Task 1. — Mathematical proof of how to apply PCA on $Z = \phi(X)$ without explicitly computing Z.

A mapping $\phi : \mathbb{R}^N \to \mathbb{R}^D \quad D >> N$

we say F is the feature space with D-dimension. (D is arbitrarily large, possibly infinite)

Firstly, we need to make the data zero-centered.

For now, we assume the data we're dealing with is zero-centered. So that

$$\sum_{k=1}^{m} \phi(x_k) = 0 \quad \text{is satisfied.}$$

The covariance matrix in F is that,

$$C_F = \frac{1}{m} \sum_{j=1}^{m} \phi(x_j)\, \phi(x_j)^T = \frac{1}{m}\, \phi(X)\, (\phi(X))^T$$

where $C_F \in \mathbb{R}^{D \times D}$

Then we need to find all eigenvalues $\lambda_i \geq 0$ and all eigenvectors $V_i \in F \setminus \{0\}$ satisfying

$$C_F V_i = \lambda_i V_i .$$

$\forall\, V_i$ with $\lambda_i \neq 0$ lie in the space of $\phi(x_1), \ldots, \phi(x_m)$.

Hence, we can get

1) $\lambda\, (\phi(x_k) \cdot V) = (\phi(x_k) \cdot C_F V) \qquad \forall\ k = 1, \ldots, m$

2) $V = \sum_{i=1}^{m} \alpha_i\, \phi(x_i) \qquad$ for $\alpha_i \in \mathbb{R} \qquad i \in \{1, \ldots, m\}$

Thus, by combining 1) & 2) yields:

$$\lambda \sum_{i=1}^{m} \alpha_i (\phi(x_k) \cdot \phi(x_i)) = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \left(\phi(x_k) \cdot \sum_{j=1}^{m} \phi(x_j)\right) \cdot (\phi(x_j) \cdot \phi(x_i)) \qquad (*)$$

Define

$$K_{ij} := \phi(x_i) \cdot \phi(x_j) \qquad K \in \mathbb{R}^{m \times m}$$

Then $(*)$ reads:

$$\lambda\, K \underline{\alpha} = \frac{1}{m} K^2 \underline{\alpha} \qquad \text{where } \alpha \text{ denotes the column vector with entries: } \alpha_1, \ldots, \alpha_m$$

$$\Rightarrow \quad K\underline{\alpha} = m\lambda \underline{\alpha} \qquad (**)$$

We solve the non-zero eigenvalues for $(**)$.

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ denote the eigenvalues of K with corresponding eigenvectors $\alpha^1, \alpha^2, \ldots, \alpha^m$.

Let λ_p be the first non-zero eigenvalue. we normalized that is

$$(V^k \cdot V^k) = 1 \qquad \forall\ k \in \{p, \ldots m\}$$

By $V = \sum_{i=1}^{m} \alpha_i \phi(x_i)$ and $(**)$, this translates into a normalization condition for $\alpha^p, \dots, \alpha^m$.

$$1 = \sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k (\phi(x_i) \cdot \phi(x_j)) = \sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k K_{ij}$$

$$= \alpha^k K \alpha^k \qquad (***)$$

$$= \lambda_k (\alpha^k \cdot \alpha^k)$$

Then compute projections onto the eigenvectors V^k in F $(k = p, \dots, m)$.

Let x be a test point, with an image $\phi(x)$ in F; then

$$(V^k \cdot \phi(x)) = \sum_{i=1}^{m} \alpha_i^k (\phi(x_i) \cdot \phi(x))$$ is the PC corresponding to ϕ.

Then, for any ϕ and any set of data $\{x_1, \dots x_m\}$, the points can be centered as $\tilde{\phi}(x_i) := \phi(x_i) - \frac{1}{M} \sum_{i=1}^{M} \phi(x_i)$. (1) Thus, the problem is similar to the problem above.

We go on to define covariance matrix $\tilde{K}_{ij} = (\tilde{\phi}(x_i) \cdot \tilde{\phi}(x_j))$ in F. Also, the familiar eigenvalue problem,

$$\tilde{\lambda} \tilde{\alpha} = \tilde{K} \tilde{\alpha}$$

with $\tilde{\alpha}$ being the expansion coefficients of an eigenvector in F in terms of the points in equation (1), $\tilde{V} = \sum_{i=1}^{M} \tilde{\alpha}_i \tilde{\phi}(x_i)$.

Because we don't have the centered data, we cannot compute $\tilde{K}$ directly; however, we can express it in terms of its non centered counterpart K. In the following, we shall use $K_{ij} = (\phi(x_i) \cdot \phi(x_j))$ and the notations $1_{ij} = 1$ for all i, j, $(1_M)_{ij} := \frac{1}{M}$, to compute $\tilde{K}_{ij} = (\tilde{\phi}(x_i) \cdot \tilde{\phi}(x_j))$

$$\tilde{K}_{ij} = \left((\phi(x_i) - \frac{1}{M} \sum_{m=1}^{M} \phi(x_m)) \cdot (\phi(x_j) - \frac{1}{M} \sum_{n=1}^{M} \phi(x_n)) \right)$$

$$= \phi(x_i) \cdot \phi(x_j) - \frac{1}{M} \sum_{m=1}^{M} \phi(x_m) \cdot \phi(x_j) - \frac{1}{M} \sum_{n=1}^{M} \phi(x_n) \phi(x_i)$$

$$+ \frac{1}{M^2} \sum_{m=1}^{M} \sum_{n=1}^{M} \phi(x_m) \phi(x_n)$$

$$= K_{ij} - \frac{1}{M}\sum_{m=1}^{M} 1_{im} k_{mj} - \frac{1}{M}\sum_{n=1}^{M} k_{in} 1_{nj} + \frac{1}{M^2}\sum_{m,n=1}^{M} 1_{im} k_{mn} 1_{nj}$$

$$= k_{ij} - \frac{1}{M}\sum_{m=1}^{M} k_{mj} - \frac{1}{M}\sum_{n=1}^{M} k_{in} + \frac{1}{M^2}\sum_{m,n=1}^{M} k_{mn}$$

$$= \tilde{k}_{ij}$$

Thus, we can compute $\tilde{k}$ from k and then solve the eigenvalue problem.

As the equation (***) the solution $\tilde{\alpha}^k$ are normalized by normalizing the corresponding vectors $\tilde{V}^k$ in F, which translates into $\tilde{\lambda}_k (\tilde{\alpha}^k \cdot \tilde{\alpha}^k) = 1$.

Task 1 — Mathematical proof of how to project the data points in $Z = \phi(X)$ by using any number of top PCs without explicitly using the PCs.

Let $Z = \phi(X)$, suppose $\tilde{Z} = \tilde{\phi}(X)$, centered the data.

We can find matrix P, Q such that $\tilde{Z} = PQ$ for P being the projected matrix and Q being the matrix for PCs in row column and corresponding to the decreasing order of variance.

Hence, each row of W is orthogonal to the other rows of W.

We claim that $\tilde{Z} W^{-1} = Y W W^{-1} = Y$. Since $W^{-1} = W$, then

$$Y = \tilde{Z} W^T$$

Again we claim that $W_i^T = \sigma_i^{-1} \tilde{Z}^T U_i$ for W_i being the ith PC; σ_i^{-1} being the ith singular value and U_i is the ith column of U, where $\tilde{Z} = U \Sigma V^T$.

Since $\tilde{Z} = U \Sigma V^T = U \Sigma W$ yields,

$$W = U^{-1} \Sigma^{-1} \tilde{Z}$$

Since U is orthogonal matrix, then

$$W = \Sigma^{-1} U^T \tilde{Z}$$

Hence, $W_i = \sigma_i^{-1} U^T \tilde{Z} \Rightarrow (W_i)^T = (\sigma_i^{-1} U_i^T \tilde{Z})^T$

$$= \tilde{Z}^T (U_i^T)^T (\sigma_i^{-1})^T$$

$$= \sigma_i^{-1} Z^T U_i$$

Then $Y_i = \tilde{Z} \sigma_i^{-1} \tilde{Z}^T U_i$

$$= \sigma_i^{-1} \tilde{Z} \tilde{Z}^T U_i = \sigma_i^{-1} K U_i$$

This tells us, each column of Y, which are the features, can be obtained without explicitly using Z or PCs.

Therefore, by finding Y, we can do PCA on $Z = \phi(X)$ without explicitly computing Z and by finding each feature vector, we can project onto any number of top PCs.

Task 2 - Kernel selection

a) Homogeneous Polynomial Kernel

$K(x_i, x_j) = (\langle x_i, x_j \rangle)^d$ $d > 0$ is the degree.

This can project the data to a specific higher dimention depending on d.

The decision boundary is the polynomial surface.

Gaussian kernel

$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$ where $\gamma = \frac{1}{2\sigma^2}$

This can project the data to an infinite-dimentional space.

The decision boundary is based on the distance between the samples.

In this case, Gaussian kernel is more appropriate to the circles datasets. Samples can be linearly seperated into 2 classes. One is in a circle region and the other one (class 2) is in a ring region around the circular region (class 1).

The data in class 1 is more gathered together than the points in class 2. Hence, the projection using Gaussian kernel should be easier to be linearly seperated.

b) How can one relate the kernel with (σ) to the data available?

The parameter σ (kernel width) helps determine the impact of distances between data points on their projected similarities.

When σ increasing, it diminished effect individual sample vectors have on the projection. This lead to a smoother decision boundary, as the projection become less sensitive to variations between data points.

However, a model trained with such a projection may be prone to underfitting. This happens because the model may become too generalised, potentially failing to capture important nuances and patterns in the data.

When σ decreasing, the projection becomes more sensitive to the specific characteristics of the data points.

As a result, the model is capable of capturing finer details in the data. This can be particularly advantageous when dealing with complex patterns.

However, there is an increased risk of overfitting. In this case, the model might become too tailored to the training data, reducing its ability to generalize well to new, unseen data.

In summary, the choice of σ in kernel methods is a balancing act between ensuring a smooth, generalizable model and creating a model that is detailed enough to capture the essential features of the data without overfitting.

C) What is the influence of the degree (d) of a Homogeneous Polynomial kernel?

Degree d of homogeneous polynomial kernel has impact on the complexity of the decision boundary in a model.

For a large d, it results in a more complex decision boundary. The model becomes highly sensetive to subtle details in the data.
However, this hightened sensitivity can lead to overfitting, where the model may perform well on training data but poorly on testing data due to its excessive specificity.

For a small degree d, it has a lower value of d, simplifies the decision boundary.
This simplification means the model will be smoother and less prone to capturing minor fluctuations in the data.
However, this can prevent overfitting, there is a risk of underfitting, where the model may be too generalized, missing important patterns in the data.

When $d = 1$, $K(x_i, x_j) = (\langle x_i, x_j \rangle)$ which projected the points to themselves.

Task 3

The minimum number of top PCs is 3. (See the computation in the code)

Task 4

We have threshold = -1.27100821, then for feature of the projected data,
if it is larger than the thereshold, classifies it to class 2, otherwise classifies to class 1.
The metric used here is accuracy. The straight line at the threshold such that all points
have been seperated into two region, ensure projected data is linearly seperable.

Task 5

The plot can be done in 3-D, since we need at least 3-PCs and cannot
be plot out in 1-D or 2-D, by using single decision stump.