Note to the Professor and TAs: Python codes written for this assignment include the following libraries:Numpy, Matplotlib, Seaborn (similar to Matplotlib but helps in better visualization), and PIL (Read images and convert to numpy arrays for Qn 4).

1. Question 1(a)

Solution: For a $d \times d$ matrix A, we know that the SVD is,

$$\begin{aligned} A &= U\Sigma V^T \\ AV &= U\Sigma && \text{(Since } V \text{ is an orthogonal matrix, } V^T = V^{-1}.\text{)} \\ AV_i &= \sigma_i U_i \end{aligned}$$

This means that for the first r non-zero singular values in Σ, the corresponding column vectors in U give the orthonormal bases to span the $\mathcal{C}(A)$. Similarly, consider SVD of AA^T,

$$\begin{aligned} AA^T &= U\Sigma U^T \\ AA^T U &= U\Sigma && \text{(Since } U \text{ is an orthogonal matrix, } U^T = U^{-1}.\text{)} \\ AA^T U_i &= \sigma_i U_i \end{aligned}$$

This means that for the first r non-zero singular values in Σ, the corresponding column vectors in U give the orthonormal bases to span the $\mathcal{C}(AA^T)$.

Hence, $\mathcal{C}(A) = \mathcal{C}(AA^T)$.

2. Question 1(b)

Solution: For a $d \times d$ matrix A, row space is not equal to column space. Consider SVD of A,

$$\begin{aligned} A &= U\Sigma V^T \\ AV &= U\Sigma && \text{(Since } V \text{ is an orthogonal matrix, } V^T = V^{-1}.\text{)} \end{aligned}$$

For the first r non-zero singular values in Σ, the corresponding column vectors in U give the orthonormal bases to span the $\mathcal{C}(A)$. Similarly, consider SVD of A^T,

$$A = U\Sigma V^T$$

Taking transpose on both sides,

$$\begin{aligned} A^T &= V\Sigma^T U^T \\ A^T &= V\Sigma^T U^{-1} && \text{(Since } U \text{ is orthogonal, } U^T = U^{-1}\text{)} \\ A^T U &= V\Sigma \end{aligned}$$

For the first r non-zero singular values in Σ, the corresponding column vectors in V give the orthonormal bases to span the $\mathcal{C}(A^T)$.

Since it is not necessary for $\mathcal{C}(A)$ and $\mathcal{C}(A^T)$ to be equal, in most general cases, $\mathcal{C}(A) \neq \mathcal{C}(A^T)$. The only scenario under which $\mathcal{C}(A) = \mathcal{C}(A^T)$ is when A is a symmetric or skew-symmetric matrix.

3. Question 2

Solution: Any kernel $K(\mathbf{x_i}, \mathbf{x_j})$ is a valid kernel if for a finite set $\{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots, \mathbf{x_n}\}$, matrix K is:

1. Positive semi-definite (or $\mathbf{v}^T K \mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^n$)
2. K is symmetric

We have already been given that K_1, K_2 be two arbitrary kernel functions mapping vectors from $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfy both the above conditions.

Note: All kernel function $K(x, y)$ will be denoted as K.

(a) Checking whether K_3 is a valid kernel.

$$\begin{aligned}\mathbf{v}^T K_3 \mathbf{v} &= \mathbf{v}^T (K_1 + K_2 + 7.5)\mathbf{v} \\ &= \mathbf{v}^T K_1 \mathbf{v} + \mathbf{v}^T K_2 \mathbf{v} + \mathbf{v}^T 7.5\mathbf{v} \\ &= \mathbf{v}^T K_1 \mathbf{v} + \mathbf{v}^T K_2 \mathbf{v} + 7.5\mathbf{v}^T \mathbf{v}\end{aligned}$$

Since $\mathbf{v}^T K_1 \mathbf{v} \geq 0$, $\mathbf{v}^T K_2 \mathbf{v} \geq 0$, and $\mathbf{v}^T \mathbf{v} \geq 0$,

$$\mathbf{v}^T K_3 \mathbf{v} \geq 0$$

Hence, K_3 is a valid kernel.

(b) Checking whether K_4 is a valid kernel.

$$\begin{aligned}\mathbf{v}^T K_4 \mathbf{v} &= \mathbf{v}^T (5K_1 - 3K_2)\mathbf{v} \\ &= 5\mathbf{v}^T K_1 \mathbf{v} - 3\mathbf{v}^T K_2 \mathbf{v}\end{aligned}$$

While we know that $\mathbf{v}^T K_1 \mathbf{v} \geq 0$, $\mathbf{v}^T K_2 \mathbf{v} \geq 0$, it is possible that $3\mathbf{v}^T K_2 \mathbf{v} > 5\mathbf{v}^T K_1 \mathbf{v}$. This makes $\mathbf{v}^T K_4 \mathbf{v} < 0$. Therefore, K_4 is not a valid kernel.

(c) Checking whether K_5 is a valid kernel.

$$\begin{aligned}\mathbf{v}^T K_5 \mathbf{v} &= \mathbf{v}^T K_1 K_2 \mathbf{v} \\ &= K_1 \mathbf{v}^T K_2 \mathbf{v} \\ &= K_1 K_2^{'}\end{aligned}$$

Since K_1 and $K_2^{'}$ are positive semi-definite matrices with only non-negative eigen values, we can say that the trace of matrix product of K_1 and K_2^{-1} will be non-negative, i.e.,

$$\begin{aligned}\mathbf{v}^T K_5 \mathbf{v} &= trace(K_1 K_2^{'}) \\ \mathbf{v}^T K_5 \mathbf{v} &\geq 0\end{aligned}$$

Hence, K_5 is a valid kernel.

(d) Checking whether K_6 is a valid kernel. We know that kernel matrix $K(x, y) = \phi(x)^T \phi(y)$. Let's assume that $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$.

$$\begin{aligned}K_6 &= (x^T y + 1)^3 \\ &= (x_1 y_1 + x_2 y_2 + 1)^3 \\ &= x_1^3 y_1^3 + x_2^3 y_2^3 + 3x_1^2 y_1^2 (x_2 y_2 + 1) + 3x_2^2 y_2^2 (x_1 y_1 + 1) + 3(x_1 y_1 + x_2 y_2) + 6x_1 x_2 y_1 y_2 + 1 \\ &= \phi(x)^T \phi(y)\end{aligned}$$

where $\phi(x) = \begin{bmatrix} x_1^3 & x_2^3 & \sqrt{3}x_1^2 x_2 & \sqrt{3}x_1 x_2^2 & \sqrt{3}x_1^2 & \sqrt{3}x_2^2 & \sqrt{3}x_1 & \sqrt{3}x_2 & \sqrt{6}x_1 x_2 & 1 \end{bmatrix}^T$
and $\phi(y) = \begin{bmatrix} y_1^3 & y_2^3 & \sqrt{3}y_1^2 y_2 & \sqrt{3}y_1 y_2^2 & \sqrt{3}y_1^2 & \sqrt{3}y_2^2 & \sqrt{3}y_1 & \sqrt{3}y_2 & \sqrt{6}y_1 y_2 & 1 \end{bmatrix}^T$.

Hence, K_6 is a valid kernel.

4. Question 3(i) and Question 3(ii)

Solution: For the 1^{st} principal component, variance ≈ 17.132. For the 2^{nd} principal component, variance ≈ 14.489.

Centering the data doesn't have any effect on the variance explained by each of the principal components. This means that the data is already a centered data with mean equal to 0 or an exponentially low value.

5. Question 3(iii)

Solution: See plots in zip folder.

6. Question 3(iv)

Solution: The Gaussian kernel performs better than the polynomial kernel. There are two main reasons for the Gaussian kernel to perform better.

1. The Gaussian kernel depends on two factors: distance between two points in feature space, and σ. One can think of σ as a closeness factor for the distance metric. For a data point x, smaller values of σ only consider points closest to x. Hence, clustering happens locally. However, as we increase value of σ towards 1 (in our question), we see that clusters are more prominently separated. Hence, higher values of σ considers points far-away from x, thereby, leading to a more global clustering mechanism. One can also say that smaller value of σ has less bias and more variance whereas higher value of σ has more bias and less variance. Therefore, with a well selected σ, Gaussian kernel will have a proper range which will amplify the connection between data points closer to each other in the feature space.

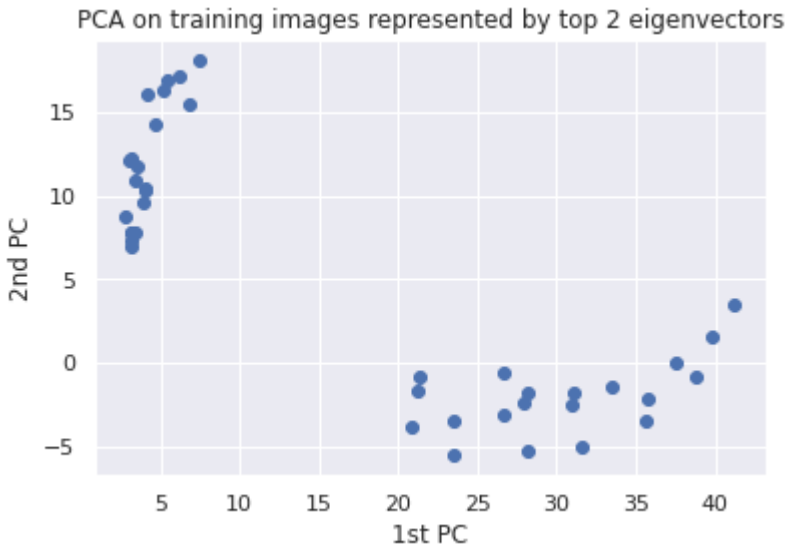
2. The polynomial kernel only extends to dimensions specified by the value of d. On the other hand, Gaussian kernel extends to infinite dimensional feature space. This is because the Taylor series expansion of Gaussian kernel shows us that the Gaussian kernel maps every point in $\mathbb{R}^2$ to an infinite dimension vector, thereby leading to better clustering.

7. Question 4

Solution: We have two distinct object categories: Category 1 is a Vaseline cream box and Category 2 is a cup. Each category contains 25 images. All images are of the dimension 80×80. Since pixel values are our features, we take 80×80 pixels and stretch it out as a column vector for each image. Therefore, our image dataset is of the shape $D \times N$ where $D = 6400$ and $N = 50$.

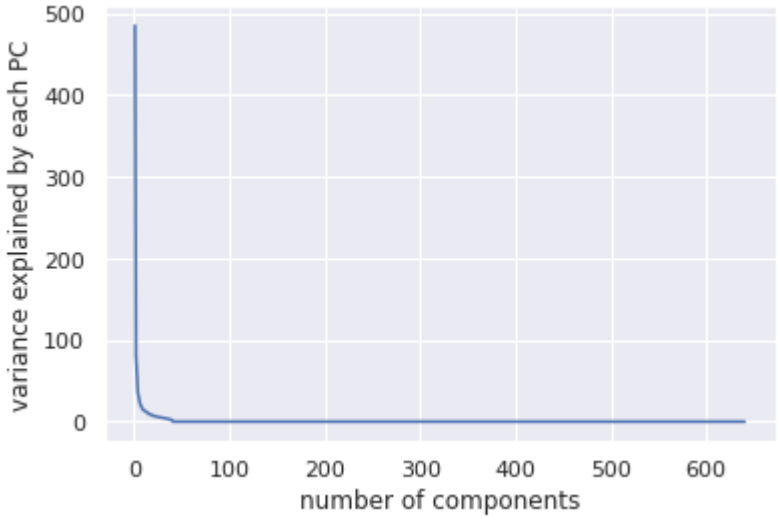
After running standard PCA on the training set, here are the following observation:

1. In higher dimensions, PCA separates the two categories into distinct clusters. We can see this by plotting the projections on top 2 eigen vectors.



2. After projecting training and test images onto the top $K\%$ eigen vectors, we compute the average distance between each point in test set and Category 1 data points. Similarly, we compute another average distance between the between each point in test set and Category 2 data points. We find that the test set images are mapped closer to their respective categories i.e., test points for which $\text{AvgDistance}_{\text{Category 1}} < \text{AvgDistance}_{\text{Category 2}}$ belong to Category 1. Similarly, test points for which $\text{AvgDistance}_{\text{Category 1}} > \text{AvgDistance}_{\text{Category 2}}$ belong to Category 2. Intuitively, this could be thought of as classification.

3. In the elbow plot of top 10% eigen vectors, we see that the variance of the dataset can be explained with fewer than top 20 eigen vectors. We can easily discard the remaining features, thereby, compressing the image size.



4. For larger sized images, standard PCA is computationally expensive. On the other hand, Kernel PCA will be computationally more efficient.

5. While standard PCA has worked well for this specific dataset, this result cannot be generalized. This is because for objects with varying backgrounds, image binarization will lead to addition of noise (for instance, if there is a sudden change in light intensity) which will affect the clustering made by PCA. However, if the images are retained in RGB or grayscale format, computing PCA may be beneficial since all low variance noise components can be discarded.