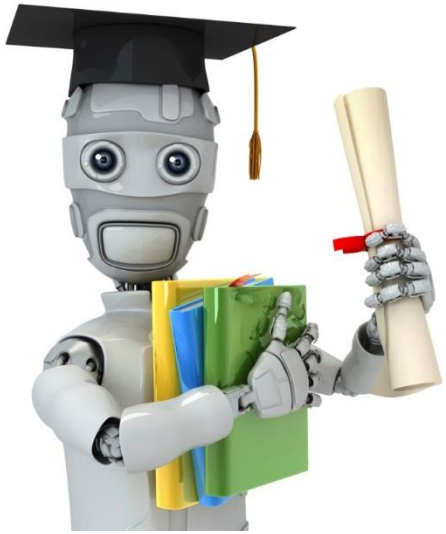


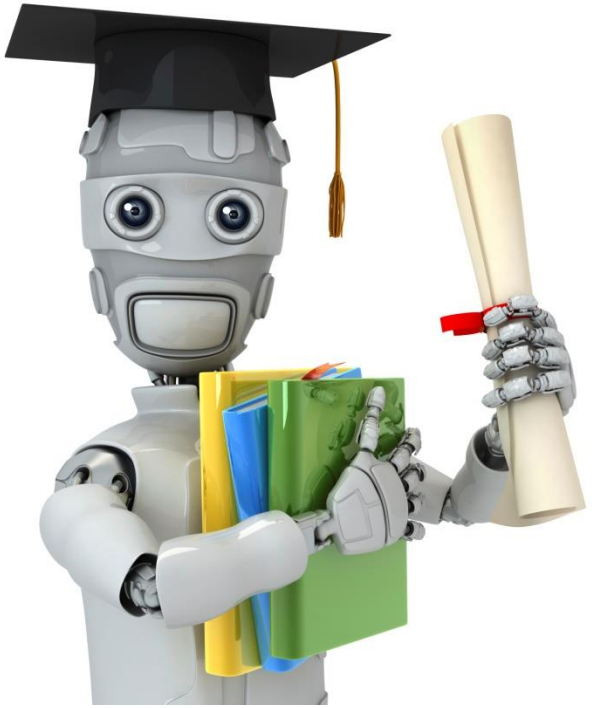
维数约简 (Dimensionality Reduction)

梁毅雄

yxliang@csu.edu.cn

Machine Learning

Some materials from Andrew Ng, Jure Leskovec, Hong-yi Lee and others

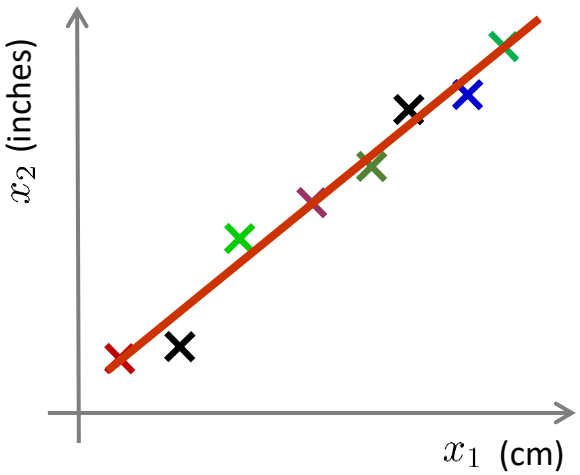


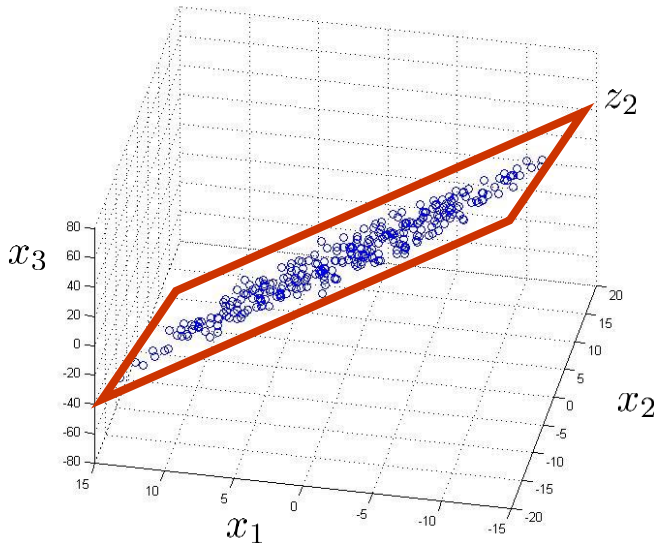
Machine Learning

维数约简

Introduction

维数约简





假设：m维数据位于或者近似位于一个d维的子空间(subspace): $d \leq m$

Axes of this subspace are effective representation of the data

维数约简

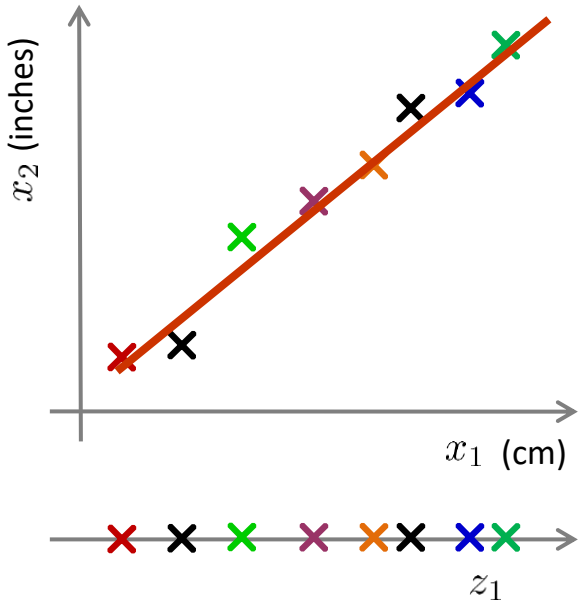
day / customer	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

- 上表中每行表示一个样本，每个样本有5个特征
- 但实际上无需用5维空间来表示这些样本，而只需要？维的空间即可

Why Reduce Dimensions?

- Discover hidden correlations/topics
  - Words that occur commonly together
- Remove redundant and noisy features
  - Not all words are useful
- Easier storage and processing of the data
- Interpretation and visualization

数据压缩



将数据从二维降到一维

$$x^{(1)} \in \mathbb{R}^2 \rightarrow z^{(1)} \in \mathbb{R}^1$$

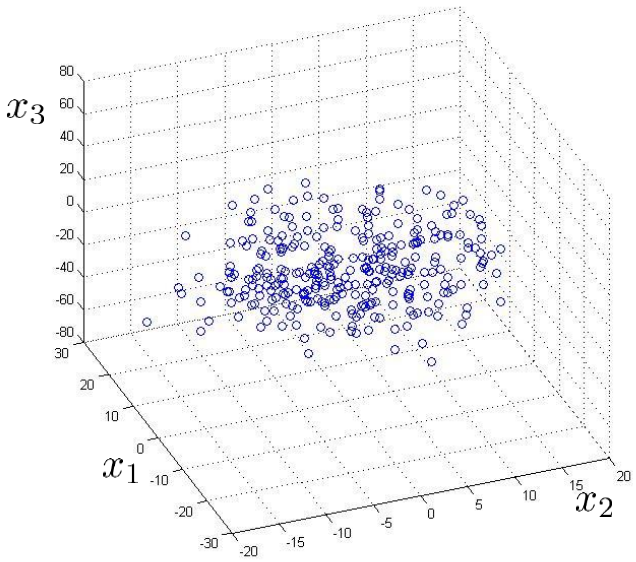
$$x^{(2)} \in \mathbb{R}^2 \rightarrow z^{(2)} \in \mathbb{R}^1$$

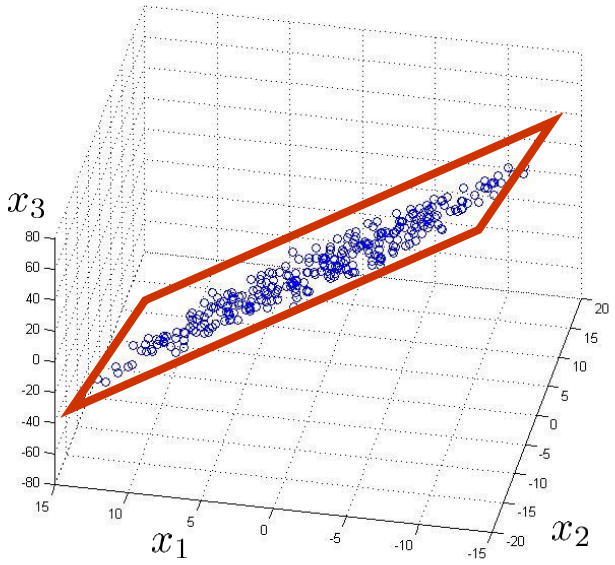
$$\vdots$$

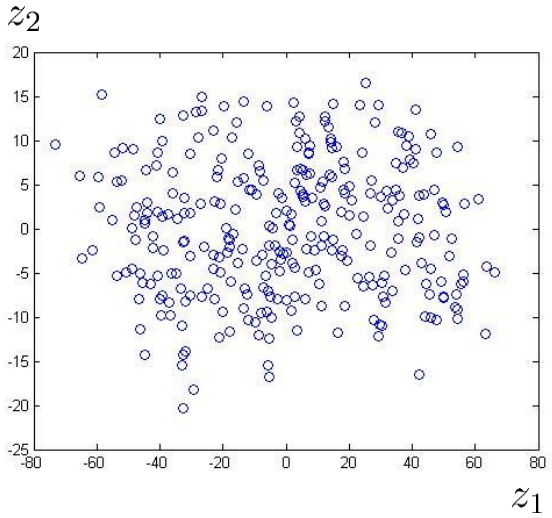
$$x^{(m)} \in \mathbb{R}^2 \rightarrow z^{(m)} \in \mathbb{R}^1$$

数据压缩

将数据从三维降到二维







$$x^{(i)} \in \mathbb{R}^3 \rightarrow z^{(i)} \in \mathbb{R}^2$$

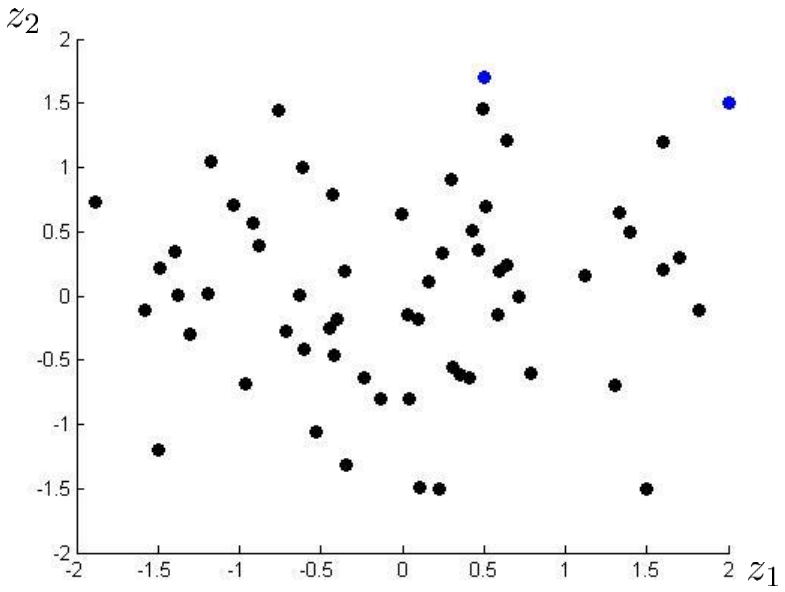
数据可视化

Country	GDP (trillions of US$)	Per capita GDP (thousands of intl. $)	Human Develop-ment Index	Life expectancy	Poverty Index (Gini as percentage)	Mean household income (thousands of US$)	…
Canada	1.577	39.17	0.908	80.7	32.6	67.293	…
China	5.878	7.54	0.687	73	46.9	10.22	…
India	1.632	3.41	0.547	64.7	36.8	0.735	…
Russia	1.48	19.84	0.755	65.5	39.9	0.72	…
Singapore	0.223	56.69	0.866	80	42.5	67.1	…
USA	14.527	46.86	0.91	78.3	40.8	84.3	…
…	…	…	…	…	…	…	

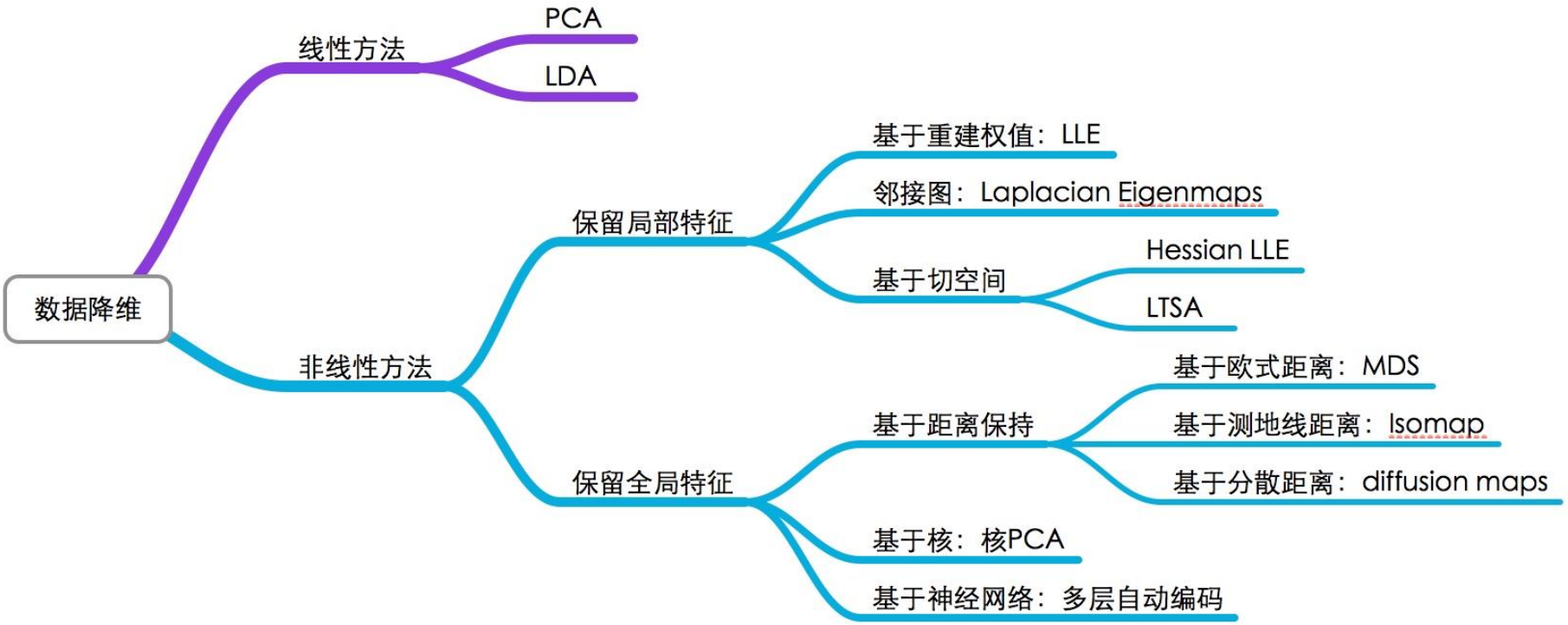
数据可视化

Country	z_1	z_2
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
…	…	…

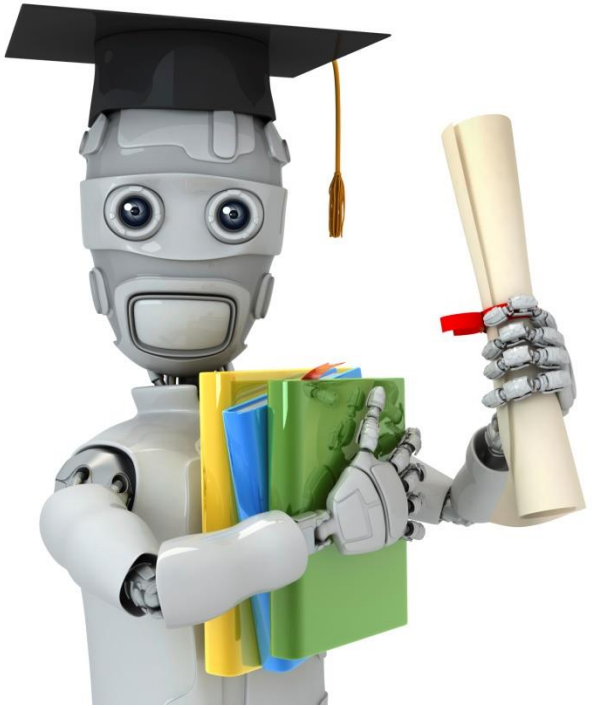
数据可视化



维数约简



http://www.jianshu.com/p/9b23902f01e0



Machine Learning

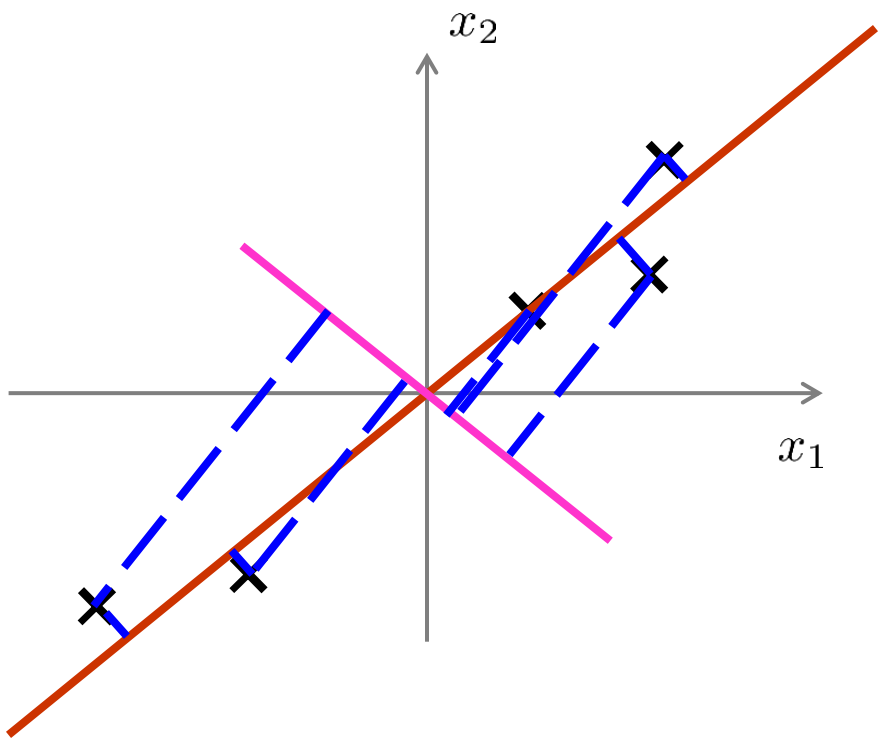
维数约简

主成分分析

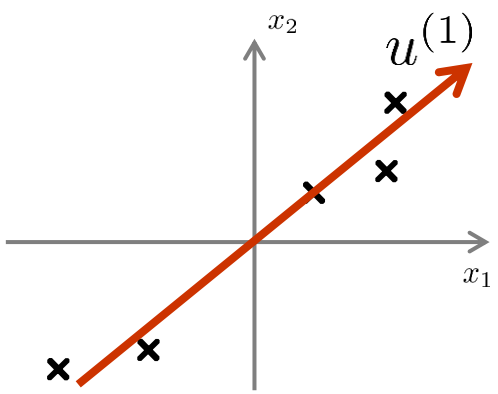
(Principal Component Analysis, PCA)

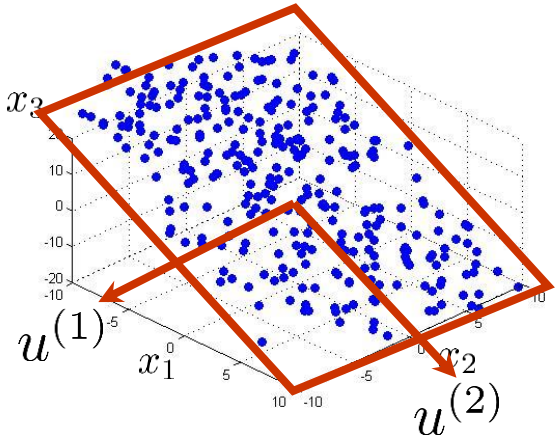
主成分分析

$$x^{(i)} \in \mathbb{R}^2 \rightarrow z^{(i)} \in \mathbb{R}^1$$



主成分分析

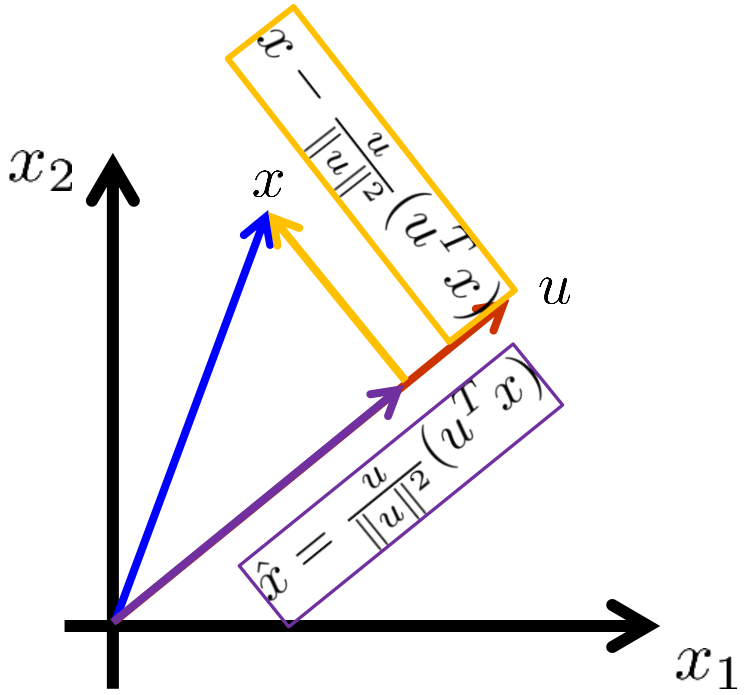




从2维降到1维：找到一个方向$u^{(1)} \in \mathbb{R}^n$来进行数据投影，使得投影误差最小

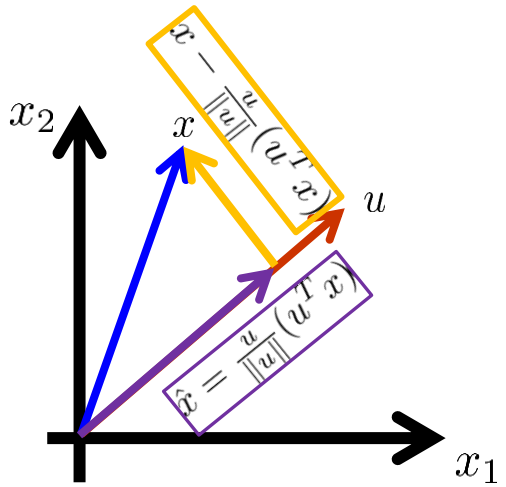
从n维降到k维：找到k个方向$u^{(1)}, u^{(1)}, \cdots, u^{(k)} (u^{(i)} \in \mathbb{R}^n, \forall i)$来进行数据投影，使得投影误差最小

Recap: 投影与投影误差



一维情况下实际上是用$\hat{x} = \frac{u}{\|u\|^2}(u^T x)$去近似$x$，对应的错误为$x - \hat{x}$. 若$u$为单位向量，则$\hat{x} = uu^T x$, 平方投影误差: $\|x - \hat{x}\|^2 = (x - \hat{x})^T(x - \hat{x})$.

主成分分析



假设数据已经过中心化处理，即$\sum_{i=1}^{m} x^{(i)} = 0$. 若已知单位矢量$u$, 对应的（平方）误差为

$$\sum_{i=1}^{m} \|x^{(i)} - uu^T x^{(i)}\|^2$$

因此可以通过求解如下优化问题来找到最优的投影方向u，

$$\min_{u:\|u\|=1} \sum_{i=1}^{m} \|x^{(i)} - uu^T x^{(i)}\|^2$$

主成分分析

$$\min_u \sum_{i=1}^{m} \|x^{(i)} - uu^T x^{(i)}\|^2$$
$$s.t.\ \|u\| = 1$$

$$\begin{aligned}\|x - uu^T x\|^2 &= (x - uu^T x)^T (x - uu^T x) \\ &= x^T x - x^T uu^T x - x^T uu^T x + x^T uu^T uu^T x \\ &= x^T x - x^T uu^T x\end{aligned}$$

因此找具有最小投影误差的方向等价于如下优化问题(最大化方差)：

$$\max_u\ u^T \left(\sum_{i=1}^{m} x^{(i)} {x^{(i)}}^T \right) u$$
$$s.t.\ \|u\| = 1$$

主成分分析

$$\max_u \ u^T \left(\sum_{i=1}^{m} x^{(i)} x^{(i)T} \right) u$$

$$s.t. \ \|u\| = 1$$

拉格朗日函数为

$$L = u^T \left(\sum_{i=1}^{m} x^{(i)} x^{(i)T} \right) u - \lambda(u^T u - 1) = u^T X X^T u - \lambda(u^T u - 1)$$

, 这里$X \in \mathbb{R}^{n \times m}$, 第$i$列对应$x^{(i)}$. 令$\frac{\partial L}{\partial u} = 0$，有

$$XX^T u = \Sigma u = \lambda u.$$

这里$\Sigma = XX^T$为样本的协方差矩阵. 显然可以选择Σ最大特征值对应的特征向量作为最优解.

$$u^T X X^T u = u^T \lambda u = \lambda.$$

主成分分析

如何找到K个投影方向$u^{(1)}, \cdots, u^{(K)}$?

可以选择Σ最大的前面K个特征值所对应的特征向量$u^{(1)}, \ldots, u^{(K)}$. 因为 Σ是对称阵，这些特征向量必定两两正交，构成K维子空间的一组标准正交基。此时 $x^{(i)}$ 对应的投影为

$$z^{(i)} = \begin{bmatrix} {u^{(1)}}^T x^{(i)} \\ {u^{(2)}}^T x^{(i)} \\ \vdots \\ {u^{(K)}}^T x^{(i)} \end{bmatrix} = U_K^T x^{(i)} \in \mathbb{R}^K.$$

这里$U_K = [u^{(1)}, u^{(2)}, \cdots, u^{(K)}] \in \mathbb{R}^{n \times K}$

主成分分析

- 输入: 样本集$\{x^{(1)}, \cdots, x^{(m)}\}$, 低维空间的维度$K$
- 过程:
  1. 对所有样本进行中心化处理： $x^{(i)} \leftarrow x^{(i)} - \frac{1}{m}\sum_j x^{(j)}$;
  2. 计算样本的协方差矩阵$\Sigma = XX^T$;
  3. 对协方差矩阵Σ 进行特征值分解;
  4. 选取最大的K个特征值所对应的特征向量$\{u^{(1)}, \cdots, u^{(K)}\}$组成投影矩阵$U_K$

奇异值分解(Singular Value Decomposition, SVD)

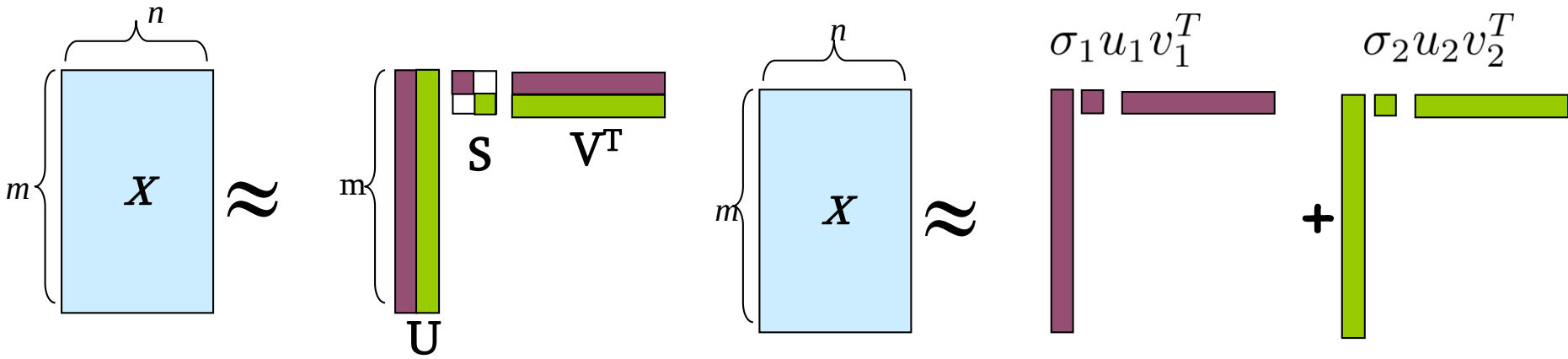
- 奇异值分解定理：

$$X = USV^T$$

- Left Singular Vectors: $U \in \mathbb{R}^{m \times r}, U^T U = I$
- Singular values: $S = \mathtt{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r) \in \mathbb{R}^{r \times r}$, 各奇异值按从大到小排列 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$
- Right Singular Vectors: $V \in \mathbb{R}^{n \times r}, V^T V = I$
- r表示X的秩: $r = \mathtt{rank}(X)$

奇异值分解(Singular Value Decomposition, SVD)

$$X = USV^T = \sum_i \sigma_i u_i v_i^T$$



主成分分析

$X = [x^{(1)}, x^{(2)}, \cdots, x^{(m)}] \in \mathbb{R}^{n \times m}$, $XX^T \in \mathbb{R}^{n \times n}$， $X^TX \in \mathbb{R}^{m \times m}$, 若特征的维数$n$大于样本个数$m$, 即$n \geq m$, 可以把求解$XX^T$的特征分解问题转化为求$X^TX$的特征分解问题。

$$X = USV^T \qquad XX^T = USV^TVSU^T = US^2U^T$$

$XX^TU = US^2 \Rightarrow$ Left Singular Vectors U的列是$\Sigma = XX^T$的特征向量，而Σ的特征值和X的奇异值的关系为：$\lambda_i = \sigma_i^2$

同理，有$X^TXV = VS^2 \Rightarrow$ Right Singular Vectors U的列是X^TX的特征向量，对应特征值和X的奇异值的关系为：$\lambda_i = \sigma_i^2$

主成分分析

$$XX^TU = US^2 \text{ 即：} \quad XX^Tu_i = \sigma_i^2 u_i$$

$$X^TXV = VS^2 \text{ 即：} \quad X^TXv_i = \sigma_i^2 v_i$$

两边左乘X，有$XX^TXv_i = \sigma_i^2 Xv_i$，即$Xv_i$ 可作为XX^T的特征向量。

是否意味着$u_i = Xv_i$?

将$X = USV^T = \sum_j \sigma_j u_j v_j^T$代入，$Xv_i = \sum_j \sigma_j u_j v_j^T v_i = \sigma_i u_i$

$$\boxed{u_i = \frac{1}{\sigma_i} Xv_i = \frac{1}{\sqrt{\lambda_i}} Xv_i}$$

主成分分析

$$z^{(i)} = U_K^T x^{(i)} = \begin{bmatrix} {u^{(1)}}^T x^{(i)} \\ {u^{(2)}}^T x^{(i)} \\ \vdots \\ {u^{(K)}}^T x^{(i)} \end{bmatrix} \in \mathbb{R}^K.$$

$$\hat{x}^{(i)} = U_K z^{(i)}$$

$$U_K = [u^{(1)}, u^{(2)}, \cdots, u^{(K)}] \in \mathbb{R}^{n \times K}$$

如何选择低维子空间的维度K?

Typically, choose K to be smallest value so that

$$\frac{\sum_{i=1}^m \|x^{(i)} - \hat{x}^{(i)}\|^2}{\sum_{i=1}^m \|x^{(i)}\|^2} = 1 - \frac{\sum_{j=1}^K \lambda_j}{\sum_{j=1}^n \lambda_j} \leq \epsilon$$

如$\epsilon = 0.01$

如何证明？

主成分分析

证明要用到矩阵迹$tr(\cdot)$的如下性质

$$tr(A) = tr(A^T)$$
$$tr(A+B) = tr(A) + tr(B)$$
$$tr(AB) = tr(BA)$$
$$tr(ABC) = tr(BCA) = tr(CAB)$$

$$z^{(i)} = U_K^T x^{(i)} \in \mathbb{R}^K$$
$$\hat{x}^{(i)} = U_K z^{(i)} = U_K U_K^T x^{(i)} \in \mathbb{R}^n$$
$$X = \left[x^{(1)}, x^{(2)}, \cdots, x^{(m)}\right] \in \mathbb{R}^{n \times m}$$
$$\hat{X} = \left[\hat{x}^{(1)}, \hat{x}^{(2)}, \cdots, \hat{x}^{(m)}\right] = U_K U_K^T X \in \mathbb{R}^{n \times m}$$
$$\sum_{i=1}^{m} \|x^{(i)}\|^2 = tr(XX^T) = tr(X^T X)$$
$$\sum_{i=1}^{m} \|x^{(i)} - \hat{x}^{(i)}\|^2 = tr\left((X - U_K U_K^T X)(X - U_K U_K^T X)^T\right)$$
$$= tr\left(XX^T - XX^T U_K U_K^T - U_K U_K^T XX^T - U_K U_K^T XX^T U_K U_K^T\right)$$
$$= tr\left(XX^T\right) - tr\left(U_K^T XX^T U_K\right)$$
$$= \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{K} \lambda_j$$

即有 $\dfrac{\sum_{i=1}^{m} \|x^{(i)} - \hat{x}^{(i)}\|^2}{\sum_{i=1}^{m} \|x^{(i)}\|^2} = 1 - \dfrac{\sum_{j=1}^{K} \lambda_j}{\sum_{j=1}^{n} \lambda_j}$

主成分分析的使用

- 加速监督学习的训练
  譬如给定数据集为 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$, 假设特征维度为$10,000$, 首先利用$x^{(1)}, x^{(2)}, \ldots, x^{(m)} \in \mathbb{R}^{10000}$基于PCA方法得到1,000个投影向量，对原始的$x$降维后得到 $z^{(1)}, z^{(2)}, \ldots, z^{(m)} \in \mathbb{R}^{1000}$, 然后再利用新的样本集 $(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \ldots, (z^{(m)}, y^{(m)})$去进行监督学习训练。

思考：进行验证或者测试时新数据如何处理？

主成分分析的使用

- Compression
  - Reduce memory/disk needed to store data
  - Speed up learning algorithm
- Visualization

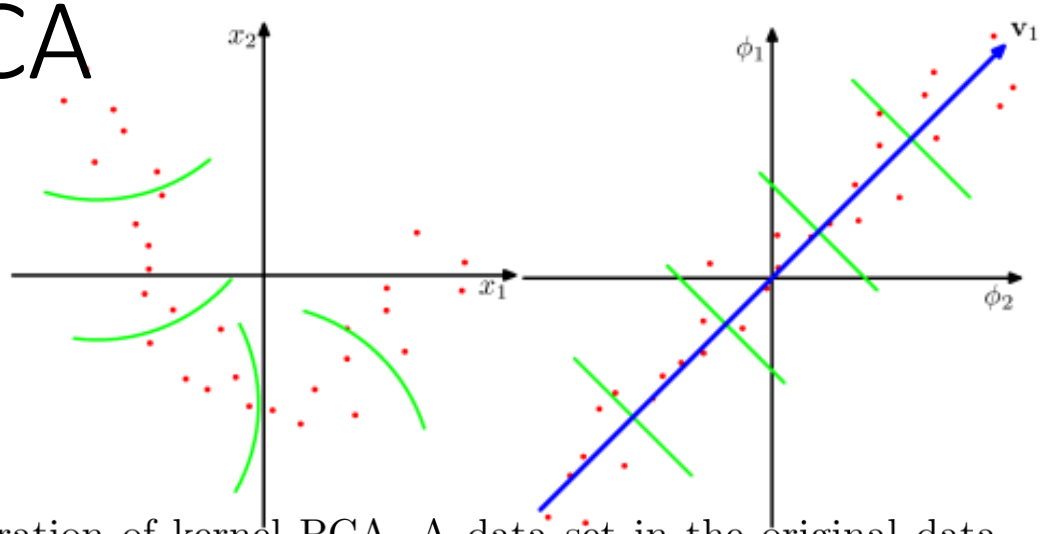
主成分分析的使用

Bad use of PCA: To prevent overfitting

- Use $z^{(i)}$ instead of $x^{(i)}$ to reduce the number of features to $k < n$. Thus, fewer features, less likely to overfit. Right?
- This might work OK, but isn't a good way to address overfitting. Use regularization instead.

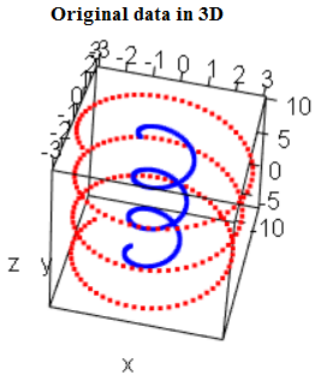
$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

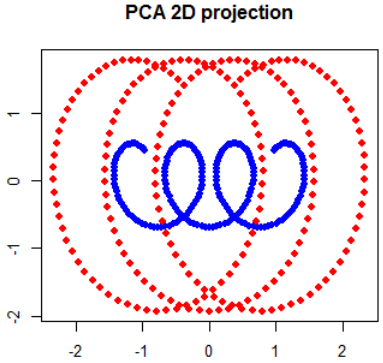
Kernel PCA

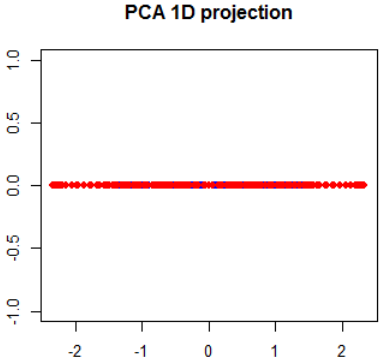


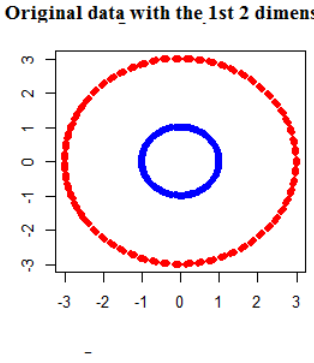
Schematic illustration of kernel PCA. A data set in the original data space (left-hand plot) is projected by a nonlinear transformation $\phi(x)$ into a feature space (right-hand plot). By performing PCA in the feature space, we obtain the principal components, of which the first is shown in blue and is denoted by the vector v_1. The green lines in feature space indicate the linear projections onto the first principal component, which correspond to nonlinear projections in the original data space. Note that in general it is not possible to represent the nonlinear principal component by a vector in x space.

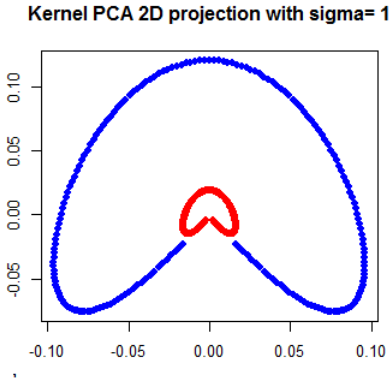
Kernel PCA

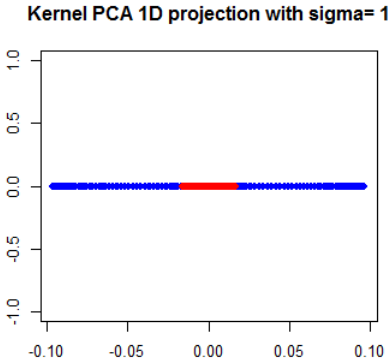

Original data in 3D
z
y
x


PCA 2D projection


PCA 1D projection


Original data with the 1st 2 dimensions


Kernel PCA 2D projection with sigma= 1


Kernel PCA 1D projection with sigma= 1

Kernel PCA

$$XX^Tu = \lambda u \Rightarrow \Phi\Phi^Tu = \lambda u$$

$$\text{这里}\Phi = [\phi(x^{(1)}), \phi(x^{(2)}), \cdots, \phi(x^{(m)})]$$

由于$\phi(x)$的维度未知，无法直接对$\Phi\Phi^T$进行特征值分解，转而求$\Phi^T\Phi$的特征分解，然后根据Left Singular Vectors和Right Singular Vectors关系间接求解出$\Phi\Phi^T$的特征向量.

$$\boxed{u_i = \frac{1}{\sqrt{\lambda_i}}\Phi v_i}$$

Kernel PCA

$$\Phi^T\Phi = \left[\phi(x^{(1)}), \phi(x^{(2)}), \cdots, \phi(x^{(m)})\right]^T \left[\phi(x^{(1)}), \phi(x^{(2)}), \cdots, \phi(x^{(m)})\right]$$

$$= \begin{bmatrix} \phi(x^{(1)})^T\phi(x^{(1)}) & \cdots & \phi(x^{(1)})^T\phi(x^{(m)}) \\ \phi(x^{(2)})^T\phi(x^{(1)}) & \cdots & \phi(x^{(2)})^T\phi(x^{(m)}) \\ \vdots & \cdots & \vdots \\ \phi(x^{(m)})^T\phi(x^{(1)}) & \cdots & \phi(x^{(m)})^T\phi(x^{(m)}) \end{bmatrix}$$

$$= \begin{bmatrix} k(x^{(1)}, x^{(1)}) & \cdots & k(x^{(1)}, x^{(m)}) \\ \vdots & \cdots & \vdots \\ k(x^{(m)}, x^{(1)}) & \cdots & k(x^{(m)}, x^{(m)}) \end{bmatrix} = \mathbf{K}$$

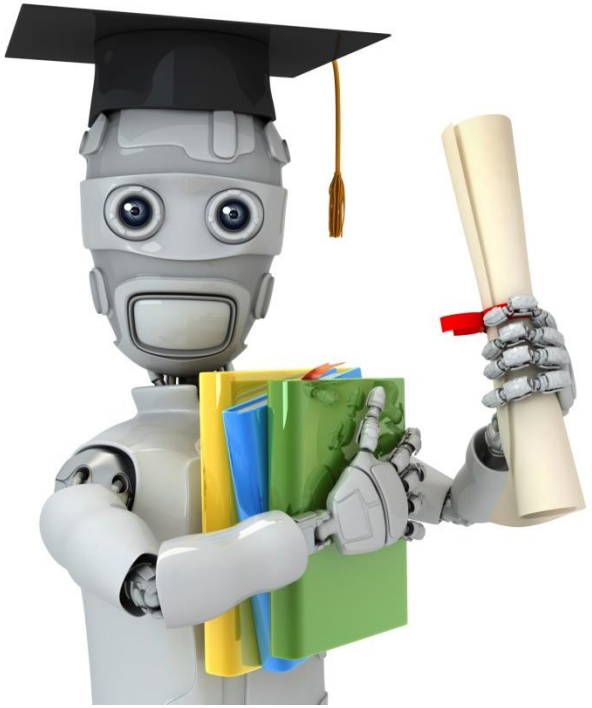
Kernel PCA

求出$\mathbf{K}$的特征值和特征向量$\{\lambda_i, v_i\}_i$后，可根据$u_i = \frac{1}{\sqrt{\lambda_i}}\Phi v_i$ 求得样本协方差矩阵$\Phi\Phi^T$的特征矢量u_i

$$u_i = \frac{1}{\sqrt{\lambda_i}}\Phi v_i$$

则$\phi(x)$在该方向上的投影为

$$\begin{aligned}\phi(x)^T u_i &= \frac{1}{\sqrt{\lambda_i}}\left[\phi(x)^T\phi(x^{(1)}), \phi(x)^T\phi(x^{(2)}), \cdots, \phi(x)^T\phi(x^{(m)})\right] v_i \\ &= \frac{1}{\sqrt{\lambda_i}}\left[k(x, x^{(1)}), k(x, x^{(2)}), \cdots, k(x, x^{(m)})\right] v_i\end{aligned}$$



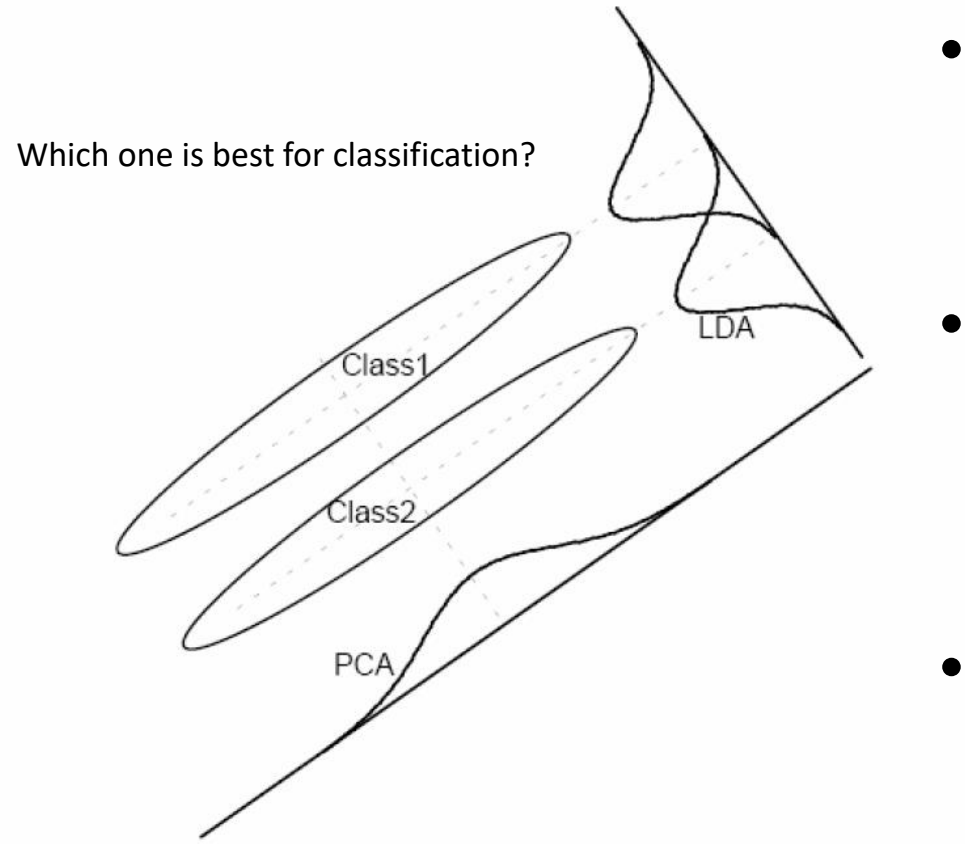
Machine Learning

维数约简

线性鉴别分析

(Linear Discriminant Analysis, LDA)

PCA vs. LDA



- PCA是根据样本投影后数据的方差来选择投影方向的（最优重构）
- 但对于分类问题而言，这样的投影反而有可能使得数据更加无法划分
- 例: 识别字母“O”和“Q”

Linear Discriminant Analysis

- LDA最早是由Fisher提出，亦称Fisher鉴别分析. 其主要目的是寻找寻找一个方向，使得沿该方向两类样本在某种意义上分开的最好
- Fisher准则: 类间方差和类内方差的比。数学形式上，寻求方向w，使得

$$J_F(w) = \frac{w^T S_b w}{w^T S_w w}$$

最大，其中类间散布矩阵S_b 和类内散布矩阵S_w 分别定义为

$$S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$S_w = \sum_{y^{(i)}=-1} (x^{(i)} - \mu_1)(x^{(i)} - \mu_1)^T + \sum_{y^{(i)}=+1} (x^{(i)} - \mu_2)(x^{(i)} - \mu_2)^T$$

Linear Discriminant Analysis

$$J_F(w) = \frac{w^T S_b w}{w^T S_w w}$$

- 如何求解最优的投影方向w? 注意到LDA的目标函数$J_F(w)$为广义Rayleigh熵，使其最大化的投影方向w必满足如下广义特征值问题

$$S_b w = \lambda S_w w$$

证明如下：因为$J_F(w)$的取值与w的长度无关，不失一般性，令$w^T S_w w = 1$, 此时只需最小化$-w^T S_b w$, 对应的拉格朗日函数为

$$L(w) = -w^T S_b w + \lambda(w^T S_w w - 1)$$

令$\frac{\partial L(w)}{\partial w} = 0$，即有 $S_b w = \lambda S_w w$.

Linear Discriminant Analysis

- 对于两类而言，实际上无需去进行广义特征值分解，因为$S_b w$ 总是位于$\mu_1 - \mu_2$的方向上（思考：为什么？）由于我们仅关心w的方向，不妨令$S_b w = \lambda(\mu_1 - \mu_2)$，因此有$S_w w = (\mu_1 - \mu_2)$, 若$S_w$可逆，即

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

实际中为了数值稳定性，常对S_w进行奇异值分解来计算S_w^{-1}.

Linear Discriminant Analysis

- Fisher鉴别分析仅提供了一个在两类条件下的一维映射，并没有提供分类规则。实际上，只需要确定一个阈值，即可作出决策。可以证明若两类数据同先验、服从高斯分布且都应的协方差矩阵相等，LDA可获得最小错误率。
- 可以证明，对于两类问题SVM得到的最优分类超平面的法向量等价于对所有支持向量采用线性鉴别分析得到的最优鉴别方向, 即支持向量机的方法可以看成是一种‘Sparsify’线性鉴别分析

Linear Discriminant Analysis

- 如何将LDA拓展到多分类任务？假定存在K类, 第k类的样本数为m_k, 对应的均值向量为μ_k，类内散布矩阵S_w仍定义为各类的散布矩阵之和

$$S_w = \sum_{k=1}^{K} \sum_{y^{(i)}=k} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T$$

再定义总体散布矩阵S_t

$$S_t = S_b + S_w = \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

Linear Discriminant Analysis

这里μ为所有样本的均值, 类间散布矩阵S_b可由下式计算

$$S_b = S_t - S_w = \sum_{k=1}^{K} m_k(\mu_k - \mu)(\mu_k - \mu)^T$$

则多类Fisher准则为：

$$J_F(W) = \frac{\mathtt{tr}(W^T S_b W)}{\mathtt{tr}(W^T S_w W)}$$

即寻找投影矩阵W使得$J_F(W)$最大化，这里$\mathtt{tr}(\cdot)$表示矩阵的迹(trace)

Linear Discriminant Analysis

$$W^* = \arg\max_W J_F(W) = \arg\max_W \frac{\mathtt{tr}(W^T S_b W)}{\mathtt{tr}(W^T S_w W)}$$

令$\frac{\partial J_F(W)}{\partial W} = 0$, 化简后有如下广义特征值分解问题

$$S_b W = \lambda S_w W.$$

$$\frac{\partial}{\partial W}\mathbf{tr}(W^T B W) = BW + B^T W$$

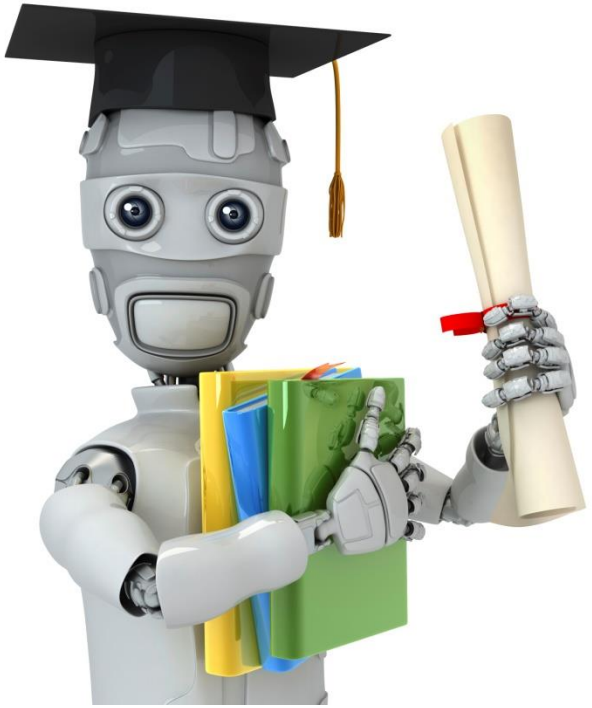
对应的解析解为$S_w^{-1} S_b$的前$K-1$个最大广义特征值所对应的特征向量组成的矩阵。

Linear Discriminant Analysis

- 传统LDA实际上是找到一个投影矩阵$W \in \mathbb{R}^{n\times(K-1)}$将数据从$n$维原始空间投影到$K-1$维空间，通常$K-1 \ll n$.
- 与PCA不同，LDA在求解投影矩阵W时，使用到了类别信息，属于典型的监督学习降维技术

  思考：

  - 为什么是$K-1$维？
  - PCA能最多降到多少维？
  - 能否采用Kernel trick将LDA拓展成非线性的Kernel discriminant analysis (KDA)? How?



Machine Learning

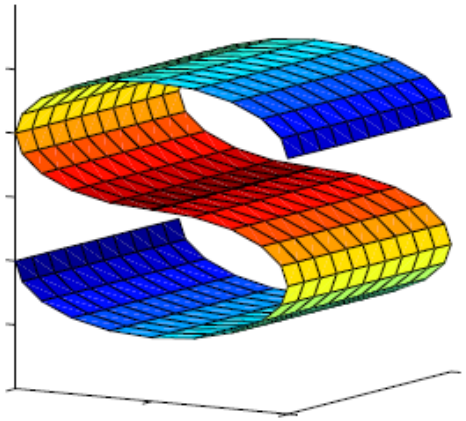
维数约简

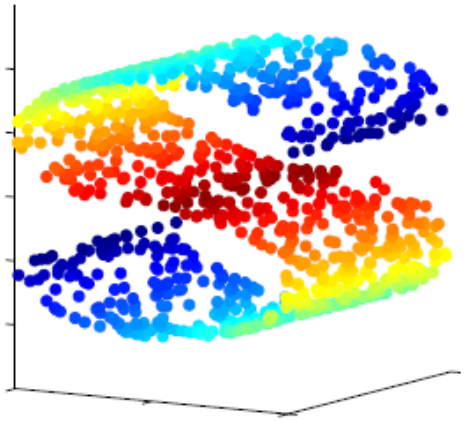
流形学习
(Manifold Learning)

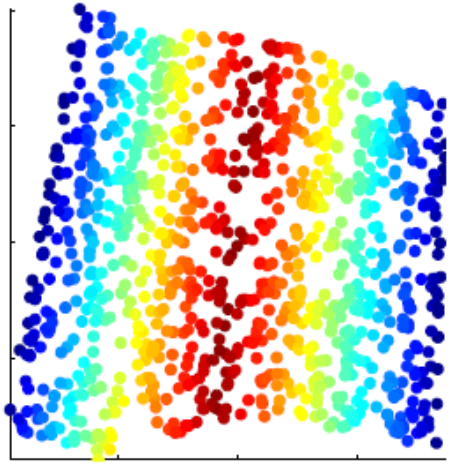
Manifold Learning

Manifold Learning (or non-linear dimensionality reduction) embeds data that originally lies in a high dimensional space in a lower dimensional space, while preserving characteristic properties.

- a manifold is a topological space that locally resembles Euclidean space near each point.



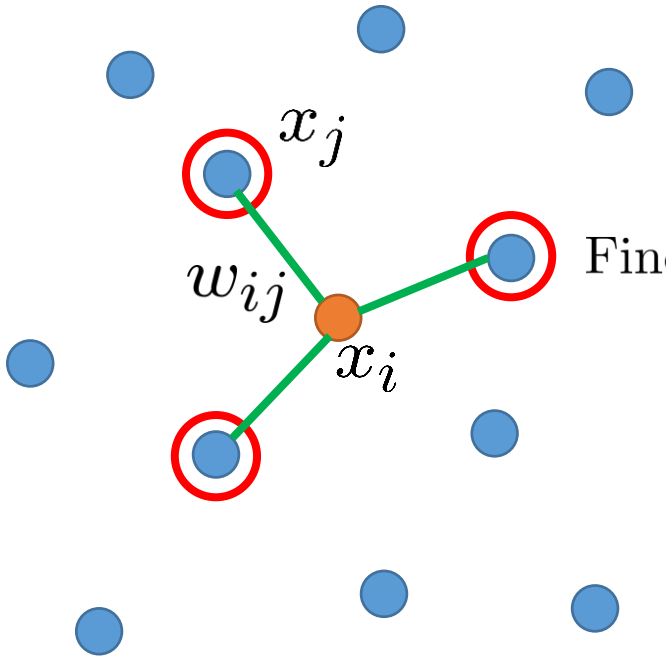




Suitable for clustering or following supervised learning

Locally Linear Embedding (LLE)

w_{ij} represents the relation between x_i and x_j



Find a set of w_{ij} minimizing

$$\sum_i \|x_i - \sum_j w_{ij} x_j\|_2$$

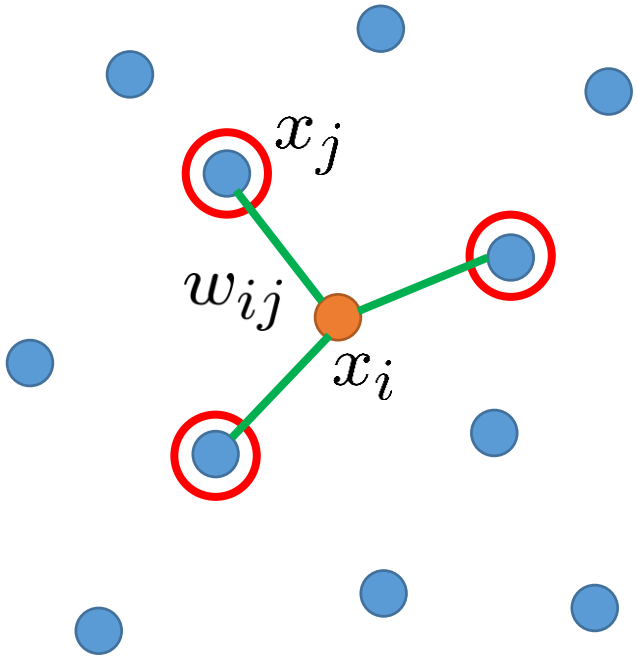
Then find the dimension reduction results z_i and z_j based on w_{ij}

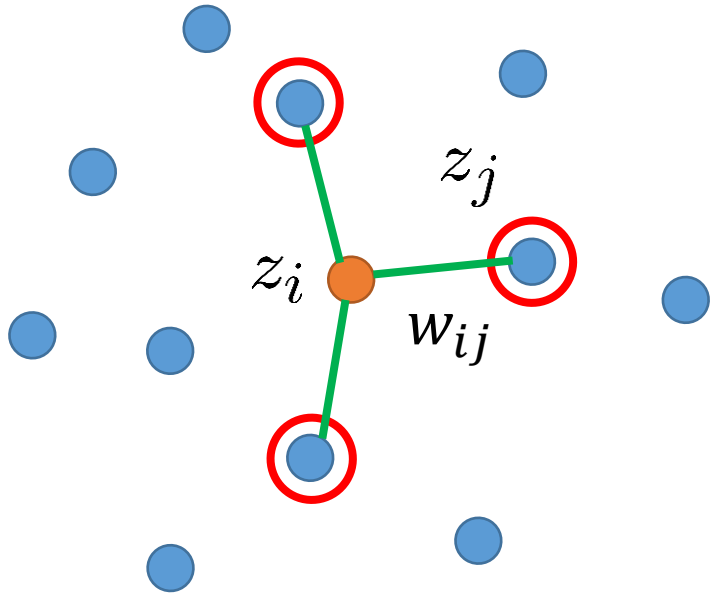
Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290, no. 5500 (2000): 2323-2326.

LLE

Find a set of z_i minimizing

$$\sum_i \|z_i - \sum_j w_{ij} z_j\|_2$$



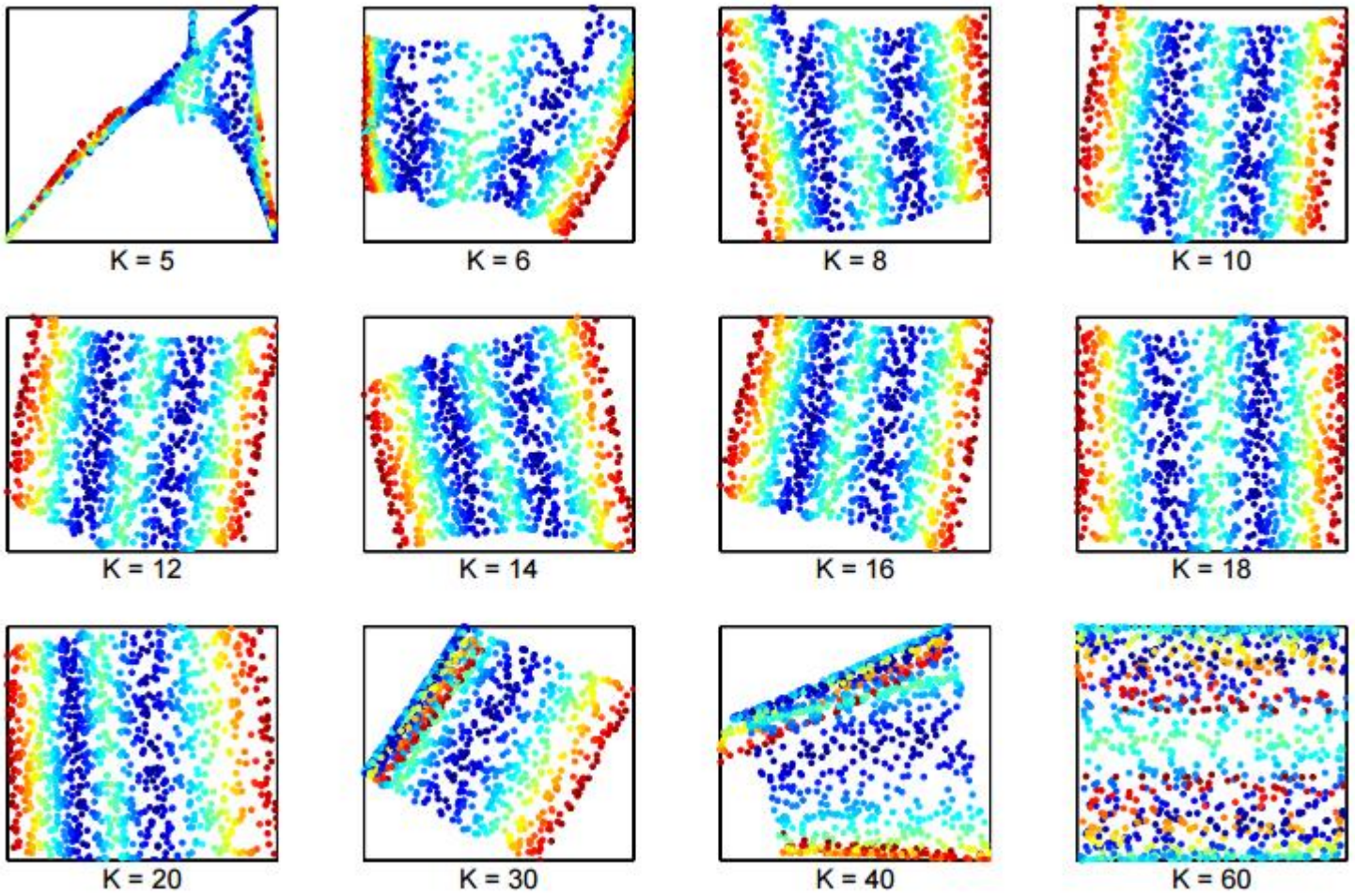


Original Space

New (Low-dim) Space

Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290, no. 5500 (2000): 2323-2326.

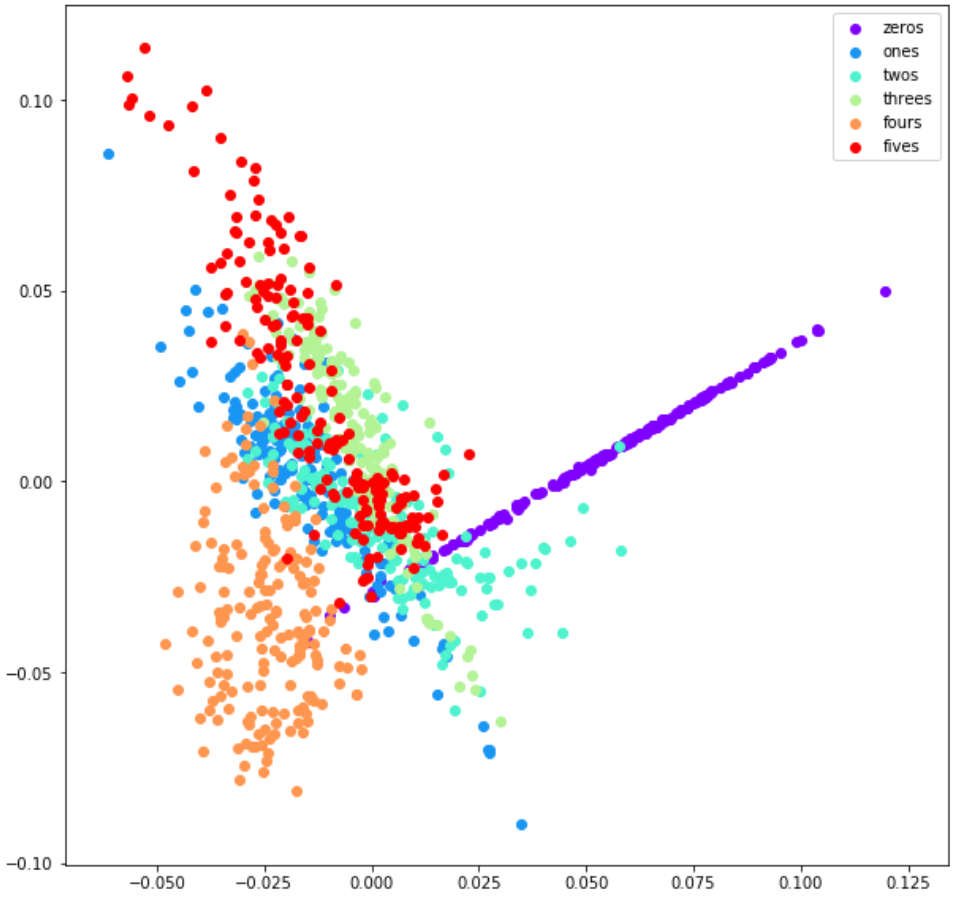
LLE



Lawrence K. Saul, Sam T. Roweis, “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds”, JMLR, 2013

LLE

Visualization of a subset of MNIST dataset after LLE

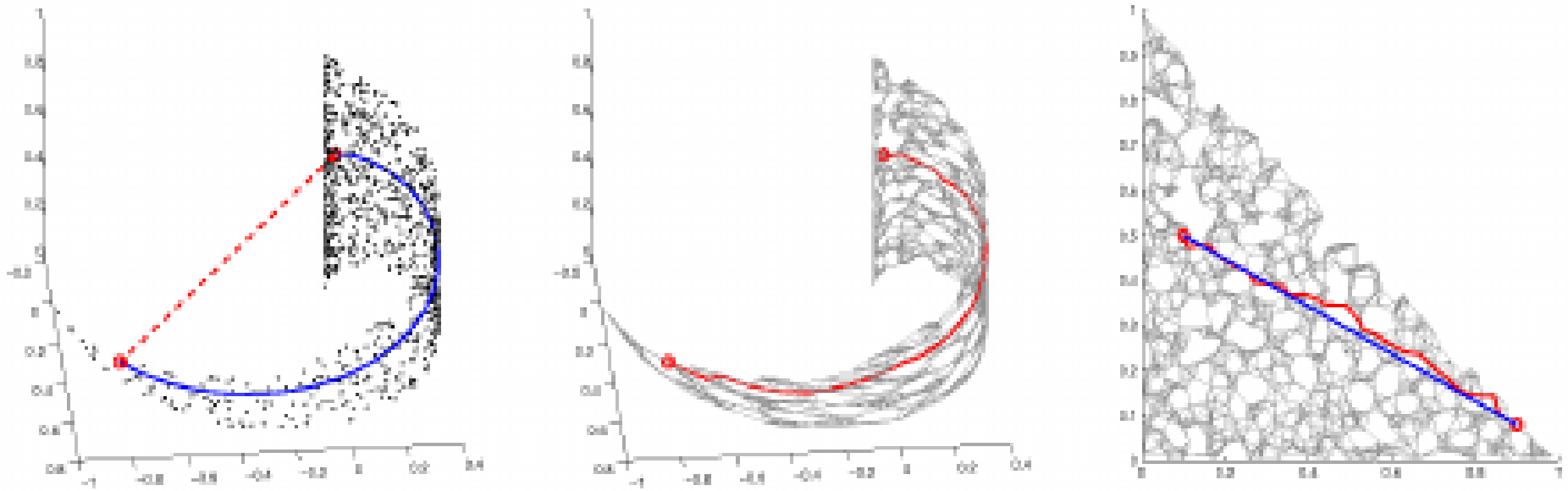


https://blog.paperspace.com/dimension-reduction-with-lle/

Laplacian Eigenmaps

- Graph-based approach

Distance defined by graph approximate the distance on manifold



Construct the data points as a ***graph***

Belkin, Mikhail, and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation." *Neural computation* 15, no. 6 (2003): 1373-1396.

Laplacian Eigenmaps

- If x_i and x_j are close in a high density region, then z_i and z_j are close to each other, i.e. We wish to choose $z_i \in \mathbb{R}$ to minimize

$$J = \frac{1}{2}\sum_{ij} w_{ij}(z_i - z_j)^2 = \mathbf{z}^T L\mathbf{z}$$

  这里$L = D - W$为Laplacian矩阵，显然也是半正定矩阵, D是对角阵，$D_{ii} = \sum_j W_{ji}$.

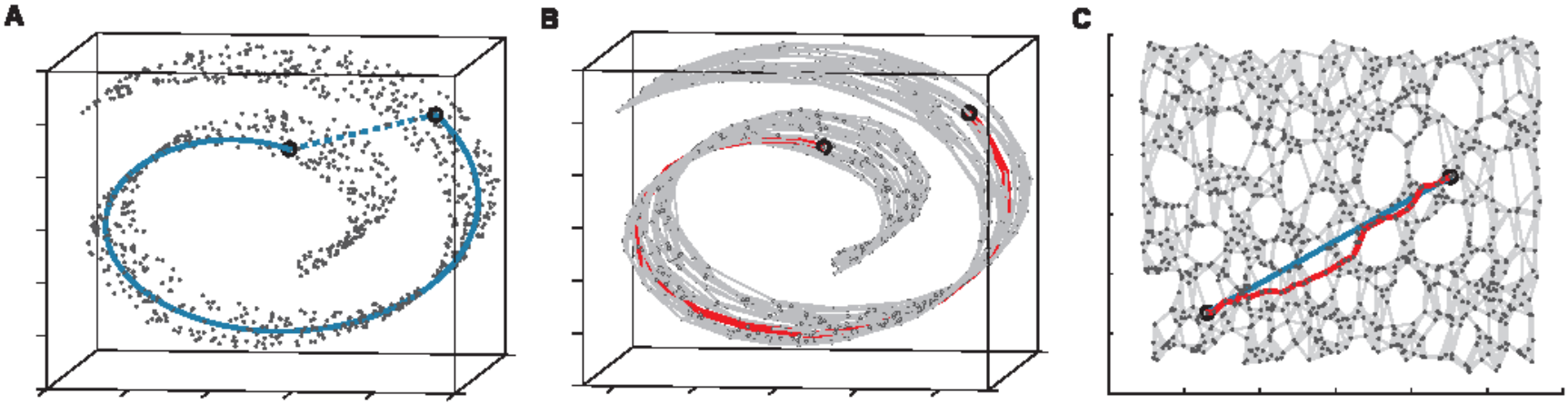
- 如何求$\mathbf{z}$? 显然当$\mathbf{z} = \mathbf{0}$，J取得最小值0. 不失一般性，可加上约束$\mathbf{z}^T D\mathbf{z} = 1$, 即

$$\min_{\mathbf{z}} \mathbf{z}^T L\mathbf{z}, \;\; s.t. \;\; \mathbf{z}^T D\mathbf{z} = 1$$

  利用拉格朗日乘子法，容易有$L\mathbf{z} = \lambda D\mathbf{z}$.

Belkin, Mikhail, and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation." *Neural computation* 15, no. 6 (2003): 1373-1396.

ISOMAP



- Constructing neighbourhood graph G
- For each pair of points in G, Computing shortest path distances ---- geodesic distances
- Use Classical MDS with geodesic distances
  - Euclidean distance→ Geodesic distance

Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." *Science* 290.5500 (2000): 2319-2323.

ISOMAP

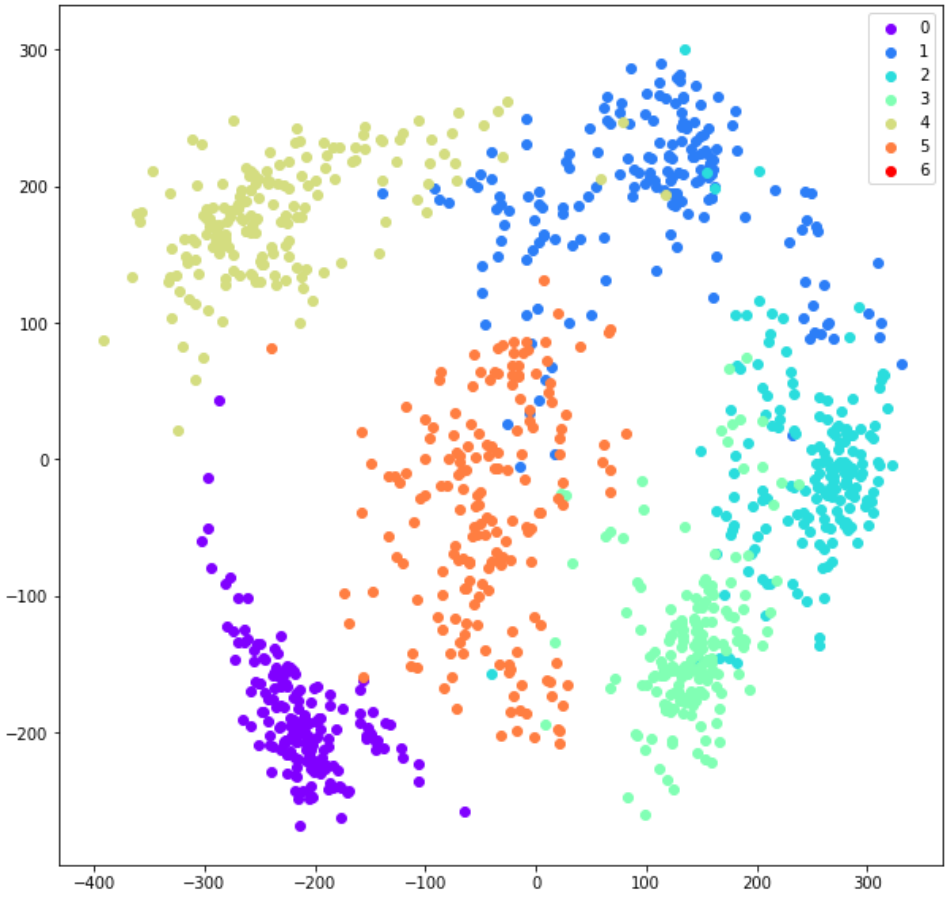
- ISOMAP算法等价于将经典的多维缩放 (Multi-dimensional Scaling, MDS)算法中的欧式距离替换成测地距离(Geodesic distance)
- MDS算法
  - 假定m个样本在原始d维空间中的距离矩阵为$\mathbf{D} \in R^{m \times m}$, 其第$i$行第$j$列元素$D_{ij}$为样本$x_i$和$x_j$的距离，目标是找到在低维空间的对应表示$z_i$和$z_j$，其在低维空间的距离等于原始空间中的距离（$\|z_i - z_j\| = \|x_i - x_j\| = D_{ij}$, 保距变换）
  - 令$\mathbf{Z} = [z_1, z_2, \cdots, z_m] \in R^{K \times m}$, $\mathbf{B} = \mathbf{Z}^T\mathbf{Z}$为降维后样本的内积矩阵，$B_{ij} = z_i^T z_j$, $D_{ij}^2 = (z_i - z_j)^T(z_i - z_j) = \|z_i\|^2 + \|z_j\|^2 - 2z_i^T z_j = B_{ii} + B_{jj} - 2B_{ij}$

ISOMAP

- MDS算法
  - 可以证明(课后自行证明, 假设降维后的样本$\mathbf{Z}$被中心化了，即$\sum_{i=1}^{m} z_i = \mathbf{0}$) $B_{ij} = -\frac{1}{2}(D_{ij}^2 - D_{i\cdot}^2 - D_{\cdot j}^2 + D_{\cdot\cdot}^2)$, 其中$D_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} D_{ij}^2$, $D_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} D_{ij}^2$, $D_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} D_{ij}^2$, 即可以通过原始的距离矩阵$\mathbf{D}$求出对应降维后样本的内积矩阵$\mathbf{B}$.
  - 通过原始的距离矩阵$\mathbf{D}$求出对应降维后样本的内积矩阵$\mathbf{B}$后, 如何求$\mathbf{Z}$?
  - 对$\mathbf{B}$进行奇异值分解，因为$\mathbf{B}$是对称阵，$\mathbf{B} = \mathbf{V}\mathbf{S}\mathbf{V}^T$, 则有$\mathbf{Z} = \mathbf{S}^{1/2}\mathbf{V}^T$，可取所有非零奇异值对应的特征向量

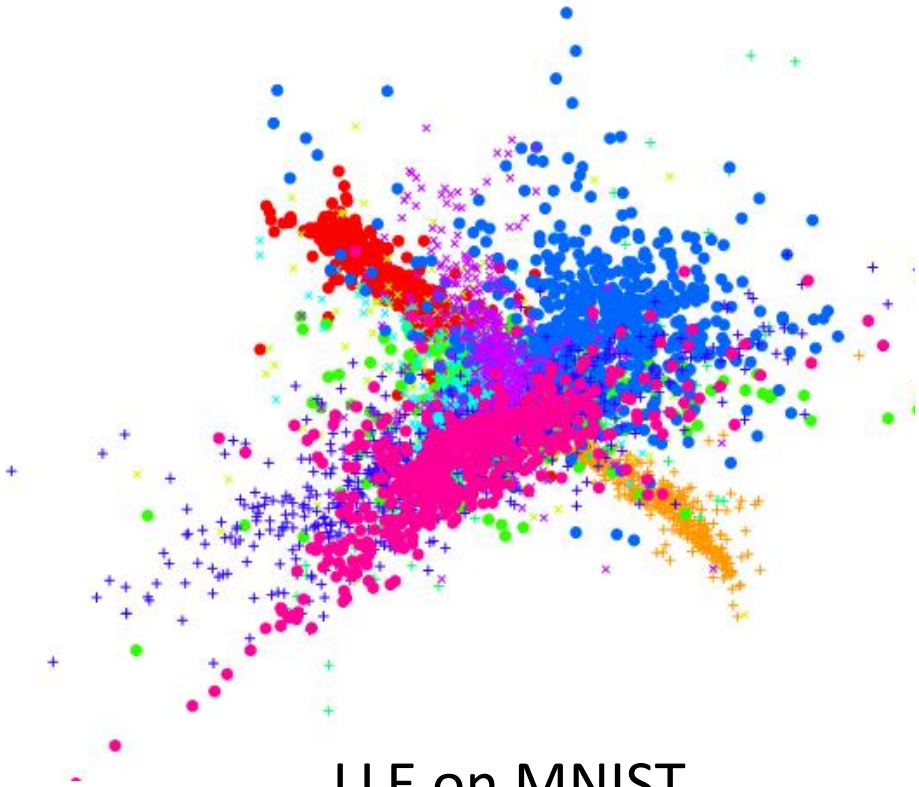
ISOMAP

Visualization of a subset of MNIST dataset after ISOMAP

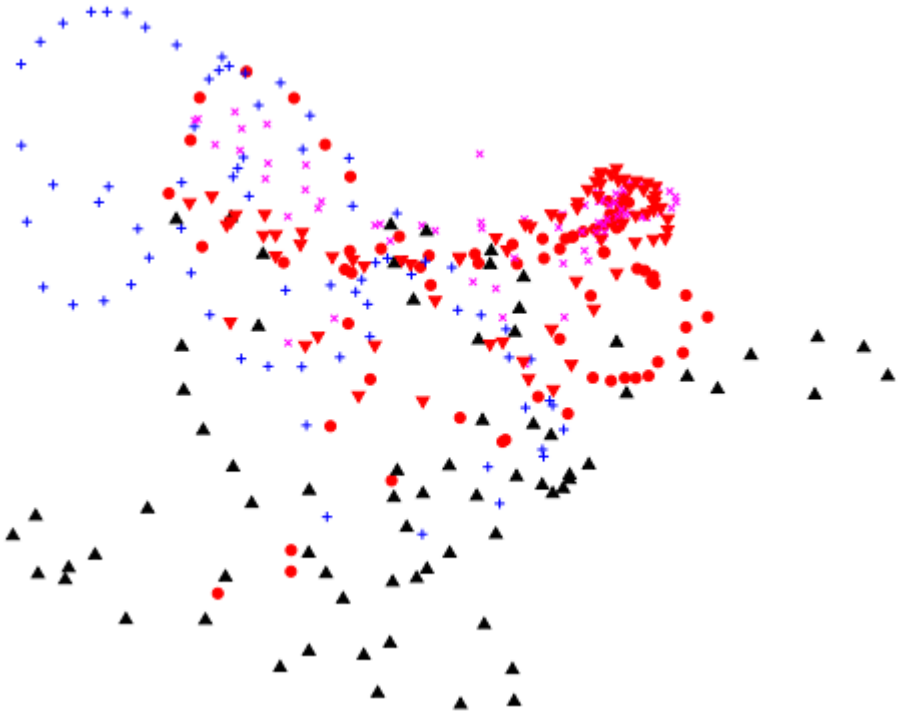


T-distributed Stochastic Neighbor Embedding (t-SNE)

- Problem of the previous approaches
  - Similar data are close, but different data may collapse



LLE on MNIST



LLE on COIL-20

t-SNE

Compute similarity between all pairs of x: $S(x_i, x_j)$

$$P(x_j|x_i) = \frac{S(x_i, x_j)}{\sum_{k \neq i} S(x_i, x_k)}$$

Compute similarity between all pairs of x: $S^{'}(z_i, z_j)$

$$Q(z_j|z_i) = \frac{S^{'}(z_i, z_j)}{\sum_{k \neq i} S^{'}(z_i, z_k)}$$

Find a set of z making the two distributions as close as possible

$$J = \sum_i KL(P(\cdot|x_i) \| Q(\cdot|z_i)) = \sum_i \sum_j P(x_j|x_i) \log \frac{P(x_j|x_i)}{Q(z_j|z_i)}$$

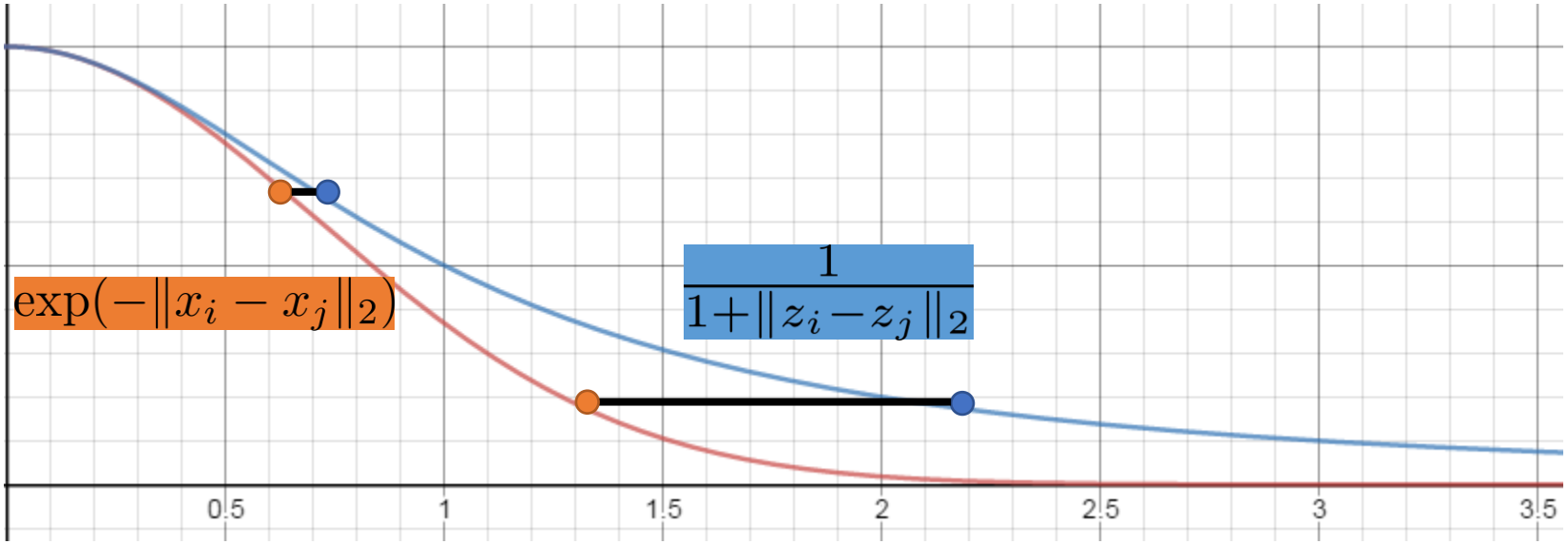
t-SNE –Similarity Measure

Ignore σ for simplicity

$$S(x_i, x_j) = \exp(-\|x_i - x_j\|_2)$$

SNE: $S'(z_i, z_j) = \exp(-\|z_i - z_j\|_2)$

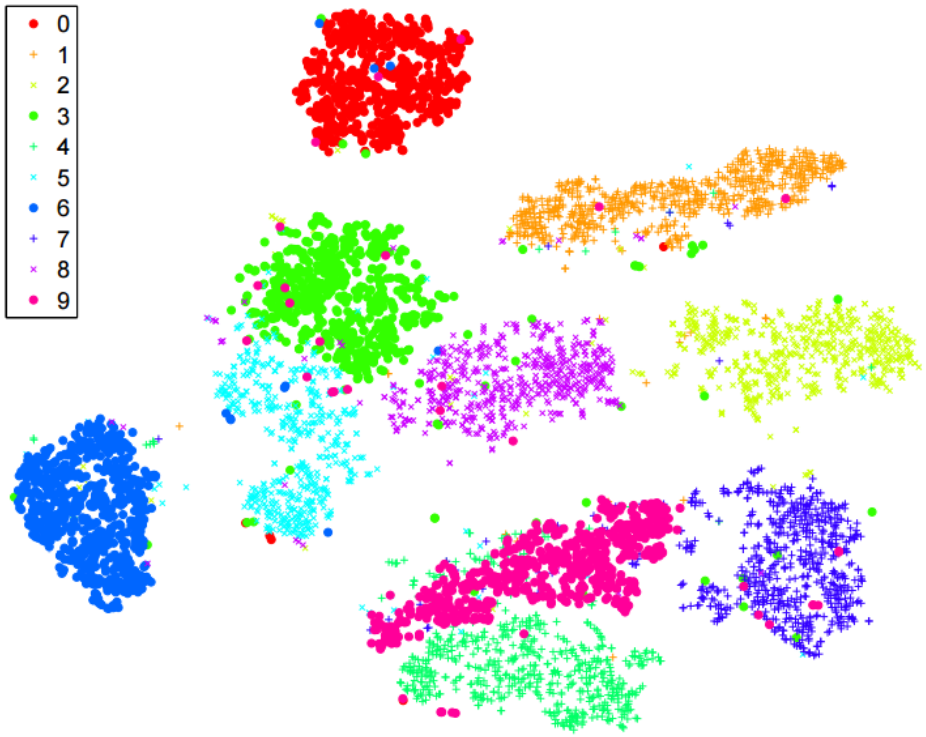
t-SNE: $S'(z_i, z_j) = \frac{1}{1+\|z_i - z_j\|_2}$



$\|x_i - x_j\|_2, \|z_i - z_j\|_2$

t-SNE

- Good at visualization



t-SNE on MNIST

t-SNE on COIL-20

t-SNE

Thanks!

Any questions?