

模式识别

第二章：距离分类器

主讲人：张治国

zhiguo Zhang@hit.edu.cn



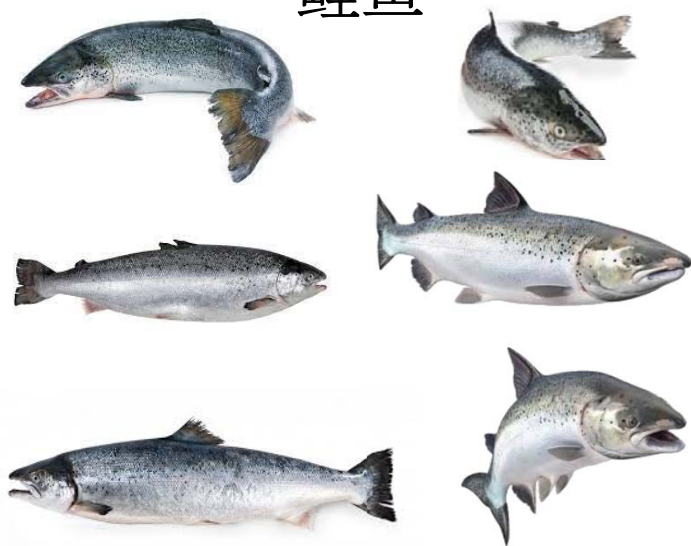
本章内容

- 距离分类器
 - ✓ – 距离分类器的一般形式
 - 模板匹配
 - ✓ – 最近邻分类
 - 最近邻分类的加速
 - ✓ – K-近邻算法
- 距离和相似性度量
 - 距离度量
 - 相似性度量
- 分类器性能评价
 - ✓ – 评价准则
 - 评价指标
 - 评价方法

距离分类器

- 人是如何识别不同对象的？

鲑鱼



海鲈鱼



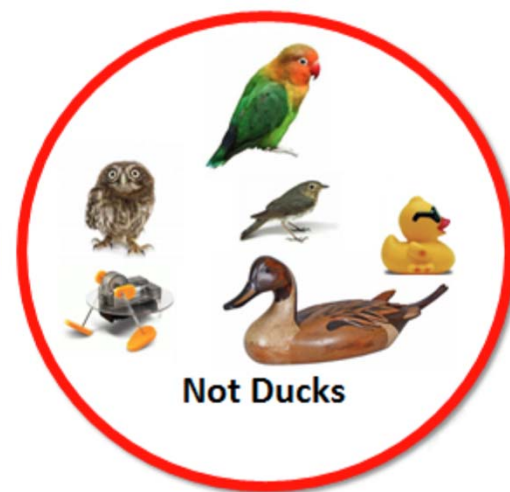
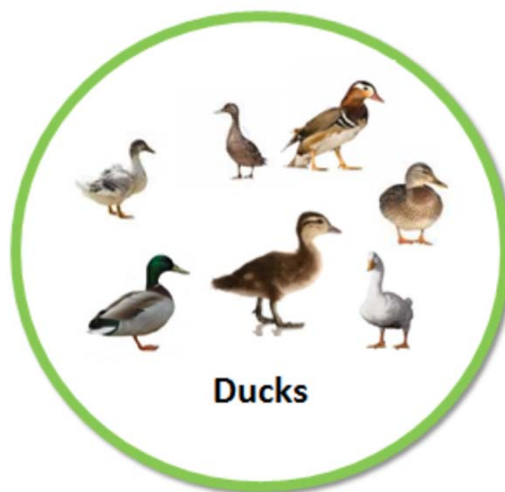
什么鱼？

距离分类器

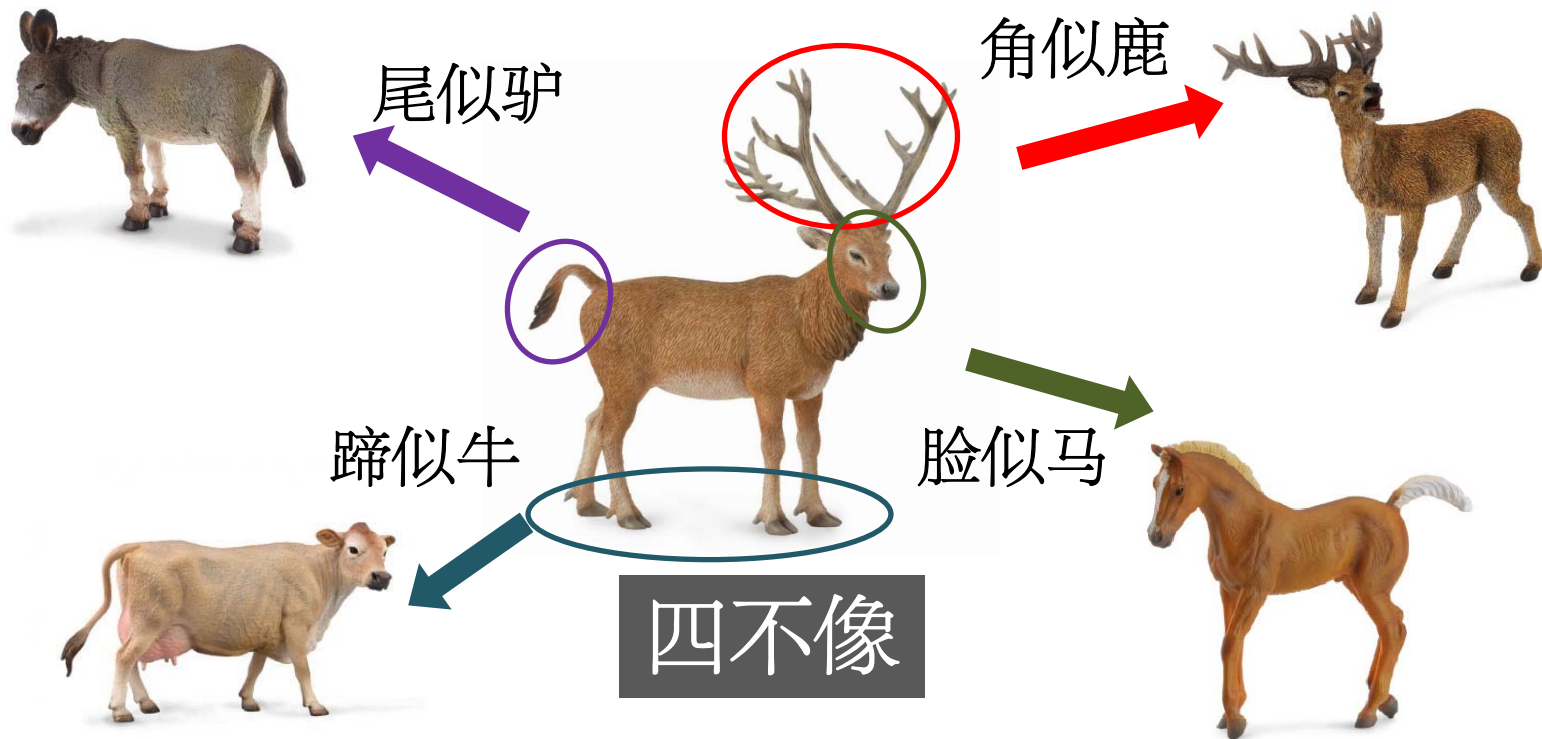
- 识别对象与某个类别**是否相似**是人在做出识别判断时的一个基本依据。
- 因此，可以根据相似性构建用于计算机识别的分类器 - **距离分类器**。

**If it looks like a duck,
quacks like a duck,
and swims like a duck**

It is a duck



距离分类器



四不像属于哪一科?

距离分类器

- 将待识别样本 \mathbf{x} 分类到与其最相似（即：特征空间内距离最接近）的类别中，一般可以通过如下过程实现。

- 输入：需要识别的样本 \mathbf{x} ;
- 计算： \mathbf{x} 与所有类别 ω_i 的相似度 $s(\mathbf{x}, \omega_i), i = 1, \dots, c$;
- 输出：相似度最大的类别 $\omega_j, j = \arg \max_{1 \leq i \leq c} s(\mathbf{x}, \omega_i)$ 。

关键问题：如何度量样本与类别的相似度（距离）？

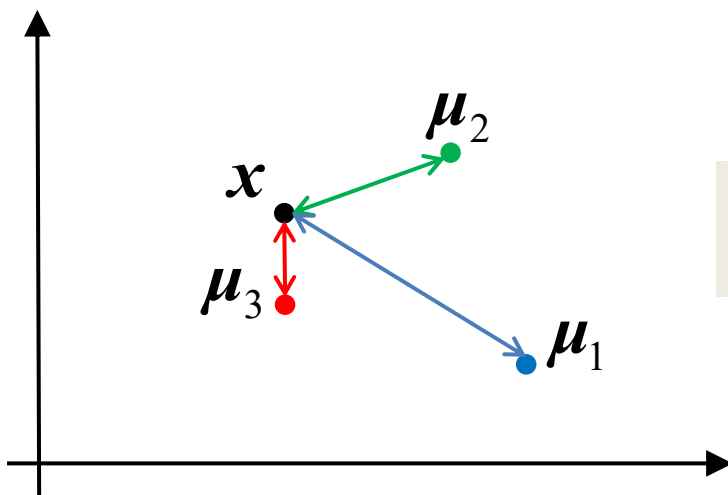
模板匹配

- **模板**：每个类别的代表样本 μ_i 。

思考：如何从多个样本中确定模板？

- 待识别样本 \mathbf{x} 和模板间的相似程度可作为样本与类别间的相似程度的度量 $s(\mathbf{x}, \omega_i) = s(\mathbf{x}, \mu_i)$ 。
- 特征空间上，两点距离 d 越大，相似度 s 越低。

$$s(\mathbf{x}, \mu_i) = -d(\mathbf{x}, \mu_i)$$



左图显示的距离是欧式距离

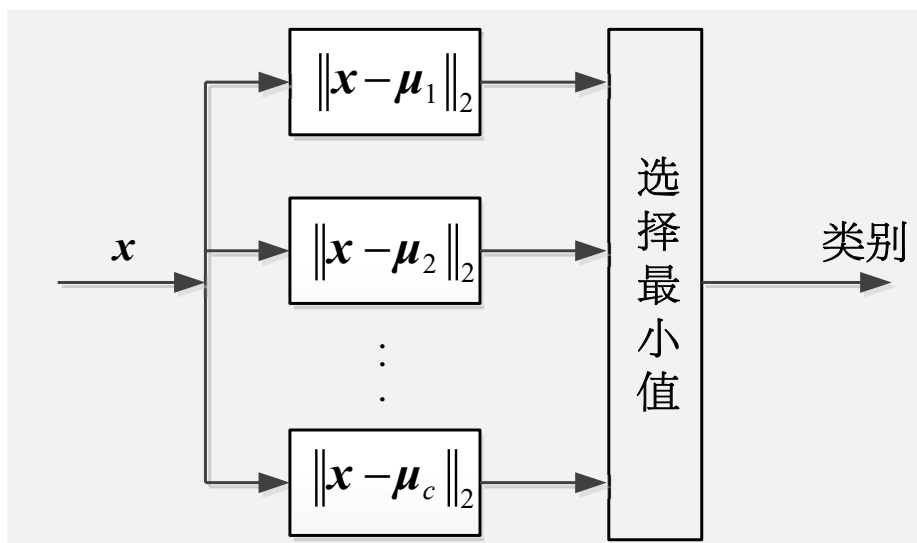
模板匹配

- 待识别样本 \mathbf{x} 和模板 μ_i 的欧氏距离计算为：

$$d(\mathbf{x}, \mu_i) = \|\mathbf{x} - \mu_i\|_2 = \sqrt{\sum_{m=1}^M (x_m - \mu_m^{(i)})^2}$$

其中 M 是特征维度, $\|\cdot\|_2$ 是矢量间的 l_2 范数。

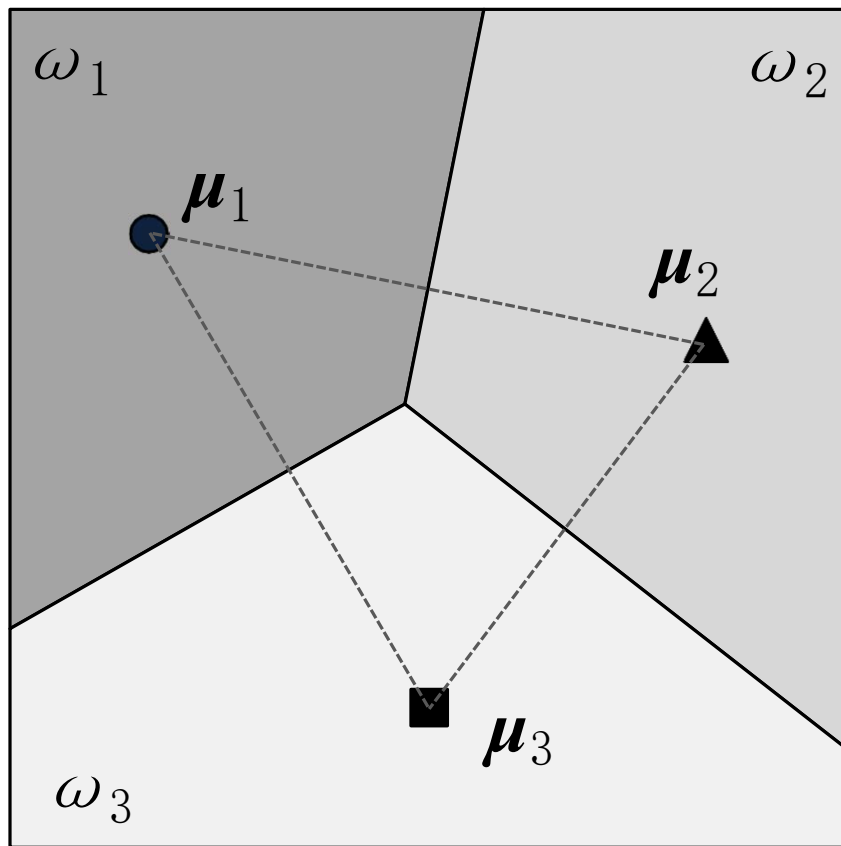
- 判别：如果 $j = \arg \min_{1 \leq i \leq c} d(\mathbf{x}, \mu_i)$, 则 $\mathbf{x} \in \omega_j$ 。



$$\begin{aligned} l_p(\mathbf{x}) &= \|\mathbf{x}\|_p \\ &= \left(\sum_{m=1}^M |x_m|^p \right)^{1/p} \end{aligned}$$

模板匹配

- 由一组连接相邻两个模板直线的垂直平分线组成 c 个区域。
- 两个区域的交界称为“判别界面”。
- 判别界面在二维空间是直线，在三维空间是平面，在高维空间是“超平面”。



最近邻分类

- 一般的模式识别问题中，每一个类别 ω_i 有多个训练样本 $D_i = \{\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_{n_i}^{(i)}\}$ ， $i = 1, \dots, c$ ， n_i 是第 i 个类别中的训练样本数。
- 样本 \mathbf{x} 与类别 ω_i 的相似度可以根据用 \mathbf{x} 与 D_i 中最近样本的距离来度量：

$$s(\mathbf{x}, \omega_i) = \max_{y \in D_i} s(\mathbf{x}, y) = \min_{y \in D_i} d(\mathbf{x}, y)$$

最近邻分类

- 最近邻分类算法：

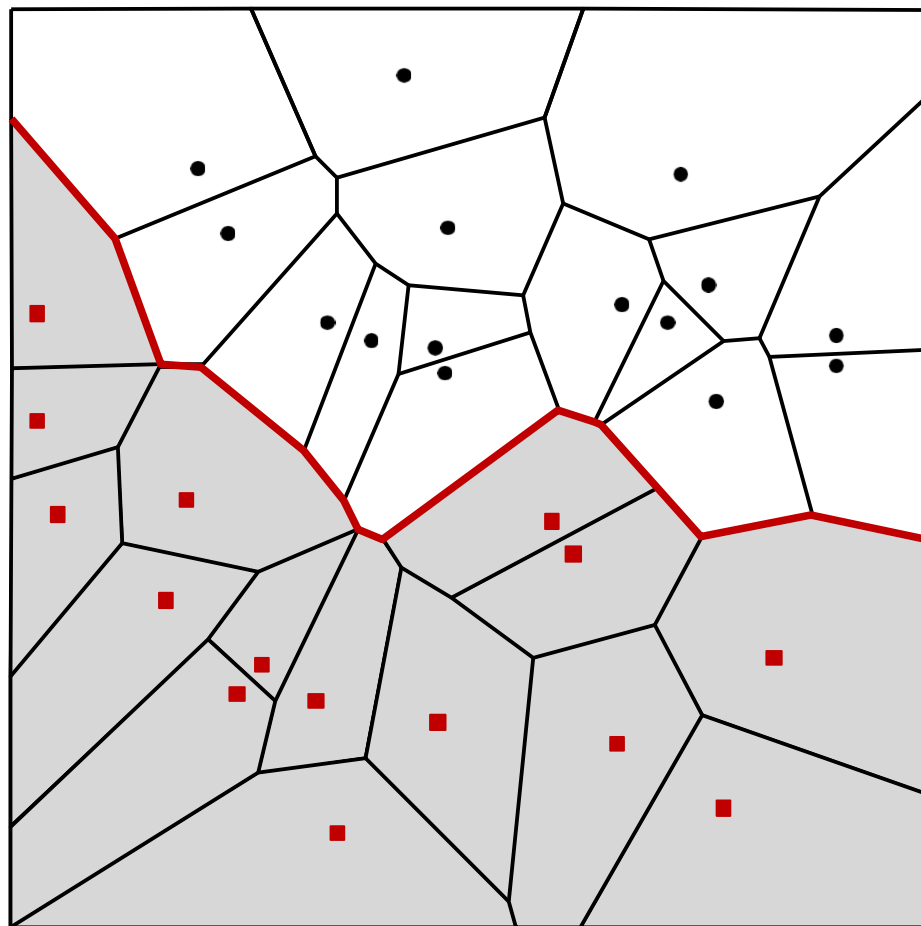
- 输入：需要识别的样本 \mathbf{x} ，训练样本集 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ ；
 - 寻找 D 中与 \mathbf{x} 最近的样本 $\mathbf{y} = \arg \min_{\mathbf{x}_i \in D} d(\mathbf{x}, \mathbf{x}_i)$ ；
 - 输出： \mathbf{y} 所属的类别。
-

其中训练样本集 $D = \bigcup_{i=1}^c D_i$ 包含了所有类别的训练样本，

$n = \sum_{i=1}^c n_i$ 是全部样本的数量。

最近邻分类

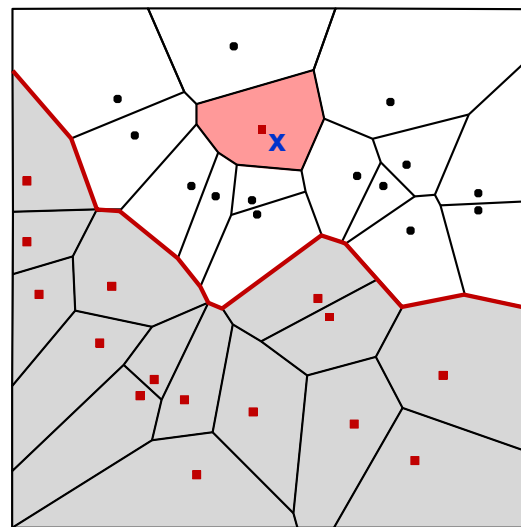
- **Voronoi网格**：由一组连接两邻点直线的垂直平分线组成的连续多边形组成。
- 最近邻分类时，如果待识别样本 x 出现在某个单元格里，那么 x 属于该类，距离 x 最近的样本 y 就是该单元格的训练样本。



- 圆点和方点是两类样本；
- 红色粗折线是分类界面。

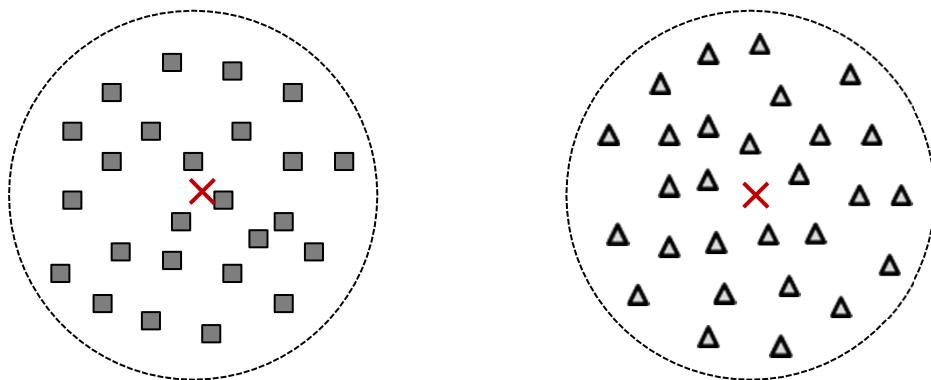
最近邻分类的特点

- 训练样本数量较多时效果良好。
- 计算量大：
 - 每次识别时需要同所有训练样本计算距离。
- 占用存储空间大：
 - 需要保存所有的训练样本。
- 易受样本噪声影响：
 - 只依赖最近的训练样本，当训练样本某些特征有偏差或者标注错误时易导致分类错误。



最近邻分类的加速

- 为加速计，可将最近邻分类转化为单模板匹配，用每个类别的训练样本学习出一个模板。
- 问题：如何学习出来一个最有代表性的模板？
- 思路：选择距离训练样本都比较近的点。



最近邻分类的加速

- 因此，模板可以在整个 M 维欧式空间中通过一个优化问题求解：

$$\mu_i = \arg \min_{\mu \in R^M} \sum_{k=1}^{n_i} d(\mathbf{x}_k^{(i)}, \mu)$$

- 选择欧式距离，我们构造下面的误差平方和准则函数

$$J_i(\mu) = \sum_{k=1}^{n_i} \left\| \mathbf{x}_k^{(i)} - \mu \right\|_2^2 = \sum_{k=1}^{n_i} (\mathbf{x}_k^{(i)} - \mu)^T (\mathbf{x}_k^{(i)} - \mu)$$

误差（欧式距离）平方和

- 优化求解：
$$\mu_i = \arg \min_{\mu \in R^M} J_i(\mu)$$

最近邻分类的加速

- $J_i(\boldsymbol{\mu})$ 的极值点是其使其梯度等于零的矢量:

$$\nabla J_i(\boldsymbol{\mu}) = \frac{\partial J_i(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \sum_{k=1}^{n_i} 2(\mathbf{x}_k^{(i)} - \boldsymbol{\mu})(-1) = 2n_i\boldsymbol{\mu} - 2\sum_{k=1}^{n_i} \mathbf{x}_k^{(i)} = 0$$

- 求解得出极值点（模板）即是第 i 类训练样本的均值:

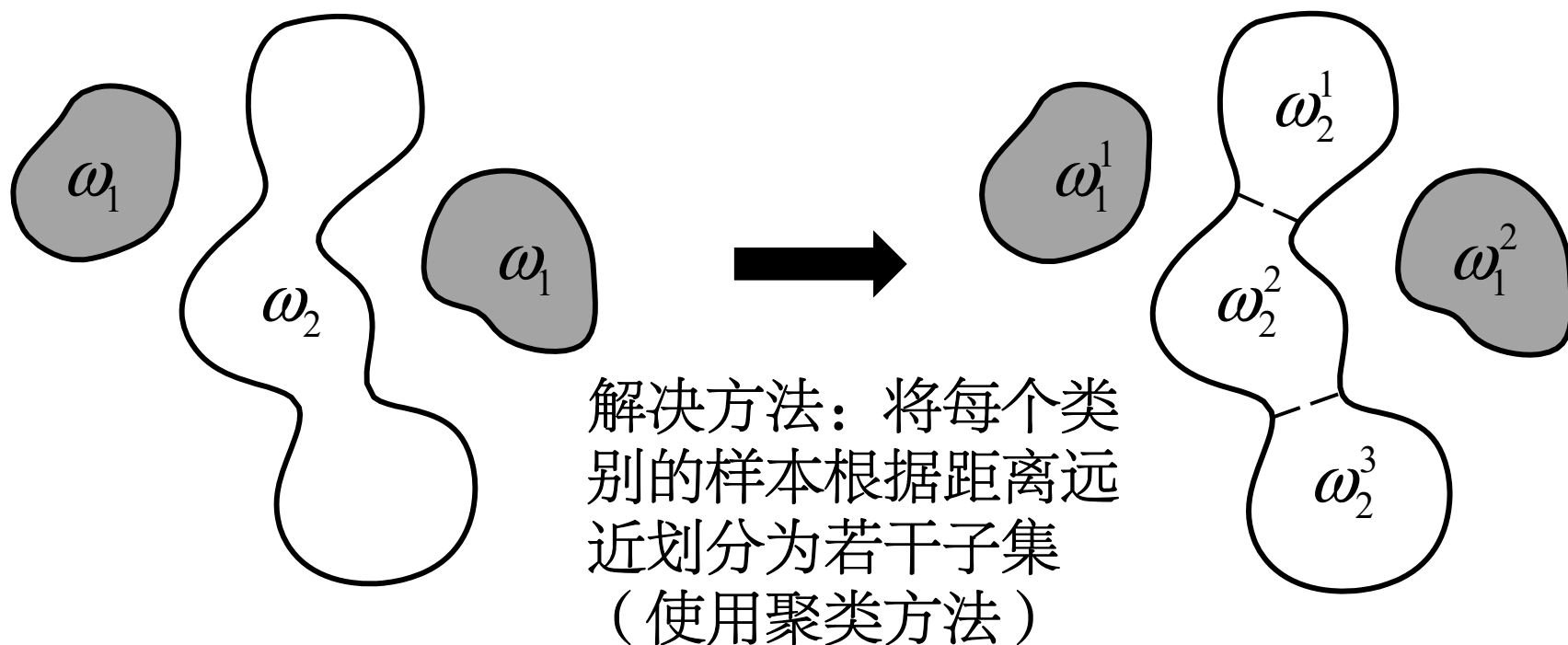
$$\boldsymbol{\mu} = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathbf{x}_k^{(i)}$$

思考：模板可以有其他形式吗？
如何求得？

- 单模板匹配学习过程：计算每个类别样本的均值作为模板，使用“模板匹配”得到结果。

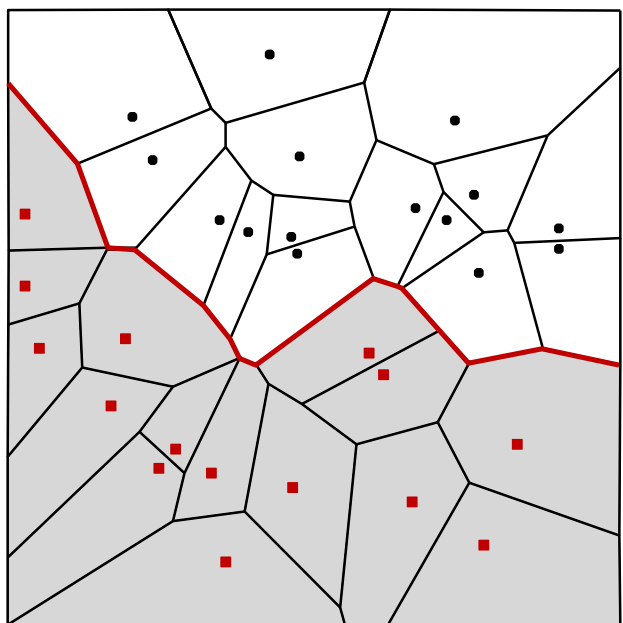
最近邻分类的加速

- 单模板匹配对样本分布有要求（分布接近球形，区域大小类似等），并不适合很多其他情况。为什么？

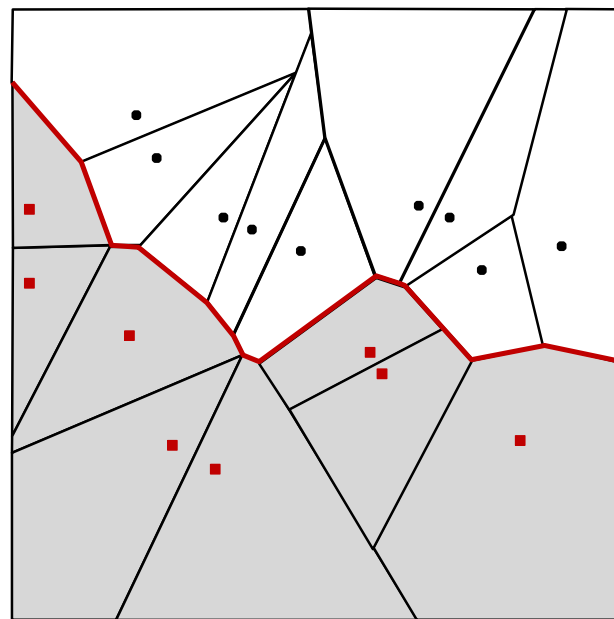


最近邻分类的加速

- 使用模板能够提高计算效率（减少匹配次数），但不能保证准确率（改变了分类界面形状）。
- **近邻剪辑**：去掉对分类面“无用”的样本点，不改变分类界面的形状。

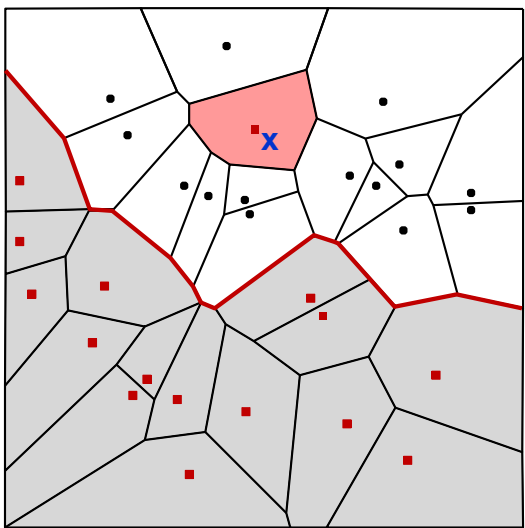


如果某样本的
网格和所有相
邻网格属于同
一类，则删除
该样本。



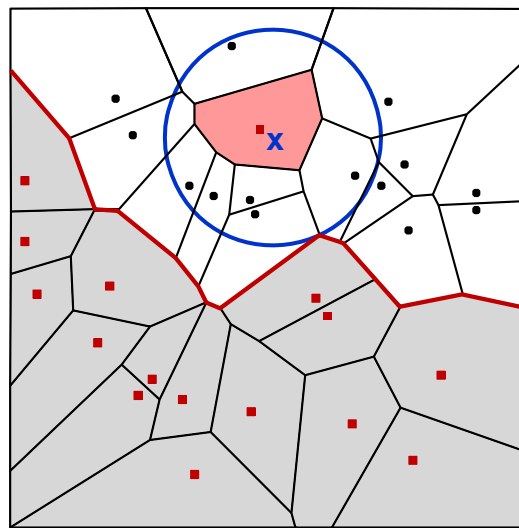
K-近邻算法

- 最近邻：
由距离最近的一个样本
的类别决定



一个噪声样本导致一片错误分类区域。

- K-近邻：
由距离最近的K个样本
的类别投票决定



例：寻找与样本最近的 $K=7$ 个样本，投票6:1，正确识别

K-近邻算法

- K-近邻算法的判别规则可以表示为：

如果 $j = \arg \max_{1 \leq i \leq c} k_i$, 则判别 $\mathbf{x} \in \omega_j$,

k_i 是与 \mathbf{x} 距离最近的 K 个样本中属于 ω_i 的样本数。

- 最近邻算法是K-近邻的特例（ $K = 1$ ）。

K-近邻算法

- K-近邻分类算法:

-
- 输入: 需要识别的样本 \mathbf{x} , 训练样本集 $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, 和参数 K ;
 - 计算 \mathbf{x} 与 D 中每个样本的距离;
 - 寻找与 \mathbf{x} 距离最近的前 K 个样本, 统计其中属于各个类别的样本数 $k_i, i = 1, \dots, c$;
 - 输出: $j = \arg \max_{1 \leq i \leq c} k_i$ 。
-

K-近邻算法

- K的选择：
 - K值选择过小，算法的性能接近于最近邻分类；
 - K值选择过大，距离较远的样本也会对分类结果产生作用，这样也会引起分类误差；
 - 适合的K值需要根据具体问题来确定。
- K-近邻的缺陷：
 - 非平衡样本：某一类样本数量很大，而其它类样本的数量相对较少，样本多的类别总是占优势。
 - 计算量：需要与每个训练样本计算距离，复杂度与最近邻算法类似。解决方法：K-D树（略）。

距离和相似性度量

- 样本间的距离可以描述它们之间的相似度。
- 欧式距离度量两点间直线长度。但特征间的距离仍有很多种度量方式，如何选择？如何定义？



距离度量

- 对于任意一个定义在两个矢量 \mathbf{x} 和 \mathbf{y} 上的函数 $d(\mathbf{x}, \mathbf{y})$ ，只要满足如下4个性质就可以称作“距离度量”：

1. 非负性： $d(\mathbf{x}, \mathbf{y}) \geq 0$

2. 对称性： $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

3. 自反性： $d(\mathbf{x}, \mathbf{y}) = 0$ ，当且仅当 $\mathbf{x} = \mathbf{y}$

4. 三角不等式： $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$

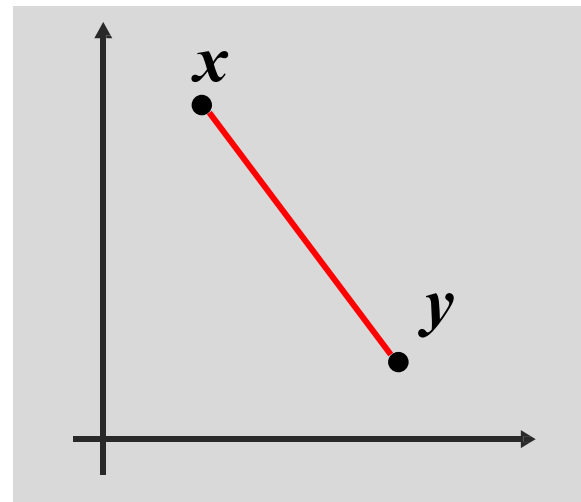
欧几里得距离

- 欧几里得距离/欧氏距离 (Euclidean Distance)

$$d(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^M (x_i - y_i)^2 \right]^{1/2}$$

- 欧氏距离是特征空间中两点的直线距离。
- 欧氏距离对应矢量的 l_2 范数：

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$$



曼哈顿距离

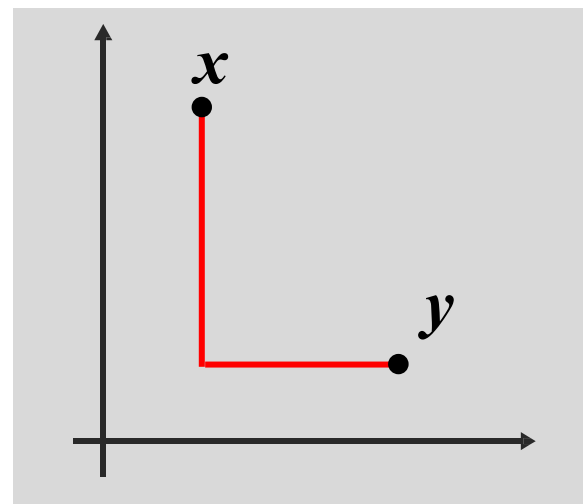
- 曼哈顿距离/街市距离 (Manhattan Distance)

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^M |x_i - y_i|$$

- 曼哈顿距离是特征空间中两点坐标之差的绝对值的和。

- 曼哈顿距离对应矢量的 l_1 范数:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1$$



切比雪夫距离

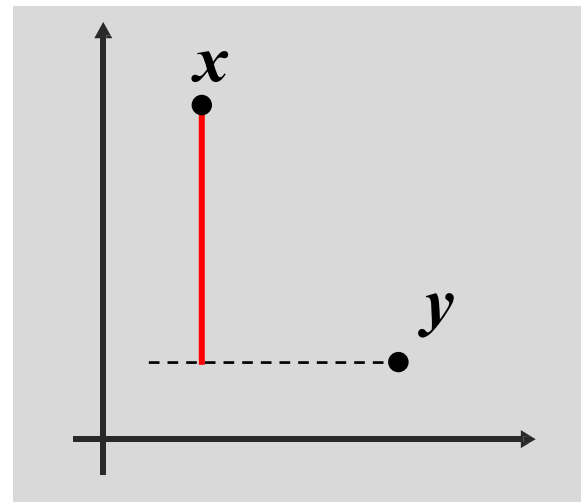
- 切比雪夫距离 (Chebyshev Distance)

$$d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq M} |x_i - y_i|$$

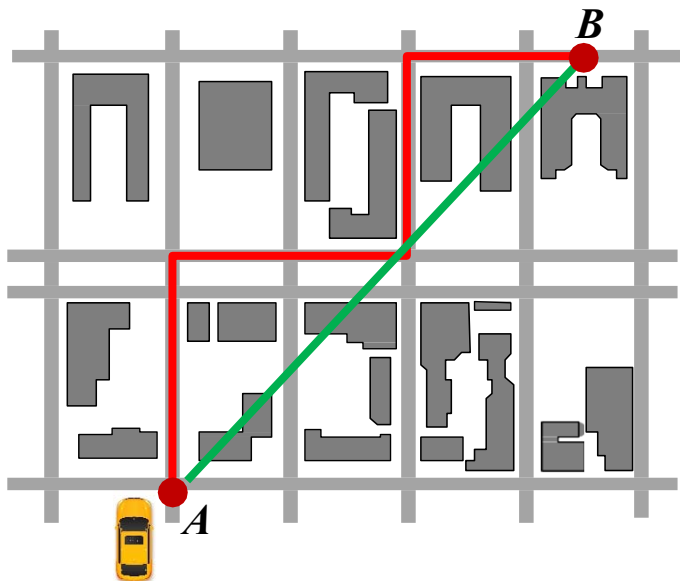
- 切比雪夫距离是各坐标数值差绝对值的最大值。
- 切比雪夫距离对应矢量的 l_∞ 范数：

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_\infty$$

$$l_p(\mathbf{x}) = \left(\sum_{m=1}^M |x_m|^p \right)^{1/p}$$

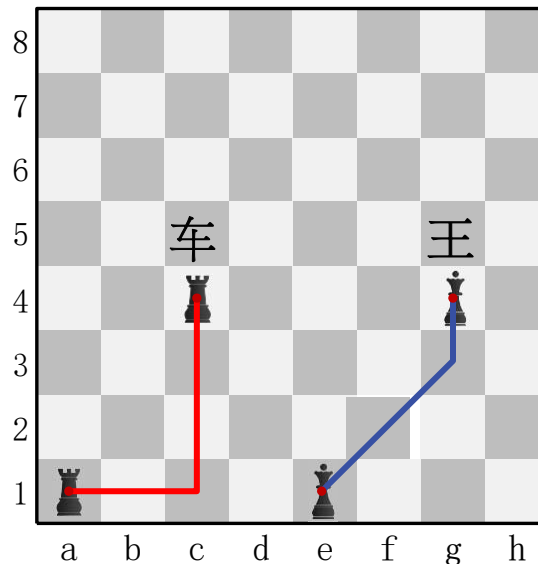


各类距离实例



街区中两点间的距离

- 欧氏距离（绿线）：
直线距离
- 曼哈顿距离（红线）：
汽车行驶的距离



王可横、直、斜走一步；车可直线走任意步。

国际象棋棋子移动格数

- 曼哈顿距离（红线）：
车移动经过的格数
- 切比雪夫距离（蓝线）：
王移动经过的格数

闵可夫斯基距离

- 闵可夫斯基距离 (Minkowski Distance)

$$d(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^M |x_i - y_i|^p \right]^{1/p}$$

- 闵可夫斯基距离对应矢量的 l_p 范数:

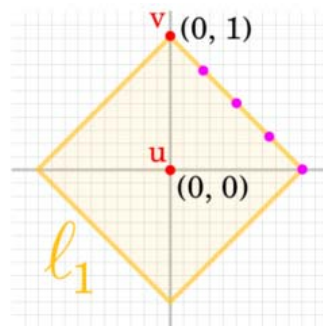
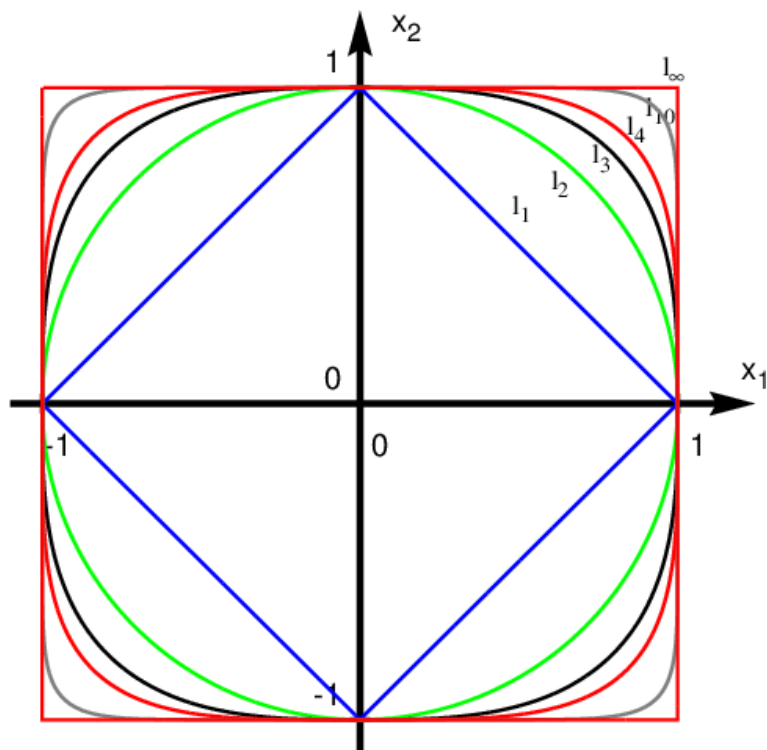
$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p$$

$$l_p(\mathbf{x}) = \left(\sum_{m=1}^M |x_m|^p \right)^{1/p}$$

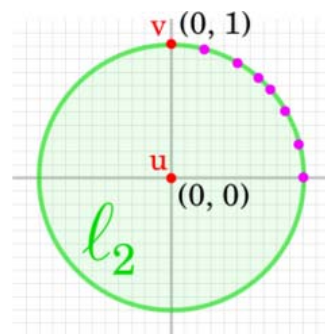
$p = 1$: 曼哈顿距离
 $p = 2$: 欧氏距离
 $p = \infty$: 切比雪夫距离

闵可夫斯基距离

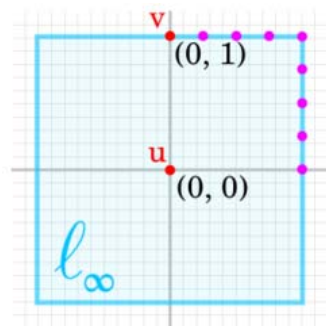
- 不同距离度量下的单位“圆”



曼哈顿距离
度量下的
单位“圆”



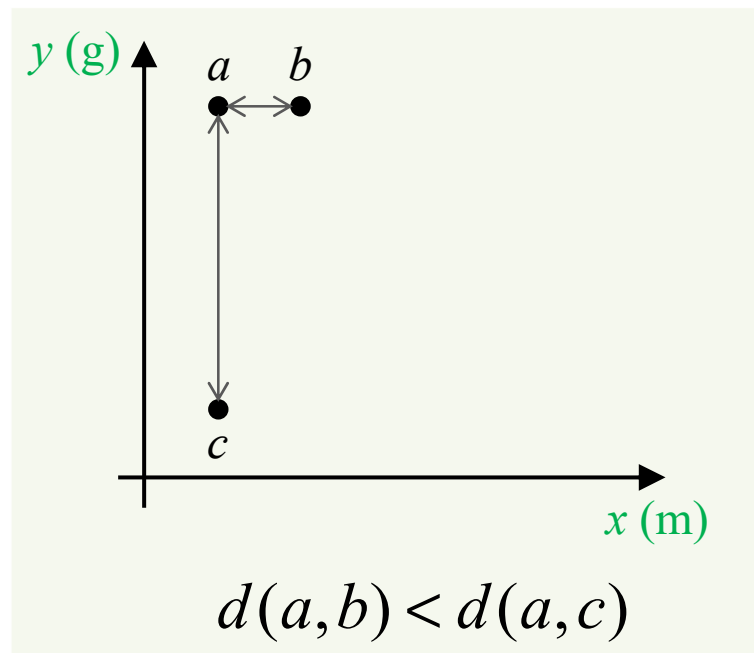
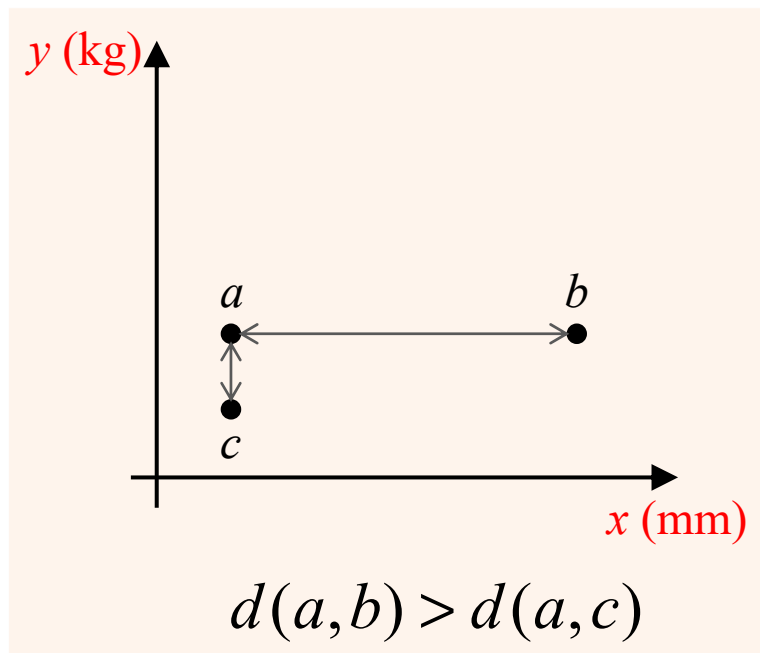
欧式距离
度量下的
单位“圆”



切比雪夫距离
度量下的
单位“圆”

样本规格化

- 不同特征有不同的取值范围，取值范围会极大影响距离。
- 例：改变特征量纲对距离的影响。



样本规格化

- **样本规格化**：使样本的每一维特征都分布在相同的范围内，计算距离度量时每一维特征上的差异都会得到相同的体现。
- **方法1. 均匀缩放**：假设每一维特征都服从均匀分布，将每一维特征平移和缩放到 $[0, 1]$ 内。
- **方法2. 高斯缩放**：假设每一维特征都符合高斯分布，将每一维特征平移和缩放为标准高斯分布。

样本规格化

- **方法1. 均匀缩放：**假设每一维特征都服从均匀分布，将每一维特征平移和缩放到 $[0, 1]$ 内。

1. 计算样本集每一维特征的最大、最小值

$$x_{j_{\min}} = \min_{1 \leq i \leq n} x_{ij}, \quad x_{j_{\max}} = \max_{1 \leq i \leq n} x_{ij}, \quad j = 1, \dots, M$$

2. 平移和缩放样本的每一维特征：

$$x'_{ij} = \frac{x_{ij} - x_{j_{\min}}}{x_{j_{\max}} - x_{j_{\min}}}, \quad i = 1, \dots, n, \quad j = 1, \dots, M$$

其中 x_{ij} 和 x'_{ij} 是规格化前和规格化后的第 i 个样本的第 j 维特征。

样本规格化

- 方法2. 高斯缩放：假设每一维特征都符合高斯分布，将每一维特征平移和缩放为标准高斯分布。

1. 计算每一维特征的均值和标准差：

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \quad s_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)^2}, \quad j = 1, \dots, M$$

2. 规格化（高斯化）每一维特征：

$$x'_{ij} = \frac{x_{ij} - \mu_j}{s_j}, \quad i = 1, \dots, n, \quad j = 1, \dots, M$$

马氏距离

- 马氏距离（Mahalanobis distance）：考虑特征之间的相关性，基于协方差对距离进行规格化，解决各个维度尺度不一致且相关的问题。

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T C^{-1} (\mathbf{x} - \mathbf{y})}$$

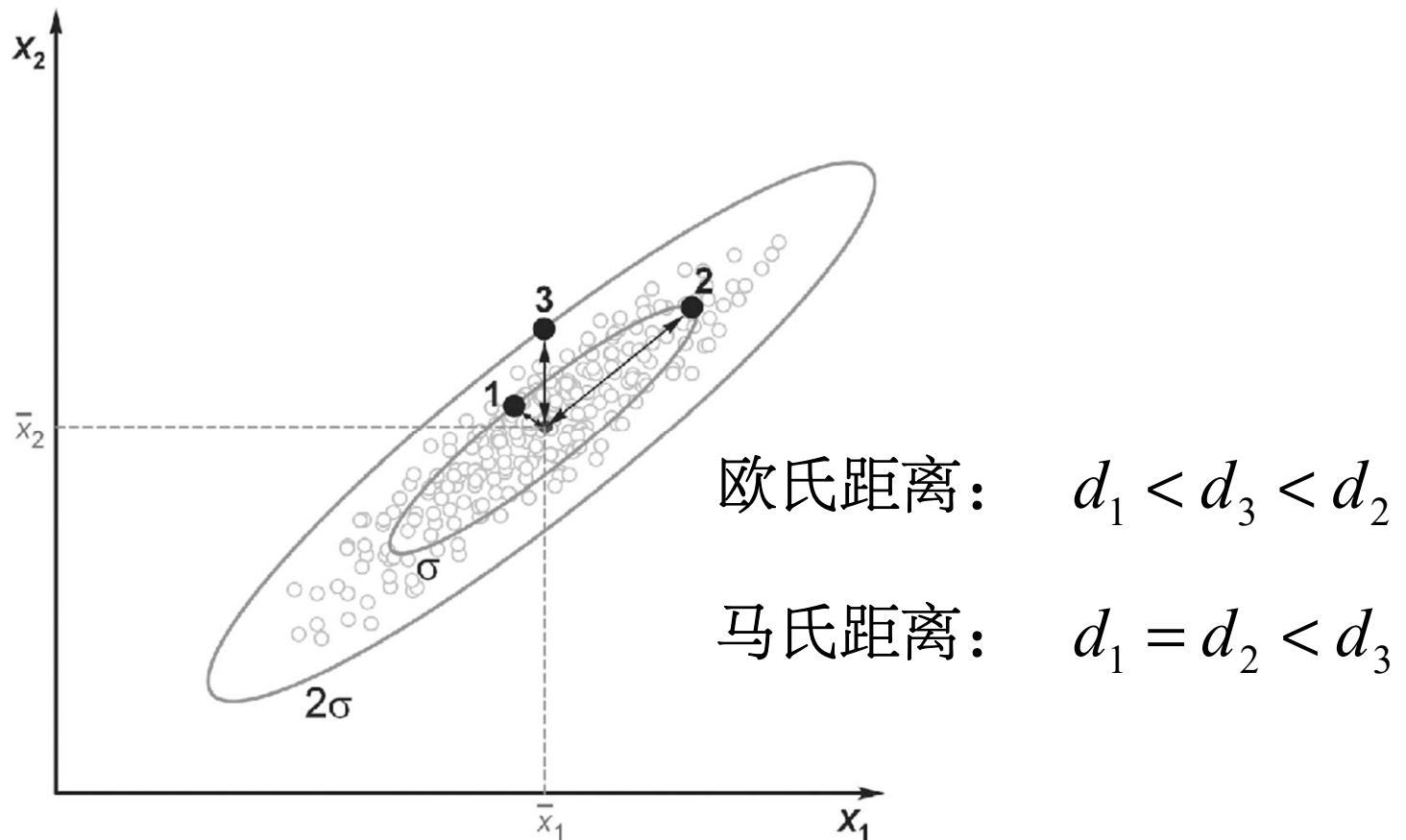
$$\text{其中 } C = \frac{1}{n-1} \sum_{i=1}^n \left(\mathbf{x}_i - \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \right) \right) \left(\mathbf{y}_i - \left(\frac{1}{n} \sum_{i=1}^n \mathbf{y}_i \right) \right)^T$$

是多维特征的协方差矩阵。

- 如各维特征独立同分布，马氏距离即欧氏距离。

马氏距离

- 马氏距离具有坐标系比例、旋转、平移不变性，并且从统计意义上去除了特征间的相关性。



加权距离

- 计算距离时可为不同特征引入不同权重，以克服量纲尺度的影响，或体现不同特征重要性（重要的特征权重高，无用特征权重为0）。
- 样本规格化，可以看做加权距离的特例。

加权欧氏距离：

$$d(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^M w_i (x_i - y_i)^2 \right]^{1/2}, w_i \geq 0$$

均匀缩放的权重：

$$w_j = \frac{1}{(x_{j_{\max}} - x_{j_{\min}})^2}$$

高斯缩放的权重：

$$w_j = \frac{1}{s_j^2}$$

汉明距离

- 汉明距离（Hamming Distance）：度量二值矢量（每个元素只取0或1， $\mathbf{x}, \mathbf{y} \in \{0,1\}^M$ ）的距离。

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^M (x_i - y_i)^2$$

- 事实上，汉明距离计算两个矢量对应位置元素不同的数量。

- 例： $\mathbf{x} = (1, 1, 0, 0, 1, 1, 1)^T$

$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 1)^T$$

\mathbf{x} 和 \mathbf{y} 的汉明距离为3。

相似性度量

- 衡量相似度不一定需要距离，在某些情况下可以选择更直接的方法衡量相似度
 - 角度相似性（两向量的夹角）
 - 相关系数
- 相似性度量随着样本间相似程度的增加而增大，距离则是随着相似程度的增加而减小。为了保持一致性可以将相似度和距离进行转换：

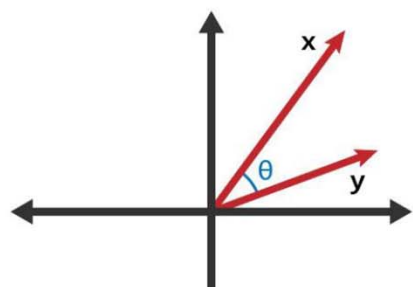
$$d(\mathbf{x}, \mathbf{y}) = 1 - s(\mathbf{x}, \mathbf{y})$$

角度相似性

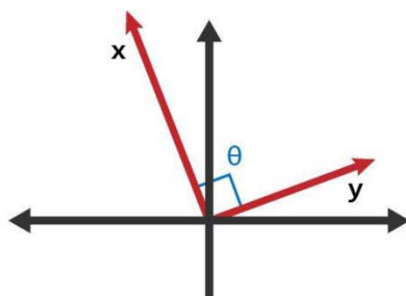
- 角度相似性：当两个样本之间的相似程度只与它们之间的夹角有关、与矢量的长度无关时，可以使用矢量夹角的余弦来度量相似性。

$$s(\mathbf{x}, \mathbf{y}) = \cos \theta_{xy} = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^M x_i y_i}{\sqrt{\sum_{i=1}^M x_i^2} \cdot \sqrt{\sum_{i=1}^M y_i^2}}$$

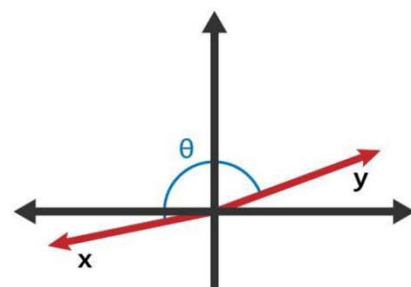
将矢量归一为单位矢量后做内积。



相似



无关（正交）



相反

相关系数

- 相关系数：数据中心化（移除均值）后矢量夹角的余弦。
- 数据中心化方式1：认为矢量 \mathbf{x} 和 \mathbf{y} 分别来自于两个样本集，样本集的均值分别为 μ_x 和 μ_y ，相关系数定义为：

$$s(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x)^T (\mathbf{y} - \mu_y)}{\|\mathbf{x} - \mu_x\| \cdot \|\mathbf{y} - \mu_y\|} = \frac{\sum_{i=1}^M (x_i - \mu_{x_i})(y_i - \mu_{y_i})}{\sqrt{\sum_{i=1}^M (x_i - \mu_{x_i})^2} \cdot \sqrt{\sum_{i=1}^M (y_i - \mu_{y_i})^2}}$$

相关系数

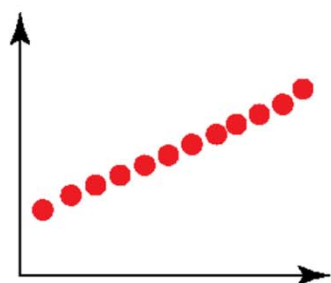
- 数据中心化方式2：将 \mathbf{x} 和 \mathbf{y} 视为一维信号，数据中心化相对于每个矢量特征均值进行：

$$\mu_x = \frac{1}{M} \sum_{i=1}^M x_i, \mu_y = \frac{1}{M} \sum_{i=1}^M y_i$$

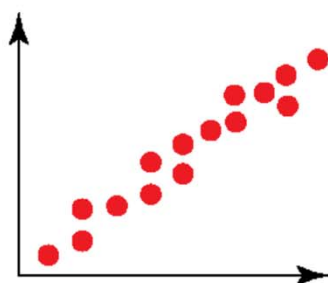
设 \mathbf{e} 是所有元素均为1的 M 维矢量，则

$$s(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x \mathbf{e})^T (\mathbf{y} - \mu_y \mathbf{e})}{\|\mathbf{x} - \mu_x \mathbf{e}\| \cdot \|\mathbf{y} - \mu_y \mathbf{e}\|} = \frac{\sum_{i=1}^M (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^M (x_i - \mu_x)^2} \cdot \sqrt{\sum_{i=1}^M (y_i - \mu_y)^2}}$$

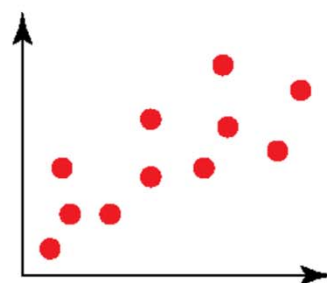
相关系数



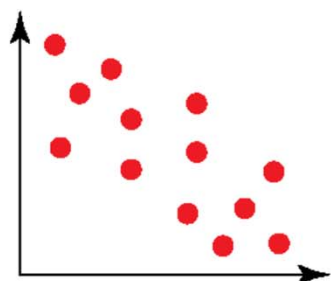
Perfect
Positive
Correlation



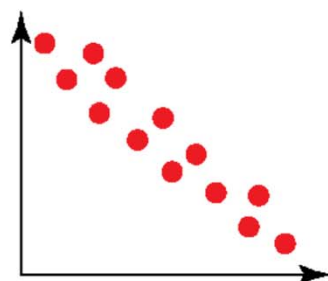
Strong
Positive
Correlation



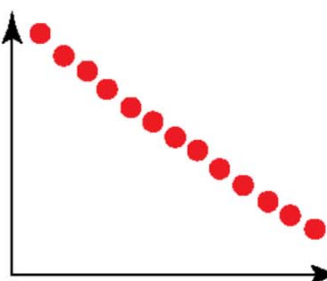
Weak
Positive
Correlation



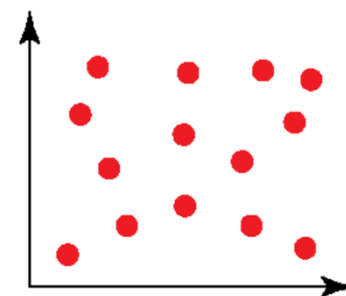
Weak
Negative
Correlation



Strong
Negative
Correlation



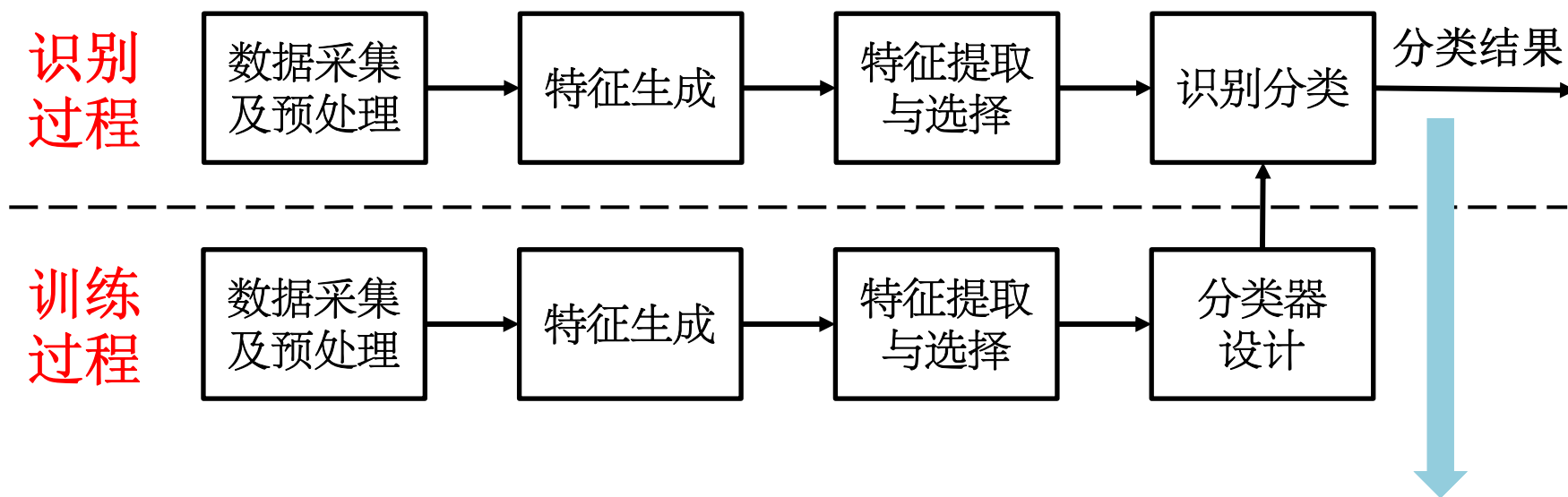
Perfect
Negative
Correlation



No
Correlation

模式识别系统

- 一个完整的模式识别系统：



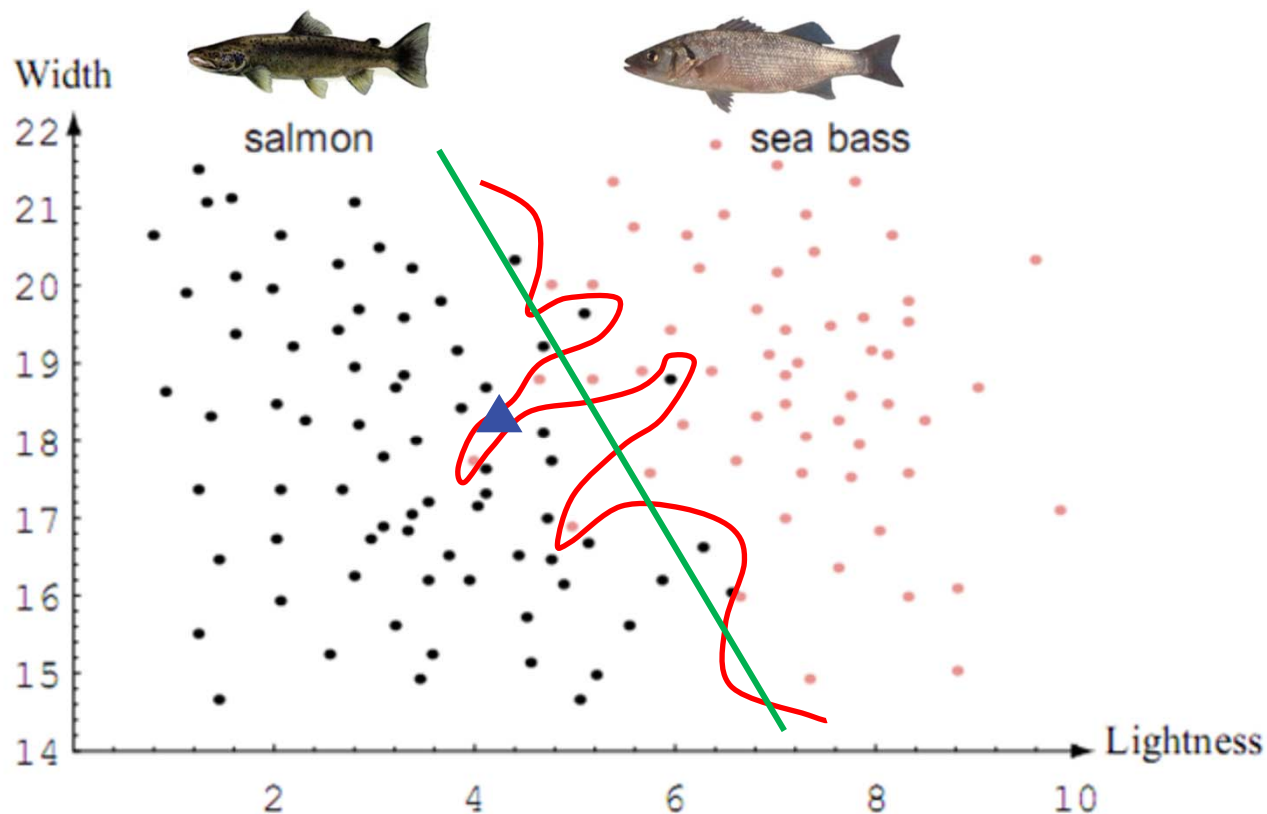
- 分类器性能评价准则
- 分类器性能评价指标
- 分类器性能评价方法

分类器性能评价准则

- 什么样的分类器是一个好的分类器？
- 模式识别的任务是确定能够提供最佳泛化性能的分类器，而不是提供最佳训练性能的分类器。
 - 训练性能：分类器识别训练数据类别的能力
 - 泛化性能（预测性能）：分类器识别未知标签的测试数据类别的能力

分类器性能评价准则

- （第一讲）思考：下图中简单的分类器（绿色直线）和复杂的分类器（红色曲线），哪一个是最好的分类器？

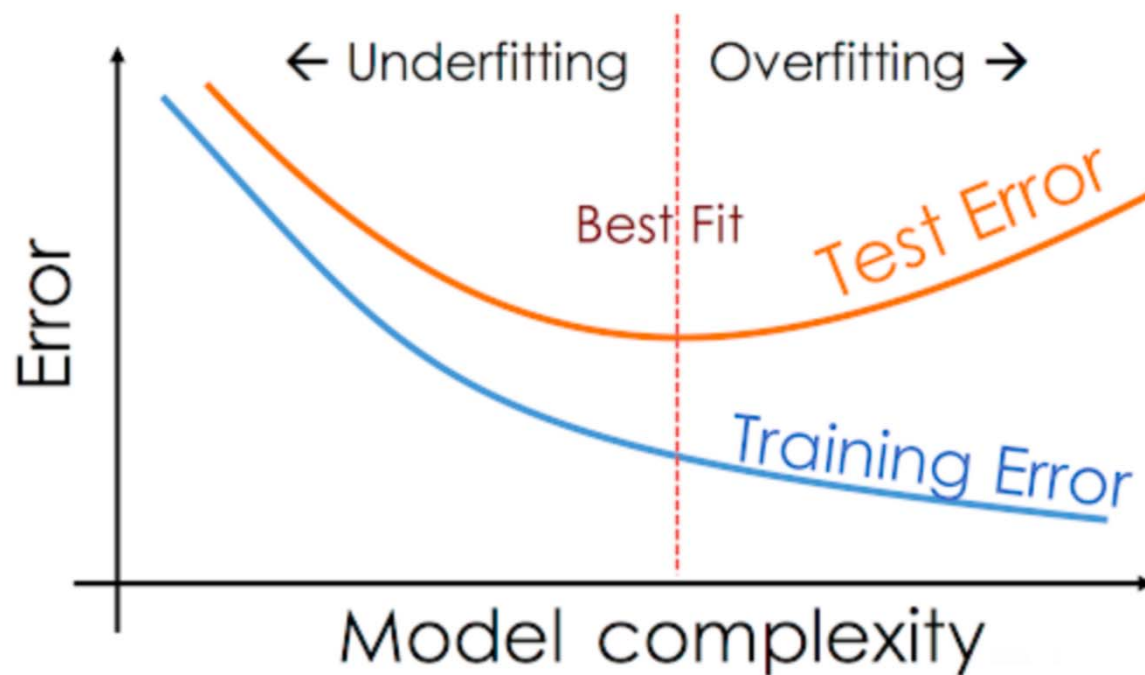


分类器性能评价准则

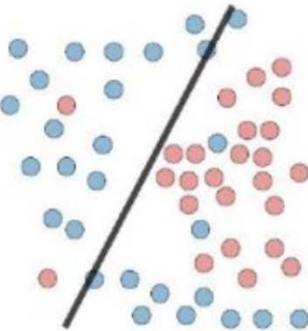
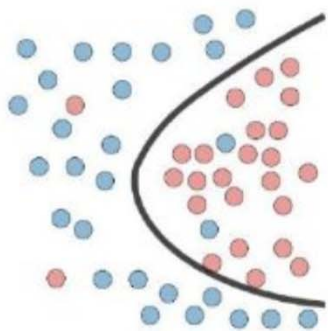
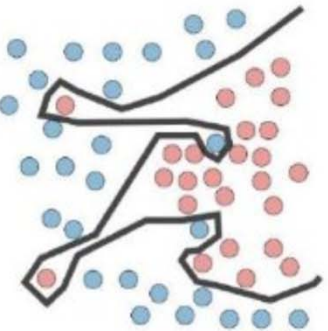
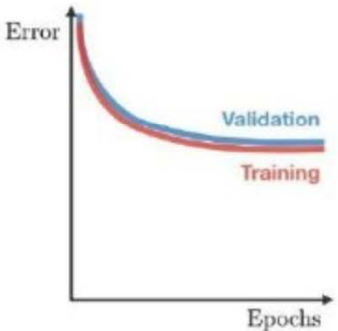
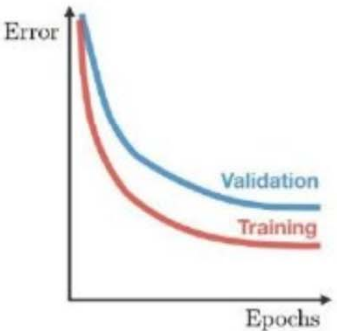
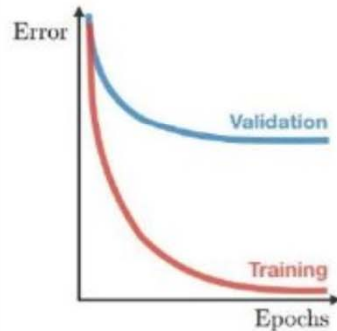
- 训练数据的完美分离通常是由噪声或随机误差引起的；一个可以完美分类训练数据的分类器事实上是学习了数据中的噪声或随机误差。
- **过拟合**：分类器过于复杂，学习噪声或随机误差而不是真正的潜在关系
- **欠拟合**：分类器过于简单，无法准确识别数据特征和类别间潜在关系

分类器性能评价准则

- 最佳分类器应该在测试数据上的表现最好（如，错误处于最小值）。



分类器性能评价准则

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> - High training error - Training error close to test error - High bias 	<ul style="list-style-type: none"> - Training error slightly lower than test error 	<ul style="list-style-type: none"> - Low training error - Training error much lower than test error - High variance
Classification			
Deep learning			
Remedies	<ul style="list-style-type: none"> - Complexify model - Add more features - Train longer 		<ul style="list-style-type: none"> - Regularize - Get more data

性能评价

- 使用什么量化指标确定最优的分类器和确定最优的分类器参数?
- 分类器设计方案和参数选择很大程度上依赖于对分类器性能的评价指标和评价方法。

<u>评价指标</u>	<u>评价方法</u>
<ul style="list-style-type: none">• 准确率/错误率• 拒识率• 敏感性、特异性、ROC曲线• 召回率、精确率、F1指标	<ul style="list-style-type: none">• 两分法• 交叉验证• Bootstrap方法

评价指标

- **准确率/错误率**：最简单的指标。对 m 个样本分类，其中 m_c 个被正确分类， m_e 个被错误分类，则

$$\text{准确率 } P_{acc} = m_c / m, \text{ 错误率 } P_{err} = m_e / m$$

- **拒识率**：只对有把握的样本判别类别，对没有把握的样本拒绝识别。对 m 个样本分类，其中 m_r 个被拒识， m_c 个被正确分类， m_e 个被错误分类，则

$$\text{拒识率 } P_{ref} = m_r / m, \text{ 准确率 } P_{acc} = m_c / (m - m_r)$$

评价指标

- 准确率不适用于不平衡数据集。
 - 分类器在处理不平衡数据时,往往会倾向于保证多数类的准确率而牺牲少数类的准确率。
- 准确率无法正确反映分类错误的代价,因为并非所有错误的代价都相等。
 - 例如疾病诊断中的两种错误: 健康人被误诊为病人, 代价是可能是实际不需要的副作用或药物成本; 但是病人被误诊为健康人, 代价是可能导致病人死亡。

评价指标

- 医学二分类诊断中，常用“阳性positive”表示患病，“阴性negative”表示正常。则可得到以下的混淆矩阵：

		真实值	
		阳性	阴性
预测输出	阳性	真阳性 True Positive (TP) <i>Correct outcome</i>	假阳性 False Positive (FP) <i>Type I error</i>
	阴性	假阴性 False Negative (FN) <i>Type II error</i>	真阴性 True Negative (TN) <i>Correct outcome</i>

评价指标

- 敏感性（Sensitivity）：分类器从所有阳性样本中正确识别阳性结果的比例

$$P_{sen} = \frac{TP}{TP + FN}$$

- 特异性（Specificity）：分类器从所有阴性样本中正确识别阴性结果的比例

$$P_{spe} = \frac{TN}{TN + FP}$$

- 准确率总是介于敏感性和特异性之间。

评价指标

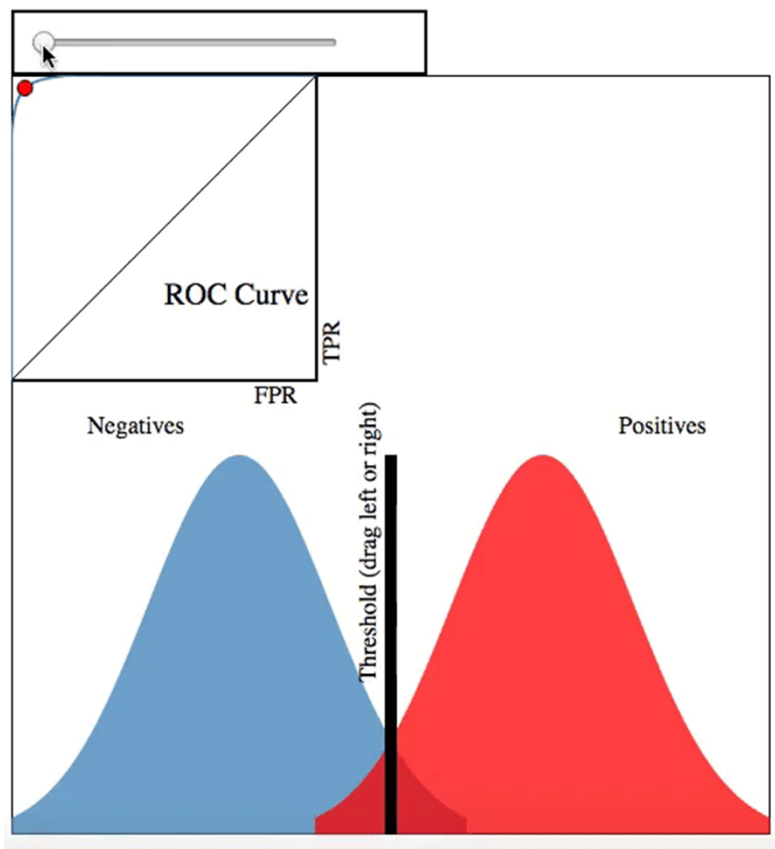
- 假阳率（false positive rate, FPR）：
正常人被误诊为患病的比率

$$FPR = 1 - P_{spe} = \frac{FP}{FP + TN}$$

- 类似可定义真阳率TPR（=敏感性）、真阴率TNR（=特异性）、假阴率FNR等。
- 敏感性和特异性（或真阳率和假阳率）的权衡：
一个模型很难同时最大化敏感性和特异性，需要
权衡哪一个更重要。

评价指标

- 研究敏感性特异性均衡常用受试者工作特征曲线（receiver operating characteristic, ROC）



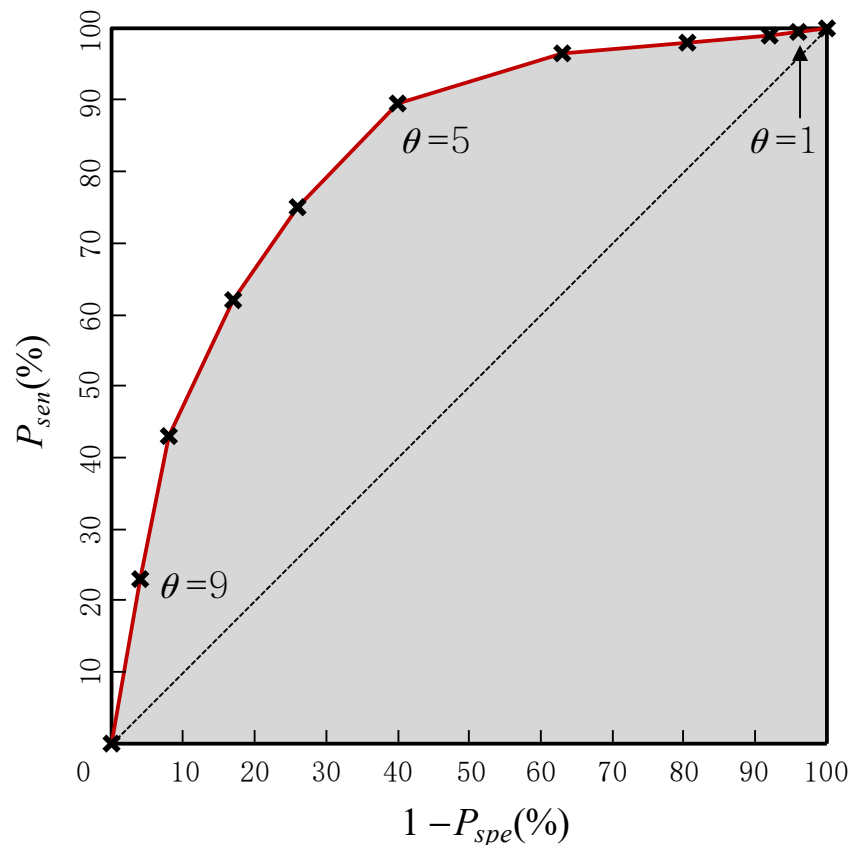
在ROC曲线中，敏感性表达为分类器在不同参数设置下的关于假阳率的一个函数。

- 左图中，红蓝两类样本由黑色粗线分类。
- 黑线（分类器参数）的移动引起不同的敏感性TPR和假阳性FPR，构成ROC曲线。
- ROC曲线的形状和红蓝两类的分布有关。

评价指标

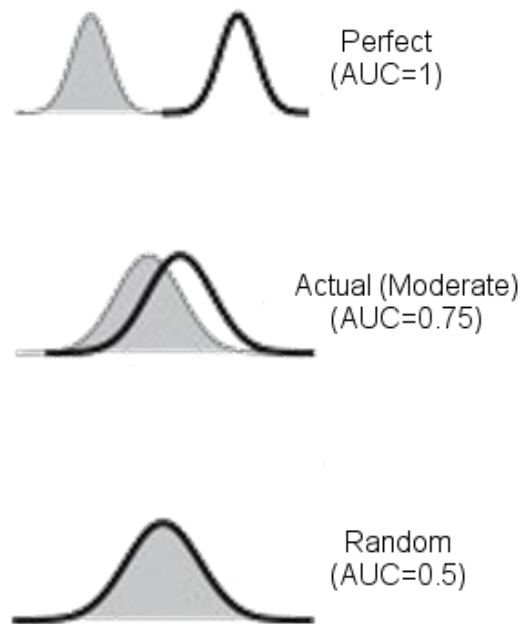
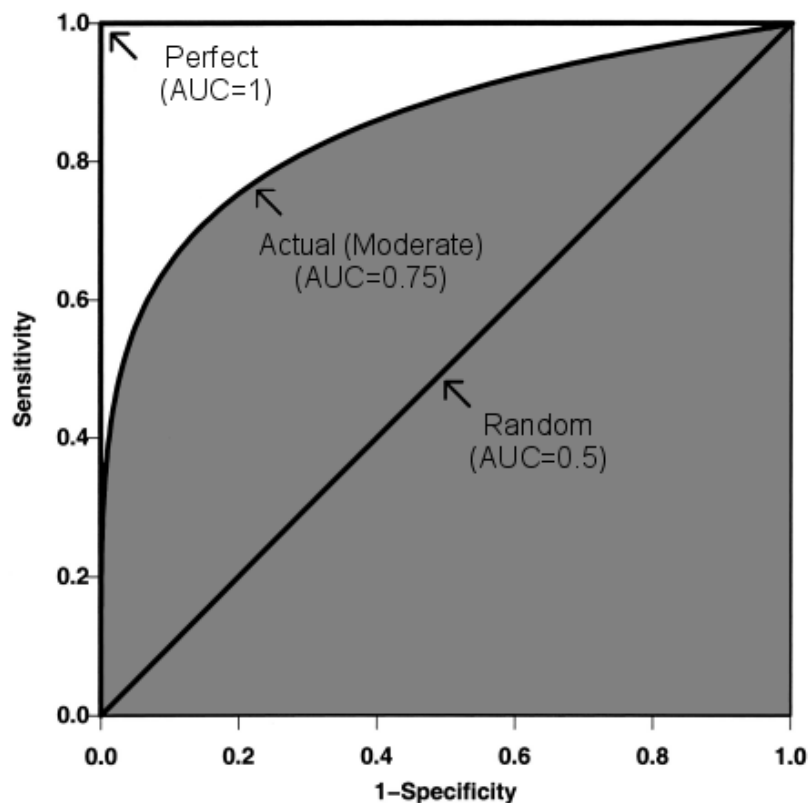
- 例：某疾病根据9个指标（阳性或阴性）诊断。分类器设定一个阈值 θ ，根据9个指标中阳性数量是否大于或等于 θ 判断是否得病。

θ	敏感性 P_{sen}	假阳率 $1 - P_{spe}$
10	0	0
9	23.1	4.3
8	43.8	8.6
7	62.2	17.1
6	72.3	25.9
5	89.6	40.4
4	96.1	63.5
3	98	80.2
2	98.9	92.3
1	99.5	96.5
0	100	100



评价指标

- ROC曲线下的面积（Area Under Curve, AUC）可以度量两个类别之间的区分程度，也可以评估不同分类器的优劣（AUC越大越好）。



评价指标

- 召回率（Recall）和精确率（Precision）：常用于信息检索和医学诊断等应用中

		实际类别	
		相关	不相关
分类结果	检索到	TP	FP
	未检索到	FN	TN

- 召回率（查全率）：相关的信息中被检索出来的比例 $P_{rec} = \frac{TP}{TP + FN}$ 与敏感性计算相同
- 精确率（查准率）：检索到的信息中与主题相关的比例 $P_{pre} = \frac{TP}{TP + FP}$

评价指标

- F1指标（F1-score）：兼顾召回率和准确率，使用二者的调和平均

$$F_1 = \frac{2}{\frac{1}{P_{rec}} + \frac{1}{P_{pre}}} = \frac{2P_{rec}P_{pre}}{P_{rec} + P_{pre}}$$

- 使用何种性能评价指标确定最优分类器由实际问题的需求决定。
 - 对漏诊后果严重，早筛可获及时治疗诊断的疾病（如乳腺癌），需要敏感性高的分类器；
 - 对误诊后果严重（患者心理压力与负担）且无有效治疗的疾病（如胰腺癌），需要特异性高的分类器。

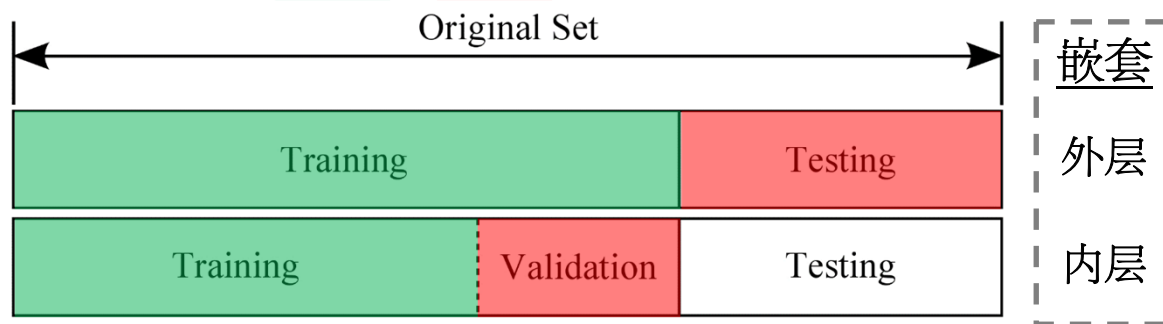
评价方法

- 各类评价指标一般是在一组样本上的随机实验的结果。如何利用有限的样本集既能够训练一个分类器，又能够准确地评价分类器的性能是一个重要问题。
- 如何将所有可用样本分成训练集和验证/测试集？
 - 训练集数据应充分、有代表性，以便对分类器进行良好的训练。
 - 验证/测试集数据也应该是足够的并且具有代表性，这样对分类器表现的评价才可靠。

评价方法

- 两分法（留出法Hold-out）：随机将样本集划分为不相交的两个子集，分别用于训练和测试。
- 划分可以随机重复若干次，结果取均值。
- 一般地，训练和测试的比例可以是7:3或8:2。
- 缺点：只能用一部分样本训练分类器，另一部分样本测试性能（即使重复若干次后）。

注：此处的“训练和测试”可以是以下两种情况：

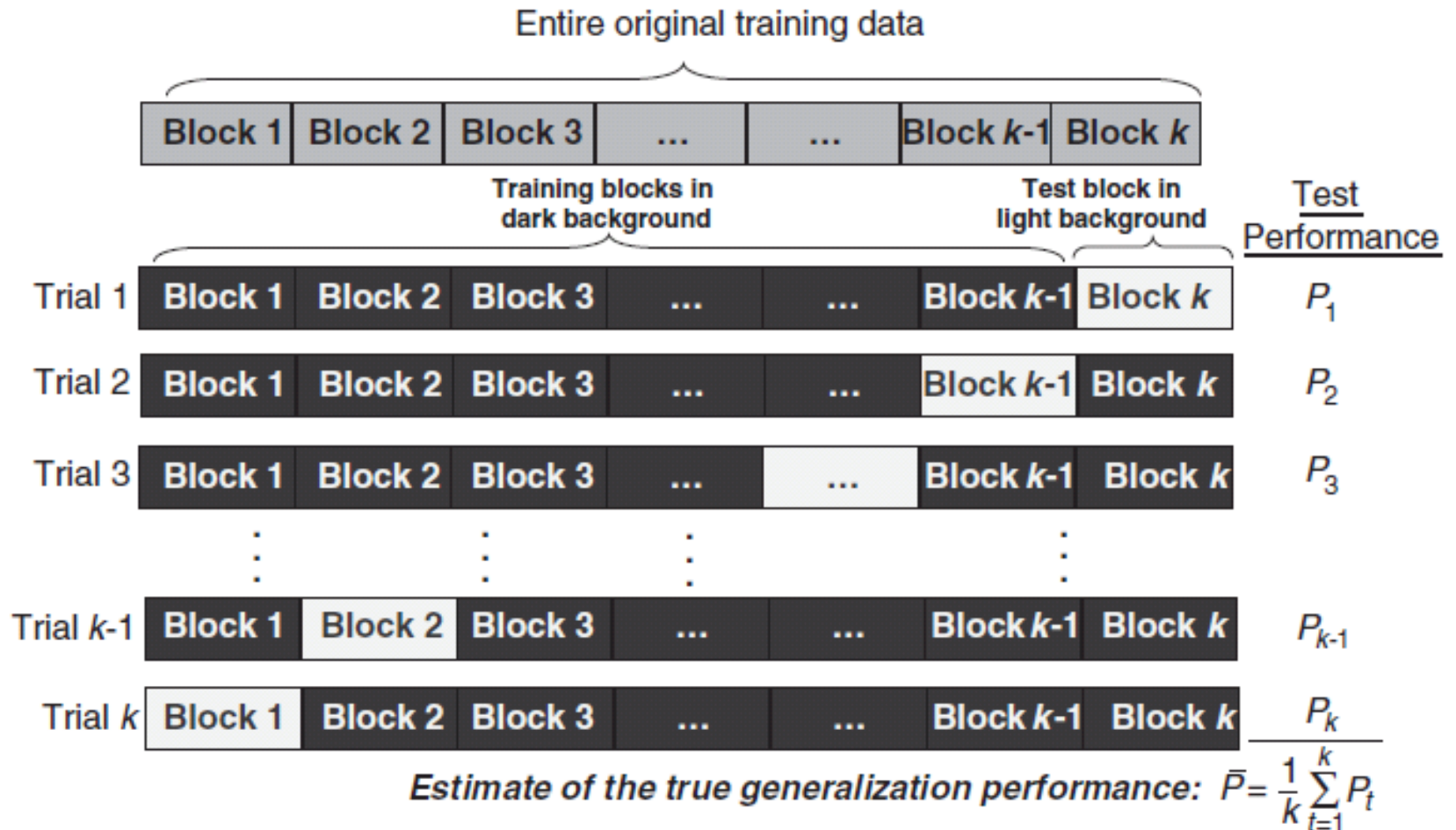


评价方法

- 交叉验证（Cross Validation, CV）：
 1. 将样本集随机划分成不相交的 k 个子集（ k 大于2，通常介于5到10之间），每个子集中的样本数量相同；
 2. 使用1个子集做测试，其余 $k-1$ 个子集训练；
 3. 交叉验证过程重复 k 次，这样每个子集都可以作为一次测试集；
 4. 以 k 次结果的平均值作为分类器的性能评价指标。
- 留一法（Leave-one-out CV, LOOCV）：
 $k = N$ ，其中 N 为样本数

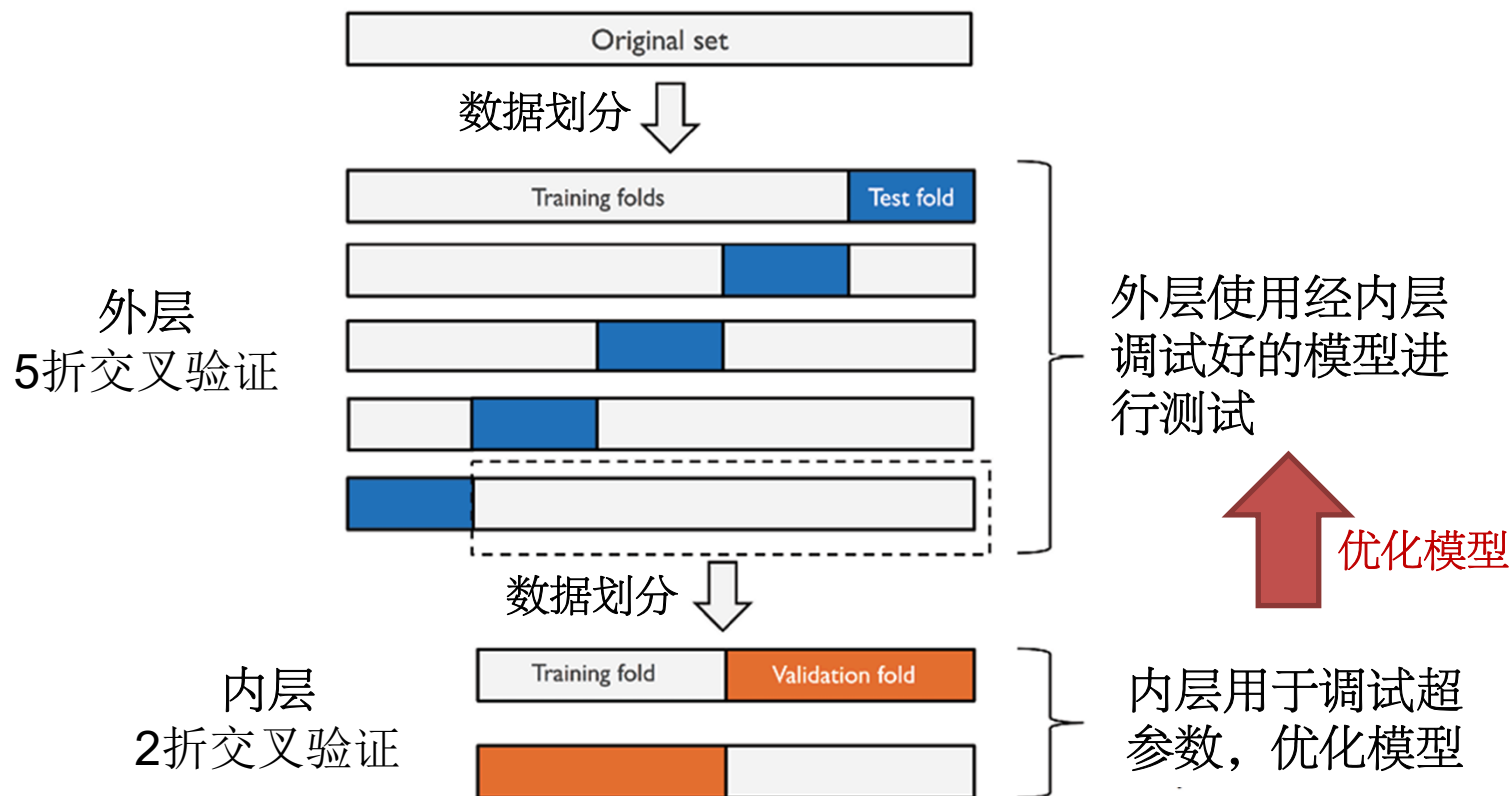
评价方法

- 交叉验证 (Cross Validation, CV)



评价方法

- 嵌套交叉验证（Nested Cross Validation）：当数据集分为训练、验证、和测试集时，交叉验证可以分两次嵌套进行。



评价方法

- Bootstrap方法:

1. 从样本集中有放回地抽取 n 个样本，组成一个Bootstrap样本集 A (A 中样本可重复)；
2. 同样方式得到另一个Bootstrap样本集 B ；
3. A 和 B 分别作为训练集和测试集，得到一个分类器性能评价；
4. 重复上述过程 k 次，取 k 次平均值作为分类器性能的评价。

数据泄露

- **数据泄露**：用于训练分类器的数据中包含了来自测试集的信息。
- 数据泄露会导致测试结果虚高，但用于新的独立测试数据时结果很差。
- 数据泄漏的常见原因：在进行预处理（如特征标准化）、特征提取与选择（如降维）时使用了所有数据，仅在分类时划分数据集。
- 解决方法：在每一折交叉验证中和每一个模式识别步骤中都要严格隔离训练和测试集。

本章小结

- 介绍了最简单直接的距离分类器的一般形式和模板匹配
- 介绍了最近邻分类器及其加速，和K-近邻算法。
- 介绍了多种距离度量和相似度度量
- 介绍了分类器评价准则（泛化能力最佳，可最小化测试误差）、评价指标（准确率、敏感性、特异性等）、评价方法（留出法、交叉验证等）