Support Vector Machine

Concept and matlab build

Kan Xie
ECE 480
Team 4

Table of Contents

- Introduction

Support vector machine is a machine learning method that is widely used for data analyzing and pattern recognizing. The algorithm was invented by Vladimir Vapnik and the current standard incarnation was proposed by Corinna Cortes and Vladimir Vapnik. This application note is to helping understand the concept of support vector machine and how to build a simple support vector machine using Matlab

- Concept

Classifying data has been one of the major parts in machine learning. The idea of support vector machine is to create a hyper plane in between data sets to indicate which class it belongs to. The challenge is to train the machine to understand structure from data and mapping with the right class label, for the best result, the hyper plane has the largest distance to the nearest training data points of any class.

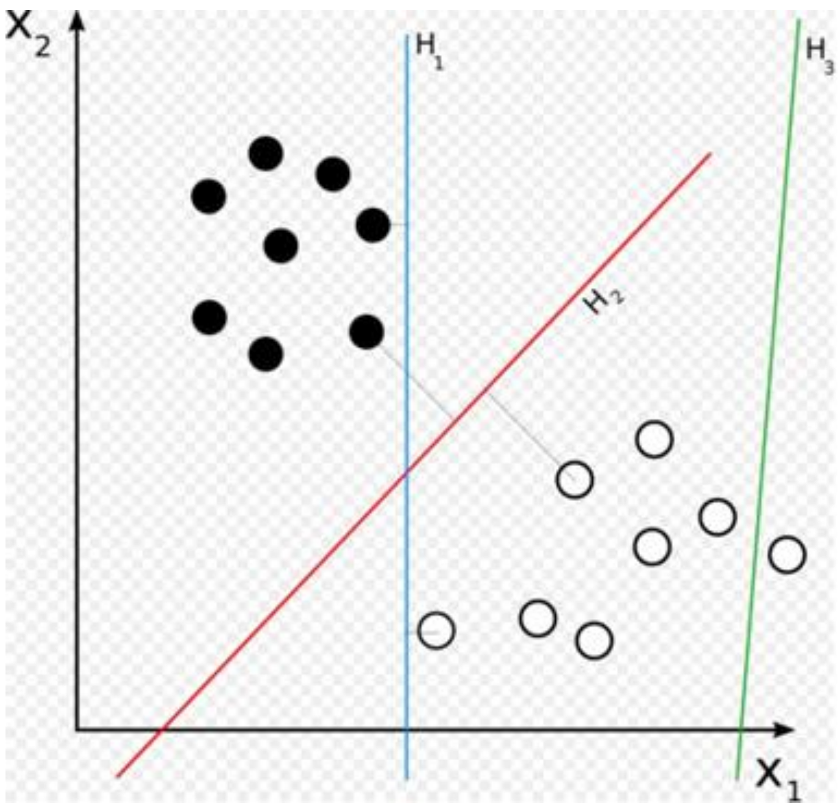


Figure 1 – Hyper Plane

(As we can see from figure 1, H3 does not separate the two classes while H1 separate the two class with a small margin, only H2 gives a maximum margin between two classes, therefor it's the right hyper plane used by support vector machine)

However, instead define a function for the hyper plane itself; we define the margin in between the two classes. From figure 2, we can see that the position of our hyper plane is depend on the value of W.

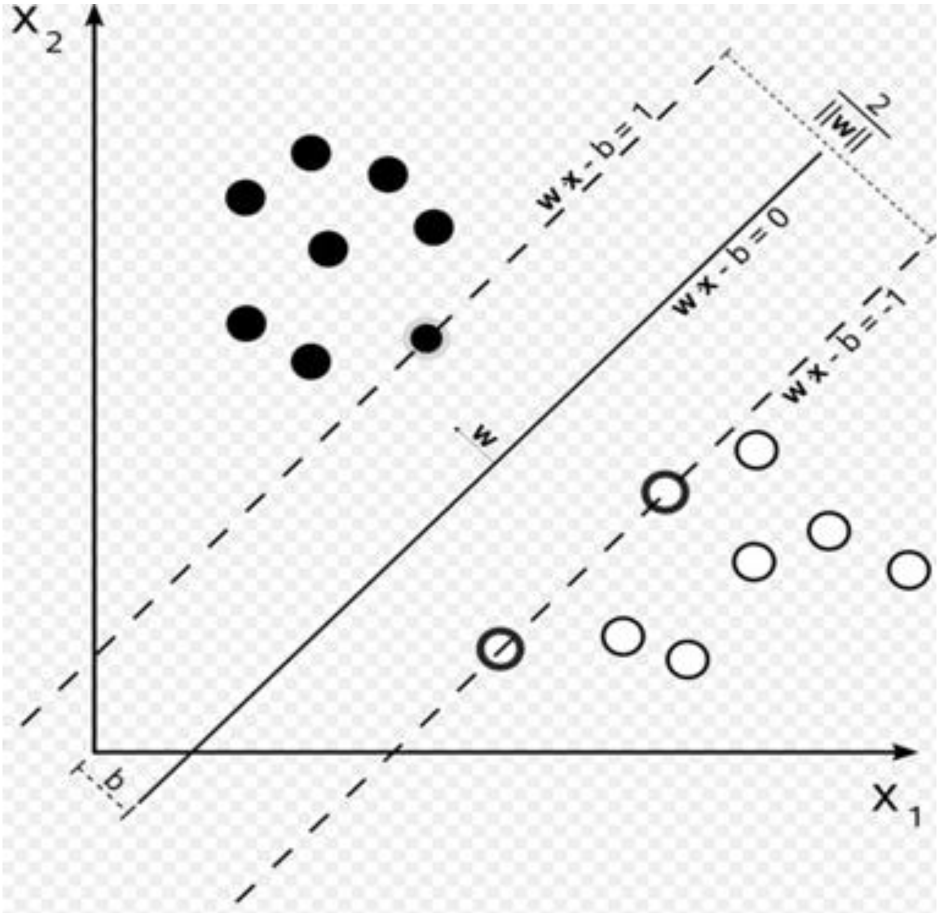


Figure 2

- Build a simple support vector machine using Matlab

1. Load the sample data

```
load dataname
```

2. Create `data`, a two-column matrix containing sepal length and sepal width measurements for 150 irises.

```
data = [meas(:,1), meas(:,2)];
```

3. From the `species` vector, create a new column vector, `groups`, to classify data into two groups: data and non-data.

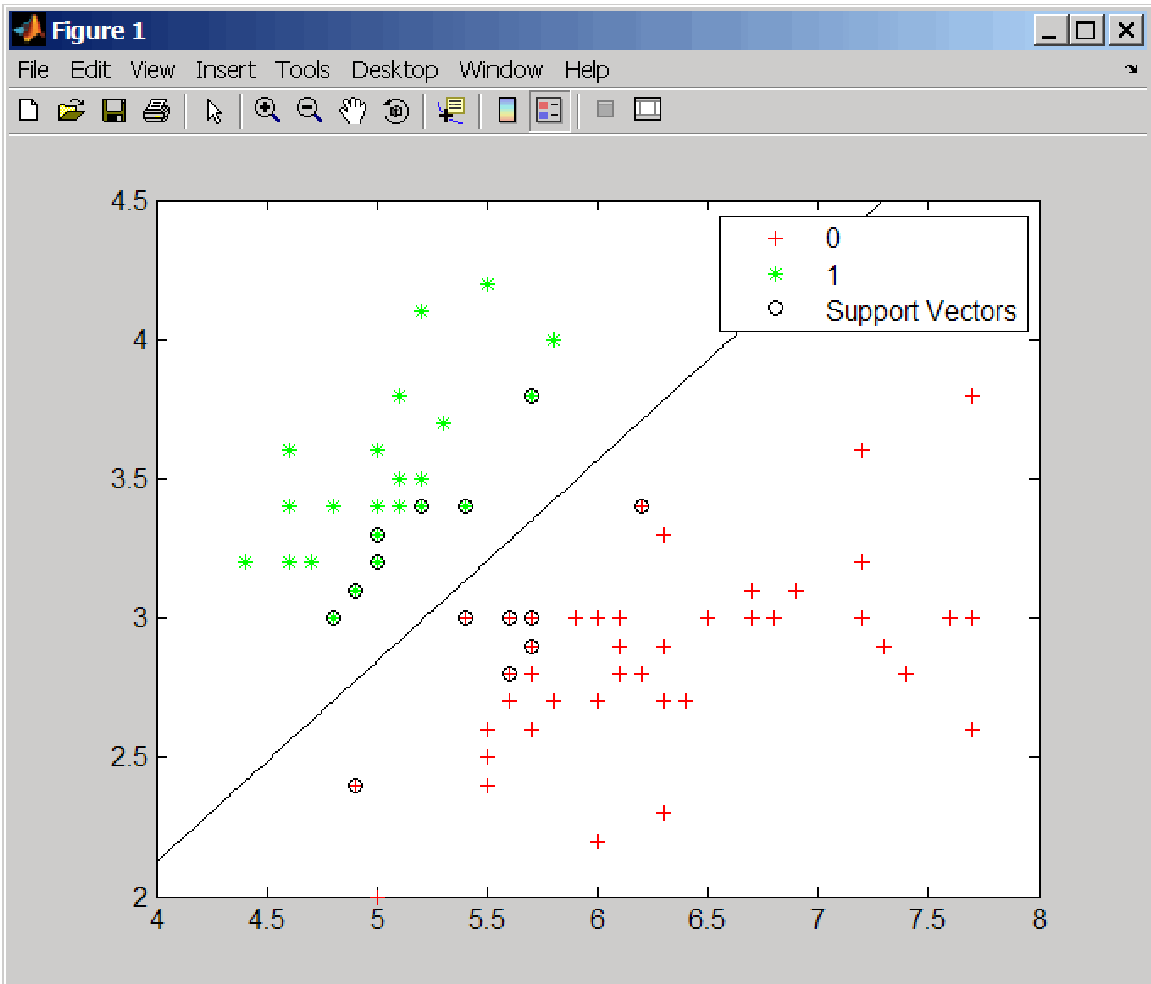
```
groups = ismember(dataset,'data');
```

4. Randomly select training and test sets.

```
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
```

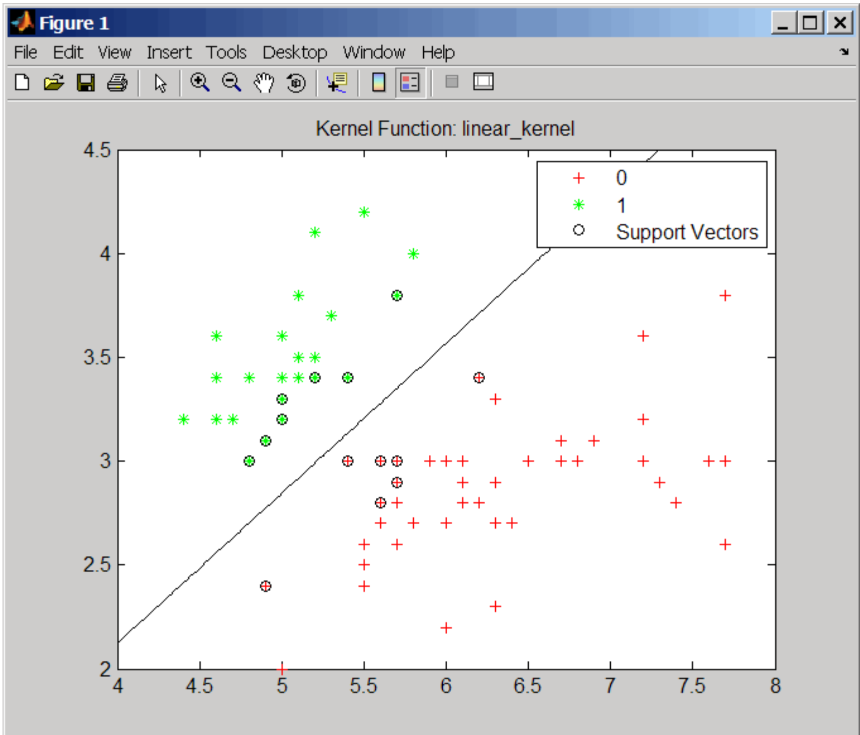
5. Train an SVM classifier using a linear kernel function and plot the grouped data.

```
svmStruct = svmtrain(data(train,:),groups(train),'showplot',true);
```



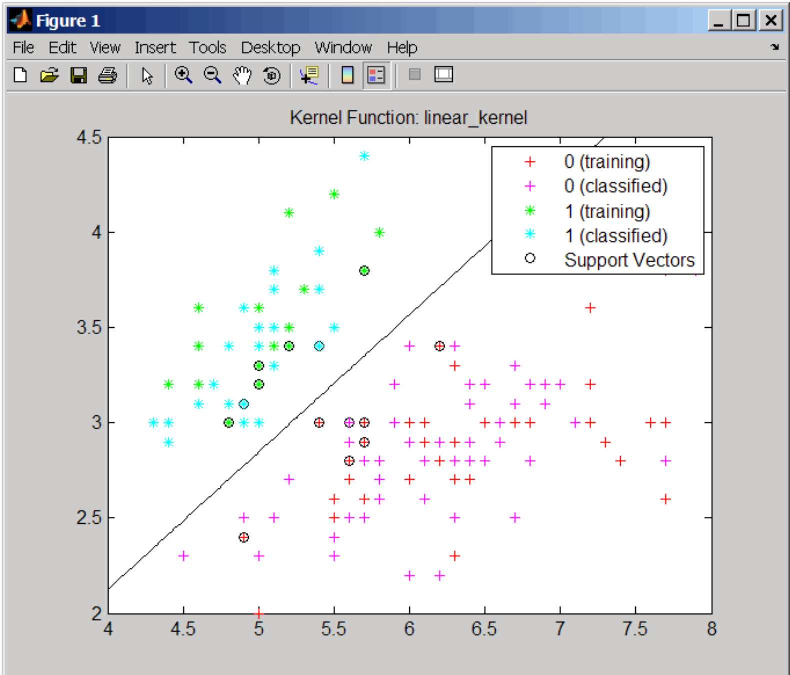
6. Add a title to the plot, using the `KernelFunction` field from the `svmStruct` structure as the title.

```
title(sprintf('Kernel Function: %s',...
              func2str(svmStruct.KernelFunction)),...
              'interpreter','none');
```



7. Use the svmclassify function to classify the test set.

```
classes = svmclassify(svmStruct,data(test,:),'showplot',true);
```

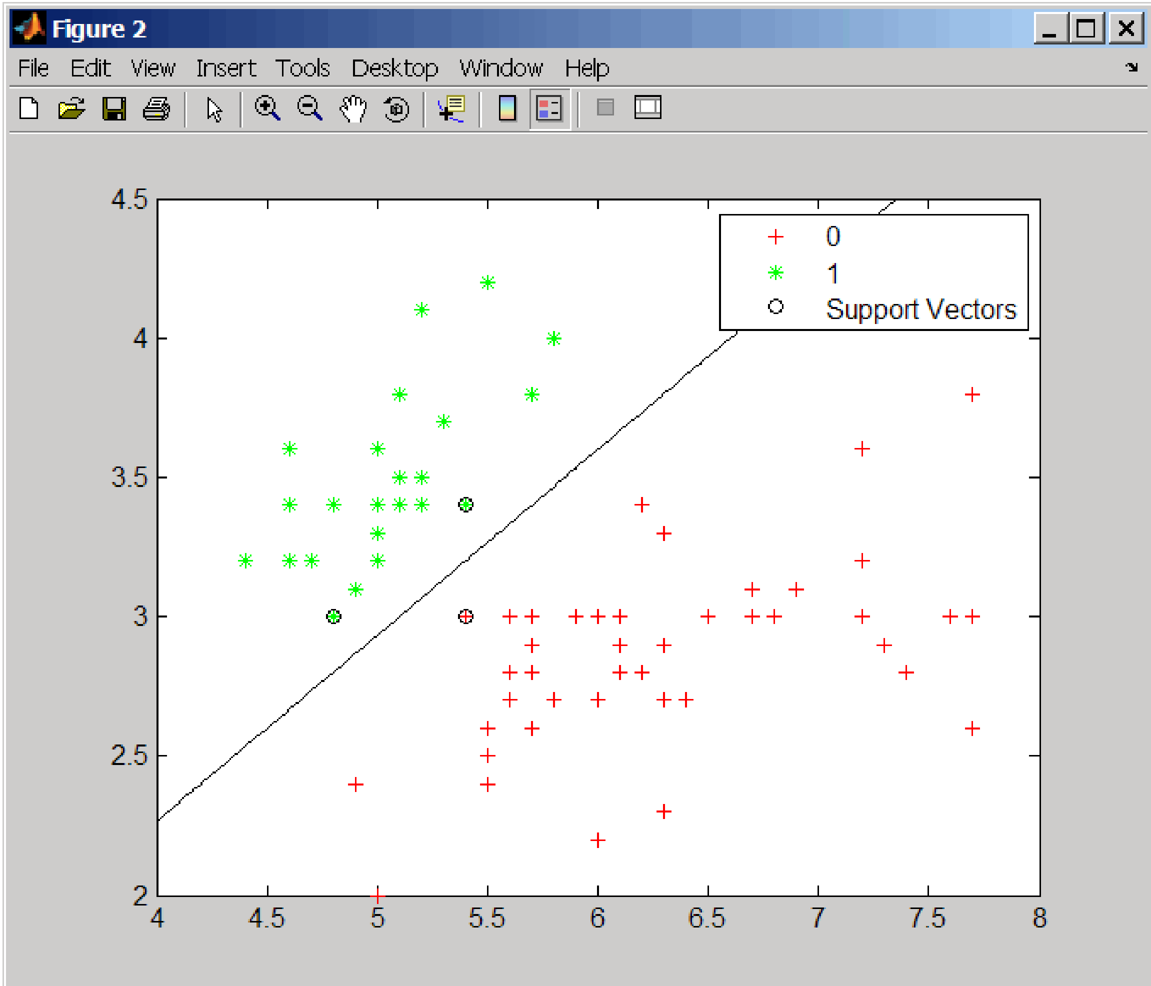


8. Evaluate the performance of the classifier.

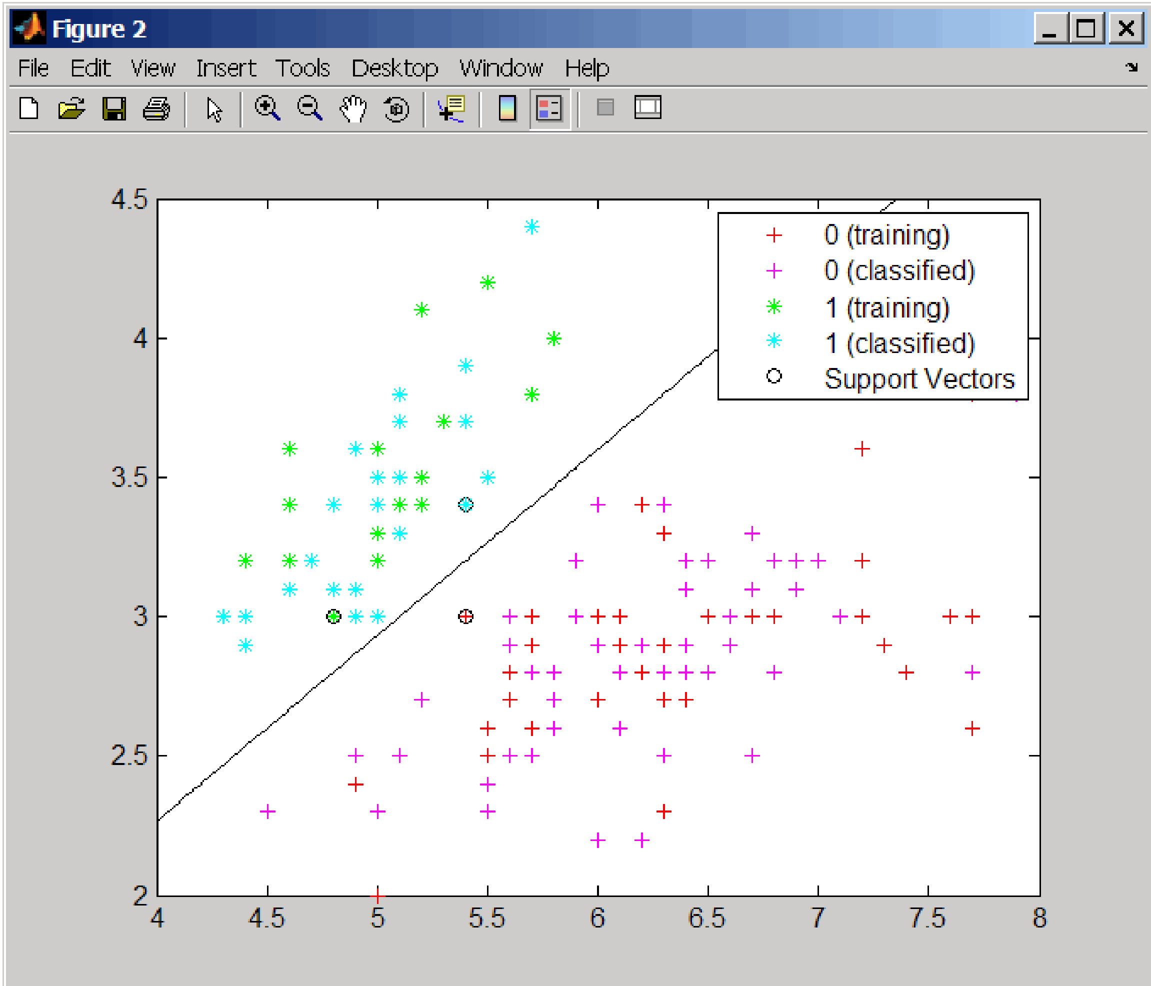
```
classperf(cp,classes,test);
cp.CorrectRate

ans =

    0.9867
```

9. Use a one-norm, hard margin support vector machine classifier by changing the `boxconstraint` property.

```
figure
svmStruct = svmtrain(data(train,:),groups(train),...
                     'showplot',true,'boxconstraint',1e6);
```




```
classes = svmclassify(svmStruct,data(test,:),'showplot',true);
```



10. Evaluate the performance of the classifier.

```
classperf(cp,classes,test);
cp.CorrectRate

ans =

    0.9867
```

- **Conclusion**

This application not went over the basic ideas of support vector machine and how to build a simple support vector machine using matlab functions, this guide is not intend to deal with complex and non-liner object with multiple attributes. However, such task can be done within matlab, please check our final design project for using support vector machine to determine object class after running Histogram of Oriented Gradients algorithm on image data base.

- Reference

. Mathworks “Train support vector machine classifier”.

<http://www.mathworks.com/help/toolbox/bioinfo/ref/svmtrain.html> (4/6/2011)

. “Support vector machine”.

<http://en.wikipedia.org/wiki/Support_vector_machine> (4/6/2011)

. Jason Weston, “Support Vector Machine (and Statistical Learning Theory) Tutorial”, NEC Labs America.