高级机器学习

统计机器学习方法回顾

大纲

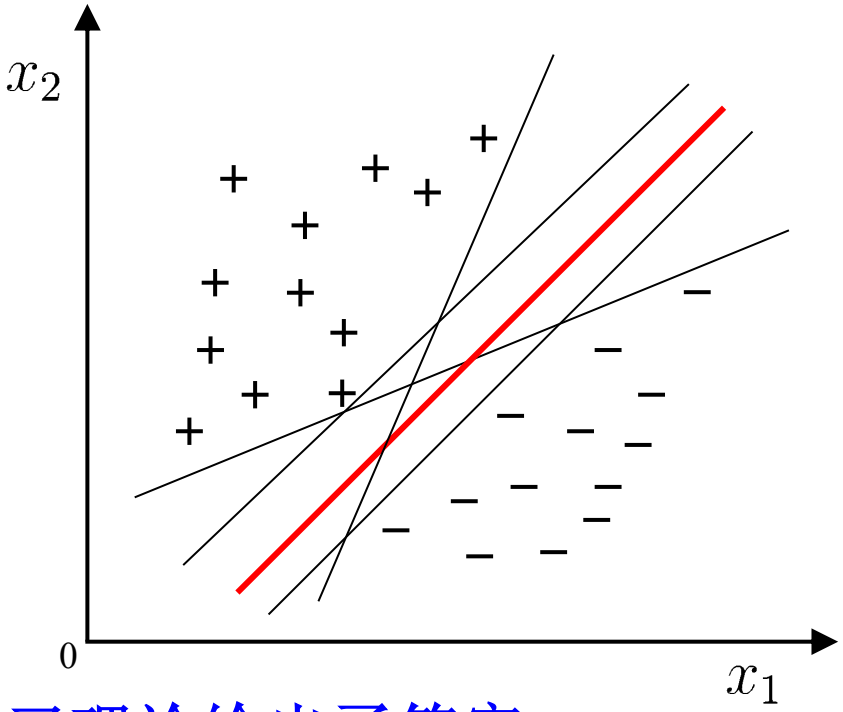
- 支持向量机
- 决策树
- 贝叶斯分类器
- 聚类
- 降维与度量学习

大纲

- 支持向量机
- 决策树
- 贝叶斯分类器
- 聚类
- 降维与度量学习

线性模型

将训练样本分开的超平面可能有很多，哪一个更好呢?

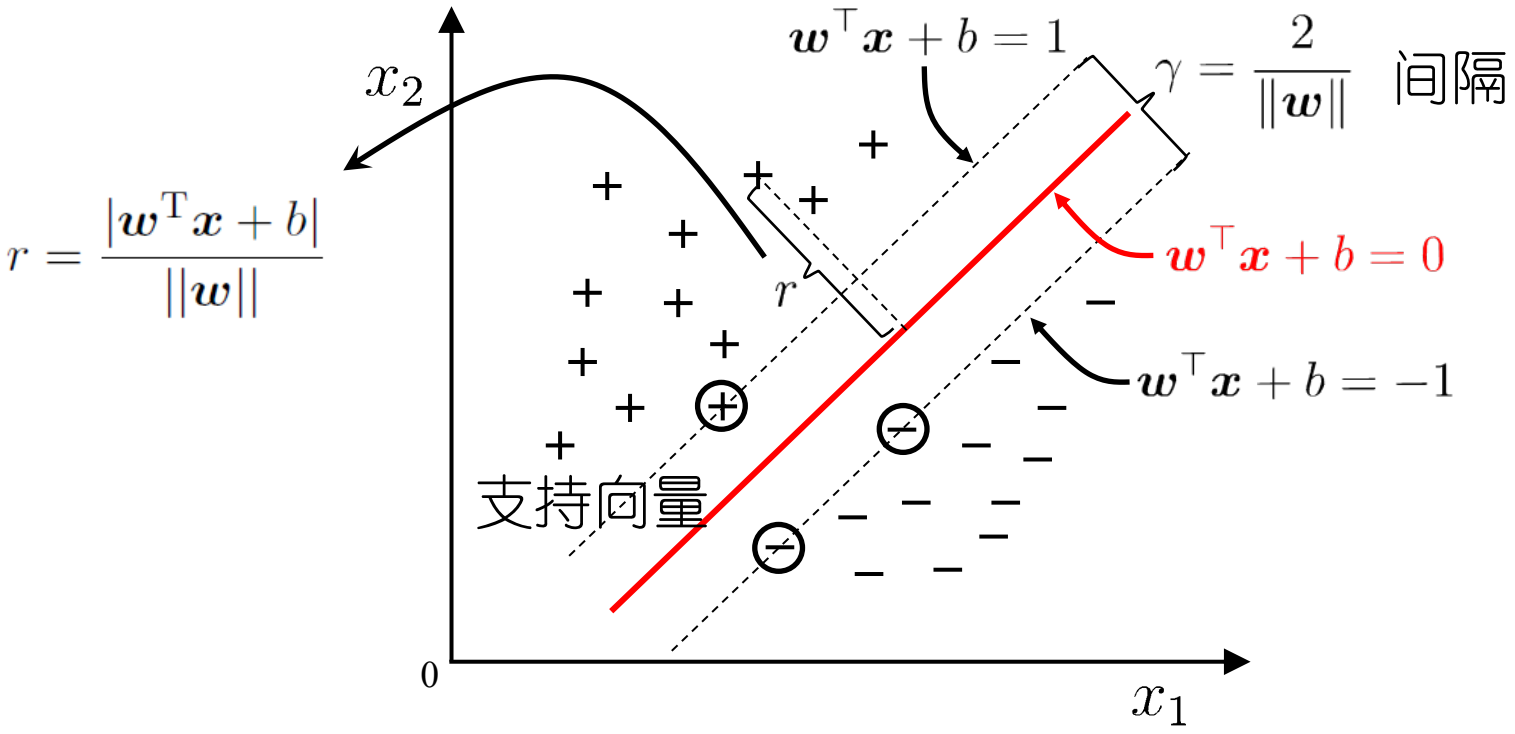


Vapnik的统计学习理论给出了答案：

"正中间"的：鲁棒性最好，泛化能力最强

间隔(margin)与支持向量(support vector)

超平面方程: $\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$

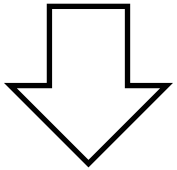


支持向量机 也被称为 “大间隔分类器”

支持向量机-基本型

最大间隔：寻找参数 $\boldsymbol{w}$ 和 b，使得 γ 最大

$$\begin{aligned} \arg\max_{\boldsymbol{w},b} \quad & \frac{2}{\|\boldsymbol{w}\|} \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m. \end{aligned}$$



$$\begin{aligned} \arg\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m. \end{aligned}$$

凸二次规划问题，能用优化计算包求解，但可以有更高效的办法

支持向量机-对偶型

拉格朗日乘子法

- 第一步：引入拉格朗日乘子 $\alpha_i \geq 0$ 得到拉格朗日函数

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b)\right)$$

- 第二步：令 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 对 $\boldsymbol{w}$ 和 b 的偏导为零可得

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \ , \quad 0 = \sum_{i=1}^{m} \alpha_i y_i$$

- 第三步：回代可得

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \ , \quad \alpha_i \geqslant 0 \ , \quad i = 1, 2, \ldots, m$$

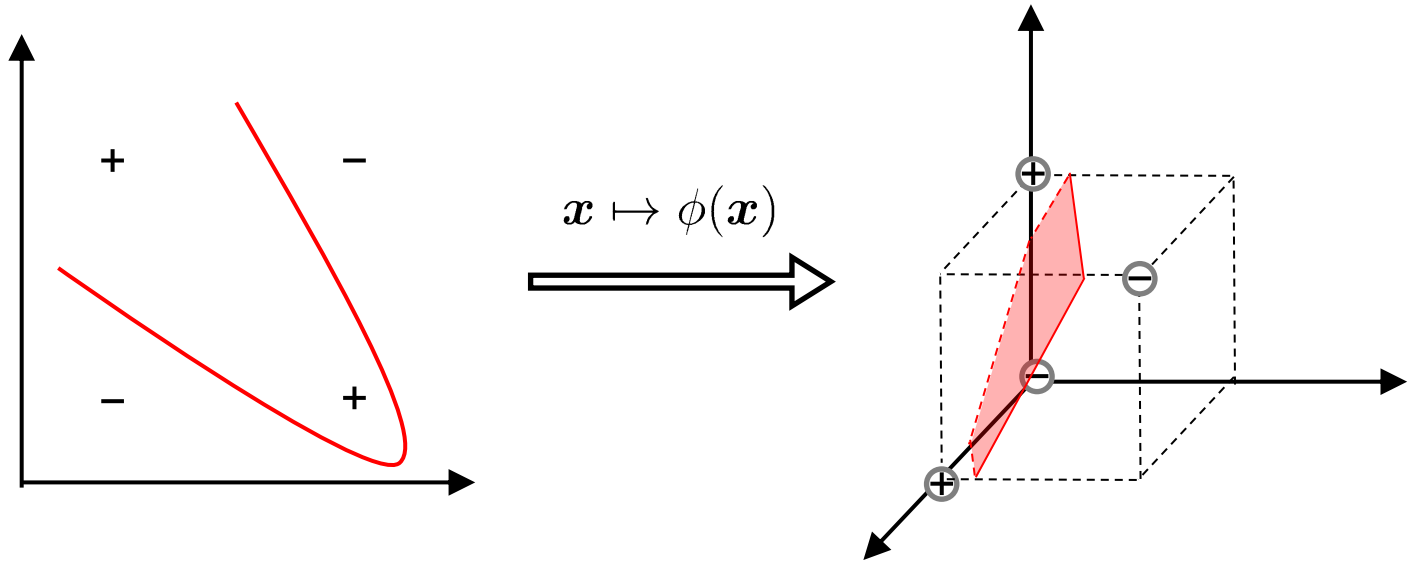
基本型和对偶型的比较

- 基本型和对偶型都是二次凸规划问题，根据凸优化理论，都可以利用成熟数值优化软件包得到全局最优解。
- 基本型和对偶型各有千秋，主要取决于效率和效果的需要
- 一般来说，基本型比较善于处理 1）低维数据；2）高维稀疏数据（所谓‘稀疏’指样本大部分属性的值为零）
- 对偶型比较善于处理高维稠密数据，此外对偶型容易吸收核函数处理非线性分类

特征空间映射

若不存在一个能正确划分两类样本的超平面，怎么办?

将样本从原始空间映射到一个更高维的特征空间，使样本在这个特征空间内线性可分



如果原始空间是有限维(属性数有限)，那么一定存在一个高维特征空间使样本可分

在特征空间中

设样本 $\boldsymbol{x}$ 映射后的向量为 $\phi(\boldsymbol{x})$，划分超平面为 $f(\boldsymbol{x}) = \boldsymbol{w}^{\top}\phi(\boldsymbol{x}) + b$

原始问题

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^{\top}\phi(\boldsymbol{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, m. \end{aligned}$$

对偶问题

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0 \ , \quad \alpha_i \geqslant 0 \ , \quad i = 1, 2, \ldots, m \end{aligned}$$

预测

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}) + b$$

只以内积形式出现

核函数 (kernel function)

基本思路：设计核函数

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j)$$

绕过显式考虑特征映射、以及计算高维内积的困难

Mercer 定理：若一个对称函数所对应的核矩阵半正定，则它就能作为核函数来使用

任何一个核函数，都隐式地定义了一个RKHS（Reproducing Kernel Hilbert Space，再生核希尔伯特空间）

“核函数选择”成为决定支持向量机性能的关键！

核函数 (kernel function)

常用核函数

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j)^d$	$d \geqslant 1$ 为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j + \theta)$	tanh 为双曲正切函数, $\beta > 0, \theta < 0$

基本经验：文本数据常用线性核，
情况不明时可先尝试高斯核

可通过函数组合得到：

若 κ_1 和 κ_2 是核函数，则对任意正数 γ_1、γ_2 和任意函数 $g(\boldsymbol{x})$，

均为核函数

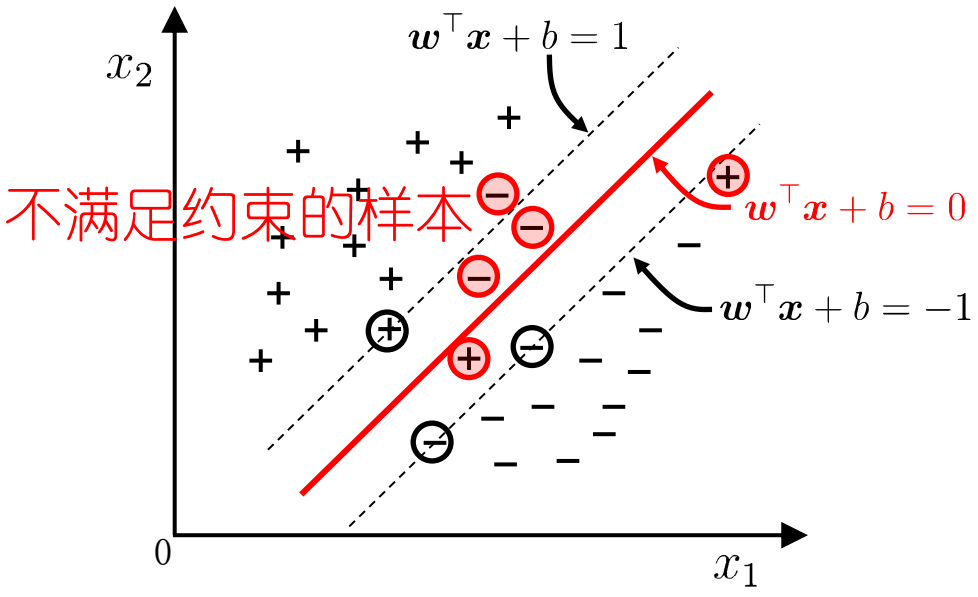
$$\begin{cases} \gamma_1 \kappa_1 + \gamma_2 \kappa_2 \\ \kappa_1 \otimes \kappa_2(\boldsymbol{x}, \boldsymbol{z}) = \kappa_1(\boldsymbol{x}, \boldsymbol{z}) \kappa_2(\boldsymbol{x}, \boldsymbol{z}) \\ \kappa(\boldsymbol{x}, \boldsymbol{z}) = g(\boldsymbol{x}) \kappa_1(\boldsymbol{x}, \boldsymbol{z}) g(\boldsymbol{z}) \end{cases}$$

软间隔支持向量机

现实中很难确定合适的核函数，使训练样本在特征空间中线性可分

即便貌似线性可分，也很难断定是否是因过拟合造成的

引入软间隔（soft margin），允许在一些样本上不满足约束



软间隔支持向量机

基本思路：最大化间隔的同时，

让不满足约束 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1$ 的样本尽可能少

$$\min_{\boldsymbol{w},b}\ \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)-1\right)$$

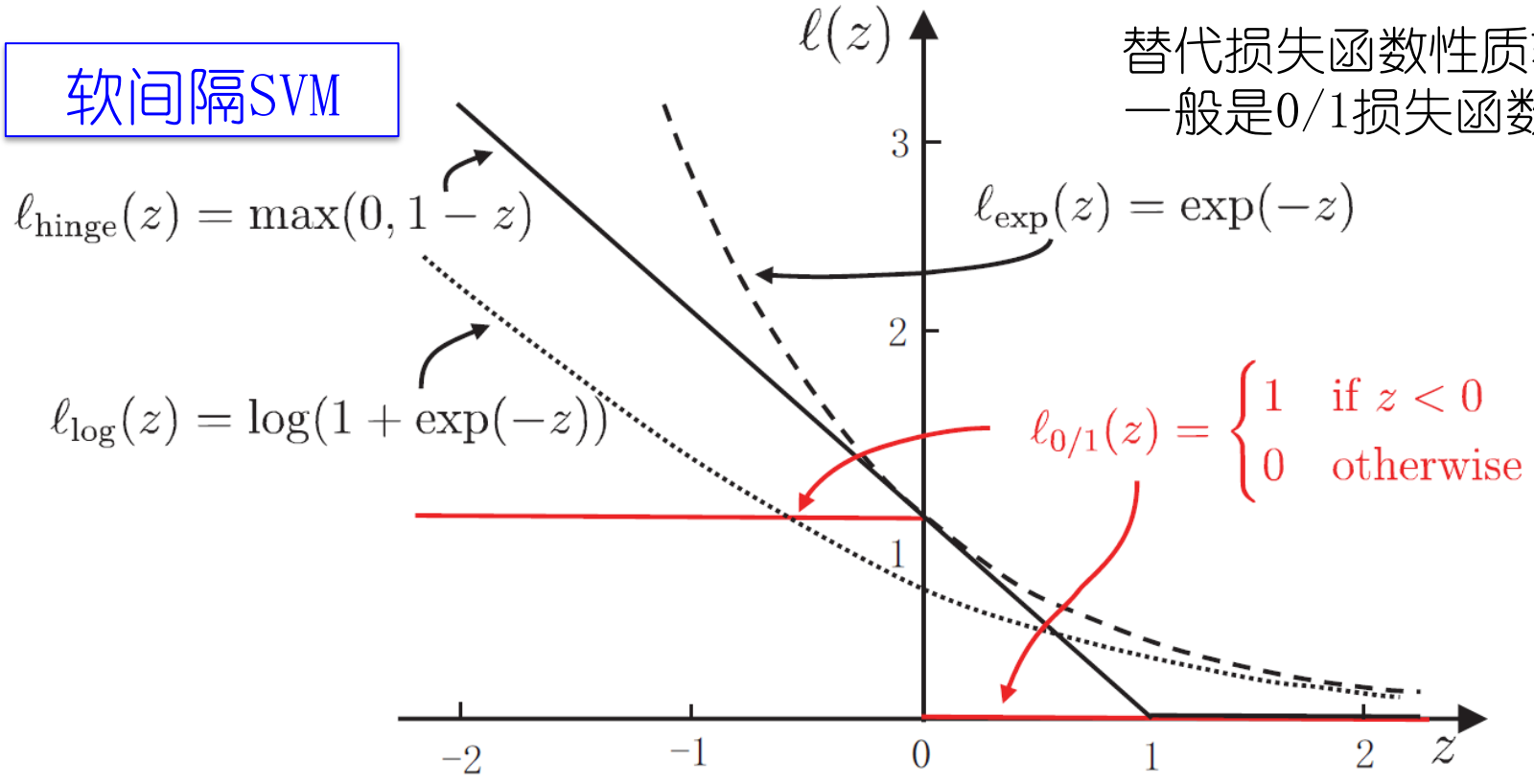
其中 $l_{0/1}$ 是 0/1损失函数：

$$\ell_{0/1}(z)=\begin{cases}1, & \text{if } z<0;\\ 0, & \text{otherwise.}\end{cases}$$

障碍：0/1损失函数非凸、非连续，不易优化！

替代损失(surrogate loss)

软间隔SVM



替代损失函数性质较好，
一般是0/1损失函数的上界

- 采用替代损失函数，是在解决困难问题时的常见技巧
- 求解替代函数得到的解是否仍是原问题的解？理论上称为替代损失的“一致性”（consistency）问题

软间隔支持向量机-基本型和对偶型

原始问题

$$\min_{\boldsymbol{w},b} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\max\left(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right)$$

引入“松弛变量” (slack variables) ξ_i

$$\begin{aligned}\min_{\boldsymbol{w},b,\xi_i} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \ & y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1 - \xi_i \ , \ \xi_i \geqslant 0 \ , \ i = 1, 2, \ldots, m.\end{aligned}$$

对偶问题

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \ & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\ \text{s.t.} \ & \sum_{i=1}^{m}\alpha_i y_i = 0 \ , \ 0 \leqslant \alpha_i \leqslant C \ , \quad i = 1, 2, \ldots, m \ .\end{aligned}$$

与“硬间隔SVM”的区别

正则化（regularization）

统计学习模型（例如SVM）的更一般形式

正则化项

$$\min_f \ \Omega(f) + C\sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i)$$

归纳偏好

结构风险
(structural risk)
描述模型本身的某些性质

经验风险
(empirical risk)
描述模型与训练数据的契合程度

- 正则化可理解为“罚函数法”
  通过对不希望的结果施以惩罚，使得优化过程趋向于希望目标
- 从贝叶斯估计的角度，则可认为是提供了模型的先验概率

现实应用中…

如何使用SVM？

- 入门级——实现并使用各种版本SVM
- 专业级——尝试、组合核函数
- 专家级——根据问题而设计目标函数、替代损失、进而……

根据当前任务“量身定制”是关键

现实应用中…

`sklearn.svm`: Support Vector Machines

The `sklearn.svm` module includes Support Vector Machine algorithms.

User guide: See the Support Vector Machines section for further details.

Estimators

`svm.LinearSVC`([penalty, loss, dual, tol, C, ...])	Linear Support Vector Classification.
`svm.LinearSVR`(*[, epsilon, tol, C, loss, ...])	Linear Support Vector Regression.
`svm.NuSVC`(*[, nu, kernel, degree, gamma, ...])	Nu-Support Vector Classification.
`svm.NuSVR`(*[, nu, C, kernel, degree, gamma, ...])	Nu Support Vector Regression.
`svm.OneClassSVM`(*[, kernel, degree, gamma, ...])	Unsupervised Outlier Detection.
`svm.SVC`(*[, C, kernel, degree, gamma, ...])	C-Support Vector Classification.
`svm.SVR`(*[, kernel, degree, gamma, coef0, ...])	Epsilon-Support Vector Regression.

`svm.l1_min_c`(X, y, *[, loss, fit_intercept, ...])	Return the lowest bound for C.

大纲

- 支持向量机
- **决策树**
- 贝叶斯分类器
- 聚类
- 降维与度量学习

决策树模型

决策树基于“树”结构进行决策

- 每个“内部结点”对应于某个属性上的“测试”（test）
- 每个分支对应于该测试的一种可能结果（即该属性的某个取值）
- 每个“叶结点”对应于一个“预测结果”

学习过程：通过对训练样本的分析来确定“划分属性”（即内部结点所对应的属性）

预测过程：将测试示例从根结点开始，沿着划分属性所构成的“判定测试序列”下行，直到叶结点

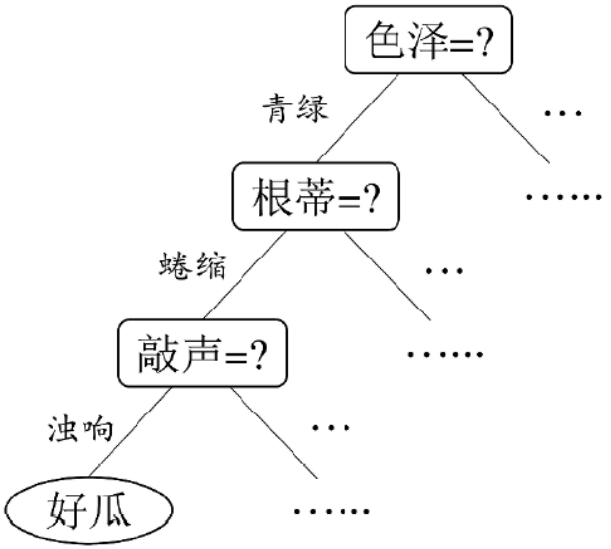


图 4.1 西瓜问题的一棵决策树

基本流程

策略：“分而治之”（divide-and-conquer）

自根至叶的递归过程

在每个中间结点寻找一个“划分”（split or test）属性

三种停止条件：

（1）当前结点包含的样本全属于同一类别，无需划分；

（2）当前属性集为空，或是所有样本在所有属性上取值相同，无法划分；

（3）当前结点包含的样本集合为空，不能划分.

划分选择

决策树学习的关键在于选择最优划分标准

- 直觉上，决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”（purity）越高越好

- 经典划分方法：
  - 信息增益（information gain）
  - 增益率（gain ratio）
  - 基尼指数（gini index）

信息增益

信息熵 (entropy) 是度量样本集合“纯度”最常用的一种指标

假定当前样本集合 D 中第 k 类样本所占的比例为 p_k，则 D 的信息熵定义为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

计算信息熵时约定：若 $p = 0$，则 $p \log_2 p = 0$.

$\mathrm{Ent}(D)$ 的最小值为 0，最大值为 $\log_2 |\mathcal{Y}|$.

$\mathrm{Ent}(D)$ 的值越小，则 D的纯度越高

信息增益直接以信息熵为基础，计算当前划分对信息熵所造成的变化

信息增益

离散属性 a 的取值：$\{a^1, a^2, \ldots, a^V\}$

D^v：D 中在 a 上取值 $= a^v$ 的样本集合

以属性 a 对 数据集 D 进行划分所获得的信息增益为：

$$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

划分前的信息熵

划分后的信息熵

第 v 个分支的权重，
样本越多越重要

增益率

信息增益：对可取值数目较多的属性有所偏好

有明显弱点，例如：考虑将“编号”作为一个属性

增益率：
$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

其中
$$\text{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

属性 a 的可能取值数目越多（即 V 越大），则 $\text{IV}(a)$ 的值通常就越大

启发式：先从候选划分属性中找出信息增益高于平均水平的，再从中选取增益率最高的

基尼指数

$$\begin{aligned}\mathrm{Gini}(D) &= \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} \\ &= 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 \ .\end{aligned}$$

反映了从 D 中随机抽取两个样例，其类别标记不一致的概率

Gini(D) 越小，数据集 D 的纯度越高

属性 a 的基尼指数：$\mathrm{Gini\_index}(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Gini}(D^v)$

在候选属性集合中，选取那个使划分后基尼指数最小的属性

过拟合和剪枝

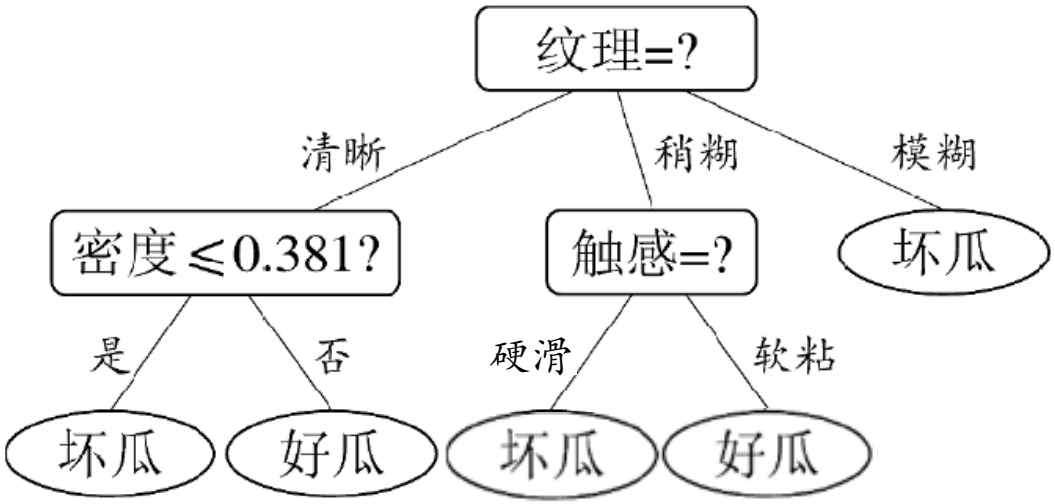
- 决策树的不足，过拟合
  - 决策树决策分支过多，以致于把训练集自身的一些特点当做所有数据都具有的一般性质而导致的过拟合
- 剪枝的基本策略
  - 预剪枝：边建树，边剪枝
  - 后剪枝：先建树，后剪枝
- 判断决策树泛化性能是否提升的方法
  - 留出法：预留一部分数据用作“验证集”以进行性能评估

连续值

基本思路：连续属性离散化

常见做法：二分法 (bi-partition)

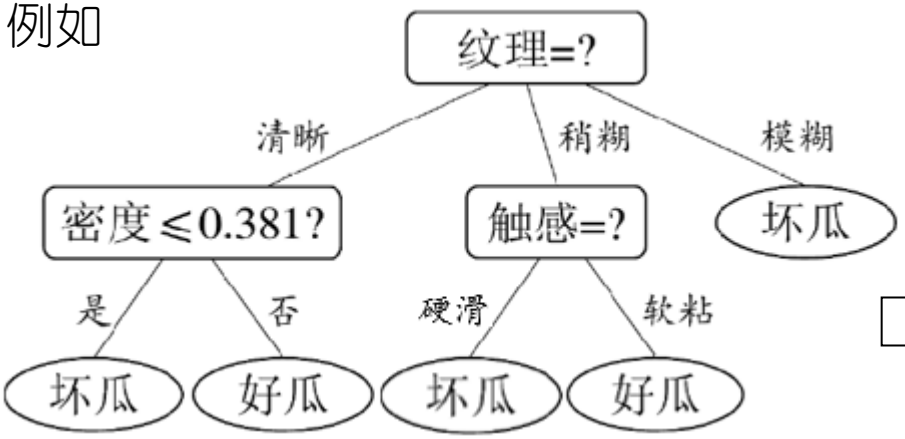
- n 个属性值可形成 n-1 个候选划分
- 然后即可将它们当做 n-1 个离散属性值处理



从树到规则

- 一棵决策树对应于一个“规则集”
- 每个从根结点到叶结点的分支路径对应于一条规则

例如



- IF (纹理=清晰) ∧ (密度≤0.381) THEN 坏瓜
- IF (纹理=清晰) ∧ (密度>0.381) THEN 好瓜
- IF (纹理=稍糊) ∧ (触感=硬滑) THEN 坏瓜
- IF (纹理=稍糊) ∧ (触感=软粘) THEN 好瓜
- IF (纹理=模糊) THEN 坏瓜

好处：

- 改善可理解性
- 进一步提升泛化能力

大纲

- 支持向量机
- 决策树
- 贝叶斯分类器
- 聚类
- 降维与度量学习

贝叶斯决策论 (Bayesian decision theory)

概率框架下实施决策的基本理论

给定 N 个类别，令 λ_{ij} 代表将第 j 类样本误分类为第 i 类所产生的损失，则基于后验概率将样本 $\boldsymbol{x}$ 分到第 i 类的条件风险为：

$$R(c_i \mid \boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j \mid \boldsymbol{x})$$

贝叶斯判定准则 (Bayes decision rule)：

$$h^*(\boldsymbol{x}) = \arg\min_{c \in \mathcal{Y}} R(c \mid \boldsymbol{x})$$

- h^* 称为贝叶斯最优分类器(Bayes optimal classifier)，其总体风险称为贝叶斯风险 (Bayes risk)
- 反映了学习性能的理论上限

判别式 vs. 生成式

$P(c \mid \boldsymbol{x})$ 在现实中通常难以直接获得

从这个角度来看，机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率

两种基本策略：

判别式 (discriminative) 模型

思路：直接对 $P(c \mid \boldsymbol{x})$ 建模

代表：

- 决策树
- 神经网络
- SVM

生成式 (generative) 模型

思路：先对联合概率分布 $P(\boldsymbol{x}, c)$ 建模，再由此获得 $P(c \mid \boldsymbol{x})$

$$P(c \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

代表：贝叶斯分类器

注意：贝叶斯分类器 ≠ 贝叶斯学习 (Bayesian learning)

贝叶斯定理

$$P(c \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

Thomas Bayes
(1701?-1761)

根据贝叶斯定理，有

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\,P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})}$$

先验概率（prior）
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定律）

证据（evidence）
因子，与类别无关

样本相对于类标记的类条件概率（class-conditional probability），亦称 似然（likelihood）

主要困难在于估计似然 $P(\boldsymbol{x} \mid c)$

朴素贝叶斯分类器 (naïve Bayes classifier)

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\,P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})}$$

主要障碍：所有属性上的联合概率难以从有限训练样本估计获得

组合爆炸；样本稀疏

基本思路：假定属性相互独立？

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\,P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})} = \frac{P(c)}{P(\boldsymbol{x})} \prod_{i=1}^{d} P(x_i \mid c)$$

d 为属性数，x_i 为 $\boldsymbol{x}$ 在第 i 个属性上的取值

$P(\boldsymbol{x})$ 对所有类别相同，于是

$$h_{nb}(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^{d} P(x_i \mid c)$$

朴素贝叶斯分类器 (naïve Bayes classifier)

- 估计 $P(c)$：$P(c) = \frac{|D_c|}{|D|}$
- 估计 $P(\boldsymbol{x}|c)$：
  - 对离散属性，令 D_{c,x_i} 表示 D_c 中在第 i 个属性上取值为 x_i 的样本组成的集合，则

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|}$$

  - 对连续属性，考虑概率密度函数，假定 $p(x_i \mid c) \sim \mathcal{N}(\mu_{c,i}, \sigma_{c,i}^2)$

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

大纲

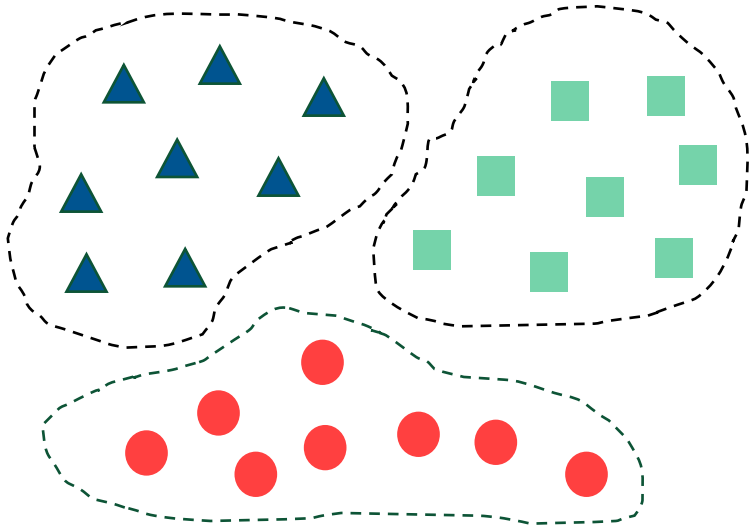
- 支持向量机
- 决策树
- 贝叶斯分类器
- **聚类**
- 降维与度量学习

聚类 (Clustering)

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



聚类 (Clustering)

聚类性能度量，亦称聚类“有效性指标”(validity index)

- 外部指标 (external index)

  将聚类结果与某个“参考模型”(reference model) 进行比较

  如 Jaccard 系数，FM 指数，Rand 指数

- 内部指标 (internal index)

  直接考察聚类结果而不用任何参考模型

  如 DB 指数，Dunn 指数等

基本想法：

- “簇内相似度”（intra-cluster similarity）高，且
- “簇间相似度”（inter-cluster similarity）低

距离计算

距离度量 (distance metric) 需满足的基本性质：

非负性: $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geqslant 0$;

同一性: $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ 当且仅当 $\boldsymbol{x}_i = \boldsymbol{x}_j$;

对称性: $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{dist}(\boldsymbol{x}_j, \boldsymbol{x}_i)$;

直递性: $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leqslant \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_k) + \mathrm{dist}(\boldsymbol{x}_k, \boldsymbol{x}_j)$.

常用距离形式：

闵可夫斯基距离 (Minkowski distance)

$$\mathrm{dist}_{\mathrm{mk}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}$$

$p = 2$: 欧氏距离(Euclidean distance)

$p = 1$: 曼哈顿距离(Manhattan distance)

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

常见聚类方法

- 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 (hierarchical clustering)
  - 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

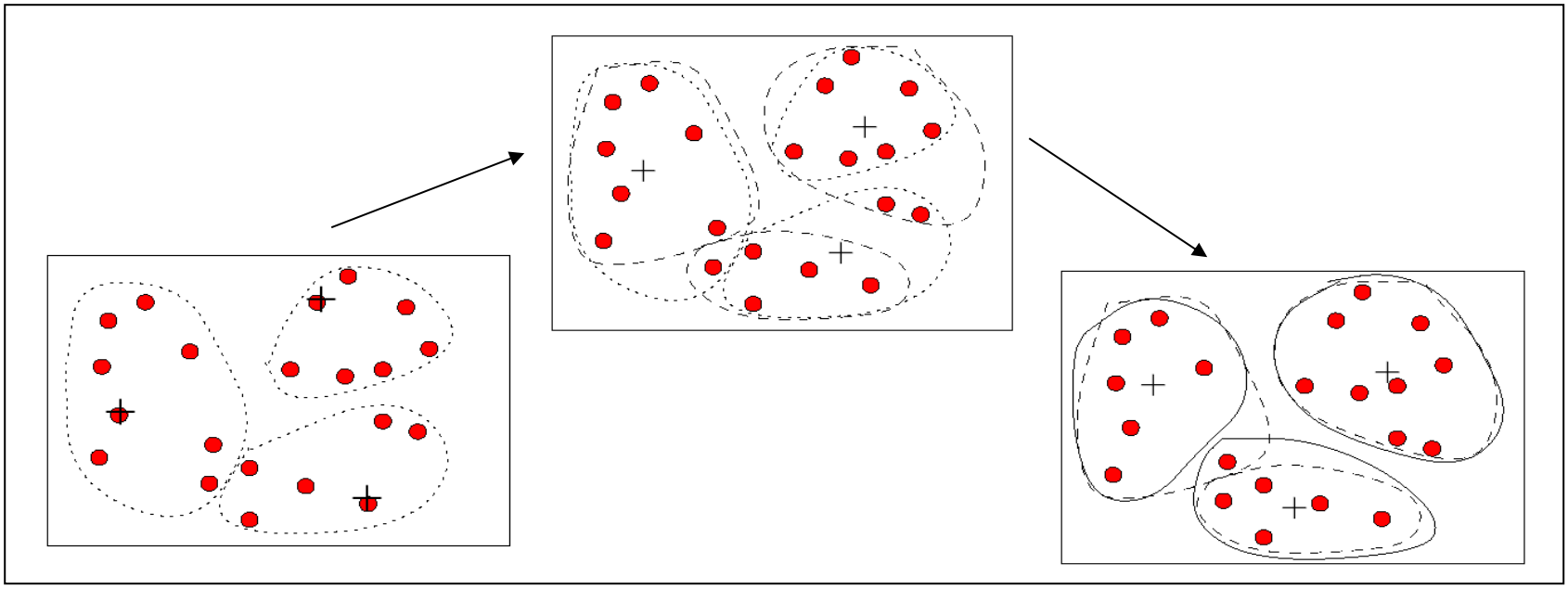
每个簇以该簇中所有样本点的“均值”表示

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



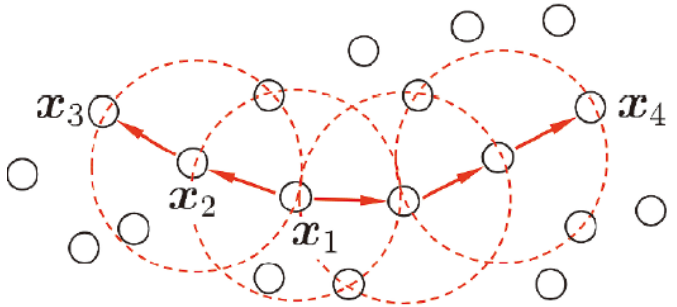
DBSCAN

关键概念：

- 核心对象(core object)：若 $\boldsymbol{x}_j$ 的 ϵ-邻域至少包含 $MinPts$ 个样本，即 $|N_\epsilon(\boldsymbol{x}_j)| \geqslant MinPts$，则 $\boldsymbol{x}_j$ 是一个核心对象；
- 密度直达(directly density-reachable)：若 $\boldsymbol{x}_j$ 位于 $\boldsymbol{x}_i$ 的 ϵ-邻域中，且 $\boldsymbol{x}_i$ 是核心对象，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度直达；
- 密度可达(density-reachable)：对 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$，若存在样本序列 $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$，其中 $\boldsymbol{p}_1 = \boldsymbol{x}_i$，$\boldsymbol{p}_n = \boldsymbol{x}_j$ 且 $\boldsymbol{p}_{i+1}$ 由 $\boldsymbol{p}_i$ 密度直达，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度可达；
- 密度相连(density-connected)：对 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$，若存在 $\boldsymbol{x}_k$ 使得 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 均由 $\boldsymbol{x}_k$ 密度可达，则称 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 密度相连.

令$MinPts = 3$，
虚线显示出ϵ 邻域
x_1 是核心对象

x_2 由 x_1 密度直达
x_3 由 x_1 密度可达
x_3 与 x_4 密度相连

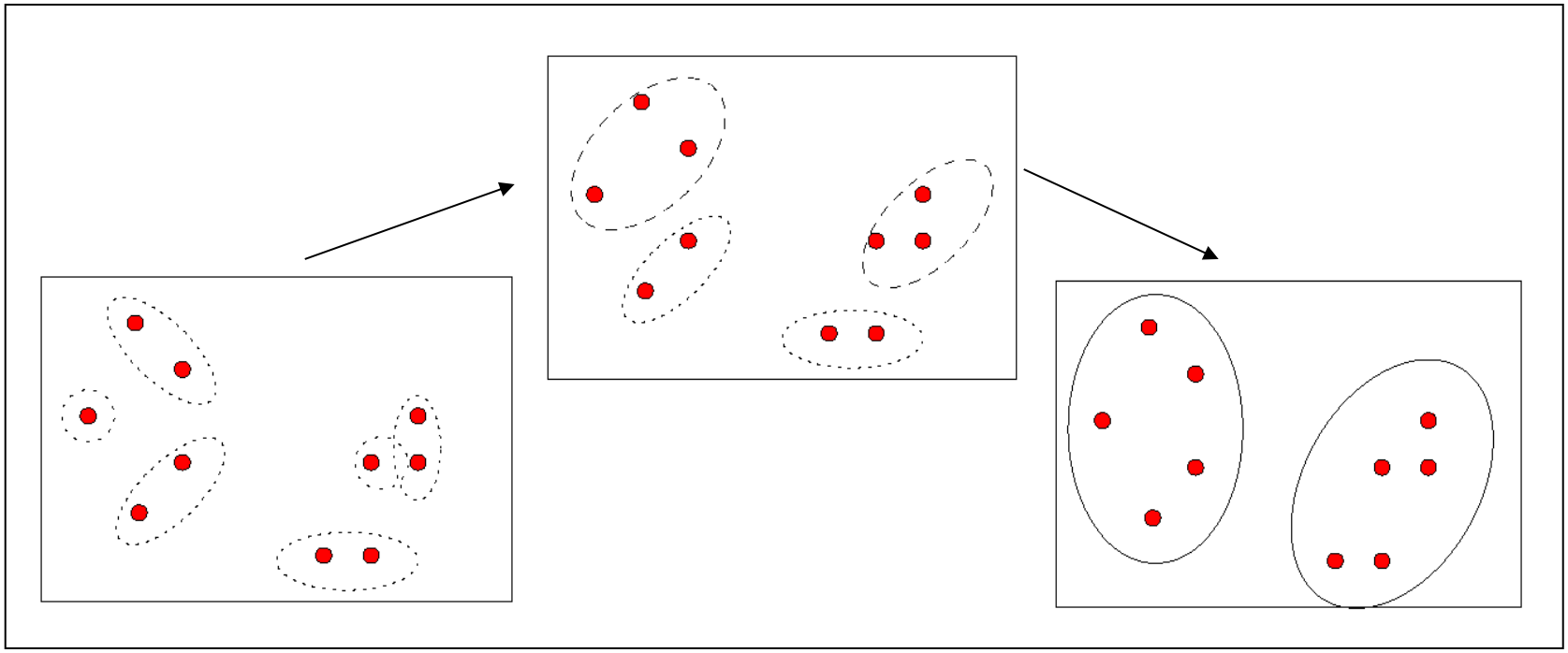


AGNES (AGglomerative NESting)

Step1: 将每个样本点作为一个簇

Step2: 合并最近的两个簇

Step3: 若所有样本点都存在与一个簇中，则停止；否则转到 Step2



AGNES

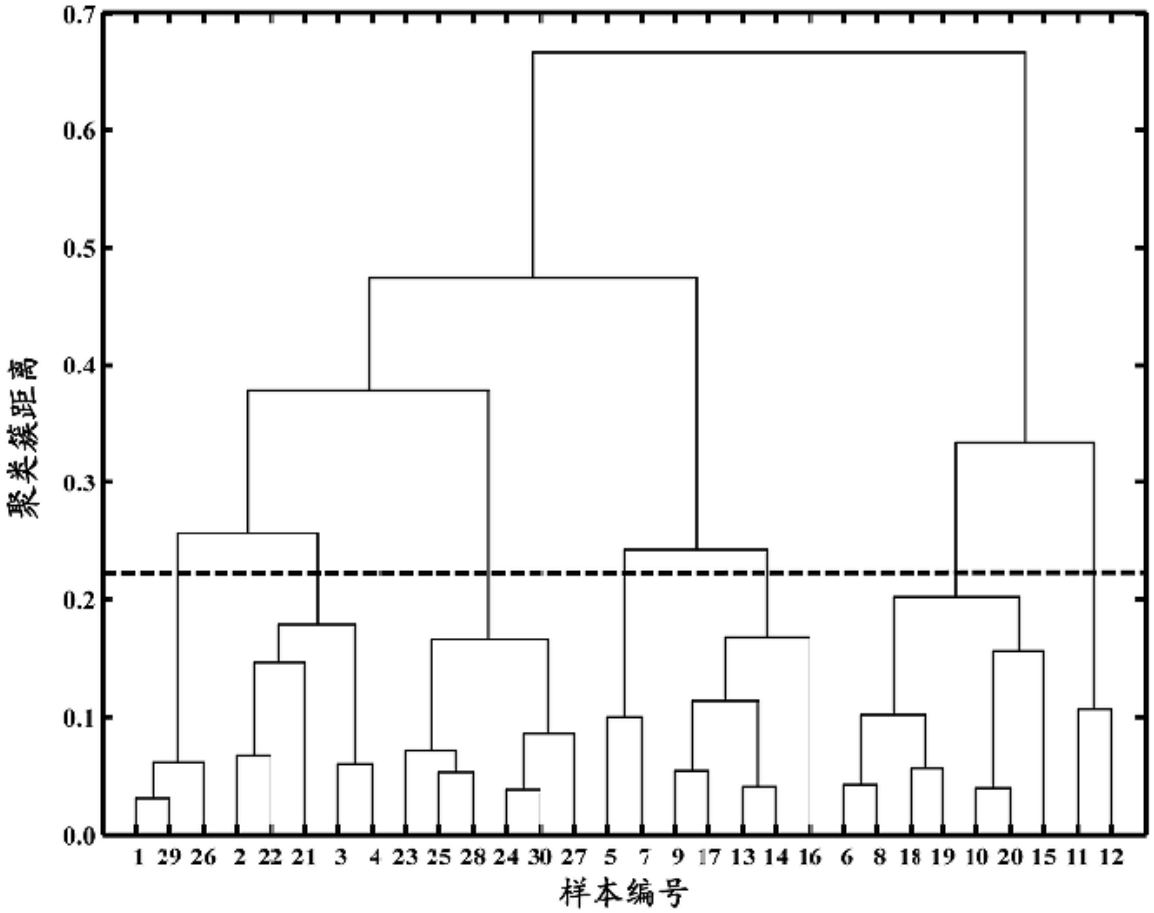


图 9.12 西瓜数据集 4.0 上 AGNES 算法生成的树状图(采用 $d_{\max}$). 横轴对应于样本编号, 纵轴对应于聚类簇距离.

大纲

- 支持向量机
- 决策树
- 贝叶斯分类器
- 聚类
- 降维与度量学习

k 近邻学习器

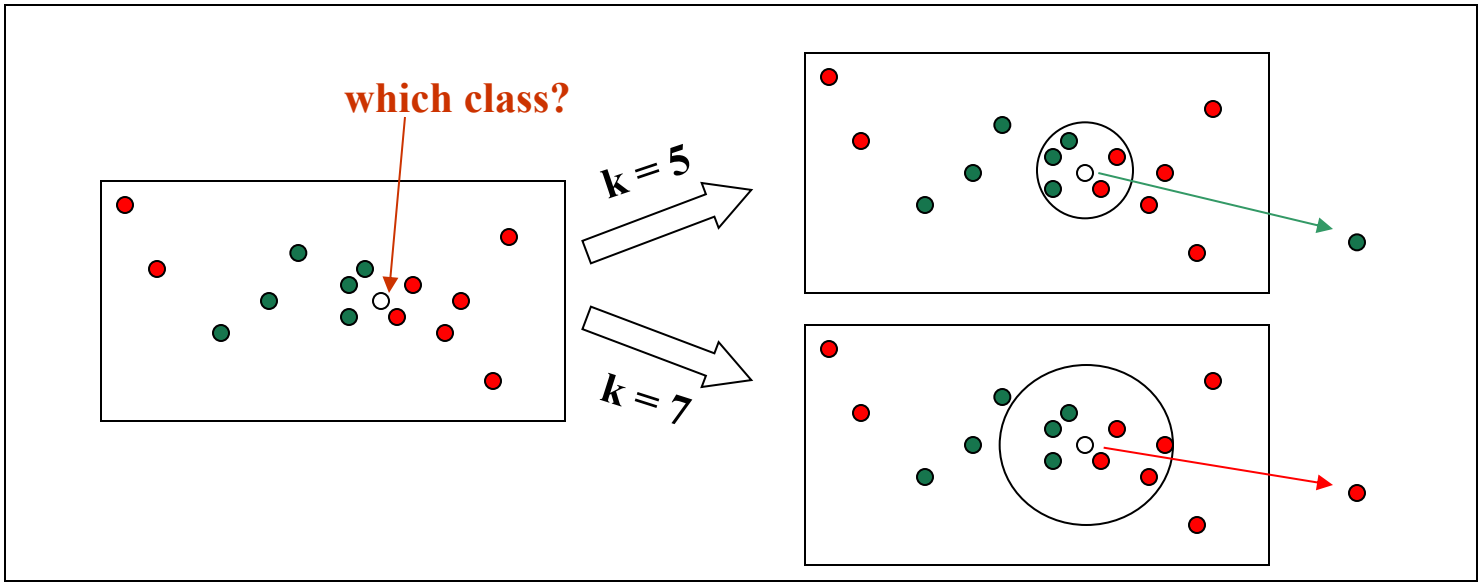
课外知识：
懒惰学习 = 事先没有分类器，见到测试样本再开始准备分类器

k 近邻 (k-Nearest Neighbor, kNN)

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

最近邻学习器重要性质

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，

$$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}).$$

$$\begin{aligned} P(err) &= 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}) \\ &\simeq 1 - \sum_{c \in \mathcal{Y}} P^2(c \mid \boldsymbol{x}) \\ &\leq 1 - P^2(c^* \mid \boldsymbol{x}) \\ &= \left(1 + P\left(c^* \mid \boldsymbol{x}\right)\right)\left(1 - P\left(c^* \mid \boldsymbol{x}\right)\right) \\ &\leq 2 \times \left(1 - P\left(c^* \mid \boldsymbol{x}\right)\right). \end{aligned}$$

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！

kNN如此完美；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)假设：样本的每个ϵ-邻域都有近邻

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

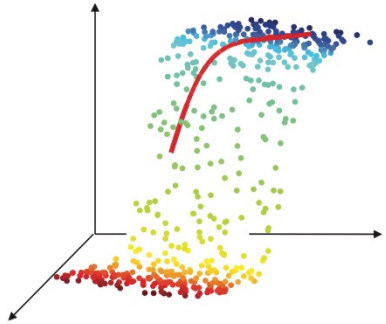
当维数很高时，甚至连计算内积都不再容易

更严重的是：样本变得稀疏，近邻容易不准

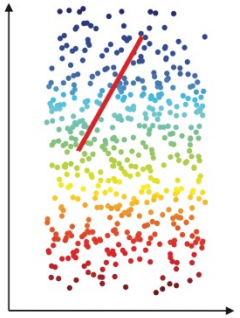
⇒ 降维（将高维样本降到低维）

为什么能进行降维？

数据样本虽是高维的，但与学习任务密切相关的也许仅是某个低维空间，即高维空间中的一个低维"嵌入"(embedding)



(a) 三维空间中观察到的样本点



(b) 二维空间中的曲面

多维缩放方法（MDS, Multiple Dimensional Scaling ）

MDS旨在寻找一个低维子空间，使得距离和样本原有距离近似不变

令原始空间的距离矩阵D，降维后样本的表示为Z，$||\mathrm{z_i}-\mathrm{z_j}||=\mathrm{dist}_{ij}$，令$B=Z^{\mathrm{T}}Z\in\mathbb{R}^{m\times m}$，$b_{ij}=z_iz_j$，假设降维后的样本$Z$被中心化，即$\sum z_i=0$

$$dist_{ij}^2=\|\mathbf{z}_i\|^2+\|\mathbf{z}_j\|^2-2\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_j=b_{ii}+b_{jj}-2b_{ij}$$

$$\sum_{i=1}^{m}dist_{ij}^2=\mathrm{tr}(\mathbf{B})+mb_{jj}\,,\qquad dist_{i\cdot}^2=\frac{1}{m}\sum_{j=1}^{m}dist_{ij}^2\,,$$

$$\sum_{j=1}^{m}dist_{ij}^2=\mathrm{tr}(\mathbf{B})+mb_{ii}\,,\qquad dist_{\cdot j}^2=\frac{1}{m}\sum_{i=1}^{m}dist_{ij}^2\,,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m}dist_{ij}^2=2m\,\mathrm{tr}(\mathbf{B})\,,\qquad dist_{\cdot\cdot}^2=\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}dist_{ij}^2\,,$$

$$b_{ij}=-\frac{1}{2}(dist_{ij}^2-dist_{i\cdot}^2-dist_{\cdot j}^2+dist_{\cdot\cdot}^2)$$

主要思路：内积保距

寻找低维子空间尽量保持样本内积不变

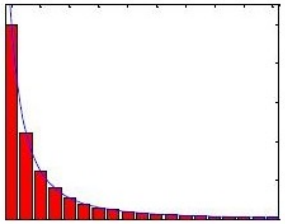
思路：特征值分解（常用技巧）

设样本之间的内积矩阵均为B

对B进行特征值分解：

$$\mathbf{B}=\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

由谱分解的数学性质，我们知道：



特征谱

谱分布长尾：存在相当数量的小特征值

$$\mathbf{Z}=\boldsymbol{\Lambda}_*^{1/2}\mathbf{V}_*^{\mathrm{T}}\in\mathbb{R}^{d^*\times m}$$

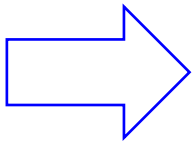
$$\mathbf{B}=\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\in\mathbb{R}^{m\times m}$$

主成分分析（PCA, Principal Component Analysis）

正交属性空间中的样本点，如何使用一个超平面对所有样本进行恰当的表达？

若存在这样的超平面，那么它大概应具有这样的性质：

- 最近重构性：样本点到这个超平面的距离都足够近
- 最大可分性：样本点在这个超平面上的投影能尽可能分开



主成分分析的两种等价推导

PCA-最近重构性

对样本进行中心化：$\sum_i \boldsymbol{x}_i = \mathbf{0}$

假定投影变换后得到的新坐标系为$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$，其中$\boldsymbol{w}_i$是标准正交基向量

$$||\boldsymbol{w}_i||_2 = 1, \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j = 0 (i \neq j).$$

若丢弃新坐标系中的部分坐标，即将维度降低到$d' < d$，则样本点在低维坐标系中的投影是$\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$ $z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$

若基于$\boldsymbol{z}_i$来重构$\boldsymbol{x}_i$，则会得到 $\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j.$

PCA-最近重构性

原样本点 $\boldsymbol{x}_i$ 与基于投影重构的样本点 $\hat{\boldsymbol{x}}_i$ 之间的距离为

$$\sum_{i=1}^{m}\left\|\sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j - \boldsymbol{x}_i\right\|_2^2 = \sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i - 2\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i + \text{const}$$

$$\propto -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\mathbf{W}\right).$$

$\boldsymbol{w}_j$ 是正交基，$\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵，于是由最近重构性，有：

$$\begin{aligned}\min_{\mathbf{W}} \quad & -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.\end{aligned}$$

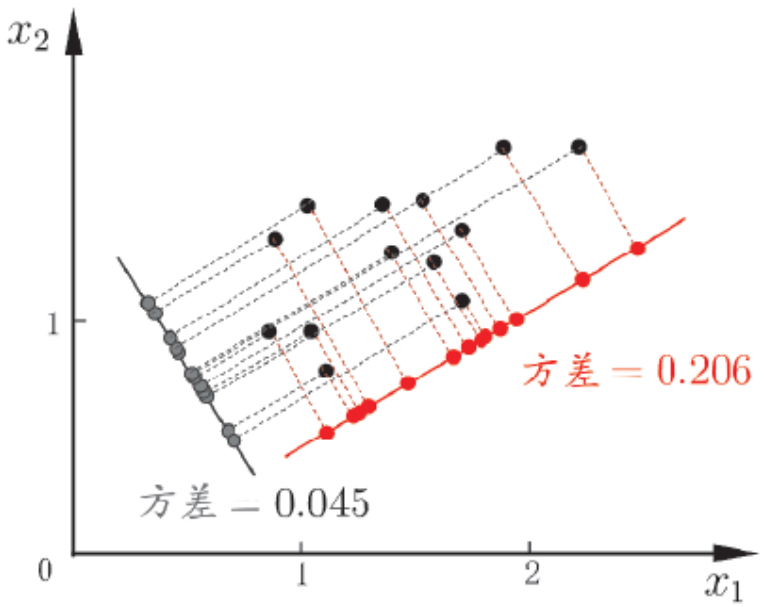
主要思路：重构误差

这就是主成分分析的优化目标

PCA-最大可分性

样本点 $\boldsymbol{x}_i$ 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化

投影后样本点的方差是 $\sum_i \mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\mathbf{W}$



于是：

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

等价于：

$$\min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

PCA求解

$$\begin{aligned} \max_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \end{aligned}$$

使用拉格朗日乘子法可得

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W} = \lambda\mathbf{W}.$$

只需对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 进行特征值分解，并将求得的特征值排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$，再取前 d' 个特征值对应的特征向量构成 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$，这就是主成分分析的解

关键变量：子空间方差

PCA的应用

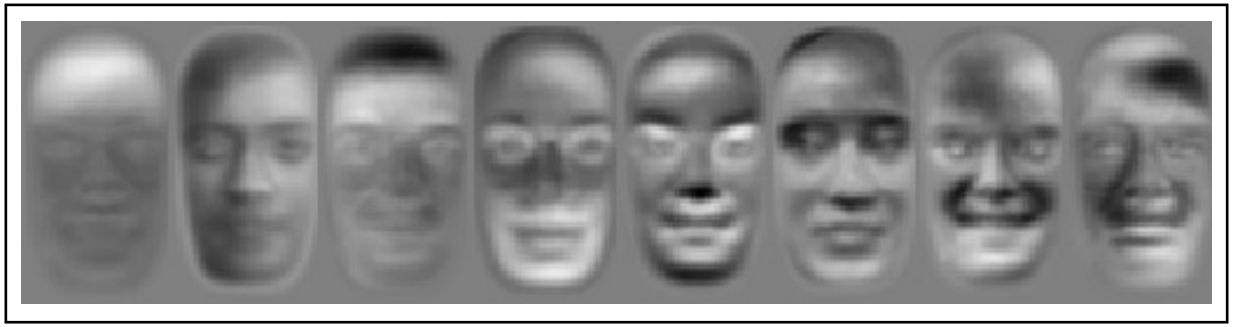
d' 的设置：

- 用户指定
- 在低维空间中对k近邻或其他分类器进行交叉验证
- 设置重构阈值，例如 t =95%，然后选取最小的 d' 使得 $\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t.$

PCA 是最常用的降维方法，在不同领域有不同的称谓

例如在人脸识别中该技术被称为“特征脸”(eigenface)

因为若将前 d′ 个特征值对应的特征向量还原为图像，则得到



流形学习 - ISOMAP

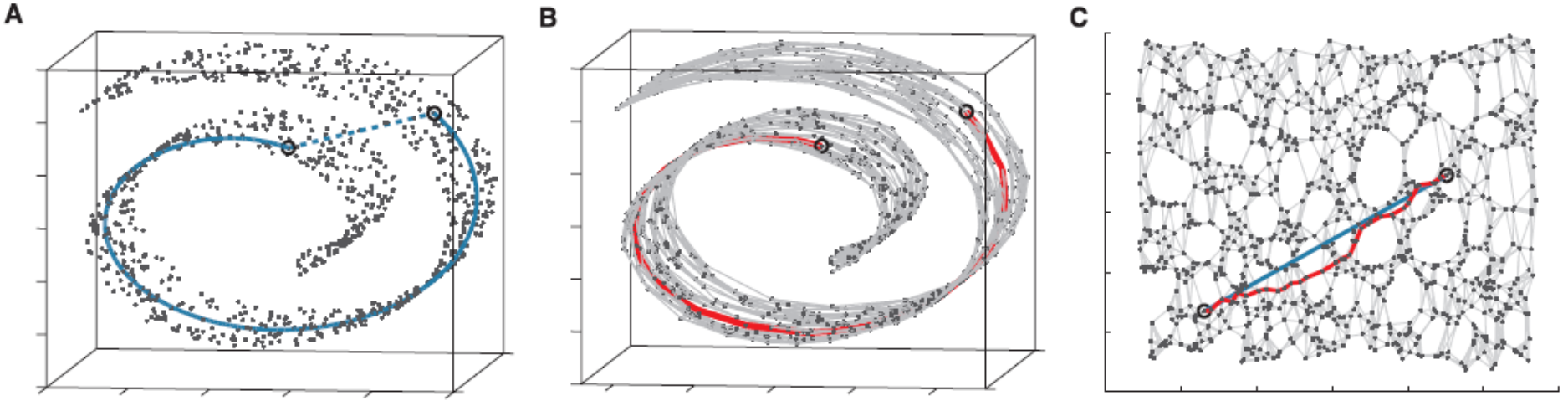


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N =$ 1000 data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

基本步骤：

关键思路：测地线距离（近似）、保距

- 构造近邻图
- 基于最短路径算法近似任意两点之间的测地线(geodesic)距离
- 基于距离矩阵通过MDS获得低维嵌入

流形学习 - LLE

基本步骤：

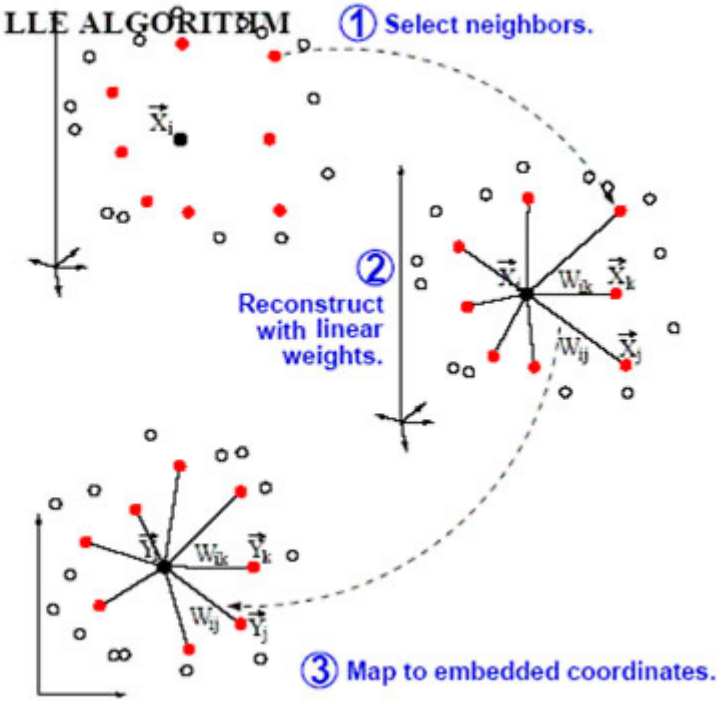
- 为每个样本构造近邻集合Q_i
- 为每个样本计算基于Q_i的线性重构系数

$$\min_{\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{x}_j \right\|_2^2$$
$$\text{s.t.} \sum_{j\in Q_i} w_{ij} = 1,$$

- 在低维空间中保持w_{ij}不变，求解下式

$$\min_{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{z}_j \right\|_2^2$$

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in Equation (1) by constrained linear fits.
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in Equation (2) by its bottom nonzero eigenvectors.



www.sciencemag.org SCIENCE VOL 290 22 DECEMBER 2000

关键思路：重构权值

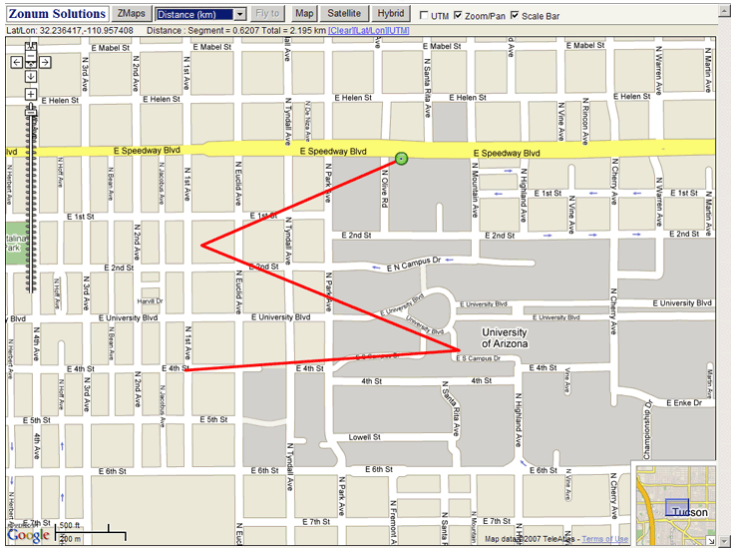
距离度量的种类

选哪个距离度量呢？能否学出来一个最好的距离度量？——距离度量学习

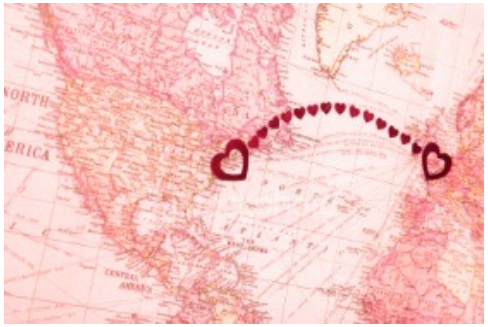
- 欧式距离

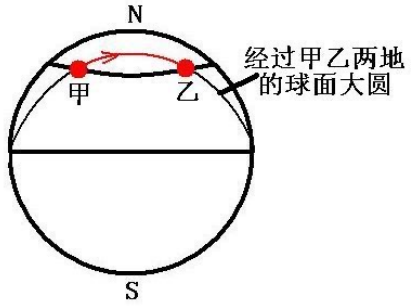
$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=i}^{n}(x_i - y_i)^2}$$

- 曼哈顿距离



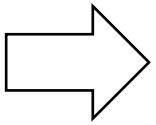
- 测地距离（ Geodesic Distance ）





距离度量学习 (distance metric learning)

降维希望找到一个“合适”低维空间下的距离度量



能否直接“学出”合适的距离？

首先，要能够通过 “参数化”来学习距离度量

马氏距离 (Mahalanobis distance) 是一个很好的选择：

$$\mathrm{dist}_{\mathrm{mah}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}\mathbf{M}(\boldsymbol{x}_i - \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2$$

其中 $\mathbf{M}$ 称为“度量矩阵”，数学上它是一个半正定对称矩阵

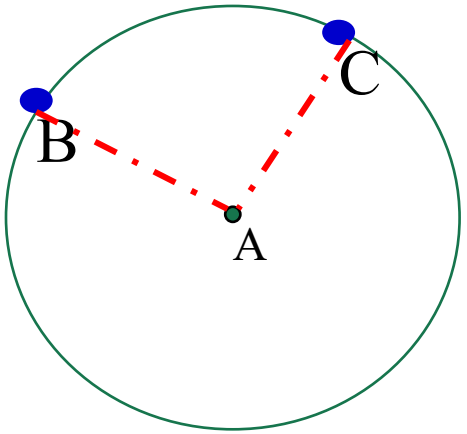
距离度量学习就是要对 $\mathbf{M}$ 进行学习

距离度量学习 (distance metric learning)

为什么是马氏距离？

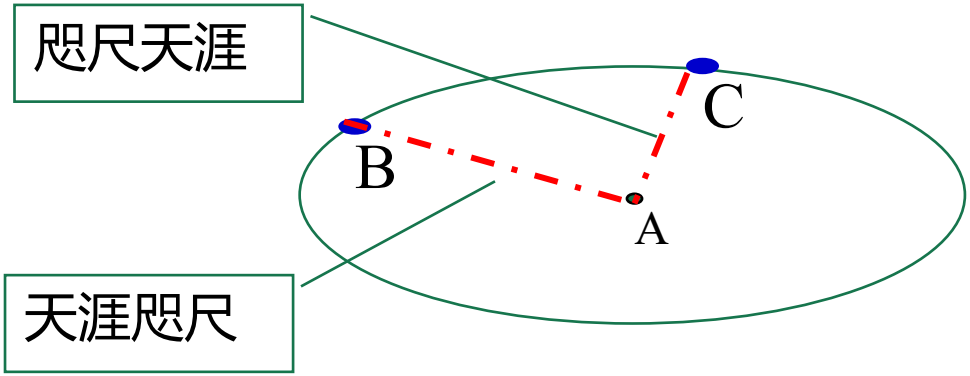
“什么是距离？

欧氏距离的缺陷——各个方向同等重要



但是：

- 有缘千里来相会
  (表面欧氏距离大但实际距离小)
- 无缘对面手难牵
  (表面欧式距离小但实际距离大)
- 马氏距离应运而生



距离度量学习 (distance metric learning)

M怎么学习？要有个合适的目标函数

- 目标1：结合具体分类器的性能

  例如，以近邻分类器的性能为目标，则得到经典的NCA算法

- 目标2：结合领域知识

  例如，若已知“必连”(must-link) 约束集合$\mathcal{M}$与“勿连”(cannot-link) 约束集合$\mathcal{C}$，则可通过求解下述凸优化问题得到 $\mathbf{M}$:

$$\min_{\mathbf{M}} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{M}} \|\boldsymbol{x}_i-\boldsymbol{x}_j\|_{\mathbf{M}}^2$$

$$\text{s.t.} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in\mathcal{C}} \|\boldsymbol{x}_i-\boldsymbol{x}_k\|_{\mathbf{M}}^2 \geqslant 1 ,$$

$$\mathbf{M} \succeq 0 ,$$

距离度量学习 – NCA：Neighborhood Component Analysis

近邻分类器在进行判别时通常使用多数投票法，替换为概率投票法. 对于任意样本 x_j，它对x_i分类结果影响的概率为

$$p_{ij} = \frac{\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2\right)}{\sum_l \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_{\mathbf{M}}^2\right)} ,$$

x_i 样本的LOO正确率：

$$p_i = \sum_{j \in \Omega_i} p_{ij} ,$$

训练集上的LOO正确率：

$$\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} \sum_{j \in \Omega_i} p_{ij} .$$

NCA的优化目标：

$$\mathbf{M} = \mathbf{P}\mathbf{P}^{\mathrm{T}} \qquad \min_{\mathbf{P}} \; 1 - \sum_{i=1}^{m} \sum_{j \in \Omega_i} \frac{\exp\left(-\left\|\mathbf{P}^{\mathrm{T}}\boldsymbol{x}_i - \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_j\right\|_2^2\right)}{\sum_l \exp\left(-\left\|\mathbf{P}^{\mathrm{T}}\boldsymbol{x}_i - \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_l\right\|_2^2\right)} .$$

小结

- 掌握支持向量机、决策树、贝叶斯分类器、KNN等学习算法
- 掌握聚类的基本思想和常见聚类算法
- 掌握降维的基本思想和常见降维算法
- 深入理解：
  - 间隔（margin）
  - 样本簇和原型的应用