

STAT3014: Sample Prac 4

User: jean

August 29, 2009

Question 1

```
> set.seed(123)
> canine1 = read.table("canine1.dat")[, 1:5]
> canine2 = read.table("canine2.dat")[, 1:5]
```

Calculating the covariance and means of the two groups.

```
> dim(canine1)
```

```
[1] 16 5
```

```
> dim(canine2)
```

```
[1] 20 5
```

```
> m1 = apply(canine1, 2, mean)
> m2 = apply(canine2, 2, mean)
> v1 = var(canine1)
> v2 = var(canine2)
> sp = (15 * v1 + 19 * v2)/34
> sp
```

	V1	V2	V3	V4	V5
V1	36.904206	2.3565000	6.1325662	4.9885588	1.3980294
V2	2.356500	0.4582941	0.5652794	0.6059412	0.2435588
V3	6.132566	0.5652794	3.2634228	1.3296912	0.1046250
V4	4.988559	0.6059412	1.3296912	1.6700588	0.3170882
V5	1.398029	0.2435588	0.1046250	0.3170882	0.4924412

```
> d = matrix(m1 - m2)
> A = solve(sp) %*% d
> A
```

```

      [,1]
V1  0.1237922
V2 -0.3446244
V3 -0.1967731
V4  2.1741676
V5  0.8418376

> t(A) %*% (m1 + m2)/2

```

```

      [,1]
[1,] 64.74534

```

```

> t(A) %*% m1

```

```

      [,1]
[1,] 69.98644

```

```

> t(A) %*% m2

```

```

      [,1]
[1,] 59.50424

```

Rule (A): Allocate a dog with measurements x to the golden jackal family (group 2) if $A^T x < 64.745$. That is $0.12 * V1 - 0.34 * V2 - 0.20 * V3 + 2.17 * V4 + 0.84 * V5 < 64.75$.

You can also calculate FLDA by first calculating the B and W matrix:

```

> W = 15 * v1 + 19 * v2
> W

```

	V1	V2	V3	V4	V5
V1	1254.7430	80.1210	208.50725	169.6110	47.53300
V2	80.1210	15.5820	19.21950	20.6020	8.28100
V3	208.5072	19.2195	110.95637	45.2095	3.55725
V4	169.6110	20.6020	45.20950	56.7820	10.78100
V5	47.5330	8.2810	3.55725	10.7810	16.74300

```

> m = (16 * m1 + 20 * m2)/36
> B = 16 * ((m1 - m) %*% t(m1 - m) + 20 * (m2 - m) %*% t(m2 - m))
> B

```

	V1	V2	V3	V4	V5
[1,]	14471.788	1534.3223	2881.1991	4017.0446	1164.3967
[2,]	1534.322	162.6713	305.4694	425.8936	123.4512
[3,]	2881.199	305.4694	573.6201	799.7564	231.8206
[4,]	4017.045	425.8936	799.7564	1115.0417	323.2105
[5,]	1164.397	123.4512	231.8206	323.2105	93.6871

```
> A1 = as.real(eigen(solve(W) %*% B)$vec[, 1])
> t(A1) %*% (m1 + m2)/2
```

```
      [,1]
[1,] 27.33915
```

```
> A/A1
```

```
      [,1]
V1 2.368228
V2 2.368228
V3 2.368228
V4 2.368228
V5 2.368228
```

Rule (B): Allocate a dog with measurements x to the golden jackal family (group 2) if $A1^T x < 64.745$. That is $0.05 * V1 - 0.15 * V2 - 0.08 * V3 + 0.92 * V4 + 0.36 * V5 < 27.34$.

Notice if you divide rule (A) by 2.36, you get rule (B).

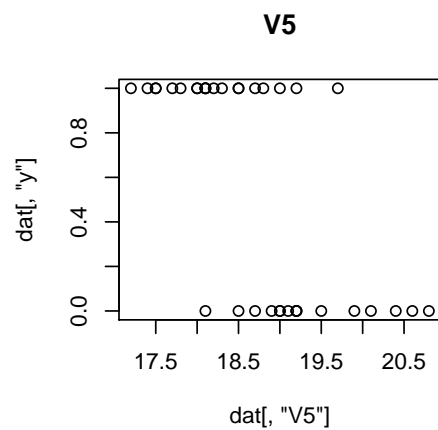
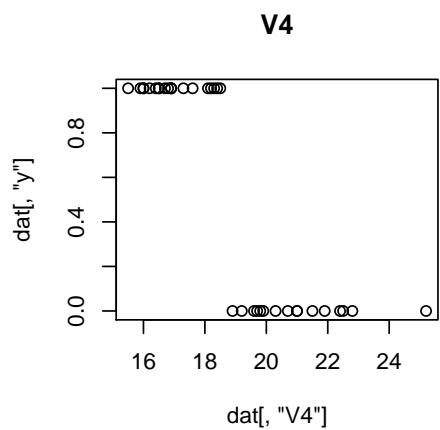
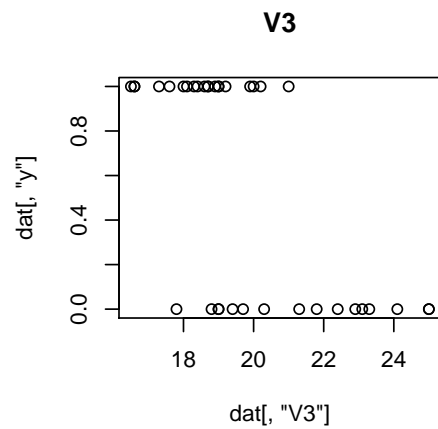
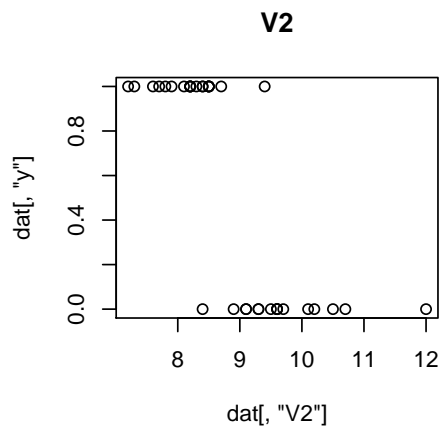
Question 2

```
> sc1 = as.matrix(canine1) %*% A
> sc2 = as.matrix(canine2) %*% A
> scp = c(sc1 >= 64.745, sc2 < 64.745)
> table(scp)
```

```
scp
TRUE
 36
```

Thus there are no misclassified dogs. We estimate the resubstitution error rate to be 0.

```
> dat = rbind(cbind(canine1, y = 0), cbind(canine2, y = 1))
> par(mfrow = c(2, 2))
> plot(dat[, "V2"], dat[, "y"], main = "V2")
> plot(dat[, "V3"], dat[, "y"], main = "V3")
> plot(dat[, "V4"], dat[, "y"], main = "V4")
> plot(dat[, "V5"], dat[, "y"], main = "V5")
```



It is clear that $V4$ is the best variable for separating the two types of dog. There is substantial overlap in the values that the other 3 variables take in the two samples.

Question 3

```
> library(MASS)
> z = lda(factor(dat[, "y"]) ~ ., dat[, -6], prior = c(1, 1)/2)
> A2 = z$scaling
> A2
```

```
LD1
V1 -0.03823555
V2 0.10644369
V3 0.06077706
V4 -0.67153239
V5 -0.26001731
```

```
> t(A2) %*% (m1 + m2)/2

      [,1]
LD1 -19.99781
```

```
> A2/A

      LD1
V1 -0.3088687
V2 -0.3088687
V3 -0.3088687
V4 -0.3088687
V5 -0.3088687
```

Rule (C): Allocate a dog with measurements x to the golden jackal family (group 2) if $A2^T x <$. That is $-0.04 * V1 + 0.11 * V2 + 0.06 * V3 - 0.67 * V4 - 0.26 * V5 > -20$.

If you divide Rule (A) by (-0.309), you will get Rule (C), this shows that the discriminant rule using lda is the same as FLDA.

Question 4

```
> predict(z, dat[, 1:5])$class

 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Levels: 0 1

> table(predict(z, dat[, 1:5])$class, dat[, 6])

   0 1
0 16 0
1  0 20
```

The resubstitution error rate in this case is zero.

Question 5

```
> cdata = rbind(read.table("canine1.dat")[, 1:9], read.table("canine2.dat")[,
+   1:9])
> dim(cdata)

 [1] 36  9

> colnames(cdata)
```

```

[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"

> rownames(cdata)

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"
[16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
[31] "31" "32" "33" "34" "35" "36"

> lab = c(rep("C1", nrow(canine1)), rep("C2", nrow(canine2)))
> cordist = 1 - (cor(t(cdata)))
> dim(cordist)

[1] 36 36

> class(cordist)

[1] "matrix"

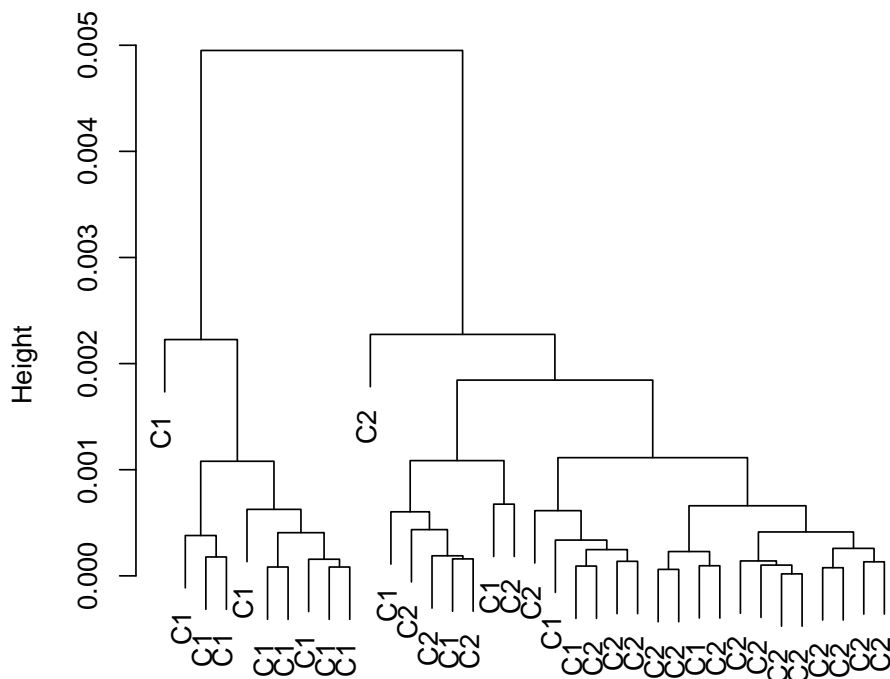
> class(as.dist(cordist))

[1] "dist"

> plot(hclust(as.dist(cordist), method = "complete"), lab = lab)

```

Cluster Dendrogram



```
as.dist(cordist)
hclust (*, "complete")
```

Most of C1 cluster

together except a few which cluster with C2. All C2 belongs to one large cluster.

Note Thank you for pointing this out in class. The correct transform from class "matrix" to class "dist" is `as.dist`. This is illustrated with the simple example below:

```
> X
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	5	9	13	17
[2,]	2	6	10	14	18
[3,]	3	7	11	15	19
[4,]	4	8	12	16	20

```
> cor(X)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	1	1	1	1
[2,]	1	1	1	1	1
[3,]	1	1	1	1	1

```
[4,] 1 1 1 1 1
[5,] 1 1 1 1 1
```

```
> dist(cor(X))
```

```
 1 2 3 4
2 0
3 0 0
4 0 0 0
5 0 0 0 0
```

```
> as.dist(cor(X))
```

```
 1 2 3 4
2 1
3 1 1
4 1 1 1
5 1 1 1 1
```

Additional information

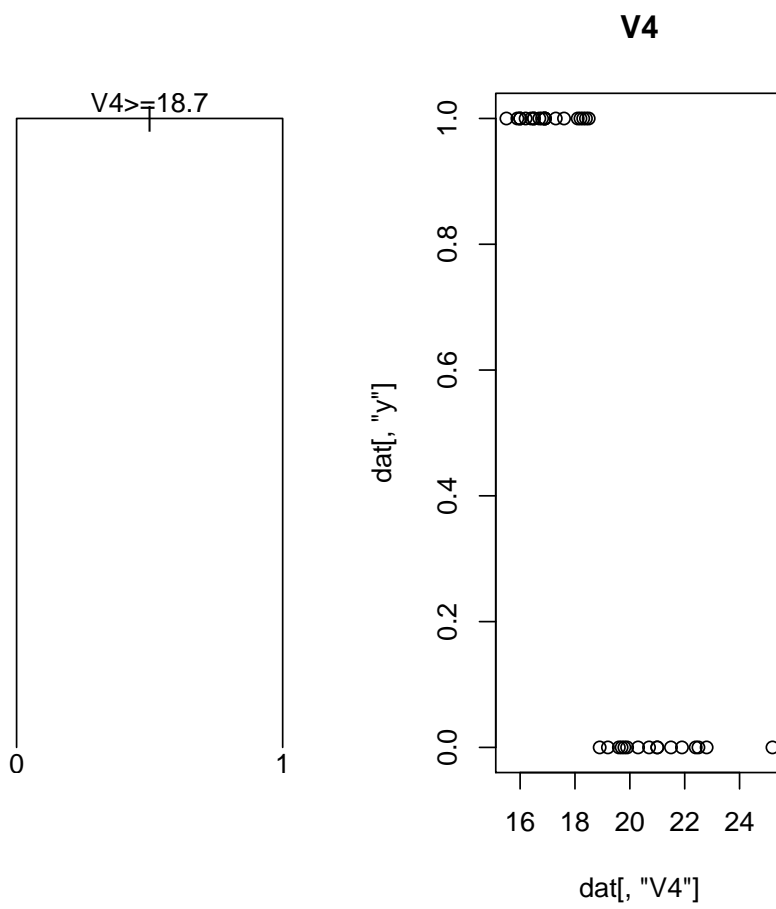
```
> library(rpart)
> par(mfrow = c(1, 2))
> ctree = rpart(factor(dat[, "y"]) ~ ., dat = dat[, -6])
> ctree
```

```
n= 36
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 36 16 1 (0.4444444 0.5555556)
 2) V4>=18.7 16 0 0 (1.0000000 0.0000000) *
 3) V4< 18.7 20 0 1 (0.0000000 1.0000000) *
```

```
> plot(ctree, compress = TRUE)
> text(ctree)
> plot(dat[, "V4"], dat[, "y"], main = "V4")
```



Again, only $V4$ is needed for effective discrimination.