STAT 503X Case Study 2: Italian Olive Oils

1 Description

This data consists of the percentage composition of 8 fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, eicosenoic) found in the lipid fraction of 572 Italian olive oils. (An analysis of this data is given in (Forina, Armanino, Lanteri & Tiscornia 1983)). There are 9 collection areas, 4 from southern Italy (North and South Apulia, Calabria, Sicily), two from Sardinia (Inland and Coastal) and 3 from northern Italy (Umbria, East and West Liguria).

The data available are:

<table>
<thead>
<tr>
<th>Region</th>
<th>Sub-regions within the larger regions (North and South Apulia, Calabria, Sicily, Inland and Coastal Sardinia, Umbria, East and West Liguria)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Palmitic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Palmitoleic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Stearic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Oleic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Linoleic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Linolenic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Arachidic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
<tr>
<td>Eicosenoic Acid</td>
<td>Percentage ×100 in sample</td>
</tr>
</tbody>
</table>

The primary question is “How do we distinguish the oils from different regions and areas in Italy based on their combinations of the fatty acids?”
# 2 Suggested Approaches

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reason</th>
<th>Type of questions addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Restructuring</strong></td>
<td>This is very clean data so I don’t see any need to restructure. The only need is to construct some visual cues to help us code region and area.</td>
<td></td>
</tr>
<tr>
<td><strong>Summary Statistics</strong></td>
<td>To get at location and scale information for each variable, and by groups</td>
<td>What is the average percent composition of eicosenoic acid? Which variables have differences in the means for different groups?</td>
</tr>
<tr>
<td><strong>Visual Inspection</strong></td>
<td>The aim of visual methods is to understand separations, help decide the best classification method, and hence interpret solutions</td>
<td>Which variables might be useful discriminators of the regions and areas? Which pairs of variables might be useful discriminators of the regions and areas? Which combinations of variables might be useful discriminators of the regions and areas?</td>
</tr>
<tr>
<td><strong>Numerical Analysis</strong></td>
<td>The aim of numerical solutions is to get the best predictive results.</td>
<td>“Can we find an accurate classification scheme for the source of the oils based on their fatty acid composition?”</td>
</tr>
<tr>
<td></td>
<td>Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Classification and Regression Trees (CART), Neural Networks (feed-forward and support vector machines)</td>
<td></td>
</tr>
</tbody>
</table>
3  Actual Approaches

3.1  Data Structuring

Data Setup: Several files were constructed to be passed into XGobi:

- **olive.dat**: the data matrix, each row contains measurements on one sample
- **olive.col**: the variable labels, one label per row, so there are as many rows as there are variables
- **olive.row**: the row labels, one label per row, so there are as many rows as there are samples
- **olive.colors**: colors for each sample in the data set, and the colors code the region of Italy which produced the sample
- **olive.glyphs**: glyphs for each sample in the data set, and the glyphs code the area within the region
- **olive.doc**: documents where the data came from, not really used in XGobi

Data was also reformatted for *svm_light*. A C program is attached that does this.

3.2  Summary Statistics

from the summary statistics we notice that most of the olive oil breaks down into oleic acid. The remaining acids are around 10% or less of the oil. Southern oils have higher average eicosenoic, palmitic and palmitoleic acids.

<table>
<thead>
<tr>
<th></th>
<th>palmitic</th>
<th>palmitoleic</th>
<th>stearic</th>
<th>oleic</th>
<th>linoleic</th>
<th>limolenic</th>
<th>arachidic</th>
<th>eicosenoic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>610</td>
<td>15.00</td>
<td>152.0</td>
<td>6300</td>
<td>448.0</td>
<td>0.00</td>
<td>0.0</td>
<td>1.00</td>
</tr>
<tr>
<td>Median</td>
<td>1201</td>
<td>110.00</td>
<td>223.0</td>
<td>7302</td>
<td>1030.0</td>
<td>33.00</td>
<td>61.0</td>
<td>17.00</td>
</tr>
<tr>
<td>Max.</td>
<td>1753</td>
<td>280.00</td>
<td>375.0</td>
<td>8410</td>
<td>1470.0</td>
<td>74.00</td>
<td>105.0</td>
<td>58.00</td>
</tr>
<tr>
<td>Mean</td>
<td>1232</td>
<td>126.10</td>
<td>228.9</td>
<td>7312</td>
<td>980.5</td>
<td>31.89</td>
<td>58.1</td>
<td>16.28</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>168.6</td>
<td>52.5</td>
<td>36.7</td>
<td>405.8</td>
<td>242.8</td>
<td>13.0</td>
<td>22.0</td>
<td>14.1</td>
</tr>
</tbody>
</table>

Southern Oils - 323 obs

<table>
<thead>
<tr>
<th></th>
<th>palmitic</th>
<th>palmitoleic</th>
<th>stearic</th>
<th>oleic</th>
<th>linoleic</th>
<th>limolenic</th>
<th>arachidic</th>
<th>eicosenoic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>875</td>
<td>35.00</td>
<td>152.0</td>
<td>6300</td>
<td>448.0</td>
<td>20.00</td>
<td>32.00</td>
<td>10.00</td>
</tr>
<tr>
<td>Median</td>
<td>1346</td>
<td>163.00</td>
<td>223.0</td>
<td>7030</td>
<td>1090.0</td>
<td>37.00</td>
<td>62.00</td>
<td>27.00</td>
</tr>
<tr>
<td>Max.</td>
<td>1753</td>
<td>280.00</td>
<td>375.0</td>
<td>8113</td>
<td>1462.0</td>
<td>74.00</td>
<td>102.00</td>
<td>58.00</td>
</tr>
<tr>
<td>Mean</td>
<td>1332</td>
<td>154.8</td>
<td>228.8</td>
<td>7100</td>
<td>1033.0</td>
<td>38.07</td>
<td>63.12</td>
<td>27.32</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>152.9</td>
<td>50.7</td>
<td>39.9</td>
<td>345.1</td>
<td>210.1</td>
<td>7.97</td>
<td>11.2</td>
<td>8.39</td>
</tr>
</tbody>
</table>

Sardinian Oils - 56 obs

<table>
<thead>
<tr>
<th></th>
<th>palmitic</th>
<th>palmitoleic</th>
<th>stearic</th>
<th>oleic</th>
<th>linoleic</th>
<th>limolenic</th>
<th>arachidic</th>
<th>eicosenoic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>1030</td>
<td>35.00</td>
<td>199.0</td>
<td>6882</td>
<td>1057</td>
<td>15.00</td>
<td>45.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Median</td>
<td>1107</td>
<td>96.00</td>
<td>222.0</td>
<td>7325</td>
<td>1146</td>
<td>27.00</td>
<td>72.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Max.</td>
<td>1213</td>
<td>135.00</td>
<td>272.0</td>
<td>7439</td>
<td>1470</td>
<td>43.00</td>
<td>105.00</td>
<td>3.00</td>
</tr>
<tr>
<td>Mean</td>
<td>1111</td>
<td>96.74</td>
<td>226.2</td>
<td>7268</td>
<td>1197</td>
<td>27.09</td>
<td>73.17</td>
<td>1.94</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>40.4</td>
<td>13.9</td>
<td>17.6</td>
<td>141.9</td>
<td>107.2</td>
<td>5.38</td>
<td>11.9</td>
<td>0.744</td>
</tr>
</tbody>
</table>

Northern Oils - 206 obs
3.3 Visual Inspection

For this section our approach is to examine differences between the 3 regions first, followed by examining the areas within region. The 3 regions are coded as color: red=South(1), green=Sardinia(2), purple=North(3). We begin by examining univariate plots, then bivariate, followed by tour methods.

Figure 1 shows the plots which illuminate large differences between regions. The top left is a plot of Region vs eicosenoic acid, which illustrates the large split between Southern oils (Region 1) and the other two regions: the presence of eicosenoic acid is an indicator that an oil is from the South. The top right plot shows oleic vs linoleic acids. These two variables almost separate the Northern oils from the Sardinian oils, along the line of negative correlation between the two variables. The bottom left shows a plot of arachidic vs linoleic acid. There is a strong separation between the two regions, but the boundary is non-linear. The bottom right shows a tour plot, where the variable arachidic acid has been manually rotated into the plot of oleic and linoleic. Clearly these three variables provide a linear separation between the two regions.

Figure 2 shows plots which illuminate the separation of areas within regions. The left side plot shows a projection where the oils from areas within the Southern region roughly cluster out. These areas are the most difficult to separate neatly. All the variables are important for separating the four groups. The middle plot shows a pairwise plot of oleic vs linoleic acid revealing the clear separation between the two oils from the island of Sardinia. The right plot shows a projection revealing some separation of the oils from areas within Northern Italy. Most of the variables have some contribution. These oils can be fairly accurately but not perfectly classified.

From the visual analysis there are some important notes:

- Eicosenoic acid alone is sufficient to cleanly separate southern oils from the other two regions.
- Four variables, oleic, linoleic, linolenic and arachidic acids are good at separating northern oils from sardinian oils. A linear boundary is possible using oleic, linoleic and arachidic. A very clean non-linear separation can be found from linoleic and arachidic acids.
- There should be no error in classifying regions. These three groups are separable in the space of the fatty acid variables.
- Southern oils are the most difficult to separate. There is considerable overlap between the four areas in the south, with Sicily (4) being the most overlapping with the other three. Almost all variables are important to get the best separation.
- The two areas of Sardinia are separable in oleic and linoleic acids.
- Northern oils can be classified with a very small amount of error. Most variables are important: oleic, linoleic, palmitic, linolenic, stearic, arachidic.

3.4 Numerical Analysis

The data is broken into training and test samples for this part of the analysis. Approximately 25% of cases within each group are reserved for the test samples. These are the cases used for the test sample, where the data has been sorted by region and area:
Figure 1: Olive oil data: Plots illustrating separation of oils from different regions.
Figure 2: Olive oil data: Plots illustrating separation of oils from different areas.

<table>
<thead>
<tr>
<th>Region</th>
<th>Area</th>
<th># Train</th>
<th># Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>42</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>158</td>
<td>48</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>27</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>49</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>38</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>38</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>40</td>
<td>11</td>
</tr>
</tbody>
</table>

3.4.1 Classical Linear Discriminant Analysis

Classical Linear discriminant analysis (LDA) assumes that the populations come from multivariate normal distributions with different mean but equal variance-covariance matrices. Linear boundaries result, and it is possible to reduce the dimension of the data into the space of maximum separation using canonical coordinates. LDA results in 4 errors in the training sample (error rate = 4/572 = 0.007) and no errors in the test sample. The variables eicosenoic, palmitic, palmitoleic, oleic and linolenic are highly correlated with the first canonical coordinate. Oleic, linoleic, and arachidic are highly correlated with the second canonical coordinate. The errors in the training sample are from the southern region, and occur because LDA assumes equal variances, so sets the boundary too close to region 1.

Following Venables & Ripley (1994) here is the approach to fitting LDA in S-plus:
Figure 3: Olive oil data: Classical linear discriminant analysis on the Regions only. Missclassifications are marked with “x”.
```python
# Run LDA on training sample
> slm.lm <- lapply(slim.train, FUN = function(x) lm(x[-(1:2)], data = slim.train[, -1]))
> slm.lm

# Plot the predictions of the current sample
> par(mfrow = c(3, 3))
> plot(slim.xpred - slim.xtrain[, -(1:2)], slim.xtrain[, -(1:2)], pch = 16, col = slm.predict)

# Generate confusion matrix
> table(slim.xtrain[, -(1:2)], slim.xpred)

# Highlight the errors
> which(class0 - slim.xtrain[slim.xtrain[, -(1:2)] != predict(slim.lm, newdata = slim.xtest[, -(1:2)])])
> text(slim.xpred, slim.xtest, as.character(sl2[sl2 != predict(slim.lm, newdata = slim.xtest[, -(1:2)])])

```

```matlab
# This m.file is used to generate, plot the data.  # Given the parameters, we must solve the equation for points with no 1.  
# > solve(lamda, dami, rami); 6 points with no 1.  
# 0.45 0.55 0.5 0.60 0.65 0.70 0.75 0.80
```

In separating the areas within regions we find that for the areas in the south we have training error of 12/246 = 0.49 and test error of 6/77 = 0.078. Three canonical coordinates are needed to separate the four areas, and all the variables are used. The Sardinian oils separate with zero error in the test and training
samples. The most important variables are oleic and linoleic. In the northern oils the training error was 1/116 = 0.0086, and the test error was 2/35 = 0.057. All the variables except eicosenoic acid were used.

### 3.4.2 CART

The CART algorithm generates a classification tree by sequentially doing binary splits on the data. The simplest case is when splits are made on individual variables rather than combinations of variables.

For example, in the olive oil data, to separate the 3 regions CART would first split on eicosenoic acid, then linoleic acid. Amazingly it only uses these two variables:

**If eicosenoic acid $\gg$ 7 then the region is South (1)**

Else

**If linoleic $\gg$ 1048.5 then the region is Sardinia (2)**

Else the region is North (3)

The missclassification rate is zero in training and 1/136 = 0.007 for the test data. (Figure 4 shows the tree and a scatterplot illustrating the split.) The missclassification arises because the CART solution is based on a small separation between the two groups. Better separation can be found in combinations of variables. The CART algorithm doesn’t take the separation between groups into account. It stops as soon as it has a perfect separation of the training sample regardless of the separation between the two groups. Perhaps a different fitting measure (than the log likelihood) would produce better results.

```r
# olive.data <- factor(olive.data, 3)
# olive.train[,2] <- olive.data
# summary(olive.train)

> summary(olive.train)

Call:  

Recursive partitioning  

Data: olive.train
  Variables selected for split: 

1) root 436 0.2 0.50 0.50  
2) eicosenoic<7 (0.0000 0.0000 0.0000 0.0000)  
3) linoleic<800.5 0.0 0.0 0.0000 0.0000  
4) linoleic<800.5 0.0 0.0 0.0000 0.0000  
5) linoleic<700.5 0.0 0.0 0.0000 0.0000  
6) linoleic<600.5 0.0 0.0 0.0000 0.0000  
7) linoleic<500.5 0.0 0.0 0.0000 0.0000  
8) linoleic<400.5 0.0 0.0 0.0000 0.0000

The relative frequencies of the terminal nodes

1) root 436 0.2 0.50 0.50  
2) eicosenoic<7 (0.0000 0.0000 0.0000 0.0000)  
3) linoleic<800.5 0.0 0.0 0.0000 0.0000  
4) linoleic<800.5 0.0 0.0 0.0000 0.0000  
5) linoleic<700.5 0.0 0.0 0.0000 0.0000  
6) linoleic<600.5 0.0 0.0 0.0000 0.0000  
7) linoleic<500.5 0.0 0.0 0.0000 0.0000  
8) linoleic<400.5 0.0 0.0 0.0000 0.0000

How the splits are decided

Consider that each case in the training sample is assigned to a leaf, so at each leaf we have a random sample size $n_k$ from a multinomial distribution specified by $p_k$. The conditional likelihood (given the training sample) is proportional to
Figure 4: CART on Regions: (left) tree view, (right) data view of boundaries.

\[ \prod_{i=1}^{\#leaves} \prod_{j=1}^{\#classes} p_{ik}^{n_{ik}} \]

Then a deviance measure (how impure/matches true classification) is given by (take negative twice the log likelihood)

\[ D = -2 \sum_{i=1}^{\#leaves} \sum_{j=1}^{\#classes} n_{ik} \log p_{ik} \]

Estimate \( p_{ik} \) by \( \hat{p}_{ik} = n_{ik}/n_i \) (maximum likelihood estimator). The splitting strategy is to choose the attribute which minimizes this deviance.

(Algorithm in S-Plus is due to Clark and Pregibon, 1992.)

**Southern Oils**
There are 11 errors in the training sample, and 12 errors in the test sample, giving error rates of 11/246 = 0.045, 12/77 = 0.16, respectively. The variables used were palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, all variables except eicosenoic! (Figure 5 shows the tree and scatterplot for the first 2 splits.)

Sardinia
Figure 5: CART on South: (left) tree view, (right) data view of first split boundary.
There are no errors in classification here for training and test samples, and only one variable is used: linoleic acid. However the separation between the two groups is very small. (Figure 6 has the tree and a histogram illustrating the split.)

North
There are 3 errors in the training sample and 3 errors in the test sample giving error rates is 3/116 = 0.026, 3/35 = 0.086, respectively. (Figure 7 has the tree and scatterplot for the first 2 splits.)

3.4.3 Neural Networks

Feed-forward neural networks (Venables and Ripley, 1999)

Feed-forward neural networks provide a flexible way to generalize linear regression functions. A simple network model as produced by nnet code in S-Plus (Venables & Ripley 1994) may be represented by the equation:

\[ y = \phi (\alpha + \sum_{h=1}^{s} w_h \phi (\alpha_h + \sum_{i=1}^{p} w_{hi}x_i)) \]

where \( x \) is the vector of explanatory variable values, \( y \) is the target value, \( p \) is the number of variables, \( s \) is the number of nodes in the single hidden layer and \( \phi \) is a fixed function, usually a linear or logistic function. This model has a single hidden layer, and univariate output values.

The code available for Splus is as follows:

```r
USAGE:

nnet(x, y, weights, size, Wts, linout=F, entropy=F, softmax=F, skip=F, rang=0.7, decay=0, maxit=100, trace=T)
```
Figure 7: CART on North: (left) tree view, (right) data view of boundary.

REQUIRED ARGUMENTS:
- x: matrix or data frame of x values for examples.
- y: matrix or data frame of target values for examples.
- weights: weights for each example -- if missing defaults to 1.
- size: number of units in the hidden layer. Can be zero if there are skip-layer units.

OPTIONAL ARGUMENTS:
- Wts: initial parameter vector. If missing chosen at random.
- linout: switch for linear output units. Default logistic.
- entropy: switch for entropy (= maximum conditional likelihood) fitting. Default by least-squares.
- softmax: switch for softmax (log-linear model) and maximum conditional likelihood fitting. Linout, entropy and softmax are mutually exclusive.
- skip: switch to add skip-layer connections from input to output.
rang: Initial random weights on [-rang, rang]. Value about
0.5 unless the inputs are large, in which case it should
be chosen so that rang * max(|x|) is about 1.
decay: parameter for weight decay. Default 0.
maxit: maximum number of iterations. Default 100.
trace: switch for tracing optimization. Default True.

VALUE:
net object. Mostly internal structure, but has:

wts: the best set of weights found
value: value of fitting criterion plus weight decay term.
fitted.values: the fitted values for the training data.

DETAILS:
A quasi-Newton optimizer is used, written in C.

Here is how I ran it for the oils data.

# Set up to start running the code
library(mars)

# MLP
attach("/home/lisbon/3.mars/mars.lib")
# paths and files

dir <- "../
dirs <- list(")
symbols <- c("(invar")
olive.x <- oilive.train[,3:9]
olive.y <- oilive.train[,1]# oilive, mpg, oilive, train, [3:9] # set random number generator into a predictable state
set.seed(777) # set random number generator into a predictable state
olive.x <- oilive.train[,oilive, mpg, oilive, train[,3:9]] # set random number generator into a predictable state
plot(oilive.x, oilive, mpg, oilive, train[,3:9]) # set random number generator into a predictable state

summary(oilive.x)

print (oilive.x)

while (olive.x[,1], round (predict (olive.x, oilive.x)))

summary (olive.x)

print (olive.x)

while (olive.x[,1], round (predict (olive.x, oilive.x[,3:9])))

I could only manage to use the linear functions, not logistic. It needed to be run numerous times to
to get to the final solution. Reconstructing this solution requires saving the random number used to set initial
conditions.

The results were quite good. For separating oils from regions the main variables used (based on correlation
swith predicted values) were eicosenoic, oleic, palmitic, palmitoleic. For separating southern oils the most
important variables are oleic, linoleic, palmitoleic, palmitic, linoleic, linolenic. Oleic and linoleic are the
most important variables for separating Sardinian oils, and stearic and oleic acids are the most important
for separating northern oils.

<table>
<thead>
<tr>
<th>Group</th>
<th>Tr. err.</th>
<th>Test err.</th>
<th>Dev</th>
<th>Seed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>0</td>
<td>0</td>
<td>4.7</td>
<td>53,28,59,17,58,2,62,27,40,62,53,2</td>
</tr>
<tr>
<td>South</td>
<td></td>
<td>0.08</td>
<td>1.7</td>
<td>21,14,49,16,12,2,32,22,36,23,28,3</td>
</tr>
<tr>
<td>Sardinia</td>
<td></td>
<td>0</td>
<td>1.5</td>
<td>5,31,20,59,28,2,62,39,9,2,38,0</td>
</tr>
<tr>
<td>North</td>
<td></td>
<td>0.11</td>
<td>0.94</td>
<td>29,14,18,56,56,3,15,39,0,31,59,3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size</th>
<th>Decay</th>
<th>Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.0001</td>
<td>T</td>
</tr>
<tr>
<td>24</td>
<td>0.1</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>0.0001</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>T</td>
</tr>
</tbody>
</table>

3.4.4 Support Vector Machines (SVM)

SVM have recently gained widespread attention due to their success at prediction in classification problems.
The subject started in the late seventies (Vapnik 1979). The definitive reference is Vapnik (1999), and Burges
(1998) gives a simpler tutorial into the subject.
The main difference between this and the previously described classification techniques is that SVMs really only work for separating between 2 groups. There are some modifications for multiple groups but these are little more than one might do manually by working pairwise through the groups. SVM algorithms work by finding a subset of points in the two groups that can be separated, which are then known as the support vectors. These support vectors can be used to define a separating hyperplane, $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w} = \sum_{i=1}^{N_S} \alpha_i y_i \mathbf{x}_i$, $N_S$ is the number of support vectors, and $\alpha$ comes from the constraint $\sum_{i=1}^{N_S} \alpha_i y_i = 0$. Actually we search for the support vectors which gives the hyperplane which gives the biggest separation between the two classes. Note that it is possible to incorporate non-linear kernel functions allows for defining non-linear separations, and also that modifications allow SVMs to perform well when the classes are not separable.

We have experimented with svm_light (http://ais.gmd.de/~thorsten/svm_light/), which is written in C. We used first order polynomial kernels on the data normalized to percentages (0–100). For separating Sicily (4) from the other three southern areas we used a Gaussian kernel. We began by developing a classification for southern Italy from the other two, then a rule for separating northern Italy from Sardinian oils. Then we worked separately in the areas within the regions. We trained on subsets of each group, and reserved a test subset for assessing the error rate.

<table>
<thead>
<tr>
<th>Groups</th>
<th>Subgroups</th>
<th>Tr. Ex.</th>
<th>Ts. Ex.</th>
<th>SV</th>
<th>Slack</th>
<th>Tr. err.</th>
<th>Test err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>1,2,3</td>
<td>436</td>
<td>136</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2,3</td>
<td>190</td>
<td>59</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Group 1</td>
<td>11,12,13,14</td>
<td>246</td>
<td>77</td>
<td>30</td>
<td>1</td>
<td>100</td>
<td>95.59</td>
</tr>
<tr>
<td></td>
<td>11,12,13</td>
<td>219</td>
<td>67</td>
<td>6</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>12,13</td>
<td>200</td>
<td>59</td>
<td>7</td>
<td>0</td>
<td>100</td>
<td>96.72</td>
</tr>
<tr>
<td>Group 2</td>
<td>25,26</td>
<td>74</td>
<td>24</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Group 3</td>
<td>37,38,39</td>
<td>116</td>
<td>35</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>37,38</td>
<td>76</td>
<td>24</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The results from running SVMs on the olive oils data are superb (Table 1). It perfectly separates the 3 regions, the oils from Sardinia, northern Italy and with a small amount of error in the southern Italian oils which would be expected given the earlier visual inspection.

It is interesting to examine the support vectors for each classification performed. Figure 8 shows a plot of eicosanoic acid. The 9 support vectors (4 from southern Italian oils, 2 from Sardinian oils, and 3 from northern oils) for separating oils from southern Italy from the other regions are plotted as large solid circles. We would expect to see these corresponding to locations on the boundaries of the two groups. But we don’t see this. The 4 support vectors from the classification of northern oils from Sardinian oils are shown as solid circles (Figure 9). These lie more as expected in the boundaries of the two groups.
Figure 8: Olive oil data: Support vectors for separating region 1 from regions 2, 3 are marked as large solid circles. (Left) Region vs eicosenoic. Interestingly, the support vectors do not lie on the edge of the two groups in the plot of eicosenoic acid alone. (Middle, Right) The manual correlation tour is used to introducing palmitic in an additive linear combination with eicosenoic, and subtract a small amount of stearic acid. The support vectors are now clearly on the boundary between the two groups.

Figure 9: Olive oil data: Support vectors for separating region 2 from regions 3 are marked as solid circles. (Left) Oleic vs linoleic. The support vectors are where we would roughly expect them to be relative to the boundary between the two groups. (Right) Correlation tour used to explore the influence of arachidic acid. Arachidic acid is added to linoleic acid, increasing the separation between the two groups, and
4 Summary

The main pieces to note about classifying oils are:

1. There is considerable separation visible in the data from the graphical inspection so should be possible to formulate a very good classification scheme. There is almost zero uncertainty in classifying the three regions from each other. Sub-areas in Sardinia and the Northern areas also separate fairly cleanly. Separating the sub-areas of Southern Italy will contain some substantial uncertainty.

2. Of all the methods tested FFNNs and SVMs perform the best in terms of prediction error. LDA and CART make some bad errors in the prediction because of the limitations of the methods: variance assumption for LDA, greedy fitting algorithm for CART and failure to see structure in combinations of variables. The complication with using the FFNN is finding good initial conditions for optimizing the fit. For svm_light, the difficulty is the awkward input data format, and trial and error on parameters to the fit to get the best results.

3. The best test error rates are 0 for regions, 8% for the southern oils, 0 for Sardinia, and 3% for the northern oils, effectively achieved by both FFNN and SVM.

4. Variables that are important for separating regions are eicosenoic acid, oleic, linoleic and arachidic acids. For separating the southern oils almost all variables are needed. For separating Sardinian oils, oleic and linoleic are sufficient. For separating the northern oils almost all variables are needed.

Appendix

Visual Methods

Tours

Tours involve views of data provided by the manipulation of low-dimensional \((D)\) projections of high-dimensional \((N)\) spaces. The grand tour proposed by Asimov (1985) provided several algorithms for tour paths which could theoretically show the viewer the data “from all sides”. A grand tour path is dense in the set of all unoriented \(D\)-dimensional planes in \(N\)-space, meaning that, if the viewer could watch until the end of time she would see every possible \(D\)-dimensional projection of the data. The technical definition of a grand tour is “a continuous \(1\)-parameter family of \(D\)-dimensional projections of \(N\)-dimensional data which is dense in the set of all \(D\)-dimensional projections in \(R^N\). The parameter is usually thought of as time.”

Technically, we define a \(D\)-dimensional projection matrix, \(P\), to be a matrix of size \(N \times D\), where the columns are orthonormal. \(P\) actually defines a plane in \(R^N\). Then a projection of the data would be written as \(x_i P\), \(i = 1, \ldots, l\).

There have been several approaches to implementing grand tours. The method proscribed in Buja, Cook, Asimov & Hurley (1997) is follows:

- **Generate a series of anchor planes, \(P_j\), \(j = 1, \ldots, t\), from the space of all such planes.**
- **Interpolate over a geodesic (shortest) path from \(P_j\) to \(P_{j+1}\).**

Generating an interpolation path between two planes requires numerous calculations. The principle is that there are a set of canonical angles between the planes which can be found using singular value decomposition of the space of the two anchor planes. These are defined by a principal orthonormal basis (projection matrix) within each plane, which may be different to the original projection matrix which defined the plane. (A \(D\)-dimensional plane can be defined by an infinite number of projection matrices.) The geodesic interpolation space is defined by the principal bases, and interpolation occurs only in this space. In this manner the
interpolation distance is in a sense the smallest possible, and there is no within-plane rotation of the data, because the interpolation is invariant in the display space. (Note that orthonormality of the projection matrix must be maintained at every interpolation step.)

There have been several recent developments in tour methods. The grand tour was originally proposed in a movie format, but computing hardware advances have spurred research into the user interface (Hurley & Buja 1990). With the user interface come mechanisms for including or excluding a variable, pausing, speed control, and backtracking over the path. To make the viewing more efficient, the high-dimensional data space can be restricted to the space of several principal components (Hurley & Buja 1990, Buja, Asimov, Hurley & McDonald 1988) or interpolating between orthogonal subspaces (Buja et al. 1988, Young & Rheingans 1990). Alternatively, more interesting projections can be given a higher probability of a visit than less interesting views during the tour path (Cook, Buja, Cabrera & Hurley (1995) in work preceded for the regression setting by McDonald, Stuetzle & Friedman (1982)). Cook & Buja (1997) describe a manual tour, where the user can adjust the projection coefficient of a single variable by rotating it into or out of the current projection. The mechanics and theory for constructing the tour paths proposed above and by Asimov (1985) are based on subspace interpolation, the details of which are given in Buja et al. (1997). A correlation tour (Buja et al. 1988) is defined as $2 \times 1$-dimensional tours, one displayed horizontally and the other vertically in a 2-dimensional display space. This is useful in situations where there are two disjoint sets of variables, eg a regression setting. Each of the modifications to the grand tour algorithm can be achieved by altering the process of selection of anchor planes, or reducing the data space.

Resources


Ripley’s web site is [www.stats.ox.ac.uk](http://www.stats.ox.ac.uk), which has the tree code used here, and also `rpart` code written by Terry Therneau and Elizabeth Atkinson at the Mayo Clinic.


Original reference is: Breiman, Friedman, Olshen, Stone. Classification and Decision Trees Wadsworth, 1984

References


23


Vapnik, V. (1999), The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science), Springer-Verlag, New York, NY.
