Advanced Numerical Analysis

Final Project

Submitted By

Md Mahin (PSID: 1900421)
Shanto Roy (PSID: 1894941)

Department of Computer Science
University of Houston

May, 2020

Problem 1

In binary classification, we are interested in finding a hyperplane that separates two clouds of points living in, say, $\mathbb{R}^p$. The support vector machine (SVM), which we talked about in class, is a pretty popular method for doing binary classification; to this day, it's (still) used in a number of fields outside of just machine learning and statistics.

One issue arises with the standard SVM, though, when the data points are not linearly separable in $\mathbb{R}^p$, i.e., we cannot find a hyperplane which separates the two classes of points. In such cases, it is often useful to map the data points to a different space (potentially of higher dimension than $\mathbb{R}^p$) where the points become separable. Such maps are called nonlinear feature maps.

In this problem, you will develop a SVM with the RBF kernel to address the nonlinearly separable problem of the standard SVM. You will implement your own RBF-SVM in part (b) of this question, but as a starting point, we will first investigate the SVM dual problem in part (a) of this question.

Throughout, we assume that we are given n data samples, each one taking the form (x_i, y_i), where $x_i \in \mathbb{R}^p$ is a feature vector and $y_i \in \{-1, +1\}$ is a class. In order to make our notation more concise, we can transpose and stack the x_i vertically, collecting these feature vectors into the matrix $X \in \mathbb{R}^{n\times p}$; doing the same thing with the y_i lets us write $y \in \{-1, +1\}^n$.

Part (a)

The primal problem of SVM with slack variables is

$$\begin{array}{ll} \underset{\beta\in\mathbb{R}^p,\ \beta_0\in\mathbb{R},\ \xi\in\mathbb{R}^n}{\text{minimize}} & \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to} & \xi_i \geqslant 0, \quad i = 1, \ldots, n, \\ & y_i(x_i^T\beta + \beta_0) \geqslant 1 - \xi_i, \quad i = 1, \ldots, n, \end{array} \tag{1}$$

where $\beta \in \mathbb{R}^p$, $\beta_0 \in \mathbb{R}$, $\xi = (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$ are our variables, and C is a positive margin coefficient chosen by the implementer. (Just to remind you of some of the intuition here: problem (1) can be viewed as another way of writing a squared ℓ_2-norm penalized hinge loss minimization problem.)

(i) Does strong duality hold for problem (1)? Why or why not? (Your answer to the latter question should be short.)

(ii) Derive the Karush-Kuhn-Tucker (KKT) conditions for problem (1). Please use $\alpha \in \mathbb{R}^n$ for the dual variables (i.e., Lagrange multipliers) associated with the constraints "$y_i(x_i^T\beta+\beta_0) \geqslant 1-\xi_i,\ i = 1, \ldots, n$", and $\mu \in \mathbb{R}^n$ for the dual variables associated with the constraints "$\xi_i \geqslant 0,\ i = 1, \ldots, n$".

(iii) Show that the SVM dual problem can be written as

$$\begin{array}{ll} \underset{\alpha\in\mathbb{R}^n}{\text{maximize}} & -(1/2)\alpha^T\tilde{X}\tilde{X}^T\alpha + 1^T\alpha \\ \text{subject to} & \alpha^T y = 0, \\ & 0 \leqslant \alpha \leqslant C1, \end{array} \tag{2}$$

where $\tilde{X} \in \mathbb{R}^{n\times p} = \text{diag}(y)X$, α is the dual variable, and the 1's here are vectors (of the appropriate and possibly different sizes) containing only ones.

(iv) Give an expression for the optimal β in terms of the optimal α variables and explain how.

Solution:

(i) Does strong duality hold for problem? Why or why not? (Your answer to the latter question should be short.)

Answer: Yes, for the equation (3), there exists a strong duality.
We can rewrite the inequality constraints of primal problem as:

$$\begin{array}{ll} \underset{\beta\in\mathbb{R}^p,\ \beta_0\in\mathbb{R},\ \xi\in\mathbb{R}^n}{\text{minimize}} & \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to} & -\xi_i \leqslant 0, \quad i = 1,\ldots,n, \\ & 1 - \xi_i - y_i(x_i^T\beta + \beta_0) \leqslant 0, \quad i = 1,\ldots,n, \end{array} \tag{3}$$

Here the inequality constraints are at the same time affine in nature where the line go through the support vectors.
From slater's condition to be strong duality, we know:

$$h_1(x) < 0 - - - - - h_m(x) < 0$$
$$l_1(x) = 0 - - - l_r(x) = 0$$

However, if there is no non-linear condition, the condition $h_1(x) < 0 - - - - - h_m(x) < 0$ can be relaxed. In case of support vector machine, the condition is linear and affine and we have support vectors on the hyperplanes. So, it automatically satisfy slater's condition.
So, we can say that the strong duality exists.

(ii) Derive the Karush-Kuhn-Tucker (KKT) conditions for problem (1). Please use $\alpha \in \mathbb{R}^n$ for the dual variables (i.e., Lagrange multipliers) associated with the constraints "$y_i(x_i^T\beta + \beta_0) \geqslant 1 - \xi_i,\ i = 1,\ldots,n$", and $\mu \in \mathbb{R}^n$ for the dual variables associated with the constraints "$\xi_i \geqslant 0,\ i = 1,\ldots,n$".

Answer:

Introducing dual variable $\alpha \in R^n$ and $\mu \in R^n$ the Lagrangian function is -

$L(\beta, \beta_0, \xi, \mu, \alpha) = \frac{1}{2}||\beta||_2^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \mu_i\xi_i + \sum_{i=1}^n \alpha_i(1 - \xi - y_i(x_i^T\beta + \beta_0))$
To derive KKT conditions, we are deriving by four conditions:

Stationarity:
Here we need to prove $0 \in \partial L(\beta, \beta_0, \xi, \mu, \alpha)$
For three variable β, β_0, μ we have to derive them separately.
So, for β
$\nabla\beta L(\beta, \beta_0, \xi, \mu, \alpha) = 0$
or, $\beta - \sum_{i=1}^n \alpha_i y_i x_i = 0$
or, $\beta = \sum_{i=1}^n \alpha_i y_i x_i$

for β_0,
$\nabla\beta_0 L(\beta, \beta_0, \xi, \mu, \alpha) = 0$
or,$-\sum_{i=1}^n \alpha_i y_i = 0$

For, ξ
$\nabla\xi L = 0$
or, $C.1 - \mu + \alpha = 0$
or, $\alpha = C.1 - \mu; i = 1, .., n$

Complementary Slackness:
For complementary slackness we need two conditions to satisfy based on two inequality constraints,
$\mu_i\xi_i = 0, i = 1, 2, ..., n$
$\alpha_i(1 - \xi - y_i(x_i^T\beta + \beta_0)) = 0, i = 1, 2, ..., n$

Primal Feasibility:
$-\xi_i \leqslant 0, \quad i = 1, \ldots, n,$
$1 - \xi_i - y_i(x_i^T \beta + \beta_0) \leqslant 0, \quad i = 1, \ldots, n$

Dual Feasibility:
$\alpha_i \geqslant 0, \mu_i \geqslant 0, i = 1, 2....n$

(iii) Show that the SVM dual problem can be written as

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} & -(1/2)\alpha^T \tilde{X}\tilde{X}^T \alpha + 1^T \alpha \\ \text{subject to} & \alpha^T y = 0, \\ & 0 \leqslant \alpha \leqslant C1, \end{array}$$

where $\tilde{X} \in \mathbb{R}^{n \times p} = \text{diag}(y)X$, α is the dual variable, and the 1's here are vectors (of the appropriate and possibly different sizes) containing only ones.

Answer:
From part 2, for dual dual variable $\alpha \in R^n$ and $\mu \in R^n$ the Lagrangian function is -

$L(\beta, \beta_0, \xi, \mu, \alpha) = \frac{1}{2}||\beta||_2^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \mu_i \xi_i + \sum_{i=1}^n \alpha_i(1 - \xi - y_i(x_i^T \beta + \beta_0))$
and,

$\beta = \sum_{i=1}^n \alpha_i y_i x_i;$

$-\sum_{i=1}^n \alpha_i y_i = 0;$

$C.1 - \mu + \alpha = 0; i = 1, .., n;$

So Lagrangian Dual Problem,

$g(\mu, \alpha) = L(\beta*, \beta_0^*, \xi*, \mu, \alpha)$

or,$= \frac{1}{2}||\beta||_2^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \mu_i \xi_i + \sum_{i=1}^n \alpha_i(1 - \xi - y_i(x_i^T \beta + \beta_0))$

or,$= \frac{1}{2}||\beta||_2^2 + C.1^T \xi - \mu^T \xi + \alpha^T(1 - \xi - y_i(x_i^T \beta + \beta_0))$

or,$= \frac{1}{2}||\beta||_2^2 + (C.1 - \mu + \alpha)^T \xi + \alpha^T(1 - y_i(x_i^T \beta + \beta_0))$

or,$= \frac{1}{2}||\beta||_2^2 - \alpha^T(y_i(x_i^T \beta + \beta_0) - 1)$; using $C.1 - \mu + \alpha = 0; i = 1, .., n;$

or,$= \frac{1}{2}||\beta||_2^2 + \alpha^T$; using $-\sum_{i=1}^n \alpha_i y_i = 0;$

or, $= \frac{1}{2}||\beta||_2^2 + \sum_{i=1}^n \alpha_i$

Now considering, $\beta = \sum_{i=1}^n \alpha_i y_i x_i;$, our duel problem will be,

$max_\alpha \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j}^n \alpha_i \alpha_j y_i y_j x_i x_j$
subject to,

$-\sum_{i=1}^{n} \alpha_i y_i = 0;$

$\alpha = C.1 - \mu; i = 1, .., n;$

Now, if we consider, $\frac{\partial L}{\partial \mu} = 0$
$c.1 - \alpha \geqslant 0$
or, $c.1 \geqslant \alpha$
now considering, $\alpha \geqslant 0$ we have $c.1 \geqslant \alpha \geqslant 0$

again from, $-\sum_{i=1}^{n} \alpha_i y_i = 0$; we can write, $1^T \alpha = 0$
and also, for $\tilde{X} = diag(y)X, where \tilde{X} \in R^{n \times p}$ we can write,

$\sum_{i,j}^{n} y_i y_j x_i x_j = y_i y_j x_i^T x_j = \tilde{X}\tilde{X}^T$

so our final duel form becomes

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} & -(1/2)\alpha^T \tilde{X}\tilde{X}^T \alpha + 1^T \alpha \\ \text{subject to} & \alpha^T y = 0, \\ & 0 \leqslant \alpha \leqslant C1; \end{array}$$

[Showed]

(iv) Give an expression for the optimal β in terms of the optimal α variables and explain how.

Answer: From stationarity of KKT conditions we know:

$\beta* = \sum_{i=1}^{n} y_i \alpha_i^* x_i$

So, according to the strong duality, if primal or dual has an optimal solution it's counter part e.g. dual or primal will also have an optimal solution. This is how we can derive optimal β with optimal α.

Problem 1

Part (b)

Now we are going to take a glimpse of the "magic" of kernels. Let's first see what is a kernel. Given a feature map $\phi : \mathbb{R}^d \to \mathcal{K}$, where $\mathcal{K}$ is a Hilbert space (i.e., a vector space with inner product), the kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the corresponding inner product function

$$K(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle. \tag{4}$$

Here the feature map, as we mentioned earlier, is used to "embed" the original data into a higher dimensional space such that they become separable. Recall the objective of the dual SVM, and it can be rewritten as

$$\begin{aligned} & -\tfrac{1}{2}\alpha^T \tilde{X}\tilde{X}^T \alpha + 1^T \alpha && (5)\\ \Leftrightarrow & -\tfrac{1}{2}\alpha^T Y X X^T Y \alpha + 1^T \alpha && (6)\\ \Leftrightarrow & -\tfrac{1}{2}\alpha^T Y G Y \alpha + 1^T \alpha, && (7)\\ & && (8) \end{aligned}$$

where $Y = \operatorname{diag}(y)$, and $G = XX^T$ is the so called Gram matrix, $G_{ij} = \langle x_i, x_j \rangle$. One nice property of the Gram matrix of a kernel K is that

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = G_{ij}. \tag{9}$$

Hence, the kernel builds a bridge between the feature maps and the original dual problem.

Now we are going to probe into the infinite dimensional space. We have seen so far how to build a kernel from a given feature map, but can we do the opposite? Suppose a map K is a kernel, can we find the corresponding feature map ϕ such that $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{K}}$? Fortunately, thanks to the Mercer's theorem, we know that we are able to construct the feature map by finding the eigenfunctions of the integral operator with the kernel.

There is no need to go into such difficulty of finding the feature maps, however, since we have the kernel-feature map equivalence (9). We only need to compute the value of the kernel function, avoiding the complexity of computing the inner product of high dimensional feature maps.

Given this intuition, we consider the radial basis function (RBF) kernel

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp\left(-\gamma \|x_i - x_j\|^2\right), \tag{10}$$

where γ controls the bandwidth of the kernel. For RBF kernel, the corresponding feature maps have infinite dimensional feature spaces. The RBF kernel is a reasonable measure of x_i and x_j's similarity, and is close to 1 when x_i and x_j are close, and near 0 when they are far apart. In the following problems, you are going to use the RBF kernel in SVM.

The RBF kernel SVM dual problem becomes: (compared to (2))

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} \quad & -(1/2)\alpha^T Y G Y \alpha + 1^T \alpha \\ \text{subject to} \quad & \alpha^T y = 0, \\ & 0 \leqslant \alpha \leqslant C1, \end{aligned} \tag{11}$$

where $G_{ij} = K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$.

(i) Implement the dual SVM in problem (11) with the RBF kernel using a standard QP solver (typically available as "quadprog" function in Matlab, R, or in `Mathprogbase.jl` in Julia; you may also refer `CVXOPT` in Python, `GORUBI`, or `MOSEK`). Load a small synthetic toy problem with inputs $X \in \mathbb{R}^{863\times 2}$ and labels $y \in \{-1, 1\}^{863}$ from `data.txt` and solve the dual SVM with $\gamma = \{10, 50, 100, 500\}$ and $C = \{0.01, 0.1, 0.5, 1\}$. Report the optimal objective values of the dual.

(ii) For each of the parameter pairs, show a scatter plot of the data and plot the decision border (where the predicted class label changes) on top. How and why does the decision boundary change with different pair of parameters?

(iii) For each of the parameter pairs, identify the support vectors (i.e., data points with nonzero α_is; in implementation select $\alpha > 1e^{-5}$) in the plots, and report the number of support vectors. What can in general be said about the location of a data point i with respect of the boundary of the margin if

- $\alpha_i = 0$;
- $\alpha_i \in (0, C)$;
- $\alpha_i = C$?

(iv) Looking back at the KKT conditions derived in part (a, ii), what can be said about the influence of the data points that lie strictly on the correct side of the margin? How would the decision boundary change if we removed these data points from the dataset and recomputed the optimal solution? (Give a qualitative answer, no need to actually implement that.)

Solution:

(i.) In code file "`Q2_dual_svm.m`", the labels are defined as-

```
% Draw Figures
figure
% Write title for the figures
title(['For Parameters- C=',num2str(C_i),' and Gamma=', num2str(Gamma_i)])
% Draw Grids
grid on
hold on
% Scatter Plot Feature 1 (+1) in ‘red‘
scatter(Feature_1(Labels == 1), Feature_2(Labels == 1), 'r', '+')
hold on
% Scatter Plot Feature 2 (-1) in ‘green‘
scatter(Feature_1(Labels == -1), Feature_2(Labels == -1), 'g', '*')
hold on
% Scatter Plot for errors in ‘blue‘
scatter(Feature_1(alpha > Max_error),Feature_2(alpha > Max_error),'b','o')
hold on
```

In our code, we have plotted "+" for all the data points from feature 1 with label 1 and "*" for all the data points from feature 2 with label -1. For the data points with nonzero α is; we have plotted a round boundary "o". The figures show all the support vectors with round boundary. See Figure 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16 for each parameter pair.

The optimal objective values are presented in Table 1.

(ii.) We plot the decision boundaries using contours which are presented in Figure 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, and 32. [note: corresponding values of γ and C is tagged accordingly.]

Observation: From all these figures, we see, with the increasing values for both γ and C, the decision boundary is more clear and obvious. It's because of the Eqn. (11) where we need to maximize the dual problem. We see, as γ controls the bandwidth and with increasing values, it maximizes the Gram Matrix. Also, with increasing value of C, the value of α increases and eventually maximizes the dual problem equation.

(iii.) The number of "`Support Vectors`" are presented in Table 1.

Table 1: Support Vectors for Given Values of C and γ

Figure	**Combination**		**# of Support Vectors**	**Optimal Values**	**Bias**
	γ	**C**			
1	10	0.01	768	-7.130305	0.916488
2	10	0.10	584	-49.480754	0.577321
3	10	0.50	447	-187.561223	0.166146
4	10	1.00	401	-326.342072	-0.242409
5	50	0.01	770	-6.716282	0.890569
6	50	0.10	448	-31.260609	0.495899
7	50	0.50	249	-79.707054	0.434681
8	50	1.00	185	-116.611534	0.177985
9	100	0.01	772	-6.794814	0.867729
10	100	0.10	449	-28.374182	0.348824
11	100	0.50	210	-60.780378	0.840059
12	100	1.00	157	-83.411436	0.673693
13	500	0.01	805	-7.306472	0.882052
14	500	0.10	745	-42.417466	-0.183961
15	500	0.50	347	-64.364836	0.176034
16	500	1.00	281	-73.963265	-1.143208

Observation for α_i :

The location of a data point i with respect of the boundary of the margin,

- IF $\alpha_i = 0$,
  The training example is correctly classified "`above the margin`"
- IF $\alpha_i \in (0, C)$,
  The training example is classified exactly "`at the margin`"
- IF $\alpha_i = C$,
  The training example is "`mis-classified`"

(iv.) From the KKT conditions derived in part (a, ii), we find the following observations:

- Removing the data points that lie strictly on the correct side of the margin, the resultant output produces a "`smaller margin`".
- The smaller margin has a greater percentage of *training points* that are being classified as *support vectors*, which may lead to "`overfitting`".

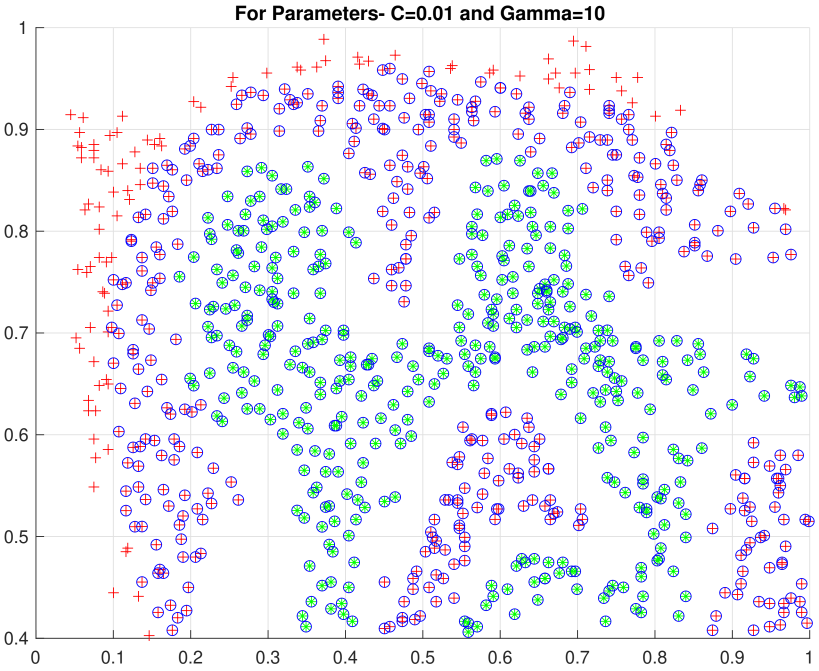

Figure 1: Dual SVM with $\gamma = 10$, and $C = 0.01$

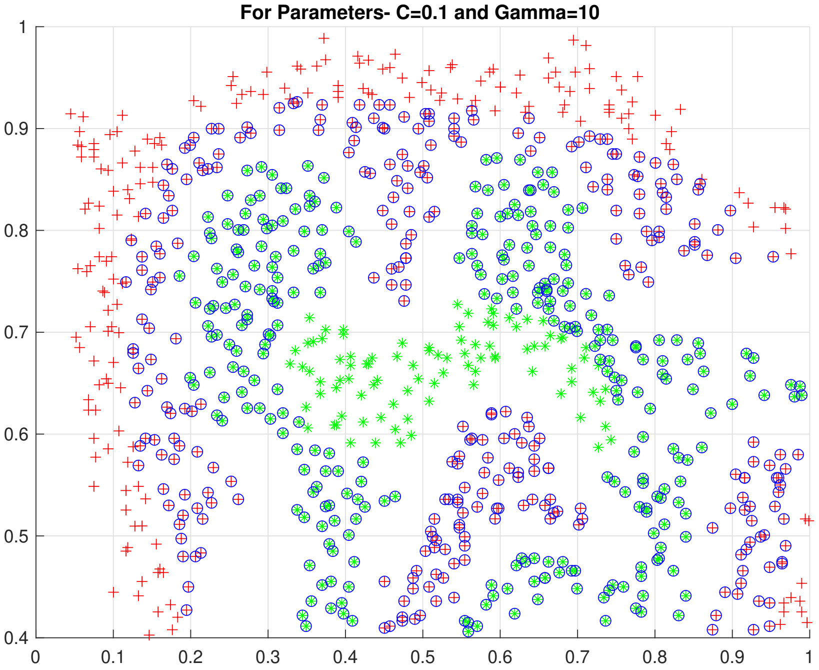

Figure 2: Dual SVM with $\gamma = 10$, and $C = 0.1$

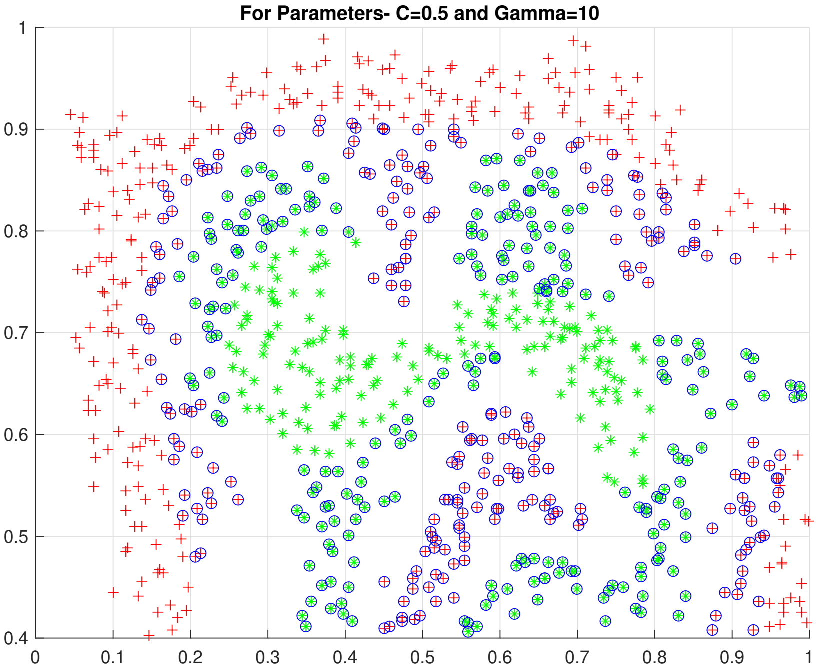

Figure 3: Dual SVM with $\gamma = 10$, and $C = 0.5$

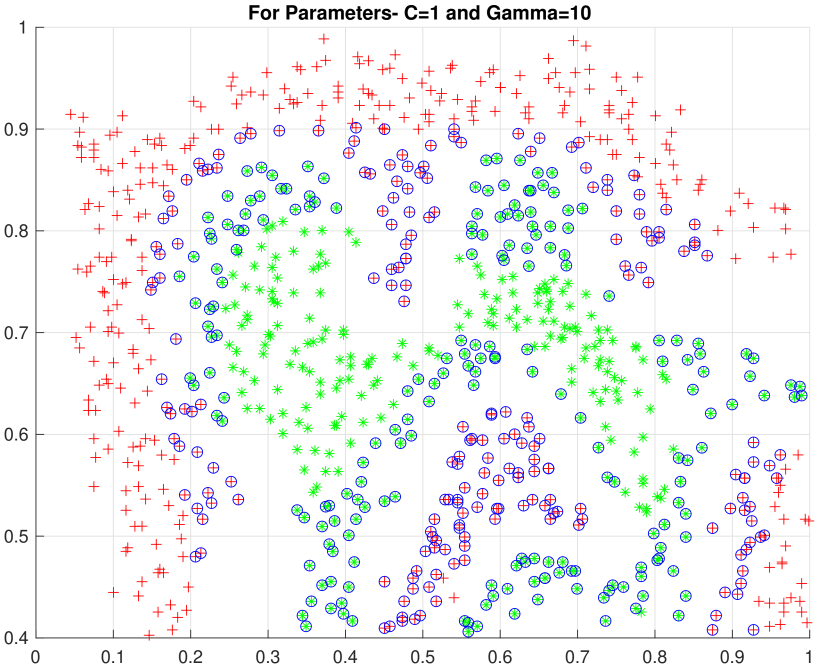

Figure 4: Dual SVM with $\gamma = 10$, and $C = 1$

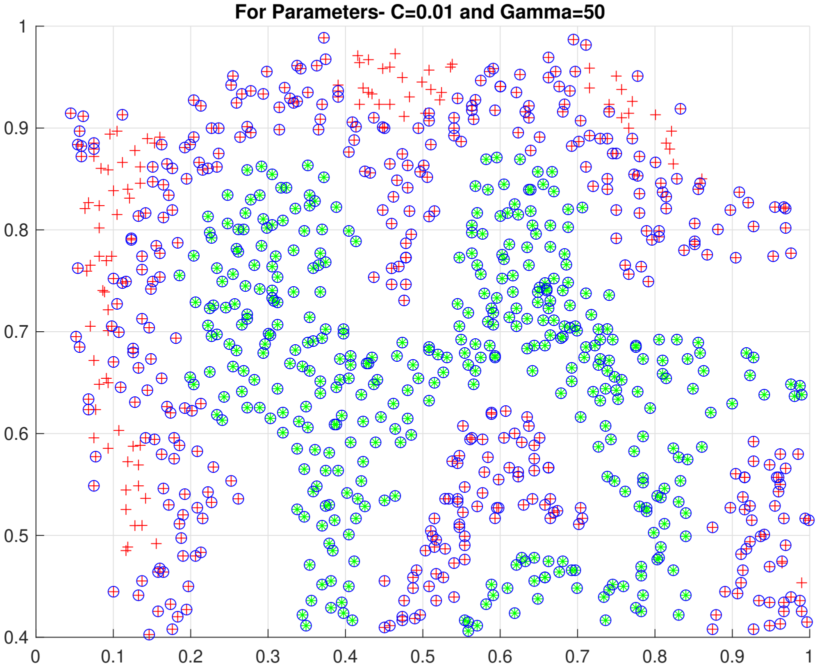

Figure 5: Dual SVM with $\gamma = 50$, and $C = 0.01$

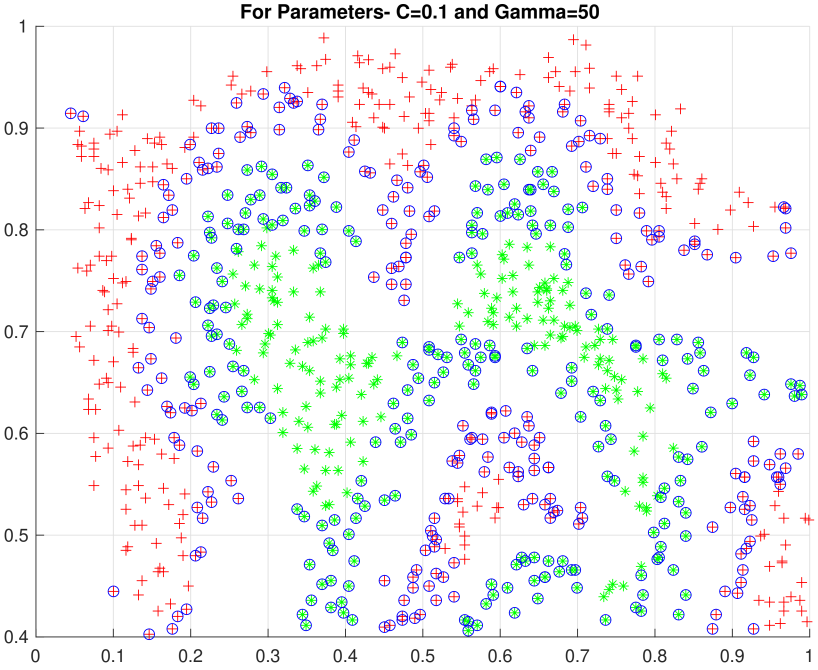

Figure 6: Dual SVM with $\gamma = 50$, and $C = 0.1$

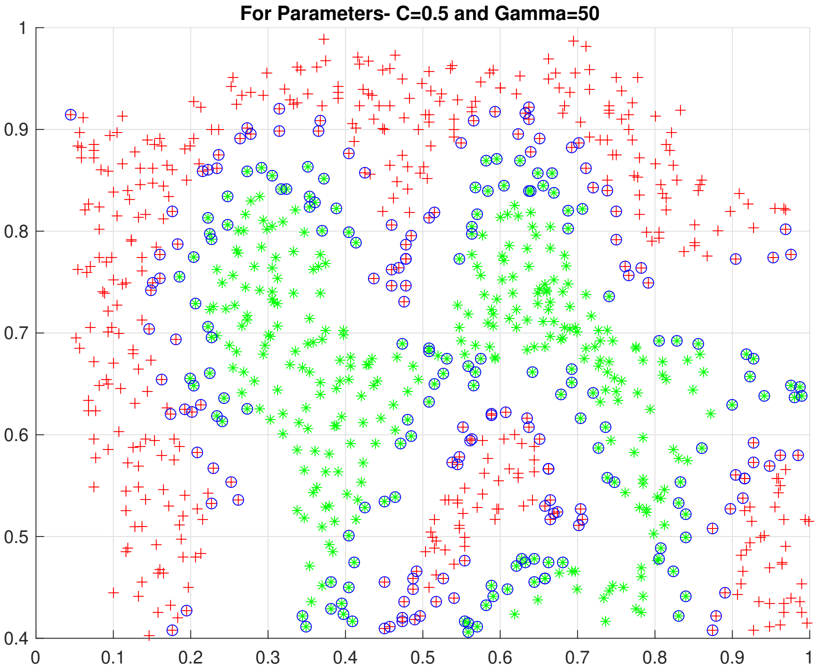

Figure 7: Dual SVM with $\gamma = 50$, and $C = 0.5$

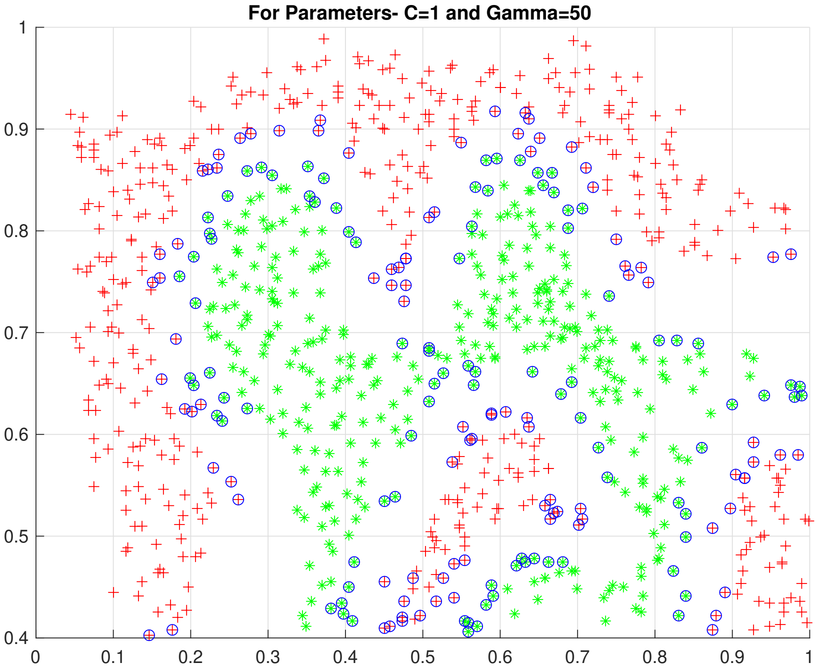

Figure 8: Dual SVM with $\gamma = 50$, and $C = 1$

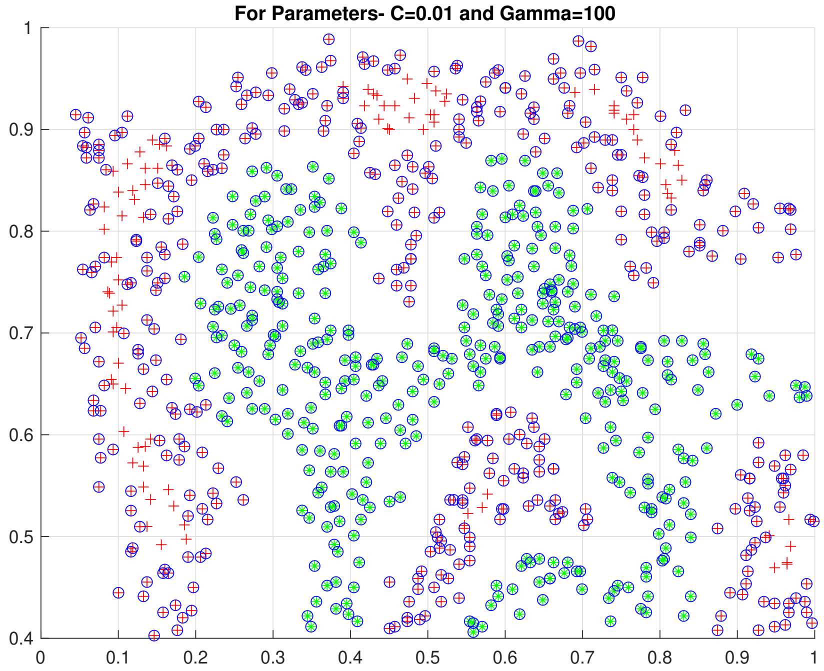

Figure 9: Dual SVM with $\gamma = 100$, and $C = 0.01$

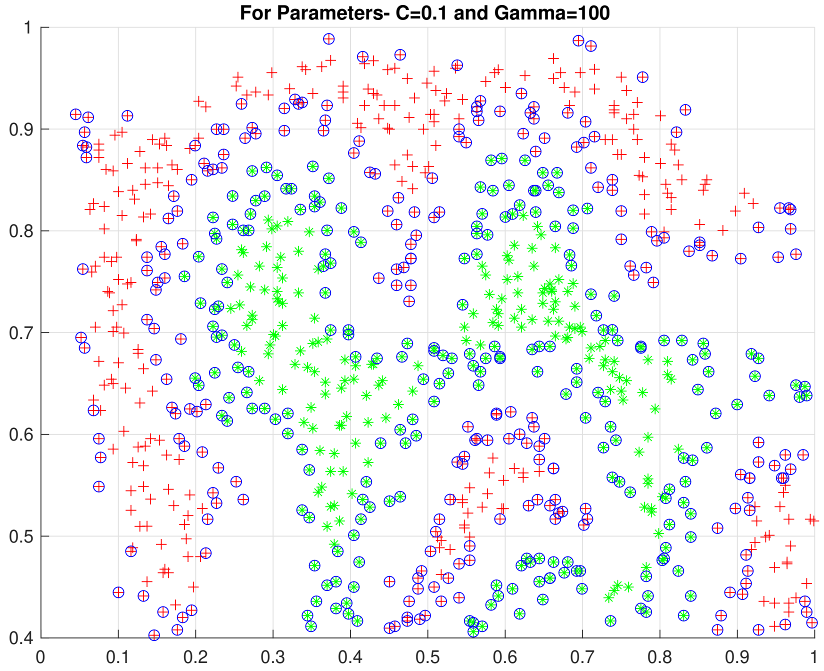

Figure 10: Dual SVM with $\gamma = 100$, and $C = 0.1$

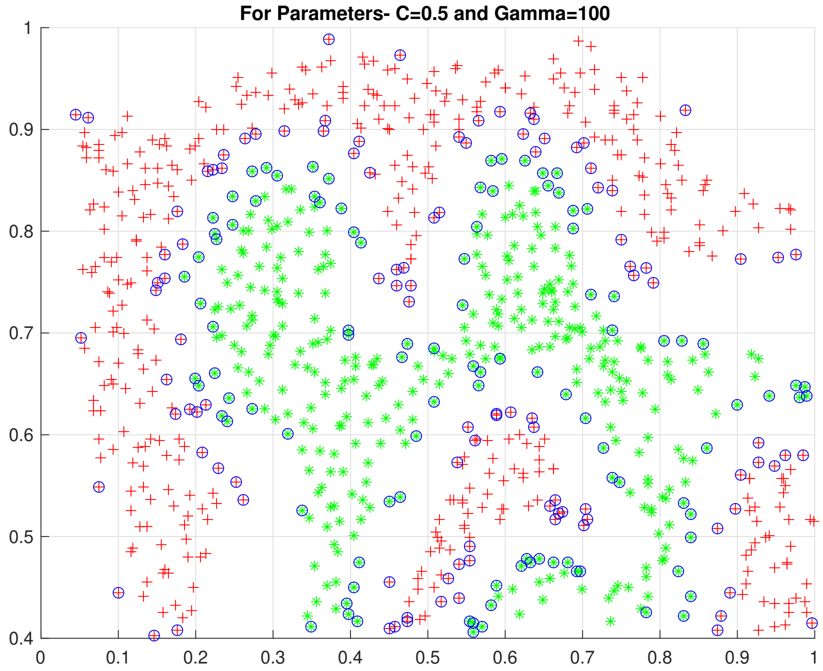

Figure 11: Dual SVM with $\gamma = 100$, and $C = 0.5$

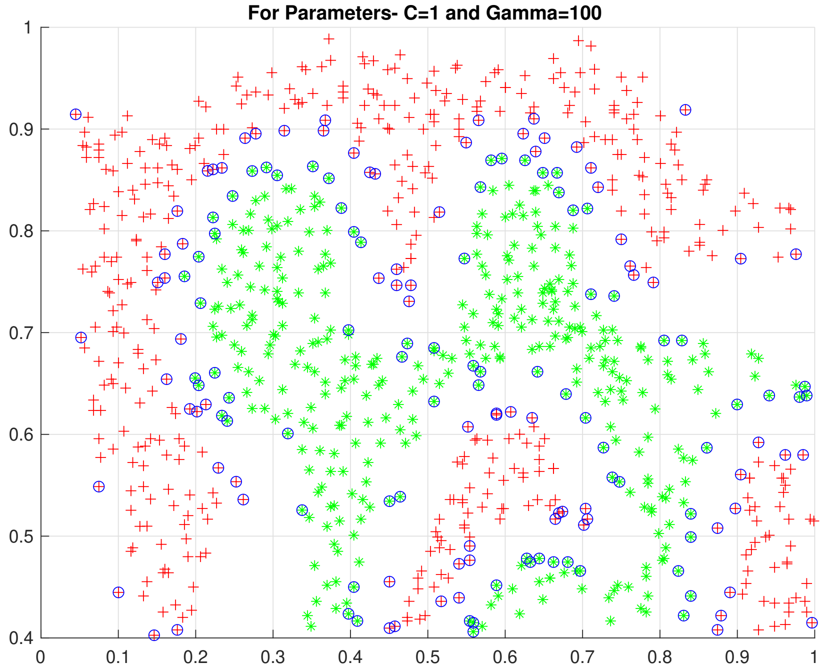

Figure 12: Dual SVM with $\gamma = 100$, and $C = 1$

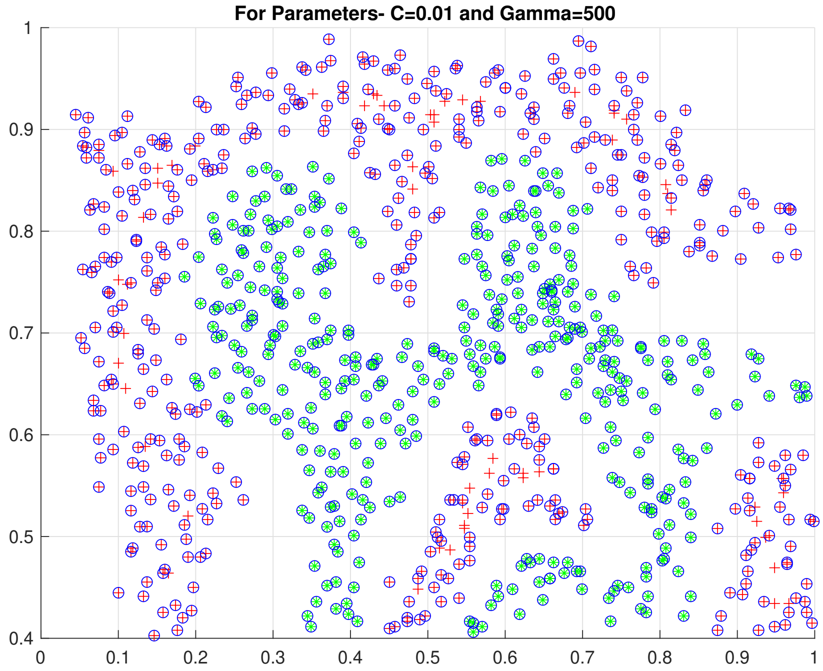

Figure 13: Dual SVM with $\gamma = 500$, and $C = 0.01$

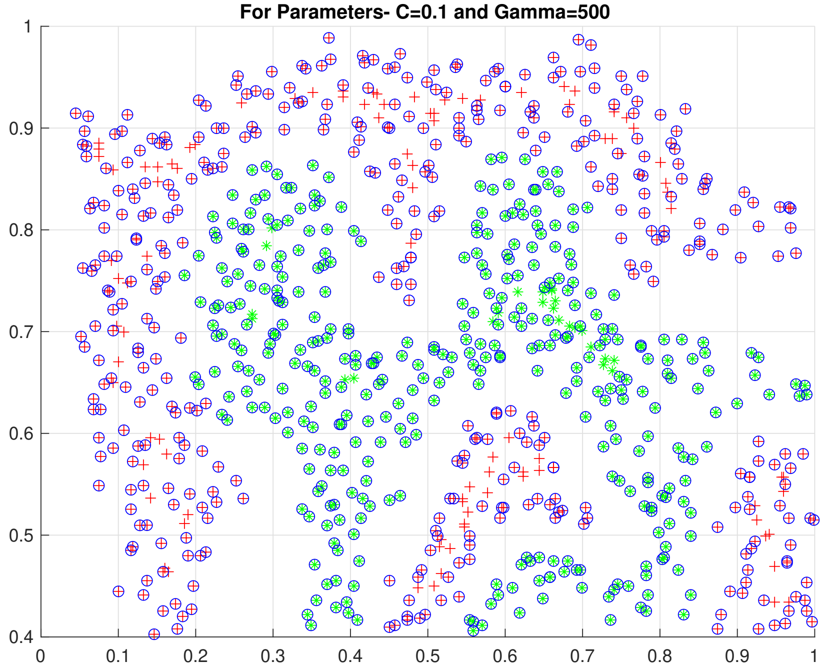

Figure 14: Dual SVM with $\gamma = 500$, and $C = 0.1$

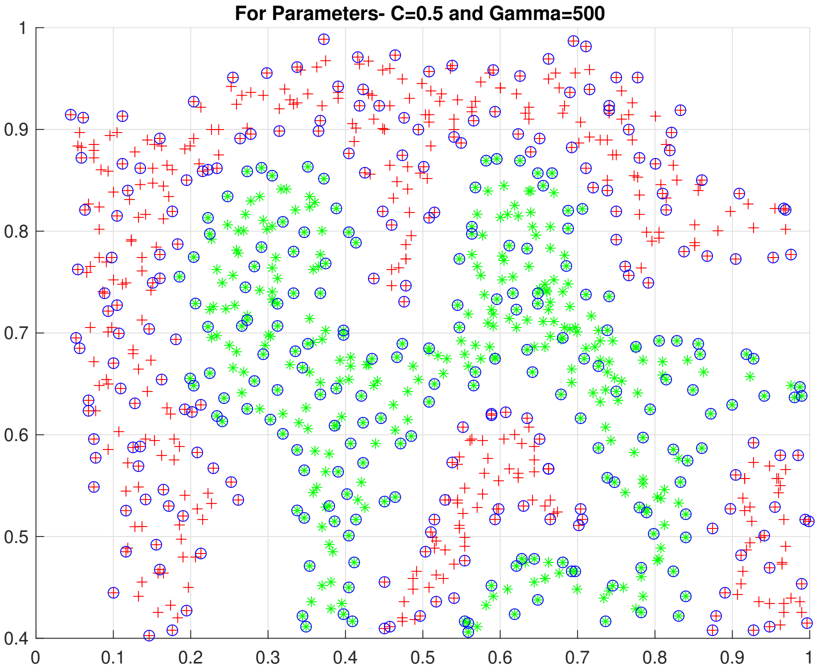

Figure 15: Dual SVM with $\gamma = 500$, and $C = 0.5$

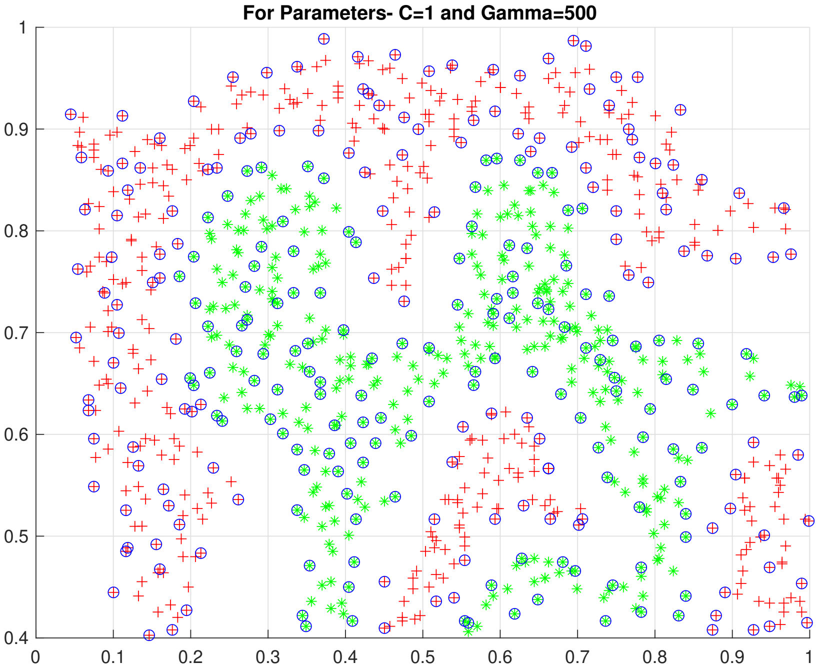

Figure 16: Dual SVM with $\gamma = 500$, and $C = 1$

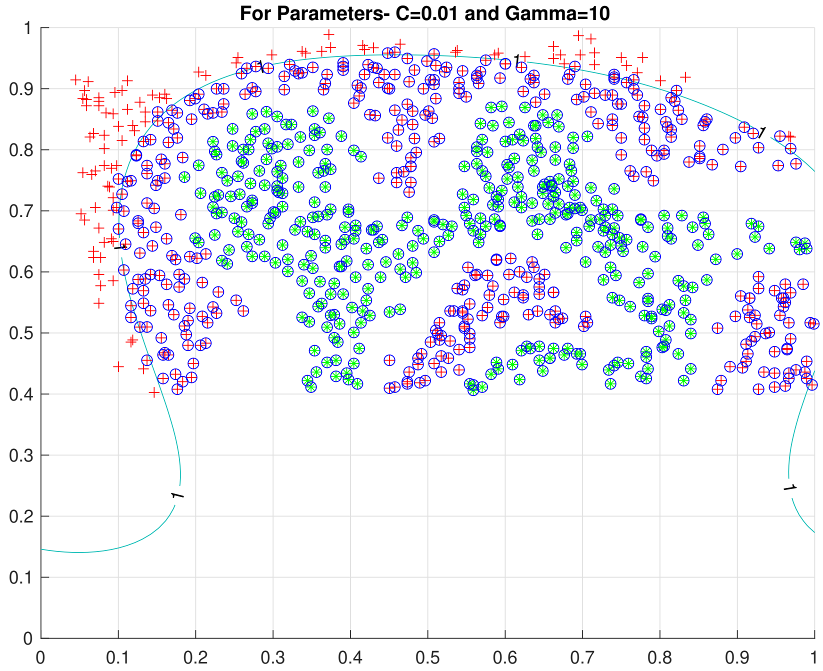

Figure 17: Dual SVM with $\gamma = 10$, and $C = 0.01$

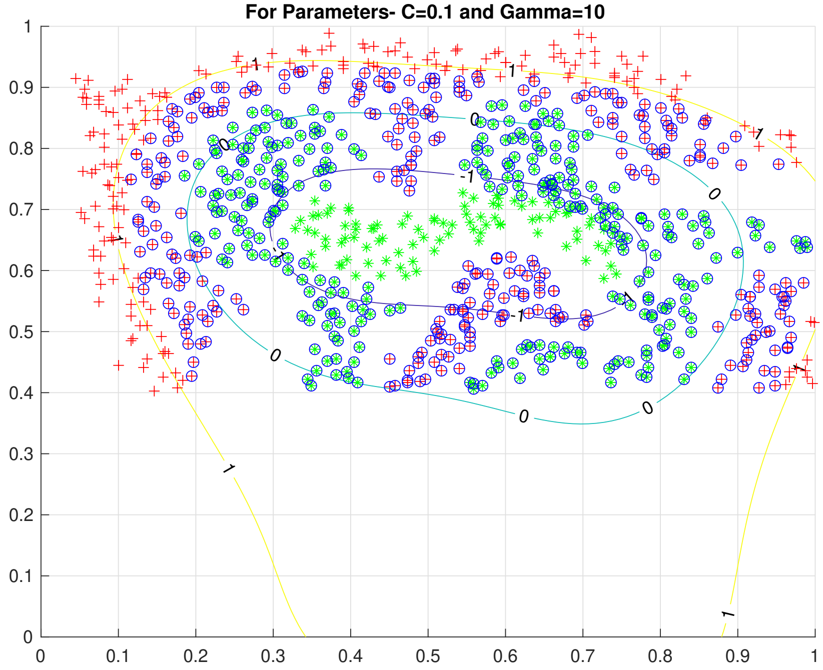

Figure 18: Dual SVM with $\gamma = 10$, and $C = 0.1$

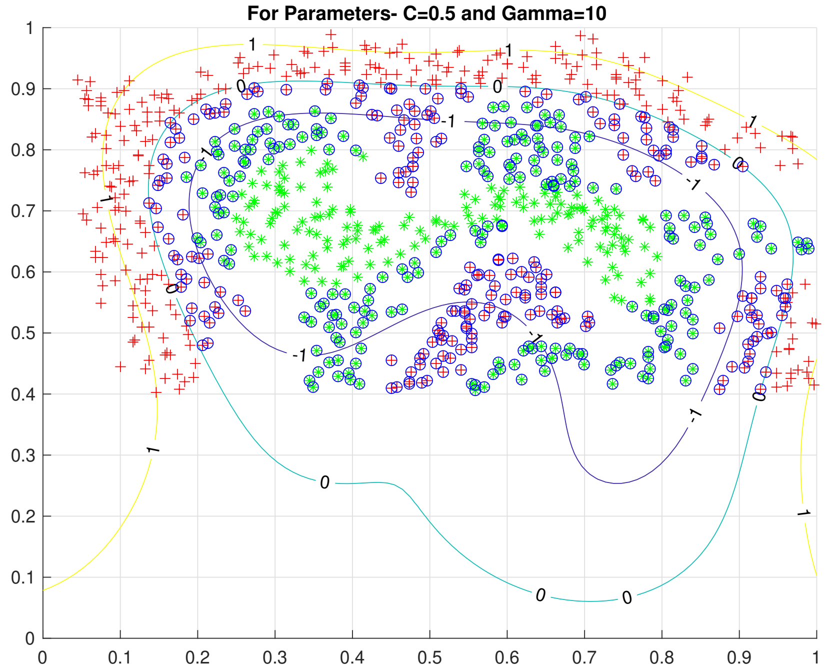

Figure 19: Dual SVM with $\gamma = 10$, and $C = 0.5$

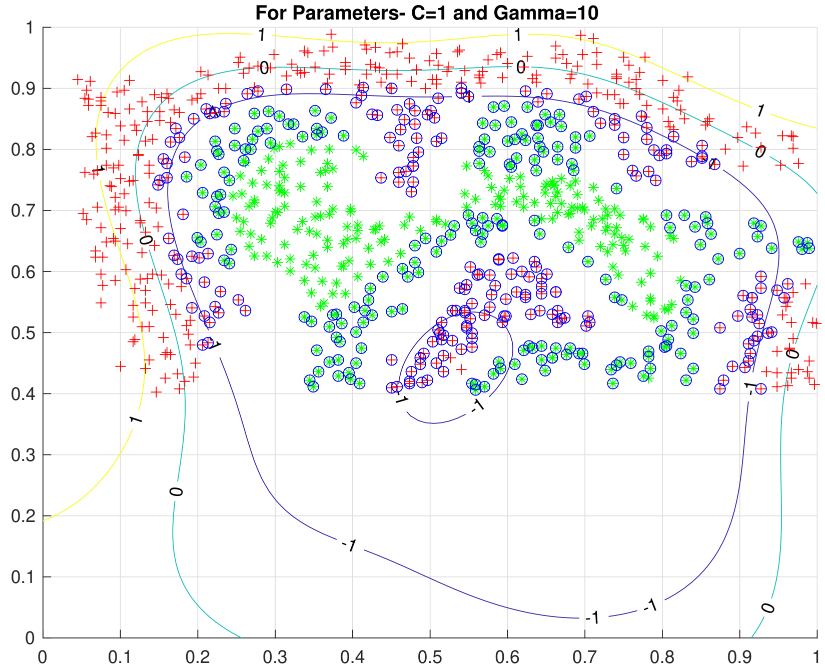

Figure 20: Dual SVM with $\gamma = 10$, and $C = 1$

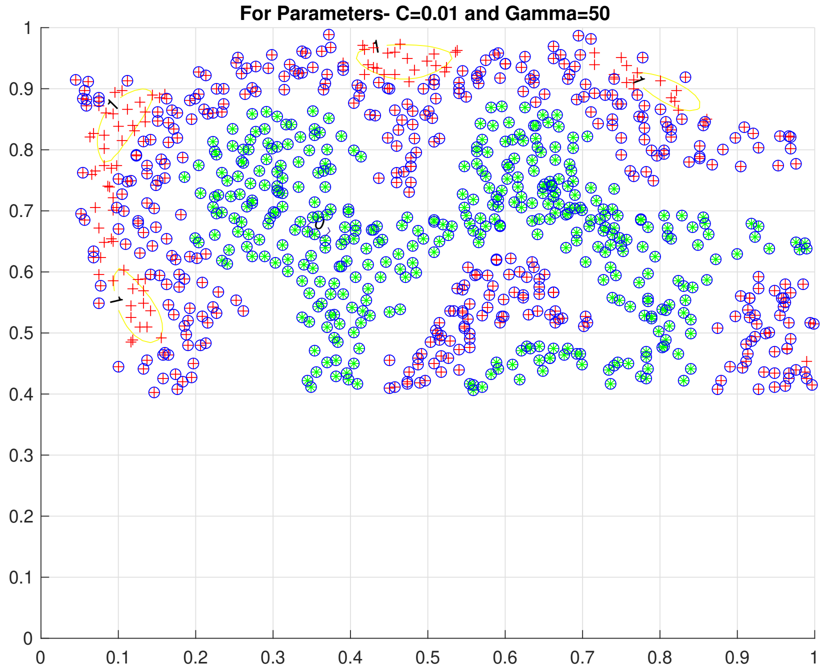

Figure 21: Dual SVM with $\gamma = 50$, and $C = 0.01$

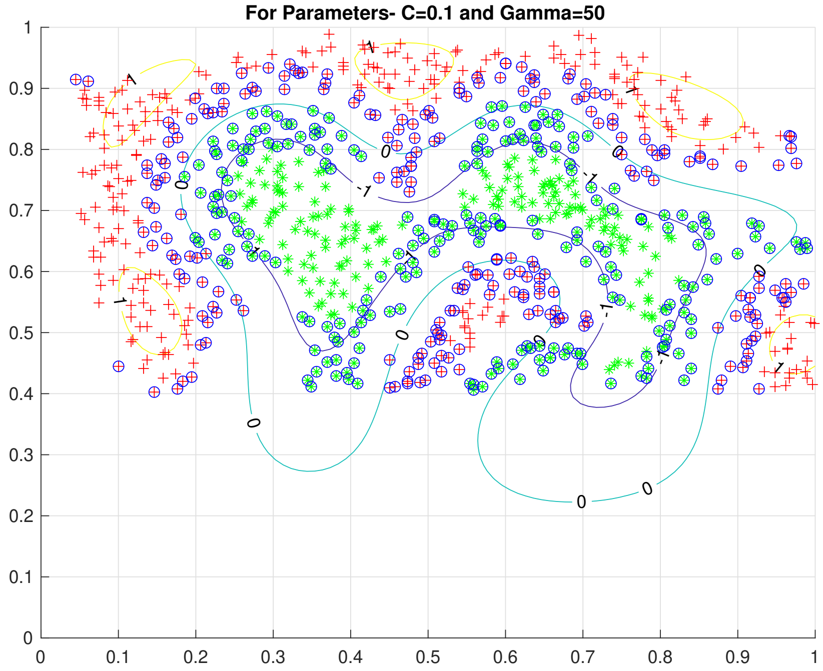

Figure 22: Dual SVM with $\gamma = 50$, and $C = 0.1$

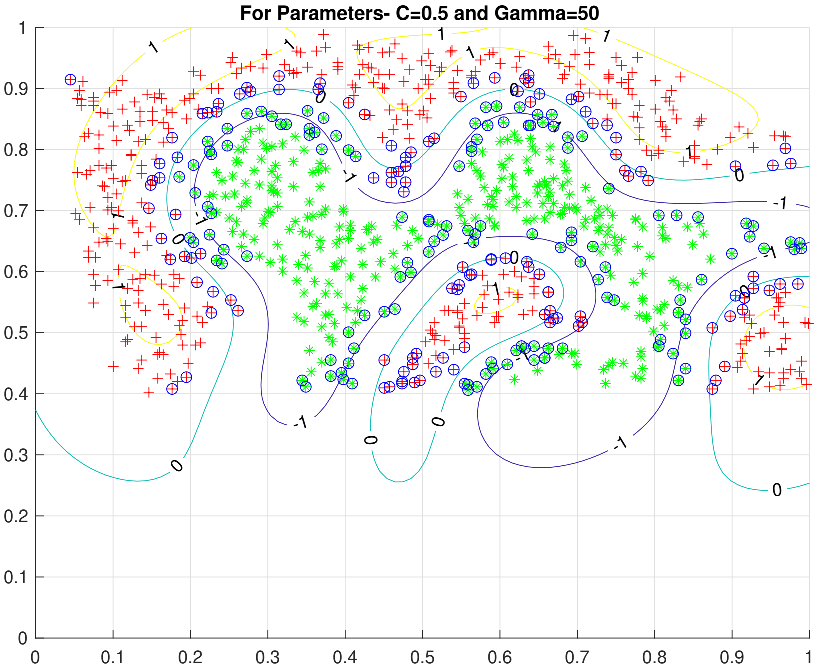

Figure 23: Dual SVM with $\gamma = 50$, and $C = 0.5$

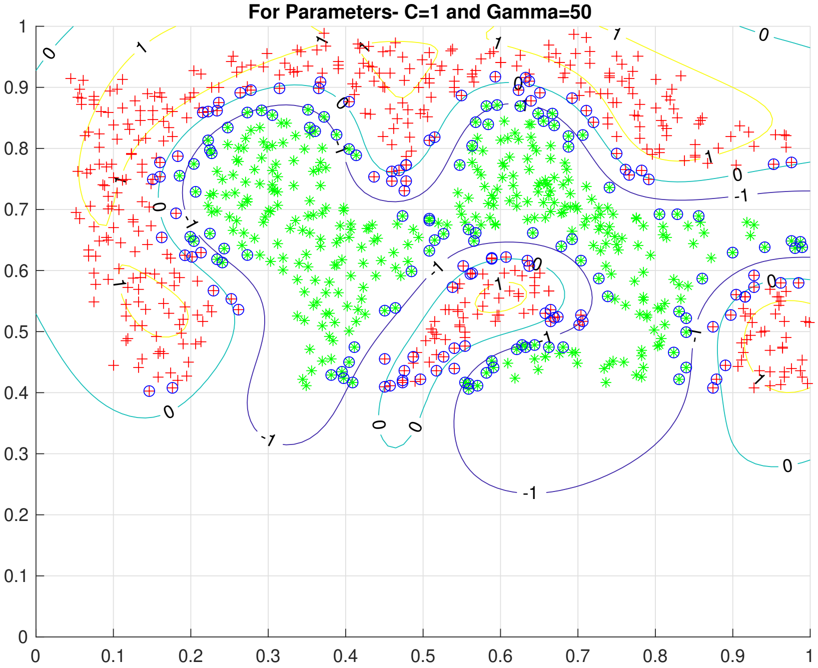

Figure 24: Dual SVM with $\gamma = 50$, and $C = 1$

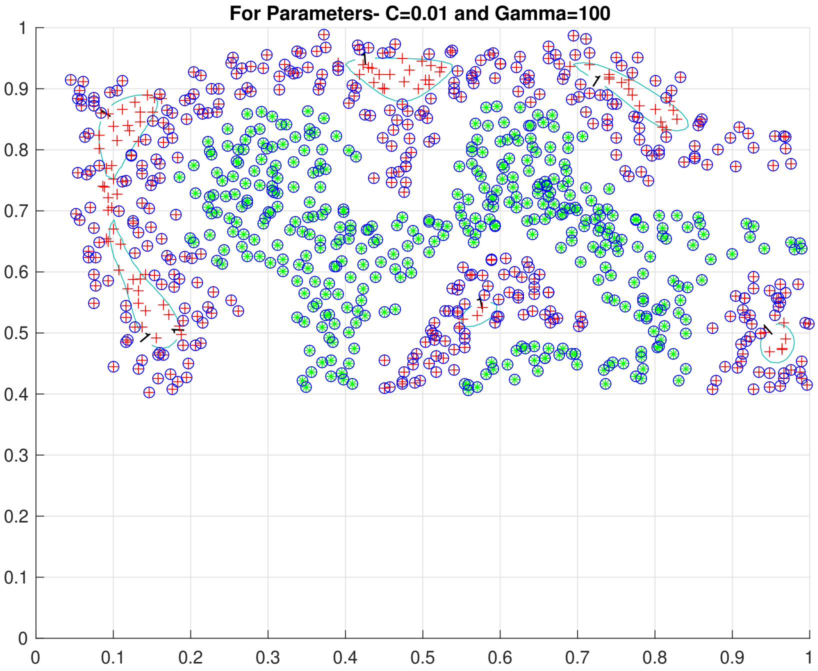

Figure 25: Dual SVM with $\gamma = 100$, and $C = 0.01$

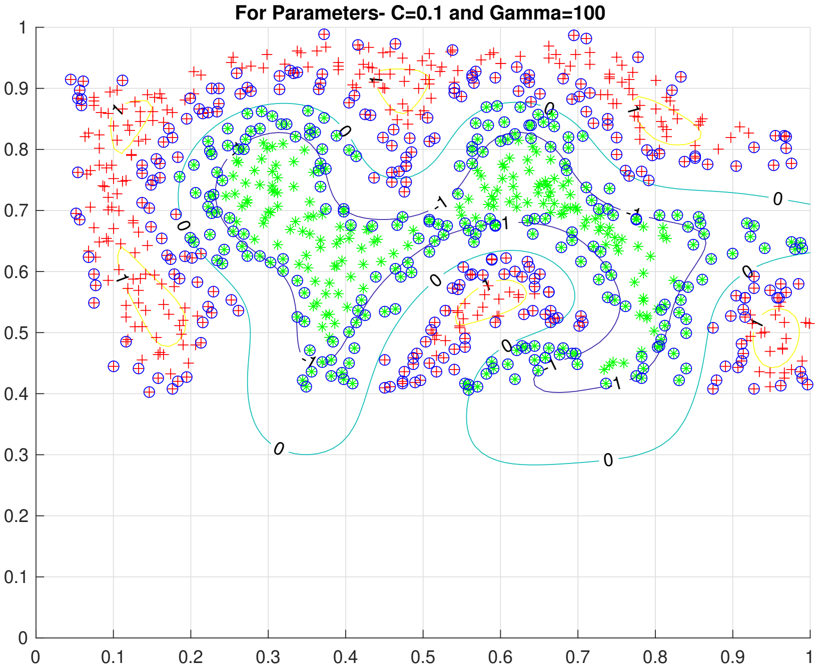

Figure 26: Dual SVM with $\gamma = 100$, and $C = 0.1$

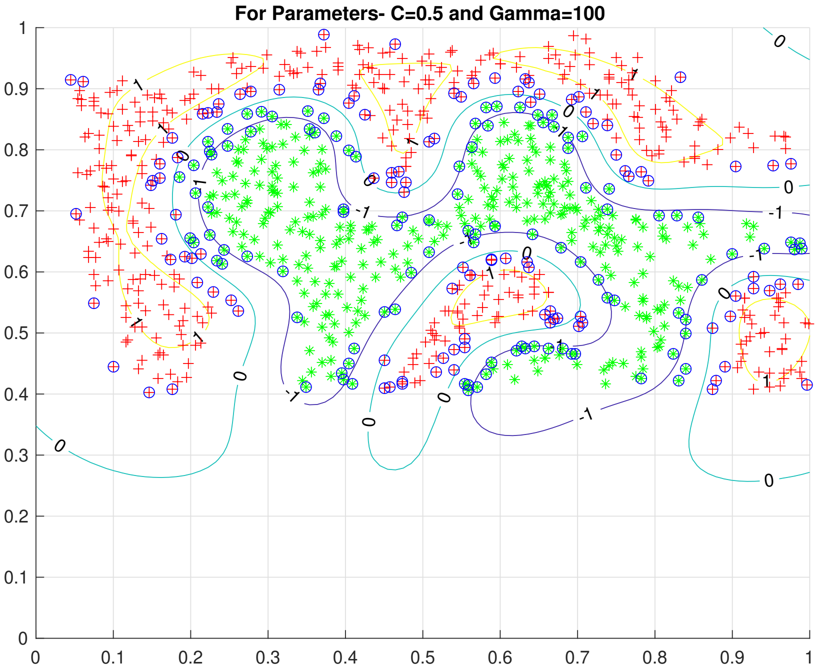

Figure 27: Dual SVM with $\gamma = 100$, and $C = 0.5$

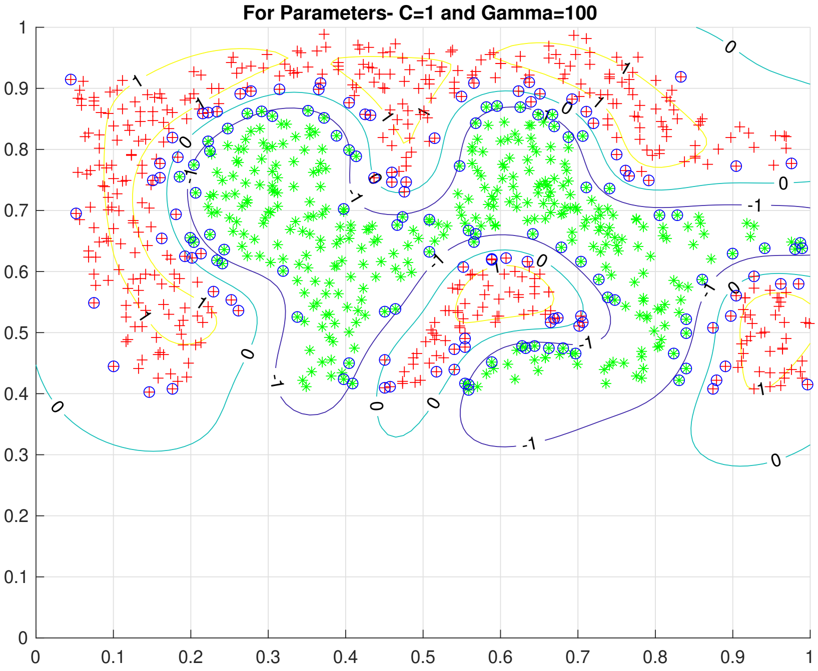

Figure 28: Dual SVM with $\gamma = 100$, and $C = 1$

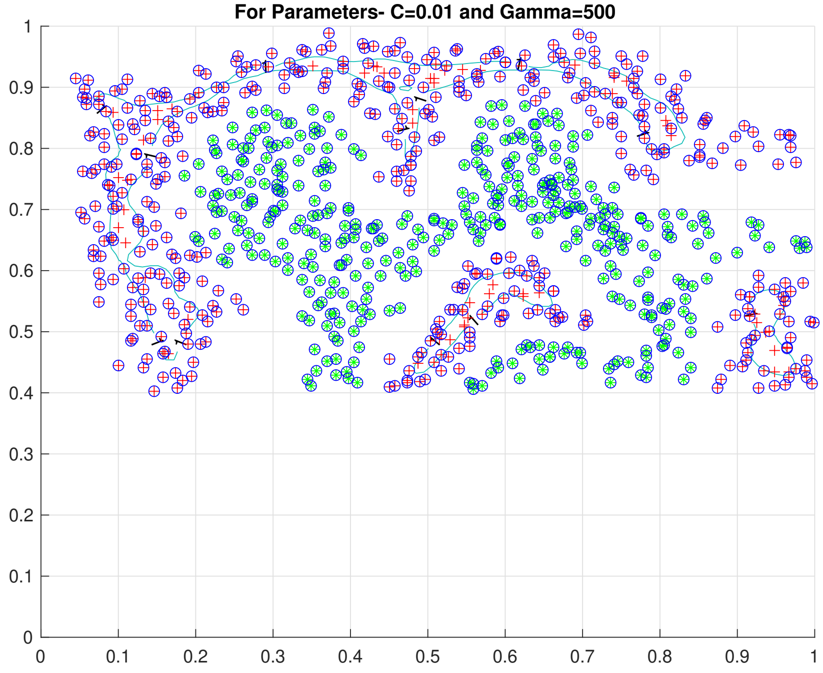

Figure 29: Dual SVM with $\gamma = 500$, and $C = 0.01$

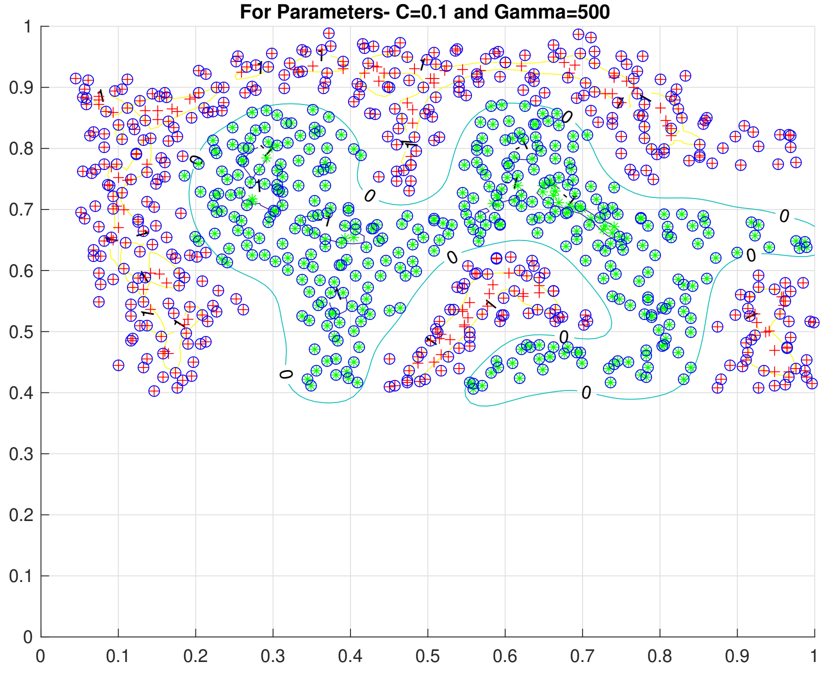

Figure 30: Dual SVM with $\gamma = 500$, and $C = 0.1$

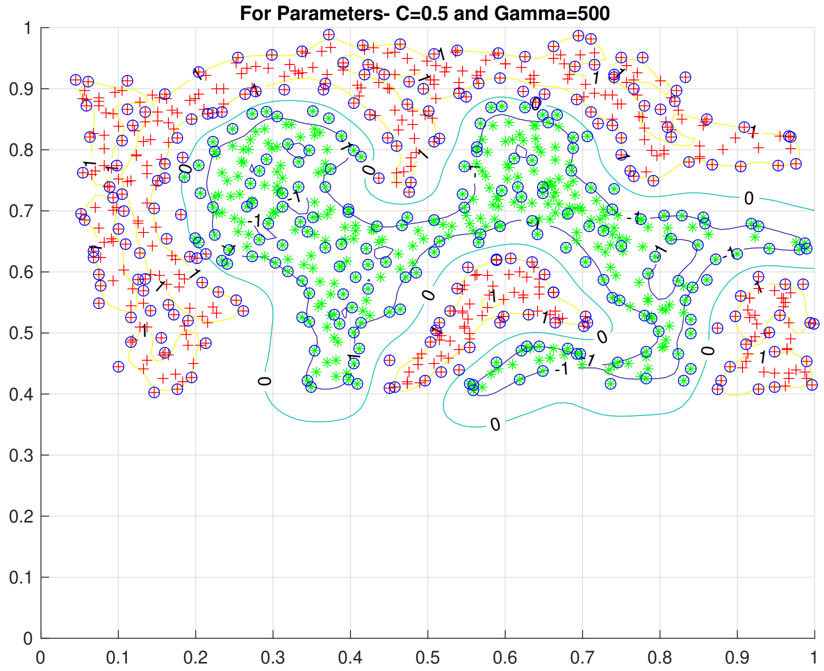

Figure 31: Dual SVM with $\gamma = 500$, and $C = 0.5$

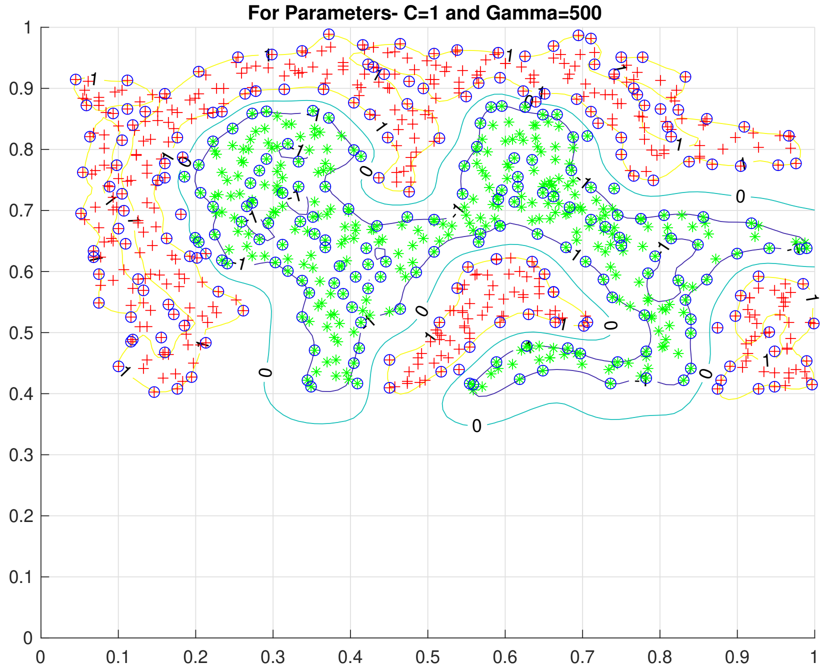

Figure 32: Dual SVM with $\gamma = 500$, and $C = 1$

Problem 1

Part (c)

In this part we are going to solve the linear SVM dual problem (11) using proximal gradient descent (or projected gradient descent). [Please refer to the slides & videos on subgradient and proximal gradient descent.]

The projected gradient descent update step is:

$$\begin{gathered} \alpha^+ = P_D(\alpha - t\nabla f(\alpha)) \\ \text{where } D = \{\alpha : \alpha^T y = 0, 0 \leqslant \alpha \leqslant C1\} \end{gathered} \tag{12}$$

In order to implement projected gradient descent it's necessary to be able to evaluate the (orthogonal) projection onto set D: $P_D(x)$ efficiently. Note that the projection operator is a constrained minimization problem:

$$P_D(x) = \operatorname*{argmin}_z g(z) = \frac{1}{2}||z - x||_2^2 \text{ subject to } z^T y = 0, 0 \leqslant z \leqslant C1 \tag{13}$$

Let's take a look at the KKT condition of this convex program. By Slater's criteria the program has strong duality. If we could solve the KKT condition we get both the primal and dual solution. Introducing dual variable for the equality constraint (and we carry over the inequality constraints without dual variables) we obtain the Lagrangian:

$$\begin{aligned} L(z, \mu) &= \frac{1}{2}||z - x||_2^2 + \mu(z^T y) \\ &= \frac{1}{2}||z - (x - \mu y)||_2^2 - \frac{1}{2}\mu^2||y||_2^2 \end{aligned} \tag{14}$$

The stationarity condition of KKT says that there exists μ^* (the optimal dual solution) such that the optimal z^* satisfies:

$$z^* = \operatorname*{argmin}_{0 \leqslant z \leqslant C1} L(z, \mu^*) = P_{\text{Box}_{[0,C1]}}(x - \mu^* y) \tag{15}$$

And the primal feasibility condition is $y^T z^* = 0$. Combining these two equations we get:

$$y^T P_{\text{Box}_{[0,C1]}}(x - \mu^* y) = 0 \tag{16}$$

The left and side is uni-variate function of μ^*, and the function is monotonic. We can use a simple bisection algorithm [a] to find its root, which gives us μ^*. Once we have μ^*, we can obtain z^* from (15). Therefore, given any point x, we can compute its projection onto D with the above method $P_D(x) = z^*$.

(i) Implement a function that computes $P_D(x)$ using the described procedure above. Verify that your implementation is correct, by comparing with CVX solution for (13) with a couple of different random x.

(ii) Implement the projected gradient descent with backtracking line search. Verify your results with the program in part (b). Use the same γ and C as part (b), question (i).

(iii) For $\gamma = 50, C = 0.5$, plot the convergence history (x-axis the iteration number k, y-axis log scale objective error $f - f^*$ of the dual problem (11). Take the optimal objective value from part (b) Matlab "quadprog" or CVX solver. Do you observe linear convergence or sublinear convergence? Explain it.

[a] https://en.wikipedia.org/wiki/Bisection_method

Solution:

(i.) **Codes:**

- Check Code “`Q3_converge.m`” for $P_D(x)$ implementation.
- Check Code “`Q3_error.m`” for error calculation and verification.

Now, let’s take a look at the error calculation for different scenarios, where the error,

$$\epsilon = \text{norm (CVX output - PD output)}$$

Table 2: Error Calculation For Different Scenarios

Combination		**Error**	**Max Error**
γ	**C**		
10	0.01	1.1591e-05	1e-5
		5.9538e-05	
		4.9271e-05	
		7.2983e-06	
50	0.10	3.7416e-05	
		6.5507e-05	
		3.0781e-05	
		3.9210e-05	
100	0.50	9.9585e-05	
		8.3046e-05	
		7.4592e-05	
		1.7769e-05	
500	1.00	7.7880e-05	
		3.1362e-05	
		2.0569e-05	
		3.5038e-05	

From Table 2, we see, the calculated error is less than the Max error and we conclude that our generated function is capable of having close results as we get from CVX calculation.

(ii.) Check Code “`Q1c_converge.m`”.

Note: As we see, for the same γ and C, we get the similar value of f^* as of the objective function calculated in “`Q.part-b(i)`”. The values are also presented in Table 1. We see, the continuous updating value of f^* finally converges at the objective value. Our code generates all convergence histories for each combination and we observe the similar convergences for all combinations.

(iii.) The Convergence history is for $\gamma = 50$, and $C = 0.5$ is presented in Figure 33.

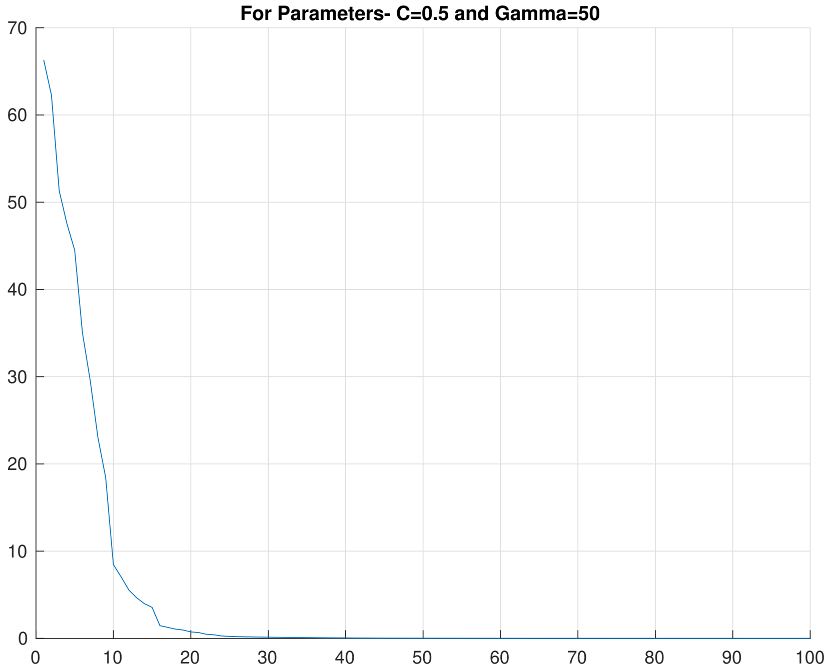

Figure 33: Convergence History for $\gamma = 50$, and $C = 0.5$

From, Figure 33, we observe, this is clearly a linear convergence.
If we consider convergence rate as μ, then in this case,

$$\mu \in (0, 1), i.e. 0 < \mu < 1$$

Note: Convergence is sublinear if $\mu = 1$.