WangZhiyi, 922182514
ZhangTianyao, 922182332

Homework 3

Question 1 - Soft SVM

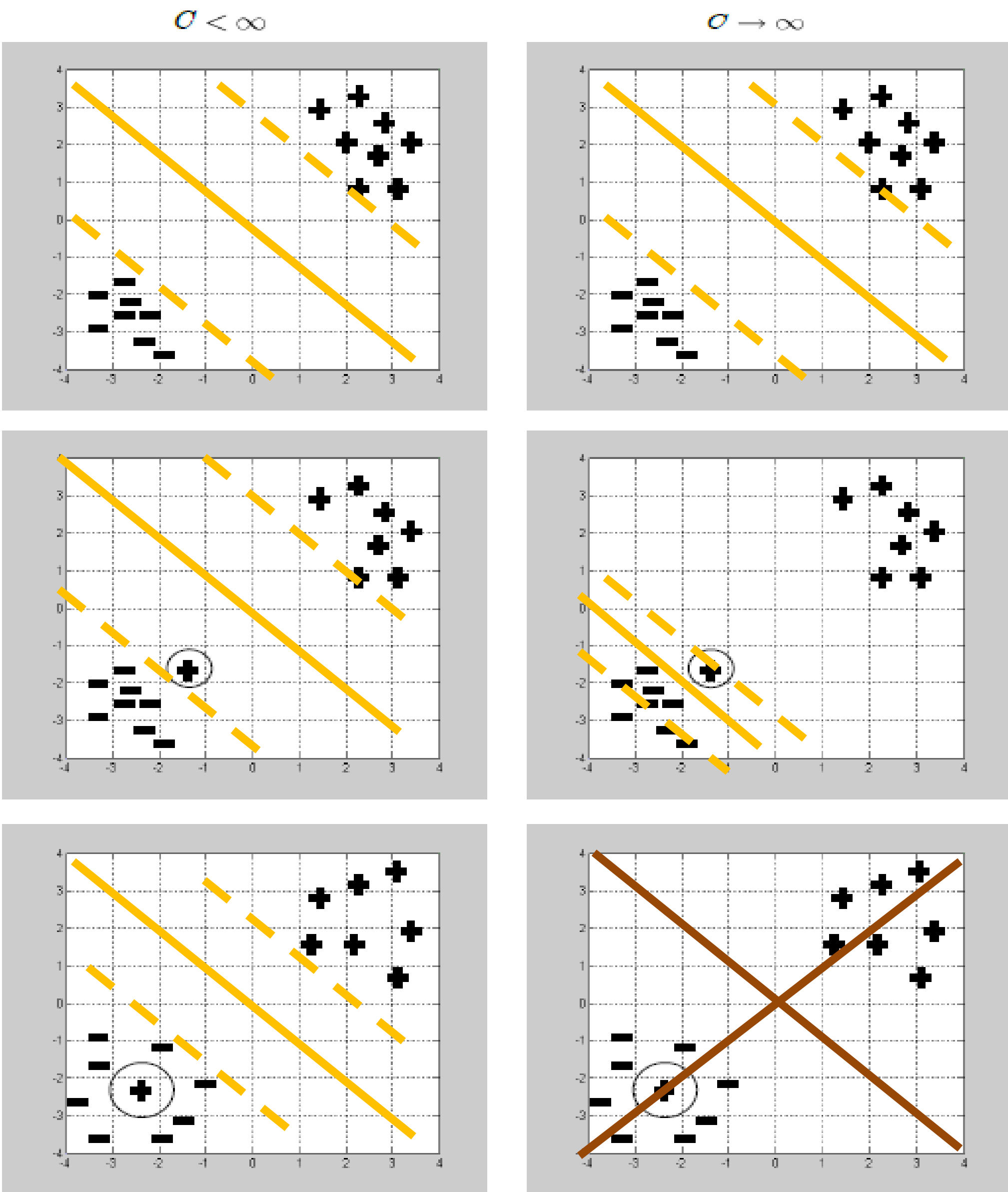
(a) Sketch the separating hyperplane for the three datasets below and for two values of C:

- In the last diagram the hyperplanes does not exist, because the ⊕ is surrounded by -

(b) In the last two problems (4 last figures) there is a circled data point, what is the suitable value of ξ (Equal to 0, between 0 to 1, greater than 1) for that point? Explain.

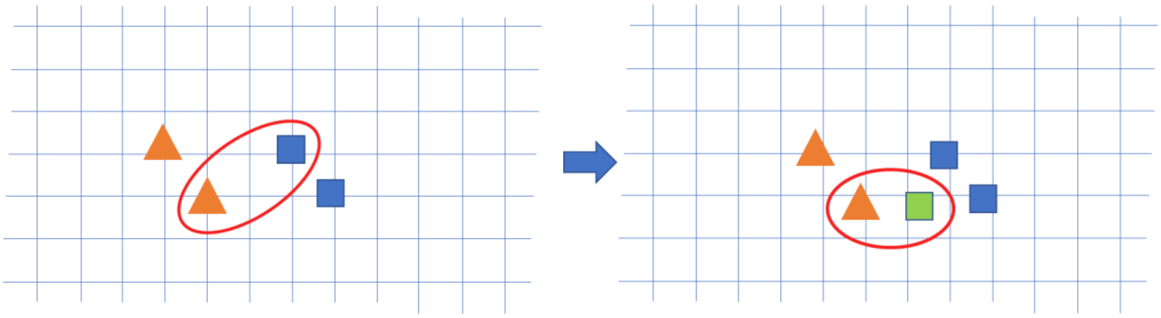
- In the 2nd row, when C<∞, ξ >1, ⊕ are on the negative side of hyprtplane.
- In the 3rd row, when C<∞, ξ >1.

You should attach this page to your homework.

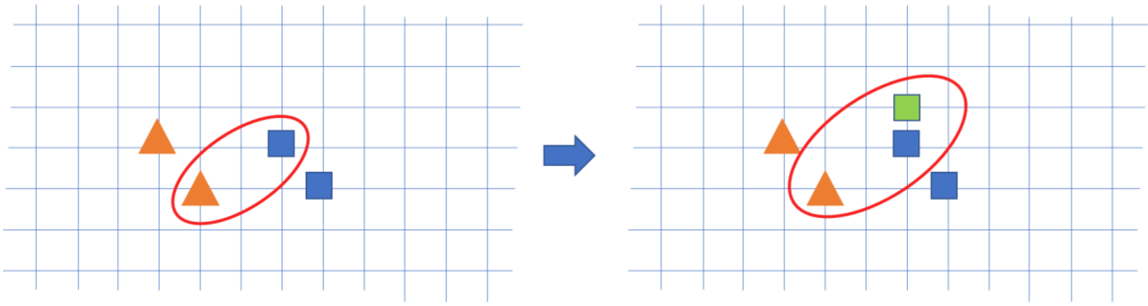


Question 2

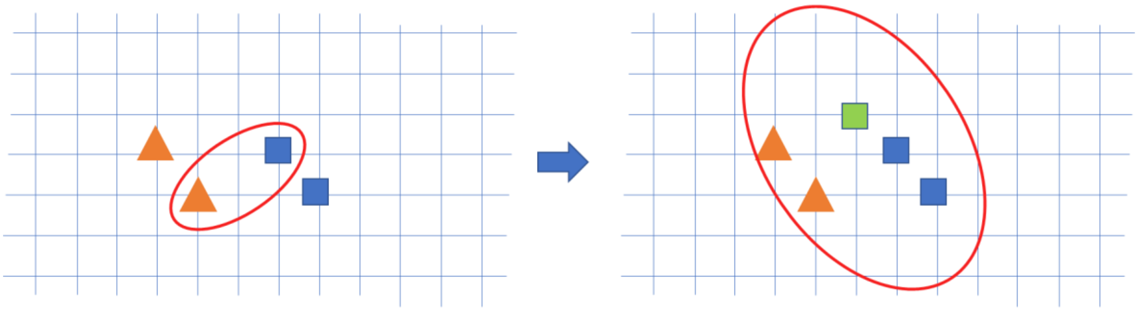
(a) The number of support vector remained $k = 2$.



(b) The number of support vector grew to $k + 1$



(c) The number of support vector grew to $n + 1$.



Question 3

(a) k_3 is also a symmetric function and PSD:

$$k_3(x,x') = k_1(x,x') + k_2(x,x') = k_1(x',x) + k_2(x',x) = k_3(x',x)$$

$$c^T K_3 c = c^T (K_1 + K_2) c = c^T K_1 c + c^T K_2 c \geqslant 0$$

(b) bYes, the classification problem is linearly separable for k_3.

$$k_3(x,x') = \Phi_1^T(x)\Phi_1^T(x') + \Phi_2^T(x)\Phi_2^T(x') = [\Phi_1, \Phi_2]^T \cdot [\Phi_1, \Phi_2] = \Phi_3^T(x)\Phi_3(x')$$

The problem is linear separable for k_1 and k_2, so the problem is linear separable for k_3

Question 4

1. NO the sample is not linear separable.
2. The features are $\left(x_1^2, x_2^2, \sqrt{2}\,x_1 x_2\right)$ according the dataset they can achieve a zero training error.
3. The dataset is mapped to:

$$\Phi(x_i)^T w = \Phi(x_i)^T \sum_{k=1}^{n} \alpha_k y_k \Phi(x_k) = \sum_{k=1}^{n} \alpha_k y_k K(x_i, x_k) \approx \alpha_k y_k K(x_i, x_i) = \alpha_i y_i$$

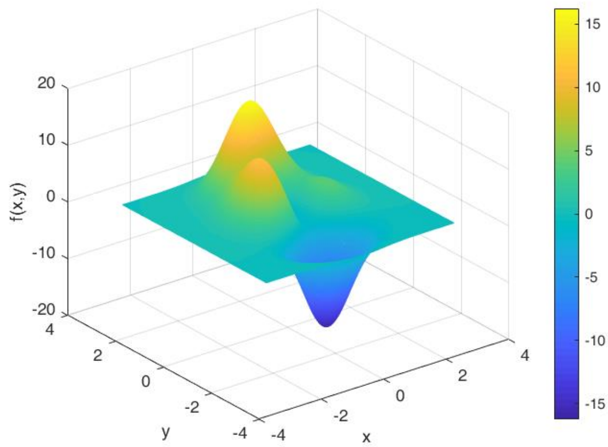
$$\Rightarrow sign\left(\Phi(x_i)^T w\right) = y_i$$

A zero training error is achieved

Question 5

(a)

```
[X,Y]=meshgrid(linspace(-3,3,300),linspace(-3,3,300));
Z=-20*(X./2-X.^2-Y.^5).*exp(-X.^2-Y.^2);
mesh(X,Y,Z);
hold on
xlabel('x');ylabel('y');zlabel('f(x,y)');
```

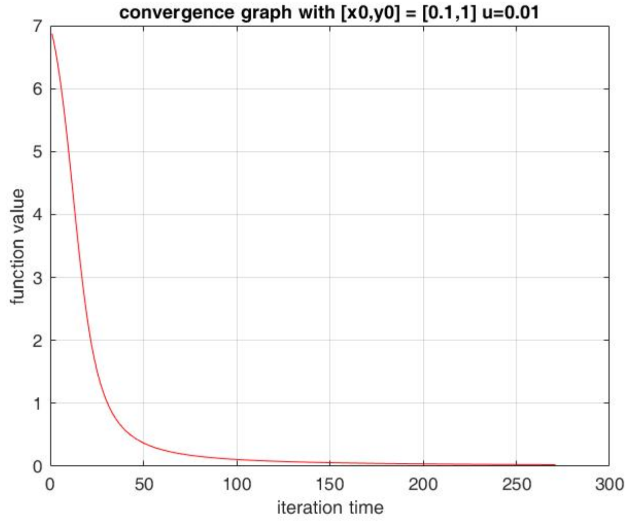


(a) Implement the gradient descent method for finding the minimum point. Attach your code to your submitted pdf file.

(b) Initialize your algorithm with the following values:

- $[x_0, y_0] = [0.1, 1]$.
- $\eta = 0.01$.

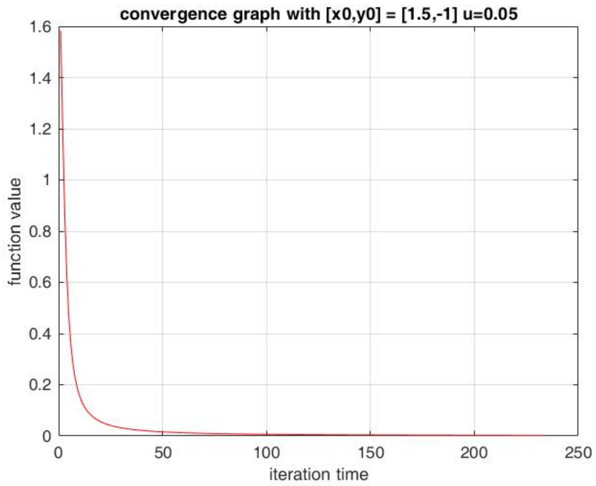
Plot the convergence graph of the algorithm (i.e. the value of the function at each step). To what point if any the algorithm converges?



(c) Initialize your algorithm with the following values:

- $[x_0, y_0] = [1.5, -1]$.
- $\eta = 0.05$.

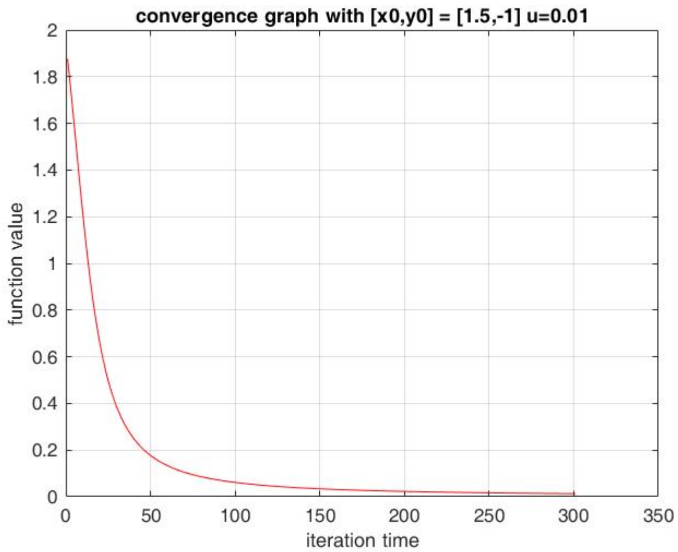
Plot the convergence graph of the algorithm. To what point if any the algorithm converges? Which phenomenon can be observed?



(d) Initialize your algorithm with the following values:

- $[x_0, y_0] = [1.5, -1]$.
- $\eta = 0.01$.

Plot the convergence graph of the algorithm. To what point if any the algorithm converges? Compare your results with the results of part (d).



```
clear all;
clc;
syms X;
syms Y;
F1 = -20.*(X./2 - X.^2 - Y.^2).*exp(-X.^2-Y.^2);
X =  linspace(-3,3,100);
Y = linspace(-3,3,100);
dfx = diff(F1,'X',1);
dfy = diff(F1,'Y',1);
disp(dfx);
disp(dfy);
W = [];

thresh = 0.001;
u = 0.01;
iter_time = 1;
v = [1.5,-1];
while iter_time < 10000
    df = [Dfx(v(1),v(2)),Dfx(v(1),v(2))];
    delta_v = u*[df(1),df(2)];
    v = v - delta_v;
    W(iter_time) = -20.*(v(1)./2 - v(1).^2 -
v(2).^2).*exp(-v(1).^2-v(2).^2);
    iter_time = iter_time + 1;
    if norm(delta_v) <= thresh
        break;
    end
end
%}
figure();
iter_time = 1 : 1 : iter_time - 1;
plot(iter_time, W,'r');
title('convergence graph with [x0,y0] = [1.5,-1]
u=0.01');
xlabel('iteration time');
ylabel('function value');
grid on;



function dfx = Dfx(X,Y)
dfx = exp(- X^2 - Y^2)*(40*X - 10) - 2*X*exp(- X^2 -
Y^2)*(20*X^2 - 10*X + 20*Y^2);
end

function dfy = Dfy(X,Y)
dfy = 40*Y*exp(- X^2 - Y^2) - 2*Y*exp(- X^2 -
Y^2)*(20*X^2 - 10*X + 20*Y^2);
end
```

The $f(x; y)$ is not is convex everywhere there are many ways to get the gradient=0