Expliquez votre choix des paramètres pour l’optimisation quadratique

Selon les notes de cours(4), Entraînement du « Kernel SVM »

Résoudre le problème de maximisation :

$$\boxed{\begin{array}{l} \underset{\lambda_1,\cdots,\lambda_N}{\arg\max} \quad \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} C_i C_j \lambda_i \lambda_j K_{ij} \\ \text{s.t.} \quad \forall\, i \leq i \leq N. \quad \lambda_i \geq 0. \\ \qquad\qquad\qquad\qquad \sum_{i=1}^{N} C_i \lambda_i = 0. \end{array}}$$

$\Rightarrow$ Transformer le problème de maximisation vers le problème de minimisation

$$\boxed{\begin{array}{l} \underset{\lambda_1,\cdots\lambda_N}{\arg\min} \quad \frac{1}{2} \sum_{i,j=1}^{N} C_i C_j \lambda_i \lambda_j K_{ij} - \sum_{i=1}^{N} \lambda_i \\ \text{s.t.} \quad \forall\, i \leq i \leq N \qquad -\lambda_i \leq 0 \\ \qquad \sum_{i=1}^{N} C_i \lambda_i = 0 \end{array}}$$

$$\min \frac{1}{2} \sum_{i,j=1}^{N} C_i C_j \lambda_i \lambda_j k_{ij} - \sum_{i=1}^{N} \lambda_i$$

$$\Rightarrow \min \frac{1}{2} \underbrace{\sum_{i=1}^{N} \lambda_i}_{\lambda^T} \sum_{i,j=1}^{N} C_i C_j k_{ij} \underbrace{\sum_{j=1}^{N} \lambda_j}_{\lambda} - \sum_{i=1}^{N} \lambda_i$$

Selon user guide de cvxopt,

$$\min \quad \frac{1}{2}x^T P x + \underline{q^T} x. \longrightarrow -\sum_{i=1}^{N} \lambda_i \Rightarrow$$

$$\text{s.t.} \quad Gx \le h.$$

$$Ax + b.$$

$$\underbrace{\overset{\leftarrow N \rightarrow}{(-1, -1, \cdots -1)}}_{q^T} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix}$$

$\Rightarrow q = -$ np.ones(N)

$$P = \sum_{i,j=1}^{N} c_i c_j * \sum_{i,j=1}^{N} k_{ij}$$

$$\underbrace{\begin{matrix} [[\,c_0c_0 & c_0c_1 & \cdots & c_0c_N] \\ [\,c_1c_0 & c_1c_1 & \cdots & c_1c_N] \\ [\;\vdots & & & \vdots\;] \\ [\,c_Nc_0 & c_Nc_1 & \cdots & c_Nc_N\,]] \end{matrix}}_{\sum_{i,j=1}^{N} c_i c_j} * \underbrace{\begin{bmatrix} k_{00} & k_{01} & \cdots & k_{0N} \\ k_{10} & k_{11} & \cdots & k_{1N} \\ \vdots & & & \\ k_{N0} & k_{N1} & \cdots & k_{NN} \end{bmatrix}}_{\sum_{i,j=1}^{N} k_{ij}}$$

$$\begin{array}{cc} \Downarrow & \Downarrow \\ \text{numpy.outer(c, c)} & \text{kernel matrix} \end{array}$$

$\Rightarrow$ P = kernel matrix * np.outer(c, c)

$Gx \leq h \Rightarrow \forall\ 1 \leq i \leq N \quad -\lambda_i \leq 0$

$$N \updownarrow \underbrace{\overset{\longleftarrow\quad N \quad\longrightarrow}{\begin{bmatrix} -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}}}_{G} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{pmatrix} \leq \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{h}$$

$\Rightarrow$ $G = -$np.eye(N)

$h =$ np.zeros(N)

$$Ax = b. \quad \Rightarrow \quad \sum_{i=1}^{N} c_i \lambda_i = 0.$$

$$\underbrace{(c_1, c_2, \cdots c_N)}_{C^T} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix} = \underset{\substack{\downarrow \\ 0}}{0.}$$

$\Rightarrow$ $A = C^T$ (dim $= 1 * N$)

$b = 0.$