十、降维与度量学习

主讲教师：周志华

k 近邻学习器

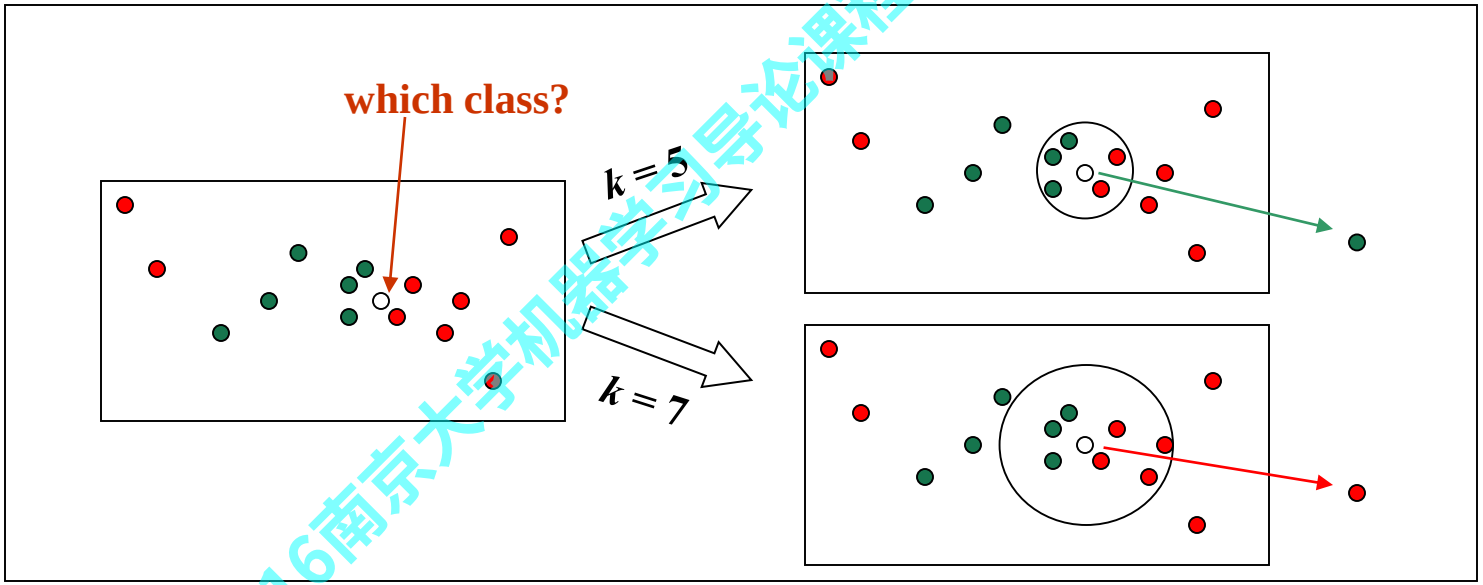
k 近邻 (k-Nearest Neighbor, kNN)

懒惰学习 (lazy learning) 的代表

基本思路：

近朱者赤，近墨者黑

(投票法；平均法)



关键： k 值选取；距离计算

最近邻学习器和贝叶斯最优分类器

给定测试样本$\boldsymbol{x}$，若其最近邻样本为$\boldsymbol{z}$,则最近邻分类器出错的概率就是$\boldsymbol{x}$和$\boldsymbol{z}$类别标记不同的概率，

$$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}).$$

$$\begin{aligned} P(err) &= 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}) \\ &\leq 1 - \sum_{c \in \mathcal{Y}} P^2(c \mid \boldsymbol{x}) \\ &\leq 1 - P^2(c^* \mid \boldsymbol{x}) \\ &= \left(1 + P\left(c^* \mid \boldsymbol{x}\right)\right)\left(1 - P\left(c^* \mid \boldsymbol{x}\right)\right) \\ &\leq 2 \times \left(1 - P\left(c^* \mid \boldsymbol{x}\right)\right). \end{aligned}$$

最近邻分离器的泛化错误率不会超过贝叶斯最优分类器错误率的两倍！

但是在真实的应用中，我们是否能够准确的找到k近邻呢？

维数灾难

但是在真实的应用中，我们是否能够准确的找到k近邻呢？

密采样(dense sampling)

如果近邻的距离阈值设为10^{-3}

假定维度为20，如果样本需要满足密采样条件
需要的样本数量近10^{60}

想象一下：一张并不是很清晰的图像：70余万维
我们为了找到恰当的近邻，需要多少样本？

维数灾难

高维空间给距离计算带来很大的麻烦

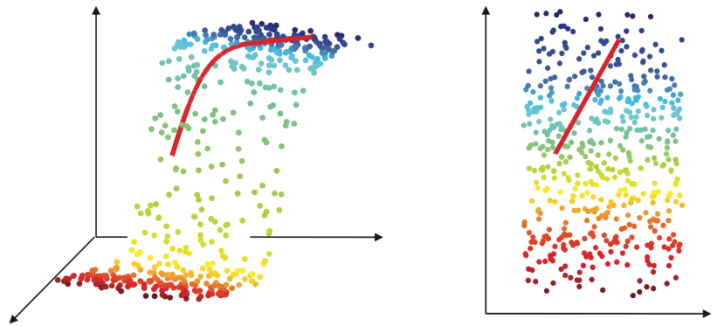
当维数很高时甚至连计算内积都不再容易

更严重的是：样本变得稀疏

⇒ 降维

为什么能进行降维？

数据样本虽是高维的，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)



(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

多维缩放方法（MDS）

MDS (Multiple Dimensional Scaling) 旨在寻找一个低维子空间，
样本在此子空间内的距离和样本原有距离尽量保持不变

考虑问题：如何在距离矩阵和内积矩阵之间建立联系？

考虑变形问题：如何在低维子空间和高维空间之间保持样本之间的内积不变？

$$dist_{ij}^2 = \|\mathbf{z}_i\|^2 + \|\mathbf{z}_j\|^2 - 2\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_j = b_{ii} + b_{jj} - 2b_{ij}$$

$$\sum_{i=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + mb_{jj}\,, \qquad dist_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} dist_{ij}^2\,,$$

$$\sum_{j=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + mb_{ii}\,, \qquad dist_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} dist_{ij}^2\,,$$

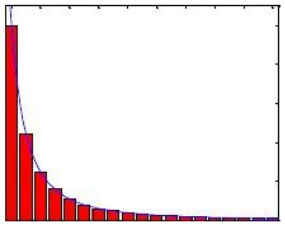
$$\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = 2m\,\mathrm{tr}(\mathbf{B})\,, \qquad dist_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2\,,$$

$$b_{ij} = -\frac{1}{2}(dist_{ij}^2 - dist_{i\cdot}^2 - dist_{\cdot j}^2 + dist_{\cdot\cdot}^2)$$

设样本之间的内积矩阵均为B
对B进行特征值分解：

$$\mathbf{B} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

由谱分解的数学性质，我们知道：



特征谱

谱分布长尾：存在相当数量的小特征值

$$\mathbf{Z} = \boldsymbol{\Lambda}_*^{1/2}\mathbf{V}_*^{\mathrm{T}} \in \mathbb{R}^{d^* \times m}$$
$$\mathbf{B} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{m \times m}$$

关键变量：距离、内积，保距

主成分分析 (Principal Component Analysis, PCA)

正交属性空间中的样本点，如何使用一个超平面对所有样本进行恰当的表达？

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

⇨ 主成分分析的两种等价推导

PCA - 最近重构性

对样本进行中心化：$\sum_i \boldsymbol{x}_i = \mathbf{0}$

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$\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$ $z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$

若基于$\boldsymbol{z}_i$来重构$\boldsymbol{x}_i$，则会得到 $\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j.$

PCA - 最近重构性（续）

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

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

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

$\boldsymbol{w}_j$ 是正交基，$\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵，于是由最近重构性，有：

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

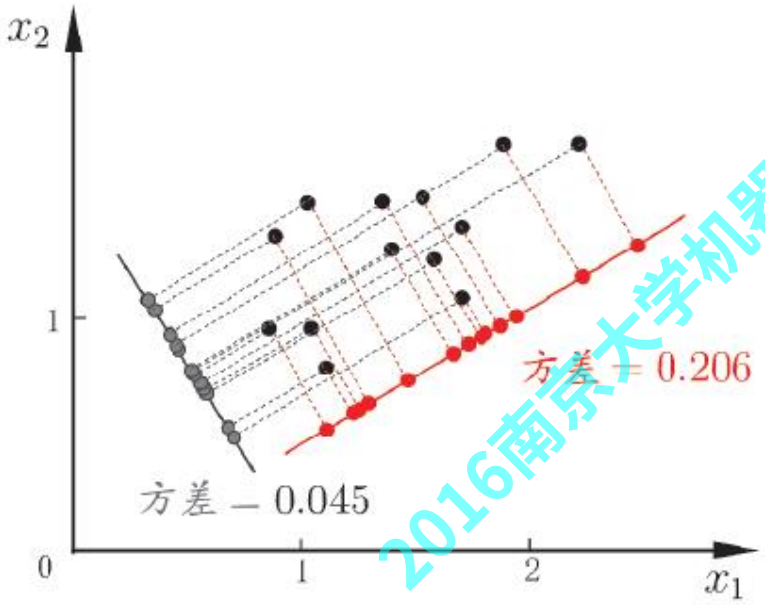
关键变量：重构误差

这就是主成分分析的优化目标

PCA - 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：

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

等价于：

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

PCA 求解

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

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

关键变量：子空间方差

PCA 应用

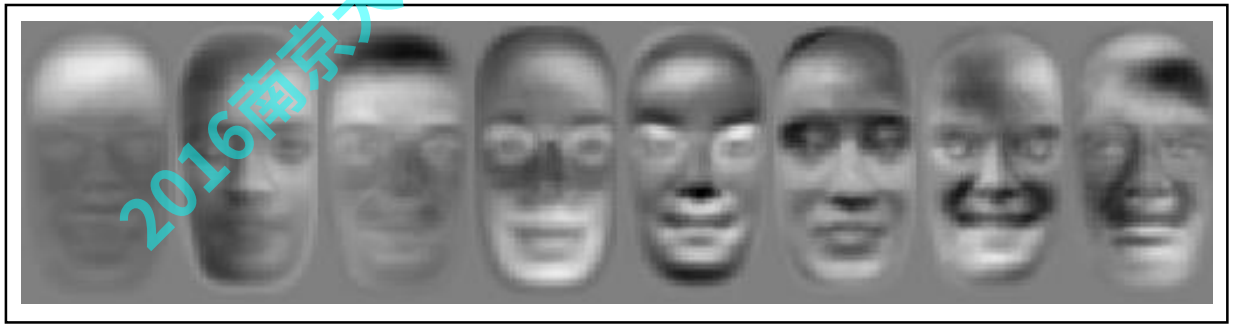
d' 的设置：

- 用户指定
- 在低维空间中对k近邻或其他分类器进行交叉验证
- 设置重构阈值，例如 t =95%，然后选取最小的 d' 使得 $\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t.$

PCA 是最常用的降维方法，在不同领域有不同的称谓

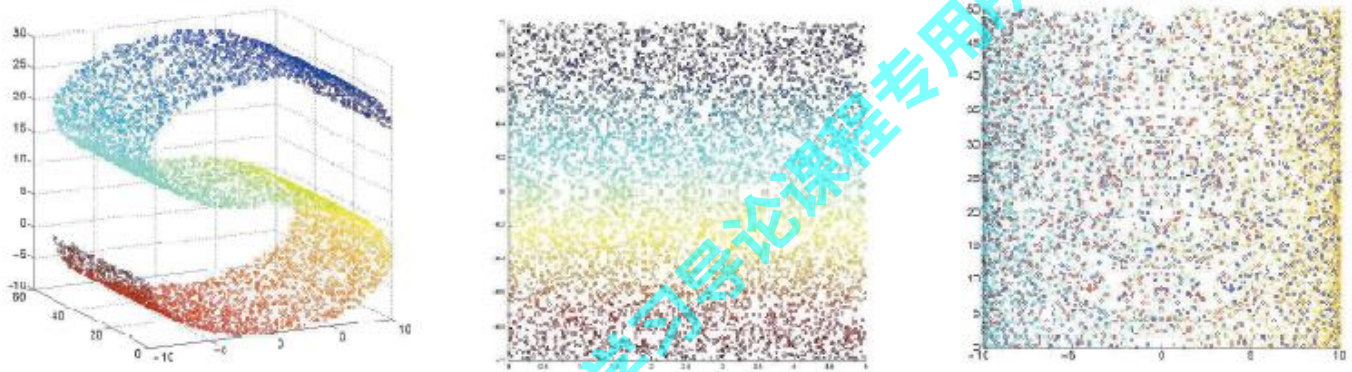
例如在人脸识别中该技术被称为“特征脸”(eigenface)

因为若将前 d' 个特征值对应的特征向量还原为图像，则得到



非线性降维

- 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
- 然而在许多现实任务中，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．

非线性降维的常用方法：

- 核化线性降维：如KPCA, KLDA, …
- 流形学习 (manifold learning)

核化PCA方法

首先，对PCA解的结构进行分析 $\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \lambda \mathbf{W}$ $\mathbf{W} = \frac{1}{\lambda}\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \frac{\boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}}{\lambda}$

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

假定z_i是由原始属性空间中样本点通过映射 ϕ 产生，即 $\mathbf{z}_i = \phi(\boldsymbol{x}_i), i = 1, 2, \ldots, m$

于是有

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

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

令 $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j)$ 可得 $\mathbf{K}\mathbf{A} = \lambda \mathbf{A}, \mathbf{A} = (\boldsymbol{\alpha_1}; \boldsymbol{\alpha_2}; \ldots; \boldsymbol{\alpha_m})$

于是

$$z_j = \boldsymbol{w}_j^{\mathrm{T}} \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i^j \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x})$$

$$= \sum_{i=1}^{m} \alpha_i^j \kappa(\boldsymbol{x}_i, \boldsymbol{x}) \,,$$

流形学习 - ISOMAP

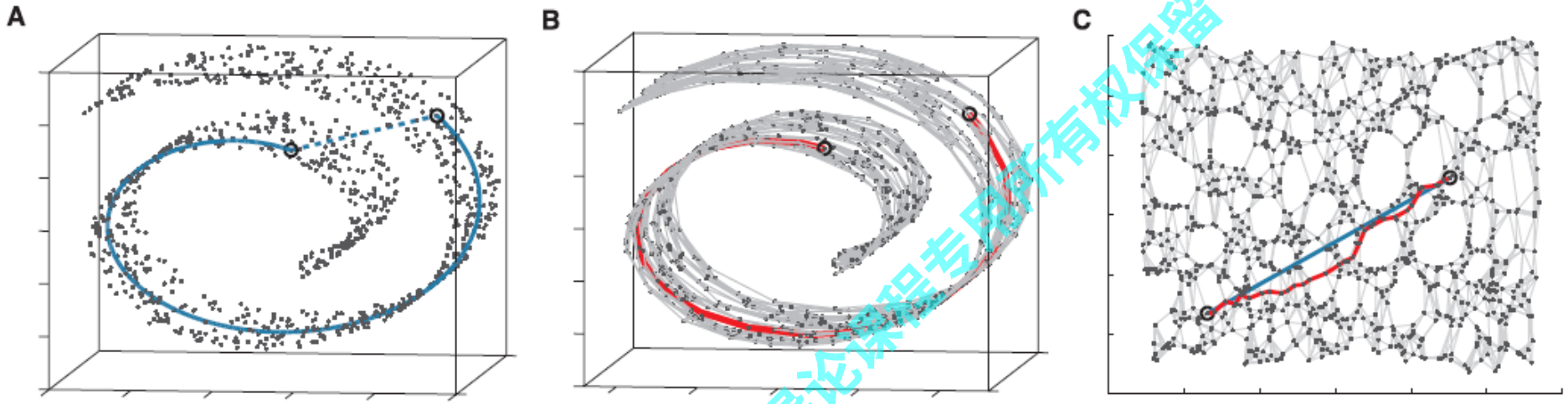


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

www.sciencemag.org SCIENCE VOL 290 22 DECEMBER 2000

基本步骤：

关键变量：测地线距离（近似）、保距

- 构造近邻图
- 基于最短路径算法近似任意两点之间的测地线(geodesic)距离
- 基于距离矩阵通过MDS获得低维嵌入

流形学习 - LLE

基本步骤：

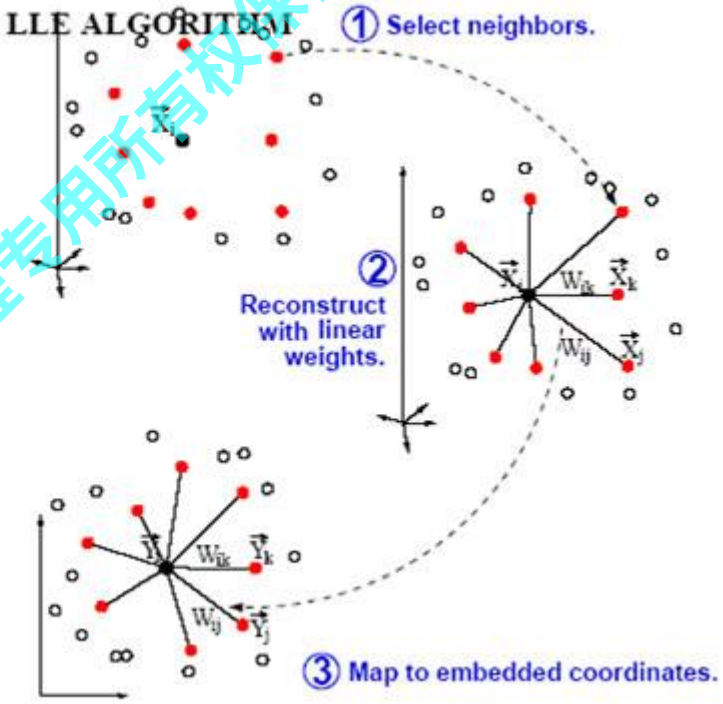
- 为每个样本构造近邻集合Q_i
- 为每个样本计算基于Q_i的线性重构系数

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

- 在低维空间中保持w_{ij}不变，求解下式

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

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in Equation (1) by constrained linear fits.
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in Equation (2) by its bottom nonzero eigenvectors.



www.sciencemag.org SCIENCE VOL 290 22 DECEMBER 2000

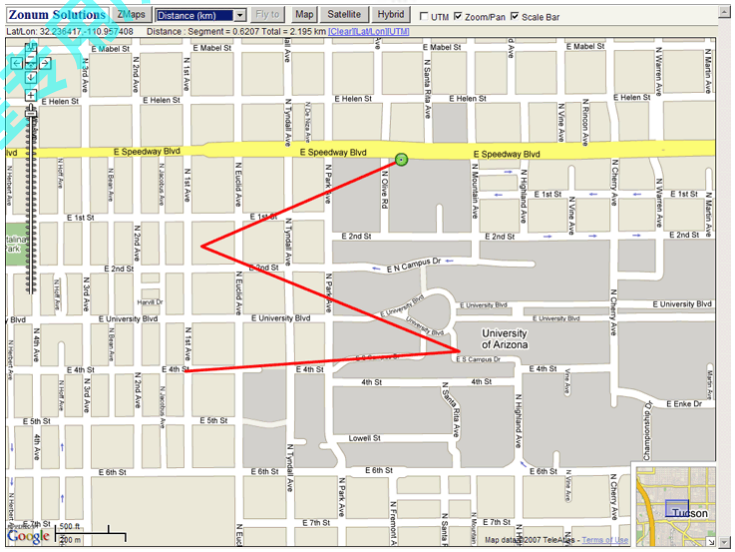
关键变量：重构权值

距离度量的种类

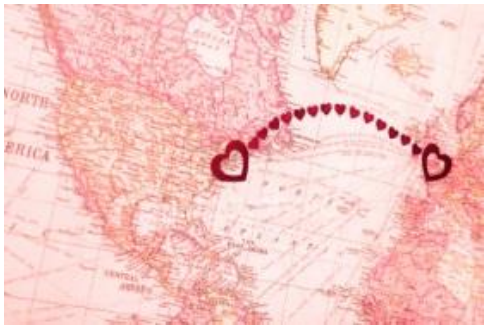
- Euclidean Distance

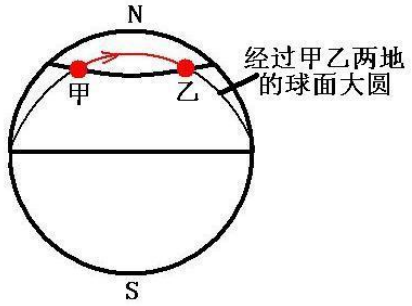
$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=i}^{n}(x_i - y_i)^2}$$

- Block Distance



- Geodesic Distance





距离度量学习 (distance metric learning)

降维的主要目的是希望找到一个“合适的”低维空间

每个空间对应了在样本属性上定义的一个距离度量

⇨ 能否直接“学出”合适的距离？

首先，要有可以通过学习来“参数化”的距离度量形式

马氏距离 (Mahalanobis distance) 是一个很好的选择：

$$\mathrm{dist}_{\mathrm{mah}}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}\mathbf{M}(\boldsymbol{x}_i - \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2$$

其中 $\mathbf{M}$ 是一个半正定对称矩阵，亦称“度量矩阵”

距离度量学习就是要对 $\mathbf{M}$ 进行学习

距离度量学习 (distance metric learning)

为什么要马氏距离 ？

我们回顾一下“什么是距离？”再思考一下“距离度量”

度量的是什么？

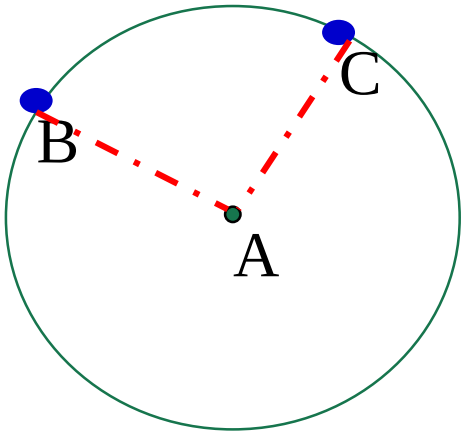
It’s a long distance to **walk**….

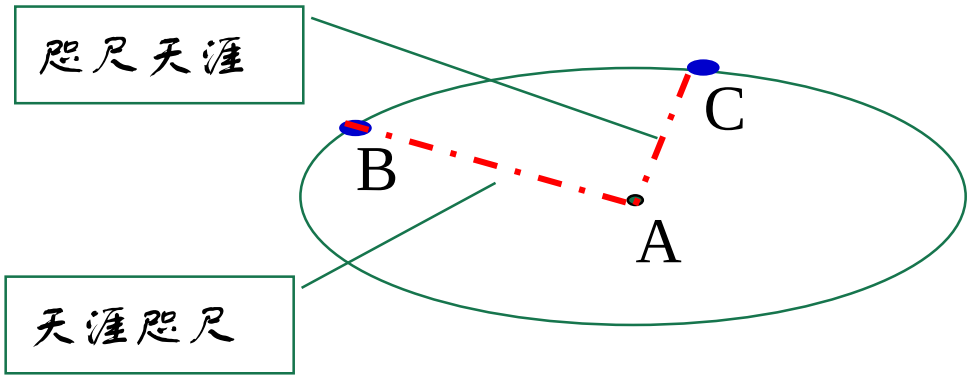
旅行的开销！

欧氏距离的缺陷 —— 各向同性

但是：

- 有缘千里来相会
  (欧氏距离大但开销少)
- 无缘对面手难牵
  (优势距离小但开销大)
- 马氏距离应运而生





距离度量学习 (distance metric learning)

其次，对 $\mathbf{M}$ 进行学习的目标是什么？

- 某种分类器的性能

  例如，若以近邻分类器的性能为目标，则得到 NCA

- 领域知识

  例如，若已知“必连”(must-link)约束集合$\mathcal{M}$与“勿连”(cannot-link)约束集合$\mathcal{C}$，则可通过求解这个凸优化问题得到 $\mathbf{M}$:

$$\begin{aligned}\min_{\mathbf{M}} & \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{M}} \|\boldsymbol{x}_i-\boldsymbol{x}_j\|_{\mathbf{M}}^2 \\ \text{s.t.} & \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in\mathcal{C}} \|\boldsymbol{x}_i-\boldsymbol{x}_k\|_{\mathbf{M}}^2 \geqslant 1 , \\ & \mathbf{M}\succeq 0 ,\end{aligned}$$

距离度量学习 – NCA：Neighborhood Component Analysis

近邻分类器在进行判别时通常使用多数投票法，替换为概率投票法：对于任意样本 x_j，它对 x_i 分类结果影响的概率为

$$p_{ij} = \frac{\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2\right)}{\sum_l \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_{\mathbf{M}}^2\right)},$$

x_i 样本的LOO正确率：

$$p_i = \sum_{j \in \Omega_i} p_{ij},$$

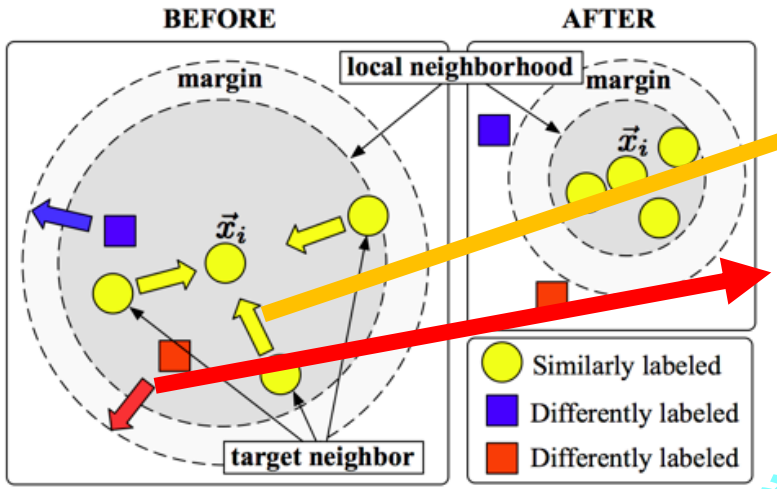
训练集上的LOO正确率：

$$\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} \sum_{j \in \Omega_i} p_{ij}.$$

NCA的优化目标：

$$\min_{\mathbf{P}} \quad 1 - \sum_{i=1}^{m} \sum_{j \in \Omega_i} \frac{\exp\left(-\left\|\mathbf{P}^{\mathrm{T}}\boldsymbol{x}_i - \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_j\right\|_2^2\right)}{\sum_l \exp\left(-\left\|\mathbf{P}^{\mathrm{T}}\boldsymbol{x}_i - \mathbf{P}^{\mathrm{T}}\boldsymbol{x}_l\right\|_2^2\right)}.$$

距离度量学习 – LMNN：Large Margin Nearest Neighbors



$$\varepsilon_{\text{pull}}(\mathbf{L}) = \sum_{j \rightsquigarrow i} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2$$

$$+$$

$$\varepsilon_{\text{push}}(\mathbf{L}) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

$$=$$

$$\varepsilon(\mathbf{L}) = (1-\mu)\,\varepsilon_{pull}(\mathbf{L}) + \mu\,\varepsilon_{push}(\mathbf{L})$$

Minimize $(1-\mu)\sum_{i,j\rightsquigarrow i}(\vec{x}_i-\vec{x}_j)^\top \mathbf{M}(\vec{x}_i-\vec{x}_j) + \mu\sum_{i,j\rightsquigarrow i,l}(1-y_{il})\xi_{ijl}$ **subject to:**

(1) $(\vec{x}_i-\vec{x}_l)^\top \mathbf{M}(\vec{x}_i-\vec{x}_l) - (\vec{x}_i-\vec{x}_j)^\top \mathbf{M}(\vec{x}_i-\vec{x}_j) \geq 1-\xi_{ijl}$

(2) $\xi_{ijl} \geq 0$

(3) $\mathbf{M} \succeq 0$.

前往第八站……