MACHINE LEARNING

机器学习

Model Evaluation

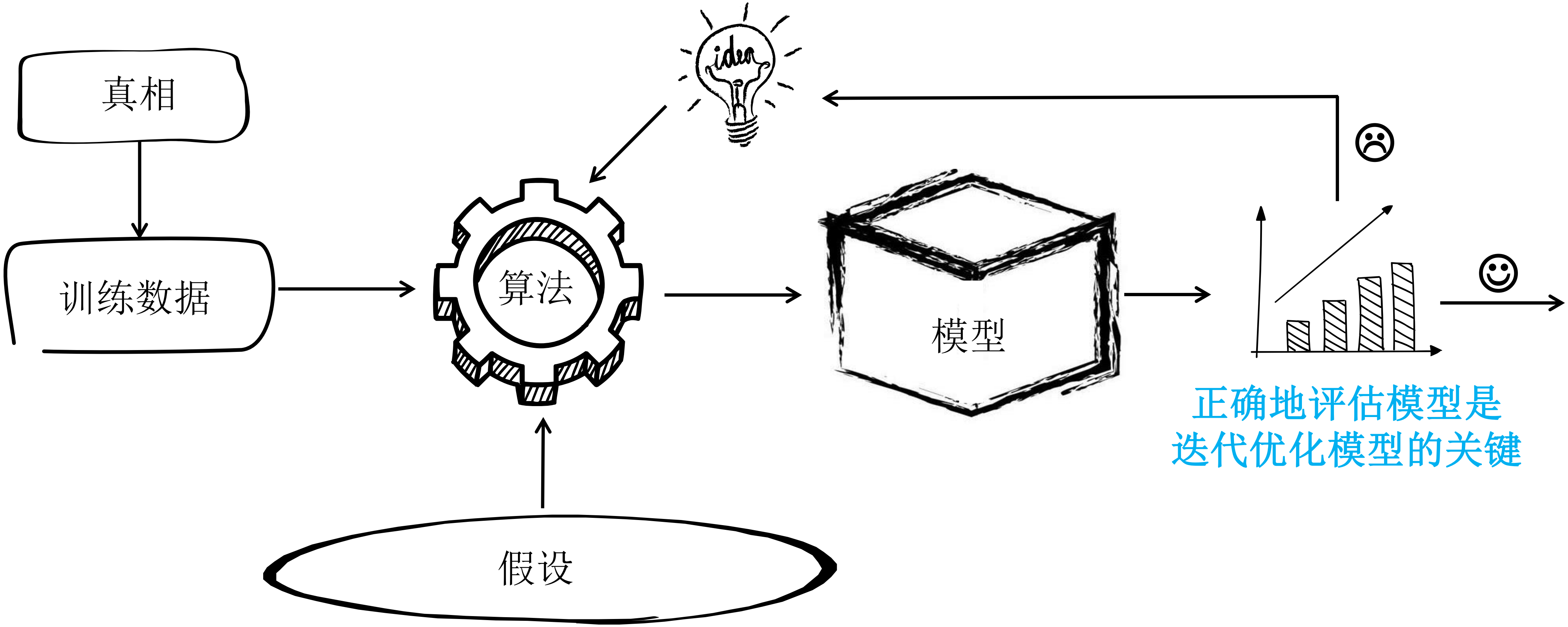
模型评估与选择

参考：
《机器学习》

Machine Learning Course

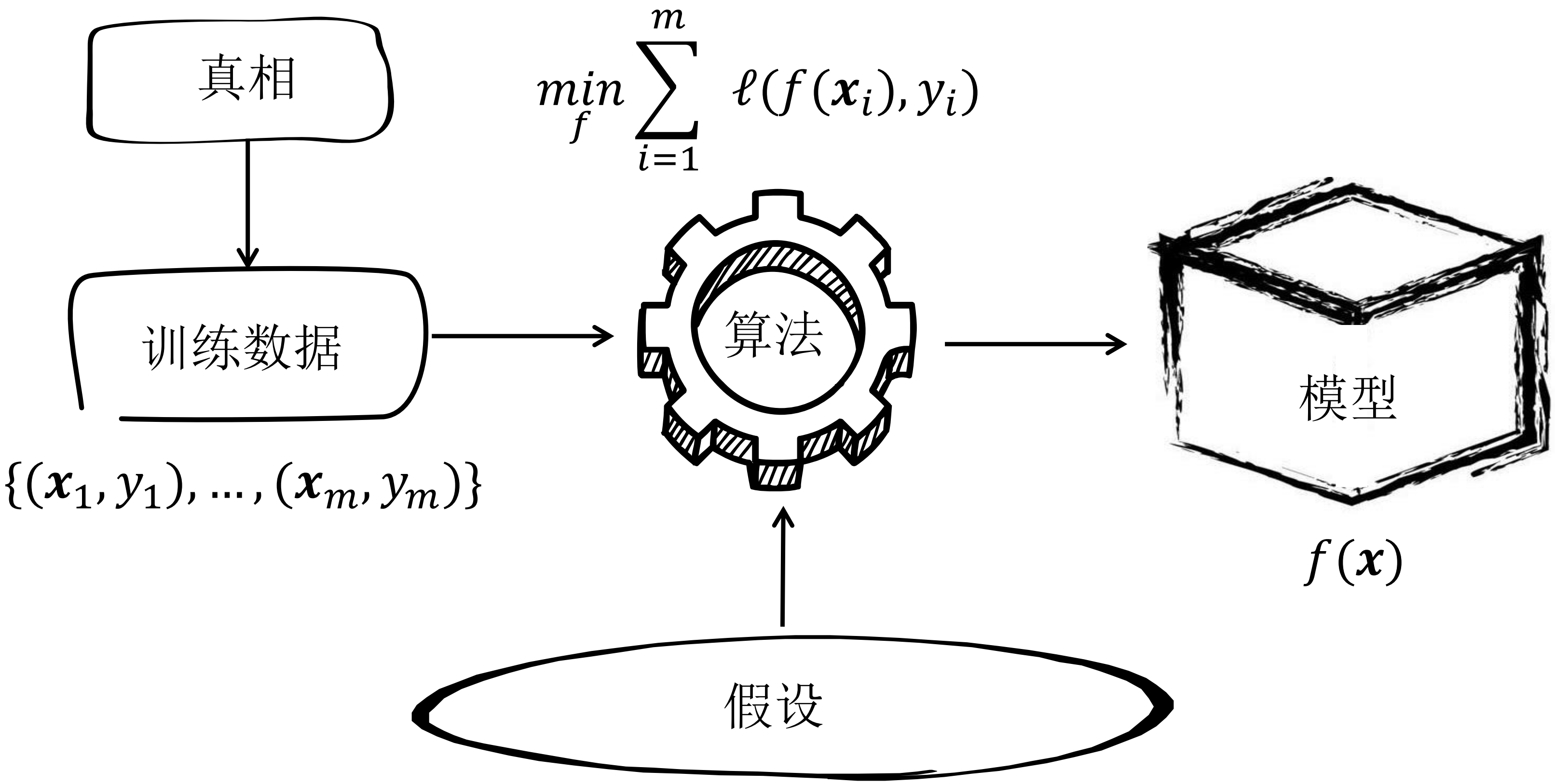
模型评估

- 评估与调优



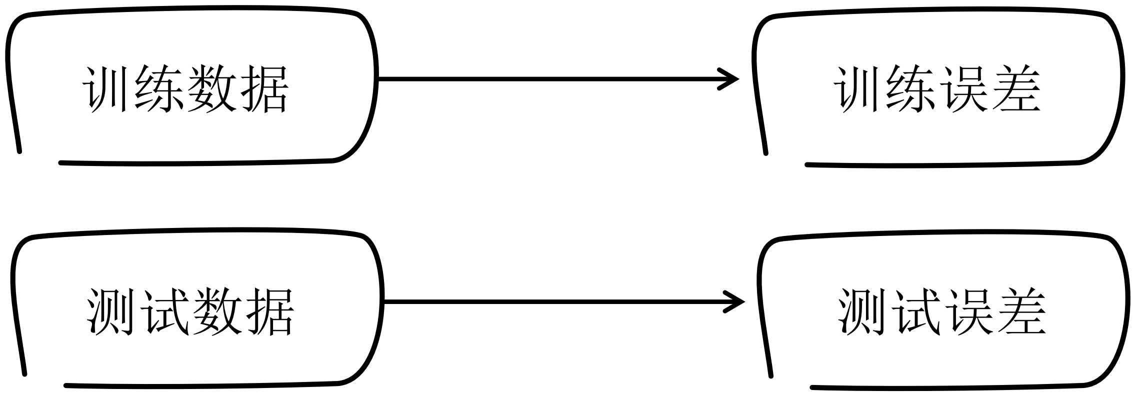
评估的对象

- 经验风险最小化



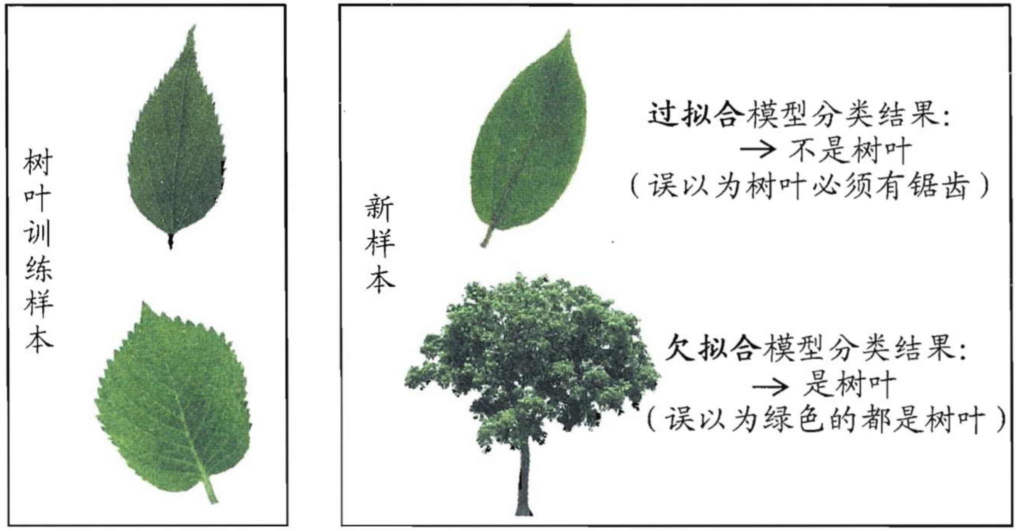
评估的对象

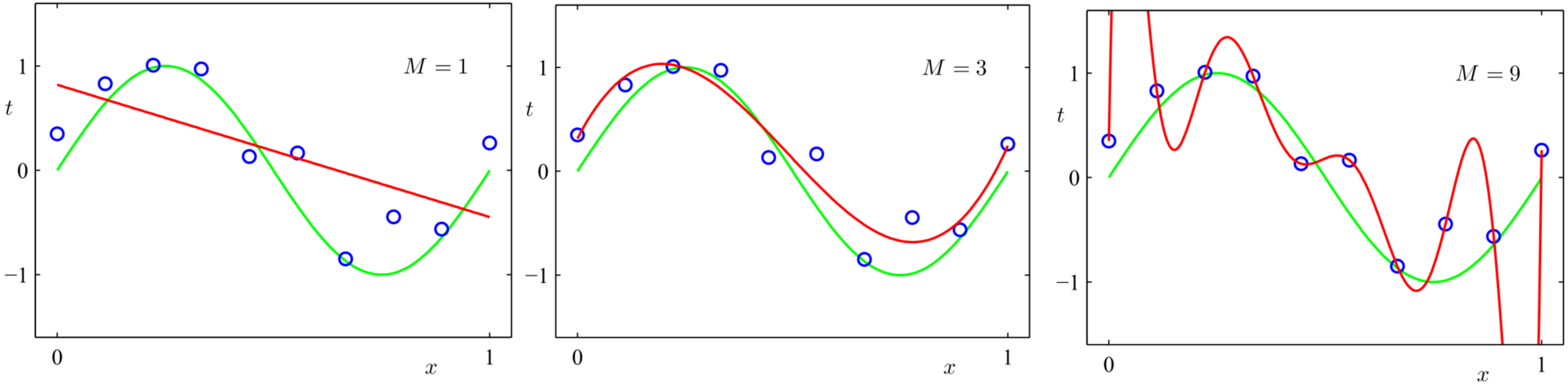
- 经验误差 vs 泛化误差



欠拟合 vs 过拟合

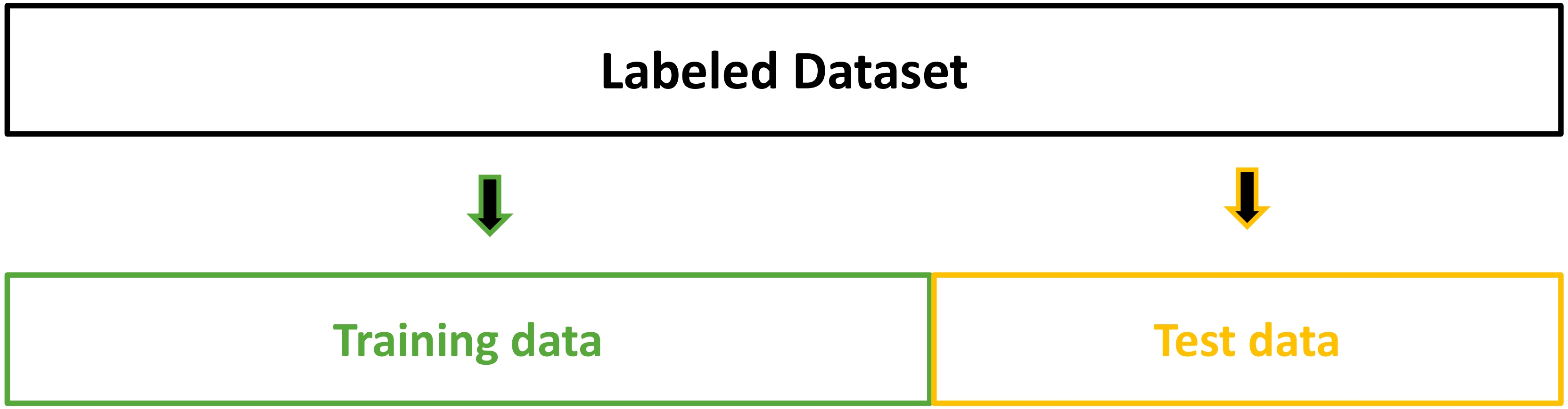
- 示例


树叶训练样本
新样本
过拟合模型分类结果:
→ 不是树叶
(误以为树叶必须有锯齿)
欠拟合模型分类结果:
→ 是树叶
(误以为绿色的都是树叶)


M = 1
M = 3
M = 9
t
x
1
0
−1

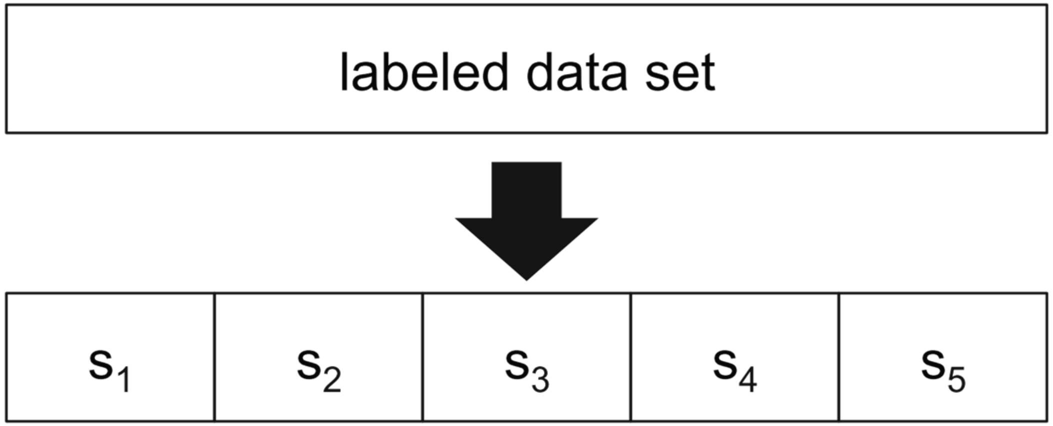
数据切分方法

- 留出法



数据切分方法

- 交叉验证法

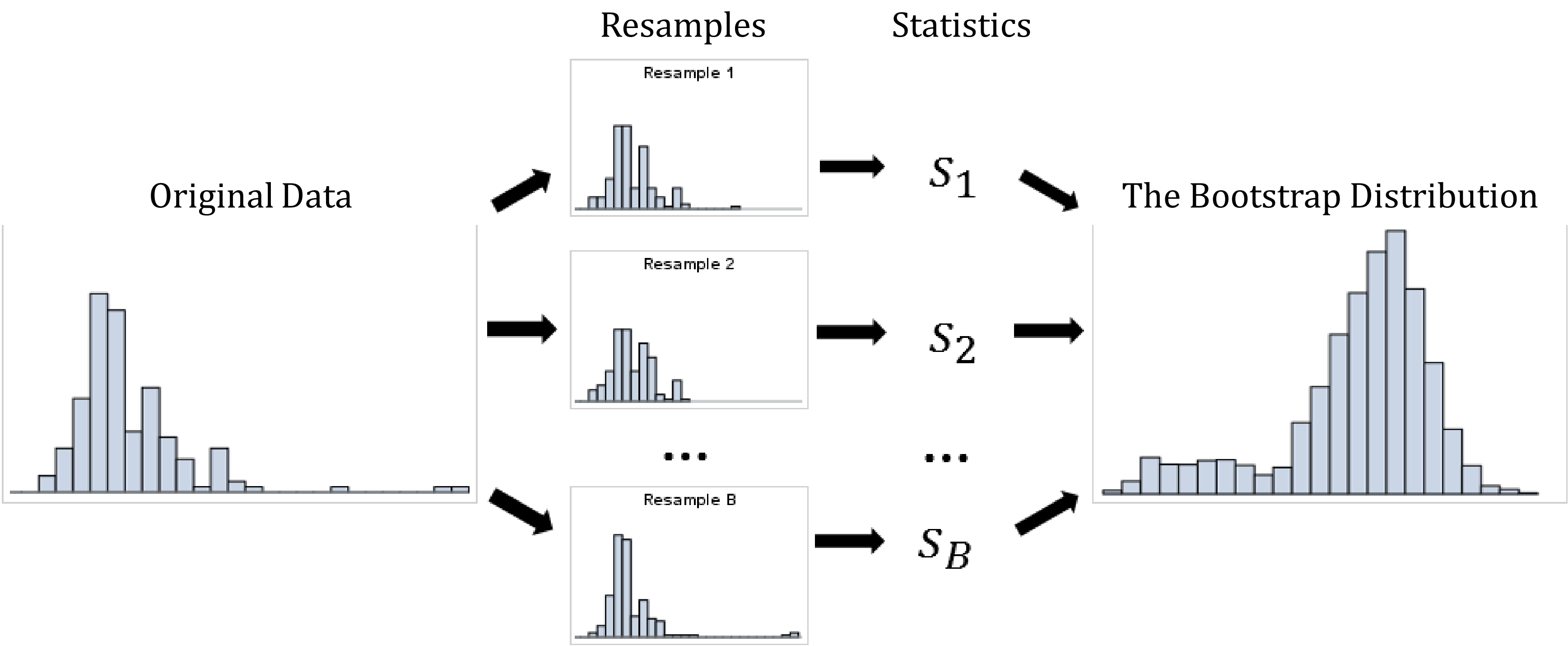


iteration	train on	test on
1	s_2 s_3 s_4 s_5	s_1
2	s_1 s_3 s_4 s_5	s_2
3	s_1 s_2 s_4 s_5	s_3
4	s_1 s_2 s_3 s_5	s_4
5	s_1 s_2 s_3 s_4	s_5

数据切分方法

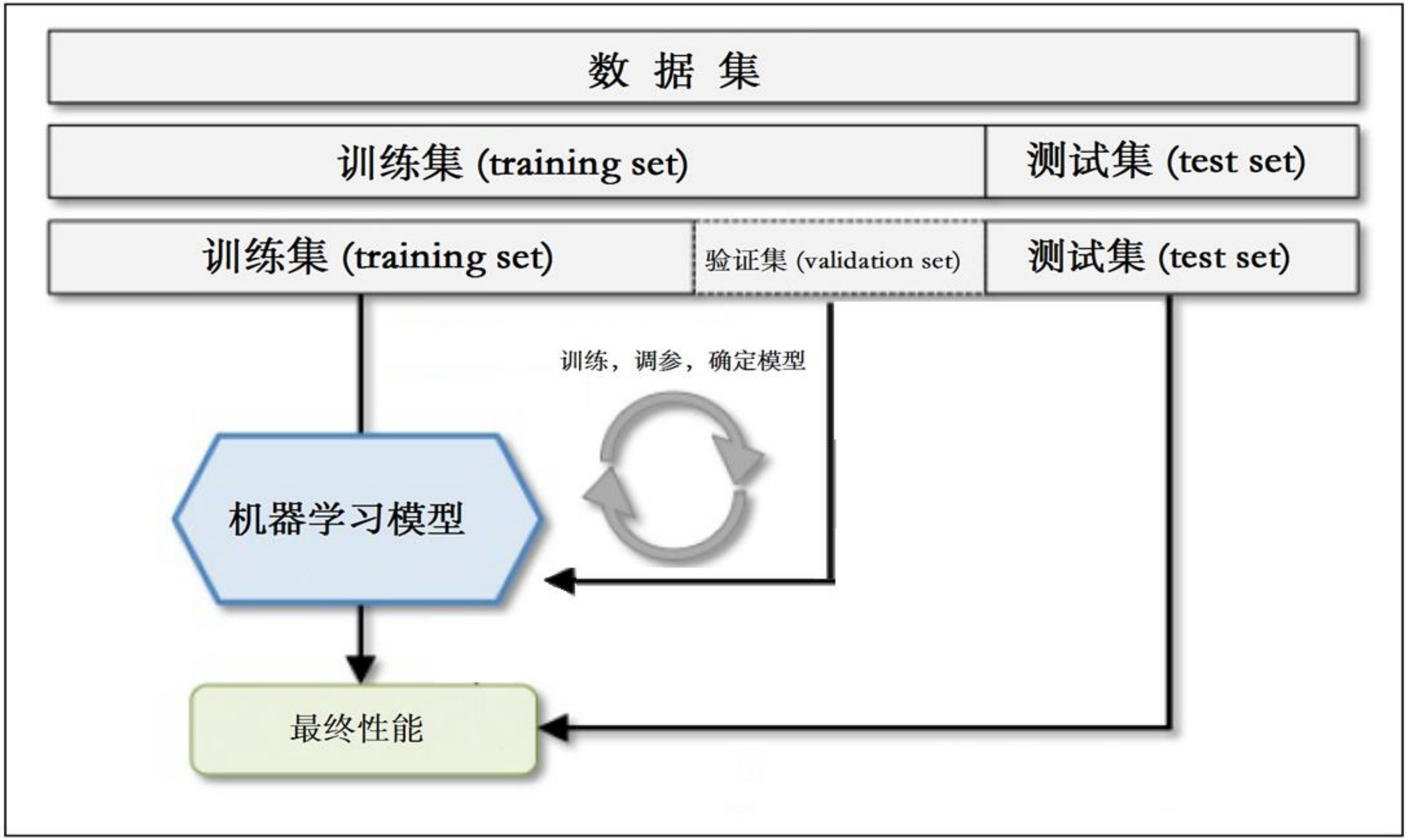
- 自助法

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



调参

- 训练集 + 验证集 + 测试集



性能度量

- 回归模型的性能度量

$$E(f;\mathcal{D}) = \int_{\boldsymbol{x}\sim\mathcal{D}} (f(\boldsymbol{x}) - y)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

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

性能度量

- 分类模型的性能度量

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f(\boldsymbol{x}_i) \neq y_i\right) \quad \overset{D}{\underset{p(\cdot)}{\Longleftrightarrow}} \quad E(f;\mathcal{D}) = \int_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{I}\left(f(\boldsymbol{x}) \neq y\right)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$\begin{aligned}\mathrm{acc}(f;D) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f(\boldsymbol{x}_i) = y_i\right) \\ &= 1 - E(f;D)\end{aligned} \qquad \begin{aligned}\mathrm{acc}(f;\mathcal{D}) &= \int_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{I}\left(f(\boldsymbol{x}) = y\right)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= 1 - E(f;\mathcal{D})\end{aligned}$$

性能度量

- 分类模型的性能度量
  - 混淆矩阵

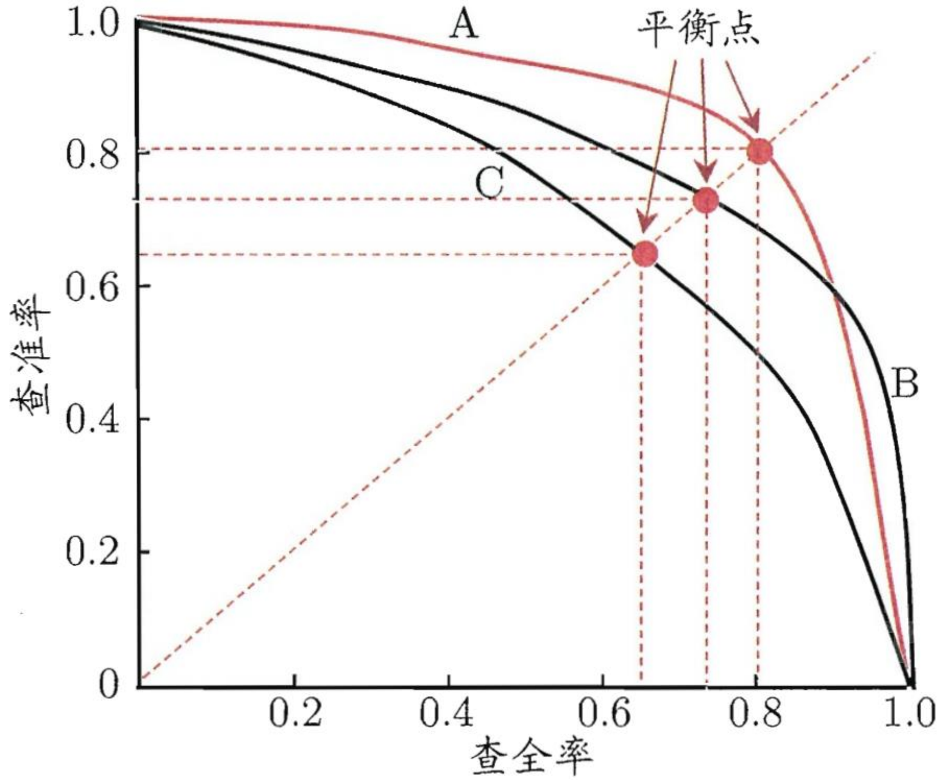
真实情况	预测结果	
	正例	反例
正例	TP(真正例)	FN(假反例)
反例	FP(假正例)	TN(真反例)

  - 查准率、查全率与F1 、PR曲线

$$P=\frac{TP}{TP+FP} \quad R=\frac{TP}{TP+FN}$$

$$F1=\frac{2\times P\times R}{P+R}=\frac{2\times TP}{\text{样例总数}+TP-TN}$$

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



性能度量

- 分类模型的性能度量
  - 多分类场景

两两类别的组合 1 ⟶ 混淆矩阵 1 ⟶ (P_1, R_1)

两两类别的组合 2 ⟶ 混淆矩阵 2 ⟶ (P_2, R_2)

... ...

两两类别的组合 n ⟶ 混淆矩阵 n ⟶ (P_n, R_n)

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

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

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

性能度量

- 分类模型的性能度量
  - 多分类场景

两两类别的组合 1 → 混淆矩阵 1

两两类别的组合 2 → 混淆矩阵 2

...   ...

两两类别的组合 n → 混淆矩阵 n

混淆矩阵 1, 混淆矩阵 2, …, 混淆矩阵 n → $\frac{\overline{TP}}{\overline{TN}}, \frac{\overline{FP}}{\overline{FN}}$

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

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

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

性能度量

- 分类模型的性能度量
  - 混淆矩阵

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)

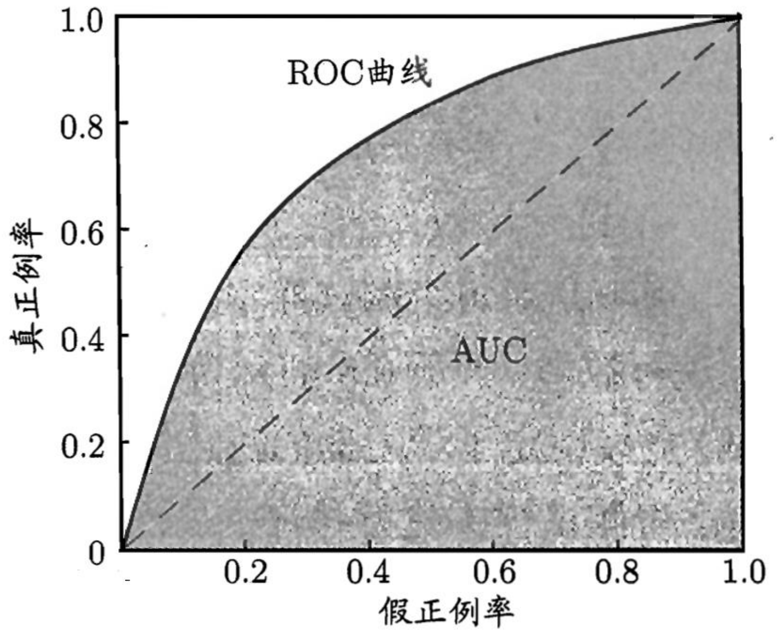
  - “真正例率” TPR 与“假正例率” FPR

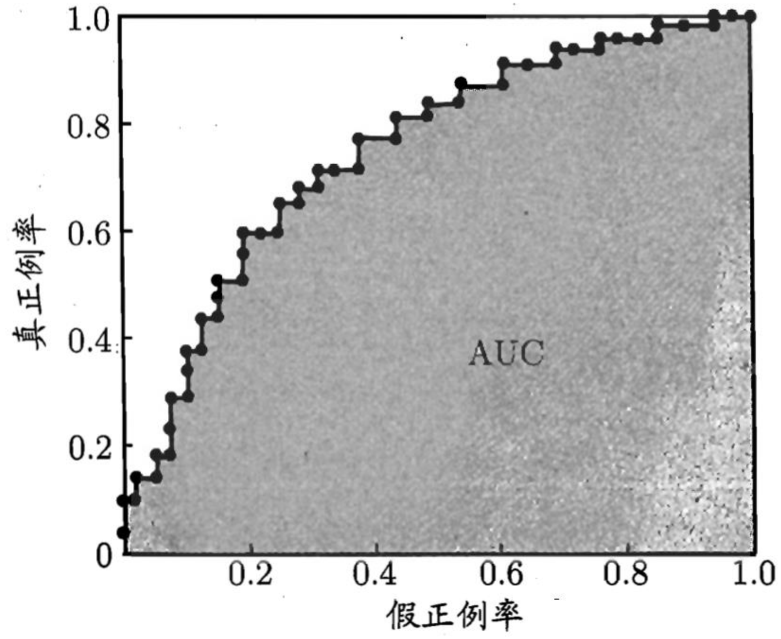
$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

$$\mathrm{FPR} = \frac{FP}{TN + FP}$$

性能度量

- 分类模型的性能度量
  - ROC 与 AUC



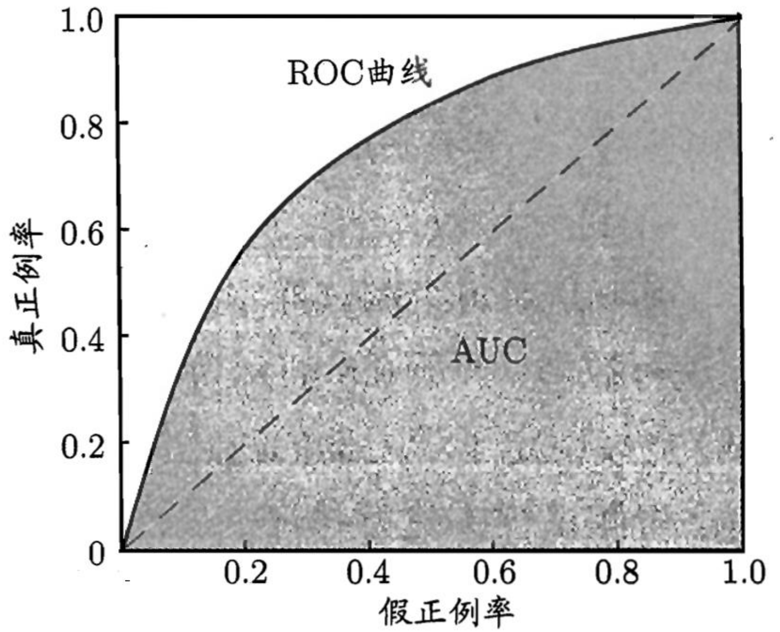


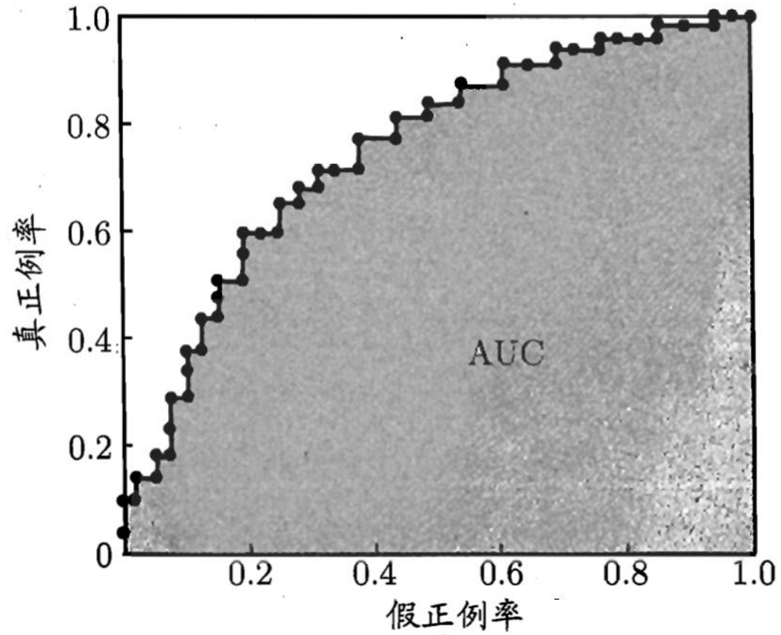
AUC ＝“ROC 曲线之下的面积”＝1 -“ROC 曲线之上的面积”＝ $1 - \ell_{\text{rank}}$

$$\ell_{\text{rank}} = \frac{1}{m^+ m^-} \sum_{\boldsymbol{x}^+ \in D^+} \sum_{\boldsymbol{x}^- \in D^-} \left(\mathbb{I}\left(f(\boldsymbol{x}^+) < f(\boldsymbol{x}^-)\right) + \frac{1}{2} \mathbb{I}\left(f(\boldsymbol{x}^+) = f(\boldsymbol{x}^-)\right) \right)$$

性能度量

- 分类模型的性能度量
  - ROC 与 AUC





思考：多分类场景下的ROC与AUC

性能度量

- 分类模型的性能度量
  - 代价敏感场景

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

$$E(f;D;cost)=\frac{1}{m}\Big(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}\left(f(\boldsymbol{x}_i)\neq y_i\right)\times cost_{01}+\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}\left(f(\boldsymbol{x}_i)\neq y_i\right)\times cost_{10}\Big)$$

性能度量

- 假设检验
  - 两种算法哪种错误率更低

Q: 泛化错误率 $\epsilon_i^A \leqslant \epsilon_i^B$

A:对k折交叉验证产生的对测试错误率：先对每对结果求差 $\Delta_i = \epsilon_i^A - \epsilon_i^B$，得到 $\Delta_1, \Delta_2, \ldots, \Delta_k$

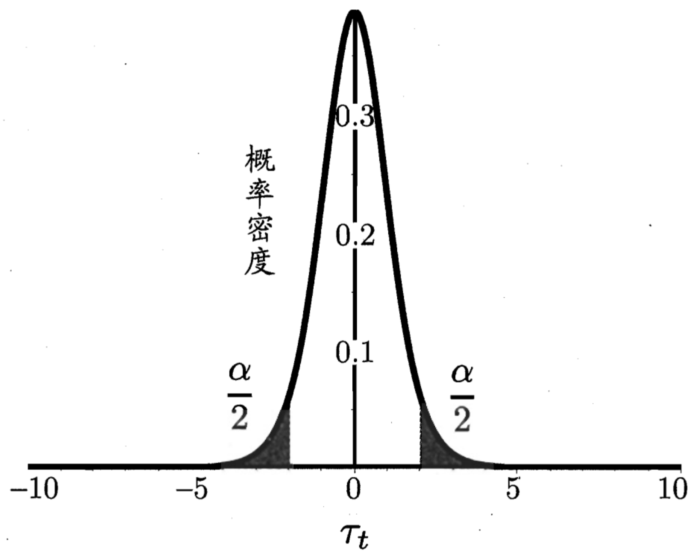
然后利用 t 检验：$\tau_t = \left| \frac{\sqrt{k}\mu}{\sigma} \right|$

差值的均值

差值的方差

服从自由度为 *k-1* 的 *t* 分布

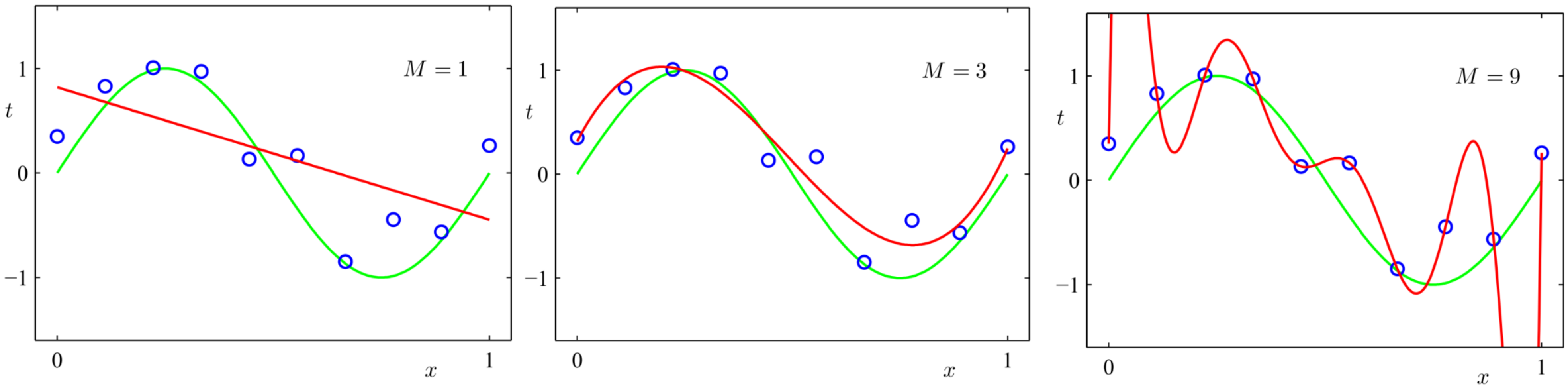
判定上述变量值是否小于临界值 $t_{\alpha/2,k-1}$，即尾部累积分布为α/2的临界值



欠拟合 vs 过拟合

- 示例

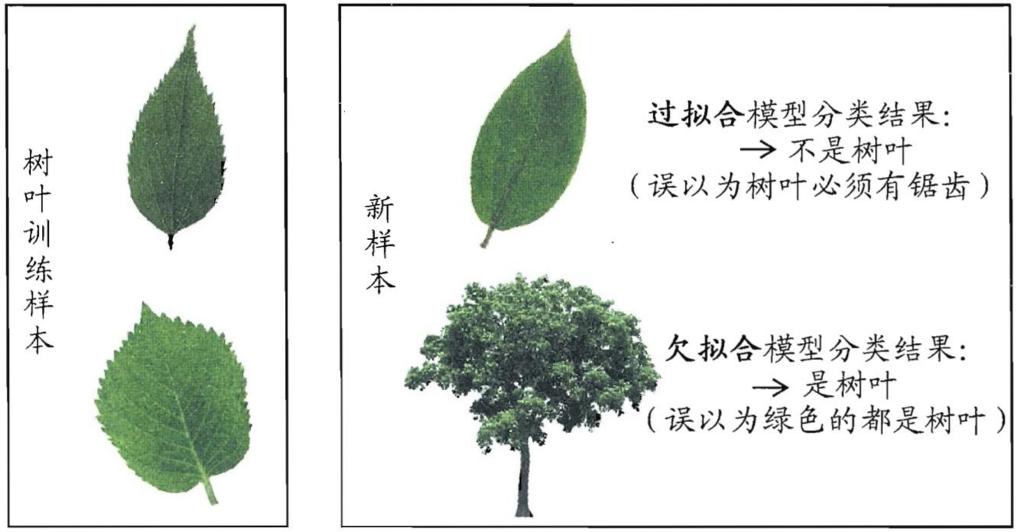
训练误差 ↔ 测试误差



欠拟合 vs 过拟合

- 示例

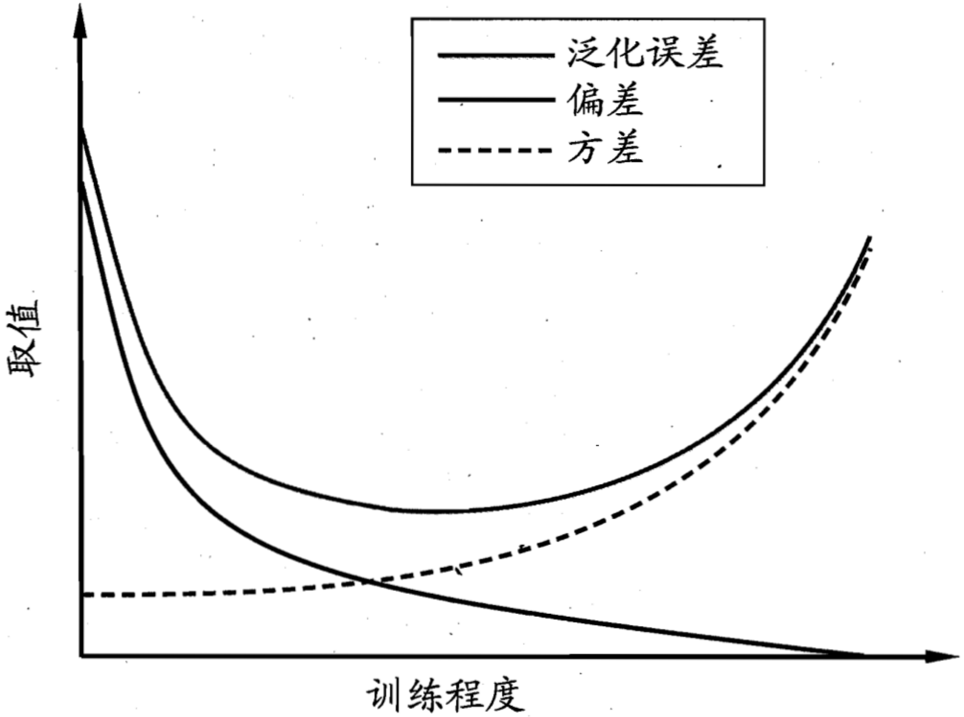
训练误差 ⟷ 测试误差



偏差与方差

- 偏差方差分解 (bias-variance decomposition)

$$E(f;D)=\mathbb{E}_D[(f(\boldsymbol{x};D)-y_D)^2]$$
$$=\mathbb{E}_D[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2]+(\bar{f}(\boldsymbol{x})-y)^2+\mathbb{E}_D[(y_D-y)^2]$$
$$=\text{bias}^2(\boldsymbol{x})+\text{var}(\boldsymbol{x})+\varepsilon^2$$



y_D : $\boldsymbol{x}$ 在数据集中的标记

$f(\boldsymbol{x};D)$:训练集 D上学得模型f在$\boldsymbol{x}$上的预测输出

$\bar{f}(\boldsymbol{x})=\mathbb{E}_D[f(\boldsymbol{x};D)]$:
以回归任务为例，学习算法的期望预测

$\text{var}(\boldsymbol{x})=\mathbb{E}_D\left[\left(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x})\right)^2\right]$:
使用样本数相同的不同训练集产生的方差为

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

$\text{bias}^2(\boldsymbol{x})=(\bar{f}(\boldsymbol{x})-y)^2$:
期望输出与真实标记的差别称为偏差

正则化技术

- 正则化 *regularization* 技术

$$\min_{f} \Omega(f) + C\sum_{i=1}^{m} \ell(f(\boldsymbol{x}_i), y_i)$$

结构风险用于描述模型的某些性质

经验风险用于描述模型与训练数据的契合程度;

- 岭回归

$$\min_{\boldsymbol{w},b} \parallel \boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{w}} \parallel_2^2 + \lambda \parallel \boldsymbol{w} \parallel_2^2 , \parallel \boldsymbol{w} \parallel_2^2 = \sum\nolimits_{j=1}^{d} {w_j}^2$$

$$\min_{\boldsymbol{w},b} (\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{w}}) + \lambda \parallel \boldsymbol{w} \parallel_2^2$$

$$\widehat{\boldsymbol{w}}_{\mathrm{RR}} = (\lambda \boldsymbol{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$$

岭回归与最大后验估计

- 岭回归的概率模型

> 最大似然估计

$q(\boldsymbol{x};\boldsymbol{\theta}) \longrightarrow p(\boldsymbol{x})$

$\mathcal{D} = \{\boldsymbol{x}\}_{i=1}^{n}$

Likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\boldsymbol{\theta}}{\mathrm{argmax}} p(\mathcal{D} \mid \boldsymbol{\theta}) = \mathrm{argmax} \prod_{i=1}^{n} q(\boldsymbol{x};\boldsymbol{\theta})$$

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\boldsymbol{\theta}\in\Theta}{\mathrm{argmax}} \log L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}\in\Theta}{\mathrm{argmax}} [\sum_{i=1}^{n} \log q(\boldsymbol{x}_i;\boldsymbol{\theta})]$$

$$p(\boldsymbol{x}) = q(\boldsymbol{x};\hat{\boldsymbol{\theta}}_{\mathrm{MLE}})$$

岭回归与最大后验估计

- 岭回归的概率模型

> 最大后验估计

$q(\boldsymbol{x};\boldsymbol{\theta}) \longrightarrow p(\boldsymbol{x})$

$\mathcal{D} = \{\boldsymbol{x}\}_{i=1}^{n}$

Likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$

Prior $p(\boldsymbol{\theta})$

Posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} p(\boldsymbol{\theta} \mid \mathcal{D})$$

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} (\sum_{i=1}^{n} \log q(\boldsymbol{x}_i \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}))$$

$$p(\boldsymbol{x}) = q(\boldsymbol{x};\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}})$$

岭回归与最大后验估计

- 岭回归的概率模型

$p(t \mid \boldsymbol{x})$?

$p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1})$，$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon$，$\epsilon \sim \mathcal{N}(0, \beta^{-1})$

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$$

$$p(\boldsymbol{w} \mid \mathbf{t}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_N, \boldsymbol{S}_N)$$

$$\boldsymbol{m}_N = \beta \boldsymbol{S}_N \boldsymbol{\Phi}^T \mathbf{t}, \boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

$$\boldsymbol{w}_{MAP}^* = \arg\max_{w} \ln p(\boldsymbol{w} \mid \mathbf{t})$$

$$= \arg\max_{w} \left(-\frac{\beta}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\}^2 - \frac{\alpha}{2} \boldsymbol{w}^T \boldsymbol{w} + \text{const}\right)$$

$$\boldsymbol{w}_{MAP}^* = \boldsymbol{w}_{RR}^*$$

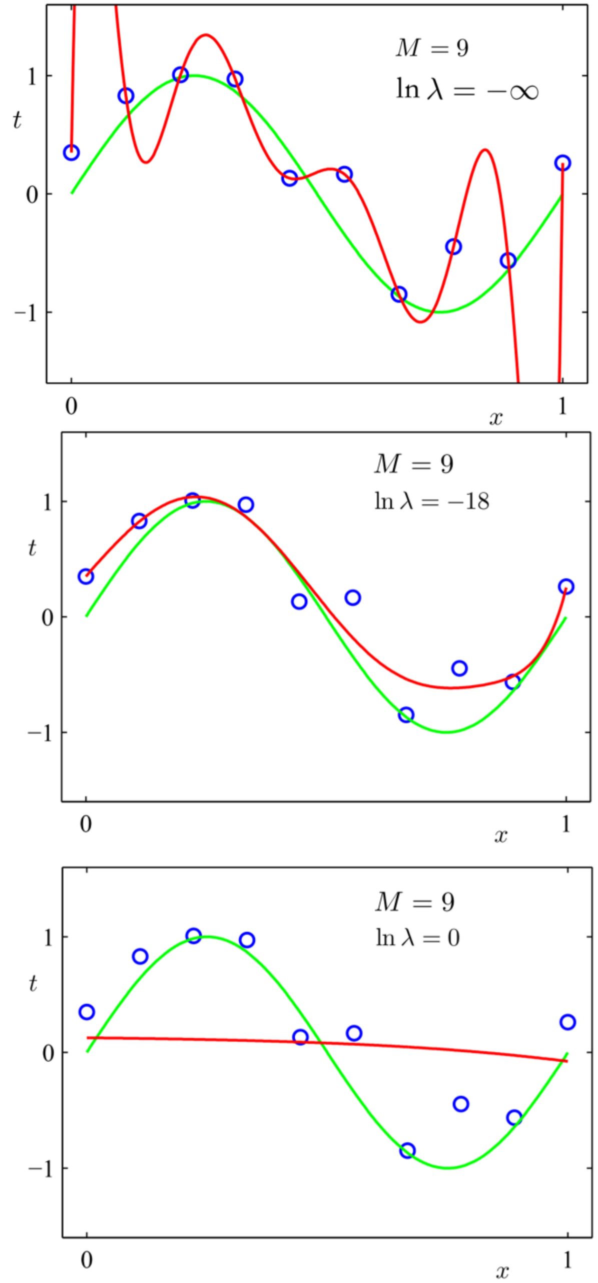
岭回归与LASSO

- 岭回归

$$\min_{\boldsymbol{w},b} \| \boldsymbol{y} - \mathbf{X}\boldsymbol{w} \|_2^2 + \lambda \| \boldsymbol{w} \|_2^2$$

$M = 9$

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

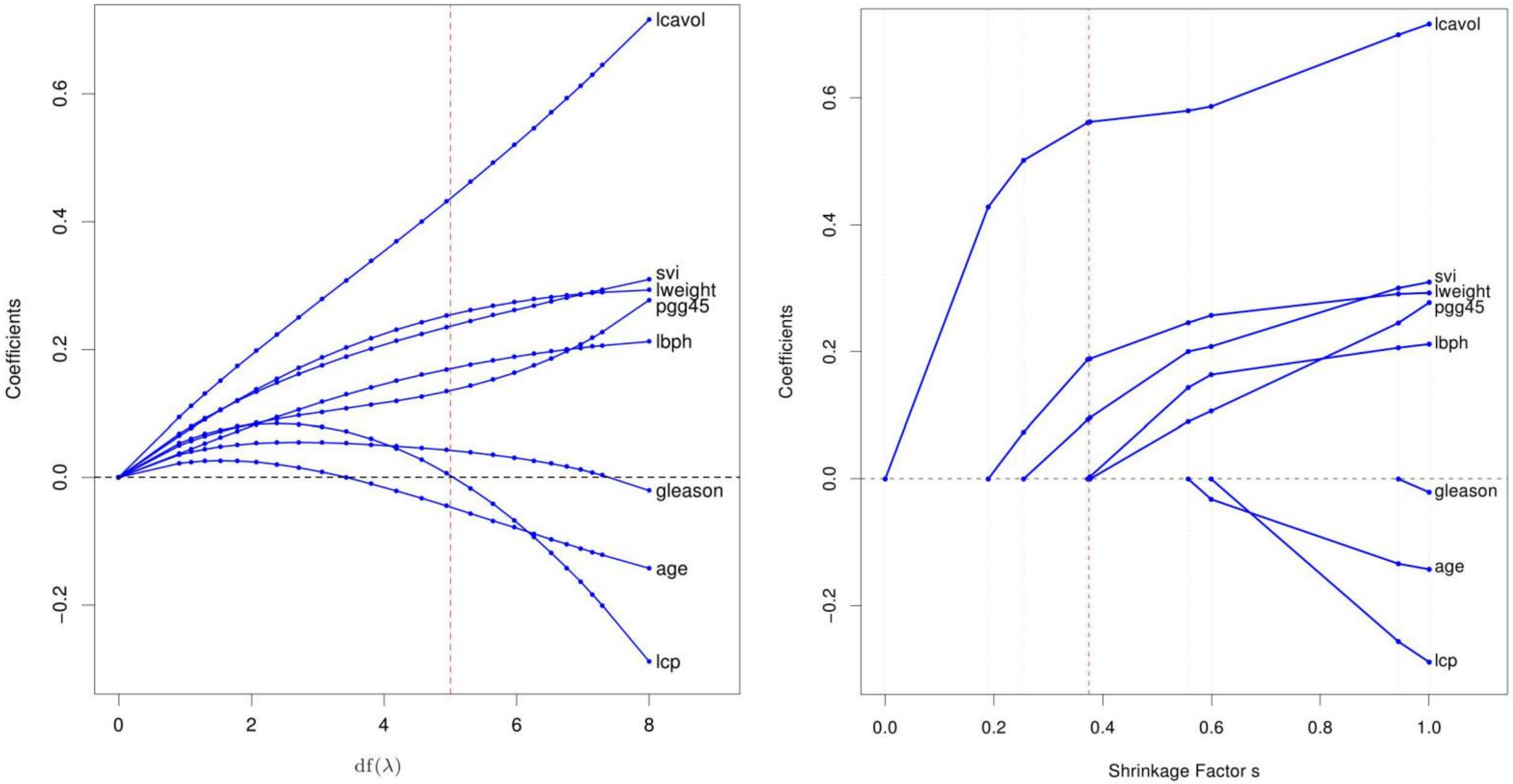


岭回归与LASSO

- LASSO

$$\min_{\boldsymbol{w},b} \| \boldsymbol{y} - \mathbf{X}\boldsymbol{w} \|_2^2 + \lambda \| \boldsymbol{w} \|_1$$

$$\| \boldsymbol{w} \|_1 = \sum_{j=1}^{d} |w_j|$$



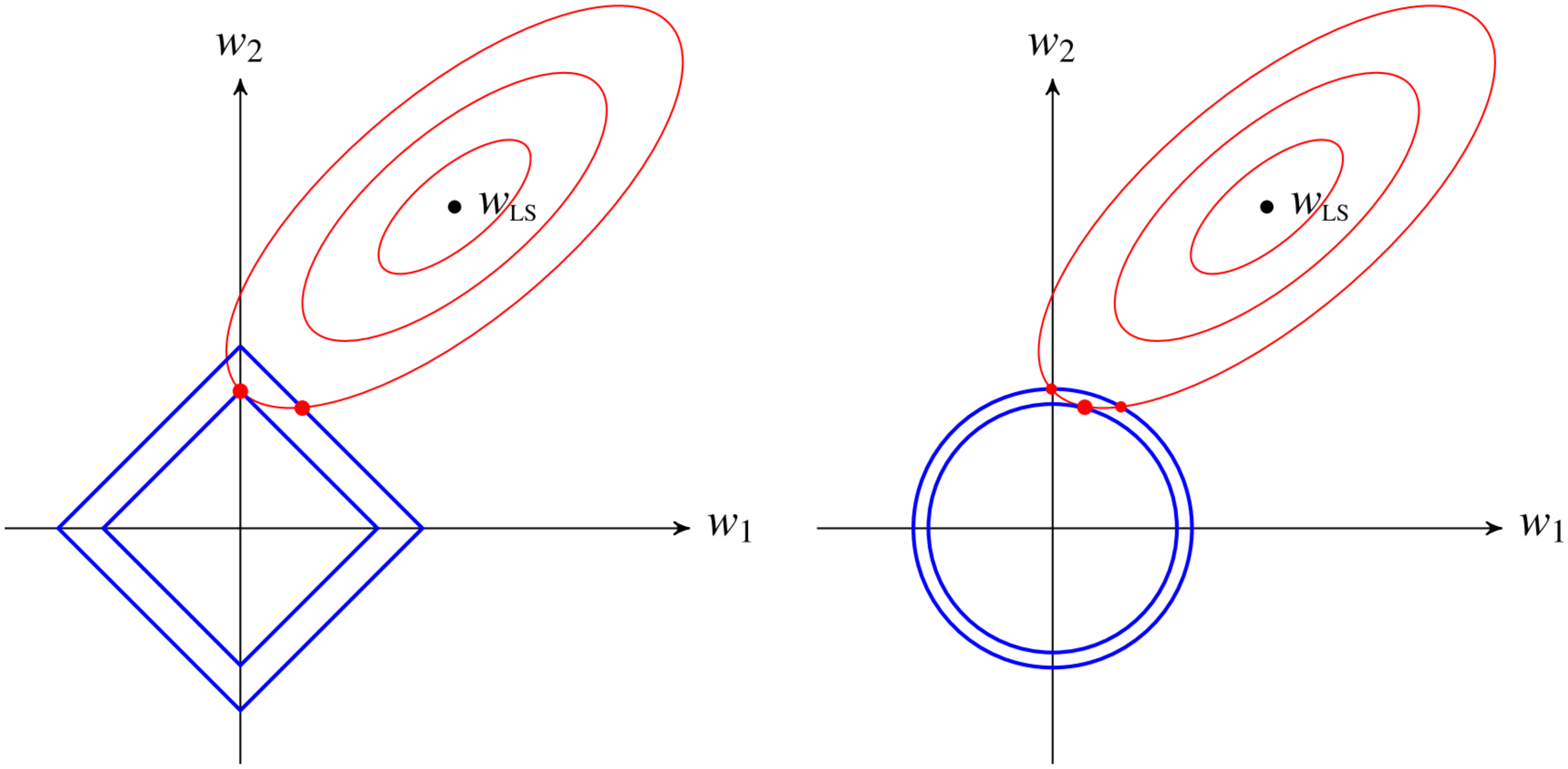
(a) $\|w\|_2$ penalty (b) $\|w\|_1$ penalty

岭回归与LASSO

- LASSO

$$\min_{\boldsymbol{w},b} \| \boldsymbol{y} - \mathbf{X}\boldsymbol{w} \|_2^2 + \lambda \| \boldsymbol{w} \|_1$$

$$\| \boldsymbol{w} \|_1 = \sum_{j=1}^{d} |w_j|$$



(a) $\|w\|_2$ penalty (b) $\|w\|_1$ penalty

MACHINE LEARNING

实践

Practice

参考：

https://scikit-learn.org/

程序示例

- 通过sklearn在线文档可以获得编程练习

3.1. Cross-validation: evaluating estimator performance

3.3. Metrics and scoring: quantifying the quality of predictions