Project 1: SVM for Classification of Spam Email Messages

Name :	LIANG MING
Matric. No. :	A0195025H
Email :	e0383690@u.nus.edu
Subject:	NEURAL NETWORKS
Assignment :	Project One
Date :	26, April, 2019

Report

Before I start the report, I would like to stress that it is briefer to report task1 and task2 together. Because g(x), with 2000 polynomials, required to be computed in task1 is hard to show entire expression in the report. At the same time, to finish task2, we also need to compute a specific g(x) to get b, so I don't write the expressions of each g(x) in the report. Accordingly, you can check the procedure of computing g(x) in the relative code or analysis in the report.

Task1 & 2

Type of SVM	Training accuracy				Test accuracy			
Hard margin with Linear kernel	93.5%				92.8%			
Hard margin with polynomial kernel	p=2	p=3	p=4（non-convex）	p=5（non-convex）	p=2	p=3	p=4 (non-convex)	p=5(non-convex)
	99.95%	99.95%	69.60%	66.45%	85.81%	85.03%	71.35%	64.19%
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	92.90%	93.35%	93.35%	93.35%	91.99%	92.38%	92.12%	92.58%
p=2	98.70%	99.25%	99.40%	99.40%	90.82%	89.52%	89.26%	89.58%
p=3	99.55%	99.75%	99.75%	99.80%	90.43%	88.87%	87.24%	87.30%
p=4	99.60%	99.80%	99.85%	99.85%	87.50%	85.81%	86.00%	86.00%
p=5	99.25%	99.45%	99.45%	99.50%	87.70%	87.24%	87.04%	86.26%

Table 1: results of SVM classification

According to the demo from teacher assistant, after loading relative data, standardize is the first step to deal with train and test data. We compute **α** firstly, which is an optimization question

$$\begin{cases} minimize: \ -Q(a) = -(\sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j d_i d_j K(x_i, x_j)) \\ Subject\ to: \ \sum_{i=1}^{N} a_i d_i = 0, 0 \le a_i \le C \end{cases} \quad (1)$$

Using quadprog function to solve matrix H and we can get a set of α

$$\mathrm{H(i,j)} = d_i d_j K(x_1, x_2) \quad (2)$$

1. Hard margin with linear kernel

For Hard margin, C is set to be a large value, 1×10^6

After getting α, select a threshold(1e-4) for choosing support vectors. For hard margin with linear kernel

$$\begin{cases} w_0 = \sum_{i=1}^{N} a_{0,i} d_i x_i \\ b_0 = \frac{1}{d^s} - w_0^T x^s \end{cases} \quad (1\text{-}1)$$

So

$$g(x) = w_0^T x + b_0 \tag{1-2}$$

Once we compute g(x), we can set each standardized train data and test data into the g(x), and use sign(g(x)) to evaluate train accuracy and test accuracy, if sign(g(x)) is equal to the relative train label or test label.

Train and test accuracy is 93.5% and 92.8% respectively, both of the test and train accuracy are big enough plus test is approximate to train. So the fitting result is good.

2. Hard margin with polynomial kernel

The kernel now is

$$K(x_1, x_2) = (x_1^T x_2 + 1)^p \tag{2-1}$$

in this case we can't compute w_0, b_0 respectively, so we use another method to get half section of $g(x)$ directly and substitute any support vector and its relative label into g(x) to get b_0

$$g(x) = \sum_{i=1}^{N} \alpha_{0,i}\, d_i K(x, x_i) + b_0 \tag{2-2}$$

$$b_0 = \frac{1}{d^s} - \sum_{i=1}^{N} \alpha_{0,i}\, d_i K(x_s, x_i) \tag{2-3}$$

After getting $g(x)$ and b_0 , we can use same approach as introduced in 'hard margin with linear kernel' (sign(g(x))) to compute train and test accuracy.

In addition to that, when p equals to 4 and 5 the kernel matrix is not semi positive definite, in this order, which mean that the optimization for α is not a convex question, so the classification result is not meaningful.

When p is 2 or 3, both the train accuracy are close to 100%, whereas accordingly test accuracy is only around 85%.

3. Soft margin with polynomial kernel.

Soft margin with polynomial kernel has the same train of thought with hard margin with polynomial kernel. The only difference is the value of C, in this task , we set C as 5 different value to study C's effect on the classification result. Quote equation 2-1, 2-2 and 2-3 to get the $g(x)$ and compare $\mathrm{sign}(g(x_s))$ with $\frac{1}{d^s}$ if they are equal to compute train and test accuracy.

From table 1, we can analysis that

- The amount of train accuracy keep rising as C is increasing no matter the value of p.
- When p = 1, kernel is $x^T x + 1$, which can be seen as linear kernel. The value of train accuracy and test accuracy are also very close to those of hard margin with linear kernel (93.5%and 92.8%).
- When p is 2,3,4,5, kernel becomes non-linear. It is interesting that, to some extent, classification result is very similar to that of hard margin with polynomial kernel: all the train accuracy are very high, close to 100%. At the same time, the difference between train and test accuracy is a little big, approximately 10% to 13%. **On the whole, when p is 2 and 3, if we set suitable C, test accuracy of soft margin can be better than that of hard margin. And combined with the analysis when p = 1, we can guess that soft margin is better than hard margin.**

Task3

In task3, I set gaussian kernel for my own SVM,

$$\mathrm{K}(x_1, x_2) = \exp(-{\|x_1 - x_2\|^2}/{\sigma^2}) \tag{3-1}$$

Type of SVM	Training accuracy				Test accuracy			
Soft margin with unlinear kernel	C = 0.2	C = 0.6	C = 1.1	C = 2.1	C = 0.2	C = 0.6	C =1.1	C = 2.1
σ =0.1	99. 85%	99. 9%	99. 90%	99. 90%	68. 10%	70. 44%	70. 96%	70. 96%
σ =0.5	99. 35%	99. 65%	99. 70%	99. 75%	71. 35%	74. 35%	75. 33%	75. 26%
σ =1	97. 40%	99. 35%	99. 95%	99. 6%	74. 09%	78. 06%	78. 91%	79. 04%
σ =5	92. 20%	95. 05%	95. 90%	96. 55%	89. 97%	91. 93%	92. 12%	92. 06%
σ =10	92. 20%	93. 55%	93. 80%	94. 60%	91. 80%	92. 58%	92. 51%	93. 16%

Table 2: results of SVM classification with gaussian kernel

According to Table 2, choose

$$\begin{cases} \sigma = 10 \\ C = 2.1 \end{cases} \tag{3-2}$$

In this case, test accuracy is highest and very close to train accuracy.

The final kernel is

$$\mathrm{K}(x_1, x_2) = \exp(-{\|x_1 - x_2\|^2}/{10^2}) \tag{3-3}$$