

K-means Clustering

Want to break N elements into K groups. How to choose the groups?

- Want high intercluster similarity: $\sum_k \sum_{n \neq m \in k} d(x_n, x_m)$ should be low.
- Want low intracluster similarity: $\sum_{k \neq k'} \sum_{n \in k} \sum_{m \in k'} d(x_n, x_m)$ should be high.

Too hard to optimize a linear combination of these... there are $\frac{1}{K!} \sum_k (-1)^{K-k} \binom{K}{k} k^N$ possible clusters (for $N = 19, K = 4$, this is over 10^{10}).

Idea: propose clusters, then iteratively fix errors.

- Initialize cluster centers
- Assign each data point to the closest center
- Recompute the centers as the means of the data points
- If assignments have changed, go to (2)

This is the k -means algorithm. It is highly sensitive to initialization, but converges quite quickly. (There are various ways to speed it up, as well.)

Initializing means:

- Random points drawn from a “reasonable” input space
- Random data points
- First a random data point, then a second data point as far away as possible, then a third data point again as far away as possible, etc.

In general, it’s a good idea to do a few runs of k -means.

k -medioids is similar, but forces each centroid to be a data point. This is advantageous for non-real-valued data.

Choosing the number of clusters

- Prior knowledge of the domain
- Visual inspection (in low D)
- Try a bunch and use them somehow decide
- Look for the “elbow” in the k versus $\text{int}(er/ra)$ cluster similarity
- use AIC or BIC