

# 无监督学习

主讲：郭春乐、刘夏雷  
南开大学计算机学院

致谢：本课件主要内容来自浙江大学吴飞教授、  
南开大学程明明教授

关于支持向量机，哪一种选择能正确填充下面的说法？支持向量机是一种\_\_模型，其目标是\_\_分类间隔。

- ☐ A 生成，最大化
- ☐ B 生成，最小化
- ☒ C 判别，最大化
- ☐ D 判别，最小化

提交

以下哪种方法不属于监督学习方法?

- ☒ A 聚类
- ☐ B 决策树
- ☐ C 线性回归
- ☐ D 朴素贝叶斯
- ☐ E 支持向量机
- ☐ F 以上都不属于监督学习方法

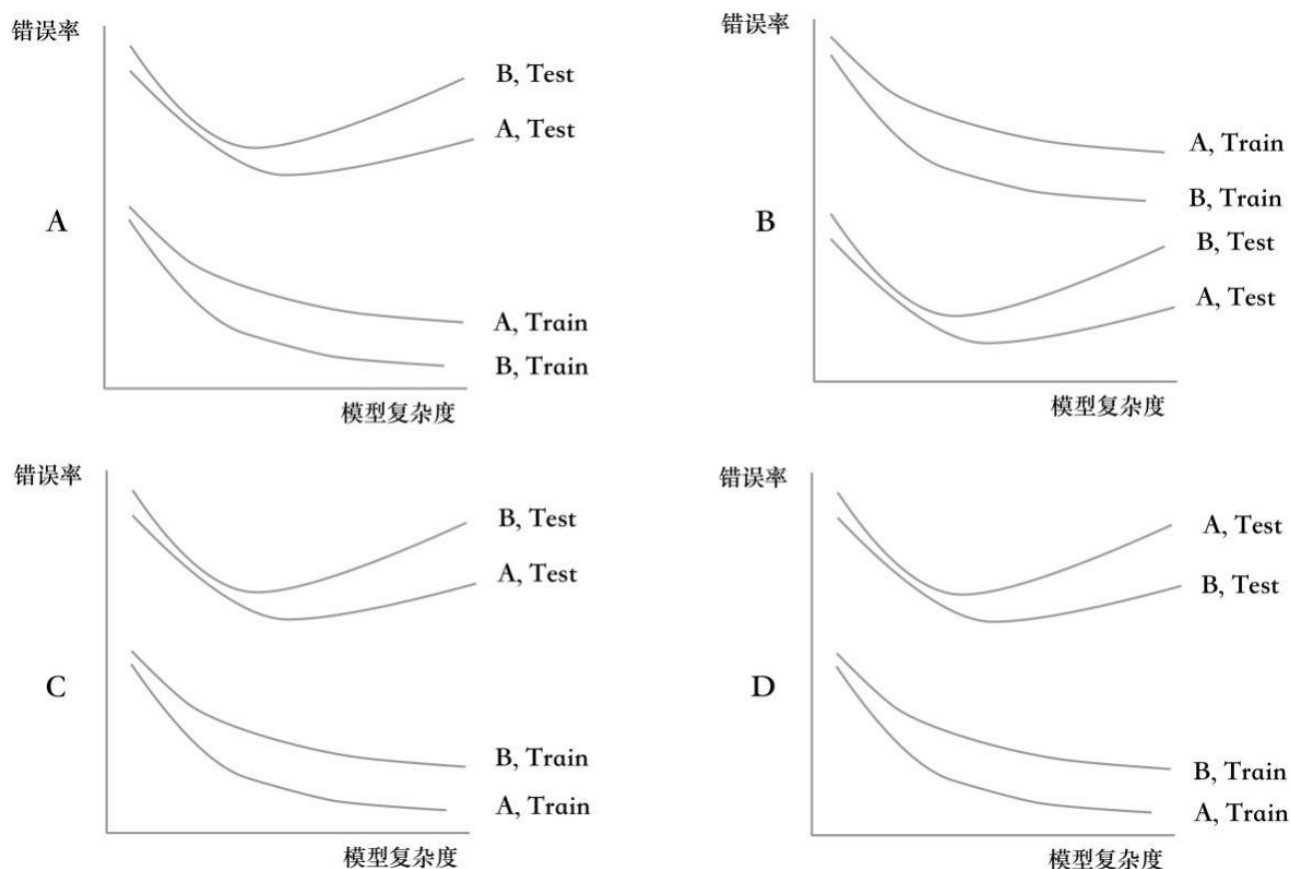
提交

在Adaboosting的迭代中，  
从第 $t$ 轮到第 $t+1$ 轮，某个被错误分类样本的惩罚增加了，可能因为该样本（）。

- ☒ A 被第 $t$ 轮训练的弱分类器错误分类
- ☐ B 被第 $t$ 轮后的集成分类器（强分类器）错误分类
- ☐ C 被到第 $t$ 轮为止训练的大多数弱分类器错误分类
- ☐ D B和C都正确
- ☐ E A，B和C都正确

提交

假设有两批从同样的真实数据分布中采样得到去完成同一任务的数据集A和B。A包含100K数据，B包含10K数据。按照9:1这一同样比例随机将A和B分别划分为训练集和测试集。图1给出了数据集A和数据集B随着模型复杂度增加所对应训练误差(A,Train以及B,Train)和测试误差(A,Test以及B,Test)的曲线图。请指出哪个图正确表示了随着模型复杂度增加所对应训练误差和测试误差的变化曲线图。



提交

对于如下数据，考虑使用Ada boosting方法来训练“是否出去玩”强分类器。每个弱分类器可考虑对单个属性的分类，比如对于“心情指数”这一属性，可考虑心情指数 $>2$ 和心情指数 $<4$ 两个方面。请问答下列问题：

- (1) Ada boosting在第一轮迭代中将会选择哪一个弱分类器？
- (2) 第一轮迭代前与迭代后每个样本的权重是多少？
- (3) 第二轮迭代选择的弱分类器是哪一个？分类器权重是多少？
- (4) 写出三轮迭代后的强分类器的表达式（每个弱分类器可用字母替代）

序号	出去玩	天气状况	有同伴	零花钱	特殊节日	心情指数 (1 差-5 好)
1	是	好	无	多	是	5
2	是	一般	有	多	是	5
3	是	一般	有	少	否	1
4	是	一般	有	少	否	3
5	是	一般	有	少	否	5
6	是	好	无	多	是	5
7	是	好	无	多	是	5
8	否	一般	无	多	是	1
9	否	一般	有	少	否	1
10	否	一般	无	少	否	5

作答

(1) 心情指数大于 1 出去玩，等于 1 不去玩。

(2)

迭代前：

1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10
------	------	------	------	------	------	------	------	------	------

迭代后：

1/16	1/16	4/16	1/16	1/16	1/16	1/16	1/16	1/16	4/16
------	------	------	------	------	------	------	------	------	------

(3)

有同伴就出去玩，没同伴不去玩

$$\frac{1}{2} \ln 3$$

(4)

三次迭代的分类器分别为：

$C_1$ : 心情指数大于 1 出去玩，等于 1 不去玩

$C_2$ : 有同伴出去玩，没同伴不去玩

$C_3$ : 天气好出去玩，天气不好不去玩

强分类器可表示为：

$$\text{sign}\left(\frac{1}{2} \ln(4) C_1 + \frac{1}{2} \ln(3) C_2 + \frac{1}{2} \ln\left(\frac{17}{7}\right) C_3\right)$$

作答

# 提纲

- K均值聚类
- 主成分分析
- 特征人脸方法
- 期望最大化算法
- 实训题目安排



# 机器学习: 从数据中学习映射函数



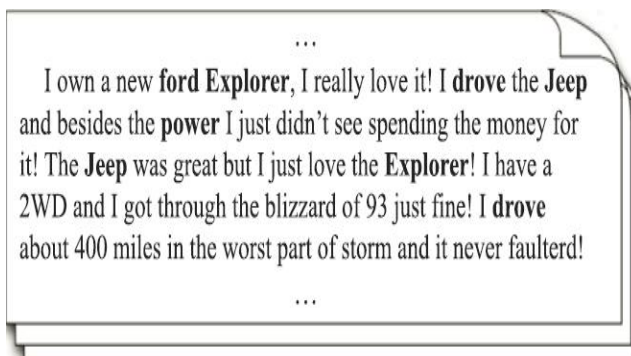
图像数据

$f\{$

81	116	...	133
104	130	...	159
...	...	...	...
155	189	...	218
197	221	...	216

- Person
- Dog
- ...

类别分类



文本数据

$f\{\text{car, money, drive, ...}\}$

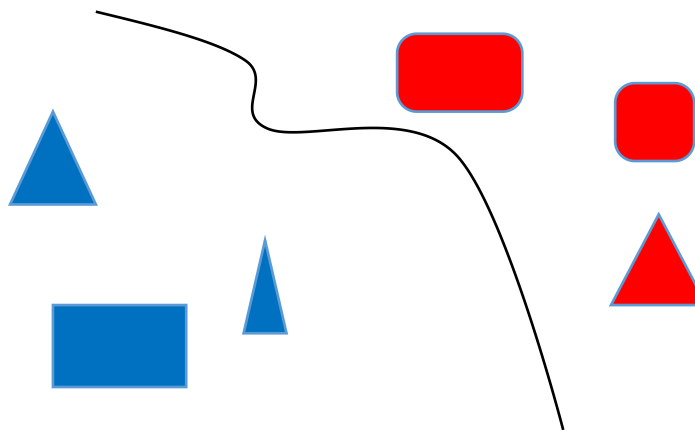
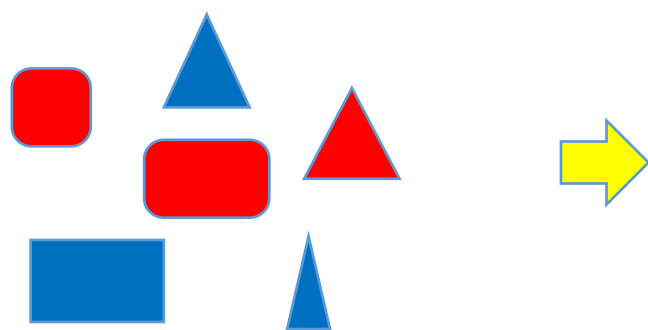
- 喜悦
- 愤怒
- ...

情感分类

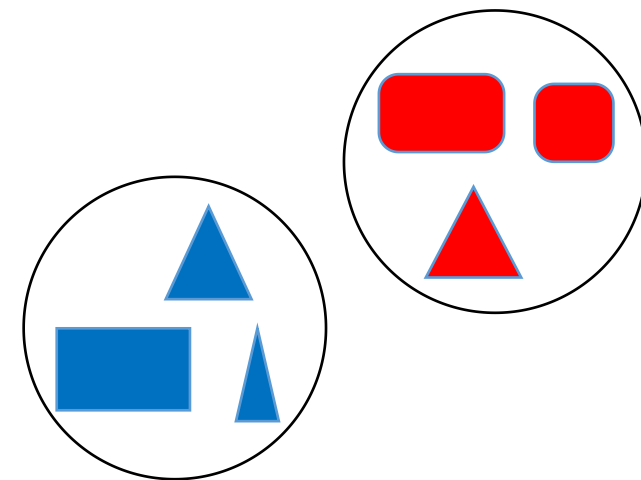
# 监督学习 versus 无监督学习

红色：汽车  
蓝色：飞机

它们是相似的  
数据的语义标签  
并不知道



左：监督学习



右：无监督学习

# 无监督学习的重要因素

数据特征	图像中颜色、纹理或形状等特征	听觉信息中旋律和音高等特征	文本中单词出现频率等特征
相似度函数	定义一个相似度计算函数，基于所提取的特征来计算数据之间的相似性		

Top suggestions for red



Red Bird



Red Fox



Red Panda



Red Dress



Red Hair



Red Shirt



Red Flowers



Red Sunflowers



Red Roses

Top suggestions for Round



Round China Cabinet



Round Sofa



Round Table



Round Eyes



Round Glasses



Round Sunglasses



Round Tablecloths



Round Loveseat



Round Beds



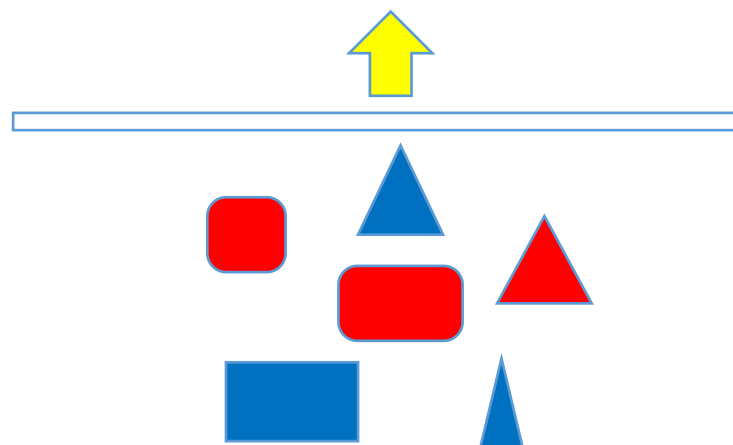
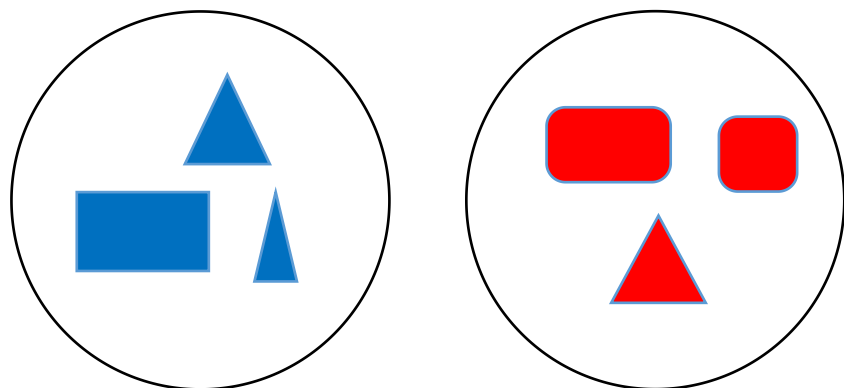
Round Pillow



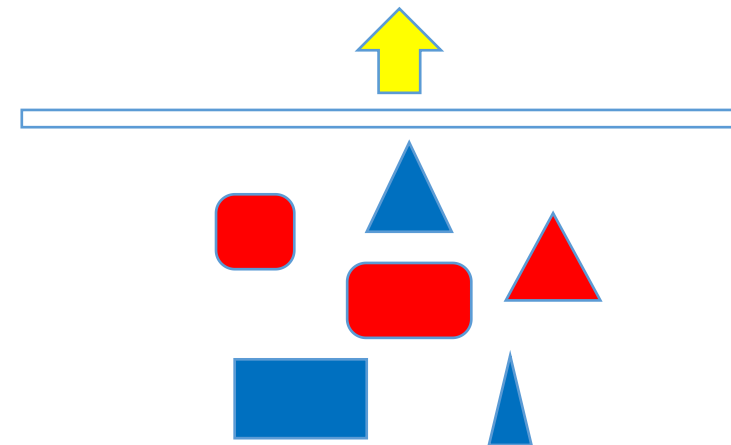
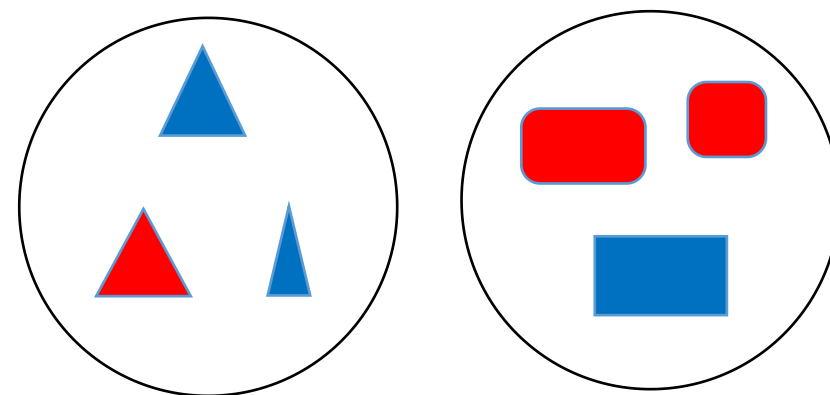
Round Rugs

# 无监督学习：数据特征和相似度函数都很重要

相似度函数：颜色相似



相似度函数：形状相似



# K均值聚类 (K-means 聚类)

- 物以类聚，人以群分（《战国策·齐策三》）
- 输入：  $n$  个数据（无任何标注信息）
- 输出：  $k$  个聚类结果
- 目的： 将  $n$  个数据聚类到  $k$  个集合（也称为类簇）

# K均值聚类算法描述

- 若干定义：

- $n$  个  $m$ -维数据  $\{x_1, x_2, \dots, x_n\}$ ,  $x_i \in R^m (1 \leq i \leq n)$

- 两个  $m$  维数据之间的欧氏距离为

- $$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{im} - x_{jm})^2}$$

- $d(x_i, x_j)$  值越小，表示  $x_i$  和  $x_j$  越相似；反之越不相似

- 聚类集合数目  $K$

- 问题：如何将  $n$  个数据依据其相似度大小将它们分别聚类到  $K$  个集合，使得每个数据仅属于一个聚类集合。

# K均值聚类算法：初始化

- 第一步：初始化聚类质心

- 初始化  $k$  个聚类质心  $c = \{c_1, c_2, \dots, c_k\}$  ,  $c_j \in R^m (1 \leq j \leq k)$

- 每个聚类质心  $c_j$  所在集合记为  $G_j$

# K均值聚类算法：对数据进行聚类

- 第二步：将每个待聚类数据放入唯一一个聚类集合中
  - 计算待聚类数据 $x_i$ 和质心 $c_j$ 之间的欧氏距离 $d(x_i, c_j)$  ( $1 \leq i \leq n, 1 \leq j \leq k$ )
  - 将每个 $x_i$ 放入与之距离最近聚类质心所在聚类集合中，即
$$\operatorname{argmin}_{c_j \in C} d(x_i, c_j)$$



# K均值聚类算法：更新聚类质心

- 第三步：根据聚类结果、更新聚类质心

- 根据每个聚类集合中所包含的数据，更新该聚类集合质心值，

即：

$$c_j = \frac{1}{|G_j|} \sum_{x_i \in G_j} x_i$$

# K均值聚类算法：更新聚类质心

- 第四步：算法循环迭代，直到满足条件
  - 在新聚类质心基础上，根据欧氏距离大小，将每个待聚类数据放入唯一一个聚类集合中
  - 根据新的聚类结果、更新聚类质心
  - 聚类迭代满足如下任意一个条件，则聚类停止：
    - 已经达到了迭代次数上限
    - 前后两次迭代中，聚类质心基本保持不变

# K均值聚类的另一个视角：最小化每个类簇的方差

- 方差：用来计算变量（观察值）与样本平均值之间的差异

$$\arg \min_G \sum_{i=1}^k \sum_{x \in G_i} ||x - G_i||^2 = \arg \min_G \sum_{i=1}^k |G_i| \text{var } G_i$$

第*i*个类簇的方差:  $\text{var}(G_i) = \frac{1}{|G_i|} \sum_{x \in G_i} ||x - G_i||^2$

- 欧氏距离与方差量纲相同
- 最小化每个类簇方差将使得最终聚类结果中每个聚类集合中所包含数据呈现出来差异性最小

# K均值聚类算法的不足

- **需要事先确定聚类数目**
  - 很多时候我们并不知道数据应被聚类的数目
- **需要初始化聚类质心**
  - 初始化聚类中心对聚类结果有较大的影响
- **算法是迭代执行，时间开销非常大**
- **欧氏距离假设数据每个维度之间的重要性是一样的**



# K均值聚类算法的应用

$K = 2$



$K = 3$



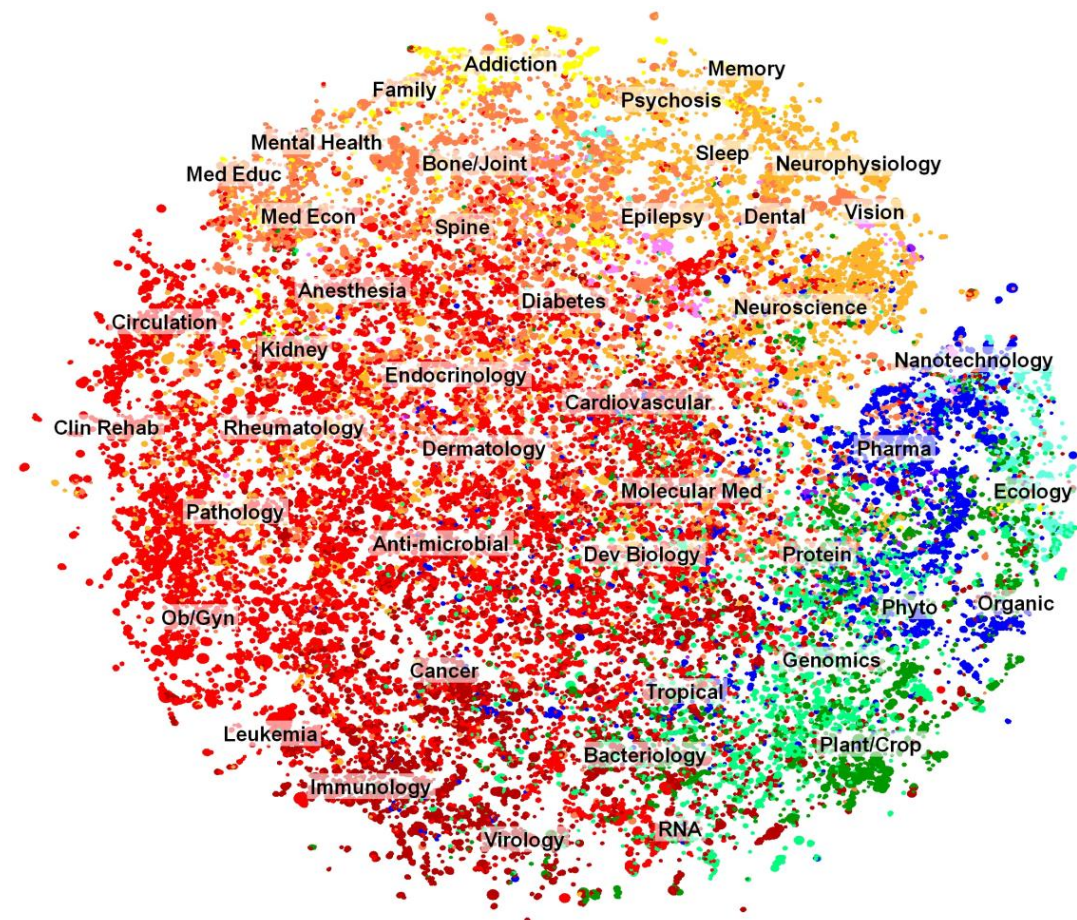
$K = 10$



Original image



图像压缩

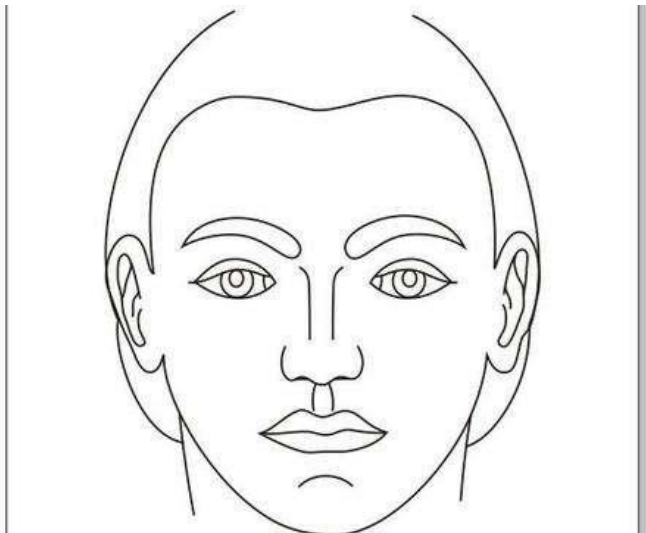


**文本分类：**将200多万篇论文聚类到29,000个类别，包括化学、工程、生物、传染疾病、生物信息、脑科学、社会科学、计算机科学等及给出了每个类别中的代表单词



# 主成分分析: Principle Component Analysis (PCA)

- 主成分分析是一种特征降维方法。人类在认知过程中会主动“化繁为简”
- 奥卡姆剃刀定律 (Occam's Razor) : “如无必要, 勿增实体”, 即“简单有效原理”



# 主成分分析: 降维后的结果要保持原始数据固有结构

- 原始数据中的结构

- 图像数据中结构: 视觉对象区域构成的空间分布
- 文本数据中结构: 单词之间的(共现)相似或不相似



200万像素点

约减



60个像素点

# 主成分分析: 若干概念-方差与协方差

- 方差等于各个数据与样本均值之差的平方和之平均数

- 假设有  $n$  个数据, 记为  $X = \{x_i\} (i = 1, \dots, n)$

$$\text{Var}(X) = \frac{1}{n-1} \sum_{i=1}^n (x_i - u)^2$$

- 其中  $u$  是样本均值,  $u = \frac{1}{n} \sum_{i=1}^n x_i$

- 方差描述了样本数据的波动程度



# 主成分分析: 若干概念-方差与协方差

- 数据样本的协方差: 衡量两个变量之间的相关度

- 假设有  $n$  个二维变量数据, 记为  $(X, Y) = \{(x_i, y_i)\} \ (i = 1, \dots, n)$

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - E(X))(y_i - E(Y))$$

- 其中  $\bar{x}$  ( $\bar{y}$ ) 和  $E(X)$  ( $E(Y)$ ) 分别是  $X$  和  $Y$  的样本均值, 分别定义如下

$$E(X) = \frac{1}{n} \sum_{i=1}^n x_i, \quad E(Y) = \frac{1}{n} \sum_{i=1}^n y_i$$

# 主成分分析: 协方差例子

编号	$x_i$	$y_i$	$x_i - E(X)$	$y_i - E(Y)$	$[x_i - E(X)][y_i - E(Y)]$
1	1	7	-8.33	-16.67	138.89
2	3	11	-6.33	-12.67	80.22
3	6	17	-3.33	-6.67	22.22
4	10	25	0.67	1.33	0.89
5	15	35	5.67	11.33	64.22
6	21	47	11.67	23.33	272.22
	$E(X) = 9.33$	$E(Y) = 23.67$	$Var(X) = 57.87$	$Var(Y) = 231.47$	$E([x_i - E(X)][y_i - E(Y)]) = 115.73$

$$X = \{x_i\}, Y = \{y_i\}$$

# 主成分分析: 协方差例子

- 对于一组两维变量（如广告投入-商品销售、天气状况-旅游出行等），可通过计算它们之间的协方差值来判断这组数据给出的两维变量是否存在关联关系：
- 当协方差 $\text{Cov}(X, Y) > 0$  时，称 $X$  与 $Y$  正相关
- 当协方差 $\text{Cov}(X, Y) < 0$  时，称 $X$  与 $Y$  负相关
- 当协方差 $\text{Cov}(X, Y) = 0$  时，称 $X$  与 $Y$  不相关（线性意义下）

# 主成分分析: 从协方差到相关系数

- 我们可通过皮尔逊相关系数（Pearson Correlation coefficient）将两组变量之间的关联度规整到一定的取值范围内。皮尔逊相关系数定义如下:

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}} = \frac{\text{Cov}(X, Y)}{\sigma_x \sigma_y}$$

# 主成分分析: 从协方差到相关系数

编号	$x_i$	$y_i$	$x_i - E(X)$	$y_i - E(Y)$	$[x_i - E(X)][y_i - E(Y)]$	$\text{corr}(X, Y)$
1	1	7	-8.33	-16.67	138.89	1.0 $y_i = 2x_i + 5$
2	3	11	-6.33	-12.67	80.22	
3	6	17	-3.33	-6.67	22.22	
4	10	25	0.67	1.33	0.89	
5	15	35	5.67	11.33	64.22	
6	21	47	11.67	23.33	272.22	
	$E(X) = 9.33$	$E(Y) = 23.67$	$Var(X) = 57.87$	$Var(Y) = 231.47$	$E([x_i - E(X)][y_i - E(Y)]) = 115.73$	

# 主成分分析: 从协方差到相关系数

## • 皮尔逊相关系数所具有的性质如下:

- 数刻画了变量 $X$ 和 $Y$ 之间线性相关程度,  $|\rho(X, Y)| \leq 1$
- $|\rho(X, Y)|$ 越大, 两者在线性相关的意义下相关度越大
- $\rho(X, Y) = 1$ 的充要条件是存在常数 $a$ 和 $b$ , 使得 $X = aY + b$
- $|\rho(X, Y)| = 0$ 表示两者不存在线性相关关系 (可能存在其他非线性相关的关系)。
- 皮尔逊相关系数是对称的, 即 $\rho(X, Y) = \rho(Y, X)$
- 正线性相关意味着变量 $X$ 增加的情况下, 变量 $Y$ 也随之增加; 负线性相关意味着变量 $X$ 减少的情况下, 变量 $Y$ 随之增加。

# 主成分分析: 从协方差到相关系数

- 相关性(correlation)与独立性(independence)

- 如果 $X$  和 $Y$  的线性不相关, 则 $|\text{Cov}(X, Y)|=0$
- 如果 $X$  和 $Y$  的彼此独立, 则一定 $|\text{Cov}(X, Y)|=0$ , 且 $X$  和 $Y$  不存在任何线性或非线性关系
- “不相关”是一个比“独立”要弱的概念, 即独立一定不相关, 但是不相关不一定相互独立 (可能存在其他复杂的关联关系)。  
独立指两个变量彼此之间不相互影响.

# 主成分分析: 算法动机

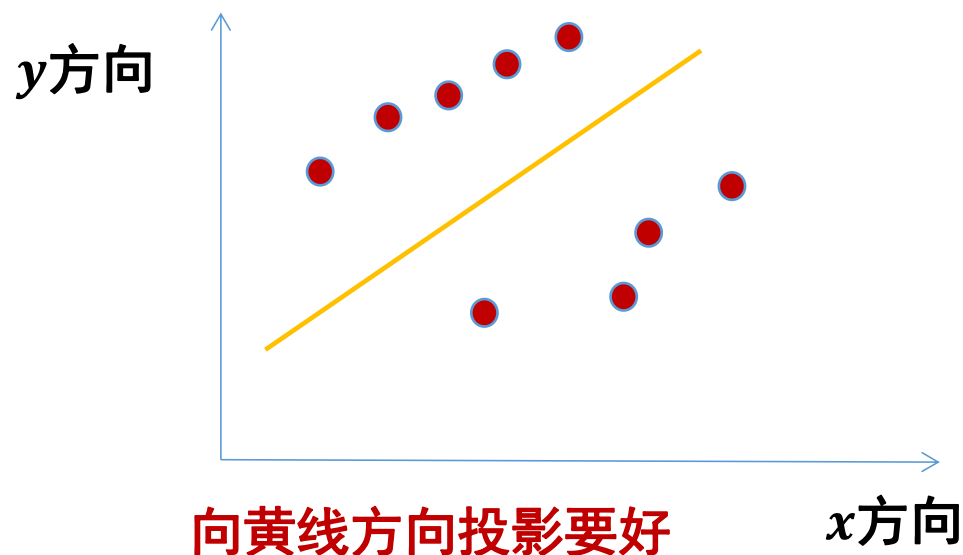
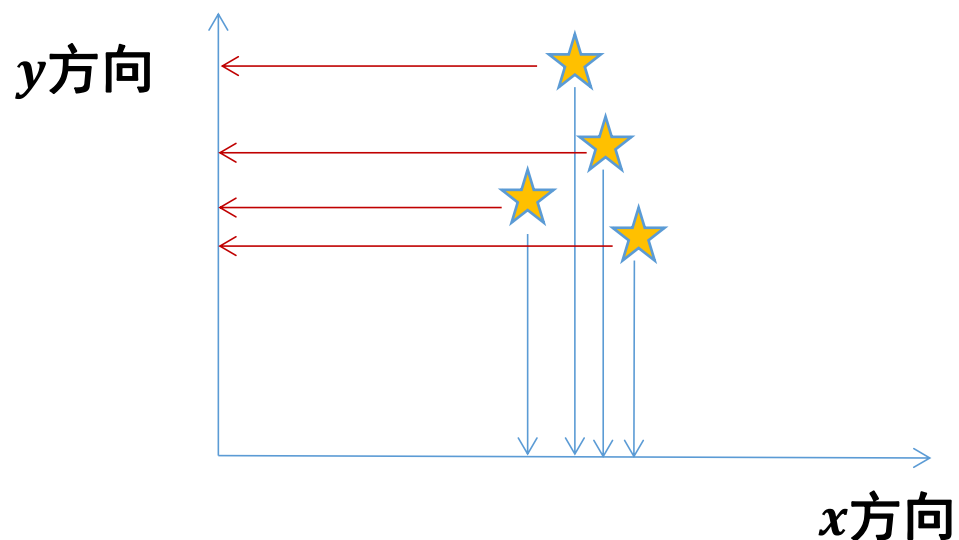
- 在数理统计中，方差被经常用来度量数据和其数学期望（即均值）之间偏离程度，这个偏离程度反映了数据分布结构。
- 在许多实际问题中，研究数据和其均值之间的偏离程度有着很重要的意义。
- 在降维之中，需要尽可能将数据向方差最大方向进行投影，使得数据所蕴含信息没有丢失，彰显个性。



# 主成分分析: 算法动机

- 保证样本投影后方差最大:

- 左下图中, 向❶方向投影 (使得二维数据映射为一维) 就比向❷方向投影结果在降维这个意义上而言要好;
- 右下图则是黄线方向投影要好。



# 主成分分析: 算法动机

- 主成分分析是将 $n$ 维特征数据映射到 $l$ 维空间( $n \gg l$ ), 去除原始数据之间的冗余性 (通过去除相关性手段达到这一目的)。
- 将原始数据向这些数据方差最大的方向进行投影。一旦发现了方差最大的投影方向, 则继续寻找保持方差第二的方向且进行投影。
- 将每个数据从 $n$ 维高维空间映射到 $l$ 维低维空间, 每个数据所得到最好的 $l$ 维特征就是使得每一维上样本方差都尽可能大。

# 主成分分析: 算法描述

- 假设有  $n$  个  $d$  维样本数据所构成的集合  $D = \{x_1, x_2, \dots, x_n\}$ 
  - 其中  $x_i (1 \leq i \leq n) \in \mathbb{R}^d$ 。
- 集合  $D$  可以表示成一个  $n \times d$  的矩阵  $X$ 。
- 假定每一维度的特征均值均为零（已经标准化）。
- 主成分分析的目的在于求取且使用一个映射矩阵  $W \in \mathbb{R}^{d \times l}$ 。
- 给定样本  $x_i \in \mathbb{R}^{1 \times d}$ ，可将  $x_i$  从  $d$  维空间映射到  $l$  维空间:  $x_i W_{d \times l}$
- 将所有降维后数据用  $z$  表示，有  $z = XW$

如何求取映射矩阵  $W$  ?

# 主成分分析: 算法描述

降维后 $n$ 个 $l$ 维样本数据 $Y$ 的方差为:

$$\text{var}(Y) = \frac{1}{n-1} \text{trace}(Y^T Y)$$

$$= \frac{1}{n-1} \text{trace}(W^T X^T X W)$$

$$= \text{trace}(W^T \frac{1}{n-1} X^T X W)$$

# 主成分分析: 算法描述

降维前 $n$ 个 $d$ 维样本数据 $X$ 的协方差矩阵记为:  $\Sigma = \frac{1}{n-1} X^T X$

主成份分析的求解目标函数为:

$$\max_{\mathbf{W}} \text{trace}(\mathbf{W}^T \Sigma \mathbf{W})$$

满足约束条件:

$$\mathbf{w}_i^T \mathbf{w}_i = 1 \quad i \in \{1, 2, \dots, l\}$$

# 主成分分析: 算法描述

- 所有带约束的最优化问题, 可通过拉格朗日乘子法将其转化为无约束最优化问题

主成份分析求解目标函数为

$$\max_{\mathbf{W}} \text{trace}(\mathbf{W}^T \mathbf{\Sigma} \mathbf{W})$$

满足约束条件

$$\mathbf{w}_i^T \mathbf{w}_i = 1 \quad i \in \{1, 2, \dots, l\}$$

拉格朗日  
函数



# 主成分分析: 算法描述

$$L(\mathbf{W}, \boldsymbol{\lambda}) = \text{trace}(\mathbf{W}^T \boldsymbol{\Sigma} \mathbf{W}) - \sum_{i=1}^l \lambda_i (\mathbf{w}_i^T \mathbf{w}_i - 1)$$

其中 $\lambda_i (1 \leq i \leq l)$ 为拉格朗日乘子,  $\mathbf{w}_i$ 为矩阵 $\mathbf{W}$ 第 $i$ 列。

对上述拉格朗日函数中变量 $\mathbf{w}_i$ 求偏导并令导数为零, 得到

$$\boldsymbol{\Sigma} \mathbf{w}_i = \lambda_i \mathbf{w}_i$$

上式表明: 每一个 $\mathbf{w}_i$ 都是 $n$ 个 $d$ 维样本数据 $\mathbf{X}$ 的协方差矩阵 $\boldsymbol{\Sigma}$ 的一个特征向量,  $\lambda_i$ 是这个特征向量所对应的特征值。

# 主成分分析: 算法描述

$$\Sigma \mathbf{w}_i = \lambda_i \mathbf{w}_i, \text{ 且 } \text{trace}(\mathbf{W}^T \Sigma \mathbf{W}) = \sum_{i=1}^l \mathbf{w}_i^T \Sigma \mathbf{w}_i = \sum_{i=1}^l \lambda_i$$

- 可见, 在主成份分析中, 最优化的方差等于原始样本数据 $\mathbf{X}$ 的协方差矩阵 $\Sigma$ 的特征根之和。
- 要使方差最大, 我们可以求得协方差矩阵 $\Sigma$ 的特征向量和特征根, 然后取前 $k$ 个最大特征根所对应的特征向量组成映射矩阵 $\mathbf{W}$ 即可。
- 注意, 每个特征向量 $\mathbf{w}_i$ 与原始数据 $\mathbf{x}_i$ 的维数是一样的, 均为 $n$ 。



# 主成分分析: 算法描述

- 输入:  $n$  个  $d$  维样本数据所构成的矩阵  $\mathbf{X}$ , 降维后的维数  $l$

- 输出: 映射矩阵  $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l\}$

- 算法步骤:

- 1: 对于每个样本数据  $\mathbf{x}_i$  进行中心化处理:  $\mathbf{x}_i = \mathbf{x}_i - \mu$ ,  $\mu = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j$

- 2: 计算原始样本数据的协方差矩阵:  $\Sigma = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$

- 3: 对协方差矩阵  $\Sigma$  进行特征值分解, 对所得特征根按其值大到小排序  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_l$

- 4: 取前  $l$  个最大特征根所对应特征向量  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l$  组成映射矩阵  $\mathbf{W}$

- 5: 将每个样本数据  $\mathbf{x}_i$  按照如下方法降维:  $(\mathbf{x}_i)_{1 \times d} (\mathbf{W})_{d \times l} = 1 \times l$

# 特征人脸方法: 动机

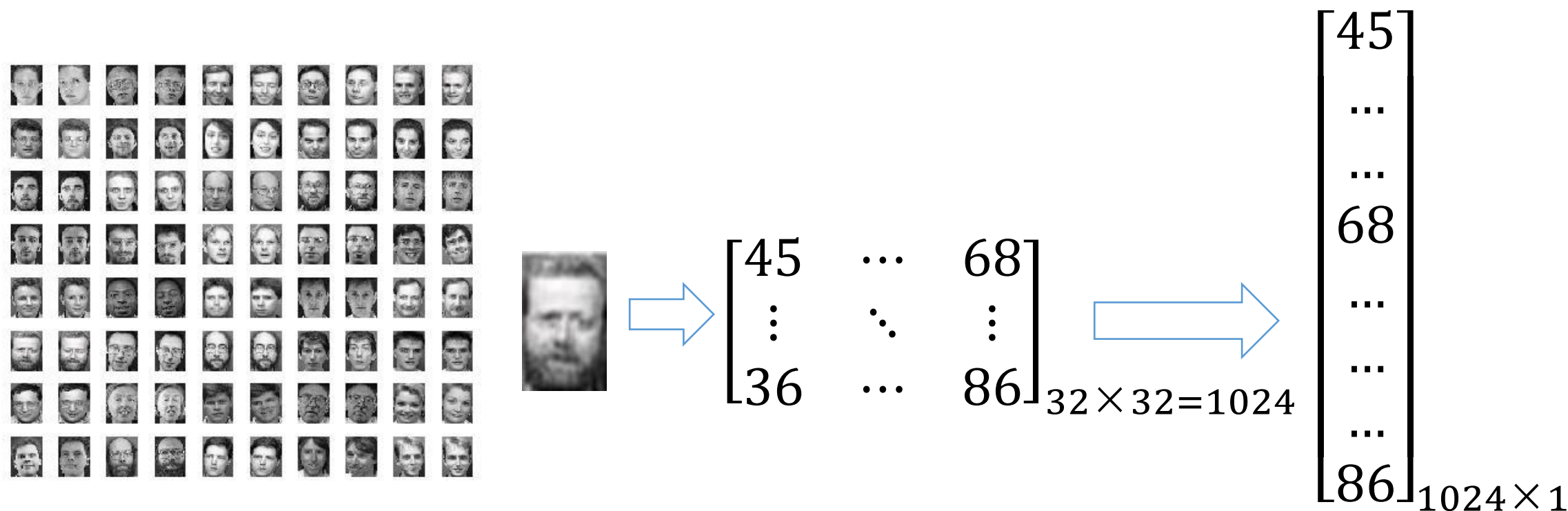
- 特征人脸方法是一种应用主成份分析来实现人脸图像降维的方法，其本质是用一种称为“特征人脸(eigenface)”的特征向量按照线性组合形式来表达每一张原始人脸图像，进而实现人脸识别。
- 由此可见，这一方法的关键之处在于如何得到特征人脸。



用（特征）人脸表示人脸，  
而非用像素点表示人脸

# 特征人脸方法: 算法描述

- 将每幅人脸图像转换成列向量
- 如将一幅 $32 \times 32$ 的人脸图像转成 $1024 \times 1$ 的列向量



# 特征人脸方法: 算法描述

- 输入:  $n$  个1024维人脸样本数据构成的矩阵 $\mathbf{X}$ , 降维后的维数 $d$
- 输出: 映射矩阵 $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l\}$  (其中 $\mathbf{w}_j$  ( $1 \leq j \leq l$ )是特征人脸)
- 算法步骤:
  - 1: 对于每个样本 $\mathbf{x}_i$ 进行中心化处理:  $\mathbf{x}_i = \mathbf{x}_i - \mu$ ,  $\mu = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j$
  - 2: 计算原始人脸样本数据的协方差矩阵:  $\Sigma = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$
  - 3: 对 $\Sigma$ 进行特征值分解, 对所得特征根从大到小排序 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$
  - 4: 取前 $l$ 个最大特征根所对应特征向量 $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l$ 组成映射矩阵 $\mathbf{W}$
  - 5: 将每个人脸图像 $\mathbf{x}_i$ 按照如下方法降维:  $(\mathbf{x}_i)_{1 \times d} (\mathbf{W})_{d \times l} = 1 \times l$

# 特征人脸方法: 算法描述

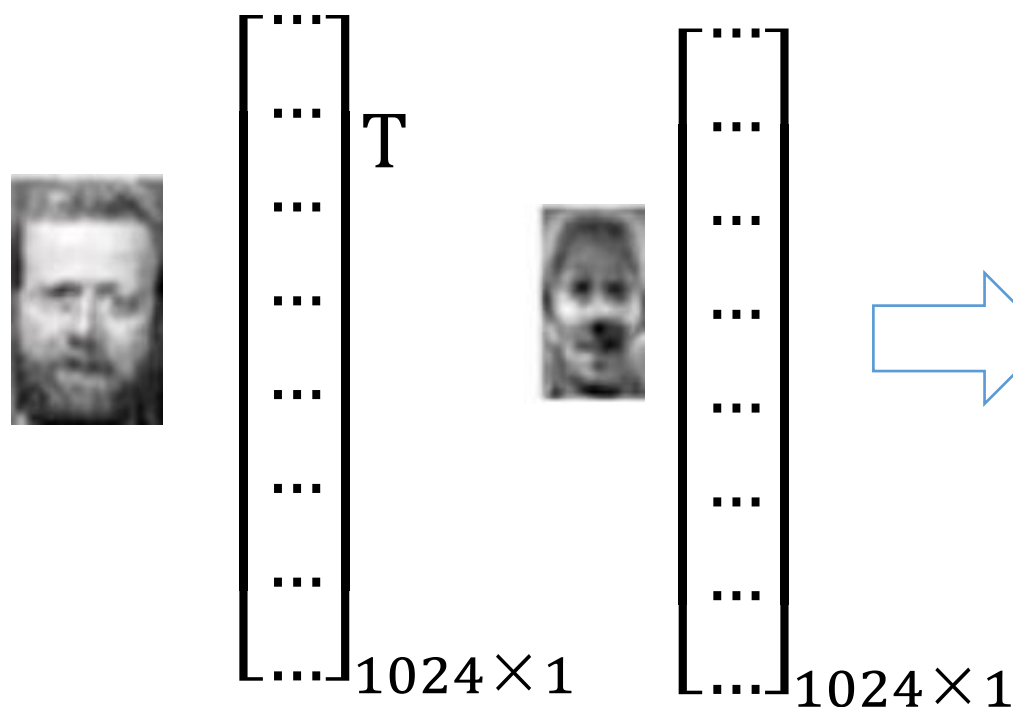
- 每个人脸特征向量  $\phi_i$  与原始人脸数据  $x_i$  的维数是一样的，均为1024。
- 可将每个特征向量还原为  $32 \times 32$  的人脸图像，称之为特征人脸，因此可得到  $n$  个特征人脸。



400个人脸（左）和与之对应的36个特征人脸

# 基于特征人脸的降维

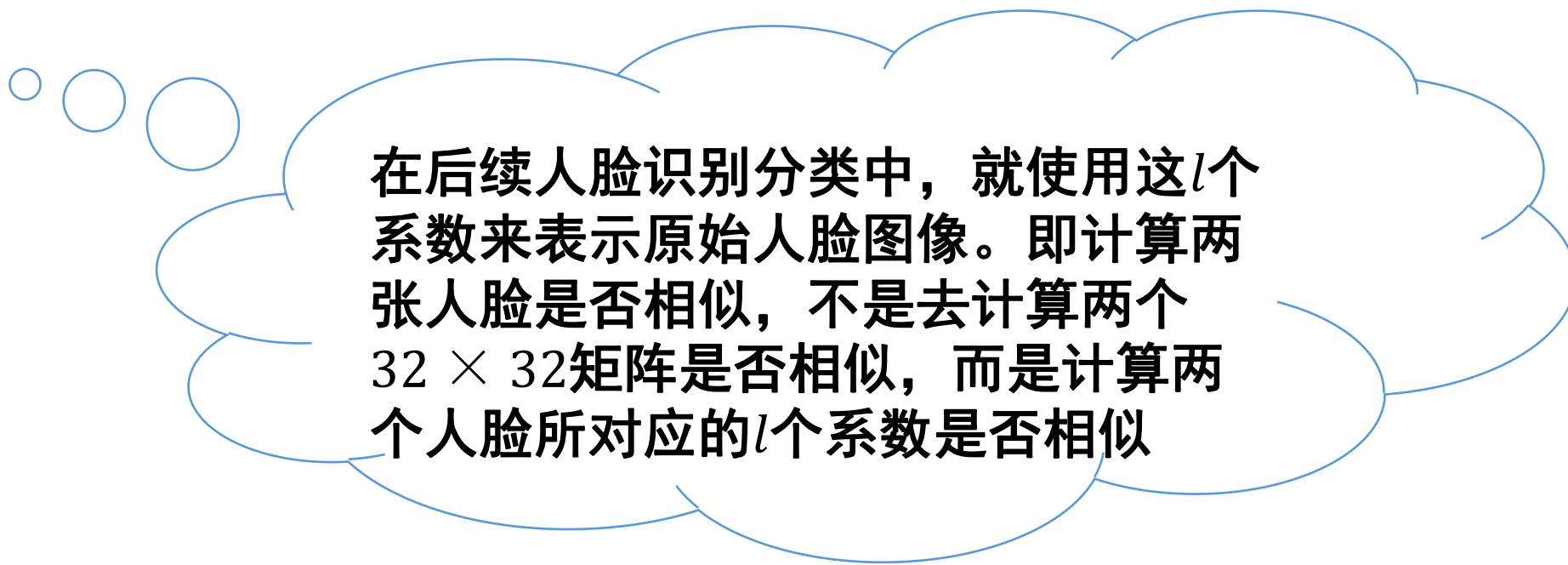
- 将每幅人脸分别与每个特征人脸做矩阵乘法，得到一个相关系数
- 每幅人脸得到 $Q$ 个相关系数 $\Rightarrow$ 每幅人脸从1024维约减到 $Q$ 维



每幅人脸图像与特征人脸  
做矩阵乘法得到一个相关  
系数

# 基于特征人脸的降维

- 由于每幅人脸是所有特征人脸的线性组合，因此就实现人脸从“像素点表达”到“特征人脸表达”的转变。每幅人脸从1024维约减到 $l$ 维。
- 使用 $l$ 个特征人脸的线性组合来表达原始人脸数据 $x_i$



在后续人脸识别分类中，就使用这 $l$ 个系数来表示原始人脸图像。即计算两张人脸是否相似，不是去计算两个 $32 \times 32$ 矩阵是否相似，而是计算两个人脸所对应的 $l$ 个系数是否相似



# 人脸表达的方法对比：聚类、主成份分析、非负矩阵分解

- 特征人脸表示：使用  $l$  个特征人脸的线性组合来表达原始人脸数据  $x_i$

$$x_i = \alpha_{i1} \times \text{img}_1 + \alpha_{i2} \times \text{img}_2 + \dots + \alpha_{il} \times \text{img}_l$$

- 聚类表示：用待表示人脸最相似的聚类质心来表示



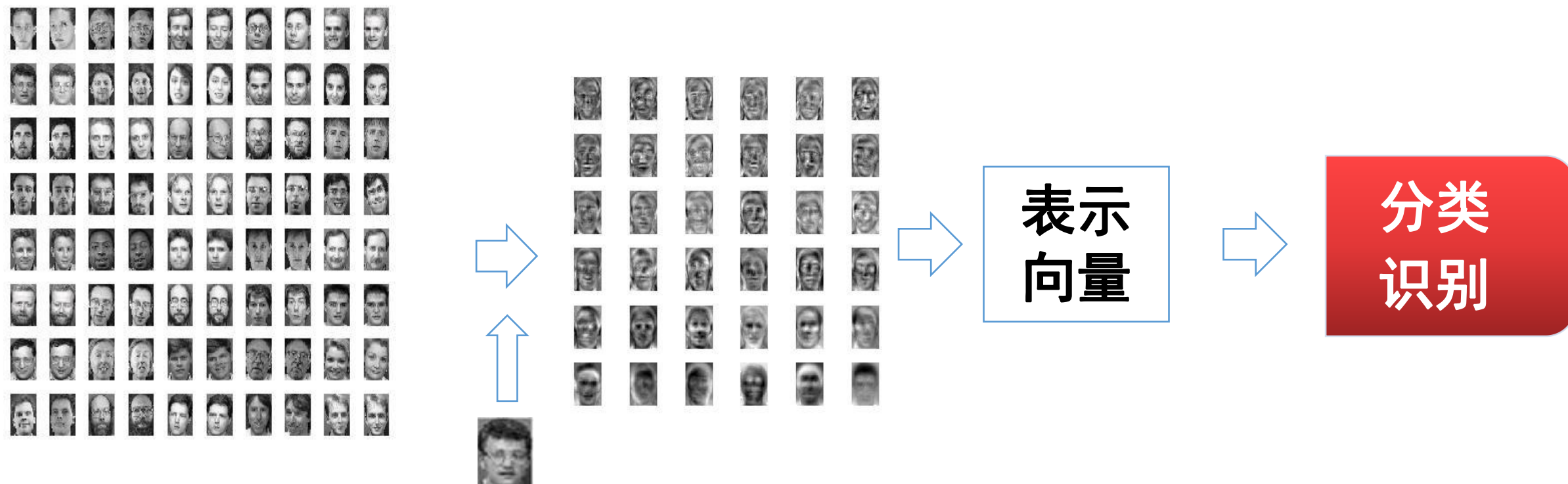


# 人脸表达的方法对比：聚类、主成份分析、非负矩阵分解

- 非负矩阵人脸分解方法表示：通过若干个特征人脸的线性组合来表达原始人脸数据 $x_i$ ，体现了“部分组成整体”



# 人脸表达后的分析与处理



# 模型参数估计

- 无论是最大似然估计算法或者是最大后验估计算法，都是充分利用已有数据，在参数模型确定（只是参数值未知）情况下，对所优化目标中的参数求导，令导数为0，求取模型的参数值。
- 在解决一些具体问题时，难以事先就将模型确定下来，然后利用数据来求取模型中的参数值。在这样情况下，无法直接利用最大似然估计算法或者最大后验估计算法来求取模型参数。

# 期望最大化算法（Expectation Maximization, EM）

- EM算法是一种重要的用于解决含有隐变量（latent variable）问题的参数估计方法。
- EM算法分为求取期望（E步骤，expectation）和期望最大化（M步骤，maximization）两个步骤。
- 在EM算法的E步骤时，先假设模型参数的初始值，估计隐变量取值；在EM算法的M步骤时，基于观测数据、模型参数和隐变量取值一起来最大化“拟合”数据，更新模型参数。基于所更新的模型参数，得到新的隐变量取值（EM算法的E步），然后继续极大化“拟合”数据，更新模型参数（EM算法的M步）。以此类推迭代，直到算法收敛，得到合适的模型参数。

# 期望最大化算法：二硬币投掷例子

- 假设有A和B两个硬币，进行五轮掷币实验：在每一轮实验中，先随机选择一个硬币，然后用所选择的硬币投掷十次，将投掷结果作为本轮实验观测结果。H代表硬币正面朝上、T代表硬币反面朝上。

表5.3 两个硬币投掷5轮（每轮10次）的结果

1	H	T	T	T	H	H	T	H	T	H
2	H	H	H	H	T	H	H	H	H	H
3	H	T	H	H	H	H	H	T	H	H
4	H	T	H	T	T	T	H	H	T	T
5	T	H	H	H	T	H	H	H	T	H

表5.3 两个硬币投掷5轮（每轮10次）的结果

# 期望最大化算法：二硬币投掷例子

- 从这十轮观测数据出发，计算硬币A或硬币B被投掷为正面的概率。记硬币A或硬币B被投掷为正面的概率为  $\theta = \{\theta_A, \theta_B\}$
- 求取期望E 步骤：
  - 初始化每一轮中硬币A和硬币B投掷为正面的概率为  $\theta_A^{(0)} = 0.60$  和  $\theta_B^{(0)} = 0.50$ 。
  - 基于 “HTTTHHTHTH” 这10次投掷结果，由硬币A投掷所得概率为：

$$\begin{aligned} & P(\text{选择硬币A投掷} | \text{硬币投掷结果}, \theta) \\ &= \frac{P(\text{选择硬币A投掷} \text{ 硬币投掷结果} \theta)}{P(\text{选择硬币A投掷, 硬币投掷结果} \theta) + (P\text{选择硬币B投掷 硬币投掷结果} \theta)} \\ &= \frac{(0.6)^5 \times (0.4)^5}{(0.6)^5 \times (0.4)^5 + (0.5)^{10}} = 0.45 \end{aligned}$$



# 期望最大化算法：二硬币投掷例子

这10次结果由硬币B投掷所得概率为：

$$P(\text{选择硬币B投掷} | \text{硬币投掷结果}, \theta)$$

$$= 1 - P(\text{选择硬币A投掷} | \text{硬币投掷结果}, \theta)$$

$$= 0.55$$

184

表5.4 两个硬币在每一轮中被选择概率以及投掷正面/反面的次数

轮次	选硬币 A 概率	选硬币 B 概率	硬币 A 为正面期望次数	硬币 A 为反面期望次数	硬币 B 为正面期望次数	硬币 B 为反面期望次数
1	0.45	0.55	2.25	2.25	2.75	2.75
2	0.80	0.20	7.24	0.80	1.76	0.20
3	0.73	0.27	5.87	1.47	2.13	0.53
4	0.35	0.65	1.41	2.11	2.59	3.89
5	0.65	0.35	4.53	1.94	2.47	1.07
合计			21.30	8.57	11.70	8.43

表5.4 两个硬币在每一轮中被选择概率以及投掷正面/反面的次数



# 期望最大化算法：二硬币投掷例子

- 期望最大化（M步骤，Maximization）：
  - 在上面的计算中，通过初始化硬币A和硬币B投掷得到正面概率 $\theta_A^{(0)}$ 和 $\theta_B^{(0)}$ ，得到每一轮中选择硬币A和选择硬币B概率这一“隐变量”，进而可计算得到每一轮中硬币A和硬币B投掷正面次数。
  - 在这些信息基础上，可更新得到硬币A和硬币B投掷为正面的概率，从而得到新的模型参数：

$$\hat{\theta}_A^{(1)} = \frac{21.30}{21.30+8.57} = 0.713 \quad \hat{\theta}_B^{(1)} = \frac{11.70}{11.70+8.43} = 0.581$$

# 期望最大化算法：二硬币投掷例子

- 接下来，可在新的概率值基础上继续计算每一轮投掷中选择硬币A或硬币B的概率，进而计算得到五轮中硬币A和硬币B投掷正面的总次数，从而得到硬币A和硬币B投掷为正面的更新概率值 $\hat{\theta}_A^{(2)}$ 和 $\hat{\theta}_B^{(2)}$ 。上述算法不断迭代，直至算法收敛，最终得到硬币A和硬币B投掷为正面的概率 $\theta = \{\theta_A, \theta_B\}$ 。

# 期望最大化算法：二硬币投掷例子

表5.5 两个硬币投掷结果为正面的概率迭代计算结果

迭代次数	硬币A为正面次数	硬币A为反面次数	硬币B为正面次数	硬币B为反面次数	硬币A投掷正面概率 $\theta_A$	硬币B投掷正面概率 $\theta_B$
1	21.30	8.57	11.70	8.43	$\hat{\theta}_A^{(1)} = 0.713$	$\hat{\theta}_B^{(1)} = 0.581$
2	19.21	6.56	13.79	10.44	$\hat{\theta}_A^{(2)} = 0.745$	$\hat{\theta}_B^{(2)} = 0.569$
3	19.41	5.86	13.59	11.14	$\hat{\theta}_A^{(3)} = 0.768$	$\hat{\theta}_B^{(3)} = 0.550$
4	19.75	5.47	13.25	11.53	$\hat{\theta}_A^{(4)} = 0.783$	$\hat{\theta}_B^{(4)} = 0.535$
5	19.98	5.28	13.02	11.72	$\hat{\theta}_A^{(5)} = 0.791$	$\hat{\theta}_B^{(5)} = 0.526$
6	20.09	5.19	12.91	11.81	$\hat{\theta}_A^{(6)} = 0.795$	$\hat{\theta}_B^{(6)} = 0.522$
7	20.14	5.16	12.86	11.84	$\hat{\theta}_A^{(7)} = 0.796$	$\hat{\theta}_B^{(7)} = 0.521$
8	20.16	5.15	12.84	11.85	$\hat{\theta}_A^{(8)} = 0.796$	$\hat{\theta}_B^{(8)} = 0.520$
9	20.17	5.15	12.83	11.85	$\hat{\theta}_A^{(9)} = 0.797$	$\hat{\theta}_B^{(9)} = 0.520$
10	20.18	5.15	12.82	11.85	$\hat{\theta}_A^{(10)} = 0.797$	$\hat{\theta}_B^{(10)} = 0.520$

表5.5 两个硬币投掷结果为正面的概率迭代计算结果

# 期望最大化算法：二硬币投掷例子

- 从这十轮观测数据出发，计算硬币A或硬币B被投掷为正面的概率。记硬币A或硬币B被投掷为正面的概率为  $\theta = \{\theta_A, \theta_B\}$
- 求取期望E 步骤：
  - 初始化每一轮中硬币A和硬币B投掷为正面的概率为  $\theta_A^{(0)} = 0.60$  和  $\theta_B^{(0)} = 0.50$ 。
  - 基于 “HTTTHHTHTH” 这10次投掷结果，由硬币A投掷所得概率

为：

$$\begin{aligned} & P(\text{选择硬币A投掷} | \text{硬币投掷结果}, \theta) \\ &= \frac{P(\text{选择硬币A投掷} \text{ 硬币投掷结果} \theta)}{P(\text{选择硬币A投掷, 硬币投掷结果} \theta) + (P\text{选择硬币B投掷 硬币投掷结果} \theta)} \\ &= \frac{(0.6)^5 \times (0.4)^5}{(0.6)^5 \times (0.4)^5 + (0.5)^{10}} = 0.45 \end{aligned}$$

# 期望最大化算法：二硬币投掷例子

- 隐变量：每一轮选择硬币A还是选择硬币B来完成10次投掷是一个隐变量，硬币A和硬币B投掷结果为正面的概率 $\theta = \{\theta_A, \theta_B\}$ 称为模型参数。
- 计算隐变量（EM 算法的 E 步）、最大化似然函数和更新模型参数（EM算法的M步）：
  - EM 算法使用迭代方法来求解模型参数 $\theta = \{\theta_A, \theta_B\}$ ：先初始化模型参数，然后计算得到隐变量（EM 算法的 E 步），接着基于观测投掷结果和当前隐变量值一起来最大化似然函数（即使得模型参数能够更好拟合观测结果），更新模型参数（EM算法的M步）。基于当前得到的模型参数，继续更新隐变量（EM算法的 E 步），然后继续最大化似然函数，更新模型参数（EM算法的M步）。以此类推，不断迭代下去，直到模型参数基本无变化，算法收敛。

# 期望最大化算法：EM算法一般形式

- 对于 $n$ 个相互独立的样本 $X = \{x_1, x_2, \dots, x_n\}$ 及其对应的隐变量 $Z = \{z_1, z_2, \dots, z_n\}$ ，在假设样本的模型参数为 $\theta$ 前提下，观测数据 $x_i$ 的概率为 $P(x_i|\theta)$ ，完全数据 $(x_i, z_i)$ 的似然函数为 $P(x_i, z_i|\theta)$ 。
- 在上面的表示基础上，优化目标为求解合适的 $\theta$ 和 $Z$ 使得对数似然函数最大：

$$(\theta, Z) = \underset{\theta, Z}{\operatorname{argmax}} L(\theta, Z) = \underset{\theta, Z}{\operatorname{argmax}} \sum_{i=1}^n \log \sum_{z_i} P(x_i, z_i|\theta)$$

# 期望最大化算法：EM算法一般形式

- 但是，优化求解含有未观测数据 $Z$ 的对数似然函数 $L(\theta, Z)$ 十分困难，EM算法不断构造对数似然函数 $L(\theta, Z)$ 的一个下界（E步骤），然后最大化这个下界（M步骤），以迭代方式逼近模型参数所能取得极大似然值。

- 求解目标：

$$(\theta, Z) = \operatorname{argmax}_{\theta, Z} L(\theta, Z) = \operatorname{argmax}_{\theta, Z} \sum_{i=1}^n \log \sum_{z_i} P(x_i, z_i | \theta)$$

$$\sum_{i=1}^n \log \sum_{z_i} P(x_i, z_i | \theta) = \sum_{i=1}^n \log \sum_{z_i} Q_i(z_i) \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$$



# 期望最大化算法：EM算法一般形式

$$\begin{aligned}\sum_{i=1}^n \log \sum_{z_i} P(x_i, z_i | \theta) &= \sum_{i=1}^n \log \sum_{z_i} Q_i(z_i) \frac{P(x_i, z_i | \theta)}{Q_i(z_i)} \\ &\geq \underbrace{\sum_{i=1}^n \sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}}_{\text{对数似然函数下界}}\end{aligned}$$

- 在上式中， $Q_i(z_i)$ 是隐变量分布，满足： $\sum_{z_i} Q_i(z_i) = 1$  ( $0 \leq Q_i(z_i)$ )。
- 上述不等式中使用了Jensen 不等式 (Jensen's inequality)。对于凹函数 $f$ ，Jensen 不等式使得下面不等式成立：

$$\log(E(f)) \geq E(\log(f)), \text{ 其中 } E(f) = \sum_i \lambda_i f_i, \lambda_i \geq 0, \quad \sum_i \lambda_i = 1$$



# 期望最大化算法：EM算法一般形式

- 令  $f_i = \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$  和  $\lambda_i = Q_i(z_i)$ ，则根据 Jensen 不等式的定义，可将  $\frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$  视为第  $i$  个样本， $Q_i(z_i)$  为第  $i$  个样本的权重。按照这样的约定，可得到如下式子：

$$E\left(\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}\right) = \sum_{z_i} \underbrace{Q_i(z_i)}_{\text{权重}} \underbrace{\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}}_{\text{样本值}}$$

于是，为了最大化  $\sum_{i=1}^n \log \sum_{z_i} P(x_i, z_i|\Theta)$  这一对数似然函数，只需最大化其下界

$\sum_{i=1}^n \sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$ 。这个下界实际上是  $\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$  的加权求和。

# 期望最大化算法：EM算法一般形式

- 由于权重 $Q_i(z_i)$ 累加之和为1，因此 $\sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$ 就是  $\log \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$  的加权平均，也就是所谓的期望，这就是EM算法中Expectation这一单词的来源。于是，EM算法就是不断最大化这一下界（M步骤），从而通过迭代的方式逼近模型参数的极大似然估计值。
- 显然，当 $\theta$ 取值给定后，对数似然函数的下界只与 $P(x_i, z_i)$ 和 $Q_i(z_i)$ 相关。于是，通过调整 $P(x_i, z_i)$ 和 $Q_i(z_i)$ 取值，使得似然函数下界不断逼近似然函数真实值。那么，当不等式取等式时，调整后的似然函数下界等于似然函数真实值。当每个样本取值均相等时（也就是每个样本取值为同一个常数），Jensen 不等式中的等式成立。

# 期望最大化算法：EM算法一般形式

• 于是令  $\frac{P(x_i, z_i|\Theta)}{Q_i(z_i)} = c$  ( $c$  为常数)，得到  $P(x_i, z_i|\Theta) = cQ_i(z_i)$ 。由于  $\sum_{z_i} Q_i(z_i) = 1$ ，可知  $\sum_{z_i} P(x_i, z_i|\Theta) = c$ 。

于是， $Q_i(z_i) = \frac{P(x_i, z_i|\Theta)}{c} = \frac{P(x_i, z_i|\Theta)}{\sum_{z_i} P(x_i, z_i|\Theta)} = \frac{P(x_i, z_i|\Theta)}{P(x_i|\Theta)} = P(z_i|x_i, \Theta)$ 。也

就是说，只要  $Q_i(z_i) = P(z_i|x_i, \Theta)$ ，就能够保证对数似然函数最大值与其下界相等。从上面的阐述可知，固定参数  $\Theta$  后，只要从观测数据  $x_i$  和参数  $\Theta$  出发，令  $Q_i(z_i) = P(z_i|x_i, \Theta)$ ，则可以得到对数似然函数最大值的下界，这就是EM算法中的E步骤。然后，固定  $Q_i(z_i)$ ，调整  $\Theta$ ，再去极大化对数似然函数最大值的下界，这就是EM算法的M步骤。

# 期望最大化算法：EM算法一般形式

- 在算法5.3中，固定上一步所得参数 $\Theta^t$ 情况下，通过计算 $Z$ 的后验概率来得到能够逼近真实最大似然估计的下界，与算法中的求取期望步骤一致。在期望最大化步骤中，通过最大化对数似然函数，可以得到新的参数 $\Theta^{t+1}$ ，即：

$$\Theta^{t+1} = \operatorname{argmax}_{\Theta} \sum_{i=1}^n \sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i | \Theta)}{Q_i(z_i)}$$

# 期望最大化算法：EM算法一般形式

- 因为  $Q_i(z_i) = p(z_i|x_i, \theta)$  是由上一步的  $\theta^t$  估计出的，与下一步要优化的  $\theta^{t+1}$  无关，所以上式等价于：

$$\theta^{t+1} = \operatorname{argmax}_{\theta} \sum_i \sum_{z_i} Q_i(z_i) \log p(x_i, z_i | \theta) - Q_i(z_i) \log Q_i(z_i)$$

$$= \operatorname{argmax}_{\theta} \sum_i \sum_{z_i} Q_i(z_i) \log p(x_i, z_i | \theta)$$

# 期望最大化算法：EM算法一般形式

- 广义EM算法中，E步骤是固定参数来优化隐变量分布，M步骤是固定隐变量分布来优化参数，两者不同交替迭代。至此，证明了EM算法能够通过不断最大化下界来逼近最大似然估计值。

# 期望最大化算法：EM算法一般形式

- 假设 $\Theta^{t+1}$ 和 $\Theta^t$ 是两个相邻的迭代更新得到的参数值，如果能证明 $L(\Theta^t) \leq L(\Theta^{t+1})$ ，即可认为EM算法能够让似然单调增长。下面给出简略的证明：

$$L(\Theta^{t+1}) \geq \sum_i \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i | \Theta^{t+1})}{Q_i(z_i)}$$
$$\Rightarrow l(\Theta^{t+1}) \geq \sum_i \sum_{z_i} Q_i^t(z_i) \log \frac{p(x_i, z_i | \Theta^{t+1})}{Q_i^t(z_i)} \text{ (特殊化)}$$

# 期望最大化算法：EM算法一般形式

- 因为  $\Theta^{t+1} = \operatorname{argmax}_{\Theta} \sum_i \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i | \Theta)}{Q_i(z_i)}$ ，即对于  $\Theta^t$ （事实上可以是任意的  $\Theta$ ）都有下面的式子成立：

$$\begin{aligned} \sum_i \sum_{z_i} Q_i^t(z_i) \log \frac{p(x_i, z_i | \Theta^{t+1})}{Q_i^t(z_i)} &\geq \sum_i \sum_{z_i} Q_i^t(z_i) \log \frac{p(x_i, z_i | \Theta^t)}{Q_i^t(z_i)} \\ &= L(\Theta^t) \end{aligned}$$

- 至此，证明了  $L(\Theta^{t+1}) \geq L(\Theta^t)$ ，即EM算法能够保证似然度取值单调增长，即EM算法能够稳定收敛



# 谢谢!