```
% IDS/ACM/CS 158: Fundamentals of Statistical Learning
% PS5, Problem 2: Soft Margin Hyperplane
% Author: Michael Li, mlli@caltech.edu
%-----------------------------------------------------------------------
clear;

D = readmatrix('dataset8.csv');
X = D(:, 1:end-1);
ys = D(:,end);
g_plus = D(ys == 1, 1:end-1);
g_minus = D(ys == -1, 1:end-1);

% C = .1
C = .1;
N = size(D,1);
p = size(D(1,1:end-1), 2);

% solve dual problem using C constraint
H = (ys*transpose(ys)) .* (X*transpose(X));
dual_margin = quadprog(H, -1*ones(1,N), zeros(1,N), 0, transpose(ys),
 0, zeros(N,1), (1/C)*ones(N,1));

% finding beta using lambdas
support_vecs = D(abs(dual_margin) > 10^-5, :);
support_plus = support_vecs(support_vecs(:,3)==1, 1:end-1);
support_minus = support_vecs(support_vecs(:,3)==-1, 1:end-1);
beta = sum(dual_margin .* ys .* X, 1);
beta0 = -1/2 * (max(beta*transpose(support_plus)) +
 min(beta*transpose(support_minus)));
dual_beta = [beta0 beta];
fprintf("\nNumber of Support Vectors C=.1 are %i\n",
 size(support_vecs, 1))
% C=.1 has 8 support vectors

% get points for decision boundary and margins
x = linspace(-3, 5, 10000);
f=@(x) (-dual_beta(2) / dual_beta(3))*x - (dual_beta(1) /
 dual_beta(3));
Y=f(x);

g=@(x) (-dual_beta(2) / dual_beta(3))*x - ((dual_beta(1) - 1) /
 dual_beta(3));
Z=g(x);

h=@(x) (-dual_beta(2) / dual_beta(3))*x - ((dual_beta(1) + 1) /
 dual_beta(3));
P=h(x);

% plot
figure
hold on
plot(x, Y, 'k')
```

```
plot(x, Z, '--k')
plot(x, P, '--k')
plot(g_plus(:,1), g_plus(:, 2), 'or')
plot(g_minus(:,1), g_minus(:, 2), 'ob')
plot(support_vecs(:,1), support_vecs(:,2), 'xk')
title('Dataset 8 with Soft Margin Hyperplane C=.1')
xlabel('X1')
ylabel('X2')

% C = 10
C = 10;

% solve dual problem using C constraint
H = (ys*transpose(ys)) .* (X*transpose(X));
dual_margin = quadprog(H, -1*ones(1,N), zeros(1,N), 0, transpose(ys),
 0, zeros(N,1), (1/C)*ones(N,1));

% finding beta using lambdas
support_vecs = D(abs(dual_margin) > 10^-5, :);
support_plus = support_vecs(support_vecs(:,3)==1, 1:end-1);
support_minus = support_vecs(support_vecs(:,3)==-1, 1:end-1);
beta = sum(dual_margin .* ys .* X, 1);
beta0 = -1/2 * (max(beta*transpose(support_plus)) +
 min(beta*transpose(support_minus)));
dual_beta = [beta0 beta];
% disp(dual_beta)
fprintf("\nNumber of Support Vectors C=10 are %i\n",
 size(support_vecs, 1))
% C=10 has 27 support vectors

% get points for decision boundary and margins
x = linspace(-3, 5, 10000);
f=@(x) (-dual_beta(2) / dual_beta(3))*x - (dual_beta(1) /
 dual_beta(3));
Y=f(x);
g=@(x) (-dual_beta(2) / dual_beta(3))*x - ((dual_beta(1) - 1) /
 dual_beta(3));
Z=g(x);
h=@(x) (-dual_beta(2) / dual_beta(3))*x - ((dual_beta(1) + 1) /
 dual_beta(3));
P=h(x);

% plot
figure
hold on
plot(x, Y, 'k')
plot(x, Z, '--k')
plot(x, P, '--k')
plot(g_plus(:,1), g_plus(:, 2), 'or')
plot(g_minus(:,1), g_minus(:, 2), 'ob')
plot(support_vecs(:,1), support_vecs(:,2), 'xk')
title('Dataset 8 with Soft Margin Hyperplane C=10')
xlabel('X1')
ylabel('X2')
```

Minimum found that satisfies the constraints.

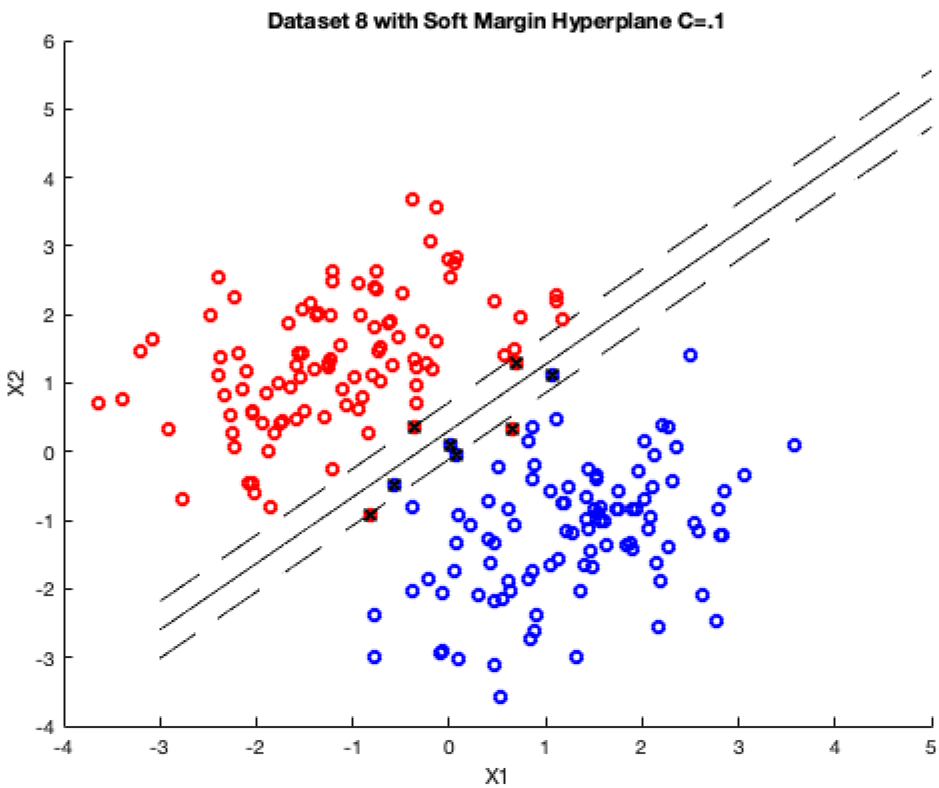
Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

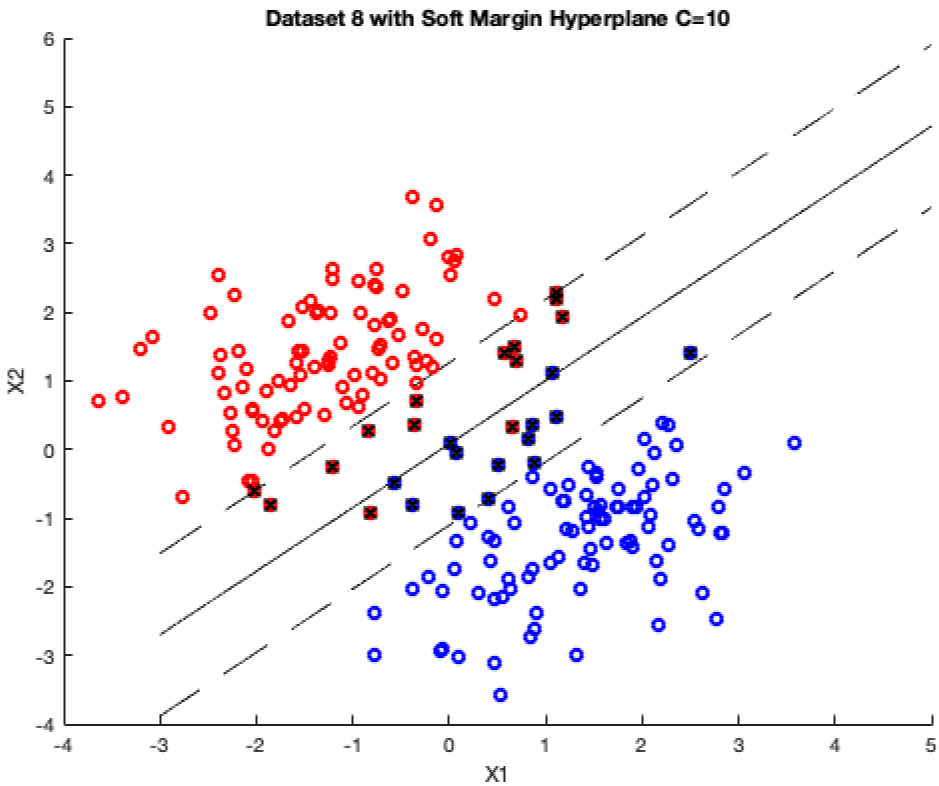
Number of Support Vectors C=.1 are 8

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

Number of Support Vectors C=10 are 27





Published with MATLAB® R2019a