1. $x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $x_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ $x_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$

(a)

$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\quad \tilde{x}_j = x_j - \mu = x_j$ For $1 \le j \le 4$

we can then form the matrix $X = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4] = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \in R^{d \times n}$

the covariance matrix $C \in R^{d \times d} = \frac{1}{n} X X^T = \frac{1}{4} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$

$C = UDU^T$

calculating the eigenvalues and eigenvectors of C gives

$\det\left(\begin{bmatrix} \frac{1}{2} - \lambda & 0 \\ 0 & \frac{1}{2} - \lambda \end{bmatrix}\right) = 0 \quad (\frac{1}{2} - \lambda)^2 = 0 \Rightarrow \lambda_1 = \frac{1}{2} \quad \lambda_2 = \frac{1}{2}$

$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ we obtain $u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

therefore the principal components of the data are

$u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad u_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

(b) Kernel function $K(x_i, x_j) = (x_i^T x_j)^r$, where $r = 10$

Compute the Gram matrix $K \in R^{n \times n}$ where $K_{ij} = K(x_i, x_j)$

$K = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

$\tilde{K} = K - 1_n K - K 1_n + 1_n K 1_n$

where $1_n = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$

$\tilde{K} = \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \end{bmatrix}$

Solving the eigen problem

$\tilde{K} a_i = \lambda_i n a_i$

where $n = 4$

$\lambda_1 = 0.5$ $\lambda_2 = 0$

gives $n\lambda_1 = 2$ $n\lambda_2 = 0$

$$u_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \qquad u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad u_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad u_4 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

To find a_i that $\|a_i\|^2 = a_i^T a_i = 1/\lambda_i n$

so $\|a_1\|^2 = \frac{1}{2}$ $\lambda_1 = 0.5$ $\left(\frac{1}{2\sqrt{2}}\right)^2 = \frac{1}{4\times 2} = \frac{1}{8}$

$$a_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \cdot \frac{1}{2\sqrt{2}}$$

Hence the projected lower-dimensional data is given by the inner product of each datapoint with the first principal component a_i

$$y_j = \sum_{l=1}^{n} \tilde{k}_{jl}\, a_{1l}$$

$y_1 = (0.5 \times -1 + -0.5 \times 1 + 0.5 \times -1 + -0.5 \times 1)/2\sqrt{2} = -\frac{\sqrt{2}}{2}$

$y_2 = (-0.5 \times -1 + 0.5 \times 1 + -0.5 \times -1 + 0.5 \times 1)/2\sqrt{2} = \frac{\sqrt{2}}{2}$

$y_3 = (0.5 \times -1 + -0.5 \times 1 + 0.5 \times -1 + -0.5 \times 1)/2\sqrt{2} = -\frac{\sqrt{2}}{2}$

$y_4 = (-0.5 \times -1 + 0.5 \times 1 + -0.5 \times -1, 0.5 \times 1)/2\sqrt{2} = \frac{\sqrt{2}}{2}$

therefore the projected datapoints $\{y_i \in R : 1 \le j \le 4\}$ are

$$\left\{ -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right\}$$

2. $X \in R^{d\times n}$ $\mu=0$ in X

(a) Covariance matrix $C = \frac{1}{n} X X^T \in R^{d\times d}$

Gram matrix $K = \frac{1}{n} X^T X \in R^{n\times n}$

(1) Eigendecomposition of C gives $C = U \Lambda U^T$

Consider $\max u^T C u$ for a unit vector u

then we have $u^T C u = u^T U \Lambda U^T u = (U^T u)^T \Lambda U^T u$

where $U^T u \in R^d$ is obtained by projecting u onto eigenvectors of C

U is orthogonal therefore $U^T u$ is also a unit vector $\|U^T u\|_2 = 1$

$\max u^T C u$ is achieved when u is aligned with the eigenvector that corresponds to the largest eigenvalue of C (i.e. λ_1)

and $\max u^T C u = \lambda_1$

(2) Similarly eigendecomposition of K gives $K = V D V^T$

$v^T K v = v^T V D V^T v = (V^T v)^T D V^T v$

where $V^T v \in R^n$ by projecting v onto eigenvectors of K

V is orthogonal, therefore $V^T v$ is a unit vector.

$\max v^T K v$ is achieved when v is aligned with the eigenvector corresponding to the largest eigenvalue of K (i.e. r_1)

and $\max v^T K v = r_1$

C and K have the same non-zero eigenvalues, therefore $\lambda_1 = r_1$

thus we can show that $\max u^T C u = \max v^T K v$

$\blacksquare$

(b) In (a) we've shown that $\max u^T C u = \max v^T K v$

where $\|u\|_2 = \|v\|_2 = 1$

the maximum value is achived by aligning u and v with the eigenvectors that corresponds to the largest eigenvalues of C and K respectively.

Let u be the eigenvectors of C that $Cu = \lambda u$ ①

v be the eigenvectors of K that $Kv = \Gamma v$ ②

rewriting ① $Cu = \lambda u$

$$\Rightarrow \frac{1}{n} X X^T u = \lambda u$$

$$\Rightarrow \frac{1}{n} X^T X X^T u = \lambda X^T u \quad \text{Multiplying both side with } X^T$$

let $v = X^T u$, then we obtain $\frac{1}{n} X^T X v = \lambda v$

which is $Kv = \lambda v$

thus the eigenvectors of K are equal to the projections of the eigenvectors of C onto the column space of X, therefore representing the same directions in the data.

Therefore PCA can be performed using Gram matrix instead of Covariance matrix. This way is more computationally efficient as gram matrix is much smalled than covariance matrix when $d << n$.

■

(c) In kernel PCA, we replace the covariance matrix with a kernel matrix K, where $K_{i,j} = K(x_i, x_j)$

and in this case $K(x_i, x_j) = x_i^T x_j$

So with (b), to show that in this case kernel PCA reduces to standard PCA, we need to show that the kernel matrix is the same as gram matrix

gram matrix $k = \frac{1}{n} X^T X$ where $k_{ij} = \frac{1}{n} x_i^T x_j$

In Kernel matrix K, $K_{ij} = K(x_i, x_j) = x_i^T x_j$

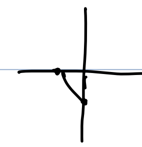
Therefore we see that gram matrix k and kernel matrix K are equal up to a scaling factor of $\frac{1}{n}$ which won't affect the eigenvalue and eigenvectors. According to (b), PCA can be performed using gram matrix instead of covariance matrix.

Therefore kernel PCA reduces to PCA when kernel function $K(x_i, x_j) = x_i^T x_j$

$\blacksquare$

2. $x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $x_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ $x_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$

(a)



Distance matrix $D \in R^{4\times 4}$

$D_{11} = 0$ $D_{12} = \|x_1 - x_2\|^2 = (\sqrt{2})^2 = 2$ $D_{13} = 4$ $D_{14} = (\sqrt{2})^2 = 2$

$D_{21} = D_{12} = 2$ $D_{22} = 0$ $D_{23} = 2$ $D_{24} = 4$

$D_{31} = D_{13} = 4$ $D_{32} = D_{23} = 2$ $D_{33} = 0$ $D_{34} = 2$

$D_{41} = D_{14} = 2$ $D_{42} = D_{24} = 4$ $D_{43} = D_{34} = 2$ $D_{44} = 0$

$$D = \begin{bmatrix} 0 & 2 & 4 & 2 \\ 2 & 0 & 2 & 4 \\ 4 & 2 & 0 & 2 \\ 2 & 4 & 2 & 0 \end{bmatrix}$$

(b) MDS

Compute $B = -\frac{1}{2} HDH$, where $H = I_N - \frac{1}{N} ee^T$

where $N = 4$, $e \in R^{N\times 1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot [1, 1, 1, 1] = \begin{bmatrix} 0.75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix}$$

$$B = -\frac{1}{2} HDH = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} = XX^T$$

eigendecomposition gives $B = UDU^T$

where

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad U = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}$$

$\hat{U} = u_1$ $\hat{D} = \text{diag}(\lambda_1) = 2$

Y is given by $Y = D^{\frac{1}{2}} \hat{U}^T$

$$Y = \sqrt{2} \cdot \left[-\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0 \right]$$

therefore the scalar output data points $\{y_j \in R : 1 \leq j \leq 4\}$ are

$y_1 = -1$, $y_2 = 0$, $y_3 = 1$ $y_4 = 0$

∎

Q4. IsoMap and LLE

① IsoMap computes pair-wise geodesic distance between datapoints, this could be very computationally expensive especially for those far apart on the manifold.

while LLE recovers global non-linear structure using locally linear fits of the data, which is computationally efficient due to sparse matrices.

② LLE requires less memory than IsoMap as LLE is a local approach that only stores the distance between K nearest neighbours of each datapoints, while IsoMap stores pairwise distances between all datapoints. ∎