

# 第四周 模型的评估与选择

李泽椿，复旦大学 生物医学工程与技术创新学院



# 目录

1 概述

2 数据集划分方法

3 模型评估方法

4 模型选择方法

5 大模型时代模型选择

# 背景——自动驾驶面临的挑战



rain

天气



多特征、高复杂度、多突发场景



行人



地势



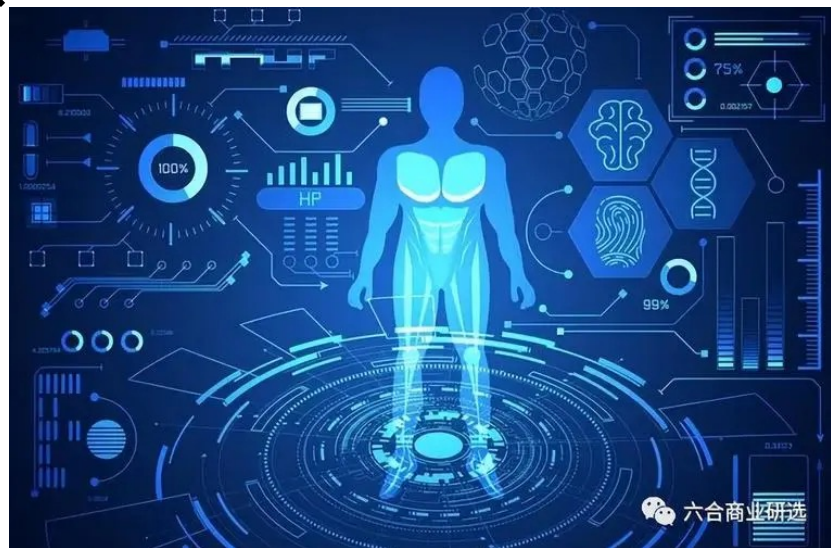
道路场景



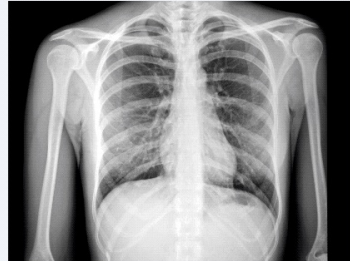
交通规则



# 背景——医疗大模型



Can you write me a report analyzing this chest X-ray?



**Findings:**

- Devices: None.
- Lungs: No pneumothorax. No substantial pleural effusion. Lungs appear clear.
- Cardiomeastinal: Normal heart size. Mediastinal contours within normal limits.
- Other: No acute skeletal abnormality.

**Impression:**

No active disease seen in chest.

Enter a question here

## 医疗大模型

年龄  
差异



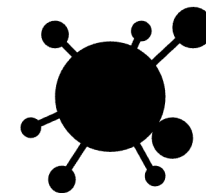
性别  
差异



特殊  
人群



罕见  
疾病



# 概述



金融领域  
收益率, 风险值



制造业  
生产效率, 质量指标



医疗领域  
治疗效果, 生理指标



人工智能  
?

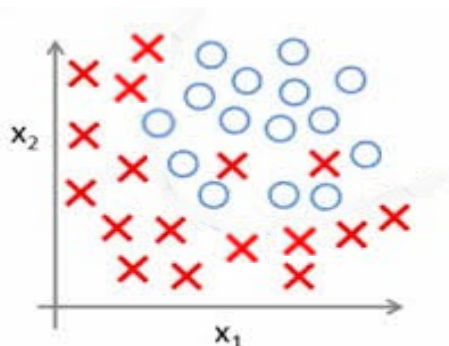
**Good or Bad**

**模型评估**

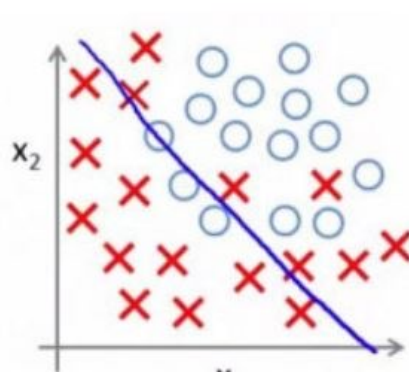
**&**

**模型选择**

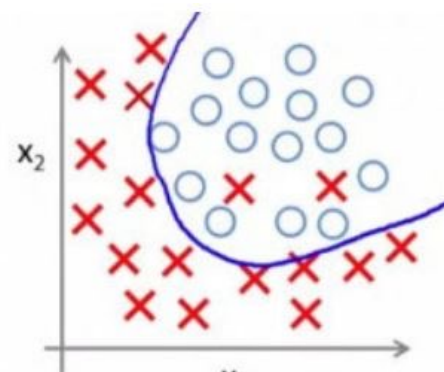
**Which one?**



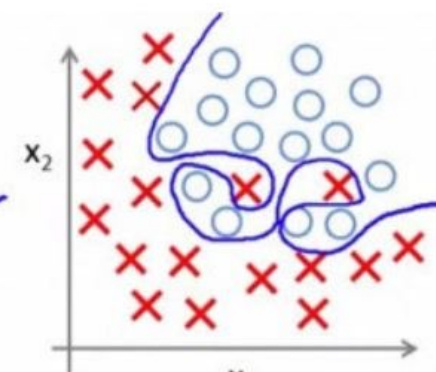
Input



Model 1



Model 2



Model 3



# 目录

1 概述

2 数据集划分方法

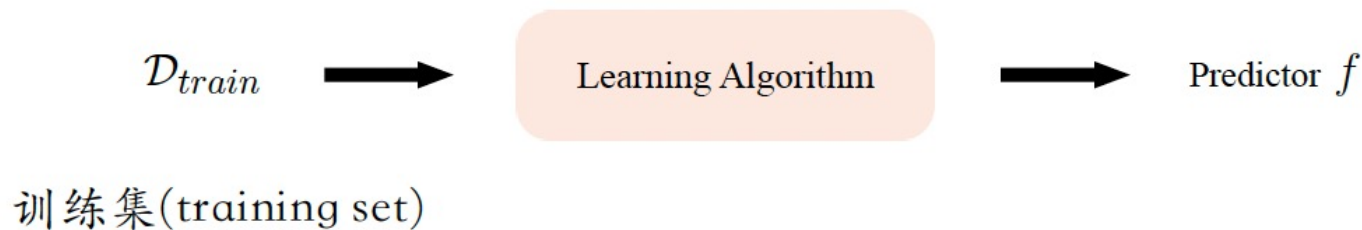
3 模型评估方法

4 模型选择方法

5 大模型时代模型选择



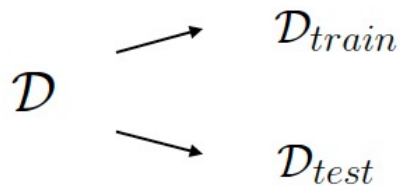
## 数据集划分



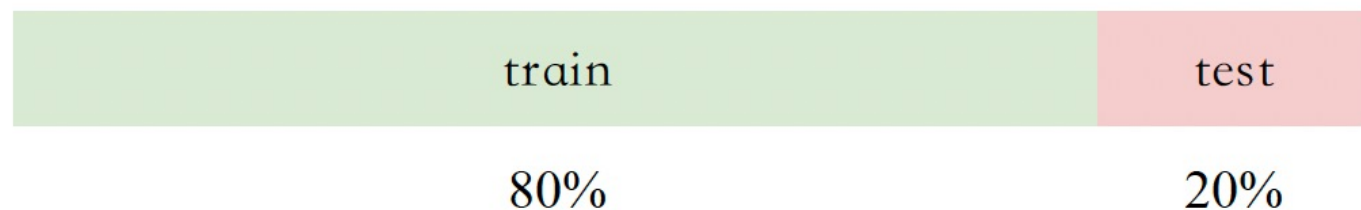
如何评估predictor的好坏?

构造测试集(testing set), 用测试集中的样本(test sample)来检验predictor的好坏。测试集应与训练集互斥, 测试样本不会用于训练过程。

通常从给定数据集(dataset)中划分出训练集和测试集。



# 留出法 (Hold-out)



将数据集若干次随机划分为无交集的训练集和测试集，重复实验评估后取平均值作为结果。

通常训练集占总样本的80%，测试集占总样本的20%。



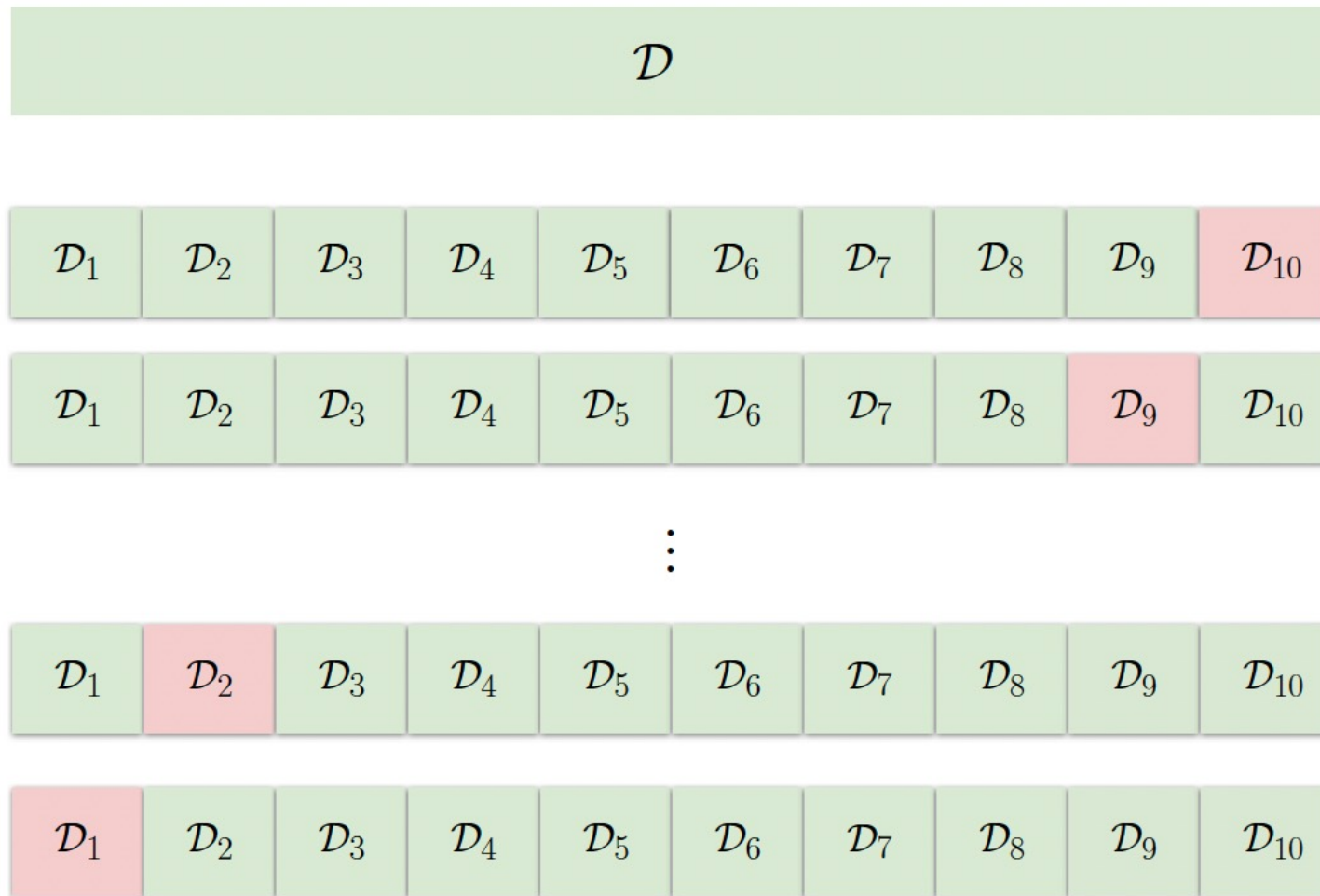
Hold-out 受到数据划分方式的影响，模型容易偏向于训练集中出现次数多的样本。



# 交叉验证法 (k-fold cross validation)



将数据集随机等分为k-fold, 每次取一个fold作为测试集, 输出k个结果的平均值。



Average Test Result

## ◆ 与留出法的区别:

交叉验证法使用了所有数据进行训练和测试, 受数据分布不平衡影响小; 增加了计算成本。



k-fold cross validation的特殊形式 ( $k$ 等于样本总数) , 即每次取出一个样本作为测试样本, 剩余样本作为训练集。



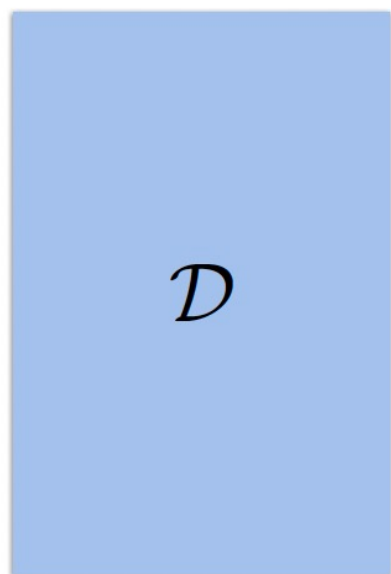
LOO不受数据划分方式的影响, 与原始数据集分布近似, 其评估结果通常认为比较准确, 但当数据集较大时, 计算开销也随之增大。



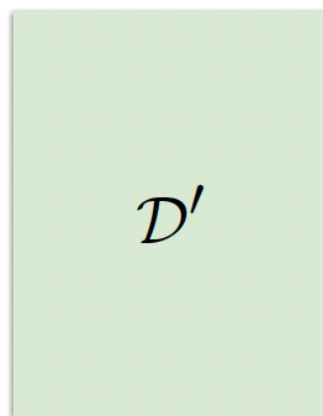
# 自助法 (Bootstrapping)



有放回地重复抽样 $m$ 次



$m$  samples



原数据集 $D$ 中部分样本重复在 $D'$ 中出现，部分样本从未在 $D'$ 中出现。

原数据集 $D$ 中不被 $D'$ 包含的数据作为测试集

样本不被取到的概率为  $\left(1 - \frac{1}{m}\right)^m$

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



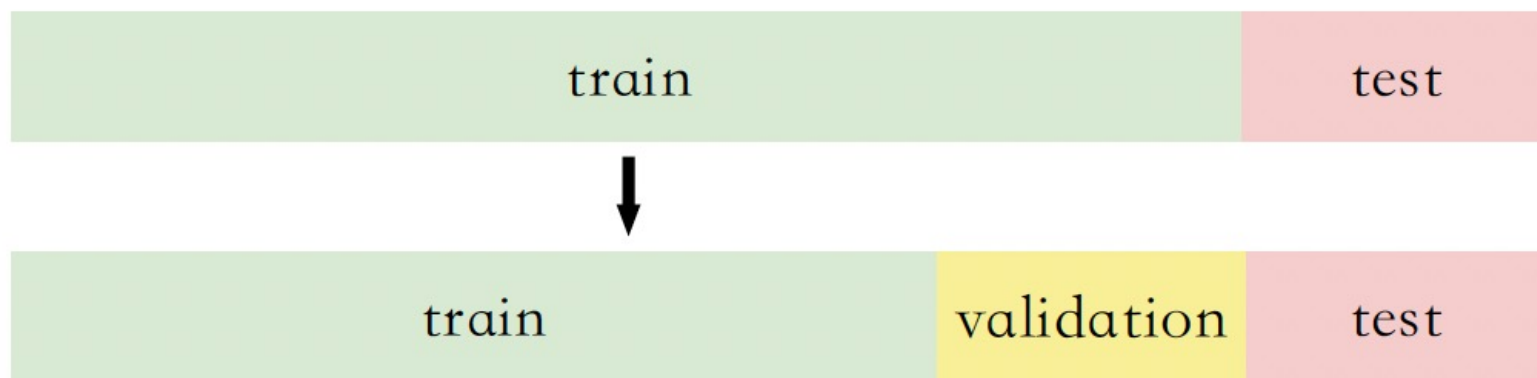
自助法在数据集较小难以划分时比较有效，但改变了原有数据集的分布，会引入估计误差。



# 验证集 (validation set)



模型评估与选择中，常常从训练集中拆分一部分数据作为验证集(validation set)。



验证集上进行调参和模型的选择，而测试集是实际应用中的数据，用于最终的性能度量。



# 目录

1 概述

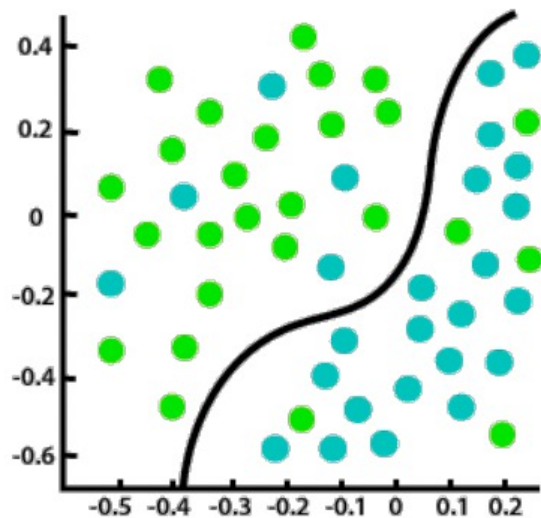
2 数据集划分方法

**3 模型评估方法**

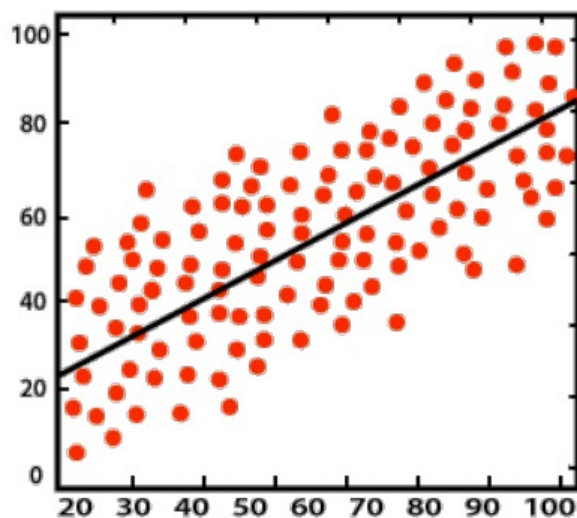
4 模型选择方法

5 大模型时代模型选择

为了衡量一个机器学习模型的好坏，需要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，不同学习任务模型性能的评价标准不同。



Classification



Regression

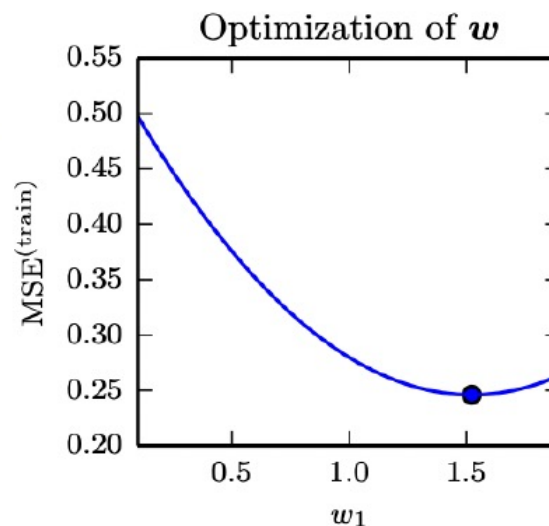
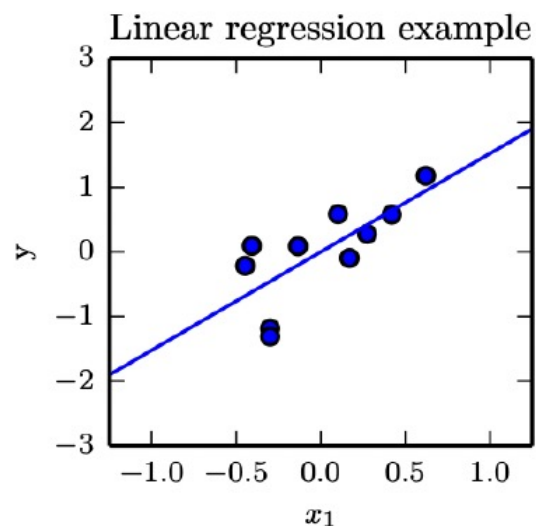
比较学习器预测结果 $f(x)$ 与真实标记 $y$

均方误差(mean squared error, MSE)

$$E(f; D) = \frac{1}{N} \sum_{n=1}^N (f(x_n) - y_n)^2$$

$$E(f; D) = \int_{x \sim D} (f(x) - y)^2 p(x) dx$$

其中 $D$ 是数据分布， $p(x)$ 是概率密度函数。





错误率(error rate)

$$E(f; D) = \frac{1}{m} \sum_{i=1}^M \mathbb{I}(f(x_i) \neq y_i)$$

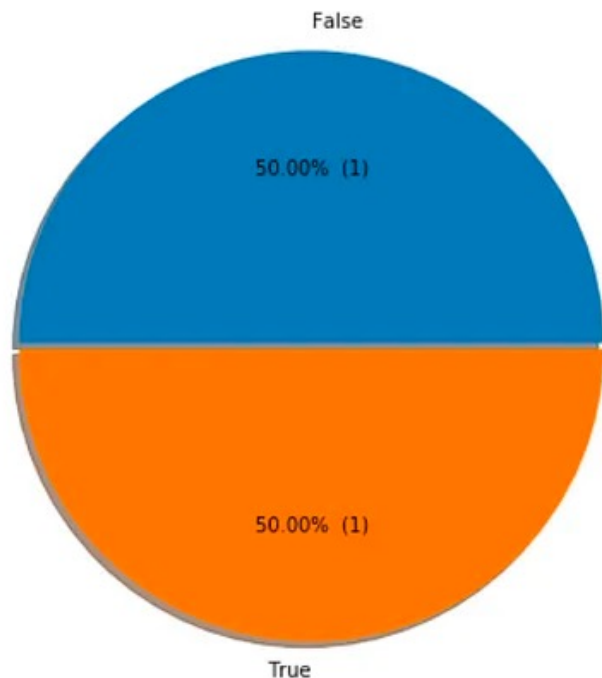
准确率(accuracy)

$$\text{acc}(f; D) = \frac{1}{m} \sum_{i=1}^M \mathbb{I}(f(x_i) = y_i) = 1 - E(f; D)$$

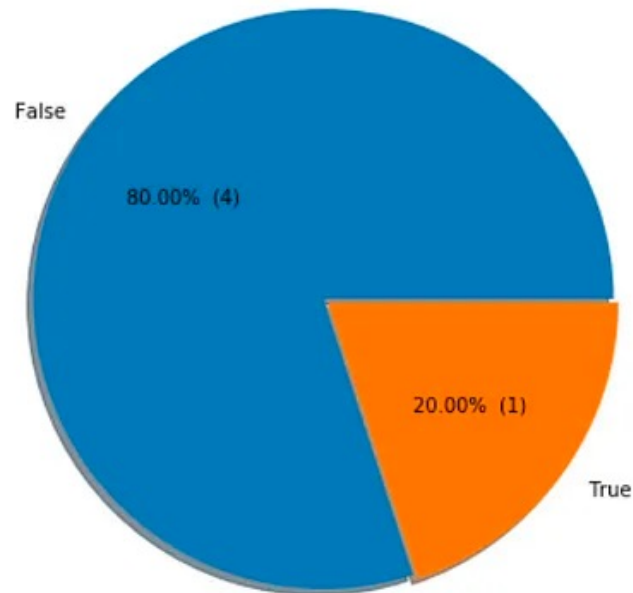


## Is your Target Imbalanced? Do you still use Accuracy? Don't!

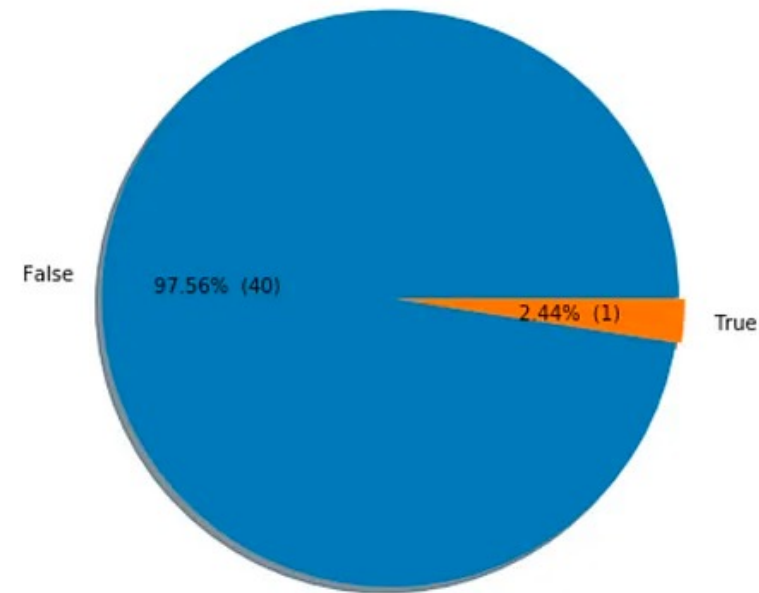
Titanic(Generic Problem)--Balanced Dataset



Actual Business Data--Imbalanced



Fraud Detection/Disease Detection--Highly Imbalanced



对于极度不均衡的数据集来说，Accuracy就失去了意义



银行的交易记录中仅有**0.1%**的交易记录是诈骗行为，99.9%是正常交易。为银行开发的信用卡诈骗检测模型进行分类时，可以达到99.9%的判别准确率。然而仅仅考虑模型的判别准确率并不能处理潜在的诈骗行为。

希望考虑的是：

模型判别为诈骗的交易记录中有多少条是真实的诈骗？

模型判别为正常的交易记录中有多少条反而为诈骗行为？ ...



# 评估指标——混淆矩阵(confusion matrix)



		Actual class	
		Positive	Negative
Predicted class	Positive	True positive (TP) N=800 0.08%	False positive (FP) N=700 0.07%
	Negative	False negative (FN) N=200 0.02%	True negative (TN) N=1,000,000 99.9%



# 评估指标——混淆矩阵(confusion matrix)

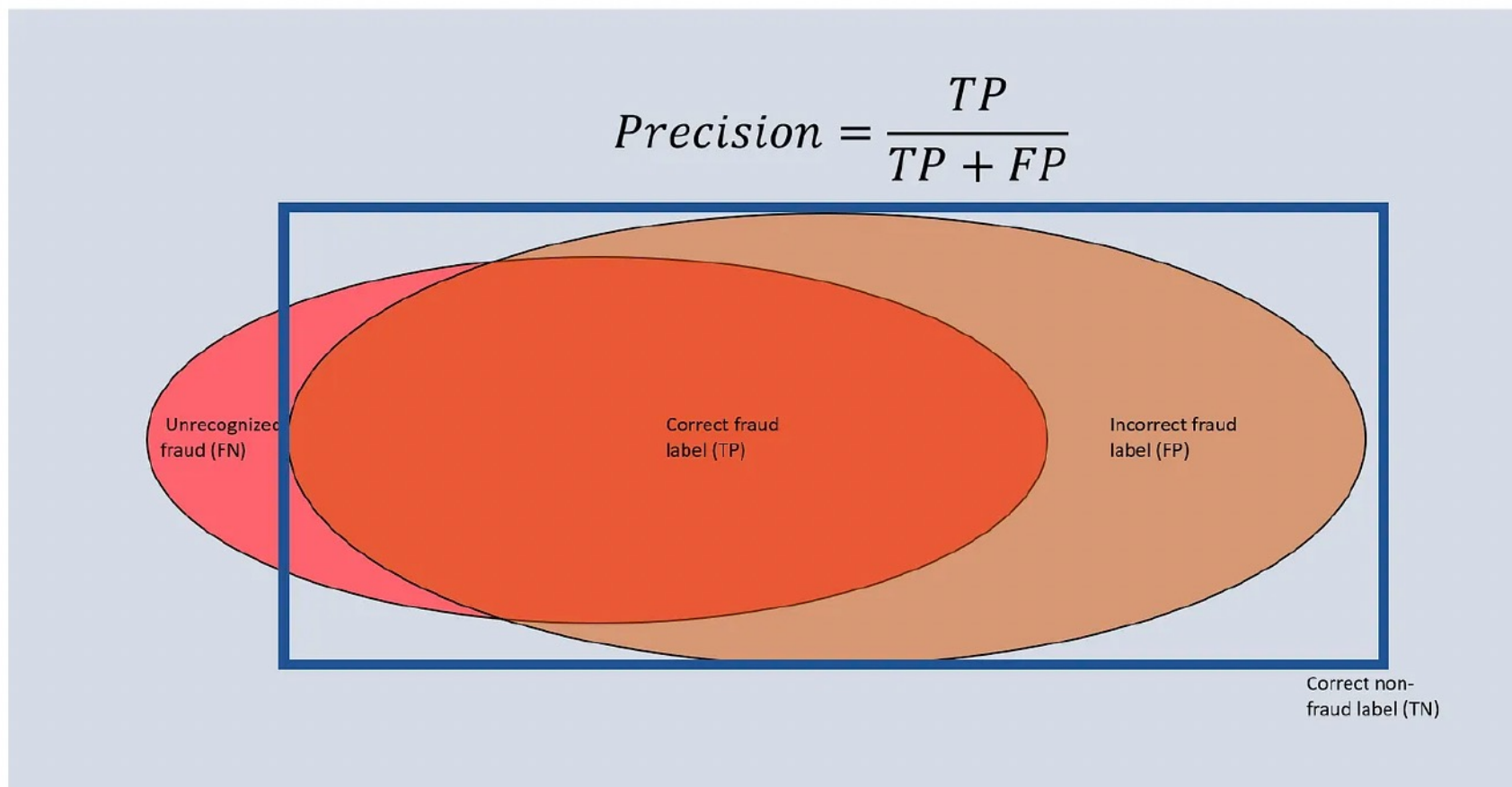


Sources: [12][13][14][15][16][17][18][19][20] view · talk · edit

		Predicted condition			
		Predicted Positive (PP)	Predicted Negative (PN)		
Actual condition	Total population = P + N			Informedness, bookmaker informedness (BM) = TPR + TNR - 1	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
	Positive (P) <sup>[a]</sup>	True positive (TP), hit <sup>[b]</sup>	False negative (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error <sup>[c]</sup> $= \frac{FN}{P} = 1 - TPR$
	Negative (N) <sup>[d]</sup>	False positive (FP), false alarm, overestimation	True negative (TN), correct rejection <sup>[e]</sup>	False positive rate (FPR), probability of false alarm, fall-out type I error <sup>[f]</sup> $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR-) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP ( $\Delta p$ ) = PPV + NPV - 1	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F <sub>1</sub> score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2 TP}{2 TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \frac{\sqrt{TPR \times TNR \times PPV \times NPV}}{\sqrt{FNR \times FPR \times FOR \times FDR}}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$



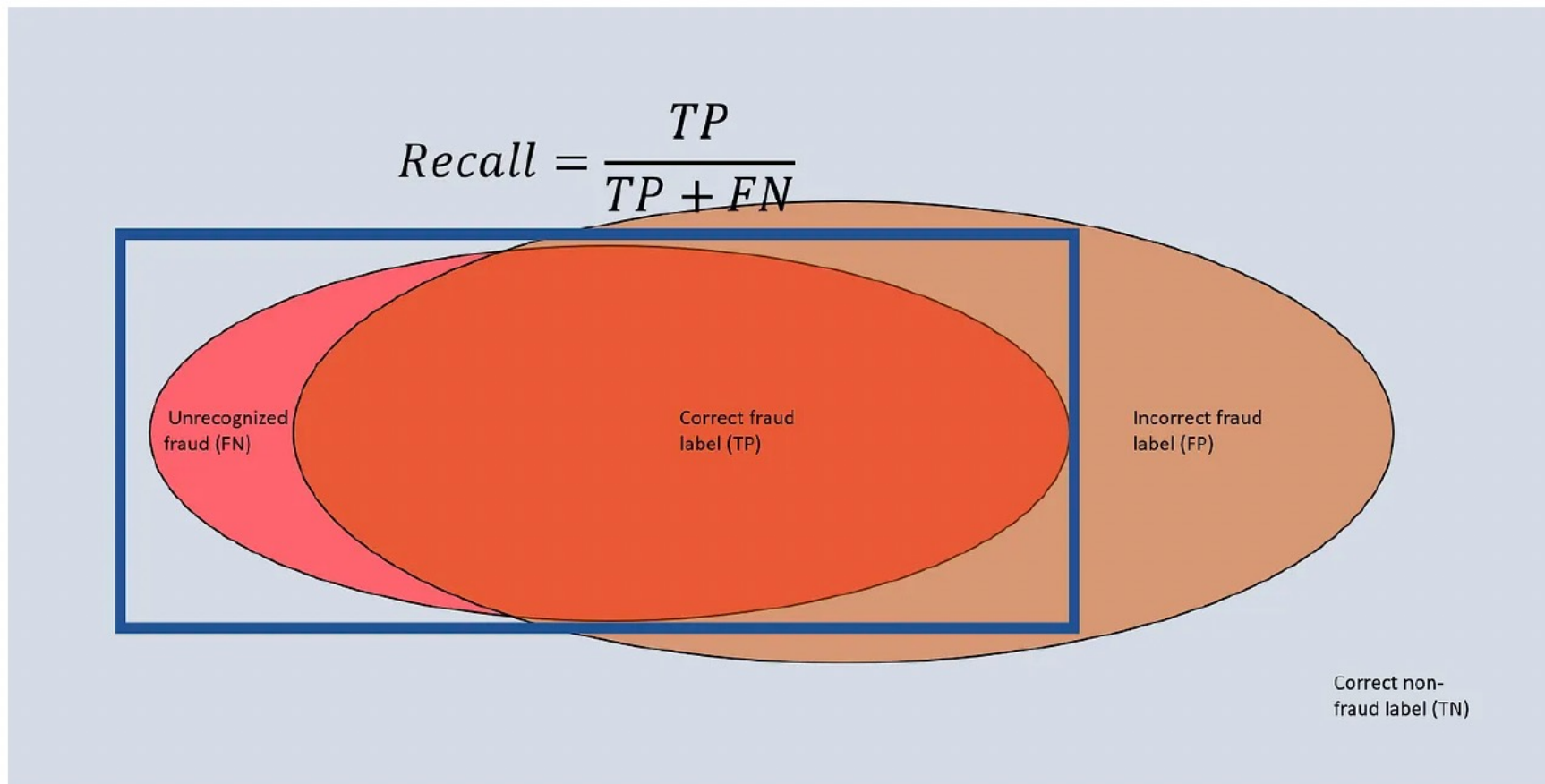
# 评估指标——精确率 (Precision)



精确率(precision)说明了模型所有判别为诈骗的交易记录中，有多少比例的记录确实为诈骗行为。



# 评估指标——召回率 (Recall)



召回率(recall)说明了所有诈骗记录中，被模型侦测出的诈骗记录的占比。



# 混淆矩阵



		Actual class	
		Positive	Negative
Predicted class	Positive	True positive (TP) N=800 0.08%	False positive (FP) N=700 0.07%
	Negative	False negative (FN) N=200 0.02%	True negative (TN) N=1,000,000 99.9%

$precision =$

$recall =$

$acc =$



# 混淆矩阵



		Actual class	
		Positive	Negative
Predicted class	Positive	True positive (TP) N=800 0.08%	False positive (FP) N=700 0.07%
	Negative	False negative (FN) N=200 0.02%	True negative (TN) N=1,000,000 99.9%

$$precision = \frac{TP}{TP + FP} = \frac{800}{800 + 700} = 0.53$$

$$recall =$$

$$acc =$$



# 混淆矩阵



		Actual class	
		Positive	Negative
Predicted class	Positive	True positive (TP) N=800 0.08%	False positive (FP) N=700 0.07%
	Negative	False negative (FN) N=200 0.02%	True negative (TN) N=1,000,000 99.9%

$$precision = \frac{TP}{TP + FP} = \frac{800}{800 + 700} = 0.53$$

$$recall = \frac{TP}{TP + FN} = \frac{800}{800 + 200} = 0.8$$

$$acc =$$





		Actual class	
		Positive	Negative
Predicted class	Positive	True positive (TP) N=800 0.08%	False positive (FP) N=700 0.07%
	Negative	False negative (FN) N=200 0.02%	True negative (TN) N=1,000,000 99.9%

$$precision = \frac{TP}{TP + FP} = \frac{800}{800 + 700} = 0.53$$

$$recall = \frac{TP}{TP + FN} = \frac{800}{800 + 200} = 0.8$$

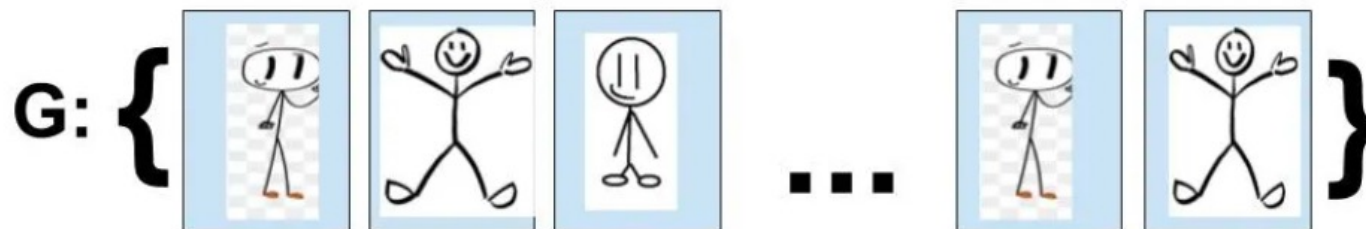
$$acc = \frac{TP + TN}{TP + TN + FP + FN} = \frac{800 + 1000000}{1001700} = 0.999$$



# Precision-Recall



在检索系统中提出



假设有5个是匹配的



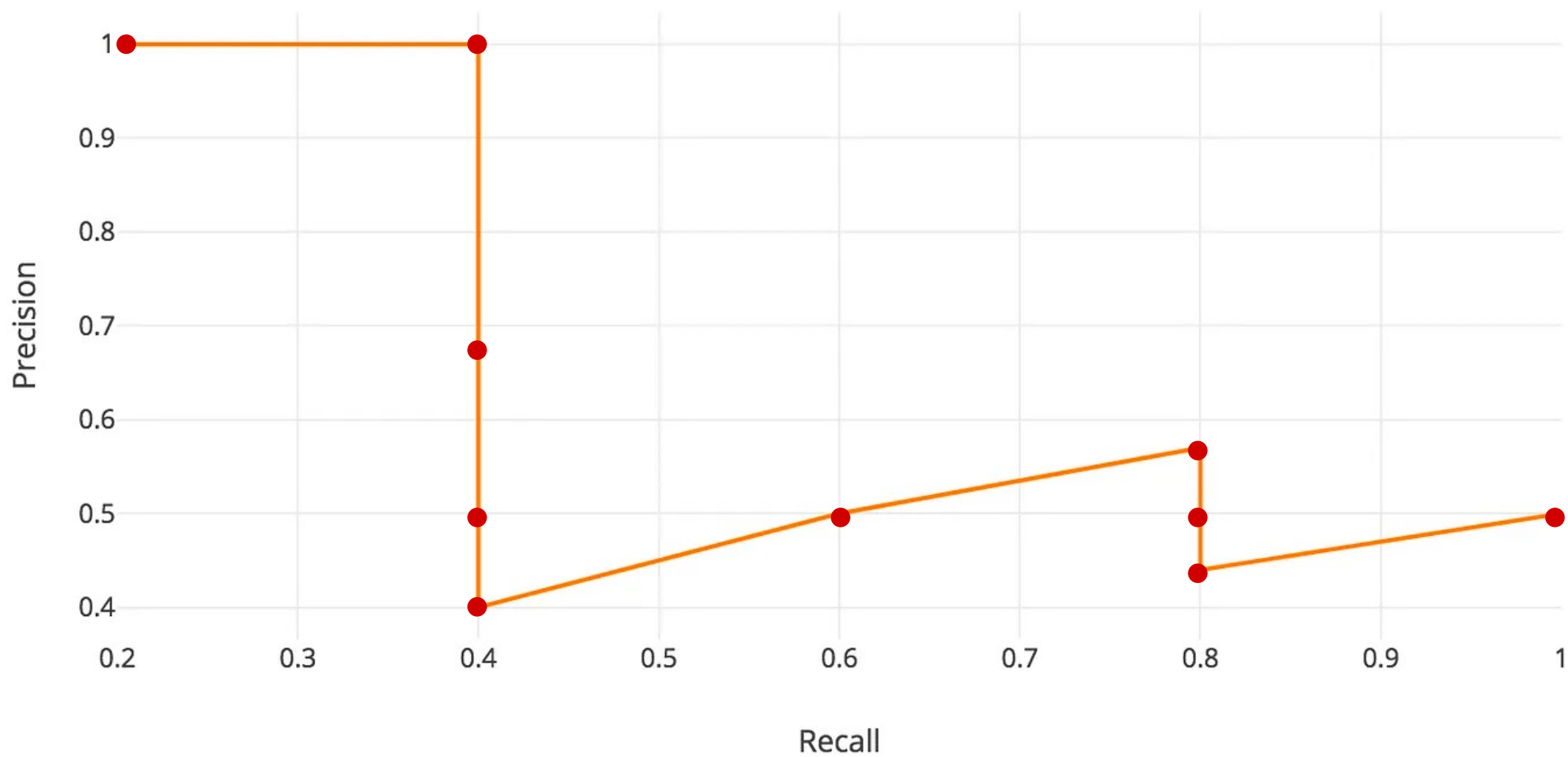
# Precision-Recall



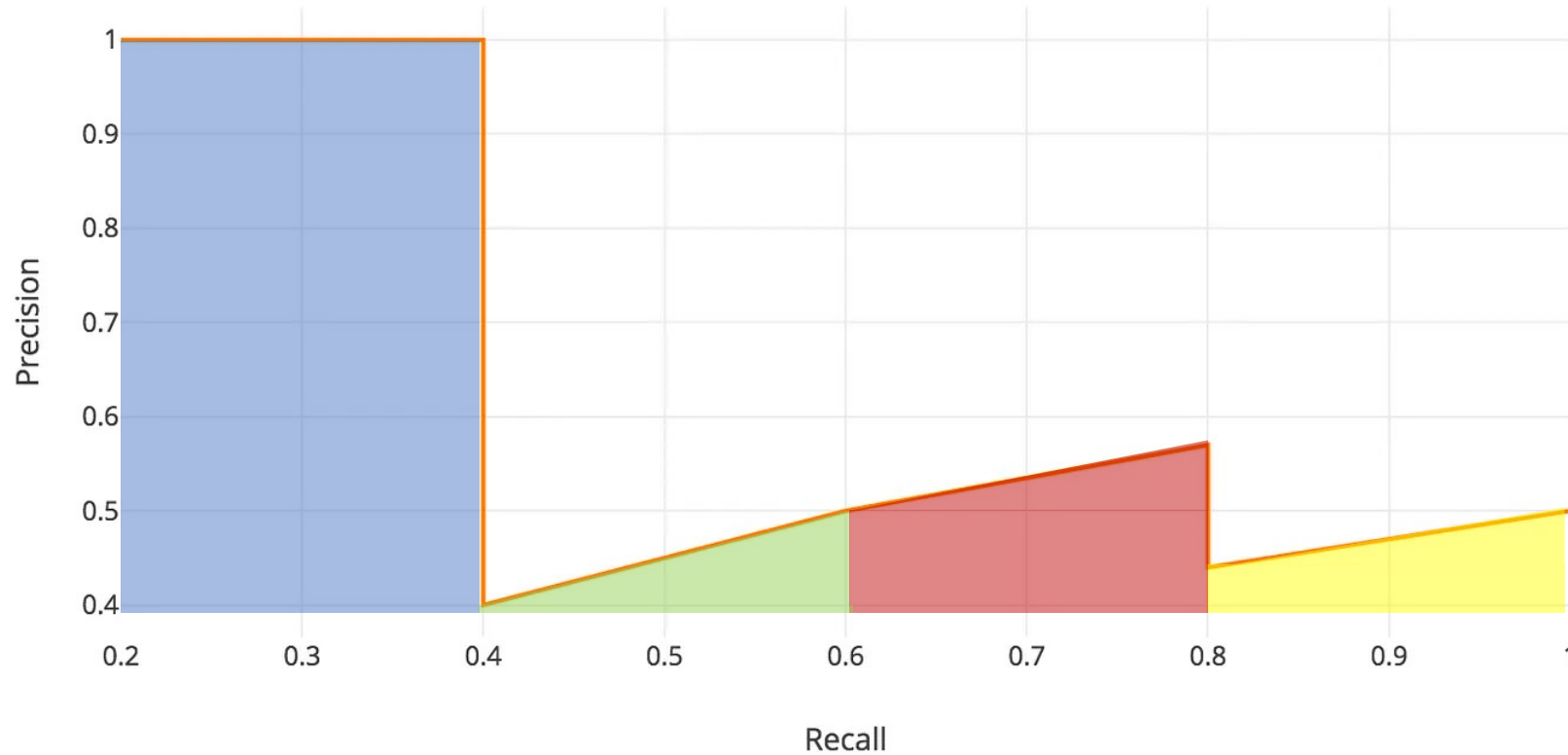
排序	是否分对	精度	召回
1	是	1	0.2
2	是	1	0.4
3	否	0.67	0.4
4	否	0.5	0.4
5	否	0.4	0.4
6	是	0.5	0.6
7	是	0.57	0.8
8	否	0.5	0.8
9	否	0.44	0.8
10	是	0.5	1.0



# P-R(Precision-Recall) 曲线



# AP (Average Precision) : P-R曲线下的面积



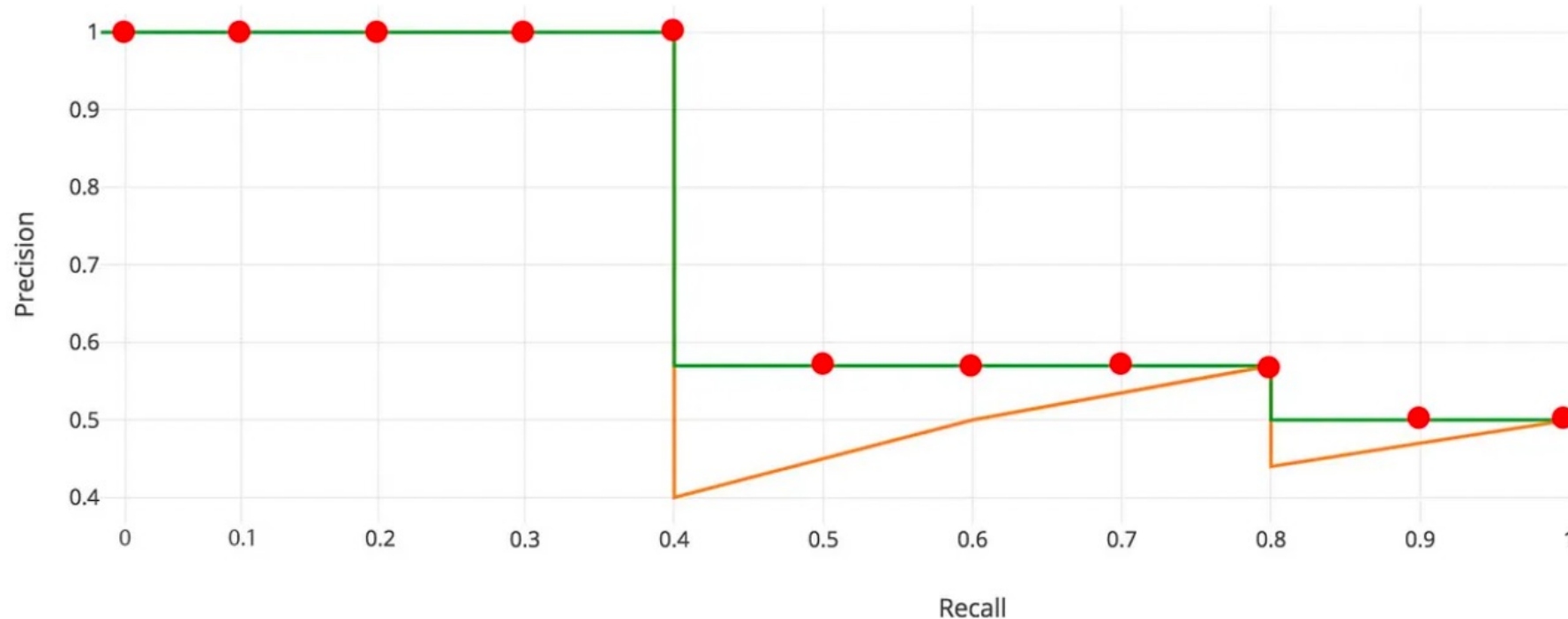
计算方法一：积分算面积

$$AP = (0.4 - 0) \times 1 + (0.4 + 0.5) \times 0.2 \div 2 + (0.5 + 0.57) \times 0.2 \div 2 + (0.44 + 0.5) \times 0.2 \div 2 = 0.691$$

计算较为繁琐



# AP (Average Precision) : P-R曲线下的面积

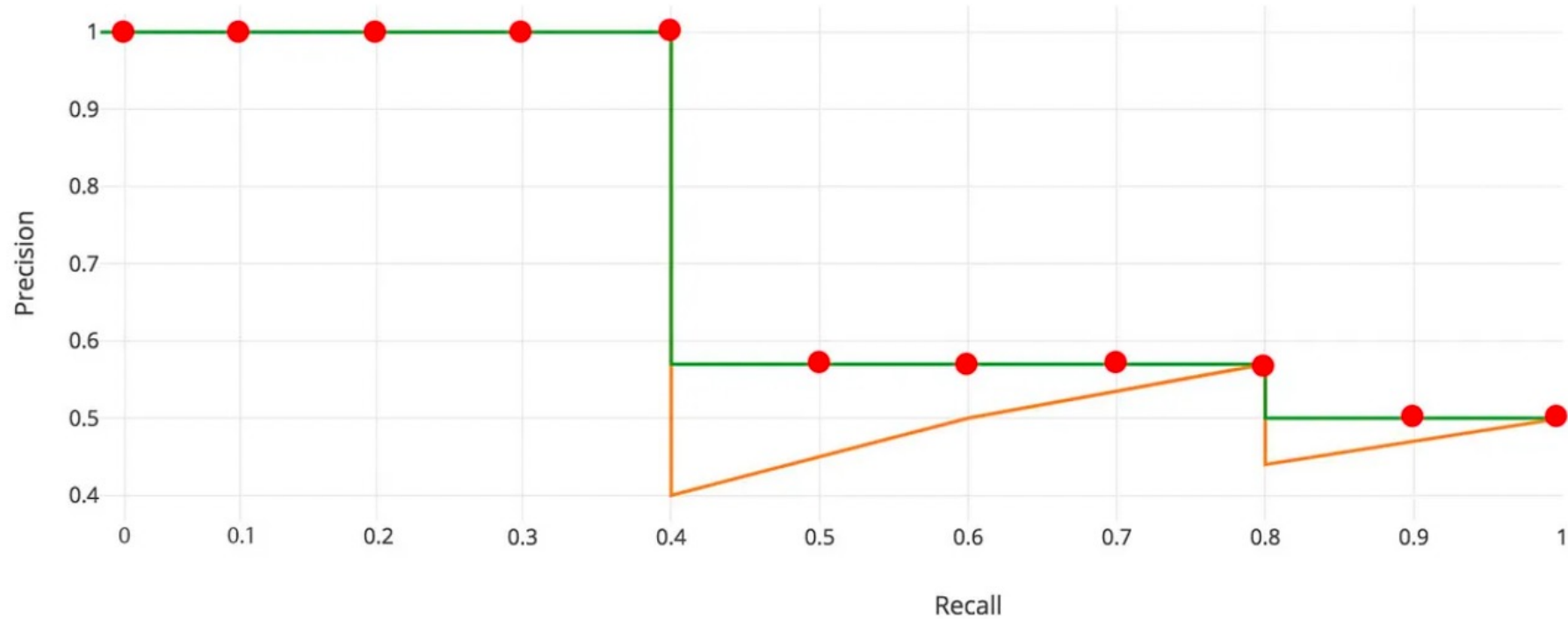


## 计算方法二：近似算法

- 将Recall所处区间 $[0,1]$ 等分为11份，即Recall=0, 0.1, 0.2, ..., 0.9, 1
- 分别以 $r_i$ 作为下限，统计大于等于 $r_i$ 的Recall最大值作为此时 $p_i$ 的取值
- 最后 $AP = \frac{1}{11} \sum_{i=1}^{11} p_i$



# AP (Average Precision) : P-R曲线下的面积



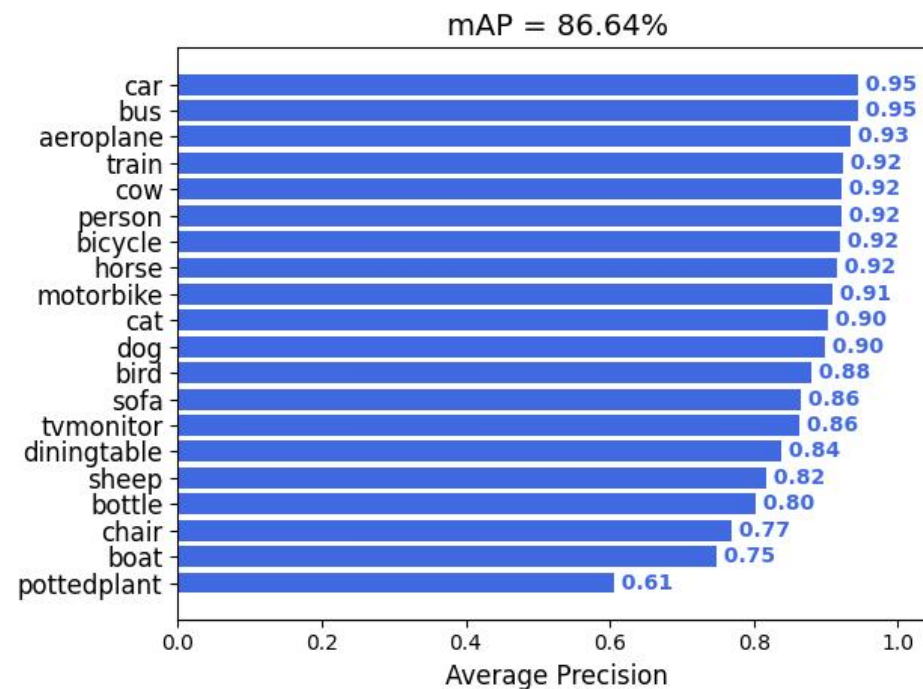
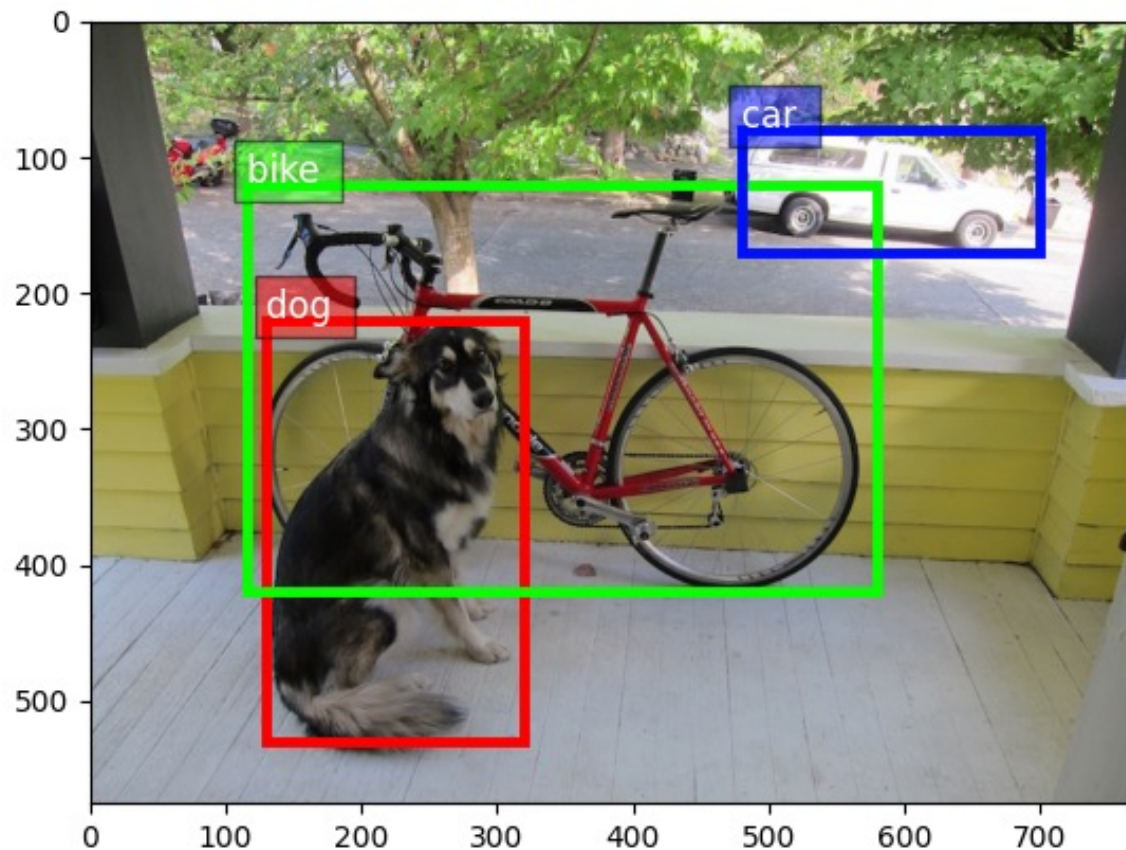
计算方法二：近似算法

$$AP_{11} = 1/11 (Ap(0) + Ap(0.1) + \dots + Ap(1.0))$$

$$AP = (5 \times 1.0 + 4 \times 0.57 + 2 \times 0.5) / 11$$



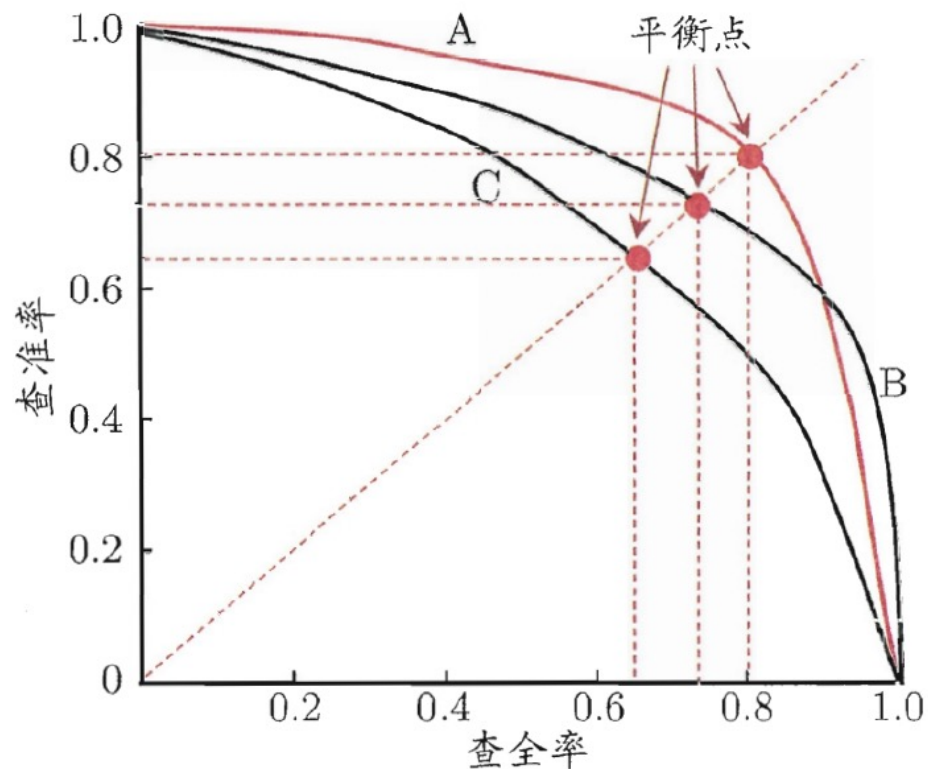
# AP应用举例：目标检测



目标检测评价指标：mAP :所有类别的平均AP



# 评估指标——平衡点 & F1-Score



trade-off

Precision  $\longleftrightarrow$  Recall

平衡点(Break even point, BEP):

P=R时, PR图的取值

F1分数(F1-score):

Precision和Recall的调和平均数

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

一般而言, 评估模型性能:

- PR曲线包裹的面积越大, 性能越好
- 平衡点越靠外, 性能越好



# 评估指标—— ROC/AUC



**ROC:** FPR与TPR联合曲线

**AUC:** ROC曲线下的面积

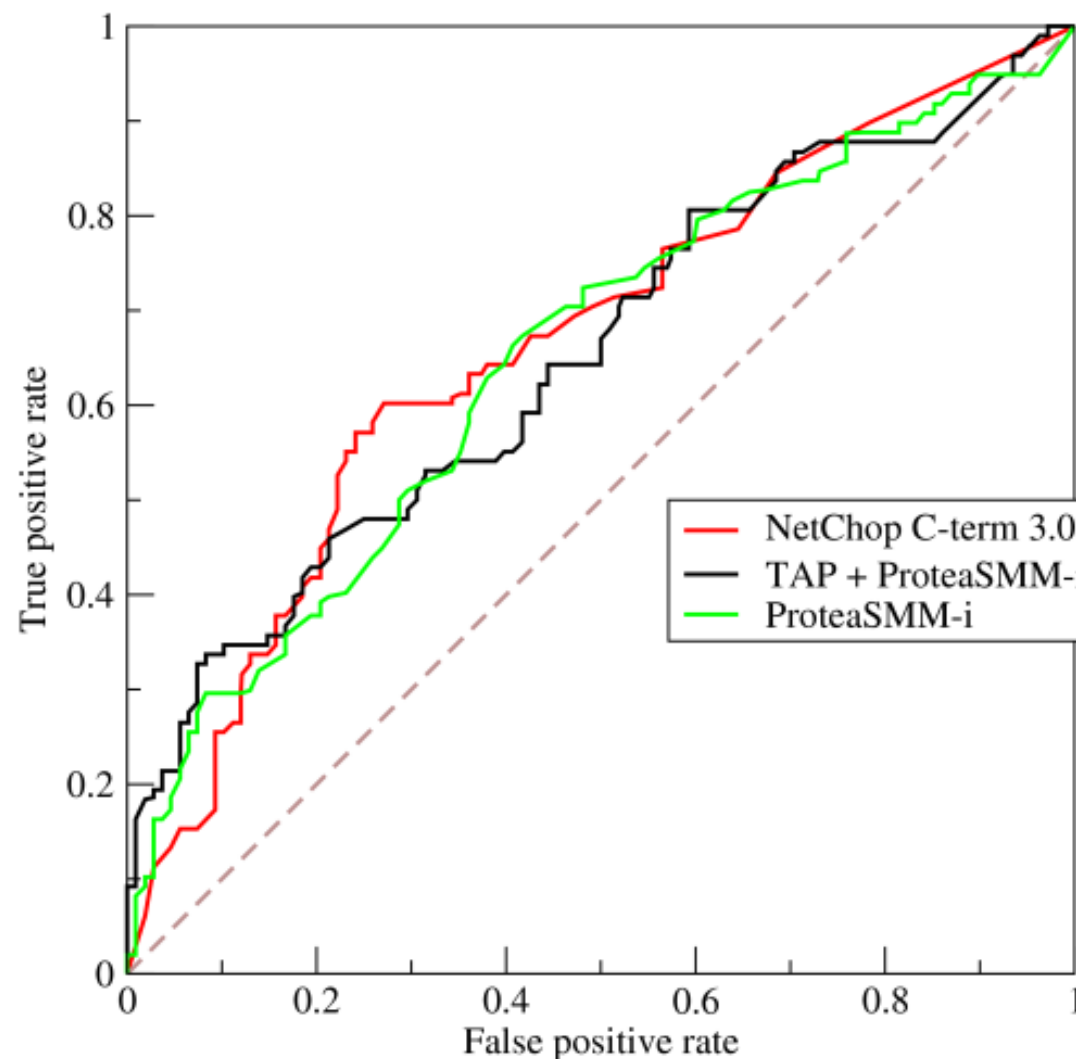
所有正样本中预测正确的概率

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

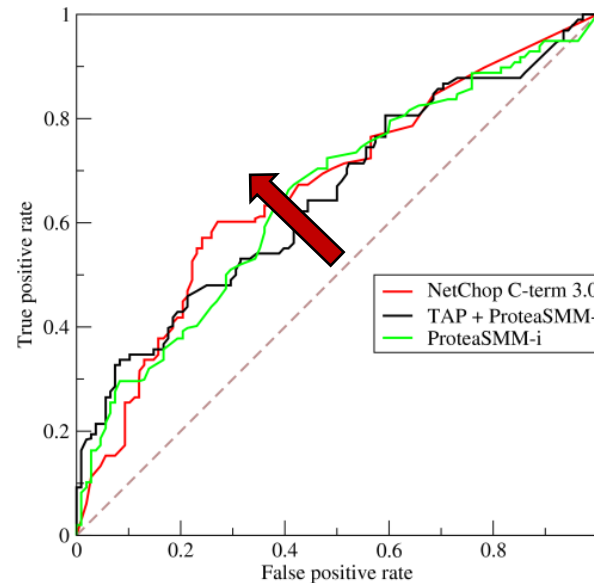
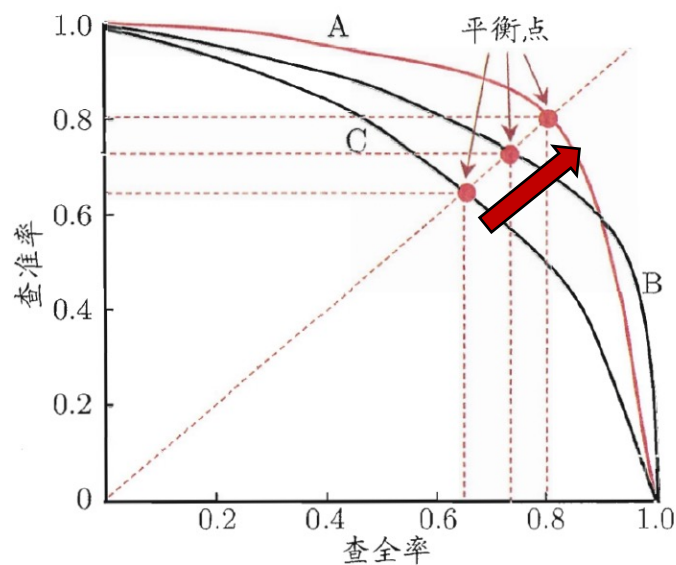
所有负样本中预测错误的概率

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

一般而言，ROC曲线包裹的面积越大，越接近(0, 1)点，性能越好。



# PR曲线 v.s. ROC曲线



相同点：二者都用于二分类评估任务，都可以对模型性能进行衡量

不同点：(1) 侧重点不同：

PR曲线：模型在不同阈值下找出正例的能力；

ROC曲线：模型在不同阈值下正确分类正例和负例的能力。

(2) 适用场景不同：

PR曲线对类别分布敏感，适合测试比较不同分布下模型的性能差异；

ROC曲线对类别间分布平衡性更具优势，适合测试模型的综合性能。



# 实际中该用哪个指标?



- **准确率 (Accuracy)** : 对于类别分布均匀的分类问题, 比如CIFAR-10分类
- **Precision (精度优先)** : 垃圾邮件过滤, 漏掉一些没关系, 但是找出来的得是真的垃圾邮件
- **Recall (查全率优先)** : 检索类型的任务或者小概率事件预测类的, 希望对的都能找出来, 无“漏网之鱼”
- **F1分数 (F1-score)** : 用于类别分布不均匀的情况, 兼顾precision和recall
- **AUC (The area under the curve)** : 大部分检测任务场景, 医疗诊断任务



5. 人工智能在医疗诊断中的应用评估。

- (1) (4分) 当我们开发一个 AI 模型来辅助医生进行疾病诊断（例如，通过分析医学影像判断是否为某种疾病）后，为什么要使用交叉验证（Cross-validation）来评估这个 AI 模型的性能？其首要目的是什么？
- (2) (4分) 简述 k-折交叉验证在评估这个医疗诊断 AI 模型时，是如何划分数据并进行评估的。
- (3) (4分) 相比于只进行一次训练集和测试集的划分，交叉验证为什么能为这个医疗 AI 模型提供一个更可靠的性能评估？

## 5. 人工智能在医疗诊断中的应用评估。

- (1) (4分) 当我们开发一个 AI 模型来辅助医生进行疾病诊断（例如，通过分析医学影像判断是否为某种疾病）后，为什么要使用交叉验证（Cross-validation）来评估这个 AI 模型的性能？其首要目的是什么？
- (2) (4分) 简述 k-折交叉验证在评估这个医疗诊断 AI 模型时，是如何划分数据并进行评估的。
- (3) (4分) 相比于只进行一次训练集和测试集的划分，交叉验证为什么能为这个医疗 AI 模型提供一个更可靠的性能评估？

### 答案：

- (1) 交叉验证的首要目的是更准确、更稳健地估计这个医疗诊断 AI 模型在未见过的真实病人数据上的泛化能力（即诊断准确性），从而减少因单次数据划分带来的评估结果的偶然性。
- (2) k-折交叉验证会将收集到的病人数据（已标记诊断结果）随机分成 k 份。在每一轮评估中，选择其中一份作为测试数据来检验 AI 模型的诊断效果，剩下的 k-1 份数据用于训练 AI 模型。这个过程会重复 k 次，每次都不同的一份数据做测试。
- (3) 因为交叉验证让 AI 模型在 k 个不同的“模拟真实场景”（不同的测试集）上进行了测试，并将这 k 次测试的性能结果（如诊断准确率）取平均。这比单次测试更能代表模型在各种可能遇到的病人数据上的综合表现，评估结果更稳定，不易受到某一次特定数据划分好坏的影响。

# 目录

1 概述

2 数据集划分方法

3 模型评估方法

4 **模型选择方法**

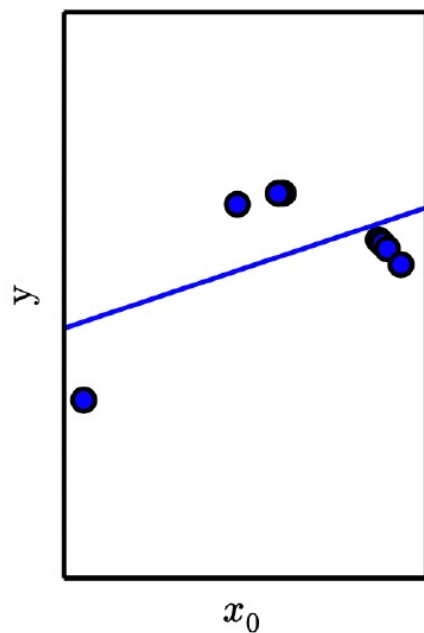
5 大模型时代模型选择



奥卡姆剃刀 “如无必要，勿增实体”

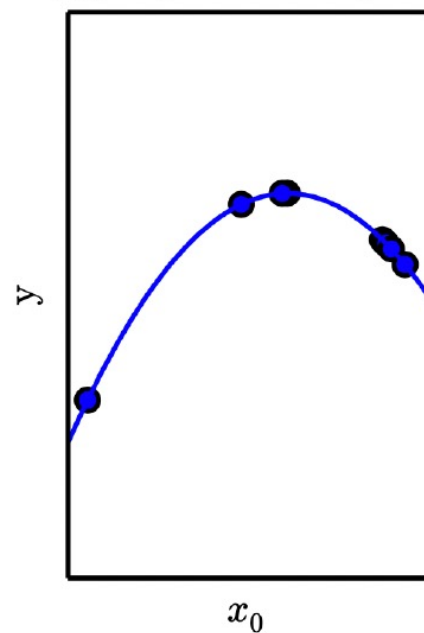
在同样能够解释已知观测现象的假设中，我们应该挑选“最简单”的那一个

Underfitting



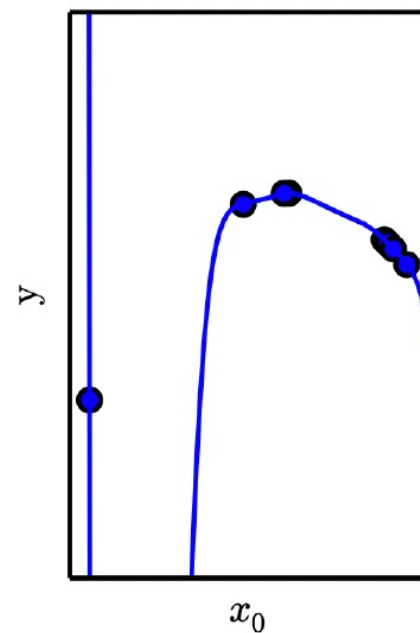
$$y = a_0 + a_1x$$

Appropriate capacity



$$y = a_0 + a_1x + a_2x^2$$

Overfitting



$$y = \sum_i^9 a_i x^i$$





经验误差（经验误差）：训练集上的误差

$$\frac{1}{m^{(train)}} \|X^{(train)} w - y^{(train)}\|$$

泛化误差（泛化误差）：测试集上的误差

$$\frac{1}{m^{(test)}} \|X^{(test)} w - y^{(train)}\|$$

泛化：在先前未观测到的输入上表现良好的能力

泛化差距（Generalization Gap）：泛化误差 - 经验误差





## 泛化能力 (generalizability)

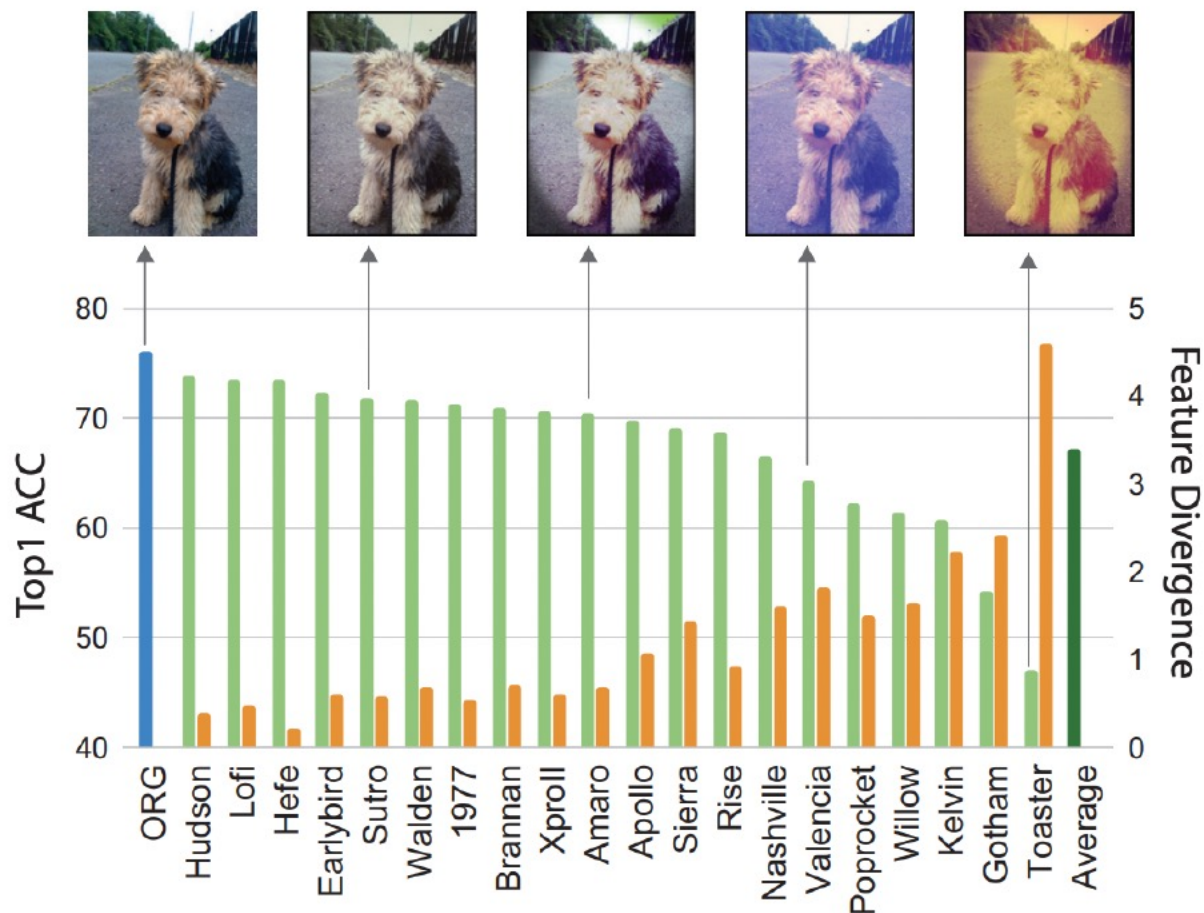
- 指由该方法学习到的模型对未知数据的预测能力
- 是学习方法本质上最重要的性质

## 泛化误差与泛化能力

- 泛化误差就是泛化能力的一种度量
- 泛化误差越小，则泛化能力越强，学习方法越有效
- 泛化误差是所学习到的模型的期望风险



## 泛化能力 (generalizability)

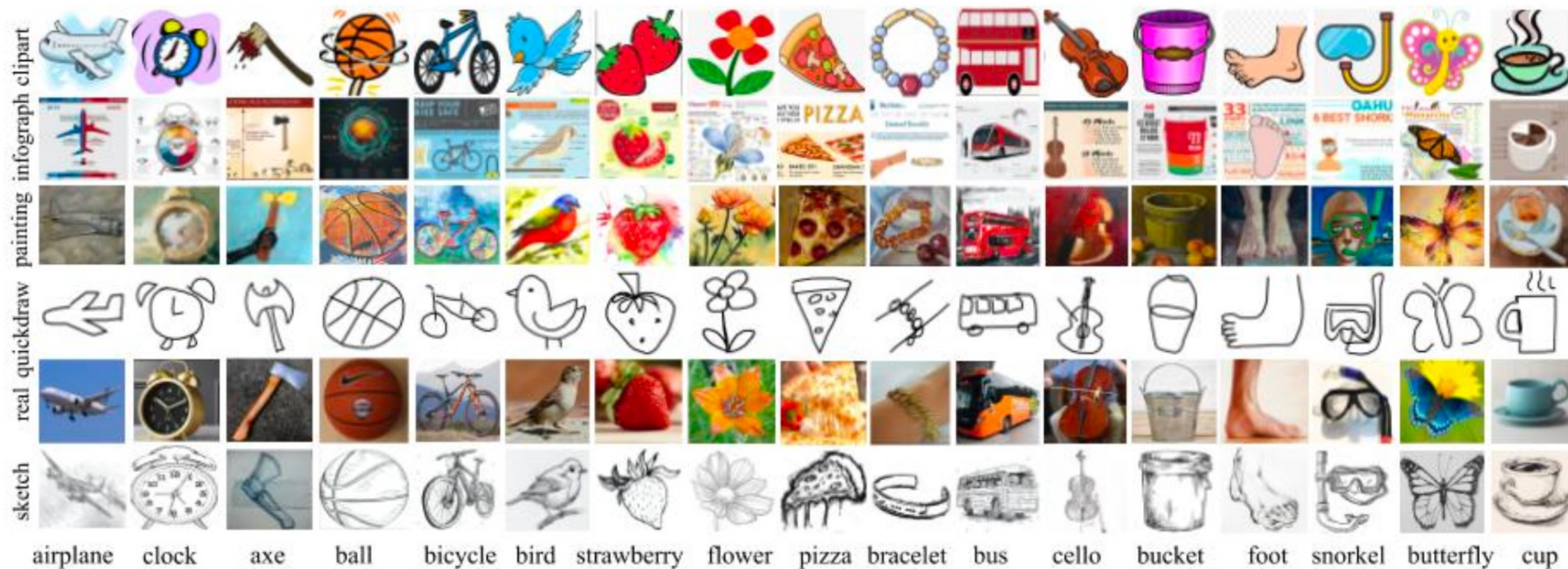


为图片添加不同instagram滤镜后图像识别分类准确率

# 泛化能力



泛化能力 (generalizability)



以二分类任务为例：

训练集  $\{(x_1, y_1), \dots, (x_n, y_n)\}, \quad x \in R^D, \quad y \in \{+1, -1\}$

模型假设空间  $\mathbf{f} = \{f_1, \dots, f_d\}$

训练误差  $\hat{R}(f) = \frac{1}{N} \sum_{n=1}^N I(y_n \neq f(x_n))$

测试误差的期望  $R(f) = E(x, y)[I(y \neq f(x_n))]$

在整个样本分布上的期望

# 泛化误差上界



以二分类任务为例：

训练集  $\{(x_1, y_1), \dots, (x_n, y_n)\}$ ,  $x \in R^D$ ,  $y \in \{+1, -1\}$  练习题册

模型假设空间  $\mathbf{f} = \{f_1, \dots, f_d\}$  学生所有可能的学习策略和知识储备

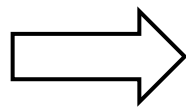
训练误差  $\hat{R}(f) = \frac{1}{N} \sum_{n=1}^N I(y_n \neq f(x_n))$  学生在做完这本练习题册后，在练习题册上的错误率

测试误差的期望  $R(f) = E(x, y)[I(y \neq f(x_n))]$  我们最关心的：学生在真正高考时的错误率

在整个样本分布上的期望



**核心矛盾：**我们能精确计算的只有“练习册上的错误率”，但我们真正关心的却是未知的“高考错误率”。我们如何从已知的训练误差，去推断未知的泛化误差呢？”



## Vapnik-Chervonenkis 维度

对任意一个函数，至少以概率  $1 - \delta$ ，以下不等式成立

$$R(f) \leq \hat{R}(f) + \sqrt{\frac{1}{2N} \left( \log d + \log \frac{1}{\delta} \right)}$$

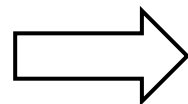
训练集上的风险  $\hat{R}(f)$       测试集上的风险  $R(f)$       误差上界

$d$ : VC维

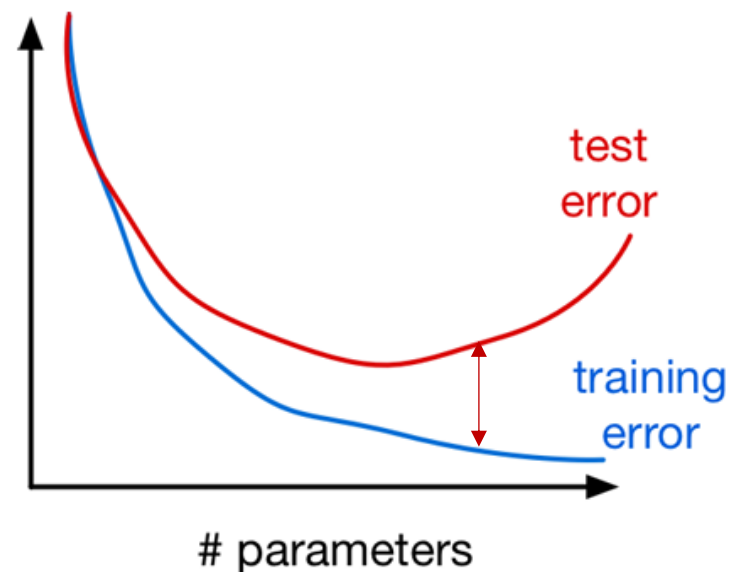
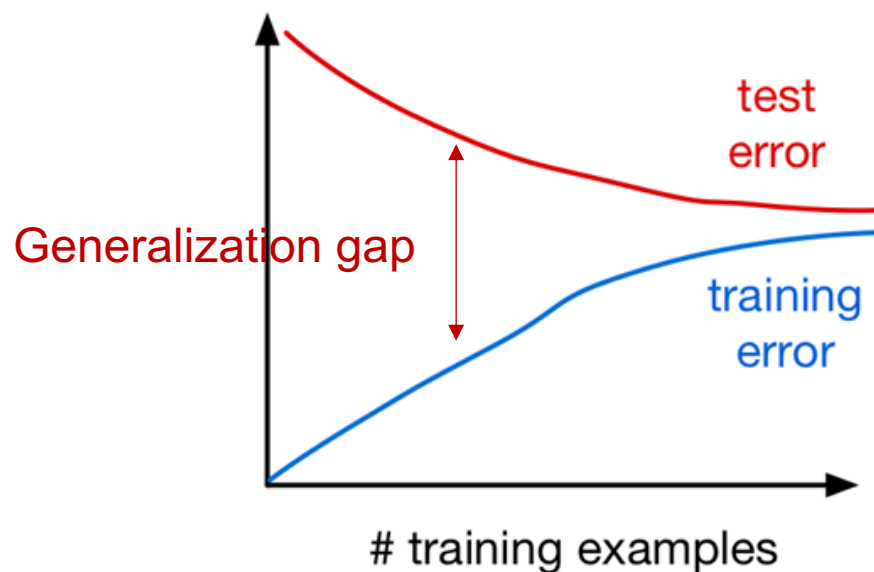
$\epsilon$

泛化误差  $\leq$  训练误差 + 模型复杂度惩罚

- 训练误差越小，泛化误差上界越小
- 样本数量  $N$  越大，泛化误差上界越小
- 假设空间越大（模型越复杂），泛化误差上界越大



要想泛化能力强，要么你本身在练习册上就表现优异，要么你的学习方法足够简单、通用，而且你见过的题足够多！



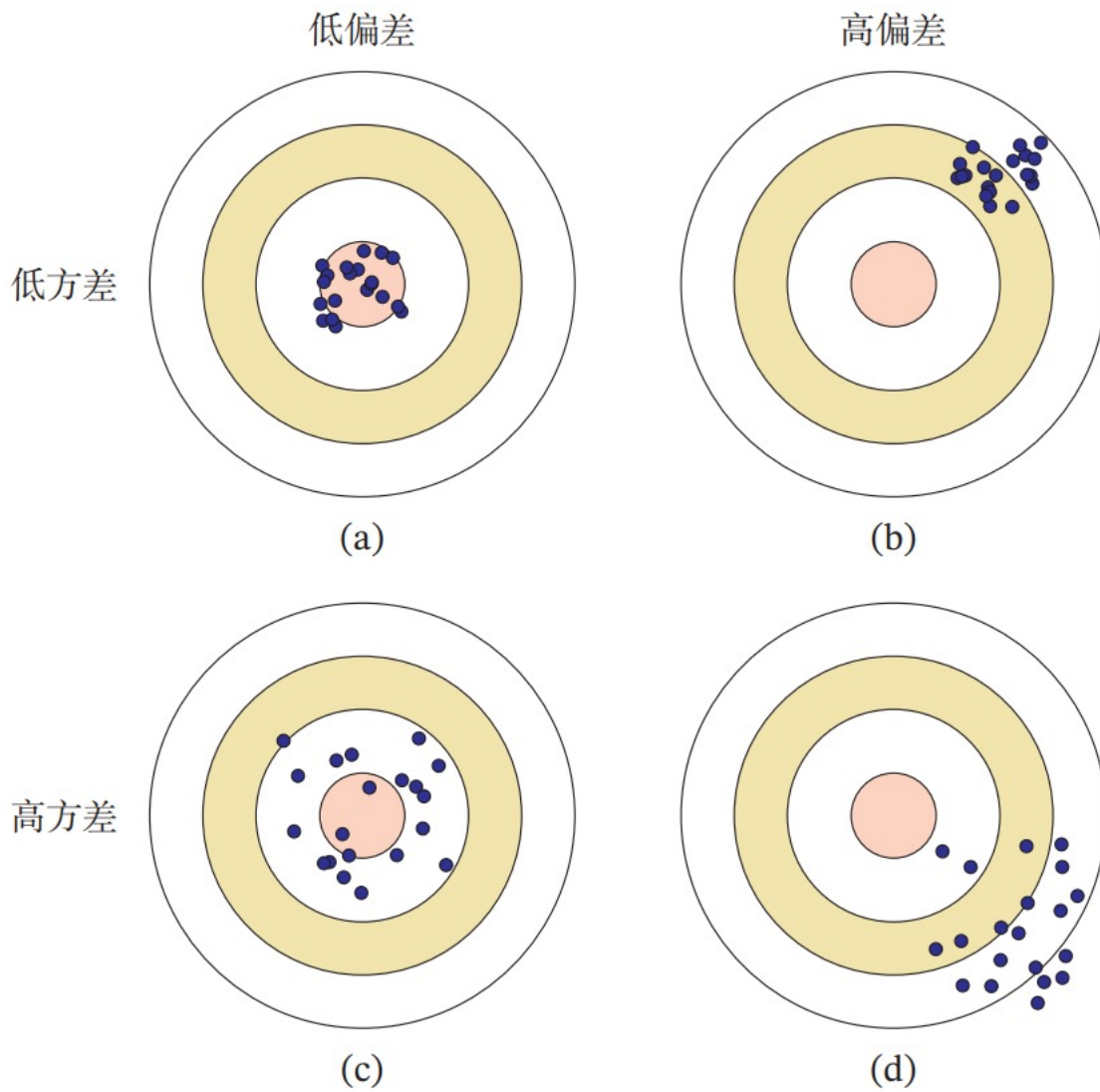
- 样本数量 $N$ 越大，泛化误差上界越小
- 假设空间越大（模型越复杂），泛化误差上界越大

一个好的模型，即是能做到——**多刷题**（充足数据），**巧学习**（控制复杂度），**得高分**（低训练误差），**赢高考**（优异泛化）的“学霸”。

# 模型选择的指标——偏差 (bias) 与方差 (variance)



## 准与确



# 模型选择的指标——偏差 (bias) 与方差 (variance)



偏差(bias): 期望输出 $\bar{f}(x)$ 与真实标记 $y$ 的偏离程度。

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

方差(variance):  $x$ 在训练集 $D$ 上学得模型 $f(x; D)$ 上的输出与期望输出 $\bar{f}(x)$ 之间的变动导致的学习性能的变化。

$$var(x) = E_D[(f(x; D) - \bar{f}(x))^2]$$

噪声:  $\epsilon^2 = E_D[(y_D - y)^2]$



假定噪声期望  $E_D [y - y_D]$  为0

$$E[y_D] = f(x; D) + \varepsilon$$

$$E[\varepsilon] = 0$$

泛化误差=错误 (**error**) =  $E(f; D) = E_D[(f(x; D) - y_D)^2]$

$$\begin{aligned} &= E_D [(f(x; D) - \bar{f}(x) + \bar{f}(x) - y_D)^2] \\ &= E_D [(f(x; D) - \bar{f}(x))^2] + E_D [(\bar{f}(x) - y_D)^2] + E_D [2 (f(x; D) - \bar{f}(x)) (\bar{f}(x) - y_D)] \\ &= E_D [(f(x; D) - \bar{f}(x))^2] + E_D [(\bar{f}(x) - y_D)^2] \\ &= \text{var}(x) + E_D [(\bar{f}(x) - y + y - y_D)^2] \\ &= \text{var}(x) + E_D [(\bar{f}(x) - y)^2] + E_D [(y - y_D)^2] + E_D [2(\bar{f}(x) - y)(y - y_D)] \\ &= \text{var}(x) + E_D [(\bar{f}(x) - y)^2] + E_D [(y - y_D)^2] \\ &= \text{var}(x) + \text{bias}^2(x) + \varepsilon^2 \end{aligned}$$



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

$\text{var}(x)$

训练集的变动（数据扰动）导致的影响

$\text{bias}^2(x)$

算法本身的拟合能力

$\varepsilon^2$

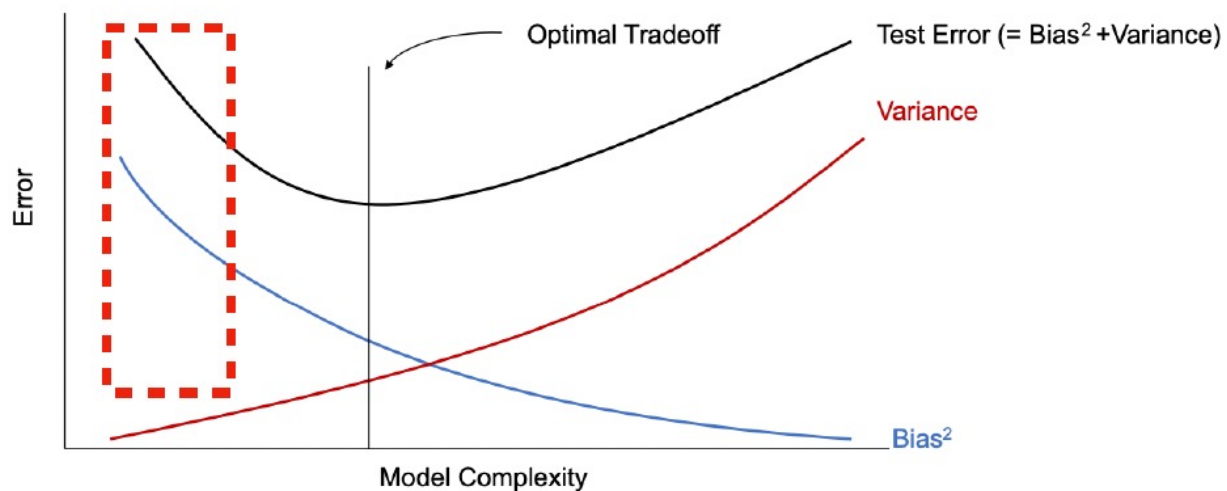
问题本身的难度



# 偏差与方差

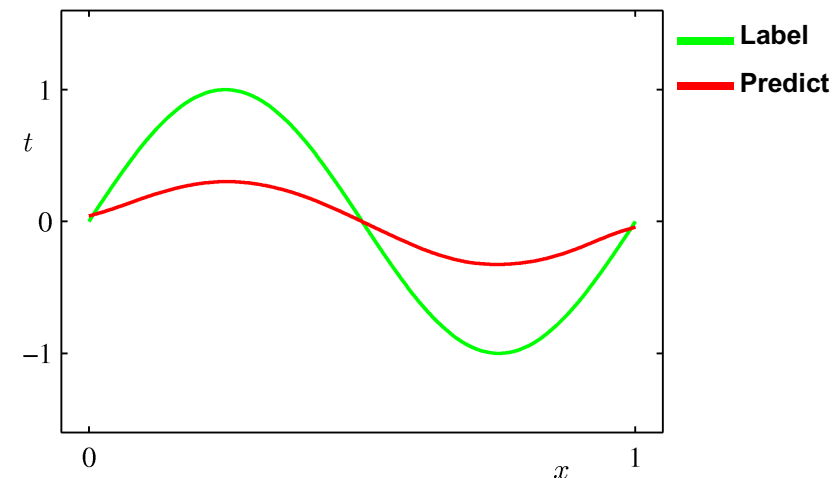
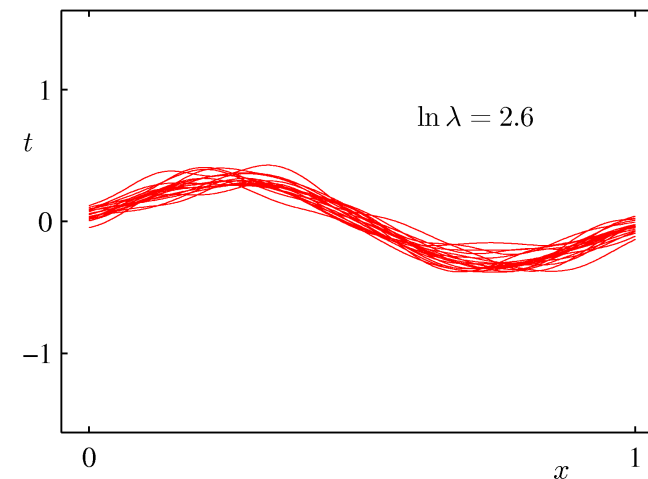


high bias  
underfit



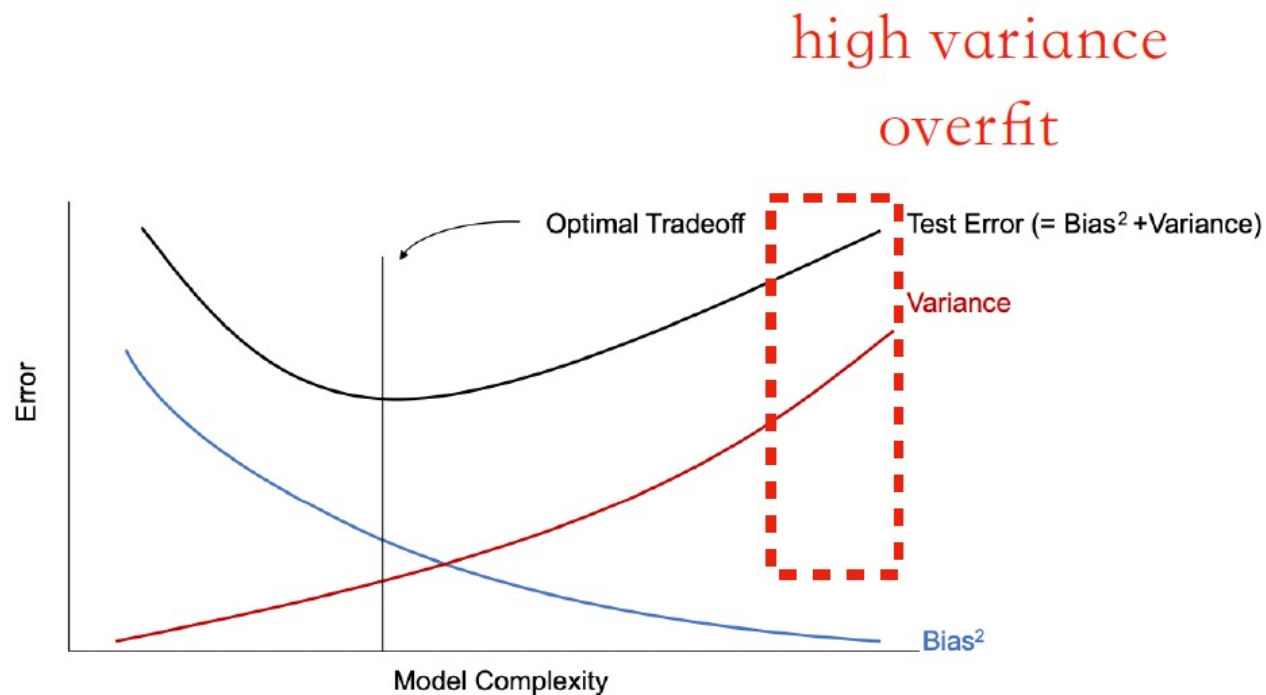
complexity  $\longleftrightarrow$  fitting  
trade-off

$$f(x) = \sin(2\pi x)$$



随机抽取25个点建立100模型， $\lambda$ ：正则化强度

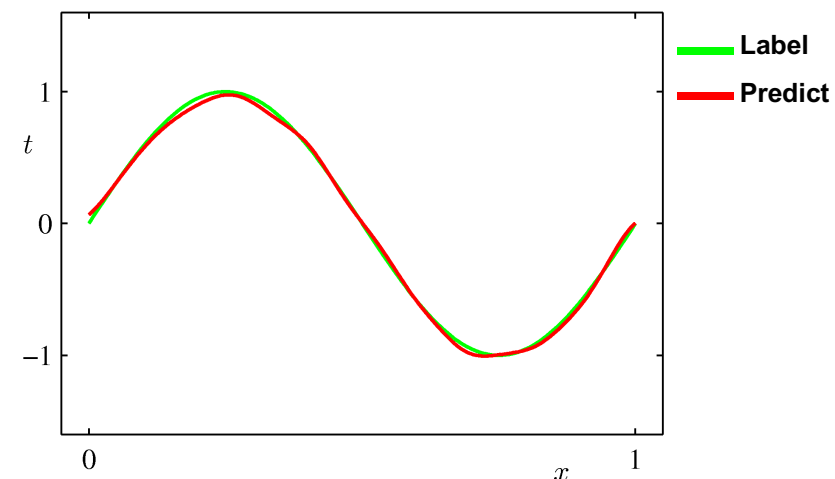
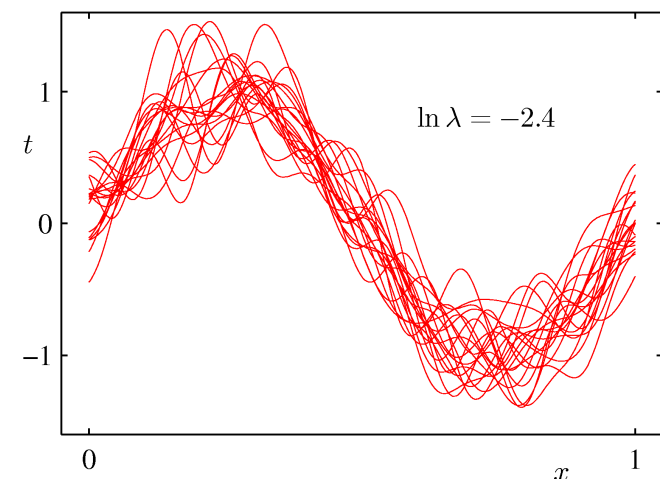
# 偏差与方差



complexity  $\longleftrightarrow$  fitting

trade-off

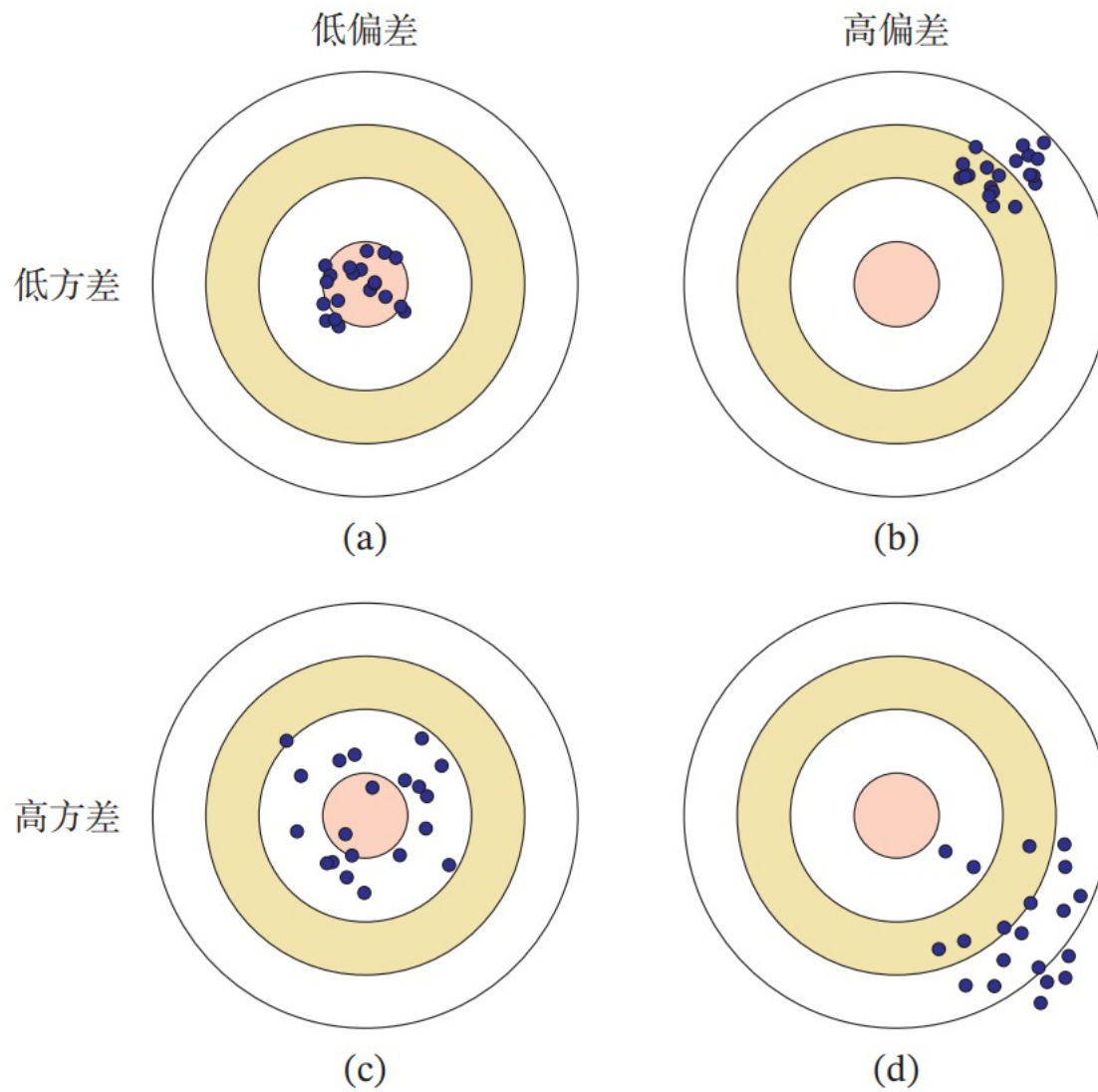
$$f(x) = \sin(2\pi x)$$



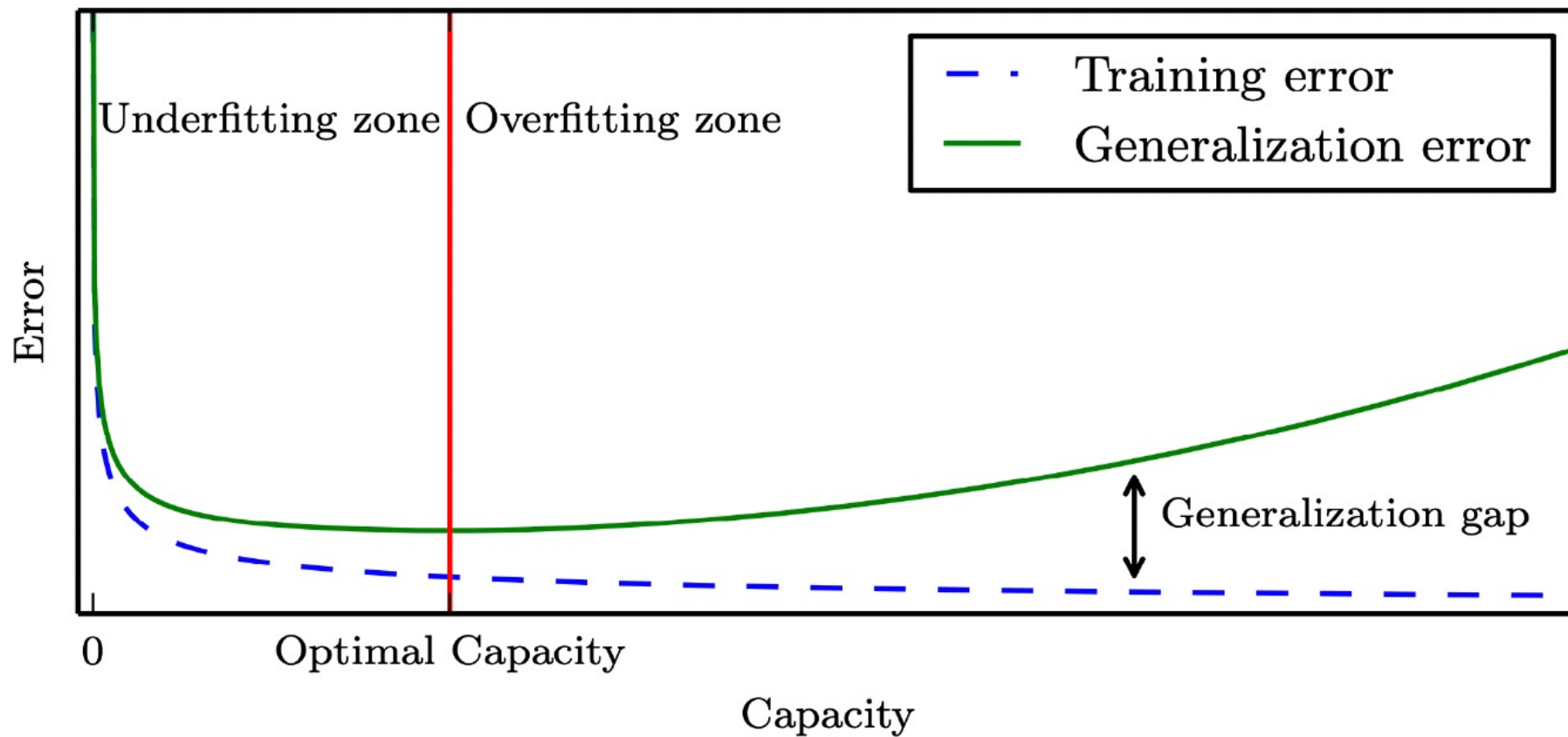
随机抽取25个点建立100模型,  $\lambda$ : 正则化强度



## 准与确



# 可能存在的问题——欠拟合和过拟合



欠拟合：经验误差和泛化误差都非常高

过拟合：经验误差降低、泛化误差反升

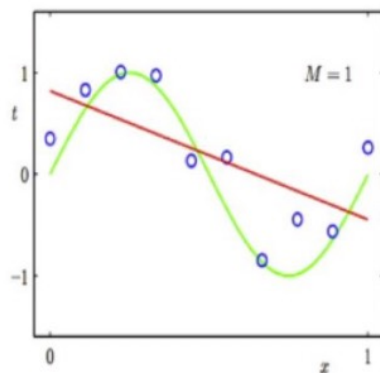


# 欠拟合和过拟合

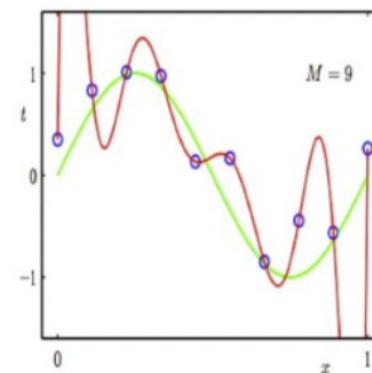
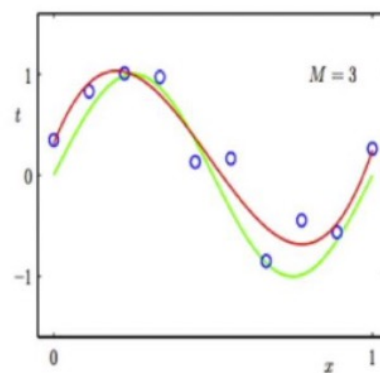


以回归和分类为例：

Regression:

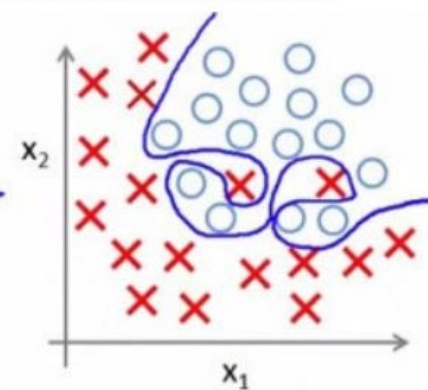
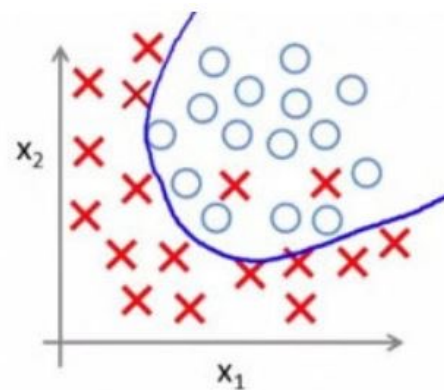
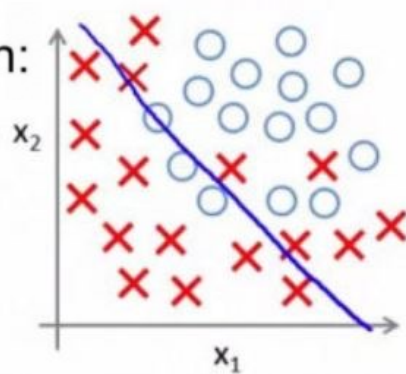


predictor too inflexible:  
cannot capture pattern



predictor too flexible:  
fits noise in the data

Classification:

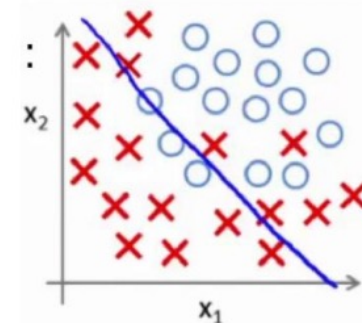
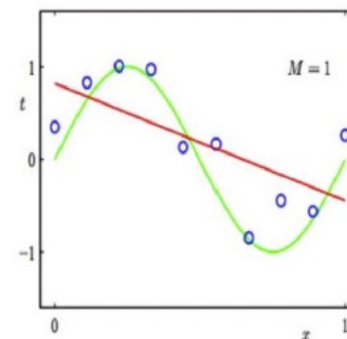


# 欠拟合和过拟合



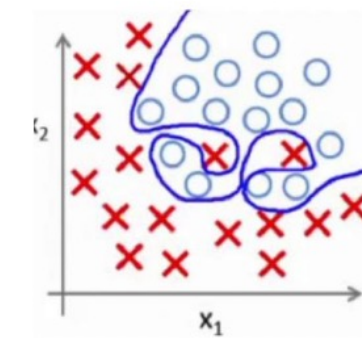
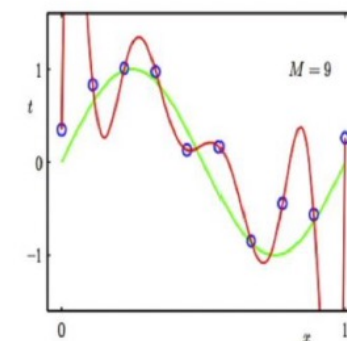
欠拟合：模型没有在训练集上取得足够低的误差  
(模型表达能力太低或训练迭代次数不足)

改用表达能力更强的模型、  
增加训练迭代次数



过拟合：训练误差和测试误差之间的差距太大  
(模型表达能力过强，训练样本不足导致拟合训练集噪声)

增加训练数据、参数正则化、早停止策略



# 缓解过拟合——正则化



正则化 (Regularization) 指对降低泛化误差而非训练误差的修改

权重衰减 (Weight Decay)

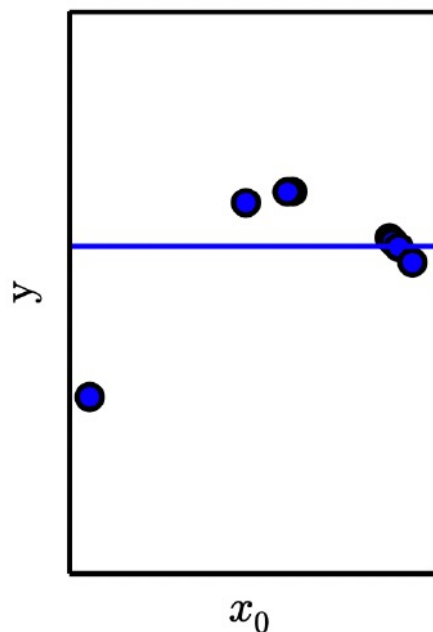
以训练九次多项式为例

$$y = \sum_i^9 a_i x^i$$

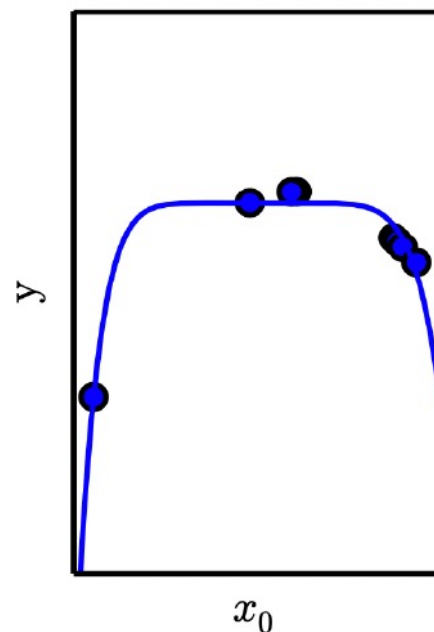
$$w = [a_0, a_1, \dots, a_9]$$

$$J(w) = MSE_{train} + \lambda w^T w$$

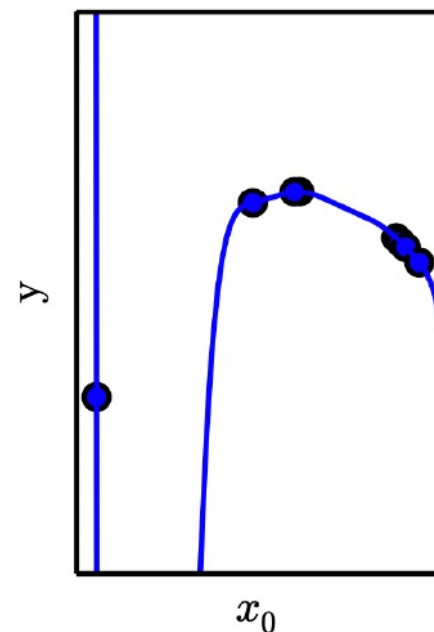
Underfitting  
(Excessive  $\lambda$ )



Appropriate weight decay  
(Medium  $\lambda$ )



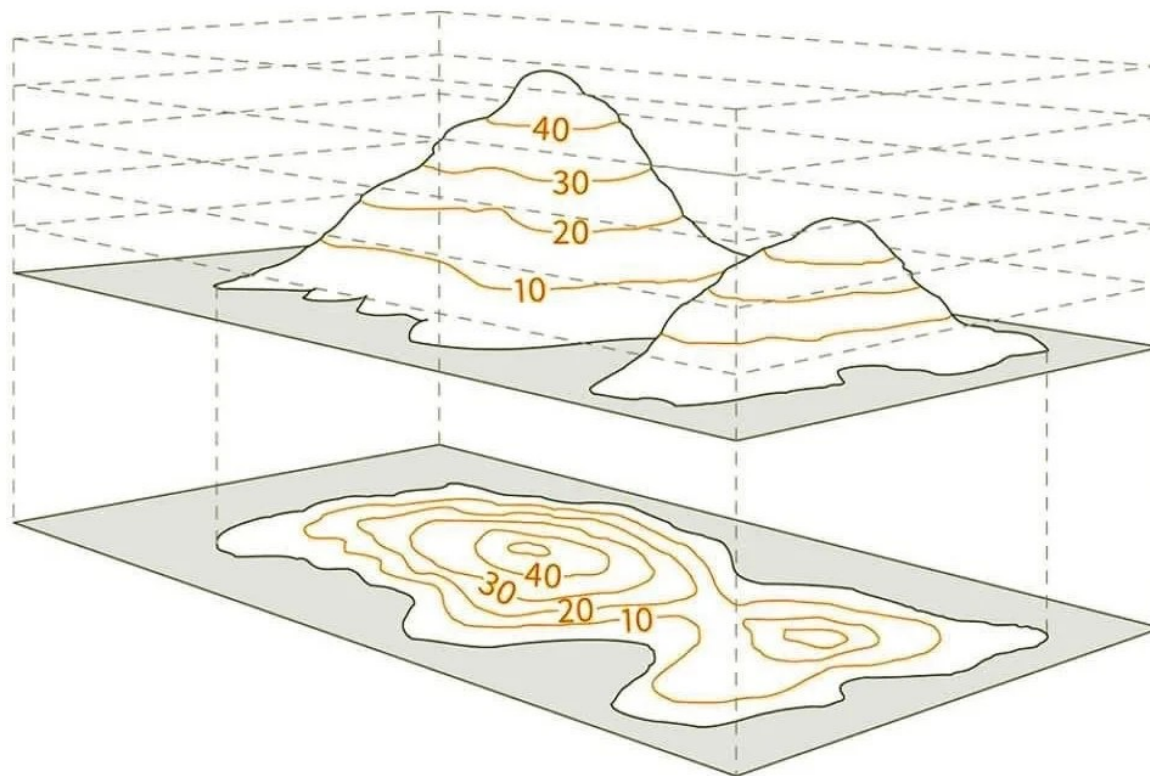
Overfitting  
( $\lambda \rightarrow 0$ )



# 缓解过拟合——正则化



损失函数等高线 (Contour line) :



- 等高线中心：山顶，损失函数最小，优化效果最优
- 等高线中间：半山腰，可以选择任意圈上的点达到同样的损失函数



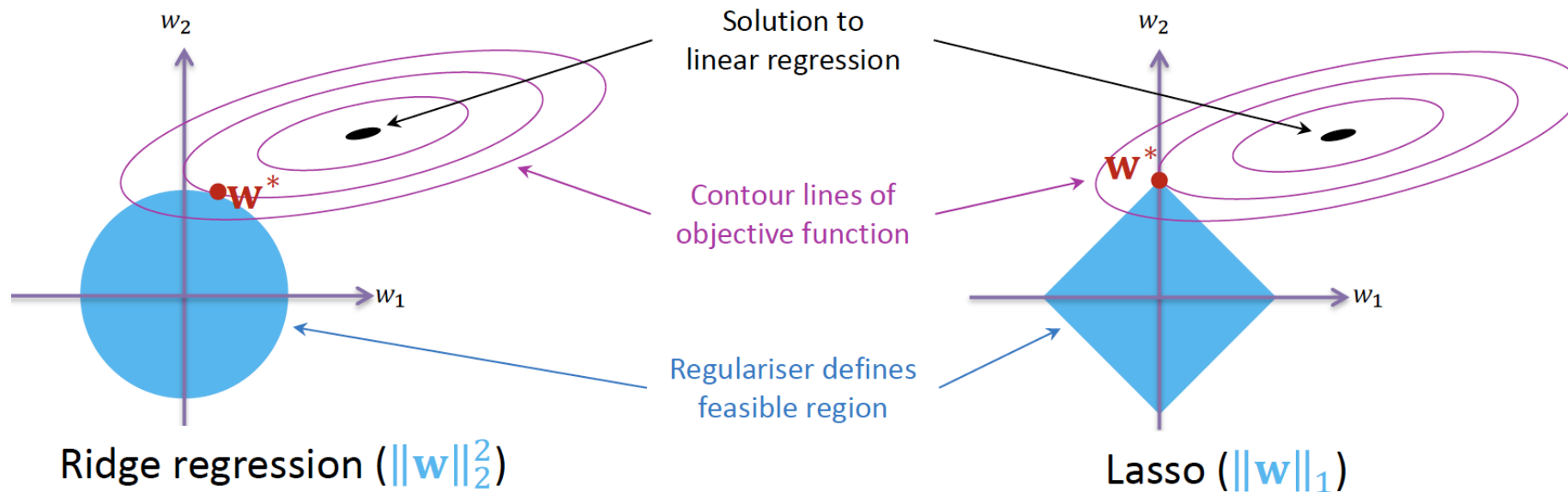
# 缓解过拟合——正则化



权重衰减 (Weight Decay) : 选择靠近坐标轴的点, 让拟合函数参数量变小

$$\|y - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

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



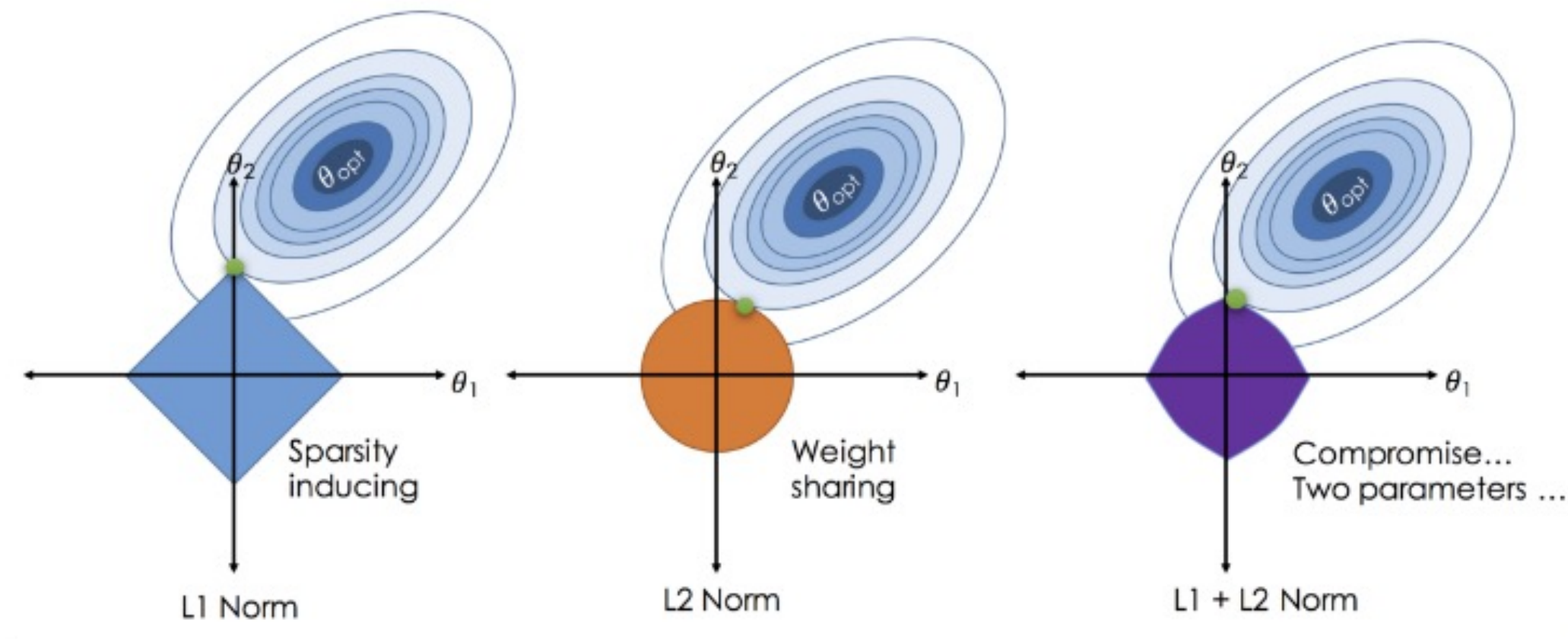
- Lasso (L1正则) : 让最优解更多的在坐标轴上, 产生稀疏化 (Sparsity) 效果
- Ridge正则 (L2正则) : 让最优解离原点最近, 让参数变小



# 缓解过拟合——正则化



权重衰减 (Weight Decay) :



Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." *Statistical Methodology* 67.2 (2005): 301-320.

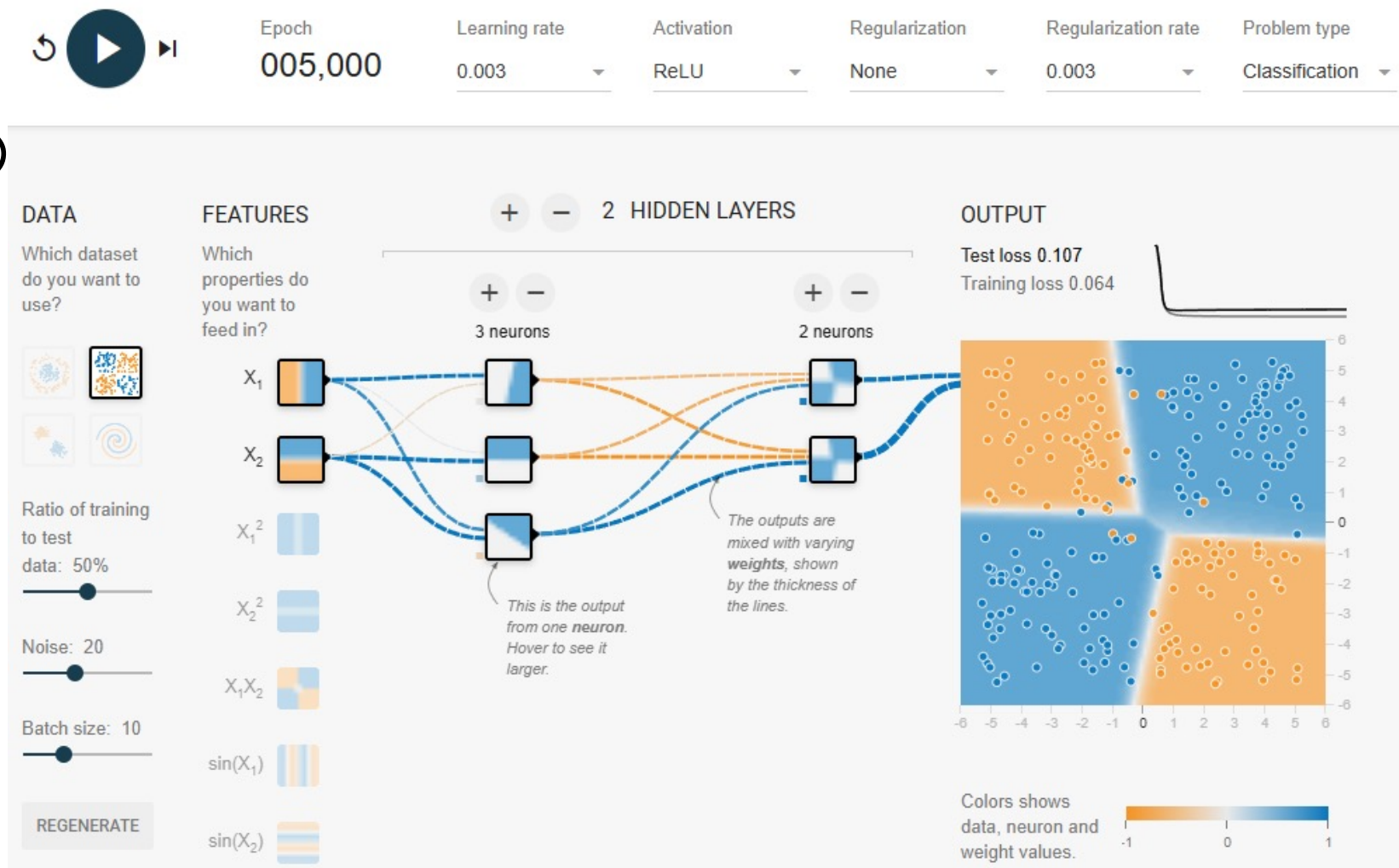


# 缓解过拟合——正则化



样例:

数据分类  
(无权重衰减)



Train loss: 0.064  
Test loss: 0.107

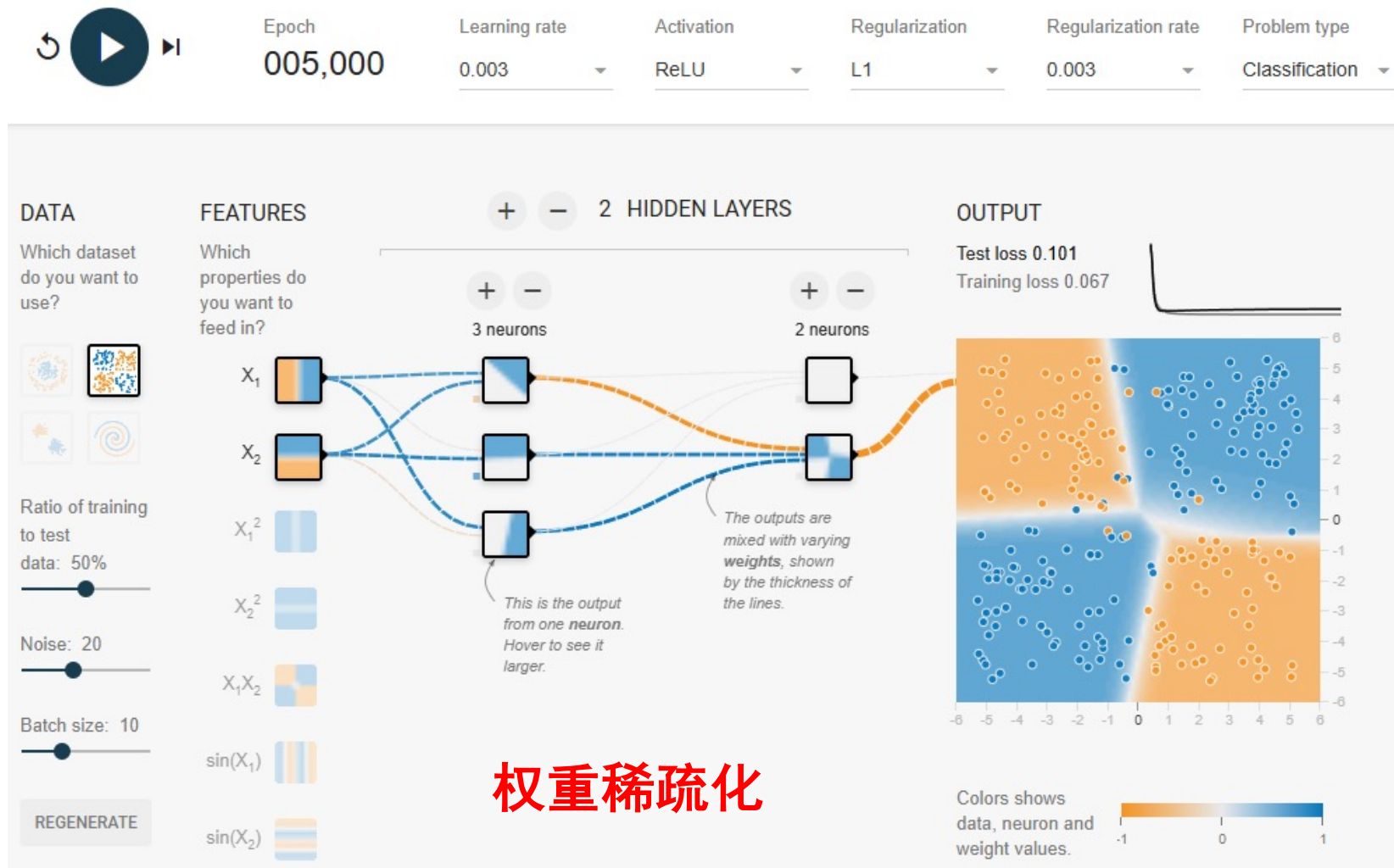


# 缓解过拟合——正则化



样例:

数据分类  
(L1 norm)



Train loss: 0.067

Test loss: 0.101

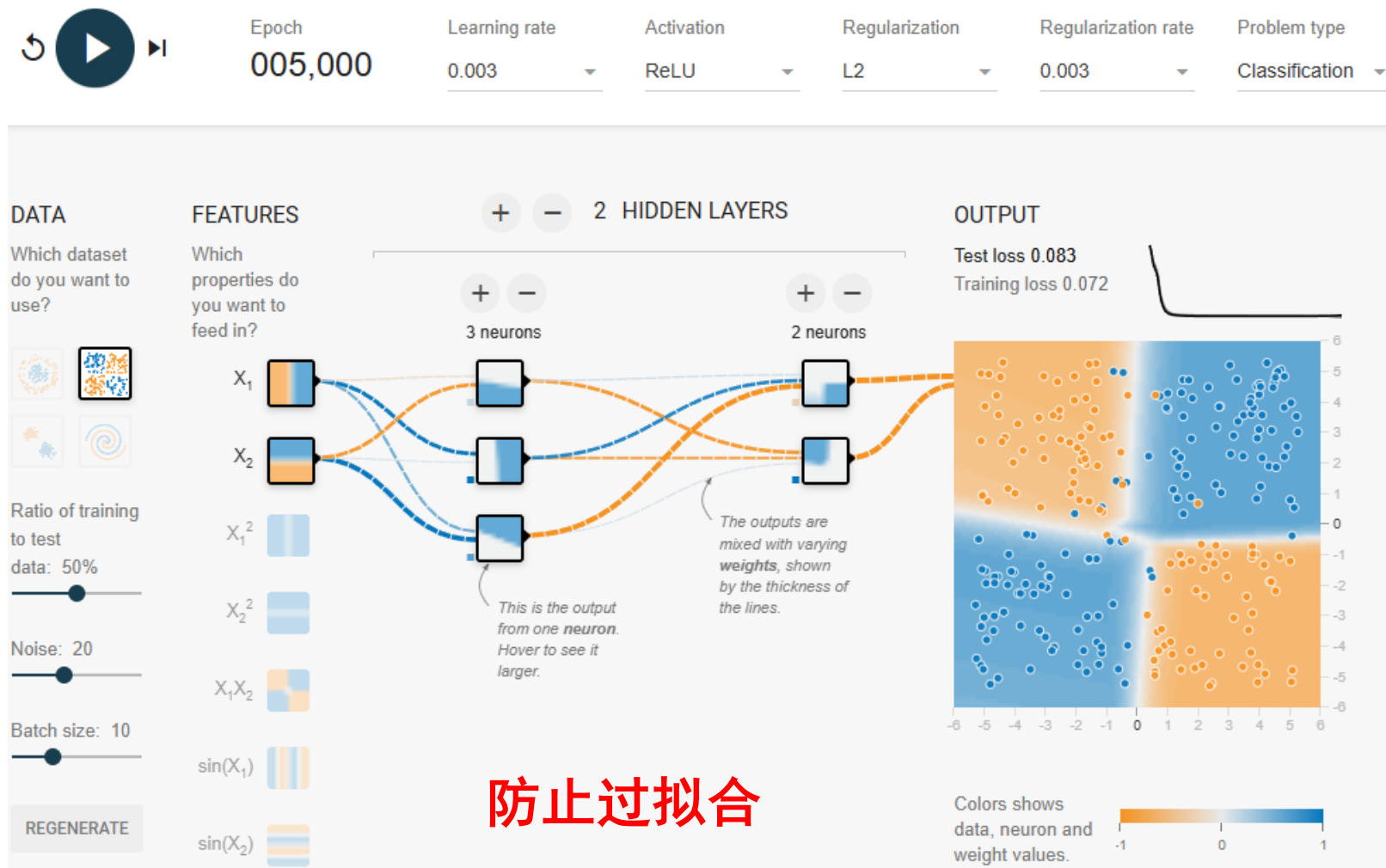


# 缓解过拟合——正则化



样例:

数据分类  
(L2 norm)



无权重衰减:

Train loss: 0.064

Test loss: 0.107

L1 norm:

Train loss: 0.067

Test loss: 0.101

L2 norm:

Train loss: 0.083

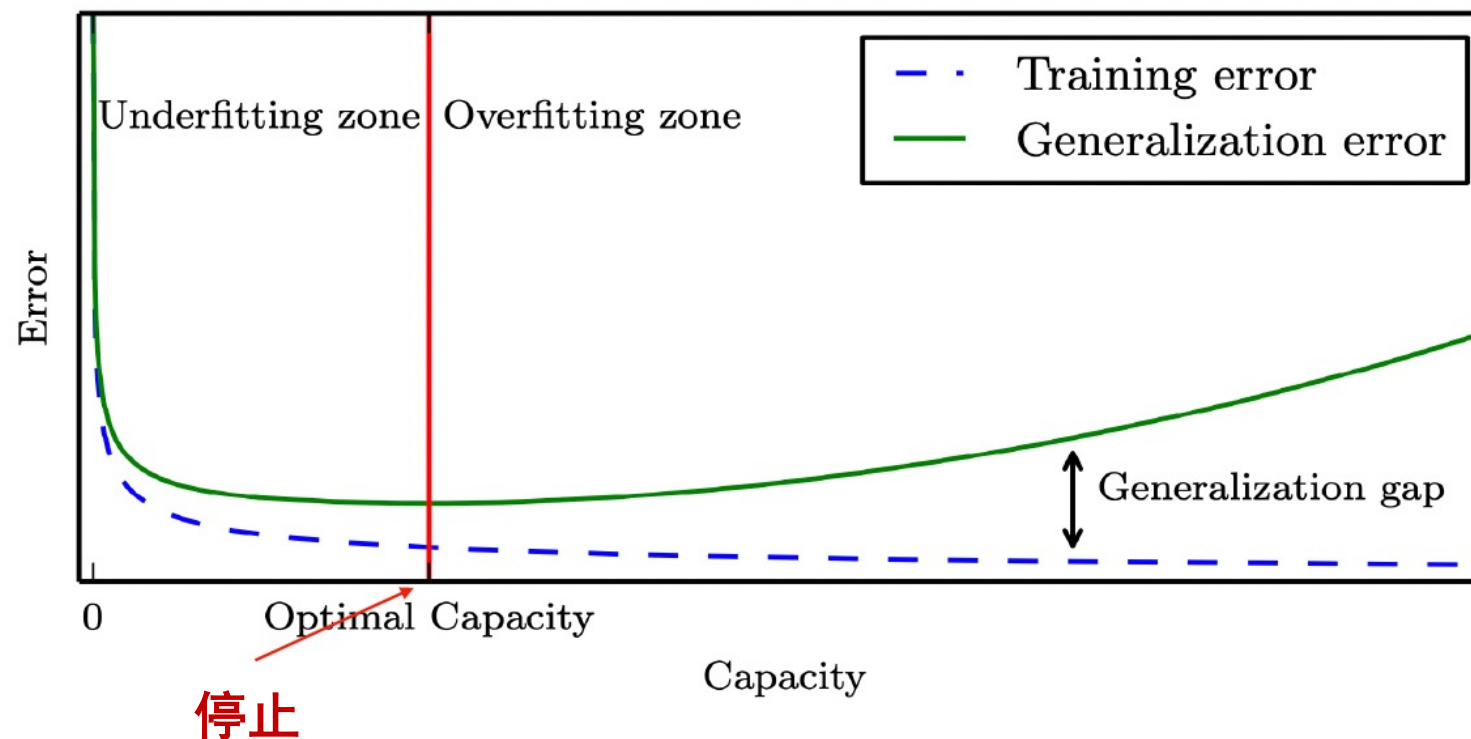
Test loss: 0.072



# 缓解过拟合——早停策略



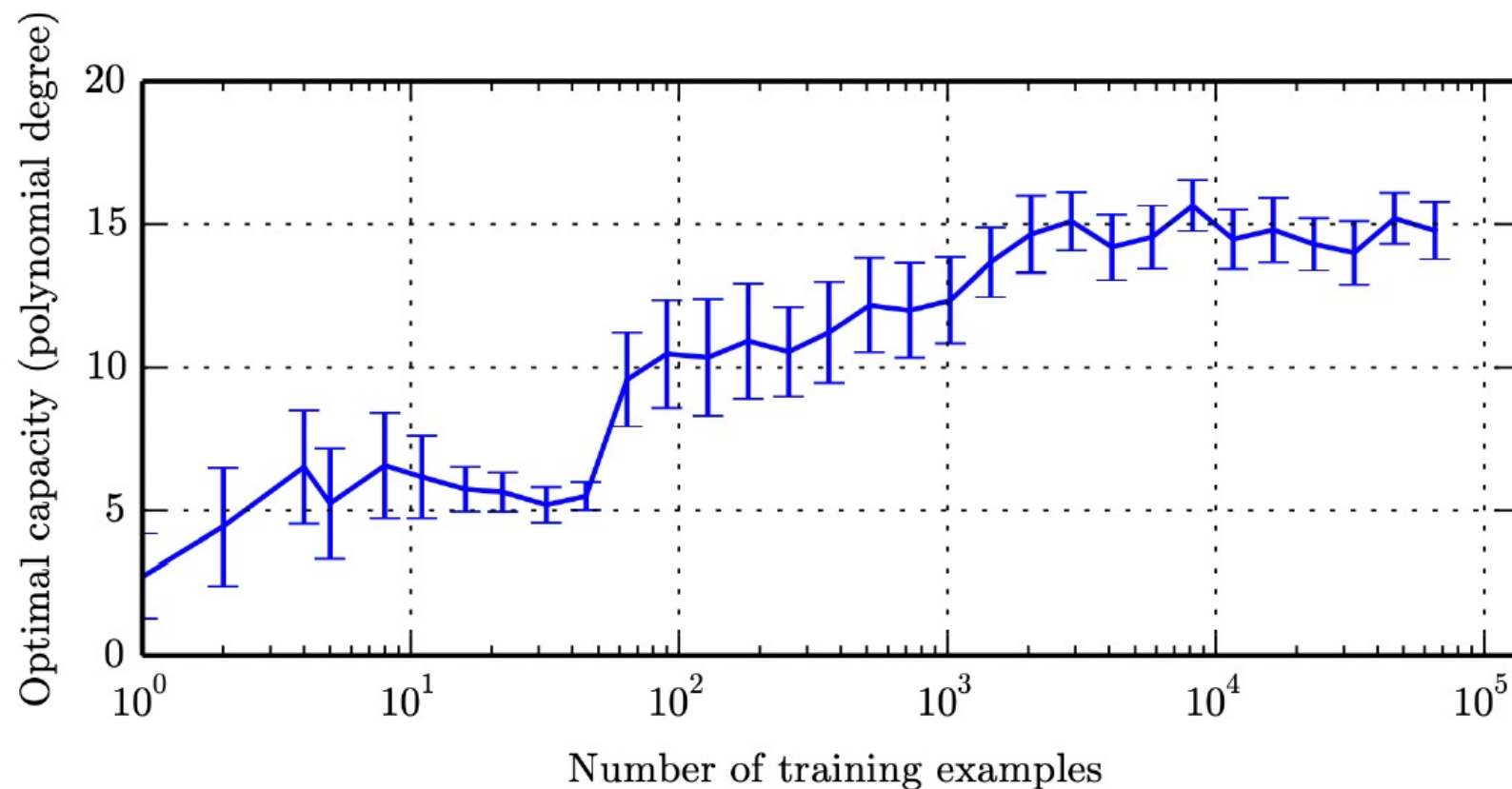
- 将数据划分为训练集和验证集，验证集用来估计误差
- 若训练集误差降低，验证集误差升高，则停止训练



# 缓解过拟合——增加训练数据



- 增加训练数据可以有效防止过拟合
- 大数据量可以抵消模型拟合能力过强的问题



3. (5分) 以下关于模型评估与选择的说法, 哪一项是正确的? ( )
- A. 在训练集上准确率最高的模型一定具有最好的泛化能力
  - B. 增加模型的复杂度总是能提高模型在测试集上的表现
  - C. 当模型在训练集和测试集上的误差都很高时, 模型可能存在欠拟合问题
  - D. K折交叉验证中, K值越大, 每次训练使用的数据量越少

3. (5分) 以下关于模型评估与选择的说法, 哪一项是正确的? ( )
- A. 在训练集上准确率最高的模型一定具有最好的泛化能力
  - B. 增加模型的复杂度总是能提高模型在测试集上的表现
  - C. 当模型在训练集和测试集上的误差都很高时, 模型可能存在欠拟合问题
  - D. K折交叉验证中, K值越大, 每次训练使用的数据量越少

# 目录

1 概述

2 数据集划分方法

3 模型评估方法

4 模型选择方法

5 **大模型时代模型选择**

**Benchmark**, 通常指的是用来评估和比较不同模型性能的一组标准化测试任务或数据集。通过基准测试, 研究者和开发者可以量化模型在特定任务上的表现, 例如准确率、推理速度、计算效率等。

- **性能评估**: 提供量化的标准来比较不同模型在各种任务上的表现
- **跨模型比较**: 允许不同团队和研究人员进行公平的模型性能比较
- **模型优化**: 帮助识别模型的弱点或改进空间, 指导优化方向
- **标准化**: 确保在不同的环境下测试的一致性, 避免实验中由于环境或设置差异导致的结果不一致

# 性能评估基准 (Benchmark)



Organization ↕	Model ↕	License ↕	GPQA 🗃️ ↕	MMLU 🗃️ ↕	MMLU Pro 🗃️ ↕	DROP 🗃️ ↕	HumanEval 🗃️ ↕
	<a href="#">o3</a>	Proprietary	87.7%	-	-	-	-
	<a href="#">o3-mini</a>	Proprietary	79.7%	86.9%	-	-	-
	<a href="#">o1-pro</a>	Proprietary	79.0%	-	-	-	-
	<a href="#">o1</a>	Proprietary	75.7%	91.8%	-	-	88.1%
	<a href="#">Gemini 2.0 Flash Thinking</a>	Proprietary	74.2%	-	-	-	-
	<a href="#">o1-preview</a>	Proprietary	73.3%	90.8%	-	-	-
	<a href="#">DeepSeek-R1</a>	Open 🗃️	71.5%	90.8%	84.0%	92.2%	-
	<a href="#">Claude 3.5 Sonnet</a>	Proprietary	67.2%	90.4%	77.6%	87.1%	93.7%
	<a href="#">QwQ-32B-Preview</a>	Open 🗃️	65.2%	-	-	-	-
	<a href="#">Gemini 2.0 Flash</a>	Proprietary	62.1%	-	76.4%	-	-
	<a href="#">o1-mini</a>	Proprietary	60.0%	85.2%	-	-	92.4%
	<a href="#">DeepSeek-V3</a>	Open 🗃️	59.1%	88.5%	75.9%	91.6%	-



# 性能评估基准 (Benchmark)



	通用模型 GPT-4.5	通用模型 GPT-4o	推理模型 OpenAI o3-mini (high)
GPQA (science)	71.4%	53.6%	79.7%
AIME '24 (math)	36.7%	9.3%	87.3%
MMMLU (multilingual)	85.1%	81.5%	81.1%
MMMU (multimodal)	74.4%	69.1%	-
SWE-Lancer Diamond (coding)*	32.6%	23.3%	10.8%
	\$186,125	\$138,750	\$89,625
SWE-Bench Verified (coding)*	38.0%	30.7%	61.0%

数据来源: <https://openai.com/index/introducing-gpt-4-5/>



## GPQA Benchmark (General-Purpose Question Answering)

### 研究生级别的知识挑战

#### 任务特点:

- **多领域**: 包括从学术到日常生活等多个领域的问题, 具有更高的多样性
- **问题类型**: 除了标准的事实性问答, 还涉及推理、解释等问题, 评估深度理解能力
- **答案类型**: 问题的答案有开放式的, 也可以是选择题的形式, 测试模型的灵活性

#### 主要用途:

- **评估模型的通用推理能力**: 模型在不同领域和任务下的适应能力
- **跨领域知识的检验**: 测试其在不同领域中的表现是否均衡

## GPQA Benchmark (General-Purpose Question Answering)

### 研究生级别的知识挑战

#### 样例题目:











Question: Consider a quantum mechanical system containing a particle of mass  $m$  moving in an isotropic three dimensional potential of the form  $V(r) = 1/2m\omega^2r^2$  corresponding to the acted force obeying Hooke's law. Here,  $\omega$  is the angular frequency of oscillation and  $r$  is the radial distance of the particle from the origin in spherical polar coordinate. What is the value of energy of the third excited state, and how many linearly independent eigenfunctions are possible for the same energy eigenvalue?

Choices:

- (A)  $11 \pi^2 \hbar^2 / (2mr^2)$ , 3
- (B)  $(9/2) \hbar\omega$ , 10
- (C)  $11 \pi^2 \hbar^2 / (2mr^2)$ , 10
- (D)  $(9/2) \hbar\omega$ , 3

# 推理代价评估 (越便宜越好)



Organization ↕	Model ↕	License ↕	Input \$/M ⓘ ↕	Output \$/M ⓘ ↕
	<a href="#">o3</a>	Proprietary	-	-
	<a href="#">o3-mini</a>	Proprietary	\$1.10	\$4.40
	<a href="#">o1-pro</a>	Proprietary	-	-
	<a href="#">o1</a>	Proprietary	\$15.00	\$60.00
	<a href="#">Gemini 2.0 Flash Thinking</a>	Proprietary	-	-
	<a href="#">o1-preview</a>	Proprietary	\$15.00	\$60.00
	<a href="#">DeepSeek-R1</a>	Open ⓘ	\$0.55	\$2.19
<b>AI</b>	<a href="#">Claude 3.5 Sonnet</a>	Proprietary	\$3.00	\$15.00
	<a href="#">QwQ-32B-Preview</a>	Open ⓘ	\$0.15	\$0.20
	<a href="#">Gemini 2.0 Flash</a>	Proprietary	-	-
	<a href="#">o1-mini</a>	Proprietary	\$3.00	\$12.00

数据来源: <https://llm-stats.com/>



# 模型选择 -- 大语言模型



Text tokens

Price per 1M tokens · Batch API price

Model	Input	Cached input	Output
gpt-4.5-preview ↳ gpt-4.5-preview-2025-02-27	\$75.00	\$37.50	\$150.00
gpt-4o ↳ gpt-4o-2024-08-06	\$2.50	\$1.25	\$10.00
gpt-4o-audio-preview ↳ gpt-4o-audio-preview-2024-12-17	\$2.50	-	\$10.00
gpt-4o-realtime-preview ↳ gpt-4o-realtime-preview-2024-12-17	\$5.00	\$2.50	\$20.00
gpt-4o-mini ↳ gpt-4o-mini-2024-07-18	\$0.15	\$0.075	\$0.60
gpt-4o-mini-audio-preview ↳ gpt-4o-mini-audio-preview-2024-12-17	\$0.15	-	\$0.60
gpt-4o-mini-realtime-preview ↳ gpt-4o-mini-realtime-preview-2024-12-17	\$0.60	\$0.30	\$2.40
o1 ↳ o1-2024-12-17	\$15.00	\$7.50	\$60.00
o3-mini ↳ o3-mini-2025-01-31	\$1.10	\$0.55	\$4.40

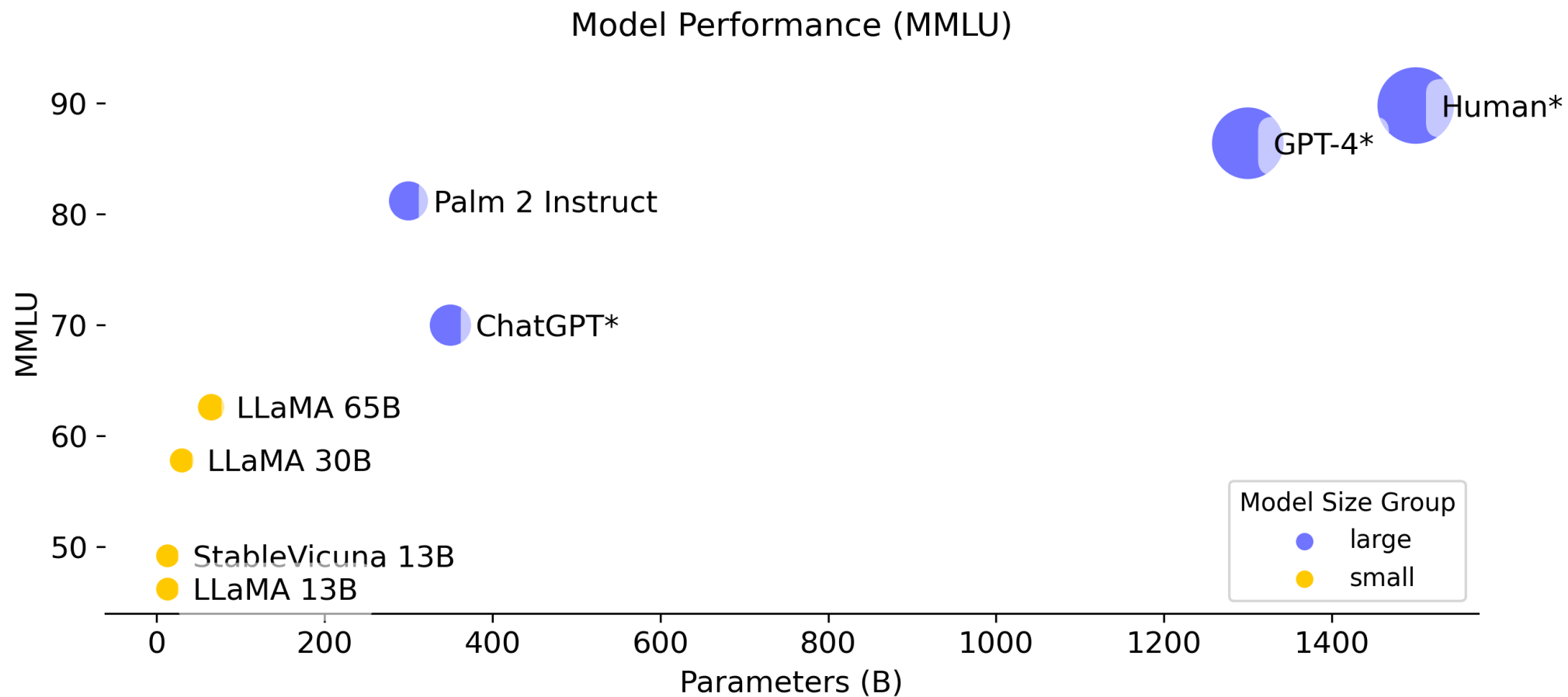
数据来源: <https://platform.openai.com/docs/pricing>



- **模型参数规模分类**

- 小模型 (<1B 参数) : 适用于移动端 / 边缘设备 (如 TinyBERT)
- 中等模型 (1B-70B 参数) : 业务场景通用 (如 LLAMA-7B)
- 大模型 (100B + 参数) : 通用智能 (如 GPT-4、Claude-3)

# 模型大小



\*Exact model size is unknown. | Data from InstructEval GitHub.



- **模型运行需要显存 (显卡内存 Video RAM)**

近似计算公式:

**推理显存  $\approx$  模型参数数量  $\times$  2 bytes (半精度浮点)**

**训练显存  $\approx$  3~6 倍推理显存**

模型规模	参数数量	训练显存需求	推理显存需求
7B	7B	30GB+	10-14GB
13B	13B	60GB+	20-26GB
70B	70B	200GB+	140GB+

# 显卡规格



显卡	显存	核心数量
NVIDIA GeForce RTX 4060	8 GB	3072
NVIDIA GeForce RTX 5090 D	32 GB	21760

显卡	显存	FP64
NVIDIA A100 80GB PCIe	80GB	9.7 TFLOPS
NVIDIA H100 80GB PCIe	80GB	24 TFLOPS

模型规模	参数数量	训练显存需求	推理显存需求
7B	7B	30GB+	10-14GB
13B	13B	60GB+	20-26GB
70B	70B	200GB+	140GB+

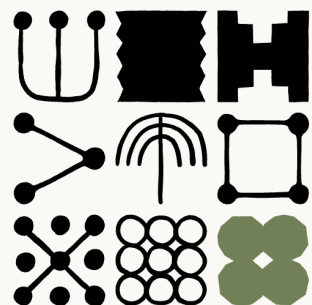


## 昇腾计算

面向“端、边、云”的全场景AI基础设施



- [Anthropic的2026年初博客](#)



## Demystifying evals for AI agents

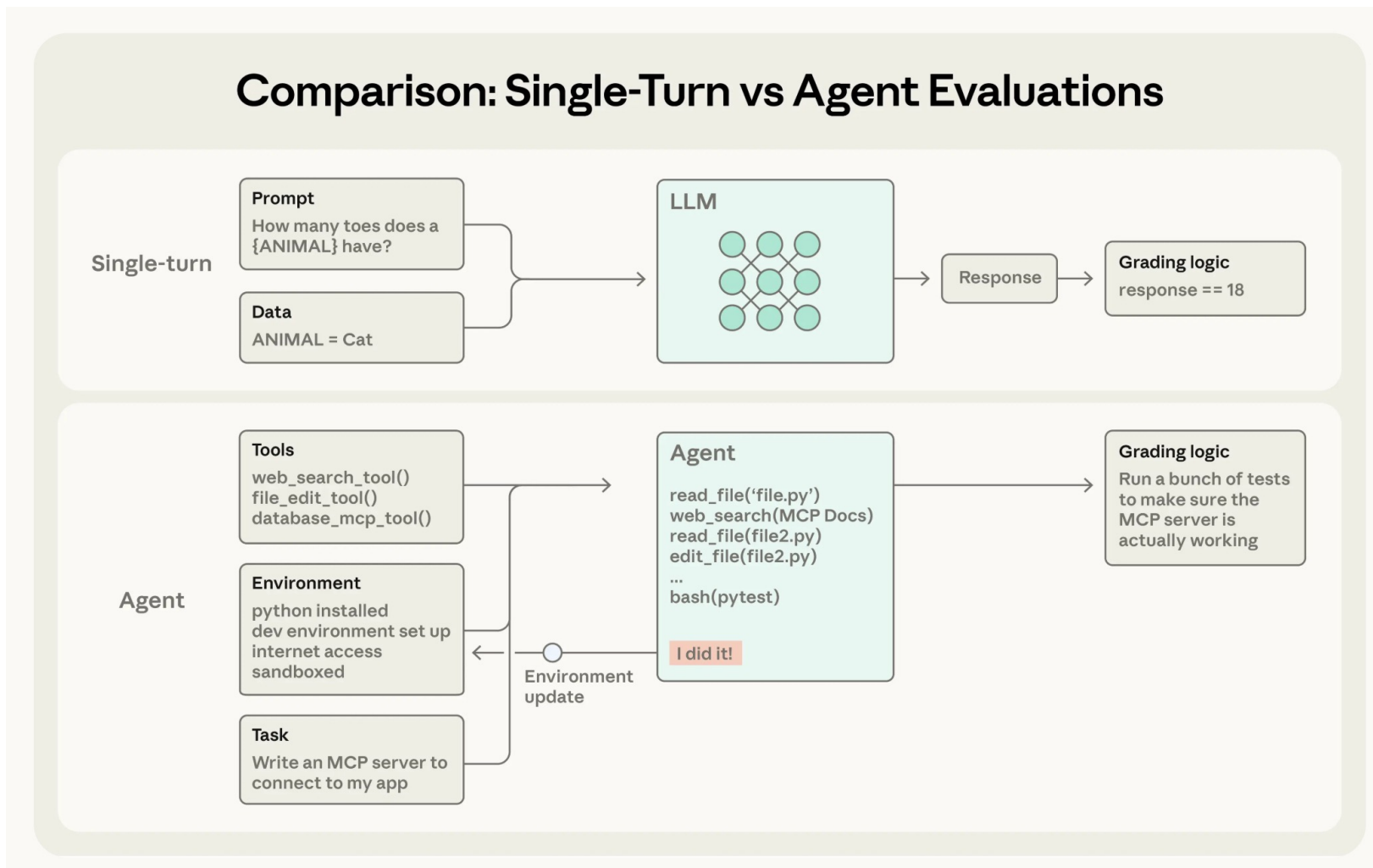
Published Jan 09, 2026

The capabilities that make agents useful also make them difficult to evaluate. The strategies that work across deployments combine techniques to match the complexity of the systems they measure.

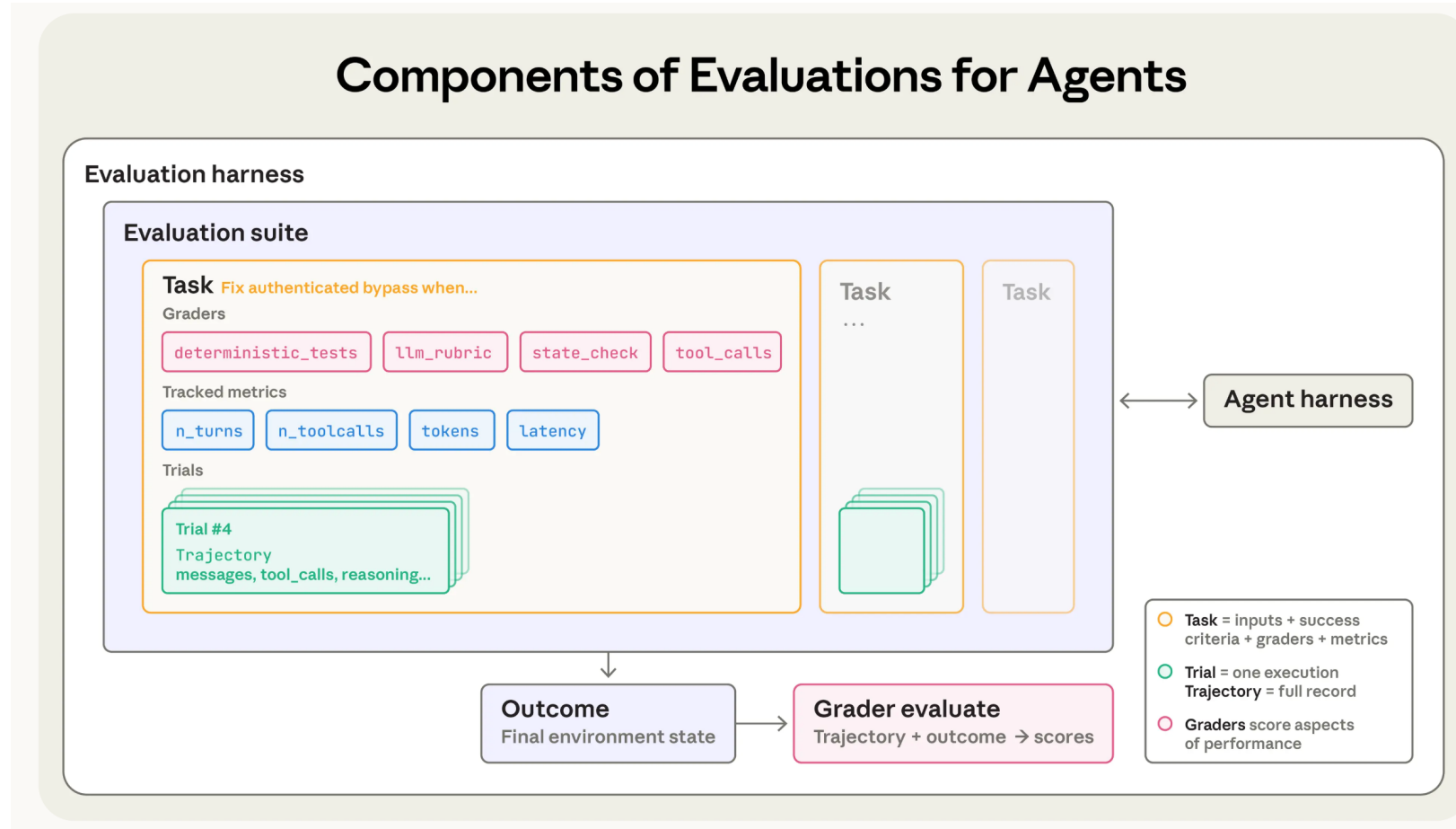
# 前沿内容：AI智能体的评测



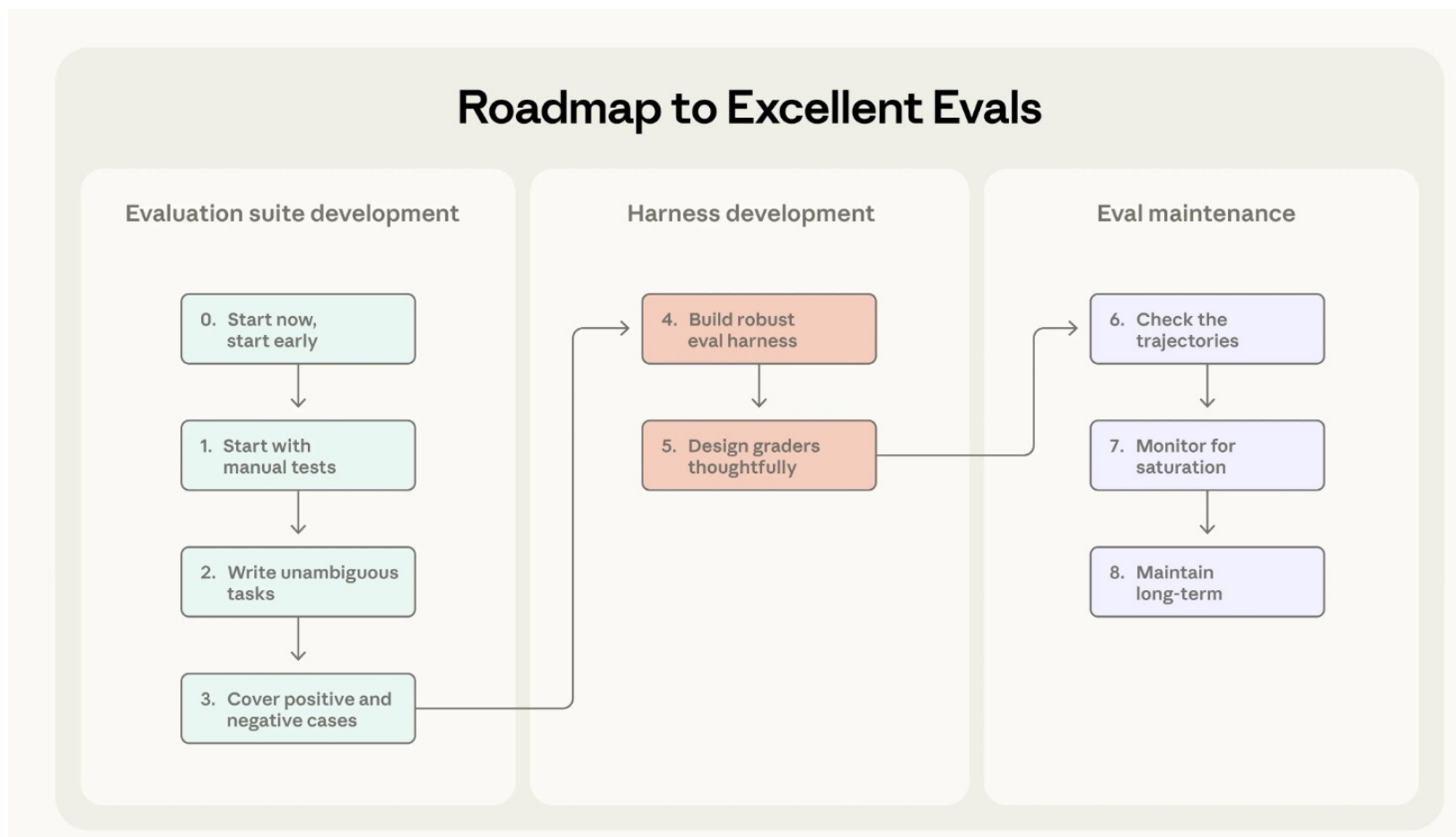
- AI智能体需要用任务成功率来评测



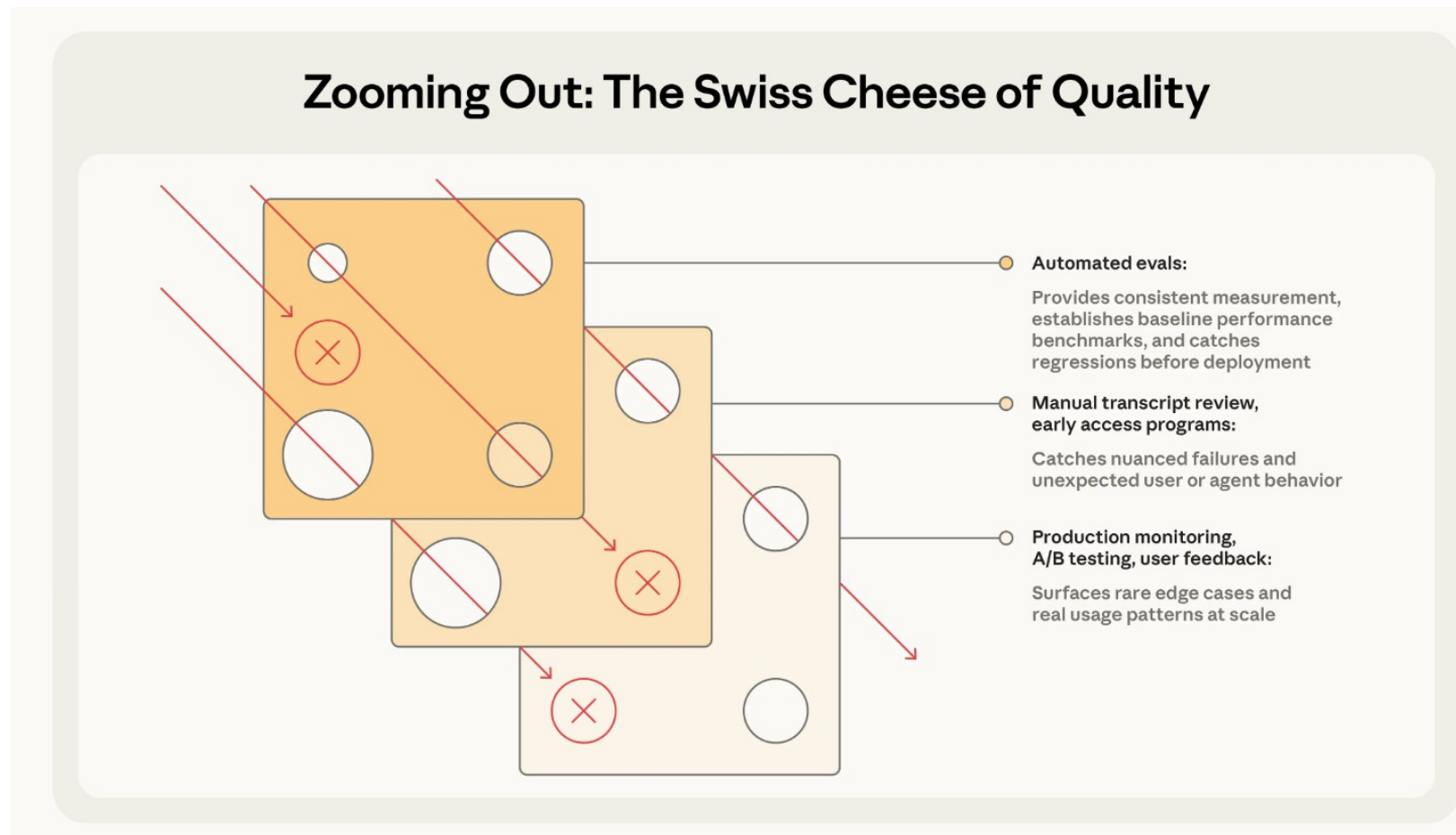
- AI智能体的评测方法组件



- AI智能体的评测流程构建



- AI智能体的安全性保障：安全叠加的瑞士奶酪理论



谢谢！