AI300学院

机器学习之无监督学习

数据降维算法

倪冰冰
上海交通大学

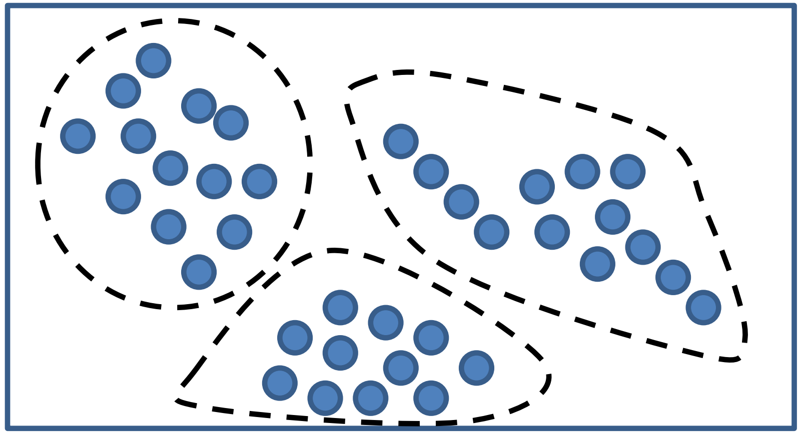
引言

- 什么是无监督学习？

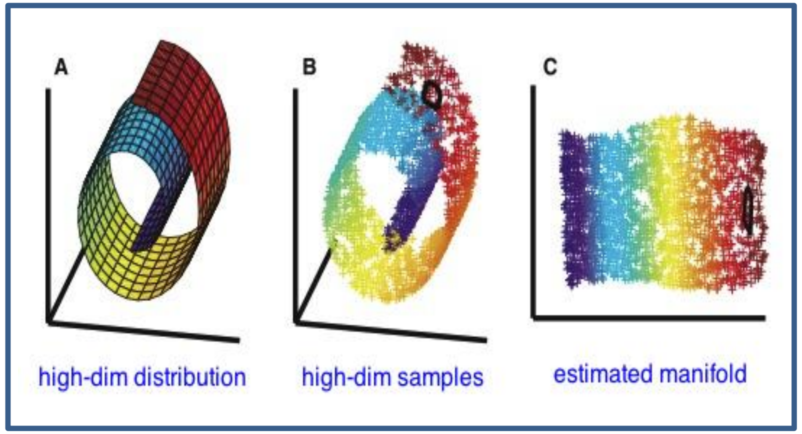
　　1.数据没有明确的标签信息。

　　2.我们希望仅依赖数据本身来探索其具有的内在结构信息。

- 无监督学习的种类有哪些？



聚类学习



表征学习(降维)

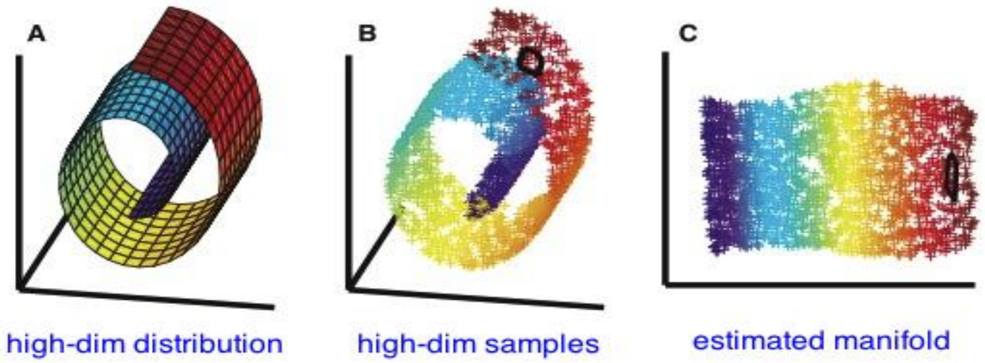
学习内容

- 什么是数据降维
- 线性降维方法: PCA & MDS
- 非线性降维方法: ISOMAP & LLE

数据降维

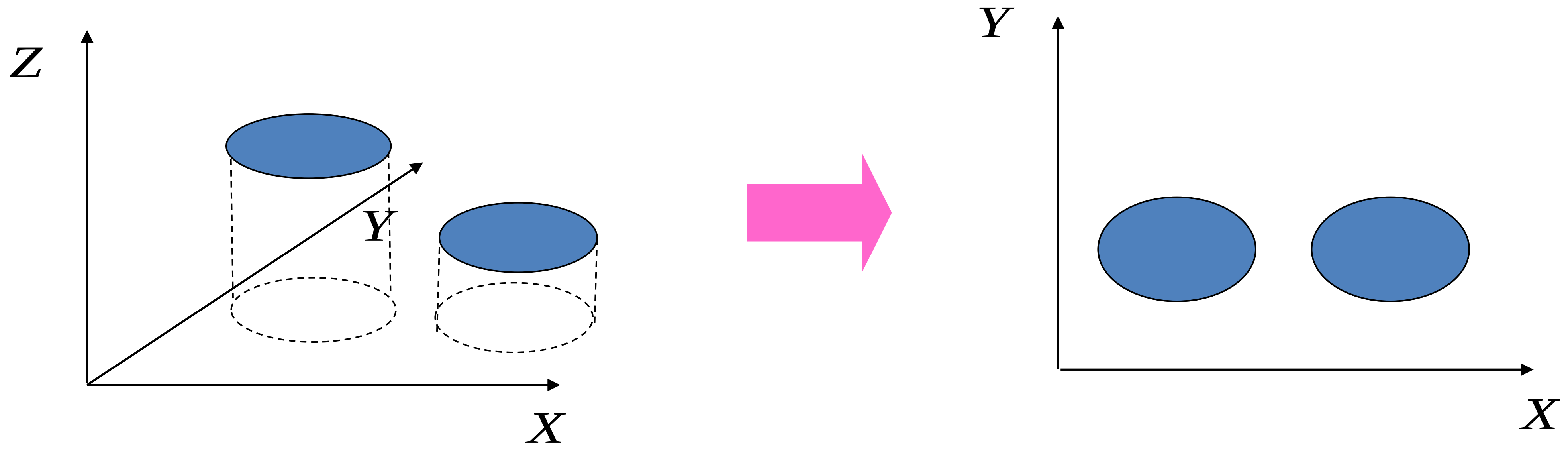
- 问题定义
  大多数的数据在日常生产中都是高维度数据：如图像、声音等
  但是他们通常可以被低维度特征向量所表示：如子空间、流形

- 解题目标
  将高维数据映射到低维空间中，并保证数据结构损失最小



数据降维

- 3D-2D示例



数据降维

- 线性方法
  - PCA (Principle Component Analysis)
    - 保留Variance
  - MDS (Multi Dimensional Scaling)
    - 保留内部点间距
- 非线性方法
  - ISOMAP
  - LLE

Principal Component Analysis

复习概率分布知识：

Expectation is the mean (average) of random variable x :

$$E[x] = \int x\, p(x) dx$$

Variance is the expected squared difference from mean m :

$$Var[x] = E[(x - m)^2] = E[x^2] - (E(x))^2$$

For vectors,

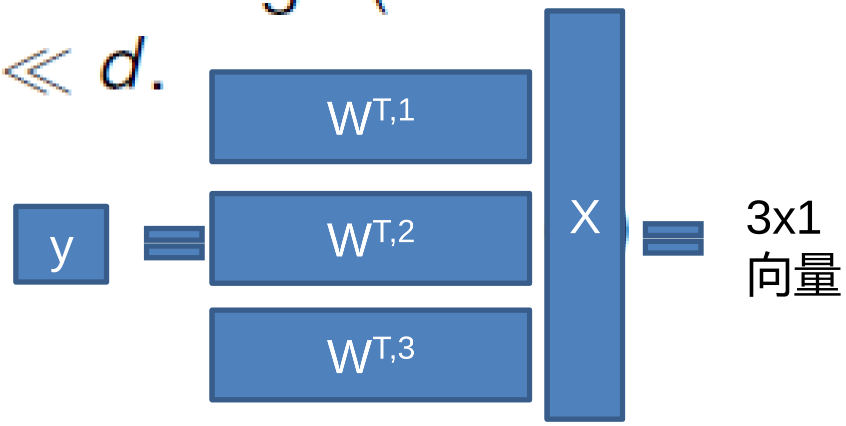
$$\text{Mean: } \mathbf{m} = E[\mathbf{x}] = [E[x_1]\ E[x_2]\ \cdots E[x_d]]^\top$$

$$\text{Covariance matrix: } \quad Var[\mathbf{x}] = E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^\top] = E[\mathbf{x}\mathbf{x}^\top] - \mathbf{m}\mathbf{m}^\top$$

Principal Component Analysis

- a.k.a. Discrete Karhunen Loeve Transform, Hotelling Transform
- Let $\mathbf{y} \in \mathbb{R}^k$ be feature vector computed from image (or another feature vector) $\mathbf{x} \in \mathbb{R}^d$, where $k \ll d$.

$$\mathbf{y} = \mathbf{W}^\top \mathbf{x}$$



- $\mathbf{W}$ is $d \times k$ and orthogonal.
  - $\mathbf{W}$ to be determined from statistics of $\mathbf{x}$.
  - Let's suppose mean and covariance matrix are: $E[\mathbf{x}] = \mathbf{0}, \quad E[\mathbf{x}\mathbf{x}^\top] = \mathbf{C}_x$

W的每个列是正交的

- We want expected error to be small. How to compute $\mathbf{W}$?

Principal Component Analysis

记得：$y=W^Tx$

- Recovered vector $\mathbf{x}_r = \mathbf{W}\mathbf{y}$
- Error: $\epsilon = \mathbf{x} - \mathbf{x}_r = \mathbf{x} - \mathbf{W}\mathbf{W}^\top\mathbf{x}$
- We want small expected error:

$$
\begin{aligned}
||\epsilon||^2 &= \epsilon^\top\epsilon \\
&= (\mathbf{x} - \mathbf{W}\mathbf{W}^\top\mathbf{x})^\top(\mathbf{x} - \mathbf{W}\mathbf{W}^\top\mathbf{x}) \\
&= \mathbf{x}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{W}\mathbf{W}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{W}\mathbf{W}^\top\mathbf{x} + \mathbf{x}^\top\mathbf{W}\mathbf{W}^\top\mathbf{W}\mathbf{W}^\top\mathbf{x} \\
&= \mathbf{x}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{W}\mathbf{W}^\top\mathbf{x}
\end{aligned}
$$

Note that $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$.

因为正交性

Principal Component Analysis

最简单：降到1维的情况

Let $k = 1$, i.e. $\mathbf{W}$ is vector, $\mathbf{y}$ is scalar. Then

$$\begin{aligned}
E[\varepsilon^\top \varepsilon] &= E[\mathbf{x}^\top \mathbf{x}] - E[(\mathbf{x}^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x})] \\
&= E[\mathbf{x}^\top \mathbf{x}] - E[\mathbf{w}^\top \mathbf{x}\mathbf{x}^\top \mathbf{w}] \\
&= E[\mathbf{x}^\top \mathbf{x}] - \mathbf{w}^\top E[\mathbf{x}\mathbf{x}^\top]\mathbf{w} \\
&= E[\mathbf{x}^\top \mathbf{x}] - \mathbf{w}^\top \mathbf{C}_x \mathbf{w}
\end{aligned}$$

Covariance矩阵

$$\boxed{C_x = \frac{1}{N}(\boldsymbol{x}_1\boldsymbol{x}_1^T + \boldsymbol{x}_2\boldsymbol{x}_2^T + \cdots + \boldsymbol{x}_N\boldsymbol{x}_N^T)}$$

Principal Component Analysis

We need to find $\mathbf{w}$ that minimizes $E[\epsilon^\top \epsilon]$ 如何最小化期望的重建误差！

This is the same as maximizing the 2^{nd} term on right-hand side:

$$\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}_X \mathbf{w}$$

But we should normalize by length of $\mathbf{w}$, so define

$$J = \frac{\mathbf{w}^\top \mathbf{C}_X \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \tag{2}$$

Goal: find $\mathbf{w}$ to maximize J

Principal Component Analysis

Take derivatives and set to 0:

$$\frac{dJ}{d\mathbf{w}} = \frac{(\mathbf{w}^\top \mathbf{w})2\mathbf{C}_X\mathbf{w} - (\mathbf{w}^\top \mathbf{C}_X\mathbf{w})2\mathbf{w}}{(\mathbf{w}^\top \mathbf{w})^2} = \mathbf{0}$$

$$\mathbf{0} = \frac{2\mathbf{C}_X\mathbf{w}}{\mathbf{w}^\top \mathbf{w}} - \left[\frac{\mathbf{w}^\top \mathbf{C}_X\mathbf{w}}{\mathbf{w}^\top \mathbf{w}}\right] \bullet \frac{2\mathbf{w}}{\mathbf{w}^\top \mathbf{w}}$$

Note: term in brackets is J, so we rearrange to get:

$$\mathbf{C}_X\mathbf{w} = J\,\mathbf{w} \quad \longleftarrow \text{ Eigenvalue problem!}$$

Thus, $\mathbf{w}$ is eigenvector of $\mathbf{C}_X$ corresponding to largest eigenvalue (= J).

Principal Component Analysis

- 为何是对应最大特征值的特征向量？

最小化：$E(\epsilon^T\epsilon) = E(x^Tx) - E[(x^Tw)(w^Tx)]$

$$E(\epsilon^T\epsilon) = E(x^Tx) - w^TC_xw$$

因为$C_xw = Jw$

$$E(\epsilon^T\epsilon) = E(x^Tx) - w^TJw$$

$$E(\epsilon^T\epsilon) = E(x^Tx) - Jw^Tw$$

最大化特征值！

Principal Component Analysis

如果要保留多个维度：

In general, PCA is: $\mathbf{y} = \mathbf{w}^\top(\mathbf{x} - \mathbf{m})$

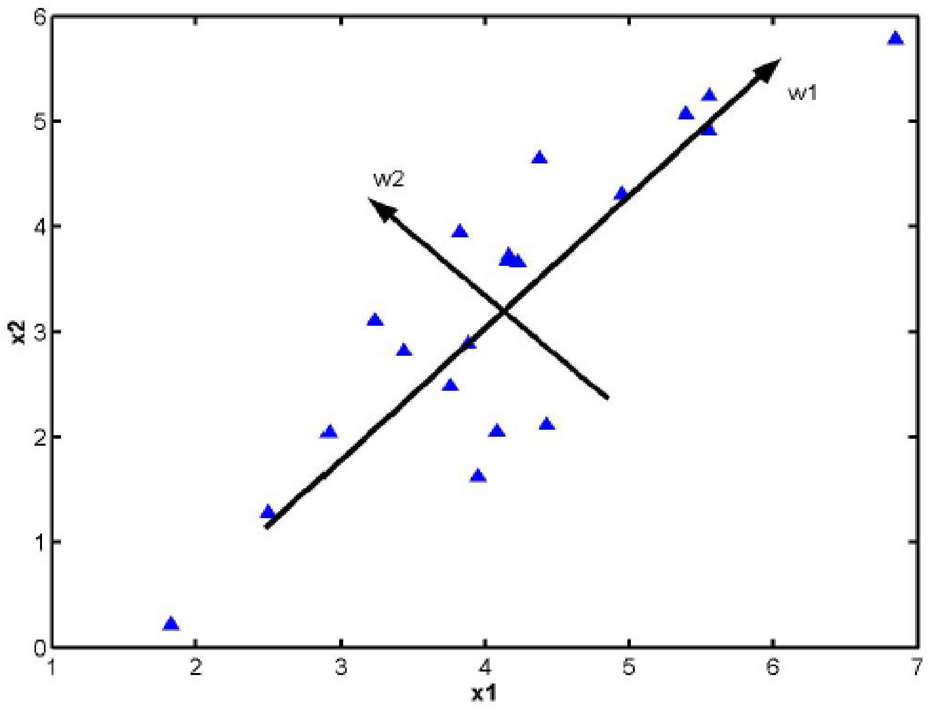
where $\mathbf{m} = E[\mathbf{x}]$ mean, and $\mathbf{W}$ is $d \times k$ matrix containing the k eigenvectors of $Var[\mathbf{x}]$ (covariance matrix) corresponding to the top k eigenvalues.

$$\mathbf{W} = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_k \\ | & | & & | \end{bmatrix}$$

$\mathbf{w}_1$: First principal component,
$\mathbf{w}_2$: Second principal component, etc.

Principal Component Analysis

几何意义：其实是拟合了训练数据的长轴短轴



- PCA is a shift and rotation of the axes.
- $\mathbf{w}_1$: direction of greatest elongation
- $\mathbf{w}_2$: direction of next greatest elongation, and orthogonal to previous eigenvector; etc.
- $\mathbf{W}$ is orthogonal because $\mathbf{C}_x$ is symmetric.
- $\mathbf{W}\mathbf{W}^\top \neq \mathbf{I}$ unless $k = d$

Principal Component Analysis

Dimensionality Reduction: $\mathbf{x} \in \mathbb{R}^d, \mathbf{y} \in \mathbb{R}^k, k \ll d$

Typically, choose k so that ratio $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{d} \lambda_j} > 90\%$ "Energy"

But how to get $\mathbf{C}_x, \mathbf{m}$? Statistics of $\mathbf{x}$

Given data $\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x}_N$, estimate $\mathbf{m}, \mathbf{C}_x$

Sample mean $\widehat{\mathbf{m}} = \frac{1}{N} \sum_{i=1}^{N} x_i$

Sample covariance matrix:

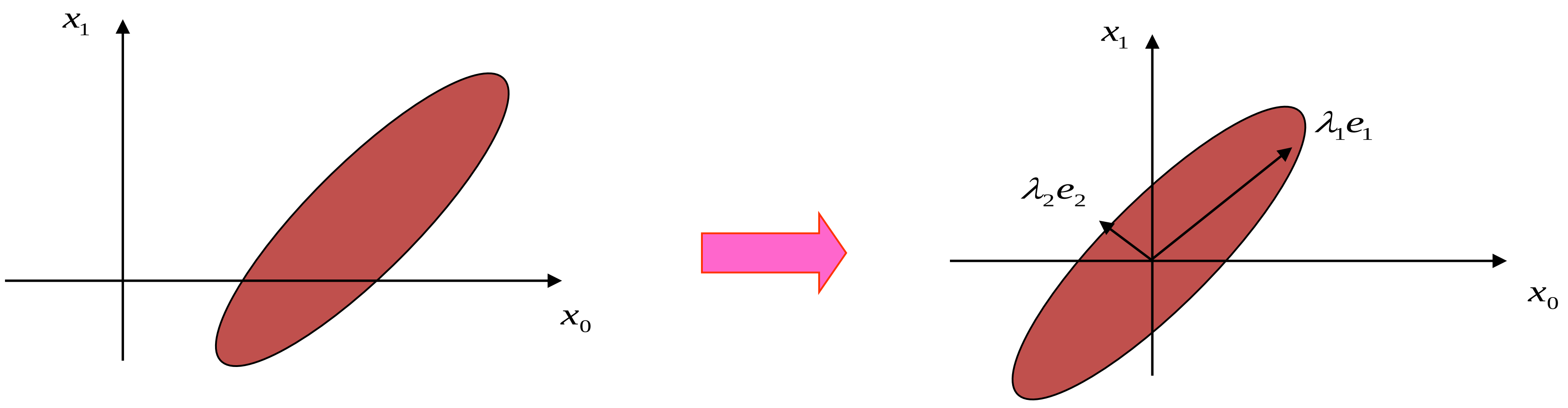
$$\widehat{\mathbf{C}_x} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top$$

$$\text{or } \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top$$

$$\text{or Scatter matrix} \quad \mathbf{S} = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top$$

线性数据降维

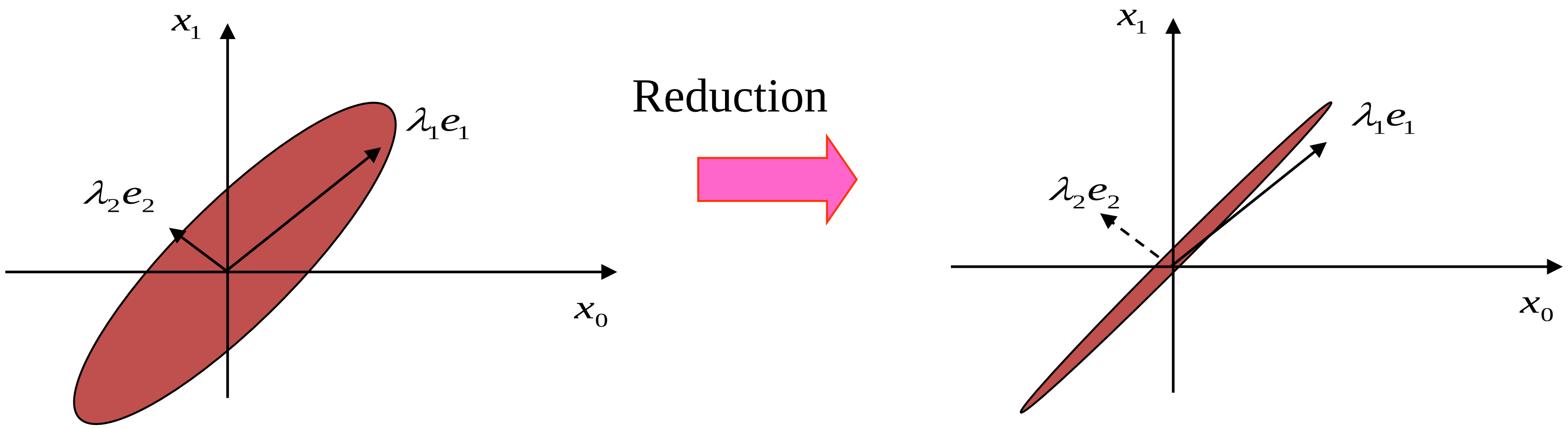
- PCA总结
  - 找到一个线性映射方向，使得映射后得到的低维度向量分布散射最大
  - 协方差矩阵的特征向量方向，即为最佳映射方向



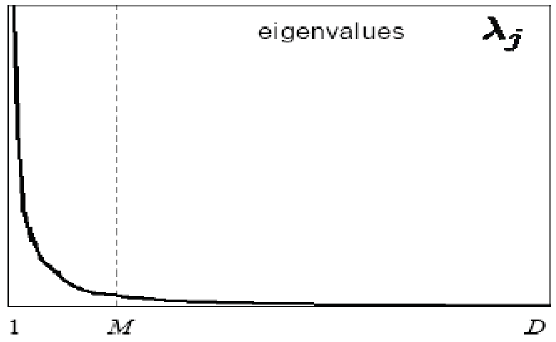
λ_k is the marginal variance along the principle direction e_k

线性数据降维

- PCA
  - 映射到 e_1 方向得到最大variance以及最小重建误差

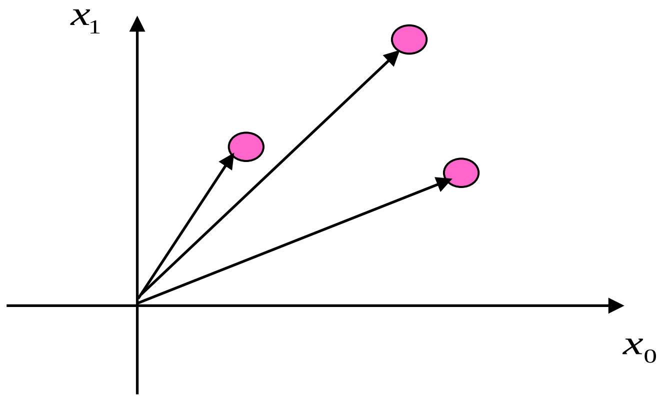


  - 选择保留多少维度 M:
    保留越多维度，重建误差越小

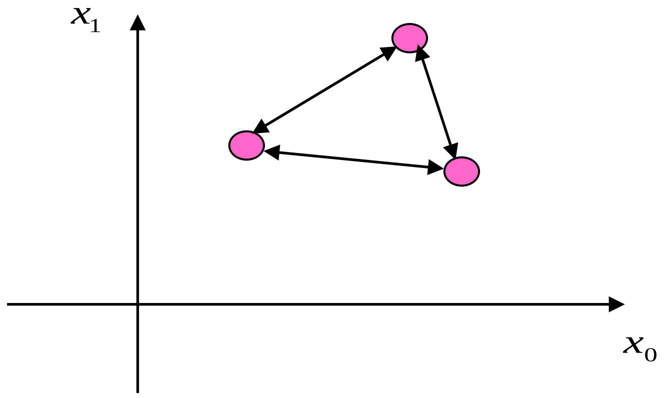


Multi-Dimensional Scaling

- Multi-Dimensional Scaling (MDS)
  - 找到映射方向使得在低维空间中高维度样本间距离不变



PCA



MDS

$$\min_{\tilde{\mathbf{x}}_1,\cdots,\tilde{\mathbf{x}}_n} \left(\sum_{i<j} \left(\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\| - d_{ij} \right)^2 \right)^{1/2}$$

Multi-Dimensional Scaling

- Find a set of points which have, under the Euclidean metric, the same distance matrix as D

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix}$$

- 假设降维后的样本为$x_1, x_2, x_3 \ldots$ 组成数据矩阵X
- 另外定义矩阵T $\mathbf{T}=\mathbf{X}\mathbf{X}^{\mathrm{T}}$
- 其中t_{ij}元素计算: $t_{ij}=\mathbf{x}_i\mathbf{x}_j$
- The distance matrix D contains terms like

$$d_{ij}^2=(\mathbf{x}_i-\mathbf{x}_j)^2=\mathbf{x}_i^2+\mathbf{x}_j^2-2\mathbf{x}_i\cdot\mathbf{x}_j$$

$$t_{ij}=-\frac{1}{2}\left(d_{ij}^2-\mathbf{x}_i^2-\mathbf{x}_j^2\right)$$

Multi-Dimensional Scaling

- We also have

$$\sum_j d_{ij}^2 = n\mathbf{x}_i^2 + \sum_j \mathbf{x}_j^2 - 2\mathbf{x}_i \sum_j \mathbf{x}_j = n\mathbf{x}_i^2 + \sum_j \mathbf{x}_j^2$$

$$\sum_i d_{ij}^2 = n\mathbf{x}_j^2 + \sum_i \mathbf{x}_i^2 - 2\mathbf{x}_j \sum_i \mathbf{x}_i = n\mathbf{x}_j^2 + \sum_i \mathbf{x}_i^2$$

$$\sum_{ij} d_{ij}^2 = n\sum_i \mathbf{x}_i^2 + n\sum_j \mathbf{x}_j^2$$

- 可以解得：

$$t_{ij} = -\frac{1}{2}\left[d_{ij}^2 - \frac{1}{n}\sum_k d_{ik}^2 - \frac{1}{n}\sum_k d_{kj}^2 + \frac{1}{n^2}\sum_{k,l} d_{kl}^2 \right]$$

- 知道了t_{ij}，相当于知道了T矩阵
- 因为$T = XX^T$，因此我们可以通过矩阵分解求得原来的X

  即可以知道X的每一个行向量，即$\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$，完成降维

Multi-Dimensional Scaling

- 对T进行特征向量分解，拆分成旋转对称的两部分：
  ：

$$\mathbf{T} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T = \underbrace{\mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}}_{\mathbf{X}} \underbrace{\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^T}_{\mathbf{X}^T}$$

- 因此，通过对T矩阵进行分解，可以获得X矩阵，从而获得低维样本
- 可以根据要求，保留一定的特征值，对应的特征向量可以组成低维样本，类似PCA
- 问题：如在一些复杂问题中，线性分解无法保留足够信息，如何处理？

AI300学院