Supervised learning – Classification (Demo)

Linear classifiers, Fisher's LDA, on > 2 classes

STAT 32950-24620

Spring 2023 (4/27)

Classification data and objectives

Classification — a type of **supervised learning**

Classifier: $R^p \rightarrow \{1, 2, \cdots, g\}$, g the number of classes.

The numerical values of the classes often are not meaningful.

```
library(MASS)  # to use lda function
```


Example (iris)

Example: classical iris data

```
data(iris)
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.
##  $ Species     : Factor w/ 3 levels "setosa","versicolo
```

```
colnames(iris) = c("SL", "SW", "PL","PW","Species")
```


Data summary

```
summary(iris)
```

```
##        SL              SW              PL              PW
##  Min.   :4.30   Min.   :2.00   Min.   :1.00   Min.   :0.
##  1st Qu.:5.10   1st Qu.:2.80   1st Qu.:1.60   1st Qu.:0.
##  Median :5.80   Median :3.00   Median :4.35   Median :1.
##  Mean   :5.84   Mean   :3.06   Mean   :3.76   Mean   :1.
##  3rd Qu.:6.40   3rd Qu.:3.30   3rd Qu.:5.10   3rd Qu.:1.
##  Max.   :7.90   Max.   :4.40   Max.   :6.90   Max.   :2.
```

Note: Variables are of comparable magnitude and spread;

therefore can be used without scaling (normalizing).

Choose feature variables

Choose feature variables to be used as predictors in classification.

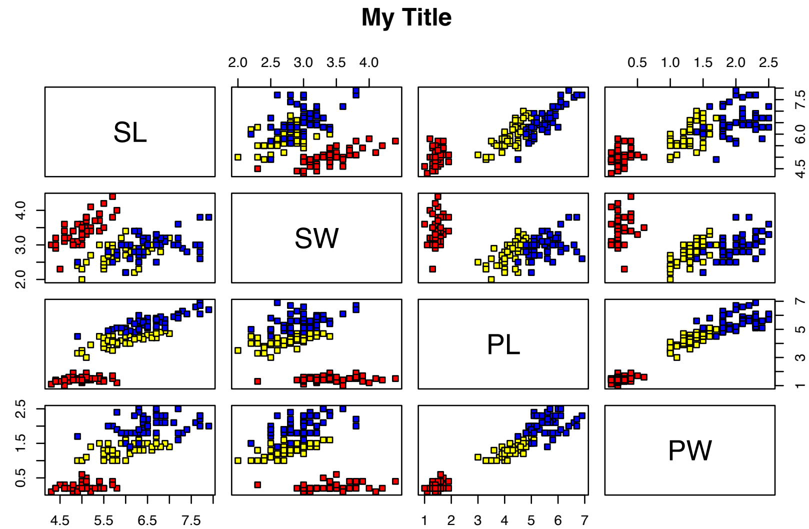
```
attach(iris)
X=iris[,1:4] #Feature var's, used for classification
```

$$\Longrightarrow \qquad p = 4, \quad g = 3$$

Classifier: $R^4 \to \{1, 2, 3\}$

Data pairwise plots

```
pairs(X, main="My Title ", pch=22,
      bg=c("red", "yellow", "blue")[unclass(Species)])
```



Sample covariance matrices

```
levels(Species)=c(1:3)
S1=cov(iris[Species==1,1:4])
# S1=cov(subset(iris[,1:4], Species==1)) # same
S2=cov(iris[Species==2,1:4])
S3=cov(iris[Species==3,1:4])
```

S_k — Sample covariance matrix of sub-population k

```
Sp=(50-1)*(S1+S2+S3)/(150-3)
```

Under equal subpopulation covariance assumption,

$$S_{pooled} = \frac{1}{n-g}\left[(n_1 - 1)S_1 + \cdots (n_g - 1)S_g\right]$$

where

$$n = n_1 + \cdots + n_g$$

Class covariance matrices S_1, S_2, S_3

```
round(S1,2); round(S2,2); round(S3,2)
```

```
##      SL   SW   PL   PW
## SL 0.12 0.10 0.02 0.01
## SW 0.10 0.14 0.01 0.01
## PL 0.02 0.01 0.03 0.01
## PW 0.01 0.01 0.01 0.01

##      SL   SW   PL   PW
## SL 0.27 0.09 0.18 0.06
## SW 0.09 0.10 0.08 0.04
## PL 0.18 0.08 0.22 0.07
## PW 0.06 0.04 0.07 0.04

##      SL   SW   PL   PW
## SL 0.40 0.09 0.30 0.05
## SW 0.09 0.10 0.07 0.05
## PL 0.30 0.07 0.30 0.05
## PW 0.05 0.05 0.05 0.08
```

Pooled covariance matrix

Pooled covariance matrix S_{pooled} and its inverse S_{pooled}^{-1}

```
round(Sp,2)
```

```
##      SL   SW   PL   PW
## SL 0.27 0.09 0.17 0.04
## SW 0.09 0.12 0.06 0.03
## PL 0.17 0.06 0.19 0.04
## PW 0.04 0.03 0.04 0.04
```

```
round(solve(Sp),2)
```

```
##       SL    SW    PL    PW
## SL 10.84 -5.38 -8.99  3.42
## SW -5.38 14.23  2.67 -8.91
## PL -8.99  2.67 14.79 -8.91
## PW  3.42 -8.91 -8.91 36.77
```

Discussion: Computational cost of inverse matrix.

Fisher's linear discriminants

Goal: Find maximum separation directions for the three classes.

By-product: Form classification regions for 3 classes (by LDA).

Assumption: Equal variance-covariance structure for all classes. (normality not required)

The directions are given by eigenvectors e_i,

$W^{-1}Be_i = \lambda_i e_i$,

scaled by $e_i' S_{pool} e_i = 1$ in R (`lda`).

`lda` in R

Try out the function `lda` in `R`

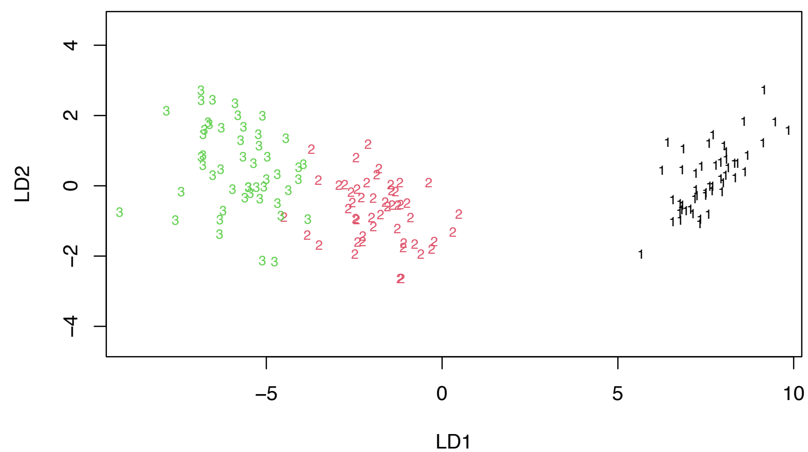
First, use all observations as training data.

```
fit = lda(X,Species) #lda(Species~X[,1]+X[,2]+X[,3]+X[,4])
attributes(fit) # prior,counts,means,scaling,lev,svd,N
```

```
## $names
## [1] "prior"   "counts"  "means"   "scaling" "lev"     "s
## [8] "call"
##
## $class
## [1] "lda"
```

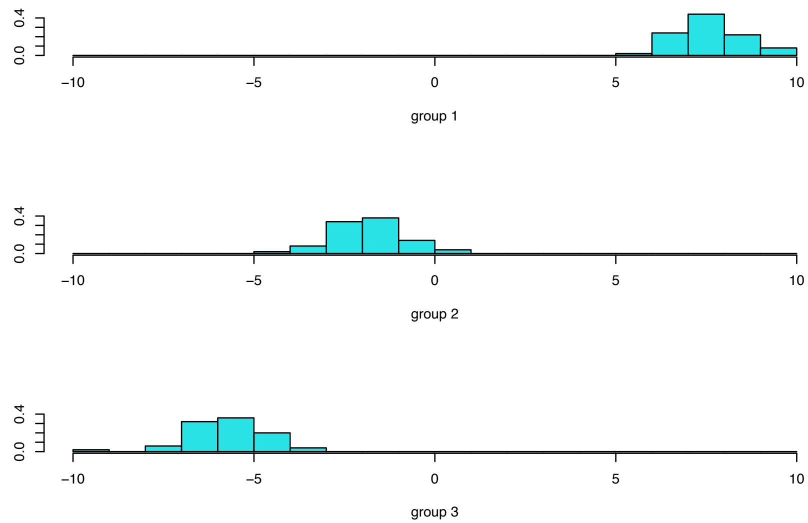

Predicted class membership in (LD1, LD2)

```
plot(fit,col=rep(1:3,each=50))
```



Project to the 1st discriminant line

```r
plot(fit,dimen=1)
```



Linear discriminant properties and normalization

$$Y_i = a_i' X, \quad var(Y_i) = a_i' S_{pool} a_i = 1$$

```r
fit$scaling  # matrix [a_1 a_2]
```

```
##         LD1     LD2
## SL  0.8294  0.0241
## SW  1.5345  2.1645
## PL -2.2012 -0.9319
## PW -2.8105  2.8392
```

```r
t(fit$scaling)%*%Sp%*%fit$scaling # normalization a'S a
```

```
##             LD1        LD2
## LD1  1.000e+00 -7.216e-16
## LD2 -8.327e-16  1.000e+00
```


Posterior probability of membership

$$p(\pi_i | x_o) = \frac{p_i \hat{f}_i(x_o)}{p_1 \hat{f}_1(x_o) + \cdots + p_g \hat{f}_g(x_o)}$$

```r
postprob = round(predict(fit,X)$posterior,3)
attributes(predict(fit,X))
```

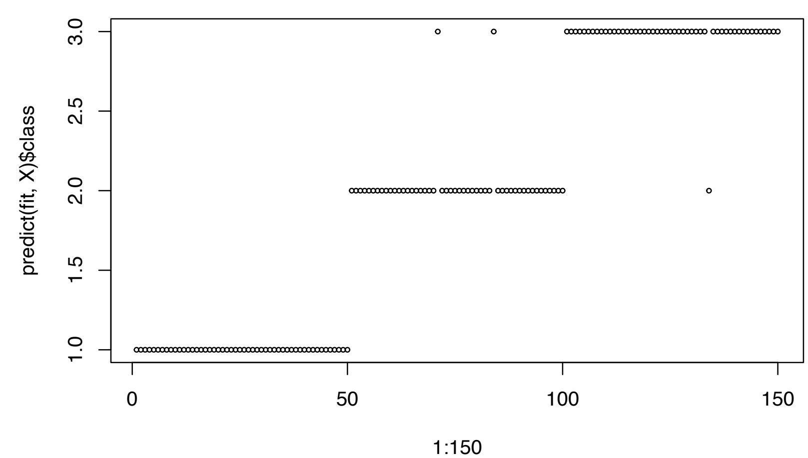
```
## $names
## [1] "class"     "posterior" "x"
```

```r
attributes(postprob)$dim
```

```
## [1] 150   3
```


Classification (and misclassification) by LDA

```r
par(mfrow=c(1,1))
plot(1:150,predict(fit,X)$class, cex=.5)  #err case 71,84,.
```



Misclassification case details

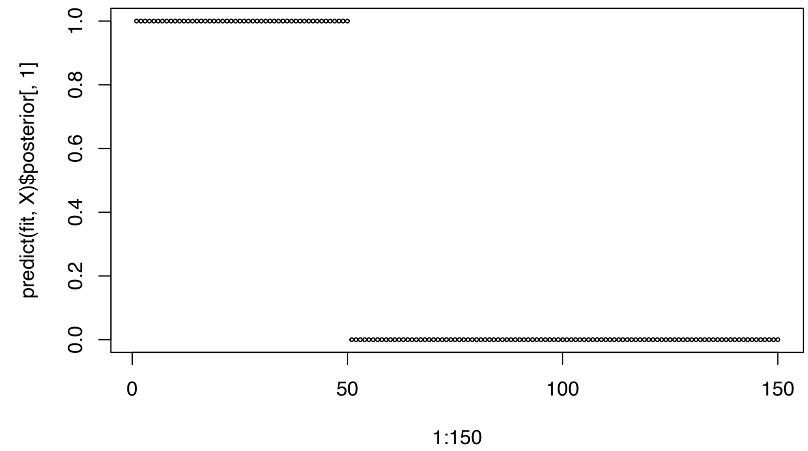
Misclassification	True species	Assigned species	Posterior $P(\pi_1 :$ *given* $x)$	Posterior $P(\pi_2 :$ *given* $x)$	Posterior $P(\pi_3 :$ *given* $x)$
Item 71	2	3	0	0.253	**0.747**
Item 84	2	3	0	0.143	**0.857**
Item 134	3	2	0	**0.729**	0.271

```
# data for the above table
cbind(postprob[c(71,84,134),],Species[c(71,84,134)])
```

```
##      1     2     3
## [1,] 0 0.253 0.747 2
## [2,] 0 0.143 0.857 2
## [3,] 0 0.729 0.271 3
```

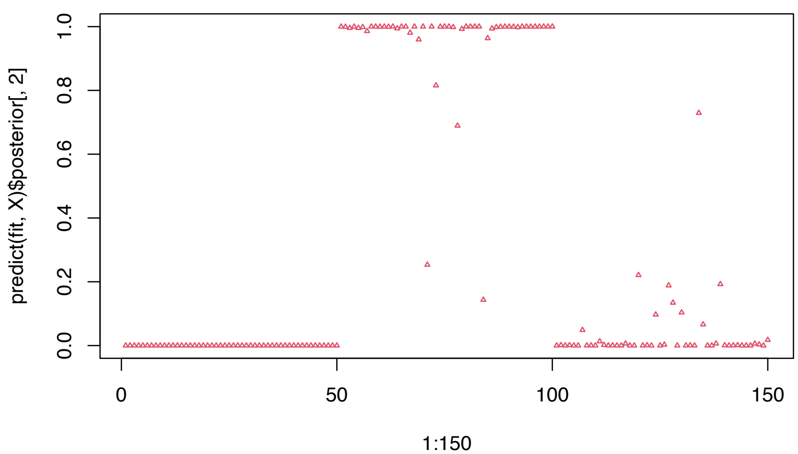

Posterior probability for class 1

```
par(mfrow=c(1,1))
plot(1:150,predict(fit,X)$posterior[,1], cex=.4) #class1
```



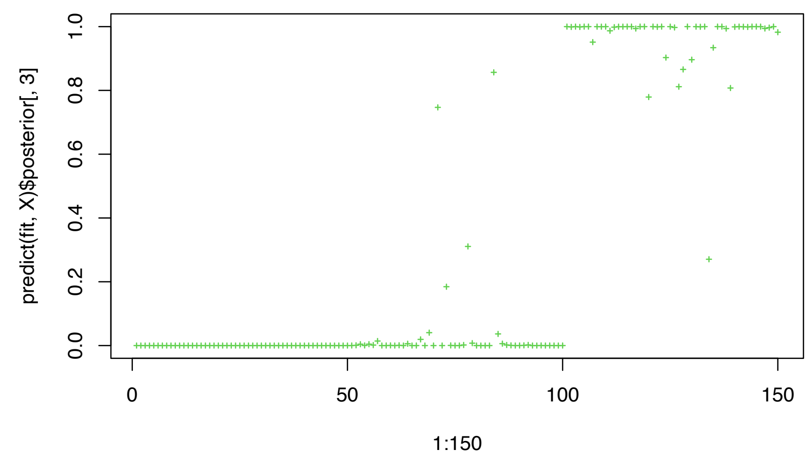
Posterior probability for class 2

```
plot(1:150,predict(fit,X)$posterior[,2],
     cex=.4,col=2,pch=2) #class2
```

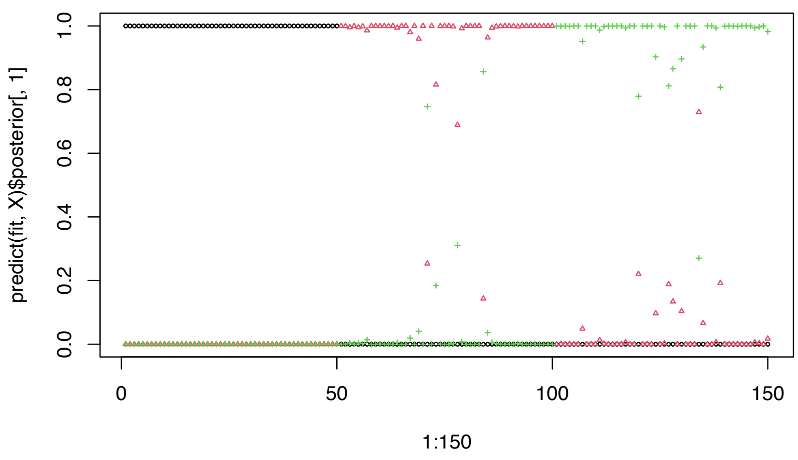


Posterior probability for class 3

```
plot(1:150,predict(fit,X)$posterior[,3],
     cex=.4,col=3,pch=3) #class3
```



Posterior probability for all



Classification proportions

Percentage of classification within each species

```
ct=table(Species, predict(fit,X)$class) #cross-count table
prop.table(ct, 1)   # (., 1) row %; (., 2) col %
```

```
##
## Species    1    2    3
##       1 1.00 0.00 0.00
##       2 0.00 0.96 0.04
##       3 0.00 0.02 0.98
```

```
diag(prop.table(ct, 1)) #correct classification by species
```

```
##    1    2    3
## 1.00 0.96 0.98
```

```
sum(diag(prop.table(ct))) #total % of correct assignments
```

```
## [1] 0.98
```


Training error — Apparent Error Rate (APER)

Table of misclassification counts based on training data

```
#ct <- table(Species, predict(fit,X)$class)
ct
```

```
##
## Species  1  2  3
##       1 50  0  0
##       2  0 48  2
##       3  0  1 49
```

Apparent Error Rate (APER) = Error rate within the training data

$$\frac{n_{1m} + n_{2m} + n_{3m}}{n_1 + n_2 + n_3} = 0.02$$

where n_{im} = misclassified members of class i

Expected Actual Error Rate E(AER)

The Expected Actual Error Rate E(AER) can be estimated by

Holdout-one cross validation

```
fitCV = lda(X,Species, CV=T)
# or lda(Species~X[,1]+X[,2]+X[,3]+X[,4], CV=T)
```

Misclassification Table by holdout-one method:

```
table(Species, fitCV$class)
```

```
##
## Species  1  2  3
##       1 50  0  0
##       2  0 48  2
##       3  0  1 49
```

Estimated E(AER) by corss validation

Expected actual error rate can be estimated

by holdout-one **cross validation**:

$$\hat{E}(AER) = \frac{0+2+1}{50+50+50} = 0.02$$

This is the same as APER! Well, it happens.

Most time, test or valication error > training err.

Use only two variables as predictors?

```
fit12 = lda(Species~X[,1]+X[,2])
```

Table of classification and misclassification

```
table(Species, predict(fit12,X[,1:2])$class)
```

```
##
## Species  1  2  3
##        1 49  1  0
##        2  0 36 14
##        3  0 15 35
```

Apparent error rate (training error)

$$APER = \frac{14+15}{50+50+50} = \frac{29}{150} = 0.19$$

Validation error rate, two predictors

Use holdout-one cross validation (two variables)

```
fitCV12 = lda(Species~X[,1]+X[,2], CV=T)
table(Species, fitCV12$class)
```

```
##
## Species  1  2  3
##        1 49  1  0
##        2  0 35 15
##        3  0 15 35
```

Estimated Expected Actual Error Rate E(AER)

$$\hat{E}(AER) = \frac{30}{150} = 0.20$$

Now this is more common and realistic:

$$\hat{E}(AER) > APER$$

Use one variable as the predictor?

```
#fit4 = lda(Species~X[,4])
fitCV4 = lda(Species~X[,4], CV=T)
#table(Species, predict(fit4,X[,4])$class)
table(Species, fitCV4$class)
```

```
##
## Species  1  2  3
##        1 50  0  0
##        2  0 48  2
##        3  0  4 46
```

Estimated expected actual error rate

$$\hat{E}(AER) = \frac{6}{150} = 0.04$$

(Some variables are better classifiers than others)

Normal Populations (classification by min ECM)

For $N(\mu_i, \Sigma_i)$:

Classification region $R_k = \{x : d_k(x) \geq d_i(x), \forall i \neq k\}$

The estimated **linear discriminant scores**

(equal-covariance, equal-cost, minimize ECM)

$$\hat{d}_k(x) = \bar{x}_k' S_{pool}^{-1} x - \frac{1}{2}\bar{x}_k' S_{pool}^{-1}\bar{x}_k + \ln(p_k), \qquad k = 1, \cdots, g.$$

Example: Three normal sub-populations

Example: $p = 2, g = 3$. Classifier: $R^2 \to \{1, 2, 3\}$

Using two petal variables as predictors

```
fit34=lda(Species~X[,3]+X[,4])
```

Obtain S_{pool} (2 predictors)

```
s1=cov(iris[Species==1,3:4]);
s2=cov(iris[Species==2,3:4])
s3=cov(iris[Species==3,3:4]);
Sp2=(50-1)*(s1+s2+s3)/(150-3);
Sp2
```

```
##          PL      PW
## PL 0.18519 0.04267
## PW 0.04267 0.04188
```


Find the linear discriminant functions

Get $\bar{x}_k, k = 1, \cdots, g$ $(g = 3)$

```
X34mean=cbind(tapply(X[,3],iris$Species,sum)/50,
              tapply(X[,4],iris$Species,sum)/50)
X34mean
```

```
##             [,1]  [,2]
## setosa     1.462 0.246
## versicolor 4.260 1.326
## virginica  5.552 2.026
```


Find the discriminant functions (cont.)

Get $\bar{x}_k' S_{pool}^{-1}$ in

$$\hat{d}_k(x) = \bar{x}_k' S_{pool}^{-1} x - \frac{1}{2}\bar{x}_k' S_{pool}^{-1}\bar{x}_k + \ln(p_k)$$

```
slp = as.matrix((X34mean)%*%solve(Sp2))
slp
```

```
##                PL     PW
## setosa      8.548 -2.834
## versicolor 20.527 10.749
## virginica  24.612 23.302
```

Find the discriminant functions (cont.)

Get $\frac{1}{2}\bar{x}_k' S_{pool}^{-1}\bar{x}_k$

```
itc = diag((X34mean)%*%solve(Sp2)%*%t(X34mean))/2
itc
```

```
##     setosa versicolor  virginica
##       5.90      50.85      91.93
```

Obtain $\hat{d}_k(x) = \bar{x}_k' S_{pool}^{-1} x - \frac{1}{2}\bar{x}_k' S_{pool}^{-1}\bar{x}_k + \ln(p_k)$,

$$\hat{d}_1 = 8.5x - 2.8y - 5.9 + \log(1/3)$$
$$\hat{d}_2 = 20.5x + 10.7y - 50.9 + \log(1/3)$$
$$\hat{d}_3 = 24.6x + 23.3y - 91.9 + \log(1/3)$$

Intersections of discriminant functions

Set $\hat{d}_1 = \hat{d}_2,\ \hat{d}_2 = \hat{d}_3,\ \hat{d}_3 = \hat{d}_1$

Solve for the intercepts and slopes of the intersection lines.

```
c(slp[1,]-slp[2,],slp[2,]-slp[3,],slp[3,]-slp[1,])
```

```
##      PL      PW      PL      PW      PL      PW
## -11.980 -13.583  -4.085 -12.553  16.064  26.136
```

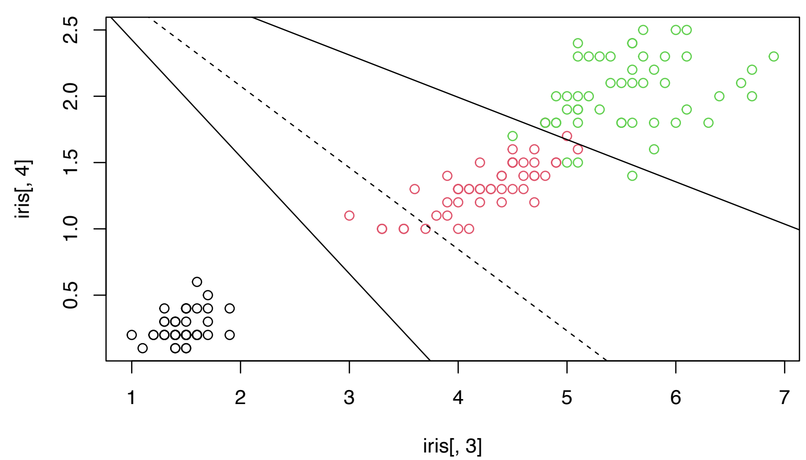
```
c(itc[1]-itc[2],itc[2]-itc[3],itc[3]-itc[1])
```

```
##     setosa versicolor  virginica
##     -44.95     -41.08      86.03
```


Plot the classification borders (code)

```
# Plot the classification borders
plot(iris[,3], iris[,4],col=rep(1:3,each=50))
abline(45/13.6, -12/13.6)  # set d1=d2
abline(41/12.55, -4/12.55) # set d2=d3
abline(86/26,-16/26,lty=2) # set d1=d3 (redundant)
```


Plot the classification borders (plot)



–>

–>