IS 621: Business Analytics and Data Mining Final

Ben Arancibia

May 19, 2015

Introduction

Part 1 Classification and Regression

Using the crime training data set for crime prediction in various neighborhoods, I will build a classification model that takes given inputs and predicts whether the neighborhood will be at risk for high crime levels. Below are the crime definitions:

- zn: proportion of residential land zoned for large lots (over 25000 square feet)
- indus: proportion of non-retail business acres per suburb
- chas: a dummy variable for whether the suburb borders the Charles River (1) or not (0)
- nox: nitrogen oxides concentration (parts per 10 million)
- rm: average number of rooms per dwelling
- age: proportion of owner-occupied units built prior to 1940
- dis: weighted mean of distances to five Boston employment centres
- rad: index of accessibility to radial highways
- tax: full-value property-tax rate per $10,000
- ptratio: pupil-teacher ratio by town
- black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- lstat: lower status of the population (percent)
- medv: median value of owner-occupied homes in $1000s
- target: whether the crime rate is above the median crime rate (1) or not (0) (target for classification)

I used a k nearest neighbor model to classify whether and area is at risk for high crime levels. The model and code is below and according to the accuracy calculation which was used from our previous class work, the accuracy is 1.

Read in the datasets

```
library(class)
crime.train <- read.csv("crime-training-data.csv", header=TRUE)
crime.test <- read.csv("crime-evaluation-data.csv", header=TRUE)
```

Use k nearest neighbors to do the prediction

```
crime.train[,1:13] <- scale(crime.train[,1:13])
crime.test[,1:13] <- scale(crime.test[,1:13])

pred.knn <- knn(crime.train[,1:13], crime.test[,1:13], cl=as.factor(crime.train[,14]), k=5)

crime.test$target <- pred.knn

head((crime.test))
```

```
##           zn      indus       chas        nox         rm        age
## 1 -0.3864190 -0.6244111 -0.2265299 -0.8386288  1.4261128 -0.3764993
## 2 -0.3864190 -0.4738235 -0.2265299 -0.1969278 -0.1737603  0.5143080
## 3 -0.3864190 -0.4738235 -0.2265299 -0.1969278  0.4124191  0.8911880
## 4 -0.3864190 -0.4738235 -0.2265299 -0.1969278 -0.3882520  0.4191362
## 5 -0.3864190 -0.7806283 -0.2265299 -0.5596284 -0.5351642 -1.1226456
## 6  0.7020852 -0.8974393 -0.2265299 -0.9874290 -0.6952984 -0.1823490
##          dis        rad        tax     ptratio      black       lstat
## 1 0.55698900 -0.8579380 -0.8543307 -0.80987496  0.4495664 -1.15653394
## 2 0.31844789 -0.6372466 -0.4877862  1.15345828  0.3103042 -0.34465682
## 3 0.31504826 -0.6372466 -0.4877862  1.15345828  0.3964054 -0.01365074
## 4 0.09563010 -0.6372466 -0.4877862  1.15345828 -1.2923529  1.92938102
## 5 0.06928292 -0.5269009 -0.6456823  0.04908333  0.4938129 -0.53882968
## 6 1.62329420 -0.1958637 -0.6174866  0.35585415  0.4743531  0.03196033
##          medv target
## 1  1.46235907      0
## 2 -0.41903856      1
## 3 -0.39623374      1
## 4 -0.98915906      1
## 5 -0.09977109      0
## 6 -0.36202651      0
```

```
accuracy <- sum(pred.knn == crime.test[,14])/length(pred.knn)
accuracy
```

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

Personally, I think that the possibility of the accuracy being 1 is highly inprobable, but it seems for this classification model it worked.

Part 2 Clustering

1. Read in the data

```
protein <- read.csv("country-protein.csv", header=TRUE, sep="\t")
```

2. Hierarchical clustering

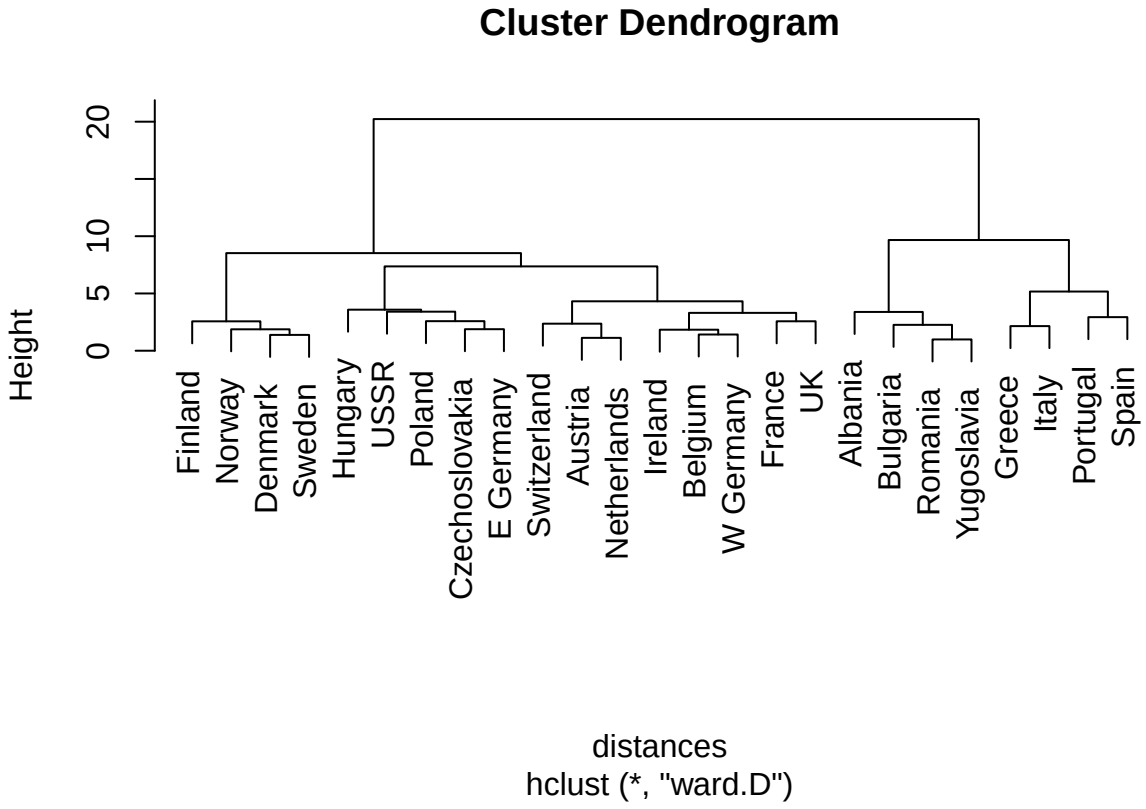
```
proteinmatrix <- scale(protein[,2:10])
attr(proteinmatrix, "scaled:center")
```

```
##   RedMeat WhiteMeat      Eggs      Milk      Fish   Cereals    Starch
##     9.828     7.896     2.936    17.112     4.284    32.248     4.276
##      Nuts    Fr.Veg
##     3.072     4.136
```

```
attr(proteinmatrix, "scaled:scale")
```

```
##   RedMeat WhiteMeat      Eggs      Milk      Fish   Cereals    Starch
##  3.347078  3.694081  1.117617  7.105416  3.402533 10.974786  1.634085
##      Nuts    Fr.Veg
##  1.985682  1.803903
```

```
distances <- dist(proteinmatrix, method="euclidean")
protein.hierarchical <- hclust(distances, method="ward.D")
plot(protein.hierarchical, labels=protein$Country)
```



3. K-means clustering

```
proteinmatrix <- scale(protein[,2:10])
protein.kmeans <- kmeans(proteinmatrix, centers=5,iter.max=100, nstart=100)
```

```
summary(protein.kmeans)
```

```
##              Length Class  Mode
## cluster      25     -none- numeric
## centers      45     -none- numeric
## totss         1     -none- numeric
## withinss      5     -none- numeric
## tot.withinss  1     -none- numeric
## betweenss     1     -none- numeric
## size          5     -none- numeric
## iter          1     -none- numeric
## ifault        1     -none- numeric
```

```
protein.kmeans$cluster
```

```
##  [1] 4 3 3 4 1 5 1 5 3 2 1 3 2 3 5 1 2 4 2 5 3 3 1 3 4
```

```
protein.kmeans$totss
```

```
## [1] 216
```

```
protein.kmeans$withinss
```

```
## [1] 16.994661 18.925874 22.110431  8.012133  5.900318
```

```
protein.kmeans$size
```

```
## [1] 5 4 8 4 4
```

```
protein$cluster <- protein.kmeans$cluster
proteinsorted <- protein[order(protein$cluster),]

knitr::kable(proteinsorted) # inspect the dataframe
```

	Country	RedMeat	WhiteMeat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Fr.Veg	cluster
5	Czechoslovakia	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0	1
7	E Germany	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6	1
11	Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2	1
16	Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6	1
23	USSR	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9	1
10	Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5	2
13	Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7	2
17	Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9	2
19	Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2	2
2	Austria	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3	3
3	Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0	3
9	France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5	3
12	Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9	3
14	Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7	3
21	Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9	3
22	UK	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3	3
24	W Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8	3
1	Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7	4
4	Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2	4

	Country	RedMeat	WhiteMeat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Fr.Veg	cluster
18	Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8	4
25	Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2	4
6	Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4	5
8	Finland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4	5
15	Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7	5
20	Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0	5

Challenge Problem

```
country <- read.csv("countries-challenge-data.csv", header=TRUE)
```

First step is to do the hierarchical clustering. Create a scaled version.

```
countrymatrix <- scale(country[,2:4])
attr(countrymatrix, "scaled:center")
```

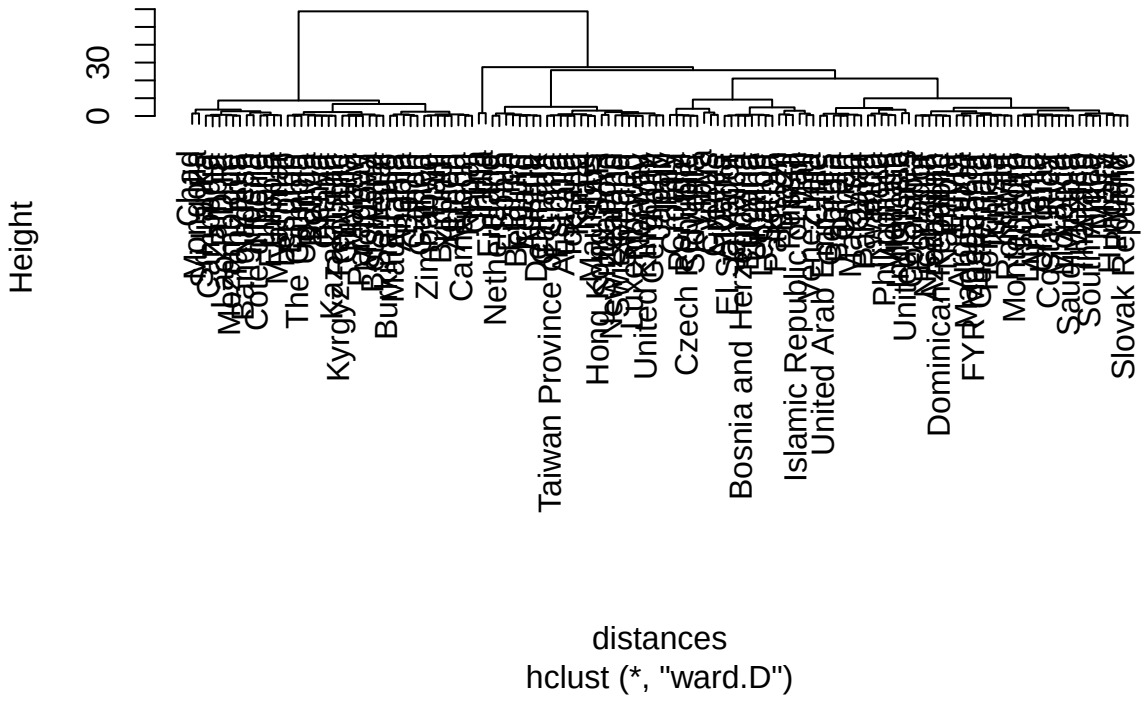
```
##   medgdpg     pop13       eti
##  3.427391 48.760072  4.050725
```

```
attr(countrymatrix, "scaled:scale")
```

```
##     medgdpg       pop13         eti
##   2.8525538 160.2819069   0.6934781
```

Create a distance matrix and then run the clustering. Plot the dendrogram.

```
distances <- dist(countrymatrix, method="euclidean")
country.hierarchical <- hclust(distances, method="ward.D")
plot(country.hierarchical, labels=country$country)
```



The next step is do kmeans clustering.

Scale the data set and perform kmeans() function to do the clustering

```
countrymatrix <- scale(country[,2:4])
country.kmeans <- kmeans(countrymatrix, centers=5,iter.max=100, nstart=100)
```

Evaluate the clusters

```
summary(country.kmeans)
```

```
##              Length Class  Mode
## cluster      138    -none- numeric
## centers       15    -none- numeric
## totss          1    -none- numeric
## withinss       5    -none- numeric
## tot.withinss   1    -none- numeric
## betweenss      1    -none- numeric
## size           5    -none- numeric
## iter           1    -none- numeric
## ifault         1    -none- numeric
```

```
country.kmeans$cluster
```

```
##   [1] 2 5 5 2 2 4 4 2 4 5 4 5 5 5 2 2 2 2 5 5 5 5 4 5 4 3 2 2 5 1 1 1 4 2 2
##  [36] 2 2 4 5 4 4 2 5 2 4 5 1 2 5 5 5 2 4 2 4 3 2 4 1 4 1 2 4 2 5 5 4 2 5 5
##  [71] 2 2 5 5 1 2 4 2 5 4 2 4 5 4 2 2 5 2 2 5 5 2 5 4 4 2 5 4 4 2 5 5 2 5 2
## [106] 1 4 2 2 5 2 2 2 4 2 1 2 1 5 4 4 4 5 2 5 2 2 5 1 4 4 4 2 2 2 5 5 5
```

```
country.kmeans$totss
```

```
## [1] 411
```

```
country.kmeans$withinss
```

```
## [1] 12.20892 28.94887  1.29813 20.31204 29.16146
```

```
country.kmeans$size
```

```
## [1] 11 50  2 33 42
```

Assign cluster labels to the data

```
country$cluster <- country.kmeans$cluster
countrysorted <- country[order(country$cluster),]

knitr::kable(head(countrysorted)) # inspect the dataframe
```

	country	medgdpg	pop13	eti	cluster
30	Croatia	-1.00	4.28	4.2	1
31	Cyprus	-4.76	0.88	4.4	1
32	Czech Republic	-0.87	10.52	4.4	1
47	Greece	-3.86	11.06	4.0	1
59	Islamic Republic of Iran	-1.67	77.10	3.0	1
61	Italy	-1.85	59.69	4.3	1

```
knitr::kable(tail(countrysorted))
```

	country	medgdpg	pop13	eti	cluster
123	Tanzania	6.96	46.28	3.5	5
125	The Gambia	6.35	1.88	3.6	5
128	Uganda	6.03	36.82	3.6	5
136	Yemen	4.40	26.66	3.0	5
137	Zambia	7.25	14.54	3.7	5
138	Zimbabwe	4.24	13.12	2.9	5