Contents

Vincent Purcell - HW 4 - ECE487

```
clear; clc; close all;
```

Problem 4.6

Problem 4.6 from the Text on page 248.

```
% Data generation based on inputs from text book
rng('default')
rng(1)
m = [-5 5 5 -5; 5 -5 5 -5];
s = 2;
N = 100;
[x1,y1] = data_generator(m,s,N);
x1 = x1';
y1 = y1';
rng(10);
[x2,y2] = data_generator(m,s,N);
x2 = x2';
y2 = y2';

C_vec = [1,100,1000]';
sigma_vec = [0.5,1,2,4]';
tol = 0.001;

% Create 12 models and plot them based on all combinations of sigma and C
for i=1:size(C_vec)
    for j=1:size(sigma_vec)
        plotSVM(x1,y1,x2,y2,tol,C_vec(i),sigma_vec(j));
    end
end

% Call Decision Tree Function
decisionTree(x1,y1,x2,y2);
```

SVM Classification

Classification and Plot Function

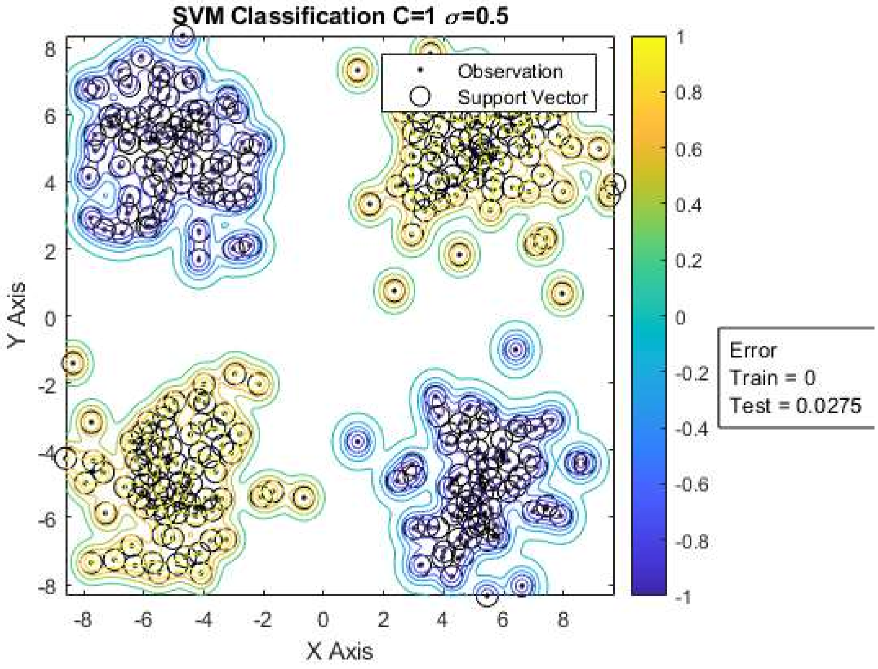
```
function plotSVM(x1,y1,x2,y2,tol,C,sigma)

    %Get classifier model and errors
    [model, test_err, train_err] = SVM_clas(x1,y1,x2,y2,tol,C,sigma);
    svInd = model.IsSupportVector;
    %Below plotting methods adapted from fitcsvm MATLAB documentation
    h = 0.02;
    [X1,X2] = meshgrid(min(x1(:,1)):h:max(x1(:,1)),...
        min(x1(:,2)):h:max(x1(:,2)));
    [~,score] = predict(model,[X1(:),X2(:)]);
    scoreGrid = reshape(score(:,1),835,916);

    figure
    plot(x1(:,1),x1(:,2),'k.')
    hold on
    plot(x1(svInd,1),x1(svInd,2),'ko','MarkerSize',10)
    contour(X1,X2,scoreGrid)
    colorbar;
    title_str = "SVM Classification C=" + num2str(C) + " \sigma=" + num2str(sigma);
    title(title_str)
    xlabel('X Axis')
    ylabel('Y Axis')
    legend('Observation','Support Vector')
    a = gca; % get the current axis;
    % set the width of the axis (the third value in Position)
    % to be 60% of the Figure's width
    a.Position(3) = 0.6;
    text1 = {"Error","Train = " + num2str(train_err) ...
        ,"Test = " + num2str(test_err)};
    annotation('textbox',[0.83 0 0 .5],'String',text1,'FitBoxToText','on')
    hold off
    snapnow
end
```

SVM Classifier

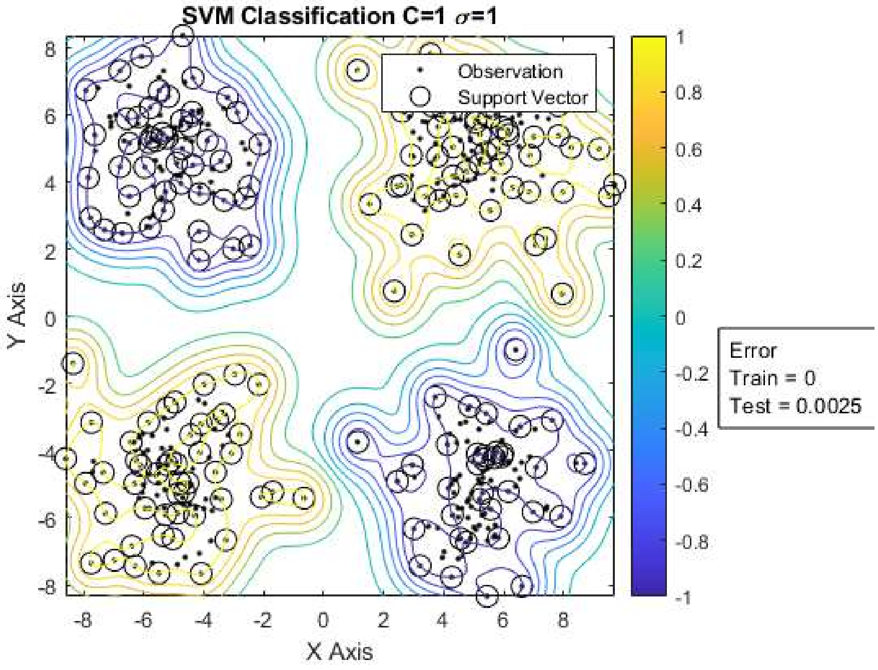
Function adapted from function on page 247 of the text

```
function [model,test_err,train_err]=SVM_clas(x1,y1,x2,y2,tol,C,sigma)

    % The following options are from the function in the textbook, it
    % required simple adaptation to the new function fitcsvm:
    % DeltaGradientTolerance = tol
    % Solver = SMO
    % Verbose = 1
    % IterationLimit = 20000
    % CacheSize = 10000
    % KernelFunction = RBF
    % KernelScale = sigma
    % BoxConstraint = C
    model = fitcsvm(x1,y1, ...
        'DeltaGradientTolerance',tol,...
        'Solver','SMO',...
        'Verbose',1,...
        'IterationLimit',20000,...
        'CacheSize',10000,...
        'KernelFunction','RBF',...
        'KernelScale',sigma,...
        'BoxConstraint',C);

    %Computation of the error probability
    test_err = loss(model,x2,y2);
    train_err = loss(model,x1,y1);
end
```

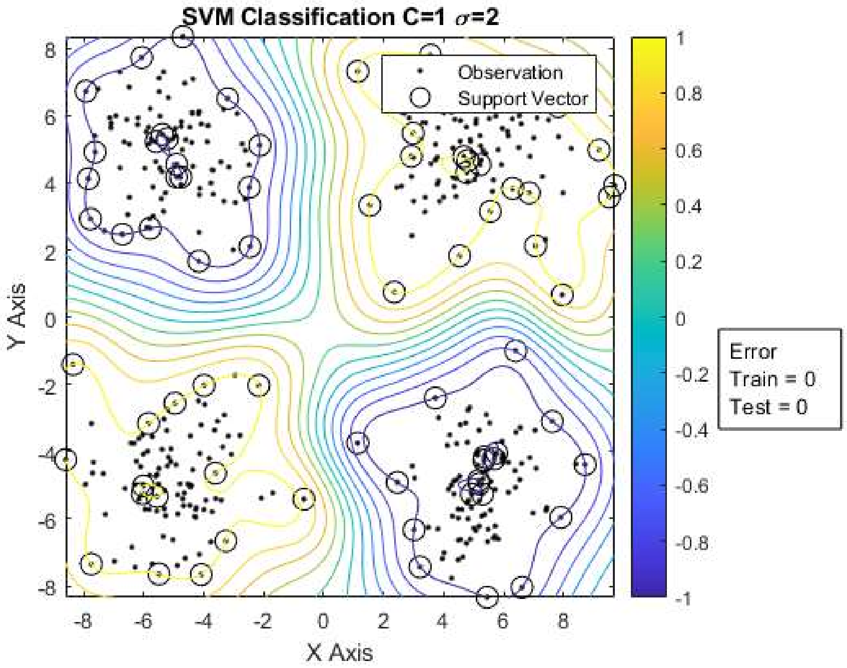
```
|===========================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===========================================================================================================================|
|            0 |active|         400  |  9.975062e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|         1000 |active|         400  |  8.407885e-04 |  1.983371e-03 |  1.018865e-03 |          297 | -7.239337e+01 |  3.339343e-16 |
|         1151 |active|         400  |  3.855007e-04 |  9.980384e-04 |  5.010877e-04 |          297 | -7.239356e+01 |  6.071532e-16 |

 Exiting Active Set upon convergence due to DeltaGradient.
```



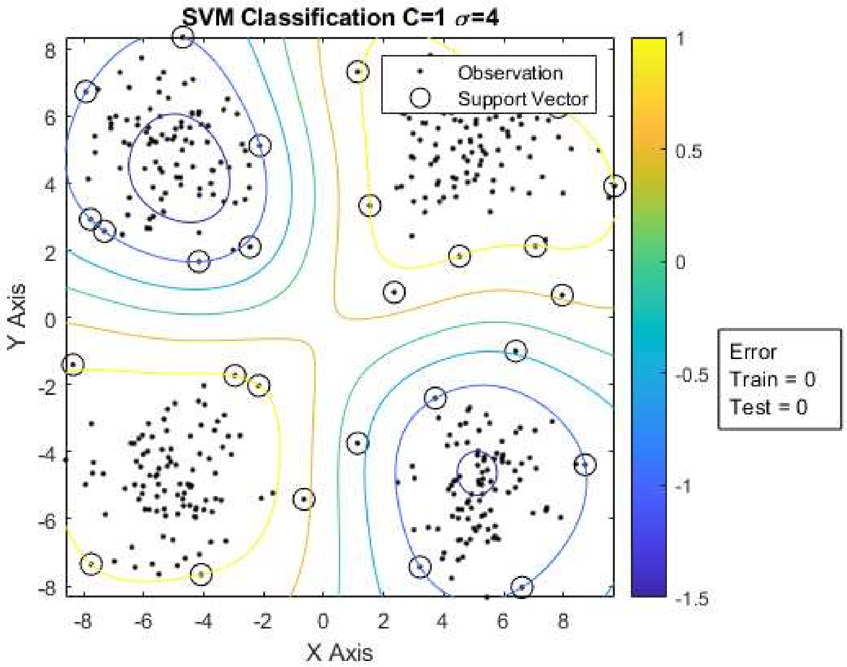
```
|====================================================================================================================================|
|   Iteration   | Set  |   Set Size   |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|               |      |              |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|====================================================================================================================================|
|             0 |active|          400 |  9.975062e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|          1000 |active|          400 |  1.165708e-03 |  2.337429e-03 |  1.254457e-03 |          188 | -3.195291e+01 |  8.413409e-17 |
|          1167 |active|          400 |  4.927793e-04 |  9.834192e-04 |  5.021250e-04 |          184 | -3.195314e+01 |  2.645453e-17 |

Exiting Active Set upon convergence due to DeltaGradient.
```



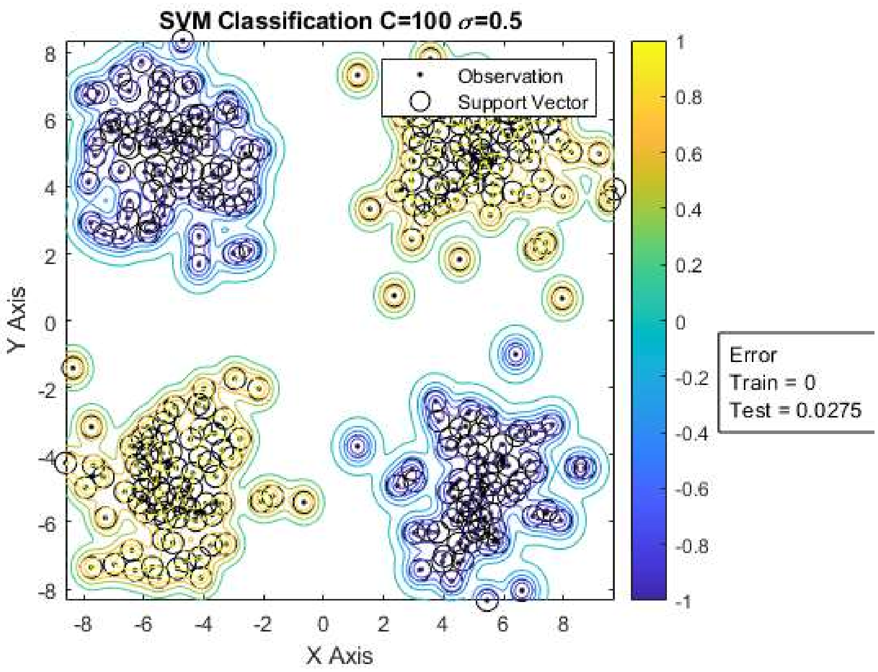
```
|====================================================================================================================================|
|   Iteration   | Set  |   Set Size   |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|               |      |              |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|====================================================================================================================================|
|             0 |active|          400 |  9.975062e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|           419 |active|          400 |  5.294154e-04 |  9.618164e-04 |  4.838165e-04 |           75 | -1.382540e+01 |  1.953732e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



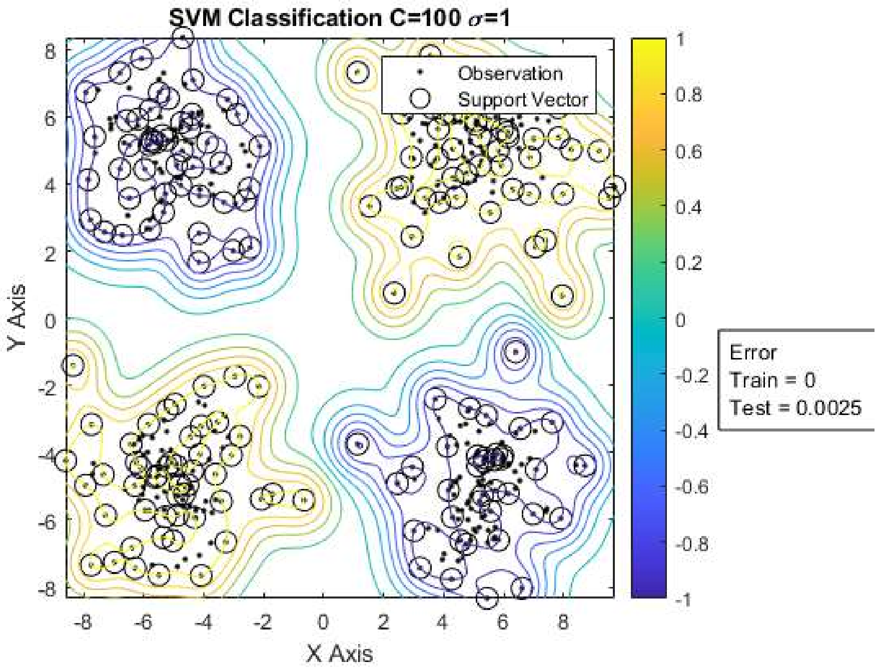
```
|===================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================|
|            0 |active|         400 |  9.975062e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|           88 |active|         400 |  1.282562e-04 |  6.374282e-04 |  3.575460e-04 |           28 | -9.747927e+00 |  2.775558e-17 |

 Exiting Active Set upon convergence due to DeltaGradient.
```



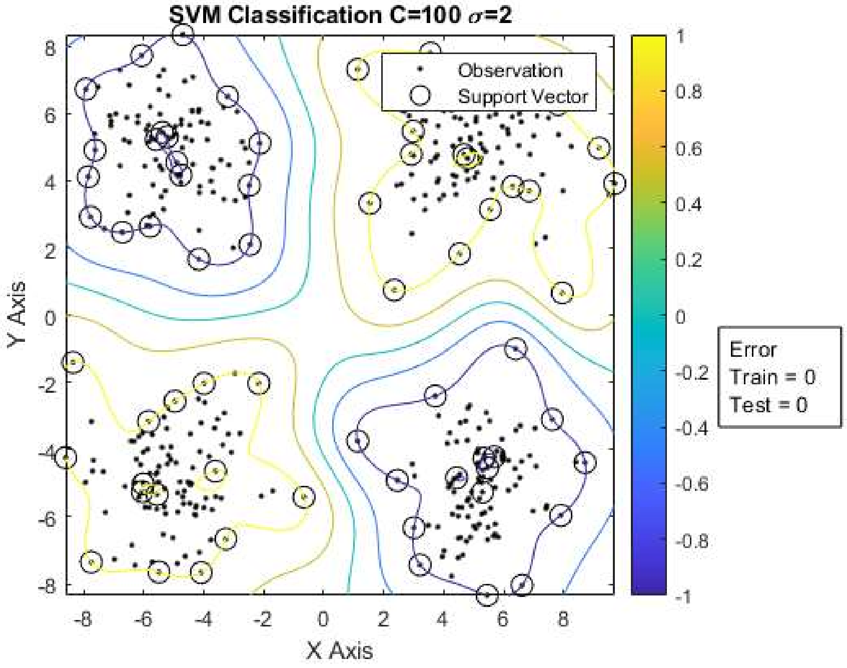
```
|===================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================|
|            0 |active|         400 |  9.999750e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|         1000 |active|         400 |  7.582241e-02 |  2.810799e-03 |  1.665751e-03 |          296 | -7.240008e+01 |  5.585810e-16 |
|         1160 |active|         400 |  4.109412e-02 |  9.976687e-04 |  5.177254e-04 |          297 | -7.240025e+01 |  7.754214e-16 |
Exiting Active Set upon convergence due to DeltaGradient.
```



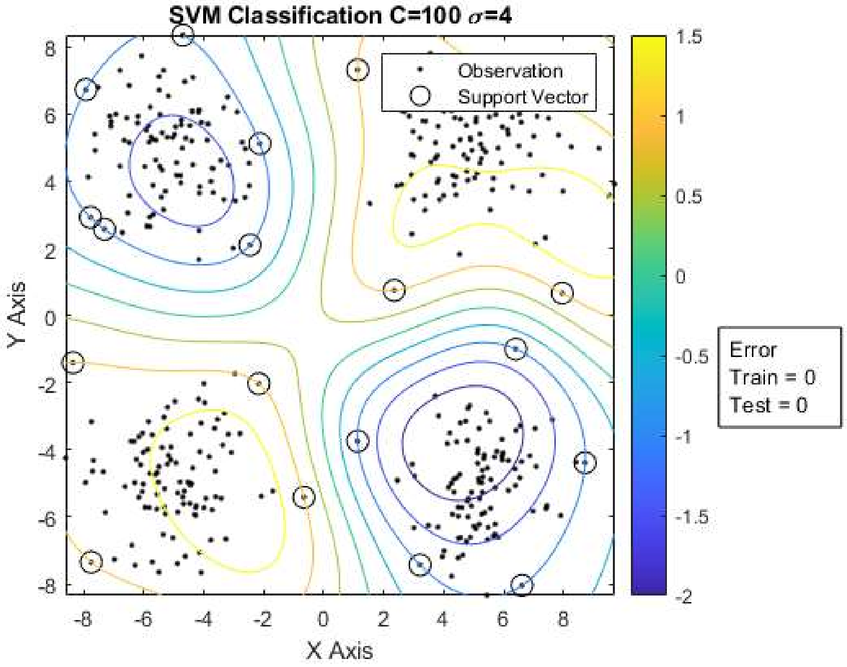
```
|===================================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================================|
|            0 |active|         400 |  9.999750e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|         1000 |active|         400 |  7.382879e-02 |  2.080937e-03 |  1.355332e-03 |          187 | -3.195580e+01 |  3.068292e-16 |
|         1063 |active|         400 |  5.763212e-02 |  9.999235e-04 |  5.191424e-04 |          186 | -3.195587e+01 |  1.040834e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



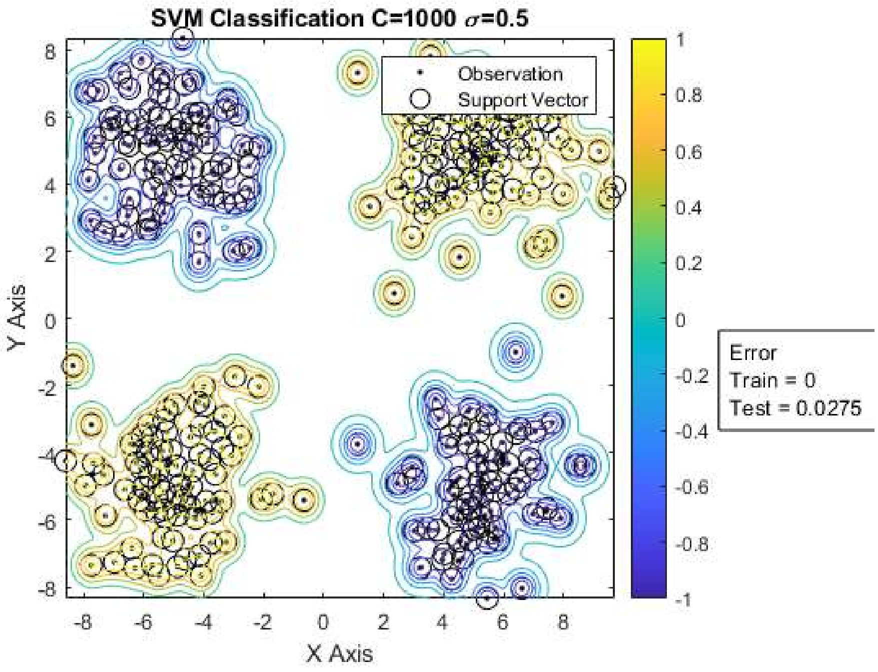
```
|===================================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================================|
|          0 |active|         400 |  9.999750e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|        386 |active|         400 |  3.377326e-02 |  8.624039e-04 |  4.503157e-04 |           68 | -1.387275e+01 |  6.982262e-17 |

Exiting Active Set upon convergence due to DeltaGradient.
```



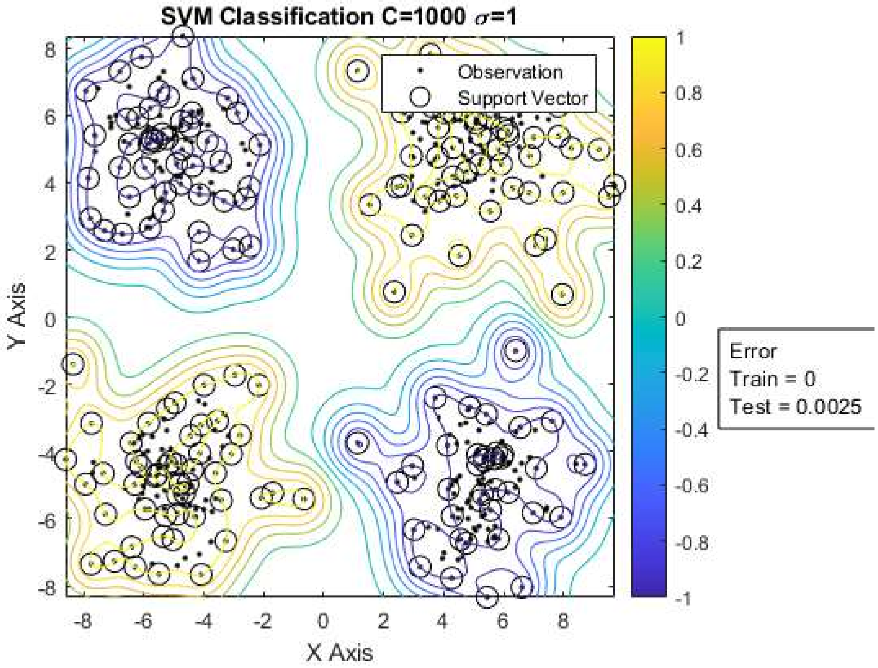
```
|=====================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|=====================================================================================================================================|
|          0 |active|         400 |  9.999750e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|         61 |active|         400 |  1.694130e-02 |  9.455347e-04 |  5.715393e-04 |           19 | -1.299826e+01 |  4.996004e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



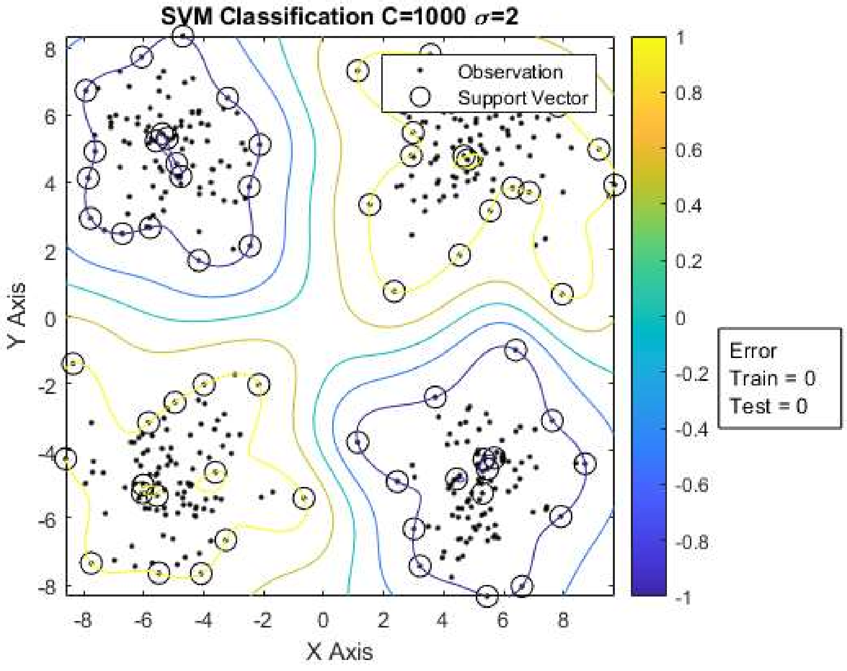
```
|=====================================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================|
|         0 |active|         400 |  9.999975e-01 |  2.000000e+00 |  1.000000e+00 |           0 |  0.000000e+00 |  0.000000e+00 |
|      1000 |active|         400 |  4.503944e-01 |  2.810799e-03 |  1.665751e-03 |         296 | -7.240008e+01 |  5.585810e-16 |
|      1160 |active|         400 |  2.999707e-01 |  9.976687e-04 |  5.177254e-04 |         297 | -7.240025e+01 |  7.754214e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



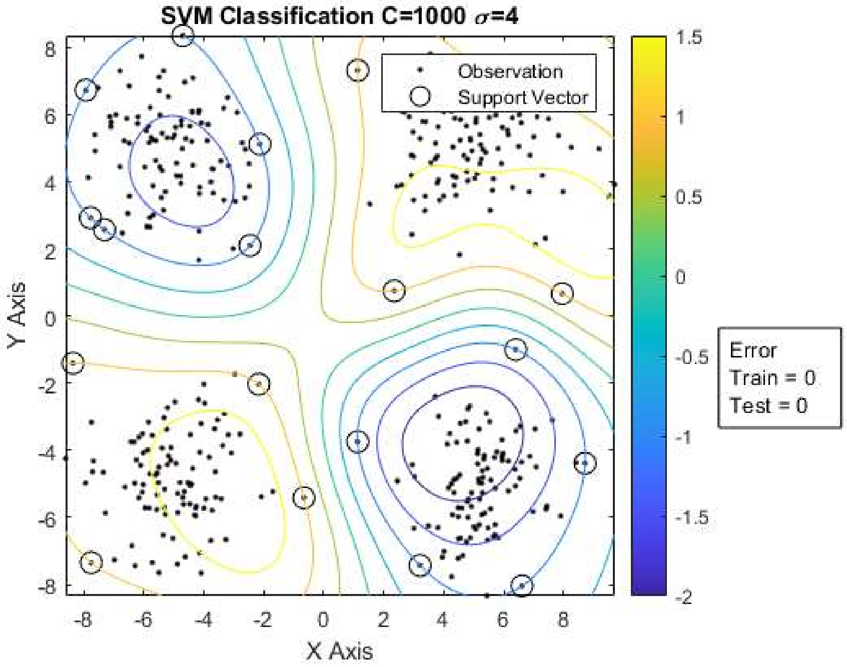
```
|===================================================================================================================|
|   Iteration  | Set  |   Set Size  |  Feasibility  |     Delta     |      KKT      |  Number of  |   Objective   |   Constraint  |
|              |      |             |      Gap      |    Gradient   |   Violation   |  Supp. Vec. |               |   Violation   |
|===================================================================================================================|
|         0 |active|         400 |  9.999975e-01 |  2.000000e+00 |  1.000000e+00 |           0 |  0.000000e+00 |  0.000000e+00 |
|      1000 |active|         400 |  4.435076e-01 |  2.080937e-03 |  1.355332e-03 |         187 | -3.195580e+01 |  3.068292e-16 |
|      1063 |active|         400 |  3.794840e-01 |  9.999235e-04 |  5.191424e-04 |         186 | -3.195587e+01 |  1.040834e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



```
|===================================================================================================================================|
|   Iteration  | Set  |   Set Size   |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |              |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================|
|            0 |active|          400 |  9.999975e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|          386 |active|          400 |  2.581275e-01 |  8.624039e-04 |  4.503157e-04 |           68 | -1.387275e+01 |  6.982262e-17 |

Exiting Active Set upon convergence due to DeltaGradient.
```



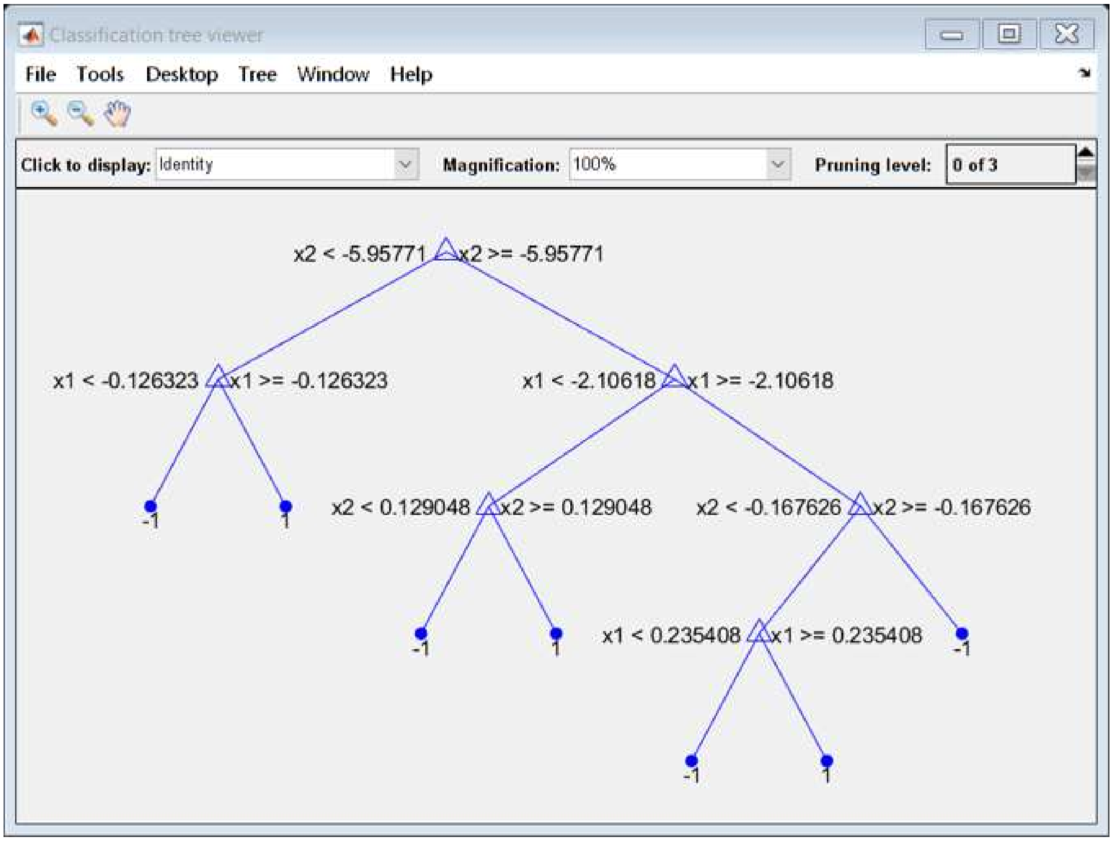
```
|===================================================================================================================================|
|   Iteration  | Set  |   Set Size   |  Feasibility  |     Delta     |      KKT      |  Number of   |   Objective   |   Constraint  |
|              |      |              |      Gap      |    Gradient   |   Violation   |  Supp. Vec.  |               |   Violation   |
|===================================================================================================================================|
|            0 |active|          400 |  9.999975e-01 |  2.000000e+00 |  1.000000e+00 |            0 |  0.000000e+00 |  0.000000e+00 |
|           61 |active|          400 |  1.475263e-01 |  9.455347e-04 |  5.715393e-04 |           19 | -1.299826e+01 |  4.996004e-16 |

Exiting Active Set upon convergence due to DeltaGradient.
```



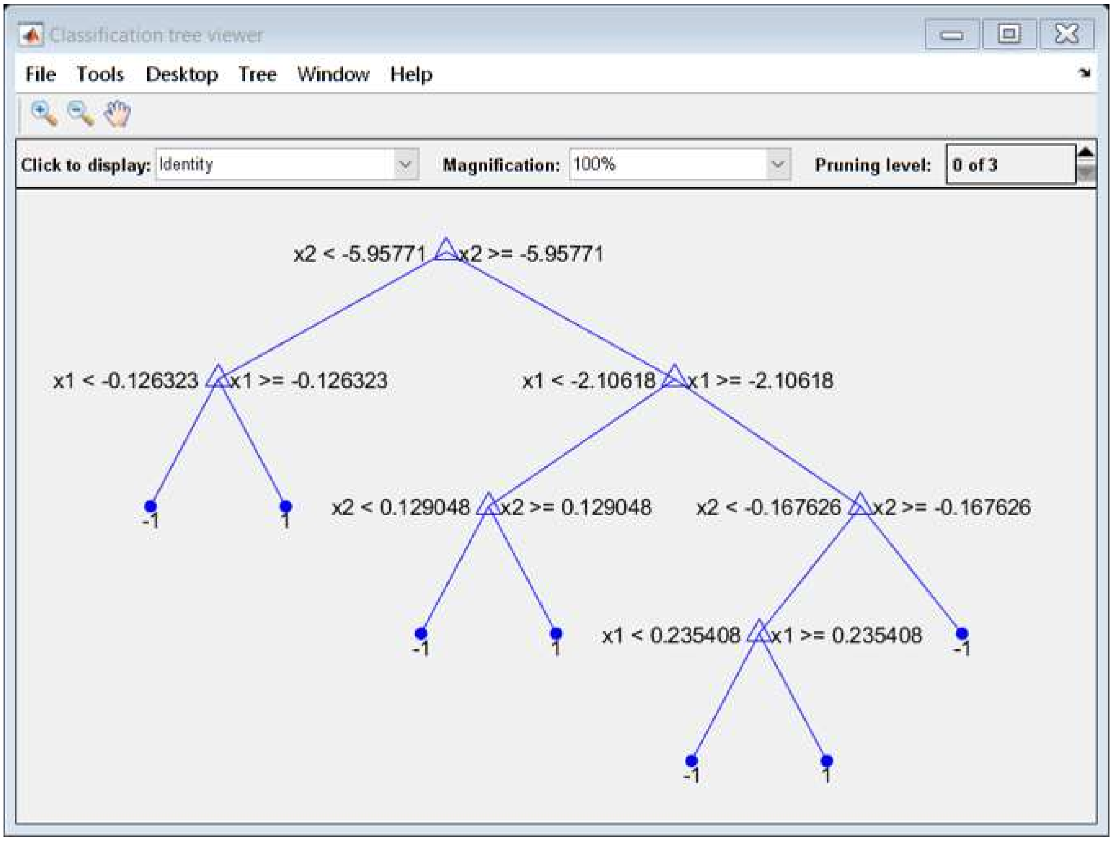
Decision Tree Classification

```
function decisionTree(x1,y1,x2,y2)
    rng(1);
    tree = fitctree(x1, y1, 'Prune', 'off','PruneCriterion','impurity');
    tree_pruned = prune(tree);

    view(tree,'Mode','graph');
    view(tree_pruned,'Mode','graph');

    test_err = loss(tree,x2,y2);
    train_err = loss(tree,x1,y1);

    test_err_p = loss(tree_pruned,x2,y2);
    train_err_p = loss(tree_pruned,x1,y1);

    fprintf('Testing Error without Pruning:  %f\n', test_err);
    fprintf('Training Error without Pruning: %f\n', train_err);
    fprintf('Testing Error with Pruning:     %f\n', test_err_p);
    fprintf('Training Error with Pruning:    %f\n', train_err_p);
end
```

```
Testing Error without Pruning:  0.005000
Training Error without Pruning: 0.000000
Testing Error with Pruning:     0.005000
Training Error with Pruning:    0.000000
```



Click to display: Identity
Magnification: 100%
Pruning level: 0 of 3



Click to display: Identity
Magnification: 100%
Pruning level: 0 of 3

Functions Received From Textbook

```
The following functions were received from the Textbook
Pattern Recognition - Theodoridis, Koutroumbas
```

Data Generation Class

Received from page 244 of the text

```
function [x,y]=data_generator(m,s,N)
    S = s*eye(2);
    [~,c] = size(m);
    x = []; % Creating the training set
    for i = 1:c
        x = [x mvnrnd(m(:,i)',S,N)'];
    end
    y=[ones(1,N) ones(1,N) -ones(1,N) -ones(1,N)];
end
```

Published with MATLAB® R2019b