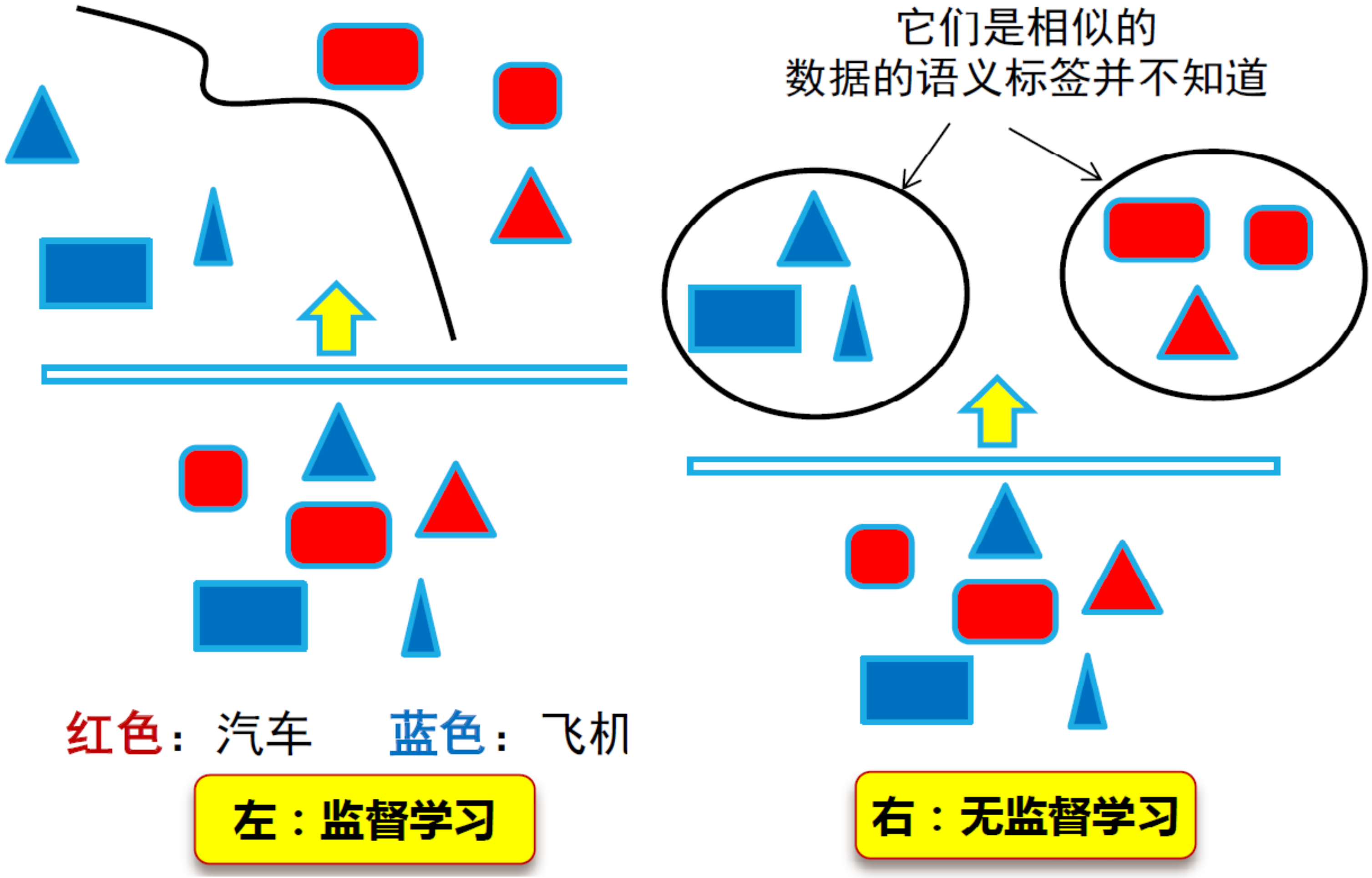
聚类和降维

Clustering and Dimension Reduction

聚类

- 聚类：对大量未知标注的数据集，按数据的内在相似性将数据集划分为多个类别，使类别内的数据相似度较大而类别间的数据相似度较小
  - Note：子集通常是不现交的；每个子集称为“簇”
- 假定样本集$D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$包含$m$个无标记样本。每个样本$\mathbf{x}_i = \{x_{i1,}, x_{i2}, \ldots, x_{in}\}^T$是一个$n$维向量。
- 聚类将样本集D划分为k个不相交的簇$\{C_i | i = 1, 2, \ldots, k\}$,其中$C_i \cap_{i \neq j} C_j = \emptyset$且$D = \cup_{i=1}^{k} C_i$.
- 用$\lambda_j \in \{1, 2, \ldots, k\}$表示簇标记，聚类结果可表示为：

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$$


它们是相似的
数据的语义标签并不知道
红色：汽车
蓝色：飞机
左：监督学习
右：无监督学习

- 数据特征
  - 图像中的颜色、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
- 距离：$d(\cdot,\cdot)$
  - 非负性：$d(\mathbf{x},\mathbf{y}) \geq 0$
  - 正定性：$d(\mathbf{x},\mathbf{y}) = 0$当且仅当$\mathbf{x} = \mathbf{y}$
  - 对称性：$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$
  - 三角不等式：$d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})$
- 余弦相似度(cosine similarity)：$\cos\theta = \frac{a^T b}{|a|\cdot|b|}$
- Person相似系数：$\rho_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$
- 杰卡德相似系数(Jaccard)：$J(A,B) = \frac{|A\cap B|}{|A\cup B|}$

- 闵可夫斯基距离：$dist(X,Y)=(\sum_{i=1}^{n}|x_i-y_i|^p)^{\frac{1}{p}}$
  - $p=2$时，为欧式距离。
  - $p=1$时，为曼哈顿距离。
- 相对熵(K-L距离) $D(p||q)=\sum_x p(x)\log\frac{p(x)}{q(x)}=E_{p(x)}\log\frac{p(x)}{q(x)}$
- Hellinger距离 $D_a(p||q)=\frac{2}{1-a^2}(1-\int p(x)^{\frac{1+\alpha}{2}}q(x)^{\frac{1-\alpha}{2}}dx$

K均值聚类(K-means聚类)

- 输入: m个数据
- 输出：k个聚类结果
- 目的：将m个数据聚类到k个集合
- 基本思想：首先给出初始划分，通过迭代改变样本和簇的隶属关系，使得每一次改进之后的划分方案都比前一次好。

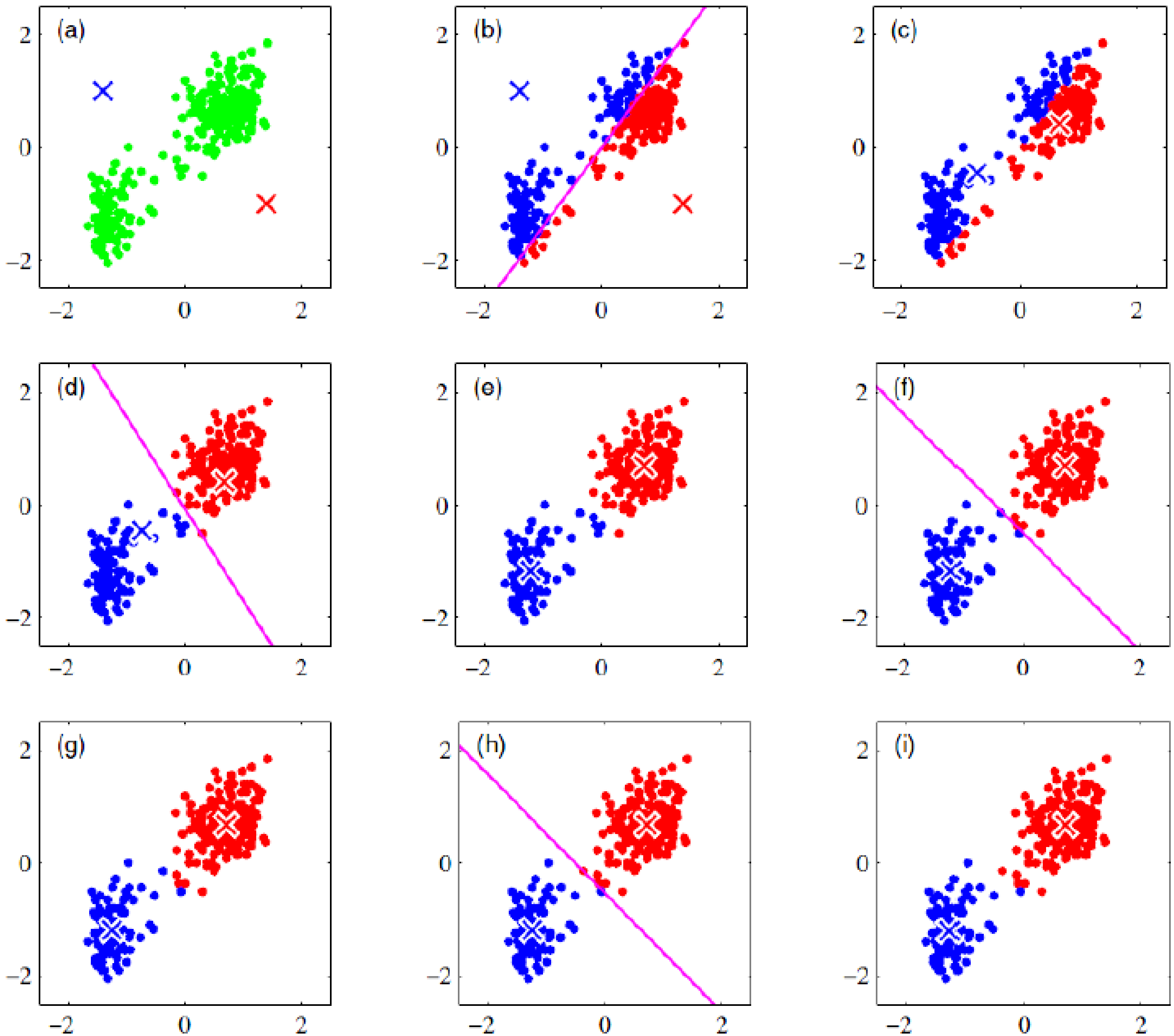
- Step 1. 初始化聚类质心
  - 初始化k个聚类质心$\mathbf{c} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}, \mathbf{c}_j \in \mathbb{R}^n (1 \leq j \leq k)$
  - 每个聚类质心$\mathbf{c}_j$所在集合记为G_j
- Step 2. 将每个待聚类数据放入唯一一个聚类集合中
  - 计算待聚类数据$\mathbf{x}_i$和质心$\mathbf{c}_j$之间的欧氏距离$d(\mathbf{x}_i, \mathbf{c}_j)(1 \leq i \leq m, 1 \leq j \leq k)$
  - 将每个$\mathbf{x}_i$放入与之距离最近质心所在聚类集合中，即:

$$\arg\min_{\mathbf{c}_j \in C} d(\mathbf{x}_i, \mathbf{c}_j)$$

- Step 3. 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：$\mathbf{c}_j = \frac{1}{|G_j|}\sum_{\mathbf{x}_i \in G_j} \mathbf{x}_i$
- Step 4. 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


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

- 优点
  - 是解决聚类问题的一种经典算法，简单、快速
  - 对处理大数据集，该算法保持可伸缩性和高效率
  - 当簇近似为高斯分布时，它的效果较好
- 缺点
  - 在簇的平均值可被定义的情况下才能使用
  - 需要事先确定聚类数目，且对初值敏感
  - 算法是迭代执行，时间开销非常大
  - 对噪声和孤立点数据敏感
  - 不适合于发现非凸形状的簇或者大小差别很大的簇。
  - 欧氏距离假设数据每个维度之间的重要性是一样的

K近邻法(K-Nearest Neighbors，KNN)

- 给定一个训练数据集，对新的输入实例，在训练数据集中找到与实例最近邻的k个实例，基于这些邻居预测
  - 分类任务：投票法---k个样本中最多的类别为预测结果
  - 回归任务：平均法---平均值或加权平均值

KNN是懒惰学习的代表，训练开销为零，待收到测试样本再进行处理。
急切学习：在训练阶段对样本进行学习的方式。

给定测试样本$\mathbf{x}$，若其最近邻样本为$\mathbf{z}$,则最近邻分类器出错的概率是$\mathbf{x}$与$\mathbf{z}$类别标记不同的概率，即$P(err) = 1 - \sum_{c \in Y} P(c|\mathbf{x})P(c|\mathbf{z})$

假设样本独立同分布，令$c^* = \arg\max_{c \in Y} P(c|\mathbf{x})$表示在贝叶斯方法的最优分类，则：

$$
\begin{aligned}
P(err) &= 1 - \sum_{c \in Y} P(c|\mathbf{x})P(c|\mathbf{z}) \\
&\approx 1 - \sum_{c \in Y} P^2(c|\mathbf{x}) \\
&\leq 1 - P^2(c^*|\mathbf{x}) \\
&= (1 + P(c^*|\mathbf{x}))(1 - P(c^*|\mathbf{x})) \\
&\leq 2 \times (1 - P(c^*|\mathbf{x}))
\end{aligned}
$$

最近邻分类器虽简单，但他的泛化错误率不超过贝叶斯分类器的错误率的两倍。

降维

- 维数灾难：在高维情形下出现的数据样本稀疏，距离计算困难等问题，是所有机器学习方法共同面临的困难
- 降维：如果我们有一组N维向量，现在要将其降到K维(K小于N)，那么我们应该如何选择K个基才能最大程度保留原有的信息。
  - 直接降维：特征选择
  - 线性降维：PCA
  - 非线性降维
- 优点：
  - 降低资源需求
  - 去除噪声
  - 增强可解释性

- 主成分分析(Principal Component Analysis, PCA)
  - 线性降维
  - 子空间方法

1. $K = 0$

- 给定数据集D,寻找1个点m^*,使得其到D中所有元素的距离总和很小，即

$$m^* = \arg\min_{m} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - m\|^2$$

利用最优性条件可得: $m^* = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i = \overline{\mathbf{x}}$

2. $K = 1$

- 一维子空间的点($\mathbf{x} - \overline{\mathbf{x}}$)都可以表示为$a\mathbf{w}$, 利用这一表示，原数据集的任一点$\mathbf{x}_i$可以近似表示为$\mathbf{x}_i \approx \overline{\mathbf{x}} + a_i\mathbf{w}$.

记 $\mathbf{a} = (a_1, a_2, \ldots, a_N)^T$

- 定义目标J来最小化平均距离：

$$J(\mathbf{w}, \mathbf{a}) \quad = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - (\overline{\mathbf{x}} + a_i\mathbf{w})\|^2$$

注意到$J(\mathbf{w}, \mathbf{a}) = J(c\mathbf{w}, \frac{1}{c}\mathbf{a})$

这样考虑最优化问题：$\min\limits_{\|\mathbf{w}\|=1} J(\mathbf{w}, \mathbf{a})$

$$
\begin{aligned}
\min_{\|\mathbf{w}\|=1} J(\mathbf{w},\mathbf{a}) &= \min_{\|\mathbf{w}\|=1} \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - (\overline{\mathbf{x}} + a_i\mathbf{w})\|^2 \\
&= \min_{\|\mathbf{w}\|=1} \frac{1}{N}\sum_{i=1}^{N} \|a_i\mathbf{w} - \ (\mathbf{x}_i - \overline{\mathbf{x}})\ \|^2 \\
&= \min_{\|\mathbf{w}\|=1} \sum_{i=1}^{N} \frac{a_i^2\|\mathbf{w}\|^2 + \|\mathbf{x}_i - \mathbf{x}\|^2 - 2a_i\mathbf{w}^T(\mathbf{x}_i - \overline{\mathbf{x}})}{N} \\
&= \min_{\|\mathbf{w}\|=1} \sum_{i=1}^{N}(a_i^2 - 2a_i\mathbf{w}^T(\mathbf{x}_i - \overline{\mathbf{x}})) \\
&= \min_{\|\mathbf{w}\|=1} J
\end{aligned}
$$

利用最优性条件：(1) $\frac{\partial J}{\partial a_i} = 2a_i - 2\mathbf{w}^T(\mathbf{x}_i - \overline{\mathbf{x}}) = 0$

得到: $a_i = (\mathbf{x}_i - \overline{\mathbf{x}}_i)^T\mathbf{w}$

(2) $\frac{\partial(J+\lambda(\mathbf{w}^T\mathbf{w}-1))}{\partial\mathbf{w}} = \sum_{i=1}^{N} -2a_i(\mathbf{x}_i - \overline{\mathbf{x}}) + 2\lambda w = 0$

得到：$\lambda\mathbf{w} = \sum_{i=1}^{N} a_i(\mathbf{x}_i - \overline{\mathbf{x}})$,

即$\lambda = \sum_{i=1}^{N} a_i(\mathbf{x}_i - \overline{\mathbf{x}})^T\mathbf{w} = \sum_{i=1}^{N} a_i^2$

$$
\begin{aligned}
\lambda \mathbf{w} &= \sum_{i=1}^{N} a_i (\mathbf{x}_i - \overline{\mathbf{x}}) \\
\Rightarrow \quad \frac{1}{N}\lambda \mathbf{w} &= \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}}) a_i \\
&= \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}}_i)^T \mathbf{w} \\
&= \mathbf{Cov}(\mathbf{x})\mathbf{w}
\end{aligned}
$$

重新写作：$\mathbf{Cov}(\mathbf{x})\mathbf{w} = \bar{\lambda}\mathbf{w}$

注意到:

$$
\begin{aligned}
\min_{\|\mathbf{w}\|=1} J &= \min_{\|\mathbf{w}\|=1} \sum_{i=1}^{N} (a_i^2 - 2a_i \mathbf{w}^T (\mathbf{x}_i - \overline{\mathbf{x}})) \\
&= \max_{\|\mathbf{w}\|=1} \sum_{i=1}^{N} a_i^2 \\
&= \max_{\|\mathbf{w}\|=1} \lambda \\
&= \max_{\|\mathbf{w}\|=1} \overline{\lambda}
\end{aligned}
$$

$a_i = (\mathbf{x}_i - \overline{\mathbf{x}}_i)^T \mathbf{w}$

3. $K > 1$

考虑到协方差矩阵是半正定的实对称矩阵，则其一定可以对角化。设其有D个特征向量$\xi_1, \xi_2, ..., \xi_D$, 与之对应的特征值分别为$\lambda_1, \lambda_2, ..., \lambda_D$,它们都是实数且满足$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D \geq 0$,则：

$$\mathbf{Cov}(\mathbf{x}) = \sum_{i=1}^{D} \lambda_i \xi_i \xi_i^T$$

构造$D \times D$的矩阵E,其第i列由ξ_i构成，可得$EE^T = E^T E = I$
那么我们有：

$$\begin{aligned}\mathbf{x} &= \overline{\mathbf{x}} + (\mathbf{x} - \overline{\mathbf{x}}) \\ &= \overline{\mathbf{x}} + EE^T(\mathbf{x} - \overline{\mathbf{x}}) \\ &= \overline{\mathbf{x}} + (\xi_1^T(\mathbf{x} - \overline{\mathbf{x}}))\xi_1 + (\xi_2^T(\mathbf{x} - \overline{\mathbf{x}}))\xi_2 + ... + (\xi_D^T(\mathbf{x} - \overline{\mathbf{x}}))\xi_D\end{aligned}$$

- PCA算法

1. 输入：一个D维训练集$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$和一个新的(更低的)维度$d(d < D)$.

2.计算均值： $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$

3.计算协方差矩阵：$\mathbf{Cov} = \frac{1}{N}\sum\limits_{i=1}^{N}(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T$

4.找到$\mathbf{Cov}(\mathbf{x})$的谱分解，得到特征向量$\xi_1, \xi_2, \ldots, \xi_D$, 及其对应的特征值分别为$\lambda_1, \lambda_2, \ldots, \lambda_D$，使得$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D \geq 0.$

5.对于任一$\mathbf{x} \in \mathbb{R}^D$, 其新的低维表示为

$$y = (\xi_1^T(\mathbf{x} - \overline{\mathbf{x}})\xi_1, \xi_2^T(\mathbf{x} - \overline{\mathbf{x}})\xi_2, \ldots, \xi_d^T(\mathbf{x} - \overline{\mathbf{x}})\xi_d)^T \in \mathbb{R}^d$$

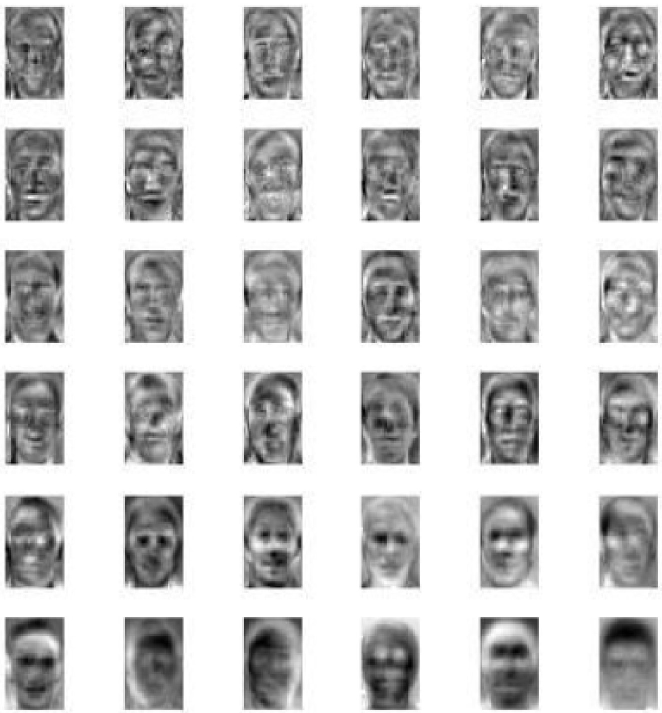
特征人脸

特征人脸方法是一种应用主成分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。

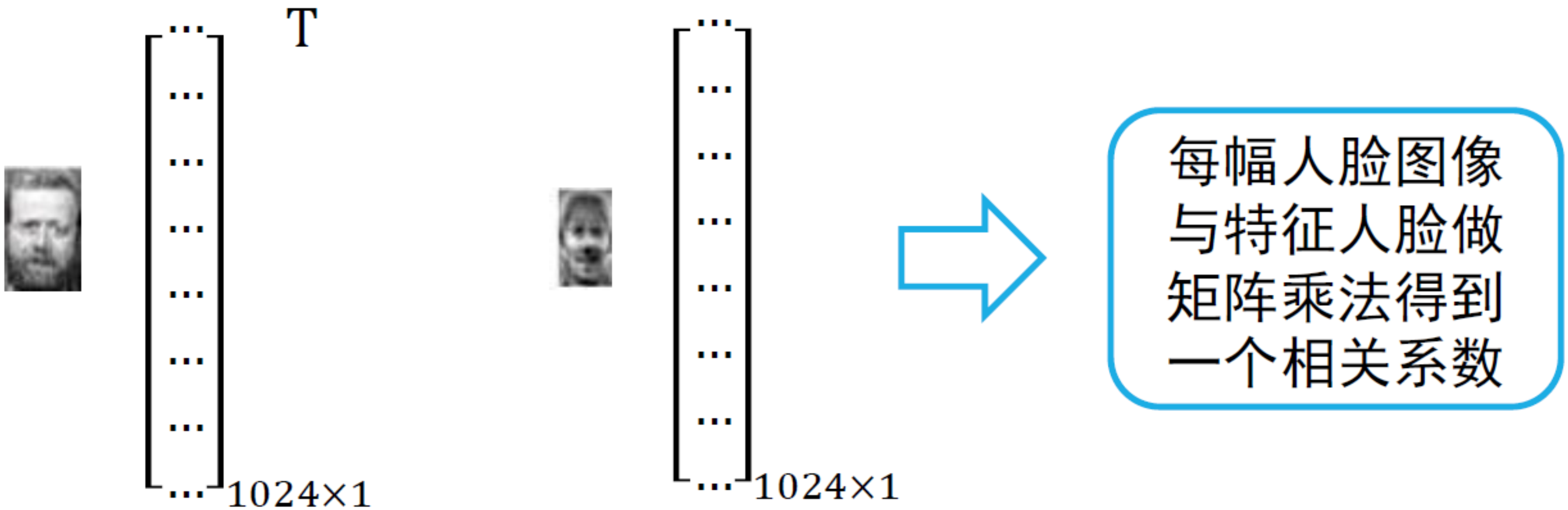
- 将每幅人脸图像转换成列向量
- 如将一幅32×32的人脸图像转成1024×1的列向量

$$\begin{bmatrix} 45 & \cdots & 68 \\ \vdots & \ddots & \vdots \\ 36 & \cdots & 86 \end{bmatrix}_{32\times 32=1024} \Rightarrow \begin{bmatrix} 45 \\ \cdots \\ \cdots \\ 68 \\ \cdots \\ \cdots \\ \cdots \\ 86 \end{bmatrix}_{1024\times 1}$$

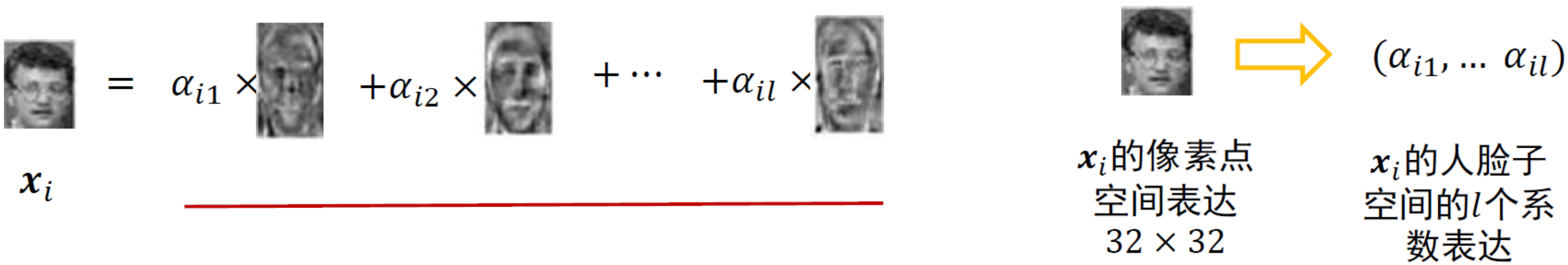
- 每个人脸特征向量$\mathbf{w}_i$与原始人脸数据$\mathbf{x}_i$的维数是一样的, 均为1024。
- 可将每个特征向量还原为32×32的人脸图像，称之为特征人脸，因此可得到l个特征人脸。
- 400个人脸和36个特征人脸

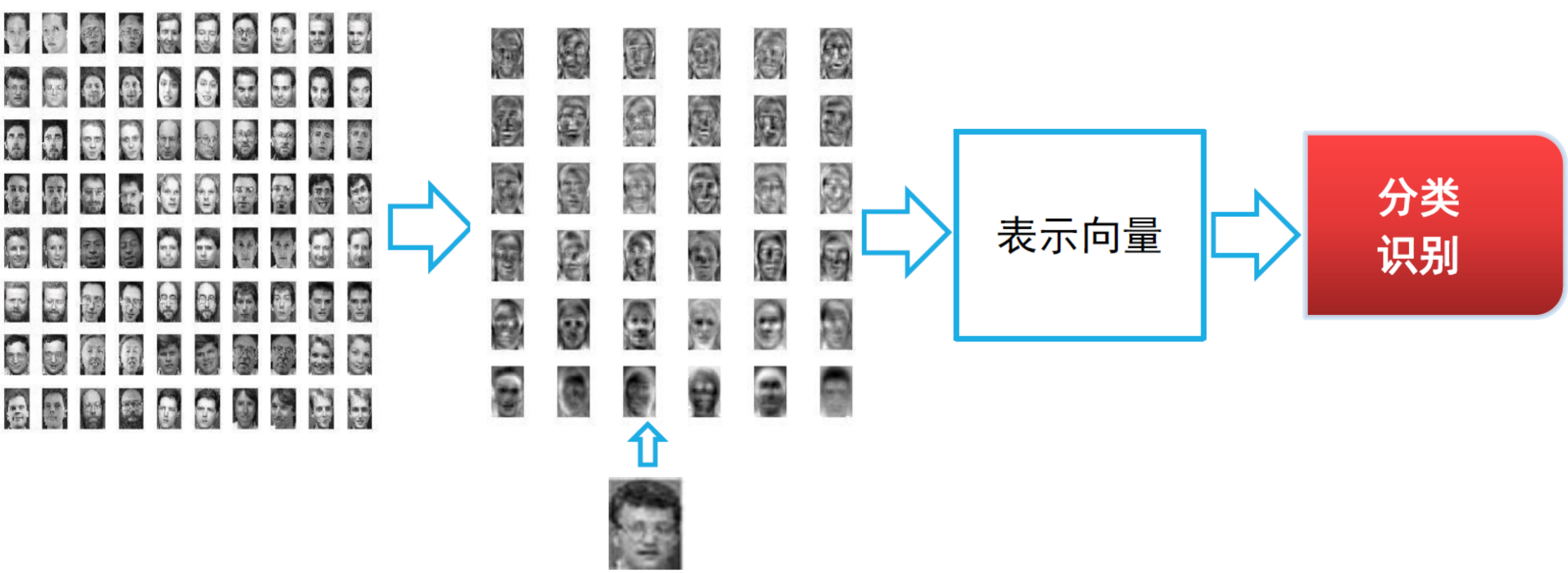


- 将每幅人脸分别与每个特征人脸做矩阵乘法，得到一个相关系数
- 每幅人脸得到l个相关系数⇒每幅人脸从1024维约减到l维



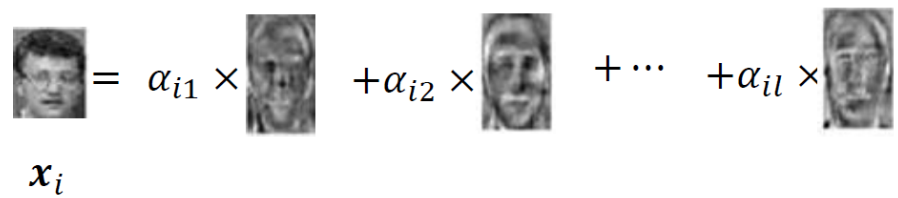
- 由于每幅人脸是所有特征人脸的线性组合，因此就实现人脸从“像素点表达”到“特征人脸表达”的转变。每幅人脸从1024维约减到l维
- 使用l个特征人脸的线性组合来表达原始人脸数据$\mathbf{x}_i$





- 人脸表达的方法对比：聚类、主成分分析、非负矩阵分解

聚类表示：
用待表示人脸最相似的聚类质心来表示



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$\boldsymbol{x}_i$

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**