



重庆邮电大学

计算机科学与技术学院

# 人工智能原理

## 无监督学习

## K均值聚类 (K-means 聚类)

- 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：继续迭代

■ 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

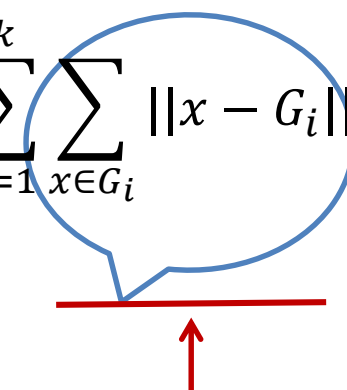
■ 根据新的聚类结果、更新聚类质心

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均值聚类算法的另一个视角：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:  $\text{var}(G_i) = \frac{1}{|G_i|} \sum_{x \in G_i} \|x - G_i\|^2$

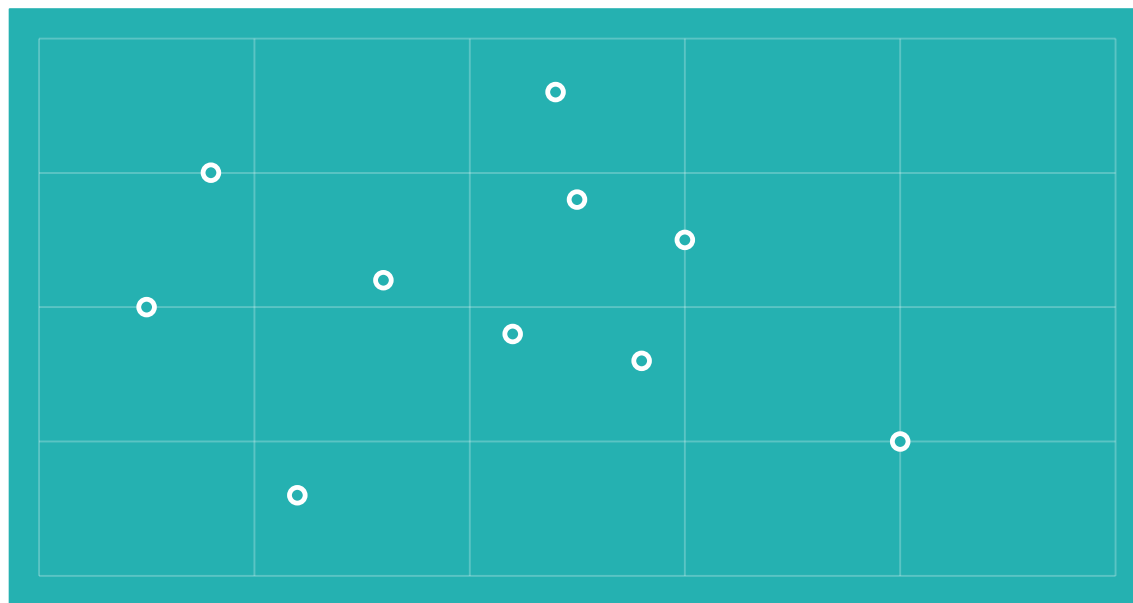
- 欧氏距离与方差量纲相同
- 最小化每个类簇方差将使得最终聚类结果中每个聚类集合中所包含数据呈现出来差异性最小。



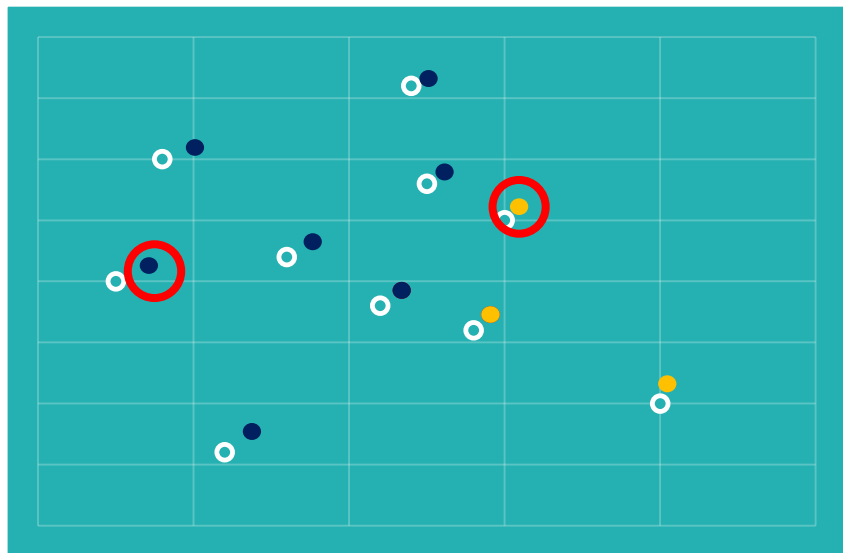
## K-means聚类算法实例求解过程

| U  | x   | y   |
|----|-----|-----|
| 1  | 0.5 | 2   |
| 2  | 0.8 | 3   |
| 3  | 1.2 | 0.6 |
| 4  | 1.6 | 2.2 |
| 5  | 2.2 | 1.8 |
| 6  | 2.4 | 3.6 |
| 7  | 2.5 | 2.8 |
| 8  | 2.8 | 1.6 |
| 9  | 3   | 2.5 |
| 10 | 4   | 1   |

给出数据对象集合如左图所示  
簇的数量 $k=2$   
样本点分布如下图



## K-means聚类算法实例求解过程



(1)任意选取两个点作为两个簇的初始中心，  
如 $C1 = (2.2, 1.8)$ ， $C2 = (2.8, 1.6)$

(2)对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇对

例如，对点 $(0.5, 2)$

$$d1 = \sqrt{(0.5 - 2.2)^2 + (2 - 1.8)^2} = 1.712$$

$$d2 = \sqrt{(0.5 - 2.8)^2 + (2 - 1.6)^2} = 2.377$$

显然  $d1 < d2$ ，故将点 $(0.5, 2)$ 分配给簇A。

对点 $(3, 2.5)$

$$d1 = \sqrt{(3 - 2.2)^2 + (2.5 - 1.8)^2} = 1.063$$

$$d2 = \sqrt{(3 - 2.8)^2 + (2.5 - 1.6)^2} = 0.912$$

这里  $d2 < d1$ ，故将点 $(3, 2.5)$ 分配给簇B。

同理，对剩余的点进行分配，最终划分结果为：

簇A： $(0.5, 2), (0.8, 3), (1.2, 0.6), (1.6, 2.2), (2.4, 3.6), (2.5, 2.8)$

簇B： $(2.8, 1.6), (3, 2.5), (4, 1)$

# K-means聚类算法实例求解过程

## (3) 计算新的簇中心

$$c1_x = \frac{0.5 + 0.8 + 1.2 + 1.6 + 2.2 + 2.4 + 2.5}{7} = 1.6$$

$$c1_y = \frac{2 + 3 + 0.5 + 2.2 + 1.3 + 3.6 + 3.8}{7} = 2.3$$

$$c2_x = \frac{2.8 + 3 + 4}{3} = 3.3$$

$$c2_y = \frac{1.5 + 2.5 + 1}{3} = 1.7$$

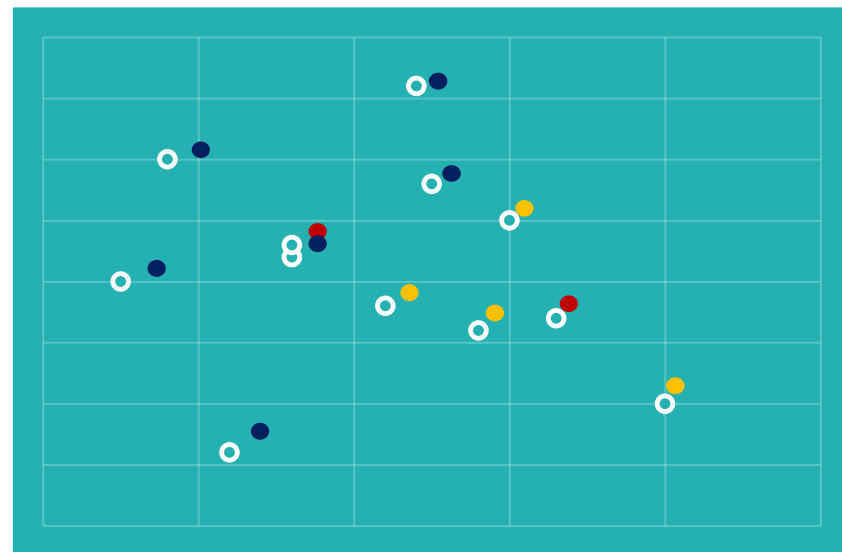
故,  $C1=(1.6,2.3)$ ,  $C2=(3.3,1.7)$

## (4) 重新计算数据集中每个点到两个簇中心的距离, 根据其值进行重新分配

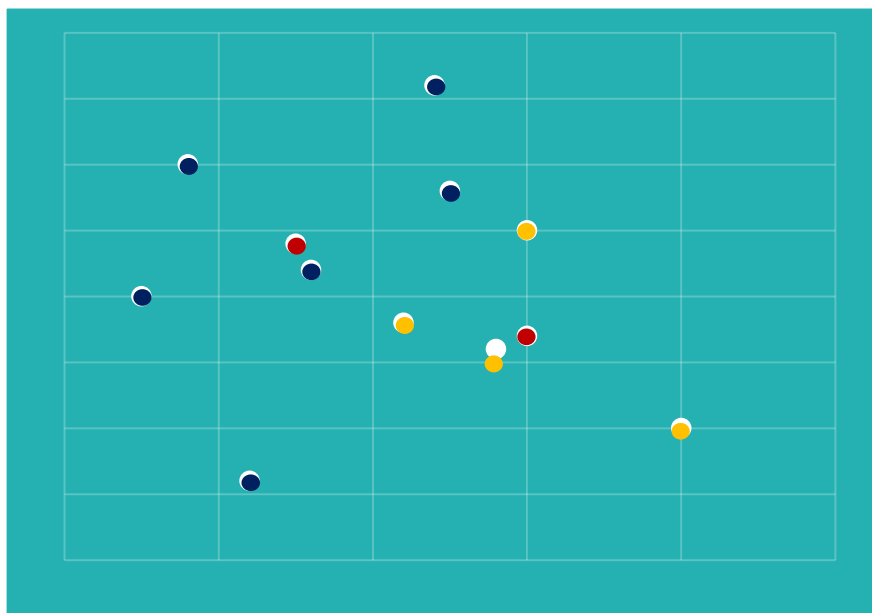
分配结果为:

簇A: (0.5,2),(0.8,3),(1.6,2.2),(1.2,0.6),(2.4,3.6),(2.5,2.8)

簇B: (2.2,1.8),(3,2.5),(2.8,1.6),(4,1)



## K-means聚类算法实例求解过程



(5) 计算新的簇中心

$C1=(1.5,2.4)$ ,  $C2=(3,1.7)$

(6) 再次分配结果为:

簇A:  $(0.5,2),(0.8,3),(1.6,2.2),(1.2,0.6),(2.4,3.6),(2.5,2.8)$

簇B:  $(2.2,1.8),(3,2.5),(2.8,1.6),(4,1)$

在三次迭代之后, 此时簇中心不变, 所以停止迭代过程, 算法停止

## 随堂练习

假设有如下8个点：

A1 (3,1), A2 (3,2), A3 (4,1), A4 (4,2), A5 (1,3), A6 (1,4),  
A7 (2,3), A8 (2,4)。

使用K-means算法对其进行聚类。设初始聚类中心分别为D1 (0,4)和D2 (3,3)。请写出简略的计算过程。

## 参考答案

### 第一轮:

|         | D1(0,4)            | D2(3,3)            |
|---------|--------------------|--------------------|
| A1(3,1) | 4.242              | 2 $\checkmark$     |
| A2(3,2) | 3.605              | 1 $\checkmark$     |
| A3(4,1) | 5                  | 2.236 $\checkmark$ |
| A4(4,2) | 4.472              | 1.414 $\checkmark$ |
| A5(1,3) | 1.414 $\checkmark$ | 2                  |
| A6(1,4) | 1 $\checkmark$     | 2.236              |
| A7(2,3) | 2.236              | 1 $\checkmark$     |
| A8(2,4) | 2                  | 1.414 $\checkmark$ |

## 参考答案

### 第一轮:

根据上表分成两簇, {A1, A2, A3, A4, A7, A8},  
{A5, A6}。重新计算新的聚类中心D3, D4。并计算新的距离表。

$$D3 = (3+3+4+4+2+2) / 6, (1+2+1+2+3+4) / 6 = (3, 2.167)$$

$$D4 = (1+1) / 2, (3+4) / 2 = (1, 3.5)$$

## 参考答案

### 第二轮:

|         | D3(3, 2.167) | D4(1,3.5) |
|---------|--------------|-----------|
| A1(3,1) | 1.167 √      | 3.201     |
| A2(3,2) | 0.167 √      | 2, 5      |
| A3(4,1) | 1.536 √      | 3.905     |
| A4(4,2) | 1.013 √      | 3.354     |
| A5(1,3) | 2.166        | 0.5 √     |
| A6(1,4) | 2.712        | 0.5 √     |
| A7(2,3) | 1.301        | 1.118 √   |
| A8(2,4) | 2.088        | 1.118 √   |



## 参考答案

### 第二轮:

根据上表分成两簇, {A1, A2, A3, A4}, {A5, A6, A7, A8}。重新计算新的聚类中心D5, D6。并计算新的距离表。

$$D5 = (3+3+4+4) / 4, (1+2+1+2)/4 = (3.5, 1.5)$$

$$D6 = (1+1+2+2) / 4, (3+4+3+4)/4 = (1.5, 3.5)$$

## 参考答案

### 第三轮:

|         | D5(3.5,1.5) | D6(1.5,3.5) |
|---------|-------------|-------------|
| A1(3,1) | 0.707 √     | 2.915       |
| A2(3,2) | 0.707 √     | 2.121       |
| A3(4,1) | 0.707 √     | 3.535       |
| A4(4,2) | 0.707 √     | 2.915       |
| A5(1,3) | 2.915       | 0.707 √     |
| A6(1,4) | 3.535       | 0.707 √     |
| A7(2,3) | 2.121       | 0.707 √     |
| A8(2,4) | 2.915       | 0.707 √     |

## 参考答案

### 第三轮：

根据上表分成两簇， $\{A1, A2, A3, A4\}$ ， $\{A5, A6, A7, A8\}$ ，和步骤四分簇一致，停止计算。

## 改进算法

### k-means++算法

k-means++算法是从选择合适的初始簇心的角度来解决k-means算法对初始簇心敏感的问题。

样本点 $x$ 到 $U$ 的距离 $D(x)$ 是该点到 $U$ 中元素 $u_j$ 的距离的最小值, 即:  $D(x) = \min_j \text{dist}(x, u_j)$

将 $D(x)$ 转化为概率 $p(x)$ 的方法:  $p(x) = \frac{D(x)^2}{\sum_{x' \in S} D(x')^2}$

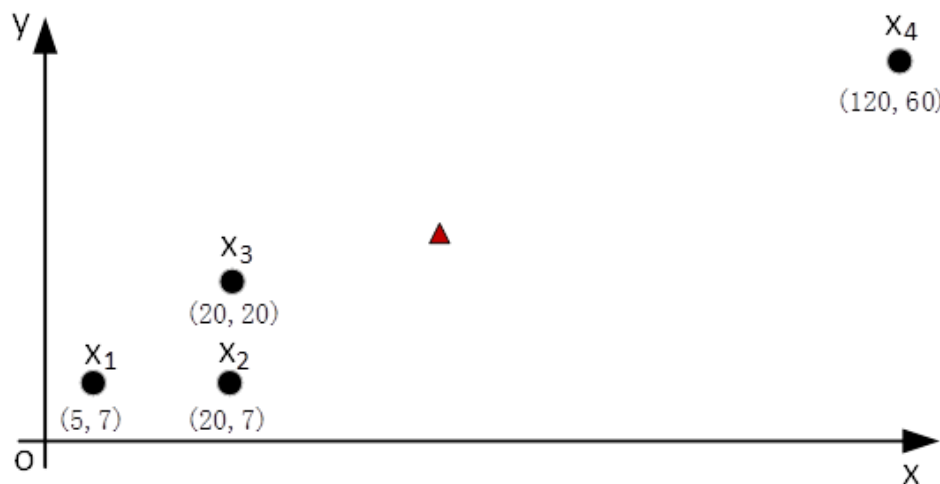
步数

操作

- 1 从样本集 $S$ 中随机选择1个样本点加入簇中心集合 $U$ 中
- 2 对任一样本点 $x$ , 计算它到 $U$ 的距离 $D(x)$
- 3 将 $D(x)$ 转化为对应样本点 $x$ 的概率 $p(x)$
- 4 按所有样本点的概率 $p(x)$ , 选择一个样本点加入簇中心集合 $U$
- 5 重复2、3、4步直至簇中心集合元素个数达到 $k$

## 改进算法

### K-中心点算法



K-中心点算法在簇中选择一个样本点作为簇中心，选择的标准是使簇内各样本点到簇中心的距离和最短。记大小为 $m$ 的簇 $C$ 内样本点为 $x_j$ ，簇中心为 $u$ ，则

$$u = \arg \min_x \sum_{j=0}^{m-1} \text{dist}(x, x_j)$$

## K均值聚类算法的不足

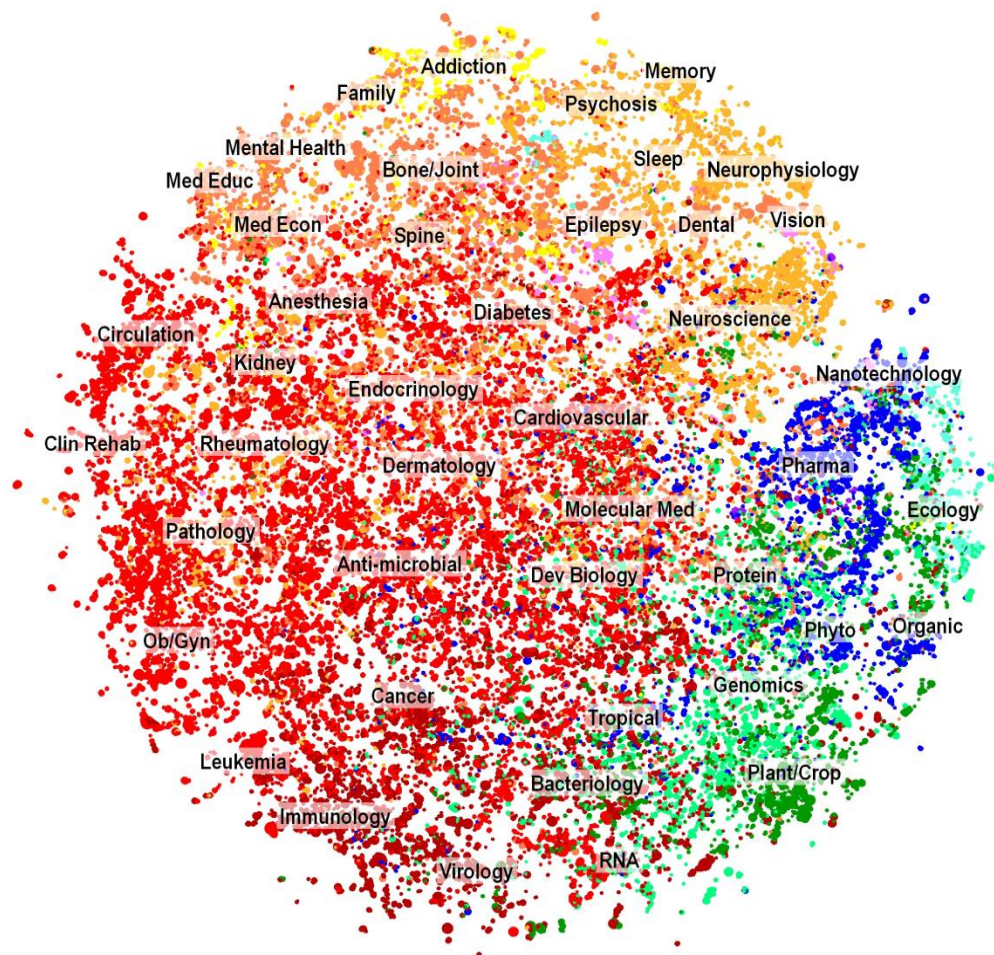
- 需要事先确定聚类数目，很多时候我们并不知道数据应被聚类的数目
- 需要初始化聚类质心，初始化聚类中心对聚类结果有较大的影响
- 算法是迭代执行，时间开销非常大
- 欧氏距离假设数据每个维度之间的重要性是一样的



# K均值聚类算法的应用



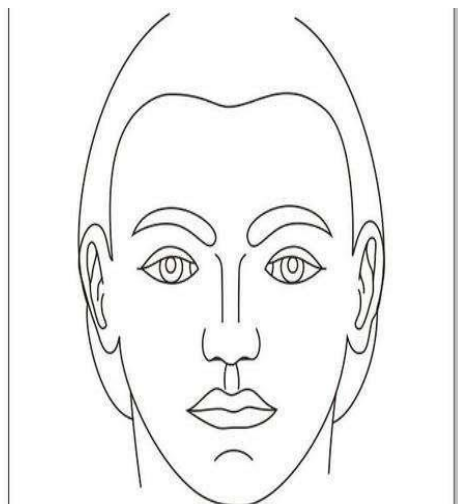
图像压缩



文本聚类：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: 若干概念-方差与协方差

## 数据样本的方差 variance

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

- 方差等于各个数据与样本均值之差的平方和之平均数
- 方差描述了样本数据的波动程度

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

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

# 主成分分析: 若干概念-方差与协方差

数据样本的协方差  
covariance

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

## ● 衡量两个变量之间的相关度

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

其中 $E(X)$ 和 $E(Y)$ 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                                 |
|    | $E(X) = 9.33$ | $E(Y) = 23.67$ | $Var(X) = 57.87$ | $Var(Y) = 231.47$ | $E([x_i - E(X)][y_i - E(Y)]) = 115.73$ |

$X = \{x_i\}, Y = \{y_i\}$   
 计算机科学与技术学院



重庆邮电大学

## 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 $cov(X, Y) > 0$ 时，称 $X$ 与 $Y$ 正相关
- 当协方差 $cov(X, Y) < 0$ 时，称 $X$ 与 $Y$ 负相关
- 当协方差 $cov(X, Y) = 0$ 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:

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

| 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                   | -16.67                                 | 1.0                 |
| 2  | 3             | 11             | -6.33                   | -12.67                   | -12.67                                 |                     |
| 3  | 6             | 17             | -3.33                   | -6.67                    | -6.67                                  |                     |
| 4  | 10            | 25             | 0.67                    | 1.33                     | 1.33                                   |                     |
| 5  | 15            | 35             | 5.67                    | 11.33                    | 11.33                                  |                     |
| 6  | 21            | 47             | 11.67                   | 23.33                    | 23.33                                  |                     |
|    | $E(X) = 9.33$ | $E(Y) = 23.67$ | $\text{Var}(X) = 57.87$ | $\text{Var}(Y) = 231.47$ | $E([x_i - E(X)][y_i - E(Y)]) = 115.73$ |                     |

# 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:

- $|corr(X, Y)| \leq 1$
- $corr(X, Y) = 1$  的充要条件是存在常数  $a$  和  $b$ , 使得  $Y = aX + b$
- 皮尔逊相关系数是对称的, 即  $corr(X, Y) = corr(Y, X)$
- 由此衍生出如下性质: 皮尔逊相关系数刻画了变量  $X$  和  $Y$  之间线性相关程度, 如果  $|corr(X, Y)|$  的取值越大, 则两者在线性相关的意义下相关程度越大。  $|corr(X, Y)| = 0$  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)。
- 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。

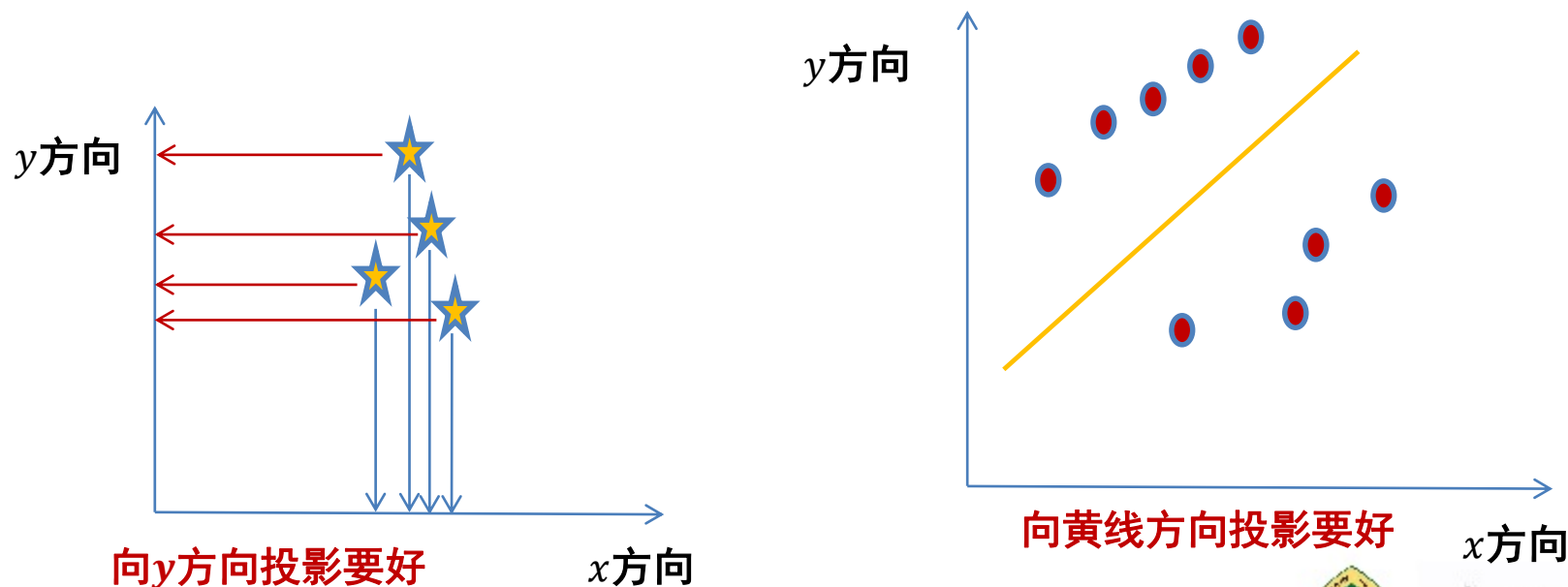
# 主成分分析: 从协方差到相关系数

- 相关性(correlation)与独立性(independence)
  - 如果 $X$ 和 $Y$ 的线性不相关, 则 $|corr(X, Y)| = 0$
  - 如果 $X$ 和 $Y$ 的彼此独立, 则一定 $|corr(X, Y)| = 0$ , 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

保证样本  
投影后方差最大

- 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。如左下图所示, 向 $y$ 方向投影 (使得二维数据映射为一维) 就比向 $x$ 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

- 主成分分析思想是将 $n$ 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 $k$ 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 $x_i (1 \leq i \leq n) \in R^d$ 。
- 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）。
- 主成分分析的目的是求取一个且使用一个 $d \times l$ 的映射矩阵 $W$ 。
- 给定一个样本 $x_i$ ，可将 $x_i$ 从 $d$ 维空间如下映射到 $l$ 维空间： $(x_i)_{1 \times d} (W)_{d \times l}$
- 将所有降维后数据用 $Y$ 表示，有  $Y = X W$

如何求取  
映射矩阵 $W$   
?

降维 原始 映射  
结果 数据 矩阵

- $Y = n \times l$
- $X = n \times d$
- $W = d \times l$

## 主成分分析: 算法描述

$$\mathbf{Y} = n \times l \quad \mathbf{X} = n \times d \quad \mathbf{W} = d \times l$$

降维后 $n$ 个 $l$ 维样本数据 $\mathbf{Y}$ 的方差为:

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

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

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

降维前 $n$ 个 $d$ 维样本数据 $\mathbf{X}$ 的协方差矩阵记为:

$$\Sigma = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$$

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

$$\mathbf{Y} = n \times l \quad \mathbf{X} = n \times d \quad \mathbf{W} = d \times l$$

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

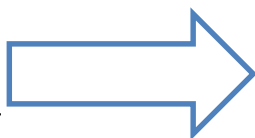
主成份分析求解目标函数为

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

拉格朗日  
函数

满足约束条件

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



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

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

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

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 $\Sigma$ 的一个特征向量， $\lambda_i$ 是这个特征向量所对应的特征值。

保证降维后结果正交以去除相关性（即冗余度）

## 主成分分析: 算法描述

$$\mathbf{Y} = n \times l \quad \mathbf{X} = n \times d \quad \mathbf{W} = d \times l$$

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

- 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 $l$ 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 $x_i$ 的维数是一样的，均为 $d$ 。

## 主成分分析与线性判别分析

**主成分分析 (PCA) 是一种无监督学习的降维方法 (无需样本类别标签)，线性区别分析 (LDA) 是一种监督学习的降维方法 (需要样本类别标签。PCA和LDA均是优化寻找一定特征向量 $w$ 来实现降维，其中PCA寻找投影后数据之间方差最大的投影方向、LDA寻找“类内方差小、类间距离大”投影方向。**

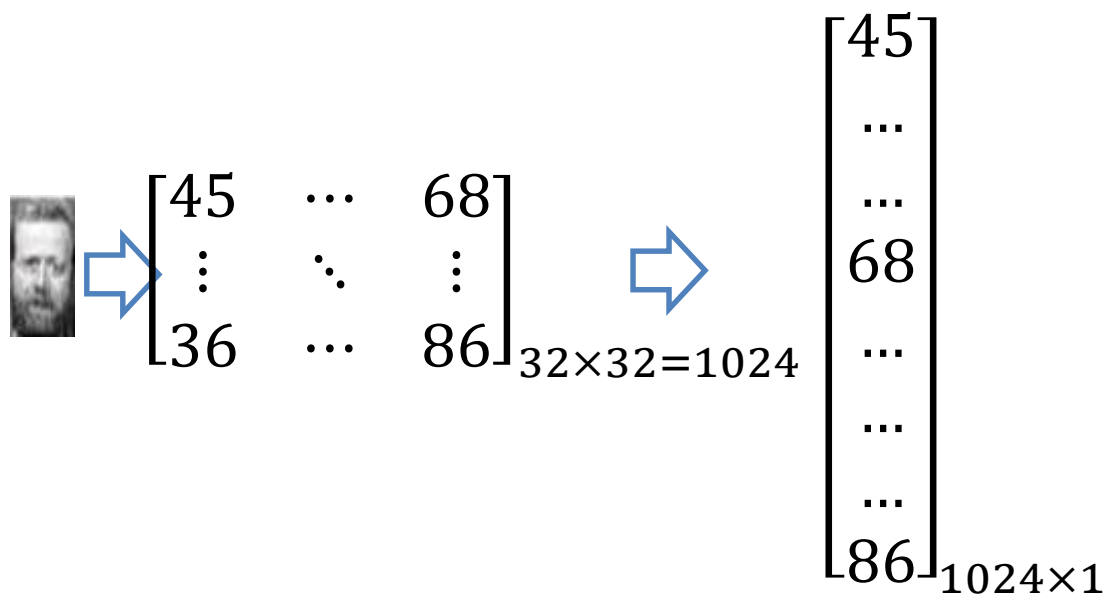
**PCA对高维数据降维后的维数是与原始数据特征维度相关 (与数据类别标签无关)。假设原始数据维度为 $d$ ，那么PCA所得数据的降维维度可以为小于 $d$ 的任意维度。LDA降维后所得到维度是与数据样本的类别个数 $K$ 有关 (与数据本身维度无关)。假设原始数据一共有 $K$ 个类别，那么LDA所得数据的降维维度小于或等于 $K - 1$ 。**

## 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: 算法描述

$$Y = n \times l \quad X = n \times d \quad W = d \times l$$

- 输入:  $n$ 个1024维人脸样本数据所构成的矩阵 $X$ , 降维后的维数 $l$
- 输出: 映射矩阵 $W = \{w_1, w_2, \dots, w_l\}$  (其中每个 $w_j (1 \leq j \leq l)$ 是一个特征人脸)

### ● 算法步骤:

- 1: 对于每个人脸样本数据 $x_i$ 进行中心化处理:  $x_i = x_i - \mu, \mu = \frac{1}{n} \sum_{j=1}^n x_j$
- 2: 计算原始人脸样本数据的协方差矩阵:  $\Sigma = \frac{1}{n-1} X^T X$
- 3: 对协方差矩阵 $\Sigma$ 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 $w_1, w_2, \dots, w_l$ 组成映射矩阵 $W$
- 5: 将每个人脸图像 $x_i$ 按照如下方法降维:  $(x_i)_{1 \times d} (W)_{d \times l} = 1 \times l$

## 特征人脸: 算法描述

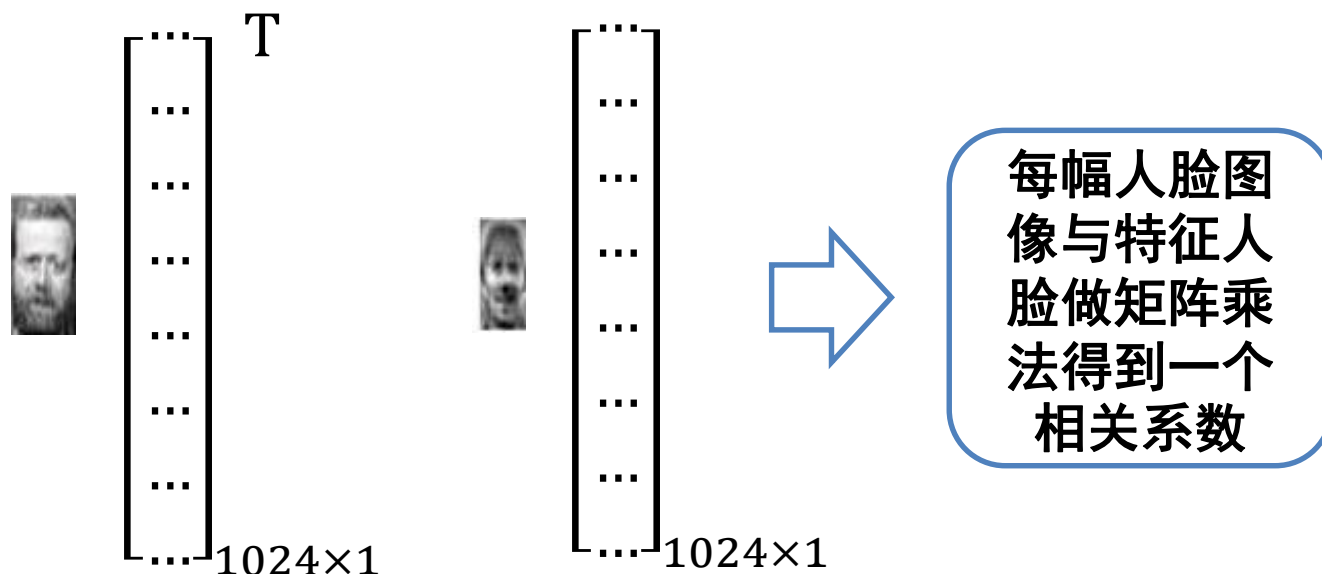
- 每个人脸特征向量 $w_i$ 与原始人脸数据 $x_i$ 的维数是一样的，均为1024。
- 可将每个特征向量还原为 $32 \times 32$ 的人脸图像，称之为特征人脸，因此可得到 $l$ 个特征人脸。



400个人脸（左）和与之对应的36个特征人脸

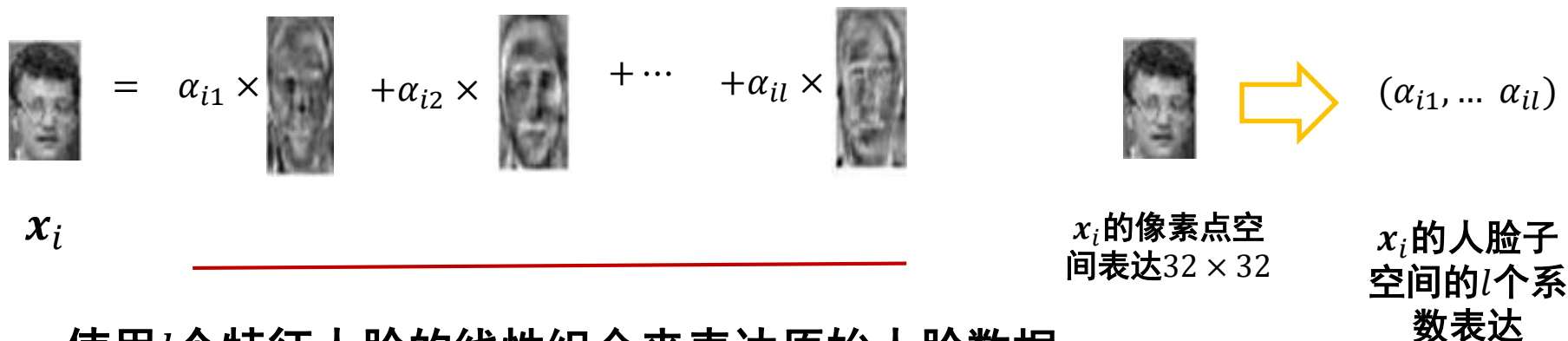
## 基于特征人脸的降维

- 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 $l$ 个相关系数 $\Rightarrow$ 每幅人脸从1024维约减到 $l$ 维



## 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



$x_i$

**聚类表示：**  
用待表示人脸最相似的  
聚类质心来表示



$x_i$

$$x_i = \alpha_{i1} \cdot \text{feature}_1 + \alpha_{i2} \cdot \text{feature}_2 + \cdots + \alpha_{il} \cdot \text{feature}_l$$

**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  $x_i$

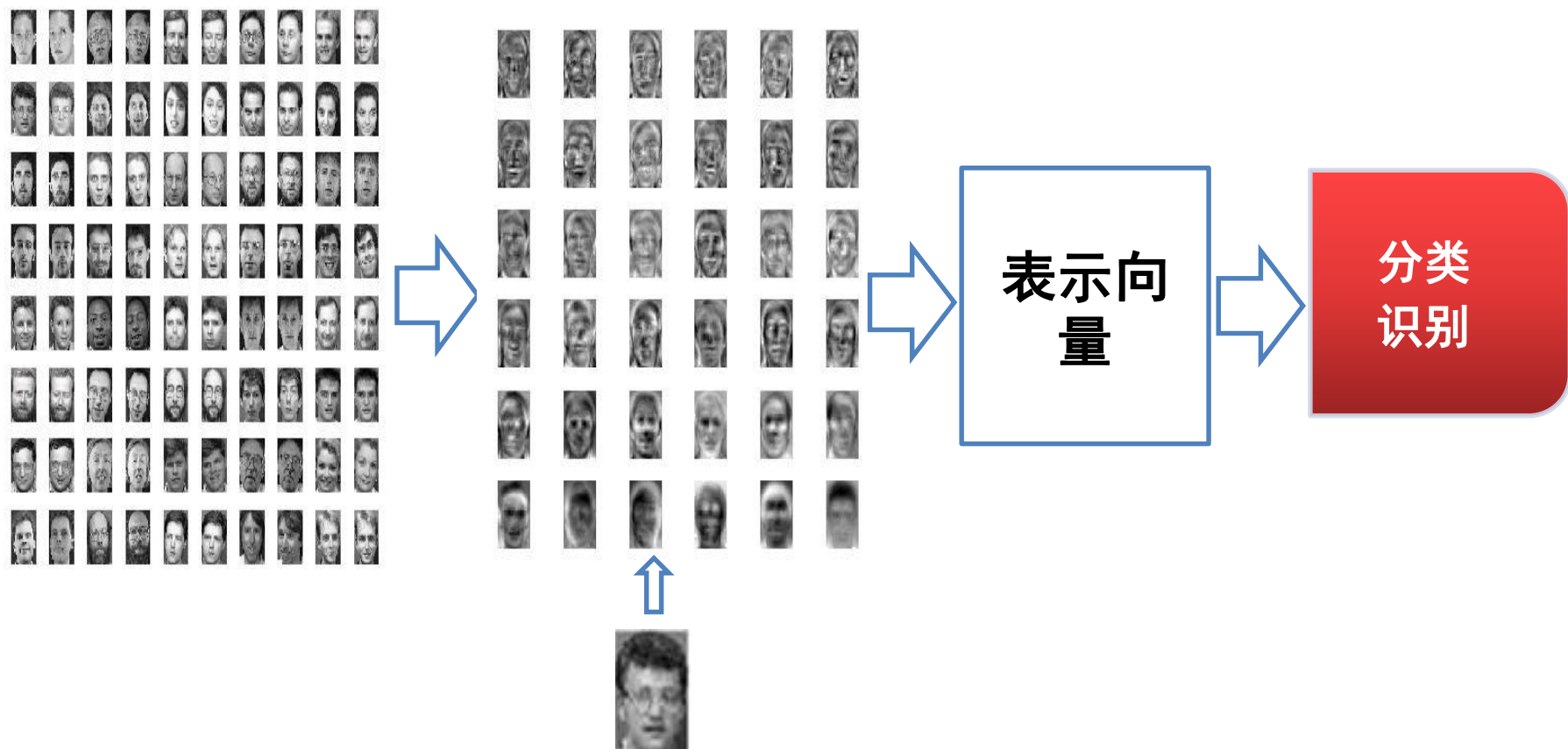


$x_i$



**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”  
Daniel D. Lee & H. Sebastian Seung, Learning the parts of objects by non-negative matrix factorization, 1999, [Nature](#)

# 人脸表达后的分析与处理





# 潜在语义分析

**潜在语义分析 (Latent Semantic Analysis, LSA或者Latent Semantic Indexing, LSI) 是一种从海量文本数据中学习单词-单词、单词-文档以及文档-文档之间隐性关系，进而得到文档和单词表达特征的方法。该方法的基本思想是综合考虑某些单词在哪些文档中同时出现，以此来决定该词语的含义与其他的词语的相似度。**

**潜在语义分析先构建一个单词-文档 (term-document) 矩阵 $A$ ，进而寻找该矩阵的低秩逼近 (low rank approximation)，来挖掘单词-单词、单词-文档以及文档-文档之间的关联关系。**

# 潜在语义分析

单词-文档矩阵(term-document): 构造与分解

通过如下九篇文章来解释LSA方法。选取每篇文章的标题

- a1: Efficient Algorithms for Non-convex Isotonic Regression through Submodular Optimization
- a2: Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search
- a3: An Improved Analysis of Alternating Minimization for Structured Multi-Response Regression
- a4: Analysis of Krylov Subspace Solutions of Regularized Non-Convex Quadratic Problems
- a5: Post: Device Placement with Cross-Entropy Minimization and Proximal Policy Optimization
- b1: CRISPR/Cas9 and TALENs generate heritable mutations for genes involved in small RNA processing of Glycine max and Medicago truncatula
- b2: Generation of D1-1 TALEN isogenic control cell line from Dravet syndrome patient iPSCs using TALEN-mediated editing of the SCN1A gene
- b3: Genome-Scale CRISPR Screening Identifies Novel Human Pluripotent Gene Networks
- b4: CHAMPIONS: A phase 1/2 clinical trial with dose escalation of SB-913 ZFN-mediated in vivo human genome editing for treatment of MPS II (Hunter syndrome)

机器学习 (Machine Learning) 类别五篇文章

基因编辑 (gene editing) 类别四篇文章



# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

从这九篇论文标题中，筛选有实际意义且至少出现在两篇文章标题中的十个单词，分别是nonconvex, regression, optimization, network, analysis, minimization, gene, syndrome, editing, human。这样，十个单词和九篇文章就可以形成一个 $10 \times 9$ 大小的单词-文档矩阵A。

|              | a1 | a2 | a3 | a4 | a5 | b1 | b2 | b3 | b4 |
|--------------|----|----|----|----|----|----|----|----|----|
| nonconvex    | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| regression   | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| optimization | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  |
| network      | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |
| analysis     | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  |
| minimization | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  |
| gene         | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 0  |
| syndrome     | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| editing      | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| human        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |

单词-文档矩阵中每一行表示某个单词在不同文档标题中所出现次数，比如单词regression分别在文档a1和文档a3的标题中各出现了一次，那么这两处相应位置值为1。

# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

|              | a1 | a2 | a3 | a4 | a5 | b1 | b2 | b3 | b4 |
|--------------|----|----|----|----|----|----|----|----|----|
| nonconvex    | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| regression   | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| optimization | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  |
| network      | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |
| analysis     | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  |
| minimization | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  |
| gene         | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 0  |
| syndrome     | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| editing      | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| human        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |

- 当用户输入“optimization”这一检索请求，由于文档a3标题中不包含这一单词，则文档a3被认为是不相关文档，但实际上文档a3所涉及“minimization”内容与优化问题相关。出现这一问题是因为单词-文档矩阵只是刻画了单词是否在文档中出现与否这一现象，而无法对单词-单词、单词-文档以及文档-文档之间语义关系进行建模。
- 如果用户检索“eat an apple”，则文档“Apple is a great company”会被检索出来，而实际上该文档中单词“Apple”所指苹果公司、而非水果，造成这一结果的原因是一些单词具有“一词多义”。
- 因此需要一种方法能够建模单词-单词、单词-文档以及文档-文档之间语义关系，解决包括“异词同义”和“一词多义”在内的诸多挑战。

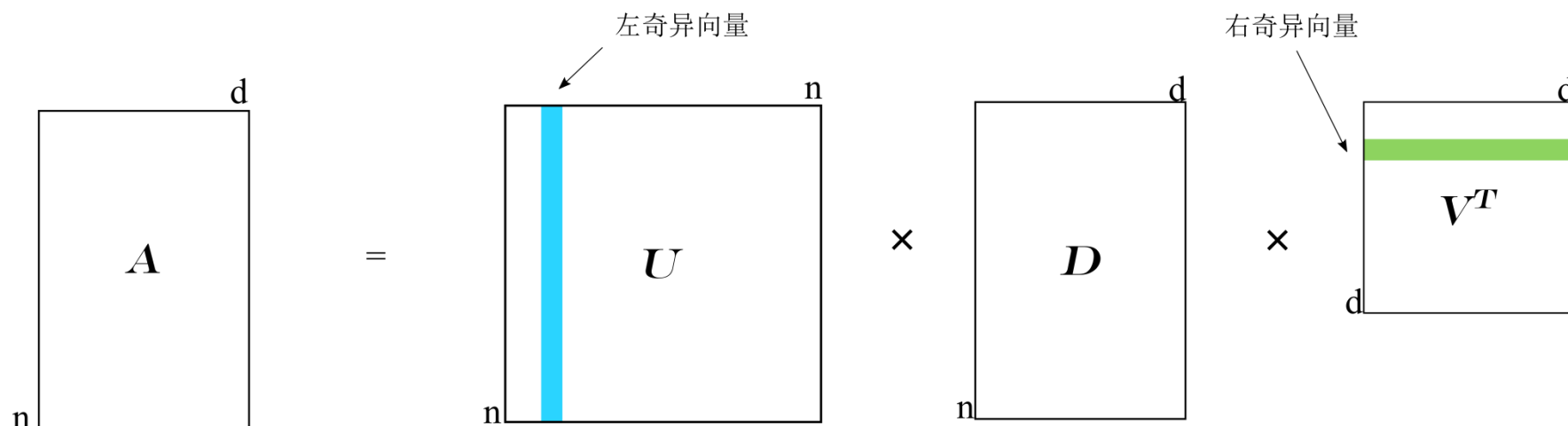
# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

- 奇异值分解(Singular Value Decomposition, SVD)将一个矩阵分解为两个正交矩阵与一个对角矩阵的乘积

$$A = UDV^T$$

- 单词个数为 $M$ 、文档个数为 $N$



# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

| a1 | a2 | a3 | a4 | a5 | b1 | b2 | b3 | b4 |
|----|----|----|----|----|----|----|----|----|
| 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |
| 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |

$$U = \begin{pmatrix} -0.09 & -0.38 & -0.14 & -0.04 & 0.71 & -0.02 & 0.12 & 0.33 & 0.45 & -0. \\ -0.1 & -0.47 & -0.2 & -0.06 & 0. & -0.06 & -0.76 & -0.38 & -0.01 & 0. \\ -0.18 & -0.53 & 0.44 & 0.51 & 0. & -0.14 & 0.16 & 0.09 & -0.42 & -0. \\ -0.28 & -0.08 & 0.57 & -0.23 & -0. & 0.32 & 0.17 & -0.5 & 0.38 & 0. \\ -0.07 & -0.35 & -0.4 & -0.46 & -0. & 0.22 & 0.46 & -0.19 & -0.45 & -0. \\ -0.09 & -0.38 & -0.14 & -0.04 & -0.71 & -0.02 & 0.12 & 0.33 & 0.45 & -0. \\ -0.52 & 0.13 & 0.16 & -0.47 & 0. & -0.65 & -0.04 & 0.16 & -0.11 & -0. \\ -0.45 & 0.16 & -0.32 & 0.34 & -0. & 0.01 & 0.11 & -0.19 & 0.08 & 0.71 \\ -0.45 & 0.16 & -0.32 & 0.34 & -0. & 0.01 & 0.11 & -0.19 & 0.08 & -0.71 \\ -0.42 & 0.1 & 0.09 & -0.11 & -0. & 0.63 & -0.32 & 0.49 & -0.22 & -0. \end{pmatrix}$$

$$D = \begin{pmatrix} 2.46 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 2.35 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 1.79 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1.51 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1.41 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1.23 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0.93 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.73 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.15 \end{pmatrix}$$

$$V^T = \begin{pmatrix} -0.15 & -0.18 & -0.1 & -0.06 & -0.11 & -0.21 & -0.58 & -0.49 & -0.54 \\ -0.59 & -0.26 & -0.51 & -0.31 & -0.39 & 0.06 & 0.19 & 0.07 & 0.18 \\ 0.05 & 0.57 & -0.42 & -0.3 & 0.17 & 0.09 & -0.26 & 0.46 & -0.3 \\ 0.27 & 0.18 & -0.37 & -0.33 & 0.32 & -0.31 & 0.13 & -0.54 & 0.38 \\ 0.5 & 0. & -0.5 & 0.5 & -0.5 & 0. & 0. & 0. & -0. \\ -0.19 & 0.15 & 0.11 & 0.16 & -0.14 & -0.53 & -0.52 & 0.25 & 0.53 \\ -0.51 & 0.36 & -0.19 & 0.62 & 0.3 & -0.04 & 0.21 & -0.2 & -0.1 \\ 0.05 & -0.56 & -0.33 & 0.19 & 0.58 & 0.21 & -0.3 & 0.2 & 0.16 \\ 0.07 & -0.27 & -0.07 & 0. & 0.14 & -0.72 & 0.38 & 0.33 & -0.36 \end{pmatrix}$$

# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

选取最大的前两个特征根及其对应的特征向量对矩阵A进行重建。下面给出了选取矩阵U、矩阵D和矩阵V的子部分重建所得矩阵 $A_2$

|              | a1 | a2 | a3 | a4 | a5 | b1 | b2 | b3 | b4 |
|--------------|----|----|----|----|----|----|----|----|----|
| nonconvex    | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| regression   | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| optimization | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  |
| network      | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |
| analysis     | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  |
| minimization | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  |
| gene         | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 0  |
| syndrome     | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| editing      | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| human        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |

$$A_2 = \begin{pmatrix} \begin{matrix} nonconvex \\ regression \\ optimization \\ network \\ analysis \\ minimization \\ gene \\ syndrome \\ editing \\ human \end{matrix} & \begin{matrix} a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \end{matrix} \\ \begin{matrix} 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ 0.68 & 0.33 & 0.59 & 0.36 & 0.45 & -0.01 & -0.06 & 0.05 & -0.06 \\ 0.79 & 0.4 & 0.68 & 0.41 & 0.53 & 0.02 & 0.02 & 0.14 & 0.02 \\ 0.22 & 0.18 & 0.17 & 0.1 & 0.15 & 0.13 & 0.36 & 0.32 & 0.33 \\ 0.51 & 0.25 & 0.44 & 0.27 & 0.34 & -0.01 & -0.06 & 0.03 & -0.06 \\ 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ 0.01 & 0.16 & -0.03 & -0.02 & 0.02 & 0.29 & 0.81 & 0.66 & 0.75 \\ -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ 0.01 & 0.13 & -0.02 & -0.01 & 0.02 & 0.23 & 0.65 & 0.53 & 0.6 \end{matrix} \end{pmatrix}$$

- 回到之前举的一个例子，用户输入“optimization”来检索与之相关的文档。尽管单词“optimization”在文档a3中没有出现，但是在重建矩阵 $A_2$ 中，对应的位置被0.68取代，说明单词“optimization”对表征文档a3所蕴含内容具有重要作用，这也符合文档a3描述的minimization问题是一个optimization问题的事实。
- 在单词-矩阵A中，文档b3所对应network、gene和human三个单词取值为1，在重建矩阵 $A_2$ 中，network、gene和human三个单词取值分别为0.32、0.66和0.53。可见，network在表征文档b3时重要性降低，因为算法认为这一单词在机器学习所相关文档表达中更具有区别性。

# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

选取最大的前两个特征根及其对应的特征向量对矩阵A进行重建。下面给出了选取矩阵U、矩阵D和矩阵V的子部分重建所得矩阵 $A_2$

$$A_2 = \begin{pmatrix} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ nonconvex & 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ regression & 0.68 & 0.33 & 0.59 & 0.36 & 0.45 & -0.01 & -0.06 & 0.05 & -0.06 \\ optimization & 0.79 & 0.4 & 0.68 & 0.41 & 0.53 & 0.02 & 0.02 & 0.14 & 0.02 \\ network & 0.22 & 0.18 & 0.17 & 0.1 & 0.15 & 0.13 & 0.36 & 0.32 & 0.33 \\ analysis & 0.51 & 0.25 & 0.44 & 0.27 & 0.34 & -0.01 & -0.06 & 0.03 & -0.06 \\ minimization & 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ gene & 0.01 & 0.16 & -0.03 & -0.02 & 0.02 & 0.29 & 0.81 & 0.66 & 0.75 \\ syndrome & -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ editing & -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ human & 0.01 & 0.13 & -0.02 & -0.01 & 0.02 & 0.23 & 0.65 & 0.53 & 0.6 \end{pmatrix}$$

- 由于 $A_2$ 是从最大两个特征根及其对应特征向量重建得到, 因此 $A_2$ 与A不是完全一样的, 两者存在一定的误差
- $A_2$ 捕获得到了原始单词-文档矩阵A中所蕴含的单词与单词之间的关联关系
- 如果两个单词在原始单词-文档矩阵A中分布一致, 则其在重建矩阵 $A_2$ 中分布也可能一致的, 如editing和syndrome。
- 对于归属于同一类别文档的单词, 可以发现它们之间的值彼此接近, 而与不是归属于同一个类别中的单词不相似, 如minimization在机器学习类别文档中均为正数、其在基因编辑类别文档中几乎为负数。

# 潜在语义分析

## 单词-文档矩阵(term-document): 构造与分解

基于单词-文档矩阵 $A$ , 对 $A^T A$ 结果归一化可得到如下文档-文档相关系数矩阵:

$$\begin{pmatrix} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ a1 & 1. & 0.22 & 0.05 & 0.22 & 0.22 & -0.22 & -0.43 & -0.43 & -0.43 \\ a2 & 0.22 & 1. & -0.33 & -0.25 & 0.37 & -0.17 & -0.33 & 0.22 & -0.33 \\ a3 & 0.05 & -0.33 & 1. & 0.22 & 0.22 & -0.22 & -0.43 & -0.43 & -0.43 \\ a4 & 0.22 & -0.25 & 0.22 & 1. & -0.25 & -0.17 & -0.33 & -0.33 & -0.33 \\ a5 & 0.22 & 0.37 & 0.22 & -0.25 & 1. & -0.17 & -0.33 & -0.33 & -0.33 \\ b1 & -0.22 & -0.17 & -0.22 & -0.17 & -0.17 & 1. & 0.51 & 0.51 & -0.22 \\ b2 & -0.43 & -0.33 & -0.43 & -0.33 & -0.33 & 0.51 & 1. & 0.05 & 0.52 \\ b3 & -0.43 & 0.22 & -0.43 & -0.33 & -0.33 & 0.51 & 0.05 & 1. & 0.05 \\ b4 & -0.43 & -0.33 & -0.43 & -0.33 & -0.33 & -0.22 & 0.52 & 0.05 & 1. \end{pmatrix}$$

基于重建单词-文档矩阵 $A_2$ , 对 $A_2^T A_2$ 结果归一化可得到如下文档-文档相关系数矩阵:

$$\begin{pmatrix} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ a1 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.95 & -0.93 & -0.95 \\ a2 & 0.97 & 1. & 0.97 & 0.97 & 0.98 & -0.85 & -0.86 & -0.83 & -0.86 \\ a3 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.96 & -0.94 & -0.96 \\ a4 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.96 & -0.94 & -0.96 \\ a5 & 1. & 0.98 & 1. & 1. & 1. & -0.94 & -0.95 & -0.93 & -0.95 \\ b1 & -0.95 & -0.85 & -0.95 & -0.95 & -0.94 & 1. & 1. & 1. & 1. \\ b2 & -0.95 & -0.86 & -0.96 & -0.96 & -0.95 & 1. & 1. & 1. & 1. \\ b3 & -0.93 & -0.83 & -0.94 & -0.94 & -0.93 & 1. & 1. & 1. & 1. \\ b4 & -0.95 & -0.86 & -0.96 & -0.96 & -0.95 & 1. & 1. & 1. & 1. \end{pmatrix}$$

- 在基于单词-文档矩阵 $A$ 所构建的文档-文档相互关系矩阵中, 同一类别文档之间的相关性系数普遍很低, 比如 $a1-a3$ ,  $a2-a3$ ,  $a2-c4$ 等。比如, 机器学习类别所包含五篇文档的平均相关性仅为0.069, 基因编辑类别所包含四篇文档的平均相关性为0.237。
- 在基于重建单词-文档矩阵 $A_2$ 所构建的文档-文档相关系数矩阵中, 由于经过隐性语义分析, 可以看到相关性矩阵中数值已可以相当反映不同文档所蕴含语义关系。机器学习类别所包含五篇文档的平均相关性上升到0.989, 基因编辑类别所包含四篇文档的平均相关性上升到1.00。机器类别所包含文档与基因编辑类别所包含文档之间的平均相关性为-0.927。



# 期望最大化算法

- 假设由 $n$ 个数据样本构成的集合 $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$ 从参数为 $\theta$ 的某个模型（如高斯模型等）以一定概率独立采样得到。于是，可以通过最大似然估计算法（maximum likelihood estimation, MLE）来求取参数 $\theta$ ，使得在参数为 $\theta$ 的模型下数据集 $\mathcal{D}$ 出现的可能性最大，即 $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(\mathcal{D}|\theta)$ 。
- 或者也可利用最大后验估计（maximum a posteriori estimation, MAP）从数据集 $\mathcal{D}$ 来如下估计参数 $\theta$ ： $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(\theta|\mathcal{D}) = \underset{\theta}{\operatorname{argmax}} \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$ 。由于 $P(\mathcal{D})$ 与 $\theta$ 无关，则可得 $\underset{\theta}{\operatorname{argmax}} \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} = \underset{\theta}{\operatorname{argmax}} P(\mathcal{D}|\theta)P(\theta)$ ，对这个式子取对数，得到 $\underset{\theta}{\operatorname{argmax}} \log P(\mathcal{D}|\theta) + \log P(\theta)$ 。可见，最大后验估计与最大似然估计相比，增加了一项与 $\theta$ 相关的先验概率 $P(\theta)$ 。
- 当然，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。
- 但是，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：二硬币投掷例子

问题：求硬币A或硬币B被投掷为正面的概率 $\Theta = \{\theta_A, \theta_B\}$ 。

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 |

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。

注意点：

- 所求取得到的 $\theta$ 值应该以最大概率（极大似然）“拟合”表5.3观测结果。
- 表5中所列观测结果是不完全数据，即从观测结果只能够知道硬币正面或反面，但是并不知道这个正面或反面来自哪一个硬币，因此每一轮中所选择投掷的硬币是一个“隐变量”。
- 由于这个问题中存在隐变量，因此下面采用EM算法来计算硬币A或硬币B被投掷为正面的概率。

## 期望最大化算法：二硬币投掷例子

问题：求硬币A或硬币B被投掷为正面的概率 $\Theta = \{\theta_A, \theta_B\}$ 。

### 求取期望 (E步骤, Expectation)

初始化每一轮中硬币A和硬币B投掷为正面的概率为 $\hat{\theta}_A^{(0)} = 0.60$ 和 $\hat{\theta}_B^{(0)} = 0.50$ 。

在第一轮试验中，所观测得到的投掷结果为“HTTTHHTHTH”，这10次投掷结果由硬币A投掷所得概率为：

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

这10次结果由硬币B投掷所得概率为：

## 期望最大化算法：二硬币投掷例子

问题：求硬币A或硬币B被投掷为正面的概率 $\Theta = \{\theta_A, \theta_B\}$ 。

### 求取期望 (E步骤, Expectation)

一旦得到了在第一轮试验中选择硬币A或硬币B投掷的概率，则可以根据这个概率来计算第一轮中硬币A和硬币B投掷正面的期望次数。比如，硬币A为正面期望次数为 $N_{\text{正面总次数}} \times P_{\text{选硬币A概率}} = 5 \times 0.45 = 2.25$ ，其他项计算类似。

同理，可以计算得到其他四轮试验中硬币A和硬币B投掷正面的次数，以及五轮试验中硬币A或硬币B投掷为正面的总次数

| 轮次 | 选硬币A<br>概率 | 选硬币B<br>概率 | 硬币A为正<br>面期望次数 | 硬币A为反<br>面期望次数 | 硬币B为正<br>面期望次数 | 硬币B为反<br>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           |

## 期望最大化算法：二硬币投掷例子

问题：求硬币A或硬币B被投掷为正面的概率 $\Theta = \{\theta_A, \theta_B\}$ 。

### 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 $\hat{\theta}_A^{(0)}$ 和 $\hat{\theta}_B^{(0)}$ ，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

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

- 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

**问题：求硬币A或硬币B被投掷为正面的概率 $\Theta = \{\theta_A, \theta_B\}$ 。**

**表5.5列出了十次迭代过程中硬币A和硬币B投掷为正面的概率。在第十次迭代时，算法收敛，得到 $\theta_A$ 和 $\theta_B$ 的值分别为0.797和0.520。**

| 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    | $\hat{\theta}_A^{(10)} = 0.797$ | $\hat{\theta}_B^{(10)} = 0.520$ |