CMPS 242 Fall 2018: Machine Learning HW5

Sepehr Ramyar, Weiting Zhan, Liwei Liu*

Problem 1

Part (a)

In this part, the major steps are implemented as follows:

- 1000 training points from the first and another 1000 training points from the second Gaussian are generated and accordingly labeled +1 and -1
- Similarly, 2000 test data points are generated with the first 1000 labeled +1 and the next 1000 labeled -1
- The SVM is fit on the training data for different γ values

The results are summarized as follows. The raw data is plotted in Fig.1. The +1-labeled data is the blue pluses and orange x's signify the data generated from the second Gaussian labeled -1. It evident that the second (orange) distribution, given the larger eigenvalues of the covariance matrix, is more spread-out.

Training an SVM with $C = 1$ and `RBF` kernel for different values of γ would result in decision boundaries that are shown in Fig.2 to Fig.6. The accuracy results and number of support vectors are reported in Table 1:

Gamma(γ)	0.1	1	10	100	1000
Accuracy	0.8675	0.8705	0.8540	0.8130	0.6730
Support Vectors	662	703	1171	1763	1988

Table 1: Accuracy and number of support vectors for different gamma (γ values.

We can see that as gamma (γ) increases, we first see improved performance (once γ=0.1 changes to γ=1) in terms of accuracy as well as a more specific decision boundary. However, as gamma increases from 1 to 1000, we see a monotonic decrease in performance. This is due to over-fitting as larger values of gamma place excessively high value on each data point and less emphasis on

*{sramyar,wzhan83, lliu61} @ucsc.edu

nearby points. Consequently, for higher gamma values, we see higher number of support vectors as the classification become more granular. Lower gamma values (like 0.1 and 1 in this case), however, place a more balanced weight for nearby points and hence generalize better. This is shown in the Figs. 2 and 3.

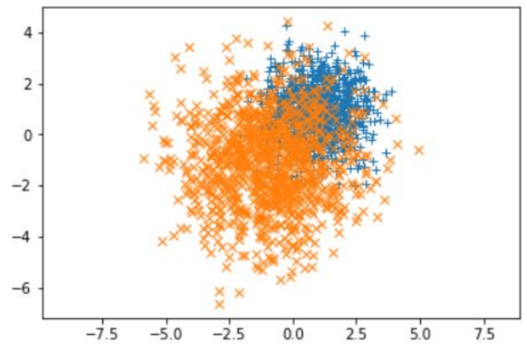

Figure 1: Training data generated from the two Gaussian distributions

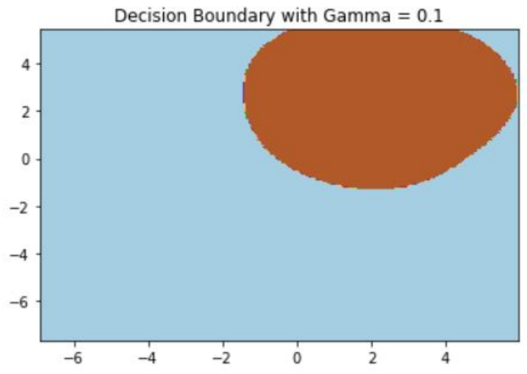

Figure 2: Decision boundary for $\gamma = 0.1$

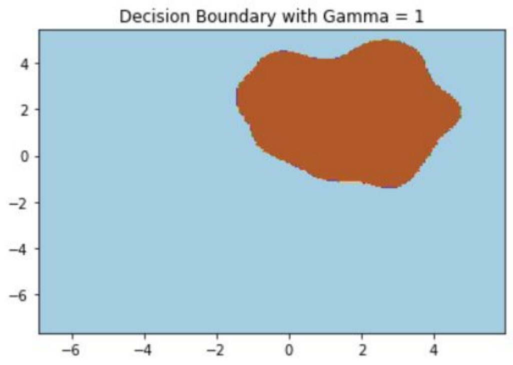

Figure 3: Decision boundary for $\gamma = 1$

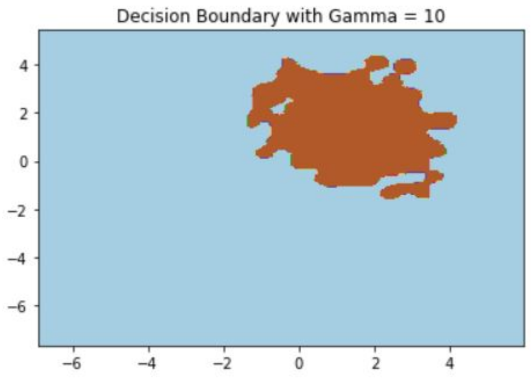

Figure 4: Decision boundary for $\gamma = 10$

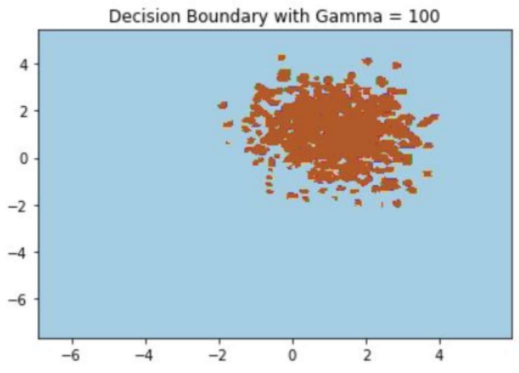

Figure 5: Decision boundary for $\gamma = 100$

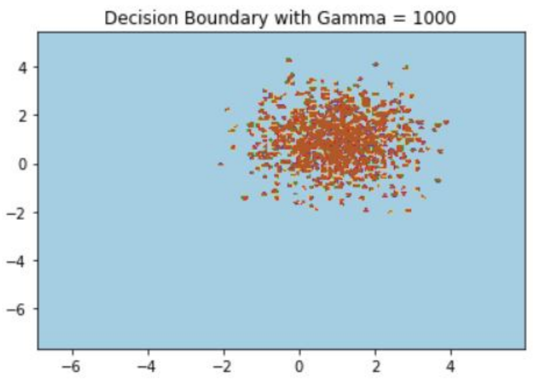

Figure 6: Decision boundary for $\gamma = 1000$

Part (b)

In terms of implementation, this section is similar to the previous one except in this case, we change the values for the parameter C. The training data from both distributions as well as the decision boundaries are illustrated in Fig.7 to Fig.12. We can see as C increases in magnitude, the decision boundary becomes more wiggly as more weight is being put on margin errors. This causes the margin to shrink.

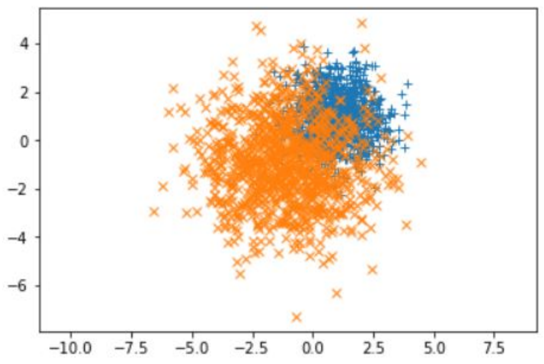

Figure 7: Training data generated from the two Gaussian distributions for part (b)

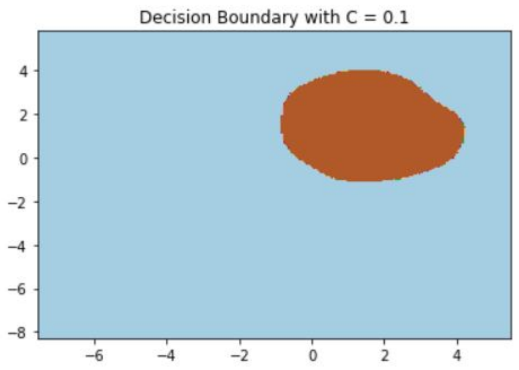

Figure 8: Decision boundary for $C = 0.1$

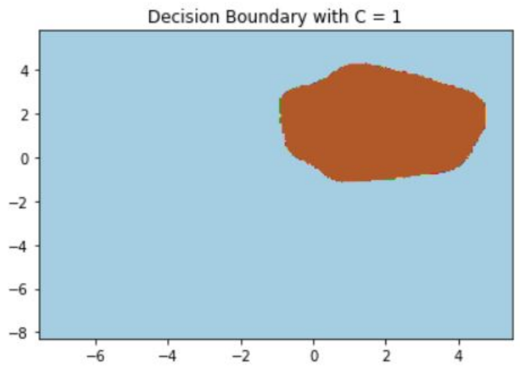

Figure 9: Decision boundary for $C = 1$

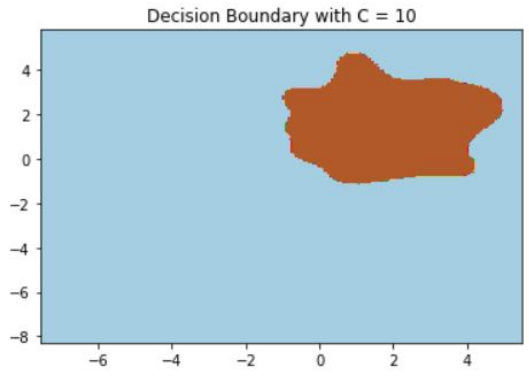

Figure 10: Decision boundary for $C = 10$

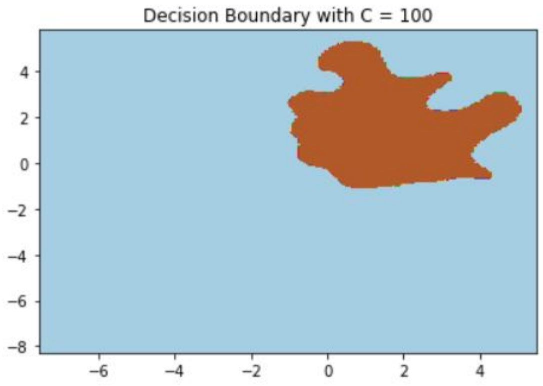

Figure 11: Decision boundary for $C = 100$

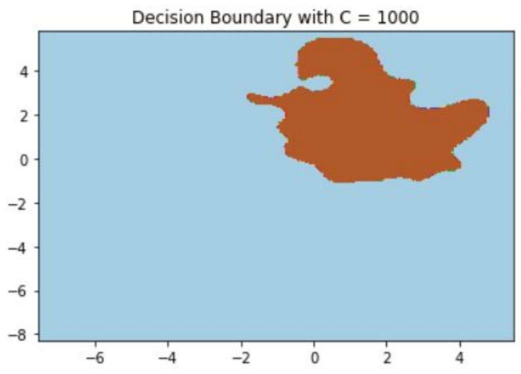

Figure 12: Decision boundary for $C = 1000$

The accuracy and number of support vectors are given in Table 2. We can see that accuracy does not dramatically change as we increase C. It does seem to slightly reduce as a thinner margin would reduce generalizability of the model thus making the test error higher. The number of support vecotors, however, decreases monotonically as C becomes larger. This is because, as explained earlier, large values of C would put a higher penalty on misclassification and hence shrink the margin. As the margin shrinks, less support vectors would be required. This is precisely the pattern we see in Table 2.

Gamma(C)	0.1	1	10	100	1000
Accuracy	0.8755	0.8745	0.8755	0.8690	0.8675
Support Vectors	805	645	594	565	550

Table 2: Accuracy and number of support vectors for different C values.

Problem 2

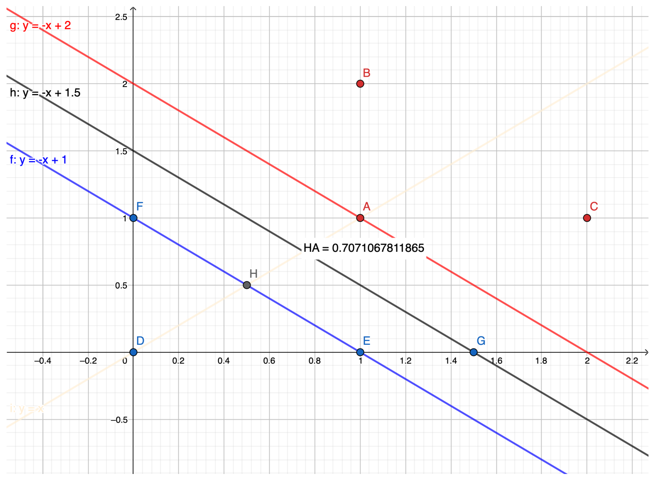

Figure 13: problem 2 solution

As shown in Figure 13, point A(1,1), E(0,1), F(0,1) as support vector. The boundary line of cluster(-1) is the line go through point (0,1) and (1,0).

$$k = \frac{0-1}{1-0} = -1 \tag{1}$$

The equation of the (-1) is :

$$f : y = -x + 1 \tag{2}$$

The boundary of (+1) is the the line go through point (1,1) and parallel to the boundary of cluster(-1). i.e, the slope is the same. The equation of the (-1) is :

$$g : y = -x + 2 \tag{3}$$

Then the optimal separating line is in the middle of line g and line f, and parallel to both.

$$h : y = -x + 1.5 \tag{4}$$

The distance between line g and line f d =0.707. So the margin is $\frac{d}{2} = \frac{\sqrt{2}}{4} =$ 0.3535

Problem 3

Un-normalized version

In this problem, different SVM models are attempted on the Pima Indians Diabetes data set. The results in this part include the un-normalized version of the data set. The results for the normalized case will follow this section.

The results for different choice of kernel and hyper-parameters (γ and C) are illustrated in the following tables. First the data is randomly sampled to form a training set (40% of data), validation set (30%), and a test set (30%). The SVM is trained on the training data and tested on the validation set to tune the hyper-parameters. After finding best hyper-parameters, the test set accuracy is reported.

	Linear	RBF	Sigmoid
$\gamma = 0.1$	0.7521	0.6086	0.6086
$\gamma = 1$	0.7521	0.6086	0.6086
$\gamma = 10$	0.7521	0.6086	0.6086
$\gamma = 100$	0.7521	0.6086	0.6086

Table 3: Accuracy results for $C = 0.1$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	0.7652	0.6086	0.6086
$\gamma = 1$	0.7652	0.6086	0.6086
$\gamma = 10$	0.7652	0.6086	0.6086
$\gamma = 100$	0.7652	0.6086	0.6086

Table 4: Accuracy results for $C = 1$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	0.7826	0.6086	0.6086
$\gamma = 1$	0.7826	0.6086	0.6086
$\gamma = 10$	0.7826	0.6086	0.6086
$\gamma = 100$	0.7826	0.6086	0.6086

Table 5: Accuracy results for $C = 10$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	0.7782	0.6086	0.6086
$\gamma = 1$	0.7782	0.6086	0.6086
$\gamma = 10$	0.7782	0.6086	0.6086
$\gamma = 100$	0.7782	0.6086	0.6086

Table 6: Accuracy results for $C = 100$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	156	308	196
$\gamma = 1$	156	308	196
$\gamma = 10$	156	308	196
$\gamma = 100$	156	308	196

Table 7: Number of support vectors for $C = 0.1$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	154	308	196
$\gamma = 1$	154	308	196
$\gamma = 10$	154	308	196
$\gamma = 100$	154	308	196

Table 8: Number of support vectors for $C = 1$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	153	308	196
$\gamma = 1$	153	308	196
$\gamma = 10$	153	308	196
$\gamma = 100$	153	308	196

Table 9: Number of support vectors for $C = 10$ with different choice of kernels and gamma

	Linear	RBF	Sigmoid
$\gamma = 0.1$	138	308	196
$\gamma = 1$	138	308	196
$\gamma = 10$	138	308	196
$\gamma = 100$	138	308	196

Table 10: Number of support vectors for $C = 100$ with different choice of kernels and gamma

In Tables 3 to 6, the validation set accuracy is reported. We can see that for all values of C and γ, the linear kernel performs consistently better than the other two kernels. Fixing C, we can see that the choice of γ does not seem to have an impact on accuracy rates. This is due to the high-dimensionality of the hypothesis space and the relative sparsity of the data. So, it is expected that changing γ would not be sufficient to compensate for the sparsity of the data. Furthermore, we can see that the performance of RBF and Sigmoid kernels does not change as we alter C and γ. This may suggest that the underlying process is more likely to be linearly separable and given the number of features (high-dimensionality), the linear kernel is able to better generalize and produce better validation set accuracy.

Tables 7 to 10 illustrate the number of support vectors for different choices of C and γ. We can see that the linear kernel has fewer support vectors than RBF and Sigmoid kernels. Furthermore, consistent with the results in Problem 1, choosing larger C's results in a decrease in the number of support vectors for the linear kernel.

Based on the accuracy results on the validation set, we can see that a choice of linear kernel along with values of 10 and 100 for C yields the best performance. Therefore, in the next step, the training and validation sets are merged and an SVM with linear kernel is trained on the merged training set. Table 16 summarizes the accuracy results for the test case. We can see that the test set accuracy for the case of $C = 10$ is the higher and next is the case of $C = 100$ which is pretty much consistent with the validation set accuracy for $C = 10$ and 100.

C	Accuracy
10	0.7826
100	0.7739

Table 11: Test set accuracy for different choices of C with linear kernel and $\gamma = 1$

Normalized version

For this part, we used the same logic as we did for the unnormalized data. First testing among different combinations of kernels, γ and C values, which works the best for the validation set. Then we returned the validation set back to training set and to generate the final hypothesis. Finally evaluate the hypothesis on the test set.

By using normalized data, the time spent on the whole process was much shorter and it is easier to check the performance of polynomial kernel function. For polynomial kernel we used a low degree value 2.

The accuracy on the validation set is reported below:

	Linear	Poly	RBF	Sigmoid
$\gamma = 0.1$	0.6537	0.6537	0.6537	0.6537
$\gamma = 1$	0.6537	0.6537	0.6537	0.6537
$\gamma = 10$	0.6537	0.6580	0.6537	0.6537
$\gamma = 100$	0.6537	0.6926	0.6537	0.6537

Table 12: Accuracy results for $C = 0.1$ with different choice of kernels and gamma

	Linear	Poly	RBF	Sigmoid
$\gamma = 0.1$	0.6494	0.6537	0.6537	0.6537
$\gamma = 1$	0.6494	0.6494	0.6494	0.6494
$\gamma = 10$	0.6494	0.6926	0.6710	0.6537
$\gamma = 100$	0.6494	0.7013	0.6970	0.6537

Table 13: Accuracy results for $C = 1$ with different choice of kernels and gamma

	Linear	Poly	RBF	Sigmoid
$\gamma = 0.1$	0.6494	0.6537	0.6494	0.6494
$\gamma = 1$	0.6494	0.6580	0.6537	0.6580
$\gamma = 10$	0.6494	0.6926	0.6883	0.6537
$\gamma = 100$	0.6494	0.7013	0.6840	0.6537

Table 14: Accuracy results for $C = 10$ with different choice of kernels and gamma

	Linear	Poly	RBF	Sigmoid
$\gamma = 0.1$	0.6537	0.6494	0.6494	0.6494
$\gamma = 1$	0.6537	0.6926	0.6970	0.5411
$\gamma = 10$	0.6537	0.7013	0.6926	0.6537
$\gamma = 100$	0.6537	0.7056	0.6623	0.6537

Table 15: Accuracy results for $C = 100$ with different choice of kernels and gamma

Based on the accuracy results on the validation set, we can see that a choice of second degree polynomial kernel along with values of 10 and 100 for C, 100 for γ yields the best performance. Then after returning the validation set back to the training set and train on these parameters again. We finally got the accuracy on the test set showed below:

C	Accuracy
10	0.6407
100	0.6494

Table 16: Test set accuracy for different choices of C with second degree polynomial kernel and $\gamma = 100$

Problem 4

3-means clustering algorithm:
Step 1 : Pick 3 start means, μ_1, μ_2 and μ_3
Step 2 : Repeat until convergence:
a. split data into 3 sets, S_1, S_2 and S_3 s.t. $x_i \in S_k$ iff μ_k closest meant to x_i
b.update each μ_k to mean of S_k

0.1 Initial means are 1, 3 and 6

Given data set :1,3,6,10,15,21,28,36, K=3
$\mu_1 = 1$, $\mu_2 = 3$, $\mu_3 = 6$.
Initial set:
K_1 = [1]
K_2 =[3]
K_3 =[6,10,15,21,28,36] mean of each cluster:
$\mu_1 = 1$, $\mu_2 = 3$, $\mu_3 = 19.3$.

Point	Distance to $\mu_1 = 1$ Cluster 1	Distance to $\mu_2 = 3$ Cluster 2	Distance to $\mu_3 = 19.3$ Cluster 3	Least distance decision
1	0	2	5	cluster 1
3	2	0	3	cluster 2
6	5	3	0	cluster 3
10	9	7	4	cluster 3
15	14	12	9	cluster 3
21	20	18	15	cluster 3
28	27	25	22	cluster 3
36	35	33	30	cluster 3

First loop:

Point	Distance to $\mu_1 = 1$ Cluster 1	Distance to $\mu_2 = 3$ Cluster 2	Distance to $\mu_3 = 19.3$ Cluster 3	Least distance decision
1	0	2	18.3	cluster 1
3	2	0	16.3	cluster 2
6	5	3	13.3	cluster 2
10	9	7	9.3	cluster 2
15	14	12	4.3	cluster 3
21	20	18	1.7	cluster 3
28	27	25	8.7	cluster 3
36	35	33	16.7	cluster 3

update data set:
K_1 =[1]
K_2 =[3,6,10]
K_3 =[15,21,28,36]
$\mu_1 = 1$, $\mu_2 = 6.3$, $\mu_3 = 25$.

Second loop:

Point	Distance to $\mu_1 = 1$ Cluster 1	Distance to $\mu_2 = 6.3$ Cluster 2	Distance to $\mu_3 = 25$ Cluster 3	Least distance decision
1	0	5.3	24	cluster 1
3	2	3.3	22	cluster 1
6	5	0.3	19	cluster 2
10	9	3.7	15	cluster 2
15	14	8.7	10	cluster 2
21	20	14.7	4	cluster 3
28	27	21.7	3	cluster 3
36	35	29.7	11	cluster 3

update data set:
K_1 = [1,3]
K_2 = [6,10,15]
K_3 = [21,28,36]
$\mu_1 = 2$, $\mu_2 = 10.3$, $\mu_3 = 28.3$.

Third loop:

Point	Distance to $\mu_1 = 2$ Cluster 1	Distance to $\mu_2 = 10.3$ Cluster 2	Distance to $\mu_3 = 28.3$ Cluster 3	Least distance decision
1	1	9.3	27.3	cluster 1
3	1	7.3	25.3	cluster 1
6	4	4.3	22.3	cluster 1
10	8	0.3	18.3	cluster 2
15	13	4.7	13.3	cluster 2
21	19	10.7	7.3	cluster 3
28	26	17.7	0.3	cluster 3
36	34	25.7	7.7	cluster 3

update data set:
K_1 = [1,3,6]
K_2 = [10,15]
K_3 = [21,28,36]
$\mu_1 = 3.3$, $\mu_2 = 12.5$, $\mu_3 = 28.3$.

Forth loop:

Point	Distance to $\mu_1 = 3.3$ Cluster 1	Distance to $\mu_2 = 12.5$ Cluster 2	Distance to $\mu_3 = 28.3$ Cluster 3	Least distance decision
1	2.3	11.5	27.3	cluster 1
3	0.3	9.5	25.3	cluster 1
6	2.7	6.5	22.3	cluster 1
10	6.7	2.5	18.3	cluster 2
15	11.7	2.5	13.3	cluster 2
21	17.7	8.5	7.3	cluster 3
28	24.7	15.5	0.3	cluster 3
36	32.7	23.5	7.7	cluster 3

The mean will not change. So it is converged. The final results are K_1 = [1,3,6]
, K_2 = [10,15] and K_3 = [21,28,36].

0.2 Initial means are 21,28, and 36

Given data set :1,3,6,10,15,21,28,36, K=3
$\mu_1 = 21$, $\mu_2 = 28$, $\mu_3 = 36$.
K_1 = [1,3,6,10,15,21]
K_2 =[28]
K_3 =[36]
Initial set:

Point	Distance to $\mu_1 = 21$ Cluster 1	Distance to $\mu_2 = 28$ Cluster 2	Distance to $\mu_3 = 36$ Cluster 3	Least distance decision
1	20	27	35	cluster 1
3	18	25	33	cluster 1
6	15	22	30	cluster 1
10	11	18	26	cluster 1
15	6	13	21	cluster 1
21	0	7	15	cluster 1
28	7	0	8	cluster 2
36	15	8	0	cluster 3

$\mu_1 = 9.3$, $\mu_2 = 28$, $\mu_3 = 36$.

First loop:

Point	Distance to $\mu_1 = 9.3$ Cluster 1	Distance to $\mu_2 = 28$ Cluster 2	Distance to $\mu_3 = 36$ Cluster 3	Least distance decision
1	8.3	27	35	cluster 1
3	6.3	25	33	cluster 1
6	3.3	22	30	cluster 1
10	0.7	18	26	cluster 1
15	5.7	13	21	cluster 1
21	11.7	7	15	cluster 2
28	18.7	0	8	cluster 2
36	26.7	8	0	cluster 3

update data set:
K_1 = [1,3,6,10,15]
K_2 =[21,28]
K_3 =[36]
$\mu_1 = 7$, $\mu_2 = 24.5$, $\mu_3 = 36$.

Second loop:

Point	Distance to $\mu_1 = 7$ Cluster 1	Distance to $\mu_2 = 24.5$ Cluster 2	Distance to $\mu_3 = 36$ Cluster 3	Least distance decision
1	6	23.5	35	cluster 1
3	4	21.5	33	cluster 1
6	1	18.5	30	cluster 1
10	3	14.5	26	cluster 1
15	8	9.5	21	cluster 1
21	14	3.5	15	cluster 2
28	21	3.5	8	cluster 2
36	29	11.5	0	cluster 3

update data set:
K_1 = [1,3,6,10,15]
K_2 =[21,28]
K_3 =[36]
$\mu_1 = 7$, $\mu_2 = 24.5$, $\mu_3 = 36$.

The mean will not change. So it is converged.
The final results are K_1 = [1,3,6,10,15] , K_2 = [21,28] and K_3 = [36].

Appendix

Problem 1

Plots of boundary with margin and support vectors.
Part (a).

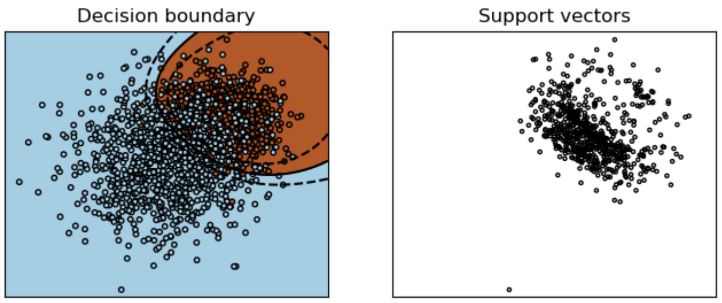

Figure 14: Decision boundary for $\gamma = 0.1$

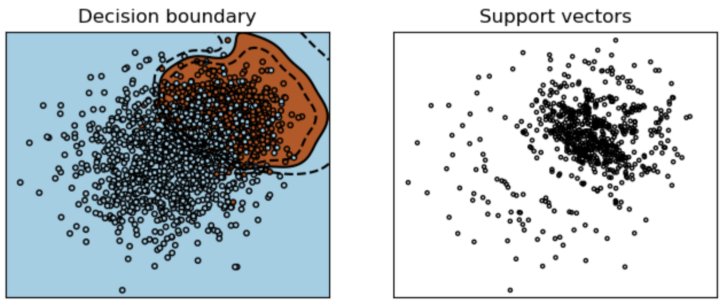

Figure 15: Decision boundary for $\gamma = 1$

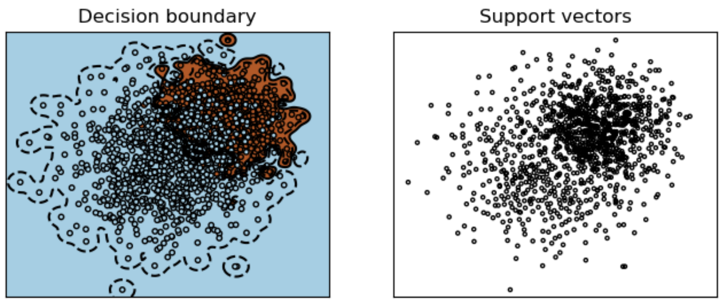

Figure 16: Decision boundary for $\gamma = 10$

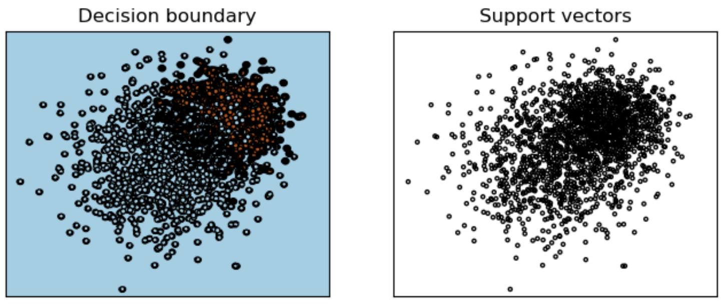

Figure 17: Decision boundary for $\gamma = 100$

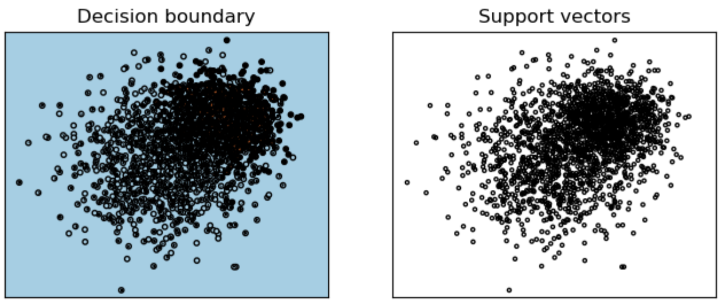

Figure 18: Decision boundary for $\gamma = 1000$

Part (b).

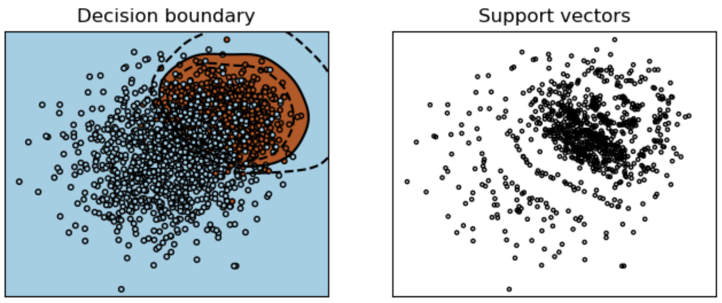

Figure 19: Decision boundary for $C = 0.1$

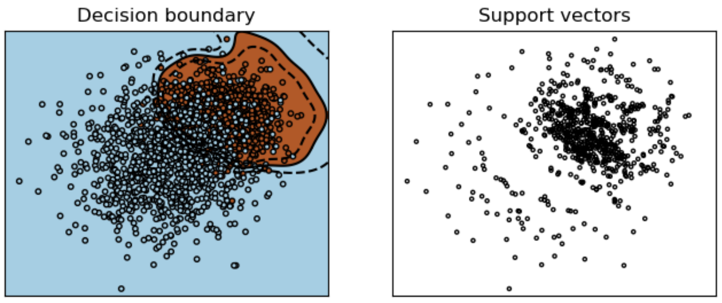

Figure 20: Decision boundary for $C = 1$

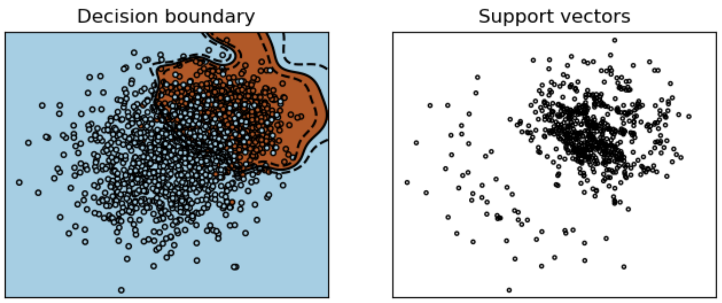

Figure 21: Decision boundary for $C = 10$

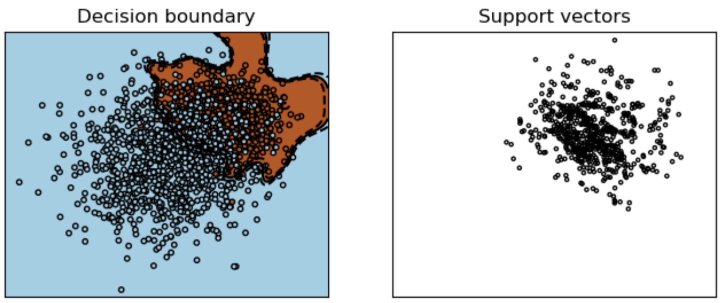

Figure 22: Decision boundary for $C = 100$

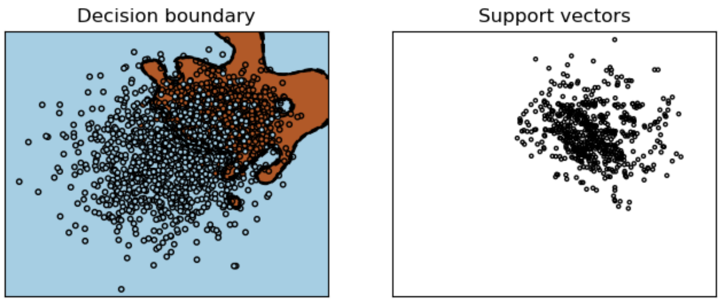

Figure 23: Decision boundary for $C = 1000$