PCA with Kernel

~S S Roy, 27th Sept,2025

1. Introduction

Principal Component Analysis (PCA):

- Finds **linear directions of maximum variance** in data.
- Reduces dimensionality while keeping most information.

Limitation:

- PCA is **linear**; it cannot capture **nonlinear patterns** (e.g., circular, spiral).

Kernel PCA (KPCA):

- Maps data into a **high-dimensional feature space** $\phi(x)$.
- Performs PCA in feature space.
- Uses **kernel trick**: compute similarity without explicitly mapping $\phi(x)$.

Difference from PCA:

Feature	PCA	Kernel PCA
Linearity	Linear	Nonlinear
Covariance	Covariance matrix in input space	Kernel (Gram) matrix in feature space

Feature	PCA	Kernel PCA
Feature mapping	Input space	High-dimensional feature space via kernel
Handles nonlinear patterns	No	Yes

---

2. Variables

Variable	Meaning
x_i	i-th sample (row) of dataset
n	Number of samples (rows)
$K(x_i, x_j)$	Kernel function (similarity between x_i and x_j)
K	Kernel matrix ($n \times n$)
H	Centering matrix: $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$
K_c	Centered kernel matrix: $K_c = HKH$
v	Eigenvector of K_c
λ	Eigenvalue of K_c
y_{ik}	Projection of sample i along principal component k: $y_{ik} = \sum_j v_{jk} K(x_i, x_j)$

---

3. Tabular Dataset

Sample	F1	F2
x1	1	0
x2	0	1
x3	1	1

- $x_1 = [1, 0]$, $x_2 = [0, 1]$, $x_3 = [1, 1]$

4. Polynomial Kernel Example

Kernel: Polynomial, degree 2

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^2$$

Step 1: Compute kernel matrix K

i\j	x1	x2	x3
x1	(11+00+1)^2=4	(10+01+1)^2=1	(11+01+1)^2=4
x2	1	4	4
x3	4	4	9

$$K = \begin{bmatrix} 4 & 1 & 4 \\ 1 & 4 & 4 \\ 4 & 4 & 9 \end{bmatrix}$$

Step 2: Center kernel matrix K_c

Centering matrix:

$$H = I_3 - \frac{1}{3}\mathbf{1}\mathbf{1}^T = \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{bmatrix}$$

$$K_c = HKH$$

Step 2a: Compute row and column means

- Row means:
  $\text{row1} = (4 + 1 + 4)/3 = 3$
  $\text{row2} = (1 + 4 + 4)/3 = 3$
  $\text{row3} = (4 + 4 + 9)/3 = 5.667$
- Column means: same as row means because K is symmetric
- Total mean: $(4 + 1 + 4 + 1 + 4 + 4 + 4 + 4 + 9)/9 = 3.556$

Step 2b: Center each element

$$K_{c_{ij}} = K_{ij} - \text{row mean}_i - \text{col mean}_j + \text{total mean}$$

i\j	x1	x2	x3
x1	4-3-3+3.556=1.556	1-3-3+3.556=-1.444	4-3-5.667+3.556=-0.111
x2	1-3-3+3.556=-1.444	4-3-3+3.556=1.556	4-3-5.667+3.556=-0.111
x3	4-5.667-5.667+3.556=-3.778	4-5.667-5.667+3.556=-3.778	9-5.667-5.667+3.556=1.222

$$K_c \approx \begin{bmatrix} 1.556 & -1.444 & -0.111 \\ -1.444 & 1.556 & -0.111 \\ -3.778 & -3.778 & 1.222 \end{bmatrix}$$

Step 3: Eigen-decomposition

Solve $K_c v = \lambda v$

- Eigenvalues: $\lambda_1 \approx 4.3, \lambda_2 \approx 0.1, \lambda_3 \approx 0$
- Eigenvector for largest eigenvalue (normalized): $v_1 \approx [-0.707, -0.707, 0]^T$

Step 4: Projection onto first principal component

$$y_i = \sum_{j=1}^{3} v_j K(x_i, x_j)$$

Sample	Calculation	y (1st PC)
x1	-0.7074 + -0.7071 + 0*4	-3.535
x2	-0.7071 + -0.7074 + 0*4	-3.535
x3	-0.7074 + -0.7074 + 0*9	-5.657

✅ Nonlinear principal component obtained.

---

5. RBF Kernel Example

Kernel: Gaussian RBF, $\sigma = 1$

$$K(x_i, x_j) = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\Big)$$

Step 1: Compute distances $\|x_i - x_j\|^2$

i\j	x1	x2	x3
x1	0	(1-0)^2+(0-1)^2=2	(1-1)^2+(0-1)^2=1
x2	2	0	1
x3	1	1	0

Step 2: Compute kernel matrix

$$K_{ij} = \exp(-\|x_i - x_j\|^2/2)$$

i\j	x1	x2	x3
x1	1	0.368	0.607
x2	0.368	1	0.607
x3	0.607	0.607	1

Step 3: Center kernel

- Row means: x1: (1+0.368+0.607)/3=0.658
- Column means: same
- Total mean: (1+0.368+0.607+0.368+1+0.607+0.607+0.607+1)/9=0.654

$$K_{c_{ij}} = K_{ij} - \text{row mean}_i - \text{col mean}_j + \text{total mean}$$

i\j	x1	x2	x3
x1	1-0.658-0.658+0.654=0.338	0.368-0.658-0.658+0.654= -0.294	0.607-0.658-0.658+0.654=-0.055
x2	-0.294	0.338	-0.044
x3	-0.055	-0.044	0.099

Step 4: Eigen-decomposition

- Eigenvalues: $\lambda_1 \approx 0.65, \lambda_2 \approx 0.12, \lambda_3 \approx 0$
- Eigenvector for largest eigenvalue: $v_1 \approx [-0.707, -0.707, 0]^T$

Step 5: Projection

$$y_i = \sum_j v_j K(x_i, x_j)$$

Sample	y (1st PC)
x1	-0.45
x2	-0.45
x3	0.9

✅ First nonlinear principal component obtained.

---

6. Summary

- Kernel PCA extends PCA to **nonlinear patterns** via kernels.
- **Steps:** Choose kernel → Compute K → Center → Eigen-decompose → Project.
- Works for **tabular data**, **numerical/categorical after encoding**.
- **Kernels:** Polynomial, RBF.
- Extracts **nonlinear principal components** for dimensionality reduction, visualization, or preprocessing.