36-350: Data Mining

Lab 4
Date: September 20, 2002 **Due: end of lab**

---

1 Introduction

This lab teaches you the steps involved in clustering data. You will compare the two methods taught in class: k-means and Ward's method.
There are 5 questions. For each one, submit your commands and a response from R demonstrating that they work. (Only hand in commands relevant to the question.) To submit a plot, click on the plot window and select

```
File -> Save as -> Postscript...
```

This saves the plot to a file which can be printed, incorporated into a Word document, or mailed to us as an attachment.

2 Starting R

Start R as in lab 1. On the class web page, go to "computer labs" and download the files for lab 4 into your work folder. Read the special functions into your running R application via the commands

```
source("lab1.r")
source("lab4.r")
```

If this fails, check that the files were downloaded correctly.

3 The data

There are two datasets used in this lab. The first is the image data from lab 2:

```
load("lab2_imgs.rda")
```

Remove infrequent colors, weight by picture frequency, and normalize the counts per image:

```
imgs <- remove.singletons(imgs)
imgs <- idf.weight(imgs)
x <- div.by.euc.length(imgs)
```

4 Clustering by k-means

The command `kmeans` will cluster by k-means:

```
cl <- kmeans(x,k)
```

where `k` is the desired number of clusters. For the image dataset, we know that there are two natural clusters, so try `k=2`. The result of `kmeans` is a list of four elements giving various information about the clusters. Type `str(cl)` to get an overview. Use the dollar sign to access components of the list, e.g. `cl$size` is a vector of the cluster sizes.
The most important component for our purposes is `cl$cluster`, which is a vector that tells you which cluster each of the images falls into. This type of vector is called a `factor` in R, and there are a variety of functions which operate on factors. The command `split(y,f)`, where `f` is a factor and `y` is a vector of the same length, will split the elements of `y` into multiple groups. By splitting a vector of the image names, you can figure out how kmeans has clustered the data.

> **Question 1:** (a) Use `split` to figure out how the images have been clustered by k-means. The vector to split is the names of the images, given by `rownames(imgs)`. (b) Have any images been put into the wrong cluster? If so, which ones?

5 Clustering by Ward's method

The command `hclust` will cluster by Ward's method. It works differently than k-means, in that it wants a distance matrix as input, not a data matrix. (This actually makes it more general since you can compute the distances any way you want, while `kmeans` always uses Euclidean distance.) The distance matrix must be computed by `dist`, and explicitly squared:

```
d <- dist(x)^2
hc <- hclust(d,method="ward")
```

Like `kmeans`, the result is a list, this time with 7 components. It doesn't give you a partition of the data, but rather a tree. You can view this tree with

```
plot(hc)
```

To determine the correct number of clusters, it helps to plot the merging cost at each step. This is done via `plot.hclust.trace`:

```
plot.hclust.trace(hc)
```

> **Question 2:** Turn in a plot of the merging costs. You will need it for the homework.

You can get a partition of any size by cutting the tree at the appropriate level. The function for this is `cutree`:

```
f <- cutree(hc,k)
```

The result is a factor of cluster numbers, just like `cl$cluster`. To match what you did with k-means, use `k=2`.

> **Question 3:** (a) Use `split` to figure out how the images have been clustered by Ward's method, when cut at two clusters. (b) Have any images been put into the wrong cluster? If so, which ones?

6 Comparing clusterings

It was shown in class that both k-means and Ward's method try to minimize the sum of squares of the clustering. You can compute this value using the function `sum.of.squares(x,f)` where `x` is the data matrix and `f` is a factor.

> **Question 4:** (a) Does k-means or does Ward's method do better at clustering the images into sailing versus racing? (b) Which clustering is better according to the sum of squares value?

7 Segmenting a color image

In this section, you will test out one of the applications of clustering discussed in class: segmenting a color image. The second data file contains the "hand" image shown in class, which you can view as follows:

```
load("lab4_img.rda")
plot.image(img)
```

The image can be segmented by simply clustering the pixels in the image, ignoring their position. This is equivalent to considering the pixels as a cloud in the color cube, and clustering the cloud.
The examples in class used Ward's method, where only neighboring pixels could merge. With k-means, any pixels in the image can be merged. This has the advantage that the fingers of the hand, which are separated in the image, may be clustered together with the rest of the hand. The downside is that the overall segmentation is noisier.
The image pixels in the (R,G,B) representation have been put into a matrix called `img.rgb`. Check it out with `str(img.rgb)`. Use `kmeans` to cluster this matrix into three parts. The resulting cluster assignments can be viewed as an image, as follows:

```
cl.img <- array(cl$cluster, c(152,122))
plot.image(cl.img)
```

This image should have three different colors, corresponding to the clusters. Pixels in the same cluster will have the same color in this plot.

> **Question 5:** Segment the hand image and send us a picture of the segmentation (as a low-quality jpeg file).