EE5904 NEURAL NETWORKS

Project 1 : SVM for Classification of Spam Email Messages

Name: Yu Shixin

Matriculation Number.: A0195017E

Email: shixin@u.nus.edu

Date:2019/4/9

Task 1:

The goal of task 1 is to compute the discrimination function g(·), so first I should **Check Mercer Condition**. Using the normalized train data to calculate the **Gram matrix** for the following SVMs, respectively. (I apply Standardization method to pre-process all the data.)

i. Hard-margin SVM with the linear kernel

$$K(X_1, X_2) = X_1^T X_2$$

ii. Hard-margin SVM with a polynomial kernel

$$K(X_1, X_2) = (X_1^T X_2 + 1)^p$$

iii. Soft-margin SVM with a polynomial kernel

$$K(X_1, X_2) = (X_1^T X_2 + 1)^p$$

The Gram matrix's eigenvalues are non-negative in theory. However, in practice, we should choose an appropriate threshold for the eigenvalues. In this task, the threshold for eigenvalues is $\mathbf{-10^{-4}}$. If any eigenvalue of gram matrix is smaller than the threshold, this kernel candidate is not admissible.

A kernel satisfying the Mercer condition ensures the existence of a global optimum for the resulting optimization problem.

The other parameters of the **quadprog** is:

parameter	**value**
H(i,j)	$\mathrm{d_i} d_j K(X_1, X_2)$
C	10^6/0.1/0.6/1.1/2.1

f	-ones(2000,1)
Aeq	train_label’
Beq	0
lb	zeros(2000,1)
ub	ones(2000,1)
x0	[]
options	optimset (‘LargeScale’,‘off’,‘MaxIter’,1000)

Then introduce those parameters to the quadprog function, and the α, which is larger than threshold (10^{-4}) determine the support vectors.

For **linear kernel**, we can calculate w_o and b_o as follows:

$$w_o = \sum_{i=1}^{N} \alpha_{o,i} d_i x_i \;, b_o = \frac{1}{d^{(s)}} - w_o^T x^{(s)}$$

Choose one of the support vector can get the b_o.

For **non-linear kernel**, we should calculate b_o as follows:

$$g(x) = \sum_{i=1}^{N} \alpha_{o,i} d_i K(x, x_i) + b_o \,,\; g\left(x^{(s)}\right) = \pm 1 = d^{(s)} \,, b_o = \frac{\sum_{i=1}^{m} b_{o,i}}{m}$$

Task 2:

Result of SVM classification

Type of SVM	**Training accuracy**	**Test accuracy**

Hard margin with Linear kernel	93.50%				92.77%			
Hard margin with polynomial kernel	P=2	P=3	P=4	P=5	P=2	P=3	P=4	P=5
	99.95%	99.95%	non-convex	non-convex	85.81%	85.03%	non-convex	non-convex
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	99.05%	99.65%	99.45%	99.50%	87.83%	87.50%	86.91%	86.26%

Table.1 Result of SVM classification

For the result of classification, all the parameter is default, with C (for hard margin) is 10^6, the threshold(for alpha) is 10^{-4} and threshold(for Gram matrix) is -10^{-4}.

Comment:

According to the result of SVM classification, I find that:

1. For hard-margin with polynomial, when $C = 10^6$,$P = 4\&5$, the result is non-convex, which means it does not exist an optimum solution for this case and not satisfy the Mercer's condition. However, even if the kernel does not satisfy the Mercer's condition, such as $P = 5$,$C = 2.1$, the min eigenvalue is -2.208, but it is still convex for quadprog function. I guess that when C is big enough($C = 10^6$),

the quadprog function can not find an optimal solution, but when C=1.1 or 2.1, the quadprog function can find an optimal solution during [0, 2.1].

2. For hard-margin, the performance of linear kernel is better than that of polynomial kernel, with **92.77%** versus **85.81% (p=2)** & **85.03% (p=3)**.
3. For the same polynomial kernel, the performance of soft-margin is better than that of hard-margin. When $P = 4\&5$, both soft-margin and hard-margin do not satisfy the Mercer's condition, but soft-margin is convex for quadprog function with a lower accuracy than $P = 1\&2\&3$.
4. For soft-margin, when **p=2&3&4&5**, almost of train data accuracy can reach to 99% around. However, the state of test data is opposite to the train data, which gets higher accuracy (**around 92%**) when **p=1** in any value of C. This reason is might that when soft-margin with p=2&3&4&5, the train data is over-fitting, and all the train data can be fitted very well.

In conclusion, just for this table, hard-margin with linear kernel or soft-margin with polynomial kernel and $P = 1$ can get a higher accuracy for test data. And polynomial kernel with $P = 2\&3$ can get a higher accuracy for train data.

Task 3:

In order to design a SVM of my own, I need to find which value of those parameters is better. I will discuss the effect as follow.

1. Kernel function:

   Except linear and polynomial kernel, the Gaussian is still a common kernel function. I choose Gaussian Kernel as kernel function.

2. The effect of C (the threshold for SVM), when sigma=25:

Value of C	**Train accuracy**	**Test accuracy**
0.1	90.3%	90.365%

1	92.7%	92.448%
10	94%	93.359%
13.5	***94.25%***	***93.555%***
15	94.4%	93.49%
20	94.65%	93.164%
50	95.25%	92.969%
100	96.15%	92.969%

From the table above, we can get the highest test accuracy when $\boldsymbol{C} = \mathbf{13.5.}$

3. The effect of sigma (for Gaussian Kernel), when C=13.5:

Value of sigma	**Train accuracy**	**Test accuracy**
0.1	99.9%	70.312%
1	99.85%	77.083%
10	98.15%	92.318%
25	***94.25%***	***93.555%***
50	95.25%	92.969%
100	96.15%	92.969%

From the table above, we can get the highest test accuracy when $\boldsymbol{sigma} = \mathbf{25}$.

This file has been saved in svm_main.m. if you run this file, please put “train.mat” and “eval.mat” with this file together.