**Nearest Neighbor Classifiers**

Load the iris.txt data into Matlab, and select the first two data features only for the moment.

You should first permute the data so that it is not in sorted order.

iris = load ('iris .txt ');

pi = randperm ( size (iris ,1));

Y= iris (pi ,5); X= iris (pi ,1:2);

1. Plot the data by their feature values, using the class value to select the color. The easiest way to do this is to use find to identify indices for which *Y* takes on each of its possible values. You can use unique to see what values *Y* contains. You can then plot each class in turn, using hold on to plot on top of the previous plot. (When you don’t want this any more, use hold off.

iris=load('iris.txt');

pi=randperm(size(iris,1));%reorderdatarandomly

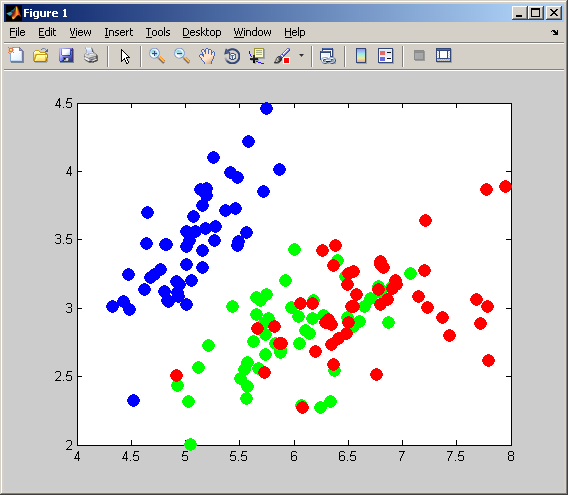
Y=iris(pi,5);X=iris(pi,1:2);%getclass&features

i0=find(Y==0);i1=find(Y==1);i2=find(Y==2);%getclassindexes

figure;

plot(X(i0,1),X(i0,2),'b.',X(i1,1),X(i1,2),'g.',...%ploteachclass

X(i2,1),X(i2,2),'r.','markersize',30);%inadifferentcolor



1. Use the provided knnClassify class to learn a 1-nearest-neighbor predictor. Use the function

class2DPlot(learner,X,Y) to plot the decision regions and training data together.

for **k=[2,9,10,53],**

learner = knnClassify(k,X,Y);

class2DPlot(learner,X,Y);

title(sprintf('K=%d',k));

end ;

1. Do the same thing for several values of *k* (say, [1*,* 3*,* 10*,* 30]) and comment on their appearance.



1. Now split the data into an 80/20 training/validation split. For *k* = [1*,* 2*,* 5*,* 10*,* 50*,* 100*,* 200], learn a model on the 80% and calculate its performance (# of data classified incorrectly) on the validation data. What value of *k* appears to generalize best given your training data? Comment on the performance at the two endpoints, in terms of over- or under-fitting.

**kvalues=[3,4,6,7,8,12,28,46,122,300];**

Ntrain=round(0.8\_size(X,1));

Xtr=X(1:Ntrain,:);Xva=X(Ntrain+1:end,:);

Ytr=Y(1:Ntrain,:);Yva=Y(Ntrain+1:end,:);

for i=1:length(kvalues)

learner=knnClassify(kvalues(i),Xtr,Ytr);%constructlearner

Yhat=predict(learner,Xva);%predictontrainingdata

err(i)=mean(Yhat~=Yva);%count#ofwrongpredictions

end;

figure;semilogx(kvalues,err,'b-','linewidth',3);

set(gca,'fontsize',18);



