Visualization and Learning Structure: Problem Session 7/12/16

Today we'll be looking at a famous olive oil data set used in clustering and classification. We have eight different chemical composition measurements on 572 Italian olive oils. Additionally, there are two sets of labels, the Region and the Area. There are three different Regions (Southern Italy, Sardinia, and Northern Italy). Each region is comprised of a group of area: Southern Italy = North Apulia, Calabria, South Apulia, and Sicily; Sardinia = Inland Sardinia, Costal Sardinia; Northern Italy = Umbrian, East Liguria, and West Liguria.

Our current goal is to build classifiers predicting Region and Area and to visualize any (dis)similarities in the multivariate structure of the data.

1. Download the olive oil data from our website and read into R.

   ```
   olive<-read.table("olive",header=T)
   ```

   Use `attach(olive)` to create a variable for each column.

   Use `table(Region, Area)` to make sure you understand the hierarchical structure of the labels. How many areas does each Region have?

2. Because there are nine areas, you'll need to create nine different colors for the areas. Otherwise, the color black will get repeated for Areas 1 and 9.

   ```
   area.spectrum<-rainbow(9)
   area.col<-area.spectrum[Area]
   ```

   Let's look (pairwise) at the chemical measurements colored by their Region/Area. Do we see any obvious class separation?

   ```
   chem.meas<-olive[,3:10]
   pairs(chem.meas,pch=16,col=Region)
   pairs(chem.meas,pch=16,col=area.col)
   ```

3. Build a linear discriminant classifier predicting Region using the eight variables. Then predict the class (Region) for each olive oil. (First do `library(MASS)`.)

   Note that the predict function below will give you the matrix of posterior probabilities for each class (572 x 3) and the predicted class with the maximum posterior probabilities.

   ```
   lda.olive<-lda(Region~.,data=chem.meas) ##the ~. means use all vars
   pred.olive.lda<-predict(lda.olive,as.data.frame(chem.meas))
   ```

   Use `table(Region, pred.olive.lda$class)` to find your misclassification rate.

   (Could also re-do your pairs plot using `col = pred.olive.lda$class`.)

4. We can visualize the posterior probabilities for each class of being assigned to that class for a more detailed assessment. (So, looking at posterior probability of being in Class 1 for all observations that were predicted to be in Class 1.)

```
par(mfrow=c(1,3))
hist(pred.olive.lda$post[pred.olive.lda$class==1,1],main="Predicted Class 1")
hist(pred.olive.lda$post[pred.olive.lda$class==2,2],main="Predicted Class 2")
hist(pred.olive.lda$post[pred.olive.lda$class==3,3],main="Predicted Class 3")
```

5. Now build a quadratic discriminant classifier predicting Region using the eight variables. Again, predict the class for each olive oil and compare to the true Region to find your misclassification rate. (In your code, replace `lda` with `qda`.)

   Visualize your posterior probability distributions; do they show an improvement?

6. We would expect the more flexible quadratic discriminant classifier to improve our misclassification rate but at a higher parameter estimation cost.

   Given that we're fitting a three class model on eight dimensions, how many more parameters do we need to estimate if we use `qda` vs `lda`?

   Thinking about your misclassification rates and parameter estimation requirements, which classifier would you pick?

7. Now build a linear discriminant classifier predicting the more fine-grained Area. Predict final class, assess misclassification rate, and visualize your posterior probs.

   Are there any areas in particular where the classifier does well? worse?

8. Does your performance improve if you use a quadratic discriminant classifier?

9. Now we'll use glyphs/icons to visualize any (dis)similarities in the eight variables across Region and Area. Since 572 is tough to see, you may want a sample:

```
library(graphics); olive.sample<-sort(sample(seq(1,572),75))  ##putting in order
```

   Try one of the icons used in class; for example,

```
stars(chem.meas[olive.sample,],col.stars=Region[olive.sample])
stars(chem.meas[olive.sample,],col.stars=area.col[olive.sample])

library(TeachingDemos)
faces(chem.meas[olive.sample,],labels=Region[olive.sample])
faces(chem.meas[olive.sample,],labels=Area[olive.sample])
```

   Do you see similar shapes/faces across the Regions? Areas?