降维算法

Machine Learning Engineer

机器学习工程师

讲师：Ivan

稀牛学院 網易 NETEASE

目 录

CONTENTS

01 无监督学习：降维介绍

02 主成分分析

03 IsoMap 算法

04 Multidimensional Scaling (MDS)

01 无监督学习：降维介绍

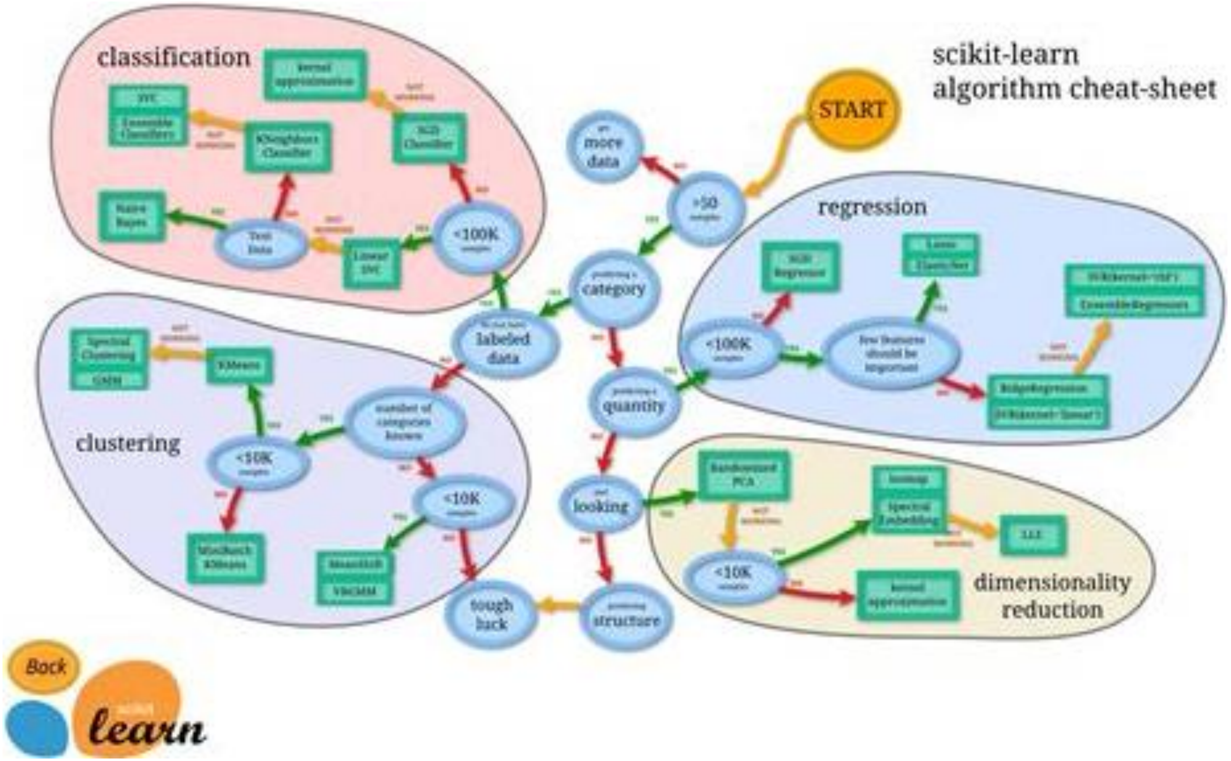
1.1 什么是降维?

1.2 降维算法的应用

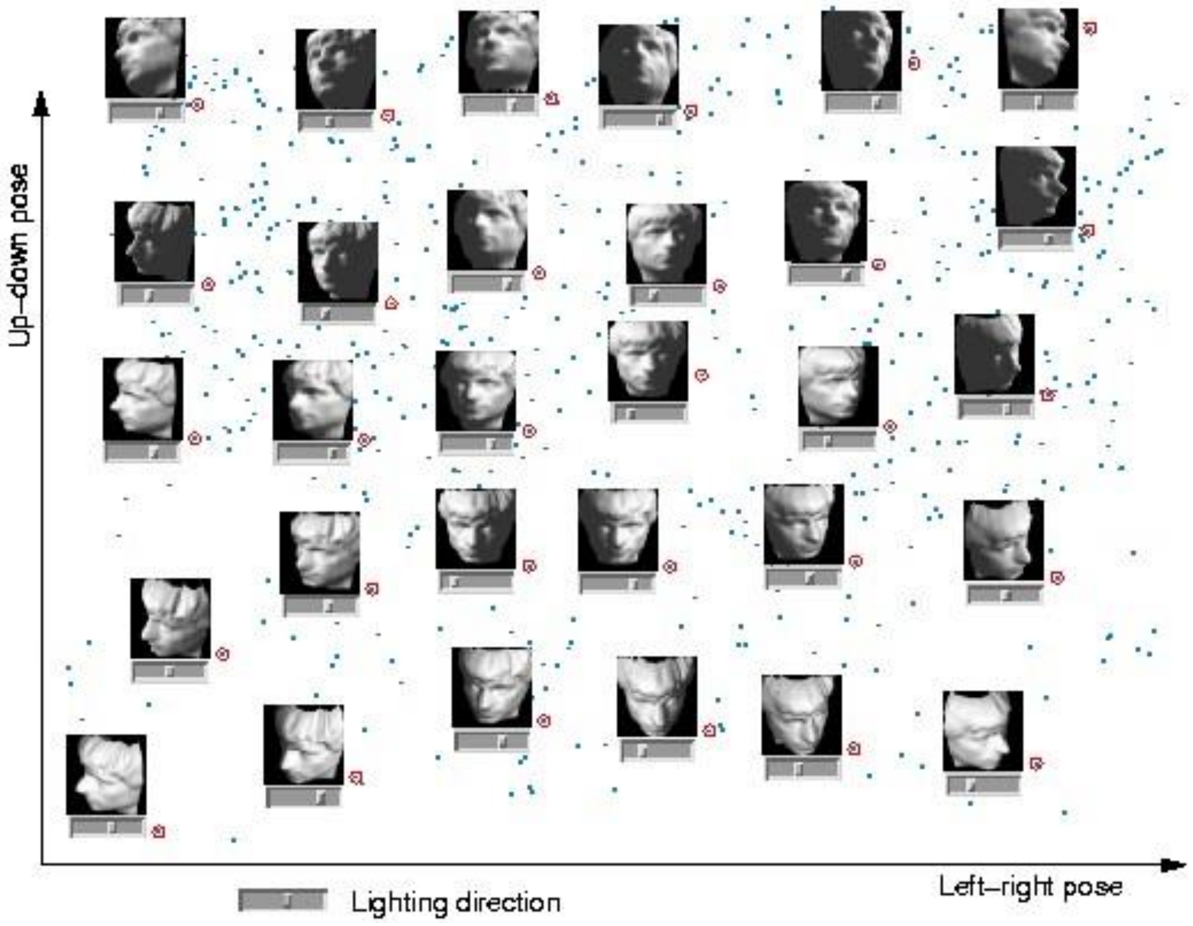
什么是降维?

什么是降维(dimensionality reduction)?

1. 无监督学习 (不需要标签)
2. 典型应用：
   - 数据可视化
   - 数据压缩
   - 数据预处理
   - ……



什么是降维(dimensionality reduction)?



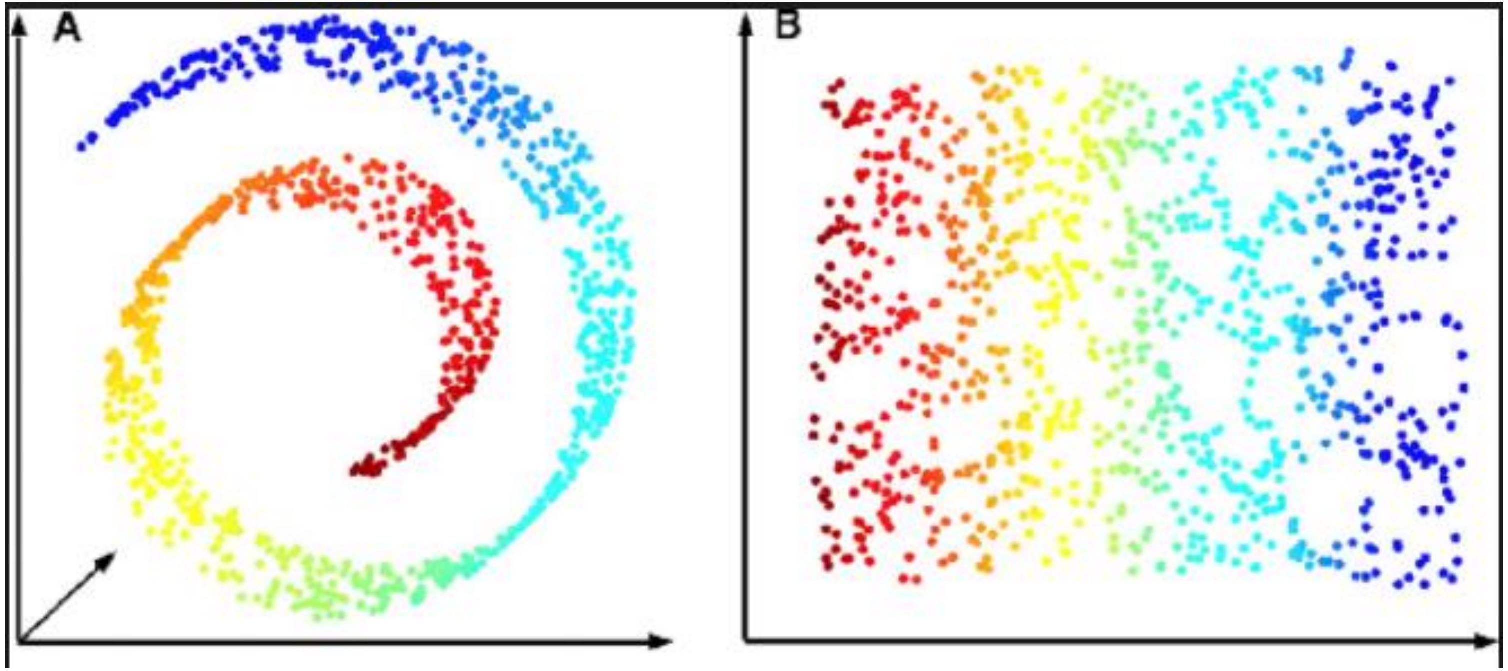
1.1 什么是降维?

降维算法大体上可以分成两类:

1. **线性降维:**
   - 主成分分析 (PCA)
   - 非负矩阵分解 (NMF)
   - 线性判别分析 (Linear Discriminant Analysis)
   - ……
2. **非线性降维:**
   - IsoMap
   - Locally Linear Embedding (LLE)
   - 自编码器 (Auto-encoder)
   - ……

降维算法的应用

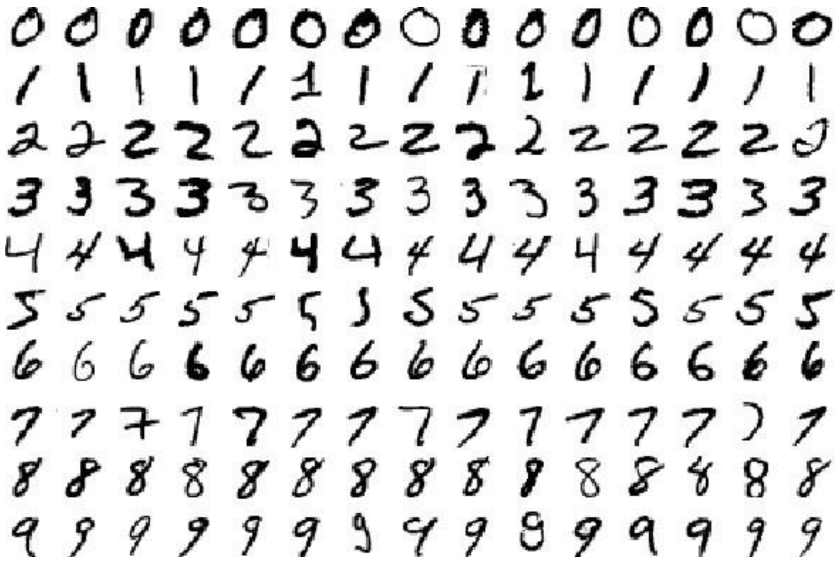
数据压缩：

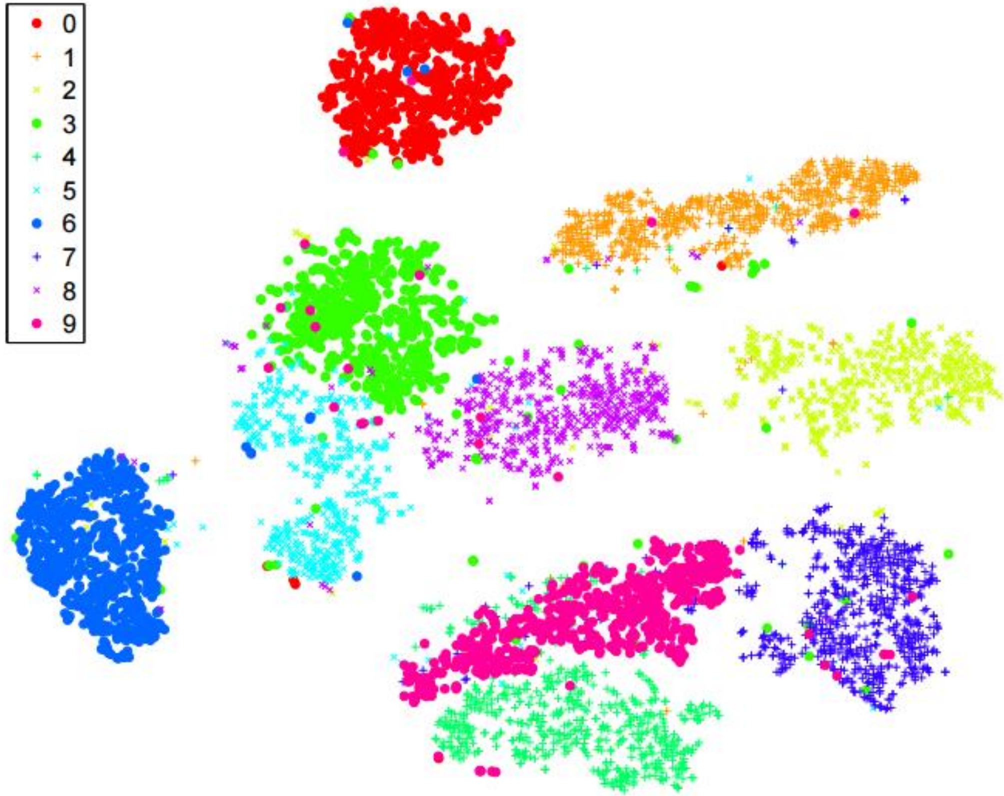


高维数据往往可以用一个低维流形(manifold)来描述(3维->2维)

降维算法的应用

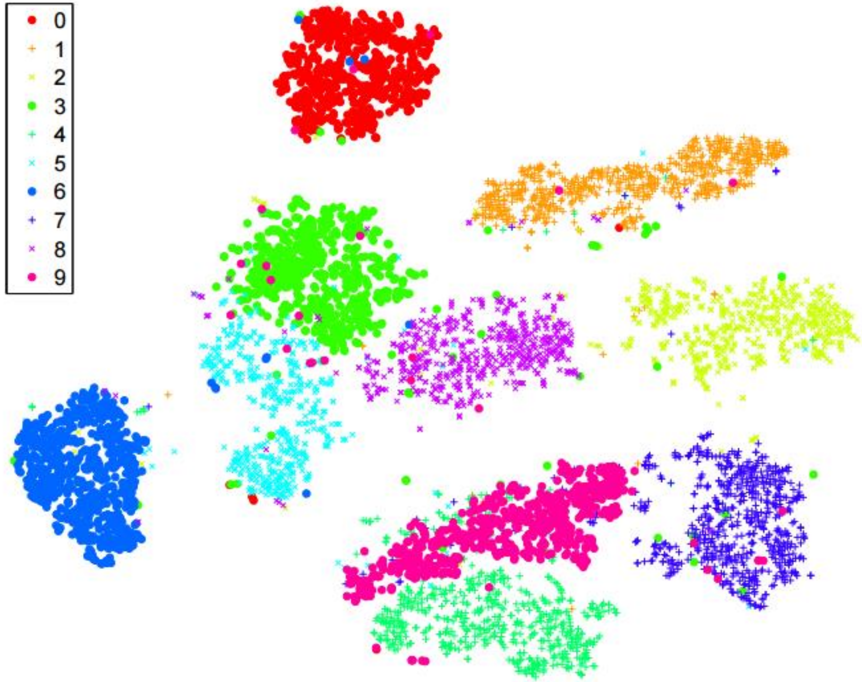
数据可视化：





- 每一个数字包含28x28 = 784个像素点
- 将数据从784 维降低到 2维，方便可视化数据的结构

要点总结



1.1 降维算法的分类：线性降维以及非线性降维

1.2 降维算法的应用：数据压缩以及可视化

02

主成分分析

2.1 线性降维

2.2 主成分分析的几何解释

2.3 主成分分析算法

2.1 线性降维

目标：给定$X=\{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

- 如何描述线性映射?

回忆：$f(x)=w_1x_1+w_2x_2+\cdots+w_dx_d$被称作一个线性函数
写成更加简单的形式：$f(x)=w^Tx$

f 就是一个线性变换，从$R^d \to R^1$

思考：如果扩展成从$R^d \to R^p, 1<p \ll d$ 的线性映射?

2.1 线性降维

目标：给定$X = \{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

思考：如果扩展成从$R^d \rightarrow R^p, 1 < p \ll d$ 的线性映射?

$$
\begin{gathered}
f_1(x) = w_{11}x_1 + w_{12}x_2 + \cdots + w_{1d}x_d = w_1^T x \\
f_2(x) = w_{21}x_1 + w_{22}x_2 + \cdots + w_{2d}x_d = w_2^T x \\
\vdots \\
\vdots \\
f_p(x) = w_{p1}x_1 + w_{p2}x_2 + \cdots + w_{pd}x_d = w_p^T x
\end{gathered}
$$

$$f: R^d \rightarrow R^p : x \rightarrow Wx, W \in R^{p \times d}$$

矩阵变换代码线性映射，W矩阵的第i行即为w_i

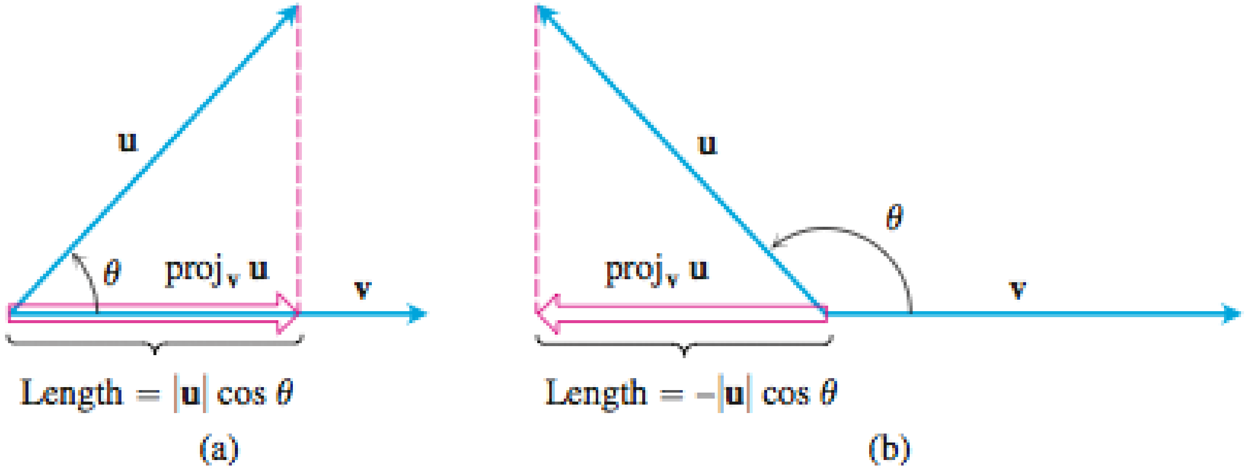
2.1 线性降维

目标：给定$X=\{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p\times d}$$

我们要求矩阵W满足如下条件：$WW^T = I_p$，即：

- 不同行的行向量之间互相垂直，即$w_i^T w_j = 0, \forall i \neq j$
- W的每一个行向量w_i是一个单位向量: $w_i^T w_i = 1, \forall i \in [p]$



2.1 线性降维

目标：给定$X=\{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p\times d}$$

我们要求矩阵W满足如下条件：$WW^T = I_p$，即：

- 不同行的行向量之间互相垂直，即$w_i^T w_j = 0, \forall i \neq j$
- W的每一个行向量w_i是一个单位向量: $w_i^T w_i = 1, \forall i \in [p]$

$\tilde{x} = Wx \in R^p$ 是x在$W = \{w_1, \dots, w_p\} \subseteq R^d$扩张成的线性子空间中的坐标，即：

$$y \approx \tilde{x}_1 w_1 + \tilde{x}_2 w_2 + \cdots + \tilde{x}_p w_p$$

是对原向量x的一个重构

简洁的表示方法: $y = W^T\tilde{x} = W^T Wx \in R^d$

2.1 线性降维

目标：给定$X=\{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p\times d}$$

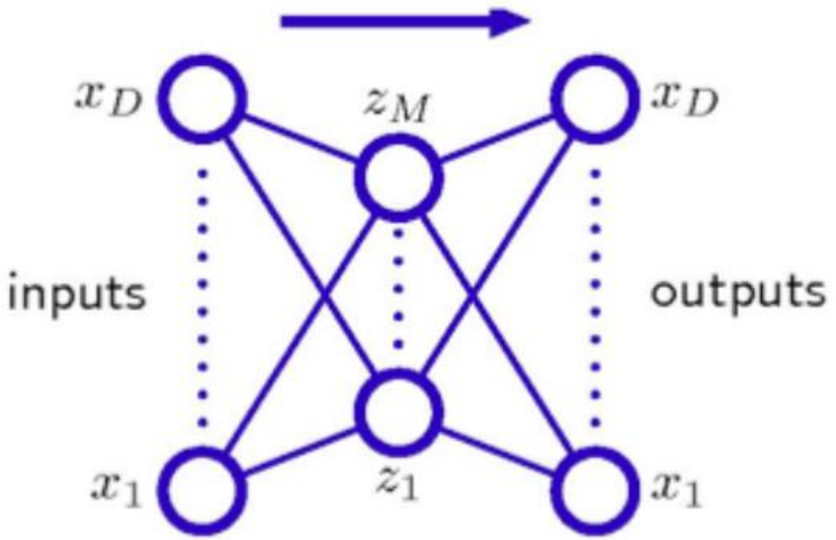
重点：$\tilde{x} = Wx \in R^p$ 是从R^d到R^p的一个投影操作(降维压缩)
重点：$y = W^T\tilde{x} = W^TWx \in R^d$是用降维后的表示对原始数据的一个重构

更一般地，第一部分被称作dimensionality reduction (encoding)，第二部分称作reconstruction (decoding)

2.1 线性降维

目标：给定$X = \{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p\times d}$$

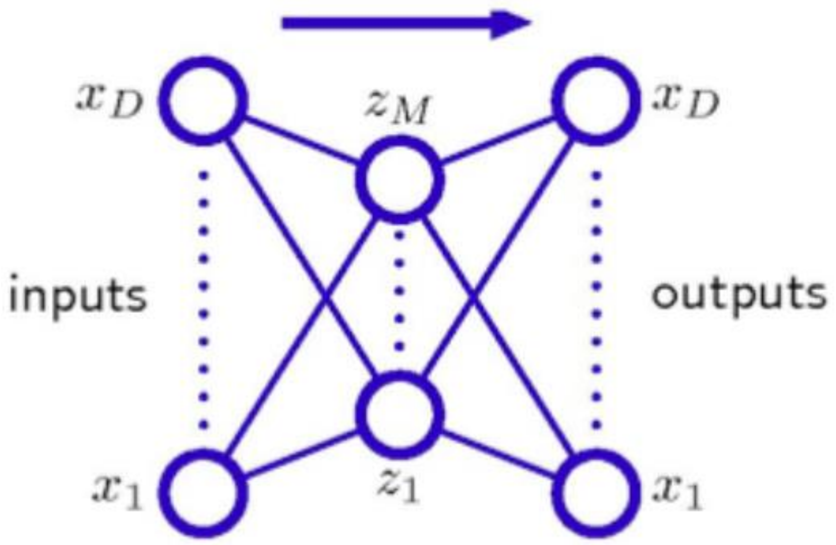


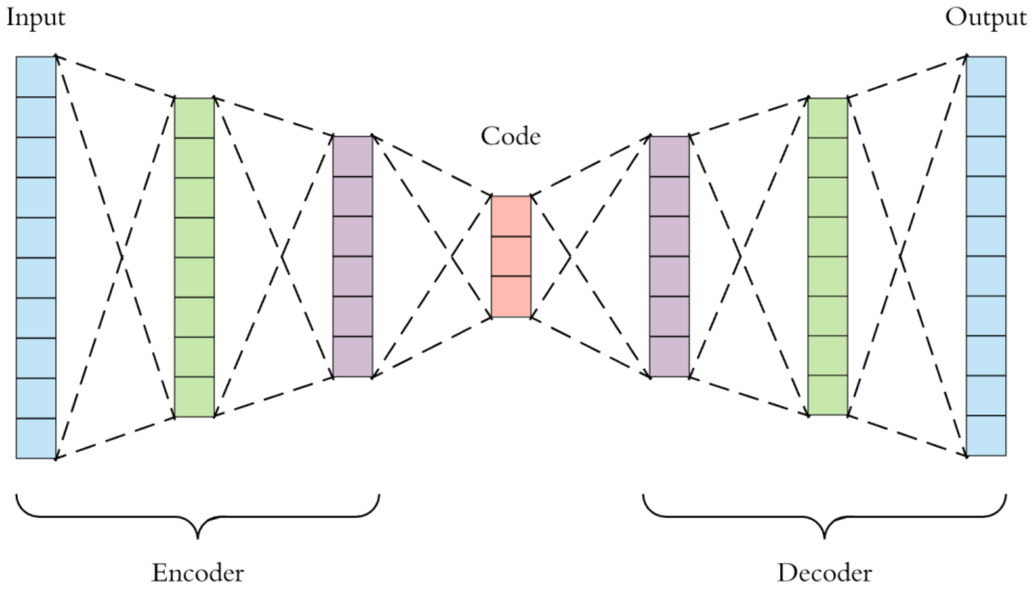
- PCA 可以用一个两层线性神经网络来刻画

2.1 线性降维

目标：给定$X = \{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p\times d}$$





- PCA 可以用一个两层线性神经网络来刻画
- 自编码器是PCA的深层、非线性扩展

主成分分析的几何解释

目标：给定$X = \{x_i\}^n{}_{i=1}, x_i \in R^d$, 寻找线性映射来将数据映射到$R^p, p \ll d$

$$f: R^d \to R^p: x \to Wx, W \in R^{p \times d}$$

问题：如何确定投影矩阵W?

- W 包含p个互相垂直的方向，我们可以依次确定每个方向
- 用什么标准来确定方向?

2.2 主成分分析的几何解释

问题：如何确定投影矩阵W?

- 最小化数据x以及它的重构向量y之间的距离

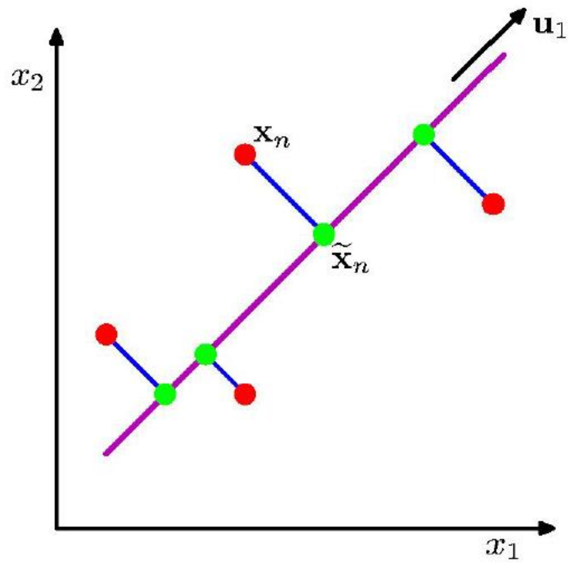
$$\min_{W} \sum_{i} ||x_i - W^T W x_i||^2$$

$$\text{s.t.} \quad WW^T = I_p$$

等价于：

$$\min_{W} \ ||X - XW^T W||_F^2$$

$$\text{s.t.} \quad WW^T = I_p$$



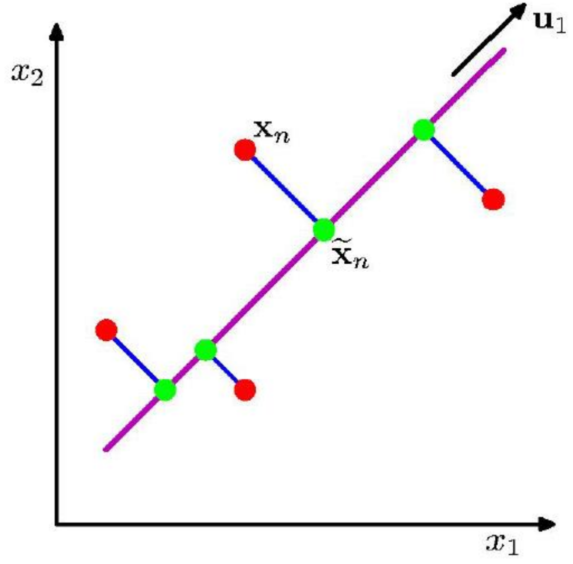
2.2 主成分分析的几何解释

问题：如何确定投影矩阵W?

$$\min_{W} \ ||X - XW^TW||_F^2$$
$$\text{s.t.} \quad WW^T = I_p$$

假设 p = 1 (先求解第一个方向)

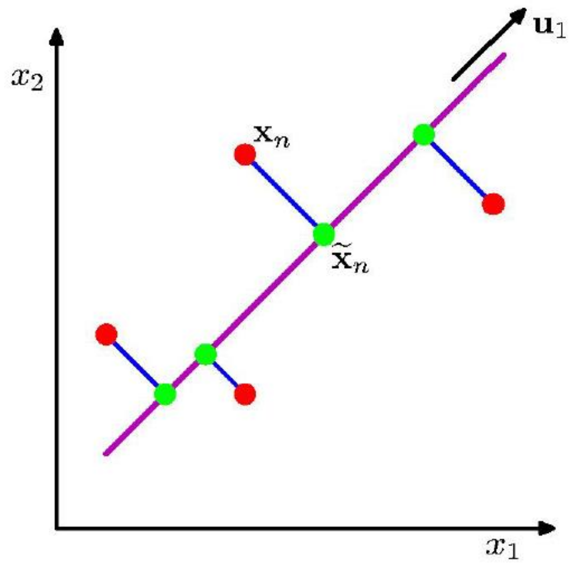
$$\min_{w} ||X - Xww^T||_F^2$$
$$\text{s.t.} \ w^Tw = 1$$



2.2 主成分分析的几何解释

$$\min_{w} ||X - Xww^T||_F^2 \quad \text{s.t. } w^T w = 1$$

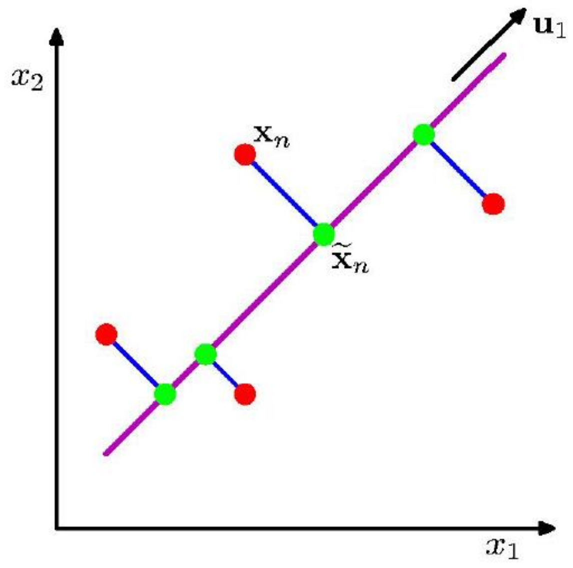
推导：



2.2 主成分分析的几何解释

$$\min_{w} ||X - Xww^T||_F^2 \quad \text{s.t. } w^Tw = 1$$

令$C = X^TX \in R^{d\times d}$，$C$是数据矩阵X所对应的协方差矩阵



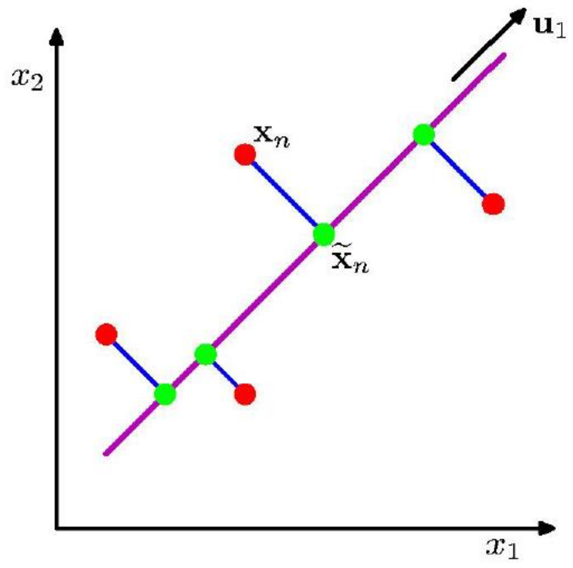
- w_1是C的最大特征向量
- $\cdots$
- $\cdots$
- w_p是C的第p大的特征向量

2.2 主成分分析的几何解释

$$\min_{w} \|X - Xww^T\|_F^2 \quad \text{s.t. } w^T w = 1$$

令$C = X^T X \in R^{d\times d}$，$C$是数据矩阵X所对应的协方差矩阵

- w_1是C的最大特征向量
- $\cdots$
- $\cdots$
- w_p是C的第p大的特征向量



另一方面，因为C是协方差矩阵，所以：

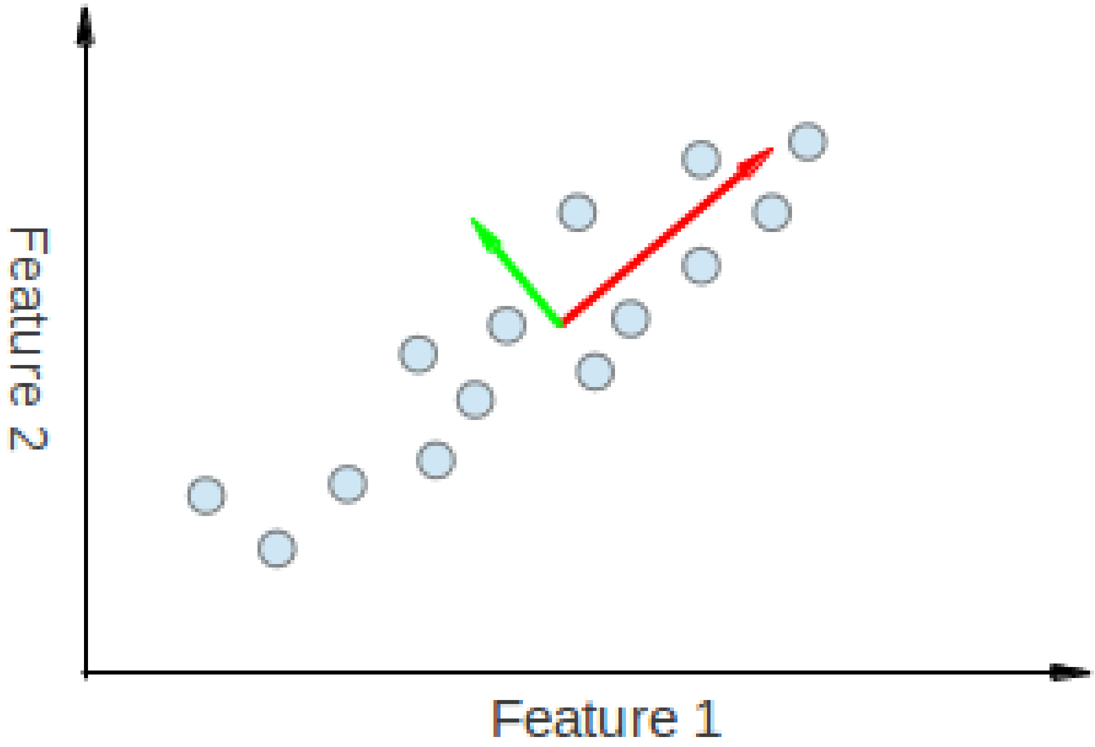
- w_1是数据方差最大的方向
- $\cdots$
- $\cdots$
- w_p是数据方差(变化)第p大的方向

2.2 主成分分析的几何解释

$$\min_{w} \left|\left|X - Xww^T\right|\right|_F^2 \quad \text{s.t. } w^Tw = 1$$

令$C = X^TX \in R^{d\times d}$，$C$是数据矩阵X所对应的协方差矩阵

- w_1是数据方差最大的方向
- w_p是数据方差(变化)第p大的方向



2.3 主成分分析算法

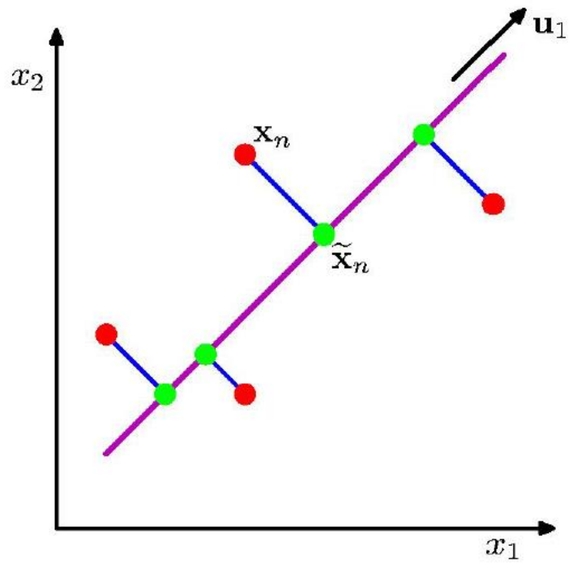
给定$X=\{x_i\}^n{}_{i=1}, x_i \in R^d$，将数据组织成矩阵$X \in R^{n\times d}$
计算协方差矩阵C：

$$C=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T \qquad \mu=\frac{1}{n}\sum_{i=1}^{n}x_i$$

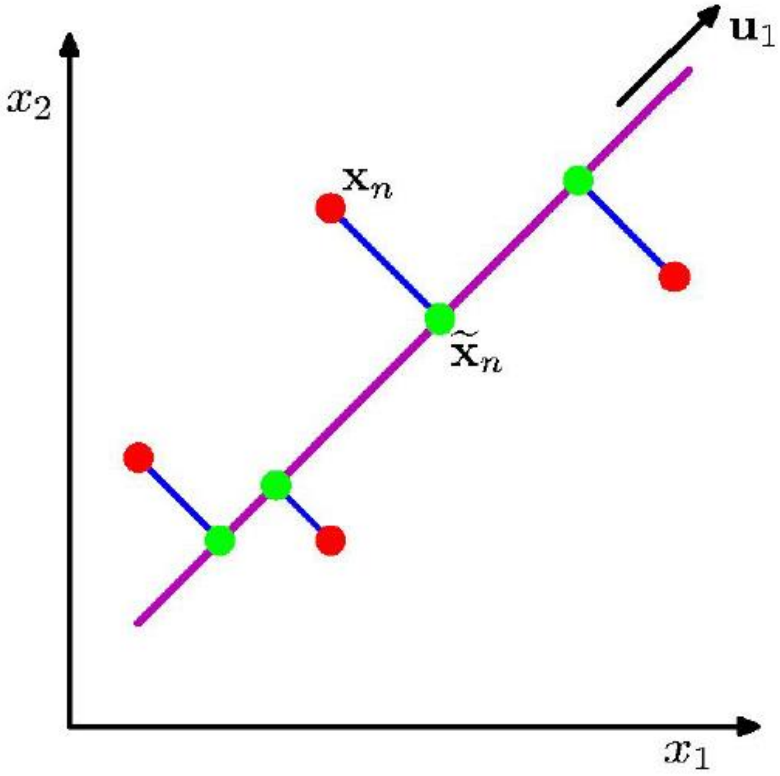
计算C的特征值分解：$C=\mathrm{Q}\Sigma Q^T, Q \in R^{d\times \mathrm{d}}$
设置$W \in R^{p\times d}$为Q^T矩阵的前p行

计算投影(压缩)后的矩阵：$\tilde{X}=XW^T$
重构数据矩阵：$Y=XW^TW$

计算复杂度：$O(nd^2+pd^2)$



要点总结



2.1 主成分分析：最小化重构误差

2.2 主成分分析：最大化压缩后数据的协方差

2.3 主成分分析的求解方法

2.4 主成分分析与自编码器的联系与区别

03

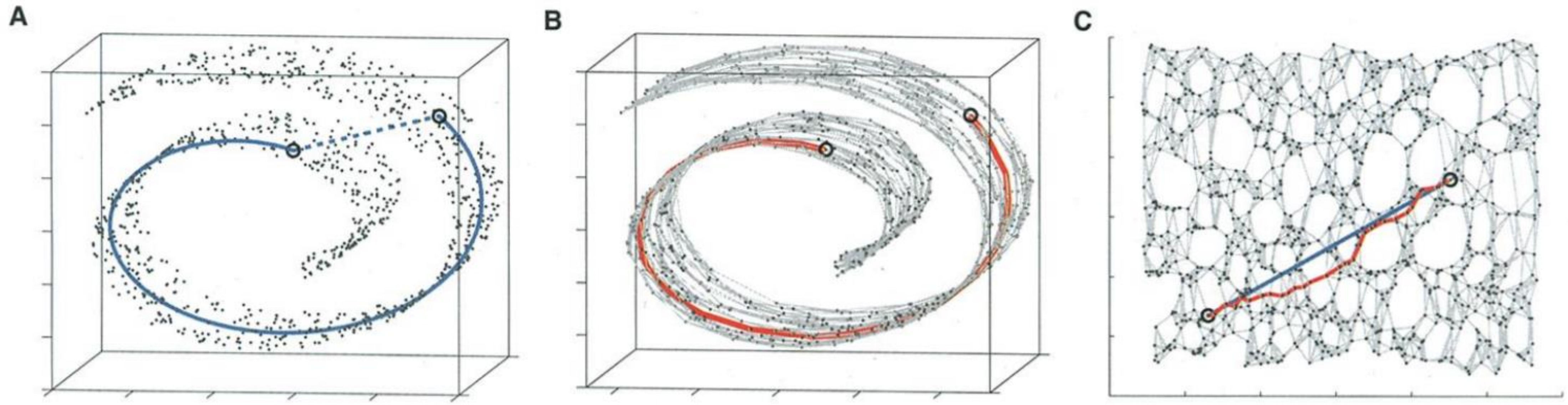
IsoMap 算法

IsoMap 算法

主成分分析(PCA)属于线性降维算法

IsoMap 是非线性降维算法，由MIT Tenebaum教授等人于2000年提出:

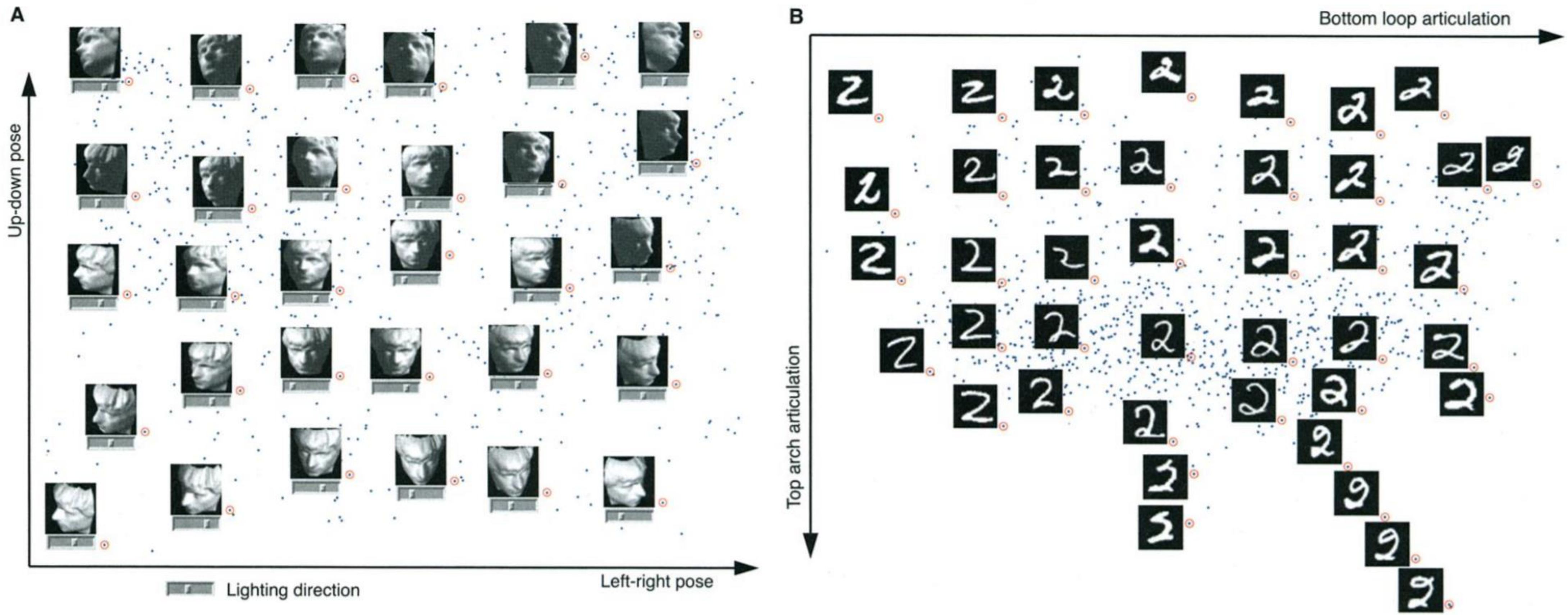
A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science 290, (2000), 2319–2323



- 主要想法：用局部的欧式距离来近似局部测地距离

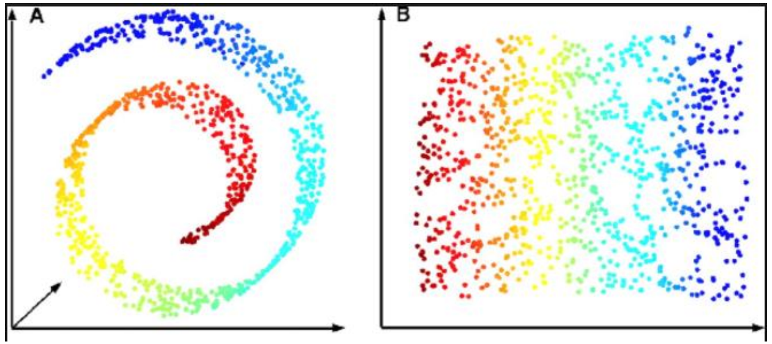
3 IsoMap 算法

主要想法：用局部的欧式距离来近似局部测地距离



3 IsoMap 算法

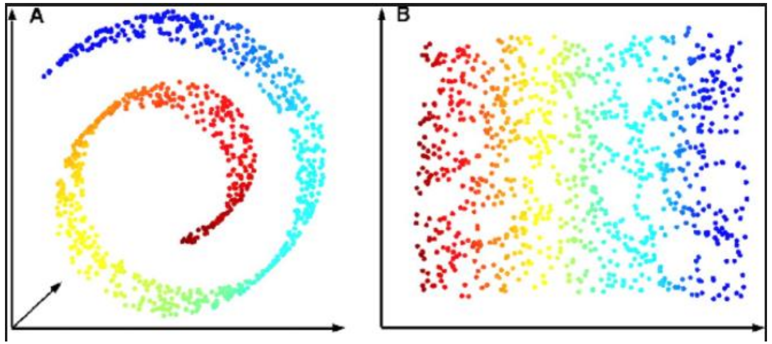
主要想法：用局部的欧式距离来近似局部测地距离
目标：两个点之间的测地距离近似等于降维之后的直线距离



1. 确定每个点的邻居节点：
   I. 在一定半径之内的所有点
   II. K近邻
2. 根据相邻节点构造一个加权图
   I. 边权等于相邻节点之间的欧式距离
3. 对于任意两个不同的节点，计算它们在加权图中的最短路径
   I. Dijkstra’s algorithm
   II. Floyd-Warshall algorithm
4. 根据两两最短路径，构造一个距离矩阵D
5. 计算低维嵌入V, 使得$\left\|V_i - V_j\right\|_2 \approx D_{ij}$

3 IsoMap 算法

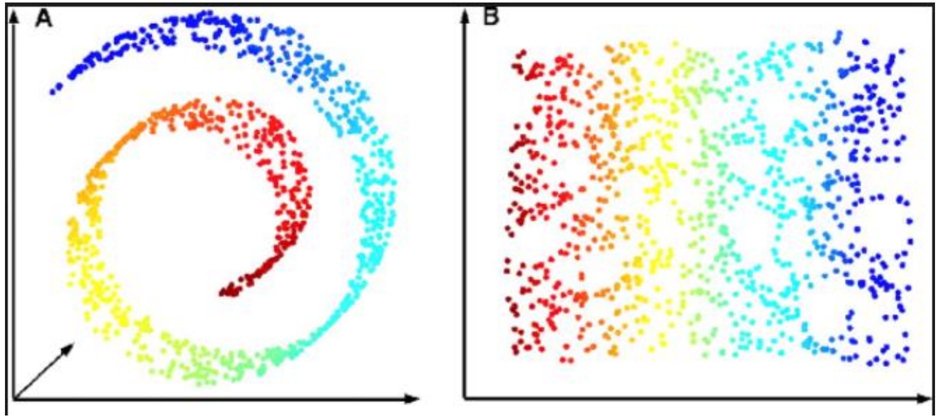
计算低维嵌入V, 使得$\left\|V_i - V_j\right\|_2 \approx D_{ij}$?



本质问题：
给定一个距离矩阵D,D_{ij}代表第i个和第j个数据点之间的距离，求一个低维嵌入方法V, 使得V_i和V_j之间的直线距离近似D_{ij}

Multi-dimensional Scaling (MDS)

要点总结



3.1 局部直线距离近似测地线距离

3.2 构造加权图计算最短路径

3.3 根据距离矩阵，使用MDS计算嵌入向量

04 Multidimensional Scaling (MDS)

4 Multidimensional Scaling

问题：
给定一个距离矩阵D, D_{ij}代表第i个和第j个数据点之间的距离，求一个低维嵌入方法V, 使得V_i和V_j之间的直线距离近似D_{ij}

$$\min_{V_1,\ldots,V_n} \quad \sum_{i<j} (\|V_i - V_j\|_2 - D_{ij})^2$$

注意：
- 上述问题的最优解不唯一，对矩阵V的平移、旋转、镜面变换等正交变换不改变$\left\|V_i - V_j\right\|_2$
- MDS通常通过数值优化方法来进行求解，包括：
  1. 一阶方法，基于梯度下降的方法
  2. 二阶方法，拟牛顿法

Multidimensional Scaling

问题：
给定一个距离矩阵D, D_{ij}代表第i个和第j个数据点之间的距离，求一个低维嵌入方法V, 使得V_i和V_j之间的直线距离近似D_{ij}

$$\min_{V_1,\ldots,V_n} \sum_{i<j} (||V_i - V_j||_2 - D_{ij})^2$$

不失一般性，可以假设:

$$\sum_{i=1}^{n} V_i = 0$$

Multidimensional Scaling

$$\min_{V_1,\ldots,V_n} \quad \sum_{i<j}(||V_i - V_j||_2 - D_{ij})^2$$

不失一般性，可以假设: $\sum_{i=1}^{n} V_i = 0$

给定嵌入矩阵V，我们可以定义Gram矩阵 $G = V^T V$，那么

$$D_{ij}^2 = \left\|V_i - V_j\right\|_2^2 = V_i^T V_i - 2V_i^T V_j + V_j^T V_j = G_{ii} - 2G_{ij} + G_{jj}$$

对上述表达式进行求和，我们有：

$$\sum_{i=1}^{n} D_{ij}^2 = tr(G) + nG_{jj} \quad \sum_{j=1}^{n} D_{ij}^2 = tr(G) + nG_{ii} \quad \sum_{i,j=1}^{n} D_{ij}^2 = 2n\, tr(G)$$

Multidimensional Scaling

$$D_{ij}^2 = \left\|V_i - V_j\right\|_2^2 = V_i^T V_i - 2V_i^T V_j + V_j^T V_j = G_{ii} - 2G_{ij} + G_{jj}$$

$$\sum_{i=1}^{n} D_{ij}^2 = tr(G) + nG_{jj} \qquad \sum_{j=1}^{n} D_{ij}^2 = tr(G) + nG_{ii} \qquad \sum_{i,j=1}^{n} D_{ij}^2 = 2n\, tr(G)$$

联立上述所有方程，在这种情况下我们可以求解出：

$$G_{ij} = -\frac{1}{2}\left(D_{ij}^2 - D_{\cdot j}^2 - D_{i\cdot}^2 + D_{\cdot\cdot}^2\right)$$

其中，

$$D_{\cdot j}^2 = \frac{1}{n}\sum_{i=1}^{n} D_{ij}^2 \qquad D_{i\cdot}^2 = \frac{1}{n}\sum_{j=1}^{n} D_{ij}^2 \qquad D_{\cdot\cdot}^2 = \frac{1}{n^2}\sum_{i,j=1}^{n} D_{ij}^2$$

Multidimensional Scaling

$$G_{ij} = -\frac{1}{2n^2}(D_{ij}^2 - D_{\cdot j}^2 - D_{i\cdot}^2 + D_{\cdot\cdot}^2)$$

我们可以根据上述公式求得整个矩阵G，由于$G = V^T V, V \in R^{n\times q}$，
对矩阵G做特征值分解，我们即可求得矩阵V

思考：如何从矩阵G的特征值分解中得到V?

答案：

要点总结

4.1 MDS的目标

4.2 MDS与PCA的联系与区别

4.3 MDS在IsoMap中的应用

THANK YOU！

Machine Learning Engineer
机器学习工程师微专业

稀牛学院 網易 NETEASE