Recap: k-Means Clustering

DS701 Session 4 — Wed Sep 16, 2026

Today’s plan

Knowledge-Check Review

KC 1: What does k-means minimize?

Name the quantity and write it as a formula.

KC 2: One iteration by hand

Points $1.2,\ 1.8,\ 2.1,\ 7.3,\ 7.9,\ 8.2,\ 9.1,\ 10.5$, initial centers $A = 4.0$, $B = 11.0$.

KC 3: Why scale the features?

Cluster customers on age (18–80) and income ($20k–$200k), unscaled. What determines the clusters?

Highlights

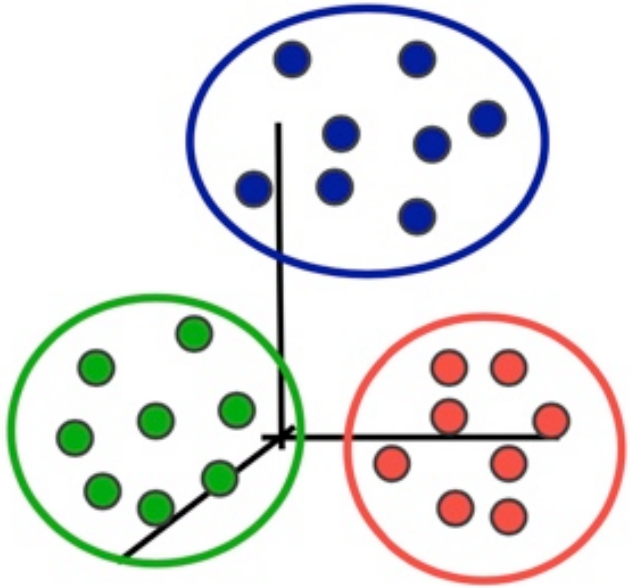
What are we actually asking for?

A **clustering** is a grouping of data objects such that objects within a group are *similar* and objects in different groups are *dissimilar*.

So we want to:

- minimize intra-cluster distances
- maximize inter-cluster distances

And it is **unsupervised** — no labels go in, and any labels we put on the clusters we invent *afterwards*.

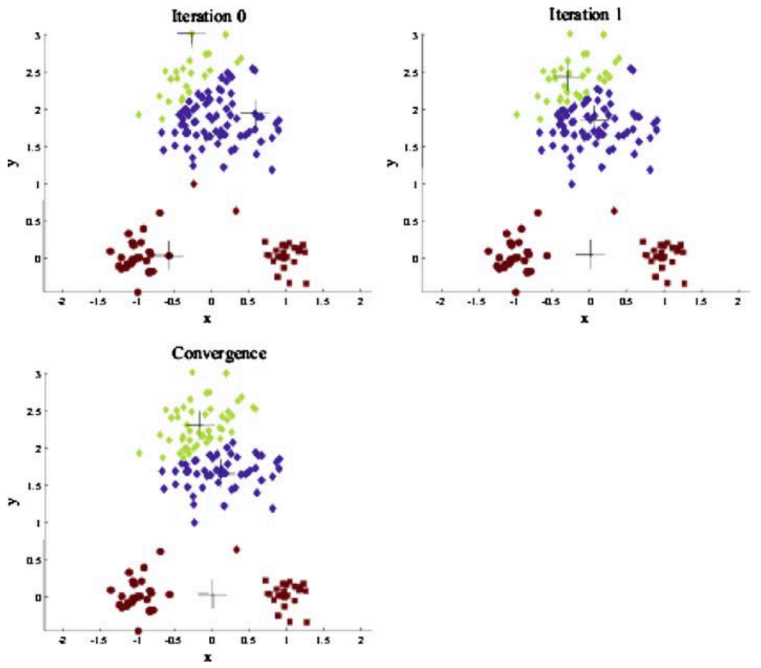


Hard problem, easy algorithm

Lloyd's algorithm

1. Pick K cluster centers $\{c_1, \dots, c_K\}$
    - random data points, or k-means++
2. **Assign**: define C_j as the points closest to center c_j
3. **Update**: set c_j to the center of mass of C_j
4. **Repeat** step 2 until convergence
    - centers move below a threshold, or inertia changes below a threshold, or max iterations

Initialization matters



Choosing K

Where k-means breaks

Assignment is “nearest center”, so cluster regions are **straight-sided Voronoi cells** and clusters are implicitly assumed to be **spheres** around their center.

Feature scaling

Because k-means looks for spherical clusters *in Euclidean distance*, the **units** of each feature are part of the model.

$$\begin{bmatrix} \text{age } 27 \\ \text{income } 75000 \\ \text{gender } 0 \end{bmatrix}, \qquad \begin{bmatrix} \text{age } 45 \\ \text{income } 42000 \\ \text{gender } 1 \end{bmatrix}$$

Where it works well

k-means is a workhorse when the assumptions roughly hold:

In-Class Activity

Activity: k-Means from scratch and in `scikit-learn`

Goal: implement one Lloyd's iteration yourself, check it against `sklearn.cluster.KMeans`, then the **stretch** — build the silhouette coefficient from its definition, sweep k, and settle a case where the elbow and the silhouette disagree — and finally break k-means on purpose and diagnose why.

- Work in groups of 2–3. Open the notebook for **your section** — A1 and B1 get different data.
- Parts marked (**autograded**) are submitted to Gradescope; the rest is participation.
- Staff will circulate — be ready to explain any part of your work.
- Dependencies: `numpy`, `matplotlib`, `scikit-learn` only.

Section A1: Open in Colab Section B1: Open in Colab Open on GitHub

The activity notebook goes live on the day of the lecture. Colab is optional — you can also open it on GitHub and run it

Going deeper

Full lecture notes: k-Means Clustering

- Minimizing a Cost Function — the WCSS objective
- The k-means Algorithm — the four steps, worked on well-separated and overlapping blobs
- Limitations of k-means — non-spherical, anisotropic, unequal size and variance
- Choosing a Good Initialization — k-means++
- Choosing the right k — hyperparameters
- Feature Scaling
- Activity: Clustering 1D Data — the pencil-and-paper walkthrough
- Next up: hierarchical clustering (04) — clusters without choosing k up front.