36-350: Data Mining

Lab 6
Date: October 3, 2003 **Due: end of lab**

Interspersed throughout this lab are questions that you will have to answer at check-off.

1. Download the files for this lab from the course web page to the desktop:

   `http://www.stat.cmu.edu/~minka/courses/36-350/lab/`

2. Open a Word or Notepad document to record your work.

Start R

3. `Start -> All Programs -> Class software -> R 1.7.0`

4. Load the special functions for this lab:

   `File -> Source R code...`

   Browse to the desktop and pick `lab6.r` (it may have been renamed to `lab6.r.txt` when you downloaded it). Another window will immediately pop up for you to pick the `mining.zip` file you downloaded.

The dataset

5. The dataset is the same as lab 5. Load it via

   ```
   data(States)
   ```

   This defines a matrix called `States`, and a standardized matrix called `StatesT`. You should use the standardized one. There is also a vector of labels called `state.region`.

Clustering

6. Using the commands from lab 4, use Ward's method to partition the data into 4 parts. *How does the sum of squares of your clustering compare to that of* `state.region`*?*

7. If `f` is a vector of cluster numbers, this command will name your clusters according `state.region`:

   ```
   f = factor(f,labels=name.clusters(f,state.region))
   ```

   In R, a vector of labels is called a **factor**. (`state.region` is also a factor).

Projections

8. Using the commands below and from lab 5, plot the m-projection of the clusters, using state labels and colors. Keep a copy for the homework.

9. Make a dichotomy between `South` and `Not South`, and plot the m-projection. Keep a copy for the homework.

10. Repeat for `Northeast`.

Parallel-coordinates

11. Make (and save) a parallel-coordinate plot of the cluster means.
12. Make (and save) two more parallel-coordinate plots: one where Texas is in a group by itself and another where New Mexico is in a group by itself. Note that the variable ordering and scaling may change slightly between the plots.
13. You can now get checked off.

Informative projection If `x` is a matrix and `f` is a factor, the combination weights for an m-, v-, or mv-projection can be computed via one of:

```
w = projection(x,f,2,type="m")
w = projection(x,f,2,type="v")
w = projection(x,f,2,type="mv")
```

`w` can be used just like PCA weights.

Scatterplot with colors To make a scatterplot of colored dots or labels, use one of the following:

```
color.plot(v2 ~ v1, x, f)
color.plot(v2 ~ v1, x, f, labels=T, asp=1)
```

The first uses dots, the second uses labels. Here `x` is a matrix and `f` is a vector of names or numbers. The usual options are also available (like `asp`).

Constructing factors If `f` is a factor, new factors can be constructed as follows:

```
fd = factor.dichotomy(f,"South")
fi = factor.isolate(f,"Texas")
```

`fd` is a factor with the labels changed to `South` and `NOT South` (only two groups). `fi` is a factor with the label for `Texas` changed to `Texas` (thus separating it from the other groups).

Parallel-coordinate plot Normally you want to plot the mean of each cluster, rather than all of the data. If `x` is a matrix and `f` is a factor:

```
xp = prototypes(x,f)
parallel.plot(xp)
```