华中科技大学
电子信息与通信学院
School of Electronic Information and Communications

机器学习导论

– Introduction to Machine Learning

第 3 章：模型评估

(Chapter 3: Model Evaluation)

华中科技大学电信学院

王邦　博士, 教授博导

wangbang@hust.edu.cn

第 03 章：模型评估

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。
- **输入空间** $\mathcal{X}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; ...x_{id})$ 由 d 个属性描述，x_{ij} 是第 j 个属性上的取值（可以是连续或离散），d 称为样本 $\boldsymbol{x}_i$ 的维数。可以将 $\boldsymbol{x}_i$ 看做是 d 维空间 $\mathcal{X}$ 中的一个向量，$\boldsymbol{x}_i \in \mathcal{X}$。

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。
- **输入空间** $\mathcal{X}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; ...x_{id})$ 由 d 个属性描述，x_{ij} 是第 j 个属性上的取值（可以是连续或离散），d 称为样本 $\boldsymbol{x}_i$ 的维数。可以将 $\boldsymbol{x}_i$ 看做是 d 维空间 $\mathcal{X}$ 中的一个向量，$\boldsymbol{x}_i \in \mathcal{X}$。
- **输出空间** $\mathcal{Y}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 y_i 称之为 $\boldsymbol{x}_i$ 的标记，令 $\mathcal{Y}$ 为所有标记的集合，即输出空间。

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。
- **输入空间** $\mathcal{X}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; ...x_{id})$ 由 d 个属性描述，x_{ij} 是第 j 个属性上的取值（可以是连续或离散），d 称为样本 $\boldsymbol{x}_i$ 的维数。可以将 $\boldsymbol{x}_i$ 看做是 d 维空间 $\mathcal{X}$ 中的一个向量，$\boldsymbol{x}_i \in \mathcal{X}$。
- **输出空间** $\mathcal{Y}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 y_i 称之为 $\boldsymbol{x}_i$ 的标记，令 $\mathcal{Y}$ 为所有标记的集合，即输出空间。

机器学习：从训练数据中学习某个“模型”，即寻找从输入空间到输出空间的某种映射：$f : \mathcal{X} \mapsto \mathcal{Y}$，从而利用该模型对新的测试数据进行预测（回归）或分类（聚类）。

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。
- **输入空间** $\mathcal{X}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; ...x_{id})$ 由 d 个属性描述，x_{ij} 是第 j 个属性上的取值（可以是连续或离散），d 称为样本 $\boldsymbol{x}_i$ 的维数。可以将 $\boldsymbol{x}_i$ 看做是 d 维空间 $\mathcal{X}$ 中的一个向量，$\boldsymbol{x}_i \in \mathcal{X}$。
- **输出空间** $\mathcal{Y}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 y_i 称之为 $\boldsymbol{x}_i$ 的标记，令 $\mathcal{Y}$ 为所有标记的集合，即输出空间。

机器学习：从训练数据中学习某个“模型”，即寻找从输入空间到输出空间的某种映射：$f : \mathcal{X} \mapsto \mathcal{Y}$，从而利用该模型对新的测试数据进行预测（回归）或分类（聚类）。

不同的学习算法可以学得“不同的模型”，那么如何评估“不同的模型”呢？

回顾

- **训练数据** $(\boldsymbol{x}_i, y_i)$：令 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 表示包含 m 个样本的数据集，其中 $(\boldsymbol{x}_i, y_i)$ 表示第 i 个样本。
- **输入空间** $\mathcal{X}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; ...x_{id})$ 由 d 个属性描述，x_{ij} 是第 j 个属性上的取值（可以是连续或离散），d 称为样本 $\boldsymbol{x}_i$ 的维数。可以将 $\boldsymbol{x}_i$ 看做是 d 维空间 $\mathcal{X}$ 中的一个向量，$\boldsymbol{x}_i \in \mathcal{X}$。
- **输出空间** $\mathcal{Y}$：第 i 个样本 $(\boldsymbol{x}_i, y_i)$ 中的 y_i 称之为 $\boldsymbol{x}_i$ 的标记，令 $\mathcal{Y}$ 为所有标记的集合，即输出空间。

机器学习：从训练数据中学习某个“模型”，即寻找从输入空间到输出空间的某种映射：$f : \mathcal{X} \mapsto \mathcal{Y}$，从而利用该模型对新的测试数据进行预测（回归）或分类（聚类）。

不同的学习算法可以学得“不同的模型”，那么如何评估“不同的模型”呢？

机器学习的目标是使得学得的模型能很好地适用于“新样本”，而不仅仅在训练样本上工作得很好。学得模型适用于新样本的能力，称之为泛化能力 (generalization)。

假设空间 (1)

演绎 (deduction) 与归纳 (induction) 是科学推理的两大基本手段。

假设空间 (1)

演绎 (deduction) 与**归纳** (induction) 是科学推理的两大基本手段。

- **演绎：** 从一般到特殊的“**特化**” (specialization) 过程，即从基础原理推演出具体状况。例如在数学公理系统中，基于一组公理和推理规则推导出与之相洽的定理。

假设空间 (1)

演绎 (deduction) 与归纳 (induction) 是科学推理的两大基本手段。

- **演绎：** 从一般到特殊的“特化” (specialization) 过程，即从基础原理推演出具体状况。例如在数学公理系统中，基于一组公理和推理规则推导出与之相洽的定理。
- **归纳：** 从特殊到一般的“泛化” (generalization) 过程，即从具体的事实归纳出一般性规律。机器学习是从“样例中进行学习”，显然是一个归纳的过程，亦称为“归纳学习” (inductive learning)。

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

（来源：周志华，《机器学习》，清华大学出版社）

例如，西瓜是否为“好瓜”由三个属性“色泽”，“根蒂”和“敲声”的取值所确定。归纳学习的目标是从样例中学得一个判断西瓜好坏的模型，可以表示为：

好瓜 $\leftrightarrow$(色泽 =？)$\wedge$(根蒂 =？)$\wedge$(敲声 =？)，

其中“？”为尚未确定的取值，$\wedge$ 为布尔运算**"and"**。

假设空间 (1)

演绎 (deduction) 与归纳 (induction) 是科学推理的两大基本手段。

- **演绎：** 从一般到特殊的“特化” (specialization) 过程，即从基础原理推演出具体状况。例如在数学公理系统中，基于一组公理和推理规则推导出与之相洽的定理。
- **归纳：** 从特殊到一般的“泛化” (generalization) 过程，即从具体的事实归纳出一般性规律。机器学习是从“样例中进行学习”，显然是一个归纳的过程，亦称为“归纳学习” (inductive learning)。

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

（来源：周志华，《机器学习》，清华大学出版社）

例如，西瓜是否为“好瓜”由三个属性“色泽”，“根蒂”和“敲声”的取值所确定。归纳学习的目标是从样例中学得一个判断西瓜好坏的模型，可以表示为：

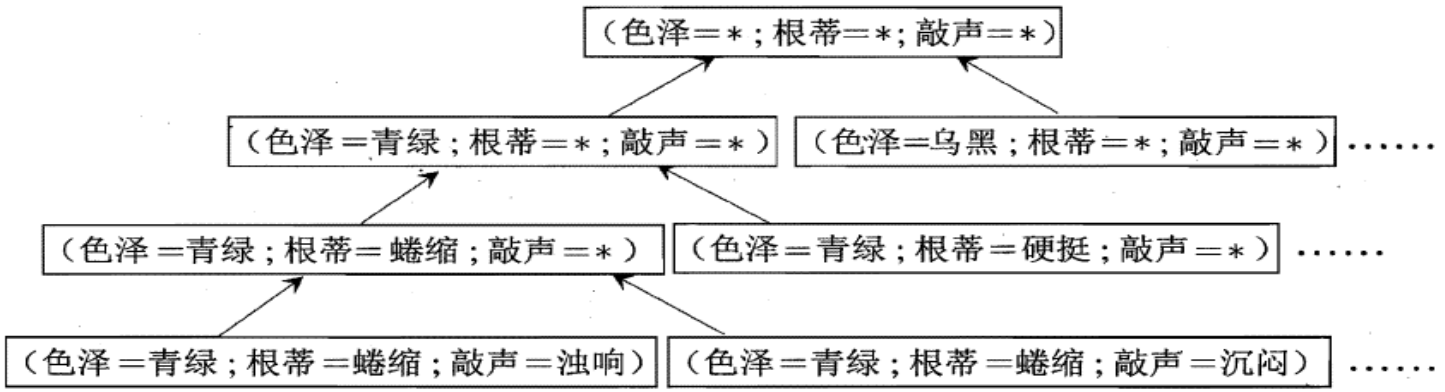
好瓜 $\leftrightarrow$(色泽 =？)$\wedge$(根蒂 =？)$\wedge$(敲声 =？)，

其中“？”为尚未确定的取值，$\wedge$ 为布尔运算**"and"**。

机器学习的过程可以看做是一个在所有“假设” (hypothesis) 组成的空间中进行**搜索**的过程，搜索目标是找到与训练集“匹配” (fit) 的假设。如果能够确定各种假设的具体表示，则可以确定相应的假设空间及其规模。

假设空间 (2)

例子中的假设空间由形如 (色泽 =?)∧(根蒂 =?)∧(敲声 =?)的所有可能取值所形成的假设组成。例如色泽有“青绿”，“乌黑”，“浅白”三种可能取值；此外，还需考虑也许“色泽”无论取什么值都合适，用通配符“*”表示，如好瓜 ↔(色泽 =*)∧(根蒂 =?)∧(敲声 =?)，即“好瓜是根蒂蜷缩、敲声浊响的瓜，什么色泽都行”。此外，还需考虑极端情况：有可能“好瓜”这个概念根本就不成立，世界上没有“好瓜”这种东西；用 ∅ 表示这个假设。若“色泽”，“根蒂”，“敲声”分别有 3,2,2 种可能取值，则假设空间的规模大小为 $4 \times 3 \times 3 + 1 = 37$；下图直观地表示出了这个西瓜问题的假设空间。

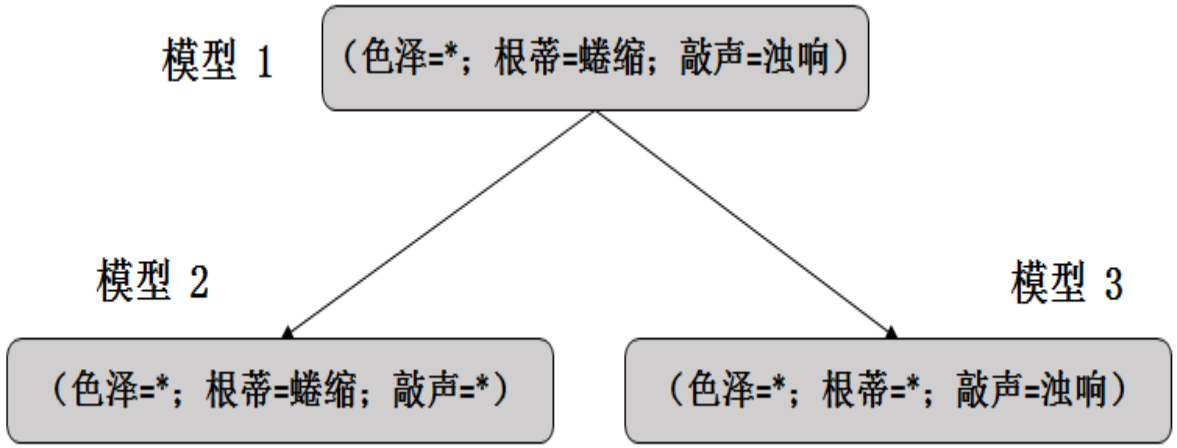


西瓜问题的假设空间　（来源：周志华，《机器学习》，清华大学出版社）

可以有许多策略对这个假设空间进行搜索，例如自顶向下、从一般到特殊，或是自底向上、从特殊到一般，搜索过程中可以不断删除与正例不一致的假设、和 (或) 与反例一致的假设。最终将会获得与训练集一致 (即对所有训练样本能够进行正确判断) 的假设，这就是我们学得的结果.

归纳偏好 (1)

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

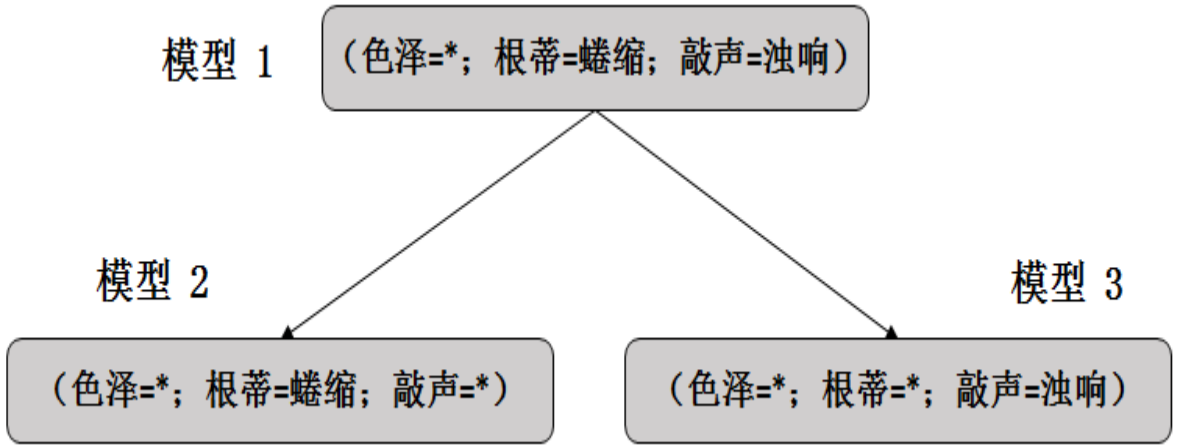


这个例子中，如果学习过程是基于有限样本训练集进行，可能学习得到多个不同的模型，且每个模型对应的假设都与训练集一致。如上图所示的三个模型都可以从训练集学习得到。但它们在面临新样本的时候，却可能产生不同的输出。

例如，对于新样本 **(色泽 = 青绿；根蒂 = 蜷缩；敲声 = 沉闷)**。如果采用模型 2：**好瓜 ↔(色泽 =*)∧(根蒂 = 蜷缩)∧(敲声 =*)**，则会将新瓜判断为好瓜；而如果采用其他两个假设，则判断的结果将不是好瓜。

归纳偏好 (1)

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



这个例子中，如果学习过程是基于有限样本训练集进行，可能学习得到多个不同的模型，且每个模型对应的假设都与训练集一致。如上图所示的三个模型都可以从训练集学习得到。但它们在面临新样本的时候，却可能产生不同的输出。

例如，对于新样本 **(色泽 = 青绿；根蒂 = 蜷缩；敲声 = 沉闷)**。如果采用模型 2：**好瓜 ↔(色泽 =*)∧(根蒂 = 蜷缩)∧(敲声 =*)**，则会将新瓜判断为好瓜；而如果采用其他两个假设，则判断的结果将不是好瓜。

应该采用哪一个模型呢？

归纳偏好 (2)

如果仅有例子中的训练样本，无法断定上述三个模型哪一个“更好”。但对于一个具体的学习算法而言，其必须要产生一个模型。这时，学习算法本身的“偏好”就会起到关键的作用。

归纳偏好 (2)

如果仅有例子中的训练样本，无法断定上述三个模型哪一个“更好”。但对于一个具体的学习算法而言，其必须要产生一个模型。这时，学习算法本身的“偏好”就会起到关键的作用。

- 如果算法喜欢“尽可能特殊”的模型，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 = 浊响)。

归纳偏好 (2)

如果仅有例子中的训练样本，无法断定上述三个模型哪一个“更好”。但对于一个具体的学习算法而言，其必须要产生一个模型。这时，学习算法本身的“偏好”就会起到关键的作用。

- 如果算法喜欢“尽可能特殊”的模型，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 = 浊响)。
- 如果算法喜欢“尽可能一般”的模型，并且由于某种原因它更“相信”根蒂，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 =*)。

归纳偏好 (2)

如果仅有例子中的训练样本，无法断定上述三个模型哪一个“更好”。但对于一个具体的学习算法而言，其必须要产生一个模型。这时，学习算法本身的“偏好”就会起到关键的作用。

- 如果算法喜欢“尽可能特殊”的模型，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 = 浊响)。
- 如果算法喜欢“尽可能一般”的模型，并且由于某种原因它更“相信”根蒂，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 =*)。

机器学习算法在学习过程中对某种类型假设的偏好，称之为“归纳偏好”(inductive bias)，或简称为“偏好”。

归纳偏好 (2)

如果仅有例子中的训练样本，无法断定上述三个模型哪一个“更好”。但对于一个具体的学习算法而言，其必须要产生一个模型。这时，学习算法本身的“偏好”就会起到关键的作用。

- 如果算法喜欢“尽可能特殊”的模型，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 = 浊响)。
- 如果算法喜欢“尽可能一般”的模型，并且由于某种原因它更“相信”根蒂，则会产生模型：好瓜 $\leftrightarrow$(色泽 =*)$\wedge$(根蒂 = 蜷缩)$\wedge$(敲声 =*)。

> 机器学习算法在学习过程中对某种类型假设的偏好，称之为“归纳偏好”(inductive bias)，或简称为“偏好”。

任何一个有效的机器学习算法必有其归纳偏好，否则它将被假设空间中看似在训练集上" 等效" 的假设所迷惑，而无法产生确定的学习结果。可以想象，如果没有偏好，我们的西瓜学习算法产生的模型每次在进行预测时随机抽选训练集上的等效假设，那么对这个新瓜"(色泽 = 青绿; 根蒂口蜷缩; 敲声 = 沉闷)" ，学得模型时而告诉我们它是好的、时而告诉我们它是不好的，这样的学习结果显然没有意义。

奥卡姆剃刀原则

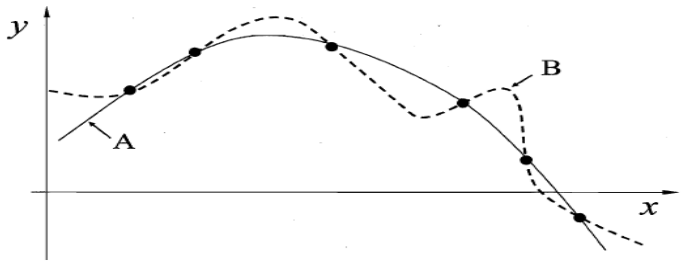
归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的“价值观”。有没有一般性的原则来引导算法确立“正确的”偏好呢?

奥卡姆剃刀原则

归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的“价值观”。有没有一般性的原则来引导算法确立“正确的”偏好呢?

奥卡姆剃刀原则 (Occam's razor)：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。同时也需注意到，奥卡姆剃刀本身存在不同的诠释，使用奥卡姆剃刀原则并不平凡。



存在多条曲线与有限样本训练集一致

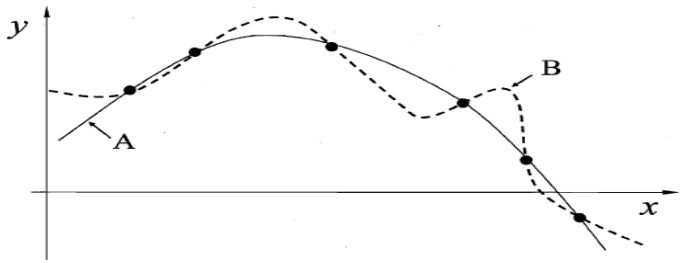
(来源：周志华，《机器学习》，清华大学出版社)

奥卡姆剃刀原则

归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的“价值观”。有没有一般性的原则来引导算法确立“正确的”偏好呢?

奥卡姆剃刀原则 (Occam's razor)：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。同时也需注意到，奥卡姆剃刀本身存在不同的诠释，使用奥卡姆剃刀原则并不平凡。



存在多条曲线与有限样本训练集一致

(来源：周志华，《机器学习》，清华大学出版社)

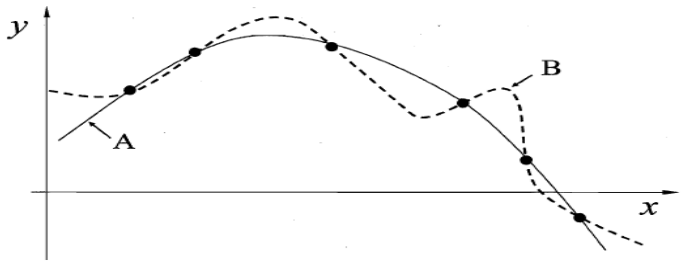
图中，曲线 A 和曲线 B 均能拟合有限的训练样本。根据奥卡姆剃刀原则，并且假设“更平滑”意味着“更简单”(例如曲线 A 更易于描述，其方程式是 $y = -x^2 + 6x + 1$，而曲线 B 则要复杂得多)，则我们更偏好“平滑”的曲线 A。

奥卡姆剃刀原则

归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的“价值观”。有没有一般性的原则来引导算法确立“正确的”偏好呢?

奥卡姆剃刀原则 (Occam's razor)：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。同时也需注意到，奥卡姆剃刀本身存在不同的诠释，使用奥卡姆剃刀原则并不平凡。



存在多条曲线与有限样本训练集一致

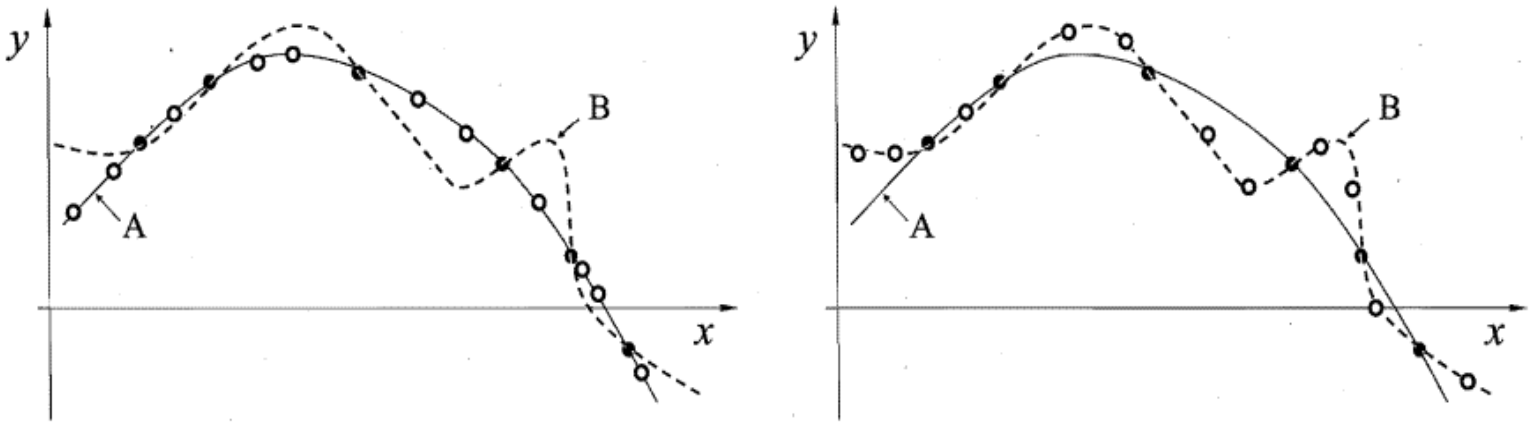
(来源：周志华，《机器学习》，清华大学出版社)

图中，曲线 A 和曲线 B 均能拟合有限的训练样本。根据奥卡姆剃刀原则，并且假设“更平滑”意味着“更简单”(例如曲线 A 更易于描述，其方程式是 $y = -x^2 + 6x + 1$，而曲线 B 则要复杂得多)，则我们更偏好“平滑”的曲线 A。

归纳偏好对应了学习算法本身所做出的关于“什么样的模型更好”的假设。但在具体的现实问题中，这个假设是否成立，即算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能。

没有免费的午餐 (1)

归纳偏好与具体问题有关: 在下图中，学习算法 $\mathfrak{L}_a$ 基于某种归纳偏好产生了曲线 A 的模型，学习算法 $\mathfrak{L}_b$ 基于另一种归纳偏好产生了曲线 B 的模型。如果基于奥卡姆剃刀原则，假设平滑曲线具有某种“描述简单性”，则我们采用曲线 A 作为学习到的模型。在图 (a) 中，我们发现曲线 A 与训练集外的样本更一致，其泛化能力强于曲线 B。但是，在图 (b) 中，我们却发现曲线 B 与训练集外的样本更一致，其泛化能力强于曲线 A。



(a) A 优于 B　　(b) B 优于 A

没有免费的午餐. (黑点: 训练样本; 白点: 测试样本)

(来源：周志华，《机器学习》，清华大学出版社)

注意到图 (a) 与图 (b) 中是两个不同的问题，他们的训练集和测试集是不一样的。因此，脱离开问题本身讨论算法性能是没有意义的。

没有免费的午餐 (2)

假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 为离散空间。令 $P(h|\boldsymbol{X}, \mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 $\boldsymbol{X}$ 产生假设 h 的概率；令 f 代表希望学习得到的真实目标函数。$\mathfrak{L}_a$ 的“训练集外误差”定义为：

$$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x})\mathbb{I}(h(\boldsymbol{x} \neq f(\boldsymbol{x}))P(h|\boldsymbol{X}, \mathfrak{L}_a),$$

其中 $\mathbb{I}(z)$ 为指示函数，若 z 为真取值 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 $\{0, 1\}^{|\mathcal{X}|}$。则对所有可能的 f 按均匀分布对误差求和，有：

$$\sum_f \mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x}) \cdot 1.$$

（证明略）上式表明，总误差与具体的学习算法无关！对于任意的两个学习算法 $\mathfrak{L}_a$ 和 $\mathfrak{L}_b$，都有：$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \mathbb{E}_{ote}(\mathfrak{L}_b|\boldsymbol{X}, f)$，即两个学习算法的期望性能相同。

没有免费的午餐 (2)

假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 为离散空间。令 $P(h|\boldsymbol{X}, \mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 $\boldsymbol{X}$ 产生假设 h 的概率；令 f 代表希望学习得到的真实目标函数。$\mathfrak{L}_a$ 的“训练集外误差”定义为：

$$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x})\mathbb{I}(h(\boldsymbol{x} \neq f(\boldsymbol{x}))P(h|\boldsymbol{X}, \mathfrak{L}_a),$$

其中 $\mathbb{I}(z)$ 为指示函数，若 z 为真取值 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 $\{0, 1\}^{|\mathcal{X}|}$。则对所有可能的 f 按均匀分布对误差求和，有：

$$\sum_f \mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x}) \cdot 1.$$

（证明略）上式表明，总误差与具体的学习算法无关！对于任意的两个学习算法 $\mathfrak{L}_a$ 和 $\mathfrak{L}_b$，都有：$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \mathbb{E}_{ote}(\mathfrak{L}_b|\boldsymbol{X}, f)$，即两个学习算法的期望性能相同。

“没有免费的午餐定理” (No Free Lunch Theorems，NFL)：针对某一域的所有问题，所有算法的期望性能是相同的。(Wolpert and Macready, 1997)

没有免费的午餐 (2)

假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 为离散空间。令 $P(h|\boldsymbol{X}, \mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 $\boldsymbol{X}$ 产生假设 h 的概率；令 f 代表希望学习得到的真实目标函数。$\mathfrak{L}_a$ 的“训练集外误差”定义为：

$$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x})\mathbb{I}(h(\boldsymbol{x} \neq f(\boldsymbol{x}))P(h|\boldsymbol{X}, \mathfrak{L}_a),$$

其中 $\mathbb{I}(z)$ 为指示函数，若 z 为真取值 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 $\{0, 1\}^{|\mathcal{X}|}$。则对所有可能的 f 按均匀分布对误差求和，有：

$$\sum_f \mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x}) \cdot 1.$$

（证明略）上式表明，总误差与具体的学习算法无关！对于任意的两个学习算法 $\mathfrak{L}_a$ 和 $\mathfrak{L}_b$，都有：$\mathbb{E}_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \mathbb{E}_{ote}(\mathfrak{L}_b|\boldsymbol{X}, f)$，即两个学习算法的期望性能相同。

> “没有免费的午餐定理”（No Free Lunch Theorems，NFL）：针对某一域的所有问题，所有算法的期望性能是相同的。(Wolpert and Macready, 1997)

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。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、甚至在相似的问题上是否为好方案，我们并不关心。NFL 定理最重要的寓意在于指出脱离具体问题，空泛地谈论“什么学习算法更好”毫无意义。要谈论算法的相对优劣，必须要针对具体的学习问题；在某些问题上表现好的学习算法，在另一些问题上却可能不尽如人意。

经验误差与泛化误差

经验误差与泛化误差

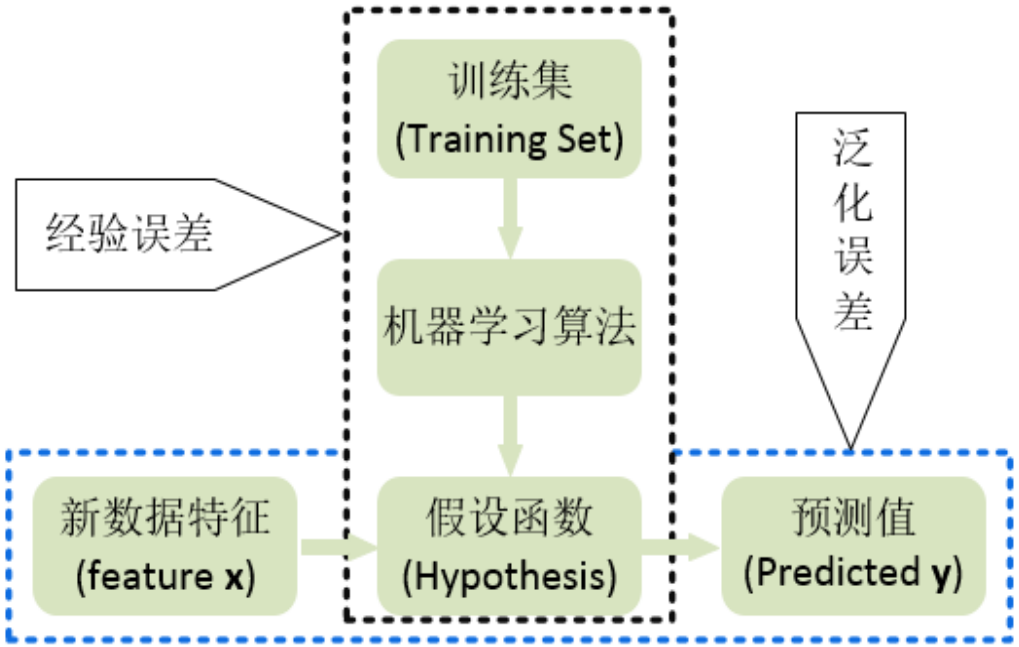
- **误差 (Error)**：学习器的实际预测输出与样本的真实输出之间的差异。

经验误差与泛化误差

- **误差 (Error)**：学习器的实际预测输出与样本的真实输出之间的差异。
- **经验误差 (Empirical Error)**：又称**训练误差 (Training Error)**，指学习器在训练集上的误差。

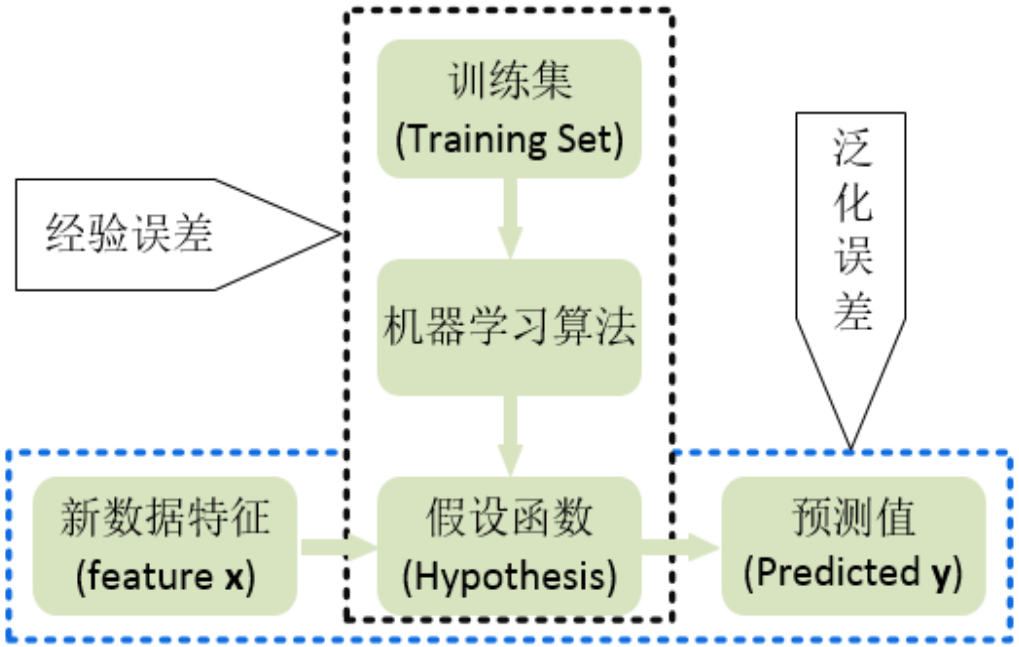
经验误差与泛化误差

- **误差 (Error)**：学习器的实际预测输出与样本的真实输出之间的差异。
- **经验误差 (Empirical Error)**：又称**训练误差 (Training Error)**，指学习器在训练集上的误差。
- **泛化误差 (Empirical Error)**：指学习器在新样本上的误差。



经验误差与泛化误差

- **误差 (Error)**：学习器的实际预测输出与样本的真实输出之间的差异。
- **经验误差 (Empirical Error)**：又称**训练误差 (Training Error)**，指学习器在训练集上的误差。
- **泛化误差 (Empirical Error)**：指学习器在新样本上的误差。



机器学习的目的是希望得到泛化误差小的学习器。

过拟合与欠拟合 (1)

过拟合与欠拟合 (1)

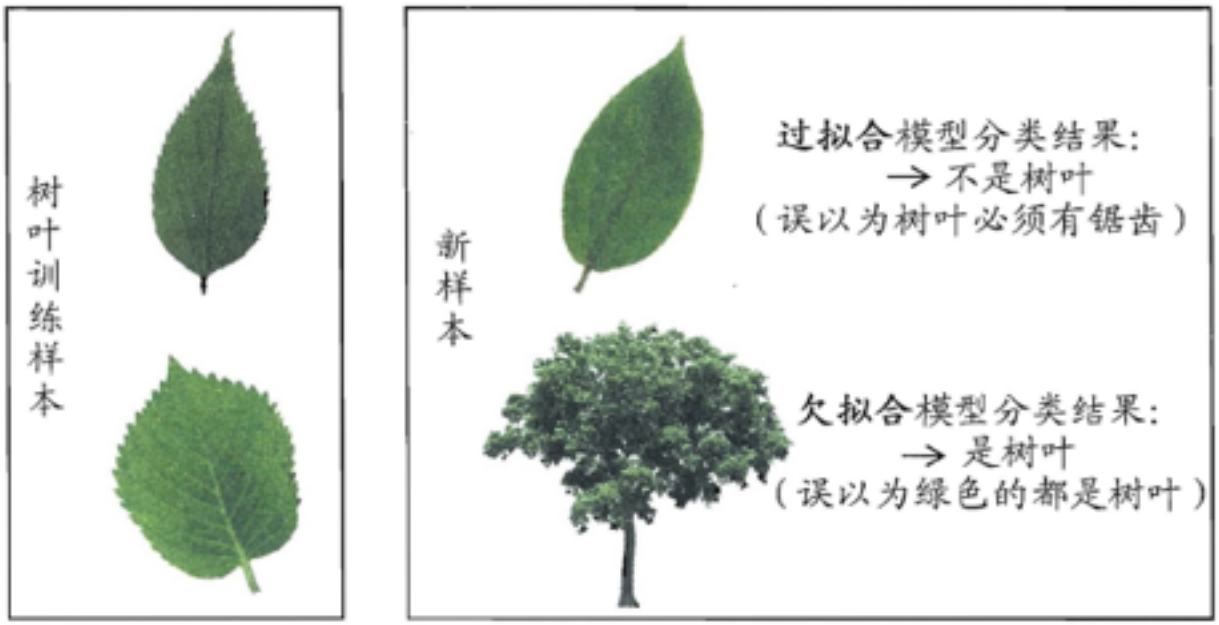
- **过拟合 (Overfitting)**：学习器学习能力过于强大，把训练样本自身的一些特点当作了所有潜在样本都会具有的性质。过拟合会导致泛化性能下降，是机器学习面临的关键障碍。

过拟合与欠拟合 (1)

- **过拟合 (Overfitting)**：学习器学习能力过于强大，把训练样本自身的一些特点当作了所有潜在样本都会具有的性质。过拟合会导致泛化性能下降，是机器学习面临的关键障碍。
- **欠拟合 (Underfitting)**：由于学习能力低下造成的。比较容易克服，比如在决策树中扩展分支、在神经网络中增加训练轮数等。

过拟合与欠拟合 (1)

- **过拟合 (Overfitting)**：学习器学习能力过于强大，把训练样本自身的一些特点当作了所有潜在样本都会具有的性质。过拟合会导致泛化性能下降，是机器学习面临的关键障碍。
- **欠拟合 (Underfitting)**：由于学习能力低下造成的。比较容易克服，比如在决策树中扩展分支、在神经网络中增加训练轮数等。



过拟合、欠拟合的直观类比

来源，周志华，《机器学习》，清华大学出版社

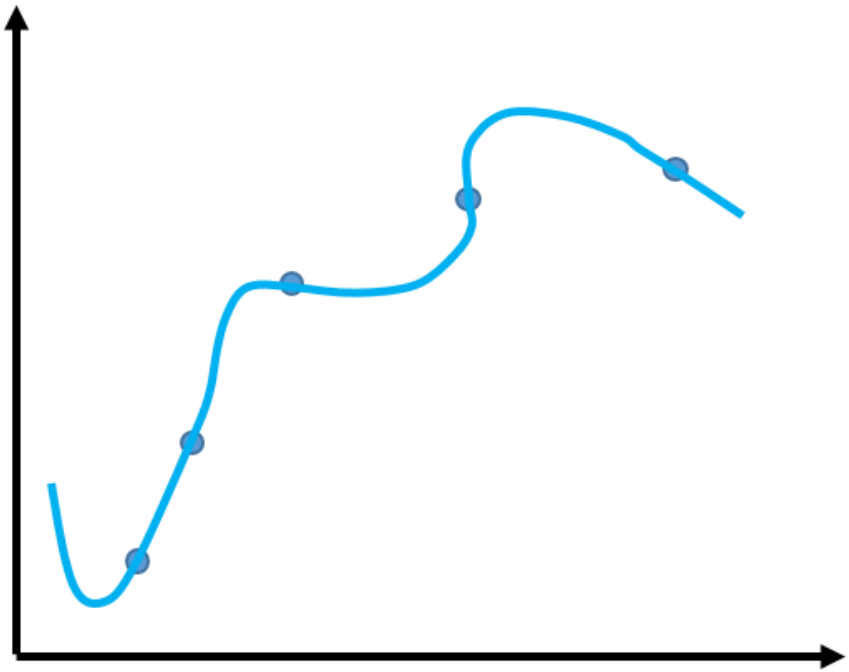
过拟合与欠拟合 (2)

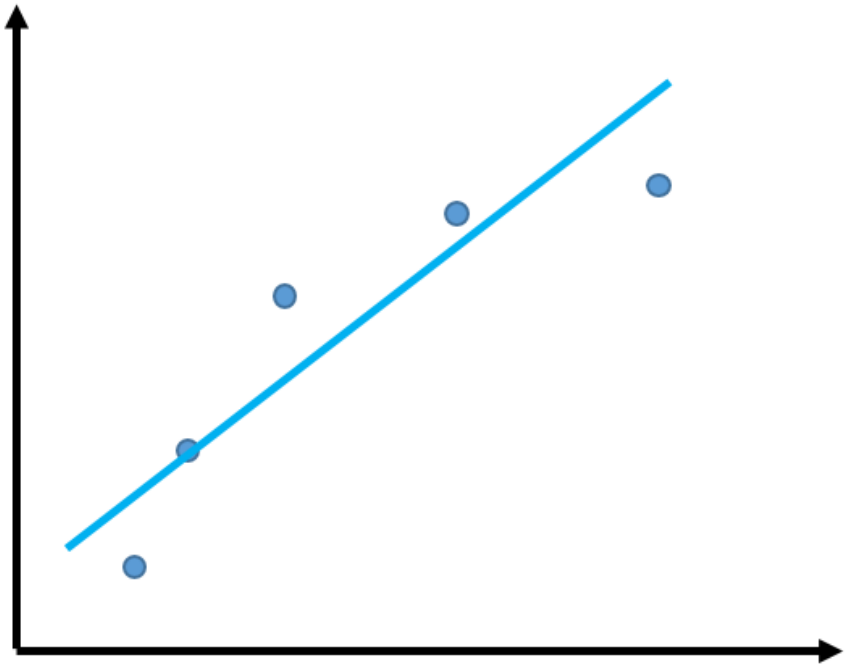
过拟合与欠拟合 (2)

- **过拟合 (Overfitting)：** 当幂次选取过高时，拟合曲线几乎能完美的经过训练集中的每一个点。

过拟合与欠拟合 (2)

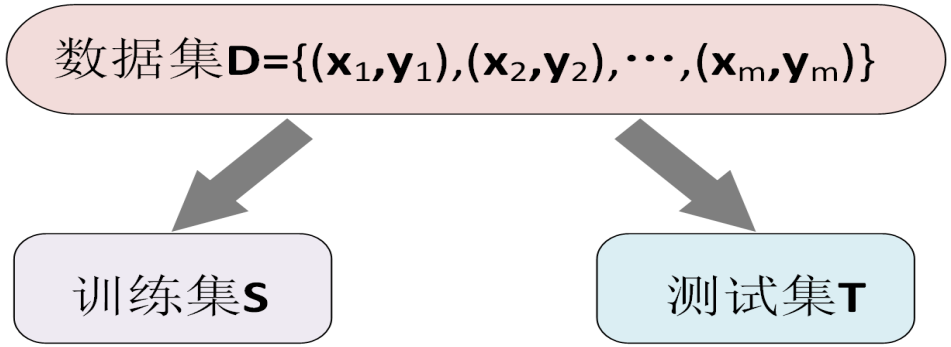
- **过拟合 (Overfitting)：** 当幂次选取过高时，拟合曲线几乎能完美的经过训练集中的每一个点。
- **欠拟合 (Underfitting)：** 当幂次选取过低时，如用线性模型（一次幂）拟合上例。直观上会给人一种拟合不足的感觉。





数据划分

评估一个机器学习模型的好坏关键在于泛化误差的大小，因此需选取一个**测试集**，以测试集上的测试误差作为泛化误差的估计。通常，我们假设测试样本也是从样本真实分布中独立同分布采样而得到的。

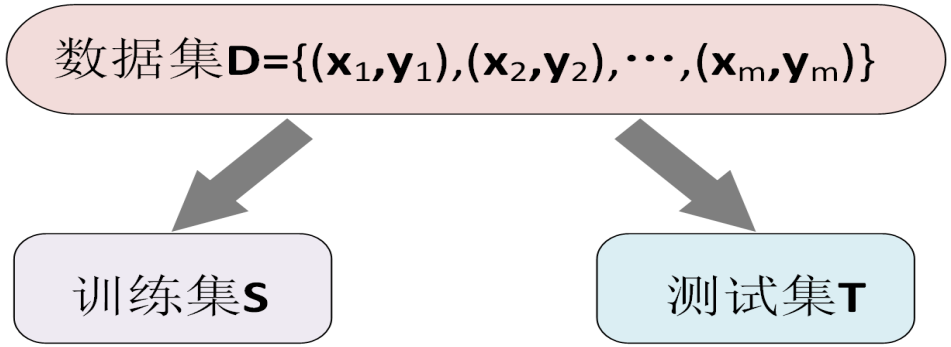


注意：测试集 (T) 应当尽量与训练集 (S) 互斥。即测试样本尽量不在训练集中出现、未在训练过程中使用过。

常用评估方法

数据划分

评估一个机器学习模型的好坏关键在于泛化误差的大小，因此需选取一个**测试集**，以测试集上的测试误差作为泛化误差的估计。通常，我们假设测试样本也是从样本真实分布中独立同分布采样而得到的。



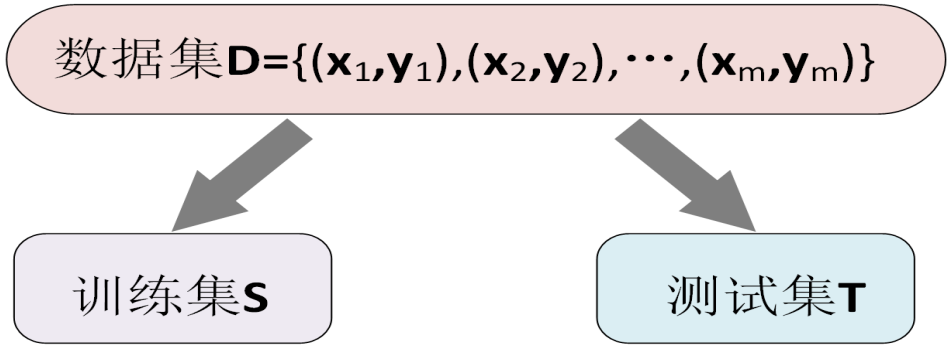
注意：测试集 (T) 应当尽量与训练集 (S) 互斥。即测试样本尽量不在训练集中出现、未在训练过程中使用过。

常用评估方法

- 留出法 (hold-out)

数据划分

评估一个机器学习模型的好坏关键在于泛化误差的大小，因此需选取一个**测试集**，以测试集上的测试误差作为泛化误差的估计。通常，我们假设测试样本也是从样本真实分布中独立同分布采样而得到的。



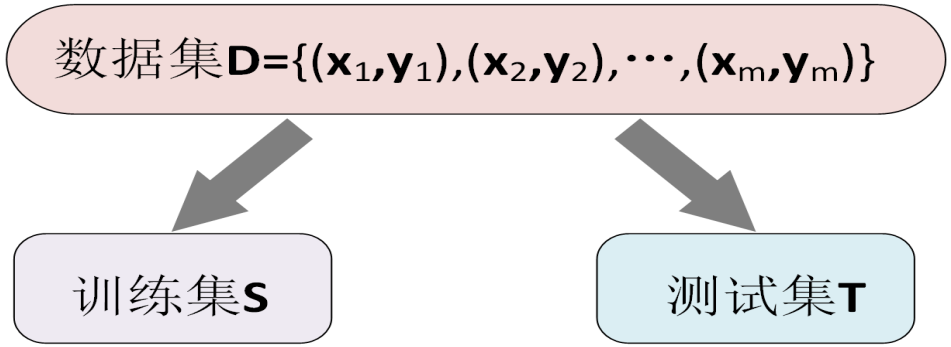
注意：测试集 (T) 应当尽量与训练集 (S) 互斥。即测试样本尽量不在训练集中出现、未在训练过程中使用过。

常用评估方法

- 留出法 (hold-out)
- 交叉验证法 (cross validation)

数据划分

评估一个机器学习模型的好坏关键在于泛化误差的大小，因此需选取一个**测试集**，以测试集上的测试误差作为泛化误差的估计。通常，我们假设测试样本也是从样本真实分布中独立同分布采样而得到的。



注意：测试集 (T) 应当尽量与训练集 (S) 互斥。即测试样本尽量不在训练集中出现、未在训练过程中使用过。

常用评估方法

- 留出法 (hold-out)
- 交叉验证法 (cross validation)
- 自助法 (bootstrapping)

留出法

留出法 (Hold-Out)：直接将数据集 D 划分为两个互斥的集合，其中一个作为训练集，另一个作为测试集。即$D = S \cup T,\ S \cap T = \emptyset$。常见做法是将大约 $2/3 \sim 4/5$ 的样本用于训练，剩余样本用于测试。

在 S 上训练模型后，用 T 来评估测试误差，作为泛化误差的估计。

注意:

留出法

留出法 (Hold-Out)：直接将数据集 D 划分为两个互斥的集合，其中一个作为训练集，另一个作为测试集。即$D = S \cup T,\ S \cap T = \emptyset$。常见做法是将大约 $2/3 \sim 4/5$ 的样本用于训练，剩余样本用于测试。

在 S 上训练模型后，用 T 来评估测试误差，作为泛化误差的估计。

注意:

- 保持数据分布的一致性，如**分层采样** (stratified sampling) 保持样本的类别比例相似。

留出法

留出法 (Hold-Out)：直接将数据集 D 划分为两个互斥的集合，其中一个作为训练集，另一个作为测试集。即$D = S \cup T,\ S \cap T = \emptyset$。常见做法是将大约 $2/3 \sim 4/5$ 的样本用于训练，剩余样本用于测试。

在 S 上训练模型后，用 T 来评估测试误差，作为泛化误差的估计。

注意：

- 保持数据分布的一致性，如**分层采样** (stratified sampling) 保持样本的类别比例相似。
- 一般要采用若干次**随机划分、重复进行实验评估后取平均值**作为留出法的评估结果。

留出法

留出法 (Hold-Out)：直接将数据集 D 划分为两个互斥的集合，其中一个作为训练集，另一个作为测试集。即$D = S \cup T,\ S \cap T = \emptyset$。常见做法是将大约 $2/3 \sim 4/5$ 的样本用于训练，剩余样本用于测试。

在 S 上训练模型后，用 T 来评估测试误差，作为泛化误差的估计。

注意:

- 保持数据分布的一致性，如**分层采样** (stratified sampling) 保持样本的类别比例相似。
- 一般要采用若干次**随机划分、重复进行实验评估后取平均值**作为留出法的评估结果。

例如，例如通过对 D 进行分层采样而获得含 70% 样本的训练集 S 和含 30% 样本的测试集 T，若 D 包含 500 个正例 500 个反例，则分层采样得到的 S 应包含 350 个正例、350 个反例，而 T 则包含 150 个正例和 150 个反例。

另外，可以把数据集 D 中的样本排序，然后把前 350 个正例放在训练集中，也可以将最后 350 个正例放到训练集，..... 这些不同的划分将导致不同的训练/测试集。

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

- 将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = \bigcup_{i=1}^{k} D_i$，$D_i \cap D_j = \emptyset (i \neq j)$。每个子集尽可能保持数据分布的一致性；

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

- 将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = \bigcup_{i=1}^{k} D_i$，$D_i \cap D_j = \emptyset (i \neq j)$。每个子集尽可能保持数据分布的一致性；
- 每次用 $k-1$ 个子集的并集作为训练集，剩下的一个作为测试集；

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

- 将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = \bigcup_{i=1}^{k} D_i$，$D_i \cap D_j = \emptyset (i \neq j)$。每个子集尽可能保持数据分布的一致性；
- 每次用 $k-1$ 个子集的并集作为训练集，剩下的一个作为测试集；
- 进行 k 次测试，结果取均值。

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

- 将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = \bigcup_{i=1}^{k} D_i$，$D_i \cap D_j = \emptyset (i \neq j)$。每个子集尽可能保持数据分布的一致性；
- 每次用 $k-1$ 个子集的并集作为训练集，剩下的一个作为测试集；
- 进行 k 次测试，结果取均值。

评估结果的稳定性和保真性很大程度上取决于 k 的取值。常见的 k 取值包括 $k = 5, 10, 20$ 等。

交叉验证法 (1)

k 折交叉验证(k-Fold Cross Validation)：

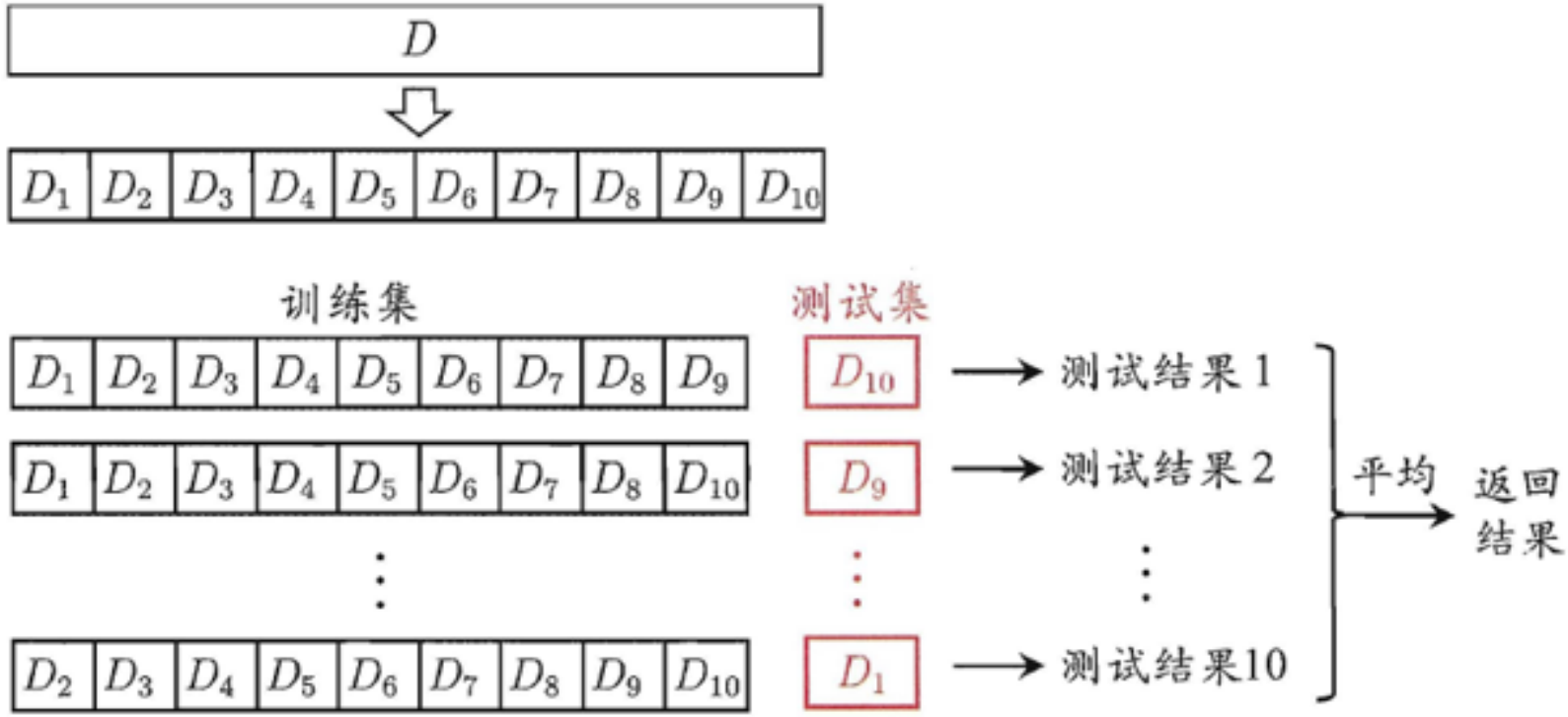
- 将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = \bigcup_{i=1}^{k} D_i$，$D_i \cap D_j = \emptyset (i \neq j)$。每个子集尽可能保持数据分布的一致性；
- 每次用 $k-1$ 个子集的并集作为训练集，剩下的一个作为测试集；
- 进行 k 次测试，结果取均值。

评估结果的稳定性和保真性很大程度上取决于 k 的取值。常见的 k 取值包括 $k = 5, 10, 20$ 等。

留一法(Leave-One-Out)：若数据集 D 中包含 m 个样本，当 $k = m$ 时，则得到交叉验证法的特例留一法。留一法不受随机样本划分方式影响，因为 m 个样本只有唯一的方式划分为 m 个子集（每个子集包含一个样本）。但当数据集较大时，需要训练 m 个模型，计算量巨大。

交叉验证法 (2)

下图为 10 折交叉验证的示意图，首先将数据集 D 划分为 10 个大小相似的子集；每次将其中的一份作为测试集，其余数据集作为训练集；最后得到 10 次的测试结果，求取平均。



10折交叉验证示意图

来源，周志华，《机器学习》，清华大学出版社

自助法

在**留出法**和**交叉验证法**中，由于保留了部分样本用于测试，因此实际评估的模型所使用的训练集比 D 小，会引入因训练样本规模不同而导致的估计偏差。

自助法 (Bootstrapping)

自助法

在**留出法**和**交叉验证法**中，由于保留了部分样本用于测试，因此实际评估的模型所使用的训练集比 D 小，会引入因训练样本规模不同而导致的估计偏差。

自助法 (Bootstrapping)

- **采用自助法生成训练集** D'：每次随机从 D 中挑出一个样本，将其拷贝放入 D'；然后再将该样本**放回**初始数据集中，重复执行 m 次后，得到包含 m 个样本的数据集 D'。注意：D 中有一些样本可能在 D' 中多次出现。

自助法

在**留出法**和**交叉验证法**中，由于保留了部分样本用于测试，因此实际评估的模型所使用的训练集比 D 小，会引入因训练样本规模不同而导致的估计偏差。

自助法 (Bootstrapping)

- **采用自助法生成训练集 D'**：每次随机从 D 中挑出一个样本，将其拷贝放入 D'；然后再将该样本**放回**初始数据集中，重复执行 m 次后，得到包含 m 个样本的数据集 D'。注意：D 中有一些样本可能在 D' 中多次出现。
- **采用自助法得到的测试集**：数据集 D 中未出现在 D' 中的所有样本作为测试集，即 $D - D'$。

显然，样本在 m 次采样中不出现在训练集 D' 的概率是：

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

即通过自助采样，初始数据集 D 中有约 36.8% 的样本未出现在采样数据集 D' 中。

自助法

在**留出法**和**交叉验证法**中，由于保留了部分样本用于测试，因此实际评估的模型所使用的训练集比 D 小，会引入因训练样本规模不同而导致的估计偏差。

自助法 (Bootstrapping)

- **采用自助法生成训练集 D'**：每次随机从 D 中挑出一个样本，将其拷贝放入 D'；然后再将该样本**放回**初始数据集中，重复执行 m 次后，得到包含 m 个样本的数据集 D'。注意：D 中有一些样本可能在 D' 中多次出现。
- **采用自助法得到的测试集**：数据集 D 中未出现在 D' 中的所有样本作为测试集，即 $D - D'$。

显然，样本在 m 次采样中不出现在训练集 D' 的概率是：

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

即通过自助采样，初始数据集 D 中有约 36.8% 的样本未出现在采样数据集 D' 中。

优点：适用于数据集较小、较难划分训练/测试集；能从初始数据集中产生不同的训练集，对集成学习等方法有很多好处。

自助法

在**留出法**和**交叉验证法**中，由于保留了部分样本用于测试，因此实际评估的模型所使用的训练集比 D 小，会引入因训练样本规模不同而导致的估计偏差。

自助法 (Bootstrapping)

- **采用自助法生成训练集 D'**：每次随机从 D 中挑出一个样本，将其拷贝放入 D'；然后再将该样本**放回**初始数据集中，重复执行 m 次后，得到包含 m 个样本的数据集 D'。注意：D 中有一些样本可能在 D' 中多次出现。
- **采用自助法得到的测试集**：数据集 D 中未出现在 D' 中的所有样本作为测试集，即 $D - D'$。

显然，样本在 m 次采样中不出现在训练集 D' 的概率是：

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

即通过自助采样，初始数据集 D 中有约 36.8% 的样本未出现在采样数据集 D' 中。

优点：适用于数据集较小、较难划分训练/测试集；能从初始数据集中产生不同的训练集，对集成学习等方法有很多好处。

缺点：产生的训练集改变了初始数据集的分布，引入估计偏差。

调参与最终模型

调参与最终模型

模型参数 (parameter)：大多数学习算法的模型原型中都有一些参数需要设定；参数配置不同，最终的模型性能可能有较大差别。

调参与最终模型

模型参数 (parameter)：大多数学习算法的模型原型中都有一些参数需要设定；参数配置不同，最终的模型性能可能有较大差别。

调参 (parameter tuning)：在进行模型评估与选择时，还需要对算法参数进行设定。学习算法的很多参数在实数范围内取值，对每个参数都训练模型不太现实；通常采用的方法是“逐步搜索”法：即在参数的取值范围内，按照某个步长进行参数取值，并评估最好的参数设置。例如某参数取值在 $[0, 1]$ 内，可以按照步长 0.1 进行参数评估，共需评估 11 个不同的参数设置。

调参与最终模型

模型参数 (parameter)：大多数学习算法的模型原型中都有一些参数需要设定；参数配置不同，最终的模型性能可能有较大差别。

调参 (parameter tuning)：在进行模型评估与选择时，还需要对算法参数进行设定。学习算法的很多参数在实数范围内取值，对每个参数都训练模型不太现实；通常采用的方法是“逐步搜索”法：即在参数的取值范围内，按照某个步长进行参数取值，并评估最好的参数设置。例如某参数取值在 $[0, 1]$ 内，可以按照步长 0.1 进行参数评估，共需评估 11 个不同的参数设置。

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留出一部分数据进行评估测试，事实上我们只使用了一部分数据训练模型。因此，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 重新训练模型. 这个模型在训练过程中使用了所有 m 个样本，这才是我们最终提交给用户的模型.

性能度量

性能度量 (Performance Measure)：衡量学习器泛化性能的标准。

性能度量反映了任务需求，在对比不同模型的能力时，使用不同的性能度量往往会导致不同的评判结果；这意味着模型的" 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，还决定于任务需求。

性能度量

性能度量 (Performance Measure)：衡量学习器泛化性能的标准。

性能度量反映了任务需求，在对比不同模型的能力时，使用不同的性能度量往往会导致不同的评判结果；这意味着模型的" 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，还决定于任务需求。

预测任务中，给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$。要评估学习器 f 的性能，就要把学习器预测结果 $f(\boldsymbol{x})$ 与真实标记 y 进行比较。

性能度量

性能度量 (Performance Measure)：衡量学习器泛化性能的标准。

性能度量反映了任务需求，在对比不同模型的能力时，使用不同的性能度量往往会导致不同的评判结果；这意味着模型的" 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，还决定于任务需求。

预测任务中，给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$。要评估学习器 f 的性能，就要把学习器预测结果 $f(\boldsymbol{x})$ 与真实标记 y 进行比较。

回归任务中，最常用的性能度量就是**均方误差** (Mean Squared Error):

$$E(f; D) = \frac{1}{m} \sum_{i=1}^{m} \left(f(\boldsymbol{x}_i) - \boldsymbol{y}_i\right)^2.$$

一般情况下，对于数据分布 $\mathcal{D}$ 和概率密度 $p(\cdot)$，均方差可描述为：

$$E(f; \mathcal{D}) = \int_{\boldsymbol{x} \sim \mathcal{D}} \left(f(\boldsymbol{x}) - \boldsymbol{y}\right)^2 p(\boldsymbol{x})\, dx.$$

以下主要介绍分类任务中常用的性能度量，包括错误率与精度，查准率、查全率与 $F1$，ROC 与 AUC，代价敏感错误率与代价曲线等。

错误率与精度

错误率 (Error Rate)：分类错误的样本占样本总数的比例。对数据集 D，分类错误率定义为:

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

对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$，错误率为：

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

错误率与精度

错误率 (Error Rate)：分类错误的样本占样本总数的比例。对数据集 D，分类错误率定义为:

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

对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$，错误率为：

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

精度 (Accuracy)：分类正确的样本占样本总数的比例。对数据集 D，精度为:

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

对于数据分布 $\mathcal{D}$ 和概率密度 $p(\cdot)$，精度为：

$$acc(f;\mathcal{D})=\int_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{I}(f(\boldsymbol{x})=\boldsymbol{y})p(\boldsymbol{x})\,d\boldsymbol{x}=1-E(f;\mathcal{D}).$$

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。
- **假正例 (False Positive, FP)**：测试集中是 Negative，模型预测结果是 Positive 的数据。

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。
- **假正例 (False Positive, FP)** ：测试集中是 Negative，模型预测结果是 Positive 的数据。
- **假反例 (False Negative, FN)**：测试集中是 Positive，模型预测结果是 Negative 的数据。

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。
- **假正例 (False Positive, FP)**：测试集中是 Negative，模型预测结果是 Positive 的数据。
- **假反例 (False Negative, FN)**：测试集中是 Positive，模型预测结果是 Negative 的数据。
- **真反倒 (True Negative, TN)**：测试集中是 Negative，模型预测结果是 Negative 的数据。

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。
- **假正例 (False Positive, FP)** ：测试集中是 Negative，模型预测结果是 Positive 的数据。
- **假反例 (False Negative, FN)**：测试集中是 Positive，模型预测结果是 Negative 的数据。
- **真反倒 (True Negative, TN)** ：测试集中是 Negative，模型预测结果是 Negative 的数据。

注意：$TP + FP + FN + TN =$ 样本总数；其中 TP，TN 表示样本类别的分类正确；FP，FN 表示样本类别的分类错误。

混淆矩阵

对于二分类问题，可将样例根据其真实类别（正例 Positive 与反例 Negative）与学习器预测类别的组合划分为四种情况：

- **真正例 (True Positive, TP)**：测试集中是 Positive，模型预测结果是 Positive 的数据。
- **假正例 (False Positive, FP)** ：测试集中是 Negative，模型预测结果是 Positive 的数据。
- **假反例 (False Negative, FN)**：测试集中是 Positive，模型预测结果是 Negative 的数据。
- **真反倒 (True Negative, TN)** ：测试集中是 Negative，模型预测结果是 Negative 的数据。

注意：$TP + FP + FN + TN =$ 样本总数；其中 TP，TN 表示样本类别的分类正确；FP，FN 表示样本类别的分类错误。

混淆矩阵 (confusion matrix)：

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)

$$\text{错误率} E = \frac{FN + FP}{TP + FN + FP + TN}.$$

$$\text{精度} acc = \frac{TP + TN}{TP + FN + FP + TN}.$$

查准率、查全率与 $F1$(1)

例如在西瓜问题中，我们经常关心‘挑出的西瓜中有多少比例是好瓜’，或者‘所有好瓜中有多少比例被挑出来了’，我们应采用哪种评价指标呢？

错误率和**精度**无法满足该要求。**查准率 P** (Precision) 与**查全率 R** (Recall) 更适用此类需求的性能度量。

$$\text{查准率}P = \frac{TP}{TP + FP}.$$

$$\text{查全率}R = \frac{TP}{TP + FN}.$$

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

混淆矩阵 (confusion matrix)

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)

注意到查准率和查全率是一对矛盾的度量。一般来说，查准率高时，查全率往往偏低；而查全率高时，查准率往往偏低。因此 $F1$ 是基于查准率与查全率的调和平均 (harmonic mean) 定义的，即 $\frac{1}{F1} = \frac{1}{2} \cdot (\frac{1}{P} + \frac{1}{R})$。

查准率、查全率与 $F1(2)$

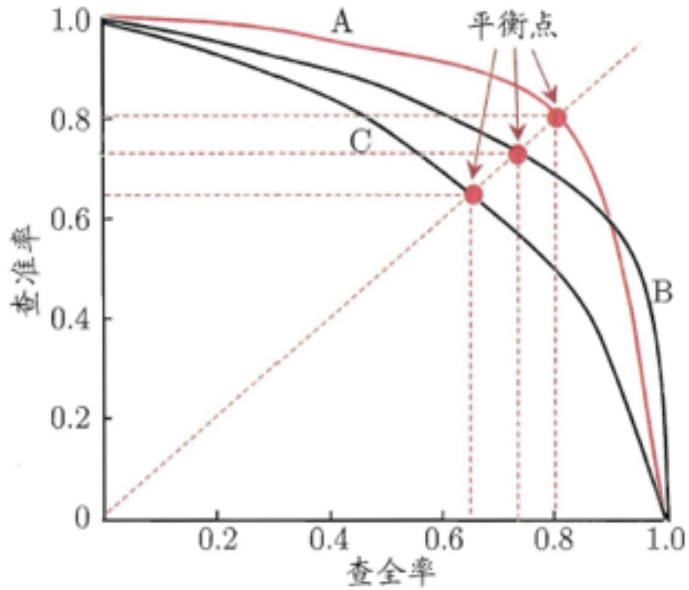
如果希望将好瓜尽可能地选出来，则可以增加瓜的数量来实现，如果将所有西瓜都选上，则查全率为 1，而查准率会降低；反之，若希望选出的瓜中好瓜比例高，则可只挑最有把握的瓜，但这会导致漏掉不少好瓜，使查全率较低。怎样综合考虑查准率和查全率？

查准率、查全率与 $F1(2)$

如果希望将好瓜尽可能地选出来，则可以增加瓜的数量来实现，如果将所有西瓜都选上，则查全率为 1，而查准率会降低；反之，若希望选出的瓜中好瓜比例高，则可只挑最有把握的瓜，但这会导致漏掉不少好瓜，使查全率较低。怎样综合考虑查准率和查全率？

P-R 曲线

根据学习器的预测结果对样本进行排序，将学习器认为“最可能”是正例的样本排在前面，将“最不可能”是正例的样本排在最后；按此顺序逐个把样本作为正例进行预测，每次都计算当前的查全率、查准率。从而得到 P-R 曲线。



P-R曲线与平衡点示意图

来源，周志华，《机器学习》，清华大学出版社

查准率、查全率与 $F1(3)$

查准率、查全率与 $F1(3)$

- 若一个学习器的 P-R 曲线被另一个学习器的曲线完全‘包住’，则后者的性能优于前者，如学习器 A 的性能优于 C；若两曲线交叉，则无法判断。

查准率、查全率与 $F1(3)$

- 若一个学习器的 P-R 曲线被另一个学习器的曲线完全‘包住’，则后者的性能优于前者，如学习器 A 的性能优于 C；若两曲线交叉，则无法判断。
- **平衡点 (Break-Even Point，BEP)**：为**查准率** $=$**查全率**时的取值。

查准率、查全率与 $F1$(3)

- 若一个学习器的 P-R 曲线被另一个学习器的曲线完全‘包住’，则后者的性能优于前者，如学习器 A 的性能优于 C；若两曲线交叉，则无法判断。
- **平衡点 (Break-Even Point，BEP)**：为**查准率** $=$**查全率**时的取值。
- $F1$ 也为查准率和查全率的综合性能度量，$F1$ 的一般形式 F_β:

查准率、查全率与 $F1$(3)

- 若一个学习器的 P-R 曲线被另一个学习器的曲线完全‘包住’，则后者的性能优于前者，如学习器 A 的性能优于 C；若两曲线交叉，则无法判断。
- **平衡点 (Break-Even Point，BEP)**：为**查准率** $=$**查全率**时的取值。
- $F1$ 也为查准率和查全率的综合性能度量，$F1$ 的一般形式 F_β:

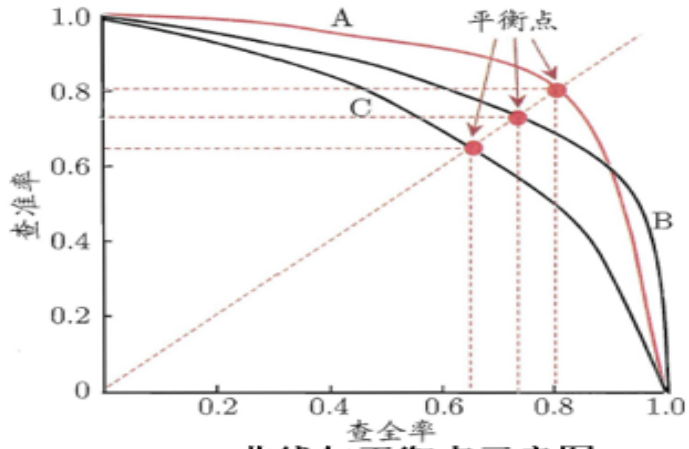
$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

其中 $\beta > 0$ 度量了查全率对查准率的相对重要性；

$\beta > 1$：查全率有更大影响；

$\beta < 1$：查准率有更大影响；

$\beta = 1$：退化为标准的 $F1$.



P-R曲线与平衡点示意图

来源，周志华，《机器学习》，清华大学出版社

宏/微查准率，宏/微查全率与宏/微 $F1$

宏/微查准率 (macro/micro-P)、宏/微查全率 (macro/micro-R) 与宏/微 $F1$(macro/micro-$F1$)

宏/微查准率，宏/微查全率与宏/微 $F1$

宏/微查准率 (macro/micro-P)、宏/微查全率 (macro/micro-R) 与宏/微 $F1$(macro/micro-$F1$)

宏查准率、宏查全率、宏 $F1$：n 个二分类混淆矩阵综合考察查准率、查全率和 $F1$。在各混淆矩阵上分别计算查准率和查全率，记为 (P_1, R_1)，(P_2, R_2)，...，(P_n, R_n)，则：

$$\text{macroP} = \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} \text{P}_\text{i}.$$

$$\text{macroR} = \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} \text{R}_\text{i}.$$

$$\text{macroF1} = \frac{2 \times \text{macroP} \times \text{macroR}}{\text{macroP} + \text{macroR}}.$$

宏/微查准率，宏/微查全率与宏/微 $F1$

宏/微查准率 (macro/micro-P)、宏/微查全率 (macro/micro-R) 与宏/微 $F1$(macro/micro-$F1$)

宏查准率、宏查全率、宏 $F1$：n 个二分类混淆矩阵综合考察查准率、查全率和 $F1$。在各混淆矩阵上分别计算查准率和查全率，记为 (P_1, R_1)，(P_2, R_2)，...，(P_n, R_n)，则：

$$\text{macroP} = \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} \text{P}_\text{i}.$$

$$\text{macroR} = \frac{1}{\text{n}} \sum_{\text{i}=1}^{\text{n}} \text{R}_\text{i}.$$

$$\text{macroF1} = \frac{2 \times \text{macroP} \times \text{macroR}}{\text{macroP} + \text{macroR}}.$$

微查准率、微查全率、微 $F1$：先将各混淆矩阵的对应元素进行平均，得到 TP，FP，TN，FN 的平均值，分别记为 $\overline{TP}$，$\overline{FP}$，$\overline{TN}$，$\overline{FN}$，则：

$$\text{microP} = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FP}}}.$$

$$\text{microR} = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FN}}}.$$

$$\text{microF1} = \frac{2 \times \text{microP} \times \text{microR}}{\text{microP} + \text{microR}}.$$

ROC 与 AUC (1)

ROC(Receiver Operating Characteristic) 与 AUC(Area Under ROC Curve)

如前所述，根据学习器的预测结果对样本进行排序。以某个**截断点 (Cut Point)**将样本判为正例和反例。根据任务需求采用不同的截断点，如若更重视**查准率**，则选择排序中靠前的位置进行截断；若更重视**查全率**，则选择靠后的位置进行截断。因此，排序本身的质量好坏，体现了综合考虑学习器在不同任务下的期望泛化性能的好坏。ROC 曲线从这个角度研究学习器的泛化性能。

ROC 与 AUC (1)

ROC(Receiver Operating Characteristic) 与 AUC(Area Under ROC Curve)

如前所述，根据学习器的预测结果对样本进行排序。以某个**截断点 (Cut Point)**将样本判为正例和反例。根据任务需求采用不同的截断点，如若更重视**查准率**，则选择排序中靠前的位置进行截断；若更重视**查全率**，则选择靠后的位置进行截断。因此，排序本身的质量好坏，体现了综合考虑学习器在不同任务下的期望泛化性能的好坏。ROC 曲线从这个角度研究学习器的泛化性能。

真正例率 (True Positive Rate,TPR)：真正例占所有正例的比率

$$TPR = \frac{TP}{TP + FN}. \tag{2}$$

ROC 与 AUC (1)

ROC(Receiver Operating Characteristic) 与 AUC(Area Under ROC Curve)

如前所述，根据学习器的预测结果对样本进行排序。以某个**截断点 (Cut Point)**将样本判为正例和反例。根据任务需求采用不同的截断点，如若更重视**查准率**，则选择排序中靠前的位置进行截断；若更重视**查全率**，则选择靠后的位置进行截断。因此，排序本身的质量好坏，体现了综合考虑学习器在不同任务下的期望泛化性能的好坏。ROC 曲线从这个角度研究学习器的泛化性能。

真正例率 (True Positive Rate,TPR)：真正例占所有正例的比率

$$TPR = \frac{TP}{TP + FN}. \tag{2}$$

假正例率 (False Positive Rate,TPR)：假正例占所有反例的比率

$$FPR = \frac{FP}{TN + FP}. \tag{3}$$

ROC 与 AUC (1)

ROC(Receiver Operating Characteristic) 与 AUC(Area Under ROC Curve)

如前所述，根据学习器的预测结果对样本进行排序。以某个**截断点 (Cut Point)**将样本判为正例和反例。根据任务需求采用不同的截断点，如若更重视**查准率**，则选择排序中靠前的位置进行截断；若更重视**查全率**，则选择靠后的位置进行截断。因此，排序本身的质量好坏，体现了综合考虑学习器在不同任务下的期望泛化性能的好坏。ROC 曲线从这个角度研究学习器的泛化性能。

真正例率 (True Positive Rate,TPR)：真正例占所有正例的比率

$$TPR = \frac{TP}{TP + FN}. \tag{2}$$

假正例率 (False Positive Rate,TPR)：假正例占所有反例的比率

$$FPR = \frac{FP}{TN + FP}. \tag{3}$$

背景：ROC 全称为‘受试者工作特征’曲线，它源于‘二战’中用于敌机检测的雷达信号分析技术，二十世纪六七十年代开始被用于一些心理学、医学检测应用中，此后被引入机器学习领域。

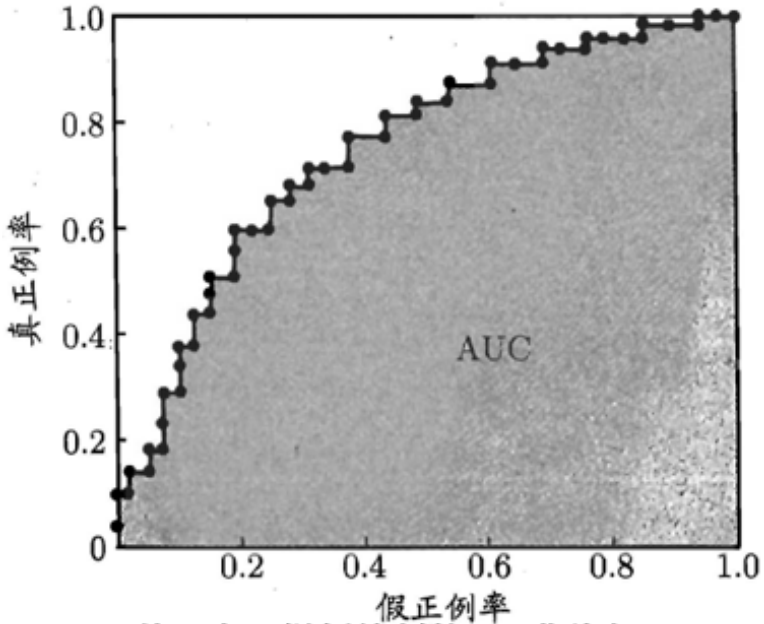
ROC 与 AUC (2)

绘制 ROC 曲线：给定 $m+(m-)$ 个正（反）例，根据学习器预测结果对样本进行排序，然后把分类阈值设为最大，即所有样本均预测为反例，此时 $TPR = TPR = 0$，在坐标 $(0,0)$ 处描点。然后，将分类阈值依次设为每个样本的预测值，即依次将每个样例划分为正例。

设前一个标记点坐标为 $(\boldsymbol{x}, \boldsymbol{y})$，

- 当前若为真正例，则对应标记点的坐标为 $(\boldsymbol{x}, \boldsymbol{y} + \frac{1}{m+})$；
- 当前若为假正例，则对应标记点的坐标为 $(\boldsymbol{x} + \frac{1}{m-}, \boldsymbol{y})$。

然后用线段连接相邻点即得 ROC 曲线。



基于有限样例绘制的ROC曲线与AUC

来源，周志华，《机器学习》，清华大学出版社

ROC 与 AUC (3)

ROC 曲线的意义：

ROC 与 AUC (3)

ROC 曲线的意义：

- 与 P-R 图相似，若一个学习器的 ROC 曲线被另一个学习器的的曲线完全‘包住’，则后者的性能优于前者；若两曲线交叉，则无法判断；

ROC 与 AUC (3)

ROC 曲线的意义：

- 与 P-R 图相似，若一个学习器的 ROC 曲线被另一个学习器的的曲线完全‘包住’，则后者的性能优于前者；若两曲线交叉，则无法判断；
- 图中虚线 y=x 上的点表示的是一个采用随机猜测策略的分类器的结果；

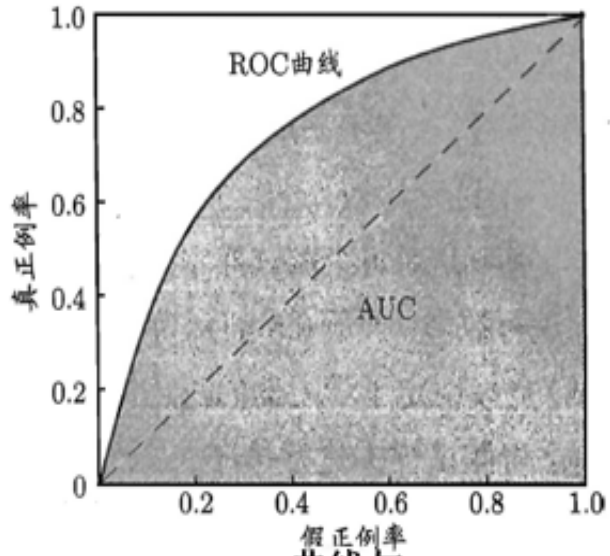
ROC 与 AUC (3)

ROC 曲线的意义：

- 与 P-R 图相似，若一个学习器的 ROC 曲线被另一个学习器的的曲线完全‘包住’，则后者的性能优于前者；若两曲线交叉，则无法判断；
- 图中虚线 y=x 上的点表示的是一个采用随机猜测策略的分类器的结果；

ROC 曲线图中的四个特殊点：

- $(0,1)$：即 $FPR=0$，$TPR=1$，则 $FN=FP=0$，所有样本分类正确；
- $(1,0)$：即 $FPR=1$，$TPR=0$，则 $TN=TP=0$，所有样本分类错误；
- $(0,0)$：即 $FPR=TPR=0$，则 $FP=TP=0$，所有样本被预测为负样本；
- $(1,1)$：即 $FPR=TPR=1$，则 $FN=TN=0$，所有样本被预测为正样本；



ROC曲线与AUC

来源，周志华，《机器学习》，清华大学出版社

ROC 与 AUC (4)

问题：如果两个学习器的 ROC 曲线交叉，怎样比较两个学习器的性能优劣？

ROC 与 AUC (4)

问题：如果两个学习器的 ROC 曲线交叉，怎样比较两个学习器的性能优劣？

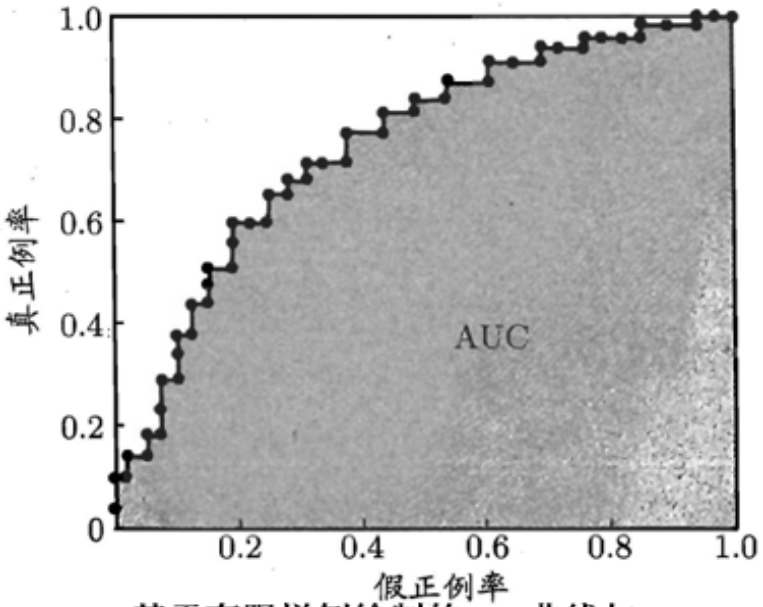
AUC(Area Under ROC Curve)为 ROC 曲线下的面积，当两个学习器的 ROC 曲线交叉时，通过比较 AUC 来判断两个学习器的性能优劣。

AUC 值越大，则学习器性能越好。

计算 AUC:

假定 ROC 曲线是由坐标为 $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$ 的点按序连接而形成，$\boldsymbol{x}_1 = 0, \boldsymbol{x}_m = 1$，则 AUC 可估算为：

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) \times (\boldsymbol{y}_i + \boldsymbol{y}_{i+1}).$$



基于有限样例绘制的ROC曲线与AUC

来源，周志华，《机器学习》，清华大学出版社

代价敏感错误率与代价曲线 (1)

引例：不同类型的错误所造成的后果不同。例如，在医疗诊断室中，错误地把患者诊断为健康人与错误地把健康人诊断为患者，看起来都是犯了‘一次错误’，但后者的影响是增加了进一步检查的麻烦，前者的后果却可能是丧失拯救生命的最佳时机。

代价敏感错误率与代价曲线 (1)

引例：不同类型的错误所造成的后果不同。例如，在医疗诊断室中，错误地把患者诊断为健康人与错误地把健康人诊断为患者，看起来都是犯了‘一次错误’，但后者的影响是增加了进一步检查的麻烦，前者的后果却可能是丧失拯救生命的最佳时机。

为权衡不同类型错误所造成的不同损失，可为不同类型的分类错误赋予**非均等代价 (Unequal Cost)**。

二分类代价矩阵：

真实类别	预测类别	
	第 0 类	第 1 类
第 0 类	0	$cost_{01}$
第 1 类	$cost_{10}$	0

$cost_{ij}$ 表示将第 i 类样本预测为第 j 类样本的代价，一般来说，$cost_{ii} = 0$；若将第 0 类判别为第 1 类造成的损失更大，则 $cost_{01} > cost_{10}$，损失程度相差越大，$cost_{01}$ 与 $cost_{10}$ 的值差别越大。

代价敏感错误率与代价曲线 (2)

均等条件下，错误代价相等，只需考虑错误次数，因此均等条件下错误率为：

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

代价敏感错误率与代价曲线 (2)

均等条件下，错误代价相等，只需考虑错误次数，因此均等条件下错误率为：

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

在**非均等条件下**的错误率称为**代价敏感 (Cost-Sensitive) 错误率**，需要考虑不同错误的代价:

$$\begin{aligned}E(f;D;cost)=&\frac{1}{m}\Big(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}(f(\boldsymbol{x}_i)\neq\boldsymbol{y}_i)\times cost_{01}\\&+\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}(f(\boldsymbol{x}_i)\neq\boldsymbol{y}_i)\times cost_{10}\Big).\end{aligned}$$

代价敏感错误率与代价曲线 (3)

代价曲线 (Cost Curve): 能够在非均等代价下，直接反应出学习器的期望总体代价（ROC 曲线在均等代价下可达到该目的）。

代价曲线图的横轴是取值为 $[0,1]$ 的正例概率代价（0 为正类，1 为负类）：

$$P(+)cost = \frac{p \times cost_{01}}{p \times cost_{01} + (1-p) \times cost_{10}}.$$

其中 p 是样例为正例的概率；纵轴是取值为 $[0,1]$ 的归一化代价：

$$cost_{norm} = \frac{FNR \times p \times cost_{01} + FPR \times (1-p) \times cost_{10}}{p \times cost_{01} + (1-p) \times cost_{10}}$$

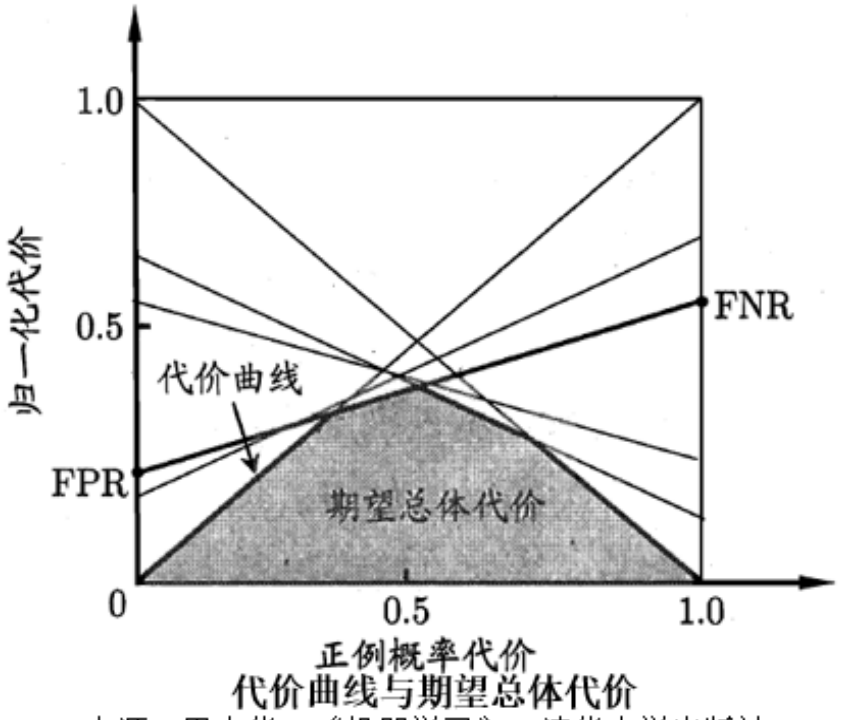
其中 FPR 是假正例率，$FNR = 1 - TPR$ 是假反例率。

代价敏感错误率与代价曲线 (4)

绘制代价曲线：

ROC 曲线上每一点对应了代价平面上的一条线段，设 ROC 曲线上点的坐标为 (FPR, TPR)，则可相应计算出 FNR，然后在代价平面上绘制一条从 $(0, FPR)$ 到 $(1, FNR)$ 的线段，线段下的面积表示了该条件下的期望总体代价；

将 ROC 曲线上的每个点转化为代价平面上的一条线段，**取所有线段的下界，围成的面积即为在所有条件下学习器的**期望总体代价。

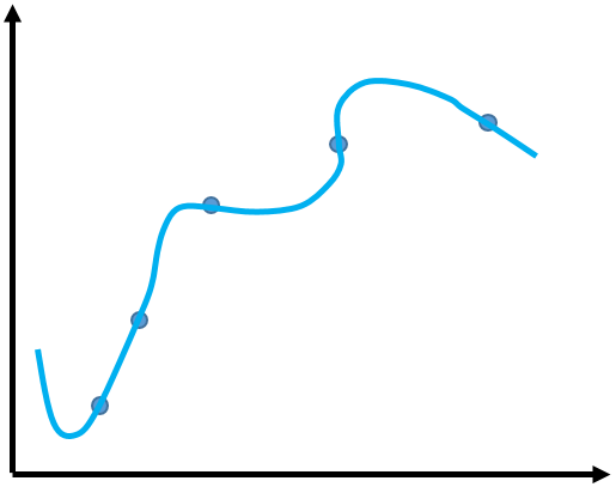


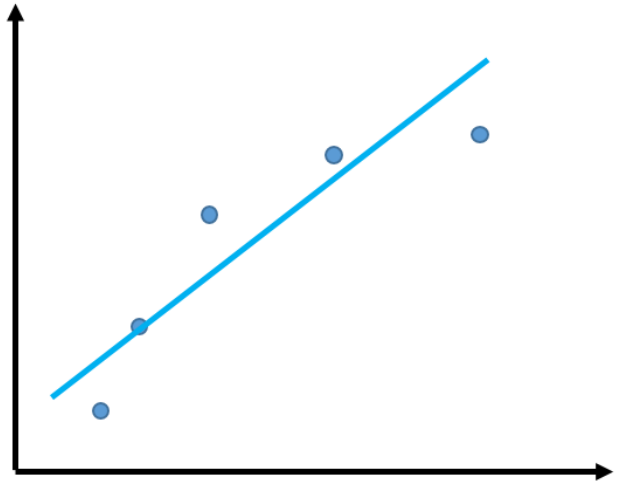
代价曲线与期望总体代价

来源，周志华，《机器学习》，清华大学出版社

偏差与方差

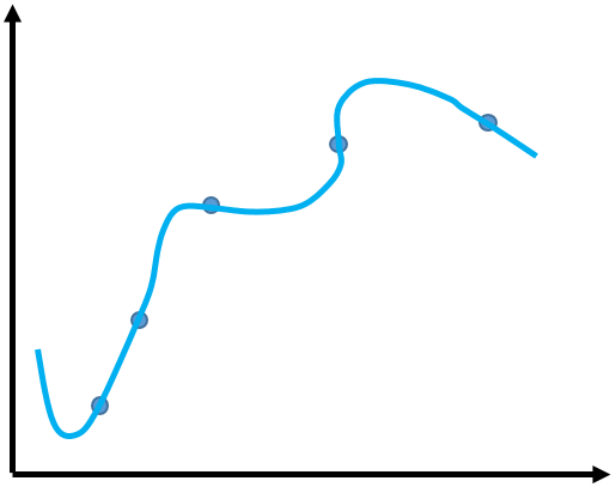
先回顾一下过拟合与欠拟合的图像。

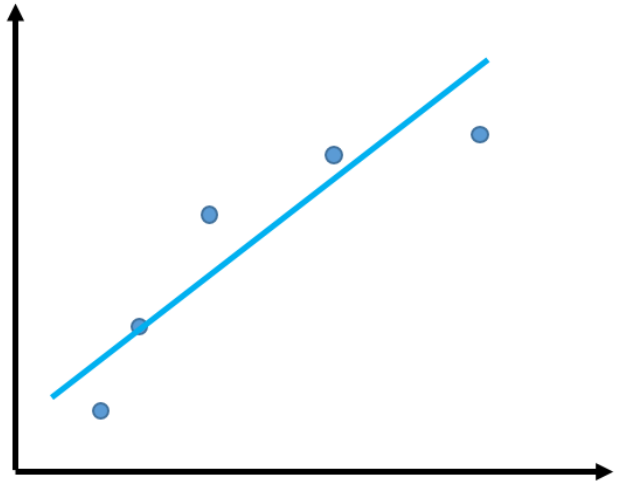




偏差与方差

先回顾一下过拟合与欠拟合的图像。

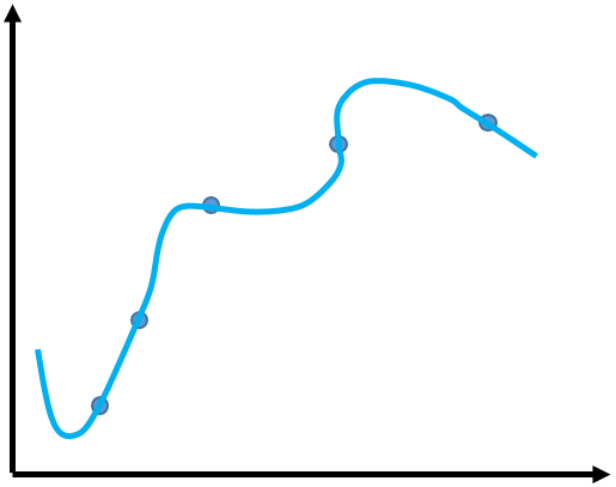


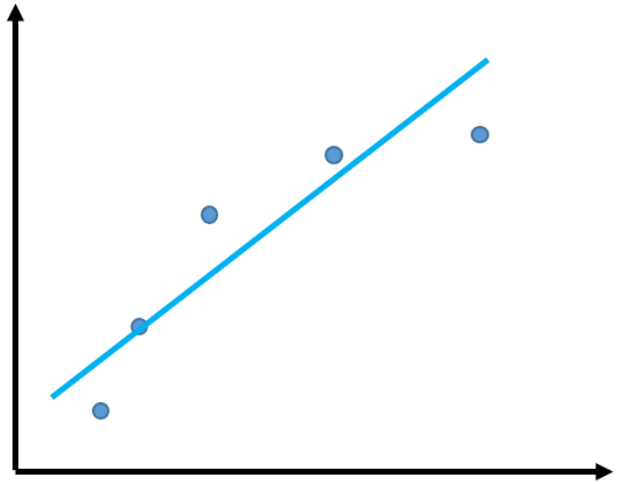


- **方差 (Variance)** 度量了同样大小的训练集的变动所导致的学习性能的变化，即**刻画了数据扰动所造成的影响**。过拟合能完美地拟合给定训练集中的每一个点，但是当训练集发生变动时，表现出高方差。

偏差与方差

先回顾一下过拟合与欠拟合的图像。





- **方差 (Variance)** 度量了同样大小的训练集的变动所导致的学习性能的变化，即**刻画了数据扰动所造成的影响**。过拟合能完美地拟合给定训练集中的每一个点，但是当训练集发生变动时，表现出高方差。
- **偏差（Bias）** 度量了学习算法的期望预测与真实结果的偏离程度，即**刻画了学习算法本身的拟合能力**。欠拟合直观上有一种拟合不足的感觉，表现为高偏差。

偏差 -方差分解

偏差 -方差分解试图对学习算法的期望泛化错误率进行拆解。

对测试样本 $\boldsymbol{x}$, 令 y_D 为 $\boldsymbol{x}$ 在数据集中的标记，y 为 $\boldsymbol{x}$ 的真实标记，$f(\boldsymbol{x}; D)$ 为训练集 D 上学得模型 f 在 $\boldsymbol{x}$ 上的预测输出。以回归任务为例，学习算法的期望预测为：

$$\bar{f}(\boldsymbol{x}) = \mathbb{E}_D[f(\boldsymbol{x}; D)].$$

使用样本数相同的不同训练集产生的方差为：

$$var(\boldsymbol{x}) = \mathbb{E}_D[(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x}))^2].$$

噪声为：

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

期望输出与真实标记的差别称为偏差，即：

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

其中，假定噪声期望为 0，即 $\mathbb{E}_D[y_D - y] = 0$

偏差 -方差分解 (2)

通过简单的多项式展开合并，可对**算法的期望泛化误差**进行分解：

$$
\begin{aligned}
E(f;D) &= \mathbb{E}_D[(f(\boldsymbol{x};D) - y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}) + \bar{f}(\boldsymbol{x}) - y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2] + \mathbb{E}_D[(\bar{f}(\boldsymbol{x}) - y_D)^2] \\
&+ \mathbb{E}_D[2(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))(\bar{f}(\boldsymbol{x}) - y_D)] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2] + \mathbb{E}_D[(\bar{f}(\boldsymbol{x}) - y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2] + \mathbb{E}_D[(\bar{f}(\boldsymbol{x}) - y + y - y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2] + \mathbb{E}_D[(\bar{f}(\boldsymbol{x}) - y)^2] + \mathbb{E}_D[(\boldsymbol{y} - y_D)^2] \\
&+ 2\mathbb{E}_D[(\bar{f}(\boldsymbol{x}) - y)(\boldsymbol{y} - y_D)] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2] + (\bar{f}(\boldsymbol{x}) - y)^2 + \mathbb{E}_D[(y - y_D)^2] \\
&= bias^2(\boldsymbol{x}) + var(\boldsymbol{x}) + \varepsilon^2.
\end{aligned}
\tag{4}
$$

偏差 -方差分解 (2)

通过简单的多项式展开合并，可对**算法的期望泛化误差**进行分解：

$$
\begin{aligned}
E(f;D) &= \mathbb{E}_D[(f(\boldsymbol{x};D)-y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x})+\bar{f}(\boldsymbol{x})-y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+\mathbb{E}_D[(\bar{f}(\boldsymbol{x})-y_D)^2] \\
&+ \mathbb{E}_D[2(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))(\bar{f}(\boldsymbol{x})-y_D)] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+\mathbb{E}_D[(\bar{f}(\boldsymbol{x})-y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+\mathbb{E}_D[(\bar{f}(\boldsymbol{x})-y+y-y_D)^2] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+\mathbb{E}_D[(\bar{f}(\boldsymbol{x})-y)^2]+\mathbb{E}_D[(\boldsymbol{y}-y_D)^2] \\
&+ 2\mathbb{E}_D[(\bar{f}(\boldsymbol{x})-y)(\boldsymbol{y}-y_D)] \\
&= \mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+(\bar{f}(\boldsymbol{x})-y)^2+\mathbb{E}_D[(y-y_D)^2] \\
&= bias^2(\boldsymbol{x})+var(\boldsymbol{x})+\varepsilon^2.
\end{aligned}
\tag{4}
$$

泛化误差可分解为偏差、方差与噪声之和。

偏差 -方差分解 (3)

偏差 -方差分解 (3)

- **偏差**衡量了算法的期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力；

偏差 -方差分解 (3)

- **偏差**衡量了算法的期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力；
- **方差**度量了同样大小的训练集的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响；

偏差 -方差分解 (3)

- **偏差**衡量了算法的期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力；
- **方差**度量了同样大小的训练集的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响；
- **噪声**表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界，即刻画了学习问题本身的难度。

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

偏差 -方差分解 (3)

- **偏差**衡量了算法的期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力；
- **方差**度量了同样大小的训练集的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响；
- **噪声**表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界，即刻画了学习问题本身的难度。

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

偏差一方差分解说明，泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。
给定学习任务，为了取得好的泛化性能，则需使偏差较小，即能够充分拟合数据，并且使方差较小，即使得数据扰动产生的影响小。

偏差 -方差窘境

偏差 -方差窘境 **(bias-variance dilemma)**：偏差与方差通常是有冲突的。

偏差 -方差窘境

偏差 -方差窘境 (bias-variance dilemma)： 偏差与方差通常是有冲突的。

- 在训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以使学习器产生显著变化，此时偏差主导了泛化错误率;

偏差 -方差窘境

偏差 -方差窘境 (bias-variance dilemma)： 偏差与方差通常是有冲突的。

- 在训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以使学习器产生显著变化，此时偏差主导了泛化错误率;
- 随着训练程度的加深，学习器的拟合能力逐渐增强，训练数据发生的扰动渐渐能被学习器学到，方差逐渐主导了泛化错误率;

偏差 -方差窘境

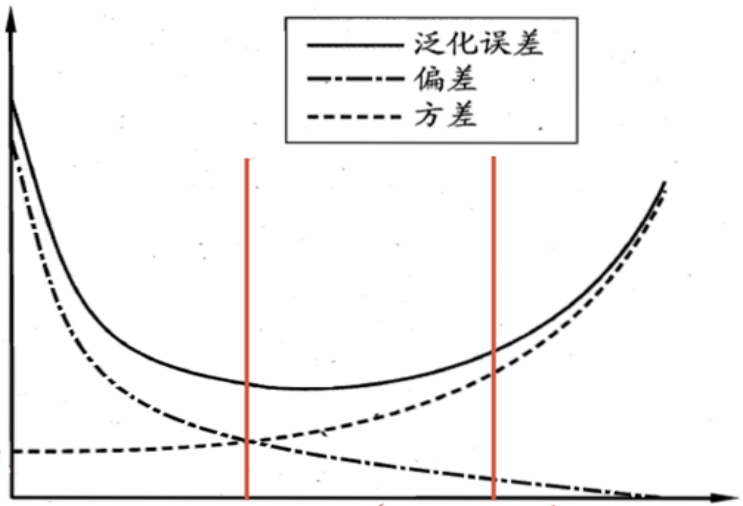
偏差 -方差窘境 (bias-variance dilemma)：偏差与方差通常是有冲突的。

- 在训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以使学习器产生显著变化，此时偏差主导了泛化错误率;
- 随着训练程度的加深，学习器的拟合能力逐渐增强，训练数据发生的扰动渐渐能被学习器学到，方差逐渐主导了泛化错误率;
- 在训练程度充足后，学习器的拟合能力已非常强，训练数据发生的轻微扰动都会导致学习器发生显著变化，若训练数据自身的非全局的特性被学习器学到了，则将发生过拟合。

偏差 -方差窘境

偏差 -方差窘境 (bias-variance dilemma)：偏差与方差通常是有冲突的。

- 在训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以使学习器产生显著变化，此时偏差主导了泛化错误率;
- 随着训练程度的加深，学习器的拟合能力逐渐增强，训练数据发生的扰动渐渐能被学习器学到，方差逐渐主导了泛化错误率;
- 在训练程度充足后，学习器的拟合能力已非常强，训练数据发生的轻微扰动都会导致学习器发生显著变化，若训练数据自身的非全局的特性被学习器学到了，则将发生过拟合。



泛化误差与偏差、方差的关系示意图
来源，周志华，《机器学习》，清华大学出版社

总结

本节主要讲解了模型学习与模型评估的方法，主要包括以下内容：

- 模型学习的基本概念：假设空间，归纳偏好，奥卡姆剃刀原则等；
- 经验误差与过拟合：经验误差，泛化误差，欠拟合以及过拟合；
- 评估方法：数据集的划分方法，包括留出法、交叉验证法以及自助法；
- 性能度量：主要介绍了衡量学习器泛化性能的标准，包括错误率与精度、查准率与查全率与 $F1$、ROC 与 AUC、代价敏感错误率与代价曲线；
- 偏差与方差：可将泛化误差分解为偏差、方差与噪声之和。