

模式识别

Pattern Recognition

第2章 模式识别系统

Hello PR

■模式识别系统构成

- 两个基本环节：先“找到目标”，再“做出判别”
- 特征的计算：“精致手工”、“数学游戏”、“暴力美学”？
- 特征的度量与分类：如何评价特征间的相似度？
- 特征的归一化：新手容易忽略的问题
- 如何评价分类效果？

■模式识别系统实践——XXX多模态信息获取

■机器学习训练的秘籍——吴恩达

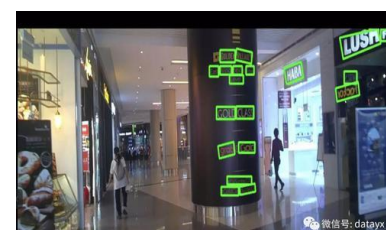
■下一步：提取更有效的特征？更好的分类方法？

模式识别系统构成

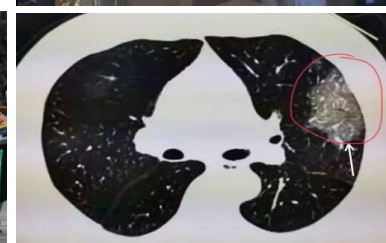
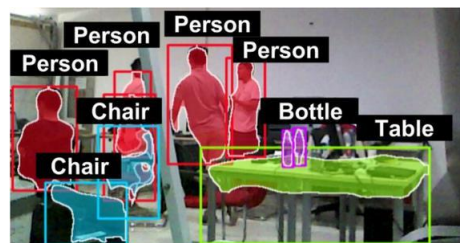
■两个步骤：先“找到目标”，再“做出判别”

视觉信号典型应用场景举例：

符号



人物



环境



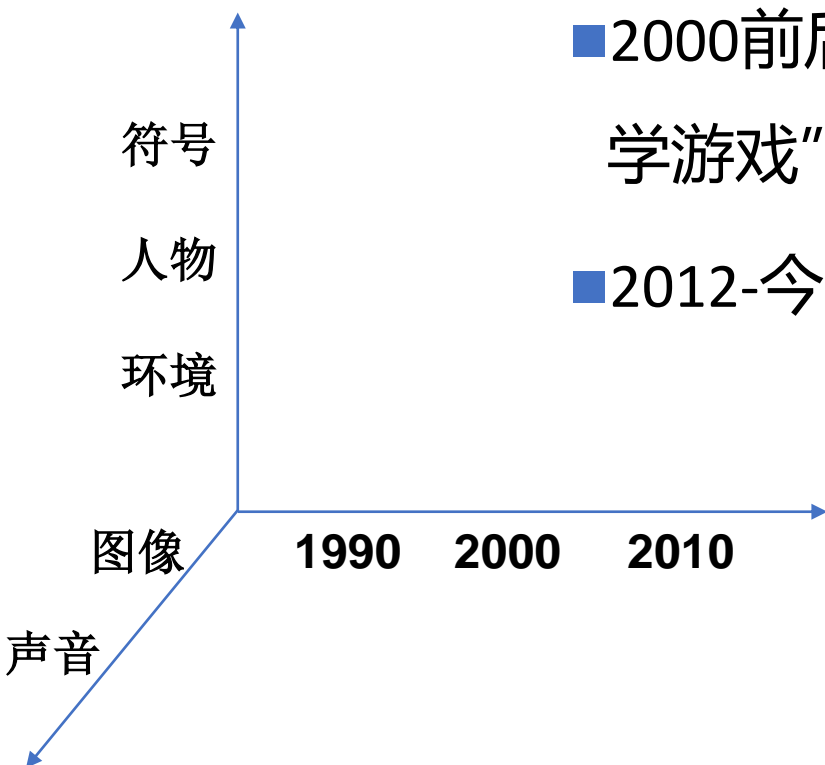
模式识别系统构成

■特征的计算：模式识别系统的核心问题

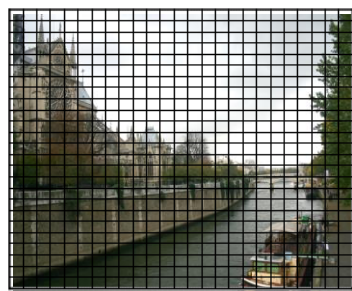
■ 8090年代：特征工程 “精致手工” ？

■ 2000前后：各种滤波、信息压缩 “数学游戏” ？

■ 2012-今：深度学习 “暴力美学” ？



特征计算、分类?



分块、加权、卷积
梯度、方向、归一
相似、距离、度量
内积、外积、统计
代价、折中、优化...

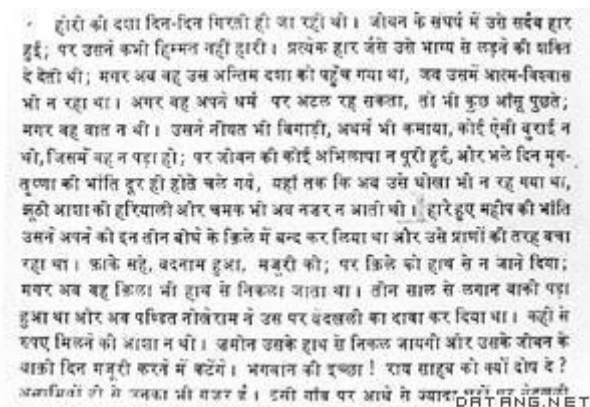
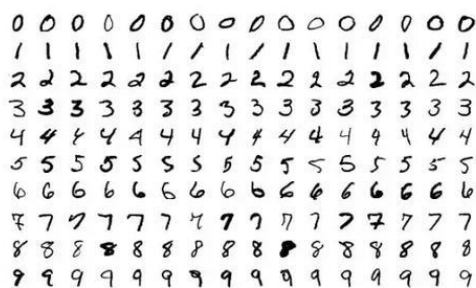
样本、显卡、金钱...

特征计算

■ “精致手工”？

——根据信号特点，直接人工设计特征计算方法

典型案例：90年代，符号、字符识别（OCR）

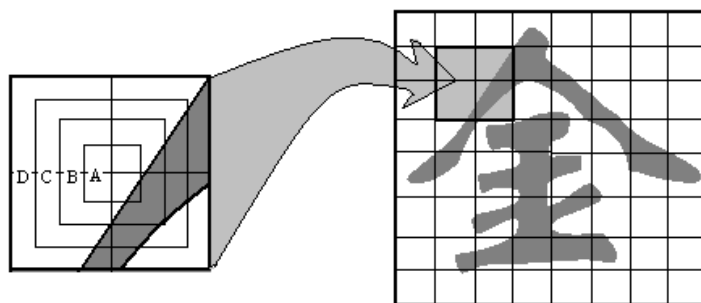
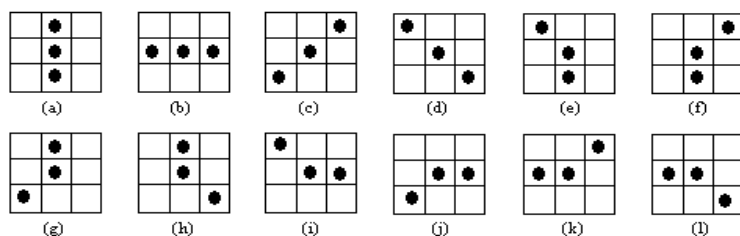


联机、脱机
印刷、打印、手写
数字、英文、中文，印地语？

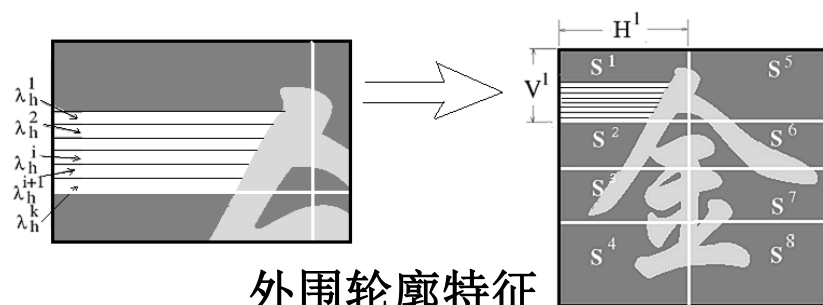
特征计算

- “精致手工”？根据信号特点，直接设计特征计算方法

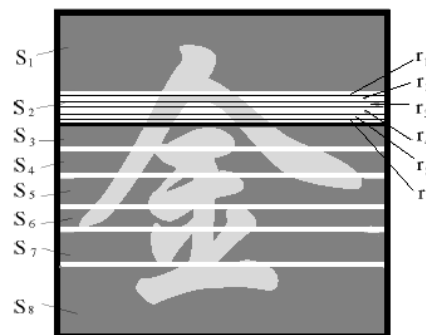
典型案例：90年代，符号、字符识别（OCR）



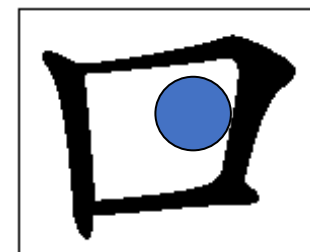
笔画方向特征（DEF）



外围轮廓特征



笔画穿越特征



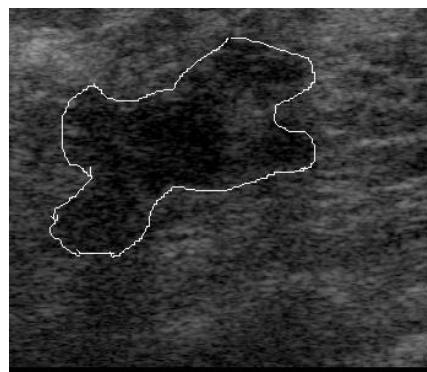
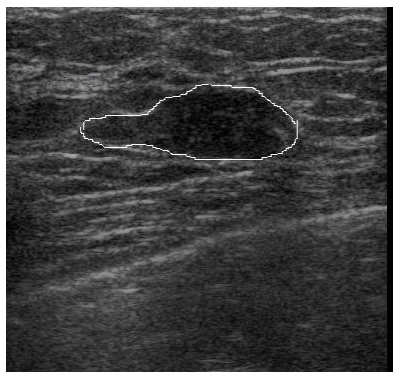
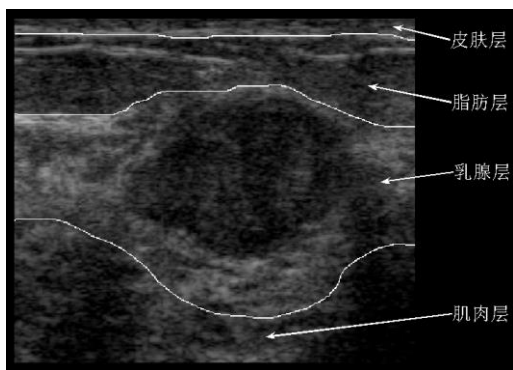
笔画密度特征

特征计算

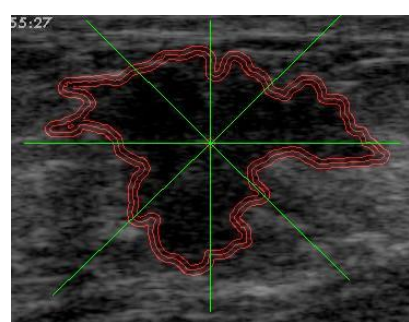
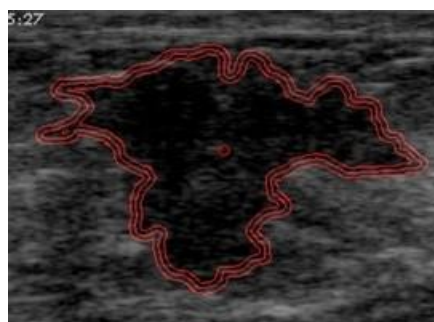
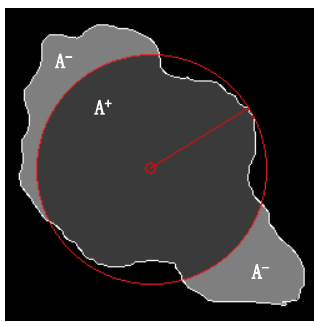
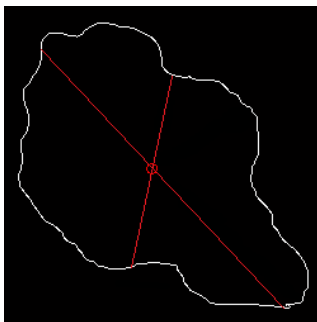
- “精致手工”？根据信号特点，直接设计特征计算方法

典型案例：2000前后，医学影像辅助诊断

目标分割



形状描述



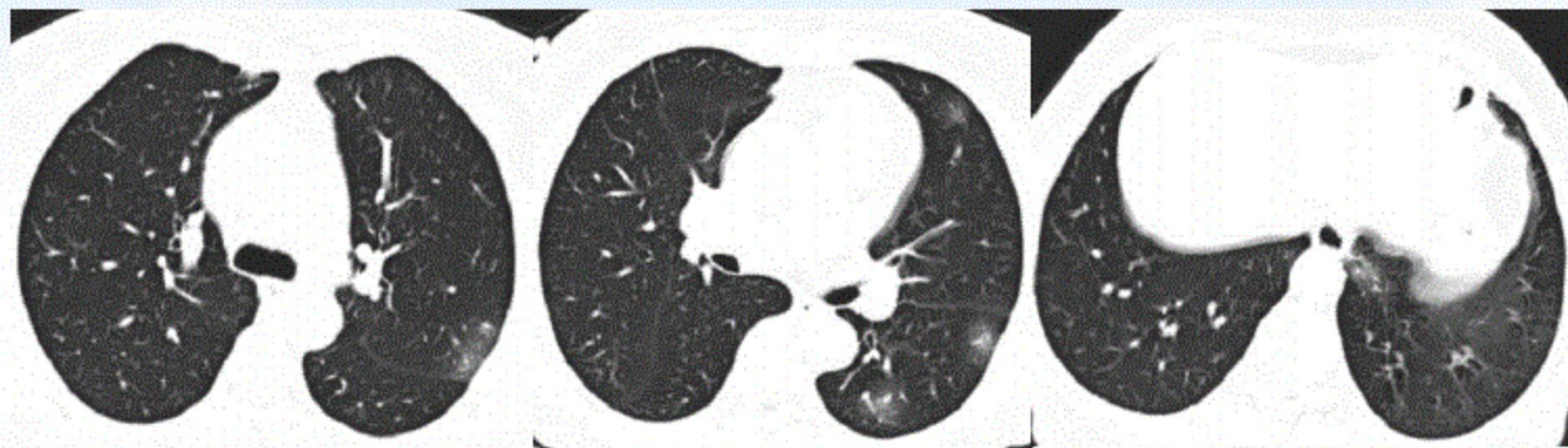
整体形状特征

边缘轮廓特征

内部纹理特征

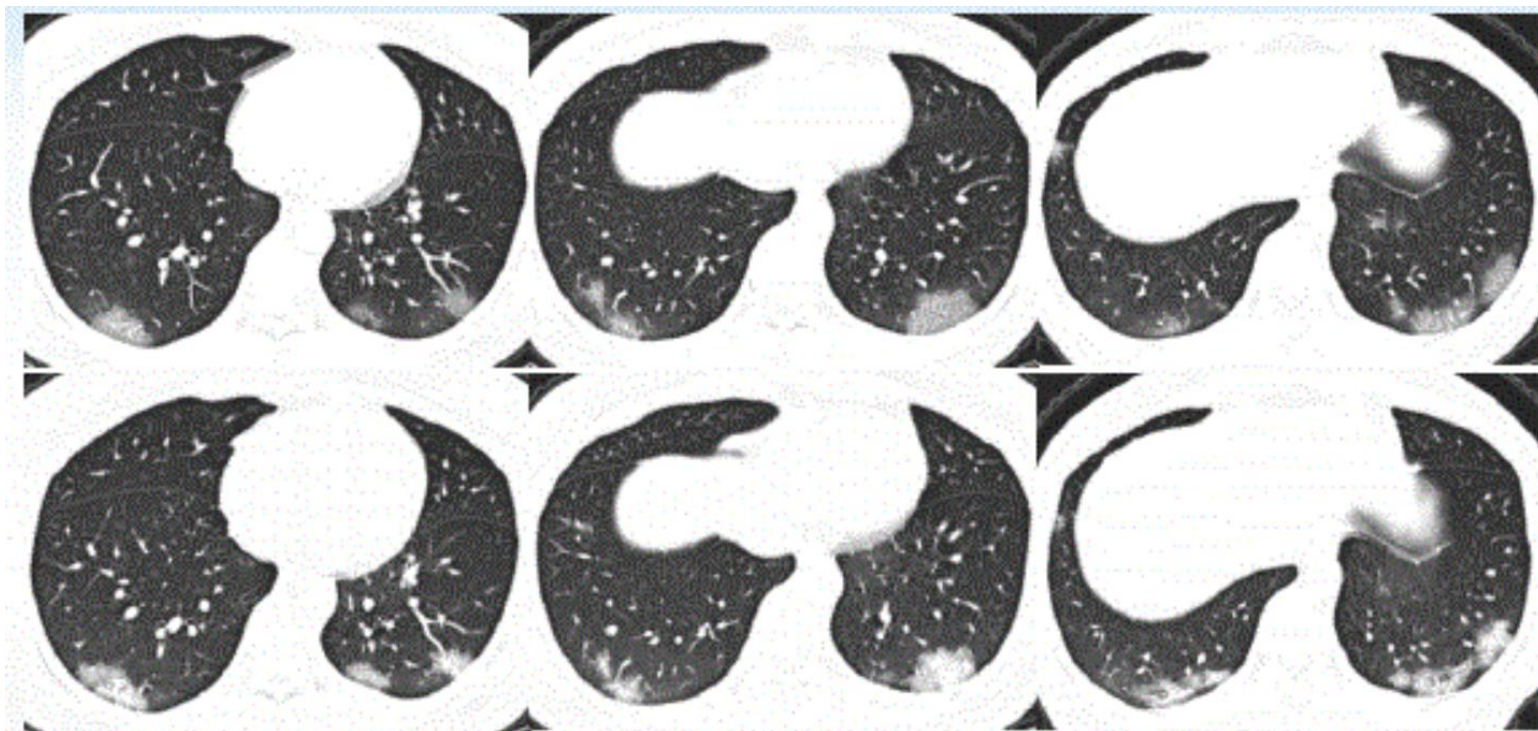
特征计算

如何检测早期新冠肺炎？



- 病变沿支气管血管束分布，多位于外周带胸膜下。
- 病变范围局限，可单发或多发，呈斑片状/结节状GGO。
- 常可见小血管增粗，伴或不伴细网格影。

发展期新冠肺炎，如何量化评估？



- 病变增多、范围扩大或融合，密度增高。
- GGO与实变并存，可见气腔结节。

特征计算

- “数学游戏”？综合滤波、投影、压缩等方法，获得相对通用、稳定有效的特征描述子。

典型案例：2000年前后（参见OpenCV）

HOG+SVM行人检测

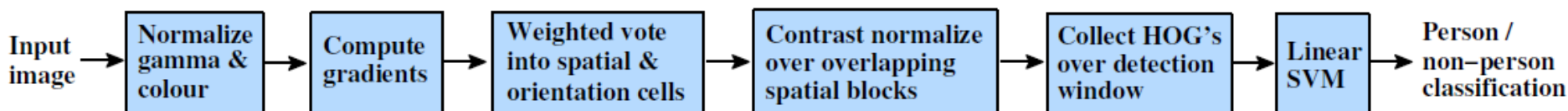
Haar+AdaBoost 人脸识别

SIFT（尺度不变特征转换）图像匹配

特征描述子——HOG

■方向梯度直方图 (Histogram of Oriented Gradient, HOG)

利用图像边缘的方向密度分布描述图像特征



1. 图片Gamma和颜色的归一化;
2. 计算梯度;
3. 构建直方图;
4. Block混叠空间块的归一化;
5. 构建HOG特征描述子;
6. SVM训练;

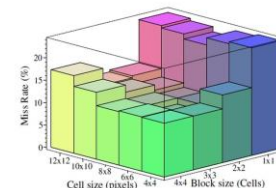
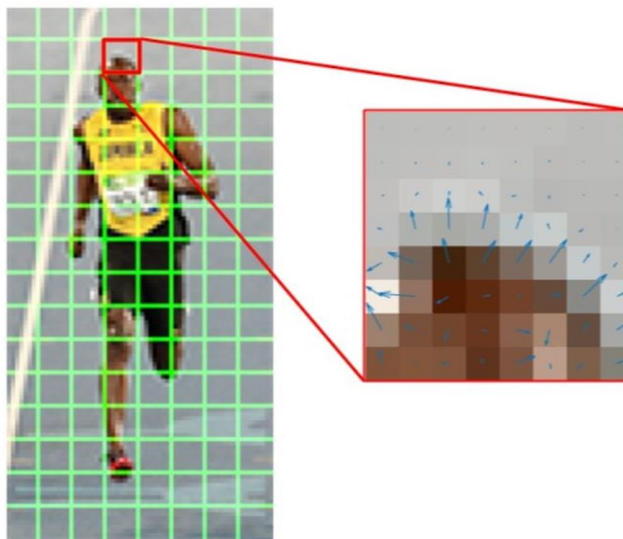
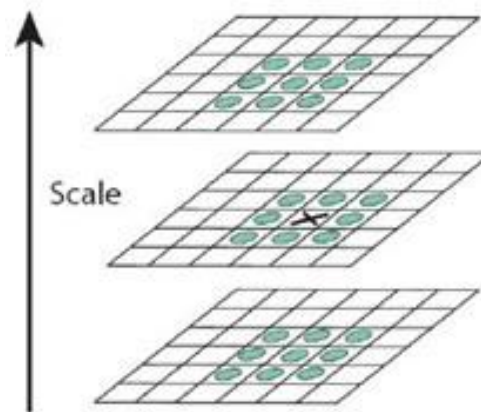
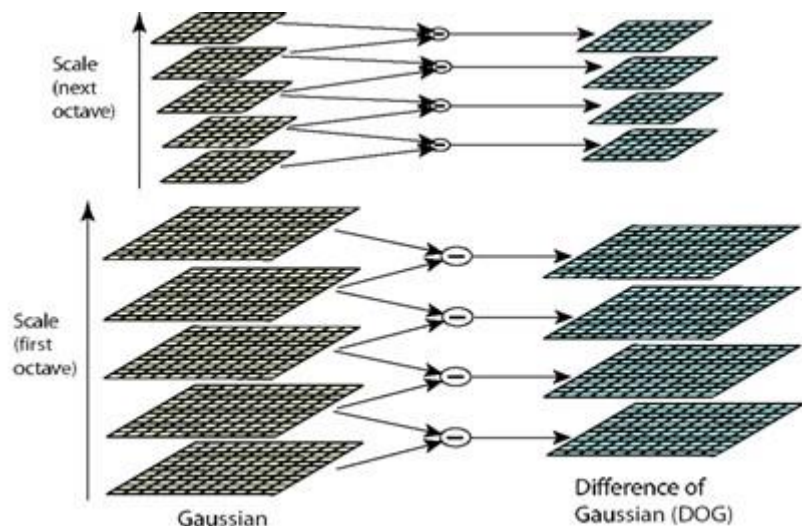


Figure 5. The miss rate at 10^{-4} FPPW as the cell and block sizes change. The stride (block overlap) is fixed at half of the block size. 3x3 blocks of 6x6 pixel cells perform best, with 10.4% miss rate.

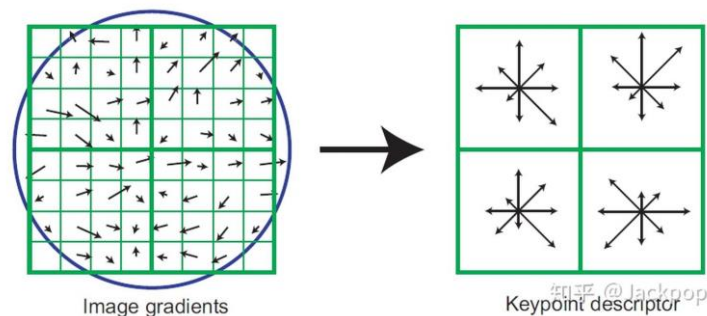
特征描述子——SHIFT

■尺度不变特征变换 (Scale-invariant feature transform)

改变旋转角度，图像亮度或拍摄视角，仍然能够得到好的检测效果



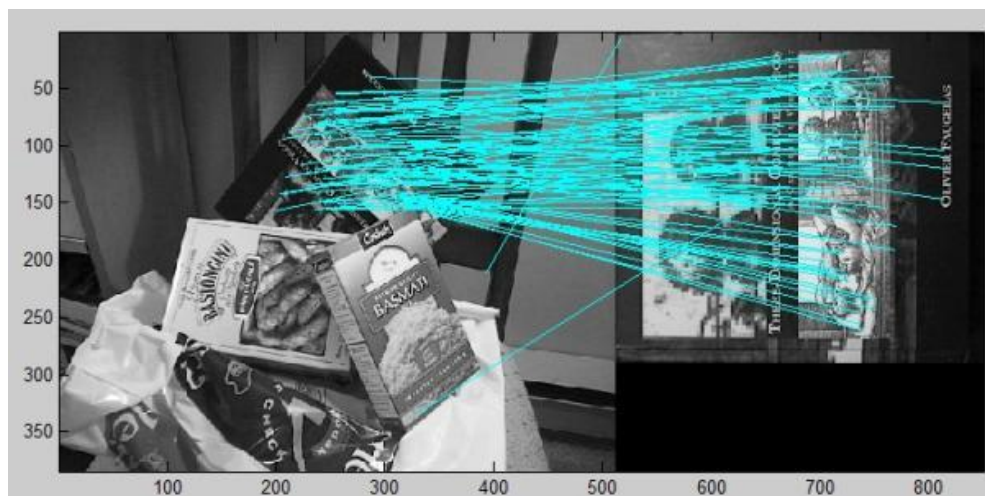
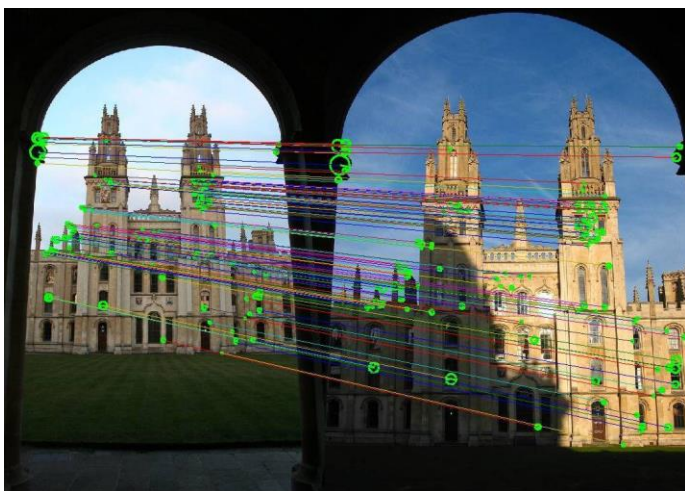
- 1) 构建多尺度高斯差分DOG边缘图像序列
- 2) 找到局部极值点，去除不对称、不清晰、不稳定的点作为特征点。
- 3) 找到特征点主方向，梯度直方图描述



特征描述子——SIFT

■ 尺度不变特征变换 (Scale-invariant feature transform)

改变旋转角度，图像亮度或拍摄视角，仍然能够得到好的检测效果



在目标检测和特征提取领域有广泛应用

特征的比较与分类

■ 将待识别样本 \mathbf{x} 分类到与其最相似的类别中

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

$$j = \arg \max_{1 \leq i \leq c} s(\mathbf{x}, \omega_i)$$


关键问题：如何度量样本 \mathbf{x} 与类别 ω_i 的相似程度？

模板匹配

- 每个类别的“先验知识”就是一个样本（模板） μ_i
- 利用 \mathbf{x} 与模板 μ_i 的相似度，作为 \mathbf{x} 与类别 ω_i 的相似度
- 用“距离”度量样本之间的相似度

$$\omega_i \rightarrow \mu_i$$

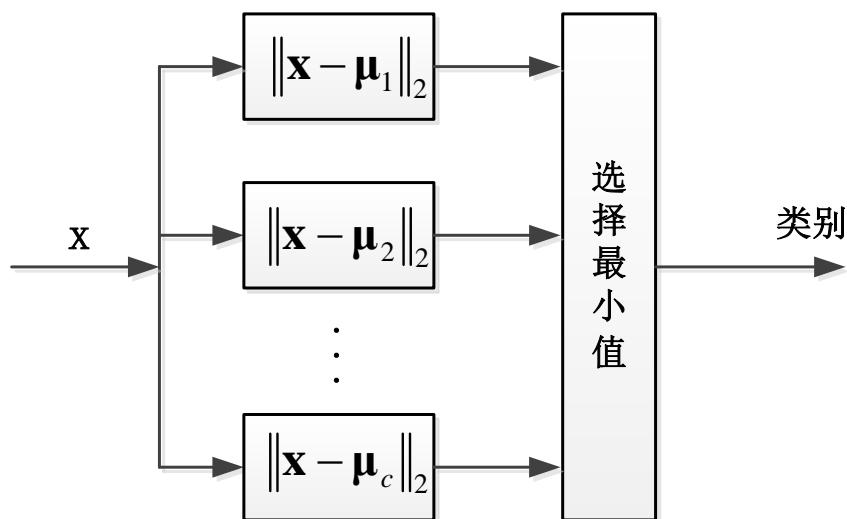
$$s(\mathbf{x}, \omega_i) = s(\mathbf{x}, \mu_i)$$

$$\begin{aligned} s(\mathbf{x}, \mu) &= -d(\mathbf{x}, \mu) \\ &= -\|\mathbf{x} - \mu\|_2 = -\sqrt{\sum_{i=1}^d (x_i - \mu_i)^2} \end{aligned}$$

欧氏距离

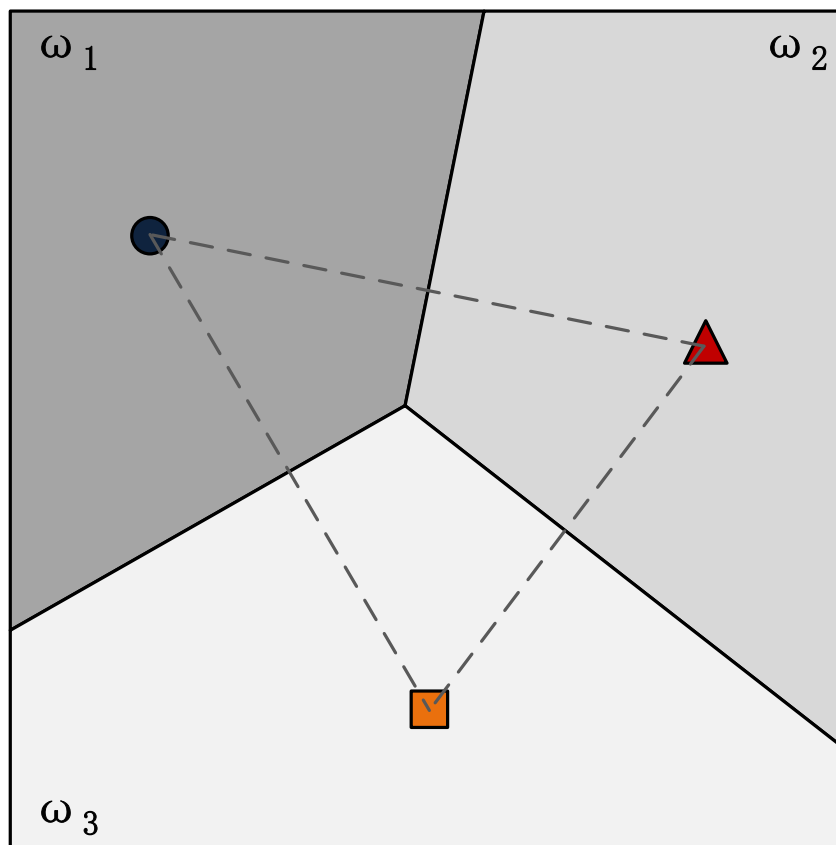
模板匹配

- $\|\cdot\|_2$ 矢量的“ l_2 范数”——矢量的长度
- 差矢量 $\mathbf{x} - \boldsymbol{\mu}$ 的 l_2 范数——两个点之间的欧氏距离
- 模板匹配过程：
$$j = \arg \min_{1 \leq i \leq c} d(\mathbf{x}, \boldsymbol{\mu}_i)$$



模板匹配——判别界面

- c 个模板将特征空间划分成了 c 个区域
- 两个区域的交界称为“判别界面”



最近邻分类器

- 每一个类别有多个训练样本 $D_i = \{\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_{n_i}^{(i)}\}$

类别 $i = 1, \dots, c$. n_i : 第*i*类训练样本个数

- 样本 \mathbf{x} 与类别 ω_i 之间相似程度:

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

最近邻分类器

-
- 输入：需要识别的样本 \mathbf{x} ，训练样本集

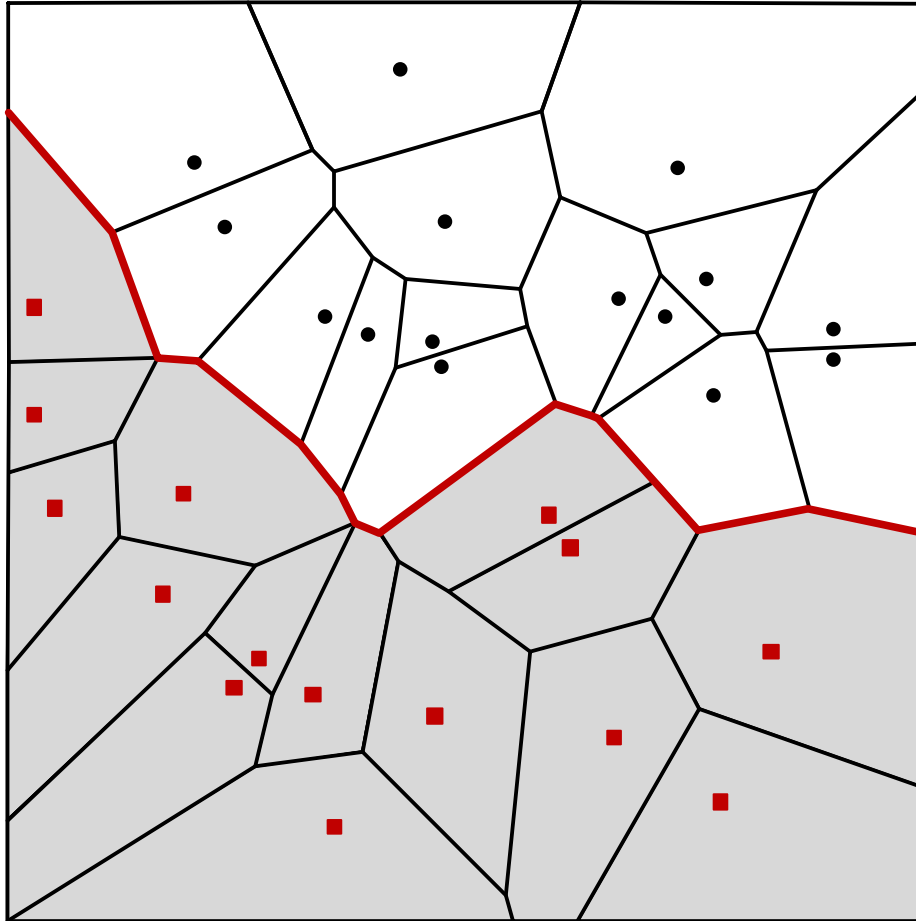
$$D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\};$$

- 寻找 D 中与 \mathbf{x} 距离最近的样本：

$$\mathbf{y} = \arg \min_{\mathbf{x}_i \in D} d(\mathbf{x}, \mathbf{x}_i);$$

- 输出： \mathbf{y} 所属的类别；
-

2.1.3 最近邻分类器-Voronoi网格



最近邻分类器——特点分析

- 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，也比较
- 易受样本噪声影响：
 - 最近邻算法只依赖最近训练样本，当训练样本某些特征有偏差、或者标注错误， 导致错误

最近邻分类器的加速

■ 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 μ

问题：什么样的 μ_i 最适合作为代表模板？

思路：距离训练样本距离都比较近的点。

模型：

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

求解：

1) 构造准则函数：

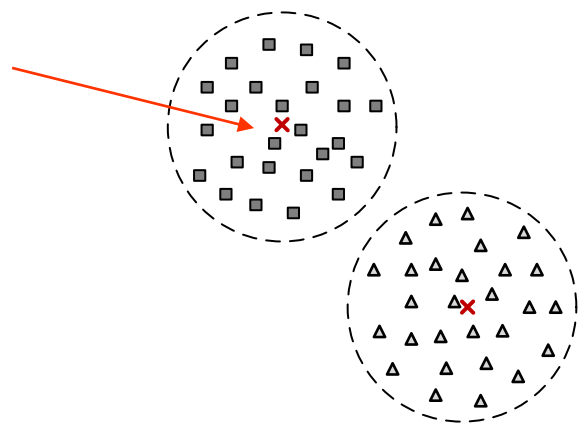
$$J_i(\mu) = \sum_{k=1}^{n_i} \underbrace{\|\mathbf{x}_k^{(i)} - \mu\|^2}_{\text{欧氏距离}}$$

误差平方和准则

2) 最优化：

$$\mu_i = \arg \min_{\mu \in R^d} J_i(\mu)$$

准则函数何处取得极值？



最近邻分类器的加速

■ 转化为单模板匹配

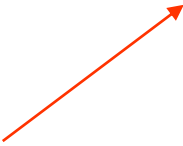
误差平方和准则函数：

$$J_i(\boldsymbol{\mu}) = \sum_{k=1}^{n_i} \underbrace{\left\| \mathbf{x}_k^{(i)} - \boldsymbol{\mu} \right\|^2}_{\text{范数的平方}} = \sum_{k=1}^{n_i} \underbrace{\left(\mathbf{x}_k^{(i)} - \boldsymbol{\mu} \right)^t \left(\mathbf{x}_k^{(i)} - \boldsymbol{\mu} \right)}_{\text{内积}}$$

极值点导数为0：

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

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

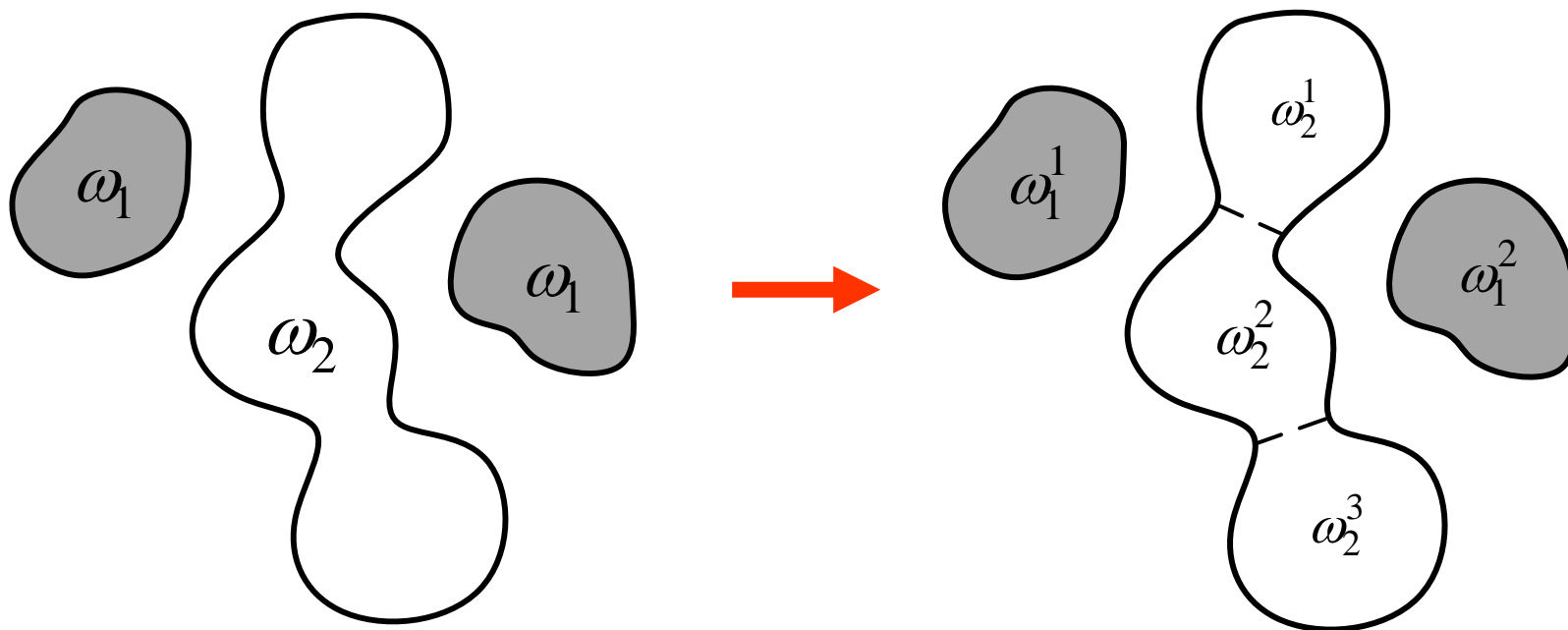
$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^t \mathbf{x}}$$


结论：计算每个类别训练样本的均值作为匹配模板



最近邻分类器的加速

■ 从“单模板”到“多模板”

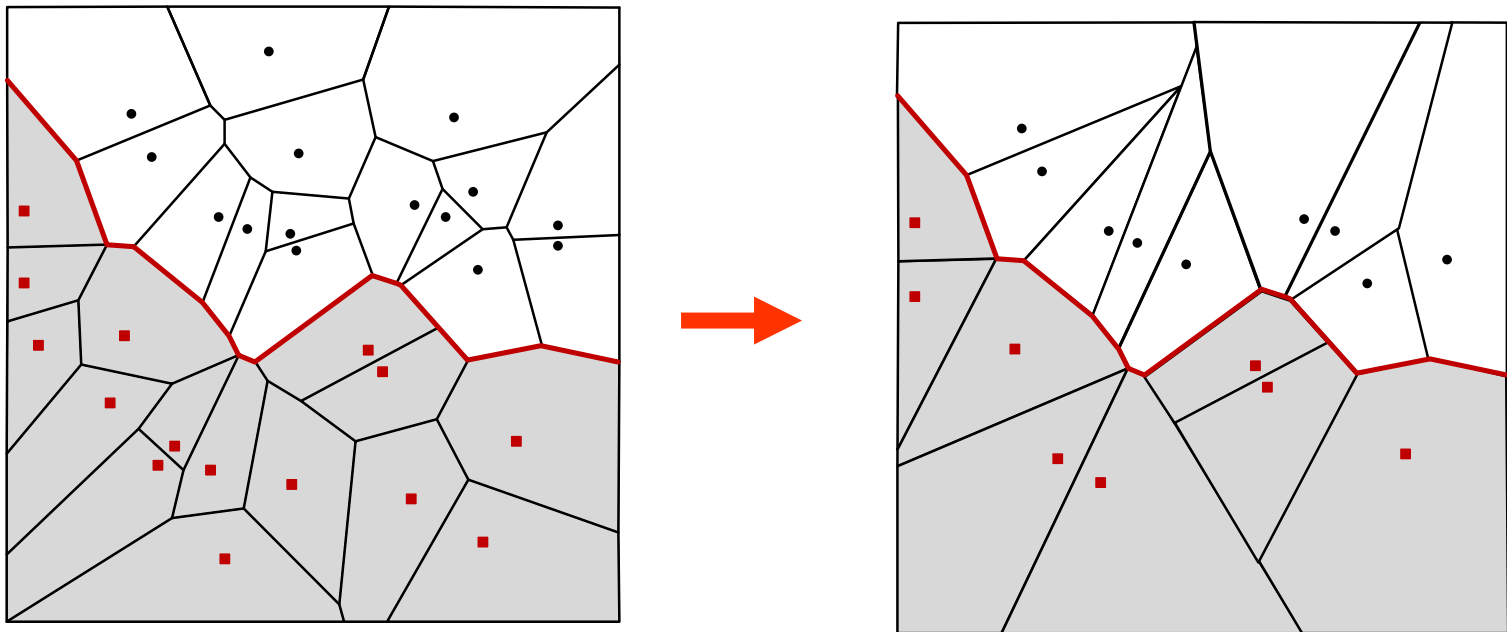


关键问题：高维特征空间如何划分？

——聚类分析

最近邻分类器的加速

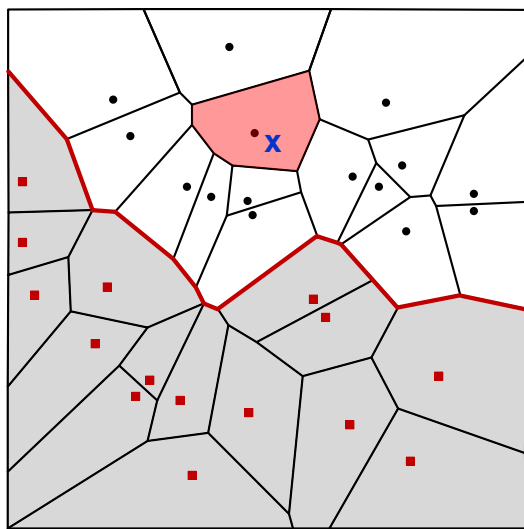
- 使用一个或多个模板
 - 能够提高计算效率——减少匹配次数
 - 不能保证准确率——改变了分类面形状
- 近邻剪辑
 - 去掉对分类面“无用”的点——不改变分类面形状



K-近邻分类器

■ 最近邻

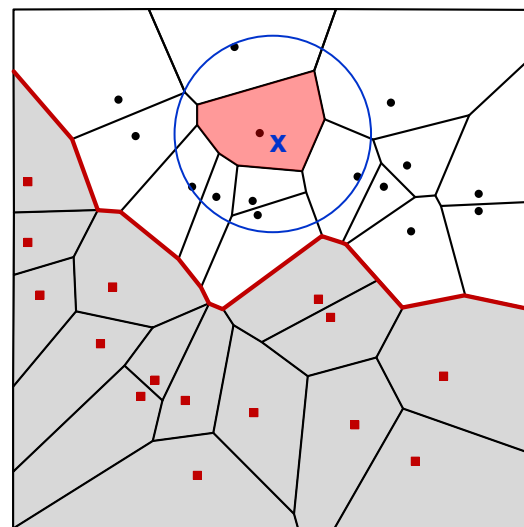
只依据距离最近的“一个”
样本的类别



一个噪声样本就将导致
一片错误分类区域

■ k-近邻

由距离最近的“K个”
样本投票来决定



寻找样本x最近的k=7个样本
投票6:1，正确识别

K-近邻分类器——特点

■ K的选择

- K值选择的过小，算法的性能接近于最近邻分类；
- K值选择的过大，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。

■ 非平衡样本集 (某一类样本数量很大，而其它类别样本的数量相对较少)

- 样本数多的类别总是占优势

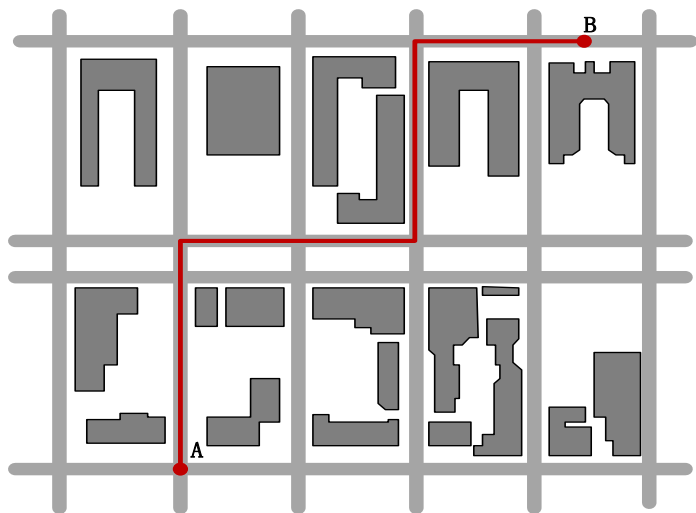
■ 计算量

- 需要与每一个训练样本计算距离。
- 寻找与其最相近K个样本。

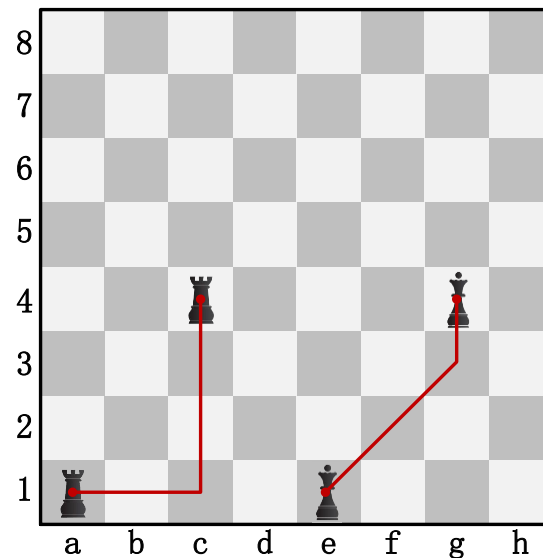
快速查找？先排序再查找——K-D树

距离和相似性度量

特征间的“距离”有很多种，如何选择？如何定义？



——街区距离



——切比雪夫距离

世界上最遥远的距离，不是生与死，而是我就站在你面前，你却不知道我爱你。

世界上最遥远的距离，不是我就站在你面前你却不知道我爱你，而是明明知道彼此相爱，却又不能在一起；

——泰戈尔，反正不是欧氏距离

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”：

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

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

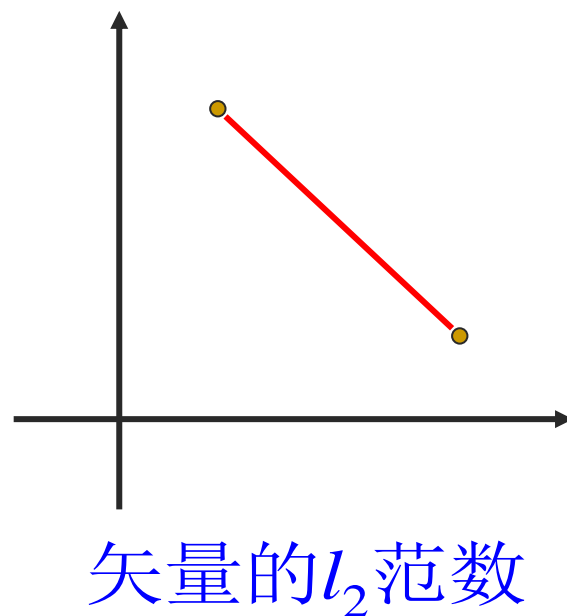
■ 自反性： $d(\mathbf{x}, \mathbf{y}) = 0$, 当且仅当 $\mathbf{x} = \mathbf{y}$

■ 三角不等式： $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$

常用的距离函数

■ 欧氏距离: (Euclidean Distance)

$$\begin{aligned}d(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|_2 \\&= \left[\sum_{i=1}^n (x_i - y_i)^2 \right]^{\frac{1}{2}} \\&= \left[(\mathbf{x} - \mathbf{y})^t (\mathbf{x} - \mathbf{y}) \right]^{\frac{1}{2}}\end{aligned}$$



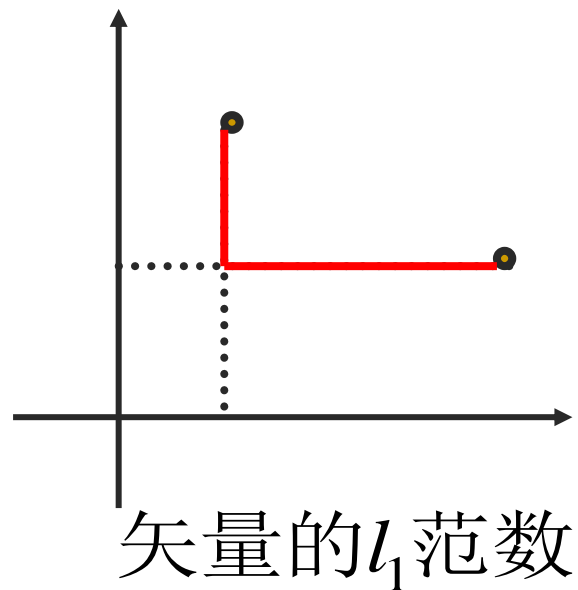
$$\mathbf{x}^t \mathbf{y} = \sum_{i=1}^n x_i y_i \text{ 是 } \mathbf{x} \text{ 与 } \mathbf{y} \text{ 之间的内积}$$

常用的距离函数

■ 街市距离：

(**Manhattan/city block/taxicab distance**)

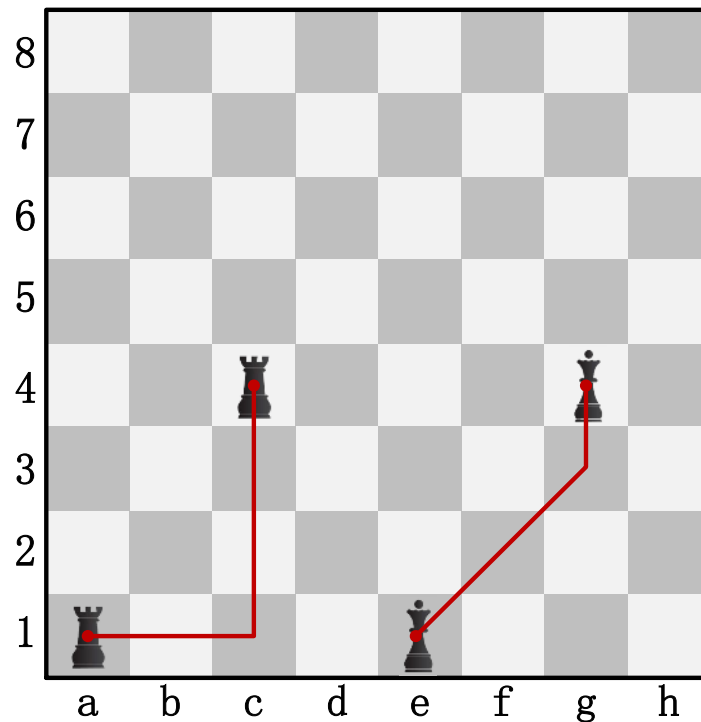
$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|_1 \\ &= \sum_{i=1}^n |x_i - y_i| \end{aligned}$$



常用的距离函数

■ 切比雪夫距离： (Chebyshev Distance)

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



常用的距离函数

■ 闵可夫斯基距离（**Minkowski Distance**）

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

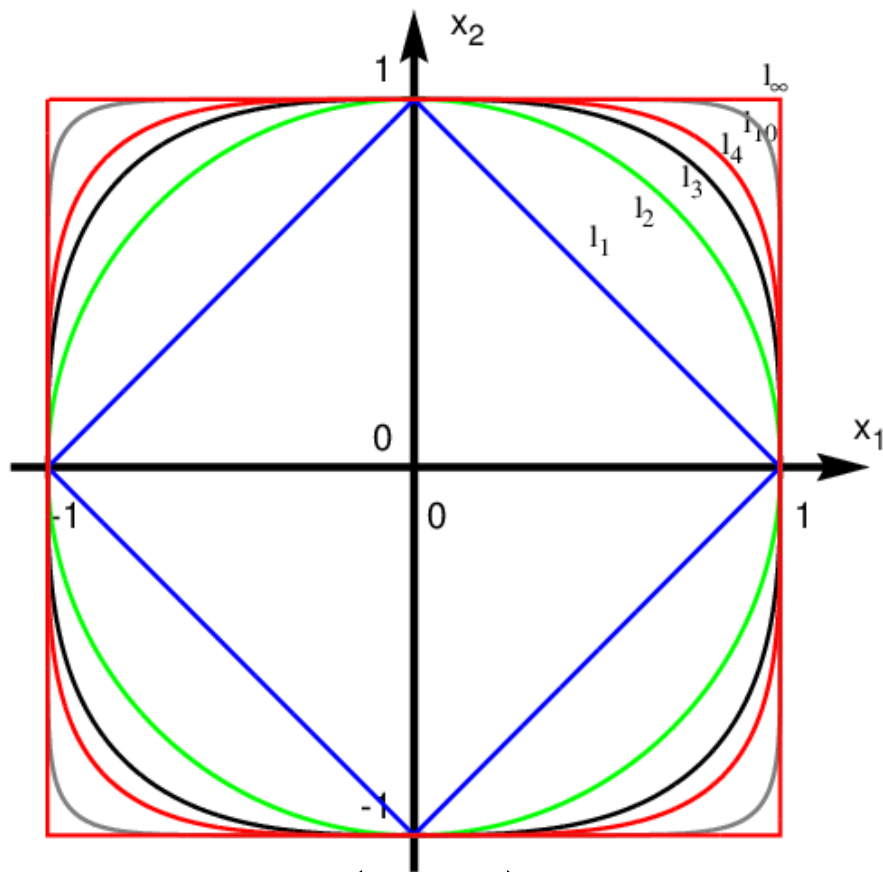
$q = 1$: 街市距离

$q = 2$: 欧氏距离

$q = \infty$: 切比雪夫距离

闵氏距离具有平移不变性旋转

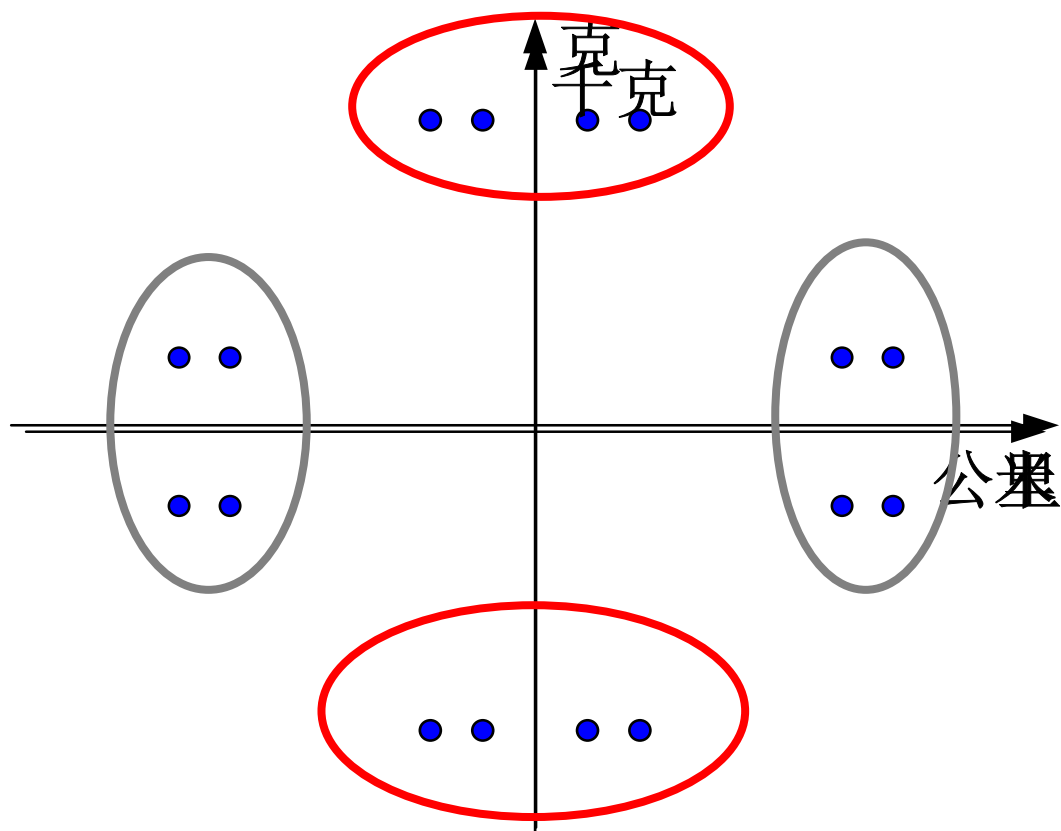
仅当 $q=2$ 时，具有旋转不变性



$$d(\mathbf{x}, 0) = 1$$

样本规格化

■ 特征量纲的影响（缩放坐标轴）



样本规格化

- 使样本每一维特征都分布在相同或相似的范围內，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：假设各维特征服从均匀分布将每一维特征都平移和缩放到 $[0,1]$ 內

1) 计算样本集每一维特征的最大、最小值

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

2) 平移和缩放样本的每一维特征：

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

样本规格化

- 使样本每一维特征都分布在相同或相似的范围内，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2-高斯缩放：假设每一维特征都符合高斯分布，平移、缩放为标准高斯分布

1) 计算每一维特征的均值和标准差：

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

进一步考虑各维特征间的协方差关系？
——马氏距离

2) 规格化每一维特征：

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

加权距离

关键问题：
如何确定每一
维特征的权重

- 计算距离时为“不同特征”引入“不同权重”

- 克服量纲影响
- 体现不同特征重要性

- 样本规格化，可以看做加权距离的特例：

加权欧氏距离：
$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^d w_j (x_j - y_j)^2 \right)^{\frac{1}{2}}, \quad w_j \geq 0$$

均匀缩放：
$$x'_{ij} = \frac{x_{ij} - x_{j\min}}{x_{j\max} - x_{j\min}} \quad \rightarrow \quad w_j = \frac{1}{(x_{j\max} - x_{j\min})^2}$$

高斯缩放：
$$w_j = \frac{1}{s_j^2} \quad \rightarrow \quad x'_{ij} = \frac{x_{ij} - \mu_j}{s_j}$$

汉明距离 (Hamming Distance)

- 二值矢量 $\mathbf{x}, \mathbf{y} \in \{0,1\}^d$ (每个元素只取0或1)
计算两个矢量对应位置元素不同的数量

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d (x_j - y_j)^2$$

例: $\mathbf{x} = (1, 1, 0, 0, 1, 1, 1)^t$, $\mathbf{y} = (1, 0, 0, 0, 0, 0, 1)^t$

$$d(\mathbf{x}, \mathbf{y}) = 3$$

2.2.2 相似性度量

- 衡量相似度，不一定需要距离，可以选择更直接的方法衡量相似度
 - 两向量的夹角——角度相似性
 - 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：

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

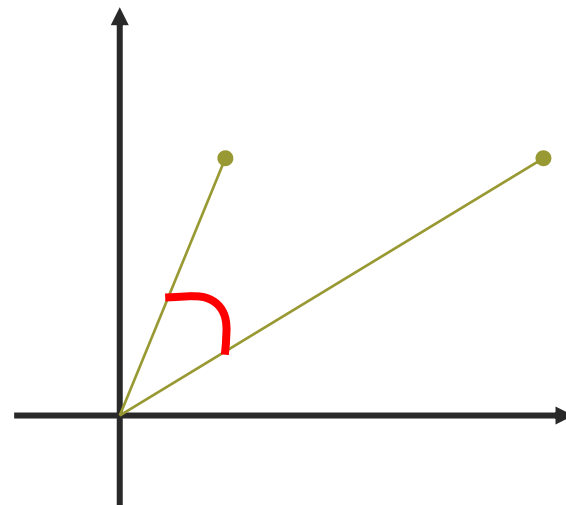
■ 角度相似性

- 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。

$$s(\mathbf{x}, \mathbf{y}) = \cos \theta_{\mathbf{xy}} = \frac{\mathbf{x}^t \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$$

将向量归一为单位向量后做内积。

$$= \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2} \cdot \sqrt{\sum_{i=1}^d y_i^2}}$$



■ 相关系数

- 认为矢量 \mathbf{x} 、 \mathbf{y} 分别来自于两个样本集

样本集的均值分别为 $\mu_{\mathbf{x}}$ 、 $\mu_{\mathbf{y}}$ ，相关系数定义为：

$$s(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_{\mathbf{x}})^t (\mathbf{y} - \mu_{\mathbf{y}})}{\|\mathbf{x} - \mu_{\mathbf{x}}\| \cdot \|\mathbf{y} - \mu_{\mathbf{y}}\|} = \frac{\sum_{i=1}^d (x_i - \mu_{xi})(y_i - \mu_{yi})}{\sqrt{\sum_{i=1}^d (x_i - \mu_{xi})^2} \cdot \sqrt{\sum_{i=1}^d (y_i - \mu_{yi})^2}}$$

- 将矢量 \mathbf{x} 、 \mathbf{y} 视为两个一维信号

$$\mu_x = \frac{1}{d} \sum_{i=1}^d x_i, \quad \mu_y = \frac{1}{d} \sum_{i=1}^d y_i, \quad \mathbf{e}: \text{所有元素均为1的} d \text{维矢量}$$

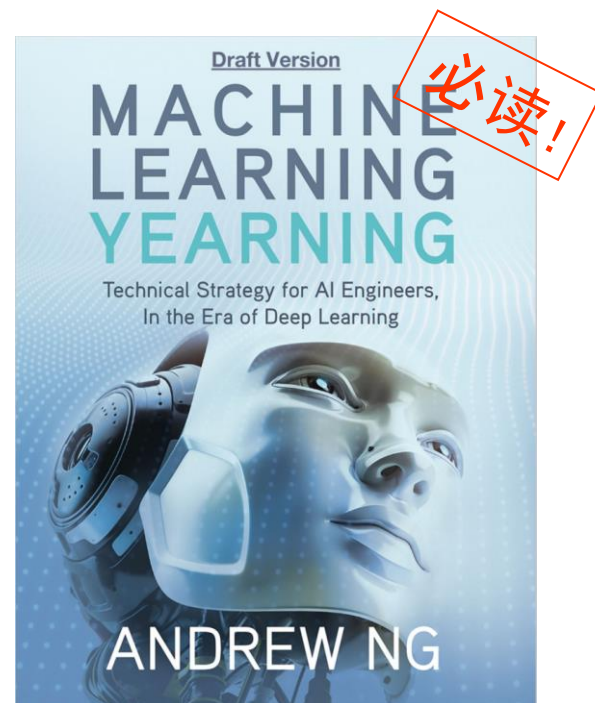
$$s(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x \mathbf{e})^t (\mathbf{y} - \mu_y \mathbf{e})}{\|\mathbf{x} - \mu_x \mathbf{e}\| \cdot \|\mathbf{y} - \mu_y \mathbf{e}\|} = \frac{\sum_{i=1}^d (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^d (x_i - \mu_x)^2} \cdot \sqrt{\sum_{i=1}^d (y_i - \mu_y)^2}}$$

| Name | Value | 说明 |
|----------------------|--------------------|-----------------|
| Distance | euclidean | 欧几里得距离度量 |
| | cityblock | 街市距离度量 |
| | chebychev: | 切比雪夫距离度量; |
| | minkowski: | 闵可夫斯基距离度量; |
| | seuclidean: | 规格化样本的欧氏距离度量 |
| | hamming | 汉明距离 |
| | cosine | 角度相似性度量 |
| | correlation | 相关系数 |
| DistParameter | minkowski | 闵可夫斯基距离中的指数p |
| | seuclidean | 规格化样本的标准差 |
| NumNeighbors | k | K-近邻算法参数 |

| Name | Value | |
|-------------------|----------|--|
| 分类策略 BreakTies | nearest | 先用K-近邻原则分类，如果出现数量相同的情况则以最近邻的原则分类 |
| | random | 先用K-近邻原则分类，如果出现数量相同的情况则在样本数量最多的几个类别中随机的选择一个； |
| | smallest | 先用K-近邻原则分类，如果出现数量相同的情况则选择类别标号最小的类别； |

模式识别/机器学习系统开发

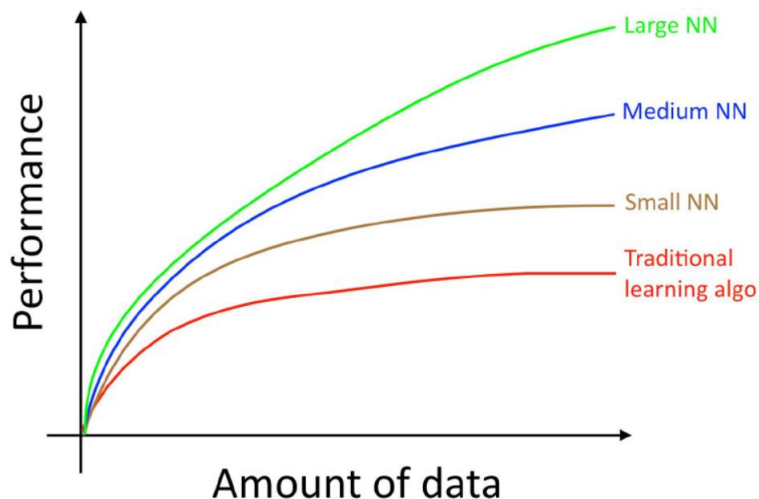
- 样本集的构建
- 分类器性能的评价
- 迭代改进
- 误差与偏差



deeplearning-ai.github.io

样本集的构建

- 穷则“特征工程”，富则“Deep Learning”



- 保证验证、测试集与实际应用（分布）的一致性

必须!

- 样本集的划分

训练集 (training set)

开发/验证集 (development/validation set)

测试集 (test set)

- 开发集、测试集的划分

- 开发集够大，才能检测小幅改进
- 样本够多时，开发、测试集比例可以降低

性能评价与迭代优化

- 根据问题特点，设定计算资源、存储空间、运行时间的可接受范围

| Classifier | Accuracy | Running time |
|------------|----------|--------------|
| A | 90% | 80ms |
| B | 92% | 95ms |
| C | 95% | 1,500ms |

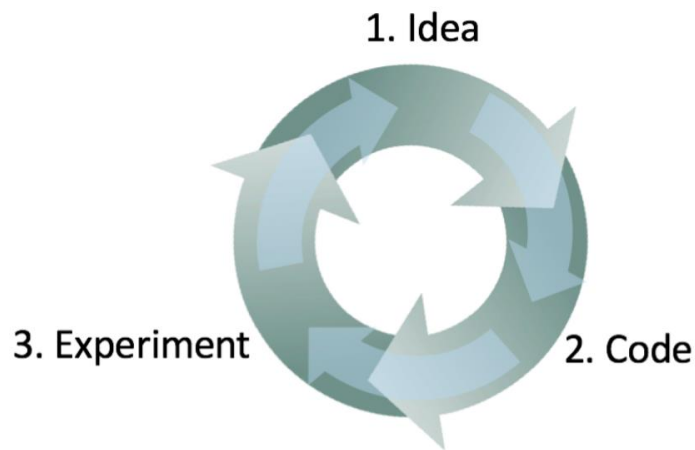
- 根据问题特点，设定优化关键指标
 - 敏感性、特异性、召回率和准确率、F1 score
- 误差分析
- 迭代优化

迭代优化！

■ 第一个想到的点子一般都行不通！

- 理想主义者：让我多想想，规划一个好方案再动手。
- 实践主义者：先设计一个基本框架，快速迭代！

虚伪！



- （几天之内）尽快构建一个简单的原型。
- 然后就会看到新的线索，告诉你该选择什么样的方向去改进原型的性能。
- 在下一次迭代中，可以根据这些线索改进系统，并构建系统的下一个版本。一次接一次，循环往复。
- 建立特定的开发集和度量指标，快速迭代！
- 每次0.1%改进，积少成多！

分类器性能评价

根据问题特点、设定计算资源、存储空间、运行时间的可接受范围，
设定优化关键指标
针对关键指标进行迭代优化

■ 评价指标

- 拒识: 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， m_r 个被拒识， m_e 个被分类错误

错误率: $P_e \approx \frac{m_e}{m - m_r}$

拒识率: $P_r \approx \frac{m_r}{m}$

分类器性能评价

■ 敏感性与特异性（医学相关领域）

- 敏感性（真阳率）：患者被诊断出来的比率

$$P_s \approx \frac{a}{a+b}$$

| | | 分类结果 | |
|------|-----|------|-----|
| | | 阳性 | 阴性 |
| 实际类别 | 患者 | a | b |
| | 正常人 | c | d |

- 特异性：正常人不被误诊的比率

$$P_n \approx \frac{d}{c+d}$$

- 假阳率：正常人被误诊的比率

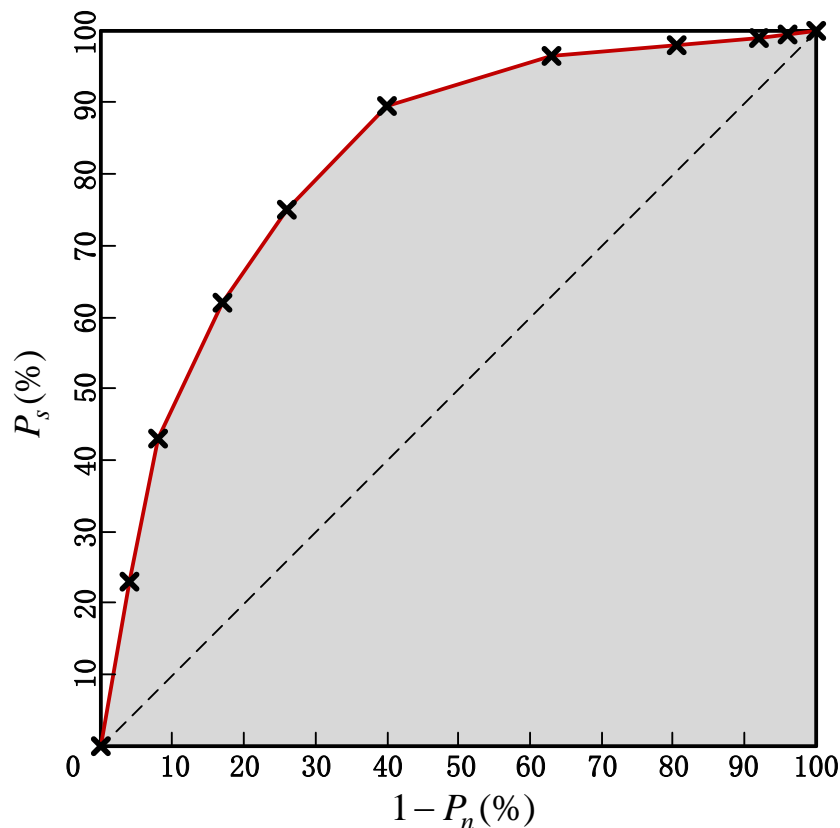
$$1-P_n \approx \frac{c}{c+d}$$

分类器性能评价

ROC曲线

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------|---|------|------|------|------|------|------|------|------|------|-----|
| P_s | 0 | 23.1 | 43.8 | 62.2 | 72.3 | 89.6 | 96.1 | 98 | 98.9 | 99.5 | 100 |
| $1-P_n$ | 0 | 4.3 | 8.6 | 17.1 | 25.9 | 40.4 | 63.5 | 80.2 | 92.3 | 96.5 | 100 |

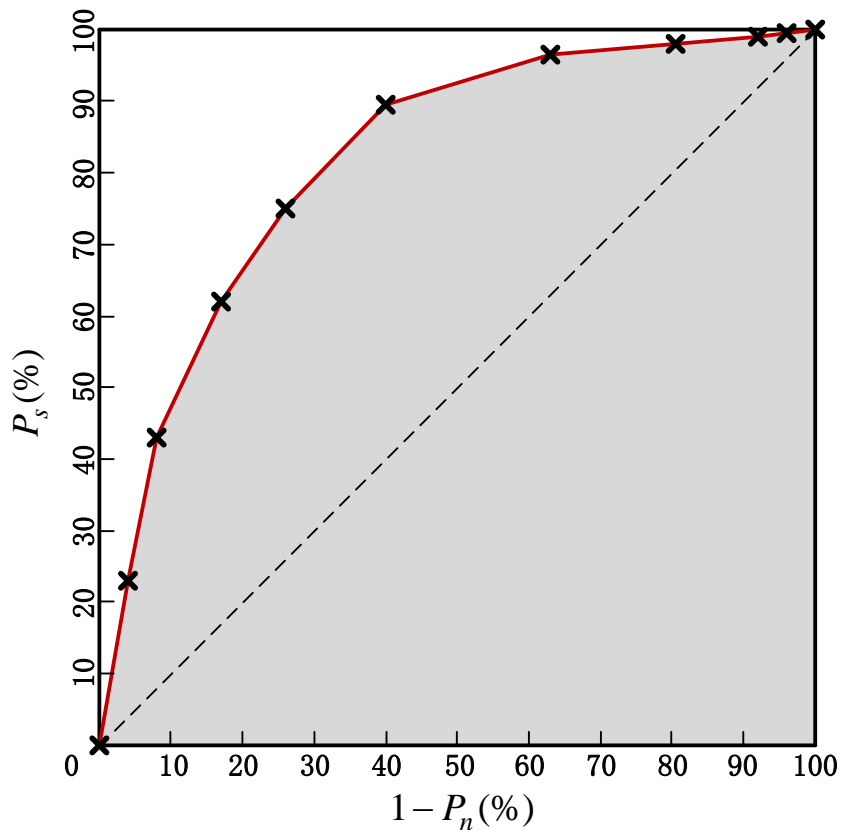
纵轴为敏感性



横轴为假阳率

分类器性能评价

■ ROC曲线



- 判断不同分类方法的优劣：
ROC曲线下方的面积越大性能越好。

- 选择分类器参数：
一般应用： $P_s = 1 - P_n$

敏感性要求高的场合：

$$P_s > 95\%$$

分类器性能评价

■ 召回率和准确率（信息检索）

| | | 分类结果 | |
|------|-----|------|------|
| | | 检索到 | 未检索到 |
| 实际类别 | 相关 | a | b |
| | 不相关 | c | d |

○ 召回率（查全率）：相关的信息中被检索出来的比例

$$P_s \approx \frac{a}{a+b} \quad \text{=敏感性}$$

○ 准确率：检索到的信息中，与主题相关的比例

$$P \approx \frac{a}{a+c}$$

分类器性能评价

■ 召回率和准确率——调和平均（F1 score）

○ 兼顾召回率和准确率，使用二者的调和平均

$$F_1 = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2RP}{R + P}$$

使用单值评估指标将帮助你更快速地作出决定。它能给出一个清晰明了的分类器性能排名，从而帮助团队明确后续的改进方向。

——吴恩达

分类器性能评价

■ 评价方法——如何进行性能评价试验？

○ 两分法：随机将训练样本集划分为不相交的两个子集

- 两子集分别用于训练、测试。重复 k 次取均值
- 只能用一部分样本学习分类器，另一部分样本测试性能

○ 交叉验证：随机地分成不相交的 k 个子集

- 每个子集中的样本数量相同。使用 $k-1$ 个训练，剩余样本测试。
- 以 k 次的平均值作为分类器的性能评价指标

分类器性能评价

■ 评价方法——如何进行性能评价试验？

○ Bootstrap方法

1. 对样本集 D 中有放回地抽取 n 个样本，
组成 Bootstrap 样本集 A 、 B （样本可能重复）；
2. 用集合 A 训练、 B 测试得到性能指标估计；
3. 上述过程重复 k 次，取 k 次的平均值作为分类器性能的评价。

- 开发集和测试集的数据，应当与你未来计划获取并对其进行良好处理的数据有着相同的分布，而不一定和训练集的数据分布一致。
- 开发集和测试集的分布应当尽可能一致
- 为你的团队选择一个单值评估指标进行优化
- 当需要考虑多项目标时，不妨将它们整合到一个表达式里（比如对多个误差指标取平均），或者设定满意度指标和优化指标
- 机器学习是一个高度迭代的过程：在出现最终令人满意的方案之前，你可能要尝试很多想法。
- 拥有开发集、测试集和单值评估指标可以帮助你快速评估一个算法，从而加速迭代进程。
- 当你要探索一个全新的应用时，尽可能在一周内建立你的开发集、测试集和评估指标；而在已经相对成熟的应用上，可以考虑花费更长的时间来执行这些工作。

- 传统的 **70% / 30%** 训练集/测试集划分对于大规模数据并不适用，实际上，开发集和测试集的比例会远低于 **30%**.
- 开发集的规模应当大到能够检测出算法精度的细微改变，但也不需要太大；测试集的规模应该大到能够使你能对系统的最终性作出一个充分的估计。
- 当开发集和评估指标已经不能提供正确的导向时，尽快修改：
 - 如果算法在开发集上过拟合，则需要获取更多的开发集数据。
 - 如果开发集与测试集的数据分布和实际数据分布不同，则需要获取新的开发集和测试集。
 - 如果评估指标无法对最重要的任务目标进行度量，则需要修改评估指标。

误差分析

■ 检查被算法误分类的开发集样本，评估想法

- 收集开发集中被误分类的样本
- 人为查看误分样本（尤其是视觉问题）

想法可以天马行空、分析必须脚踏实地！
观察、记录错分样本，分析原因，修正思路。

| 图像 | 狗 | 大猫 | 模糊 | 备注 |
|-------|-----|-----|-----|----------------|
| 1 | ✓ | | | 罕见美国比特犬 |
| 2 | | | ✓ | |
| 3 | | ✓ | ✓ | 狮子；雨天在动物园拍摄的图片 |
| 4 | | ✓ | | 树木后的美洲豹 |
| ... | ... | ... | ... | ... |
| 占全体比例 | 8% | 43% | 61% | |

特征可视化
中间结果可视化
运动过程可视化
误差曲线可视化。。。

问题发现——找到造成这些误差的原因。

问题排序——找到主要矛盾，确定问题优先级

问题探索——判断哪些想法更有前景，获得探索新方向的灵感。

开发集的拆分——Eyeball 与 Blackbox

- 将大型开发集分为两部分：
 - **Eyeball**: 人工检查
 - **Blackbox**: 只用于分类效果评估与调参，不人工观测

提高工作效率——观测部分样本就能发现原因

预警过拟合——当**Eyeball**上的性能比**Blackbox**上高时，意味过拟合，此时可更新**Eyeball**

在视听觉问题中，观察、分析错分样本是提升“洞察力”的关键！

- 当你开始一个新项目，尤其是在一个你不擅长的领域开展项目时，很难正确预判出最有前景的方向。
- 所以，**不要在一开始就试图设计和构建一个完美的系统**。相反，应尽可能快（例如在短短几天内）地构建和训练一个系统雏形。然后使用误差分析法去帮助你识别出最有前景的方向，并据此**不断迭代**改进你的算法。
- 通过手动检查约 **100** 个被算法错误分类的开发集样本来执行误差分析，并计算主要的错误类别。使用这些信息来确定优先修正哪种类型的错误。
- 考虑将开发集分为人为检查的 **Eyeball** 开发集和非人为检查的 **Blackbox** 开发集。如果在 **Eyeball** 开发集上的性能比在 **Blackbox** 开发集上好很多，说明你已过拟合 **Eyeball** 开发集，下一步应该考虑为其获取更多数据。
- **Eyeball** 开发集应该足够大，以便于算法有足够多的错误分类样本供你分析。对大多数应用来说，含有**1000-10000**个样本的 **Blackbox** 开发集已足够。
- 如果你的开发集不够大，无法按照这种方式进行拆分，那么就使用 **Eyeball** 开发集来执行人工误差分析、模型选择和调超参。

误差的构成：偏差与方差

- 在训练集和开发集上分别统计错误率：

训练错误率 = 15%
开发错误率 = 30% → 萌新？（high bias and high variance）

训练错误率 = 1%
开发错误率 = 11% → 过拟合（overfitting）

训练错误率 = 15%
开发错误率 = 16% → 欠拟合（underfitting）

训练错误率 = 0.5%
开发错误率 = 1% → 成功！

误差的构成：偏差与方差

非正式
说法

- 偏差 (**bias**)：训练集上的错误率
偏差=最优错误率（贝叶斯错误率）+可避免偏差
- 方差 (**variance**)：
“开发集错误”与“训练集错误”的差值
- 总误差（均方误差**MSE**）=偏差+方差
- 将误差分为偏差与方差，能够更有针对性的改进
 - 较高可避免偏差？——加大模型规模
 - 较高方差？——增加训练集数据量
 - 可避免偏差为负？——**over-memorized**
 - 加大模型规模是否过拟合？——正则化、**dropout**

减少可避免偏差的技术

- 加大模型规模：使算法更好地拟合训练集，从而减少偏差。当这样做增大方差时，加入正则化。
- 对训练集进行误差分析，根据结果修改输入特征：
 - 增加额外的特征，消除某个特定类别的误差。这些新的特征对处理偏差和方差都有所帮助。
 - 添加更多的特征将增大方差；可以加入正则化来抵消方差的增加。
- 减少或者去除正则化（**L2 正则化**，**L1 正则化**，**dropout**）
 - 减少可避免偏差，但会增大方差。
- 修改模型架构（比如神经网络架构）使之更适用于你的问题
 - 这将同时影响偏差和方差。

有一种方法并不能奏效？

- 添加更多的训练数据：这项技术可以帮助解决方差问题，但它对于偏差通常没有明显的影响。

减少“可避免偏差”的技术

- 加大模型规模：使算法更好地拟合训练集，从而减少偏差。当这样做增大方差时，加入正则化。
- 对训练集进行误差分析，根据结果修改输入特征：
 - 增加额外的特征，消除某个特定类别的误差。这些新的特征对处理偏差和方差都有所帮助。
 - 添加更多的特征将增大方差；可以加入正则化来抵消方差的增加。
- 减少或者去除正则化（**L2 正则化**，**L1 正则化**，**dropout**）
 - 减少可避免偏差，但会增大方差。
- 修改模型架构（比如神经网络架构）使之更适用于你的问题
 - 这将同时影响偏差和方差。

有一种方法并不能奏效？

- 添加更多的训练数据：这项技术可以帮助解决方差问题，但它对于偏差通常没有明显的影响。

减少方差的技术

必须的：

- 添加更多的训练数据：最简单最可靠策略。
- 加入提前终止（**Early stopping**）：免费午餐？

常用的：

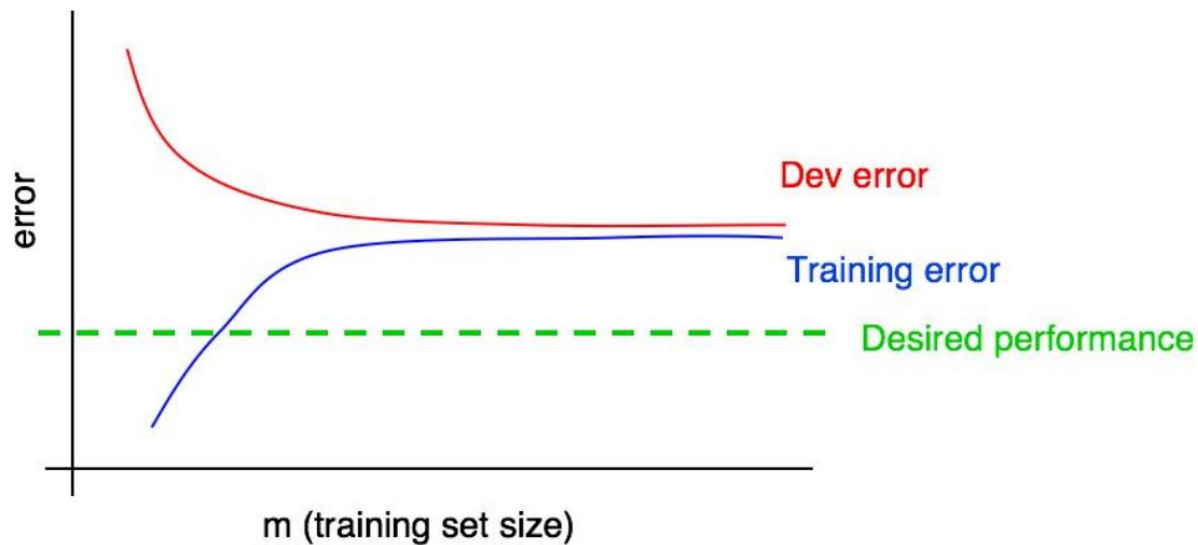
- 计算成本不敏感时进行正则化
- 样本少时进行特征选择
- 降低计算成本时，减少模型规模（谨慎使用，会增加偏差）

同时影响偏差和方差的：

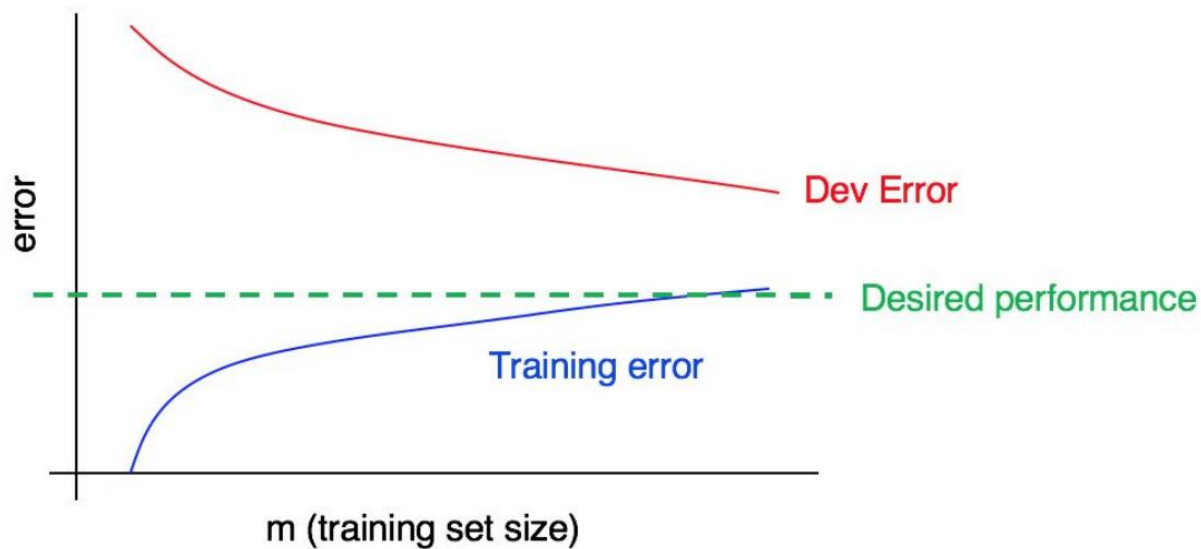
- 根据误差分析结果修改输入特征
- 修改模型架构（比如神经网络架构）使之更适用于你的问题

学习曲线

高偏差



高方差



模块化的优点：

模块化

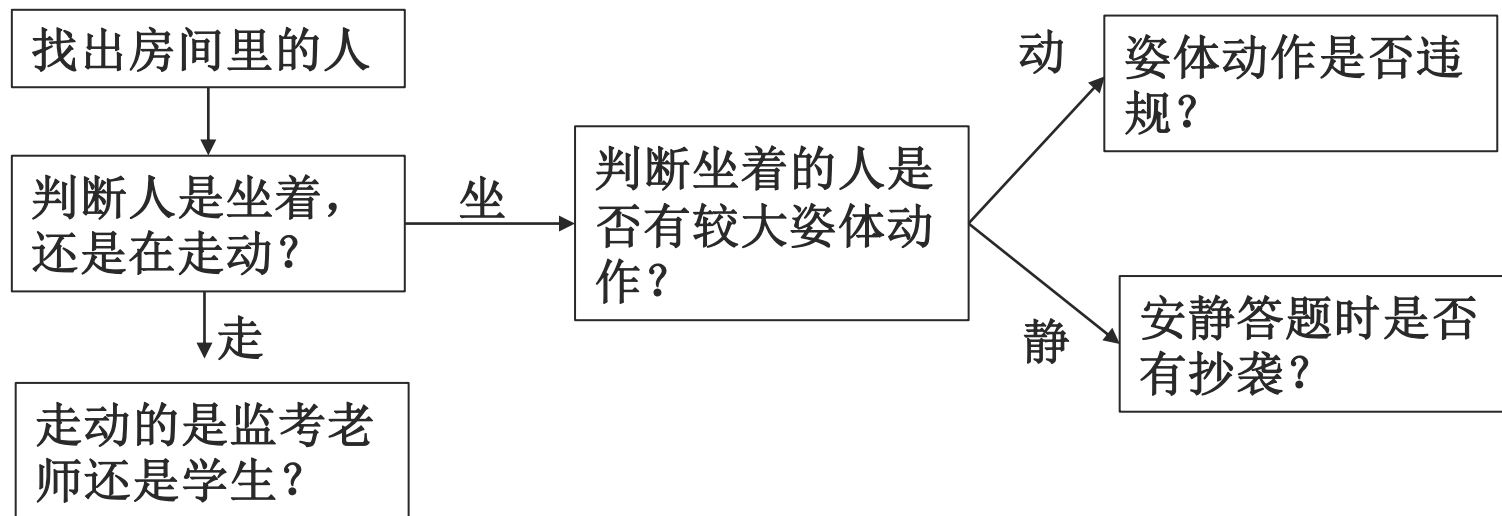
端到端



模式识别系统-模块化流水线vs端到端？

模块化流水线：把一个复杂问题拆分为一系列简单问题：
每个问题有更多先验知识，只需从少量数据中学习，
对每个组件模块进行误差分析

智能监考：



模式识别系统-端到端

有些问题难以拆分组件
直接学习更为丰富的输出？

$x =$



$y =$ “A yellow bus driving down a road with green trees and green grass in the background.”

阅读“机器学习训练的秘籍”回答以下问题

- 总结一下开发集、测试集的划分原则
- 你在ImageNet数据集上训练、测试结果很好，但是对你自己拍摄的图片效果很差。你觉得该怎么办？
- 对比讨论“端到端”和“流水线组件”两种思路的优缺点、适用场景

新手练习：手写数字识别

■ UCI 数据集

- Semeion Handwritten Digit Data Set

■ MatLab 或 Python (NumPy, SciPy, Matplotlib, scikit-learn)

■ 分类方法？

- 距离分类器、**SVM**？

■ 如何评价？

- 如何评价识别结果？
- 如何改进？

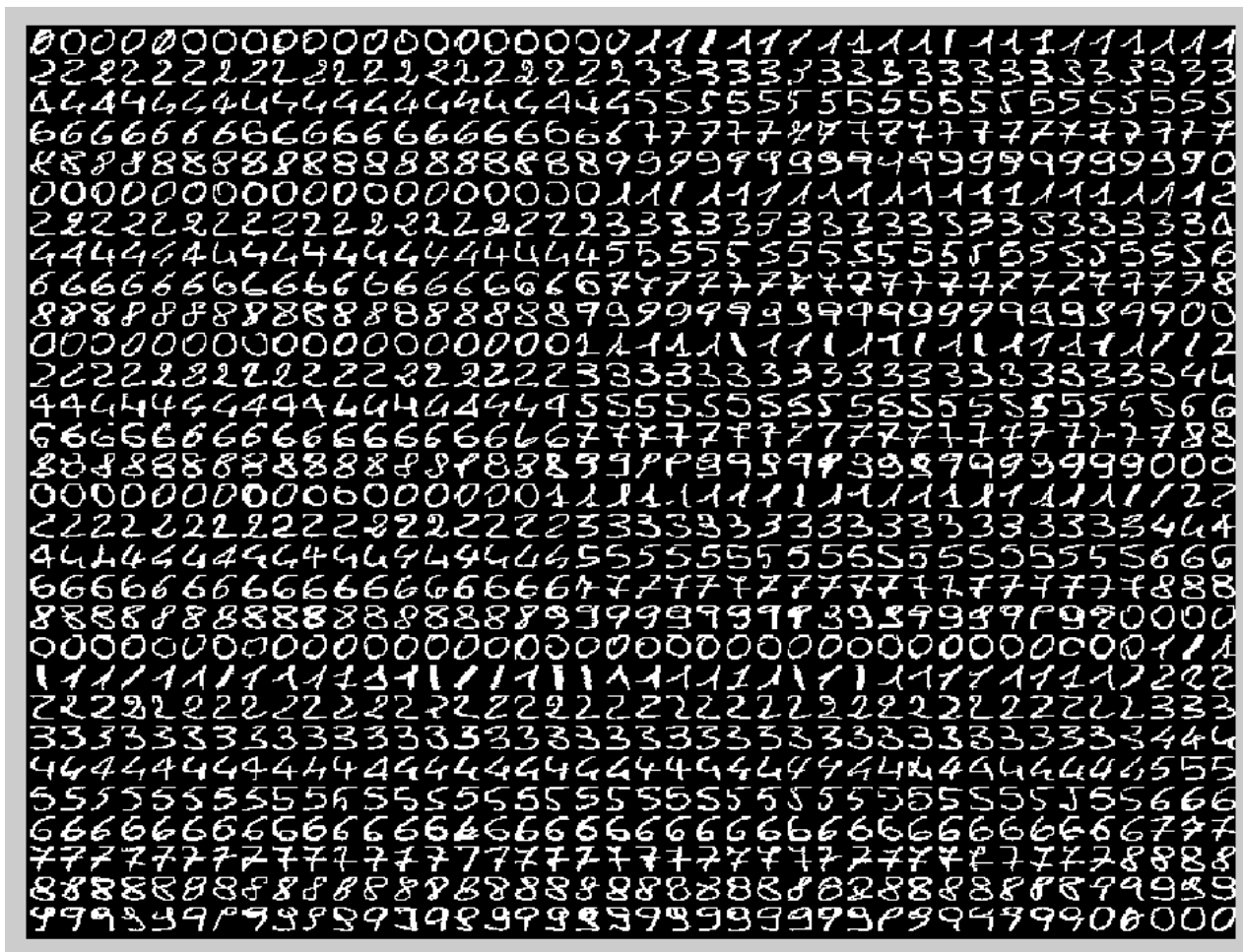
➤ 如果你从来没有做过一个模式识别或机器学习程序，那么请你务必练习一下。

➤ 如果你不熟悉基于**Python**的机器学习库，请开始熟悉。

➤ 高手请忽略😊

UCI 数据集 <http://archive.ics.uci.edu/ml/>

Semeion Handwritten Digit Data Set



新手练习：手写数字识别

■ 用Python实现该字符集的识别

- 代码+注释，规范命名
- 模版匹配，最近邻，**k**近邻
- 如何实验并评价各类方法的效果？
- 你有哪些想法或改进？能使识别率更高？

高手设计：街道违停抓拍

- 为警车设计一套视觉系统，能够自动抓拍街边违停车牌，给出一个初步方案？
 - 传感器、硬件平台、云系统构建
 - 是否需要地理信息系统？
 - 是否需要街景匹配？
 - 什么叫违停？如何用算法定义？
 - 如果太麻烦了，你能否给出一两种典型违停的检测？
 - 对于这种典型状态，可以用什么方法解决？
 - 样本集如何获得？
 - 估计要多少成本？