1 Fuzzy clustering

The fuzzy c-means clustering algorithm uses fuzzy logic – each point has a degree of belonging to each of the clusters rather than membership in a single cluster. We represent that degree of belonging using a “partition matrix” $W$, which is really a set of weights. If we have $N$ data points and $c$ clusters, then the matrix of weights is $N \times c$.

In Fuzzy c-means clustering, we find the weights that most effectively place our $N$ data points in $c$ clusters.

The algorithm is

1. Assign weights randomly. I choose from a uniform distribution on the range 0 to 1.

2. Repeat until converged. Convergence is indicated when the weights change less than some small threshold $\epsilon$.

   (a) Compute the mean of each cluster. This is a weighted component-wise arithmetic mean.

   $$\bar{c}_j = \frac{\sum_{i=1}^{N} w_{i,j}^m \bar{x}_i}{\sum_{i=1}^{N} w_{i,j}^m}$$

   where $\bar{x}_i$ is the $i^{th}$ data point, $w_{i,j}$ is the weight of the $i^{th}$ data point in the $j^{th}$ cluster, and $m$ is the fuzziness coefficient ($m > 1$, and the smaller the value of $m$, the sharper the clusters are).

   (b) For each point $\bar{x}_i$, compute the new weights. The weight for the $k^{th}$ cluster is the inverse of the sum of the “amplified” ratios where each ratio is the ratio of the distance between point and the mean of the $j^{th}$ cluster to the distance between the point and mean another cluster. If the point is very close to cluster $j$, then its distance to it will be small, putting a small number in the denominator, leading to a large weight.

   $$w_{i,j} = \frac{1}{\sum_{h=1}^{c} \left( \frac{d(\bar{c}_h, \bar{x}_i)}{d(\bar{c}_h, \bar{x}_j)} \right)^{2/(m-1)}}$$

Note: The Wikipedia page on fuzzy clustering is an excellent reference for this algorithm. That was my main reference for these notes.