PCA算法推导

线性pca

A应用：

1、降维

2、表面法向量估计

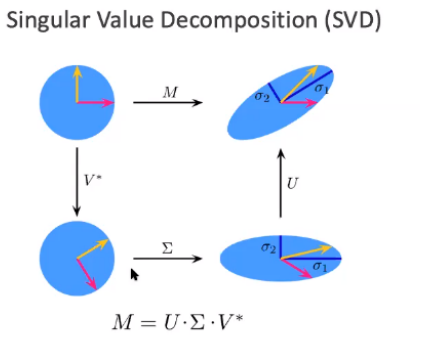
3、分类

4、关键点检测

5、特征描述

B基础概念：

一、奇异值分解



对于一个实矩阵M（m×n阶），如果可以分解为M＝UΣV’，其中U和Σ为分别为m×n与n×m阶正交阵，V为n×n阶对角阵，且Σ＝diag（a1,a2,...,ar,0,..., 0）。且有a1=a2=a3=...=ar=0.那么a1,a2,...,ar称为矩阵A的奇异值。正交矩阵：若一个方阵其行与列皆为正交的单位向量，则该矩阵为正交矩阵，且该矩阵的转置和其逆相等。两个向量正交的意思是两个向量的内积为 0

二、特征值分解、谱定理

针对对称阵：

Let $A \in R^{n,n}$ be symmetric, and $\lambda_i \in R, i = 1, 2, \cdots, n$ be the eigenvalues of A. There exists a set of orthonormal vectors $u_i \in R_n, i = 1, 2, \cdots, n$, such that $Au_i = \lambda_i u_i$. Equivalently, there exists an orthogonal matrix $U = [u_1, \cdots, u_n]$ (i.e., $UU^T = U^TU = I_n$), such that,

$$A = U\Lambda U^T = \sum_{i=1}^{n} \lambda_i u_i u_i^T, \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$$

三、瑞利熵

Given a symmertric matrix $A \in S^n$,

$$\boxed{\lambda_{\min}(A) \leq \frac{x^T A x}{x^T x} \leq \lambda_{\max}(A), \forall x \neq 0}$$

$$\lambda_{max}(A) = \max_{x:\|x\|_2=1} x^T A x$$

$$\lambda_{min}(A) = \min_{x:\|x\|_2=1} x^T A x$$

The maximum and minimum are attained for $x = u_1$ and for $x = u_n$, respectively, where u_1 and u_n are the largest and smallest eigenvector of A, respectively.

证明:

- Apply the spectral theorem, U is orthogonal, Λ is diagonal

$$x^T A x = x^T U \Lambda U^T x = \bar{x}^T \Lambda \bar{x} = \sum_{i=1}^{n} \lambda_i \bar{x}_i^2$$

- Obviously,

$$\lambda_{\min} \sum_{i=1}^{n} \bar{x}_i^2 \leq \sum_{i=1}^{n} \lambda_i \bar{x}_i^2 \leq \lambda_{\max} \sum_{i=1}^{n} \bar{x}_i^2$$

- Also, orthogonal matrix U doesn't change the norm of any vector

$$\sum_{i=1}^{n} x_i^2 = x^T x = x^T U U^T x = (U^T x)^T (U^T x) = \bar{x}^T \bar{x} = \sum_{i=1}^{n} \bar{x}_i^2$$

- Combining the above 3 equations,

$$\lambda_{\min} x^T x \leq x^T A x \leq \lambda_{\max} x^T x$$

C、PCA

过程:

- Normalize the data to be zero mean

$$\tilde{X} = [\tilde{x}_1, \cdots, \tilde{x}_m], \tilde{x}_i = x_i - \bar{x}, i = 1, \cdots, m \qquad \bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

- PCA is to get largest variance when projected to a direction $z \in \mathbb{R}^n$, $\|z\|_2 = 1$

$$\alpha_i = \tilde{x}_i^T z, i = 1, \cdots, m$$

- The mean variance of the projections is

$$\frac{1}{m} \sum_{i=1}^{m} \alpha_i^2 = \frac{1}{m} \sum_{i=1}^{m} z^T \tilde{x}_i \tilde{x}_i^T z = \frac{1}{m} z^T \tilde{X} \tilde{X}^T z$$

- So, maximize it,

$$\max_{z \in R^n} z^T (\tilde{X} \tilde{X}^T) z, \text{s.t.:} \|z\|_2 = 1$$

- Now, maximize this

$$\max_{z \in R^n} z^T (\tilde{X} \tilde{X}^T) z, \text{s.t.:} \|z\|_2 = 1$$

- Recall the Rayleigh Quotients $\lambda_{\min}(A) \leq \frac{x^T A x}{x^T x} \leq \lambda_{\max}(A), \forall x \neq 0$

- Recall our Spectral Theorem $A = U \Lambda U^T = \sum_{i=1}^{n} \lambda_i u_i u_i^T, \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$

- Apply to PCA

$$H = \tilde{X} \tilde{X}^T = U_r \Sigma^2 U_r^T$$

- First principle vector $z_1 = u_1$, u_1 is the first column of U_r

z1是u1：Σ对角线元素是分量比重，对角线元素最大值对应的u1即是最大分量，即对应z1

- Let's take a look at $H = \tilde{X}\tilde{X}^T = U_r\Sigma^2 U_r^T$
- Perform SVD on $\tilde{X}$: $\tilde{X} = U_r \Sigma V_r^T = \sum_{i=1} \sigma_i u_i v_i^T$

Spectral Theorem and SVD are closely related

- Find z_2 by deflation

$$\tilde{x}_i^{(1)} = \tilde{x}_i - u_1(u_1^T \tilde{x}_i), i = 1, \cdots, m$$

$$\tilde{X}^{(1)} = [\tilde{x}_1^{(1)}, \cdots, \tilde{x}_m^{(1)}] = (I_n - u_1 u_1^T)\tilde{X}$$

- Combine the above equations:

$$\begin{aligned}\tilde{X}^{(1)} &= \sum_{i=1}^{r} \sigma_i u_i v_i^T - (u_1 u_1^T) \sum_{i=1}^{r} \sigma_i u_i v_i^T \\ &= \sum_{i=1}^{r} \sigma_i u_i v_i^T - \sum_{i=1}^{r} \sigma_i u_1 u_1^T u_i v_i^T \\ &= \sum_{i=1}^{r} \sigma_i u_i v_i^T - \sigma_1 u_1 v_1^T \quad //\text{U is orthogonal} \\ &= \sum_{i=2}^{r} \sigma_i u_i v_i^T\end{aligned}$$

---

u是正交阵

- We have removed the first components, finding z_2 is by

$$\max_{z \in R^n} z^T(\tilde{X}^{(1)}\tilde{X}^{(1)T})z, \text{s.t.:} \|z\|_2 = 1$$

$$\tilde{X}^{(1)} = \sum_{i=2}^{r} \sigma_i u_i v_i^T$$

- The result is simply $z_2 = u_2$, u_2 is the 2nd column of U_r

- $z_3, \cdots z_m$ can be found by similar deflation.

总结：

Given $x_i \in \mathbb{R}^n, i = 1,2, \cdots m$, perform PCA by:

1. Normalized by the center

$$\tilde{X} = [\tilde{x}_1, \cdots, \tilde{x}_m], \tilde{x}_i = x_i - \bar{x}, i = 1, \cdots, m \qquad \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i.$$

2. Compute SVD $H = \tilde{X}\tilde{X}^T = U_r\Sigma^2 U_r^T$
3. The principle vectors are the columns of U_r
(Eigenvector of X = Eigenvector of H)

应用：pca降维、升维　　eigenface

2、核pca

非线性基底的定义φ（xi）

- Input data $x_i \in \mathbb{R}^{n_0}$, non-linear mapping $\phi: \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}$
- Follow the standard Linear PCA on the lifted space $\mathbb{R}^{n_1}$
  1. Assume $\phi(x_i)$ is already zero-center
  $$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) = 0$$
  2. Compute correlation matrix
  $$\tilde{H} = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi^T(x_i)$$
  3. Solve the eigenvectors/eigenvalues by $\tilde{H}\tilde{z} = \tilde{\lambda}\tilde{z}$

- Note that eigenvectors can be expressed as linear combination of features
$$\tilde{z} = \sum_{j=1}^{N}\alpha_j\phi(x_j)$$

- Proof:
$$\tilde{H}\tilde{z} = \tilde{\lambda}\tilde{z}$$
$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\underbrace{\phi^T(x_i)\tilde{z}}_{\text{scalar}} = \tilde{\lambda}\tilde{z}$$

- Find the eigenvector $\tilde{z}$ = find the coefficient α_j

- Put that linear combination into $\tilde{H}\tilde{z} = \tilde{\lambda}\tilde{z}$
$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi^T(x_i)\left(\sum_{j=1}^{N}\alpha_j\phi(x_j)\right) = \tilde{\lambda}\sum_{j=1}^{N}\alpha_j\phi(x_j)$$
$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\left(\sum_{j=1}^{N}\alpha_j\phi^T(x_i)\phi(x_j)\right) = \tilde{\lambda}\sum_{j=1}^{N}\alpha_j\phi(x_j)$$
- Let's define kernel function $k(x_i, x_j) = \phi^T(x_i)\phi(x_j)$
$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\left(\sum_{j=1}^{N}\alpha_j k(x_i, x_j)\right) = \tilde{\lambda}\sum_{j=1}^{N}\alpha_j\phi(x_j)$$

- Multiply both sides by $\phi(x_k), k = 1, \cdots, N$,
$$\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_j k(x_k, x_i)k(x_i, x_j) = N\tilde{\lambda}\sum_{j=1}^{N}\alpha_j k(x_k, x_j), k = 1, \cdots, N$$

核函数定义：φ的内积即可简化

- Now define the Gram matrix $K \in \mathbb{R}^{N\times N}, K(i,j) = k(x_i, k_j)$
  - K is symmetric because $k(x_i, x_j) = k(x_j, x_i)$
- The above equation can be written as
$$K^2\alpha = N\tilde{\lambda}K\alpha$$
- Remove K on both sides
$$K\alpha = N\tilde{\lambda}\alpha$$
$$K\alpha = \lambda\alpha$$
- Again, get the eigenvectors α_r and eigenvalues $\lambda_r, r = 1, \cdots, l$
- However, we have to ensure that $\tilde{z}$ is unit vector
  Note that we are solving the linear PCA in the feature space
$$\tilde{H}\tilde{z} = \tilde{\lambda}\tilde{z} \qquad \tilde{z} = \sum_{j=1}^{N}\alpha_j\phi(x_j)$$

- The normalization of $\tilde{z}$ leads to

$$1 = \tilde{z}_r^T \tilde{z}_r$$
$$1 = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ri}\alpha_{rj}\phi^T(x_i)\phi(x_j)$$
$$1 = \alpha_r^T K \alpha_r$$

- Note that $K\alpha = \lambda\alpha$, we have $\alpha_r^T \lambda_r \alpha_r = 1, \forall r$
- That is, normalize α_r to be norm $1/\lambda_r$
- Now, the r^{th} principle vector in the lifted space is given below, which is unknown

$$\tilde{z}_r = \sum_{j=1}^{N} \alpha_{rj}\phi(x_j)$$

解决核函数均值为0的问题，将核函数做一转换：

- Normalize $\phi(x_i)$ to be zero mean

$$\tilde{\phi}(x_i) = \phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)$$

- The normalized kernel $\tilde{k}(x_i, x_j)$ is given by

$$\begin{aligned}\tilde{k}(x_i, x_j) &= \tilde{\phi}^T(x_i)\tilde{\phi}(x_j) \\ &= \left(\phi(x_i) - \frac{1}{N}\sum_{k=1}^{N}\phi(x_k)\right)^T \left(\phi(x_j) - \frac{1}{N}\sum_{l=1}^{N}\phi(x_l)\right) \\ &= k(x_i, x_j) - \frac{1}{N}\sum_{k=1}^{N}k(x_i, x_k) - \frac{1}{N}\sum_{k=1}^{N}k(x_j, x_k) + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}k(x_k, x_l)\end{aligned}$$

- In the matrix form $\tilde{K} = K - 2\mathbb{1}_{\frac{1}{N}}K + \mathbb{1}_{\frac{1}{N}}K\,\mathbb{1}_{\frac{1}{N}}$, where $\mathbb{1}_{\frac{1}{N}}(i,j) = \frac{1}{N}, \forall i, j$

常用的核函数：用实验确定选用哪种核函数

- Kernel choices
  - Linear $k(x_i, x_j) = x_i^T x_j$
  - Polynomial $k(x_i, x_j) = (1 + x_i^T x_j)^p$
  - Gaussian $k(x_i, x_j) = e^{-\beta\|x_i - x_j\|_2}$
  - Laplacian $k(x_i, x_j) = e^{-\beta\|x_i - x_j\|_1}$

核pca的总结：

- Select a kernel $k(x_i, x_j)$, compute the Gram matrix $K(i,j) = k(x_i, x_j)$
- Normalize K

$$\tilde{K} = K - 2\mathbb{1}_{\frac{1}{N}}K + \mathbb{1}_{\frac{1}{N}}K\,\mathbb{1}_{\frac{1}{N}}$$

- Solve the eigenvector/eigenvalues of $\tilde{K}$

$$\tilde{K}\alpha_r = \lambda_r\alpha_r$$

- Normalize α_r to be $\alpha_r^T\alpha_r = \frac{1}{\lambda_r}$
- For any data point $x \in \mathbb{R}^n$, compute its projection onto r^{th} principle component $y_r \in \mathbb{R}$

$$y_r = \phi^T(x)\tilde{z}_r = \sum_{j=1}^{N}\alpha_{rj}k(x, x_j)$$

3、法向量

pca的应用：投影最小的方向，pca最小特征值对应的投影向量，曲率：最小特征值对应特征值之和

- So we normalize the data points by its center, similar to what we did in PCA proof.

$$\tilde{X} = [\tilde{x}_1, \cdots, \tilde{x}_m], \tilde{x}_i = x_i - \bar{x}, i = 1, \cdots, m$$

- Now the problem becomes,

$$\min_{n \in R^n} \sum_{i=1}^{m} (\tilde{x}_i^T n)^2, \text{s.t.: } \|n\|_2 = 1$$

$$\min_{n \in R^n} \sum_{i=1}^{m} n^T \tilde{x}_i \tilde{x}_i^T n, \text{s.t.: } \|n\|_2 = 1$$

$$\min_{n \in R^n} n^T \left(\sum_{i=1}^{m} \tilde{x}_i \tilde{x}_i^T \right) n, \text{s.t.: } \|n\|_2 = 1$$

$$\min_{n \in R^n} n^T \tilde{X} \tilde{X}^T n, \text{s.t.: } \|n\|_2 = 1$$

PCA:

$$\boxed{\max_{z \in R^n} z^T (\tilde{X} \tilde{X}^T) z, \text{s.t.:} \|z\|_2 = 1}$$

法向量就是最小特征值对应的法向量

带权重的法向量估计、CVPR深度学习法向量估计

4、滤波

1、SOR滤波：

1. For each point, find a neighborhood
2. Compute its distance to its neighbors $d_{ij}, i = [1, \dots, m], j = [1, \dots, k]$
3. Model the distances by Gaussian distribution $d \sim N(\mu, \sigma)$

$$\mu = \frac{1}{nk} \sum_{i=1}^{m} \sum_{j=1}^{k} d_{ij}, \quad \sigma = \sqrt{\frac{1}{nk} \sum_{i=1}^{m} \sum_{j=1}^{k} (d_{ij} - \mu)^2}$$

4. For each point, compute its mean distance to its neighbors
5. Remove the point, if the mean distance is outside some confidence according to the Gaussian distribution
   E.g. Remove if

$$\sum_{j=1}^{k} d_{ij} > \mu + 3\sigma \text{ or } \sum_{j=1}^{k} d_{ij} < \mu - 3\sigma$$

2、降采样：

1. Compute the min or max of the point set $\{p_1, p_2, \cdots p_N\}$
   $$x_{max} = \max(x_1, x_2, \cdots, x_N), x_{min} = \min(x_1, x_2, \cdots, x_N), y_{max} = \cdots\cdots$$
2. Determine the voxel grid size r
3. Compute the dimension of the voxel grid
   $$D_x = (x_{max} - x_{min})/r$$
   $$D_y = (y_{max} - y_{min})/r$$
   $$D_z = (z_{max} - z_{min})/r$$
4. Compute voxel index for each point
   $$h_x = \lfloor (x - x_{min})/r \rfloor$$
   $$h_y = \lfloor (y - y_{min})/r \rfloor$$
   $$h_z = \lfloor (z - z_{min})/r \rfloor$$
   $$h = h_x + h_y * D_x + h_z * D_x * D_y$$
5. Sort the points according to the index in Step 4
6. Iterate the sorted points, select points according to Centroid / Random method
   0, 0, 0, 0, 3, 3, 3, 8, 8, 8, 8, 8, 8, 8, 8, ...

trick：格子数计算 选择2^32还是2^64

提升：不用排序、采用哈希表，哈希函数将

1. Compute the min / max of each coordinate
2. Determine the voxel grid size r
3. Compute the dimension of the voxel grid
4. Compute voxel index for each point h
5. Use a *hash* function to map each point to a container G_i in $\{G_1, G_2, ..., G_M\}$
6. Iterate $\{G_1, G_2, ..., G_M\}$ and get M point!

- That *hash* function is

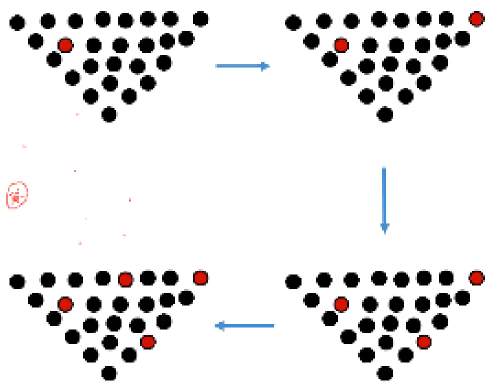
$$\text{hash}(h_x, h_y, h_z) : \mathbb{R}^3 \to \mathbb{R}$$

E.g., $hash(h_x, h_y, h_z) = (h_x + h_y * D_x + h_z * D_x * D_y)\ \%\ container\_size$

将排序需要的格子空间映射到哈希表，不用排序

3、FPS最远点采样：去除中心聚集点

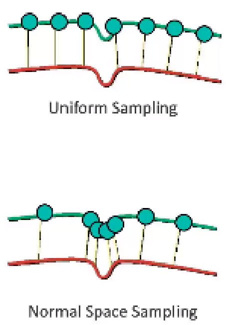
1. Randomly choose a point to be the first FPS point
2. Iterate until we get the desired number of points
   a. For each point in the original point cloud, compute its distance to the nearest FPS point
   b. Choose the point with the largest value, add to FPS set



4、NSS降采样：法相空间采样,配准的时候保留法相特征

Used in Iterative Closest Point

1. Construct a set of buckets in the normal space
2. Put all points into bucket according to the surface normals
3. Uniformly pick points from all buckets until we have the desired number of points

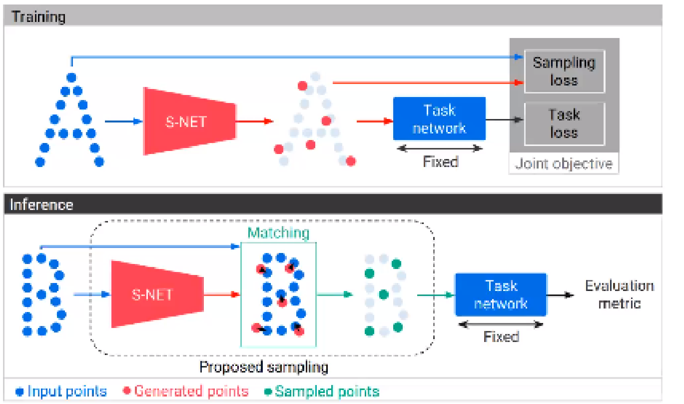


第一步：各个方向建立容器、第二部：根据法向量选择容器

第三：每个方向容器内点云数量一定

5、深度学习降采样：

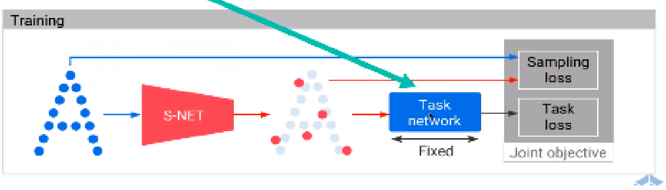
降采样之后的点尽可能保证原来的语义



Problem statement Given a point set $P = \{p_i \in \mathbb{R}^3, i = 1, \ldots, n\}$, a sample size $k \leq n$ and a task network T, find a subset S^* of k points that minimizes the task network's objective function f:

$$S^* = \underset{S}{\operatorname{argmin}}\, f(T(S)), \quad S \subset P, \quad |S| = k \leq n. \quad (1)$$

Sampling based on Semantics



再加入集合约束

5上采样

双边滤波：两个高斯核　　BF双边滤波

Given image I, for each pixel p, find its neighbor S.

each pair (p, q)

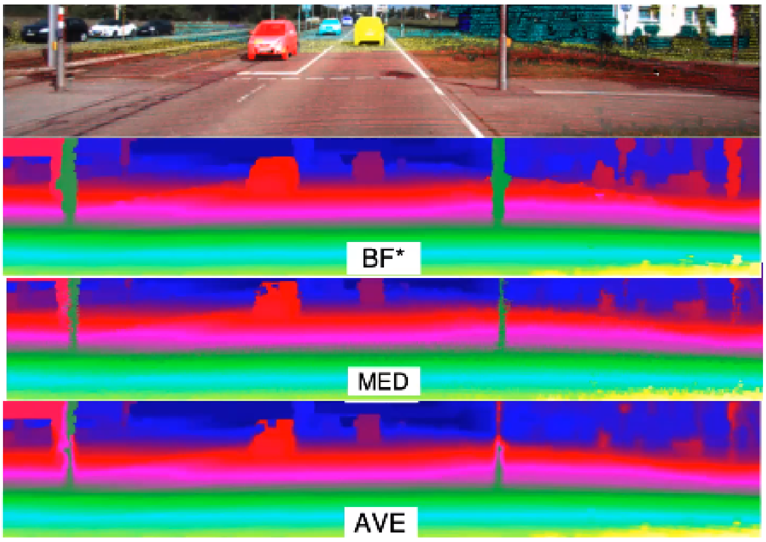
- Compute distance weight G_{σ_s} intensity weight G_{σ_r}

$$G_\sigma(x) = \frac{1}{2\pi\,\sigma^2}\ \exp\left(-\frac{x^2}{2\,\sigma^2}\right)$$

- Apply Bilateral Filter to get intensity of pixel p

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})\, I_{\mathbf{q}}$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})$$



高斯滤波：高斯核