

Pattern Recognition: Week 9

Clustering and k-means

John Quinn

May 26, 2008

Clustering

- In the previous methods discussed, we have assumed that labelled training data is available, i.e. that we know what the different classes are.
- However, sometimes we have a quantity of *unlabelled* data and want to find what different classes exist within it.
- For example, imagine a collection of pages which have been scanned from several different books. We don't know which books they came from (or maybe even how many books there were), but want to organise the pages into sets which belong to each other.
- The k-means algorithm does this by iteratively finding the center of each cluster in the data space.

K-means algorithm

- 1 **Initialisation:** set the means μ_1, \dots, μ_K to have the same position as K randomly selected data points.
- 2 **Assignment:** for each data point, work out which is the nearest mean.

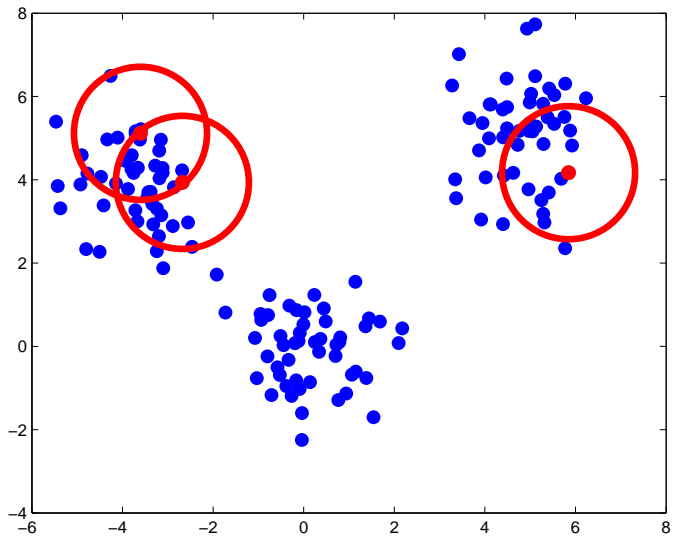
$$k_i = \arg \min_j d(\mu_j, x_i) \quad (1)$$

- 3 **Update:** re-calculate the K means to be the average of all the data points that “belong” to it.

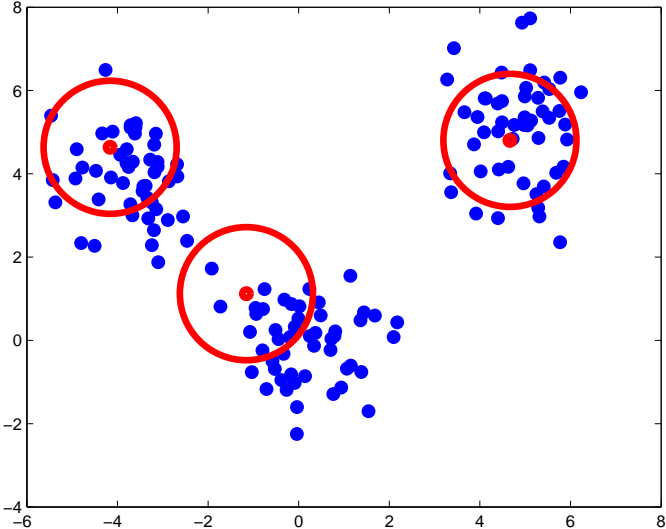
$$\mu_i = \frac{1}{N_i} \sum_{j:k_j=i} x_j \quad (2)$$

where N_i is the number of data points that belong to set i .
If the means have converged (no change), then stop.
Otherwise go to step 2.

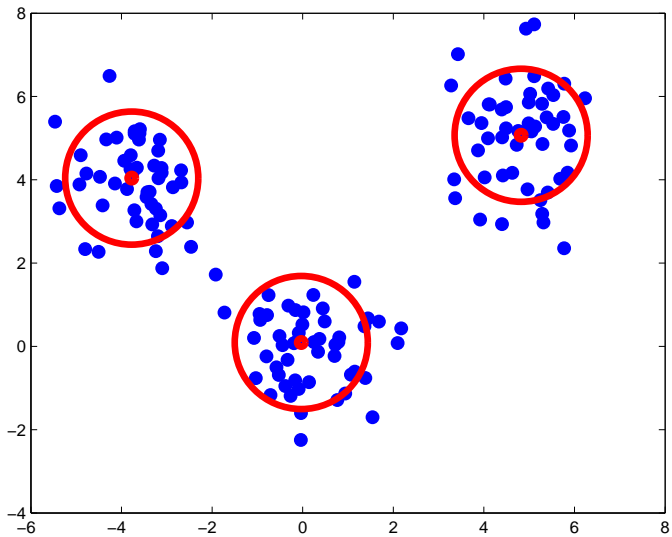
Initialisation



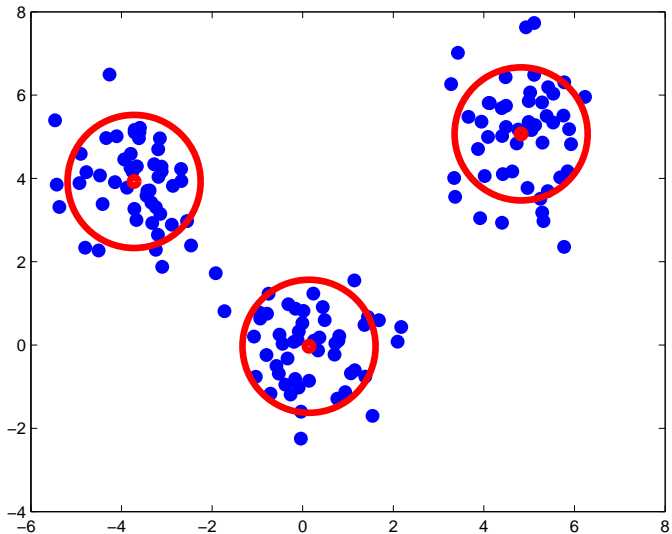
Iteration 1



Iteration 2



Iteration 3 (convergence)



Performance of k-means

Advantages

- Often gives good results.
- Converges quickly.

Disadvantages

- You have to know in advance how many clusters there are.
- There can be some problem cases where the algorithm gets stuck or does the wrong thing (think about long, thin, parallel clusters).
- No provision for uncertainty. In the algorithm, a point is definitely in one class or another. In real life, there can be ambiguity.