

n = number of instances in training

$$f_\lambda = \frac{c_\lambda}{n}$$



Example

label λ	tp	fp	fn	precision	recall	f_{λ}^{test}	f_{λ}^{train}
c_1	3	2	7	0.6	0.3	0.25	0.1
c_2	1	7	9	0.125	0.1	0.25	0.2
c_3	6	6	8	0.5	0.429	0.35	0.3
c_4	6	9	0	0.4	1	0.15	0.4
total	16	24	24			1	1
macro-averaged		0.406	0.457				
micro-averaged		0.4	0.4				
lfb-micro-averaged		0.395	0.579				

two versions of the micro-averaged precision and recall differ.



■ Macro-averaging vs. Micro-averaging

- Macro-averaged gives equal weight to each class
- Micro-averaged gives equal weight to each per-instance classification decision
- Macro-averaging is a good choice when you get a sense of effectiveness on small classes
- Micro-averaging is a good choice on the large classes because large classes dominate small classes in micro-averaging
- Macro-averaging evaluates the system performance overall across the sets of data, can not get any specific decision with it
- Micro-average can be a useful measure when the dataset varies in size



目录

- 8.1 分类及相关定义
- 8.2 从归类理论到经典分类理论
 - 8.2.1 PAC理论
 - 8.2.2 统计机器学习理论
- 8.3 分类测试公理
- 8.4 分类结果评价
- 8.5 拓展应用
- 8.6 作业



拓展1：经验风险最小化与分类公平性

- **问题：**如何有效保证敏感信息不影响经验模型的结果？

问题举例：确定是否应该贷款给一个人的模型，最好不使用诸如人种、性别等敏感信息。

- **拓展：Fair Empirical Risk Minimization**

Michele等[1]提出了一种**基于经验风险最小化**的方法，该方法**将公平性约束**纳入了学习问题。它促使学习的分类器的条件风险相对于敏感变量近似为常数。

$$\min \left\{ \hat{L}(f) : f \in \mathcal{F}, |\hat{L}^{+,a}(f) - \hat{L}^{+,b}(f)| \leq \hat{\epsilon} \right\}$$

由“来自Group-a(比如男性)的正例样本产生的Loss”与“来自Group-b(比如女性)的正例样本产生的Loss”之间的差异小于预先确定的上界 $\hat{\epsilon}$

Michele Donini, Luca Oneto, Shai Ben-David, John S. Shawe-Taylor, Massimiliano Pontil. Empirical Risk Minimization Under Fairness Constraints[C]. Advances in Neural Information Processing Systems 31 (NIPS 2018)



拓展2: PAC-Bayes理论

- **问题:** VC维理论能定量地度量样本空间的复杂性, 但**不能利用先验信息**。考虑单一分类器无法保证结果的不确定性, 为保证学习到的分类器具备泛化能力, 引入**系列分类器**。
- **拓展:** 现代学习理论将 PAC 学习和贝叶斯推理有机地结合在一起, 产生了 PAC-Bayes 学习理论。PAC-Bayes 边界是 “Occam’s razor” 边界的一个推广。与传统的学习算法不同, PAC-Bayes 边界输出的是一**系列分类器的分布, 而不是单个的分类器**。



拓展2: PAC-Bayes理论

(Germain(2009)定理 2. 1) 若由概念空间 C 中的概念 c 得到的样本服从任意分布 D , 对任意先验分布为 P 的假设空间 H , 对任意 $\delta \in (0, 1]$ 以及任意凸函数 $\Psi: [0, 1] \times [0, 1] \rightarrow \mathbb{R}$, 有:

$$Pr_{S \sim D^m} \left(\forall Q \text{ on } H: \Psi(R_S(G_Q), R(G_Q)) \leq \frac{1}{m} \left[KL(Q \| P) + \ln \left(\frac{1}{\delta} E_{S \sim D^m} E_{h \sim Q} e^{m\Psi(R_S(h), R(h))} \right) \right] \right) \geq 1 - \delta, \quad (11)$$

其中, $KL(Q \| P) = E_{h \sim Q} \ln \frac{Q(h)}{P(h)}$.

- P : 分类器 h 的真实分布; Q : 分类器 h 的经验分布;
- G_Q : 由于贝叶斯最优分类器算法开销大并且可能得到不属于假设空间中的假设, 因此人们设计了另外一种常用的近似最优分类器来替代, 即Gibbs分类器。该分类器在得到 Q 之后不采用加权处理, 而是通过采样的方式依概率分布 Q 随机地选择一个分类器 h 。
- $R_S(G_Q)$: Gibbs分类器的经验误差; $R(G_Q)$: Gibbs分类器的真实误差;
- 形式类似 $Pr(R(h) \leq \epsilon) \geq 1 - \delta$
- Ψ : 一个用来表示一切形式的“误差”的凸函数, 当该函数取不同形式时, 可以得到不同的界
- PAC-Bayesian与传统机器学习结合的核心思路就是选择特定的凸函数 Ψ 并对式 (11) 得到的泛化误差界进行优化



拓展2：PAC-Bayes理论

■ 优点：

- ✓ 对于连续分类器空间在实践中比大部分 VC 维相关的边界更紧
- ✓ 基于这种更紧的边界，能够设计更好的分类算法，有效避免过拟合
- ✓ 基于 PAC-Bayes 边界推导的学习算法实质上是对类别假设空间的“平均”，因而能够获得更好的分类性能



应用1：分类理论与深度学习

- 关于**深度神经网络的泛化误差界**分析，Zhang等[1]通过实验结果指出传统基于VC维和Rademacher复杂度的泛化误差分析**无法解释**深度神经网络中参数数目远远大于样本数目却仍具有良好泛化性这一现象。
- Arpit等[2]进一步指出分析**深度神经网络的泛化误差界**不能简单考虑深度神经网络理论上所能表达的假设空间复杂度，而**需结合**深度神经网络采用的**优化算法及训练数据**，考虑分析深度神经网络所能优化假设构成的**假设空间的复杂度**。

[1]Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals. Understanding deep learning requires rethinking generalization[C]. ICLR 2017

[2] *Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML' 17). JMLR.org, 233–242.*



- 8.1 分类及相关定义
- 8.2 从归类理论到经典分类理论
 - 8.2.1 PAC理论
 - 8.2.2 统计机器学习理论
- 8.3 分类测试公理
- 8.4 分类结果评价
- 8.5 拓展应用
- 8.6 作业



作业

1. 求下表中预测类别的准确率、召回率和 F_1 得分；当阈值设定为 $\text{score} > 2$ 时计算拒绝无效假设的准确率、召回率和 F_1 得分；当 $\text{score} > -1$ 时计算拒绝无效假设的准确率、召回率和 F_1 得分

ID	Score	Predicted Class	True Class
1	-4.80	-	-
2	-4.43	-	-
3	-2.09	-	-
4	-1.30	-	-
5	-0.53	-	+
6	-0.30	-	+
7	0.49	+	-
8	0.98	+	-
9	2.25	+	+
10	3.37	+	+
11	4.03	+	+
12	4.90	+	+



作业

2. 通过经验风险最小化推导极大似然估计：证明模型是条件概率分布，损失函数是对数损失函数时，经验风险最小化等价于极大似然估计。

3. 基于某用于诊断病人是否患病的黑箱模型，预测得到100个样本0~1之间的患病分数。调节阈值可得到不同的诊断结果（见下表）：随着阈值从0.0逐步增加到1.0，会有越来越多的样本被判断为正常人。试根据该表格画出ROC曲线并计算去线下面积AUC。

Threshold	TP	FP	TN	FN
0.0	50	50	0	0
0.1	48	47	3	2
0.2	47	40	9	4
0.3	45	31	16	8
0.4	44	23	22	11
0.5	42	16	29	13
0.6	36	12	34	18
0.7	30	11	38	21
0.8	20	4	43	33
0.9	12	3	45	40
1.0	0	0	50	50



北京交通大学《机器学习》课程组成员

- 于 剑: jianyu@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/6463/>
- 景丽萍: lpjing@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/8249/>
- 田丽霞: lxtian@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/7954/>
- 黄惠芳: hfhuang@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/7418/>
- 杨 凤: fengyang@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/8518/>
- 吴 丹: wudan@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/8925/>
- 万怀宇: hywan@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/8793/>
- 王 晶: wj@bjtu.edu.cn, <http://faculty.bjtu.edu.cn/9167/>