无监督学习

Fei Gao

gaofei@hdu.edu.cn

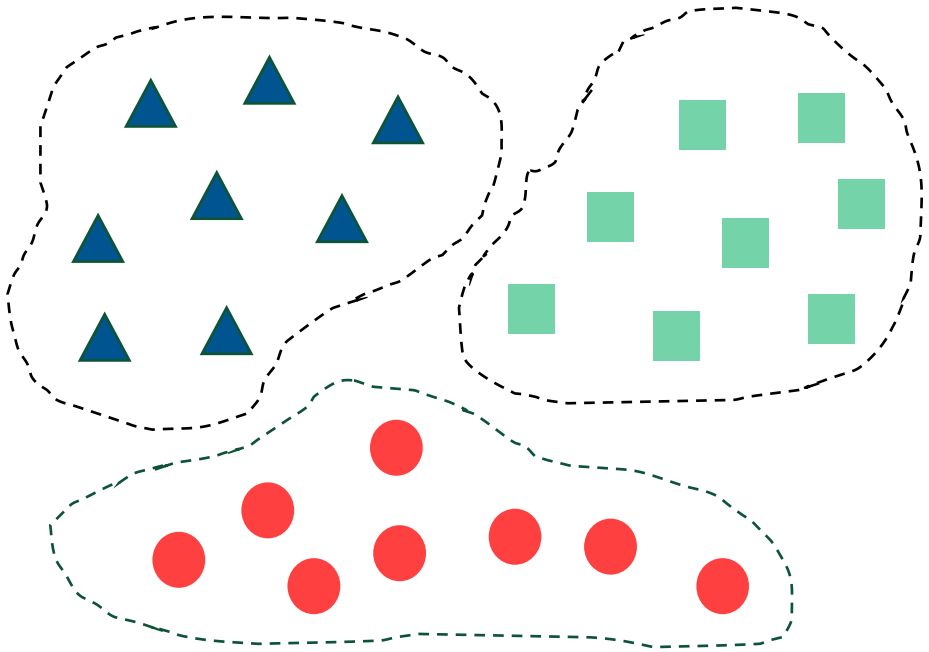
https://fei-hdu.github.io/

杭州電子科技大学
HANGZHOU DIANZI UNIVERSITY

目录

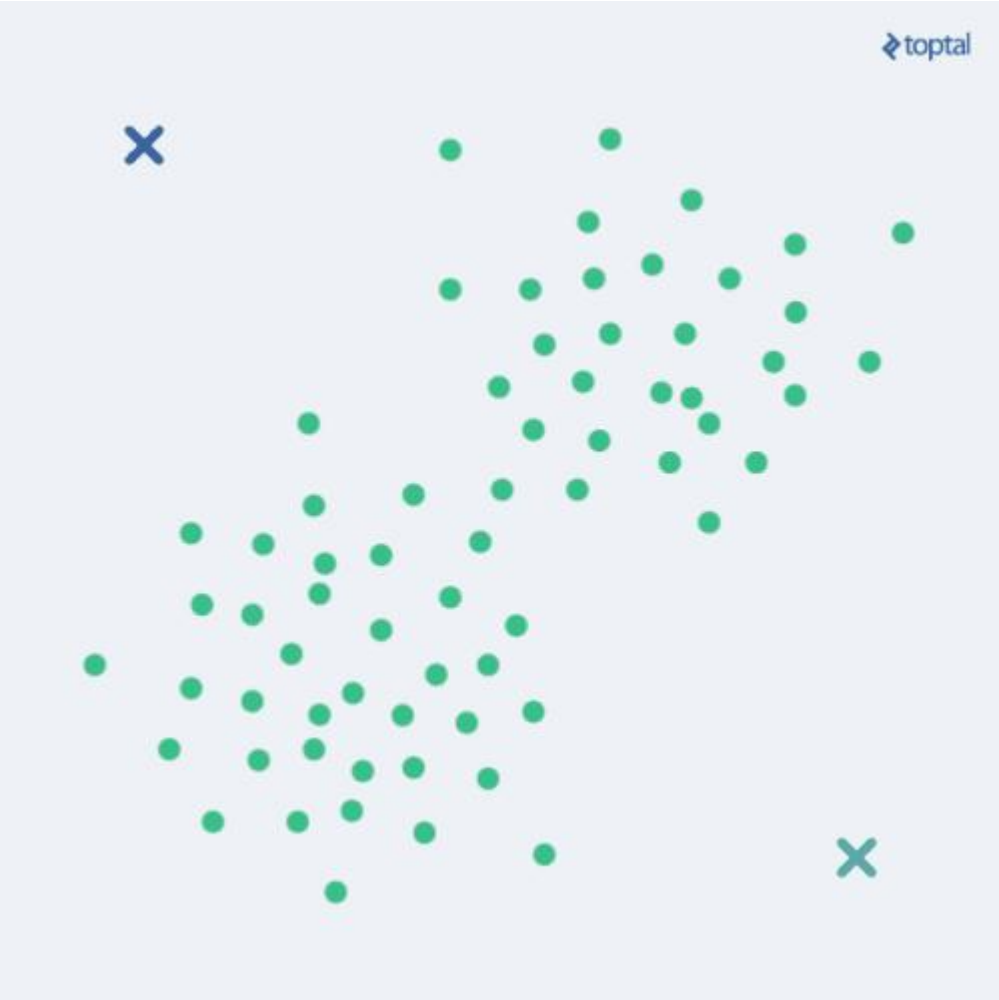
聚类：K-means

- **聚类目标：**
  - 将数据集中的样本划分为若干个通常不相交的子集（“簇”，cluster）.
  - 聚类既可以作为一个单独过程（用于找寻数据内在的分布结构），也可作为分类等其他学习任务的前驱过程.



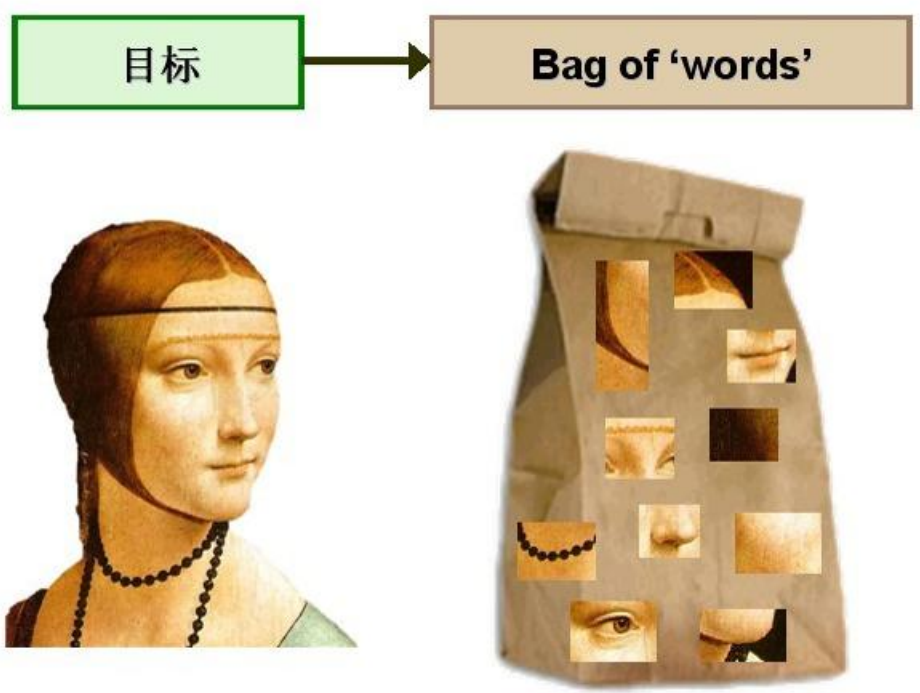
聚类：K-means

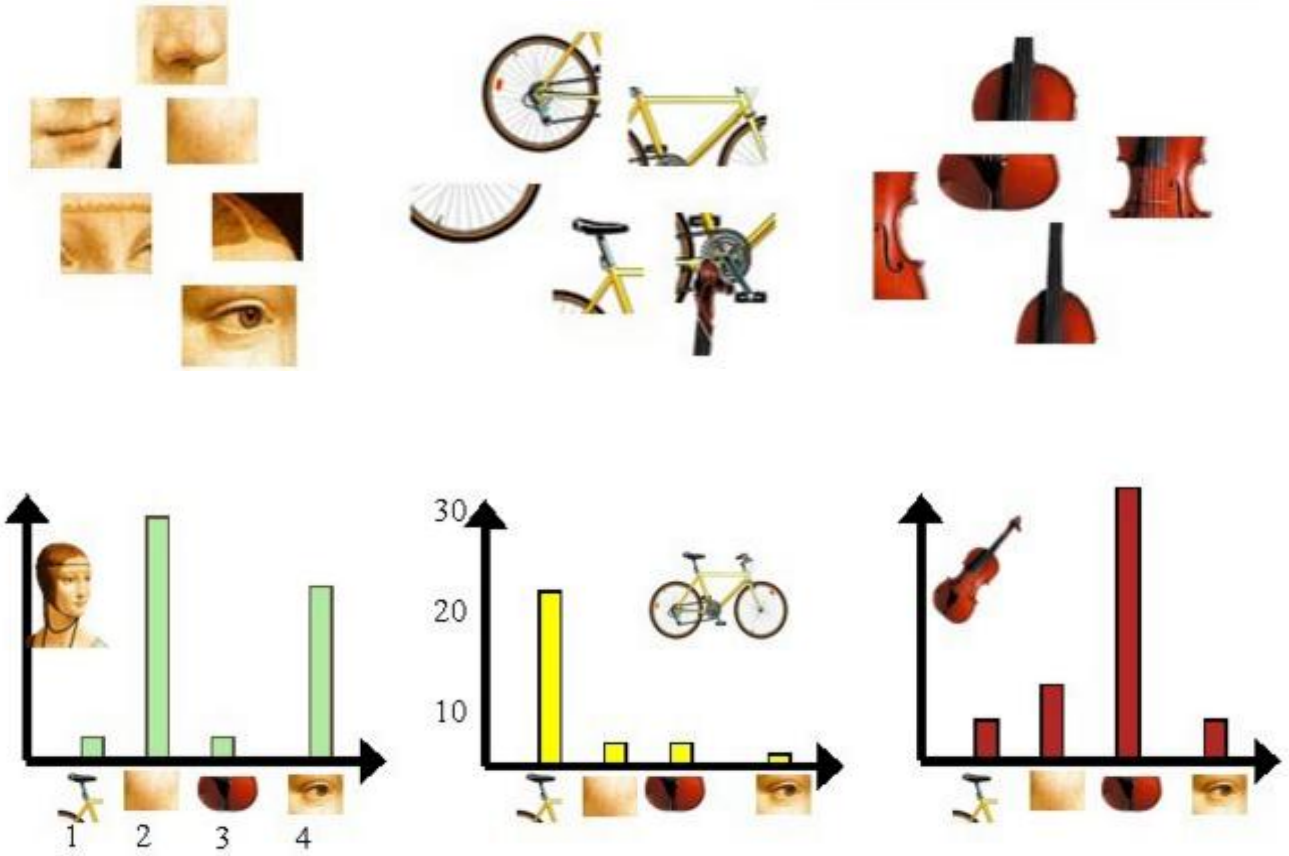
- **K均值聚类（K-means）**
  - 算法6.3



聚类：K-means

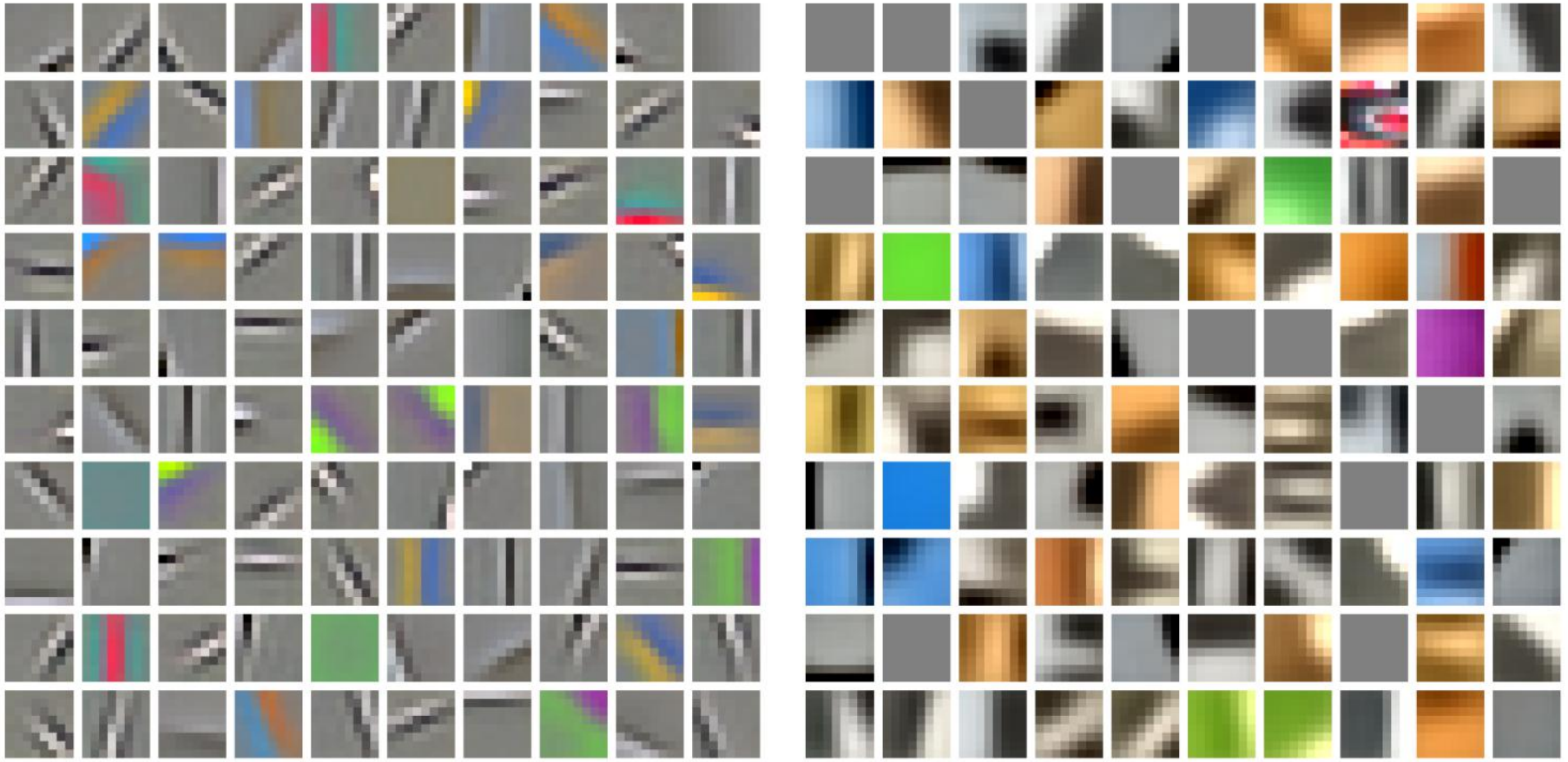
- **视觉词向量 Visual Bag-of-Words**





聚类：K-means

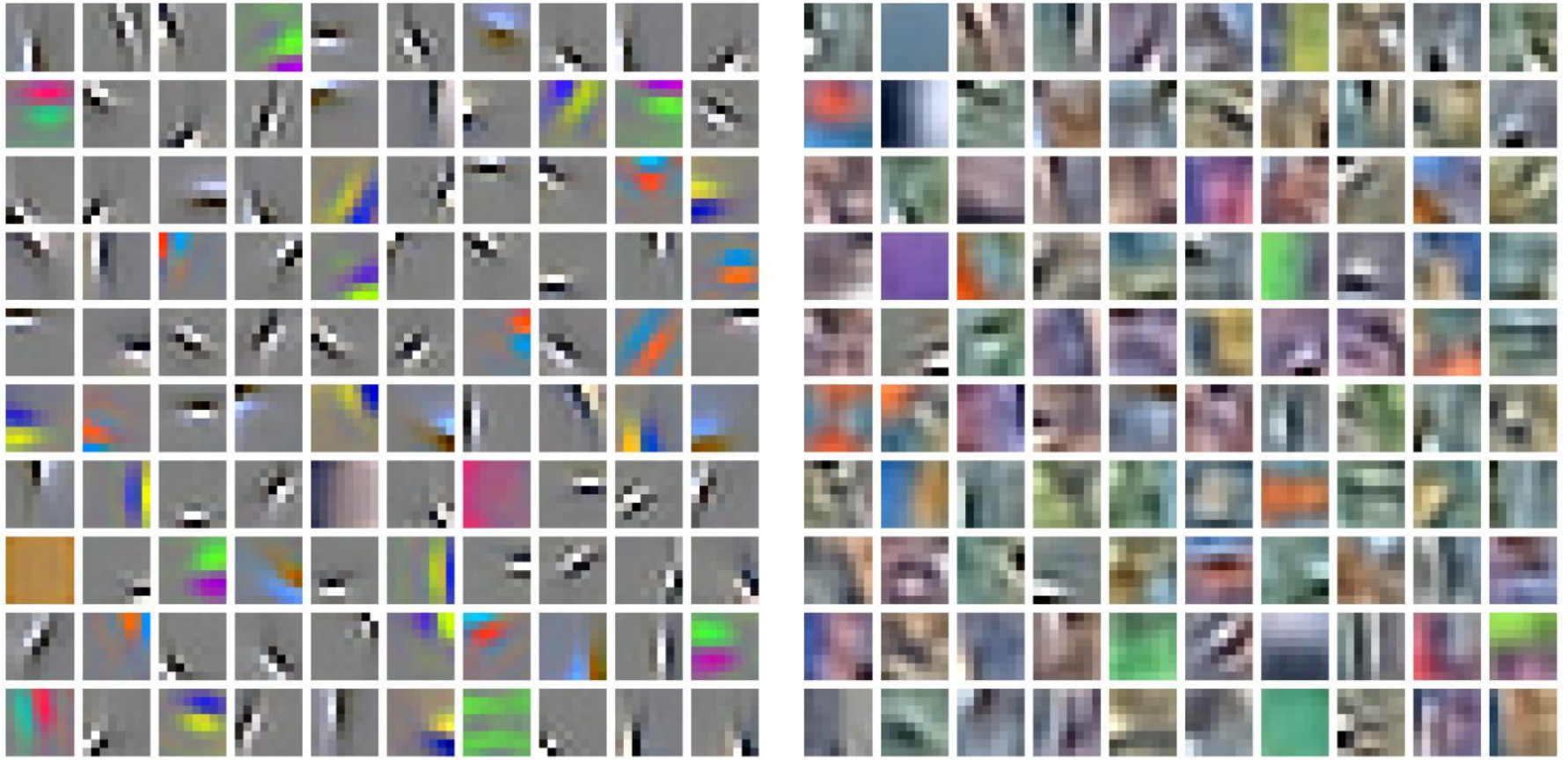
- **无监督特征学习（PCA+*k*-means）**



(a) K-means (with and without whitening)

聚类：K-means

- 无监督特征学习（PCA+*k*-means）

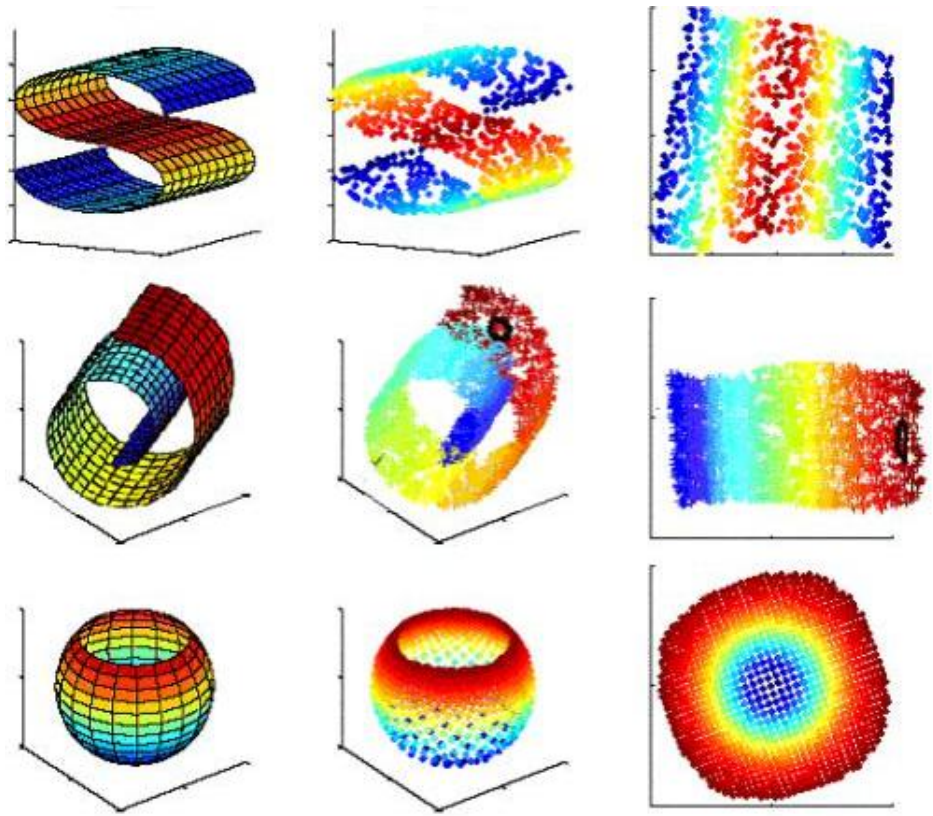


(c) Sparse Autoencoder (with and without whitening)

目录

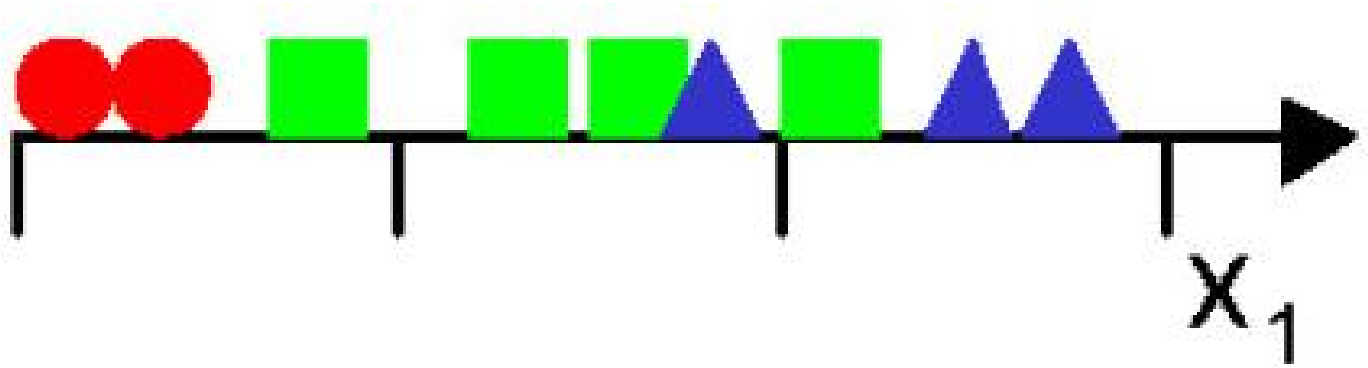
大纲

- k近邻学习
- **维数灾难**
- 主成分分析
- 流形学习
- 度量学习



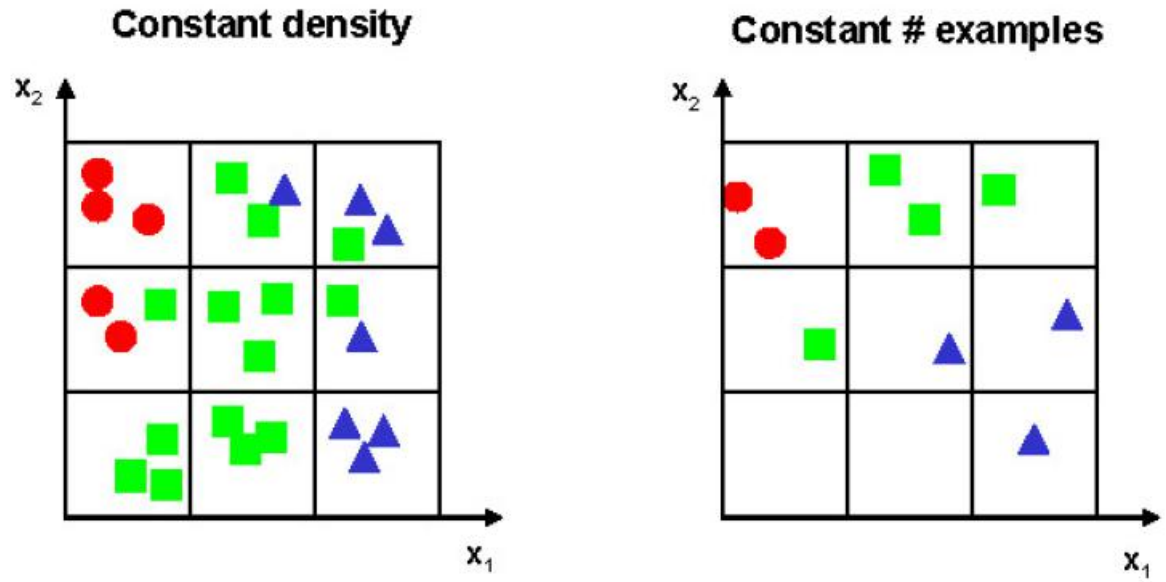
维数灾难 (curse of dimensionality)

- **用３类模式识别问题举例**
  - 一个简单的方法是：
    - 将特征空间划分为小bins
    - 计算每个bins里边每个样本对于每一类的ratio
    - 对于一个新的样本，找到它所在的bin然后将它划分到该bin里最主要的类里
    - 这个方法类似于k-nn算法，和桶算法类似
  - 比如我们



维数灾难 (curse of dimensionality)

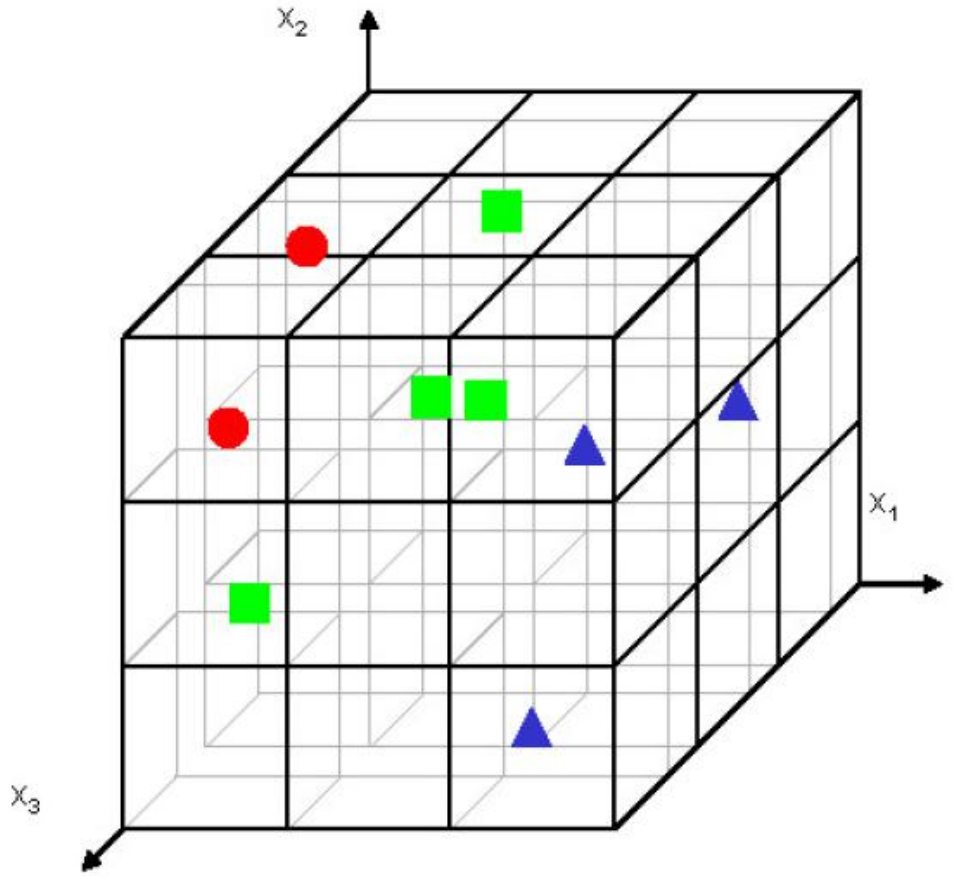
- 在一维我们会注意到类之间会有太多重叠,于是我们就决定引入第二个特征试图增加可分性.



在二维空间，如果
我们选择保留样本密度，那么样本点数量就从 9 激增至9*3=27
我们选择保留样本数量，那么 2 维的散点图就十分稀疏
该怎么抉择呢？再增加一个feature？

维数灾难 (curse of dimensionality)

- **在３维空间，事情变得更糟糕：**
  - bins的数量增加到27
  - 维持样本密度不变，样本点的数量就增加到了81
  - 维持样本点个数不变，那么3D散点图几乎就是空的



很明显，我们用相同的bins来划分样本空间的办法是十分低效率的

维数灾难 (curse of dimensionality)

- **Richard E.Bellman(发明动态规划的美国应用数学家) 在1961年提出了这个术语：**

The "Curse of dimensionality", is a term coined by Bellman to describe the problem caused by the exponential increase in volume associated with adding extra dimensions to a (mathematical) space.

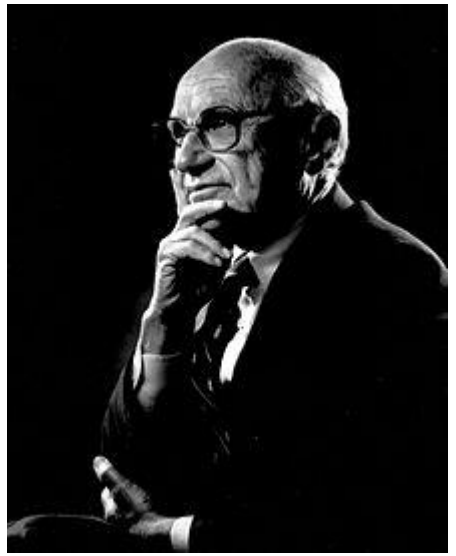
随着维数增加，（数学）空间的体积指数型增长。

维数灾难 (curse of dimensionality)

- **维度灾难更加深刻的含义**
  - 保持样本密度需要指数级增长的样本数量
  - 给定一个密度为N的样本和D维，需要的总样本数为 N^D
  - 密度估计的目标函数复杂性也随着维度增长呈指数增长

定义在高维空间的函数比定义在低维空间的函数复杂的多，并且那些复杂性是难以辨明的复杂．这意味着，为了更好的学习它，越复杂的函数需要越密集的样本点。

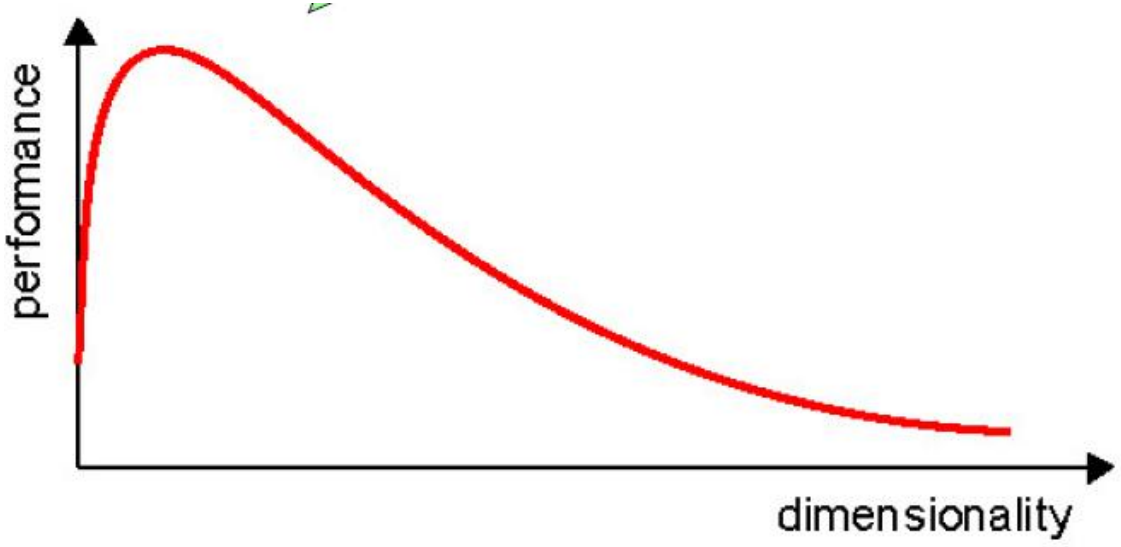
--by Friedman(famous for his friedman test)



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严重障碍，被称为“维数灾难”。

维数灾难 (curse of dimensionality)

- **我们如何打败维度灾难？**
  - 引入先验知识？
  - 提高目标函数的光滑性（例如光滑核方法），可使问题的推论性增强，甚至变为解析问题．然而，这需要大量的训练样本和训练时间．
  - 减少维度！（是否和前面加入特征矛盾）
- **在实践中，维度灾难意味着，给定一个样本大小，存在一个维数的上限，超过了这个上线，分类器的性能就会不升反降．**



维数灾难 (curse of dimensionality)

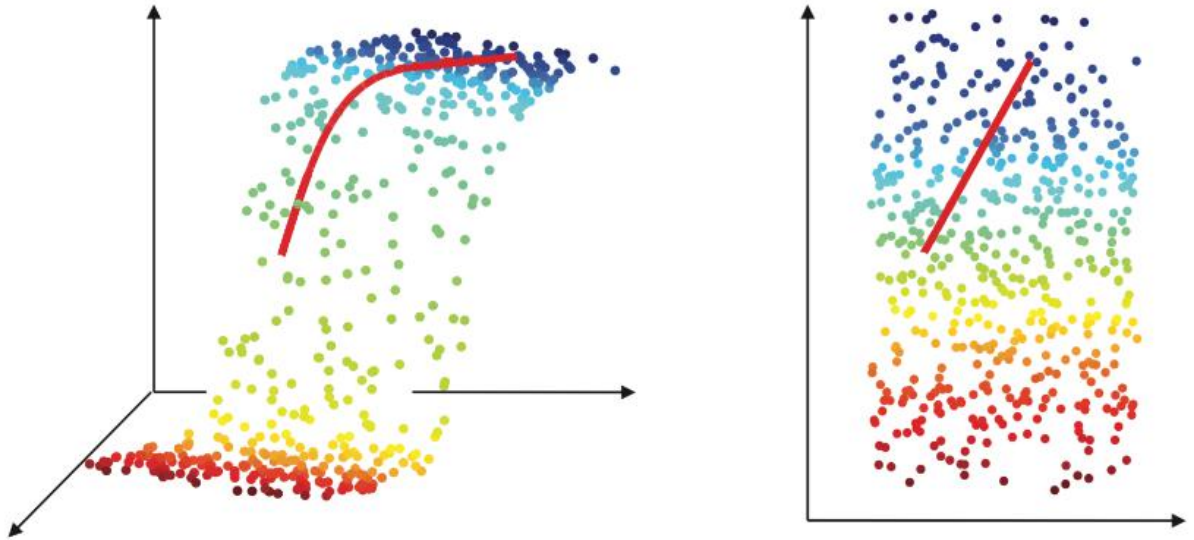
- **特征选择　vs. 特征提取**
  - Feature extraction:从现有的特征组合中生成一个特征子集
  - Feature selection:选择包含全部特征信息的子集（新的特征由原来的一维或者多维特征映射获得）

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{\text{linear feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \\ y_M \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_N \end{bmatrix}$$

- 给定一个特征空间 $x_i \in R^N$ 找出一个映射： $y = f(x) : R^N \to R^M$ 其中 $M < N$，这样，变换后的特征向量 $y_i \in R^M$ 保存了（大多数）原来特征空间 R^N 内的信息（或结构）
- 一个最优的映射的引入$y = f(x)$不会增加分类错误（就是对同一个算法说和原来的分类正确率一样）

低维嵌入

- **缓解维数灾难: 降维(dimension reduction)**
  - 通过某种数学变换，将原始高维属性空间转变为一个低维“子空间” (subspace)
    - 子空间: 密度大幅度提高，距离计算容易



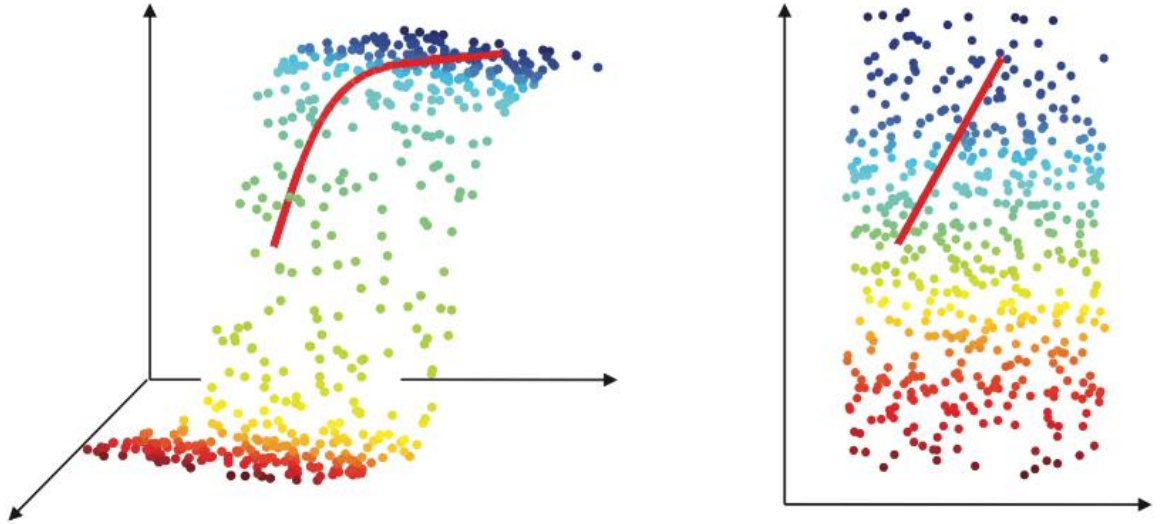
(a) 三维空间中观察到的样本点　　(b) 二维空间中的曲面

图 10.2　低维嵌入示意图

低维嵌入

- **缓解维数灾难的一个重要途径是降维(dimension reduction)**
  - 即通过某种数学变换，将原始高维属性空间转变为一个低维“子空间” (subspace)，在这个子空间中样本密度大幅度提高，距离计算也变得更为容易。

- 为什么能进行降维？
  - 数据样本虽然是高维的，但与学习任务密切相关的也许仅是某个低维分布，即高维空间中的一个低维“嵌入”(embedding)，因而可以对数据进行有效的降维。

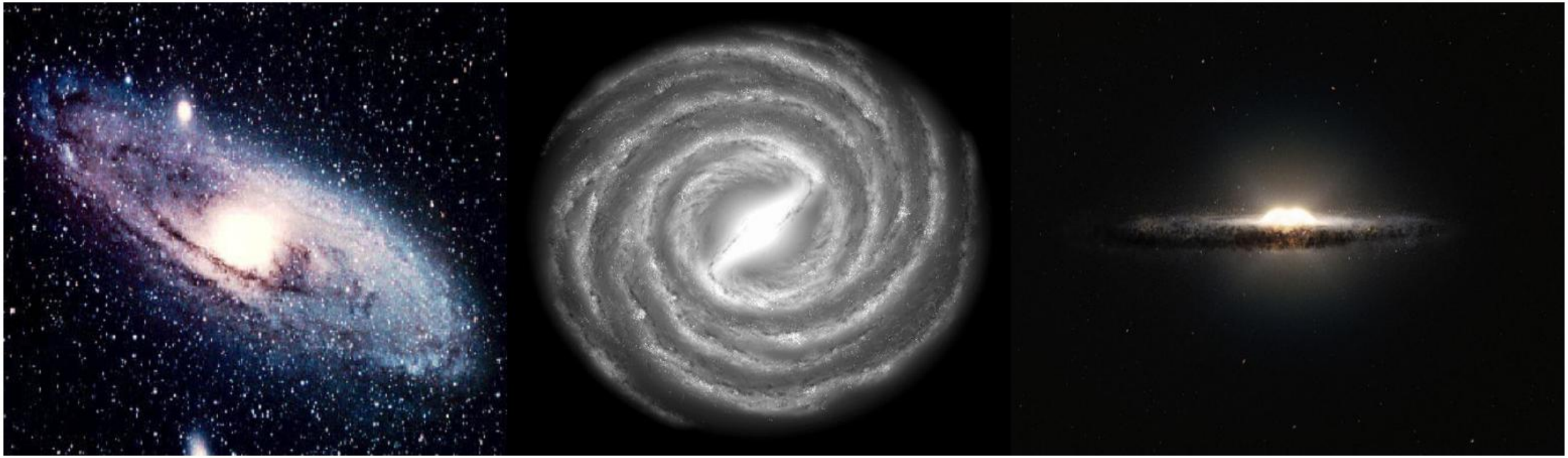


(a) 三维空间中观察到的样本点　　(b) 二维空间中的曲面

图 10.2　低维嵌入示意图

低维嵌入

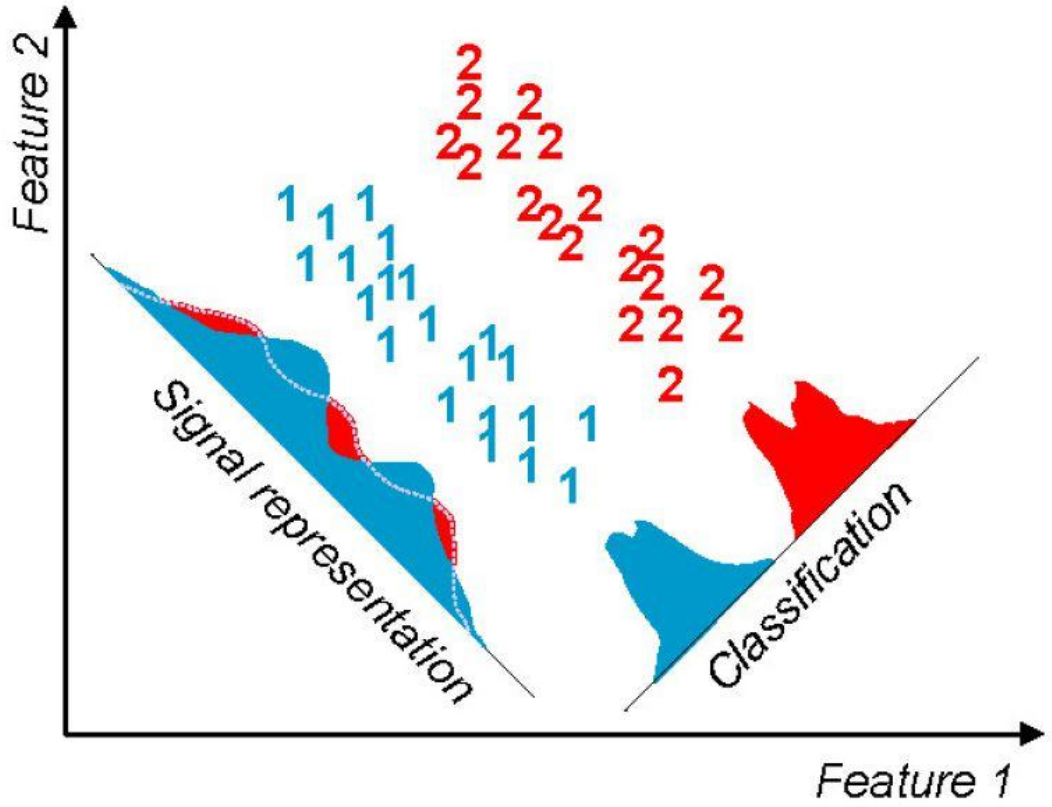
- **为什么能进行降维？**
  - 数据样本虽然是高维的，但与学习任务密切相关的也许仅是某个低维分布，即高维空间中的一个低维“嵌入” (embedding)。



在3维空间中，数据分别如银河系，每颗星代表一个数据点（图1）。从图3中可以看出，数据在其中一维上，方差较其它两维要小得多，即，在该方向上的特征值的模较小。那么在保留大部分信息的前提下，去掉这一维，得到的二维数据（图2）也是能较好反应原始数据的。

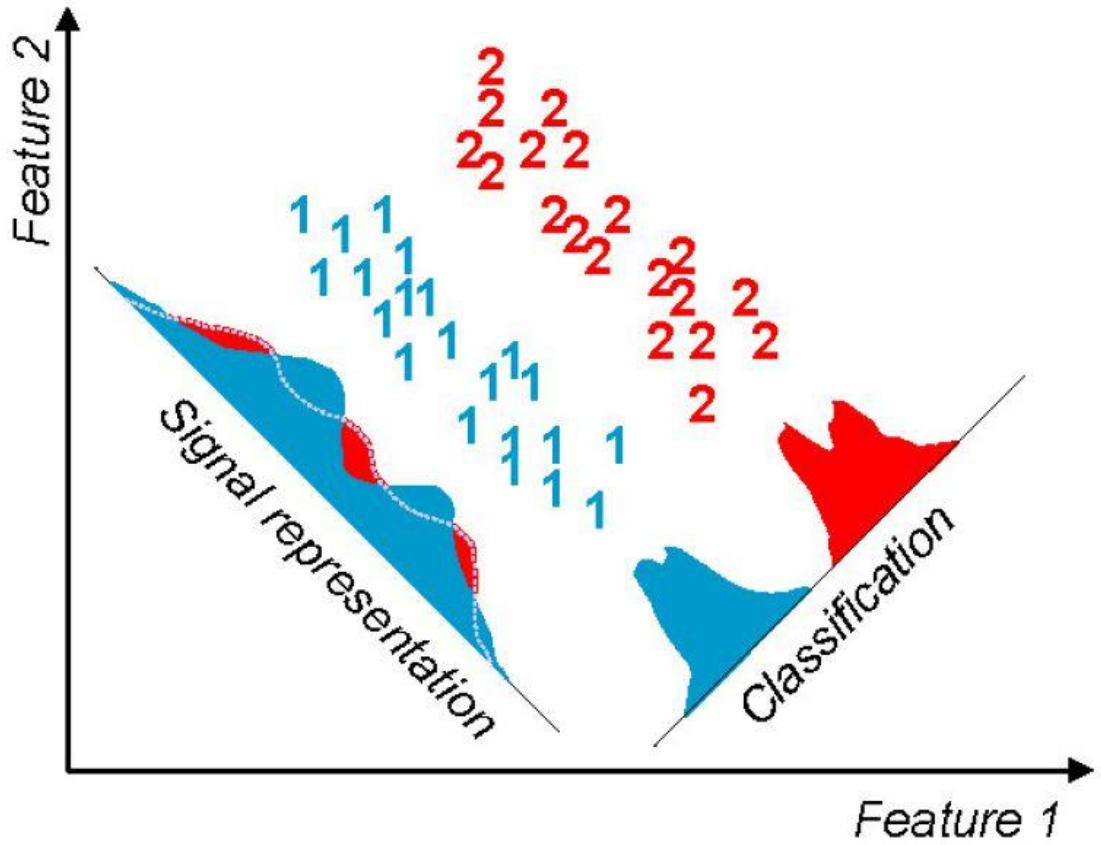
主成分分析

- **主成分分析(Principal Component Analysis, 简称PCA)**
  - 对于正交属性空间中的样本点，如何用一个超平面对所有样本进行恰当的表达？



主成分分析

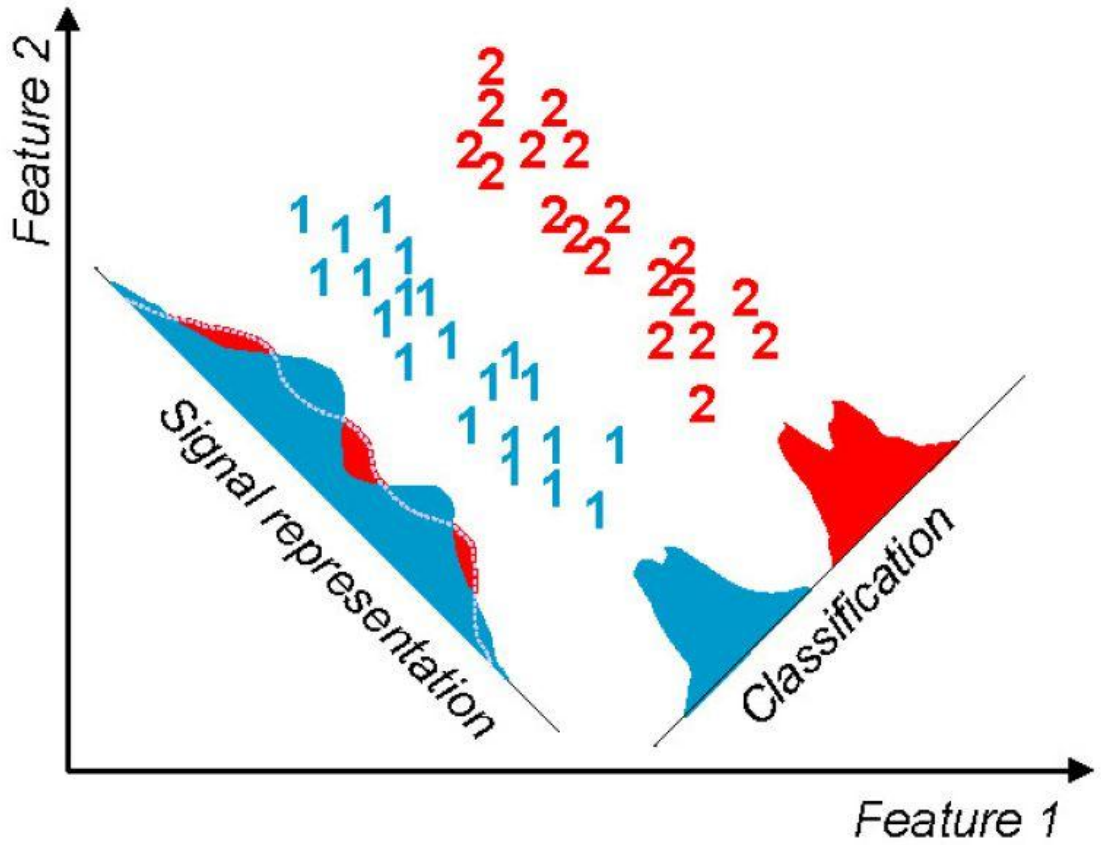
- **主成分分析 PCA**
  - 若存在这样的超平面，那么它大概应具有这样的性质：
    - 最近重构性：样本点到这个超平面的距离都足够近；
    - 最大可分性：样本点在这个超平面上的投影能尽可能分开。



主成分分析

- **主成分分析 PCA**
  - 若存在这样的超平面，那么它大概应具有这样的性质：
    - 最近重构性：样本点到这个超平面的距离都足够近；
    - 最大可分性：样本点在这个超平面上的投影能尽可能分开。

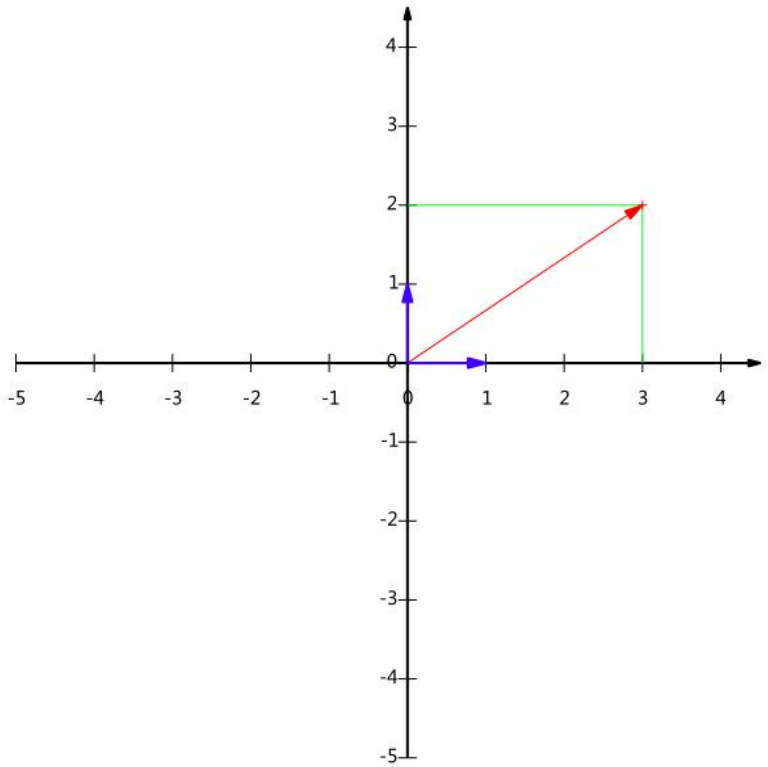
基于最近重构性和最大可分性，能分别得到主成分分析的两种等价推导。

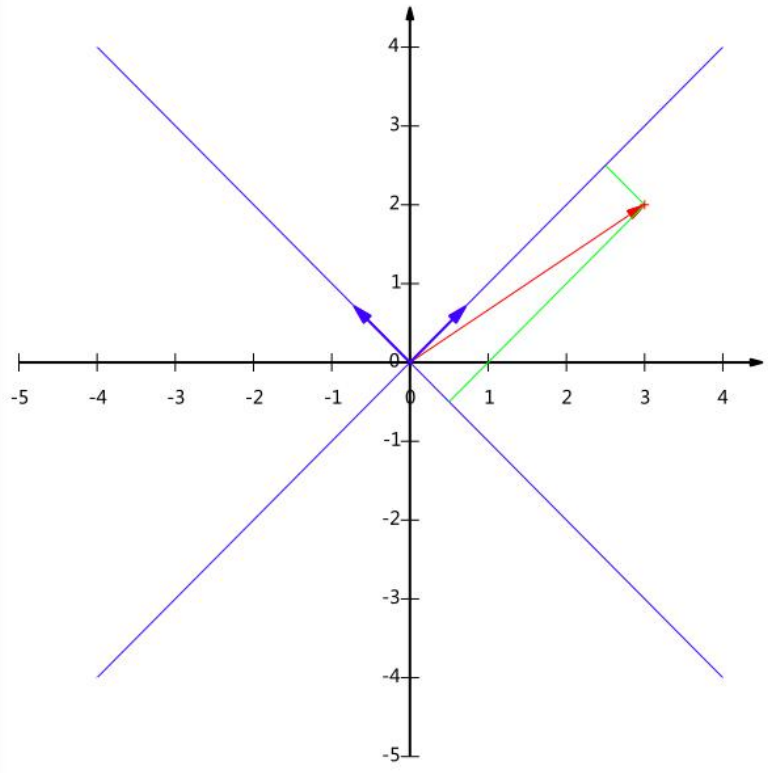


主成分分析

- **最近重构性**
  - 对样本进行中心化，$\sum_i \boldsymbol{x}_i = \mathbf{0}$ ， 再假定投影变换后得到的新坐标系为$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$， 其中$\boldsymbol{w}_i$ 是标准正交基向量，

$$||\boldsymbol{w}_i||_2 = 1, \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j = 0 (i \neq j).$$





主成分分析

- **最近重构性**
  - 对样本进行中心化，$\sum_i \boldsymbol{x}_i = \mathbf{0}$，再假定投影变换后得到的新坐标系为 $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$，其中 $\boldsymbol{w}_i$ 是标准正交基向量，

$$||\boldsymbol{w}_i||_2 = 1, \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j = 0 (i \neq j).$$

  - 若丢弃新坐标系中的部分坐标，即将维度降低到 $d' < d$，则样本点在低维坐标系中的投影是 $\boldsymbol{z}_i = \left(z_{i1}; z_{i2}; \ldots; z_{id'}\right)$，$z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$ 是 $\boldsymbol{x}_i$ 在低维坐标下第 j 维的坐标，
  - 若基于 $\boldsymbol{z}_i$ 来重构 $\boldsymbol{x}_i$，则会得到

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j.$$

主成分分析

- **最近重构性**
  - 考虑整个训练集，原样本点 $\boldsymbol{x}_i$与基于投影重构的样本点 $\hat{\boldsymbol{x}}_i$ 之间的距离为

$$\sum_{i=1}^{m}\left\|\sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j-\boldsymbol{x}_i\right\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i+\text{const}$$

$$\propto -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\mathbf{W}\right).$$

  - 根据最近重构性应最小化上式。考虑到 $\boldsymbol{w}_j$ 是标准正交基，$\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵，有

$$\begin{aligned}\min_{\mathbf{W}} \quad & -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})\\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.\end{aligned}$$

  - 这就是主成分分析的优化目标。

主成分分析

- **最大可分性**
  - 样本点 $\boldsymbol{x}_i$ 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。

- 方差

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu)^2$$

- 将每个字段的均值都化为0

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}a_i^2$$

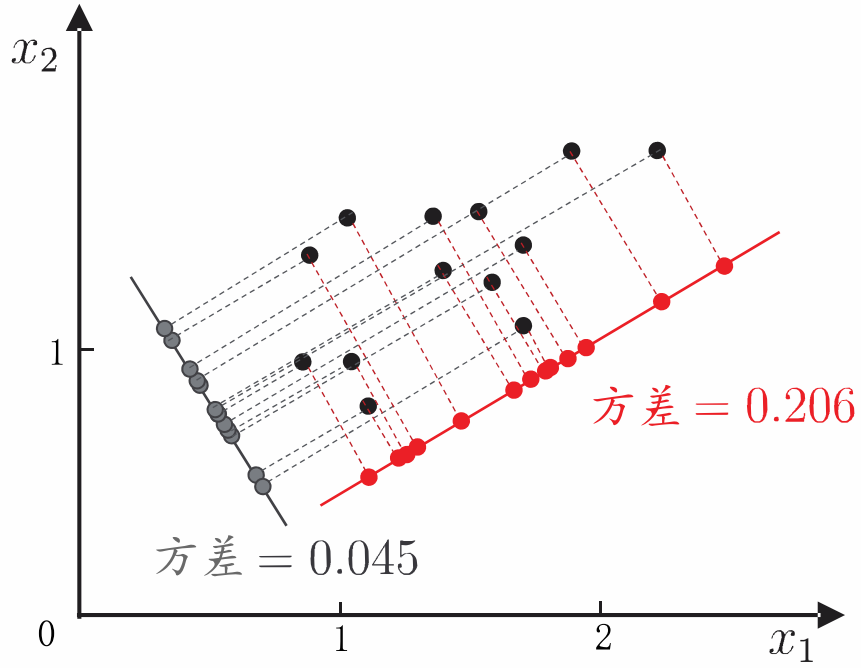
主成分分析

- **最大可分性**
  - 样本点 $\boldsymbol{x}_i$ 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。
  - 投影后样本点的方差是 $\sum_i \mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\mathbf{W}$ ，于是优化目标可写为

$$\begin{aligned} \max_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \end{aligned}$$

$$X = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ b_1 & b_2 & \cdots & b_m \end{pmatrix}$$

$$\frac{1}{m}XX^{\mathsf{T}} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} a_i^2 & \frac{1}{m}\sum_{i=1}^{m} a_i b_i \\ \frac{1}{m}\sum_{i=1}^{m} a_i b_i & \frac{1}{m}\sum_{i=1}^{m} b_i^2 \end{pmatrix}$$



主成分分析

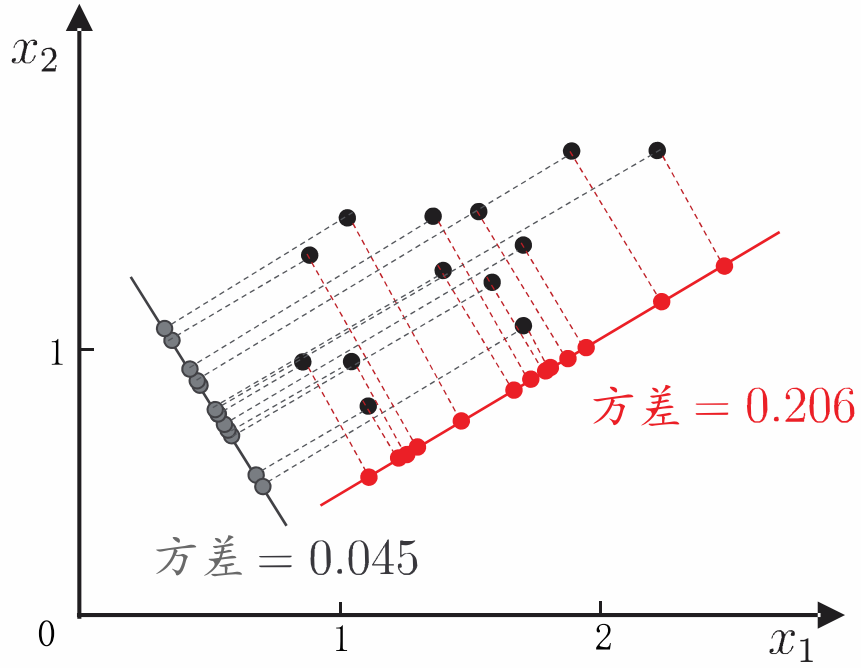
- **最大可分性**
  - 样本点 $\boldsymbol{x}_i$ 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。
  - 投影后样本点的方差是 $\sum_i \mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\mathbf{W}$ ，于是优化目标可写为

$$\begin{aligned} \max_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \end{aligned}$$

与

$$\begin{aligned} \min_{\mathbf{W}} \quad & -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \end{aligned}$$

等价。



主成分分析

- **PCA的求解**

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \qquad\qquad \min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \qquad\qquad \text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

  - 对优化式使用拉格朗日乘子法可得

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W} = \lambda\mathbf{W}.$$

  - 只需对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$进行特征值分解，并将求得的特征值排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ ，
  - 再取前 d' 个特征值对应的特征向量构成 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$ ，这就是主成分分析的解。

主成分分析

- **PCA算法**

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
低维空间维数 d'.

过程：

1: 对所有样本进行中心化: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;
4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

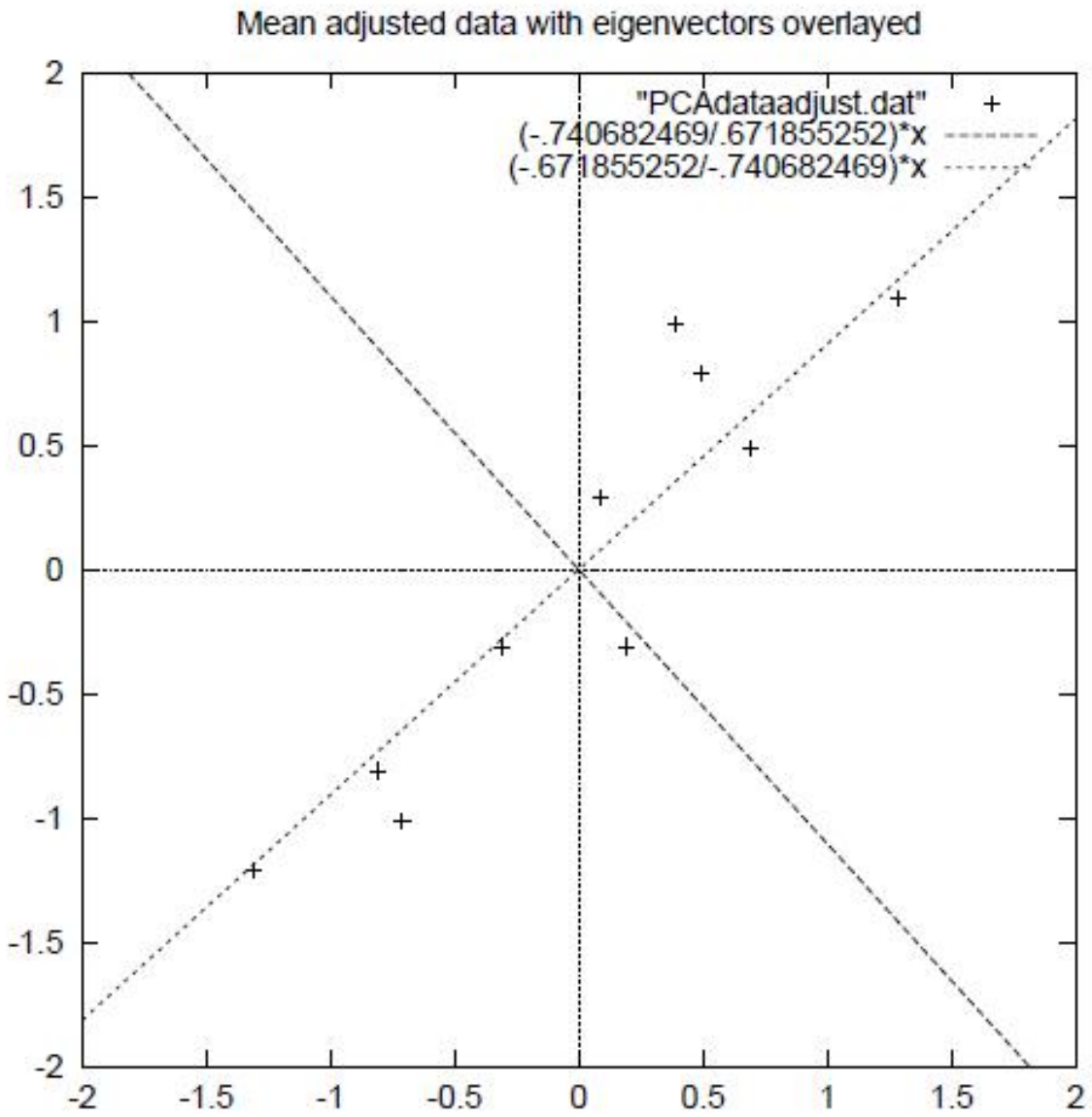
输出： 投影矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$.

主成分分析

正号表示预处理后的样本点，

斜着的两条线就分别是正交的特征向量（由于协方差矩阵是对称的，因此其特征向量正交），

最后一步的矩阵乘法就是将原始样本点分别往特征向量对应的轴上做投影。

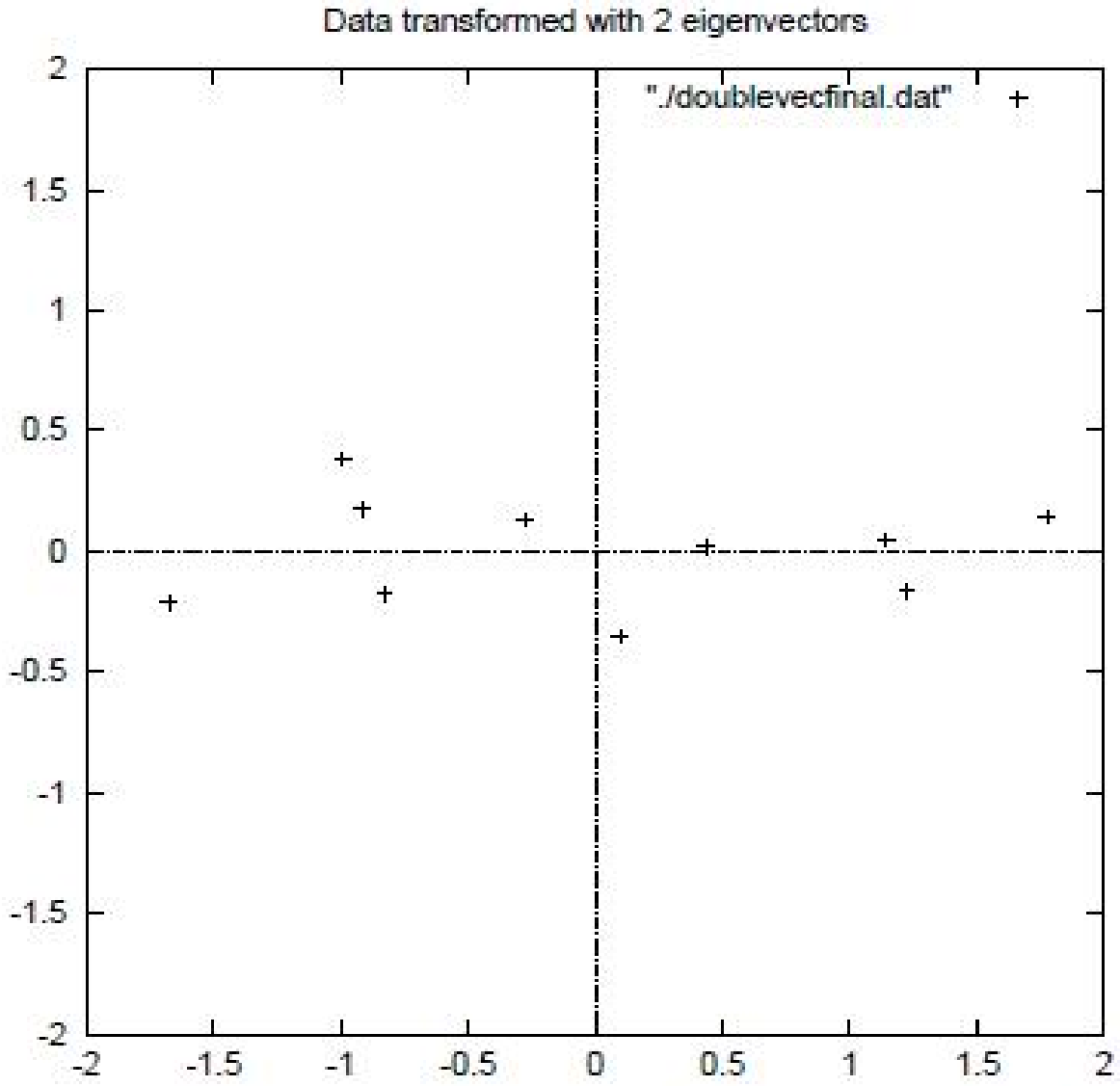


主成分分析

正号表示预处理后的样本点，

斜着的两条线就分别是正交的特征向量（由于协方差矩阵是对称的，因此其特征向量正交），

最后一步的矩阵乘法就是将原始样本点分别往特征向量对应的轴上做投影。



主成分分析

python中的numpy包计算均值和协方差：

```
1  import numpy as np
2  X = [[2, 0, -1.4],
3      [2.2, 0.2, -1.5],
4      [2.4, 0.1, -1],
5      [1.9, 0, -1.2]]
6  print(np.mean(X,axis=0))
7  print(np.cov(np.array(X).T))
```

eig函数返回特征值和特征向量的元组：

```
1  import numpy as np
2  w, v = np.linalg.eig(np.array([[1, -2], [2, -3]]))
3  print('特征值: {}\n特征向量: {}'.format(w,v))
```

主成分分析

```
import numpy as np
x = np.mat([[ 0.9, 2.4, 1.2, 0.5, 0.3, 1.8, 0.5, 0.3, 2.5, 1.3],
            [ 1, 2.6, 1.7, 0.7, 0.7, 1.4, 0.6, 0.6, 2.6, 1.1]])
x = x.T
T = x - x.mean(axis=0)
C = np.cov(x.T)
w,v = np.linalg.eig(C)
v_ = np.mat(v[:,0])     #每个特征值对应的是特征矩阵的每个列向量
v_ = v_.T       #默认以行向量保存，转换成公式中的列向量形式
y = T * v_
print(y)
```

主成分分析

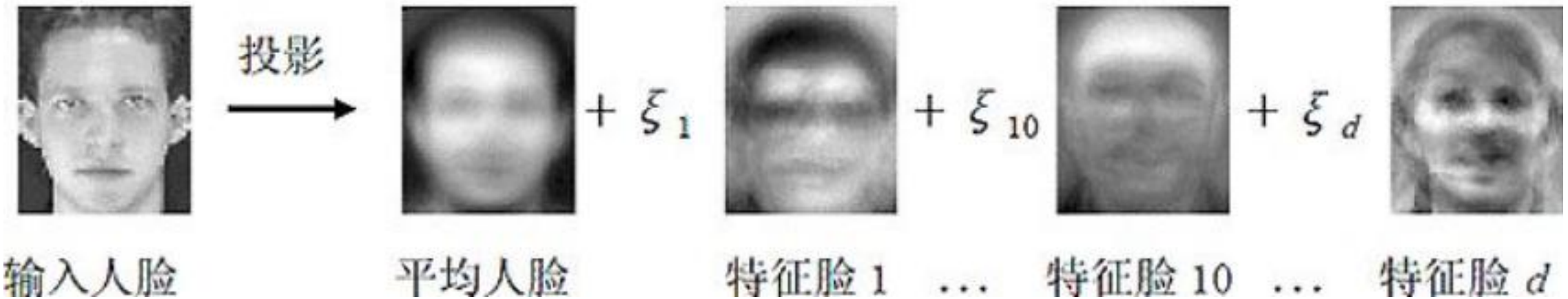
- 图像PCA降维（保留50个特征值），然后重构

主成分分析

- **特征脸：人脸图像PCA降维，不同特征值重构**

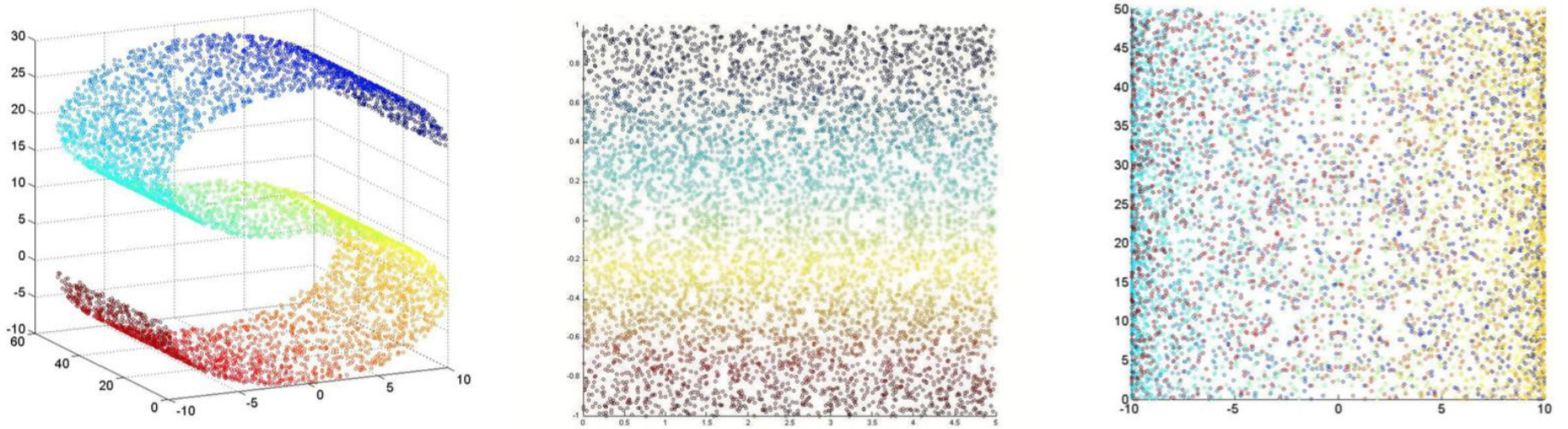
主成分分析

- **特征脸**



核化线性降维：核化主成分分析 (Kernelized PCA, KPCA)

- 线性降维方法假设从高维空间到低维空间的函数映射是线性的，然而，在不少现实任务中，可能需要非线性映射才能找到恰当的低维嵌入：



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

图 10.6　三维空间中观察到的 3000 个样本点，是从本真二维空间中矩形区域采样后以 S 形曲面嵌入，此情形下线性降维会丢失低维结构. 图中数据点的染色显示出低维空间的结构.

核化线性降维

- **核化主成分分析 (Kernelized PCA, 简称KPCA)**
  - 假定 $\boldsymbol{z}_i$ 是由原始属性空间中的样本点 $\boldsymbol{\phi}$ 通过映射 $\boldsymbol{x}_i$产生，即

$$\boldsymbol{z}_i = \boldsymbol{\phi}(\boldsymbol{x}_i),\ i = 1, 2, \ldots, m.$$

$$\mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{\alpha}_i$$

  - 若 $\boldsymbol{\phi}$ 能被显式表达出来，则通过它将样本映射至高维空间，再在特征空间中实施PCA即可，即有

$$\left(\sum_{i=1}^{m} \boldsymbol{\phi}(\boldsymbol{x}_i) \boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \right) \mathbf{W} = \lambda \mathbf{W}.$$

  - 并且

$$\mathbf{W} = \sum_{i=1}^{m} \boldsymbol{\phi}(\boldsymbol{x}_i) \boldsymbol{\alpha}_i.$$

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_j).$$

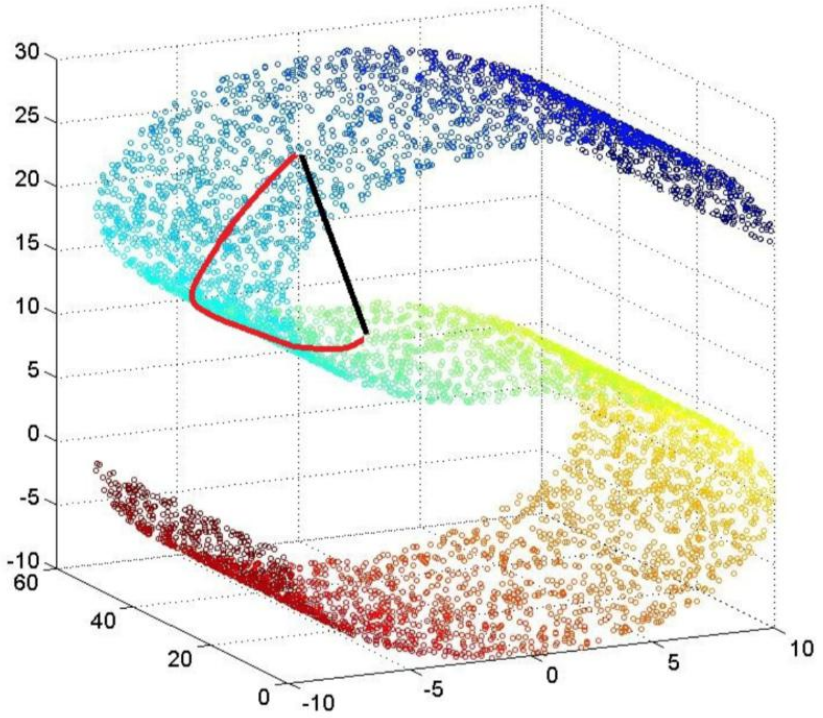
$$\mathbf{K}\mathbf{A} = \lambda \mathbf{A}.$$

$$(\mathbf{K})_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \mathbf{A} = (\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \ldots; \boldsymbol{\alpha}_m).$$

目录

流形学习

- **流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”是在局部与欧氏空间同胚的空间，换言之，它在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。**



(a) 测地线距离与高维直线距离

- 当维数被降至二维或三维时，能对数据进行可视化展示，因此流形学习也可被用于可视化。
  - t-SNE

流形学习

- 流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”是在局部与欧氏空间同胚的空间，换言之，它在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。

- 若低维流形嵌入到高维空间中，则数据样本在高维空间的分布虽然看上去非常复杂，但在局部上仍具有欧氏空间的性质，因此，可以容易地在局部建立降维映射关系，然后再设法将局部映射关系推广到全局。

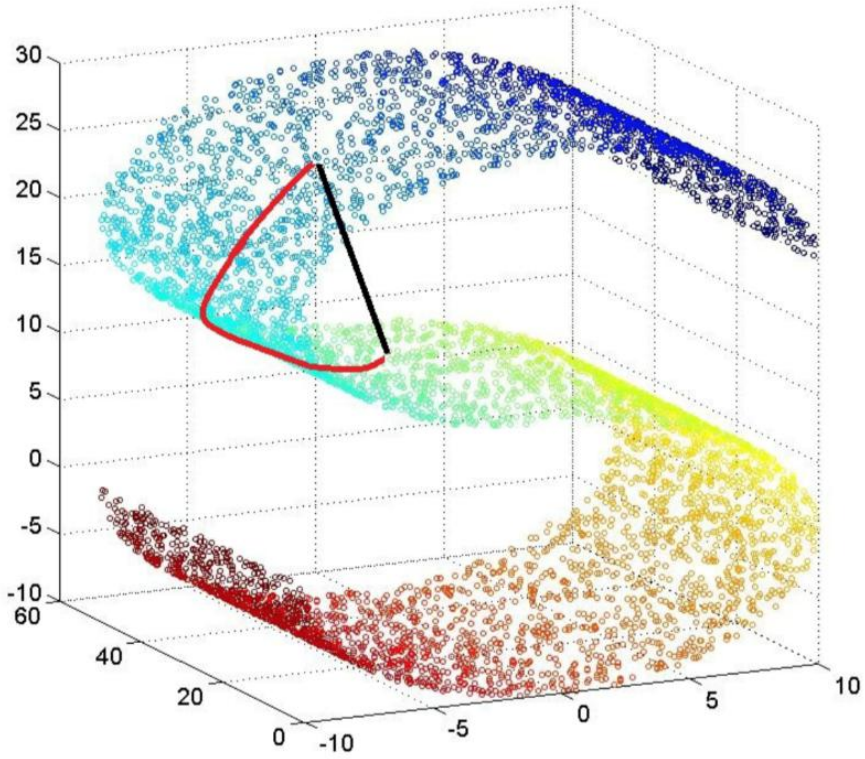
- 当维数被降至二维或三维时，能对数据进行可视化展示，因此流形学习也可被用于可视化。
  - t-SNE

流形学习

- **等度量映射(Isometric Mapping，Isomap)**

  - 低维流形嵌入到高维空间之后，直接在高维空间中计算直线距离具有误导性，因为高维空间中的直线距离在低维嵌入流形上不可达。

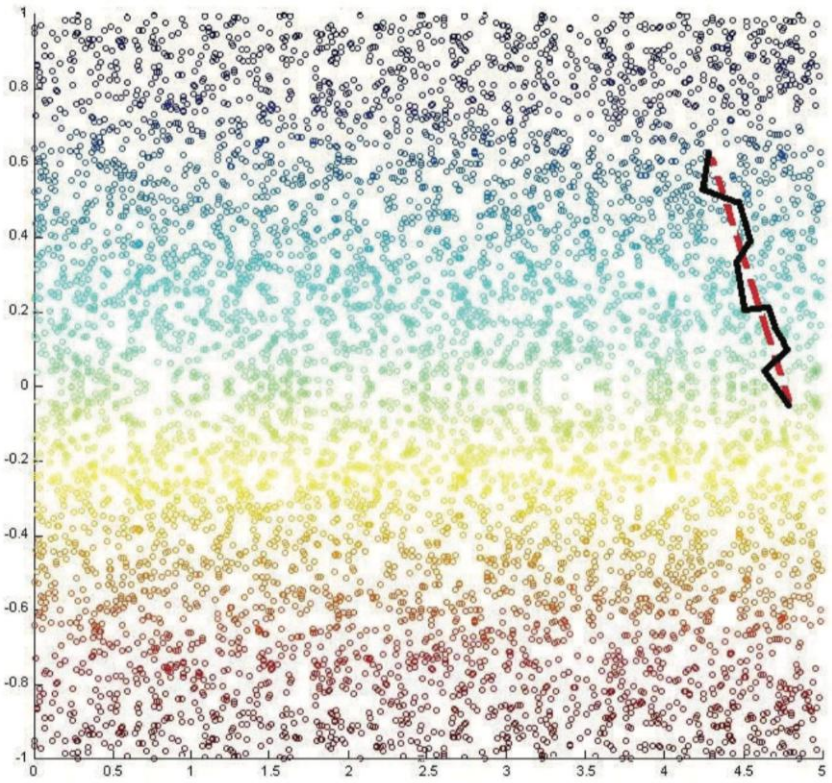
  - 低维嵌入流形上两点间的本真距离是“测地线”(geodesic)距离。



(a) 测地线距离与高维直线距离

流形学习

- **等度量映射(Isometric Mapping，Isomap)**
  - 测地线距离的计算：
    - 利用流形在局部上与欧氏空间同胚这个性质，对每个点基于欧氏距离找出其近邻点，
    - 然后就能建立一个近邻连接图，图种近邻点之间存在连接，而非近邻点之间不存在连接，
    - 于是，计算两点之间测地线距离的问题，就转变为计算近邻连接图上两点之间的最短路径问题。
  - 最短路径的计算可通过Dijkstra算法或Floyd算法实现。
  - 得到距离后可通过多维缩放方法获得样本点在低维空间中的坐标。



(b) 测地线距离与近邻距离

流形学习

- **等度量映射(Isometric Mapping，Isomap)**

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程:

1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离, 与其他点的距离设置为无穷大;
4: **end for**
5: 调用最短路径算法计算任意两样本点之间的距离 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
6: 将 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 作为 MDS 算法的输入;
7: **return** MDS 算法的输出

输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.8 Isomap 算法

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**

Nonlinear Dimensionality Reduction by Locally Linear Embedding

Sam T. Roweis[1] and Lawrence K. Saul[2]

Science, 2000,

290(5500):2323-2326.

被引量 1.3万 Baidu学术

流形学习

AlphaGo

DeepMind, 2016

Lecun Y, Bengio Y, Hinton G.

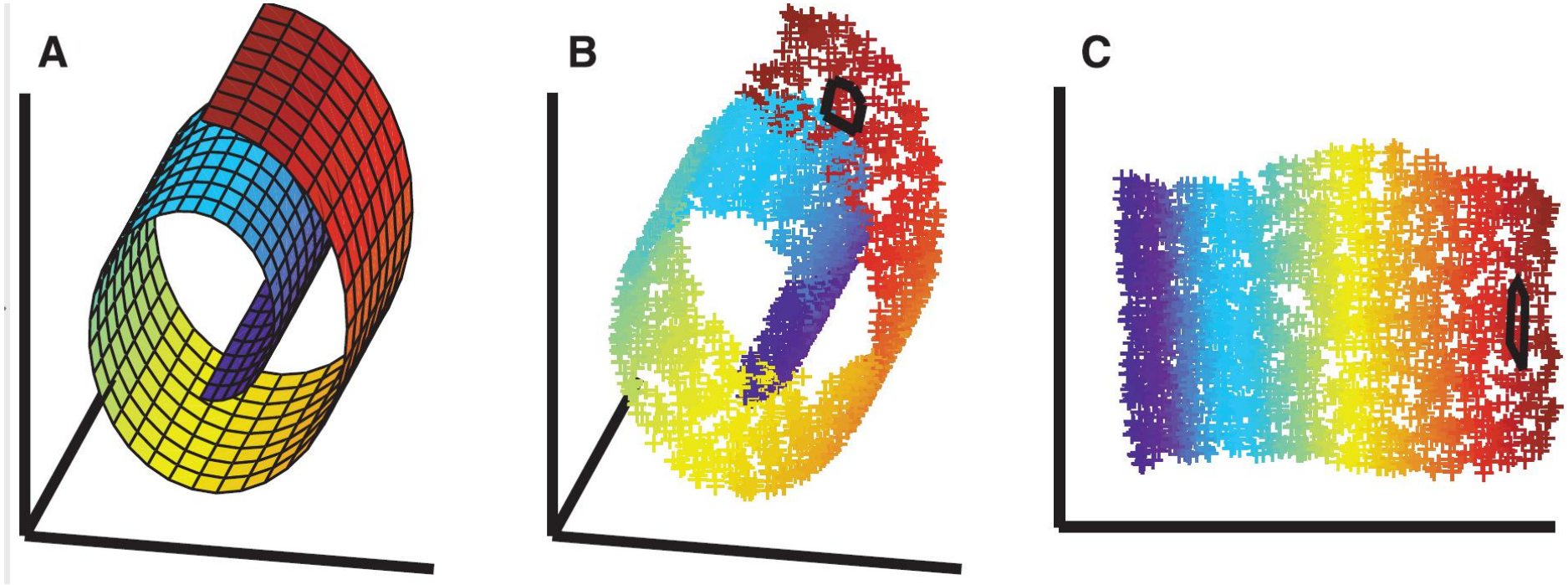
Deep learning[J].

Nature, 2015,

521(7553):436.

流形学习

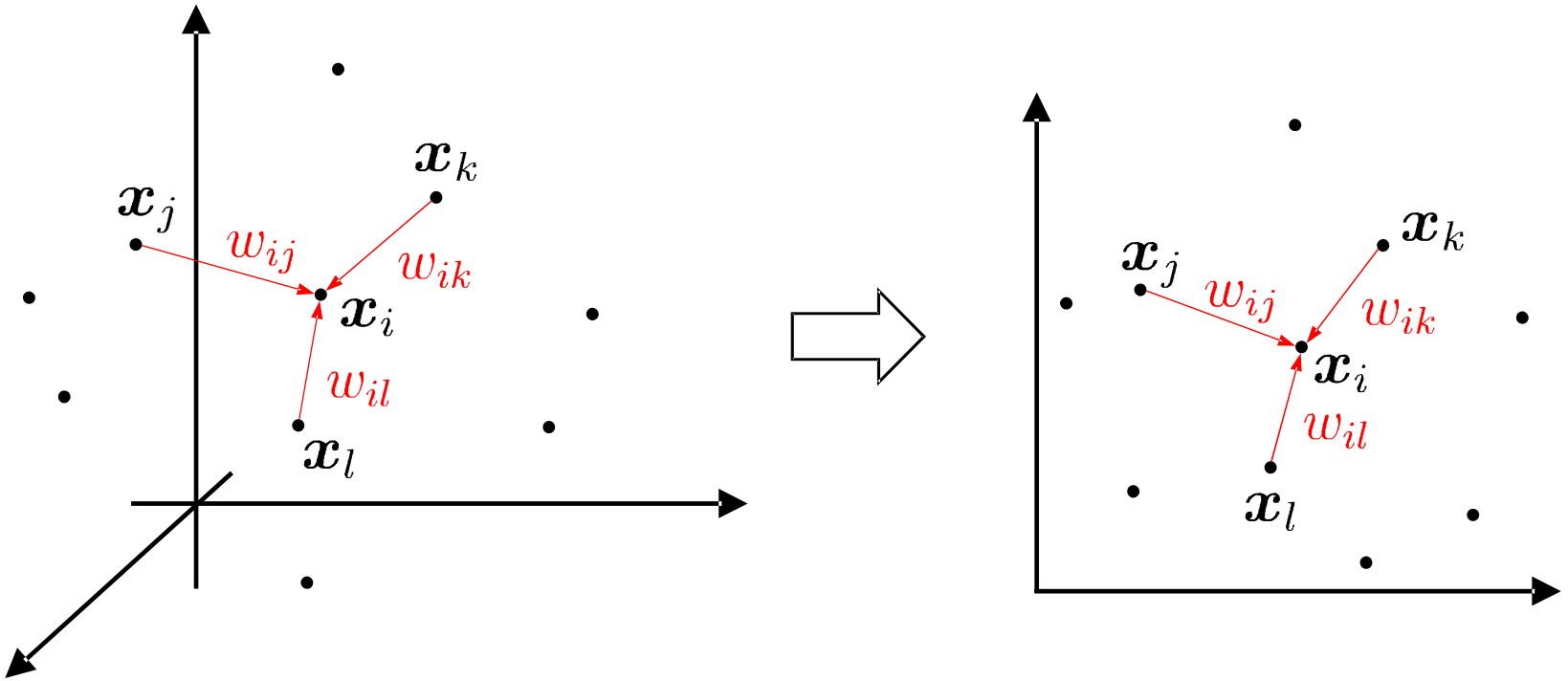
- **局部线性嵌入 (Locally Linear Embedding，LLE)**
  - 局部线性嵌入试图保持邻域内的线性关系，并使得该线性关系在降维后的空间中继续保持。



The problem of nonlinear dimensionality reduction, as illustrated (10) for three-dimensional data (B) sampled from a two-dimensional manifold (A).

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**
  - 局部线性嵌入试图保持邻域内的线性关系，并使得该线性关系在降维后的空间中继续保持。



$$\boldsymbol{x}_i = w_{ij}\boldsymbol{x}_j + w_{ik}\boldsymbol{x}_k + w_{il}\boldsymbol{x}_l$$

流形学习

- **LLE**

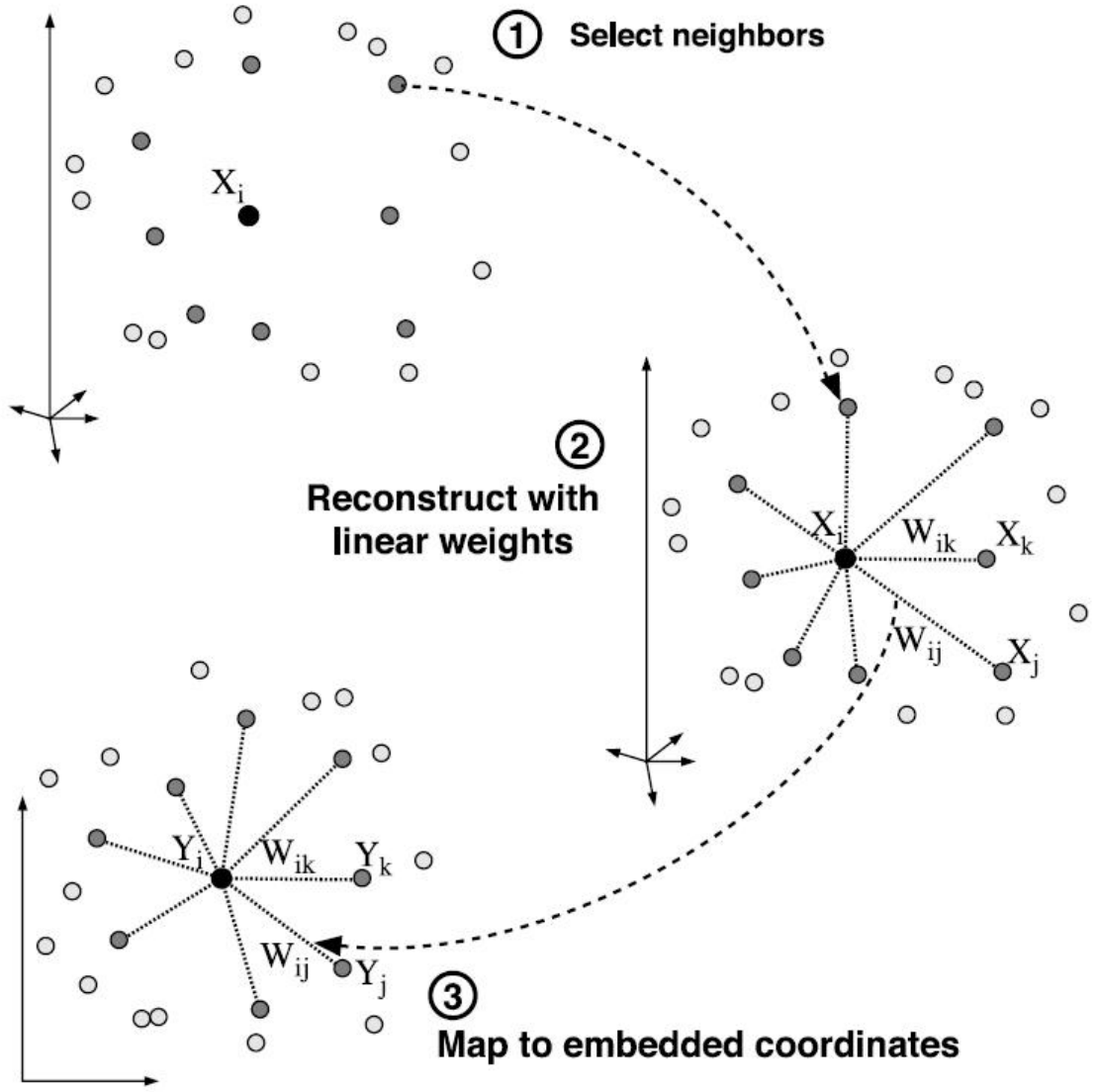
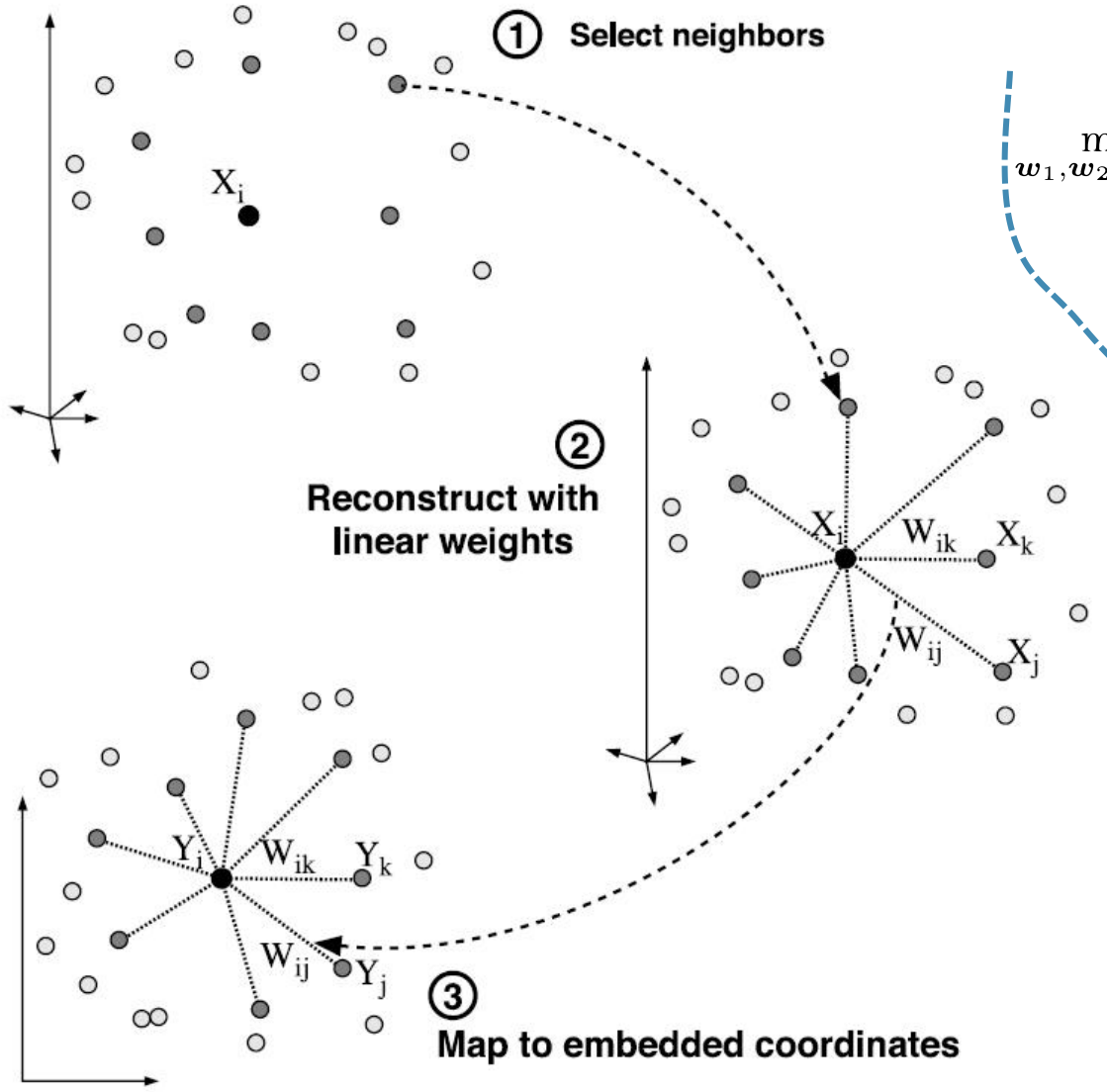


Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.

流形学习

- **LLE**

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2$$

$$\text{s.t.} \sum_{j \in Q_i} w_{ij} = 1,$$

$$w_{ij} = \frac{\sum_{k \in Q_i} C_{jk}^{-1}}{\sum_{l,s \in Q_i} C_{ls}^{-1}}.$$

流形学习

- **LLE**

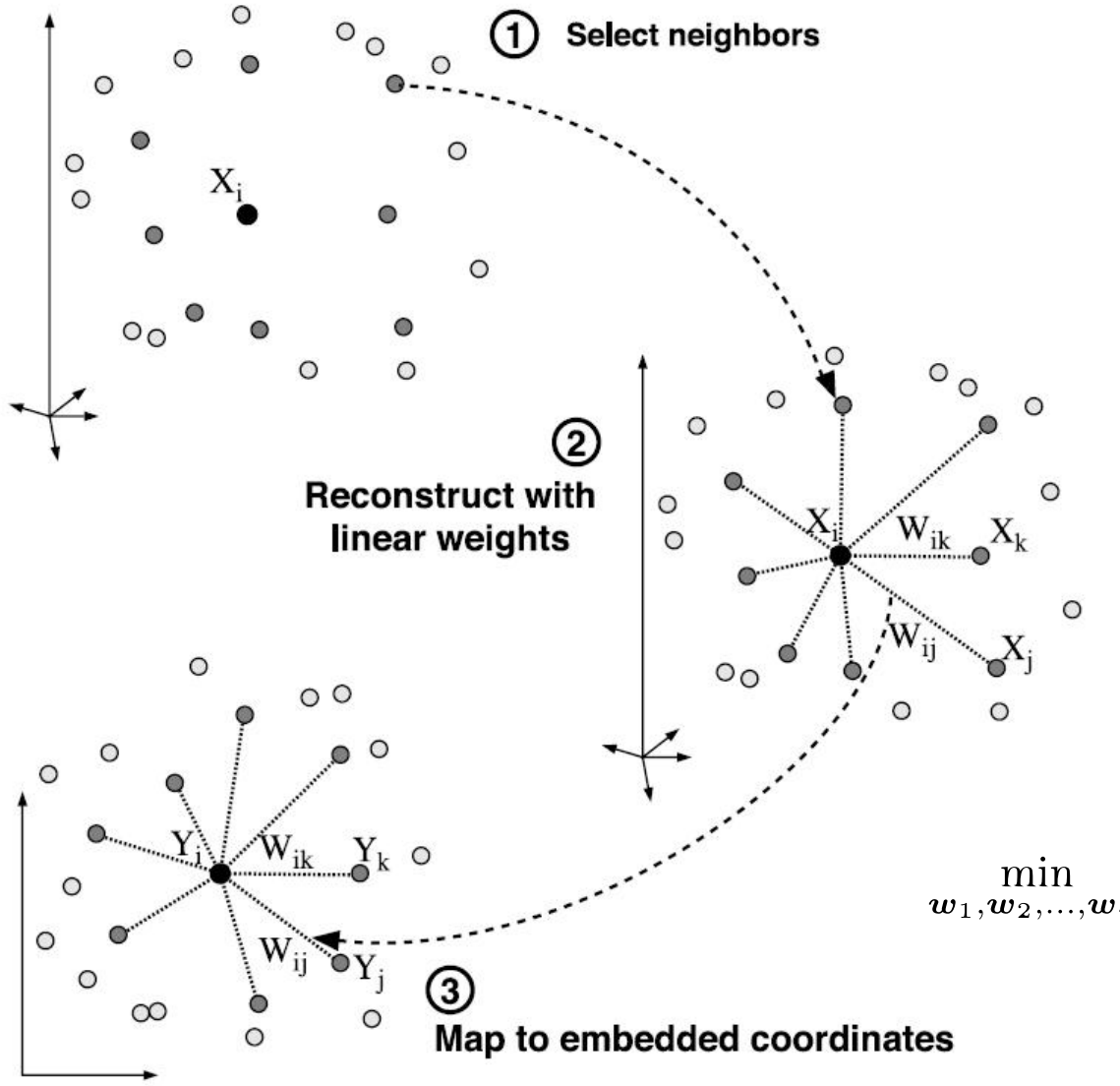


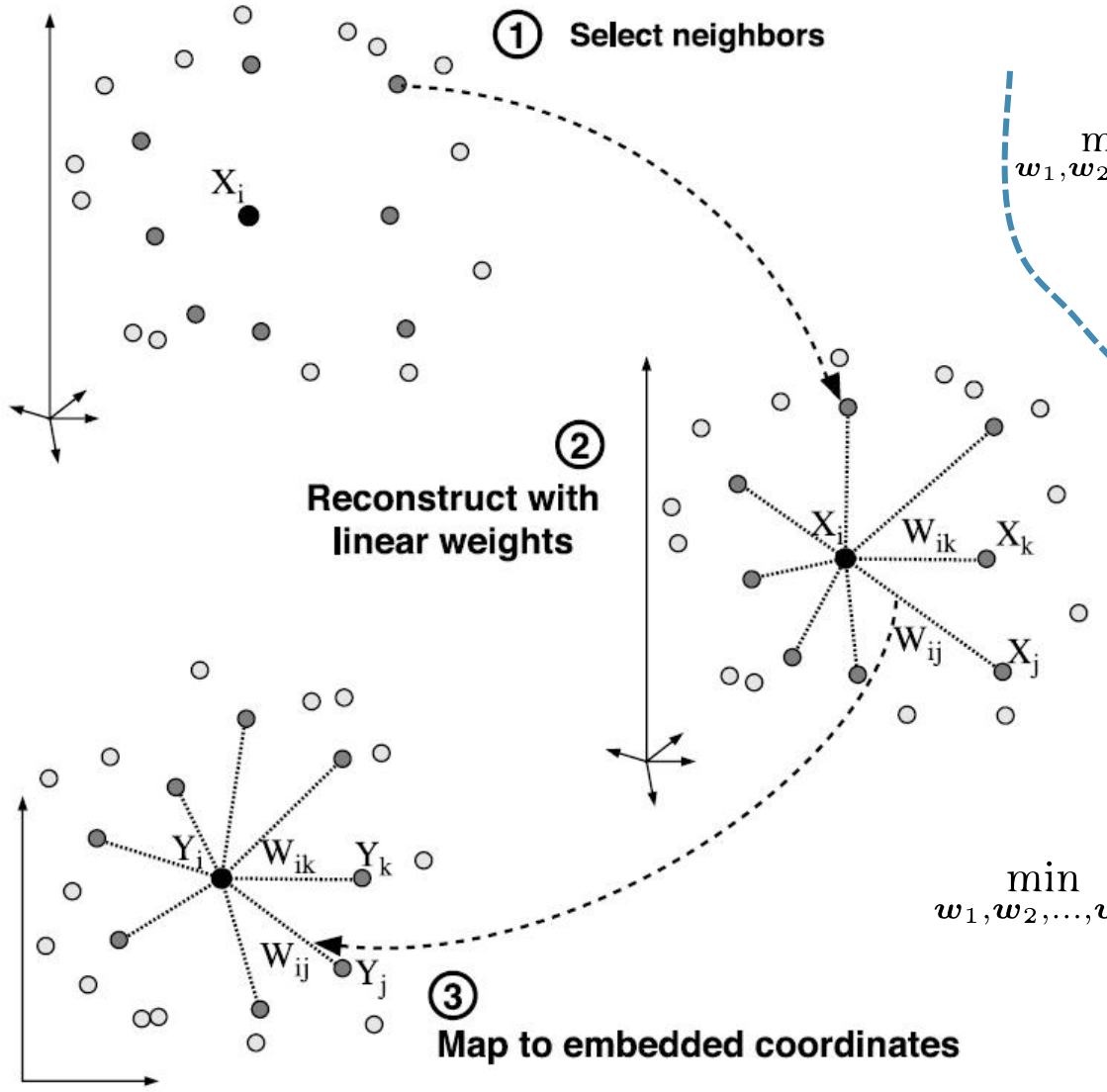
Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2$$

流形学习

- **LLE**

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2$$

$$\text{s.t.} \sum_{j \in Q_i} w_{ij} = 1,$$

$$w_{ij} = \frac{\sum_{k \in Q_i} C_{jk}^{-1}}{\sum_{l,s \in Q_i} C_{ls}^{-1}}.$$

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2$$

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**
  - LLE先为每个样本 $\boldsymbol{x}_i$ 找到其近邻下标集合 Q_i，然后计算出基于 Q_i的中的样本点对 $\boldsymbol{x}_i$ 进行线性重构的系数 $\boldsymbol{w}_i$ ：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2$$
$$\text{s.t.} \sum_{j \in Q_i} w_{ij} = 1,$$

  - 其中 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 均为已知，令 $C_{jk} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_k)$ ，w_{ij} 有闭式解

$$w_{ij} = \frac{\sum_{k \in Q_i} C_{jk}^{-1}}{\sum_{l,s \in Q_i} C_{ls}^{-1}}.$$

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**
  - LLE在低维空间中保持$\boldsymbol{w}_i$不变，于是 $\boldsymbol{x}_i$对应的低维空间坐标 $\boldsymbol{z}_i$ 可通过下式求解：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2$$

重构误差最小

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**
  - LLE在低维空间中保持$\boldsymbol{w}_i$不变，于是 $\boldsymbol{x}_i$对应的低维空间坐标 $\boldsymbol{z}_i$ 可通过下式求解：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2$$

重构误差最小

  - 令

$$\mathbf{Z} = (\mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_m) \in \mathbb{R}^{d' \times m}, (\mathbf{W})_{ij} = w_{ij},$$

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathrm{T}} (\mathbf{I} - \mathbf{W}),$$

  - 则优化式可重写为右式，并通过特征值分解求解。

$$\min_{\mathbf{Z}} \operatorname{tr}(\mathbf{Z}\mathbf{M}\mathbf{Z}^{\mathrm{T}})$$
$$\text{s.t. } \mathbf{Z}\mathbf{Z}^{\mathrm{T}} = \mathbf{I}.$$

流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程:

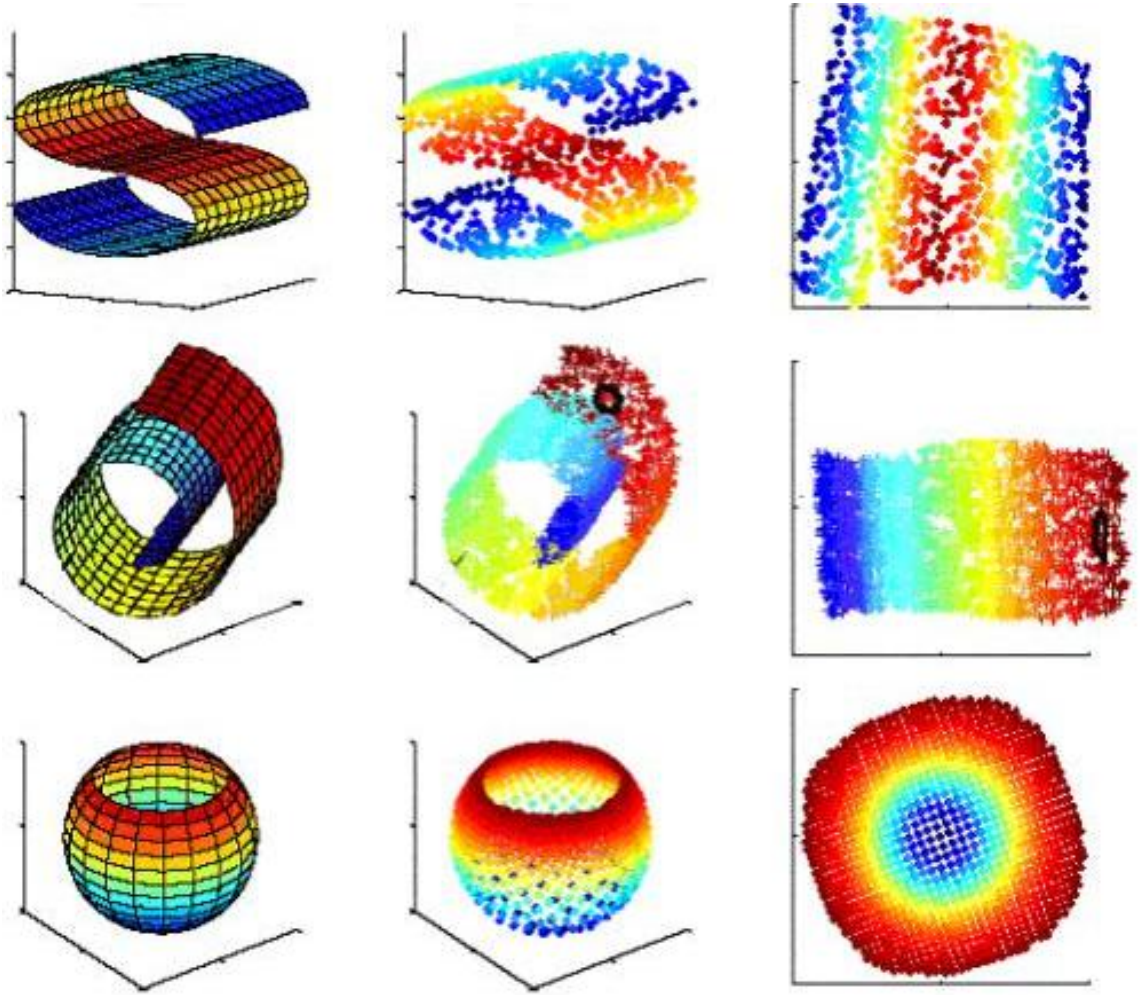
1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: 从式(10.27)求得 w_{ij}, $j \in Q_i$;
4: 对于 $j \notin Q_i$, 令 $w_{ij} = 0$;
5: **end for**
6: 从式(10.30)得到 $\mathbf{M}$;
7: 对 $\mathbf{M}$ 进行特征值分解;
8: **return** $\mathbf{M}$ 的最小 d' 个特征值对应的特征向量

输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.10 LLE 算法

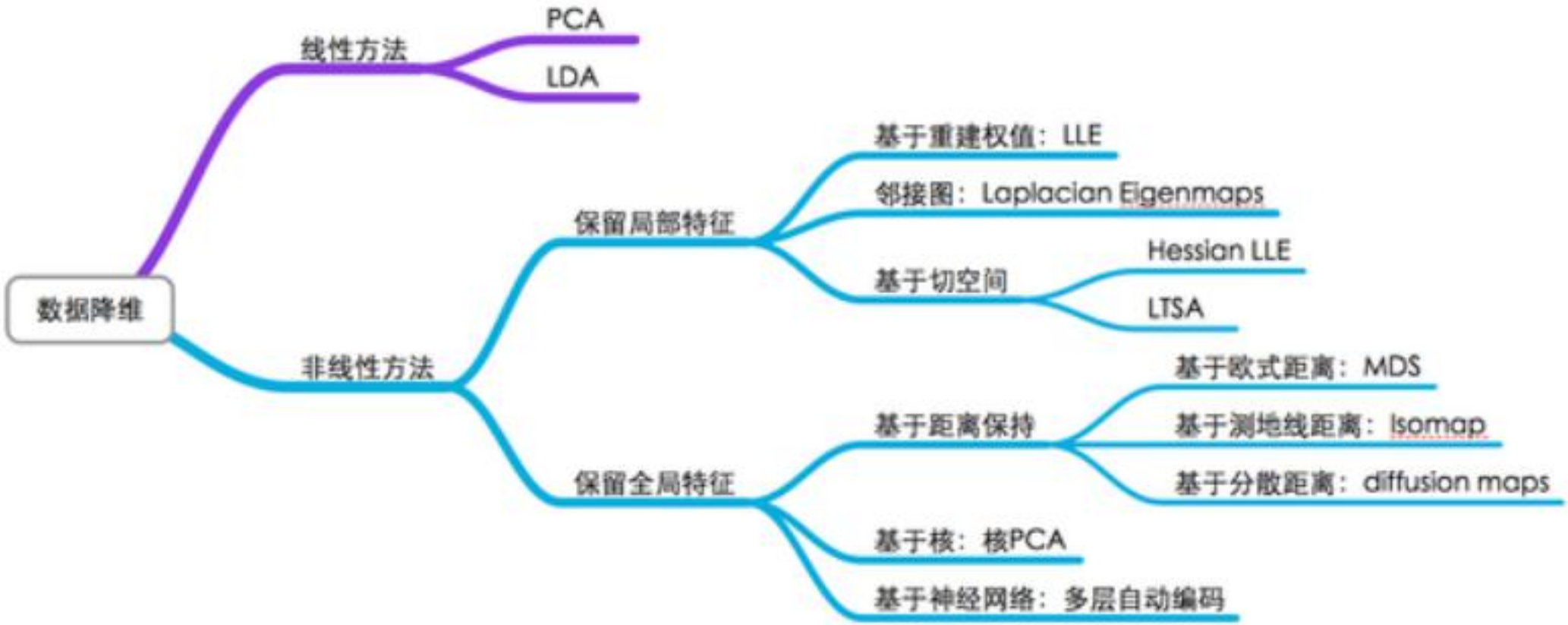
流形学习

- **局部线性嵌入 (Locally Linear Embedding，LLE)**



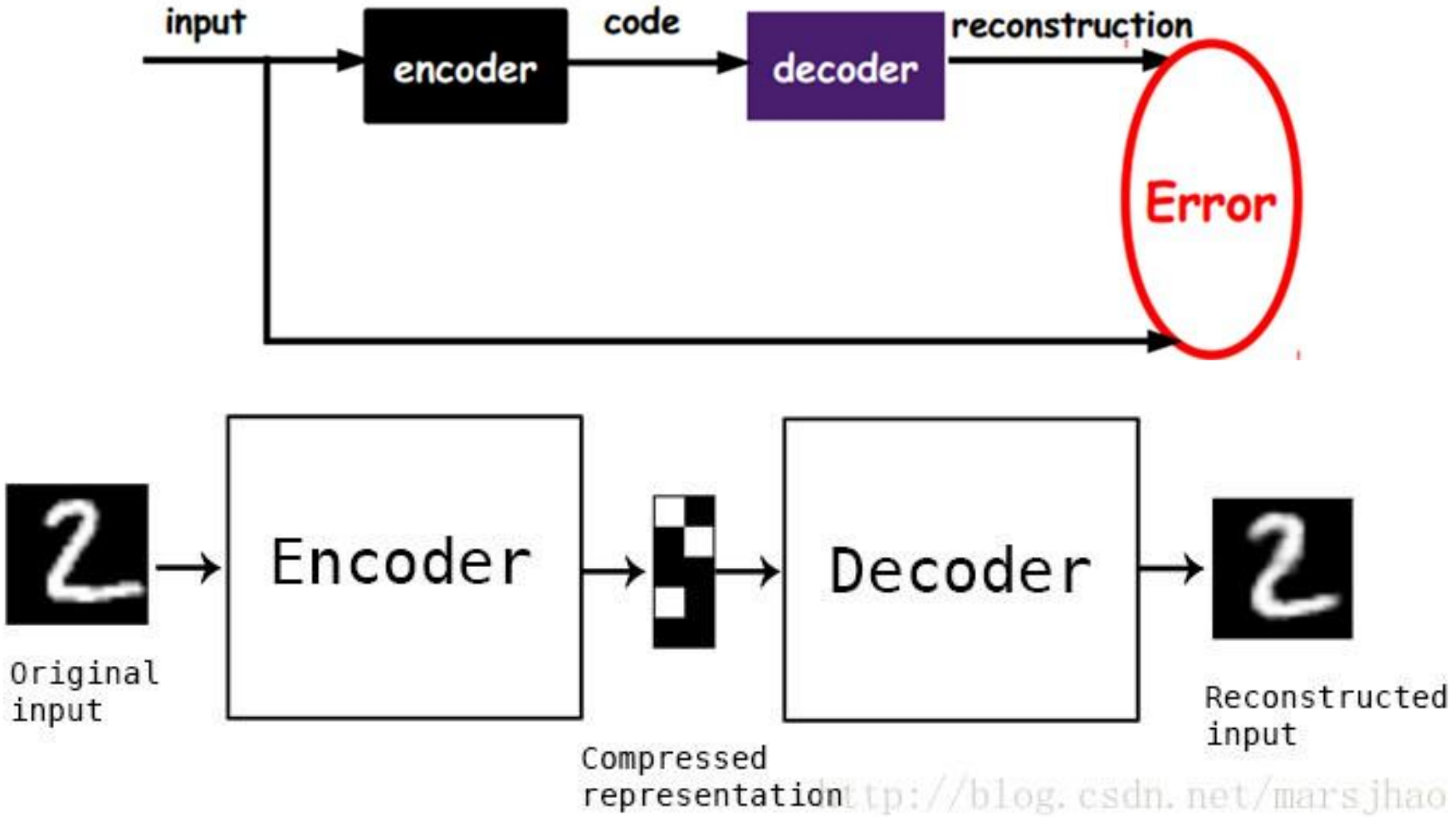
降维方法

- **降维方法类型**


数据降维
线性方法
PCA
LDA
非线性方法
保留局部特征
基于重建权值：LLE
邻接图：Laplacian Eigenmaps
基于切空间
Hessian LLE
LTSA
保留全局特征
基于距离保持
基于欧式距离：MDS
基于测地线距离：Isomap
基于分散距离：diffusion maps
基于核：核PCA
基于神经网络：多层自动编码

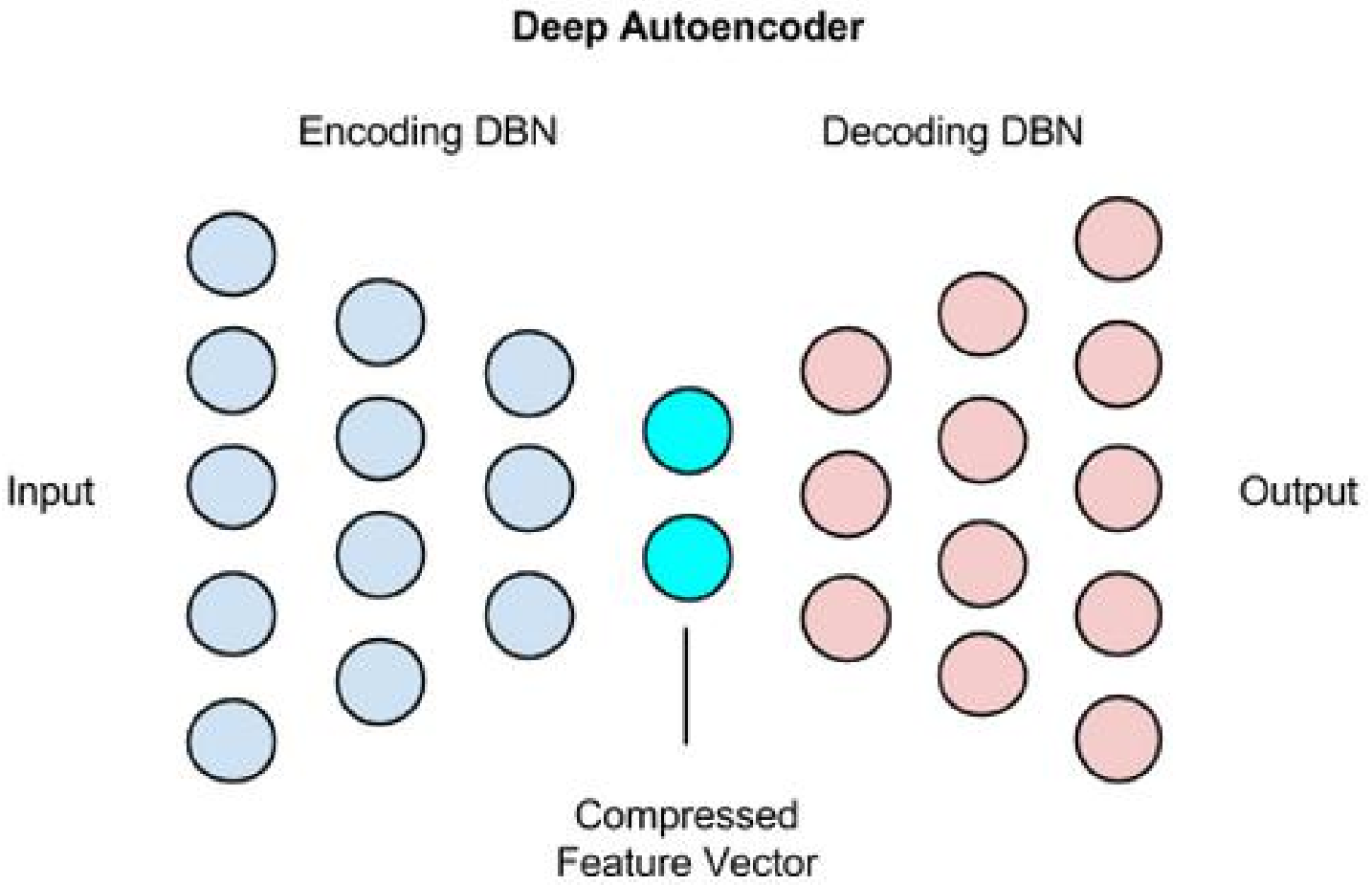
降维方法

- 多层自动编码器



降维方法

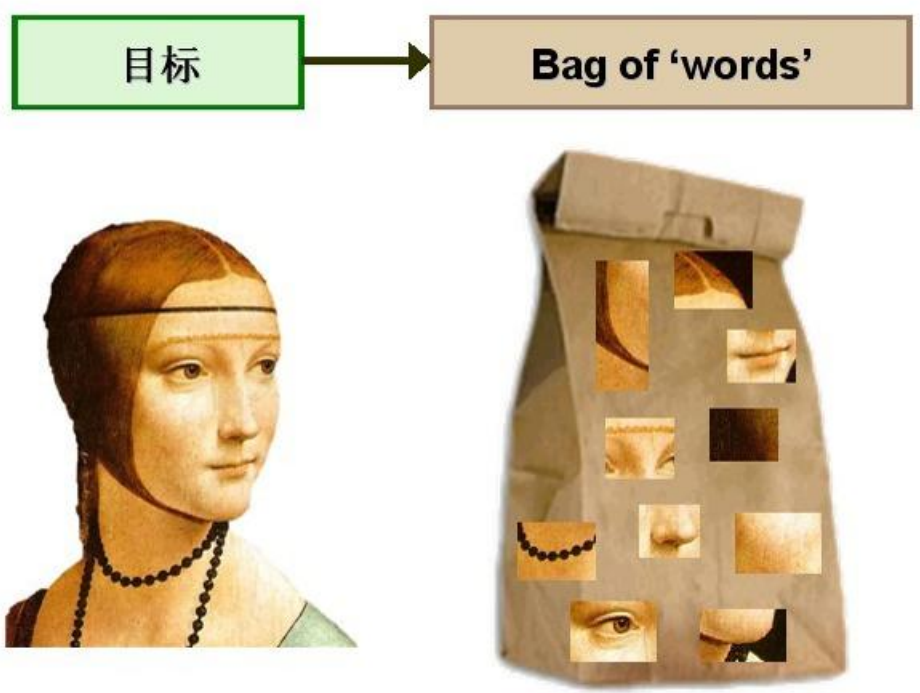
- 多层自动编码器

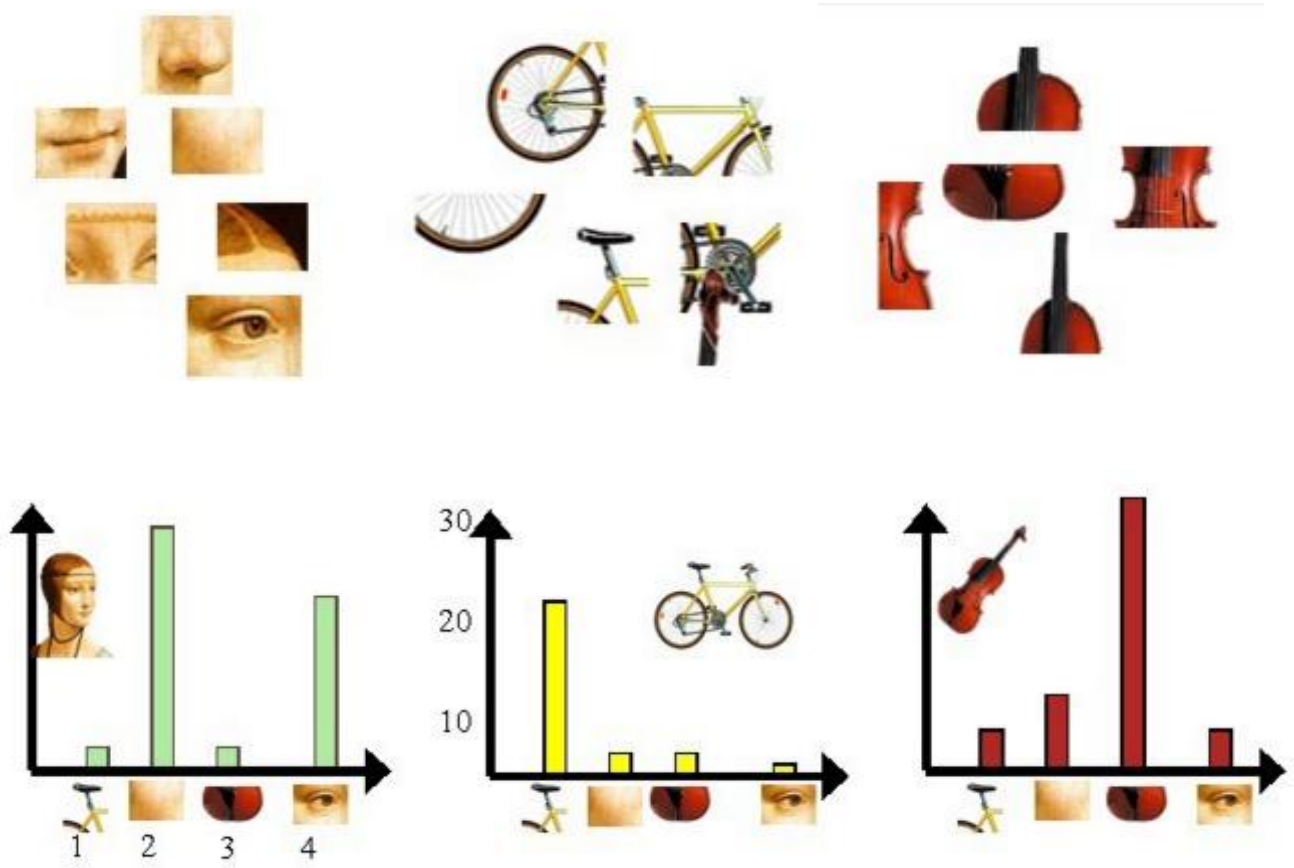


目录

论文解读

- **视觉词向量 Visual Bag-of-Words**





无监督特征学习（PCA+k-means）

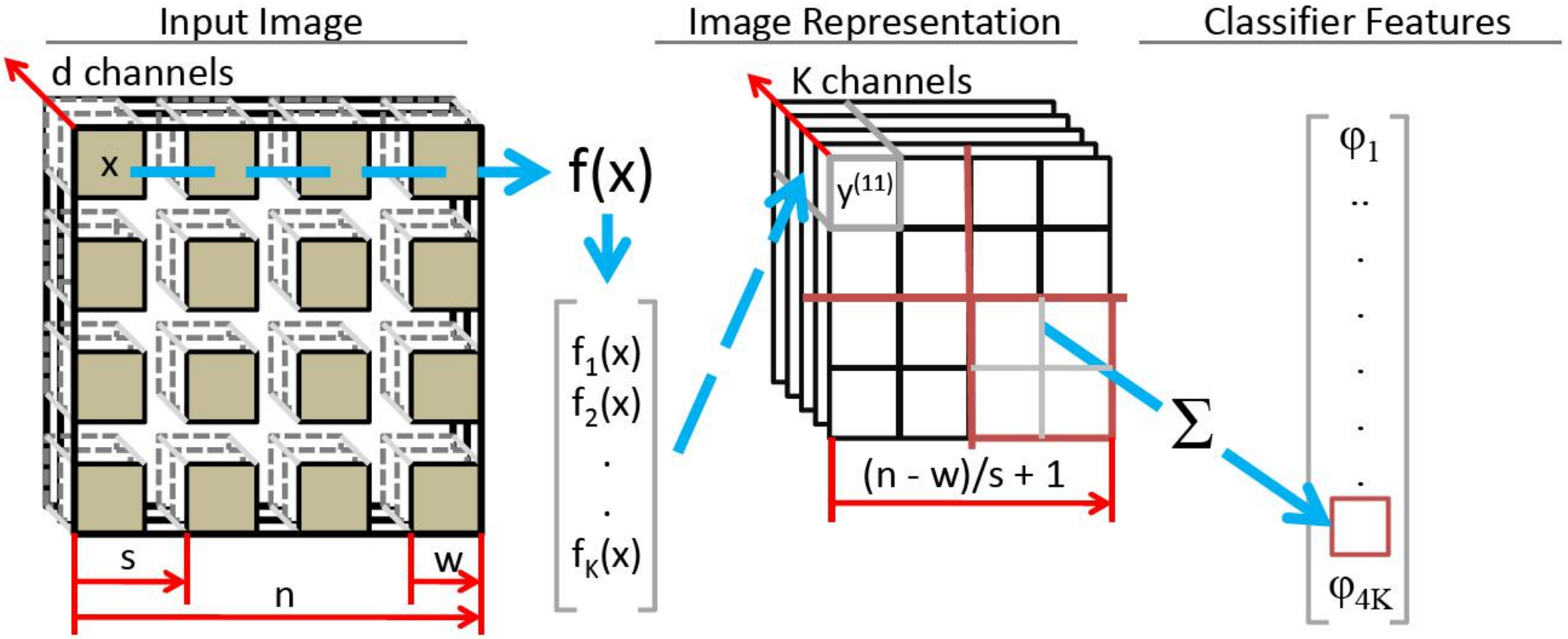
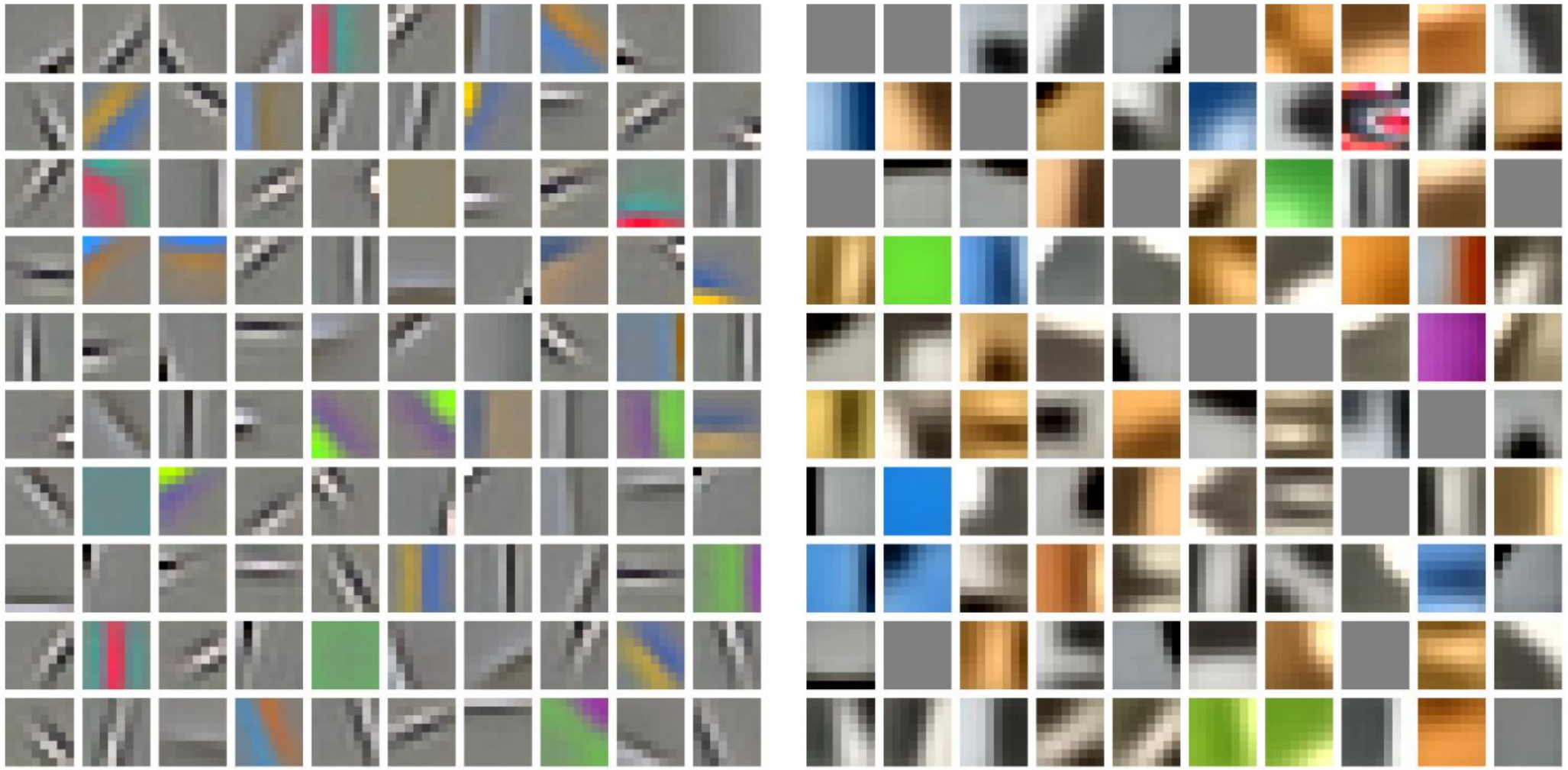


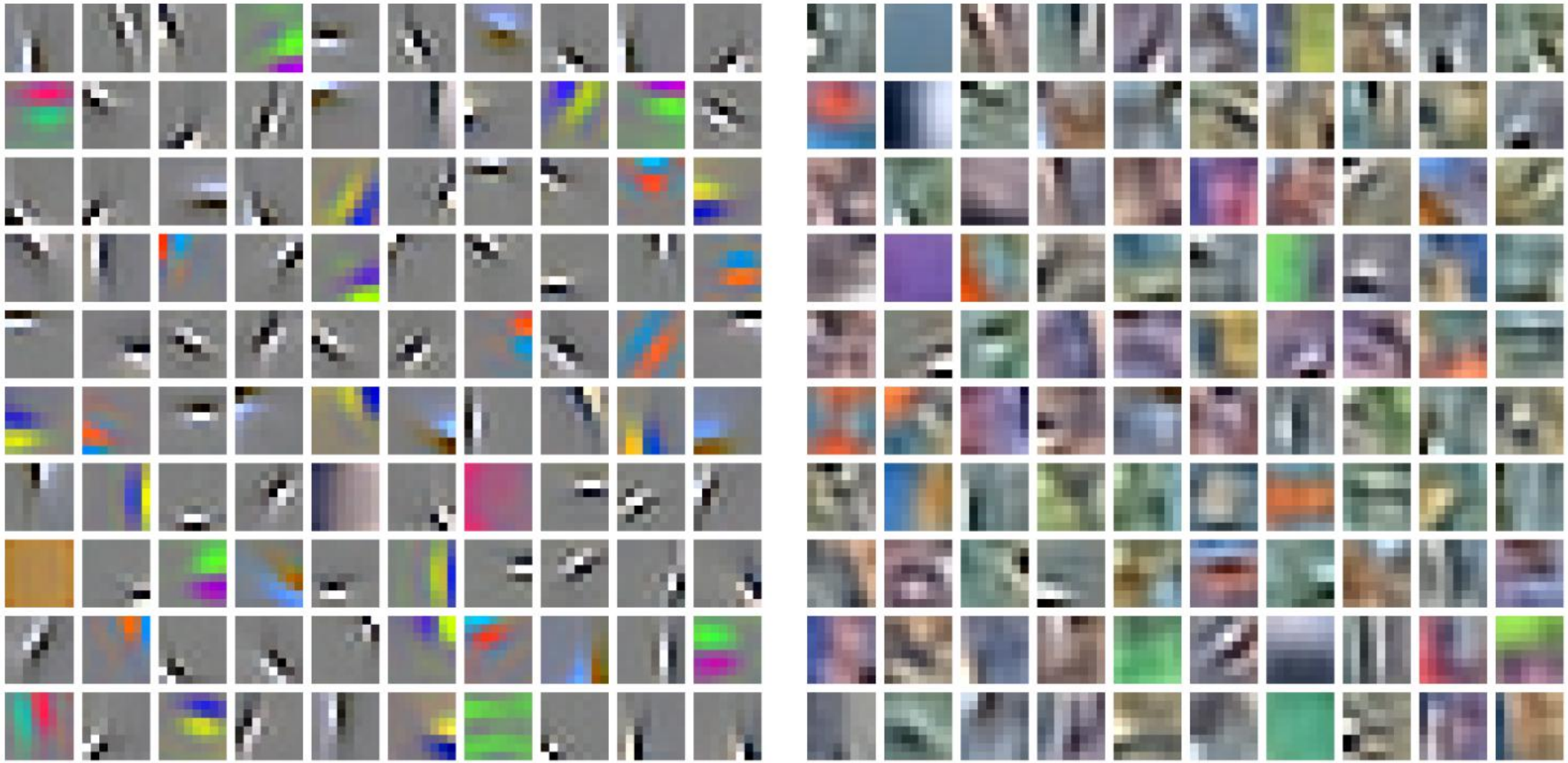
Figure 1: Illustration showing feature extraction using a w-by-w receptive field and stride s. We first extract w-by-w patches separated by s pixels each, then map them to K-dimensional feature vectors to form a new image representation. These vectors are then pooled over 4 quadrants of the image to form a feature vector for classification. (For clarity we have drawn the leftmost figure with a stride greater than w, but in practice the stride is almost always smaller than w.

无监督特征学习（PCA+k-means）



(a) K-means (with and without whitening)

无监督特征学习（PCA+k-means）



(c) Sparse Autoencoder (with and without whitening)

无监督特征学习（PCA+k-means）

- **编码**

1. Extract random patches from unlabeled training images.
2. Apply a pre-processing stage to the patches.
3. Learn a feature-mapping using an unsupervised learning algorithm.

K-means clustering: We apply K-means clustering to learn K centroids $c^{(k)}$ from the input data. Given the learned centroids $c^{(k)}$, we consider two choices for the feature mapping f. The first is the standard 1-of-K, hard-assignment coding scheme:

$$f_k(x) = \begin{cases} 1 & \text{if } k = \arg\min_j ||c^{(j)} - x||_2^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The second is a non-linear mapping that attempts to be "softer" than the above encoding, but also yield sparse outputs through simple competition:

$$f_k(x) = \max\{0, \mu(z) - z_k\} \tag{3}$$

where $z_k = ||x - c^{(k)}||_2$ and $\mu(z)$ is the mean of the elements of z.
We refer to these as `K-means (hard)` and `K-means (triangle)` respectively.

Thank You !

Q & A