CMSE 820 HW 5

Hao Lin

October 20, 2018

1 Question 1

1.1 a

If $f_s(x)$ is convex for any $s \in S$, prove that $f(x) = \max_{s \in S} f_s(x)$ is convex.
Proof:

Consider some arbitrary x, y and $t \in [0, 1]$. By the definition of $f(x)$, there exist $a, b, c(t) \in S$ such that

$$f(x) = f_a(x),\ f(y) = f_b(y),\ f[tx + (1 - t)y] = f_{c(t)}[tx + (1 - t)y].$$

Since $f_c(t)$ are convex, we have

$$f[tx+(1-t)y] = f_{c(t)}[tx+(1-t)y] \le tf_{c(t)}(x)+(1-t)f_{c(t)}(y) \le tf_a(x)+(1-t)f_b(y) = tf(x)+(1-t)f(y).$$

Hence, $f(x)$ is convex.

1.2 b

Prove that the dual problem for any general minimization problem is a convex optimization problem.
Proof:

Let's consider the following general minimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & h_i(x) \le 0,\ i = 1, ..., m, \\ & l_j(x) = 0,\ j = 1, ..., n. \end{aligned}$$

The corresponding Lagrangian reads

$$L(x, v, u) = f(x) + v^T h(x) + u^T l(x),\ v \in \mathbb{R}^m,\ u \in \mathbb{R}^n.$$

The dual function is

$$g(u,v) = \min_x L(x,u,v).$$

Now consider some arbitrary $u_1, u_2 \in \mathbb{R}^m$, $v_1, v_2 \in \mathbb{R}^n$ and $t \in [0,1]$, we have

$$\begin{aligned} g[tu_1 + (1-t)u_2, tv_1 + (1-t)v_2] &= \min_x \{f(x) + t[u_1^T h(x) + v_1^T l(x)] + (1-t)[u_2^T h(x) + v_2^T l(x)]\} \\ &\geq t \min_x [f(x) + u_1^T h(x) + v_1^T l(x)] + (1-t) \min_x [f(x) + u_2^T h(x) + v_2^T l(x)] \\ &= tg(u_1, v_1) + (1-t) g(u_2, v_2), \end{aligned}$$

which shows that the dual function $g(u,v)$ is concave.

The constraint $u \geq 0$ simply adds a linear term to the Lagrangian, which is both concave and convex. The dual problem

$$\max_{u \geq 0, v} g(u,v) = -\min_{u \geq 0, v} -g(u,v)$$

is thus a convex optimization problem.

2 Question 2

Consider a quadratic program:

$$\begin{aligned} &\min_x \frac{1}{2} x^T Q x + c^T x \\ &\text{subject to } Ax = b,\ x \geq 0, \end{aligned}$$

where $x \in \mathbb{R}^p$ and $Q \in \mathbb{R}^{p \times p}$ is a positive definite matrix. Derive the dual problem.

The Lagrangian for the primal problem reads

$$L(x,u,v) = \frac{1}{2} x^T Q x + c^T x - u^T x + v^T (Ax - b).$$

Note that $L(x,u,v)$ is a strictly convex function of x, because the quadratic term is strictly convex and the affine terms are convex. Thus, $\min_x L(x,v,u)$ has a unique solution x^* satisfying $\frac{\partial L}{\partial x}|_{x=x^*} = 0$.

$$\left.\frac{\partial L}{\partial x}\right|_{x=x^*} = Qx^* + c - u + A^T v = 0 \Leftrightarrow x^* = -Q^{-1}(c - u + A^T v).$$

Note that since Q is positive definite, $\mathrm{Ker}\{Q\} = \{0\}$, from which its invertibility follows.

The dual problem is

$$\max_{u \geq 0, v} g(u,v) = \max_{u \geq 0, v} L(x^*, u, v) = \max_{u \geq 0, v} -\frac{1}{2}(c - u + A^T v)^T Q^{-1} (c - u + A^T v) + v^T b.$$

3 Question 3

Given an $X \in \mathbb{R}^{p\times n}$, a new data point $x^* \in \mathbb{R}^p$ and a feature map $\phi : \mathbb{R}^p \to \mathbb{R}^\infty$, we can perform PCA on the feature space to achieve the four goals in the following manner:

3.1 a. Encode training data:

We can use ϕ to map the original data matrix to a data matrix $\Phi(X)$ in the feature space, and perform dual PCA on the matrix $\Phi^T\Phi$, i.e.,

$$\Phi^T\Phi = V(\Sigma^2)V^T,$$

where $\Phi = U\Sigma V^T$ is the compact SVD of Φ. The first d principal components $Y \in \mathbb{R}^{d\times n}$ are given by

$$Y = \Sigma_d V_d^T,$$

where $\Sigma_d \in \mathbb{R}^{d\times d}$ and $V_d \in \mathbb{R}^{n\times d}$ are truncated from Σ and V, respectively.

3.2 b. Reconstruct training data

The first d principal component eigenvectors are given by

$$U_d = \Phi V_d(\Sigma_d^{-1}).$$

The training data can be reconstructed in the feature space as

$$\tilde{\Phi}(X) = U_d Y = \Phi V_d(\Sigma_d^{-1})Y.$$

3.3 c. Encode testing example

$$y^* = U_d^T\phi(x^*) = [\Phi V_d(\Sigma_d^{-1})]^T\phi(x^*).$$

3.4 d. Reconstruct test example

$$\tilde{\phi}(x^*) = U_d y^* = \Phi V_d(\Sigma_d^{-1})y^*.$$

4 Question 4

```
function G = GaussianKernel(x,y,sigma)
    G = exp(-((norm(x-y)/sigma)^2)/2);
end
```

```
function P = PolynomialKernel(x,y,a)
    P = dot(x,y)^a;
end
```

```
function K = KappaMatrix(X,type,tuning_parameter)
    p = size(X,1);
    n = size(X,2);
    H = eye(n)-ones(n,1)*ones(1,n)/n;
    if type == "Polynomial"
        X = X*H;
    end
    K = zeros(n,n);
    for i=1:n
        xx = X(:,i);
        for j=i:n
            yy = X(:,j);
            if type == "GaussianKernel"
                K(i,j)=GaussianKernel(xx,yy,tuning_parameter);
            elseif type == "Polynomial"
                K(i,j)=PolynomialKernel(xx,yy,tuning_parameter);
            end
        K(j,i)=K(i,j);
        end
    end
    K = H*K*H;
end
```

```
function [Sigma,V,Y] = KernelPCA(kappa,d)
    [U,S,V] = svd(kappa);
    V = V(:,1:d);
    Sigma = sqrtm(S(1:d,1:d));
    Y = Sigma*V.';
end
```

5 Question 5

The kernels used are as follows:

$$\text{Gaussian kernel } k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2(0.2)^2}\right),$$
$$\text{Polynomial kernel } k(x,y) = (x^T y)^2.$$

5.1 (1)

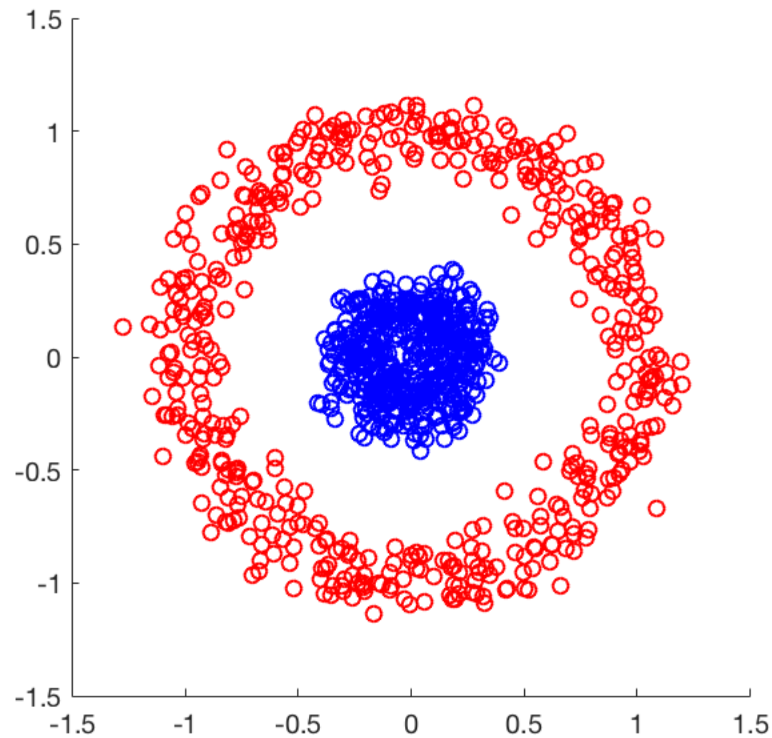

Figure 1: Original space.

5.2 (2)

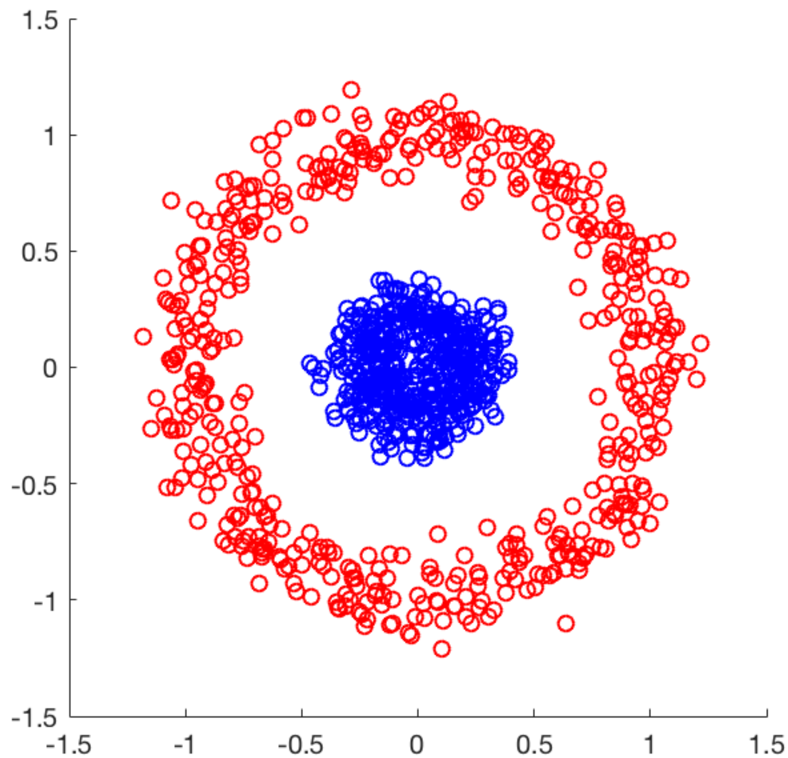

Figure 2: Plot of PCA using the 1st and 2nd principal components. The PCA plot amounts to simply a rotation of the original data plot, and the data points are not linearly separable.

5.3 (3)

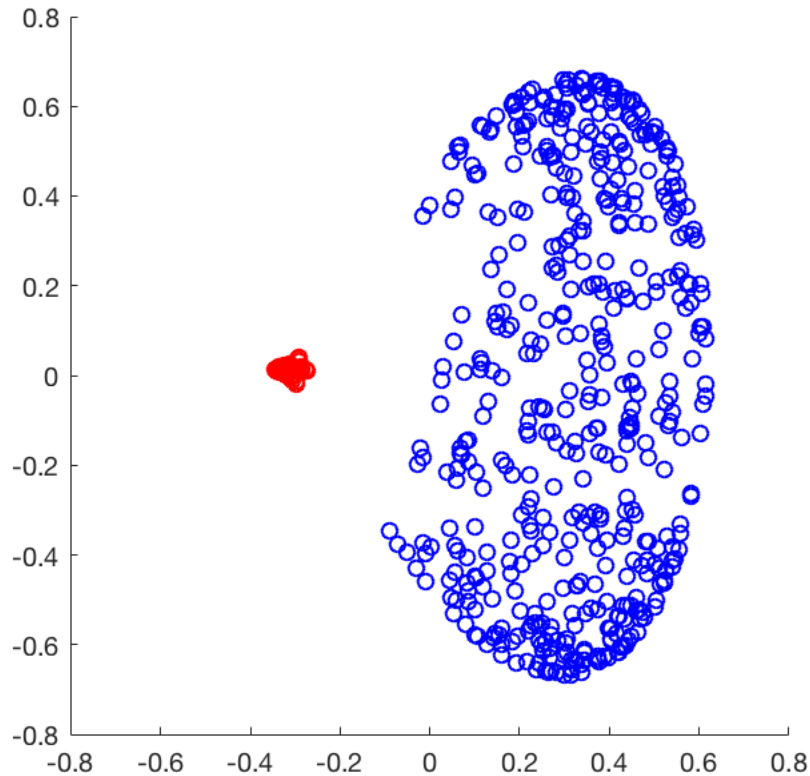

Figure 3: Plot of KPCA with Gaussian kernel using the 1st and 2nd principal components. The data points with different colors have distinctly different first principal components. It is possible to separate the two groups by drawing a straight line on the plot.

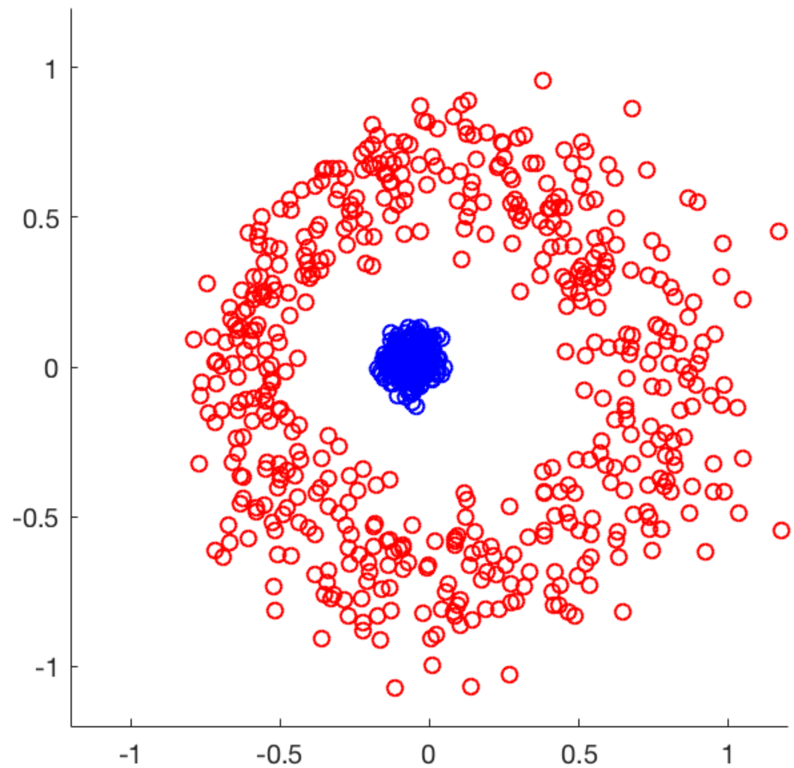

Figure 4: Plot of KPCA with polynomial kernel using the 1st and 2nd principal components. The first two principal components are not enough to separate the two groups of data points linearly.

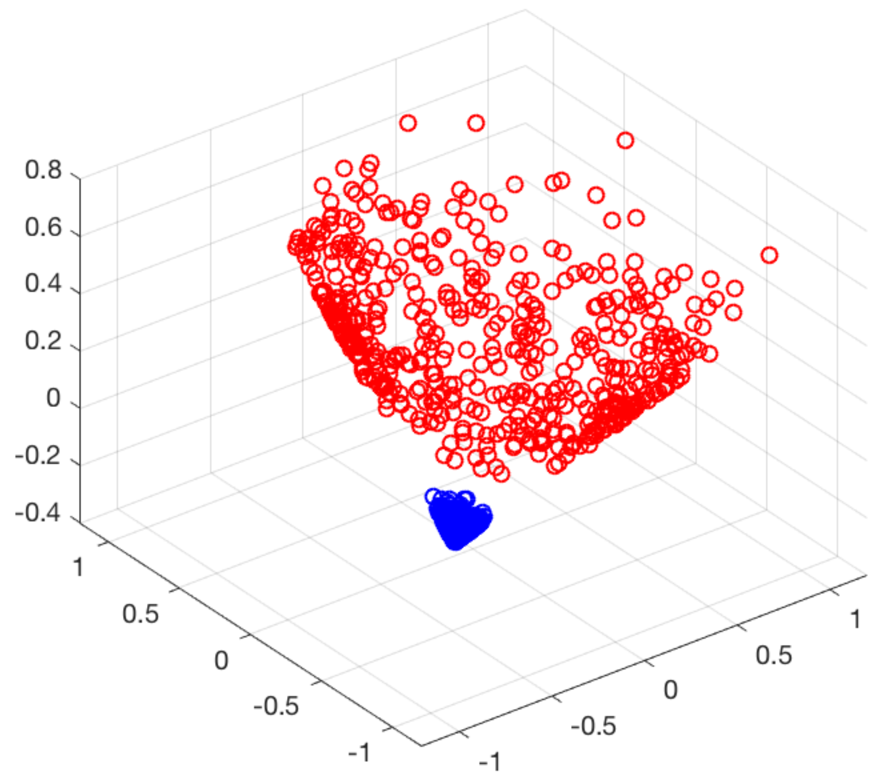

Figure 5: Plot of KPCA with polynomial kernel using the first 3 principal components. It is clear from this 3 dimensional plot that the feature that separates the two colored groups should be the 3rd principal component.

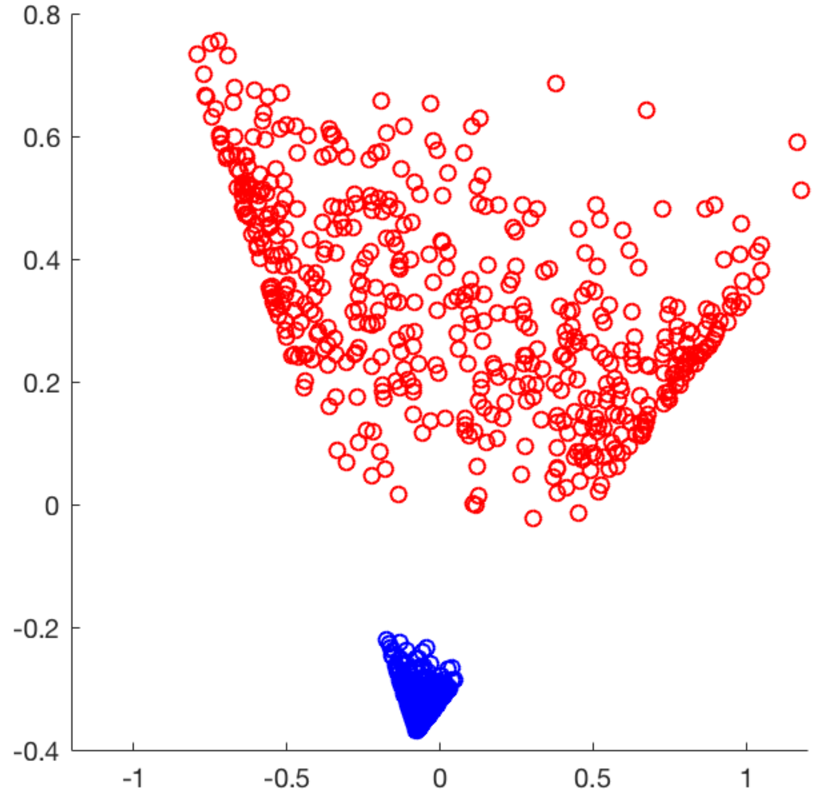

Figure 6: Plot of KPCA with polynomial kernel using the 1st and 3rd principal components. Now the two groups are linearly separable.

The results from KPCA with the Gaussian kernel gives the best results. With the Gaussian kernel, the data are embedded into a space of infinite dimensions, which makes it more likely for the data points to be linearly separable.

5.4 (4)

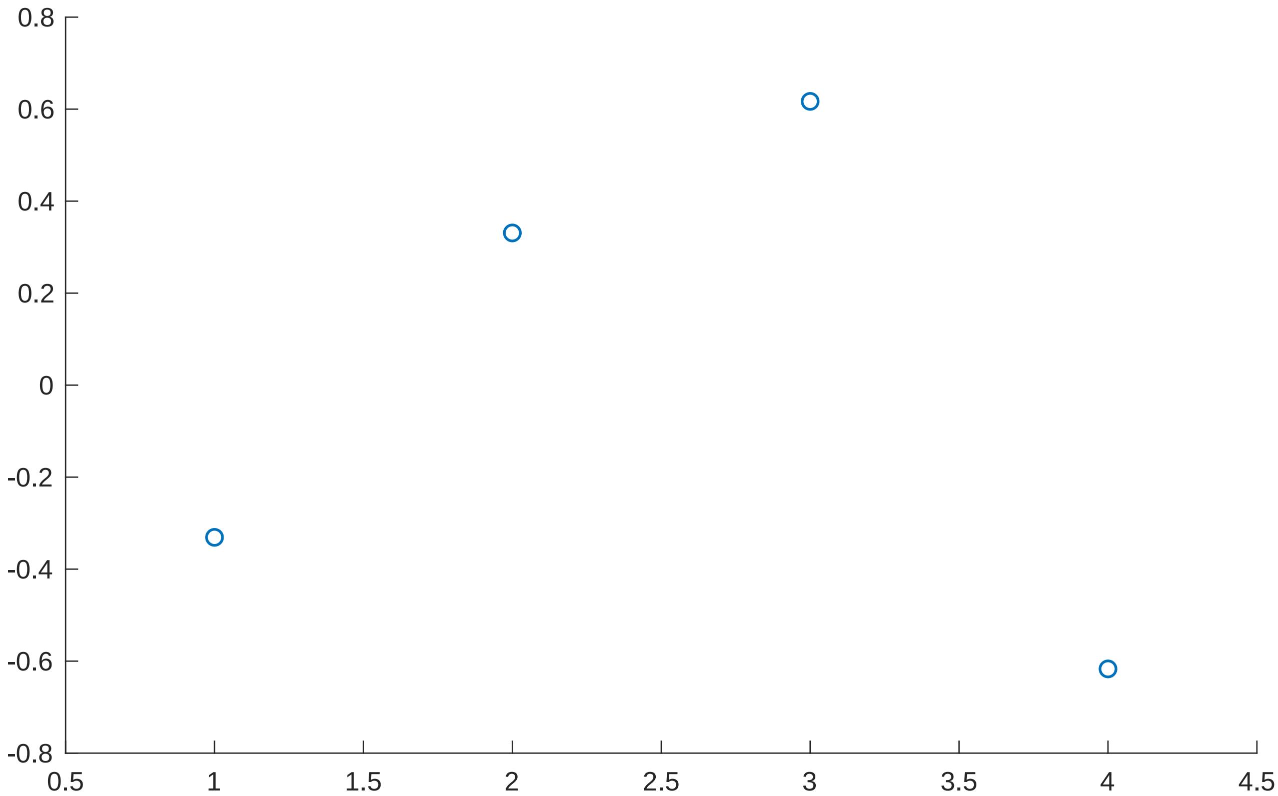

Figure 7: Projection onto the first component of PCA. From left to right are points $(0, -0.7), (0, 0.7), (0.7, 0)$ and $(-0.7, 0)$.

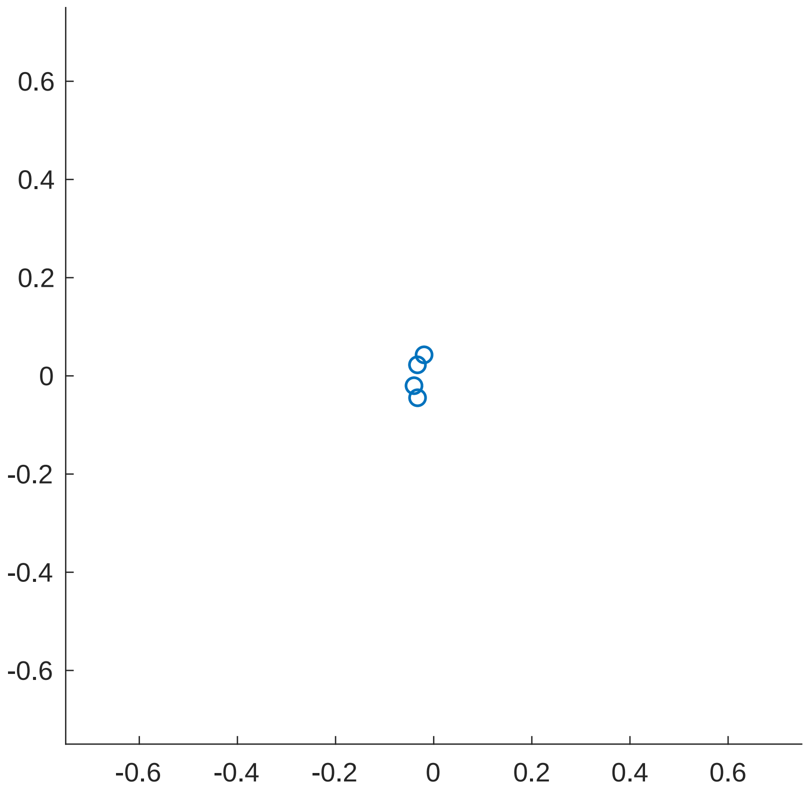

Figure 8: Projection onto the first 2 components of KPCA with Gaussian kernel.

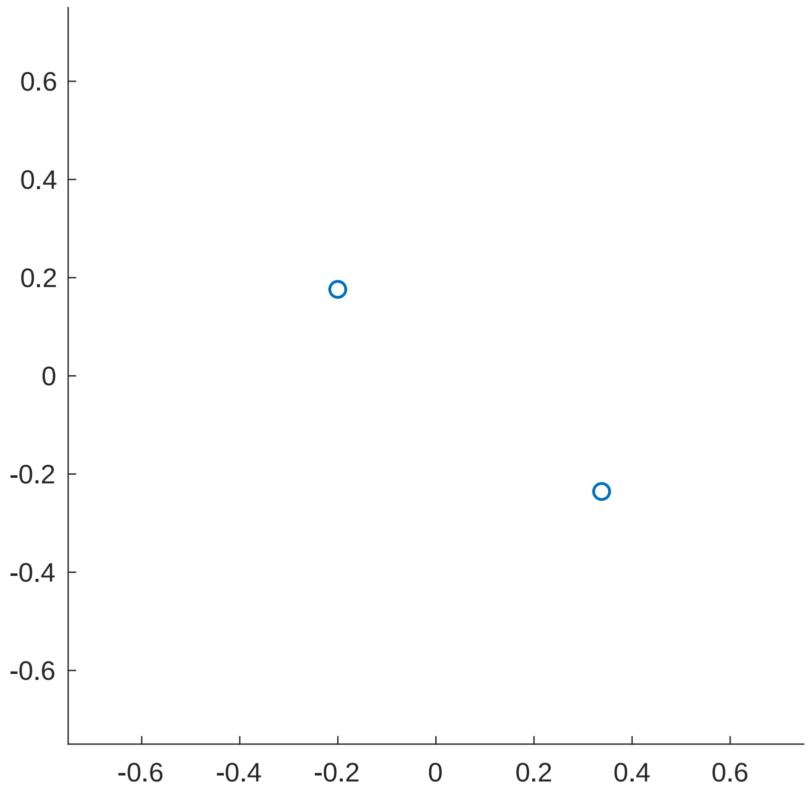

Figure 9: Projection onto the first 2 components of KPCA with polynomial kernel.