

数据挖掘与机器学习

潘斌

panbin@nankai.edu.cn

范孙楼227

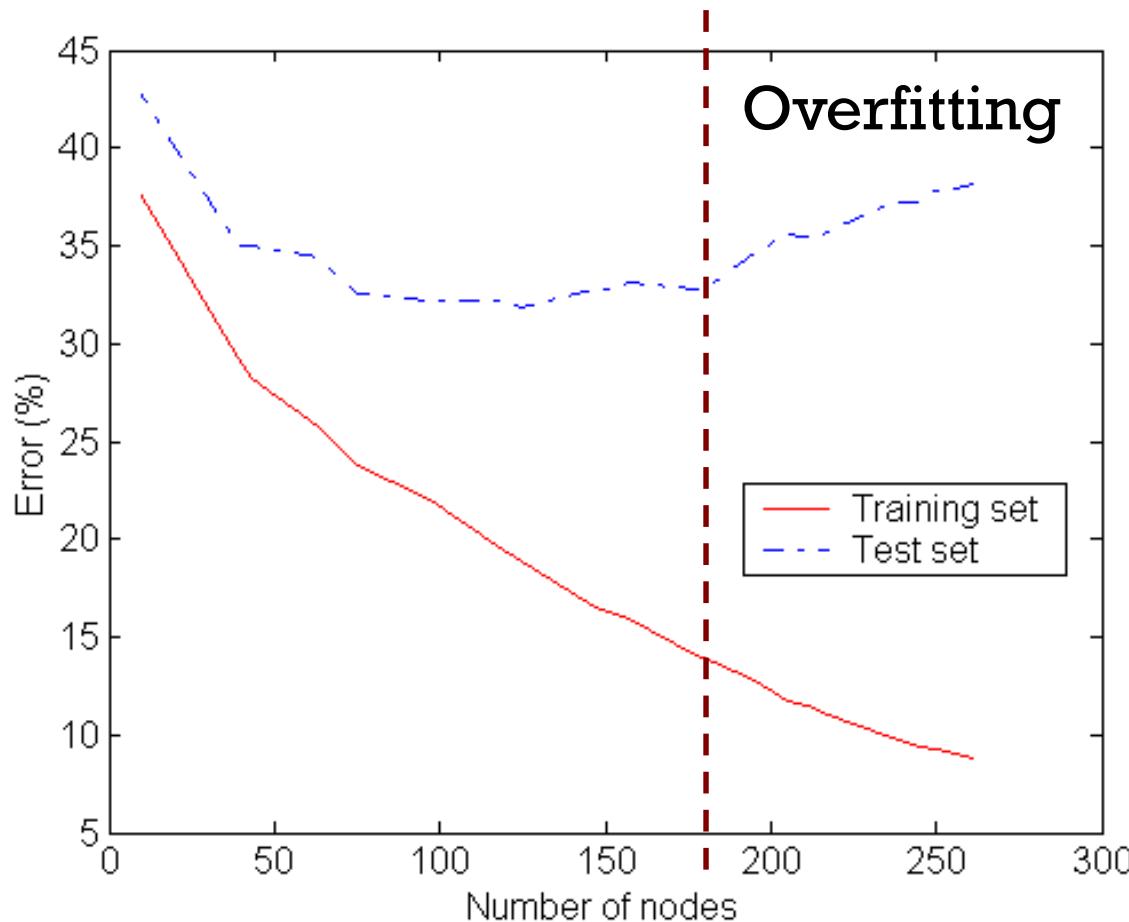
1

上节回顾

- 概念学习的求解方法
 - Find-S
 - 列表后消除算法
 - 候选消除算法
 - 奥卡姆剃刀原理和没有免费午餐原理
- 模型的评估
 - PR曲线
 - ROC曲线



■ 过拟合与欠拟合



Underfitting: when model is too simple, both training and test errors are large



模型M1：锯齿 \cap 绿色 \rightarrow 树叶



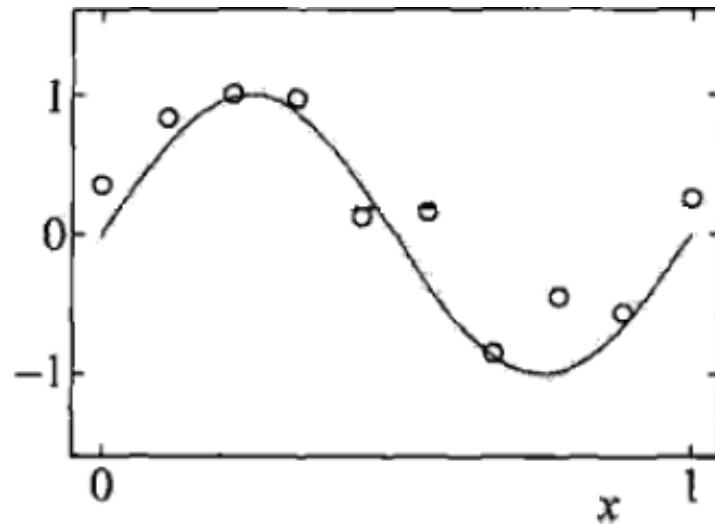
模型M2：绿色 \rightarrow 树叶



例 假设给定一个训练数据集：

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

在 M 次多项式函数中选择一个对已知数据以及未知数据都有很好预测能力的函数.



设 M 次多项式为

$$f_M(x, w) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$

求以下经验风险最小化:

$$L(w) = \frac{1}{2} \sum_{i=1}^N (f(x_i, w) - y_i)^2 = \frac{1}{2} \sum_{i=1}^N \left(\sum_{j=0}^M w_j x_i^j - y_i \right)^2$$

平方
损失



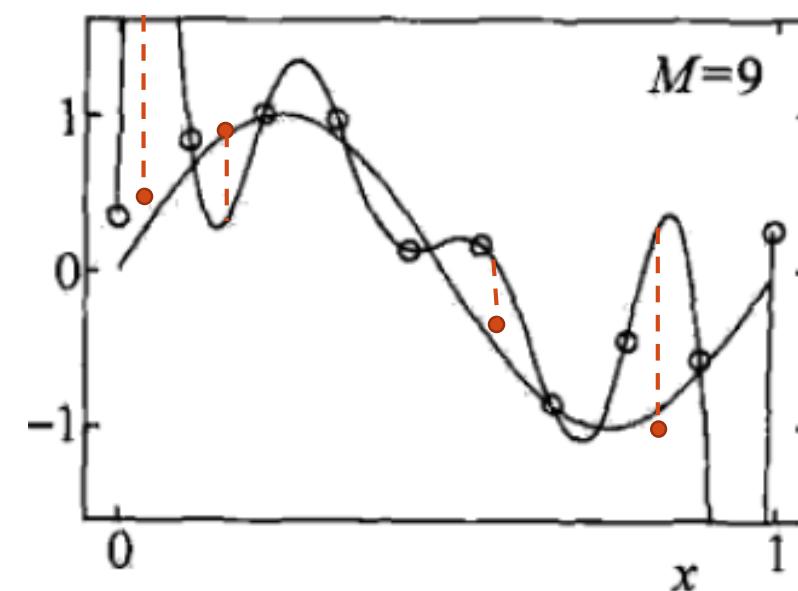
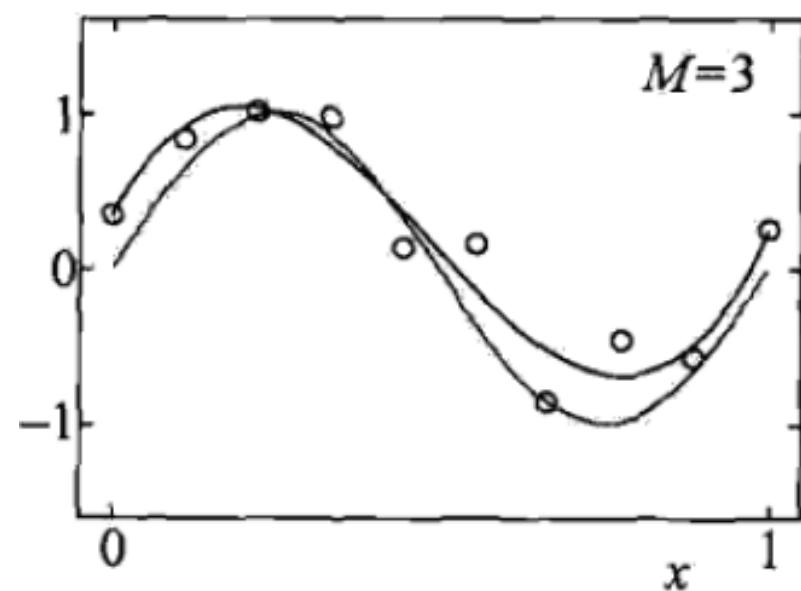
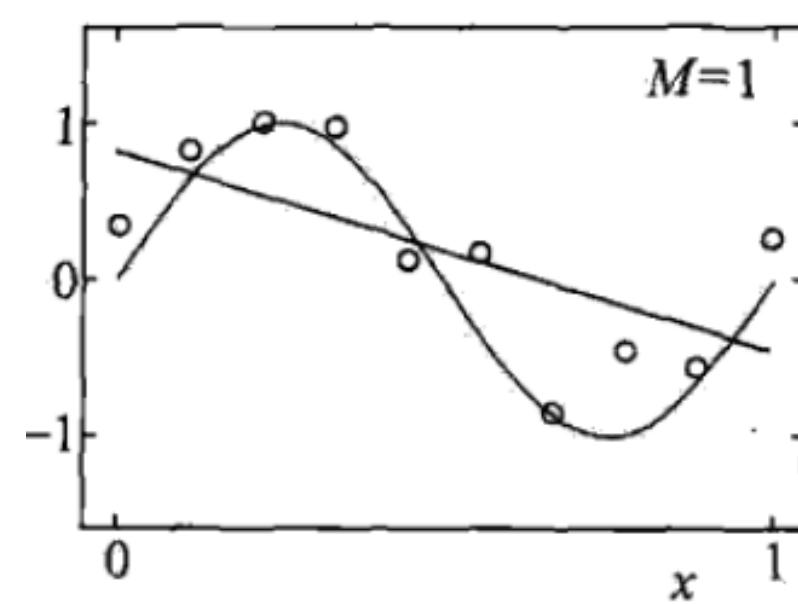
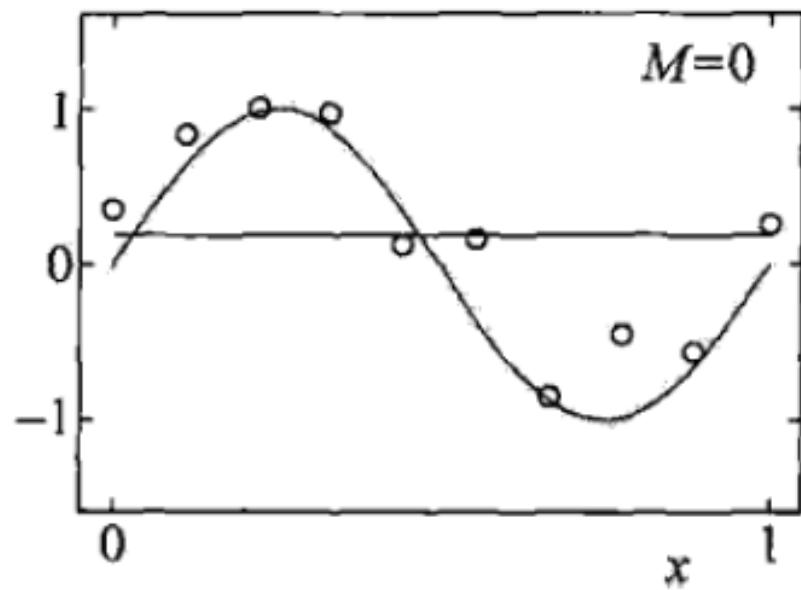
对 w_j 求偏导数并令其为 0， 可得

$$w_j = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^{j+1}}, \quad j = 0, 1, 2, \dots, M$$

于是求得拟合多项式系数 $w_0^*, w_1^*, \dots, w_M^*$.



$$f_M(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^M w_j x^j$$



模型的检验方法——留出法（HOLDOUT）

- 将原始数据集D划分为二不相交集合，一个作为训练集S，一个作为测试集T
- 在训练集上训练出模型
- 在测试集上评估模型的性能
- 二者的划分常依分析者的判断
 - 50-50
 - 2/3作为训练集，1/3作为检验集

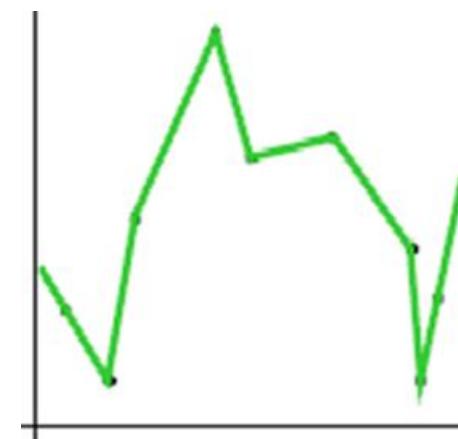
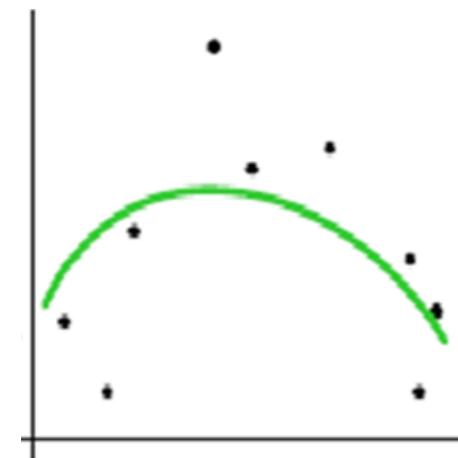
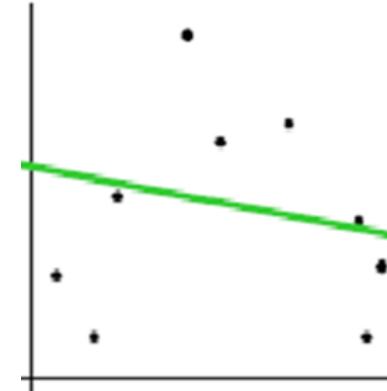
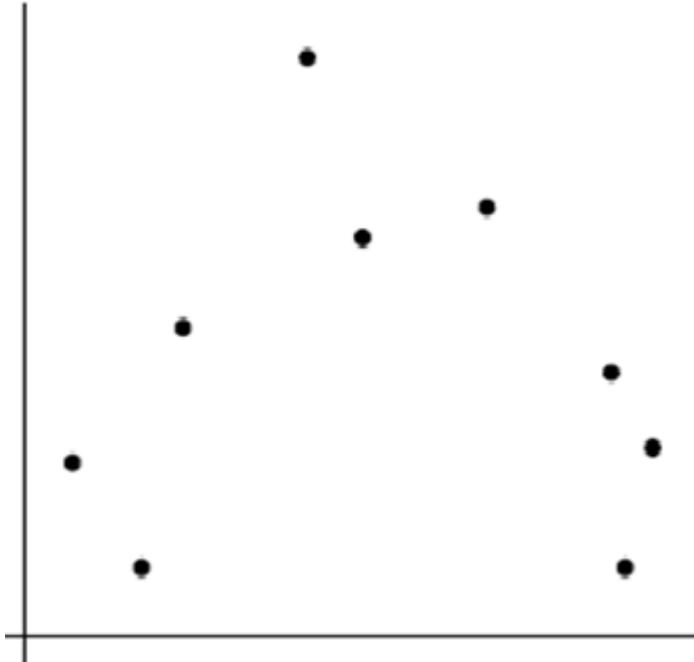


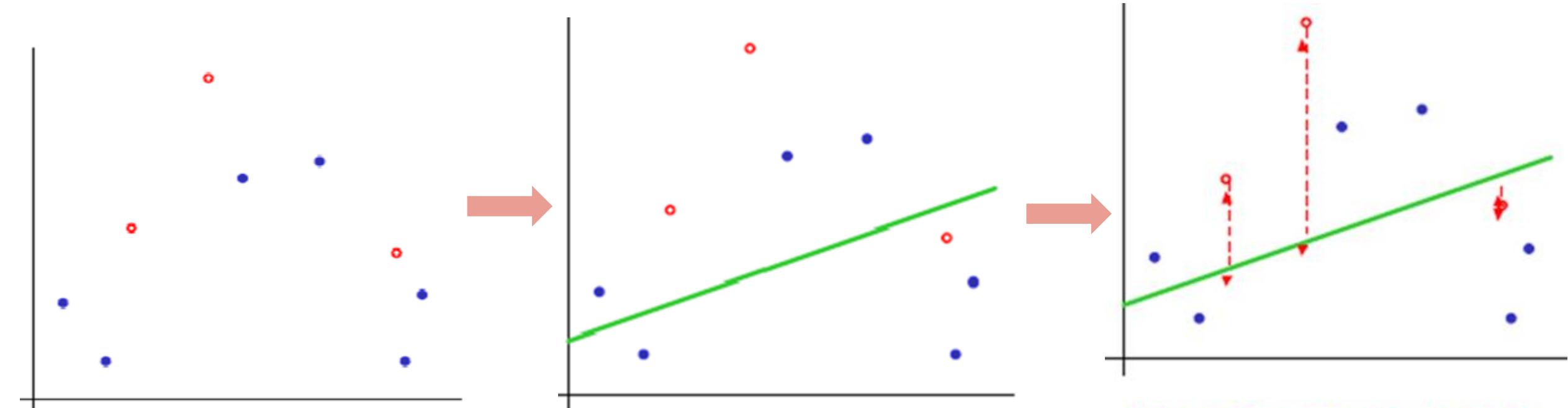
以二分类任务为例, 假定 D 包含 1000 个样本, 将其划分为 S 包含 700 个样本, T 包含 300 个样本, 用 S 进行训练后, 如果模型在 T 上有 90 个样本分类错误, 那么其错误率为 $(90/300) \times 100\% = 30\%$, 相应的, 精度为 $1 - 30\% = 70\%$.

- 划分要尽可能保持原始数据分布的一致性
- 一般采用多次随机划分、重复评估后取平均值



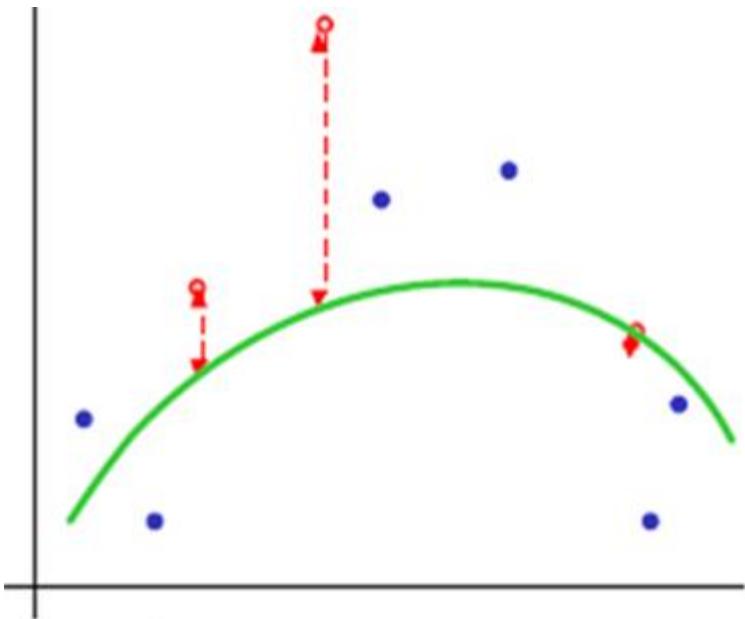
例：回归问题



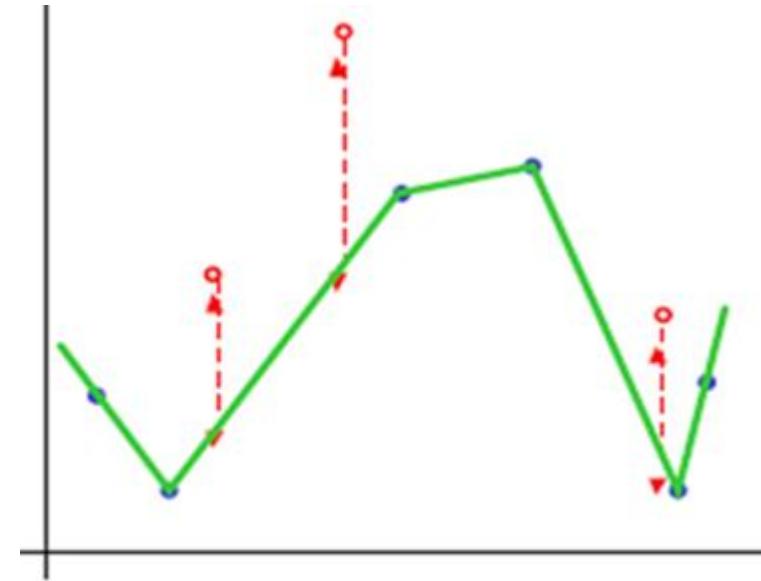


Mean Squared Error = 2.4





Mean Squared Error = 0.9



Mean Squared Error = 2.2

■ 留一法 (Leave-One-Out Cross Validation)

- 记 L 为整个训练样本集的大小。
- 对于第 i 个训练样本，将其取出，对剩下 $L-1$ 个样本进行训练，得到模型，并用第 i 个训练样本对该模型的性能进行测试
- 该过程重复 L 次
- 用这 L 个模型的准确率的平均值作为评价模型性能的指标



$MSE_{LOOCV} = 2.12$

$MSE_{LOOCV} = 0.962$

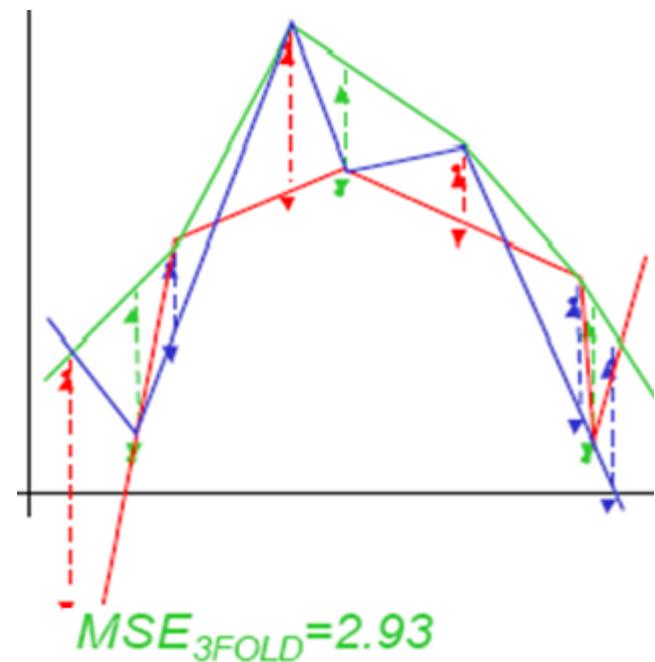
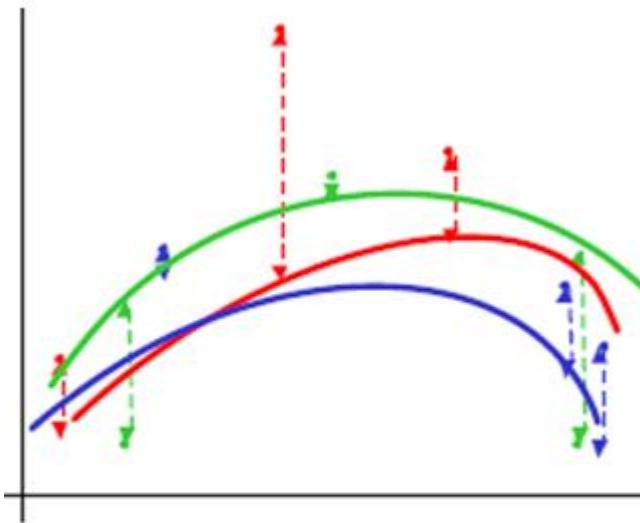
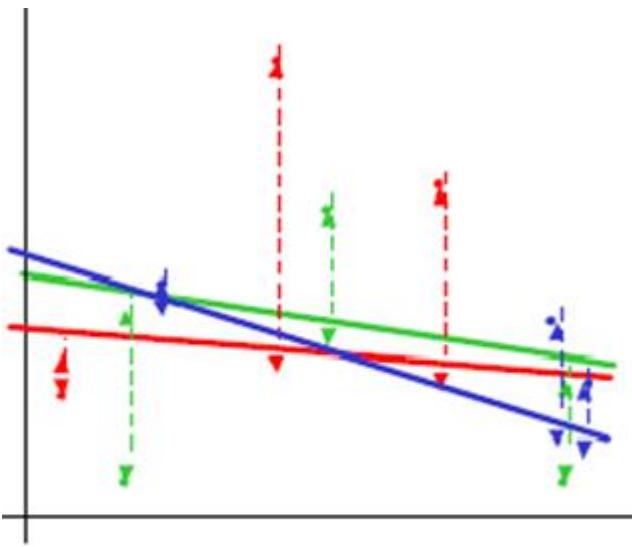
$MSE_{LOOCV} = 3.33$



■ K 折交叉验证法 (K-fold Cross Validation)

- 将训练样本集随机分为K个集合，通常分为K等份
- 对其中的K-1个集合进行训练，得到模型，并用剩下的一个集合对该模型的性能进行测试。
- 该过程重复K次，取K次过程中的测试错误的平均值作为评价模型性能的指标。





自助法 (BOOTSTRAPPING)

- 以自助采样法为基础
 - 数据集 D 中包含 m 个样本
 - 每次随机从 D 中挑选一个样本到 D'
 - 该样本被放回初始数据集 D 中
 - 重复 m 次得到包含 m 个样本的数据集 D'



不同模型（学习器）性能的比较

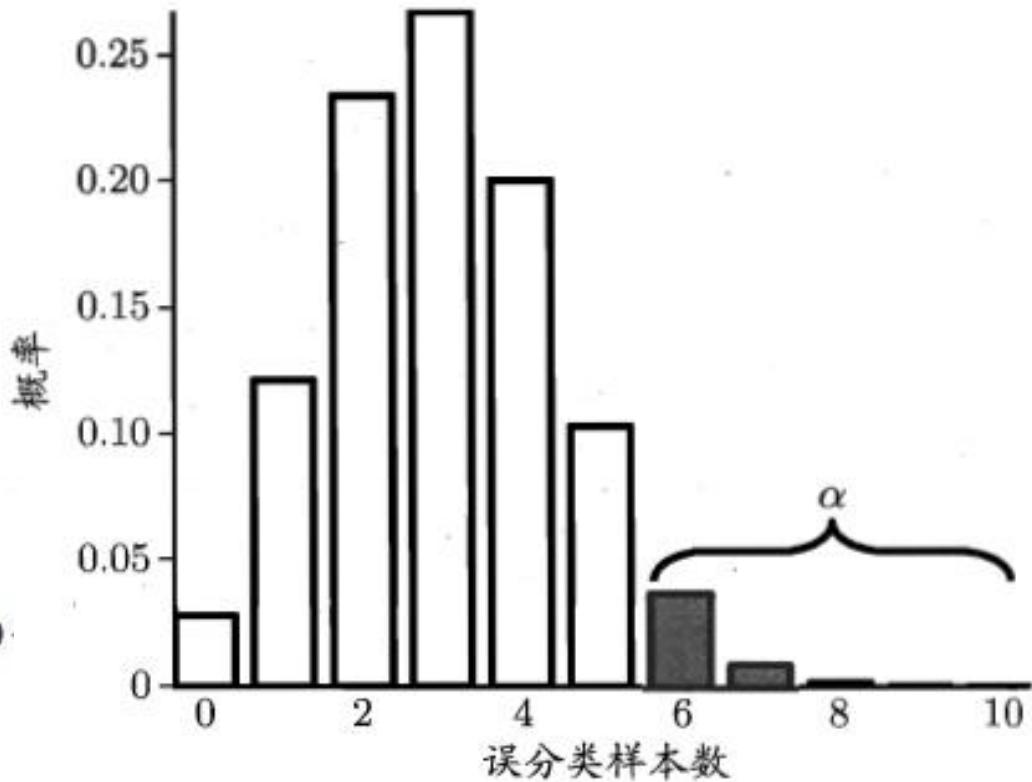
- 考虑以下二个模型：
 - 模型M1: accuracy = 85%, tested on 30 instances
 - 模型M2: accuracy = 75%, tested on 5000 instances
- 模型M1是否优于模型M2?



模型检验

- $H_0: \varepsilon \leq \varepsilon_0$
- 二项检验

二项分布 $B(n, p)$ 在 n 很大时逼近正态分布 $N(np, np(1-p))$



二项分布示意图 ($m = 10, \epsilon_0 = 0.3$)



模型检验

- $H_0: \varepsilon \leq \varepsilon_0$
- 近似正态检验

$$\frac{\frac{x}{m} - \varepsilon_0}{\sqrt{\varepsilon_0(1-\varepsilon_0)/m}} \sim AN(0,1)$$



模型检验

- 单总体情况。这种情况下 t 统计量的定义为

$$t = \frac{\bar{X} - u_0}{\sigma / \sqrt{N}}$$

式中 \bar{X} 为样本的均值, u_0 为总体的均值, σ 为总体标准差, N 为样本个数, 由于总体标准差无法得知, 因此一般用样本标准差 S 来估计总体标准差。

- $H_0: \varepsilon \leq \varepsilon_0$

k 个测试错误率, $\hat{\epsilon}_1, \hat{\epsilon}_2, \dots, \hat{\epsilon}_k$,

- t 检验 (交叉验证法)

$$\mu = \frac{1}{k} \sum_{i=1}^k \hat{\epsilon}_i ,$$

$$\sigma^2 = \frac{1}{k-1} \sum_{i=1}^k (\hat{\epsilon}_i - \mu)^2 .$$

$$\tau_t = \frac{\sqrt{k}(\mu - \varepsilon_0)}{\sigma} \text{ 服从自由度为 } k-1 \text{ 的 } t \text{ 分布}$$



- 准确率的置信区间

- X : 模型正确预测的记录数

- p : 模型真正的准确率

- 当 N 充分大时

$$P\left(\left|\frac{acc - p}{\sqrt{p(1-p)/N}}\right| \leq Z_{1-\alpha/2}\right) = 1 - \alpha$$



例 考虑一个模型，它在 100 个检验记录上具有 80% 的准确率。在 95% 的置信水平下，模型的真实准确率的置信区间是什么？

随着记录数 N 的增大所产生的置信区间：

N	20	50	100	500	1000	5000
置信 区间	0.584	0.670	0.711	0.763	0.774	0.789
	-0.919	-0.888	-0.867	-0.833	-0.824	-0.811



- Given two models, say M1 and M2, which is better?
 - M1 is tested on D1 (size=n1), found error rate = e_1
 - M2 is tested on D2 (size=n2), found error rate = e_2
 - Assume D1 and D2 are independent
 - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1^2)$$

$$e_2 \sim N(\mu_2, \sigma_2^2)$$

- Approximate:

$$\hat{\sigma}_i^2 = \frac{e_i(1 - e_i)}{n_i}$$



- To test if performance difference is statistically significant: $d = e1 - e2$
 - $d \sim N(d_t, \sigma^2_t)$ where d_t is the true difference
 - Since D1 and D2 are independent, their variance adds up:

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \approx \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1 - e1)}{n1} + \frac{e2(1 - e2)}{n2}$$

- At $(1-\alpha)$ confidence level, $d_t = d \pm Z_{1-\alpha/2} \hat{\sigma}_t$



- An Illustrative Example

- Given: M1: n1 = 30, e1 = 0.15
M2: n2 = 5000, e2 = 0.25

- $d = |e_2 - e_1| = 0.1$ (2-sided test)

$$\hat{\sigma}_d^2 = \frac{0.15(1 - 0.15)}{30} + \frac{0.25(1 - 0.25)}{5000} = 0.0043$$

- At 95% confidence level, $Z_{1-\alpha/2} = 1.96$

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

- => Interval contains 0 => difference may not be statistically significant



- 交叉验证配对t检验

- 两个学习器A、B

- K折交叉验证法得测试集上的误差为 $\epsilon_1^A, \epsilon_2^A, \dots, \epsilon_k^A$ 和 $\epsilon_1^B, \epsilon_2^B, \dots, \epsilon_k^B$

- 若A、B性能相同，则在相同训练/测试集上的误差应一样，即

$$\Delta_i = \epsilon_i^A - \epsilon_i^B = 0$$

- 计算差值的均值和方差 $\bar{\Delta} = \frac{\sum_{i=1}^K \Delta_i}{K}$ $S^2 = \frac{\sum_{i=1}^K (\Delta_i - \bar{\Delta})^2}{K-1}$

- $H_0: \epsilon^A - \epsilon^B = 0$

$$\frac{\sqrt{K\bar{\Delta}}}{S} \sim t(K-1)$$



■ McNemar 检验

■ 考察二学习器

两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	e_{00}	e_{01}
错误	e_{10}	e_{11}

■ 若二学习器性能相同，则 $e_{01} = e_{10}$

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \sim \chi^2(1)$$

修正系数
(《机器学习》有误)



■ McNemar 检验

		宣讲后		合计
		有必要	无必要	
宣讲前	有必要	28	6	34
	无必要	49	17	66
合计		77	23	100

- 计算统计量；查表找临界值



■ Friedman 检验 和 Nemenyi 检验

- 用于多算法的比较 (H_0 : 所有算法性能相同)
- 基于算法排序
 - 使用交叉验证法得到每个算法在每个数据集上的测试结果
 - 在每个数据集上根据测试性能好坏排序，并赋序值 1, 2,。若算法性能相同，则平分序值
- 计算平均序值

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875



- 若算法性能相同，则平均序值应相同。

- k: 算法个数

- N: 数据集个数

- r_i : 第*i*个算法的平均序数

$$\begin{aligned}\tau_{\chi^2} &= \frac{k-1}{k} \cdot \frac{12N}{k^2-1} \sum_{i=1}^k \left(r_i - \frac{k+1}{2} \right)^2 \\ &= \frac{12N}{k(k+1)} \left(\sum_{i=1}^k r_i^2 - \frac{k(k+1)^2}{4} \right)\end{aligned}$$

在 k 和 N 都较大时，服从自由度为 $k-1$ 的 χ^2 分布。

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(k-1) - \tau_{\chi^2}} \sim F(k-1, (k-1)(N-1))$$



- 若 H_0 被拒绝，则算法性能显著不同。需要进一步区分各算法。

Nemenyi 检验计算出平均序值差别的临界值域

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}},$$

若两个算法的平均序值之差超出了临界值域 CD , 则以相应的置信度拒绝“两个算法性能相同”这一假设.

q_α 的值可以查看下表获得:

α	算法个数 k									
	2	3	4	5	6	7	8	9	10	
0.05	1.960	2.344	2.569	2.728	2.850	2.949	3.031	3.102	3.164	
0.1	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920	



■ 例 (续)

■ $K = 3, N = 4, \alpha = 0.05$

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(k-1) - \tau_{\chi^2}} \sim F(k-1, (k-1)(N-1))$$

 F 检验的常用临界值 $\alpha = 0.05$

数据集 个数 N	算法个数 k	
	2	3
4	10.128	5.143
5	7.709	4.459
8	5.591	3.739
10	5.117	3.555
15	4.600	3.340
20	4.381	3.245

$$\tau_F = 24.429$$

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

Nemenyi 检验中常用的 q_α 值

α	算法个数 k		
	2	3	4
0.05	1.960	2.344	2.569
0.1	1.645	2.052	2.291

拒绝原假设！

$$CD = 1.657$$

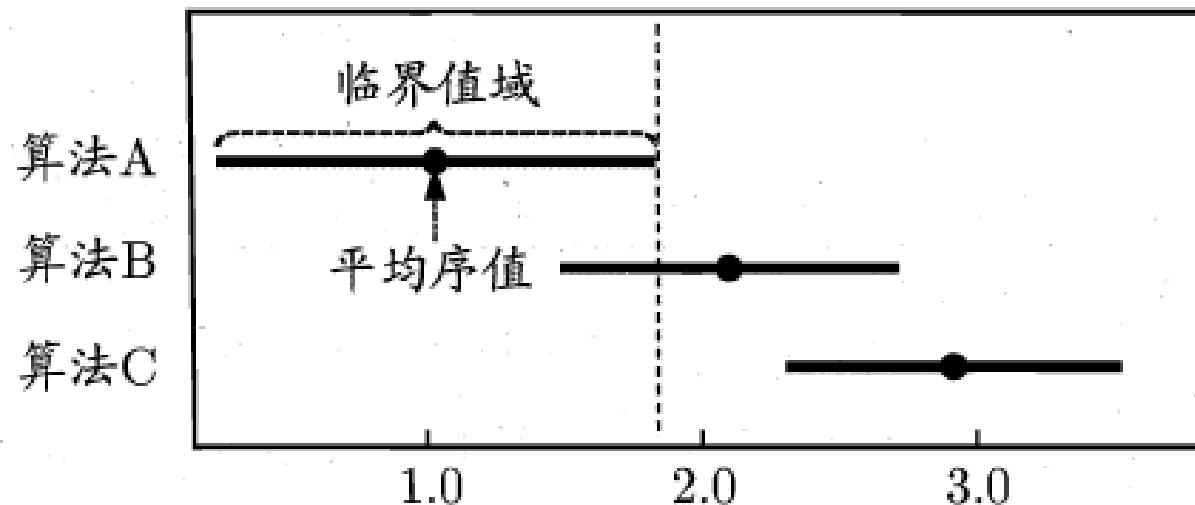
$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

A、C 性能显著不同，AB\BC 否



■ Friedman 检验图

- 横轴：平均序值；纵轴：各算法
- 点：每个算法的平均序值；横线段：临界值域
- 若两个算法的横线段有交叠，则算法无显著差别；否则说明有显著差别。



Friedman 检验图



噪 声 (标注错误) 期 望 为 0

噪 声 与 模 型 无 关

• 偏 差 - 方 差 分 解

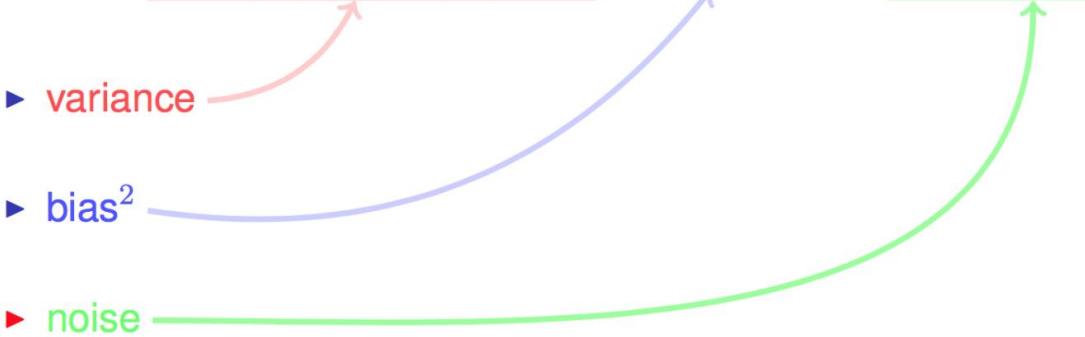
假定 $\mathbb{E}_D[y_D - y] = 0$. $\bar{f}(\mathbf{x}) = \mathbb{E}_D[f(\mathbf{x}; D)]$

$$\begin{aligned}
 E(f; D) &= \mathbb{E}_D [(f(\mathbf{x}; D) - y_D)^2] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}) + \bar{f}(\mathbf{x}) - y_D)^2] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] \\
 &\quad + \mathbb{E}_D [2(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))(\bar{f}(\mathbf{x}) - y_D)] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y + y - y_D)^2] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)^2] + \mathbb{E}_D [(y - y_D)^2] \\
 &\quad + 2\mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)(y - y_D)] \\
 &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)^2] + \mathbb{E}_D [(y - y_D)^2] \\
 &= Variance + Bias + Noise
 \end{aligned}$$

符 号	涵义
\mathbf{x}	测试样本 (总 体)
D	数据集 (多 个)
y_D	\mathbf{x} 在数据集中的标记
y	\mathbf{x} 的真实标记
f	训练集 D 学得的模型
$f(\mathbf{x}; D)$	由训练集 D 学得的模型 f 对 \mathbf{x} 的预测输出
$\bar{f}(\mathbf{x})$	模型 f 对 \mathbf{x} 的 期望 预测 输出

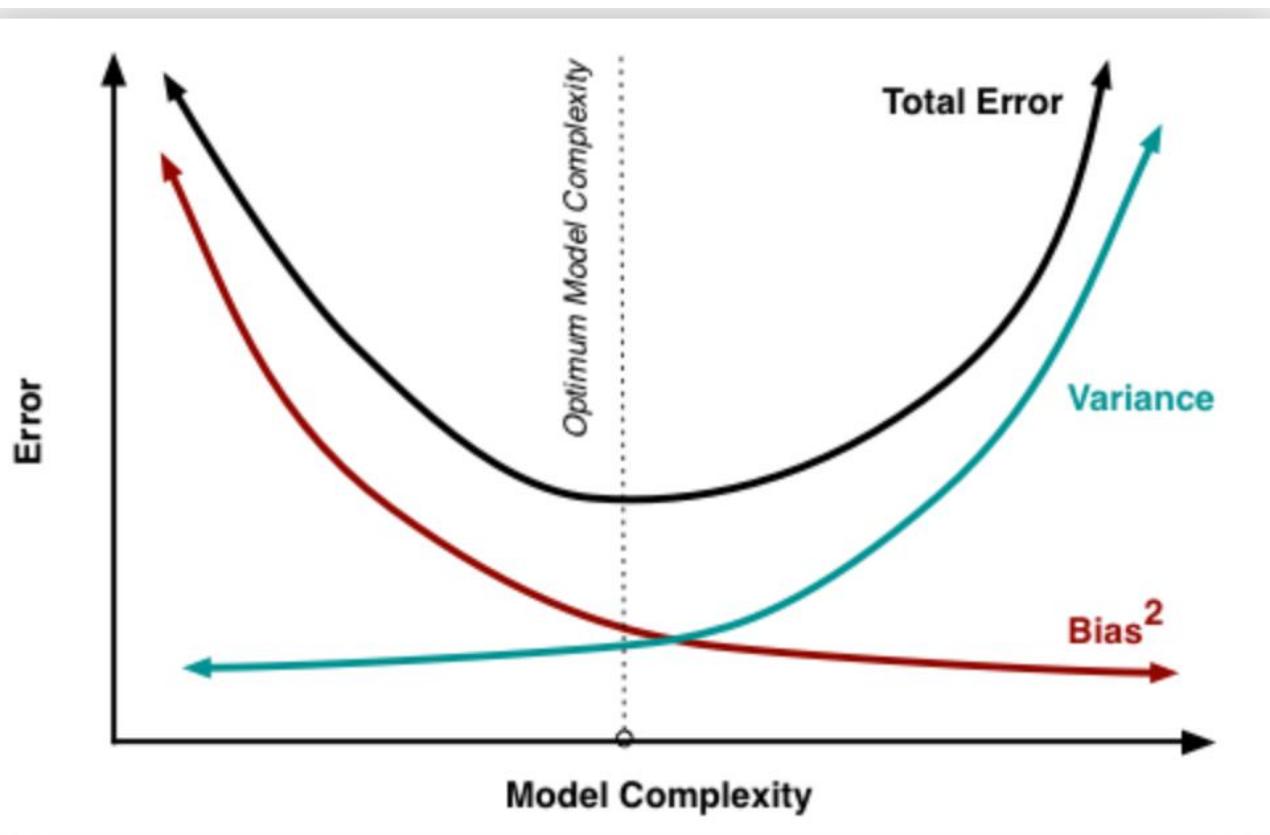


$$Err(\mathbf{x}) = \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + (\bar{f}(\mathbf{x}) - y)^2 + \mathbb{E}_D \left[(y_D - y)^2 \right]$$



■ 欠拟合

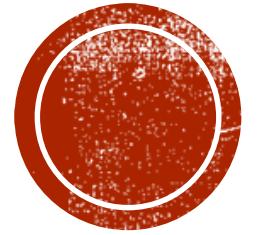
- 拟合能力不足
- 偏差占主导
- 训练数据的扰动不足以使学习器发生变化
- 增加模型参数



■ 过拟合

- 拟合能力过强
- 方差占主导
- 训练数据的轻微扰动会导致模型变化
- 减少参数，正则化
- 集成学习





分 类

Classification

新的问题

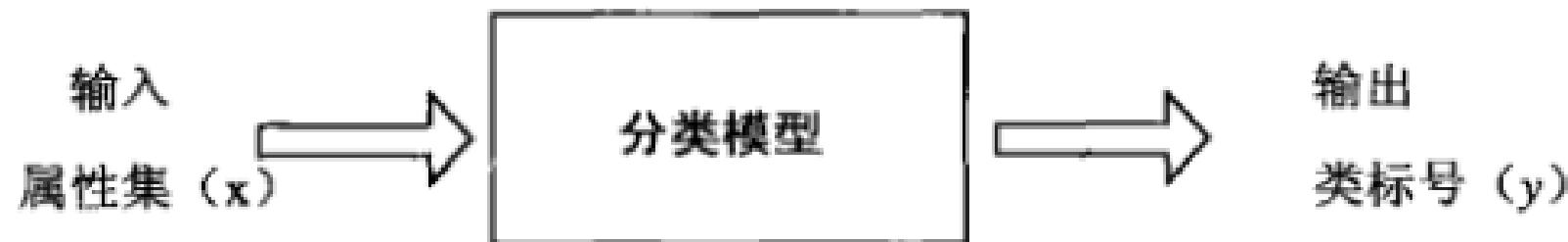
脊椎动物的数据集

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	否	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	半	否	否	否	哺乳类
青蛙	冷血	无	否	否	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	是	是	是	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	否	否	鸟类
猫	恒温	软毛	是	否	否	否	否	哺乳类
豹纹鲨	冷血	鳞片	否	是	否	否	否	鱼类
海龟	冷血	鳞片	否	否	否	否	否	爬行类
企鹅	恒温	羽毛	是	半	否	否	是	鸟类
豪猪	恒温	刚毛	否	半	否	否	否	哺乳类
鳗	冷血	鳞片	是	否	否	否	否	鱼类
蝾螈	冷血	无	否	半	否	否	是	两栖类



分 类

- 分类任务就是通过学习得到一个目标函数 f , 把每个属性集 X 映射到一个预先定义的类标号 y
 - 目标函数也称分类模型



分类器的任务是根据输入属性集 x 确定类标号 y



解决分类问题的一般方法

- 使用一种学习算法 (learning algorithm) 确定分类模型
 - 拟合输入数据中的类标号和属性集之间的联系
 - 正确预测未知样本的类标号
 - 决策树、基于规则的分类法、KNN、神经网络、支持向量机、贝叶斯分类.....
- 提取有效的样本特征



特征的分类：

根据人类视觉系统对场景的感知，特征分为**低层特征**和**高层语义特征**两个方面。

低层特征包括：外观特征、边缘特征、纹理特征、形状特征等。

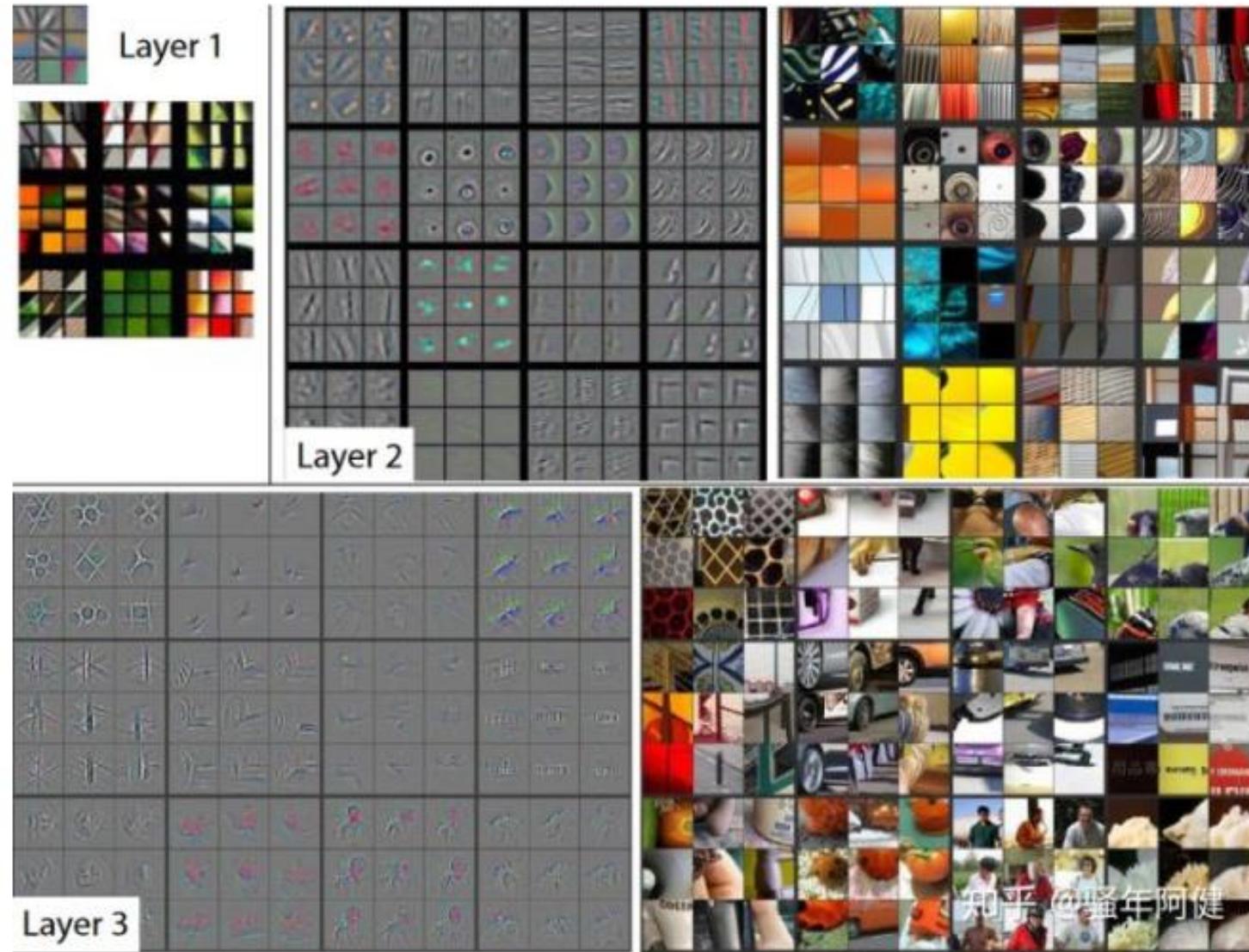
高层语义特征包括：场景语义、行为语义和情感语义等3个层次。

在底层图像特征与高层语义特征之间，往往存在着“语义鸿沟”，视觉物体的**中层语义表示**是连接低层与高层语义的纽带。

低层特征描述的是图像的视觉特征，中层语义是由视觉特征推导得到的中间语义特征，对应于**视觉词袋**和**中间语义（主题语义）**两种表示方法。



低层、中层和高层特征



知乎 @ 骚年阿健



有效的特征与特征描述举例：

- 统计特征（统计直方图、方差、均值、熵）
- 几何特征（包括面积、周长、位置、方向、距离）
- 边界特征（链码、傅立叶描述子）
- 形状描述（区域描述、图像矩）
- 灰度与颜色描述（梯度特征、灰度直方图特征、梯度方向直方图、特征直方图、颜色直方图和颜色直方图距离、颜色不变量特征）
- 角点特征（Harris、Susan、Fast、Brisk）
- 变换特征（傅里叶、小波）
- 纹理特征（灰度共生矩阵、LBP特征，结构相似性特征，Gabor特征）
- 局部不变特征描述子（SIFT、SURF、GLOH、WLD、D-nets） ...



- 1 基本概念
- 2 颜色特征描述
- 3 形状特征描述
- 4 图像的纹理分析技术
- 5 小结



◆ 目的

让计算机具有认识或者识别图像的能力，即图像识别。特征选择是图像识别中的一个关键问题。特征选择和提取的基本任务是如何从众多特征中找出最有效的特征。

◆ 特征形成

根据待识别的图像，通过计算产生一组原始特征，称之为特征形成。



◆ 特征提取

原始特征的数量很大，或者说原始样本处于一个高维空间中，通过映射或变换的方法可以将高维空间中的特征描述用低维空间的特征来描述，这个过程就叫特征提取。

◆ 特征选择

从一组特征中挑选出一些最有效的特征以达到降低特征空间维数的目的，这个过程就叫特征选择。

选取的特征应具有如下特点：

可区别性

可靠性

独立性好

数量少

8.2 颜色特征描述

8.2.1 颜色矩

8.2.2 颜色直方图



8.2.1 颜色矩

- ◆ 颜色矩是以数学方法为基础的，通过计算矩来描述颜色的分布。
- ◆ 颜色矩通常直接在RGB空间计算
- ◆ 颜色分布的前三阶矩表示为：

$$\mu_i = \frac{1}{N} \sum_{j=1}^N p_{i,j}$$

$$\sigma_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^2 \right)^{\frac{1}{2}}$$

$$s_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^3 \right)^{\frac{1}{3}}$$





8.2.2 颜色直方图

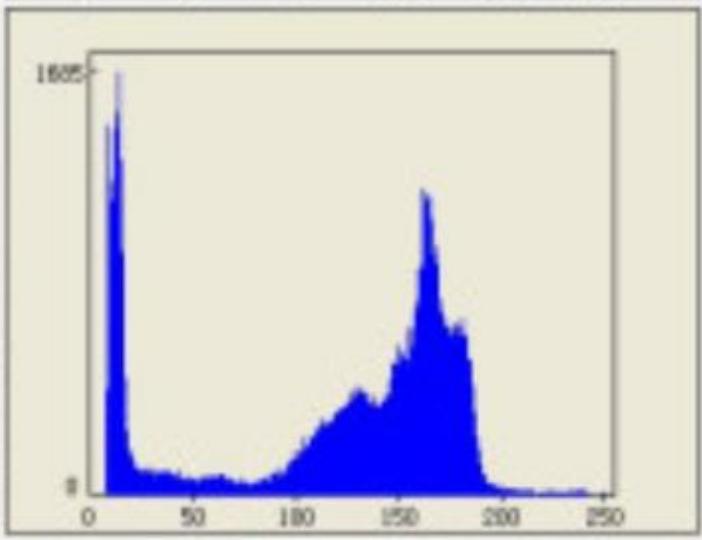
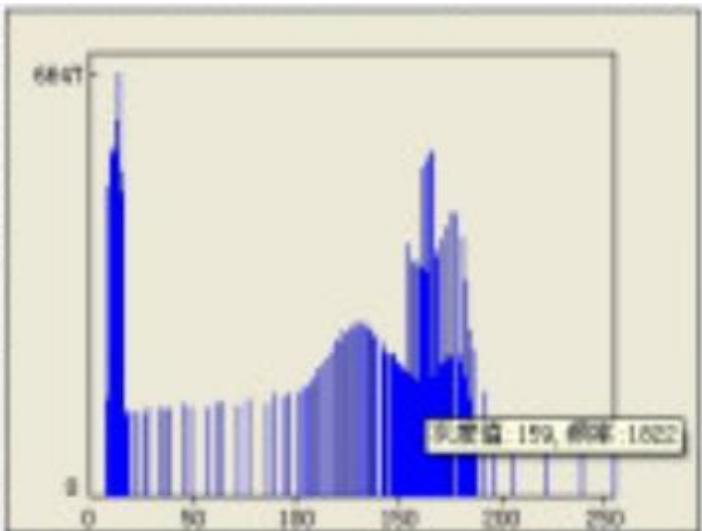
- ◆ 设一幅图像包含M个像素，图像的颜色空间被量化成N个不同颜色。颜色直方图H定义为：

$$p_i = h_i \quad (8-4)$$

h_i 为第i种颜色在整幅图像中具有的像素数。

- ◆ 归一化为：

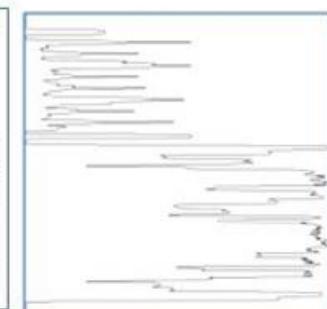
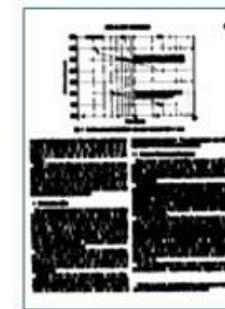
$$p_i = h_i / M \quad (8-5)$$



投影直方图

- 对于一个二值图像 $f(x, y)$, 其垂直和水平投影直方图可以分别记为 $p_v(y)$ 和 $p_h(x)$, 其中

- $p_v(y) = \sum_{x=0}^{w(f)} f(x, y), \quad p_h(x) = \sum_{y=0}^{h(f)} f(x, y),$
- $w(f)$ 和 $h(f)$ 分别为 $f(x, y)$ 的像素宽度和高度。



8.3 形状特征描述



形状特征提取

- 基于图像内物体形状的检索是基于内容检索当中一个**最具挑战性**的问题之一，因为寻找符合人眼感知特性的形状特征不是一件简单的工作。
- 首要的**困难**是要将不同物体从图像中分割出来，这是计算机视觉的困难问题之一。
- 形状的描述也是困难的问题，常用的方法有**傅立叶描述子**、**矩不变量**、各种简单的**形状因子**（如**面积**、**圆度**、**偏心度**、**主轴方向**）等。
- 除了这些全局特征以外，有时也用一些局部特征（如**直线段**、**圆弧**、**角点**、**高曲率点**等），以解决遮挡问题。

福

寿

形状特征提取

Hu不变矩

Zernike不变矩

Hough变换

Radon变换

傅里叶描述子

HU不变矩

- 矩是一种完备的数学表示，其优点是能够直接用于包含感兴趣目标的区域而不需事先把目标分离出来。
- 区域的矩是用所有属于区域内的点计算出来的，因而不太受噪声等的影响。
- 由于低阶矩对噪声和量化误差不敏感，所以矩为描述区域提供了一种虽不完备但却有用的总体表示。
- 矩在图像检索中，特别是在商标图像检索中，得到了广泛的应用。
（背景简单）
- Hu不变矩具有良好的尺度、平移和旋转不变性

HU不变矩

◎ $f(x, y)$ 的 $j + k$ 阶矩定义为：

$$\blacksquare m_{jk} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^j y^k f(x, y) dx dy,$$

◎ $f(x, y)$ 的 $j + k$ 阶中心规格矩定义为：

$$\blacksquare \mu_{jk} = \frac{1}{m_{00}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^j (y - \bar{y})^k f(x, y) dx dy,$$

$$\circ \text{其中 } \bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}, j + k = 2, 3, 4, \dots.$$

◎ 中心规格矩同时具有位置无关性和大小无关性。

◎ 使得二阶中心矩 μ_{11} 为最小的旋转角 θ 可以由如下公式得出

$$\blacksquare \tan 2\theta = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}.$$

◦ 相对于主轴计算的中心规格矩，在物体缩放、平移和旋转时保持不变。

HU不变矩

- $\emptyset_1 = \mu_{20} + \mu_{02}$
- $\emptyset_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$
- $\emptyset_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$
- $\emptyset_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$
- $\emptyset_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$
- $\emptyset_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$
- $\emptyset_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$

HU不变矩



(a)



(b)



(c)



(d)

	\emptyset_1	\emptyset_2	\emptyset_3	\emptyset_4	\emptyset_5	\emptyset_6	\emptyset_7
(a)	0.001286	0.000000	0.000000	0.000000	-0.000000	0.000000	0.000000
(b)	0.411764	0.001129	0.000195	0.004179	0.000004	0.000120	-0.000002
(c)	0.001615	0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000
(d)	0.310856	0.002382	0.005049	0.000931	-0.000001	-0.000009	0.000002

ZERNIKE不变矩

◎ Zernike不变矩是来自数学领域的概念，其定义是一组多项式序列，每个多项式在单位圆上是正交的。

◎ 存在奇数和偶数Zernike多项式，其中偶数多项式可以定义为

- $Z_n^m(\rho, \varphi) = R_n^m(\rho) \cos(m\varphi)$,

- 而奇数多项式则定义为

- $Z_n^{-m}(\rho, \varphi) = R_n^m(\rho) \sin(m\varphi)$,

- 其中， m 和 n 为非负整数，且 $n \geq m$ ， φ 是以弧度为单位的方位角， ρ 为规范化的径向距离，径向多项式 R_n^m 则定义为

- $$\odot R_n^m(\rho) = \sum_{k=0}^{(n-m)/2} \frac{(-1)^k (n-k)!}{k! \left(\frac{n+m}{2} - k\right)! \left(\frac{n-m}{2} - k\right)!} \rho^{n-2k}$$

ZERNIKE不变矩

- ◎ 在Zernike不变矩的基础上，伪Zernike不变矩被提出，并具有更好的鲁棒性和对图像噪声的不敏感性。
 - 寻找目标像素所占据的外接矩形区域；
 - 根据所设定的阶数 n ，进行重复次数 l 的计算，从而进一步得到所需计算的不变矩系数的个数；
 - 针对每一对阶数 n 和重复次数 l ，进行如下四个子步骤的计算：
 - 将步骤1所得到的外接矩形区域归一化为一个单位圆；
 - 将所有图像像素从笛卡尔坐标系转换至以单位圆圆心为原点的极坐标系中；
 - 在极坐标系下计算每一个像素离开原点的径向距离；
 - 统计所有像素的径向距离得到在当前阶数 n 和重复次数 l 下的不变矩系数；
 - 将所有的不变矩系数组合起来得到一个系数序列，便是一幅二值图像所对应的伪Zernike不变矩的计算结果。

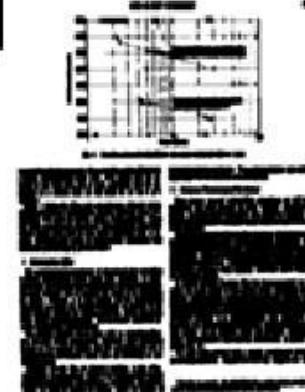
伪ZERNIKE不变矩计算



(a)



(b)



(c)



(d)

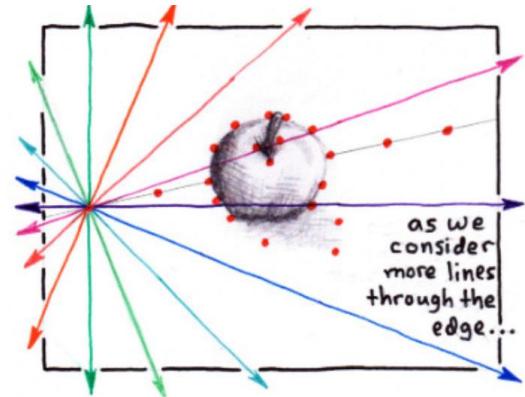
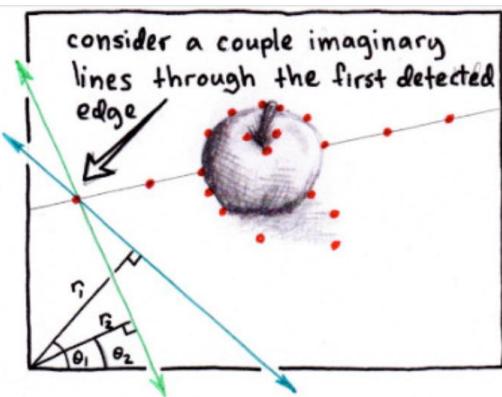
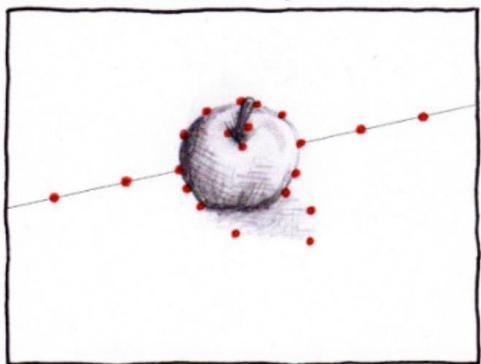
	A_{00}	A_{11}	A_{22}	A_{20}	A_{33}	A_{31}
(a)	0.318224	0.083217	0.289287	0.034964	0.163702	0.004639
(b)	0.318229	0.045778	0.423569	0.037042	0.074096	0.080350
(c)	0.318310	0.142310	0.385979	0.073231	0.185505	0.038667
(d)	0.318289	0.039561	0.186443	0.094189	0.019000	0.054490

HOUGH变换

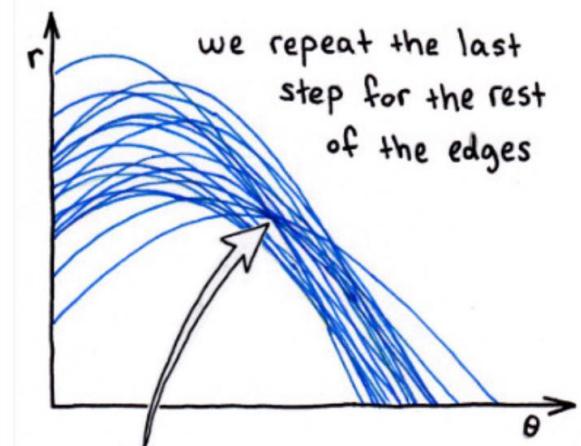
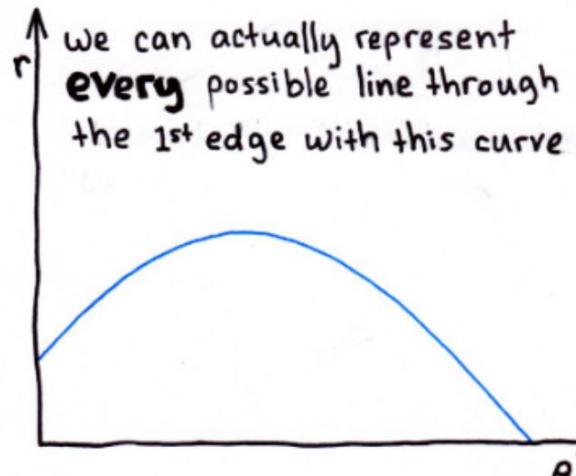
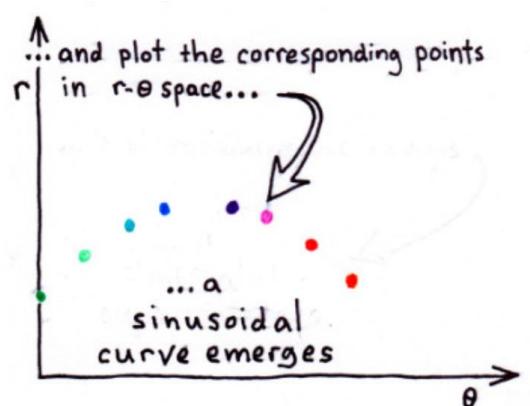
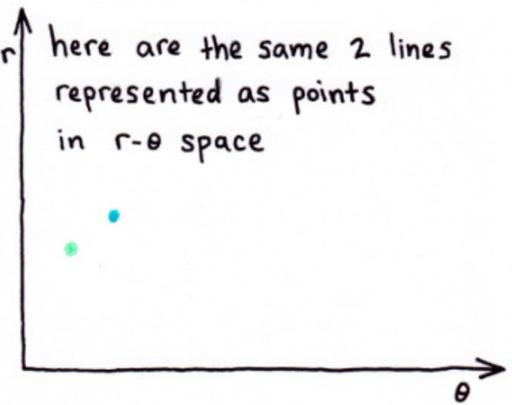
- ◎ 经典的Hough变换主要涉及图像中的直线检测，但是后来Hough变换得到了扩展，被用于任意形状位置的检测，其中最常用的是圆形或者椭圆。
 - Hough变换最简单的实例就是用于直线检测的线性变换。
- ◎ 使用参数空间 (r, θ) ，直线公式可以重写为
 - $y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$,
 - 这也就等价于公式
 - $r = x \cos\theta + y \sin\theta$ 。

直角坐标系中的一条直线对应于极坐标系下的一个点。

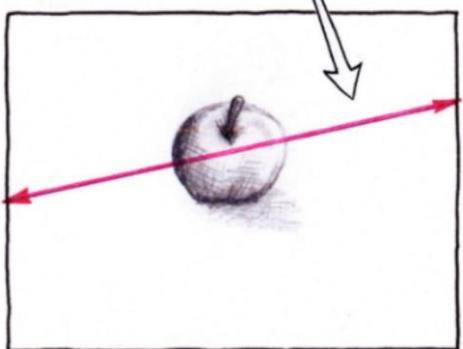
detected edges



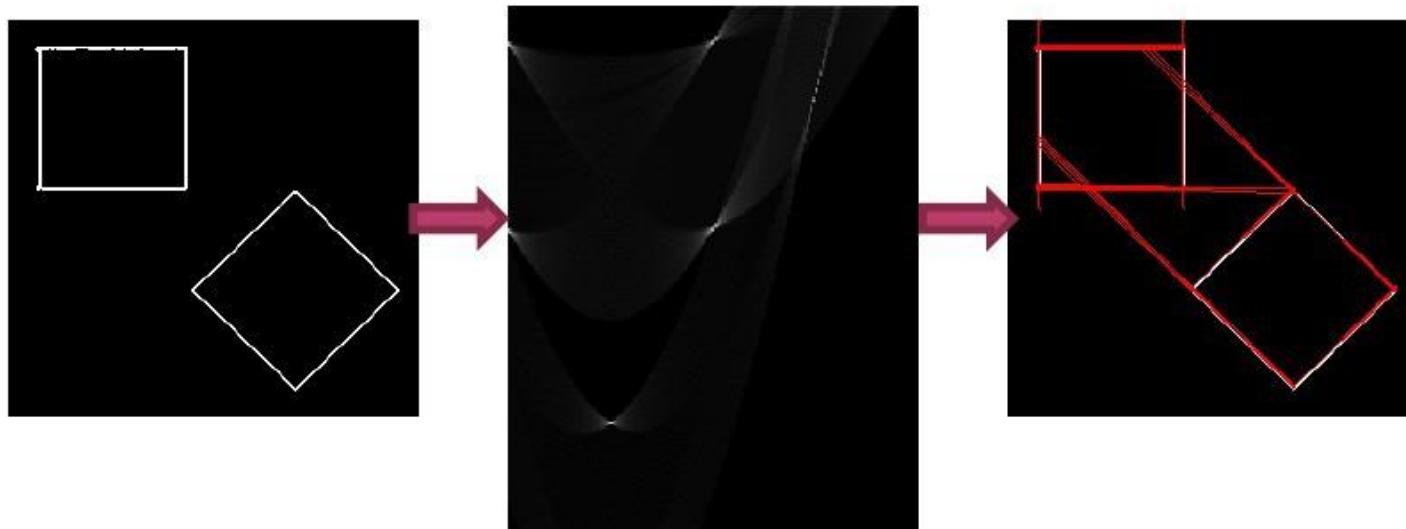
here are the same 2 lines represented as points in $r-\theta$ space



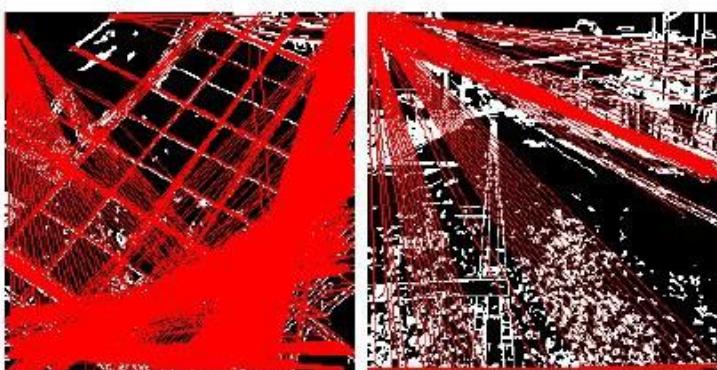
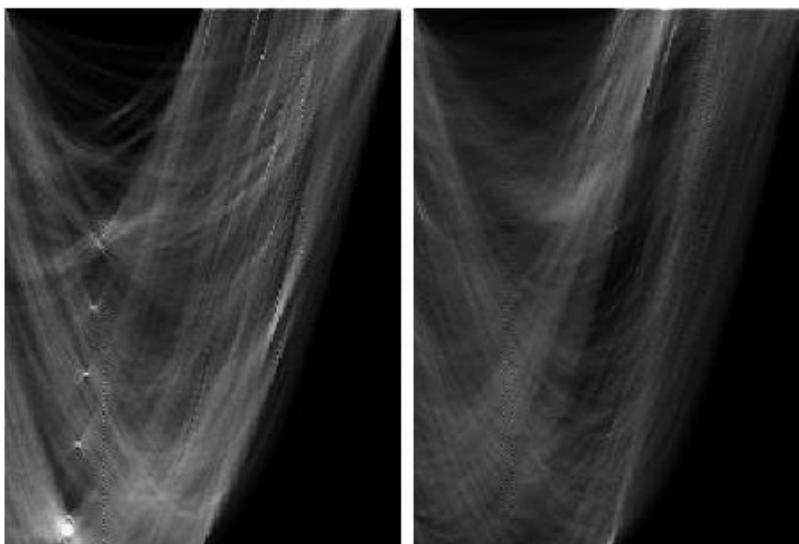
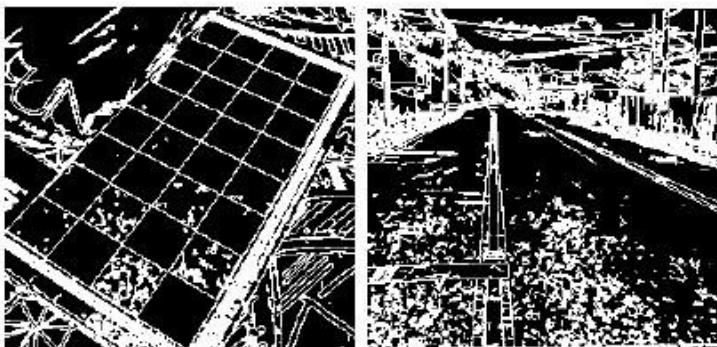
The point with the most curve-crossings identifies an imaginary line in the original image that passes through the most edges



HOUGH变换



HOUGH变换



RADON变换

- ◎ 二维Radon变换属于一种积分变换，是由众多直线函数上的积分组成。
- ◎ Radon变换被广泛的应用于断层摄影术。
 - 如果 f 表示一个未知的密度函数，那么Radon变换表示一种发散数据，可用于断层摄影术的输出。
 - 因此，Radon逆变换可以用于从发散数据中重构原始密度函数，从而形成了断层摄影术重构技术的数学基础。
- ◎ Radon变换数据通常被称为正弦图，因为Dirac三角函数的Radon变换结果看上去是呈一个正弦波的分布。
 - 许多小物体的Radon变换在显示上都会呈现出模糊正弦波的样子，而这些正弦波具有不同的振幅和相位。

RADON变换

◎ Radon变换的数学公式可以描述为

- $R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(\rho - x\cos\theta - y\sin\theta) dx dy,$
- 其中 $\rho = x\cos\theta + y\sin\theta$ 表示一条直线， $\delta(\cdot)$ 为 Dirac 三角函数。

◎ Dirac 三角函数大致可以认为是一种在原点无穷大而其余地方为零的函数，即

- $\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases},$

- 并且满足如下条件

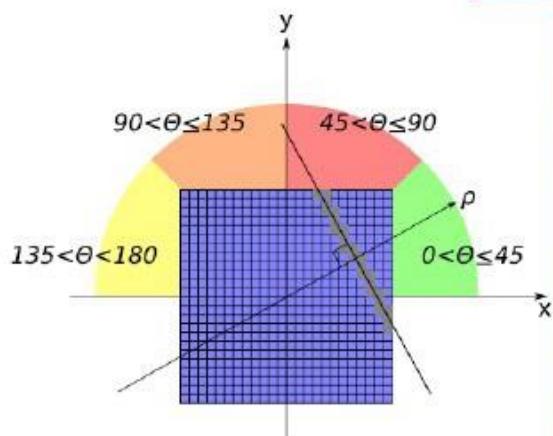
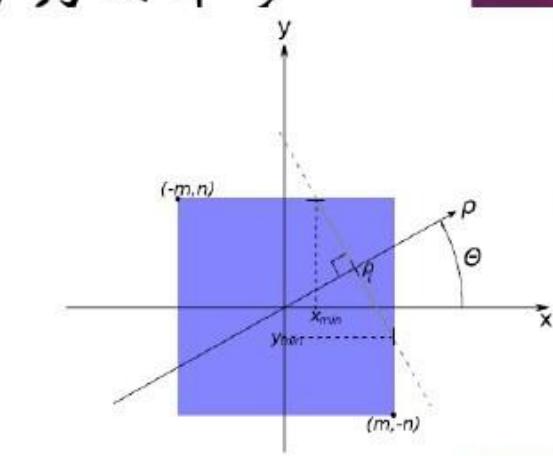
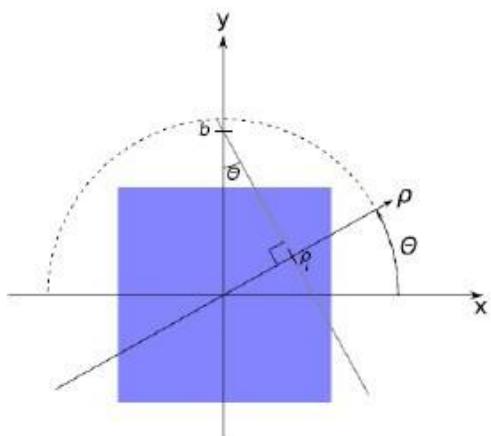
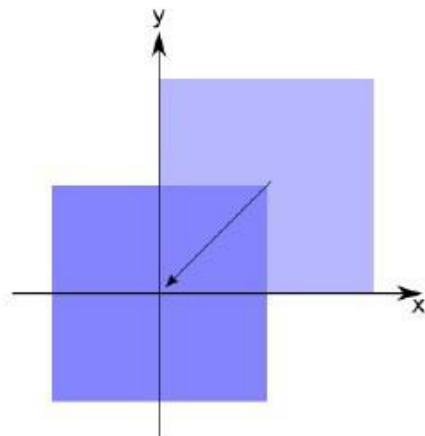
- $\int_{-\infty}^{\infty} \delta(x) dx = 1.$

◎ Radon 变换实际上是从笛卡尔直角坐标 (x, y) 到极坐标 (ρ, θ) 的映射

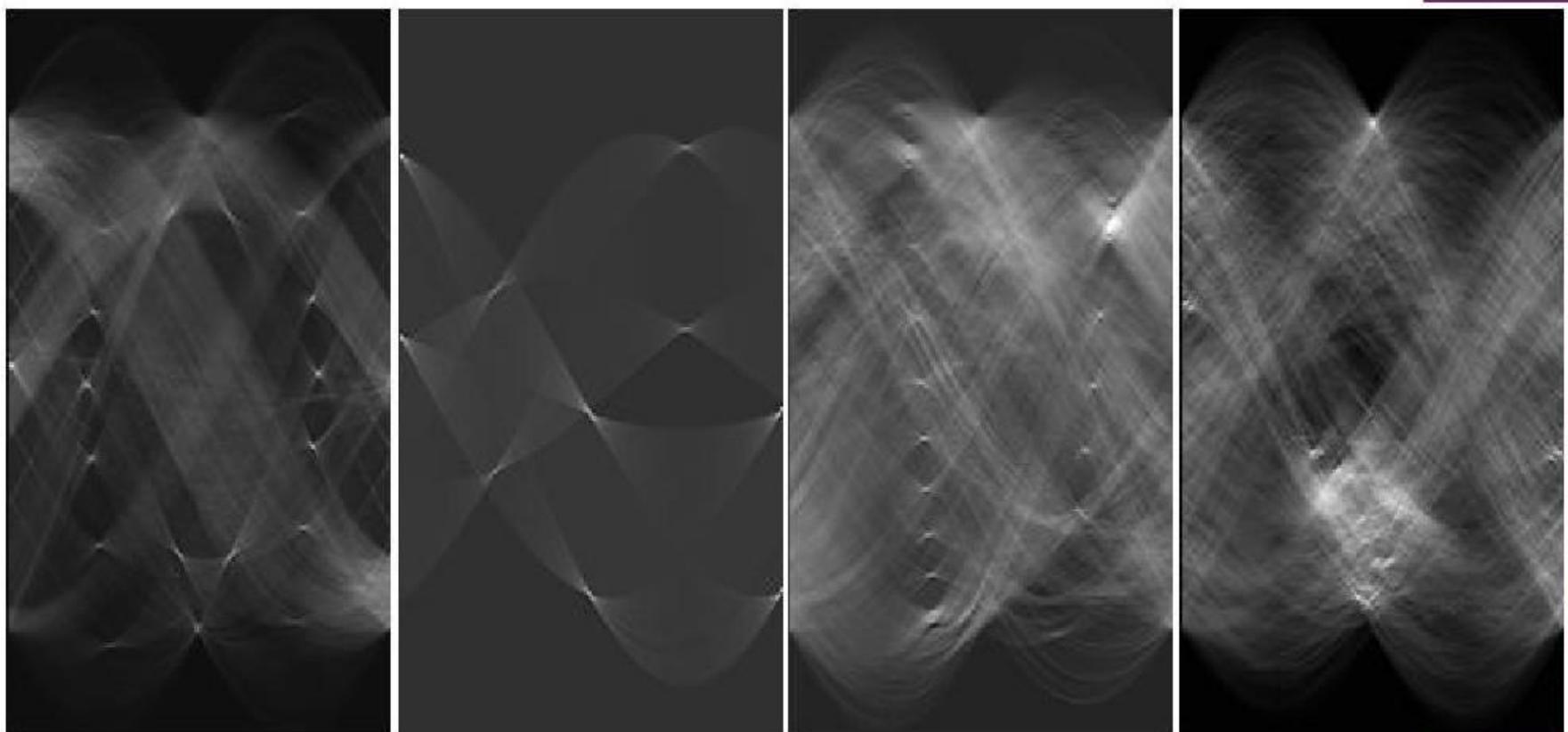
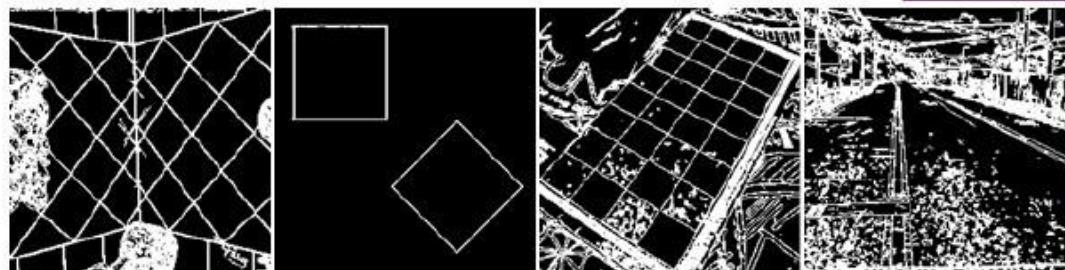
RADON变换

◎ Radon变换的具体实现，大致可以分为四个步骤

- 将坐标原点移至图像中心
- 直线参数计算
- 直线坐标的最大最小值计算
- 根据角度进行分区讨论



RADON变换



傅里叶描述子

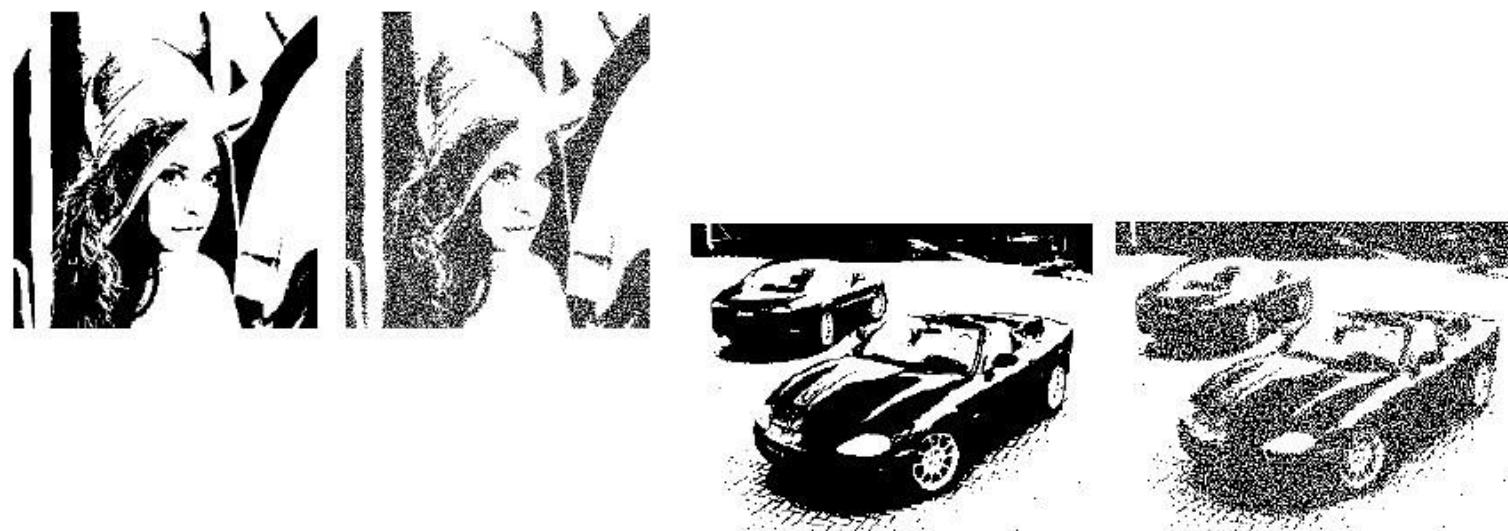
- ① 傅里叶描述子，就是对封闭曲线进行傅里叶变换，将变换后得到的傅里叶系数作为表示图像形状的特征。
- ② 可以将形状表示为具有 N 个顶点的集合 $\{z(i): i, \dots, N\}$ ，那么傅里叶描述子 $\{c(k): k = -\frac{N}{2} + 1, \dots, \frac{N}{2}\}$ 就是集合 Z 的傅里叶变换系数集合，即

- $z_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} c_k e^{2\pi j \frac{ki}{N}}$ ，
- 而 $c(k)$ 与 $z(i)$ 之间的逆变换为
 - $c_k = \frac{1}{N} \sum_{i=1}^N z_i e^{-2\pi j \frac{ik}{N}}$ 。

- ③ 为了进行傅里叶描述子的计算，就需要两步操作：
 - 一是从二值图像中得到一维复数数组，用以作为一维傅里叶变换的输入
 - 二是进行一维傅里叶正变换。

傅里叶描述子

- 为了验证所得到的傅里叶描述子是否正确，可以对傅里叶描述子进行傅里叶逆变换，然后从所得到的一维复数数组中恢复二值图像，通过比较新旧图像就可以看出所得到的傅里叶描述子是否正确。



■ 区域形状的描述

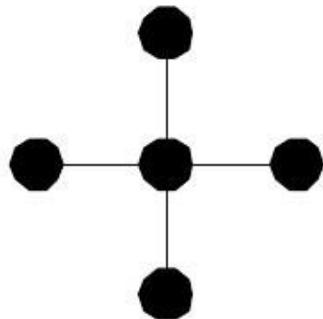
- 常针对二值图
- 属于手工特征



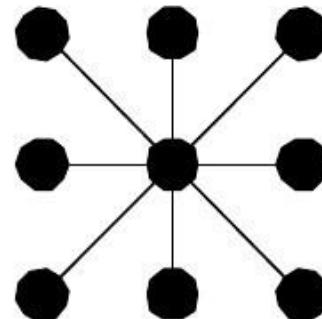
8.3.1几个基本概念

◆ 邻域与邻接

- ▶ 对于任意像素 $(i, j), (s, t)$ 是一对适当的整数，则把像素的集合 $\{(i+s, j+t)\}$ 叫做像素 (i, j) 的邻域.
- ▶ 直观上看，这是像素 (i, j) 附近的像素形成的区域.
- ▶ 最经常采用的是4-邻域和8-邻域



(a)



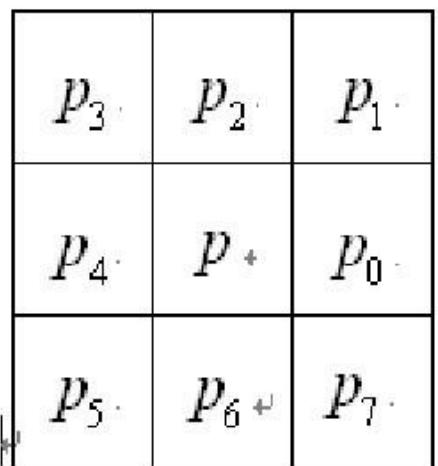
(b)

4-邻域和8-邻域



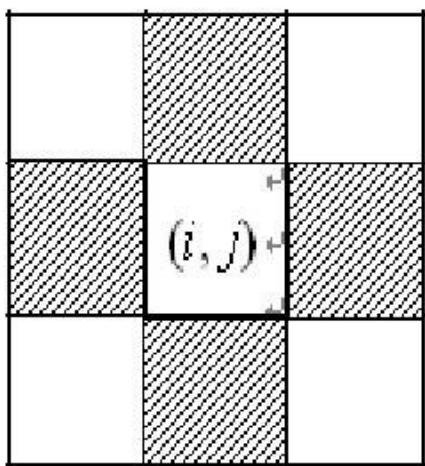
◆ 邻域与邻接

- ▶ 互为4-邻域的两像素叫4-邻接。
- ▶ 互为8-邻域的两像素叫8-邻接。



(a)

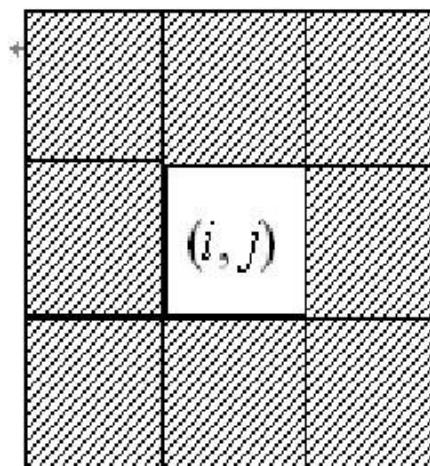
(a) 像素的编号



(b)

图 8-1 邻接像素的种类

(b) 4-邻接

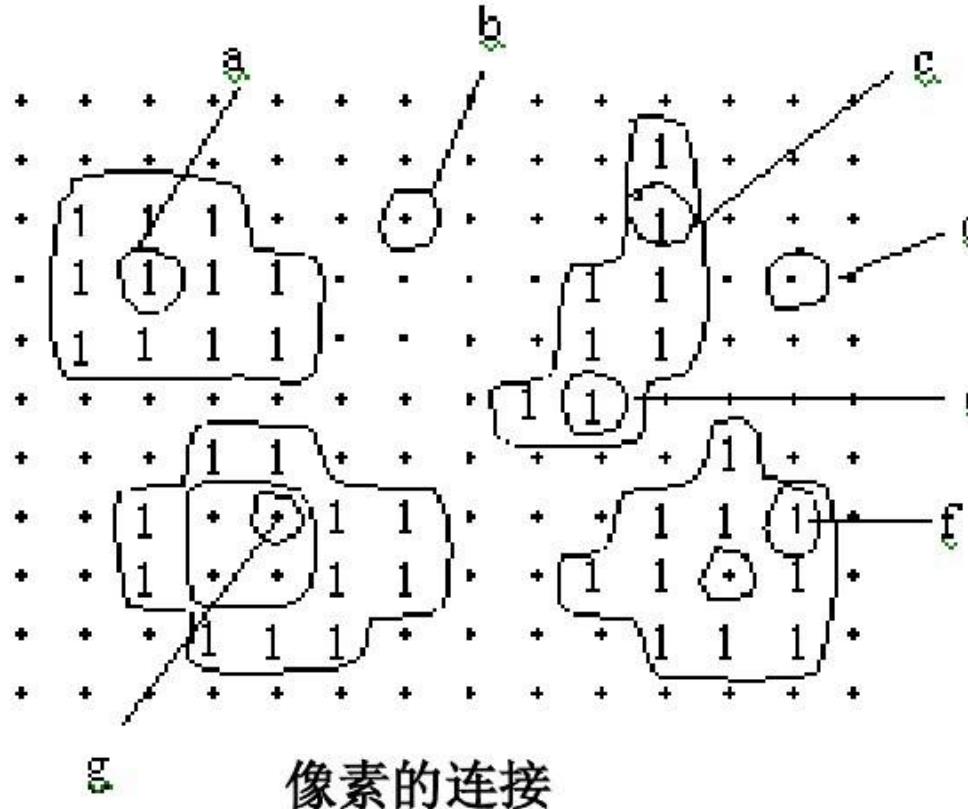


(c)

(c) 8-邻接

◆ 像素的连接

对于图像中具有相同值的两个像素A和B，如果所有和A、B具有相同值的像素序列 $L_0 (= A), L_1, L_2, \dots, L_{n-1}, L_n (= B)$ 存在，并且 L_{i-1} 和 L_i 互为4-邻接或8-邻接，那么像素和叫做4-连接或8-连接，以上的像素序列叫4-路径或8-路径。



像素的连接

◆ 连接成分

- ▶ 在图像中，把互相连接的像素的集合汇集为一组，于是具有若干个0值的像素和具有若干个1值的像素的组就产生了。把这些组叫做连接成分，也称作连通成分。
- ▶ 在研究一个图像连接成分的场合，若1像素的连接成分用4-连接或8-连接，而0像素连接成分不用相反的8-连接或4-连接就会产生矛盾。
- ▶ 假设各个1像素用8-连接，则其中的0像素就被包围起来。如果对0像素也用8-连接，这就会与左下的0像素连接起来，从而产生矛盾。因此0像素和1像素应采用互反的连接形式，即如果1像素采用8-连接，则0像素必须采用4-连接。

0	0	0	0	0
0	1	1	1	0
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

连接性矛盾示意图



► 在0-像素的连接成分中，如果存在和图像外围的1行或1列的0-像素不相连接的成分，则称之为孔。不包含有孔的1像素连接成分叫做单连接成分。含有孔的1像素连接成分叫做多重连接成分。

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

(a)孤立点

0	0	0	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(b)

⊕

孔

1	1	1	1	1
1	0	1	1	1
1	1	1	0	1
1	0	0	0	1
1	1	1	1	1

(c)

图 8-4 连接成分

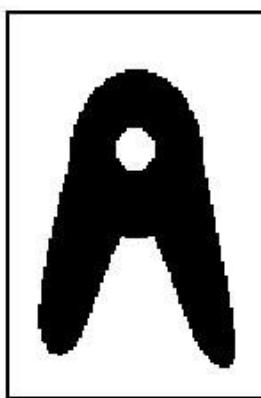
(a)孤立点 (b)单连接成分(c)多重连接成分

8.3.2 区域内部空间域分析

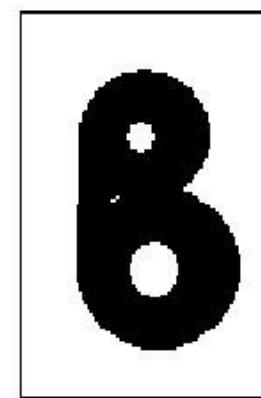
- ◆ 区域内部空间域分析是不经过变换而直接在图像的空间域，对区域内提取形状特征。

1. 欧拉数 欧拉数=连接成分数-孔数

- ✓ 图像的欧拉数是图像的拓扑特性之一，它表明了图像的连通性。下图 (a) 的图形有一个连接成分和一个孔，所以它的欧拉数为0，而下图 (b) 有一个连接成分和两个孔，所以它的欧拉数为-1。
- ✓ 可见通过欧拉数可用于目标识别。



(a)



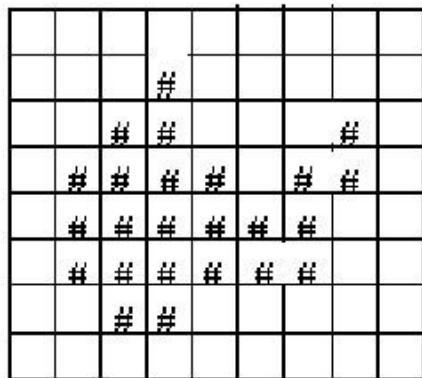
(b)

具有欧拉数为0和-1的图形

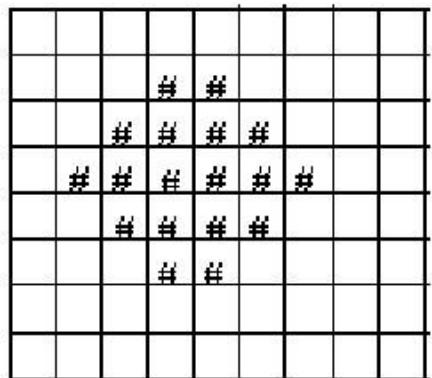


2.凹凸性

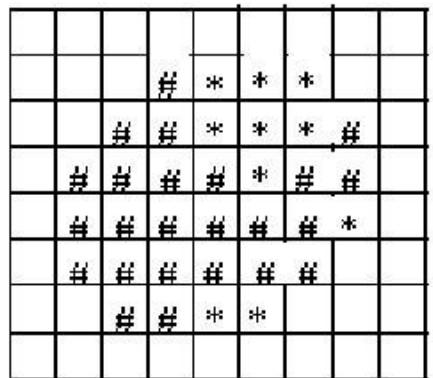
- ✓ 凹凸性是区域的基本特征之一，区域凹凸性可通过以下方法进行判别：区域内任意两像素间的连线穿过区域外的像素，则此区域为凹形。相反，连接图形内任意两个像素的线段，如果不通过这个图形以外的像素，则这个图形称为是凸的。任何一个图形，把包含它的最小的凸图形叫这个图形的凸闭包。
- ✓ 凸图形的凸闭包就是它本身。从凸闭包除去原始图形的部分后，所产生的图形的位置和形状将成为形状特征分析的重要线索。凹形面积可将凸封闭包减去凹形得到。



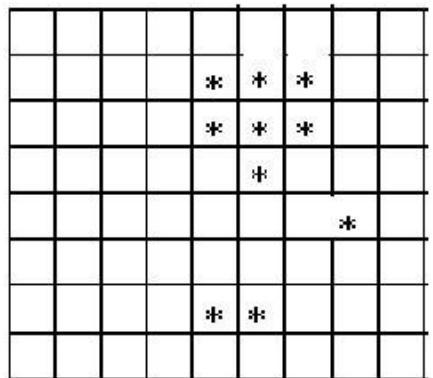
(a) 凹形



(b) 凸形



(c)(a) 中凹形的凸封闭包



(d) 凹形面积

区域的凹凸性



3. 距离

- ✓ 距离在实际图像处理过程中往往作为一个特征量出现，因此对其精度的要求并不是很高。所以对于给定图像中三点A, B, C，当函数 $D(A, B)$ 满足下式的条件时，把 $D(A, B)$ 叫做A和B的距离，也称为距离函数。

$$\begin{cases} D(A, B) \geq 0 \\ D(A, B) = D(B, A) \\ D(A, C) \leq D(A, B) + D(B, C) \end{cases}$$

- ✓ 第一个式子表示距离具有非负性，并且当A和B重合时，等号成立；
- ✓ 第二个式子表示距离具有对称性
- ✓ 第三个式子表示距离的三角不等式。



✓ 计算点(i, j)和(h, k)间距离常采用的几种方法：

(1) 欧氏距离，用 d_e 来表示。

$$d_e[(i, j), (h, k)] = [(i - h)^2 + (j - k)^2]^{\frac{1}{2}}$$

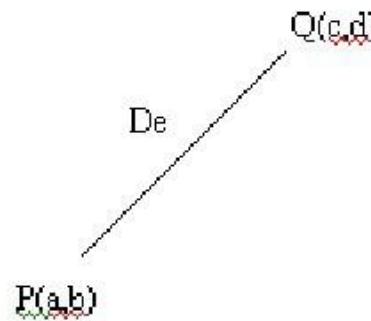
(2) 4-邻域距离，也称为街区距离。

$$d_s[(i, j), (h, k)] = [(i - h)^2 + (j - k)^2]^{\frac{1}{2}}$$

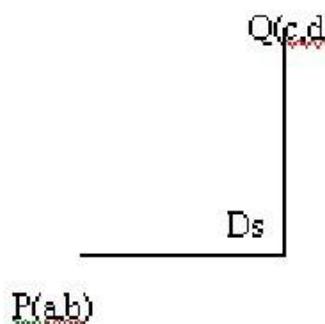
(3) 8-邻域距离，也称为棋盘距离。

$$d_g[(i, j), (h, k)] = \max(|i - h|, |j - k|)$$

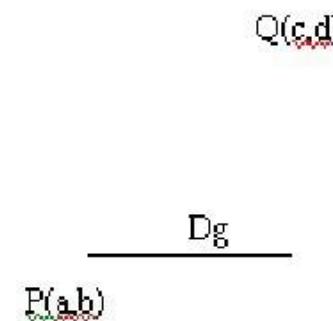
这三种距离之间的关系： $d_g \leq d_s \leq d_e$ ，如图所示。街区距离和棋盘距离都是欧式距离的一种近似。



(a) 欧式距离



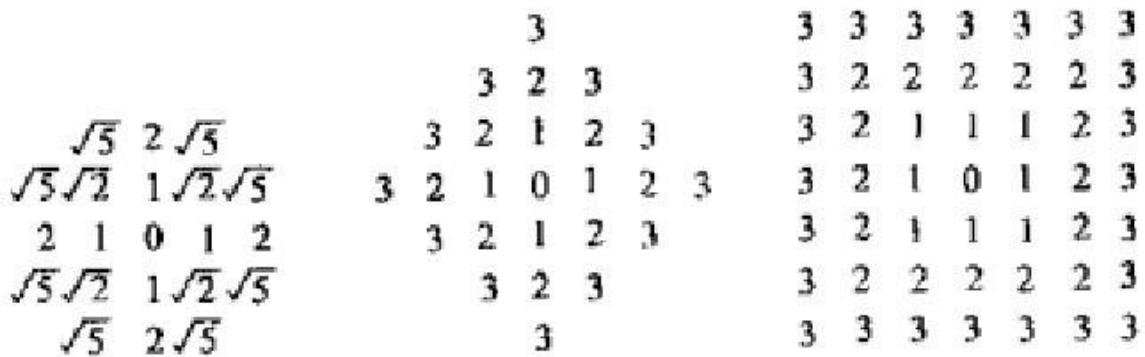
(b) 街区距离



(c) 棋盘距离

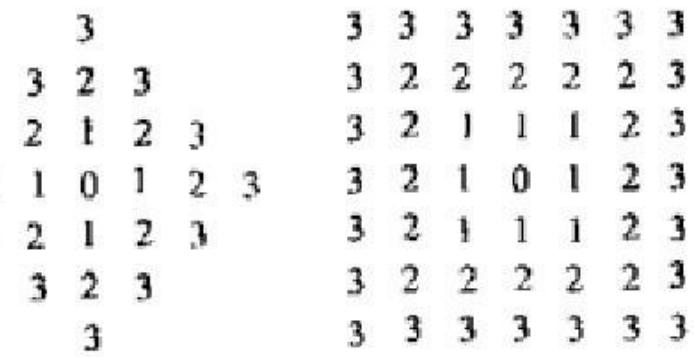


- ✓ 下图中表示了以中心像素为原点的各像素的距离。从离开一个像素的等距离线可以看出，在欧氏距离中大致呈圆形，在棋盘距离中呈方形，在街区距离中呈倾斜45度的正方形。街区距离是图像中两点间最短的4-连通的长度，而棋盘距离则是两点间最短的8-连通的长度。
- ✓ 此外，把4-邻域距离和8-邻域距离组合起来而得到的八角形距离有时也被采用，它的等距线呈八角形。



(a)

(a) 欧几里德距离



(b)

(b) 4-邻域距离

(c)

离开单个像素的距离

(c) 8-邻域距离

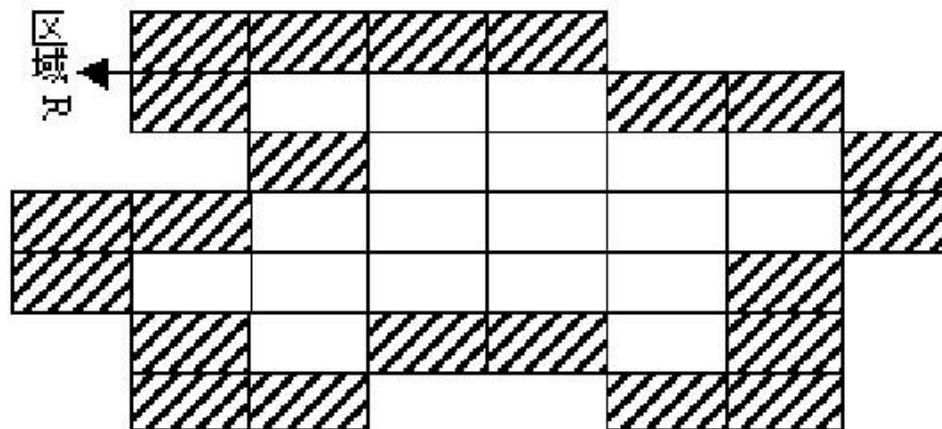


4.区域的测量

- ✓ 区域的大小及形状表示方法主要包括以下几种：

(1) 面积S：图像中的区域面积S可以用同一标记的区域内像素的个数总和来表示。

福



王

区域的面积和周长

按上述表示法区域R的面积 $S=41$ 。区域面积可以通过扫描图像，累加同一标记像素得到，或者是直接在加标记处理时计数得到。



(2) 周长L: 区域周长L是用区域中相邻边缘点间距离之和来表示。采用不同的距离公式, 关于周长L的计算有很多方法。常用的有两种:

一种计算方法是采用欧式距离, 在区域的边界像素中, 设某像素与其水平或垂直方向上相邻边缘像素间的距离为1, 与倾斜方向上相邻边缘像素间的距离为 $\sqrt{2}$ 。周长就是这些像素间距离的总和。这种方法计算的周长与实际周长相符, 因而计算精度比较高。

另一种计算方法是采用8邻域距离, 将边界的像素个数总和作为周长。也就是说, 只要累加边缘点数即可得到周长, 比较方便, 但是, 它与实际周长间有差异。根据这两种计算周长的方式, 以区域的面积和周长图为例, 区域的周长分别是 $14 + 8\sqrt{2}$ 。

(3) 圆形度 R_0 : 圆形度 R_0 用来描述景物形状接近圆形的程度，它是测量区域形状常用的量。其计算公式为：

$$R_0 = 4\pi S / L^2$$

式中为S区域面积；L为区域周长 R_0 值的范围为 $0 < R_0 \leq 1$ ， R_0 值的大小反映了被测量边界的复杂程度，越复杂的形状取值越小。 R_0 值越大，则区域越接近圆形。



(4) 形状复杂性e: 形状复杂性常用离散指数表示，其计算公式为：

$$e = L^2 / S$$

该式描述了区域单位面积的周长大小，e值越大，表明单位面积的周长大，即区域离散，则为复杂形状；反之，则为简单形状。e值最小的区域为圆形。

典型连续区域的计算结果为：圆形e=12.6；正方形e=16.0；正三角形e=20.8。

此外，常用的特征量还有区域的幅宽、占有率和直径等。

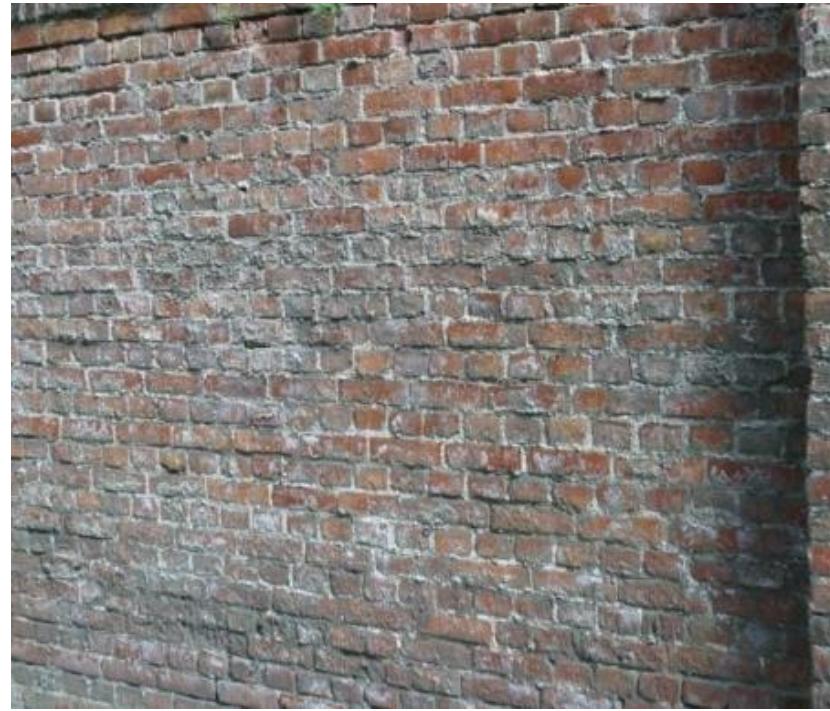
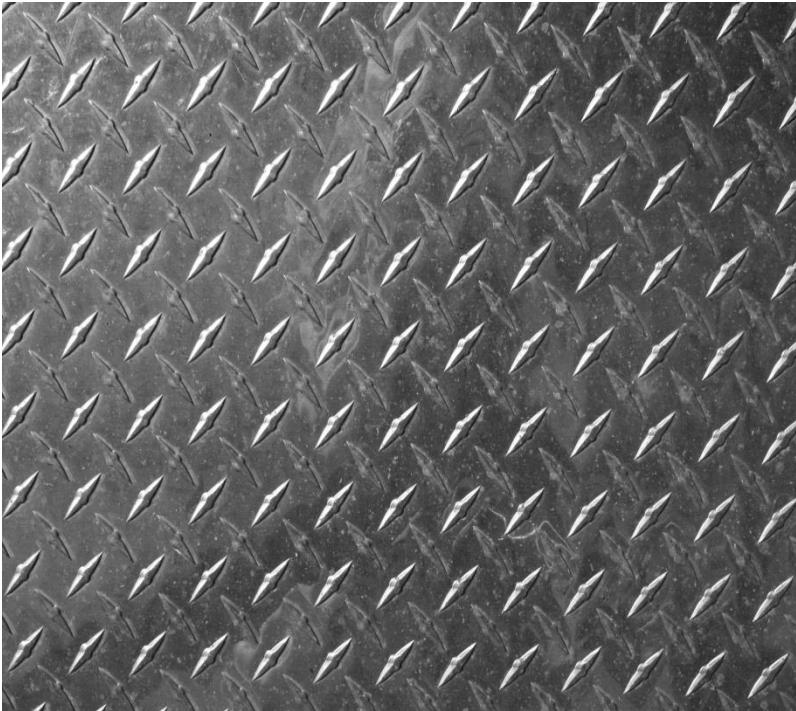


8.4 图像的纹理分析技术

8.4.1 纹理分析概念

- ◆ 指的是图像像素灰度级或颜色的某种变化，主要研究如何获得图像纹理特征和结构的定量描述和解释，以便于图像分析、分割和理解。
- ◆ 一般来说，可以认为纹理由许多相互接近、相互编织的元素构成，并常富有周期性。
- ◆ 纹理的定义大体可以从三个方面来描述：
 - ▶ 具有某种局部的序列性，并在该序列更大的区域内不断重复；
 - ▶ 序列由基本部分非随机排列组成；
 - ▶ 各个部分大致都是均匀的统一体。





图像的纹理特征



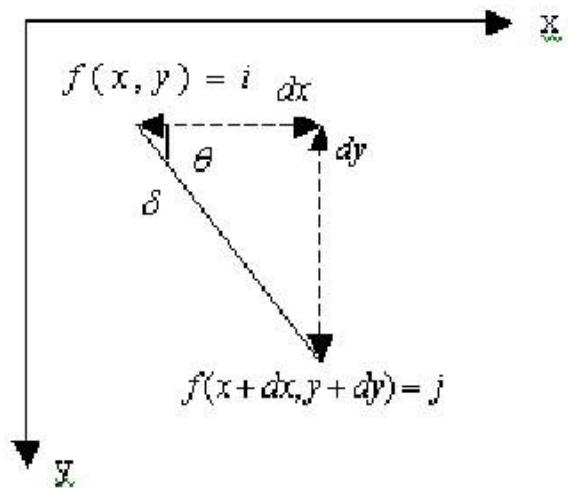
- ◆ 纹理分析是指通过一定的图像处理技术抽取出纹理特征，从而获得纹理的定量或定性描述的处理过程。
- ◆ 纹理特征是从图像中计算出来的一个值，它对区域内部灰度级变化的特征进行量化。
- ◆ 纹理分析基本过程是从像素出发，在纹理图像中提取出一些辨识力比较强的特征，作为检测出的纹理基元，并找出纹理基元排列的信息，建立纹理基元模型，然后再利用此纹理基元模型对纹理图像进一步分割、分类或是辨识等处理。



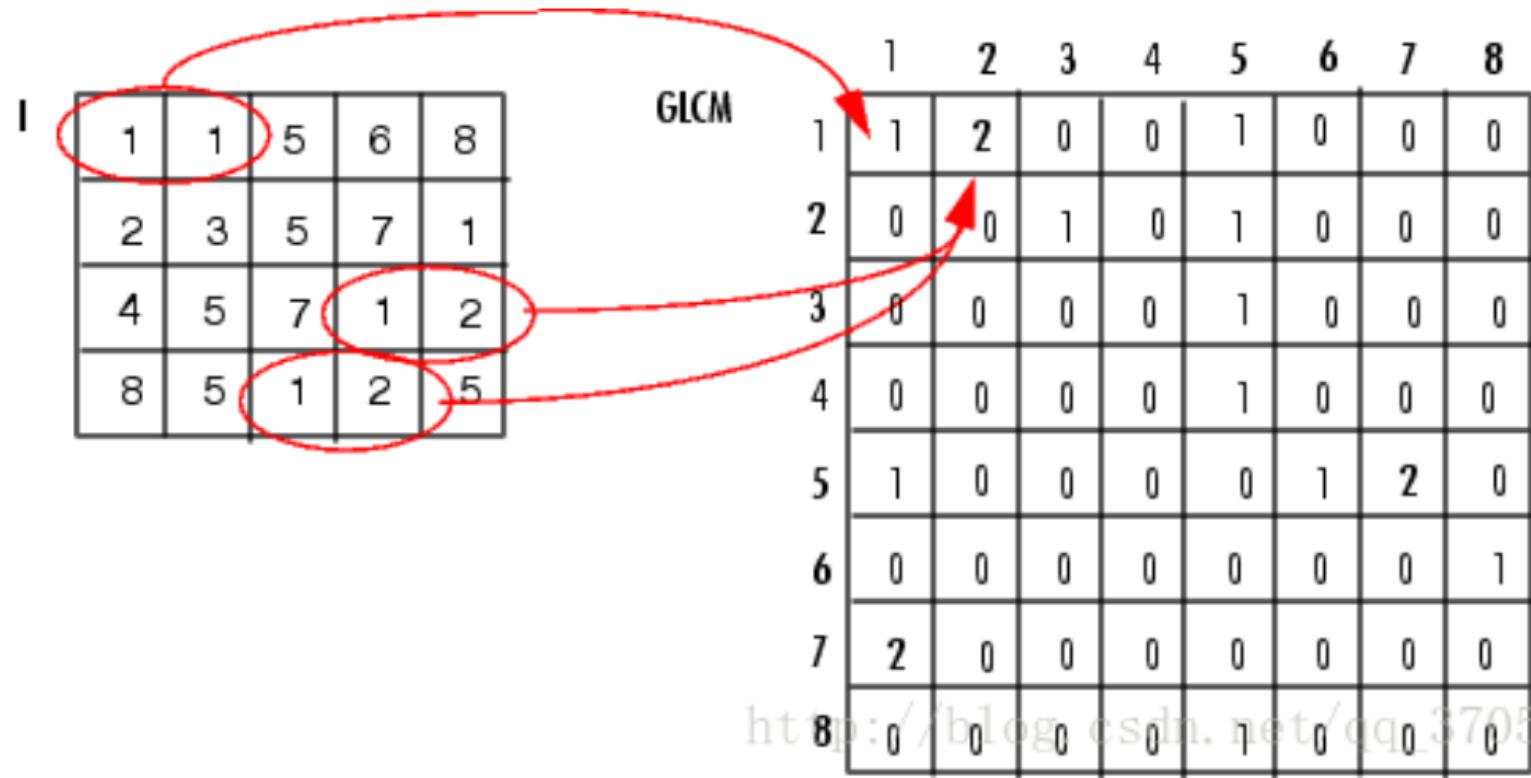
8.4.2 空间灰度共生矩阵

- ✓ 灰度共生矩阵就是从 $N \times N$ 的图像 $f(x, y)$ 的灰度为 i 的像素出发，统计与距离为 $\delta = (\alpha x^2 + \alpha y^2)^{1/2}$ ，灰度为 j 的像素同时出现的概率 $P(i, j, \delta, \theta)$ 。用数学表达式则为：

$$P(i, j, \delta, \theta) = \{(x, y), (x + \alpha x, y + \alpha y) \mid f(x, y) = i, f(x + \alpha x, y + \alpha y) = j\}$$



灰度共生矩阵的像素对



- 分方向
- 构造矩阵
- 统计个数
- 形成特征向量

http://blog.csdn.net/cqq_37059



1. 0° 方向灰度共生矩阵

当 $\theta = 0^\circ$, $\Delta x = 1, \Delta y = 0$ 由于所给图像中只有4个灰度级, 因此所求得的灰度共生矩阵的大小为 4×4 。

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3

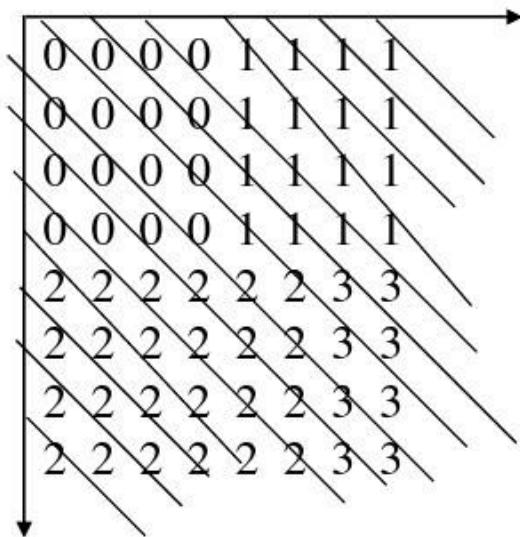
一幅数字灰度图像

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3

0° 方向灰度共生矩阵计算示意图

2. 45° 方向灰度共生矩阵

当 $\theta = 45^\circ$ 时, $\alpha_x = 1, \alpha_y = -1$ 。



45° 方向灰度共生矩阵计算示意图



3. 90° 方向灰度共生矩阵

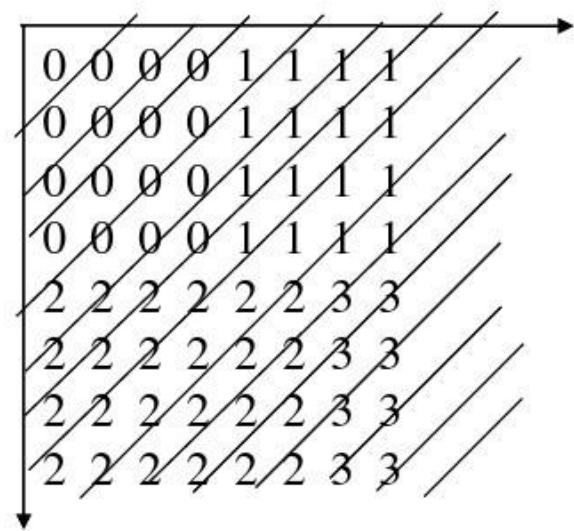
当 $\theta = 90^\circ$ 时, $d_x = 0, d_y = -1$ 。

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3
2	2	2	2	2	2	3	3

90° 方向灰度共生矩阵计算示意图

4. 135° 方向灰度共生矩阵

当 $\theta = 135^\circ$ 时, $\alpha_x = -1, \alpha_y = -1$ 。



135° 方向灰度共生矩阵计算示意图

灰度共生矩阵计算结果

$$P(0^\circ) = \begin{bmatrix} 12 & 4 & 0 & 0 \\ 4 & 12 & 0 & 0 \\ 0 & 0 & 20 & 4 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

(a)

$$P(45^\circ) = \begin{bmatrix} 9 & 3 & 4 & 0 \\ 3 & 9 & 1 & 2 \\ 4 & 1 & 15 & 3 \\ 0 & 2 & 3 & 3 \end{bmatrix}$$

(b)

$$P(90^\circ) = \begin{bmatrix} 12 & 0 & 4 & 0 \\ 0 & 12 & 2 & 2 \\ 4 & 2 & 18 & 0 \\ 0 & 2 & 0 & 6 \end{bmatrix}$$

(c)

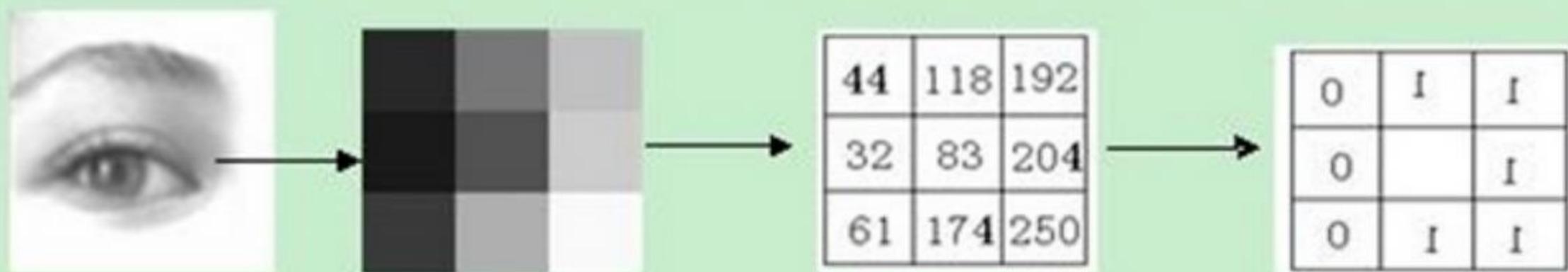
$$P(135^\circ) = \begin{bmatrix} 9 & 3 & 3 & 0 \\ 3 & 9 & 3 & 1 \\ 3 & 3 & 15 & 3 \\ 0 & 1 & 3 & 3 \end{bmatrix}$$

(d)

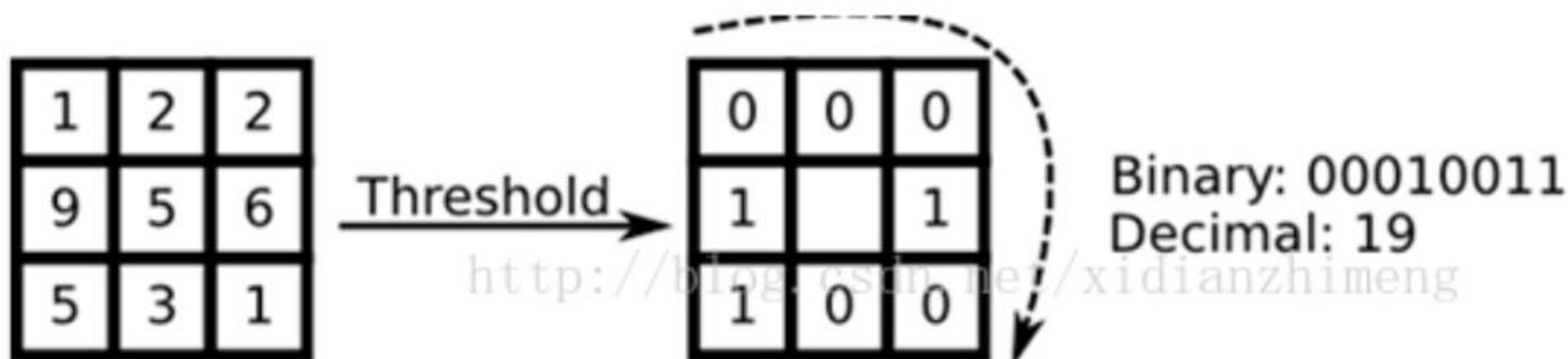
局部二值模式特征 (Local Binary Pattern, LBP)

- LBP算子定义为在 $3*3$ 的窗口内，以窗口中心像素为阈值，将相邻的8个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为1，否则为0。这样， $3*3$ 邻域内的8个点经比较可产生8位二进制数（通常转换为十进制数即LBP码，共256种），即得到该窗口中心像素点的LBP值，并用这个值来反映该区域的纹理信息（例如亮点和暗点）
- 最终得到LBP图，然后计算直方图





$$(01111100)_{10} = 124$$



视觉词袋 (Bag-of-words, BOW)

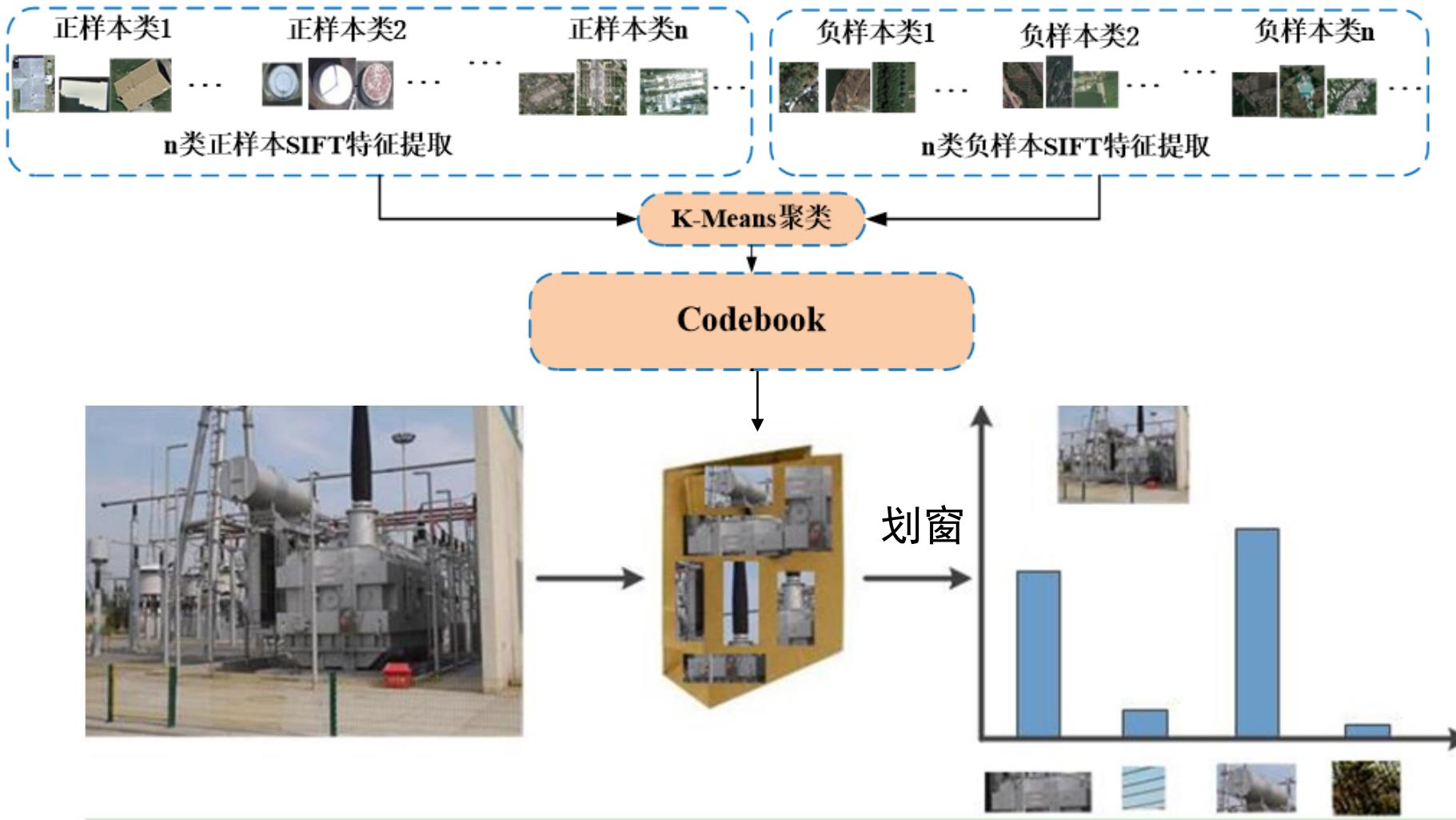
词袋是一种统计关键词出现频率的文档表示方法，2004年被引入计算机视觉领域。词袋模型将图像局部特征空间进行量化，得到一组离散的具有代表性的特征向量，成为视觉单词，视觉单词的集合称为视觉词典，通常采用的视觉词典的构建方法是利用K-means算法对局部特征进行聚类，每个聚类中心对应于一个视觉词汇，视觉词袋方法将图像中的局部特征匹配到距离最近的一个视觉词汇，用视觉词汇出现频率的直方图表示图像。

特征袋 (Bag-of-Features, BoF) :

该模型将多幅图像作为文档集合，分别提取出文档集合中单个文档的局部特征，然后采用聚类算法生成码书 (codebook)，每个聚类的中心即代表一个视觉词汇，码书则表示为视觉词汇的集合，通过量化提取的输入图像特征来计算属于码书中视觉词汇的统计直方图，即为图像的特征袋模型。



基于特征袋模型的分类器



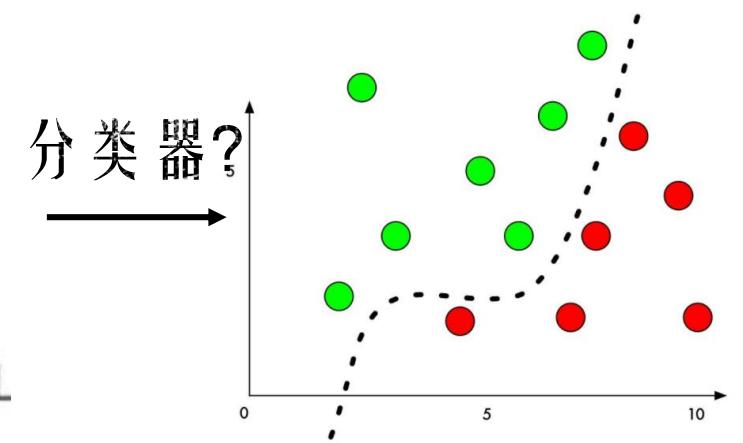
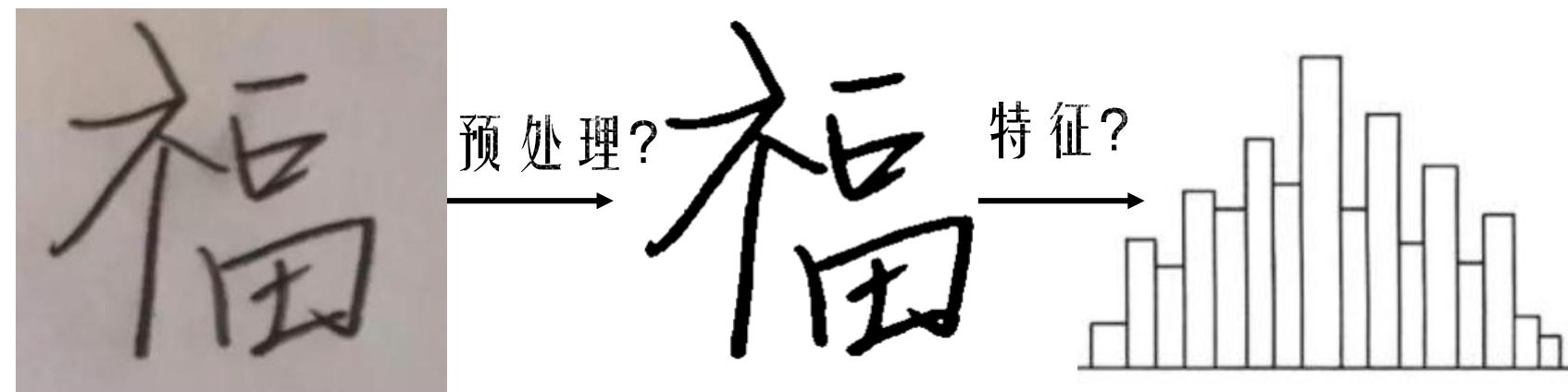
图像

视觉词汇字典

直方图



大作业1：福字识别





THE END!

