%load the data set

>> data=load('iris.txt')

%echo the input

data =

5.1000 3.5000 1.4000 0.2000 1.0000

4.9000 3.0000 1.4000 0.2000 1.0000

4.7000 3.2000 1.3000 0.2000 1.0000

4.6000 3.1000 1.5000 0.2000 1.0000

5.0000 3.6000 1.4000 0.2000 1.0000

5.4000 3.9000 1.7000 0.4000 1.0000

4.6000 3.4000 1.4000 0.3000 1.0000

5.0000 3.4000 1.5000 0.2000 1.0000

4.4000 2.9000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

5.4000 3.7000 1.5000 0.2000 1.0000

4.8000 3.4000 1.6000 0.2000 1.0000

4.8000 3.0000 1.4000 0.1000 1.0000

4.3000 3.0000 1.1000 0.1000 1.0000

5.8000 4.0000 1.2000 0.2000 1.0000

5.7000 4.4000 1.5000 0.4000 1.0000

5.4000 3.9000 1.3000 0.4000 1.0000

5.1000 3.5000 1.4000 0.3000 1.0000

5.7000 3.8000 1.7000 0.3000 1.0000

5.1000 3.8000 1.5000 0.3000 1.0000

5.4000 3.4000 1.7000 0.2000 1.0000

5.1000 3.7000 1.5000 0.4000 1.0000

4.6000 3.6000 1.0000 0.2000 1.0000

5.1000 3.3000 1.7000 0.5000 1.0000

4.8000 3.4000 1.9000 0.2000 1.0000

5.0000 3.0000 1.6000 0.2000 1.0000

5.0000 3.4000 1.6000 0.4000 1.0000

5.2000 3.5000 1.5000 0.2000 1.0000

5.2000 3.4000 1.4000 0.2000 1.0000

4.7000 3.2000 1.6000 0.2000 1.0000

4.8000 3.1000 1.6000 0.2000 1.0000

5.4000 3.4000 1.5000 0.4000 1.0000

5.2000 4.1000 1.5000 0.1000 1.0000

5.5000 4.2000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

5.0000 3.2000 1.2000 0.2000 1.0000

5.5000 3.5000 1.3000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

4.4000 3.0000 1.3000 0.2000 1.0000

5.1000 3.4000 1.5000 0.2000 1.0000

5.0000 3.5000 1.3000 0.3000 1.0000

4.5000 2.3000 1.3000 0.3000 1.0000

4.4000 3.2000 1.3000 0.2000 1.0000

5.0000 3.5000 1.6000 0.6000 1.0000

5.1000 3.8000 1.9000 0.4000 1.0000

4.8000 3.0000 1.4000 0.3000 1.0000

5.1000 3.8000 1.6000 0.2000 1.0000

4.6000 3.2000 1.4000 0.2000 1.0000

5.3000 3.7000 1.5000 0.2000 1.0000

5.0000 3.3000 1.4000 0.2000 1.0000

7.0000 3.2000 4.7000 1.4000 2.0000

6.4000 3.2000 4.5000 1.5000 2.0000

6.9000 3.1000 4.9000 1.5000 2.0000

5.5000 2.3000 4.0000 1.3000 2.0000

6.5000 2.8000 4.6000 1.5000 2.0000

5.7000 2.8000 4.5000 1.3000 2.0000

6.3000 3.3000 4.7000 1.6000 2.0000

4.9000 2.4000 3.3000 1.0000 2.0000

6.6000 2.9000 4.6000 1.3000 2.0000

5.2000 2.7000 3.9000 1.4000 2.0000

5.0000 2.0000 3.5000 1.0000 2.0000

5.9000 3.0000 4.2000 1.5000 2.0000

6.0000 2.2000 4.0000 1.0000 2.0000

6.1000 2.9000 4.7000 1.4000 2.0000

5.6000 2.9000 3.6000 1.3000 2.0000

6.7000 3.1000 4.4000 1.4000 2.0000

5.6000 3.0000 4.5000 1.5000 2.0000

5.8000 2.7000 4.1000 1.0000 2.0000

6.2000 2.2000 4.5000 1.5000 2.0000

5.6000 2.5000 3.9000 1.1000 2.0000

5.9000 3.2000 4.8000 1.8000 2.0000

6.1000 2.8000 4.0000 1.3000 2.0000

6.3000 2.5000 4.9000 1.5000 2.0000

6.1000 2.8000 4.7000 1.2000 2.0000

6.4000 2.9000 4.3000 1.3000 2.0000

6.6000 3.0000 4.4000 1.4000 2.0000

6.8000 2.8000 4.8000 1.4000 2.0000

6.7000 3.0000 5.0000 1.7000 2.0000

6.0000 2.9000 4.5000 1.5000 2.0000

5.7000 2.6000 3.5000 1.0000 2.0000

5.5000 2.4000 3.8000 1.1000 2.0000

5.5000 2.4000 3.7000 1.0000 2.0000

5.8000 2.7000 3.9000 1.2000 2.0000

6.0000 2.7000 5.1000 1.6000 2.0000

5.4000 3.0000 4.5000 1.5000 2.0000

6.0000 3.4000 4.5000 1.6000 2.0000

6.7000 3.1000 4.7000 1.5000 2.0000

6.3000 2.3000 4.4000 1.3000 2.0000

5.6000 3.0000 4.1000 1.3000 2.0000

5.5000 2.5000 4.0000 1.3000 2.0000

5.5000 2.6000 4.4000 1.2000 2.0000

6.1000 3.0000 4.6000 1.4000 2.0000

5.8000 2.6000 4.0000 1.2000 2.0000

5.0000 2.3000 3.3000 1.0000 2.0000

5.6000 2.7000 4.2000 1.3000 2.0000

5.7000 3.0000 4.2000 1.2000 2.0000

5.7000 2.9000 4.2000 1.3000 2.0000

6.2000 2.9000 4.3000 1.3000 2.0000

5.1000 2.5000 3.0000 1.1000 2.0000

5.7000 2.8000 4.1000 1.3000 2.0000

6.3000 3.3000 6.0000 2.5000 3.0000

5.8000 2.7000 5.1000 1.9000 3.0000

7.1000 3.0000 5.9000 2.1000 3.0000

6.3000 2.9000 5.6000 1.8000 3.0000

6.5000 3.0000 5.8000 2.2000 3.0000

7.6000 3.0000 6.6000 2.1000 3.0000

4.9000 2.5000 4.5000 1.7000 3.0000

7.3000 2.9000 6.3000 1.8000 3.0000

6.7000 2.5000 5.8000 1.8000 3.0000

7.2000 3.6000 6.1000 2.5000 3.0000

6.5000 3.2000 5.1000 2.0000 3.0000

6.4000 2.7000 5.3000 1.9000 3.0000

6.8000 3.0000 5.5000 2.1000 3.0000

5.7000 2.5000 5.0000 2.0000 3.0000

5.8000 2.8000 5.1000 2.4000 3.0000

6.4000 3.2000 5.3000 2.3000 3.0000

6.5000 3.0000 5.5000 1.8000 3.0000

7.7000 3.8000 6.7000 2.2000 3.0000

7.7000 2.6000 6.9000 2.3000 3.0000

6.0000 2.2000 5.0000 1.5000 3.0000

6.9000 3.2000 5.7000 2.3000 3.0000

5.6000 2.8000 4.9000 2.0000 3.0000

7.7000 2.8000 6.7000 2.0000 3.0000

6.3000 2.7000 4.9000 1.8000 3.0000

6.7000 3.3000 5.7000 2.1000 3.0000

7.2000 3.2000 6.0000 1.8000 3.0000

6.2000 2.8000 4.8000 1.8000 3.0000

6.1000 3.0000 4.9000 1.8000 3.0000

6.4000 2.8000 5.6000 2.1000 3.0000

7.2000 3.0000 5.8000 1.6000 3.0000

7.4000 2.8000 6.1000 1.9000 3.0000

7.9000 3.8000 6.4000 2.0000 3.0000

6.4000 2.8000 5.6000 2.2000 3.0000

6.3000 2.8000 5.1000 1.5000 3.0000

6.1000 2.6000 5.6000 1.4000 3.0000

7.7000 3.0000 6.1000 2.3000 3.0000

6.3000 3.4000 5.6000 2.4000 3.0000

6.4000 3.1000 5.5000 1.8000 3.0000

6.0000 3.0000 4.8000 1.8000 3.0000

6.9000 3.1000 5.4000 2.1000 3.0000

6.7000 3.1000 5.6000 2.4000 3.0000

6.9000 3.1000 5.1000 2.3000 3.0000

5.8000 2.7000 5.1000 1.9000 3.0000

6.8000 3.2000 5.9000 2.3000 3.0000

6.7000 3.3000 5.7000 2.5000 3.0000

6.7000 3.0000 5.2000 2.3000 3.0000

6.3000 2.5000 5.0000 1.9000 3.0000

6.5000 3.0000 5.2000 2.0000 3.0000

6.2000 3.4000 5.4000 2.3000 3.0000

5.9000 3.0000 5.1000 1.8000 3.0000

**%generate a sample for training**

>> sampleindex=randint(1,20,150) % this generates an array of 20 indices in the range 1 to 150

%(or maybe 0 to 150?)

% see the indices of the data to be used as training

sampleindex =

Columns 1 through 12

115 30 58 82 34 96 72 22 117 15 44 35

Columns 13 through 20

79 13 60 15 16 117 43 90

**%eliminate duplicate**

>> uniquesampleindex=unique(sampleindex)

uniquesampleindex =

Columns 1 through 12

13 15 16 22 30 34 35 43 44 58 60 72

Columns 13 through 18

79 82 90 96 115 117

**% extract the data points to be used for training**

>> sample=data(uniquesampleindex, :)

sample =

4.8000 3.0000 1.4000 0.1000 1.0000

5.8000 4.0000 1.2000 0.2000 1.0000

5.7000 4.4000 1.5000 0.4000 1.0000

5.1000 3.7000 1.5000 0.4000 1.0000

4.7000 3.2000 1.6000 0.2000 1.0000

5.5000 4.2000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

4.4000 3.2000 1.3000 0.2000 1.0000

5.0000 3.5000 1.6000 0.6000 1.0000

4.9000 2.4000 3.3000 1.0000 2.0000

5.2000 2.7000 3.9000 1.4000 2.0000

6.1000 2.8000 4.0000 1.3000 2.0000

6.0000 2.9000 4.5000 1.5000 2.0000

5.5000 2.4000 3.7000 1.0000 2.0000

5.5000 2.5000 4.0000 1.3000 2.0000

5.7000 3.0000 4.2000 1.2000 2.0000

5.8000 2.8000 5.1000 2.4000 3.0000

6.5000 3.0000 5.5000 1.8000 3.0000

>> [rsample, c]= size(sample);

>> rsample

rsample =

18

**% change labels 2 and 3 to -1**

>> for i=1:rsample, if (sample(i,5)==2 || sample(i,5)==3), sample(i,5)=-1; end, end

>> sample

**%check**

sample =

4.8000 3.0000 1.4000 0.1000 1.0000

5.8000 4.0000 1.2000 0.2000 1.0000

5.7000 4.4000 1.5000 0.4000 1.0000

5.1000 3.7000 1.5000 0.4000 1.0000

4.7000 3.2000 1.6000 0.2000 1.0000

5.5000 4.2000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

4.4000 3.2000 1.3000 0.2000 1.0000

5.0000 3.5000 1.6000 0.6000 1.0000

4.9000 2.4000 3.3000 1.0000 -1.0000

5.2000 2.7000 3.9000 1.4000 -1.0000

6.1000 2.8000 4.0000 1.3000 -1.0000

6.0000 2.9000 4.5000 1.5000 -1.0000

5.5000 2.4000 3.7000 1.0000 -1.0000

5.5000 2.5000 4.0000 1.3000 -1.0000

5.7000 3.0000 4.2000 1.2000 -1.0000

5.8000 2.8000 5.1000 2.4000 -1.0000

6.5000 3.0000 5.5000 1.8000 -1.0000

**% sort examples by label**

>> [sortedclass, sortedindex]=sort(sample(:,5));

**% sorted training data is**

>> sortedsample=sample(sortedindex,:)

sortedsample =

4.9000 2.4000 3.3000 1.0000 -1.0000

5.2000 2.7000 3.9000 1.4000 -1.0000

6.1000 2.8000 4.0000 1.3000 -1.0000

6.0000 2.9000 4.5000 1.5000 -1.0000

5.5000 2.4000 3.7000 1.0000 -1.0000

5.5000 2.5000 4.0000 1.3000 -1.0000

5.7000 3.0000 4.2000 1.2000 -1.0000

5.8000 2.8000 5.1000 2.4000 -1.0000

6.5000 3.0000 5.5000 1.8000 -1.0000

4.8000 3.0000 1.4000 0.1000 1.0000

5.8000 4.0000 1.2000 0.2000 1.0000

5.7000 4.4000 1.5000 0.4000 1.0000

5.1000 3.7000 1.5000 0.4000 1.0000

4.7000 3.2000 1.6000 0.2000 1.0000

5.5000 4.2000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

4.4000 3.2000 1.3000 0.2000 1.0000

5.0000 3.5000 1.6000 0.6000 1.0000

**% split training data into negative examples**

>> negsample=sortedsample(find(sortedsample(:,5)== -1),:)

negsample =

4.9000 2.4000 3.3000 1.0000 -1.0000

5.2000 2.7000 3.9000 1.4000 -1.0000

6.1000 2.8000 4.0000 1.3000 -1.0000

6.0000 2.9000 4.5000 1.5000 -1.0000

5.5000 2.4000 3.7000 1.0000 -1.0000

5.5000 2.5000 4.0000 1.3000 -1.0000

5.7000 3.0000 4.2000 1.2000 -1.0000

5.8000 2.8000 5.1000 2.4000 -1.0000

6.5000 3.0000 5.5000 1.8000 -1.0000

**% extract positive examples**

>> possample=sortedsample(find(sortedsample(:,5)== +1),:)

possample =

4.8000 3.0000 1.4000 0.1000 1.0000

5.8000 4.0000 1.2000 0.2000 1.0000

5.7000 4.4000 1.5000 0.4000 1.0000

5.1000 3.7000 1.5000 0.4000 1.0000

4.7000 3.2000 1.6000 0.2000 1.0000

5.5000 4.2000 1.4000 0.2000 1.0000

4.9000 3.1000 1.5000 0.1000 1.0000

4.4000 3.2000 1.3000 0.2000 1.0000

5.0000 3.5000 1.6000 0.6000 1.0000

**%extract size of positive examples; rpos is the number of positive examples in the training data**

[rpos, colpos]=size(possample);

**% START FindS**

**% set S as a cell array of size rpos, 1: S{i} will contain S after the ith positive example was used.**

>> S=cell(rpos,1);

**% S set is initialized to the first positive example**

>> S{1}=possample(1,1:4)

S =

[1x4 double]

[]

[]

[]

[]

[]

[]

[]

[]

[]

**% just checking the 2nd pos example**

>> possample(2,1:4)

ans =

5.5000 4.2000 1.4000 0.2000

>> S{1}

ans =

4.7000 3.2000 1.6000 0.2000

**% Now, here I chose to update S by a positive example, e+, is by taking the min of the current S and e+;**

>> S{2}=min(possample(2,1:4), S{1})

S =

[1x4 double]

[1x4 double]

[]

[]

[]

[]

[]

[]

[]

[]

**% repeat for each of the remaining examples:**

>> for i=3:rpos, S{i}=min(possample(i,1:4), S{i-1}); end

**% last element of S contains the bounds on the attribute values:**

>> S{rpos}

ans =

4.4000 3.0000 1.2000 0.1000

**% NOTE: By taking the minimum of the attribute values, we find the most specific description consistent with the %positive examples!**

**% Furthermore, this description is NOT among the data points:**

>> Sindex=0; for i=1:150, if data(i,1:4)==[4.4000 3.0000 1.2000 0.1000], Sindex=i; end, end

>> Sindex

Sindex =

0

**% The remaining data is test data**

>>testindex=setdiff(1:150, uniquesampleindex);

>> length(testindex)

ans =

132

>> testdata=data(testindex,:);

>> [rtest, ctest]=size(testdata)

**% relabel clases 2 and 3**

>> for i=1:rtest, if (testdata(i,1:5) == 2|| testdata(i,1:5) == 3), testdata(I,1:5)=-1; end, end

**% labels assigned to the test data by the algorithm**

>> for i=1:rtest, if testdata(i,1:4)**>=**S{rpos}, testlabel(i)=1; else testlabel(i)=-1; end; end

**%find agreement between the labels assigned and the true labels of the test data**

>> agree=find (testdata(:,5)==testlabel');

>> numagree=length(agree)

numagree =

128

>> accuracy=numagree/rtest

accuracy =

0.9697 % pretty good accuracy!!!!

>>Percentage = accuracy\*100

Percentage =

96.9697

%CONCLUSION: to describe in words this concept, based on these training examples (note how few positive examples

% we have, only 9) we would say, **“if the 1st attribute >= 4.4000, the 2nd Attribute >= 3.0000 , 3rd Attribute >= 1.2000,**

**%and 4th Attribute >= 0.1000 , then the instance for which these inequalities hold is a positive instance, a setosa iris**

**%flower”**