



北京邮电大学

Beijing University of Posts and Telecommunications

无监督学习

戚 琦

网络与交换技术国家重点实验室 网络智能研究中心 科研楼511

qiqi8266@bupt.edu.cn

13466759972



表 12.1 10 种统计学习方法特点的概括总结

方法	适用问题	模型特点	模型类型	学习策略	学习的损失函数	学习算法
感知机	二类分类	分离超平面	判别模型	极小化误分点到超平面距离	误分点到超平面距离	随机梯度下降
k 近邻法	多类分类, 回归	特征空间, 样本点	判别模型			
朴素贝叶斯法	多类分类	特征与类别的联合概率分布, 条件独立假设	生成模型	极大似然估计, 极大后验概率估计	对数似然损失	概率计算公式, EM 算法
决策树	多类分类, 回归	分类树, 回归树	判别模型	正则化的极大似然估计	对数似然损失	特征选择, 生成, 剪枝
逻辑斯蒂回归与最大熵模型	多类分类	特征条件下类别的条件概率分布, 对数线形模型	判别模型	极大似然估计, 正则化的极大似然估计	逻辑斯蒂损失	改进的迭代尺度算法, 梯度下降, 拟牛顿法
支持向量机	二类分类	分离超平面, 核技巧	判别模型	极小化正则化合页损失, 软间隔最大化	合页损失	序列最小最优优化算法 (SMO)
提升方法	二类分类	弱分类器的线性组合	判别模型	极小化加法模型的指数损失	指数损失	前向分步加法算法
EM 算法 ^①	概率模型参数估计	含隐变量概率模型		极大似然估计, 极大后验概率估计	对数似然损失	迭代算法
隐马尔可夫模型	标注	观测序列与状态序列的联合概率分布模型	生成模型	极大似然估计, 极大后验概率估计	对数似然损失	概率计算公式, EM 算法
条件随机场	标注	状态序列条件下观测序列的条件概率分布, 对数线性模型	判别模型	极大似然估计, 正则化极大似然估计	对数似然损失	改进的迭代尺度算法, 梯度下降, 拟牛顿法

- 感知机
- K近邻法
- 朴素贝叶斯
- 决策树
- 逻辑斯蒂回归与最大熵模型
- 支持向量机
- 提升方法
- EM算法
- 隐马尔科夫模型
- 条件随机场

参考：李航，《统计机器学习方法》



■ 无监督学习概述

■ 聚类 划分聚类

密度聚类

■ 降维---PCA

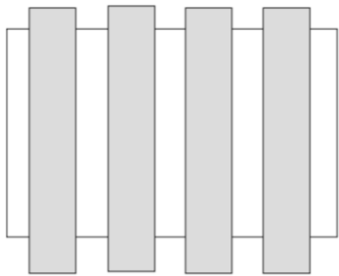
■ EM算法

■ 自动编码器

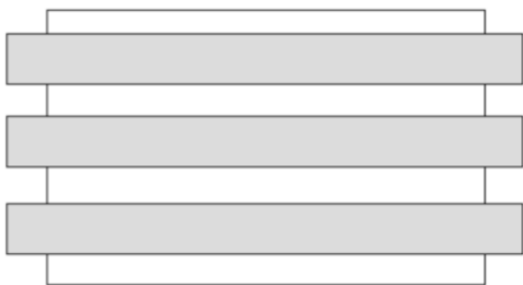


- 无监督学习的基本想法是对给定数据（矩阵数据）进行某种“**压缩**”，从而**找到数据的潜在结构**。假定损失最小的压缩得到的结果就是最本质的结构。

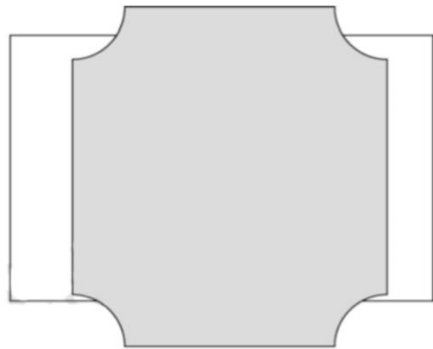
考虑发掘数据的纵向结构，把相似的样本聚到同类，即对数据进行**聚类**。



考虑发掘数据的横向结构，把高维空间的向量转换为低维空间的向量，即对数据进行**降维**。



同时考虑发掘数据的纵向与横向结构，假设数据由含有隐式结构的概率模型生成得到，从数据中学习该**概率模型**。





无监督学习基本问题---聚类

- 聚类 (clustering) 是将样本集合中相似的样本 (实例) 分配到相同的类, 不相似的样本分配到不同的类。
- 聚类时, 样本通常是欧氏空间中的向量, 类别不是事先给定, 而是从数据中自动发现, 但类别的个数通常是事先给定的。样本之间的相似度或距离由应用决定。
- 如果一个样本只能属于一个类, 则称为硬聚类 (hard clustering)

$$z_i = g_{\theta}(x_i), i = 1, 2, \dots, N$$

- 如果一个样本可以属于多个类, 则称为软聚类 (soft clustering), 每一个样本依概率属于每一个类

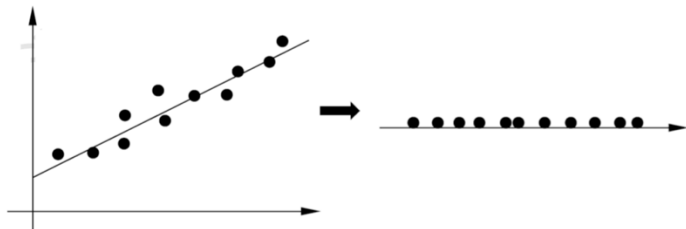
$$P_{\theta}(z_i|x_i), i = 1, 2, \dots, N,$$



无监督学习基本问题---降维

- 降维 (dimensionality reduction) 是将训练数据中的样本 (实例) 从高维空间转换到低维空间。
- 假设样本原本存在于低维空间, 或者近似地存在于低维空间, 通过降维则可以更好地表示样本数据的结构, 即更好地表示样本之间的关系。
- 高维空间通常是高维的欧氏空间, 而低维空间是低维的欧氏空间或者流形 (manifold)。
- 从高维到低维的降维中, 要保证样本中的信息损失最小。
- 降维有线性的降维和非线性的降维。

二维空间的样本存在于一条直线的附近, 可以将样本从二维空间转换到一维空间。通过降维可以更好地表示样本之间的关系。



无监督学习基本问题---概率模型估计

- 假设训练数据由一个概率模型生成，由训练数据学习概率模型的结构和参数。
- 概率模型的结构类型，或者说概率模型的集合事先给定，而模型的具体结构与参数从数据中自动学习。学习的目标是找到最有可能生成数据的结构和参数。
- 概率模型包括混合模型、概率图模型（EM算法）等。
- 概率图模型又包括有向图模型和无向图模型。

假设数据由高斯混合模型生成，学习的目标是估计这个模型的参数

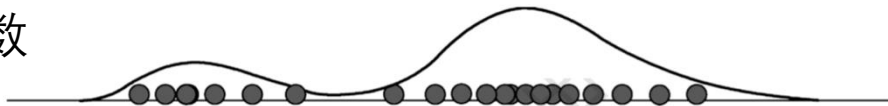


图 13.4 概率模型估计的例子



■ 无监督学习概述

■ 聚类 划分聚类

密度聚类

■ 降维---PCA

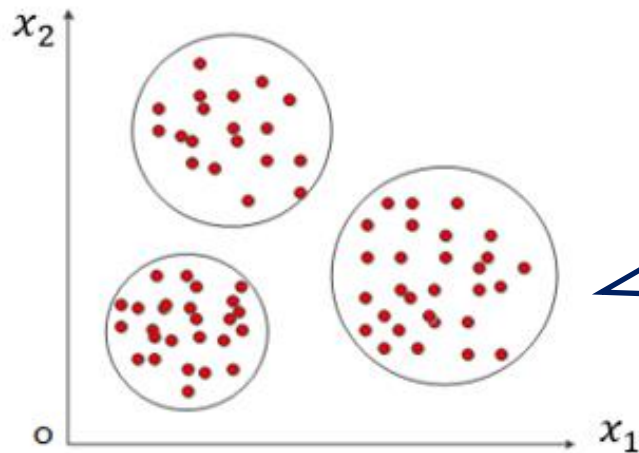
■ EM算法

■ 自动编码器



- 基本思想：对样本数据进行划分，实现对样本数据的聚类分析。
- 首先需要确定划分块的个数即**聚簇的个数**，然后通过适当方式将样本数据聚集成指定个数的聚簇。
- **k -均值聚类**和**模糊 c -均值聚类**是两种最典型、最常用的划分型聚类算法，这两种算法均**使用样本数据的均值确定各聚簇的聚类中心**，并通过计算各样本数据到各聚簇聚类中心的**某种距离**实现对样本数据之间的相似性度量。

基本思想：基于同类样本在特征空间中应该相距不远的基本思想，将集中在特征空间某一区域内的样本划分为同一个簇，其中区域位置的界定主要通过样本特征值的均值确定。



对具有两个属性特征 X_1, X_2 的某示例样本数据集进行聚类，簇数 $k = 3$ ，每个聚簇的聚类中心坐标值为该簇中所有示例样本特征的均值。

通常用欧式距离（2-范数）或曼哈顿距离（1-范数）等范数度量两个示例样本之间的距离。

对于给定的样本数据集 D : $D = \{X_1, X_2, \dots, X_n\}$
每个样本具有 m 个特征, 即 $X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$

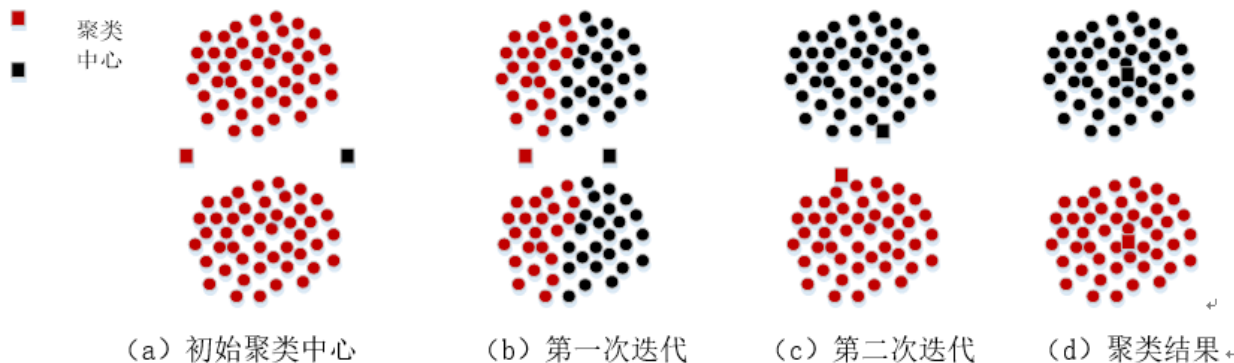
以欧式距离为例, 聚类过程: 假设按照某种方式将数据集 D 中所有样本划分为 k 个簇 C_1, C_2, \dots, C_k , 则与该划分相对应的类内距离 $d(C_1, C_2, \dots, C_k)$ 为:

$$d(C_1, C_2, \dots, C_k) = \sum_{j=1}^k \sum_{X_i \in C_j} \left(\sum_{t=1}^m (x_{it} - u_{jt})^2 \right)^{\frac{1}{2}}$$

其中 u_{jt} 表示为第 j 个簇 C_j 聚类中心 U_j 的第 t 个坐标分量。

k -Means 聚类算法过程

- (1) 令 $s = 0$ ，并从 D 中随机生成 k 个作为初始聚类中心的数据点 $u_1^0, u_2^0, \dots, u_k^0$;
- (2) 计算 D 中各样本与各簇中心之间的距离 w ，并根据 w 值将其分别划分到簇中心点与其最近的簇中;
- (3) 分别计算各簇中所有样本数据的均值，并分别将每个簇所得到的均值作为该簇新的聚类中心 $u_1^{s+1}, u_2^{s+1}, \dots, u_k^{s+1}$;
- (4) 若 $u_j^{s+1} = u_j^s$ ，则终止算法并输出最终簇，否则令 $s = s + 1$ ，并返回步骤 (2)。



某机构15支足球队在2017-2018年间的积分，各队在各赛事中的水平发挥有所不同。若将球队的**水平分为三个不同的层次水平**。聚类分析哪些队伍的整体水平比较相近。

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
赛事1	50	28	17	25	28	50	50	50
赛事2	50	9	15	40	40	50	40	40
赛事3	9	4	3	5	2	1	9	9
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
赛事1	40	50	50	50	40	40	50	
赛事2	40	50	50	50	40	32	50	
赛事3	5	9	5	9	9	17	9	

先对积分数据进行归一化处理：

$$a'_i = \frac{a_i - \min(a_i)}{\max(a_i) - \min(a_i)}$$

$\min(a_i)$ 和 $\max(a_i)$ 分别表示第 i 个属性值 a_i 在所有球队中的最小值和最大值。

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
赛事1	1	0.3	0	0.24	0.3	1	1	1
赛事2	1	0	0.15	0.76	0.76	1	0.76	0.76
赛事3		0.19						
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
赛事1	0.7	1	1	1	0.7	0.7	1	
赛事2	0.76	1	1	1	0.76	0.68	1	
赛事3	0.25	0.5	0.25	0.5	0.5	1	0.5	

将球队分为3个层次水平，聚类簇数 $k = 3$ 。通过随机采样选择编号为2、9、12的三支队伍所对应数据点作为初始聚类中心，即三个簇的聚类中心分别为： $\mu_1 = (0.3, 0, 0.19)$, $\mu_2 = (0.7, 0.76, 0.5)$, $\mu_3 = (1, 1, 0.5)$

计算每个数据点到聚类中心的欧氏距离

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
μ_1	1.2594	0	0.3407	0.7647	0.7710	1.2354	1.0787	1.0787
μ_2	0	0.9131	0.9995	0.5235	0.5946	0.6306	0.3000	0.3000
μ_3	0.3407	1.2594	1.3636	0.8353	0.8609	0.5000	0.2400	0.2400
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
μ_1	0.8609	1.2594	1.2221	1.2594	0.9131	1.1307	1.2594	
μ_2	0.2500	0.3842	0.4584	0.3842	0	0.5064	0.3842	
μ_3	0.4584	0	0.2500	0	0.3842	0.6651	0	

分别将每个数据点分配到聚类中心与其距离最近的簇中，得到第一次聚类结果为：

$$C_1 = \{X_2, X_3\}; C_2 = \{X_4, X_5, X_9, X_{13}, X_{14}\};$$

$$C_3 = \{X_1, X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$$

根据上述第一次聚类结果，对聚类中心做调整。对于 C_1 ，有：

$$\mu'_1 = \left(\frac{0.3 + 0}{2}, \frac{0.15 + 0}{2}, \frac{0.19 + 0.13}{2} \right) = (0.15, 0.075, 0.16)$$

同理可将第二个簇 C_2 和第三个簇 C_3 的聚类中心进行调整，分别得到

$$\mu'_2 = (0.528, 0.744, 0.412), \mu'_3 = (1.094, 0.40625)$$

计算各数据点与更新后的聚类中心的距离

队伍	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
μ'_1	1.3014	0.1704	0.1704	0.6967	0.7083	1.2664	1.1434	1.1434
μ'_2	0.5441	0.8092	0.8443	0.3308	0.4197	0.6768	0.4804	0.4804
μ'_3	0.1113	1.1918	1.3040	0.7965	0.8014	0.4107	0.2030	0.2030
队伍	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	
μ'_1	0.8831	1.3014	1.2595	1.3014	0.9420	1.1722	1.3014	
μ'_2	0.2368	0.5441	0.5609	0.5441	0.1939	0.6160	0.5441	
μ'_3	0.3832	0.1113	0.1674	0.1113	0.3622	0.7142	0.1113	

根据表可得到第二次聚类结果如下： $C_1 = \{X_2, X_3\}$ ； $C_2 = \{X_4, X_5, X_9, X_{13}, X_{14}\}$ ； $C_3 = \{X_1, X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$

聚类结果并未发生变化，故聚类中心收敛，停止迭代。

由上述聚类结果可知， X_2 、 X_3 两支球队的整体水平比较相近， $X_4, X_5, X_9, X_{13}, X_{14}$ 的整体水平比较相近，其余球队的整体水平比较相近。

与划分聚类的区别：**基于划分的聚类算法**主要通过样本数据之间的距离进行聚类操作，适合于对类圆形聚簇的聚类，如果将其用于对具有任意形状的聚簇进行聚类则有时不能获得满意的效果。

密度聚类算法：将聚簇看作是数据空间中被稀疏区域分开的稠密区域，由此得到以密度为度量标准的样本数据聚类方法。

三种具有代表性的密度聚类算法，即**DBSCAN算法**、**OPTICS算法**和**DENCLUE算法**。

密度聚类概念：两个参数 ϵ 和***MinPts***

ϵ ：领域半径

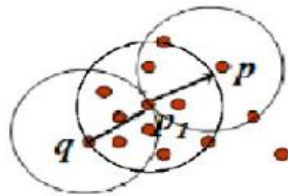
MinPts：在以 ϵ 为半径的领域内**最少包含点的个数**（密度阈值）

核心对象：一个对象的 ϵ -邻域至少包含***MinPts***个对象

边界对象：不是核心点，但落在某个核心点的 ϵ 邻域内的对象

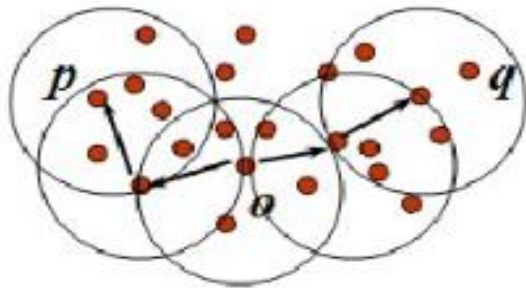
噪声对象：不属于任何簇的对象。

密度直达 (Directly density reachable, DDR) :如果 q 是一个核心对象, p_1 属于 q 的邻域 (就在邻域内), 那么称 q 密度直达 p_1 。



密度可达 (density reachable) :点 p 关于 ϵ 和 $MinPts$ 是从 q 密度可达的, 如果存在一个**节点链** $p_1, \dots, p_n, p_1 = q, p_n = p$, p_i 直接密度可达 p_{i+1} , 则称 p 密度可达 q 。

密度相连:点 p 关于 ϵ 和 $MinPts$ 与点 q 是密度相连的, 如果存在点 o 使得, p 和 q 都是关于 ϵ 和 $MinPts$ 是从 o 密度可达的 (如果存在 o , o 密度可达 q 和 p , 则称 p 和 q 是密度连通的)。

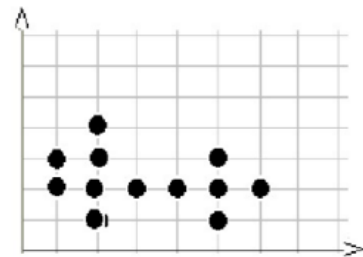


算法:

- (1) 任意选取一个点 p 。
- (2) 得到所有从 p 关于 ϵ 和 $MinPts$ 密度可达的点。
- (3) 如果 p 是一个核心点, 则找到一个聚类。
- (4) 如果 p 是一个边界点, 没有从 p 密度可达的点, DBSCAN 将访问数据中的下一个点。
- (5) 继续这一过程, 直到数据中的所有点都被处理。

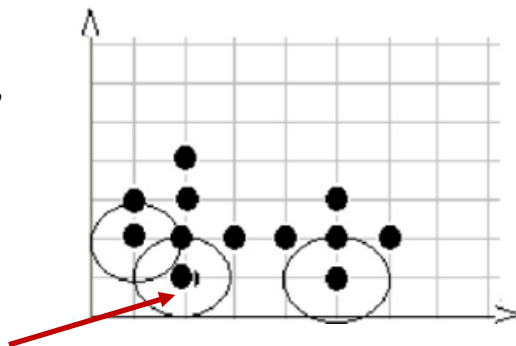
数据集 D ，试用DBSCAN算法对其进行密度聚类分析，取 $\varepsilon = 1$ 、 $MinPts = 4$ 、 $n = 12$ 。

序号	1	2	3	4	5	6	7	8	9	10	11	12
属性A	2	5	1	2	3	4	5	6	1	2	5	2
属性B	1	1	2	2	2	2	2	2	3	3	3	4



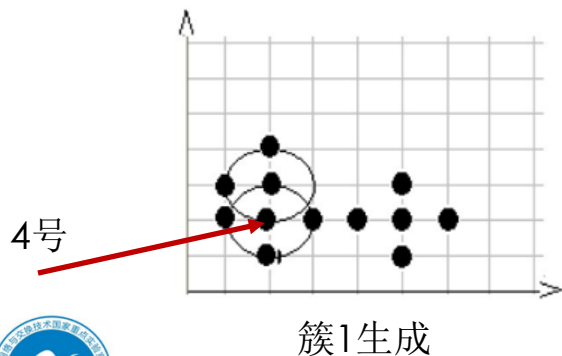
横纵的各自对应属性A和B

首先，在数据集 D 中任意选择一个样本点。选择1号样本点，由于以1号样本点为圆心且半径为1的圆中只包含2个样本点，小于4个，故1号样本点不是核心对象点。同理可得第2号、第3号样本点均不是核心样本点

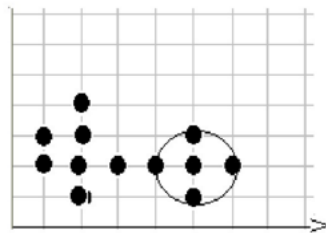
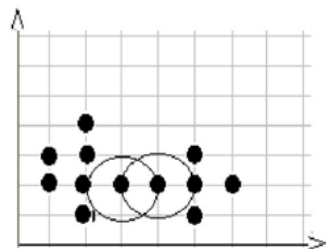


1号 判断样本点1、2、3

第二步，已知4号样本点是一个核心对象点。从4号样本点出发寻找所有与其具有可达关系的其余样本点，可以找到4个直接可达样本点、2个间接可达样本点，将这7个样本点组成一个样本子集合 $D_1 = \{1, 3, 4, 5, 9, 10, 12\}$ ，则 D_1 就是一个所求的聚簇

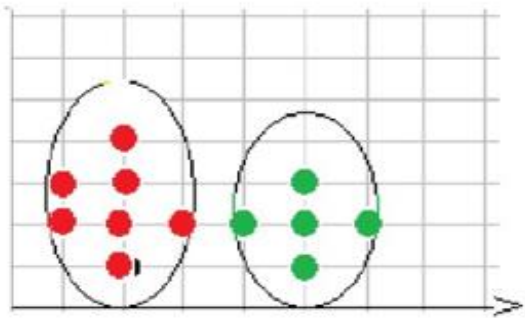


第三步，选择第5号样本点，因为第5号样本点已在簇 D_1 内，故选择下个样本点，选择第6号样本点，易知第6号样本点不是核心对象点



第四步，选择第7号样本点，第7号样本点是核心对象点。与第二步同理获得一个新的聚簇 $D_2 = \{2, 6, 7, 8, 11\}$

第五步，在数据集 D 选择第8号样本点，此样本点已经在簇2里面，故选择下一个样本点；同理发现样本点9、10和12已在聚簇 D_1 内，样本点11已在聚簇 D_2 内。此时已完成对数据集 D 中所有样本点的聚类分析，结束聚类过程并输出聚簇 D_1 和 D_2 。



最终聚类结果

DBSCAN算法的缺点：两个初始参数 ϵ （邻域半径）和 $MinPts$ （ ϵ 邻域最小点数）需要用户手动设置输入，聚类的类簇结果对这两个参数的取值非常敏感，在样本点密度分布不够均匀的场合，使用DBSCAN算法则难以获得满意的效果。

为避免DBSCAN算法在使用全局固定参数方面的局限，可以使用OPTICS密度聚类算法。



■ 无监督学习概述

■ 聚类 划分聚类

密度聚类

■ 降维---PCA

■ EM算法

■ 自动编码器



- 1901年Pearson提出针对非随机变量的主成分分析法，即PCA方法，1933年Hotelling将PCA推广到随机变量。
- PCA是常用的无监督学习方法，利用**正交变换**把由**线性相关**变量表示的观测数据转换为少数几个**由线性无关变量**表示的数据，线性无关的变量称为主成分。
- 主成分的个数通常小于原始变量的个数，**PCA属于降维方法**
- 主要用于发现数据中的基本结构，即数据变量之间的关系，是数据分析的有力工具，也可用于数据预处理
- 主要思想：**将高维空间数据投影到低维空间上，然后将数据包含信息量大的主成分保留下来，忽略掉对数据描述不重要的次要信息。**

假设 $D = \{X_1, X_2, \dots, X_n\}$ 为任意给定的示例样本数据集，其中 $X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$ ，即数据集中每个样本均为一个 m 维列向量，则 $X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$ 的基向量组通常取标准正交基：

$$(1, 0, \dots, 0)^T, (0, 1, \dots, 0)^T, \dots, (0, 0, \dots, 1)^T。$$

对于任意一组基向量 $\{w_1, w_2, \dots, w_m\}$ ，可将数据集 D 中任意给定的一个样本 X_i 表示为：

$$X_i = \sum_{j=1}^m \theta_{ij} w_j$$

其中 θ_{ij} 为样本 X_i 的第 j 个分量在基向量 $\{w_1, w_2, \dots, w_m\}$ 下的坐标。

样本数据在不同基向量下的坐标通常会有所不同

例如，对于某个样本数据向量 η ，通常以标准正交基 $(1,0)^T, (0,1)^T$ 为基向量将其表示为 $X = (3,2)^T$ ，即有： $(3,2)^T = 3(1,0)^T + 2(0,1)^T$ 表示。 $(3,2)^T$ 是以 $(1,0)^T$ 和 $(0,1)^T$ 为基向量组成的二维空间中从原点到坐标点 $(3,2)$ 的向量。也可以表示为

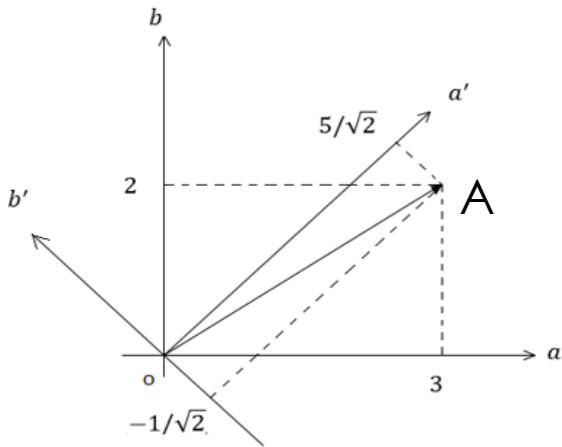
$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (3,2)^T$$

事实上，如果使用标准正交基 $(1/\sqrt{2}, 1/\sqrt{2})^T$ 和 $(-1/\sqrt{2}, 1/\sqrt{2})^T$ 作为二维空间的坐标系，则样本数据向量 η 在该坐标系中表示形式为：

$$X' = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} (3,2)^T = (5/\sqrt{2}, -1/\sqrt{2})^T$$

将样本数据向量 η 从某个坐标系映射到另一个坐标系，其实并未改变向量自身，而只改变了该向量的定量表示形式

向量 A 的坐标分量取值在这两个坐标系中的变换过程如图，其中原坐标系的坐标轴为 a 轴和 b 轴，变换后的坐标轴分别为 a' 轴和 b' 轴（正交的）。



若希望将数据集 D 中样本数据由 m 降至 k 维（ $k < m$ ），则可选择 k 个线性无关的 m 维向量 w_1, w_2, \dots, w_k 作为 k 维空间的一组基，将 D 中 m 维向量通过线性变换映射到以 w_1, w_2, \dots, w_k 为基向量的 k 维空间中，由此实现对数据集 D 中样本向量的降维。

令 $m \times n$ 阶矩阵 $\mathbf{X} = (X_1, X_2, \dots, X_n)$ 表示数据集 D 的初始数据，矩阵 $\mathbf{X}' = (X'_1, X'_2, \dots, X'_n)$ 表示数据集 D 在以 $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$ 为基向量的 k 维空间中样本数据，则有：

$$\mathbf{X}' = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k)^T \mathbf{X} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix}^T$$

m 维

n 个样本

记矩阵 $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k)^T$ ，并称之为变换矩阵，则 $\mathbf{X}' = \mathbf{W}\mathbf{X}$

任意选择的一组 k 个线性无关 m 维向量 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ 作为基向量构成变换矩阵 \mathbf{W} ，将数据集 D 中样本数据降至 k 维。

如何选择一组适当的基向量 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ 是实现了对样本数据进行有效降维的关键技术。

对数据集 D 中样本数据的降维应尽可能保留原数据有效信息。数据点分布的分散度可以用方差度量，方差越大的属性，其包含的信息量就越大（向基向量投影之后，尽可能比较分散），故要求所选基向量 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ 使得映射后数据方差尽可能地变大。

第一步:对所有数据特征进行中心化和归一化

数据集 D 中的样本数据进行标准化操作。假定数据集 D 中包含 n 个样本数据 X_1, X_2, \dots, X_n ，每个样本数据 X_i 均为具有 m 个属性的 m 维向量。 u_j 和 s_j 分别是样本数据集 D 中所有样本的第 j 维分量均值和标准差，即有：

$$u_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \quad s_j = \left(\frac{1}{n-1} \sum_{i=1}^n (x_{ij} - u_j)^2 \right)^{\frac{1}{2}}$$

令第 i 个样本数据 X_i 的第 j 维分量取值为 x_{ij} ，将 X_i 的各个分量值 x_{ij} 转化为标准值 z_{ij} ：

$$z_{ij} = \frac{x_{ij} - u_j}{s_j}, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m$$

第二步: 计算样本的协方差矩阵

将标准化后数据组成新的数据矩阵 $\mathbf{Z} = (Z_1, Z_2, \dots, Z_n)$ 并构造其协方差矩阵 \mathbf{C} :

$$\mathbf{C} = \frac{1}{n} \mathbf{Z}^T \mathbf{Z} = \begin{pmatrix} \frac{1}{n} \sum_{i=1}^n z_{i1}^2 & \frac{1}{n} \sum_{i=1}^n z_{i1} z_{i2} & \dots & \frac{1}{n} \sum_{i=1}^n z_{i1} z_{im} \\ \frac{1}{n} \sum_{i=1}^n z_{i2} z_{i1} & \frac{1}{n} \sum_{i=1}^n z_{i2}^2 & \dots & \frac{1}{n} \sum_{i=1}^n z_{i2} z_{im} \\ \dots & \dots & \dots & \dots \\ \frac{1}{n} \sum_{i=1}^n z_{im} z_{i1} & \frac{1}{n} \sum_{i=1}^n z_{im} z_{i2} & \dots & \frac{1}{n} \sum_{i=1}^n z_{im}^2 \end{pmatrix}$$

协方差矩阵 \mathbf{C} 中主对角线元素为样本属性数据的方差，非主对角线元素表示样本数据的两个属性之间的协方差。

协方差是对两个随机变量联合分布线性相关程度的一种度量

目标: 构造适当的标准正交基 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$, 使得标准化样本数据 $\mathbf{Z} = (Z_1, Z_2, \dots, Z_n)$ 变换到以 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ 为坐标系的 k 维线性空间中能够得到最大的数据方差。

也就是: 方差最大 (对角线), 协方差为0 (非对角线)。

第三步:求最大方差(对协方差矩阵求特征值和特征向量)

首先构造第一个基向量 \mathbf{w}_1 , 以标准化样本数据 (Z_1, Z_2, \dots, Z_n) 关于第一个属性的方差作为目标函数进行最大值优化求解, 目标函数:

$$J(\mathbf{w}_1) = \frac{1}{n} \sum_{i=1}^n (Z_i^T \mathbf{w}_1)^2$$

由于 $Z_i^T \mathbf{w}_1$ 是一个实数, 其转置还是其自身, 可将上述目标函数转化为:

$$J(\mathbf{w}_1) = \frac{1}{n} \sum_{i=1}^n (Z_i^T \mathbf{w}_1)^T (Z_i^T \mathbf{w}_1)$$

即有:

$$J(\mathbf{w}_1) = \frac{1}{n} \mathbf{w}_1^T \left(\sum_{i=1}^n Z_i Z_i^T \right) \mathbf{w}_1$$

得到:

$$J(\mathbf{w}_1) = \frac{1}{n} \mathbf{w}_1^T \mathbf{Z} \mathbf{Z}^T \mathbf{w}_1$$

$$\sum_{i=1}^n Z_i Z_i^T = \mathbf{Z} \mathbf{Z}^T$$

由于 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ 为标准正交基，故需满足约束条件 $\mathbf{w}_1^T \mathbf{w}_1 = 1$ 。将该约束条件与上述目标函数进行联立，可得到如下条件优化问题：

$$\max_{\mathbf{w}_1} \mathbf{w}_1^T \mathbf{Z} \mathbf{Z}^T \mathbf{w}_1;$$

$$\text{s. t. } \mathbf{w}_1^T \mathbf{w}_1 = 1$$

又因为协方差矩阵 $\mathbf{C} = \frac{1}{n} \mathbf{Z} \mathbf{Z}^T$ ，故可由此构造拉格朗日函数：
$$\max_{\mathbf{w}_1} \mathbf{w}_1^T \mathbf{C} \mathbf{w}_1 - \alpha (\mathbf{w}_1^T \mathbf{w}_1 - 1)$$

目标函数对 \mathbf{w}_1 的偏导数为0，则有 $2\mathbf{C}\mathbf{w}_1 - 2\alpha\mathbf{w}_1 = 0$ ，即有：

$$\mathbf{C}\mathbf{w}_1 = \alpha\mathbf{w}_1$$

因此， \mathbf{w}_1 是协方差矩阵 \mathbf{C} 的一个特征向量， α 是与该特性向量对应的特征根。又因：

$$\mathbf{w}_1^T \mathbf{C} \mathbf{w}_1 = \alpha \mathbf{w}_1^T \mathbf{w}_1 = \alpha$$

$\mathbf{w}_1^T \mathbf{w}_1 = 1$

故要使得 $\mathbf{w}_1^T \mathbf{Z} \mathbf{Z}^T \mathbf{w}_1$ 取得最大化，即使得 $\mathbf{w}_1^T \mathbf{C} \mathbf{w}_1$ 取得最大化， \mathbf{w}_1 即为协方差矩阵 \mathbf{C} 的最大特征根 λ_1 所对应的特征向量。根据 \mathbf{w}_1 可获得样本数据 \mathbf{X} 或 \mathbf{Z} 的第一个主成分 $\mathbf{z}'_1 = \mathbf{w}_1^T \mathbf{Z}$ 。

如果第一主成分 \mathbf{z}'_1 不足以代表 m 维数据 \mathbf{X} 或 \mathbf{Z} 的信息，可以考虑计算样本数据的第二个主成分 $\mathbf{z}'_2 = \mathbf{w}_2^T \mathbf{Z}$ 。对于第二个主成分 \mathbf{z}'_2 的构造，可通过求解如下条件优化问题获得：

$$\begin{aligned} & \max_{\mathbf{w}_2} \mathbf{w}_2^T \mathbf{C} \mathbf{w}_2 \\ & \text{s. t. } \mathbf{w}_2^T \mathbf{w}_2 = 1, \quad \boxed{\mathbf{w}_2^T \mathbf{w}_1 = 0} \quad \text{正交} \end{aligned}$$

构造拉格朗日函数： $\max_{\mathbf{w}_2} \mathbf{w}_2^T \mathbf{C} \mathbf{w}_2 - \alpha(\mathbf{w}_2^T \mathbf{w}_2 - 1) - \beta(\mathbf{w}_2^T \mathbf{w}_1 - 0)$


令上述优化问题的目标函数对 \mathbf{w}_2 的偏导数为0，则有：

$$2\mathbf{C}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0$$

用 \mathbf{w}_1^T 左乘上式，得： $2\mathbf{w}_1^T \mathbf{C} \mathbf{w}_2 - 2\alpha\mathbf{w}_1^T \mathbf{w}_2 - \beta\mathbf{w}_1^T \mathbf{w}_1 = 0$

由于 $\mathbf{w}_1^T \mathbf{w}_2 = 0$ 且 $\mathbf{w}_1^T \mathbf{C} \mathbf{w}_2$ 作为标量等于其转置 $\mathbf{w}_2^T \mathbf{C} \mathbf{w}_1$ ，注意到 \mathbf{w}_1 为协方差矩阵 \mathbf{C} 中以 λ_1 为特征根的特征向量，故有：

$$\mathbf{w}_1^T \mathbf{C} \mathbf{w}_2 = \mathbf{w}_2^T \mathbf{C} \mathbf{w}_1 = \lambda_1 \mathbf{w}_2^T \mathbf{w}_1 = 0$$



$$\mathbf{C}\mathbf{w}_1 = \lambda_1 \mathbf{w}_1$$

由此可得 $\beta \mathbf{w}_1^T \mathbf{w}_1 = 0$ ，即有 $\beta = 0$ ，从而可将

$2\mathbf{C}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0$ 简化为：

$$\mathbf{C}\mathbf{w}_2 = \alpha\mathbf{w}_2$$

因此，协方差矩阵 \mathbf{C} 中除 λ_1 之外的最大特征值 $\lambda_2 = \alpha$ 所对应的特征向量即为所求的第二个正交基向量 \mathbf{w}_2 ，由此可得第二主成分 $\mathbf{z}'_2 = \mathbf{w}_2^T \mathbf{Z}$ 。

可同理依此求出第三、第四、.....、第 k 个基向量和相应的主成分。

----- 以上为推导过程，下面是计算方法 -----

直接求 \mathbf{C} 的特征值和特征向量，对协方差矩阵 \mathbf{C} 做对角化处理：

$$\boldsymbol{\lambda} = \mathbf{P}\mathbf{C}\mathbf{P}^T$$

其中 $\boldsymbol{\lambda}$ 为 \mathbf{C} 的全部特征根组成的对角矩阵， \mathbf{P} 是 \mathbf{C} 的全部特征向量组成的正交矩阵。

选择矩阵 \mathbf{P} 的前 k 行组成主分量分析的变换矩阵 \mathbf{W}

基本PCA方法的基本步骤如下：

(1) 对数据集 D 中样本数据按如下公式进行标准化，并组成新的数据矩阵 \mathbf{Z} ；

$$z_{ij} = \frac{x_{ij} - u_j}{s_j}, i = 1, 2, \dots, n, j = 1, 2, \dots, m$$

(2) 根据数据矩阵 \mathbf{Z} 计算协方差矩阵 $\mathbf{C} = \frac{1}{m} \mathbf{Z}^T \mathbf{Z}$ ；

(3) 求出协方差矩阵 \mathbf{C} 全部特征根并将这些特征根按照从大到小次序排列，选择前 k 个特征值所对应特征向量按行排列构成变换矩阵 \mathbf{W} ；

(4) 使用变换矩阵 \mathbf{W} 对原数据进行降维 $\mathbf{X}' = \mathbf{W}\mathbf{X}$ ，或对标准化数据进行降维 $\mathbf{Z}' = \mathbf{W}\mathbf{Z}$ 。

若 $k = m$ ，则转换后数据保留了原数据的全部信息；若 $k = 0$ ，则相当于完全不展示原数据的信息。如何确定 k 值？

在确定 k 的具体取值时，通常会考虑不同 k 值可保留方差的百分比，并称这种分量方差占总方差的百分比为**该分量对总方差的贡献率**，简称为**方差贡献率**。

令 $\lambda_1, \lambda_2, \dots, \lambda_n$ 表示协方差矩阵 \mathbf{C} 的全部特征值且按由大到小顺序排列， \mathbf{w}_i 为特征值 λ_i 所对应的特征向量，若保留变换后样本数据前 k 个分量，则得到相应累计方差贡献率 Ω 为：

$$\Omega = \sum_{i=1}^k \lambda_i / \sum_{i=1}^n \lambda_i$$

通常选择 k 以保留99%或97%的累计方差贡献率，即选取满足 $\Omega \geq 0.99$ 或 $\Omega \geq 0.97$ 的最小 k 值。

现有我国大陆30个省、直辖市、自治区的经济发展状况数据集如表所示，包括8项经济指标：国民生产总值（ a_1 ）；居民消费水平（ a_2 ）；固定资产投资（ a_3 ）；职工平均工资（ a_4 ）；货物周转量（ a_5 ）；居民消费指数（ a_6 ）；商品零售价格指数（ a_7 ）；工业总产值（ a_8 ），试用基本PCA方法将这8项经济指标融合成3项综合指标。

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
北京	1394.89	2505	519.01	8144	373.9	117.3	112.6	843.43
天津	920.11	2720	345.46	6501	342.8	115.2	110.6	582.51
河北	2849.52	1258	704.87	4839	2033.3	115.2	115.8	1234.85
山西	1092.48	1250	290.9	4721	717.3	116.9	115.6	697.25
内蒙古	832.88	1387	250.23	4134	781.7	117.5	116.8	419.39
辽宁	2793.37	2397	387.99	4911	1371.1	116.1	114	1840.55
吉林	1129.2	1872	320.45	4430	497.4	115.2	114.2	762.47
黑龙江	2014.53	2334	435.73	4145	824.8	116.1	114.3	1240.37
上海	2462.57	5343	996.48	9279	207.4	118.7	113	1642.95
江苏	5155.25	1926	1434.95	5934	1025.5	115.8	114.3	2026.64
浙江	3524.79	2249	1006.39	6619	754.4	116.6	113.5	916.59
安徽	2003.58	1254	474	4609	908.3	114.8	112.7	824.14
福建	2160.52	2320	553.97	5857	609.3	115.2	114.4	433.67
江西	1205.1	1182	282.84	4211	411.7	116.9	115.9	571.84
山东	5002.34	1527	1229.55	5145	1196.6	117.6	114.2	2207.69

河南	3002.74	1034	670.35	4344	1574.4	116.5	114.9	1367.92
湖北	2391.42	1527	571.68	4685	849	120	116.6	1220.72
湖南	2195.7	1408	422.61	4797	1011.8	119	115.5	843.83
广东	5381.72	2699	1639.83	8250	656.5	114	111.6	1396.35
广西	1606.15	1314	382.59	5150	556	118.4	116.4	554.97
海南	364.17	1814	198.35	5340	232.1	113.5	111.3	64.33
四川	3534	1261	822.54	4645	902.3	118.5	117	1431.81
贵州	630.07	942	150.84	4475	301.1	121.4	117.2	324.72
云南	1206.68	1261	334	5149	310.4	121.3	118.1	716.65
西藏	55.98	1110	17.87	7382	4.2	117.3	114.9	5.57
陕西	1000.03	1208	300.27	4396	500.9	119	117	600.98
甘肃	553.35	1007	114.81	5493	507	119.8	116.5	468.79
青海	165.31	1445	47.76	5753	61.6	118	116.3	105.8
宁夏	169.75	1355	61.98	5079	121.8	117.1	115.3	114.4
新疆	834.57	1469	376.95	5348	339	119.7	116.7	428.76

将数据进行标准化处理，并依据标准化处理后数据建立协方差矩阵。

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	1.0000	0.2668	0.9506	0.1899	0.6172	-0.2726	-0.2636	0.8737
a_2	0.2668	1.0000	0.4261	0.7178	-0.1510	-0.2351	-0.5927	0.3631
a_3	0.9506	0.4261	1.0000	0.3989	0.4306	-0.2805	-0.3591	0.7919
a_4	0.1899	0.7178	0.3989	1.0000	-0.3562	-0.1342	-0.5384	0.1033
a_5	0.6172	-0.1510	0.4306	-0.3562	1.0000	-0.2532	0.0217	0.6586
a_6	-0.2726	-0.2351	-0.2805	-0.1342	-0.2532	1.0000	0.7628	0.1252
a_7	-0.2636	-0.5927	-0.3591	-0.5384	0.0217	0.7628	1.0000	-0.1921
a_8	0.8737	0.3631	0.7919	0.1033	0.6586	-0.1252	-0.1921	1.0000

编号	特征值	方差贡献率	累计方差贡献率
1	3.754	46.925%	46.925%
2	2.197	27.4625%	74.3875%
3	1.215	15.1875%	89.575%
4	0.403	5.0375%	94.6125%
5	0.213	2.6625%	97.275%
6	0.138	1.725%	99%
7	0.065	0.8125%	99.8125%
8	0.015	0.1875%	100%

对协方差矩阵进行对角化处理，为所求特征值并按从大到小的次序排列。最后两列分别是各特征值的方差贡献率及其累计值。

选择较大的三个特征值所对应特征向量作为基向量进行降维，所选三个特征值的方差百分比累计值为89.575%。根据特征值计算得到三个基向量的分量取值。

基向量元素取值表

	w_1	w_2	w_3
w_{i1}	0.45679	0.25851	0.1099
w_{i2}	0.31301	-0.40379	0.24587
w_{i3}	0.47056	0.10839	0.19243
w_{i4}	0.23996	-0.48777	0.33405
w_{i5}	0.2509	0.49801	-0.24933
w_{i6}	-0.26244	0.16988	0.7227
w_{i7}	-0.31966	0.40102	0.39716
w_{i8}	0.42468	0.28769	0.19147

可将8维的原始数据 $\mathbf{X} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8)^T$ 降维为3维的综合指标数据 $\mathbf{Z}' = (\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3)^T$ ，其中：

$$\mathbf{z}'_i = \mathbf{w}_i^T \mathbf{X}, \quad i = 1, 2, 3$$

即有：

$$\begin{aligned} \mathbf{z}'_1 &= 0.4568\mathbf{a}_1 + 0.3130\mathbf{a}_2 + 0.4706\mathbf{a}_3 + 0.2400\mathbf{a}_4 \\ &+ 0.2509\mathbf{a}_5 - 0.2624\mathbf{a}_6 - 0.3197\mathbf{a}_7 + 0.4247\mathbf{a}_8 \end{aligned}$$

$$\begin{aligned} \mathbf{z}'_2 &= 0.2585\mathbf{a}_1 - 0.4038\mathbf{a}_2 + 0.1084\mathbf{a}_3 - 0.4878\mathbf{a}_4 \\ &+ 0.4980\mathbf{a}_5 + 0.1699\mathbf{a}_6 + 0.4010\mathbf{a}_7 + 0.2877\mathbf{a}_8 \end{aligned}$$

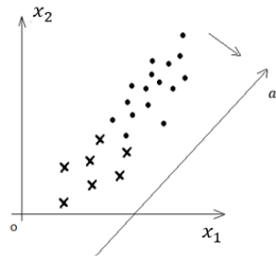
$$\begin{aligned} \mathbf{z}'_3 &= 0.1099\mathbf{a}_1 + 0.2459\mathbf{a}_2 + 0.1924\mathbf{a}_3 + 0.3340\mathbf{a}_4 \\ &- 0.2493\mathbf{a}_5 + 0.7227\mathbf{a}_6 + 0.3972\mathbf{a}_7 + 0.1915\mathbf{a}_8 \end{aligned}$$

带入各省市标准化后的数据，各省份经济数据的**主成分**如下表：

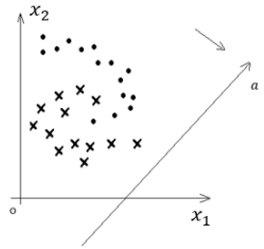
	北京	天津	河北	山西	内蒙古	辽宁	吉林
\mathbf{z}'_1	-0.8266	0.6564	1.3585	-0.9888	-1.6211	1.6632	-0.3868
\mathbf{z}'_2	2.2582	-2.6378	2.3513	0.3905	0.7253	0.9719	-0.4226
\mathbf{z}'_3	0.5399	-1.1725	-1.3128	-0.5717	-0.3819	-0.6231	-1.2106
	河南	湖北	湖南	广东	广西	海南	四川
\mathbf{z}'_1	1.023	-0.2825	-0.4101	4.6123	-1.1412	-0.5639	0.5699
\mathbf{z}'_2	2.1457	1.4488	1.063	-1.2982	0.3656	-2.2874	1.9764
\mathbf{z}'_3	-0.9401	1.1445	0.2546	0.0959	0.3815	-2.4087	0.852
	上海	江苏	浙江	安徽	福建	江西	山东
\mathbf{z}'_1	3.1951	3.5689	1.883	0.4451	0.4181	-1.3898	3.0006
\mathbf{z}'_2	-3.2802	1.2629	-0.4864	0.1197	-0.9188	0.3006	2.0659
\mathbf{z}'_3	2.8822	0.3835	0.2257	-1.862	-0.6569	-0.5293	0.5468
	云南	西藏	陕西	甘肃	青海	宁夏	新疆
\mathbf{z}'_1	-2.0197	-2.0175	-1.7772	-2.1163	-2.3478	-2.1619	-1.7232
\mathbf{z}'_2	0.7238	-2.0169	0.7078	0.1682	-1.074	-0.9936	-0.0422
\mathbf{z}'_3	1.8898	0.0156	0.459	0.6939	0.2626	-0.4881	1.019

经基本PCA方法降维后得到的三个数据向量 $\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3$ 均由原始数据的8个指标 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8$ 通过线性组合得到， \mathbf{z}'_1 的前3个指标 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_8$ 的组合系统均较大，这三个指标分量对 \mathbf{z}'_1 的构成起主要作用，故可将 \mathbf{z}'_1 看成是由国民生产总值、固定资产投资和工业总产值所刻画的反映经济发展状况的综合指标。同理，可将 \mathbf{z}'_2 看成是由货物周转量、商品零售价格指数、工业总产值所刻画的反映经济发展状况的综合指标，将 \mathbf{z}'_3 单独看成是居民消费指数指标。

基本PCA方法的缺点：使用线性映射方式将原始高维数据降至低维，有时很难得到良好的低维数据表示。



(a) 适合线性映射降维的数据分布



(b) 不适合线性映射降维的数据分布

使用核映射技术对基本主分量分析方法进行改进，提出一种名为核PCA分析的改进方法。

核PCA分析首先通过核映射技术对原始高维数据做进一步的升维变换，在更高维的空间中使用线性映射方式进行降维。

基本PCA分析的线性变换矩阵 \mathbf{W} 通过对协方差矩阵矩阵 \mathbf{C} 进行对角化而得。

核PCA分析使用某个核函数 $K(X_i, X_j) = \varphi^T(X_i)\varphi(X_j)$ ，所对应的核矩阵 \mathbf{K} 代替协方差矩阵 \mathbf{C} 。

给定数据集 $D = \{X_1, X_2, \dots, X_n\}$ 的情况下，核PCA分析的基本步骤如下：

- (1) 针对数据集 D 选择合适的核函数 $K(X_i, X_j)$;
- (2) 计算核函数所对应的核矩阵 K ;
- (3) 求核矩阵 K 所对应的特征值并将其按从大到小的次序排列，选择前 k 个特征值所对应特征向量按行排列构成主成分分析的变换矩阵 W ;
- (4) 使用变换矩阵 W 对高维数据进行降维： $X' = WX$ 或 $Z' = WZ$ 。



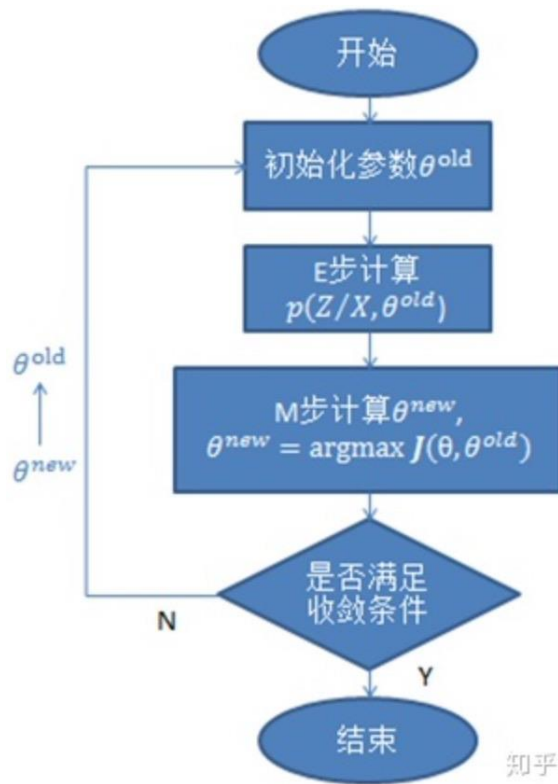
- 无监督学习概述
- 聚类 划分聚类
密度聚类
- 降维---PCA
- EM算法
- 自动编码器



例1: 假设我们需要调查学校的男生和女生的体重分布。(抽样) 假设你在校园里随便地邀请了100个男生和100个女生, 共200个人(也就是200个体重的样本数据)坐在教室里面。你开始喊:“男的左边, 女的右边!”。然后你就先统计抽样得到的100个男生的体重。假设他们的体重是服从高斯分布的。但是这个分布的均值 μ 和方差 σ^2 我们不知道, 这两个参数就是我们要估计的。记作 $\theta=[\mu, \sigma^2]^T$

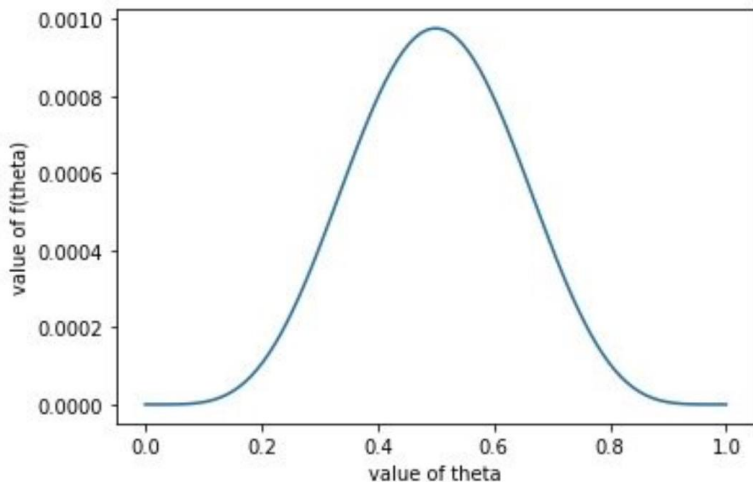
数学语言: 在学校那么多男生(体重)中, 我们独立地按照概率密度 $p(x|\theta)$ 抽取100个(体重), 组成样本集 X , 我们想通过样本集 X 来估计出未知参数 θ 。这里概率密度 $p(x|\theta)$ 我们知道了是高斯分布 $N(\mu, \sigma^2)$ 的形式, 其中的未知参数是 $\theta=[\mu, \sigma^2]^T$ 。抽到的样本集是 $X=\{x_1, x_2, \dots, x_N\}$, 其中 x_i 表示抽到的第 i 个人的体重, 这里 N 就是100, 表示抽到的样本个数。

- EM算法：期望最大化 (Expectation Maximization) 算法，是一种**迭代算法**，是在概率模型中寻找**参数极大似然估计**的算法，其中概率模型依赖于无法观测的隐含变量。
- 同时估计出**每个样本所属的簇类别**以及**每个簇概率分布的参数**。
- 主要用于从含有**隐含变量**的数据中计算**最大似然估计**。
- 1977年由美国数学家Arthur Dempster、Nan Laird和Donald Rubin提出



知乎 |

- 一枚硬币，想知道这枚硬币抛出去之后正面朝上的概率是多少
- 实验抛了10次硬币，其中正面朝上的次数是5次，反面朝上的次数也是5次。我们认为硬币每次正面朝上的概率是50%?
- 从表面上来看，这个结论理所应当，但问题在于实验结果和实验参数之间不是强耦合。如果硬币被人做过手脚，正面朝上的概率是60%，抛掷10次，也有可能得到5次正面5次反面的概率。同理，如果正面朝上的概率是70%，也有一定的概率可以得到5次正面5次反面的结果。
- 从概率角度入手，抛硬币是一个二项分布的事件，假设抛掷硬币正面朝上的概率是 p ，那么反面朝上的概率就是 $1-p$ 。带入二项分布的公式，计算10次抛掷之后，5次是正面结果在当前 p 参数下出现的概率是多少。



- 抛硬币事件:横坐标 p , 纵坐标为10次抛出**5**次为正面的概率
- 正面朝上的概率是**0.5**的时候, **10**次抛掷出现**5**次正面的概率最大。
- 将正面朝上的概率看成是实验当中的参数, 把似然看成是概率。

最大似然估计:使得当前实验结果出现概率最大的参数。

通过实验结果和概率, **找出最有可能导致这个结果的原因或者说参数**, 这个就叫做最大似然估计。



由于每个样本都是独立地从 $p(x|\theta)$ 中抽取的，抽到男生A（的身高）的概率是 $p(x_A|\theta)$ ，抽到男生B的概率是 $p(x_B|\theta)$ ，那因为他们是独立的，同时抽到男生A和男生B的概率是 $p(x_A|\theta) * p(x_B|\theta)$ ，同理，同时抽到这100个男生的概率就是他们各自概率的乘积。

数学语言：从分布是 $p(x|\theta)$ 的总体样本中抽取到这100个样本的概率，也就是样本集X中各个样本的联合概率，用下式表示：

$$L(\theta) = L(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n p(x_i; \theta), \theta \in \Theta$$



在概率密度函数的参数为 θ 时，得到X这组样本的概率

X已知（抽取到的这100个人的身高可以测出来）； θ 未知。上式只有 θ 是未知数，所以它是 θ 的函数。这个函数反映的是**在不同的参数 θ 取值下，取得当前这个样本集的可能性**，因此称为**参数 θ 相对于样本集X的似然函数**（likelihood function），记为 **$L(\theta)$** 。

【例1续】某些男生和女生情侣，无法让他们分开坐。那现在这200个人已经混到一起了，随便指出一个人（的身高），无法确定这个人（的身高）是男生（的身高）还是女生（的身高）。也就是说不知道抽取的那200个人里面的每一个人到底是从男生的那个身高分布里面抽取的，还是女生的那个身高分布抽取的。

用数学的语言就是，**抽取得到的每个样本都不知道是从哪个分布抽取的。**

这时有两个问题需要估计（两个问题互相依赖）：

一是这个人是男的还是女的？

二是男生和女生对应的身高的高斯分布的参数是多少？

想估计知道A和B两个互相依赖的参数的方法：先随便设一个A值出来，看另一方B如何变化，然后再根据B变化调整A变化，然后如此迭代着不断互相推导，最终就会收敛到一个解。

这就是**EM算法的基本思想**。

似然函数: $L(\theta) = L(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n p(x_i; \theta), \theta \in \Theta$

$$\ln L(\theta) = \sum_i \ln p(x_i; \theta), \theta \in \Theta$$

改为对数函数，相乘变相加

因不知样本性别，设为 z_i $p(x_i; \theta) = \sum_{z_i} p(x_i, z_i; \theta)$

引入隐变量 z （未知的），假设其概率分布为! (#C)

已知：样本集 $X=\{x(1), \dots, x(m)\}$ ，包含 m 个独立的样本；

未知：每个样本 i 对应的类别 $z(i)$ 是未知的（相当于聚类）；

输出：估计概率模型 $p(x, z)$ 的参数 θ ；

目标：找到适合的 θ 和 z ，让 $L(\theta)$ 最大。

$$L_{EM}(\theta) = \prod_{i=1}^n \prod_{z_i} p(x_i, z_i; \theta), \theta \in \Theta$$

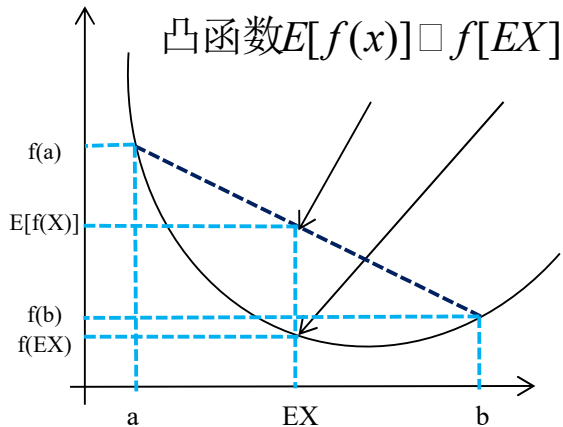
Jensen不等式

$\ln x$ 凹函数

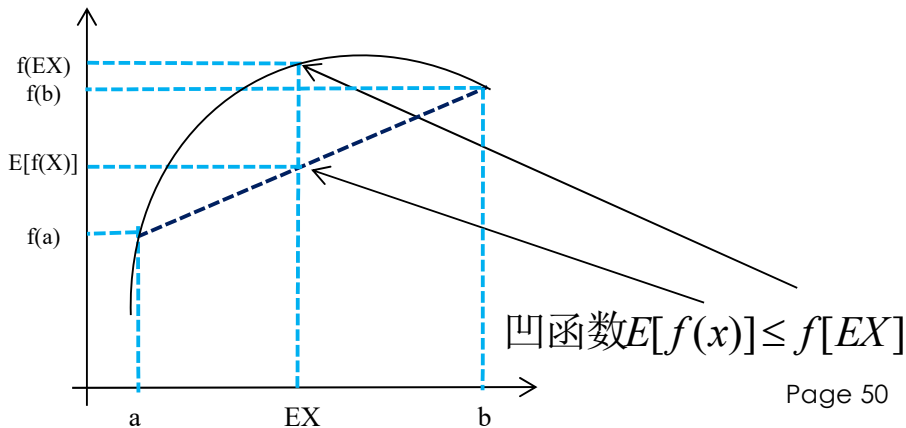
$$\prod E[\ln(x)] \leq \ln[EX]$$

$$\ln L_{EM}(\theta) = \sum_i \ln \prod_{z_i} p(x_i, z_i; \theta) = \sum_i \ln \prod_{z_i} Q(z_i) \frac{p(x_i, z_i; \theta)}{Q(z_i)} = \sum_i \sum_{z_i} Q(z_i) \ln \frac{p(x_i, z_i; \theta)}{Q(z_i)}$$

凸函数有**Jensen**不等式：函数的期望值大于等于期望值的函数值

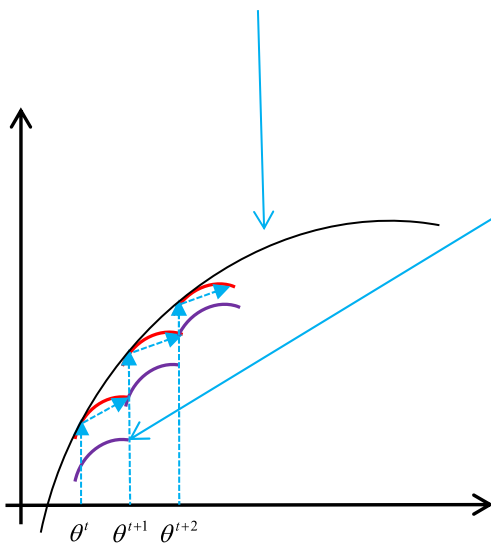


$$\frac{p(x_i, z_i; \theta)}{Q(z_i)} \text{ 的期望 } E[x] \quad \ln \frac{p(x_i, z_i; \theta)}{Q(z_i)} \text{ 的期望 } E[\ln(x)]$$



$$\ln L_{EM}(\theta) = \sum_i \ln \sum_{z_i} p(x_i, z_i; \theta) = \sum_i \ln \sum_{z_i} Q(z_i) \frac{p(x_i, z_i; \theta)}{Q(z_i)} \geq \sum_i \sum_{z_i} Q(z_i) \ln \frac{p(x_i, z_i; \theta)}{Q(z_i)}$$

在固定参数 θ 后，调整 $Q(z)$ 使下界拉升至与 $L(\theta)$ 在此点 θ 处相等，然后，固定 $Q(z)$ ，调整 θ 使下届达到最大值，此时为新的 θ ；再固定 θ ，调整 $Q(z)$ ，直到收敛到似然函数 $L(\theta)$ 的最大值。



解决了 $Q(z)$ 如何选择的问题，就是**E步**，建立 $L(\theta)$ 的下界。接下来的**M步**，就是在给定 $Q(z)$ 后，调整 θ ，去极大化 $L(\theta)$ 的下界。

- 初始化分布参数 θ ； 重复以下步骤直到收敛
- E步骤：根据参数初始值或上一次迭代的模型参数来计算出隐性变量的后验概率，其实就是隐性变量的期望。作为隐变量的现估计值：

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta).$$

计算方法

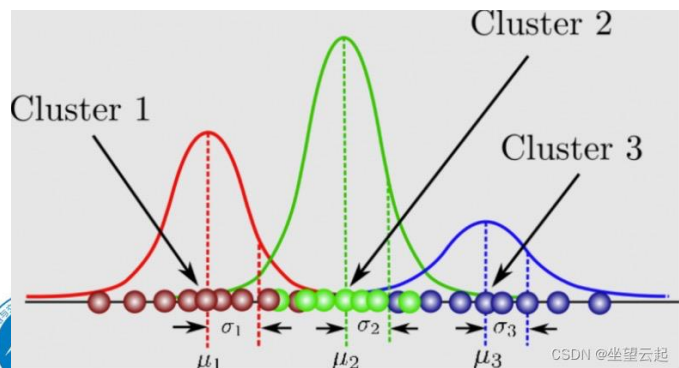
$$\begin{aligned} Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{\sum_z p(x^{(i)}, z; \theta)} \\ &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{p(x^{(i)}; \theta)} \\ &= p(z^{(i)}|x^{(i)}; \theta) \end{aligned}$$

- M步骤：将似然函数最大化以获得新的参数值：

$$\theta := \arg \max_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

总结：EM算法（期望最大算法）是一种从不完整数据或有数据丢失的数据集（存在隐含变量）中求解概率模型参数的最大似然估计方法。

- 高斯混合模型 (GMM) 是一个概率概念，用于对真实世界的数据集进行建模。
- GMM是高斯分布的泛化，可用于表示可聚类为多个高斯分布的任何数据集。
- 高斯混合模型 (GMM) 用于根据概率分布将数据分类为不同的类别
- 假设所有数据点都是从具有未知参数的高斯分布的混合中生成的
- GMM 的两部分组成：均值向量 (μ) 和协方差矩阵 (Σ)，多个高斯分布加权



单变量高斯分布 $N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right]$

多维数据可用高斯混合模型

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left[-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)^T\right]$$

■ 高斯混合模型参数估计的EM算法（以高维数据为例）：

第一步：参数赋初始值，开始迭代

第二步：E步，计算混合项系数 Z_{ij} 的期望 $E[Z_{ij}]$ ：

$$E[Z_{ij}] = \frac{|\Sigma_j|^{-1/2} e^{-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)}}{\sum_{k=1}^K |\Sigma_k|^{-1/2} e^{-\frac{1}{2}(x^{(i)} - \mu_k)^T \Sigma_k^{-1} (x^{(i)} - \mu_k)}}$$

第三步：M步，计算新一轮迭代的参数模型 θ ：

$$\begin{aligned} \text{均值 } \mu_j &= \frac{\sum_{i=1}^N E[Z_{ij}] x^{(i)}}{\sum_{i=1}^N E[Z_{ij}]} & \text{协方差 } \Sigma_j &= \frac{\sum_{i=1}^N E[Z_{ij}] \cdot \{(x^{(i)} - \mu_j)^T (x^{(i)} - \mu_j)\}}{\sum_{i=1}^N E[Z_{ij}]} & \text{系数 } w_j &= \frac{\sum_{i=1}^N E[Z_{ij}]}{N} \end{aligned}$$

重复第二、三步，直到收敛。

用程序随机从4个高斯模型中生成500个2维数据，

真实参数：混合项 $w=[0.1, 0.2, 0.3, 0.4]$ ，

均值 $\mu=\{[5, 35], [30, 40], [20, 20], [45, 15]\}$ ，

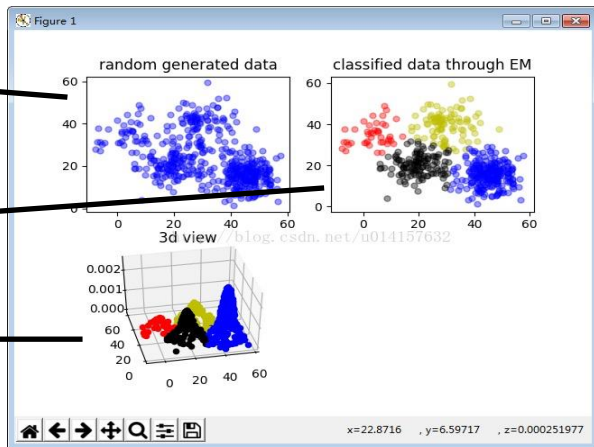
协方差矩阵 $\Sigma=\{[30, 0], [0, 30]\}$

以这500个数据作为观测数据，根据EM算法来估计以上参数（未估计协方差矩阵）

生成的观测数据

分类后的结果

高斯混合模型的
三维可视化图



用EM算法迭代计算结果：

- 混合项系数估计为 $[0.08790, 0.18614, 0.25716, 0.46878]$
- 均值估计为 $\{[3.7483 \ 34.9302]$
 $[29.9127 \ 39.8799] \ [19.9754$
 $20.2637] \ [45.2073 \ 15.4781]\}$



■ 无监督学习概述

■ 聚类 划分聚类

密度聚类

■ 降维---PCA

■ EM算法

■ 自动编码器

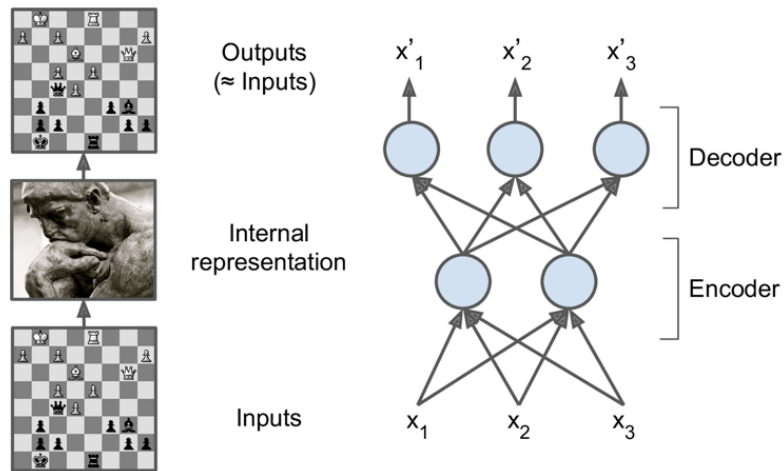


记下两组数字

40, 27, 25, 36, 81, 57, 10, 73, 19, 68

50, 25, 76, 38, 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20

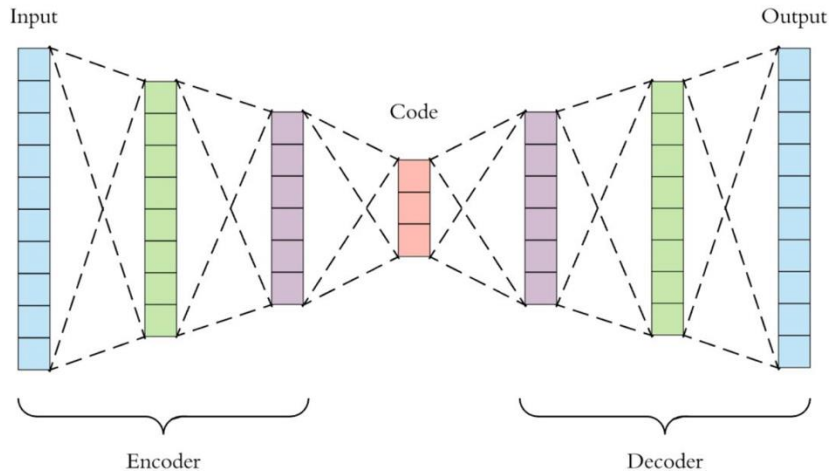
一个自编码器接收输入，将其转换成高效的内部表示，然后再输出输入数据的类似物体



- 自动编码器是一种能够通过无监督学习，学到**输入数据高效表示**的人工神经网络，自编码器能从数据样本中进行无监督学习，这意味着可将其应用到某个数据集中取得良好的性能，且不需要任何新的特征工程，只需要适当地训练数据。
- **输入数据的高效表示称为编码**，其维度一般远小于输入数据，使得自编码器可用于**降维**，设置合适的维度和稀疏约束，自编码器可以学习到比PCA等技术更有意思的数据投影。
- 自编码器还可以用于**数据去噪**。
- 自编码器可作为强大的特征检测器（feature detectors），应用于深度神经网络的**预训练**。
- 自编码器还可以随机生成与训练数据类似的数据，这被称作**生成模型**（generative model）

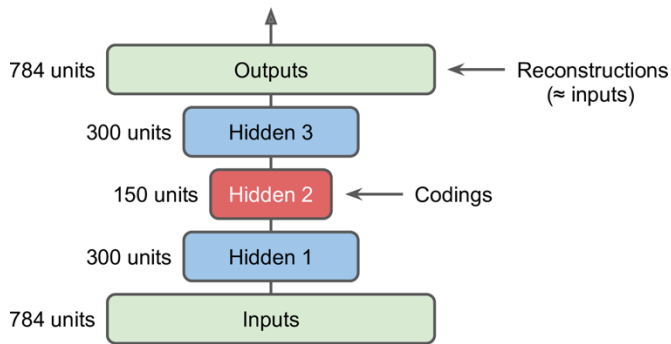
- 自编码器通常包括两部分：**encoder**（也称为识别网络）将输入转换成内部表示，**decoder**（也称为生成网络）将内部表示转换成输出。
- 编码器将输入压缩为潜在空间表征，可以用函数 $f(x)$ 来表示，解码器将潜在空间表征重构为输出，可以用函数 $g(x)$ 来表示，编码函数 $f(x)$ 和解码函数 $g(x)$ 都是神经网络模型。
- 与多层感知机类似，输入神经元和输出神经元的个数相等，输出是在设法重建输入，损失函数是重建损失

训练自编码器，可以使输入通过编码器和解码器后，保留尽可能多的信息，但也可以训练自编码器来使新表征具有多种不同的属性。不同类型的自编码器旨在实现不同类型的属性。



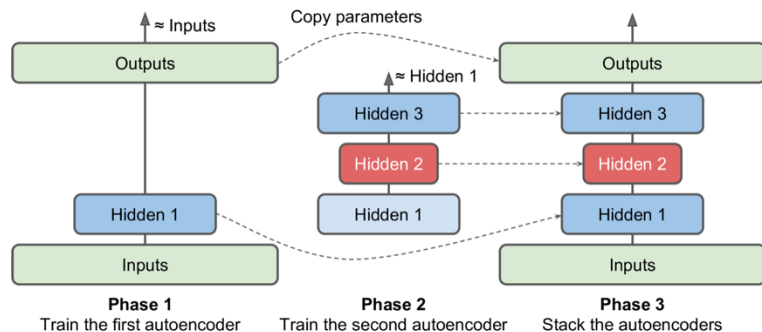


■ 自编码器可以有多个隐层，被称作**栈式自编码器（或者深度自编码器）**



增加隐层可以学到更复杂的编码，但不能使自编码器过于强大，encoder过于强大只能重建数据，不能学到有用的数据表示

与训练整个栈式自编码器不同，可以训练多个浅层的自编码器，然后再将它们合并为一体，提高训练速度

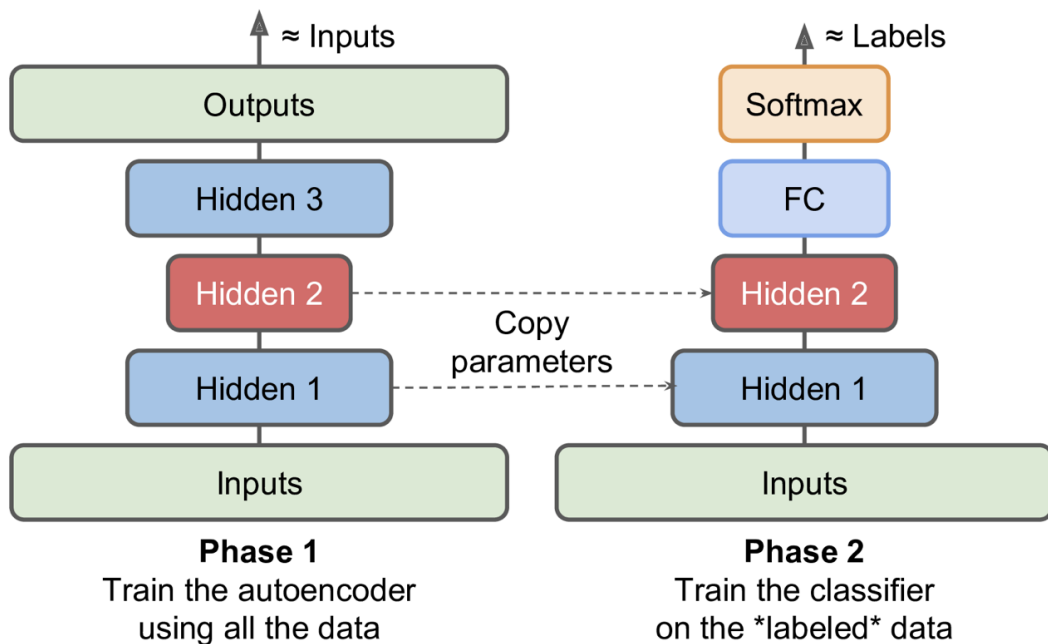


第一个自编码器学习去重建输入。然后，第二个自编码器学习去重建第一个自编码器隐层的输出。最后，这两个自编码器被整合到一起，可以使用这种方式，创建很深的栈式自编码器。





- 如果我们要处理一个复杂的有监督学习问题又没有足够的标注数据，一个解决方案是找到一个解决类似任务的训练好的模型，复用低层。类似的，如果有一个很大的数据集但绝大部分是未标注数据，可以使用所有的数据先训练一个栈式自编码器，然后复用低层来完成真正的任务。



- 使用自编码器去噪的思想在1980s提出（如在1987年Yann LeCun的硕士论文中有所提及）。在一篇2008年的论文中，Pascal Vincent等人表明自编码器可用于特征提取。在一篇2010年的论文中，Vincent等人提出栈式去噪自编码器（stacked denoising autoencoders）。
- 一种强制自编码器学习有用特征的方式是输入增加噪声，通过训练之后得到无噪声的输出。防止自编码器简单的将输入复制到输出，从而提取出数据中有用的模式，使学习到的低维表示具备更高的鲁棒性。
- 噪声可以是添加到输入的纯高斯噪声，也可以是随机丢弃输入层的某个特征，类似于dropout

