EE5904 Part II

Project 1: SVM for Classification of Spam Email Messages

Student Name: Xin Zhengfang

Student Number: A0206597U

Email Address: e0427425@u.nus.edu

Data Preprocessing

In all tasks, I use standardizing method to preprocess the input data by subtracting the mean and dividing the standard deviation of train data. Therefore, we can obtain a “standard normal” input. The formula is shown in Eq. 1.

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

Mercer’s condition check

In all tasks, before doing the kernel mapping, we will do Mercer’s condition check to demonstrate availability of the kernel. If any eigenvalue of kernel gram matrix K is negative, we will do not compute discriminant function g(·).

For training set $S = \{(\mathbf{x}_i, d_i)\}, i = 1,2,\ldots,N$ the Gram matrix is

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1,\mathbf{x}_1) & \ldots & K(\mathbf{x}_1,\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_N,\mathbf{x}_1) & \ldots & K(\mathbf{x}_N,\mathbf{x}_N) \end{bmatrix} \in R^{N\times N} \tag{2}$$

K is should be positive semi-definite (i.e., its eigenvalues are nonnegative) The nonnegative threshold is -1e-4 because numerical reason.

Quadprog toolbox

In the tasks, we use *quadprog*, a toolbox in MatLab, to optimize the SVM dual problem. The function of *quadprog* is shown in Eq. 3.

$$x = quadprog(H, f, A, b, Aeq, beq, 1b, ub, x\theta, options) \tag{3}$$

The meanings of parameters are shown in Eq.4.

$$\min_x \frac{1}{2} x^T H x + f^T x \text{such that} \begin{cases} A \cdot x \le b & \cdot \\ Aeq \cdot x = beq & \cdot \\ b \le x \le ub & \cdot \end{cases} \tag{4}$$

Task 1

i. A hard-margin SVM with the linear kernel

The dual problem functions of hard margin linear kernel SVM is shown in Eq. 4.

$$\text{Maximizing: } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} . \alpha_i - \frac{1}{2}\sum_{i=1}^{N} . \sum_{j=1}^{N} . \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{Subject to :}(1) \sum_{i=1}^{N} . \alpha_i d_i = 0 \tag{5}$$

$$(2)\ \alpha_i \geq 0$$

Therefore, in the code:

$$\mathrm{H}(i_c j) = d_i d_j x_i^T x_j$$
$$f = -\mathrm{ones}(2000,1)$$
$$Aeq = \mathrm{train\_label}'$$
$$\mathrm{Beq} = 0$$
$$lb = zeros\ (2000,1)$$
$$\mathrm{ub} = \mathrm{ones}(2000,1) * C$$
$$x_0 = []$$
$$\mathrm{options} = \mathrm{optimset}('\mathrm{LargeScale}', '\mathrm{Maxiter}', 1000)$$
$$\mathrm{C} = 10^6$$

ii. A hard-margin SVM with a polynomial kernel

The dual problem functions of hard margin ploy kernel SVM is shown in Eq. 6.

$$\text{Maximize: } \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} . \alpha_i - \frac{1}{2}\sum_{i=1}^{N} . \sum_{j=1}^{N} . \alpha_i \alpha_j d_i d_j \left(\mathbf{x}_i^T \mathbf{x}_j + 1\right)^p$$

$$\text{Subject to:} \quad \sum_{i=1}^{N} . \alpha_i d_i = 0 \text{ and } 0 \leq \alpha_i \tag{6}$$

Therefore, in the code:

$$\mathrm{H}(i_c j) = d_i d_j (x_i^T x_j + 1)\hat{}p$$
$$f = -\mathrm{ones}(2000,1)$$
$$Aeq = \mathrm{train\_label}'$$
$$\mathrm{Beq} = 0$$
$$lb = zeros\ (2000,1)$$
$$\mathrm{ub} = \mathrm{ones}(2000,1) * C$$

$$x_0 = []$$
$$\text{options} = \text{optimset}('\text{LargeScale}', '\text{Maxiter}', 1000)$$
$$\text{C} = 10^6$$

iii. A soft-margin SVM with a polynomial kernel

The dual problem functions of soft margin ploy kernel SVM is shown in Eq. 7.

$$\begin{aligned} &\text{Maximize:} \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} . \alpha_i - \frac{1}{2} \sum_{i=1}^{N} . \sum_{j=1}^{N} . \alpha_i \alpha_j d_i d_j \left(\mathbf{x}_i^T \mathbf{x}_j + 1\right)^p \\ &\text{Subject to:} \quad \sum_{i=1}^{N} . \alpha_i d_i = 0 \text{ and } 0 \le \alpha_i \le C \end{aligned} \tag{7}$$

Therefore, in the code:

$$\text{H}(i_c j) = d_i d_j (x_i^T x_j + 1)\hat{}p$$
$$f = -\text{ones}(2000{,}1)$$
$$Aeq = \text{train\_label}'$$
$$\text{Beq} = 0$$
$$lb = \text{zeros}\ (2000{,}1)$$
$$\text{ub} = \text{ones}(2000{,}1) * C$$
$$x_0 = []$$
$$\text{options} = \text{optimset}('\text{LargeScale}', '\text{Maxiter}', 1000)$$
$$\text{C} = 0.1 \sim 2.1$$

Task 2

With the previous work as the basis, we can directly get the following results shown in Table 1.

Table 1. Result of SVM classification
(NA: Not Admissible)

Type of SVM	Training accuracy				Test accuracy			
Hard margin with Linear kernel	93.85%				92.51%			
Hard margin with polynomial kernel	p=2	p=3	p=4	p=5	p=2	p=3	p=4	p=5
	100.00%	100.00%	NA	NA	85.81%	86.65%	NA	NA
Soft margin with polynomial kernel	C = 0.1	C = 0.6	C = 1.1	C = 2.1	C = 0.1	C = 0.6	C = 1.1	C = 2.1
p=1	93.20%	94.05%	93.80%	93.70%	92.38%	93.23%	92.71%	92.51%
p=2	98.95%	99.40%	99.50%	99.55%	91.21%	90.10%	89.58%	89.52%
p=3	99.60%	99.80%	99.80%	99.80%	91.08%	90.23%	90.36%	89.71%
p=4	NA	NA	NA	NA	NA	NA	NA	NA
p=5	NA	NA	NA	NA	NA	NA	NA	NA

Comments

1. As for not admissible kernel, they can still get results, because we have set a max iteration number 1000. The test accuracies of these not admissible kernels are generally bad than those admissible kernels. Therefore, we didn't list them in the table.
2. The hard margin with linear kernel outperforms most of polynomial kernel's results. One reason is the overfitting phenomenon such as hard margin polynomial kernel and soft margin polynomial kernel with high *p*.
3. As shown in the group results in soft margin with polynomial kernel, the overfitting will go further when p or C increases.

In conclusion, the data tend to be linearly separable, therefore, we should choose linear kernel or small p value polynomial kernel and small C to get good results.

Task 3

In this task, I use common kernel choice, radial basis function. The dual problem functions are shown in Eq, 8.

$$\text{Maximize:} \quad Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} . \alpha_i - \frac{1}{2}\sum_{i=1}^{N} . \sum_{j=1}^{N} . \alpha_i \alpha_j d_i d_j \exp\left(-\gamma * |\mathbf{x_i} - \mathbf{x}_j|^2\right)$$
$$\text{Subject to:} \quad \sum_{i=1}^{N} . \alpha_i d_i = 0 \text{ and } 0 \le \alpha_i \le C \tag{6}$$

The performance is shown Table 2. My choice of gamma is 0.00125, and C is 450.

Table 2. Result of Radial Basis function

Training accuracy	Test accuracy
96.60%	94.21%