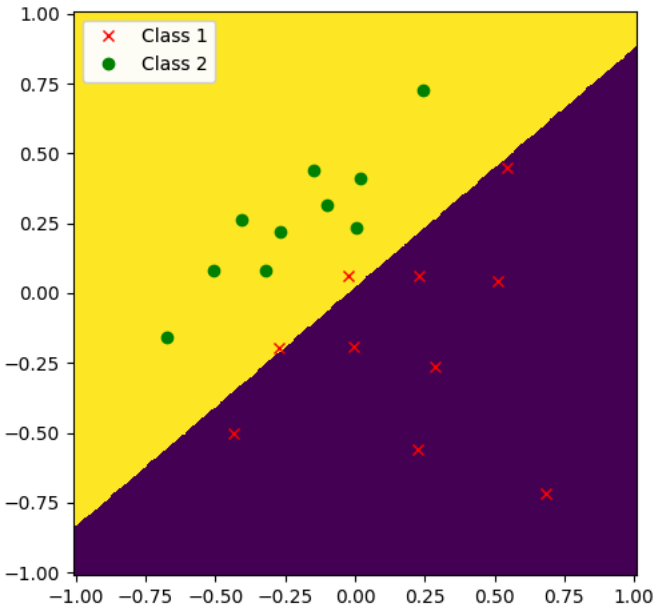
EE559 Week10 HW 1139093874 Pin-Hsuan Lee

1.

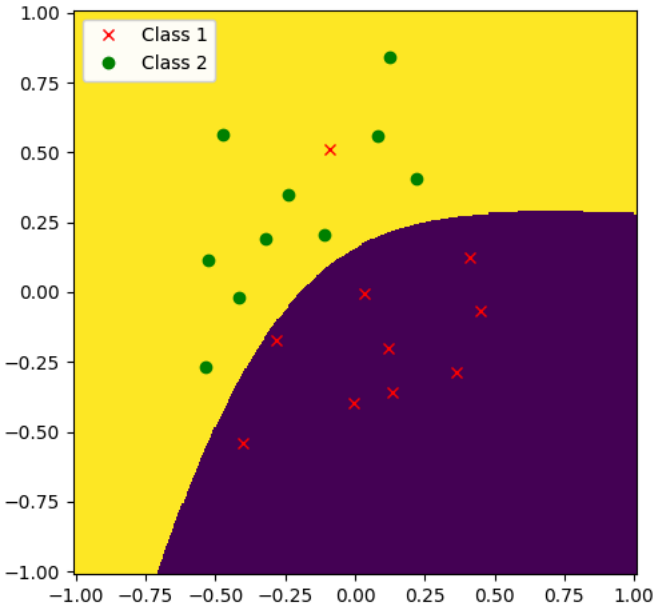
(a)

In SVM, C is a penalty for misclassifying a data point. When C is small, the misclassified data points will not make huge influences to the classifier, so it can accommodate more misclassified data points. When C is large, the classifier is heavily penalized for misclassified data and therefore curves to avoid misclassified data points.

When C=1, accuracy is 90%.



When C=100, accuracy is 100%.



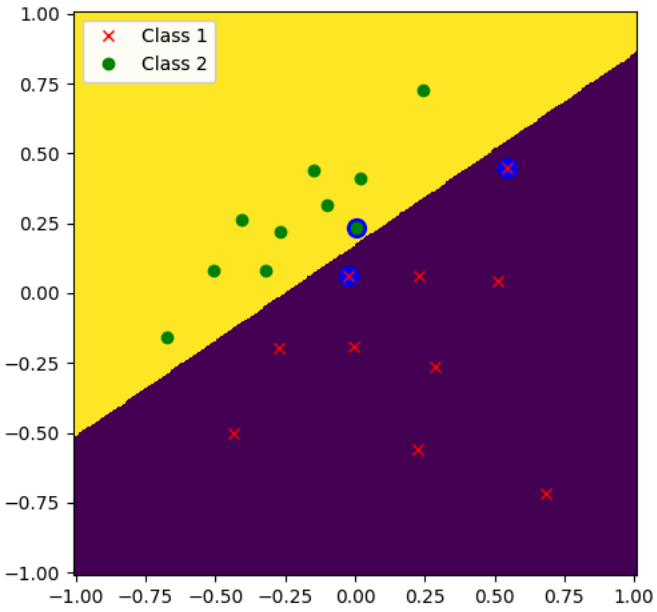
(b)

```
W0= -1.7983676588125648
W1= -7.119663837669007
W2= 10.402648206879697
The decision boundary equation is ( -7.119663837669007 )X1+( 10.402648206879697 )X2+( -1.7983676588125648 )=0
```

There are three pairs of support vectors as follows

```
[[-0.023855   0.06042  ]
 [ 0.54579    0.45029  ]
 [ 0.0064864  0.23394  ]]
```

The support vectors are circled in the plot below when C = 100 with Linear Kernel.



(c)

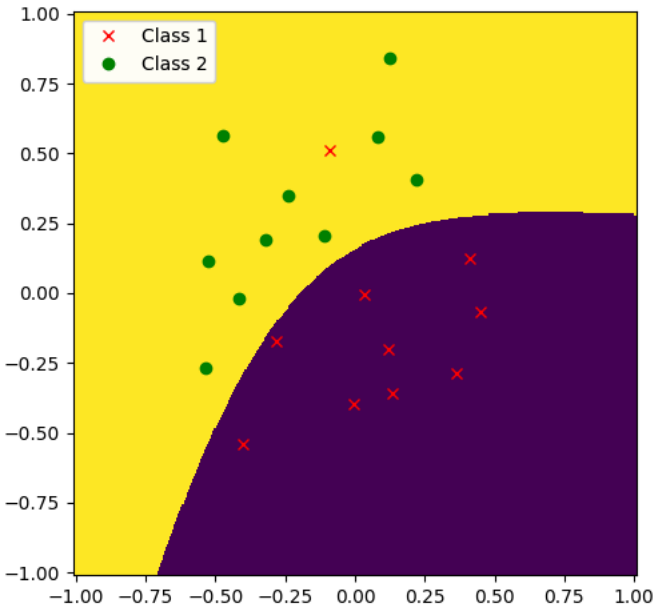
```
g(X) of support vector 1 when C=100 is -1.0000000733052992
g(X) of support vector 2 when C=100 is -1.000000523698073
g(X) of support vector 3 when C=100 is 0.5890468751882154
```

The first and second support vectors are on the boundary of the margin (corresponding to $g(x) = \pm 1$). However, the third support vector is within the margin. The situation happens because when c is small, the misclassified data points are more acceptable in the classifier. Nevertheless, when c is large, the misclassified points will penalize a lot to the classifier and the slack variables will minimize to zero. As we can see in the following example, when C=10000000, the support vectors are on the boundary in order to reduce the penalty.

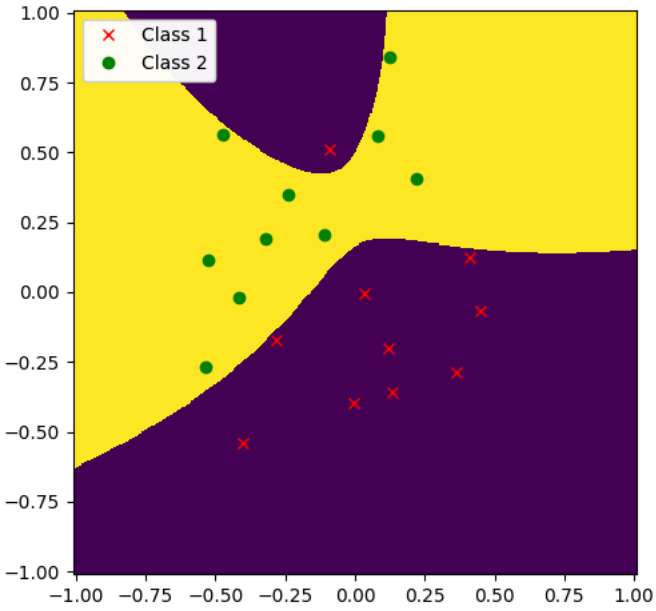
```
g(X) of support vector 1 when C=10000000 is -0.9996883756962438
g(X) of support vector 2 when C=10000000 is -1.0001565385588216
g(X) of support vector 3 when C=10000000 is 0.999844283004963
```

(d)

When C=50, the accuracy is 95%



When C=5000, the accuracy is 100%

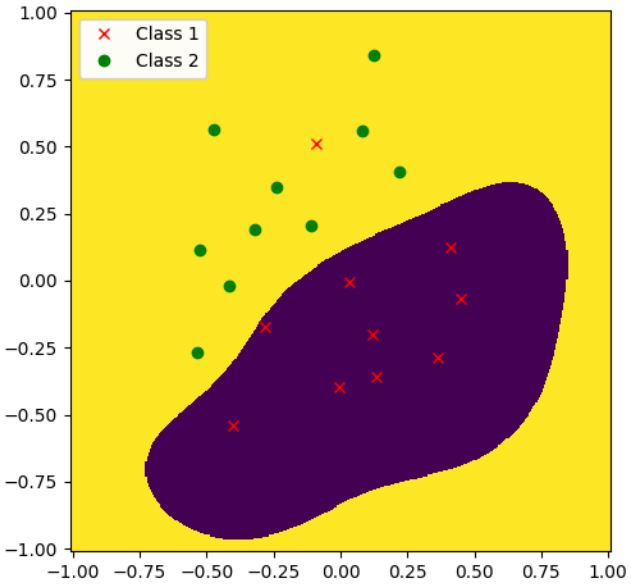


First, the data in original feature space is nonlinear and the Gaussian Kernel transforms the original feature space to a higher dimensional feature space, therefore, the decision boundary is nonlinear.

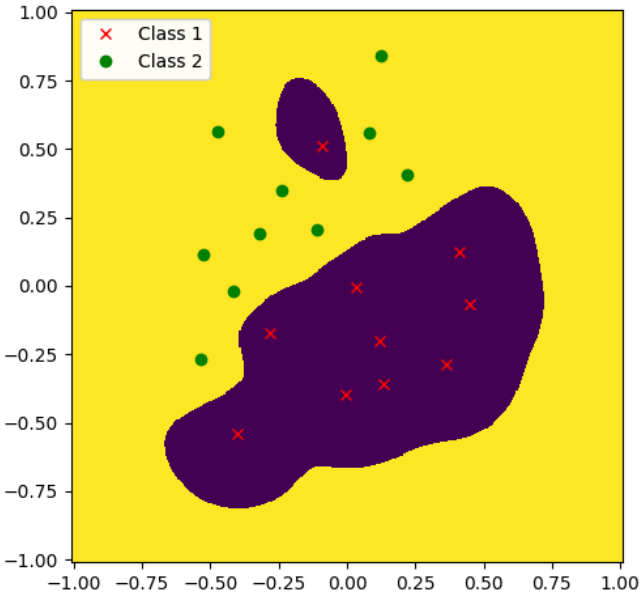
Second, when we observe the C penalty parameter in RBF, we find that we get better accuracy as C becomes larger and the reason is the same as we answered before. When C is small, the misclassified data points are more acceptable in the classifier, therefore, the decision boundary is not so precise and the accuracy can be lower. Nevertheless, when C is larger, the misclassified points will penalize a lot to the classifier and the slack variables will become smaller to keep a balance. As a result, the decision boundary will be more precise which causes a better accuracy.

(e)

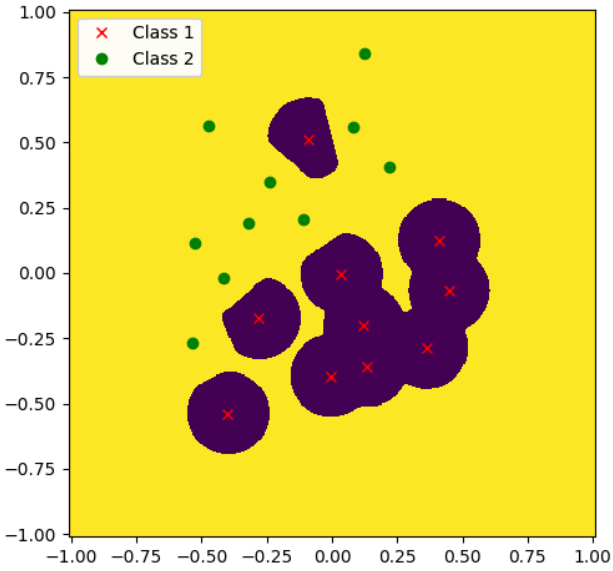
When $\gamma = 10$, the accuracy of the classifier is 95%.



When $\gamma = 50$, the accuracy of the classifier is 100%.



When γ = 500, the accuracy of the classifier is 100%.



γ is a parameter of the RBF kernel and can be thought as the "spread" of the kernel and therefore the decision region. When γ is low, the decision region is wider. When γ gets larger, the decision boundary becomes narrower which fits the data points better. However, when γ is too larger as we can see when γ = 500, the decision region looks like circling each of the training data points and causes the overfitting. Therefore, We can know that when we use γ which is too large to test the test dataset, we will get poor accuracy.

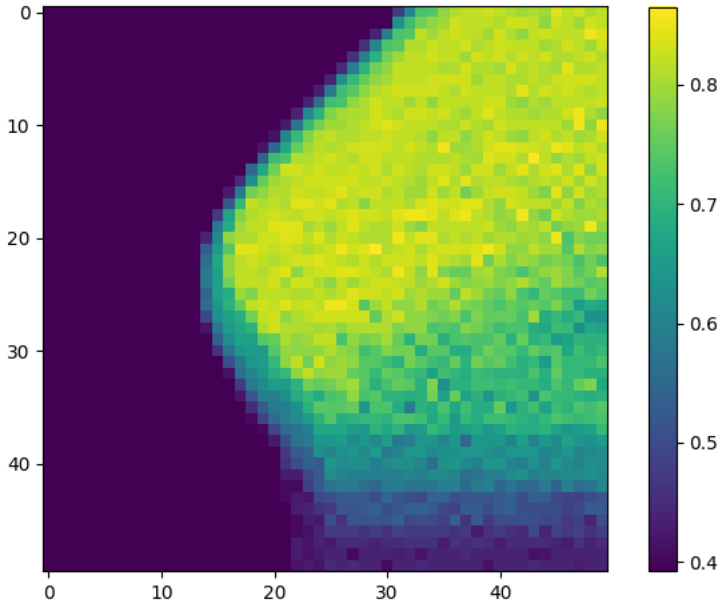
2.

(a)The average cross-validation accuracy is 0.8431 when using Gaussian Kernel and setting γ=1, C=1.

(b)

(i)

The plot showing below is the accuracy map when we set γ range from 10^{-3} to 10^{3} and C range from 10^{-3} to 10^{3}. We generate 50 samples within the range for both γ and C and put it into the 2D array of size(Nγ,N_C). The y-axis is the order of rows of the array and the x-axis is the order of columns of that.



(ii)

Most of the time, we can have a max accuracy. However, we can still get more than one max accuracy which are the same values sometimes. If that happens, we can choose the pair with the lowest standard deviation as our best choice.

In my trial this time, there is a clear choice of pair.

```
The max mean accuracy is 0.866 and the unbiased standard deviation is 0.062
and it happened when gamma=0.373,C=10.985
```

(c)

(i)

```
1. (r,C)= (0.655,8.286)
2. (r,C)= (0.121,59.636)
3. (r,C)= (0.039,8.286)
4. (r,C)= (0.002,1000.000)
5. (r,C)= (0.022,323.746)
6. (r,C)= (0.494,6.251)
7. (r,C)= (0.494,44.984)
8. (r,C)= (0.212,44.984)
9. (r,C)= (0.494,33.932)
10. (r,C)= (0.494,14.563)
11. (r,C)= (0.029,323.746)
12. (r,C)= (0.160,14.563)
13. (r,C)= (0.373,10.985)
14. (r,C)= (0.655,1.526)
15. (r,C)= (0.373,2.024)
```

```
16. (r,C)= (0.091,323.746)
17. (r,C)= (0.160,2.683)
18. (r,C)= (0.091,244.205)
19. (r,C)= (0.281,33.932)
20. (r,C)= (0.039,79.060)
```

(ii)

In real situation, we know nothing about the test data. The reason why we shuffle the data and then average the mean accuracy is to simulate the real condition. Therefore, it's more well-defined to pick up the best values for [γ,C] over the T=20 runs.

```
The the final chosen best values for (r,C) is ( 0.373,2.683 )
mean cross-validation accuracy : 0.838
standard deviation : 0.016880
```

(d)

The estimate is more than one standard deviation. It's because that in the real situation, the classification depends on 13 features. However, we only use 2 features to train the classifier and test on the test dataset. Therefore, the classifier is more than one standard deviation.

```
When we use the chosen (r,C), the accuracy is 0.798
The estimate is approximately 2.360 standard deviation of the mean cross-validation accuracy from (c)
```