课程7：
聚类以及k近邻

聚类

- 聚类任务
- 性能度量
- 距离计算
- 聚类算法

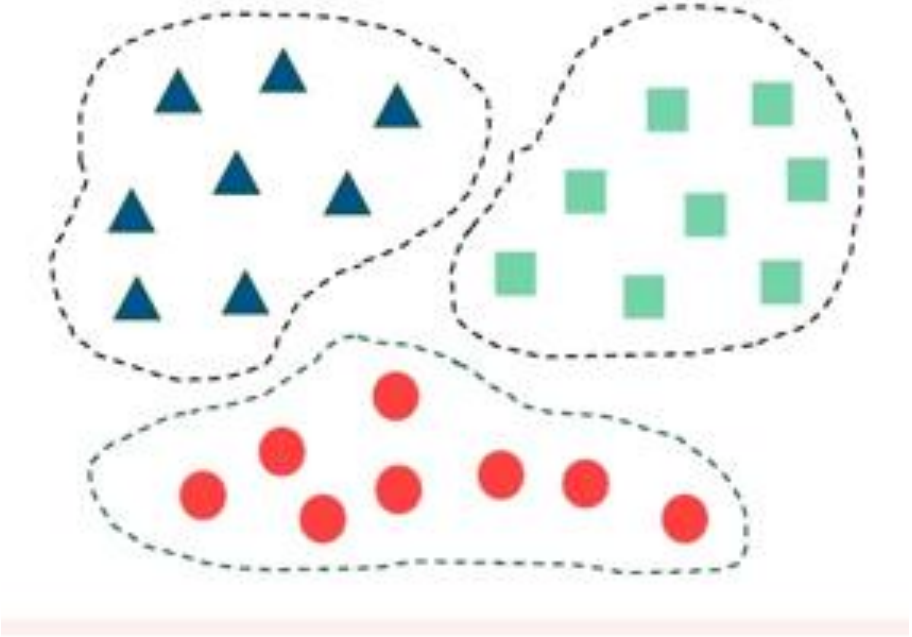
聚类任务

- 在“无监督学习”任务中研究最多、应用最广

目标：将数据样本划分为若干个通常不相交的“簇”（cluster）

既可以作为一个单独过程（用于找寻数据内在的分布结构）

也可作为分类等其他学习任务的前驱过程



两大基本问题

- ✓ 性能度量 —— 什么是好的聚类
- ✓ 距离计算 —— 样本间距离怎么确定

性能度量

- 评估聚类结果的好坏并确立优化的目标

思想：聚类结果的 簇内相似度（intra-cluster similarity）高且 簇间相似度（inter-cluster similarity）低。

聚类性能度量大致有两类：

✓ 聚类结果与某个 参考模型（reference model）进行比较（例如将领域专家给出的划分结果作为参考模型），称为 外部指标（external index）。

✓ 直接考察聚类结果而不利用任何参考模型，称为 内部指标（internal index）。

外部指标

对数据集 $\boldsymbol{D}=\{\boldsymbol{x_1},\boldsymbol{x_2},\cdots,\boldsymbol{x_m}\}$，假定通过聚类给出的簇划分为$\boldsymbol{C}=\{C_1,C_2\cdots,C_k\}$；参考模型给出的簇划分为$\boldsymbol{C}^*=\{C_1^*,C_1^*,\cdots,C_s^*\}$。相应地，令 λ 与 λ^*分别表示与$\boldsymbol{C}$ 和$\boldsymbol{C}^*$对应的簇标记向量。把样本两两配对考虑，定义

四个集合互不相交

$$a=\left|SS\right|,SS=\left\{(x_i,x_j)\,|\,\lambda_i=\lambda_j,\lambda_i^*=\lambda_j^*,i<j\right\}$$

$$b=\left|SD\right|,SD=\left\{(x_i,x_j)\,|\,\lambda_i=\lambda_j,\lambda_i^*\neq\lambda_j^*,i<j\right\}$$

$$c=\left|DS\right|,DS=\left\{(x_i,x_j)\,|\,\lambda_i\neq\lambda_j,\lambda_i^*=\lambda_j^*,i<j\right\}$$

$$d=\left|DD\right|,DD=\left\{(x_i,x_j)\,|\,\lambda_i\neq\lambda_j,\lambda_i^*\neq\lambda_j^*,i<j\right\}$$

其中集合 SS 包含了在 C 中隶属于相同簇且在 C^* 中也隶属于相同簇的样本对，集合 SD 包含了在C中隶属于相同簇但在 C^* 中隶属于不同簇的样本对，……由于每个样本对 (x_i,x_j) $(i<j)$ 仅能出现在一个集合中，因此有 $a+b+c+d=m(m-1)/2$ 成立。

外部指标

基于上式，我们可以导出常用的聚类性能度量外部指标

- ✓ Jaccard系数（Jaccard Coefficient，简称JC）

$$JC = \frac{a}{a+b+c}$$

- ✓ FM指数（Fowlkes and Mallows Index，简称FMI）

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$

- ✓ Rand指数（Rand Index，简称RI）

$$RI = \frac{2(a+d)}{m(m-1)}$$

为了使聚类结果与参考模型更加接近，上述性能度量值应越大越好

内部指标

考虑簇划分的结果$\boldsymbol{C}=\{C_1,C_2\cdots,C_k\}$，定义

$$avg(C)=\frac{2}{|C|(|C|-1)}\sum_{1\le i<j<|C|}dist(x_i,x_j)$$

$$diam(C)=\max_{1\le i<j<|C|}dist(x_i,x_j)$$

$$d_{\min}(C_i,C_j)=\min_{x_i\in C_i,x_j\in C_j}dist(x_i,x_j)$$

$$d_{cen}(C_i,C_j)=dist(\mu_i,\mu_j)$$

$dist(.,.)$用于计算两个样本建的距离，μ表示簇C内样本的中心点（簇内所有样本的平均值）

基于上式，我们可以导出常用的聚类性能度量内部指标

✓ DB指数

$$DBI=\frac{1}{k}\sum_{i=1}^{k}\max\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(C_i,C_i)}\right)$$

✓ Dunn指数

$$DI=\min_{1\le i\le k}\left\{\min_{j\ne i}\left(\frac{d_{\min}(C_i,C_j)}{\max_{1\le l\le k}diam(C_l)}\right)\right\}$$

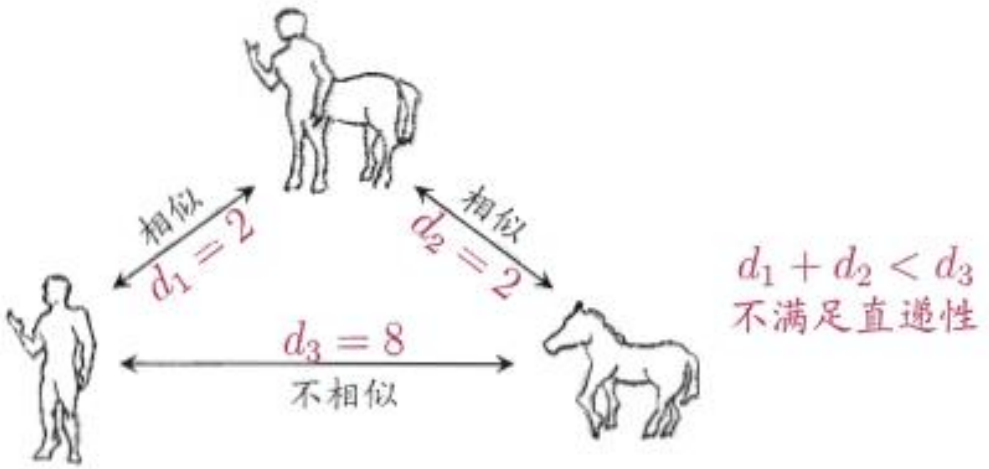
根据簇内样本相似度高的原则，DBI的值越小越好，而DI则相反，值越大越好

距离计算

- 在聚类算法和性能度量中涉及到样本之间的距离计算，因此有必要对其进行讨论

对函数$dist(\cdot,\cdot)$，若它是一个“距离度量”（distance measure），需要满足以下性质：

- ✓ 非负性：$dist(x_i, x_j) \geq 0$
- ✓ 同一性：$dist(x_i, x_j) = 0$当且仅当x_i=x_j
- ✓ 对称性：$dist(x_i, x_j) = dist(x_j, x_i)$
- ✓ 直递性：$dist(x_i, x_j) \leq dist(x_i, x_j) + dist(x_i, x_j)$

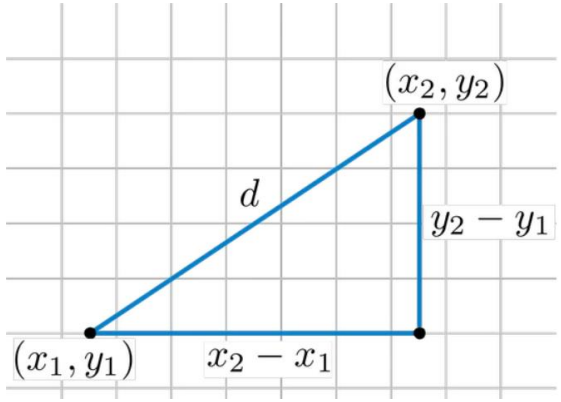


非度量距离的一个例子

距离计算

最常用的距离 —— “闵可夫斯基距离”（Minkowski distance）

$$dist_{mk}(x_i, x_j) = \left(\sum_{u=1}^{n} \left| x_{iu} - x_{ju} \right|^{p} \right)^{\frac{1}{p}}$$

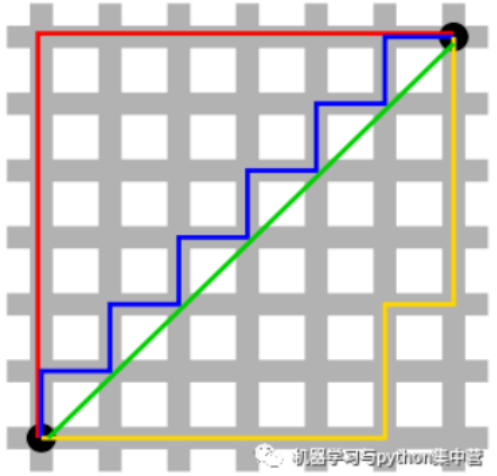


当p=2时，闵可夫斯基距离即欧式距离

$$dist_{ed}(x_i, x_j) = \left\| x_i - x_j \right\|_2$$

当p=1时，闵可夫斯基距离即曼哈顿距离

$$dist_{man}(x_i, x_j) = \left\| x_i - x_j \right\|_1$$



距离计算

某些样本属性(例如定义域为{飞机，火车，轮船})无法使用数字大小进行衡量，即无法定义有序关系，这类属性称为无序属性

无序属性如何计算距离?

对无序属性可采用VDM。令$m_{u,a}$表示在属性u上取值为a的样本数，$m_{u,a,i}$表示在第i个样本簇中在属性u上取值为a的样本数，k为样本簇数，则属性u上两个离散值a与b之间的VDM距离为

$$VDM_p(a,b)=\sum_{i=1}^{k}\left|\frac{m_{u,a,i}}{m_{u,a}}-\frac{m_{u,b,i}}{m_{u,b}}\right|^p$$

若样本同时包含有序和无序属性，可将闵可夫斯基距离与VDM距离结合起来处理

必须记住

聚类的“好坏”不存在绝对标准

the goodness of clustering depends on the opinion of the user

常见的聚类算法

- 原型聚类
  - ✓ 亦称“基于原型的聚类”(prototype-based clustering)
  - ✓ 假设：聚类结构能通过一组原型刻画
  - ✓ 过程：先对原型初始化，然后对原型进行迭代更新求解
  - ✓ 代表：k均值聚类，学习向量量化(LVQ)，高斯混合聚类
- 密度聚类
  - ✓ 亦称“基于密度的聚类”(density-based clustering)
  - ✓ 假设：聚类结构能通过样本分布的紧密程度确定
  - ✓ 过程：从样本密度的角度来考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇
  - ✓ 代表：DBSCAN, OPTICS, DENCLUE
- 层次聚类
  - ✓ 假设：能够产生不同粒度的聚类结果
  - ✓ 过程：在不同层次对数据集进行划分，从而形成树形的聚类结构
  - ✓ 代表：AGNES（自底向上），DIANA（自顶向下）

k-means算法

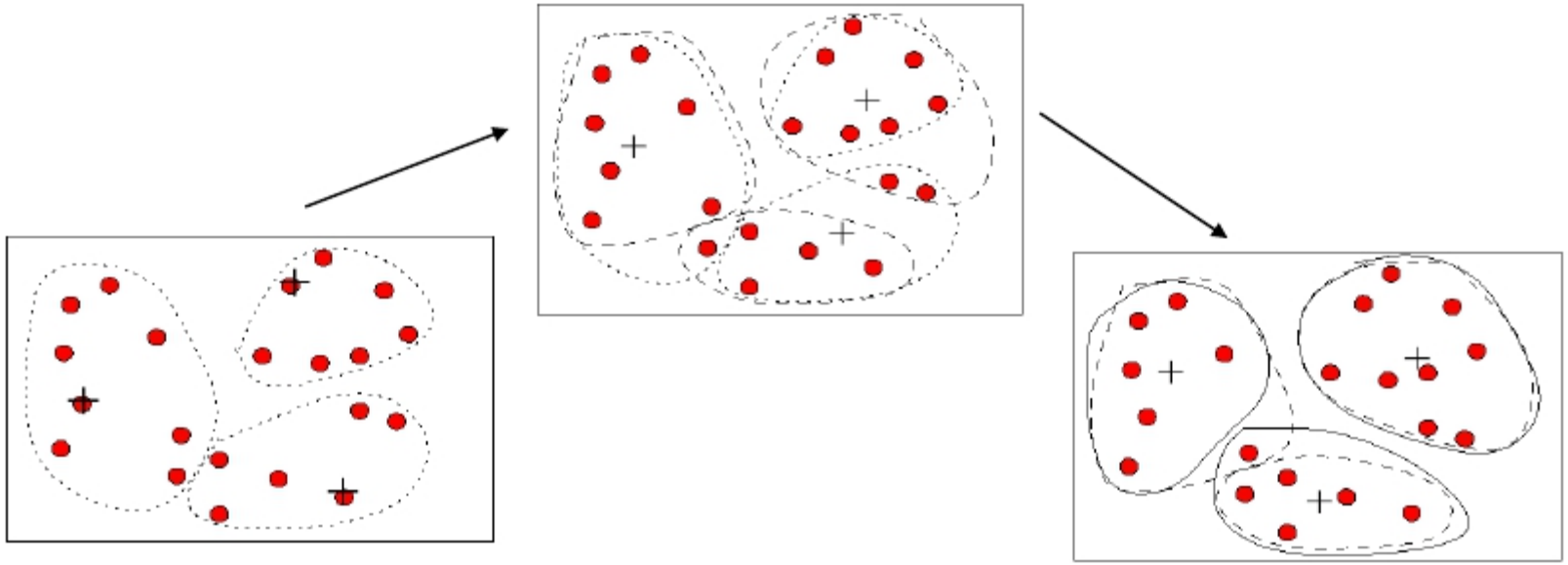
思想：每个簇以该簇中所有样本点的“均值”表示

Step1：随机选取k个样本点作为簇中心

Step2：将其他样本点根据其与簇中心的距离，划分给最近的簇

Step3：更新各簇的均值向量，将其作为新的簇中心

Step4：若所有簇中心未发生改变，则停止；否则执行 Step 2



k-means算法

以西瓜数据集举例说明，假定聚类簇数$k = 3$，

① 随机选取三个样本x_6, x_{12}, x_{27}作为初始均值向量，即

$\mu_1 = (0.403,0237)\ \ \mu_2 = (0.343,0.099)\ \ \mu_3 = (0.532,0.472\ )$

② 计算x_1与各均值向量距离，分别为0.369,0.506,0.166，x_1与μ_3的距离最近，因此x_1被划入到簇C_3中。对其他样本做相同的操作，可得簇划分为

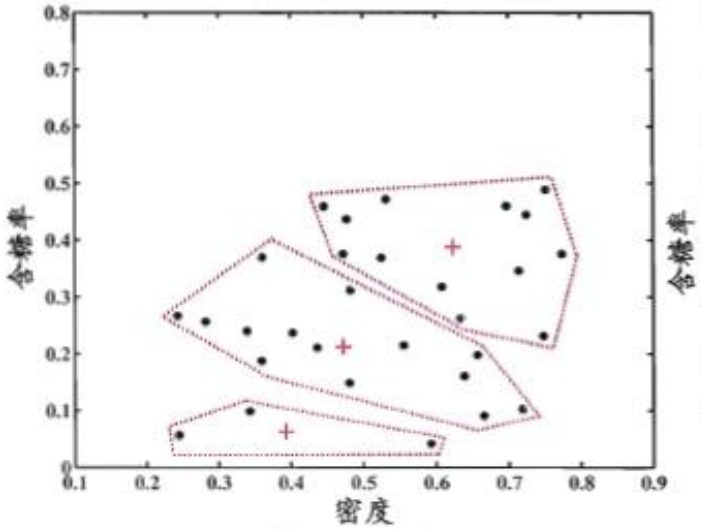
$$C_1 = \{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{23}\}$$
$$C_2 = \{x_{11}, x_{12}, x_{16}\}$$
$$C_3 = \{x_1, x_2, x_3, x_4, x_{21}, x_{22}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\}$$

③ 从C_1、C_2、C_3求出新的均值向量 $\mu_1' = (0.403,0237)\ \ \mu_2' = (0.343,0.099)\ \mu_3' = (0.532,0.472\ )$

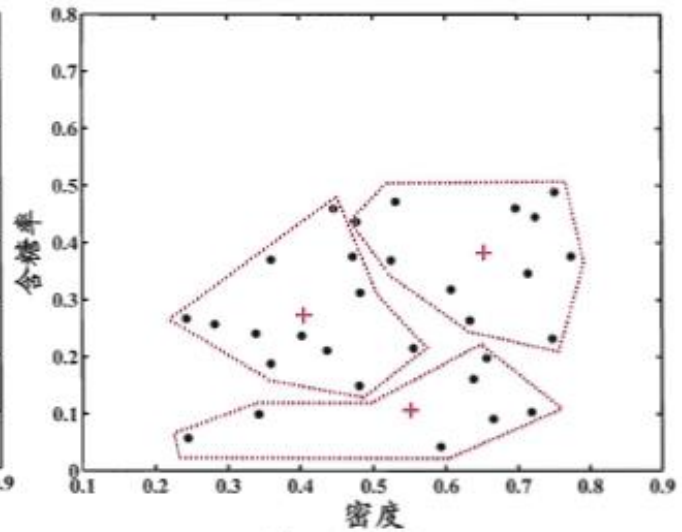
编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459

西瓜数据集

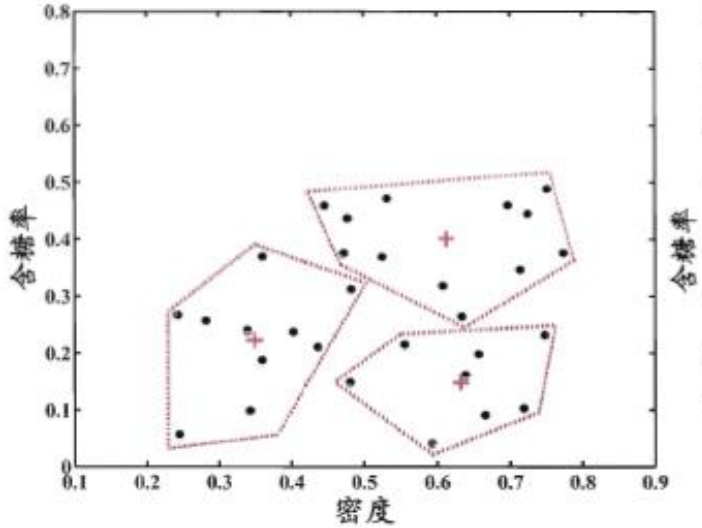
重复上述过程，直至簇中心不发生改变或迭代次数到达上限值



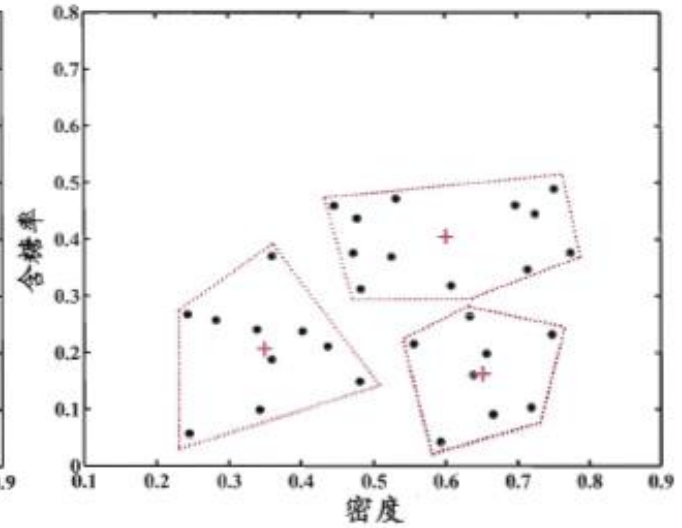
(a) 第一轮迭代后



(b) 第二轮迭代后



(c) 第三轮迭代后



(d) 第四轮迭代后

迭代过程中簇划分发生变化

学习向量化(LVQ)

也是试图找到一组原型向量来刻画聚类结构，但假设数据样本带有类别标记实际上是通过聚类来形成类别的“子类”结构，每个子类对应一个聚类簇

输入: 样本集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
原型向量个数 q, 各原型向量预设的类别标记 $\{t_1, t_2, \ldots, t_q\}$;
学习率 $\eta \in (0, 1)$.

过程:

1: 初始化一组原型向量 $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$
2: **repeat**
3: 从样本集 D 随机选取样本 $(\boldsymbol{x}_j, y_j)$;
4: 计算样本 $\boldsymbol{x}_j$ 与 $\boldsymbol{p}_i$ $(1 \leqslant i \leqslant q)$ 的距离: $d_{ji} = ||\boldsymbol{x}_j - \boldsymbol{p}_i||_2$;
5: 找出与 $\boldsymbol{x}_j$ 距离最近的原型向量 p_{i^*}, $i^* = \arg\min_{i \in \{1,2,\ldots,q\}} d_{ji}$;
6: **if** $y_j = t_{i^*}$ **then**
7: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} + \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$ $\boldsymbol{x}_j$ 与 $\boldsymbol{p}_{i^*}$ 的类别相同
8: **else**
9: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} - \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$ $\boldsymbol{x}_j$ 与 $\boldsymbol{p}_{i^*}$ 的类别不同
10: **end if**
11: 将原型向量 $\boldsymbol{p}_{i^*}$ 更新为 $\boldsymbol{p}'$
12: **until** 满足停止条件

输出: 原型向量 $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$

和 K-means 的不同：

- ✓ 每个样例有类别标签，即 LVQ 是一种监督式学习；
- ✓ 输出不是每个簇的划分，而是每个类别的原型向量；
- ✓ 每个类别的原型向量不是简单的均值向量，考虑了附近非/同样例的影响。

K近邻

K近邻学习器

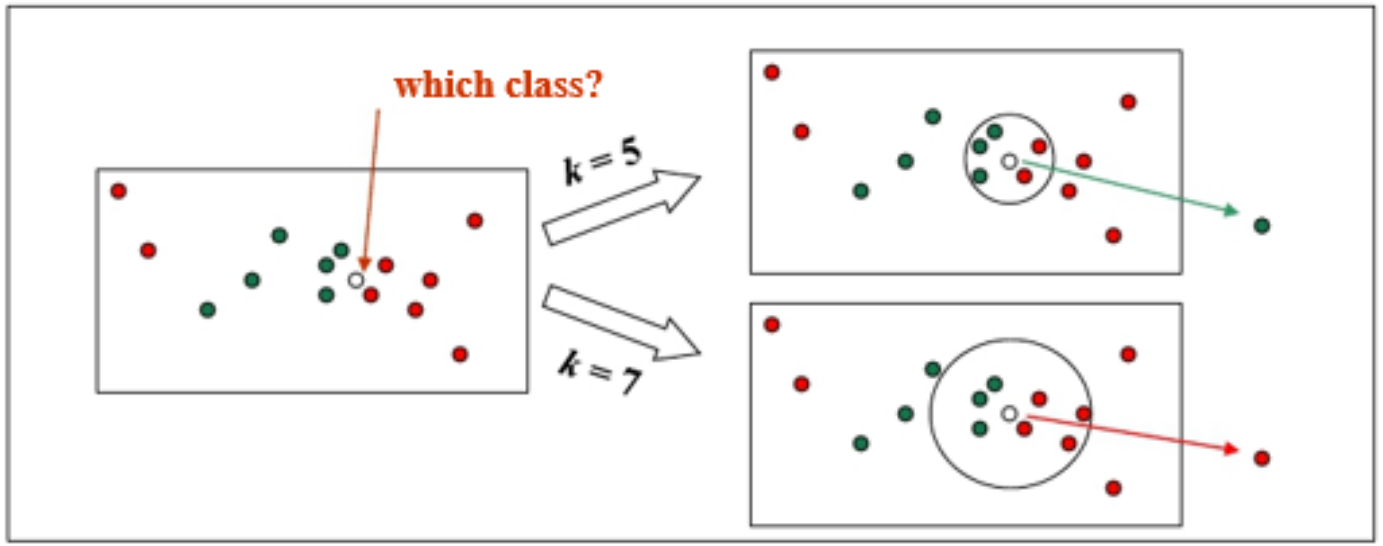
*k*近邻（*k*-Nearest Neighbor, *k*NN)

懒惰学习 （lazy learning）的代表

基本思路：

近朱者赤，近墨者黑

（投票法；平均法）



关键： k 值选取；距离计算

最近邻学习器和贝叶斯最优分类器

给定测试样本$\boldsymbol{x}$，若其最近邻样本为$\boldsymbol{z}$则最近邻分类器出错的概率就是$\boldsymbol{x}$和$\boldsymbol{z}$类别标记不同的概率，即

$$P(err) = 1 - \sum_{c \in y} P(c \mid x) P(c \mid z)$$

$$\simeq 1 - \sum_{c \in y} P^2(c \mid x)$$

$$\leq 1 - P^2(c^* \mid x) = (1 + P(c^* \mid x))(1 - P(c^* \mid x))$$

$$\leq 2 \times (1 - P(c^* \mid x))$$

最近邻分离器的泛化错误率不会超过贝叶斯最优分类器错误率的两倍!

但是在真实的应用中，我们是否能够准确的找到k近邻呢？

维数灾难

但是在真实的应用中，我们是否能够准确的找到*k*近邻呢？

密采样（dense sampling）

如果近邻的距离阈值设为10^{-3}。假定维度为20，如果样本需要满足密采样条件 需要的样本数量近10^{60}！

想象一下：一张并不是很清晰的图像：70余万维
我们为了找到恰当的近邻，需要多少样本？

维数灾难

高维空间给距离计算带来很大的麻烦

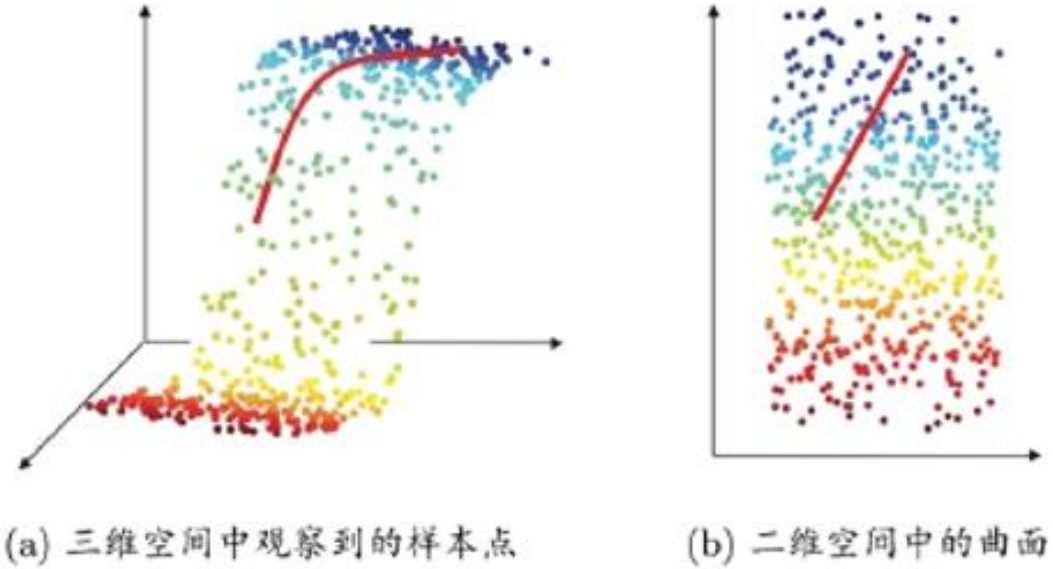
当维数很高时甚至连计算内积都不再容易

更严重的是：样本变得稀疏

提高k近邻算法效率的办法：

✓ 降维

✓ 利用树结构对K近邻算法改进



(a) 三维空间中观察到的样本点　(b) 二维空间中的曲面

参考资料：

1. A. Rodriguez, A. Laio. Clustering by fast search and find of density peaks[J], Science 344, 1492 (2014);
2. I. S. Dhillon, Y. Guan and B. Kulis. Kernel k-means: Spectral clustering and normalized cuts[C]. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD), 2004:551-556
3. E. Castro. Why so many clustering algorithms - a position paper[J]. SIGKDD Explorations, 2002:65-75
4. S. Guha, R. Rastogi and K. Shim. ROCK: A robust clustering algorithm for categorical attributes[C]. In Proceedings of the 15th International Conference on Data Engineering(ICDE), 1999:512-521

相关论文会放到课程网页中，如有需要请自行下载。