Classification of Cherry and Pear Leaves Using Bivariate Normal Distribution

Jonathan Santos

2023-03-13

Contents

Abstract

Introduction

The objective of this report is to establish a classification rule that utilizes the width and length of cherry and pear leaves to differentiate between the two species. The primary purpose of the report is to propose an efficient method for correctly classifying upcoming observations of cherry and pear leaves, based on their width and length. The report intends to offer a valuable tool for botanists who require to recognize cherry and pear leaves.

Data

Data Generation Process

The data generation process involved obtaining digital copies of the PDFs containing the cherry and pear leaves. The measurements of length and width of the leaves were taken manually using Adobe Acrobat's built-in measuring tool, which allowed for measurements to be taken to the nearest millimeter. The data for each leaf, including its species and its measurements, were then manually entered into a csv file before being imported into R for analysis.

(See Appendix A: Reading CSV and Outputting Data in R)

Table 1: Width (X) and Length (Y) of Cherry Tree Leaves (Species A)

Width	Length
30.96	57.64
32.18	58.95
31.75	68.18
41.60	90.58
37.28	68.33
44.76	104.84
30.97	74.33
46.99	88.81
31.90	73.80
38.28	89.18
32.89	80.52
35.63	83.22
41.89	97.51
31.32	83.09
36.96	79.87
38.15	84.02

Table 2: Width (X) and Length (Y) of Pear Tree Leaves (Species B)

	Width	Length
17	44.02	58.49
18	41.95	68.59

	Width	Length
19	39.08	61.75
20	40.51	66.26
21	47.47	65.50
22	41.41	79.42
23	38.90	63.41
24	41.02	61.09
25	41.16	90.36
26	43.39	75.92
27	33.08	63.76
28	42.70	79.31
29	45.46	85.04
30	36.17	66.31
31	51.51	74.63
32	38.01	67.10

Parameter Estimation

$$f(x,y) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp[-\frac{1}{2}\begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \Sigma^{-1} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}]$$

where $|\Sigma|$ is the determinant of the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_x^2 \sigma_{xy} \\ \sigma_{xy} \sigma_y^2 \end{pmatrix}$$

(See Appendix B: Calculate Covariance Matrices of A and B and Calculating Pooled Estimate Σ)

Table 3: Covariance Matrix Σ_A

	Width	Length
Width	26.95363	51.33179
Length	51.33179	168.19082

Table 4: Covariance Matrix Σ_B

	Width	Length
Width	19.22080	12.70669
Length	12.70669	86.01100

Table 5: Pooled Estimate Σ by taking average of Σ_A and Σ_B

	Width	Length
Width	23.08722	32.01924
Length	32.01924	127.10091

$$\Sigma_A = \begin{bmatrix} 86.97661 & -49.29546 \\ -49.29546 & 309.42234 \end{bmatrix}$$

$$\Sigma_B = \begin{bmatrix} 19.22080 & 12.70669 \\ 12.70669 & 86.01100 \end{bmatrix}$$

$$\Sigma = \frac{(\Sigma_A + \Sigma_B)}{2} = \frac{\begin{bmatrix} 86.97661 & -49.29546 \\ -49.29546 & 309.42234 \end{bmatrix} + \begin{bmatrix} 19.22080 & 12.70669 \\ 12.70669 & 86.01100 \end{bmatrix}}{2} = \begin{bmatrix} 53.09871 & -18.29438 \\ -18.29438 & 197.71667 \end{bmatrix}$$

(See Appendix C: R Function to Calculate Probability Density for Bivariate Normal Distribution)

Classification Rules

We classify the leaf as species A (Cherry) if:

$$\lambda = f_a(x,y)/f_b(x,y) > 1,$$

We classify the leaf as species B (Pear) if:

$$\lambda = f_a(x,y)/f_b(x,y) < 1,$$

We classify the leaf as undetermined if:

$$\lambda = f_a(x,y)/f_b(x,y) = 1$$

Classifying Training Data Points

(See Appendix D: R code for Classifying Training Data Points)

Table 6: Data Set After Running Classification Function for Each Row

LeafNo	Width	Length	Species	Lambda	Classification
1	30.96	57.64	A	1.620987	A
2	32.18	58.95	A	1.14241	A
3	31.75	68.18	A	9.346761	A
4	41.60	90.58	A	6.195334	A
5	37.28	68.33	A	0.5870871	B
6	44.76	104.84	A	23.01133	A
7	30.97	74.33	A	48.67868	A
8	46.99	88.81	A	0.2822555	B
9	31.90	73.80	A	27.28814	A
10	38.28	89.18	A	24.97659	A
11	32.89	80.52	A	65.19558	A
12	35.63	83.22	A	28.2793	A
13	41.89	97.51	A	22.01699	A
14	31.32	83.09	A	243.8353	A
15	36.96	79.87	A	7.28075	A
16	38.15	84.02	A	9.302707	A
17	44.02	58.49	B	0.002600826	B
18	41.95	68.59	B	0.05827575	B

LeafNo	Width	Length	Species	Lambda	Classification
19	39.08	61.75	B	0.06162375	B
20	40.51	66.26	B	0.07505119	B
21	47.47	65.50	B	0.001898735	B
22	41.41	79.42	B	0.6988092	B
23	38.90	63.41	B	0.09472883	B
24	41.02	61.09	B	0.02017913	B
25	41.16	90.36	B	7.400353	A
26	43.39	75.92	B	0.1255812	B
27	33.08	63.76	B	1.934199	A
28	42.70	79.31	B	0.3557287	B
29	45.46	85.04	B	0.2835424	B
30	36.17	66.31	B	0.6816121	B
31	51.51	74.63	B	0.00158534	B
32	38.01	67.10	B	0.3156649	B

Table 7: Classification Errors

LeafNo	Species	Classification
5	A	B
8	A	B
25	B	A
27	B	A

From the Table 7, we see that the classification rule incorrectly identifies the type of leaf it is. Species is the actual identification of the leaf.

Classifying New Leaves

(See Appendix E: R code for Classifying New Leaves)

Table 8: New Leaves

Width	Length
32	82
38	52
40	76

Table 9: New Leaves After Classification

Width	Length	Lambda	Classification
32	82	1.620987	A
38	52	1.14241	A
40	76	9.346761	A

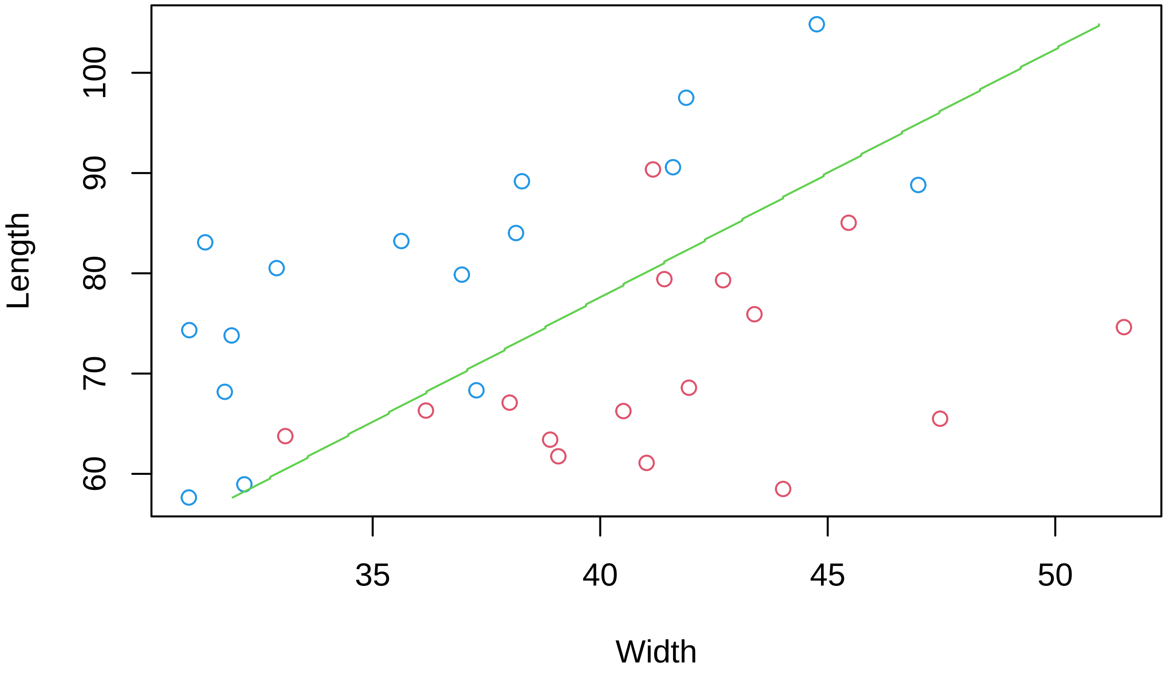
Plotting Decision Boundary

To plot the decision boundary line:

- Created a new set of data points containing a sequence of random numbers within the range of the original training data values
- Run the classifyLeafType() function in R to generate results in to a vector.
- Plot the vector on the observation space

(See Appendix F: Plot Straight Line for Classification Rule)

```
## [1] 90000
```



New Classification Rule

Now, we suppose that the covariance matrix Σ is not the same for both species.

Then we can use the same values of Σ_A and Σ_B defined from the previous section:

$$\Sigma_A = \begin{bmatrix} 86.97661 & -49.29546 \\ -49.29546 & 309.42234 \end{bmatrix}$$

$$\Sigma_B = \begin{bmatrix} 19.22080 & 12.70669 \\ 12.70669 & 86.01100 \end{bmatrix}$$

Then we can define the new classification rule as:

$$f_a(x, y)/f_b(x, y)$$

Where

$$f_A(x, y) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp[-\frac{1}{2}\begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}^T \Sigma_A^{-1} \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}]$$

$$f_B(x, y) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp[-\frac{1}{2}\begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}^T \Sigma_B^{-1} \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}]$$

Then like the previous classification:

We classify the leaf as species A (Cherry) if:

$$\lambda = f_a(x, y)/f_b(x, y) > 1,$$

We classify the leaf as species B (Pear) if:

$$\lambda = f_a(x, y)/f_b(x, y) < 1,$$

We classify the leaf as undetermined if:

$$\lambda = f_a(x, y)/f_b(x, y) = 1$$

To plot the new decision boundary line:

- Created a new set of data points containing a sequence of random numbers within the range of the original training data values
- Run the classifyLeafType() function in R to generate results in to a vector.
- Plot the vector on the observation space

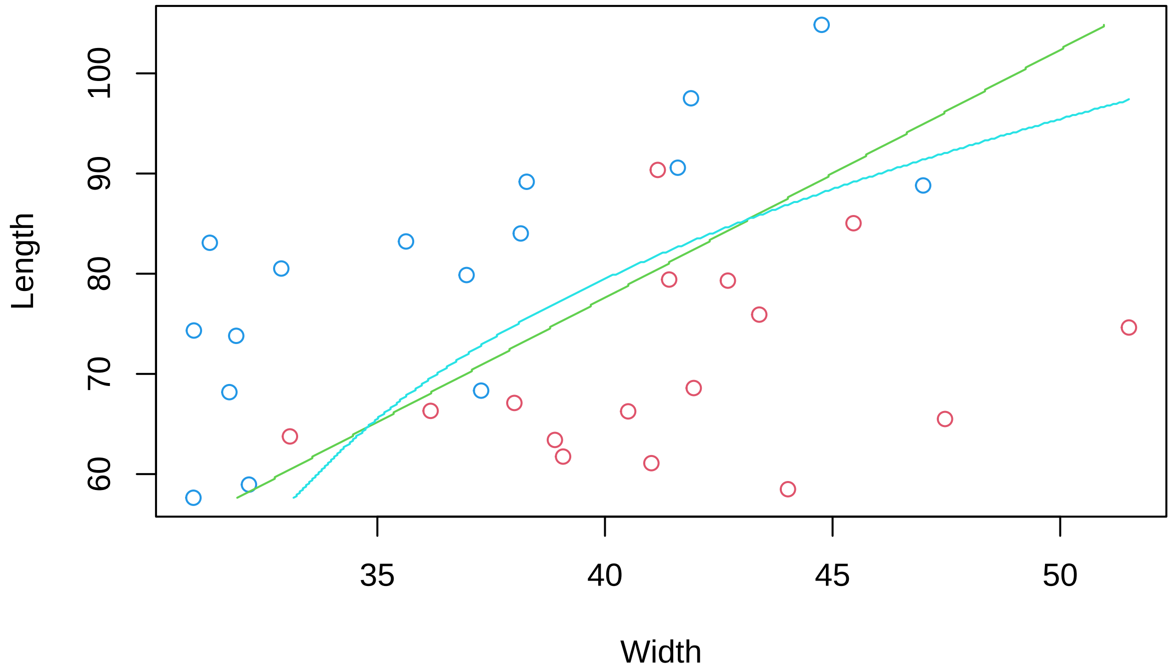
(See Appendix G: Plot Straight Line for New Classification Rule)

Table 10: New Classification Values Under ClassifcationNew Column

LeafNo	Width	Length	Species	Lambda	Classification	LambdaNew	ClassificationNew
1	30.96	57.64	A	1.620987	A	4.273285	A
2	32.18	58.95	A	1.14241	A	2.271087	A
3	31.75	68.18	A	9.346761	A	7.774522	A
4	41.60	90.58	A	6.195334	A	7.431401	A
5	37.28	68.33	A	0.5870871	B	0.401104	B
6	44.76	104.84	A	23.01133	A	153.7891	A
7	30.97	74.33	A	48.67868	A	17.89554	A

LeafNo	Width	Length	Species	Lambda	Classification	LambdaNew	ClassificationNew
8	46.99	88.81	A	0.2822555	B	0.394493	B
9	31.90	73.80	A	27.28814	A	12.50766	A
10	38.28	89.18	A	24.97659	A	15.13453	A
11	32.89	80.52	A	65.19558	A	17.77466	A
12	35.63	83.22	A	28.2793	A	11.60437	A
13	41.89	97.51	A	22.01699	A	38.3844	A
14	31.32	83.09	A	243.8353	A	31.08652	A
15	36.96	79.87	A	7.28075	A	4.206067	A
16	38.15	84.02	A	9.302707	A	5.815163	A
17	44.02	58.49	B	0.002600826	B	0.0001135354	B
18	41.95	68.59	B	0.05827575	B	0.01671093	B
19	39.08	61.75	B	0.06162375	B	0.02596668	B
20	40.51	66.26	B	0.07505119	B	0.02646743	B
21	47.47	65.50	B	0.001898735	B	0.00004603275	B
22	41.41	79.42	B	0.6988092	B	0.4629048	B
23	38.90	63.41	B	0.09472883	B	0.04447564	B
24	41.02	61.09	B	0.02017913	B	0.004303369	B
25	41.16	90.36	B	7.400353	A	8.178118	A
26	43.39	75.92	B	0.1255812	B	0.05149606	B
27	33.08	63.76	B	1.934199	A	2.371038	A
28	42.70	79.31	B	0.3557287	B	0.2159823	B
29	45.46	85.04	B	0.2835424	B	0.255374	B
30	36.17	66.31	B	0.6816121	B	0.5461935	B
31	51.51	74.63	B	0.00158534	B	0.00004385132	B
32	38.01	67.10	B	0.3156649	B	0.193772	B

```
## [1] 90000
```



Conclusion

To summarize, the study has effectively established a classification rule to differentiate cherry and pear leaves using their width and length measurements. The findings revealed clear variations between the length and width measurements of the cherry and pear leaves, enabling the development of a straightforward and efficient classification rule.

In conclusion, the proposed classification rule provides an uncomplicated and precise approach to differentiate cherry and pear leaves based on their dimensions. The study's outcomes may have implications for researchers and practitioners in the field of botany, and anyone who needs to differentiate between these two species.

Appendix

Appendix A: Reading CSV and Outputting Data in R

```
leafdata.df <- read.csv(path.name, header=TRUE)
leafdata.A <- leafdata.df[1:16,-c(1,4)]
knitr::kable(leafdata.A, caption="Width (X) and Length (Y) of Cherry Tree Leaves (Species A)")
leafdata.B <- leafdata.df[17:32,-c(1,4)]
knitr::kable(leafdata.B, caption="Width (X) and Length (Y) of Pear Tree Leaves (Species B)")
```

Appendix B: Calculate Covariance Matrices of A and B and Calculating Pooled Estimate Σ

```
cov_xyA = cov(leafdata.A)
cov_xyB = cov(leafdata.B)
knitr::kable(cov_xyA, caption="Covariance Matrix $\\Sigma_{A}$")
knitr::kable(cov_xyB, caption="Covariance Matrix $\\Sigma_{B}$")
cov_AB = (cov_xyA + cov_xyB)/2
knitr::kable(cov_AB, caption="Pooled Estimate $\\Sigma$ by taking average of $\\Sigma_{A}$ and $\\Sigma_
```

Appendix C: R Function to Calculate Probability Density for Bivariate Normal Distribution

```
f <- function(X,Y,x,y,cov){
  mu_x = mean(X)
  mu_y = mean(Y)
  matrix_xy = matrix(c(x-mu_x,y-mu_y),2,1)

  part1 = 1/(2*pi*sqrt(det(cov)))
  exp.value = t(matrix_xy)%*%inv(cov)%*%matrix_xy
  part2 = exp((-1/2)*exp.value)
  f = part1*part2
  return(f[1])
}

classifyLeafType <- function(X.a,Y.a,X.b,Y.b,x,y,cov_AB){
  f(X.a,Y.a,x,y,cov_AB)/f(X.b,Y.b,x,y,cov_AB)
}
```

Appendix D: R code for Classifying Training Data Points

```
X.a = leafdata.A$Width
Y.a = leafdata.A$Length
X.b = leafdata.B$Width
Y.b = leafdata.B$Length

for(i in 1:32){
  lambda = classifyLeafType(X.a,Y.a,X.b,Y.b,leafdata.df$Width[i],leafdata.df$Length[i],cov_AB)
  leafdata.df$Lambda[i] = format(lambda,scientific=FALSE)
  if(lambda>1){
    leafdata.df$Classification[i]="A"
  }
  else if (lambda<1) {
    leafdata.df$Classification[i]="B"
  }
  else {
    leafdata.df$Classification[i]="U"
  }
```

```
}

knitr::kable(leafdata.df,caption="Data Set After Running Classification Function for Each Row")
InvalidRows<-data.frame()
for(i in 1:32){
  if(leafdata.df$Classification[i] != leafdata.df$Species[i]){
    InvalidRows<-rbind(InvalidRows,data.frame(LeafNo=leafdata.df$LeafNo[i],Species=leafdata.df$Species[
  }
}
knitr::kable(InvalidRows,caption="Classification Errors")
```

Appendix E: R code for Classifying New Leaves

```
NewLeaves <- data.frame(Width=c(32,38,40),Length=c(82,52,76))
knitr::kable(NewLeaves,caption="New Leaves")
for(i in 1:dim(NewLeaves)[1]){
  lambda = classifyLeafType(X.a,Y.a,X.b,Y.b,leafdata.df$Width[i],leafdata.df$Length[i],cov_AB)
  NewLeaves$Lambda[i] = format(lambda,scientific=FALSE)
  if(lambda>1){
    NewLeaves$Classification[i]="A"
  }
  else if (lambda<1) {
    NewLeaves$Classification[i]="B"
  }
  else {
    NewLeavesf$Classification[i]="U"
  }
}
knitr::kable(NewLeaves, caption="New Leaves After Classification")
```

Appendix F: Plot Straight Line for Classification Rule

```
leaf.df <- leafdata.df[1:32,c(2:4)]

N.dimensions <- 300
X.space <- seq(from = min(leaf.df$Width), to = max(leaf.df$Width), length.out = N.dimensions)
Y.space <- seq(from = min(leaf.df$Length), to = max(leaf.df$Length), length.out = N.dimensions)
Observation.space <- expand.grid(Width = X.space, Length = Y.space)

predict.values<- c()
for(i in 1:(N.dimensions^2)){
  lambda = classifyLeafType(X.a,Y.a,X.b,Y.b,Observation.space[i,1],Observation.space[i,2],cov_AB)
  if (lambda >1){
    predict.values <- append(predict.values,1)
  } else {
    predict.values <- append(predict.values,2)
  }
}
length(predict.values)
plot(leaf.df[, 1:2], col = c(4,2)[as.factor(leaf.df$Species)]); contour(x = X.space, y = Y.space, z = ma
```

Appendix G: Plot Straight Line for New Classification Rule

```
classifyLeafType.new <- function(X.a,Y.a,X.b,Y.b,x,y,cov_A,cov_B){
  f(X.a,Y.a,x,y,cov_A)/f(X.b,Y.b,x,y,cov_B)
}

for(i in 1:32){
  lambda = classifyLeafType.new(X.a,Y.a,X.b,Y.b,leafdata.df$Width[i],leafdata.df$Length[i],cov_xyA,cov_
  leafdata.df$LambdaNew[i] = format(lambda,scientific=FALSE)
  if(lambda>1){
    leafdata.df$ClassificationNew[i]="A"
  }
  else if (lambda<1) {
    leafdata.df$ClassificationNew[i]="B"
  }
  else {
    leafdata.df$ClassificationNew[i]="U"
  }
}

knitr::kable(leafdata.df,caption="New Classification Values Under ClassifcationNew Column")

predict.values1<- c()
for(i in 1:(N.dimensions^2)){
  lambda1 = classifyLeafType.new(X.a,Y.a,X.b,Y.b,Observation.space[i,1],Observation.space[i,2],cov_xyA,
  if (lambda1 >1){
    predict.values1 <- append(predict.values1,1)
  } else {
    predict.values1 <- append(predict.values1,2)
  }
}
length(predict.values1)
plot(leaf.df[, 1:2], col = c(4,2)[as.factor(leaf.df$Species)]); contour(x = X.space, y = Y.space, z = ma
```