Incorporating Kernels

- Recall that we are minimizing $\|\beta\|$ and by setting the derivatives of Lagrangian primal function we have $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$. ... Most things are as in linear kernel.

- The trickiest thing is that now we need to find α_i's by solving a modified Lagrange dual function.

- Lagrange dual function:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to} \quad & 0 \le \alpha_i \le C, \\ & \sum_{i=1}^{N} \alpha_i = 0 \end{aligned} \tag{1}$$

- This is a quadratic programming which we can solve using `cvxopt`.

- However, `cvxopt` solves problems of kind:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2} \alpha^T P \alpha + q^T \alpha \\ \text{subject to} \quad & G\alpha \preceq h, \\ & A\alpha = b \end{aligned} \tag{2}$$

  where $G\alpha \preceq h$ means that the inequality is taken element-wise over the vectors $G\alpha$ and h.

- So, we need to choose matrices P, q, G, h, A, and b so that by inserting them in (2) we get (1).

- Instead of maximizing L_D, we minimize $-L_D$.

- We turn $\frac{1}{2}\alpha^T P\alpha + q^T\alpha$ into $-L_D$ by constructing P and q as follows:

$$P = \begin{pmatrix} & \vdots & \\ \dots & y_i y_j K(x_i, x_j) & \dots \\ & \vdots & \end{pmatrix}_{N\times N} \qquad q = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}_{N\times 1}$$

- We turn $G\alpha \preceq h$ into $0 \leq \alpha_i \leq C$ by letting:

$$G = \left(\begin{array}{c} -I_{N\times N} \\ \hline I_{N\times N} \end{array}\right)_{2N\times N} \qquad h = \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline C \\ \vdots \\ C \end{array}\right)_{2N\times 1}$$

- We turn $A\alpha = b$ into $\sum_{i=1}^{N} \alpha_i = 0$ by letting $A = y^T$ and $b = 0$.

- Now that we have expressed (1) in the form of (2), we use `cvxopt` to solve for α_i's.