DSE 210: Probability and Statistics using Python

Worksheet 10 — Clustering

1. *Suboptimality of Lloyd's algorithm.* Consider the following data set consisting of five points in $\mathbb{R}^1$:

$$-10, -8, 0, 8, 10.$$

We would like to cluster these points into $k = 3$ groups.

   (a) What is the optimal k-means solution? Give the locations of the centers as well as the k-means cost.

   (b) Suppose we call Lloyd's k-means algorithm on this data, with $k = 3$ and with initialization $\mu_1 = -10, \mu_2 = -8, \mu_3 = 0$. What is the final set of cluster centers obtained by the algorithm? What is the k-means cost of this set of centers?

2. For this problem, we'll be using the *animals with attributes* data set. Go to

`http://attributes.kyb.tuebingen.mpg.de`

and, under "Downloads", choose the "base package" (the very first file in the list). Unzip it and look over the various text files.

This is a small data set that has information about 50 animals. The animals are listed in `classes.txt`. For each animal, the information consists of values for 85 features: does the animal have a tail, is it slow, does it have tusks, etc. The details of the features are in `predicates.txt`. The full data consists of a 50×85 matrix of real values, in `predicate-matrix-continuous.txt`. There is also a binarized version of this data, in `predicate-matrix-binary.txt`.

   (a) Load the real-valued array, and also the animal names, into Python. Run k-means on the data (from `sklearn.cluster`) and ask for $k = 10$ clusters. For each cluster, list the animals in it. Does the clustering make sense?

   (b) Now hierarchically cluster this data, using `scipy.cluster.hierarchy.linkage`. Choose Ward's method, and plot the resulting tree using the `dendrogram` method, setting the `orientation` parameter to `'right'` and labeling each leaf with the corresponding animal name.

   You will run into a problem: the plot is too cramped because the default figure size is so small. To make it larger, preface your code with the following:

```
from pylab import rcParams
rcParams['figure.figsize'] = 5, 10
```

   (or try a different size if this doesn't seem quite right). Does the hierarchical clustering seem sensible to you?

   (c) Turn in an iPython notebook with a transcript of all this experimentation.