Python
机器学习实战

机器学习理论

生成方法和判别方法

机器学习原动力

Learn from expert

Learn from data

Learn a function input → ouput

机器学习思想

有监督学习

- 判别方法：（discriminative approach）：决策树、支持向量机、k近邻、逻辑回归
- 生成方法（generative approach）：朴素贝叶斯、HMM

无监督学习

机器学习思想

生成模型：无穷样本→概率密度模型 →产生模型→预测

判别模型：有限样本→判别函数 →预测模型→预测

生成模型更普适、判别模型更直接

生成方法关注数据是如何产生的；寻找的是数据分布模型

判别方法关注数据的差别，寻找的是分类面

由生成模型可以得到判别式模型，但由判别式模型得不到生成式模型

机器学习思想

例如我们有以下(x,y)形式的数据：(1,0), (1,0), (2,0), (2, 1)

	y=0	y=1
x=1	1/2	0
x=2	1/4	1/4

	y=0	y=1
x=1	1	0
x=2	1/2	1/2

P(x,y) ，P(x)

P(y|x)

机器学习解决问题的框架

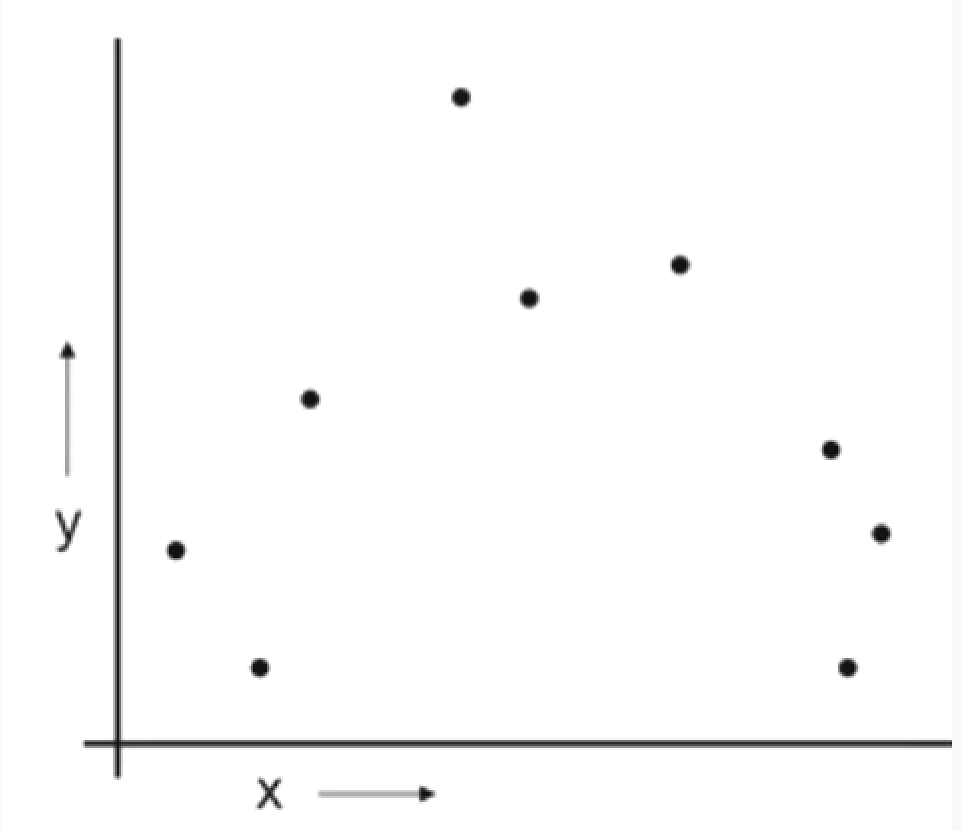
- 定义目标
- 定义模型
- 定义损失函数
- 训练样本
- 优化

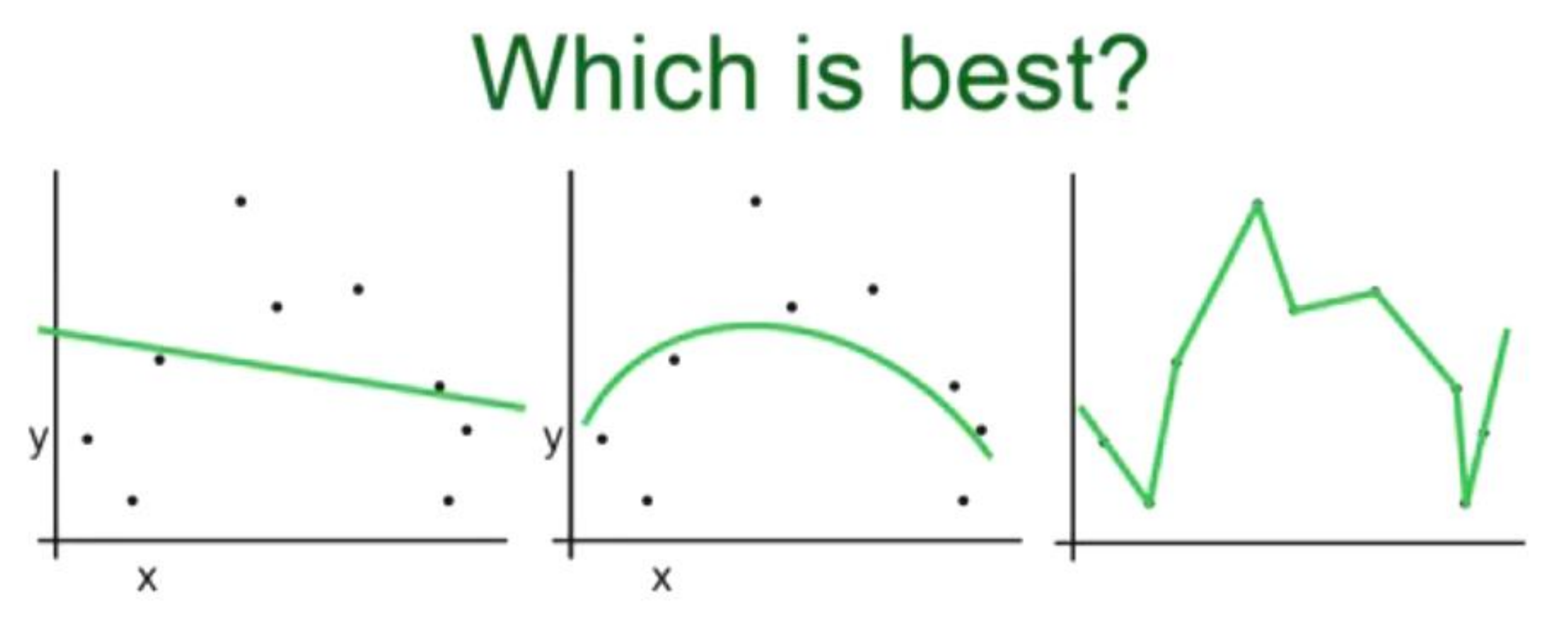
模型、目标和算法

损失函数和正则化

损失函数

- 损失
- 目标：损失最小

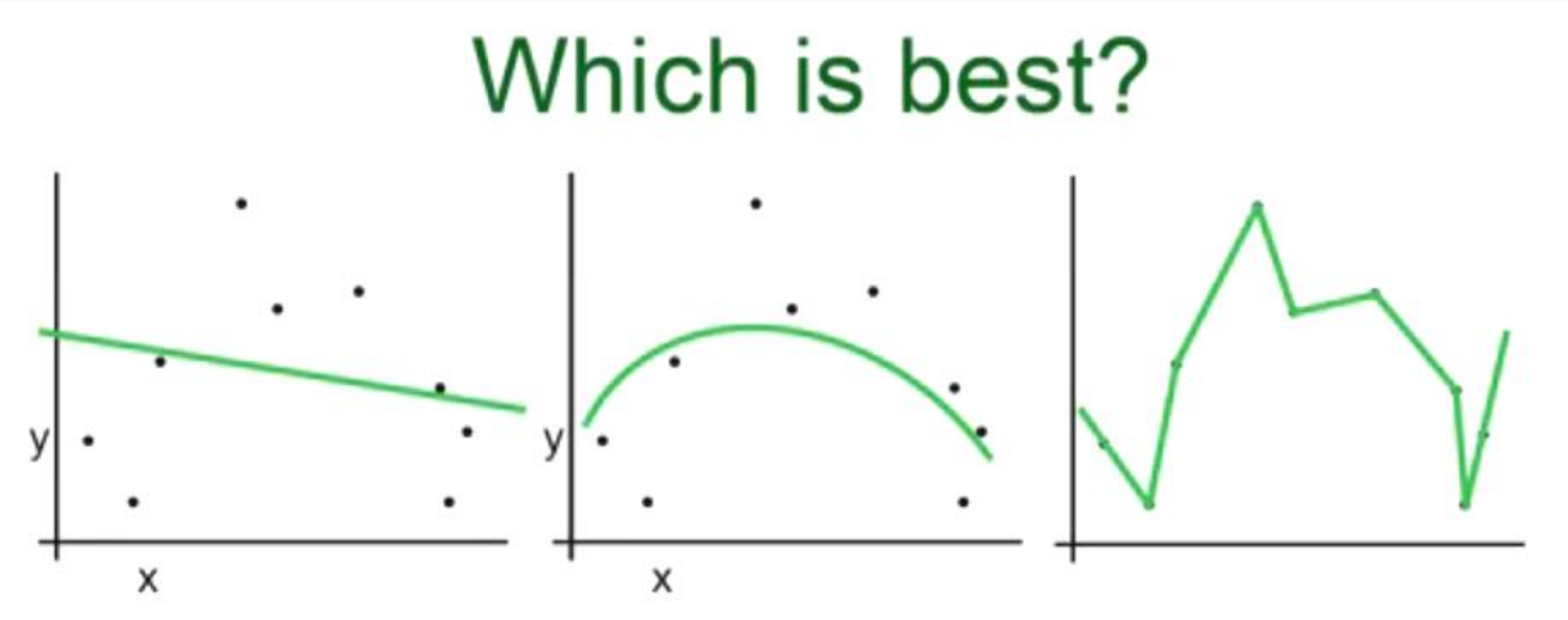


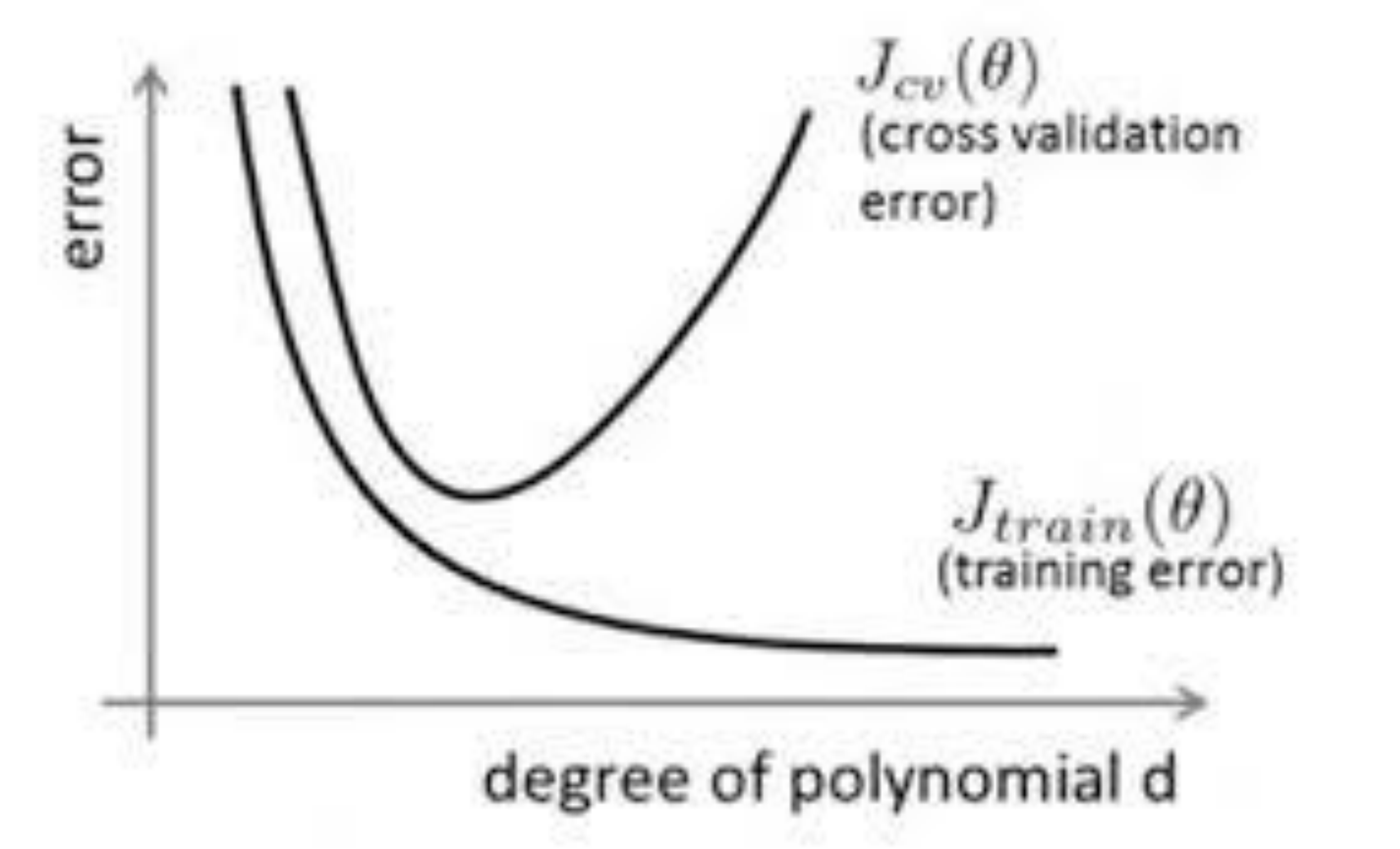


机器学习思想

过拟合

损失最小不是唯一目标

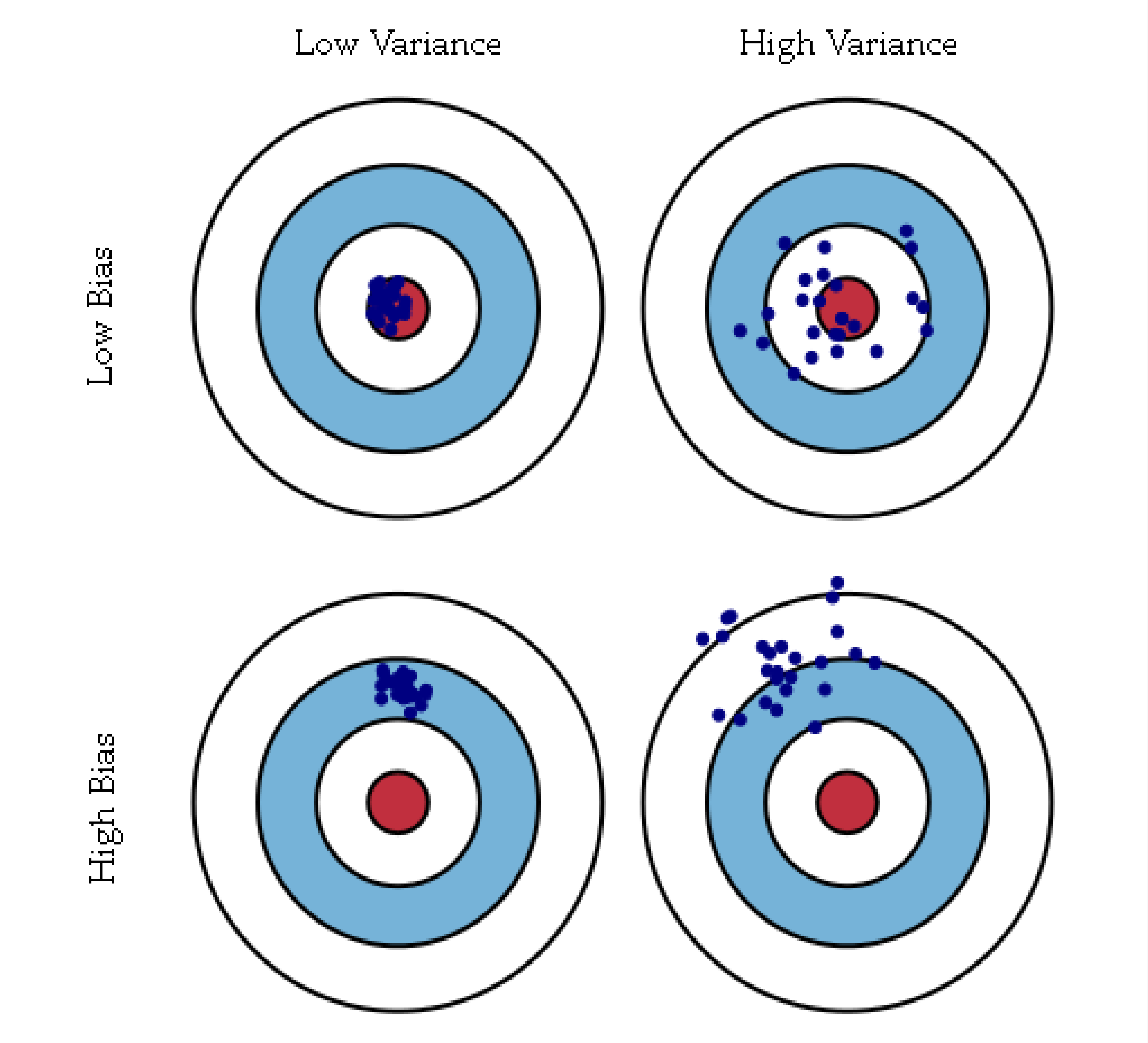




偏差（bias）和方差（variance）

bias表示预测值的均值与实际值的差值，反映的是模型本身的优劣

variance表示预测结果本身的方差，反映的是算法性能的优劣



损失函数一般形式

- 虽外表形式不一，但其本质作用应是唯一的，即用于衡量最优的策略
- 结构风险=经验风险+惩罚项（正则化项）
- L1
- L2

$$J(w) = \sum_i L(m_i(w)) + \lambda R(w)$$
$$m_i = y^{(i)} f_w(x^{(i)})$$
$$y^{(i)} \in \{-1, 1\}$$
$$f_w(x^{(i)}) = w^T x^{(i)}$$

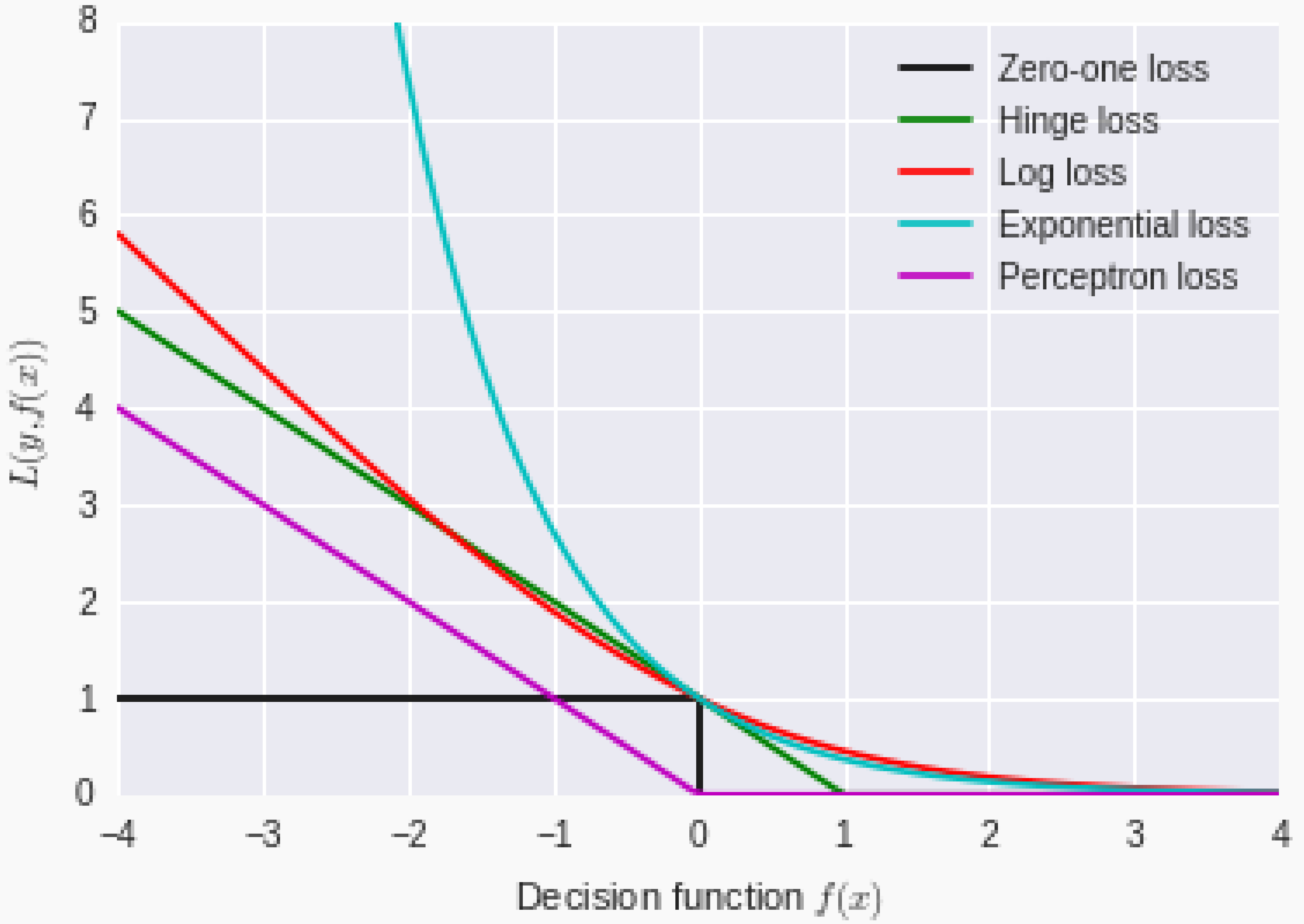
$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + k \sum_{j=1}^{p} \beta_j^2 \right\}$$

$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + k \sum_{j=1}^{p} |\beta_j| \right\}$$

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

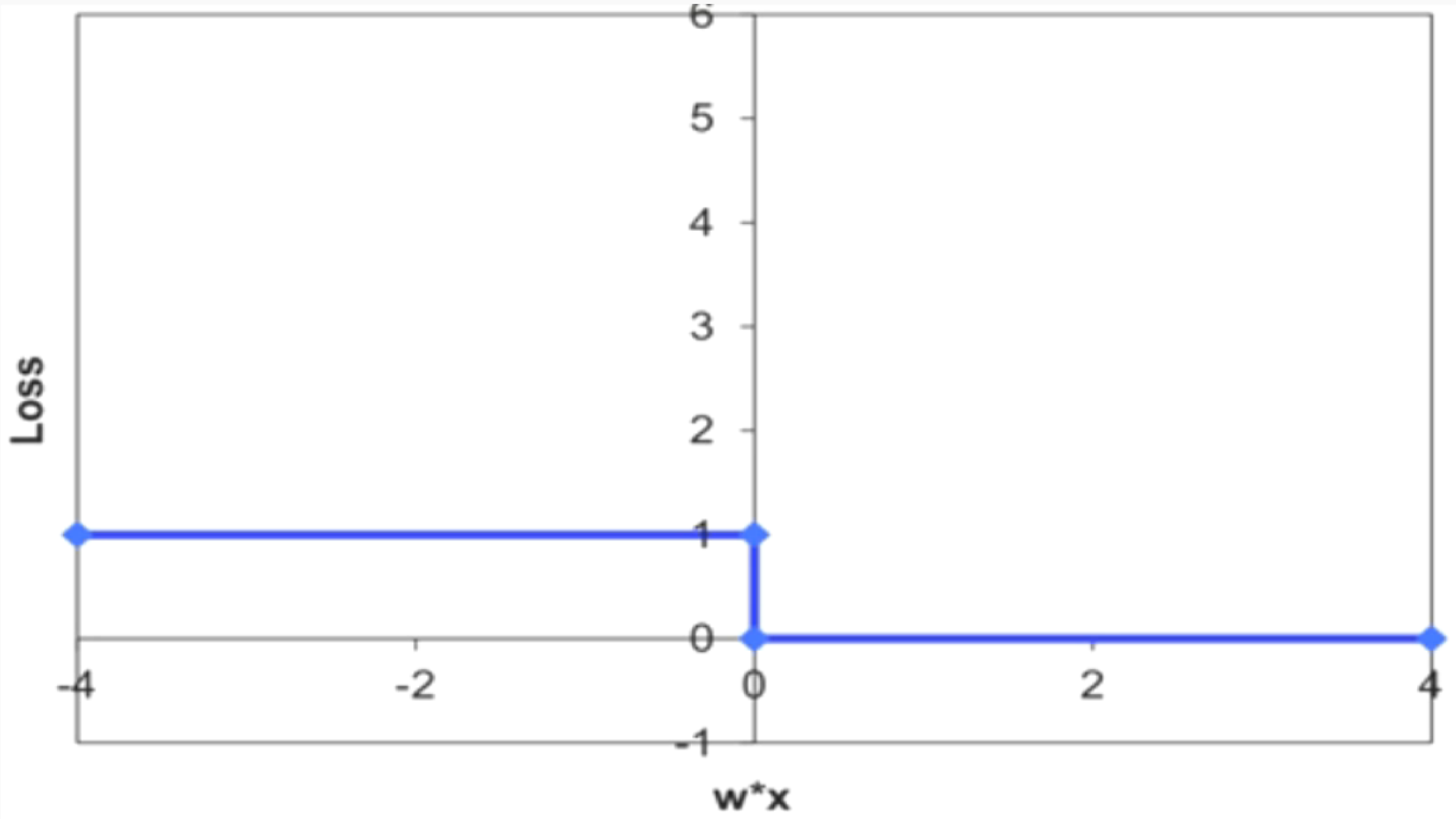
损失函数的类型

- 0-1 Loss
- Hinge Loss：SVM
- Log Loss：LR
- Exp Loss：Boost
- Square Loss



0-1 Loss

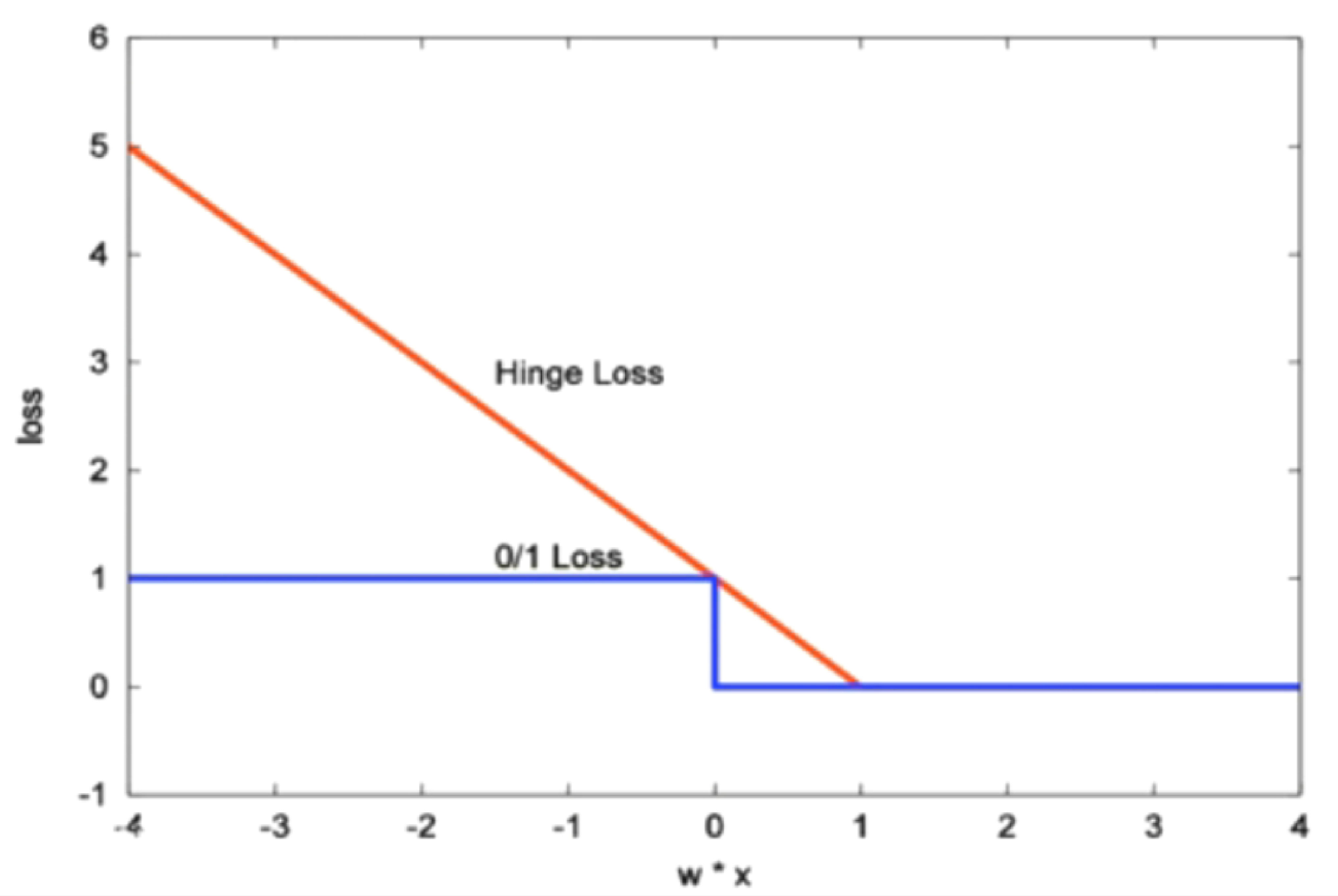
$$L_{01}(m) = \begin{cases} 0 & \text{if } m \geq 0 \\ 1 & \text{if } m < 0 \end{cases}$$



Hinge Loss

$$\ell(y) = \max(0, 1 - t \cdot y)$$

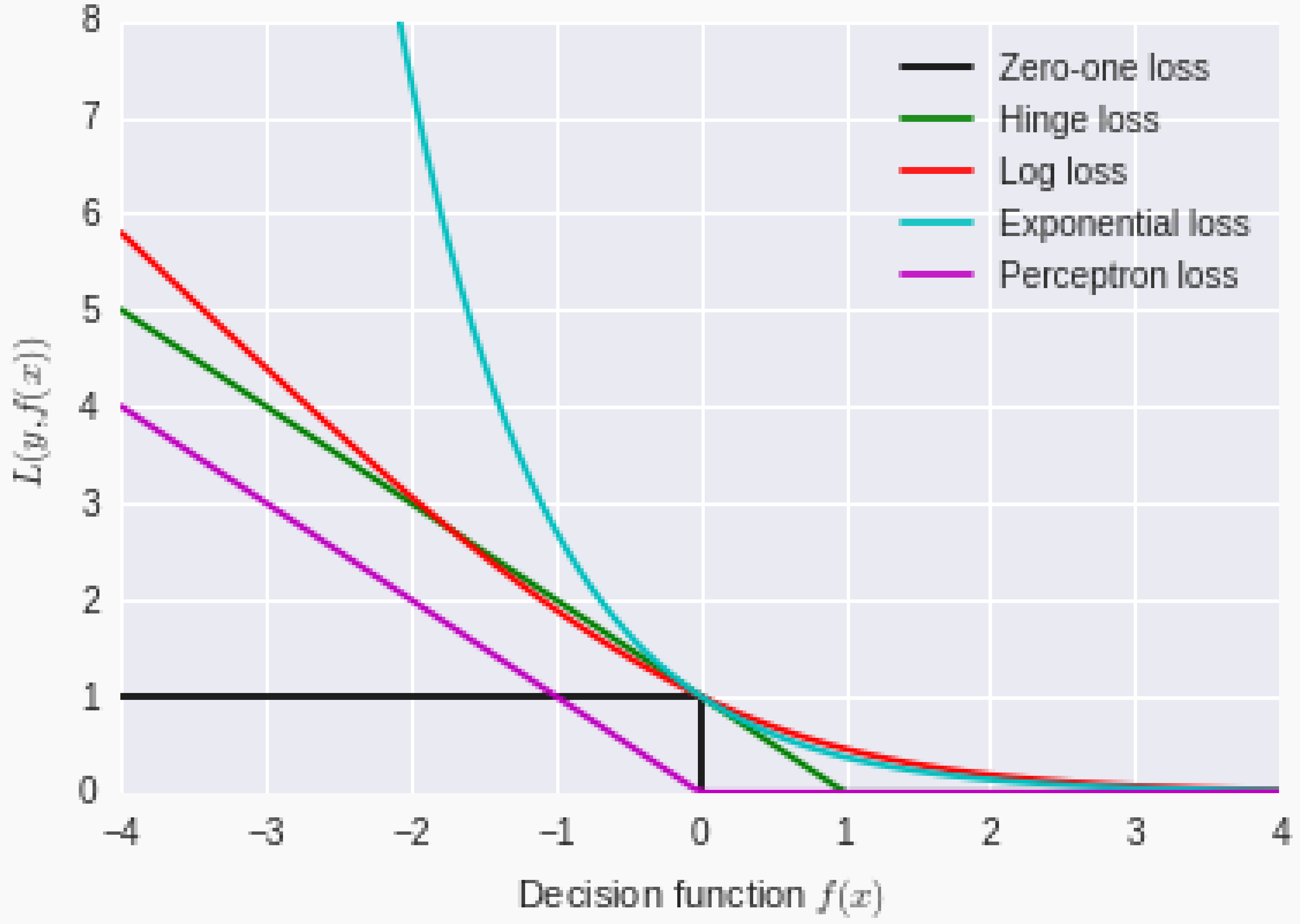
$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^Tx^{(i)} + b) \geq 1 - \xi_i, \quad i = 1,\ldots,m \\ & \xi_i \geq 0, \quad i = 1,\ldots,m. \end{aligned}$$



Log Loss

$$L(\mathbf{w}) \;=\; -\frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) \;=\; -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right],$$

$$\begin{aligned}\ell(\theta) &= \log L(\theta)\\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)})\log(1-h(x^{(i)}))\end{aligned}$$



模型选择和参数选择

从工程的角度看机器学习

- 模型表达
- 模型优化
- 模型的评估

性能评价

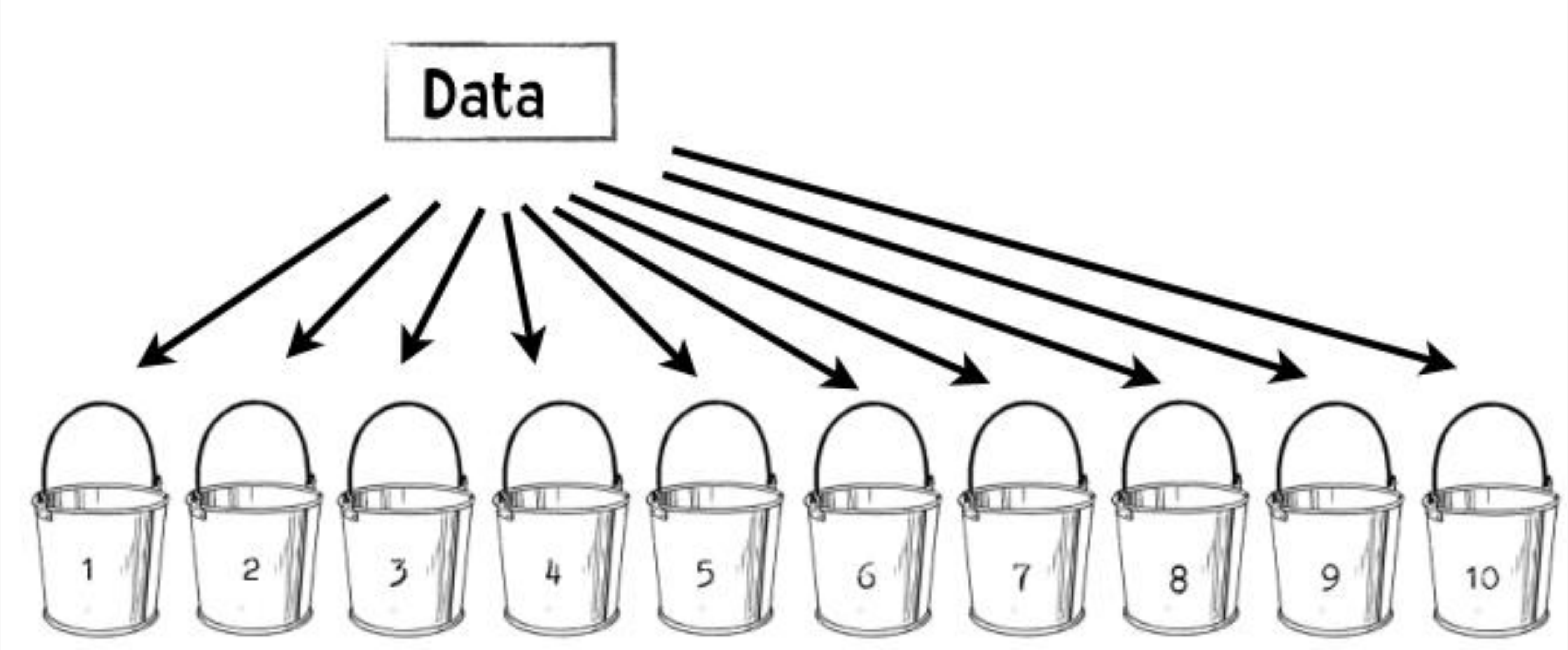
30-70 Test

交叉验证：K-Fold Cross Validation

缺点：结果的不确定性

LOO：

优点：结果确定



性能评估

- 混淆矩阵
- 正例和负例
- 准确率 $= \dfrac{A + D}{A + B + C + D}$

- 灵敏度（Sensitivity, Recall Rate，TP） $\dfrac{A}{A + C}$
- 假阴性（漏诊，FN） $\dfrac{C}{A+C}$
- 特异度 $\dfrac{D}{B+D}$
- 假阳性 （误诊，FP)） $\dfrac{B}{B+D}$

混淆矩阵（Confusion Matrix）

		实际		
		Yes	No	合计
预测	Yes	A	B	A+B
	No	C	D	C+D
	合计	A+C	B+D	

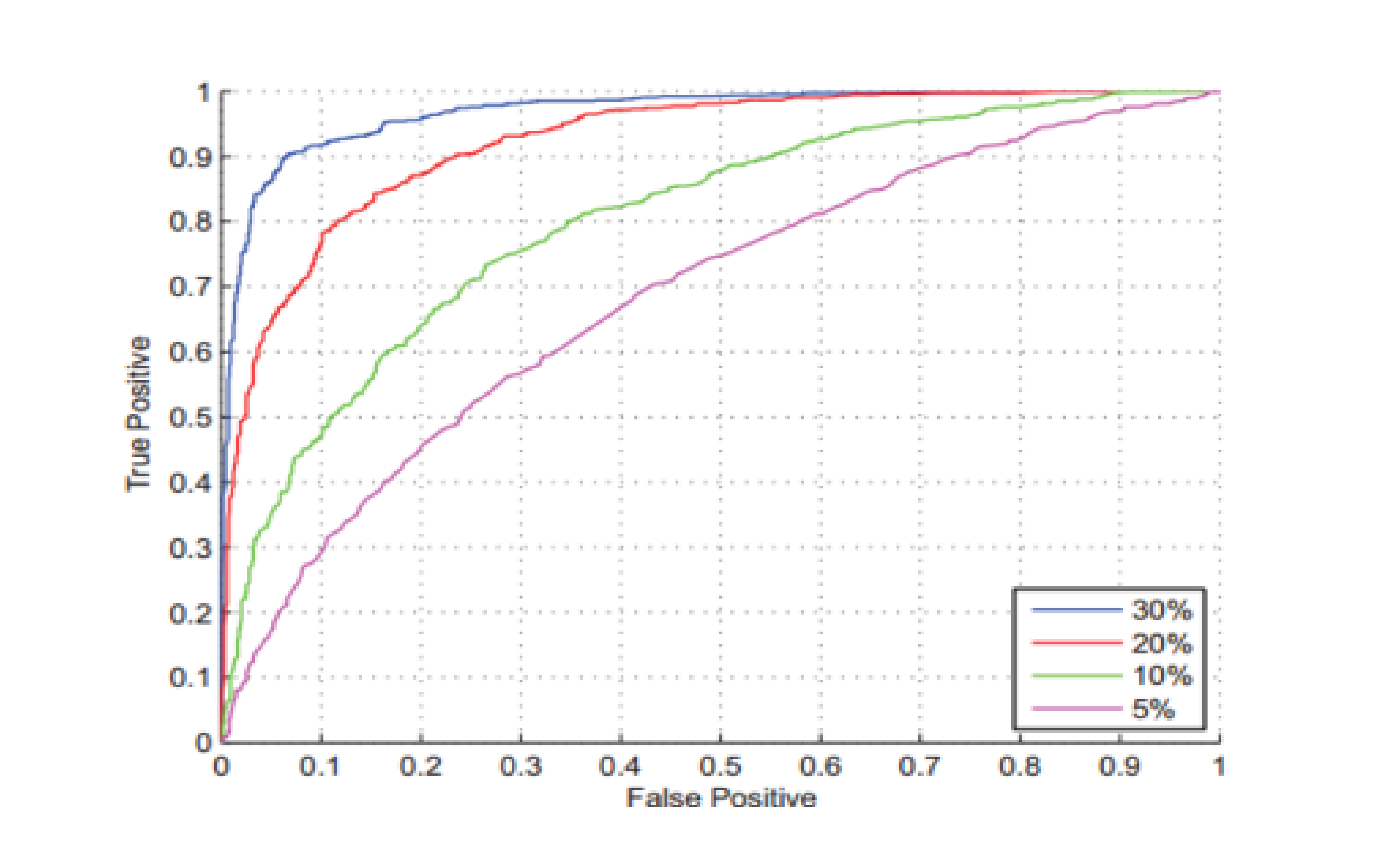
不要过度迷信准确率

HIV筛查测试ELISA的准确率相当高，有些数据表明高达99.9%以上。如果一个人查出来是阳性的话就一定被感染吗？

贝叶斯公式

ROC曲线

- FN代价高于FP，右上角
- FP代价高于FN，左下角
- 二者差不多，左上角



自适应学习率算法

二分线性精确搜索

基于阿米霍步长准则[Armijo-Goldstein]的模糊搜索

- 目标函数值应该有足够的变化(下降或者上升)
- 搜索的步长α不应该太小

$$\frac{f(x_k+\alpha d_k)-f(x_k)}{\alpha}<cf'(x_k)^T d_k$$

机器学习思想

$$f(x_k + \alpha d_k) < f(x_k) + c\alpha f'(x_k)^T d_k$$

Algorithm 3: Backtracking Line Search with Armijo condition

Input: $c_1 \in (0,1)$,$\tau \in (0,1)$,search direction d and step size α_0

Initialize $k = 0$;

while $f(x + \alpha_k d) > f(x) + c_1 \alpha_k f'(x)^T d$ **do**

  update $\alpha_{k+1} = \tau \alpha_k$;

  $k = k + 1$;

return α_k;