前沿计算研究实践 (II)	**2019 年春季**

专题 8：降维

教授：陈宝权　　作者：陈炤桦、王恒屹、王星瀚、王宇萱、魏安江、赵云飞

免责声明：该笔记尚未进行正式出版物的通常审查，未经教师允许不能向课程外分发.

降维是将高维度的数据保留下最重要的一些特征，去除噪声和不重要的特征，从而实现提升数据处理速度的目的。在实际的生产和应用中，降维在一定的信息损失范围内，可以为我们节省大量的时间和成本。降维也成为应用非常广泛的数据预处理方法。

8.1 主成分分析

PCA(Principal Component Analysis)，即主成分分析，是一种使用最广泛的数据降维算法。PCA 的主要思想是将 d 维特征映射到 k 维上，这 k 维是全新的正交特征也被称为主成分，是在原有 d 维特征的基础上重新构造出来的 k 维特征。

对向量 $\mathbf{x_1}, \cdots, \mathbf{x_n} \in \mathbb{R}^d$ 取均值

$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i}$$

令 $\mathbf{y_i} = \mathbf{x_i} - \mathbf{m}, Y^T = \{\mathbf{y_1}, \cdots, \mathbf{y_n}\}$，x 的散度矩阵可以写成

$$S = Y^T Y$$

显然 S 是实对称矩阵，可以正交对角化，有

$$S = V^T \Lambda V$$

其中 $\Lambda = diag\{\lambda_1, \cdots, \lambda_d\}, \lambda_1 \geq \cdots \geq \lambda_d, V = \{\mathbf{v_1}, \cdots, \mathbf{v_d}\}$

我们可以用这些特征向量来表示每个向量 $x = \mathbf{m} + a_1\mathbf{v_1} + \cdots + a_d\mathbf{v_d}$，我们还可以只选择前 k 个特征值对应的特征向量分量。因为前面的特征值 (方差) 比较大，数据在对应的特征向量方向上变化大，我们可以只选前面比较大的特征值对应的特征向量达到降维的目的。

8.2 Kernel PCA

PCA 有一些不尽如人意的地方，它只能对线性可分的数据进行分类，我们使用核函数将数据映射到高维空间然后在高维空间进行分类并降维。

选取映射

$$\Phi : \mathbb{R}^d \longrightarrow \mathbb{R}^{\hat{d}}$$

$$\mathbf{x} \longrightarrow \Phi(\mathbf{x})$$

继续写出散度矩阵，令 $X_\Phi^T = \{\Phi(\mathbf{x_1}), \cdots, \Phi(\mathbf{x_n})\}$

$$C = X_\Phi^T X_\Phi$$

事实上，我们并不需要显式地给出 $\Phi(x)$，我们可以通过利用计算核函数的方法求 C 的特征向量。核函数

$$K_{n\times n} := K_{ij} =< \Phi(\mathbf{x_i}), \Phi(\mathbf{x_j}) >$$

$$K = X_\Phi X_\Phi^T$$

由于 PCA 需要中心化，我们对它做零均值化，引入

$$J = I_{n\times n} - \mathbf{1}_n$$

$$Y_\Phi = JX_\Phi$$

$$\bar{K} = Y_\Phi Y_\Phi^T = K - \mathbf{1}_n K - K\mathbf{1}_n + \mathbf{1}K\mathbf{1}_n$$

然后对 $\bar{K}$ 进行 PCA，找出较大特征值对应的特征向量，而且

$$\bar{K}\alpha = Y_\Phi Y_\Phi^T = \lambda\alpha$$

$$Y_\Phi^T Y_\Phi Y_\Phi^T = \lambda Y_\Phi^T \alpha$$

可以知道 $v = Y_\Phi^T \alpha$ 是 $\bar{C} = Y_\Phi^T Y_\Phi$ 的特征值为 λ 的特征向量，然而 v 的表达式带有 Φ 我们仍无法写出，但我们其实关注的是 $\Phi(\mathbf{y_i})$ 在 v 方向的投影值

$$\frac{v^T}{||v||}\Phi(\mathbf{y_i}) = \frac{1}{\sqrt{\alpha^T Y_\Phi Y_\Phi^T \alpha}}\alpha^T Y_\Phi \Phi(\mathbf{y_i}) = \frac{\alpha^T}{\sqrt{\lambda}||\alpha||}\begin{pmatrix}\bar{K}(y_1, y_i)\\ \cdots \\ \bar{K}(y_n, y_i)\end{pmatrix}$$

8.3 核的选择

8.3.1 一些常用的核

- $K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\frac{1}{2\sigma^2}(\mathbf{x_i}-\mathbf{x_j})^2}$ 高斯核
- $K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma(\mathbf{x_i}-\mathbf{x_j})^2}$ RBF 核
- $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^p$ 多项式核
- $K(\mathbf{x_i}, \mathbf{x_j}) = tanh(\eta \mathbf{x_i} \cdot \mathbf{x_j} + v)$ sigmoid 核

8.3.2 普遍情况

虽然这些东西看着都没有问题，好像都可以看作内积，但我们内心还是非常不安，他们真的可以被写成内积吗?

是的。下面我们用高斯核举个例子，剩下的核也可用同样的方法证明它是核。我们可以对数据的每一维写出像，总的映射就是他们的直和

$$K(\mathbf{x_i},\mathbf{x_j})_k = e^{-\frac{1}{2\sigma^2}(x_{ik}-x_{jk})^2} = e^{\frac{-x_{ik}^2-x_{jk}^2}{2\sigma^2}} e^{\frac{x_{ik}x_{jk}}{\sigma^2}} = \Phi(\mathbf{x_i})^T\Phi(\mathbf{x_j})_k$$

我们有

$$e^{\frac{x_{ik}x_{jk}}{\sigma^2}} = \sum_{m=0}^{\infty}\frac{1}{m!}(\frac{x_{ik}}{\sigma})^m(\frac{x_{jk}}{\sigma})^m$$

我们能够得到

$$\Phi(\mathbf{x_i})_k = e^{\frac{-x_{ik}^2}{2\sigma^2}}\left(1,\sqrt{\frac{1}{1!}}\frac{x_{ik}}{\sigma},\sqrt{\frac{1}{2!}}(\frac{x_{ik}}{\sigma})^2,\cdots\right)$$

对更一般的情况，我们有 Mercer 定理

定理 8.1 ***Mercer* 定理**：设 $K(x,y)\in L^2(\mathbb{R}^d\times\mathbb{R}^d)$ 中的正定函数，那么存在一个函数序列 $\{\varphi_i(x)\}_{i=1}^{\infty}$ 使得 $K(x,y)=\sum_{i=1}^{\infty}\varphi_i(x)\varphi_i(y)$，在 L_2 范数下收敛。

8.4 多维缩放

8.4.1 定义

定义 8.2 多维缩放 将三维空间中的二维表面转换为二维平面，平面顶点之间的欧几里德距离与表面的测地线大致相同。

8.4.2 过程

我们可以通过快速行进算法计算给定表面的两个选定点之间的测地距离。

在原始三维空间中提高向量的维数，使高维空间中的欧几里德距离等于 d，以及两个数据之间的距离，以矩阵 E 表示。然后使用 J 表示 $\mathbf{X}$ 为零，对 $B=JEJ$ 做 PCA，$\hat{X}_{ij}$ 为第 j 个数据的第 i 个坐标值。

通常选择少量数据点用于降维，并且通过差分方法映射剩余数据点：尺寸减小平面中 (t，s) 的位置根据 (v 中的四个点的比例) 确定 (u,v) 平面。

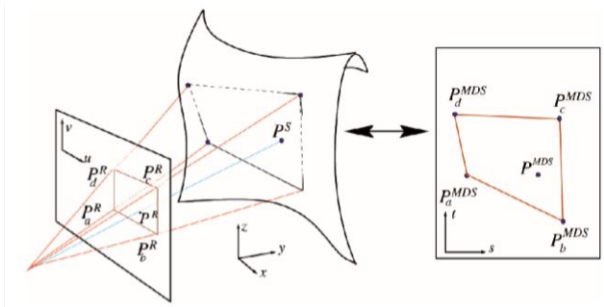
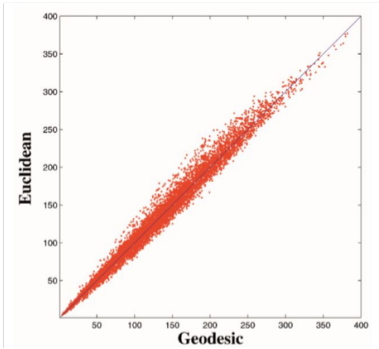


Fig. 7. A method for mapping between the surface and the flat plane. In our experiments the surface is given as a range image, i.e., a specific (u, v) parametric form. Left: The subgrid rectangle's vertices selected for the MDS procedure. Each vertex corresponds to a surface point in 3D. Right: After flattening, we find four corresponding points which are overlaid on the texture image.

8.4.3 结果与应用



8.5 方向包围盒

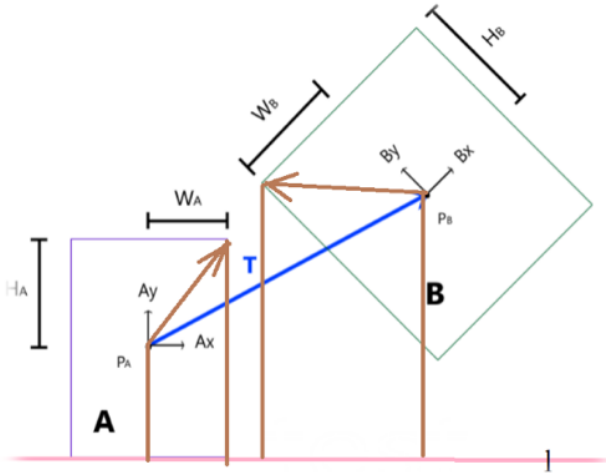
8.5.1 定义

定义 8.3 *OBB,* 即方向包围盒，总是沿物体的主成分方向生成一个最小的矩形边界框，可以与物体一起旋转，以便更准确地检测。

8.5.2 应用

$$|T \cdot L| > |(W_A Ax) \cdot L| + |(H_A Ay) \cdot L| + |(W_B Bx) \cdot L| + |(H_B By) \cdot L|$$

尝试 L 的四个值（两个矩形的四个方向），如果有一个满足条件，则没有碰撞。



8.6 Electors Voting for Fast Automatic Shape Correspondence

8.6.1 任务

找到曲线骨架的两组特征节点之间的 1-1 语义对应关系，这两个特征节点在语义上相似但可能在几何上非常不同。

8.6.2 解决方案

首先，它使用快速修剪测试执行组合搜索，以消除指数级的许多不良对应关系，留下仍然是一大组 (可能是数千个) 更可靠的对应关系，我们称之为 elector。其次，elector 对个别候选特征对进行投票，以建立最终的对应关系。

- 空间配置：通过使用 MDS 执行空间嵌入来减轻不同姿势的影响。
- 以节点为中心修剪：

$$c_p = \frac{1}{|P|} \sum_{p_i \in P} geo(p, p_i)$$

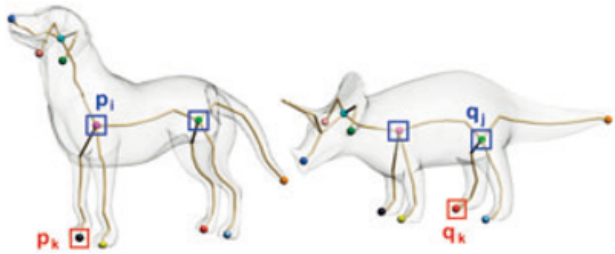
  较大的 c_p 值意味着 p 离形状中心更远。如果 (p_k, q_k) 的中心差异大于阈值，我们会拒绝它们。
- 按路径长度和路径半径修剪：如果满足以下任一条件，我们拒绝 (p_k, q_k)：

$$\frac{|geo(p_k, p_j) - geo(q_k, q_j)|}{(geo(p_k, p_j) + geo(q_k, q_j))/2} > \epsilon_C, \forall (p_j, q_j) \in \Omega$$

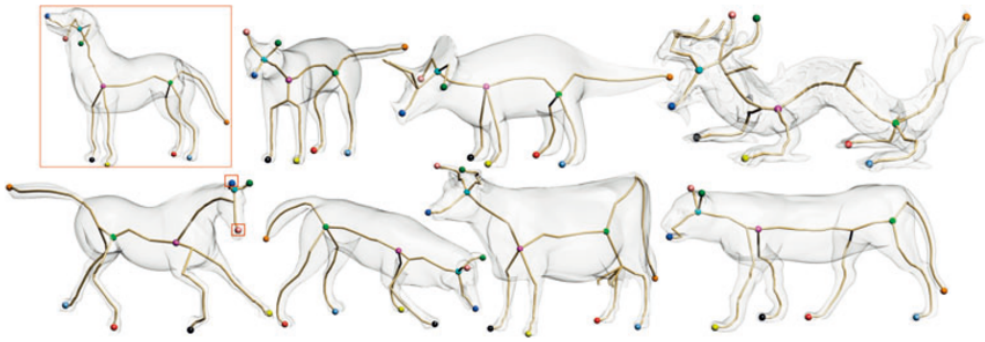
$$\frac{|rad(p_k, p_j) - rad(q_k, q_j)|}{(rad(p_k, p_j) + rad(q_k, q_j))/2} > \epsilon_C, \forall (p_j, q_j) \in \Omega$$

geo $(\cdot,\cdot)$ 和 rad $(\cdot,\cdot)$ 是沿着骨架路径的两个特征节点的测地距离和平均路径半径，并且它们都被标准化为 $[0,1]$。所以模型的大小不变。

- 按拓扑一致性修剪：为了保持类似的本地拓扑结构，我们拒绝 (p_k, q_k)。



- 投票过程：s_{ij} 是所有选择器中 (p_i, q_j) 对的总出现次数。所有特征对首先按其分数排序，并将具有最高分数的对迭代地添加到最终对应关系中。



8.7 非刚性表面匹配

8.7.1 任务描述

处理可变形物体的不同姿势：

- 非刚性曲面的匹配
- 形状检索

8.7.2 测地距离

不同姿势下的物体可视为表面的弯曲。弯曲的良好特性是它不会改变点之间的测地距离。因此，使用测地距离来描述表面的内在结构是一种自然的方法。

在三角域算法上快速行进：从 N 个点中选择 n 个点。算法可以计算 n 点之间的测地距离，其余部分计算在 $O(nN)$ 中。

8.7.3 三角化

在给定的表面上有无数的点，我们如何选择？

- 三角形表面。较短的三角形获得更准确的表示。
- 我们从顶点选择一个小子集来提高计算效率。

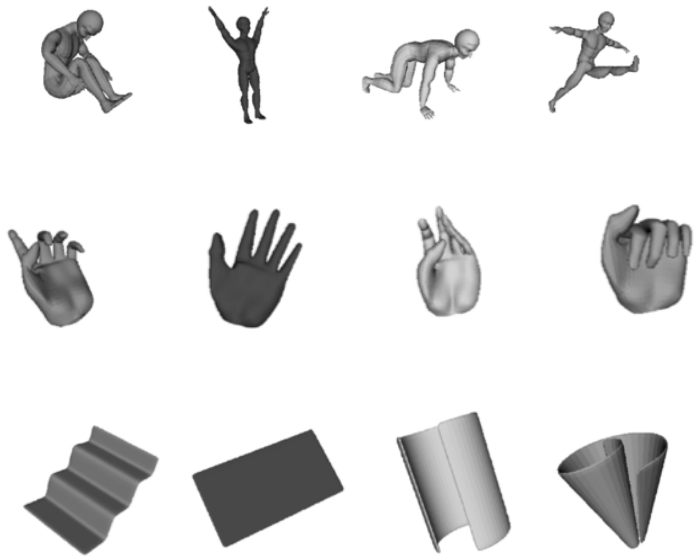
- Voronoi 采样：选择过程必须保持表面结构。在每次迭代中，我们选择与所选顶点的其余部分具有最大测地距离的顶点。
- 最后我们获得了一个 $n \times n$ 矩阵的测地距离。

8.7.4 MDS Mapping

- 设置尺寸 m
- Applicate MDS 算法，将顶点映射到 m 维 Euclid 空间
- 获取“签名表面”

8.7.5 匹配

现在我们已经将非刚性曲面的匹配问题转换为匹配的刚性曲面：我们只需要比较由 m 维空间中的坐标给出的签名曲面。一个简单的方法：让 M_i 成为曲面 S_i 的前几个矩的向量，然后将矩 - 距离矩阵 D_M 定义为 $[D_M]_{ij} = \left||M_i - M_j|\right|_2^2$

现在 D_M 描述了曲面之间的成对不相似性，我们再次应用 MDS，将每个曲面映射到新欧几里德空间中的一个点，然后应用像聚类，k-最近邻等算法。

8.7.6 结果