FDU 统计机器学习 1. 统计学习概论

本文参考以下教材:

- 统计学习方法 (第2版, 李航) 第 1 章
- 机器学习 (周志华) 第 $1, 2$ 章

欢迎批评指正!

1.1 基本分类

统计学习 (statistical learning) 是关于计算机基于数据构建概率统计模型并运用模型对数据进行预测和分析的学科.

- **监督学习** (supervised learning) 根据标注数据中构造预测模型.
  标注数据给出输入和输出的对应关系，监督学习的本质是从中学习输入到输出的映射的统计规律.
  - **半监督学习** (semi-supervised learning) 根据少量标注数据和大量未标注数据构造预测模型以节约成本
  - **主动学习** (active learning) 是指智能系统不断主动给出对学习最有帮助的实例让 "教师" 进行标注，
    然后再利用标注数据构造预测模型.
- **无监督学习** (unsupervised learning) 根据无标注数据构造预测模型.
  无标注数据是自然得到的数据，无监督学习的本质是学习数据中的统计规律或潜在结构.
- **强化学习** (reinforcement learning) 是指智能系统在与环境的连续互动中学习最优行为策略的机器学习问题.
  智能系统观测到的是与环境互动得到的数据序列，强化学习的本质是学习最优的序贯策略.

统计学习还可根据模型特征进行分类:

- 根据目标模型分布是条件概率分布还是函数可分为**概率模型** (probabilistic model) 和**确定性模型** (deterministic model)
  (最具代表性的一组例子是概率图模型和支持向量机)
- 其中确定性模型还可根据函数的线性与否分为**线性模型** (linear model) 和**非线性模型** (non-linear model)
  (最具代表性的一组例子是感知机和神经网络)

监督学习模型还可以分为**生成模型** (generative model) 和**判别模型** (discriminative model)

- 生成方法根据数据学习输入输出 (X, Y) 的联合概率分布 $p(x, y)$
  然后根据 $p(y|x) = \frac{p(x,y)}{p_X(x)}$ 得到条件概率分布 $p(y|x)$
- 判别方法根据数据直接学习决策函数 $f(x)$ 或条件概率分布 $p(y|x)$

生成方法可以还原出联合概率分布 $p(x, y)$，而判别方法不能;
生出方法的收敛速度更快;
生出方法可以处理存在隐变量的情况;
判别方法直接面对预测，往往学习的准确率更高;
判别方法可以对数据进行各种程度上的抽象，定义并使用更多的特征，因此可以简化学习问题.

统计学习还可根据其使用的技巧进行分类:

- **频率学派** (Frequentist) 与 **Bayes 学派**的最主要区别: 是否允许**先验概率分布**的使用
  - 频率学派不假设任何的先验知识，不参照过去的经验，只按照当前已有的数据进行概率推断
  - Bayes 学派会假设先验知识的存在，然后再用采样逐渐修改先验知识并逼近真实知识
- **核方法** (kernal method) 通过使用核函数表示和学习非线性模型.
  它通过将线性模型中的内积运算替换为核函数进行推广，提升其表现.
  (例如核函数支持向量机)

1.2 基本构成

统计学习方法都是由模型、策略和算法构成的.

1.2.1 模型

记 $\mathcal{X}$ 和 $\mathcal{Y}$ 分别为输入空间和输出空间.

假设空间既可以定义为决策函数族 $\mathscr{F}=\{f_\theta(\cdot):\theta\in\Theta \text{ such as } \begin{cases}\mathrm{dom}(f_\theta)=\mathcal{X}\\ \mathrm{Range}(f_\theta)\subseteq\mathcal{Y}\end{cases}\}$

也可以表示为条件概率分布族 $\mathscr{F}=\{p_\theta(\cdot|\cdot):\theta\in\Theta\}$

假设空间中的任意一个对象都是一个模型.

1.2.2 策略

统计学习依照一定的策略从假设空间中选取最优模型.

记训练集为 $D_{\text{train}}=\{(x_i,y_i):i=1,\dots,N\}$

(1) 损失函数

损失函数 $\mathrm{loss}(y,f(x))$ 度量预测错误 $(f(x)\neq y)$ 的程度 (即代价)，它是非负实值函数.

损失函数值越小，模型该次预测的准确率就越高.

- ① 0-1 **损失函数:**

  $\mathrm{loss}(y,f(x;\theta))=I(y\neq f(x;\theta))=\begin{cases}0, & \text{if } f(x;\theta)=y\\ 1, & \text{otherwise}\end{cases}$

  其缺点是数学性质不是很好:

  不连续且导数为 0，难以优化.

  因此经常用连续可微的损失函数替代.

- ② **平方损失函数 (Quadratic Loss Function):**

  $\mathrm{loss}(y,f(x;\theta))=\frac{1}{2}\|y-f(x;\theta)\|_2^2$

- ③ **交叉熵损失函数 (Cross-Entropy Loss Function):**

  假设样本标签 $y\in\{1,2,\dots,n\}$,

  模型的输出 $f(x)\in[0,1]^n$ 为类别标签的条件概率分布,

  即满足 $\begin{cases}f_i(x;\theta)=p(y=i|x;\theta)\in[0,1] & (\forall\, i=1,2\dots,n)\\ \sum_{i=1}^n f_i(x;\theta)=1\end{cases}$

  这样的 $f(x;\theta)$ 称为一个 **one-hot 向量**.

  我们定义真实分布 $y=p_r(y|x)$ 与模型预测分布 $f(x;\theta)$ 之间的交叉熵为:

$$\begin{aligned}\mathrm{loss}(y,f(x;\theta))&=-y^{\mathrm{T}}\log\left(f(x;\theta)\right)\\&=-\sum_{i=1}^n y_i\log\left(f_i(x;\theta)\right)\end{aligned}$$

在实际应用中，如果样本类别为 k，则它属于第 k 类的概率为 1，属于其他类的概率为 0
(即 $y = e_k$，其中 e_k 为 $\mathbb{R}^n$ 的第 k 个标准基向量)

- **④ Hinge 损失函数:**
  对于二分类问题，假设样本标签 $y \in \{-1, +1\}$，模型输出 $f(x;\theta) \in \mathbb{R}$
  $\text{loss}(y, f(x;\theta)) = \max\{0, 1 - yf(x;\theta)\} = [1 - yf(x;\theta)]_+$
  其中 $[x]_+ := \max\{0, x\}$

(2) 风险函数

假设模型的输入输出为随机变量 (X, Y)，且服从联合概率分布 $p(x, y)$
我们定义**期望风险函数** (expected risk function) 为模型 f 关于联合分布 p 的期望损失，即有:

$$\begin{aligned}\text{Risk}_{\text{exp}}(f) &= \text{E}_p[\text{loss}(Y, f(X))] \\ &= \int_{\mathcal{X}\times\mathcal{Y}} \text{loss}(y, f(x))p(x, y)\text{d}x\text{d}y\end{aligned}$$

但由于联合分布 $p(x, y)$ 是未知的，故我们定义**经验风险函数** (empirical risk function):

$$\text{Risk}_{\text{emp}}(f) := \frac{1}{N}\sum_{i=1}^{N}\text{loss}(y_i, f(x_i))$$

$$\text{where the train set is } D_{\text{train}} := \{(x_i, y_i) : i = 1, \ldots, N\}$$

根据大数定律，当训练集容量 $N \to \infty$ 时，我们有 $\text{Risk}_{\text{emp}}(f) \to \text{Risk}_{\text{exp}}(f)$ 成立.
但在实际应用中训练样本是有限的，故仅使用经验风险函数往往不理想，我们需对此策略进行修正.
这就关系到监督学习的两个基本策略: 经验风险最小化和结构风险最小化.

经验风险最小化 (empirical risk minimization, ERM) 策略认为经验风险最小的模型是最优的模型.
此时问题就转化为:

$$\min_{f\in\mathcal{F}}\text{Risk}_{\text{emp}}(f) = \min_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}\text{loss}(y_i, f(x_i))$$

当训练样本量 N 足够大时，经验风险最小化策略效果会很好.
特别地，当模型是条件概率分布且损失函数是对数损失函数时，经验风险最小化就等价于极大似然估计.
但当训练样本量 N 很小时，经验风险最小化策略效果通常不好，会造成**过拟合** (over-fitting) 的现象.

结构风险最小化 (structural risk minimization, SRM) 策略是为防止过拟合而提出的策略.
它在经验风险的基础上添加了表示模型复杂度的**正则化项** (regularizer):

$$\text{Risk}_{\text{srm}}(f) := \text{Risk}_{\text{emp}}(f) + \lambda J(f)$$

其中定义在假设空间 $\mathcal{F}$ 上的泛函 $J(f)$ 与模型的复杂度正相关，而超参 $\lambda \geq 0$ 是正则化系数.
正则化项 $\lambda J(f)$ 就代表了对复杂模型的惩罚.

结构风险小的模型往往在对训练数据有好的拟合的同时，还对测试数据有较好的预测.
特殊地，当模型是条件概率分布、损失函数是对数损失函数且模型复杂度由模型的先验概率表示时，
结构风险最小化就等价于最大后验估计.

1.2.3 算法

在确定了假设空间 $\mathcal{F}$、学习准则和训练集 $D_{\text{train}} = \{(x_i, y_i) : i = 1, \dots, N\}$ 之后，
如何找到最优的模型 $f(x; \theta^\star)$ 就成了一个**优化问题** (Optimization Problem)
机器学习的训练过程其实就是最优化问题的求解过程.

参数与超参数：
在机器学习中，优化又可以分为**参数优化**和**超参数优化**.
模型 $f(x; \theta)$ 中的 θ 称为模型的**参数**，可以通过优化算法进行学习.
除了可学习的参数 θ 之外，还有一类参数是用来定义模型结构或优化策略的，
这类参数叫作**超参数** (Hyper-Parameter).
在 Bayes 方法中，超参数可以理解为参数的参数，即**控制模型参数的参数**.

常见的超参数：
聚类算法中的类别个数、梯度下降法中的步长、正则项的系数、神经网络的层数、支持向量机中的核函数等.
超参数的选取一般都是组合优化问题，很难通过优化算法来自动学习.
因此，超参数优化是机器学习的 一个**经验性**很强的技术，
通常是按照人的经验设定，或者通过搜索的方法对一组超参数组合进行不断试错调整.

(1) 梯度下降法

最简单、常用的优化算法是**梯度下降法** (Gradient Descent).
首先初始化参数 θ_0，然后迭代计算训练集 D_{train} 上经验风险函数 $\text{Risk}_{\text{emp}}(D_{\text{train}}, \theta)$ 的最小值：

$$\begin{aligned}\theta_{k+1} &= \theta_k - \alpha \cdot \nabla_\theta \text{Risk}_{\text{emp}}(\theta_k) \\ &= \theta_k - \alpha \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \text{loss}(y^{(i)}, f(x^{(i)}; \theta_k))\end{aligned}$$

其中 θ_k 是第 k 次迭代时的参数值，α 为搜索步长，又称**学习率** (learning rate).

- **提前停止策略 (Early Stop Strategy):**
  在梯度法的训练过程中，由于过拟合的原因，
  在训练样本上收敛的参数，并不一定在测试集上最优.
  因此，除了训练集和测试集之外，有时也会使用**验证集** (Validation Set) 来进行模型选择.
  每次迭代时 (例如第 k 次迭代)，
  将新得到的模型 $f(x; \theta_{k+1})$ 在验证集上计算错误率，
  并与上一次迭代的模型 $f(x; \theta_k)$ 在验证集上的错误率进行比较.

如果在验证集上的错误率不再下降，就停止迭代.

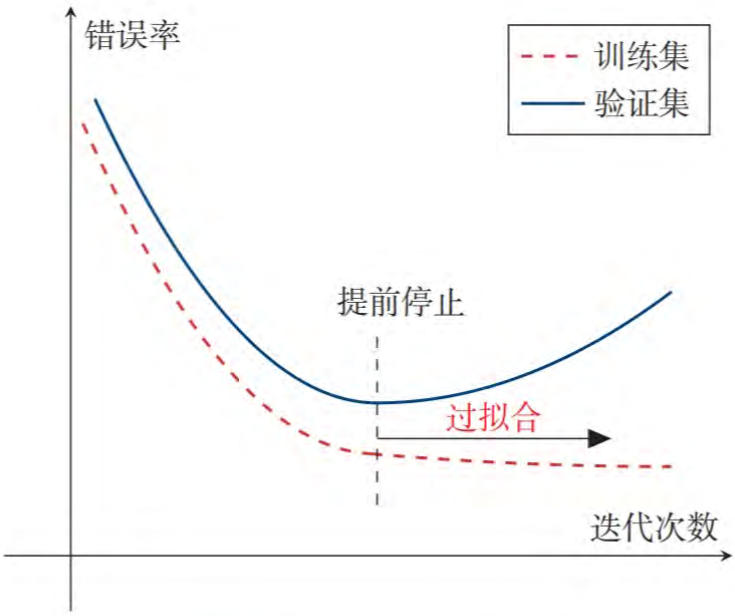


图 2.4 提前停止的示例

(2) 随机梯度下降法

前面介绍的梯度法又称**批量梯度下降法** (Batch Gradient Descent, BGD)
它使用 n 个样本的经验风险 $\mathrm{Risk}_{\mathrm{emp}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{loss}(y_i, f(x_i;\theta))$ 的梯度，
来近似期望风险 $\mathrm{Risk}_{\mathrm{exp}}(\theta) = \mathrm{E}[\mathrm{loss}(y, f(x;\theta))]$ 的梯度.
当样本量 N 很大时，每次迭代的计算开销会很大.

为了减少每次迭代的计算复杂度，
我们可以随机抽取一个样本 (x_s, y_s) $(s \in \{1, \ldots, n\})$，
利用其损失函数 $\mathrm{loss}(y_s, f(x_s;\theta_k))$ 的梯度来计算参数 θ_{k+1}
这称为**随机梯度下降法** (Stochastic Gradient Descent, SGD)
经过足够次数的迭代后，随机梯度下降法也可以收敛到局部最优解.
然而它有一个缺点：是无法充分利用计算机的并行计算能力.

二者的区别在于:
每次迭代的优化目标是所有样本的平均损失函数 $\mathrm{Risk}_{\mathrm{emp}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{loss}(y_i, f(x_i;\theta))$
还是随机抽取的单个样本 (x_s, y_s) 的损失函数 $\mathrm{loss}(y_s, f(x_s;\theta))$

(3) 小批量梯度下降法

小批量梯度下降法 (Mini-Batch Gradient Descent) 是批量梯度下降法和随机梯度下降法的折中.
每次迭代时，我们随机选取一小部分训练样本来计算梯度并更新参数.
第 k 次迭代时，随机选取一个包含 n_0 个样本的子集 S_k，
计算在这个子集上每个样本损失函数的梯度并取平均，然后进行参数更新：

$$\theta_{k+1} \leftarrow \theta_k - \alpha\frac{1}{n_0}\sum_{(x,y)\in S_k}\nabla_\theta \mathrm{Loss}(y, f(x;\theta_k))$$

其中 n_0 一般在 $1 \sim 100$ 之间，通常设置为 2 的幂.
在实际应用中，小批量随机梯度下降法有收敛快、计算开销小的优点，
因此逐渐成为大规模的机器学习中的主要优化算法.

1.3 模型选择

1.3.1 训练误差与测试误差

记假设空间为 $\mathcal{F}$，选取的模型为 f，损失函数为 $\text{loss}(y, f(x))$
训练集为 $D_{\text{train}} = \{(x_i, y_i) : i = 1, \ldots, N\}$，测试集为 $D_{\text{test}} = \{(x'_i, y'_i) : i = 1, \ldots, N'\}$

> 将数据集 D 划分为两个互斥的集合 D_{train} 和 D_{test} 的方法称为**留出法** (hold-out)
> 其划分要尽可能保持数据分布的一致性，避免因划分引入额外的偏差.
> 其中保留类别比例的采样方式称为**分层采样** (stratified sampling)
>
> 单次使用留出法得到的估计结果往往不够稳定可靠，
> 一般需要若干次随机划分、重复实验评估后取平均值作为留出法的评估结果.
>
> 测试集越小，评估结果的方差越大;
> 训练集越小，评估结果的偏差越大 (因为训练集和测试集的差距会越大)

训练误差即模型 f 关于训练集的平均损失 (即经验风险函数):

$$\text{Risk}_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^{N} \text{loss}(y_i, f(x_i))$$

欠拟合 (Underfitting):
和过拟合相反的一个概念是**欠拟合**，
即模型不能很好地拟合训练数据，在训练集上的错误率比较高.
欠拟合一般是由于模型能力不足造成的.

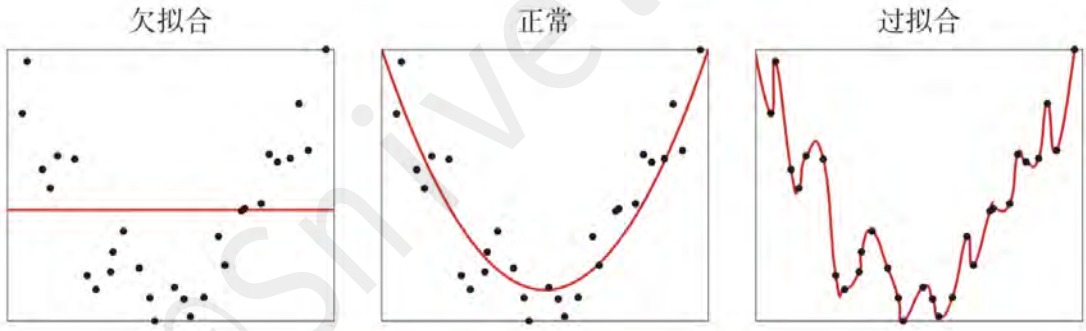


图 2.3　欠拟合和过拟合示例

测试误差即模型 f 关于测试集的平均损失:

$$\text{Error}_{\text{test}} := \frac{1}{N'} \sum_{i=1}^{N'} \text{loss}(y'_i, f(x'_i))$$

测试误差的大小反映了学习方法对未知数据的预测能力，称为**泛化能力** (generalization ability)

当假设空间含有不同复杂度 (例如参数量) 的模型时，就面临**模型选择** (model selection) 的问题.
若一味追求降低训练误差 (即经验风险) (即提供对训练数据的拟合能力),
则选取的模型往往会比真模型 (如果存在) 更高.
这种现象称为**过拟合** (over-fitting)

过拟合是指模型复杂度过高，以至于出现对已知数据拟合得很好，但对位置数据预测得很差的现象.
可以说模型选择旨在避免过拟合并提高模型的预测能力.

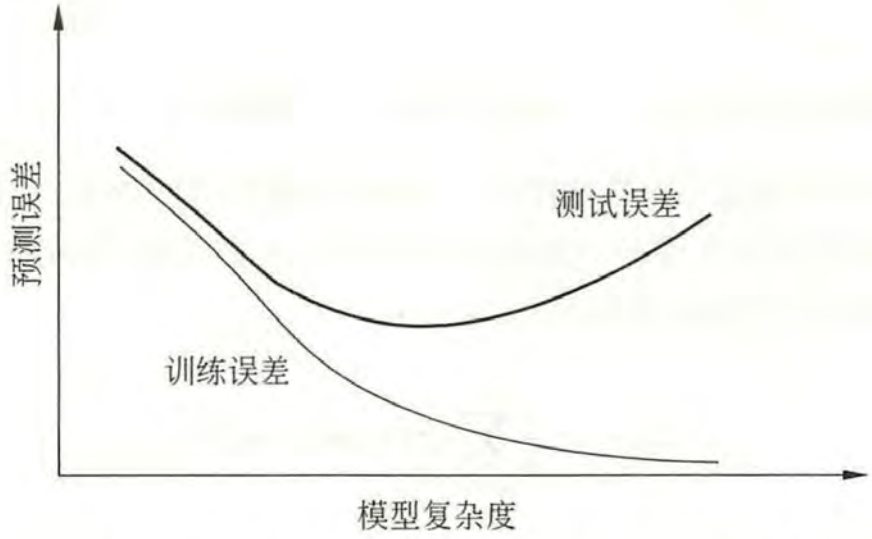


图 1.9　训练误差和测试误差与模型复杂度的关系

1.3.2 偏差-方差分解

如何在模型的拟合能力和泛化能力之间取得一个较好的平衡，对机器学习算法来说至关重要.
偏差-方差分解 (Bias-Variance Decomposition) 为我们提供了一个有效的工具.
这里我们以回归问题为例进行介绍，但其结论适用于一般的分类问题.

以回归问题为例：
假设样本的**真实分布**为 $p_r(x,y)$.
我们采用平方损失函数，定义模型 $f(x;\theta)$ 的**期望风险**为：

$$\mathrm{Risk}_{\exp}(\theta)=\mathrm{E}_{(x,y)\sim p_r(x,y)}[(y-f(x;\theta))^2]$$

最优模型为 $f(x;\theta^\star)=\mathrm{E}_{y\sim p_r(y|x)}[y]$
模型 $f(x;\theta)$ 的期望风险可以分解为：

$$\begin{aligned}\mathrm{Risk}(\theta)&=\mathrm{E}_{(x,y)\sim p_r(x,y)}[(y-f(x;\theta))^2]\\&=\mathrm{E}_{(x,y)\sim p_r(x,y)}[(y-f(x;\theta^\star)+f(x,\theta^\star)-f(x;\theta))^2]\\&=\mathrm{E}_{(x,y)\sim p_r(x,y)}[(y-f(x;\theta^\star))^2]+\mathrm{E}_{x\sim p_r(x)}[(f(x,\theta^\star)-f(x;\theta))^2]\\&\overset{\Delta}{=}\mathrm{Var}(\varepsilon)+\mathrm{E}_{x\sim p_r(x)}[(f(x,\theta^\star)-f(x;\theta))^2]\end{aligned}$$

其中第一项由样本分布以及噪声引起，无法通过优化模型来减少.
而第二项是当前模型 $f(x;\theta)$ 与最优模型 $f(x;\theta^\star)$ 的差距，是机器学习算法优化的真实目标.

对于单个样本 x，所有可能的训练集 D 得到的模型 $f(x;\theta_D)$ 和最优模型 $f(x;\theta^\star)$ 的期望差距为：

$$\begin{aligned}\mathrm{E}_D[(f(x;\theta_D)-f(x,\theta^\star))^2]&=\mathrm{E}_D[(f(x;\theta_D)-\mathrm{E}_D[f(x;\theta_D)]+\mathrm{E}_D[f(x;\theta_D)]-f(x;\theta^\star))^2]\\&=\{\mathrm{E}_D[f(x;\theta_D)]-f(x;\theta^\star)\}^2+\mathrm{E}_D[(f(x;\theta_D)-\mathrm{E}_D[f(x;\theta_D)])^2]\\&\overset{\Delta}{=}\mathrm{bias}^2(x)+\mathrm{variance}(x)\end{aligned}$$

- 第一项为**偏差** (bias)，
  代表一个模型在不同训练集上的平均性能和最优模型的差异，
  用于衡量一个模型的拟合能力.
- 第二项是**方差** (variance)，
  代表一个模型在不同训练集上的性能的变异程度，
  用于衡量一个模型是否容易过拟合，也就是衡量模型的泛化能力.

综上所述，我们有：

$$
\begin{aligned}
\text{Risk}(\theta) &= \mathrm{E}_{(x,y)\sim p_r(x,y)}[(y - f(x;\theta^\star))^2] + \mathrm{E}_{x\sim p_r(x)}[(f(x,\theta^\star) - f(x;\theta))^2] \\
&\overset{\Delta}{=} \mathrm{Var}(\varepsilon) + \mathrm{E}_{x\sim p_r(x)}[(f(x,\theta^\star) - f(x;\theta))^2] \\
&= \mathrm{Var}(\varepsilon) + \mathrm{E}_{x\sim p_r(x)}[\mathrm{E}_D[(f(x;\theta_D) - f(x;\theta_D))^2]] \\
&= \mathrm{Var}(\varepsilon) + \mathrm{E}_{x\sim p_r(x)}[\text{bias}^2(x) + \text{variance}(x)] \\
&\overset{\Delta}{=} \mathrm{Var}(\varepsilon) + \text{bias}^2 + \text{variance}
\end{aligned}
$$

其中 $\begin{cases} \text{bias}^2 = \mathrm{E}_{x\sim p_r(x)}[\text{bias}(x)^2] = \mathrm{E}_{x\sim p_r(x)}\{(\mathrm{E}_D[f(x;\theta_D)] - f(x;\theta^\star))^2\} \\ \text{variance} = \mathrm{E}_{x\sim p_r(x)}[\text{variance}(x)] = \mathrm{E}_{x\sim p_r(x)}\{\mathrm{E}_D[(f(x;\theta_D) - \mathrm{E}_D[f(x;\theta_D)])^2]\} \end{cases}$
分别代表模型的**拟合能力**和**泛化能力**.

下图给出了四种偏差和方差的5组合情况，
其中数据点代表不同训练集 D 上得到的模型 $f(x;\theta_D)$，同心圆的圆心代表最优模型 $f(x,\theta^\star)$.

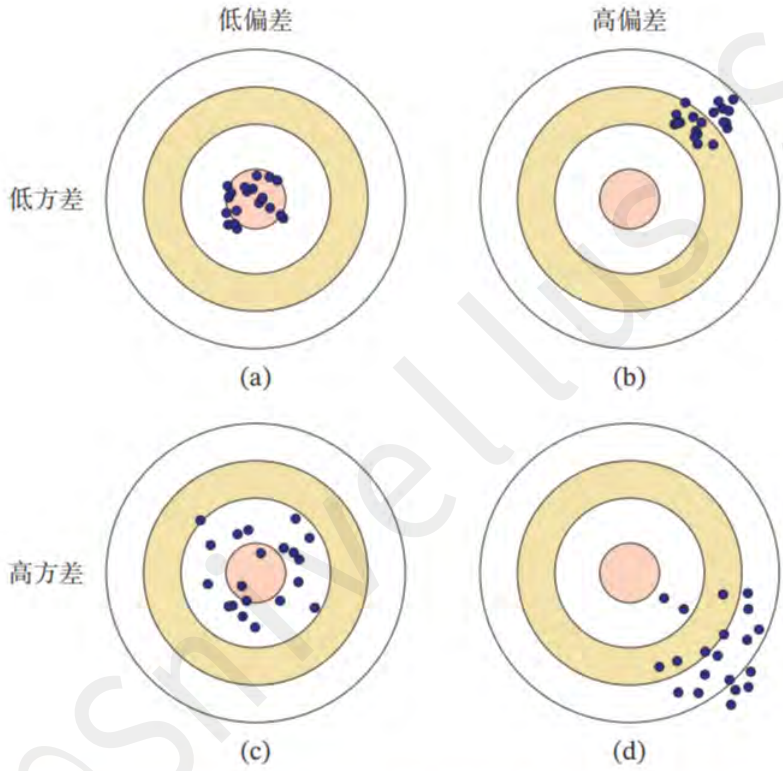


图 2.6 机器学习模型的四种偏差和方差组合情况

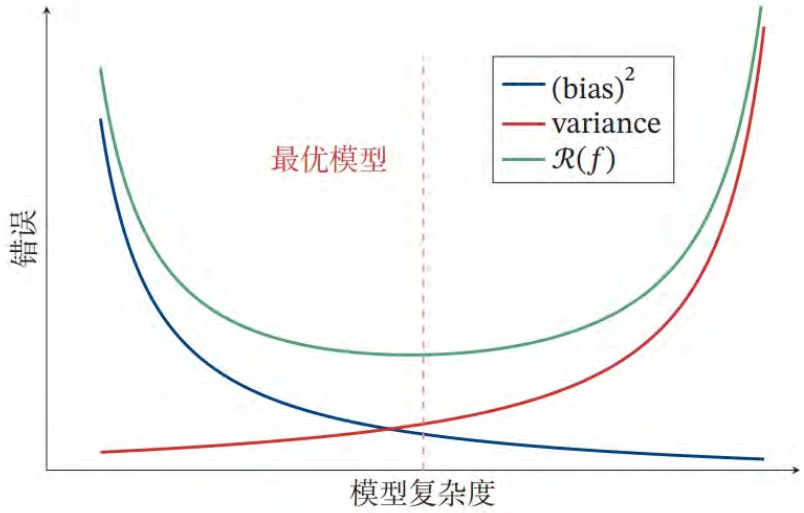


图 2.7　机器学习模型的期望错误、偏差和方差随复杂度的变化情况

1.3.3 正则化

正则化是结构风险最小化策略的实现
即在经验风险 $\text{Risk}_{\text{emp}}(f)$ 上加上一个正则化项 (一般是模型复杂度的单调递增函数)
例如可以是模型参数向量的范数.

正则化策略的优化问题一般具有如下形式:

$$\min_{f\in\mathscr{F}} \text{Risk}_{\text{srm}}(f) := \text{Risk}_{\text{emp}}(f) + \lambda J(f) \text{ where } \text{Risk}_{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N}\text{loss}(y_i, f(x_i))$$

(Occam 剃刀原理): 在所有可供选择的模型中，能够很好地解释已知数据并且十分简单才是最好的模型.
从 Bayes 估计的角度来看，正则化项对应于模型的先验概率.
通常复杂的模型有较小的先验概率，简单的模型有较大的先验概率.

1.3.4 交叉验证

另一种常用的模型选择方法是**交叉验证** (cross validation):

若给定的样本数据充足，则可以随机地将数据集切分成三部分，
分别为训练集 (training set)、验证集 (validation set) 和测试集(test set).
训练集用来训练模型，验证集用于模型的选择，而测试集用于最终对学习方法的评估.

但很多实际应用中，数据是不充足的，此时可以使用交叉验证.
其基本思想是重复地使用数据
即按多种划分方式从训练集中划分出验证集，取多次验证的平均结果用来模型选择.

- **k-折交叉验证:**
  随机地将数据划分为 k 个互不相交、大小相同的子集
  然后用其中 $k-1$ 个子集训练模型，用剩下的子集验证模型
  如此反复多次，最后选出平均验证误差最小的模型.

  特殊地，当 $k=N$ 时称为**留一交叉验证**.

1.3.5 自助法

自助法 (bootstrapping) 从给定的包含 m 个样本的数据集 D 中有放回地采样 m 次得到数据集 D'
其中样本在 m 次采样中始终不被采到地概率是 $(1-\frac{1}{m})^m \to \frac{1}{e} \approx 0.368$
最终初始数据集 D 中的一部分样本会在 D' 中多次出现，而另一部分样本不出现.
此时我们将 D' 作为训练集，而 $D \backslash D'$ 作为测试集.

自助法在数据集较小、难以有效划分时很有用.
但其缺点是改变了初始数据集的分布，这会引入估计偏差.
因此在初始数据量足够时，留出法和交叉验证法更常用.

1.4 泛化误差

设假设模型的输入输出为随机变量 (X, Y)，且服从联合概率分布 $p(x, y)$
模型 f 的泛化误差即为期望风险:

$$\begin{aligned}\mathrm{Risk}_{\exp}(f) &= \mathrm{E}_p[\mathrm{loss}(Y, f(X))] \\ &= \int_{\mathcal{X}\times\mathcal{Y}} \mathrm{loss}(y, f(x)) p(x, y) \mathrm{d}x\mathrm{d}y\end{aligned}$$

学习方法的泛化能力分析往往是通过研究泛化误差的概率上界进行的
简称为**泛化误差上界** (generalization error bound)
换言之，我们通过比较两种学习方法的泛化误差上界的大小来比较它们的优劣.

泛化误差上界通常具有以下性质:

- 它是训练样本容量 N 的函数; 当样本容量增加时，泛化上界趋于零
- 它是假设空间 $\mathscr{F}$ 容量的函数; 假设空间容量越大，模型就越难学，泛化误差上界就越大.

考虑二分类问题的泛函误差上界 (一般的假设空间要找到泛化误差上界比较困难，我们不作介绍)
输入空间 $\mathcal{X} = \mathbb{R}^n$，输出空间 $Y = \{-1, 1\}$，假设空间为由有限集合 $\mathscr{F} = \{f_1, \dots, f_d\}$
损失函数为 0-1 损失函数 $\mathrm{loss}(y, f(x)) = \begin{cases} 0 & f(x) = y \\ 1 & \mathrm{otherwise} \end{cases}$
设训练数据集 $D_{\mathrm{train}} = \{(x_i, y_i) : i = 1, \dots, N\}$ 是从随机变量 (X, Y) 的联合概率分布 $p(x, y)$ 独立同分布产生的.

模型 $f \in \mathscr{F}$ 的期望风险 (泛化误差) 为:

$$\begin{aligned}\mathrm{Risk}_{\exp}(f) &= \mathrm{E}_p[\mathrm{loss}(Y, f(X))] \\ &= \int_{\mathcal{X}\times\mathcal{Y}} \mathrm{loss}(y, f(x)) p(x, y) \mathrm{d}x\mathrm{d}y\end{aligned}$$

模型 $f \in \mathscr{F}$ 的经验风险 (训练误差) 为:

$$\mathrm{Risk}_{\mathrm{emp}}(f) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{loss}(y_i, f(x_i))$$

(泛化误差上界, 统计学习方法 定理 1.1)
对于上述二分类问题，至少依概率 $1-\delta$ $(0 < \delta < 1)$ 有如下不等式成立:

$$\mathrm{Risk}_{\exp}(f) \le \mathrm{Risk}_{\mathrm{emp}}(f) + \varepsilon(d, N, \delta) \text{ for all } f \in \mathscr{F} = \{f_1, \dots, f_d\}$$

$$\text{where } \varepsilon(d, N, \delta) = \sqrt{\frac{1}{2N}(\log(d) - \log(\delta))}$$

其中 d 为假设空间中模型的个数，N 为训练样本量.

- 上述定理表明:
  - 训练误差 $\mathrm{Risk}_{\mathrm{emp}}(f)$ 越小，泛化误差上界也越小.
  - 第二项 $\varepsilon(d,N,\delta)$ 关于 N 严格单调递减，且当 $N\to\infty$ 时趋于零.
    因此当训练样本量 N 趋近于正无穷时，泛化误差上界趋近于泛化误差.
  - 第二项 $\varepsilon(d,N,\delta)$ 是 $O(\sqrt{log(d)})$ 阶的函数.
    因此假设空间 $\mathscr{F}$ 包含的函数越多，泛化误差上界也越大.

> (引理: Hoeffing 不等式)
> 设 $X_1,\dots,X_n$ 为独立随机变量，且 $X_i\in[a_i,b_i]\ (i=1,\dots,n)$
> 定义随机变量 $\bar{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$ (仍是随机变量)
> 则对于任意 $t>0$ 都有以下不等式成立:
>
> $$\mathrm{P}\{\bar{X}-\mathrm{E}(\bar{X})\geq t\}\leq\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right)$$
>
> $$\mathrm{P}\{\mathrm{E}(\bar{X})-\bar{X}\geq t\}\leq\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right)$$

证明:
任意给定模型 $f\in\mathscr{F}$

注意到 $\mathrm{loss}(y_1,f(x_1)),\dots,\mathrm{loss}(y_N,f(x_N))$ 是 N 个独立的随机变量，且有 $\mathrm{loss}(y_i,f(x_i))\in[0,1]\ (i=1,\dots,n)$
经验风险 $\mathrm{Risk}_{\mathrm{emp}}(f)=\frac{1}{N}\sum_{i=1}^{N}\mathrm{loss}(y_i,f(x_i))$ 是它们的经验均值 (仍是随机变量)
其期望即为泛化误差: $\mathrm{E}[\mathrm{Risk}_{\mathrm{emp}}(f)]=\mathrm{Risk}_{\mathrm{exp}}(f)$

根据第一条 Hoeffding 不等式可知，对于任意 $\varepsilon>0$ 都有:

$$\mathrm{P}\{\mathrm{Risk}_{\mathrm{exp}}(f)-\mathrm{Risk}_{\mathrm{emp}}(f)\leq\varepsilon\}\leq\exp\left(-\frac{2N^2\varepsilon^2}{N\cdot(1-0)^2}\right)=\exp\left(-2N\varepsilon^2\right)$$

要保证至少依 $1-\delta$ 概率有 "从假设空间 $\mathscr{F}=\{f_1,\dots,f_d\}$ 中任取一个模型有 $\mathrm{Risk}_{\mathrm{emp}}(f)-\mathrm{Risk}_{\mathrm{exp}}(f)\leq\varepsilon$ 成立"
我们只需令 $\delta=d\cdot\exp\left(-2N\varepsilon^2\right)$ 即可.
解得 $\varepsilon=\varepsilon(d,N,\delta)=\sqrt{\frac{1}{2N}(\log(d)-\log(\delta))}$ (保留正根)
因此我们有:

$$\mathrm{P}\{\mathrm{Risk}_{\mathrm{emp}}(f)-\mathrm{Risk}_{\mathrm{exp}}(f)\leq\varepsilon(d,N,\delta)\text{ for all }f\in\mathscr{F}=\{f_1,\dots,f_d\}\}\leq1-\delta$$

$$\text{where }\varepsilon(d,N,\delta)=\sqrt{\frac{1}{2N}(\log(d)-\log(\delta))}$$

即至少依概率 $1-\delta$ 有如下不等式成立:

$$\mathrm{Risk}_{\mathrm{exp}}(f)\leq\mathrm{Risk}_{\mathrm{emp}}(f)+\varepsilon(d,N,\delta)\text{ for all }f\in\mathscr{F}=\{f_1,\dots,f_d\}$$

$$\text{where }\varepsilon(d,N,\delta)=\sqrt{\frac{1}{2N}(\log(d)-\log(\delta))}$$

命题得证.

1.5 评价指标

为了衡量一个机器学习模型的好坏，需要给定一个测试集，
用模型对测试集中的每一个样本进行预测，并根据预测结果计算评价分数.

考虑分类问题，给定测试集 $D_{\text{test}} = \{(x^{(i)}, y_i)\}_{i=1}^{n}$
假设标签 $y_i \in \{1, \dots, m\}$,
我们使用学习好的模型 $f(x; \theta^\star)$ 对测试集中的每个样本进行预测，结果为 $\{\hat{y}^{(i)}\}_{i=1}^{n}$

- **(1) 正确率 (Accuracy):**
  $$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^{n} I(\hat{y}_i = y_i)$$
- **(2) 错误率 (Error Rate):**
  $$\text{Error Rate} = 1 - \text{Accuracy} = \frac{1}{n} \sum_{i=1}^{n} I(\hat{y}_i \neq y_i)$$
- **(3) 精确率 (Precision) & 召回率 (Recall):**
  对于类别 $c \in \{1, \dots, m\}$ 来说，模型在测试集上的结果分为四种情况.
  我们使用**混淆矩阵** (confusion matrix) 来表示：

表 2.3　类别 c 的预测结果的混淆矩阵

		预测类别	
		$\hat{y} = c$	$\hat{y} \neq c$
真实类别	$y = c$	TP_c	FN_c
	$y \neq c$	FP_c	TN_c

  - **真正例 (True Positive, TP):**
    一个样本真实类别为 c 且模型预测正确.
    这类样本数量记为 $\text{TP}_c = \sum_{i=1}^{n} I(y_i = \hat{y}_i = c)$
  - **真负例 (True Negative, TN):**
    一个样本真实类别不为 c 且模型预测正确. (这种情况通常不需要关注)
    这类样本数量记为 $\text{TN}_c = \sum_{i=1}^{n} I(y_i = \hat{y}_i \neq c)$
  - **假正例 (False Positive, FP):**
    一个样本真实类别不为 c，但模型预测为 c，预测不正确.
    这类样本数量记为 $\text{FP}_c = \sum_{i=1}^{n} I(y_i \neq c \wedge \hat{y}_i = c)$
  - **假负例 (False Negative, FN):**
    一个样本真实类别为 c，但模型预测不为 c，预测不正确.
    这类样本数量记为 $\text{FN}_c = \sum_{i=1}^{n} I(y_i = c \wedge \hat{y}_i \neq c)$

  我们进一步定义：

  - 类别 c 的**精确率 (Precision):** (又称查准率)
    所有预测为 c 类的样本中预测正确的比例：$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}$
  - 类别 c 的**召回率 (Recall):** (又称查全率)
    所有真实为 c 类的样本中预测正确的比例：$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$

- 类别 c 的 F **值 (F Measure):**
  $\mathrm{F}_c = \left[\frac{1}{1+\beta^2}\frac{1}{\mathrm{Precision}_c} + \frac{\beta^2}{1+\beta^2}\frac{1}{\mathrm{Recall}_c}\right]^{-1} = \frac{(1+\beta^2)\cdot\mathrm{Precision}_c\cdot\mathrm{Recall}_c}{\beta^2\mathrm{Precision}_c+\mathrm{Recall}_c}$
  其中 β 用于平衡精确率和召回率的重要性.
  通常取 $\beta=1$，记为 $\mathrm{F1}_c = \frac{2}{\frac{1}{\mathrm{Precision}_c}+\frac{1}{\mathrm{Recall}_c}} = \frac{2\cdot\mathrm{Precision}_c\cdot\mathrm{Recall}_c}{\mathrm{Precision}_c+\mathrm{Recall}_c}$，
  即精确率和召回率的**调和平均**.
- **宏平均 (Macro Average):**
  即每一类的性能指标的算术平均
  $$\begin{cases}\mathrm{precision}_{\mathrm{macro}} = \frac{1}{m}\sum\limits_{i=1}^{m}\mathrm{precision}_c \\ \mathrm{Recall}_{\mathrm{macro}} = \frac{1}{m}\sum\limits_{i=1}^{m}\mathrm{Recall}_c \\ \mathrm{F1}_{\mathrm{macro}} = \frac{2}{\frac{1}{\mathrm{Precision}_{\mathrm{macro}}}+\frac{1}{\mathrm{Recall}_{\mathrm{macro}}}} = \frac{2\cdot\mathrm{Precision}_{\mathrm{macro}}\cdot\mathrm{Recall}_{\mathrm{macro}}}{\mathrm{Precision}_{\mathrm{macro}}+\mathrm{Recall}_{\mathrm{macro}}}\end{cases}$$
  (部分文献定义 $\mathrm{F1}_{\mathrm{macro}} = \frac{1}{m}\sum\limits_{i=1}^{m}\mathrm{F1}_c$)

(ROC 曲线和 AUC)

ROC 曲线的纵轴为 $\mathrm{TPR} = \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$，横轴为 $\mathrm{FPR} := \frac{\mathrm{FP}}{\mathrm{TN+FP}}$
而 AUC 是 **ROC 曲线下的面积** (Area Under Curve)，其取值范围为 $[0,1]$
最差的情况是随机猜测，AUC 理论上是 0.5;
若 AUC 很接近于 0，说明模型可能出错了，它很 "准确" 地将正例预测为负例，将负例预测为正例.
如果把它的预测取反，其 AUC 就很接近于 1 了.

如果一条 ROC 曲线将另一个包裹 (更往左上角凸)，则前者一致地优于后者.
(我们希望 TPR 越大越好，FPR 越小越好，这里有一个 trade-off)

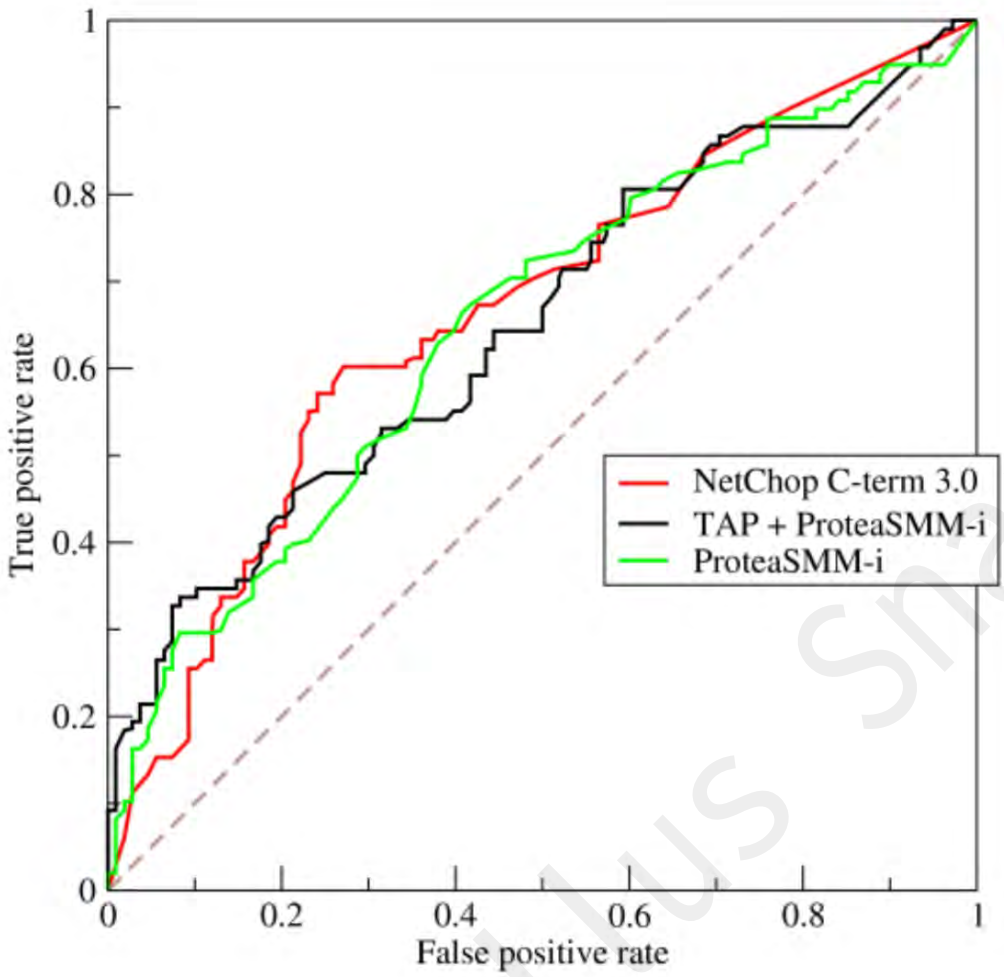
有限样本集上不同分类器的 ROC 曲线:
(设其样本量为 N，正例和负例的集合为 D^+ 和 D^-，个数分别为 N^+, N^-)

- ① 设分类阈值为 1，则 $(\mathrm{FPR},\mathrm{TPR}) = (0,0)$
- ② 将预测值从高到低排序，将阈值依次设为预测值，分别计算 $(\mathrm{FPR},\mathrm{TPR})$ (可以利用已经计算的结果快速更新)
  若当前样本是真正例，则向纵轴方向走 (FPR 部分不变，TPR 部分增长一个单位 $\frac{1}{N^+}$)
  若当前样本是假正例，则向横轴方向走 (FPR 部分增长一个单位 $\frac{1}{N^-}$，TPR 部分不变)

排序损失 l_{rank} 和曲线下面积 AUC 的计算公式为:

$$l_{\mathrm{rank}} := \frac{1}{N^+N^-}\sum_{(x^+,1)\in D^+}\sum_{(x^-,0)\in D^-}\{\mathbb{1}(f(x^+)<f(x^-)) + \frac{1}{2}\mathbb{1}(f(x^+)=f(x^-))\}$$

$$\mathrm{AUC} := 1 - l_{\mathrm{ROC}} = \frac{1}{N^+N^-}\sum_{(x^+,1)\in D^+}\sum_{(x^-,0)\in D^-}\mathbb{1}(f(x^+)\geq f(x^-))$$



(成本收益曲线)
纵轴为收益 (召回率 $\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$)，横轴为成本 (覆盖率 $\frac{\mathrm{TP+FP}}{\mathrm{P+N}}$)

- **(4) 交叉验证 (Cross-Validation):**
  为避免划分训练集和测试集时的随机性对评价结果造成的影响，
  我们可以把原始数据集平均分为 $k \geq 3$ 组不重复的子集，
  每次选 $k-1$ 组子集作为训练集，剩下的 1 组子集作为验证集.
  这样可以进行 k 次试验并得到 k 个模型，
  将这 k 个模型在各自验证集上的错误率的平均作为分类器的评价.

- **(5) 类别不均衡**
  假设阈值 τ 满足 $\frac{\tau}{1-\tau} > \frac{N^+}{N^-}$ (这意味着正例不足)
  - ① 过采样: 对已有正例插值得到新的正例
  - ② 欠采样: 将负例划分成多份，分别与正例学习，进行模型集成.
    (这样从全局来看不会丢失样本信息)

The End