EE5904 Neural Networks

Project 1: SVM for Classification of Spam Email Messages

Name: Zhang Fei

Matriculation No.: A0117981X

Email: E0506561@u.nus.edu

Introduction: This project aims to implement SVMs with different kernels. And our task is to train SVM classifiers to predict accuracies on train set, test set and evaluation set data. Different parameters in the SVM models are investigated to observe the effect on the model.

Task 1: Implement the discriminant function g(·)

To check whether there exist such discriminant function g(·), we can apply Mercer Condition theorem. Firstly, retrieve Gram Matrix denoted by K for each SVM model:

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

We need to calculate the eigenvalues for Gram Matrix K. The discriminant function g(·) exists only when K has non-negative eigenvalues. However, in practical, some eigenvalues may be very small which will easily fail Mercer Condition check. Therefore, a threshold eigenvalue -10^{-4} is set for Mercer Condition check. Then we can compute discriminant function as such:

$$g(x) = w_0^T \varphi(x) + b_0 = \sum_{i=1}^{N} \alpha_{0,i} d_i K(x, x_i) + b_0$$

1. A Hard-margin SVM with the linear kernel:

$$\mathrm{K}(x_1, x_2) = x_1^T x_2$$

2. A hard-margin SVM with a polynomial kernel:

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

A hard-margin SVM can be approximated by a soft-margin SVM with a very large C value. C is set to 1×10^6.

3. A soft-margin SVM with a polynomial kernel:

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

Parameters in quadprog function:

Parameters	Value
H	$d_i d_j K(X_1, X_2)$
f	-1 * one(2000,1)
A	[ ]
b	[ ]

Aeq	train_label’
beq	0
lb	zeros(2000,1)
ub	C * one(2000,1)
X0	[ ]
C	[10^6, 0.1, 0.6, 1.1, 2.1]
options	optimset (‘LargeScale’,‘off’,‘MaxIter’,1000)

The quadprog function will produce a matrix of α values. The α values that are greater than the threshold value 1x 10^{-4} will determine the support vectors.

Task 2: Evaluate SVMs Classifiers on train set and test set

Table1 Results of SVM Classification

Type of SVM	Train accuracy				Test accuracy			
Hard margin with Linear kernel	93.3%				92.64%			
Hard margin with Polynomial kernel	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
	100%	100%	Non Convex	Non Convex	85.29%	86.65%	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	93.2%	94.05%	93.8%	93.7%	92.38%	93.23%	92.71%	92.51%
P = 2	98.95%	99.4%	99.5%	99.55%	91.21%	90.04%	89.58%	89.58%
P = 3	99.6%	99.8%	99.8%	99.8%	91.08%	90.23%	90.43%	89.71%
P = 4	99.8%	99.9%	99.9%	99.9%	89.00%	88.22%	88.35%	88.28%
P = 5	99.7%	99.9%	99.7%	99.7%	89.13%	88.54%	88.54%	88.48%

Comment:

For A Hard-margin SVM with the linear kernel:

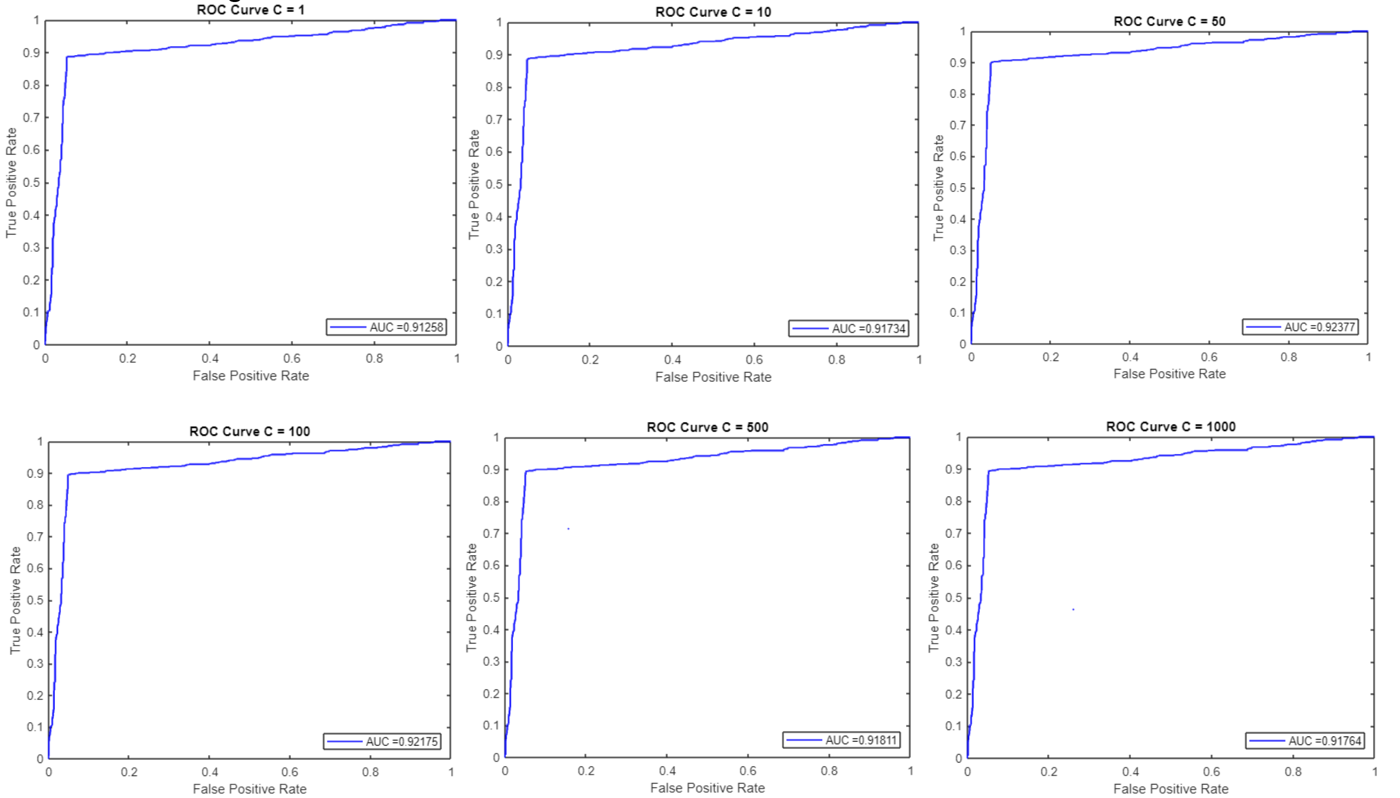


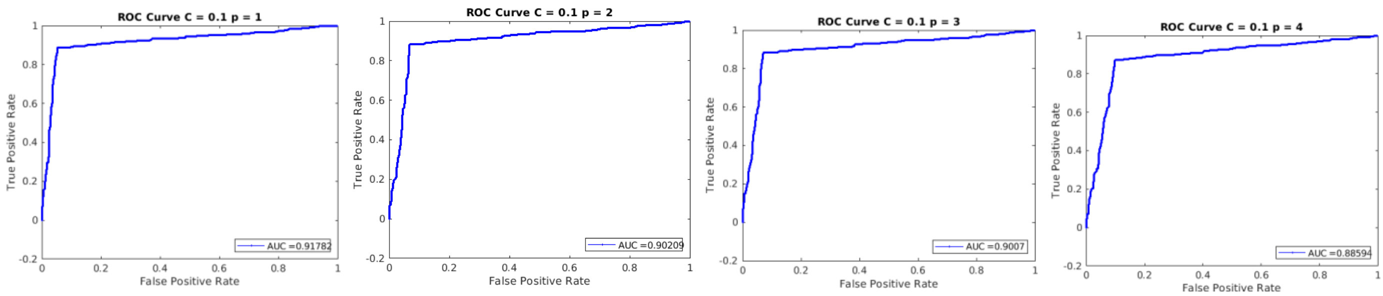
Figure 1 Linear kernel SVM ROC with different C

Figure 1 shows the ROC curves of test data with different C on linear kernel SVM model. Varying the C parameter does not seem to affect the performance of the model, and each area under ROC does not change much.

For A hard-margin SVM with a polynomial kernel:

We can see that the accuracies in train set are perfect 100% for polynomial orders 2 and 3. However, the accuracies in test set are between 85%-86%. The hard-margin SVM with a polynomial kernel overfits train data set. Thus, it cannot generalize on test set. And when increasing polynomial order, the results are non-convex. It is difficult for program to find out optimal solution. And the polynomial order of 2 and 3 does not affect the performance too much. The best accuracy obtained on test set is 86.65% when p is 3.

A soft-margin SVM with a polynomial kernel:



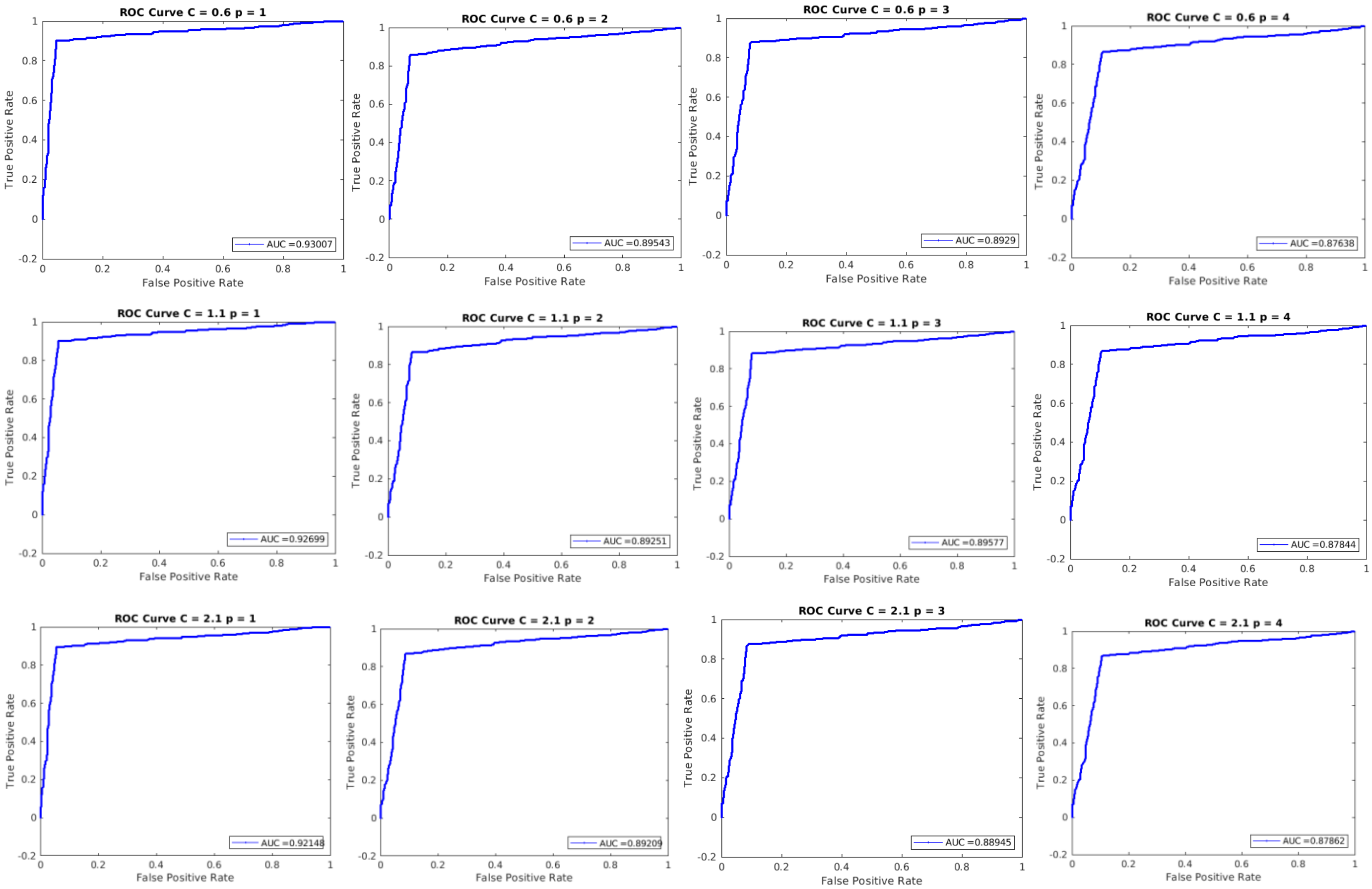


Figure 2 Polynomial kernel SVM ROC with different C and p

Figure 2 shows the soft-margin Polynomial kernel SVM ROC with different C and p. with the same p value, as C increase, the ROC curves and areas of AUC don't change much. The choice of C does affect much of the test data performance, but the optimal value for C may not necessarily be the lowest one. When increase the order of polynomial kernel with C unchanged, accuracies on train set increase while decrease on test set. As we can see that as p increase, the ROC curves are pulled away oppose the perfect point, and area of each AUC decreases. High order of polynomial kernel makes train data overfitted. The best accuracy obtained is 93.23% when C = 0.6 and p =1.

In conclusion, when given an arbitrary dataset, we will not know which kernel works best. We can use the simplest kernel first. Compare the best results from linear model and soft-margin polynomial model, they look like perform equally well. However, for polynomial model, the complexity grows with the size of train set size and polynomial order. It will get much expensive to train the kernel SVM and predict when the projection into infinite higher dimension space. Instead of tuning more hyperparameters, the linear kernel is preferred with the equally well performance.

Task **3: Design my own SVM**

I implemented SVM with Radial basis function kernel:

$$K(x_1, x_2) = exp(-\gamma||x_1 - x_2||^2)$$

Table 3 Results of RBF kernel

C	Train accuracy	Test accuracy
C = 0.5	91.95%	91.67%
C = 1	92.50%	91.47%
C = 10	94.30%	92.9%
C = 20	95.00%	93.16%
C = 50	95.45%	93.49%
C = 100	96.20%	93.45%
C =200	96.60%	93.75%
C =500	97.00%	93.23%

Table 3 shows the accuracy results of train set and test set. The parameter γ is 0.002. and as C increases, both accuracies on train set and test set increase. I choose the RBF kernel with parameters C = 500 and γ = 0.002