Economics 1660: Big Data Prof. Daniel Björkegren

PS 1: Clustering and Visualization

1. Clustering: Dissimilarity (Source: HTF Exercise 14.1)

Weights for Clustering. Show that weighted Euclidean distances:

$$\boldsymbol{d_e^w(x_i, x_j)} = \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{p}} \boldsymbol{w_l}\left(\boldsymbol{x_{\{il\}}} - \boldsymbol{x_{jl}}\right)^{\boldsymbol{2}}}{\sum_{\boldsymbol{l=1}}^{\boldsymbol{p}} \boldsymbol{w_l}}$$

satisfies

$$d_e^w(x_i, x_j) = d_e(z_i, z_j) = \sum_{l=1}^{p} (z_{il} - z_{jl})\text{^}2,$$

Where

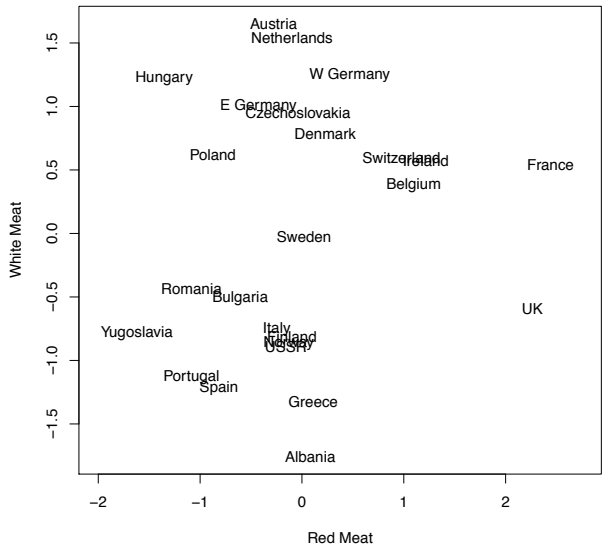
$$z_{il} = x_{il} \left(\frac{w_l}{\sum_{l=1}^{p} w_l}\right)^{\frac{1}{2}}.$$

Thus, weighted Euclidean distance based on x is equivalent to unweighted Euclidean distance based on z.

2. Clustering: Code

Clustering techniques can bring out structure in data. This problem asks you to find structure that differentiates European countries based on their food consumption.

On the website find the dataset protein.csv[1].



[1] Source: Hand et al. 1993 A Handbook of Small Data Sets

If you feel confident you may write code from scratch; you may also use starter python code for this problem available on the course website as protein_starter.py to get a head start.

- Compute a K-means cluster of the data using Red Meat and White Meat consumption
    - Fix K and compute the clusters that result from different random draws of cluster means. Some draws will lead to better partitions than others: use the partition that minimizes the within cluster point scatter.
    - Graph how the within cluster point scatter varies as K increases. (Hint: Should the within cluster point scatter ever increase as you add another cluster?)
    - What is your preferred number of clusters? What countries are in which clusters? How does the algorithm's result compare to how you would have classified the countries visually, based on the plot?
- Extend your analysis to use Red Meat, White Meat, Eggs, and Milk consumption:
    - First, look at the columns of data. Without running an algorithm, how would you cluster the countries based on these 4 dimensions? (If you find it helpful to graph the data feel free to do so.)
    - Next, compute K-means clusters of the data. Graph how the within cluster point scatter varies as K increases
    - What is your preferred number of clusters? What countries are in which clusters? How does the algorithm's result compare to the clusters you created on your own?

- *Optional:* Hierarchical clustering. Create a hierarchical cluster of the countries using the 4 dimensions of protein consumption. Show your results in a graph or tree. What appear to define the major divisions between clusters?

Hint:
The K-means algorithm has the following steps:[2]

1. Randomly choose an initial set of points as centers.
2. For each center identify the subset of training points that is closer to it than any other center.
3. Calculate the means of each feature for the data points in each cluster. This mean vector is the new cluster center.
4. Repeat steps 2 and 3 many times until each center stabilizes.

[2] The Elements of Statistical Learning; Hastie, Tibshirani, Friedman; Springer, 2008.

3. Napoleon's March

Can data reveal the story of Napoleon's march into Russia? This question asks you to explore a dataset visually.

Open the graphical user interface of the statistical package R, and install the ggplot2 graphing package with the command:

```
install.packages('ggplot2', dependencies=TRUE)
```

Set the working directory to the directory where the two data sets are; for example, you can use a command like:

```
setwd("/Users/USERNAME/Downloads/PS1_package/")
```

Then open the datasets:

```
troops <- read.table("tables/minard-troops.txt", header=T)
cities <- read.table("tables/minard-cities.txt", header=T)
```

You can look at the data tables by simply entering their name:

```
troops
```

Then, create plots to explore the dataset. For example, the command:

```
ggplot(troops, aes(long, lat)) + geom_point()
```

plots the latitude (y) and longitude (x) of each observation in the dataset. The following plots the number of survivors (y) and longitude (x) and connects the points:

```
ggplot(troops, aes(long, survivors)) + geom_path()
```

Assemble several plots to tell the story of Napoleon's march, and turn these in on one page.