26 Kernel PCA: Unlocking Nonlinear Dimensions

26.1 Revisiting PCA: The Linear Lens

26.1.1 PCA Refresher: Variance, Covariance, and Eigenvectors

As shown in Figure 1, the typical workflow of Principal Component Analysis (PCA) can be understood intuitively as a series of steps to find the directions in which the data varies the most. First, we start with a dataset $\boldsymbol{X}$ of shape $n \times D$, where n is the number of samples and D is the number of features.

Often, we standardize the data by subtracting the mean and dividing by the standard deviation for each feature. This step ensures that all features are on a comparable scale and prevents those with larger variance from dominating the analysis. In some cases, simply centering the data (subtracting the mean so that the dataset has zero mean) is sufficient.

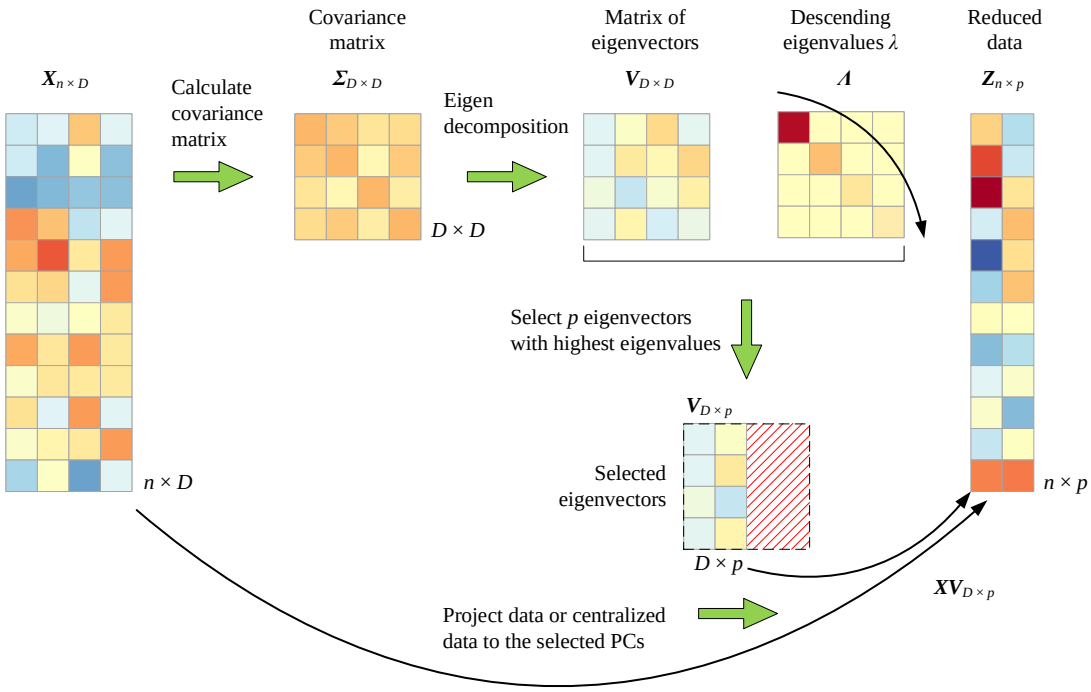

Figure 1. PCA workflow: From data centering to covariance decomposition and dimensionality reduction.

Next, we compute the covariance matrix $\boldsymbol{\Sigma}$ of the data, which is a $D \times D$ matrix capturing how each pair of features varies together. This matrix forms the basis for discovering the principal components. By performing an eigenvalue decomposition of $\boldsymbol{\Sigma}$, we obtain a set of eigenvectors and corresponding eigenvalues. The eigenvectors indicate the directions in feature space along which the data varies, while the eigenvalues quantify how much variance each direction captures.

To reduce dimensionality, we sort the eigenvalues from largest to smallest and select the top p eigenvectors corresponding to the largest eigenvalues. These top eigenvectors define a new p-dimensional subspace that captures the most important variation in the data.

Finally, we project the original centered data onto this subspace, obtaining a lower-dimensional representation of the dataset known as the principal component scores. These scores provide a compressed yet informative view of the data, preserving as much variability as possible in fewer dimensions.

This process, from computing the covariance to projecting onto the top eigenvectors, is illustrated in Figure 1, giving a clear overview of PCA's core steps: standardization or centering, covariance computation, eigenvalue decomposition, selection of principal components, and projection into a reduced space.

26.1.2 Limitations of PCA: When Linear Assumptions Fail

Principal Component Analysis (PCA) is a powerful and widely used tool for dimensionality reduction, but it is not a universal solution. PCA works best when the underlying data approximately follows a multivariate Gaussian distribution. However, when the data has a nonlinear structure, PCA often fails to capture the true directions of maximum variance.

Figure 2 illustrates such a case: when data lies on a nonlinear manifold, PCA cannot effectively reduce its dimensionality or reveal meaningful patterns.

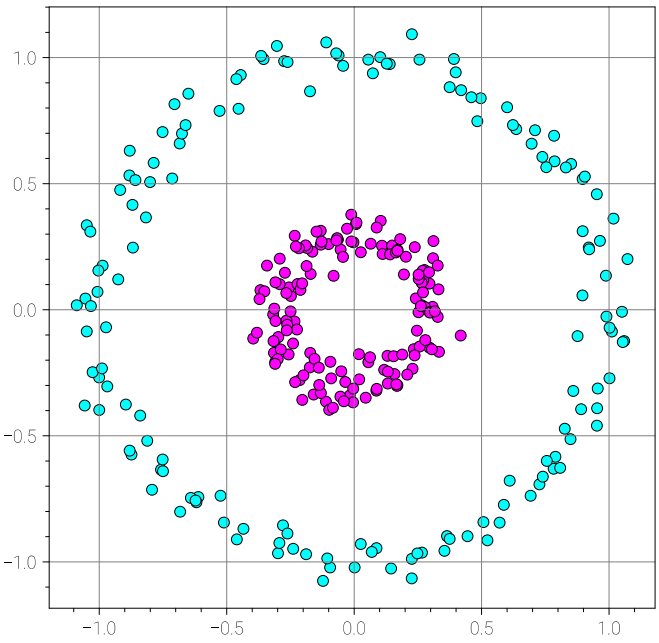

Figure 2. Example of nonlinear data structure where Principal Component Analysis (PCA) fails to perform effective dimensionality reduction. Generate by Ch26_01_kernel_PCA.ipynb.

To address this limitation, we introduce Kernel Principal Component Analysis (Kernel PCA, or KPCA)—an extension of PCA that allows for nonlinear feature extraction. Instead of directly performing PCA in the original input space, KPCA uses a kernel function to implicitly map the data into a higher-dimensional feature space, where the data may become linearly separable. In this new space, we can then apply the standard PCA procedure to extract principal components.

26.1.3 Two Gram Matrice and Their Secrets

As shown in Figure 3, suppose we have a data matrix $\boldsymbol{X}$ of shape $n \times D$, where n is the number of samples and D is the number of features. Let us assume the data matrix $\boldsymbol{X}$ has already been standardized, meaning each feature has zero mean and unit variance. PCA begins by constructing the Gram matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ of shape $D \times D$, which represents the covariance relationships among the features.

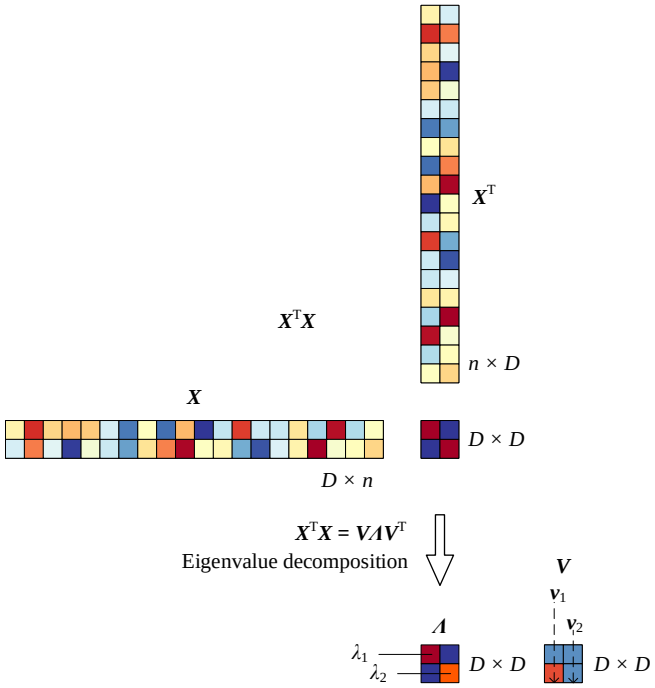

Figure 3. Eigenvalue decomposition of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$

By performing eigenvalue decomposition on this Gram matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, we obtain

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{V}$ contains the eigenvectors, and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues that indicate the amount of variance explained by each principal component.

Once the eigenvectors $\boldsymbol{V}$ are obtained, we can project the data into the new coordinate system to compute the factor scores:

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V} \tag{2}$$

26.1.4 Singular Value Decomposition: A Unified Perspective

Alternatively, by using the Singular Value Decomposition (SVD) of $\boldsymbol{X}$,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} \tag{3}$$

the factor scores can be written as

$$\boldsymbol{Z} = \boldsymbol{US} \tag{4}$$

Expand the equation above, we get

$$\underbrace{\begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_D \end{bmatrix}}_{\boldsymbol{Z}} = \underbrace{\begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_D \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_D \end{bmatrix}}_{\boldsymbol{S}} = \begin{bmatrix} s_1\boldsymbol{u}_1 & s_2\boldsymbol{u}_2 & \cdots & s_D\boldsymbol{u}_D \end{bmatrix} \tag{5}$$

Here, each column vector $\boldsymbol{u}_j$ in $\boldsymbol{U}$ is a unit vector, and $\boldsymbol{S}$ is a diagonal matrix whose elements s_j are the singular values.

Meanwhile, the data matrix $\boldsymbol{X}$ also defines the second Gram matrix, $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$.

As shown in Figure 4, Each element of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ measures the similarity between two samples, so this matrix can also be interpreted as a linear kernel matrix.

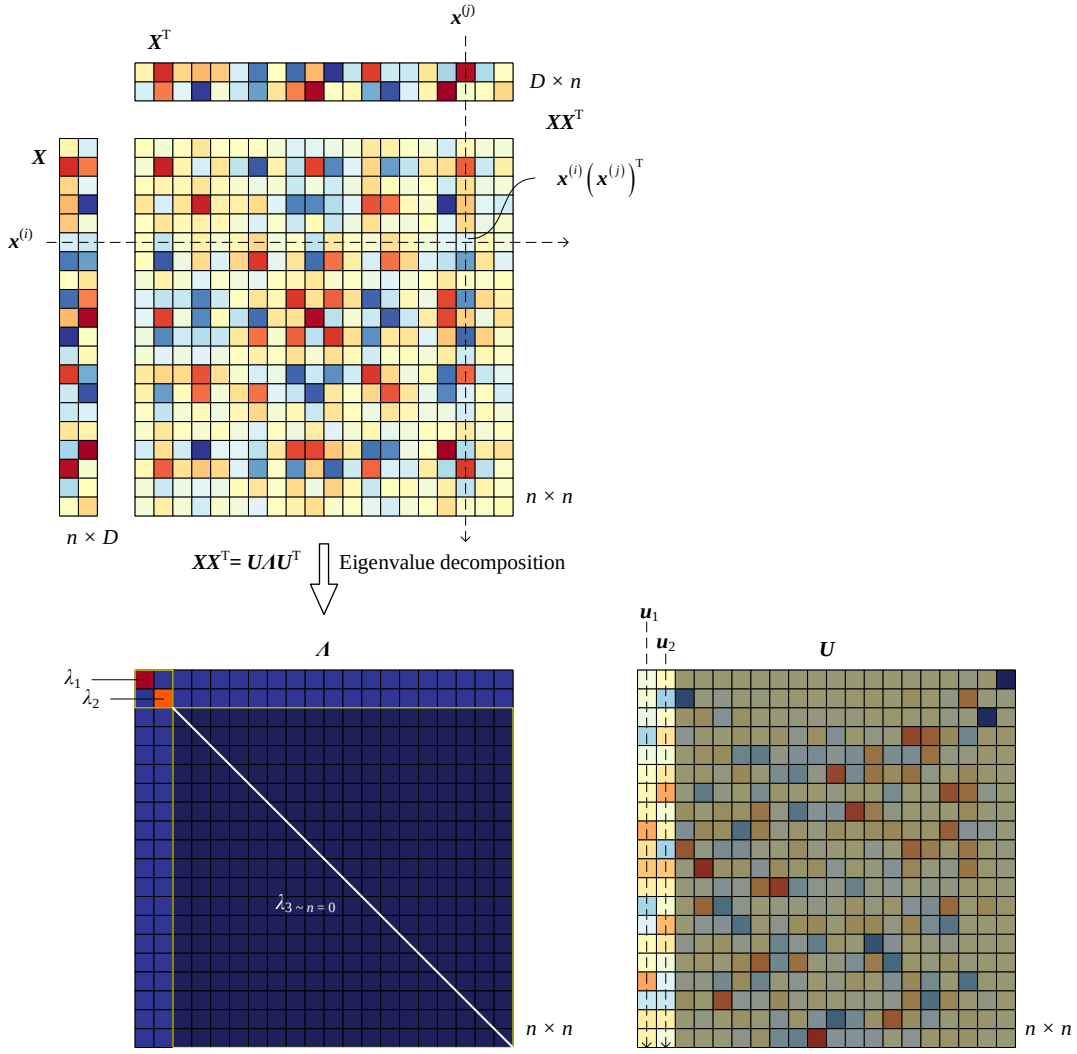

Figure 4. Eigenvalue decomposition of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$

We can also perform eigenvalue decomposition on the second Gram matrix

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}} \tag{6}$$

Although the two Gram matrices $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ and $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ have different shapes, their nonzero eigenvalues are identical. This connection between the two decompositions becomes crucial for understanding KPCA, where we generalize this idea using kernel functions.

As shown in Figure 5, the Singular Value Decomposition (SVD) provides a unified view of both Gram matrices.

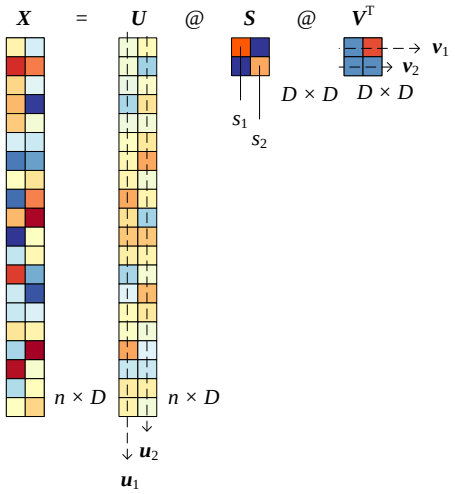

Figure 5. Singular Value Decomposition (SVD) illustrates the relationship between the eigenvalue decompositions of the two Gram matrices.

Next, we will use the Gaussian kernel as an example to illustrate how Kernel Principal Component Analysis (KPCA) is implemented using kernel techniques.

26.2 Enter the Kernel World: Nonlinear Feature Extraction

26.2.1 From Distance to Similarity: Pairwise Euclidean Matrix

To construct a Gaussian kernel matrix, we must first compute the pairwise Euclidean distances between all data points.

As shown in Figure 6, for a dataset consisting of n samples, we calculate the Euclidean distance (L2 norm) between any two points:

$$\left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)} \right\|_2 \tag{7}$$

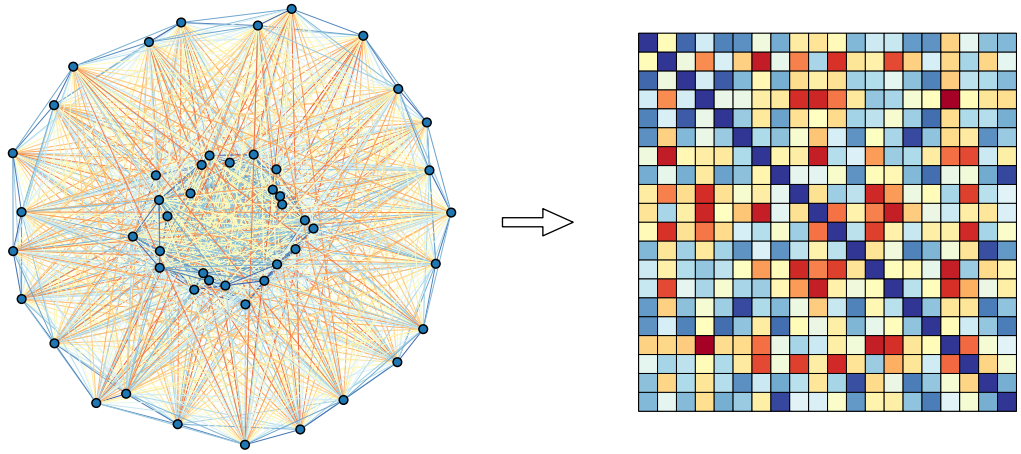

Figure 6. Pairwise Euclidean distances. Generate by Ch26_01_kernel_PCA.ipynb.

Figure 7 illustrates how the pairwise Euclidean distance matrix is computed and then squared as a preparatory step for building the Gaussian kernel matrix used in Kernel PCA.

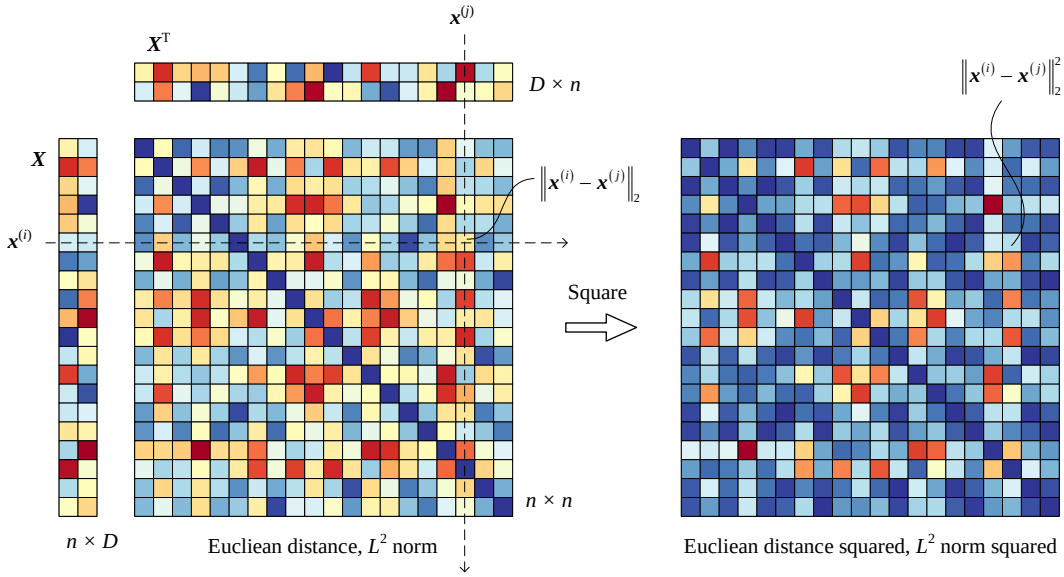

Figure 7. Squared Euclidean distances. Generate by Ch26_01_kernel_PCA.ipynb.

26.2.2 Gaussian Kernel: Measuring Closeness in High Dimensions

On the left of Figure 7, the matrix $\boldsymbol{X}$ represents the dataset, where each row $\boldsymbol{x}^{(i)}$ is a sample with 2 features. By computing the Euclidean distance between every pair of samples $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, we obtain a symmetric distance matrix. Each entry in this matrix corresponds to the L2 norm, which measures how far apart two samples are in the feature space.

The right-hand of Figure 7, shows the squared Euclidean distances. In the Gaussian kernel function,

$$\kappa\left(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}\right)=\exp\left(-\gamma\left\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\right\|_2^2\right) \tag{8}$$

the squared distance appears in the exponent. γ is a parameter that controls the "width" of the Gaussian function. A larger γ makes the kernel more sensitive to local variations, while a smaller γ produces smoother relationships.

As shown in Figure 8, this form transforms the notion of "distance" into a measure of similarity: points that are close to each other (small distance) produce kernel values near 1, while points far apart yield values near 0.

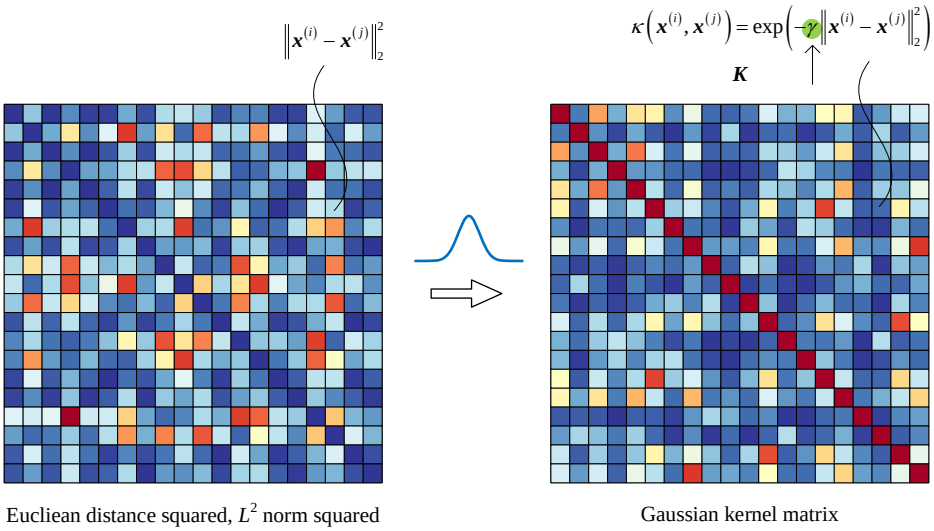

Figure 8. Gaussian kernel matrix. Generate by Ch26_01_kernel_PCA.ipynb.

As we have seen before, kernel technique are powerful tools in machine learning for handling nonlinear data. The key idea is to implicitly map the input data into a high-dimensional feature space using a kernel function. Instead of explicitly computing this mapping (which could be very expensive), the kernel function allows us to calculate the inner product in that high-dimensional space directly from the original features. This is known as the kernel trick, and it makes it possible to model complex nonlinear structures efficiently.

26.2.3 Centering the Kernel Matrix: Zero Mean in Feature Space

Before performing eigenvalue decomposition, the kernel matrix $\boldsymbol{K}$ must be centered to remove the mean effect and ensure that the transformed data has zero mean in the feature space.

Let $\boldsymbol{M}$ be the centering matrix defined as:

$$\boldsymbol{M}=\boldsymbol{I}-\frac{1}{n}\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}} \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{1}$ is a column vector of ones.

The centered kernel matrix $\boldsymbol{K}_c$ can then be obtained as:

$$\boldsymbol{K}_c=\boldsymbol{M}\boldsymbol{K}\boldsymbol{M}^{\mathrm{T}} \tag{10}$$

This operation ensures that each row and column of $\boldsymbol{K}_c$ has a mean value of zero.

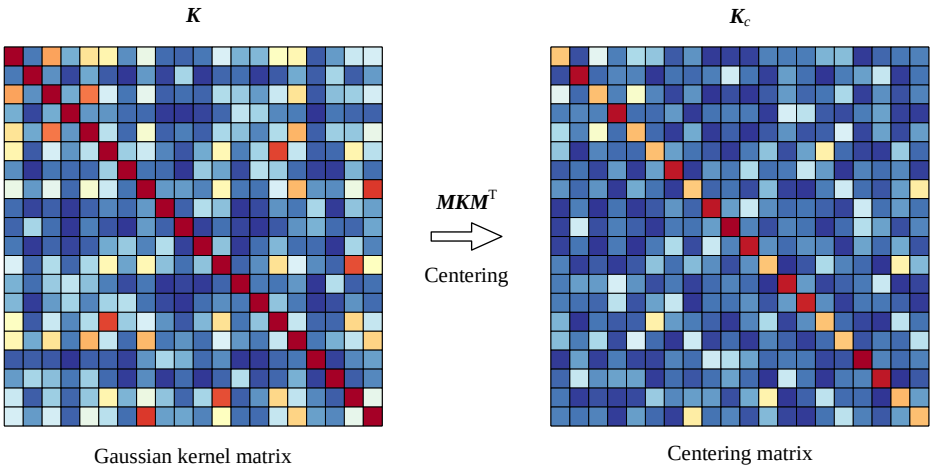

Figure 9. Centering process for the Gaussian kernel matrix. Generate by Ch26_01_kernel_PCA.ipynb.

26.2.4 Eigen-Decomposition in Kernel Space: Nonlinear Principal Components

Then, as shown in Figure 10, we perform eigenvalue decomposition on the centered kernel matrix $\boldsymbol{K}_c$.

By sorting the eigenvalues from largest to smallest, we select the top eigenvectors corresponding to the most significant components. These eigenvectors represent the principal components in the kernel-induced feature space.

Projecting the data onto these components gives us the nonlinear factor scores, i.e., the new coordinates that best capture the nonlinear structure of the data.

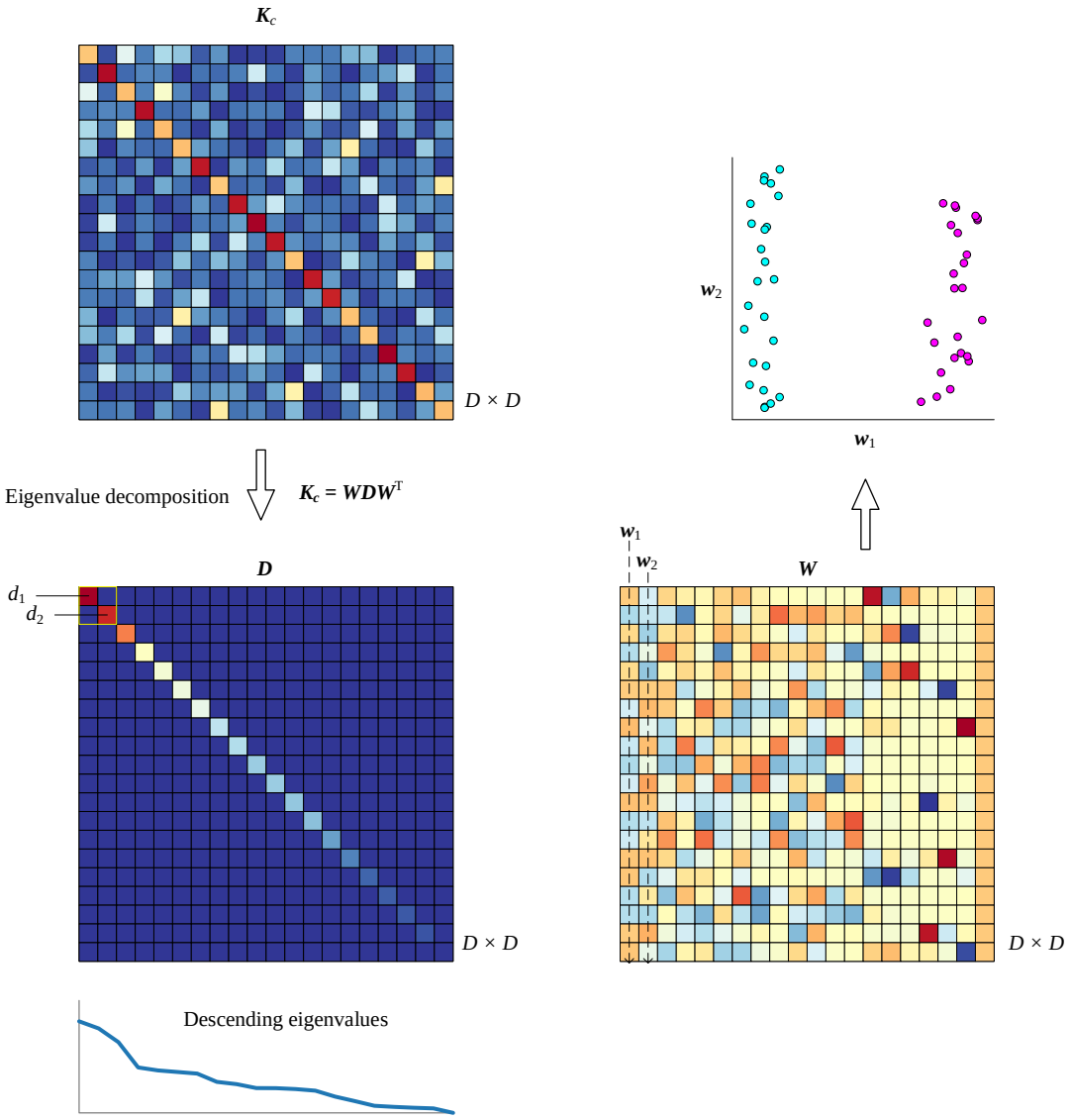

Figure 10. Eigenvalue decomposition of the Gaussian kernel matrix. Generate by Ch26_01_kernel_PCA.ipynb.

26.2.5 Nonlinear Mapping in Action: Dimensionality Lifting

Finally, as shown in Figure 11, the Gaussian kernel implicitly performs a nonlinear mapping that "unfolds" complex structures in the data. Although the original data may have only two features, the kernel mapping effectively places it in a higher-dimensional space, where the underlying structure becomes more separable or more linearly organized.

This process—often described as dimensionality lifting—is the essence of KPCA. It allows linear methods like PCA to work in spaces where nonlinear relationships can be represented linearly after the transformation.

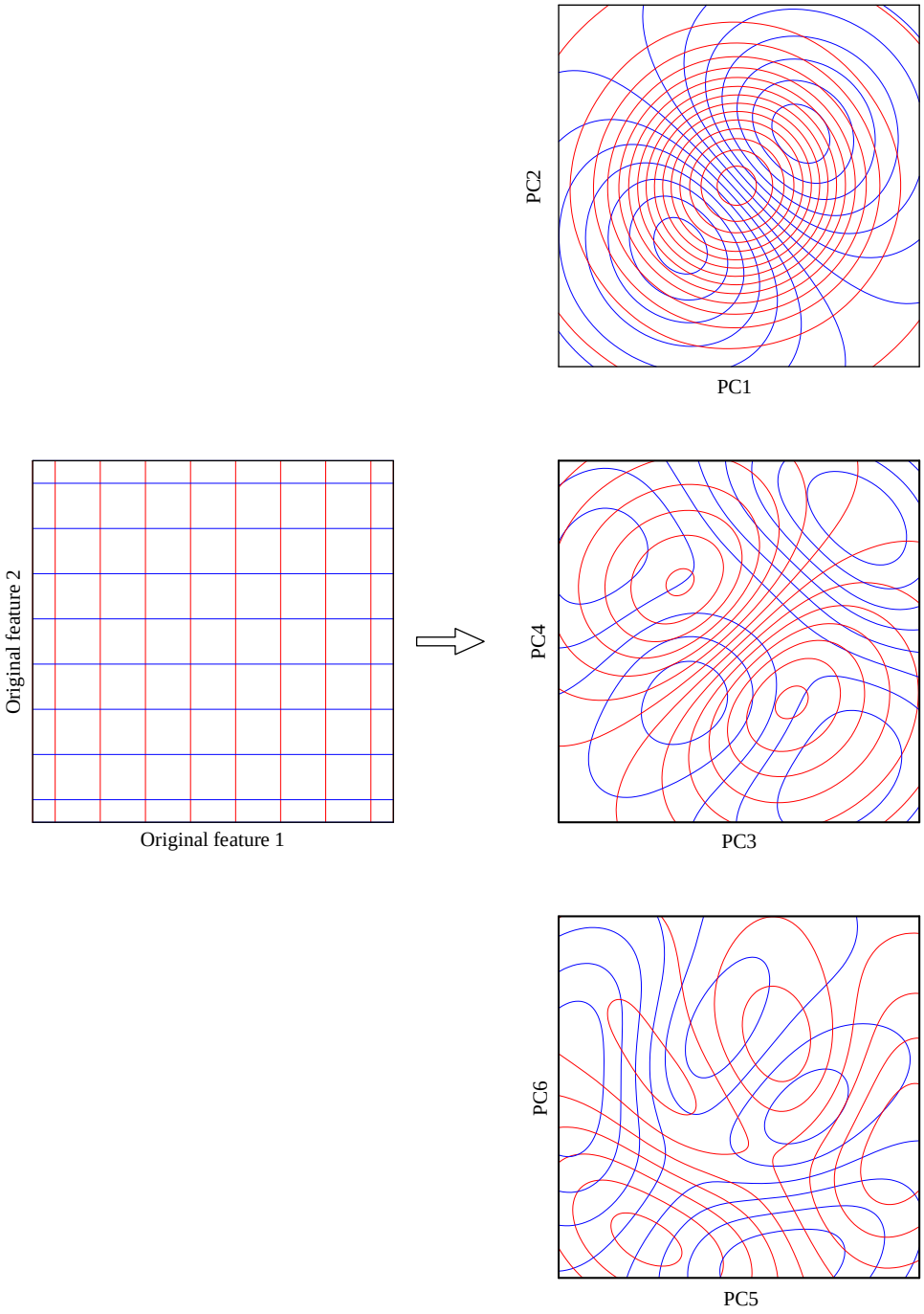

Figure 11. Nonlinear mapping effect illustrated by mesh deformation.

26.3 Conclusion

This chapter introduces Kernel Principal Component Analysis (KPCA) as a nonlinear extension of PCA. While PCA is effective for linearly structured data that roughly follows a multivariate Gaussian distribution, it struggles with nonlinear patterns. KPCA overcomes this by using kernel functions—such as the Gaussian kernel—to implicitly map data into a higher-dimensional feature space, where nonlinear relationships become linearly separable.

The chapter first reviews PCA's mathematical foundation, including Gram matrices, eigenvalue decomposition, and singular value decomposition (SVD). It then explains how KPCA builds on these ideas using

the Gaussian kernel matrix, which measures pairwise similarity based on the squared Euclidean distance between samples. This transformation converts distances into similarity scores, emphasizing close points while diminishing the influence of distant ones.

Next, the kernel matrix is centered to ensure zero-mean features in the new space, followed by eigenvalue decomposition to extract nonlinear principal components. The final stage demonstrates how kernel mapping effectively performs a dimensionality lift, revealing the hidden structure of complex data manifolds. Through this process, KPCA provides a powerful framework for uncovering nonlinear patterns that traditional PCA cannot capture.