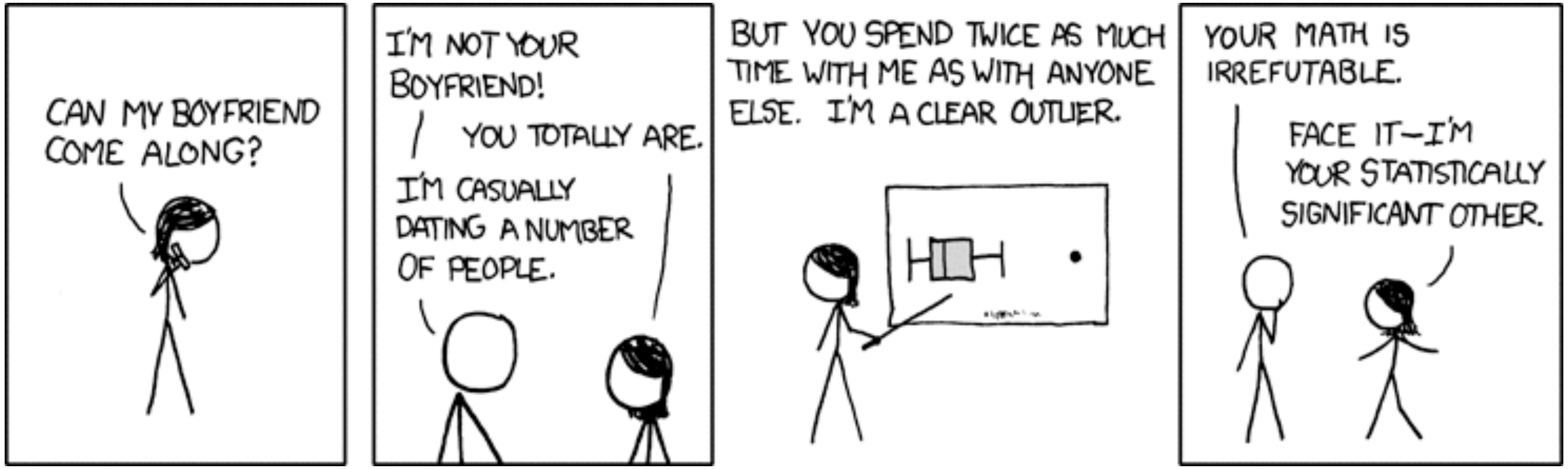
Introduction to Data Science
CS 5963 / Math 3900
Lecture 12: Clustering II

Alexander Lex
alex@sci.utah.edu

Braxton Osting
osting@math.utah.edu

THE UNIVERSITY OF UTAH



[xkcd]

Recap: Supervised vs. Unsupervised Learning

Supervised Learning

Data: both the features, x, and a response, y, for each item in the dataset.

Goal: 'learn' how to predict the response from the features.

Examples:

- Regression
- Classification

Unsupervised Learning

Data: Only the features, x, for each item in the dataset.

Goal: 'discover 'interesting' things about the dataset.

Examples:

- Clustering
- Principal Component Analysis (PCA)

Clustering

Goal: Discover unknown subgroups in data. Partition the dataset into groups where ‘similar’ items are in the same group and ‘dissimilar’ items are in different groups.

Examples:

- Social Network Analysis: Clustering can be used to find communities
- Handwritten digits where the digits are unknown
- Ecology: cluster organisms that share attributes into species, genus, etc…

To make clustering concrete, we must define what it means for items to be ‘similar’.

Examples: Euclidean distance, Pearson correlation, Manhattan distance, weighted distances, Jaccard coefficient, ….

Types of Clustering Algorithms

Partition Algorithms

- divide data into set of bins
- # bins either manually set (e.g., k-means) or automatically determined (e.g., affinity propagation)

Hierarchical Algorithms

- Produce “similarity tree” – dendrogram
- discrete cluster can be produced by “cutting” a dendrogram

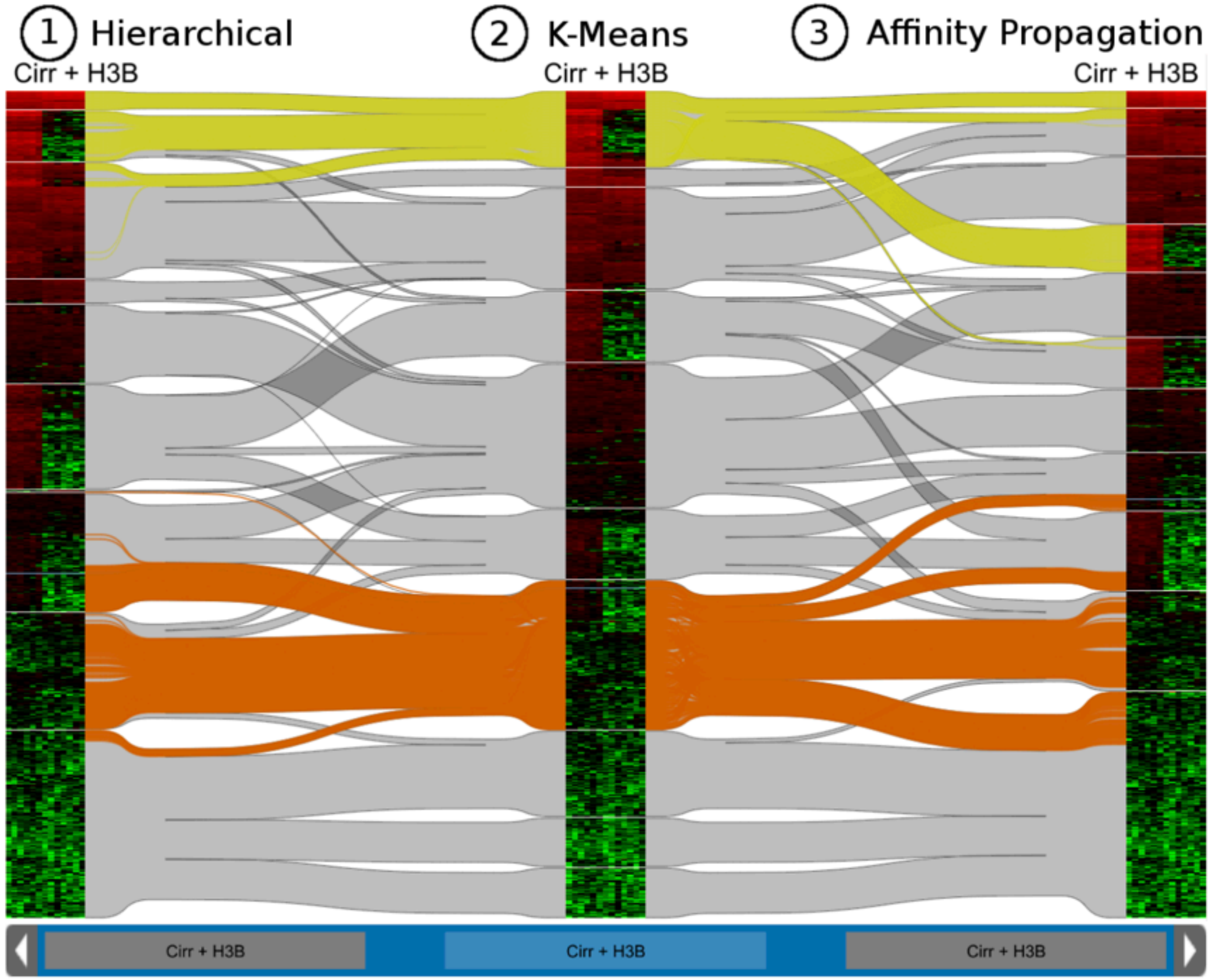
Bi-Clustering

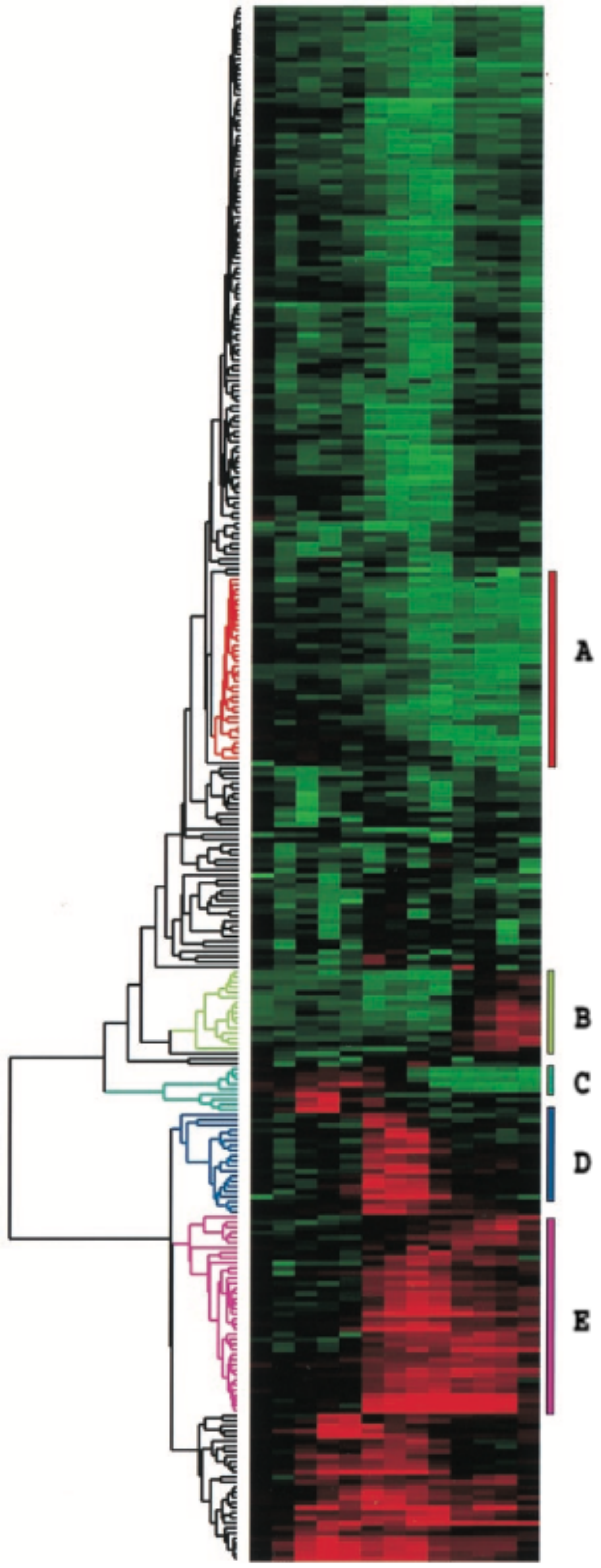
- Clusters dimensions & records

Fuzzy clustering

- probabilistic cluster assignment
- allows occurrence of elements in multiples clusters

Visualization: Important to judge cluster quality





K-means clustering

Goal: Find a collection clusters, C_i, with centers, μ_i, and assign each datapoint to a cluster, as to minimize the aggregate intra-cluster distance (*inertia*)

$$\underset{C}{argmin} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

Each summand is the squared distance from the point to the center of its cluster

The inner sum measures the inherent spread for each cluster

Lloyd's Algorithm for k-means

Algorithm:

Input: set of features $x_1 \ldots x_n$, and k (nr clusters)

Pick k starting points as centers, $mu_1 \ldots mu_k$

While not converged:

1. assign each point x_i to the closest center, mu_j
2. for each cluster j, compute a new center mu_j
   by calculating the mean of all x_i assigned to cluster j

Typical convergence criteria:

- no point has changed cluster
- distance between old and new centroid below threshold
- number of max iterations reached

Properties:

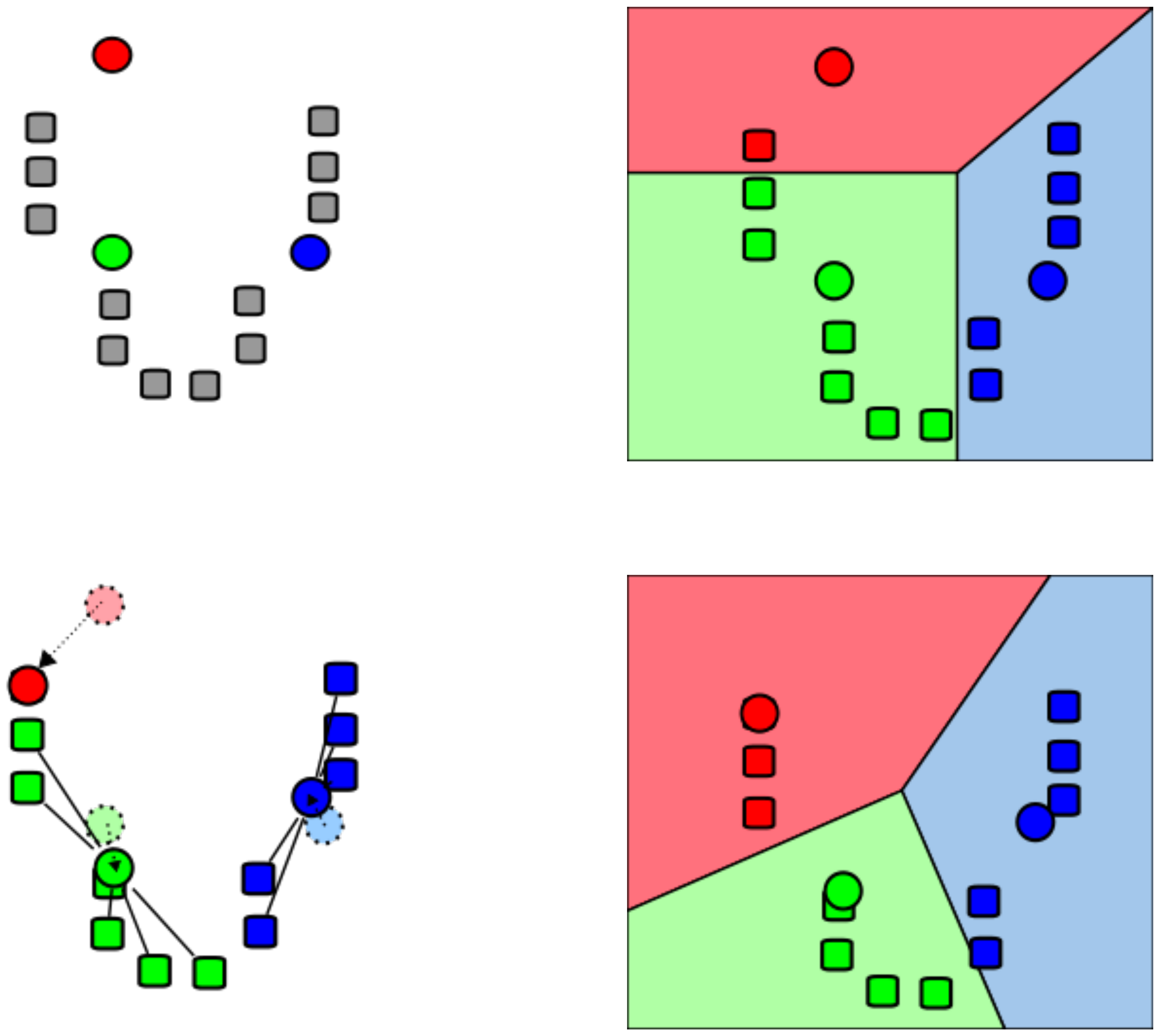
- Converges to a local optimum, not necessarily global
- In practice: run multiple times and pick the solution with min inertia
- Convergence is $O(n*k*d*i)$ where

  n is the number of records,

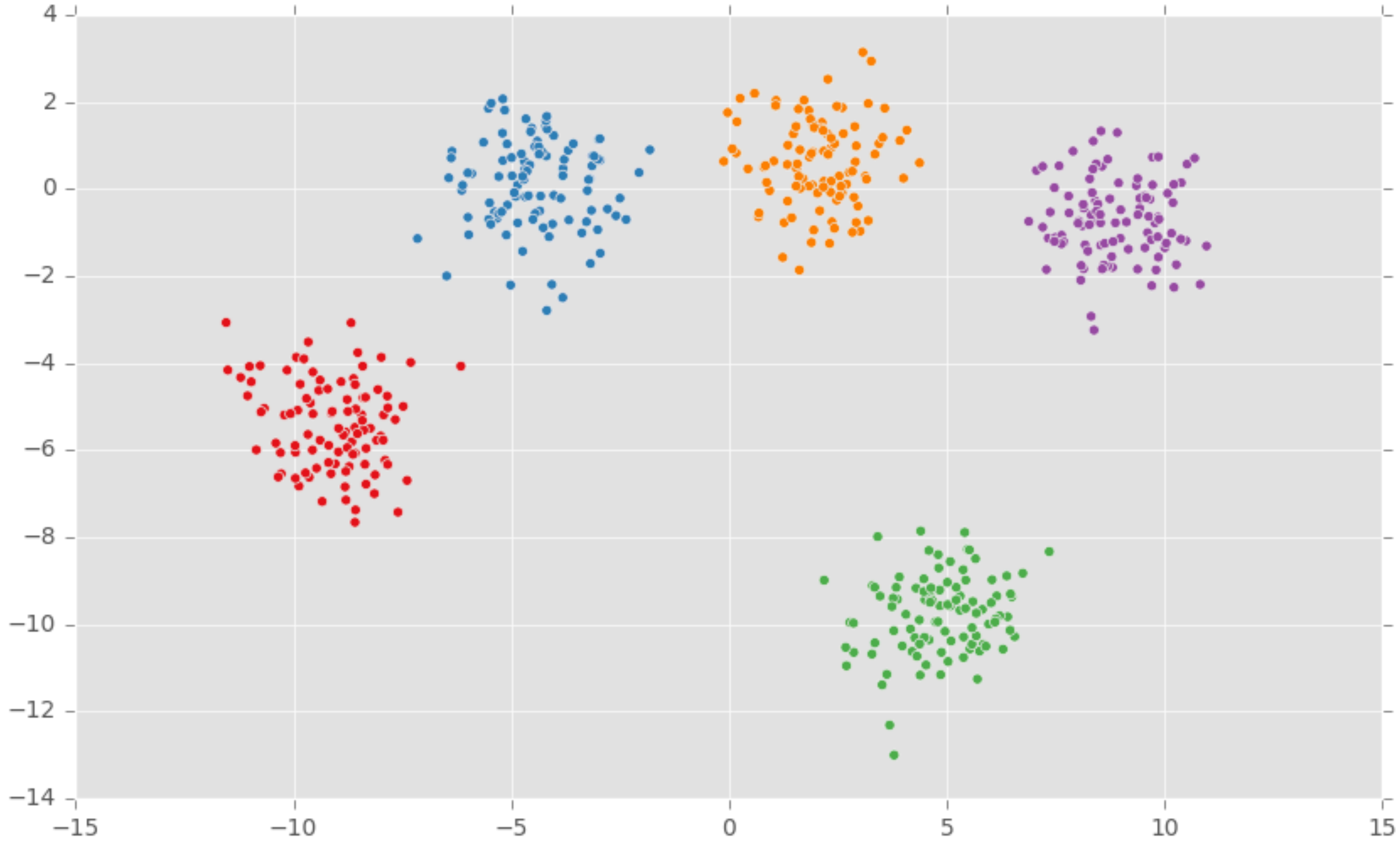
  k is the number of clusters

  d is the number of dimensions

  i is the number of iterations needed until convergence
- In practice: i small and very fast
- As a function of k, the inertia of the optimal partition is decreasing
- Decision boundaries give convex sets



In the previous lecture, we saw how to use scikit-learn to compute k-means clusters



Choosing the k in k-means?

In some applications, this is obvious

Example: If you're clustering handwritten digits and you know there are 10 digits, 0,1,…,9, it makes sense to choose k=10.

In many applications, there is no obvious choice.

Example: In ecology, it might not be obvious how many different species there are.

Practical Solution: Compute the optimal clusters for many different values of k and choose the value of k at the "elbow". Remember, the inertia is increasing with k.

Hierarchical Clustering

Hierarchical Clustering

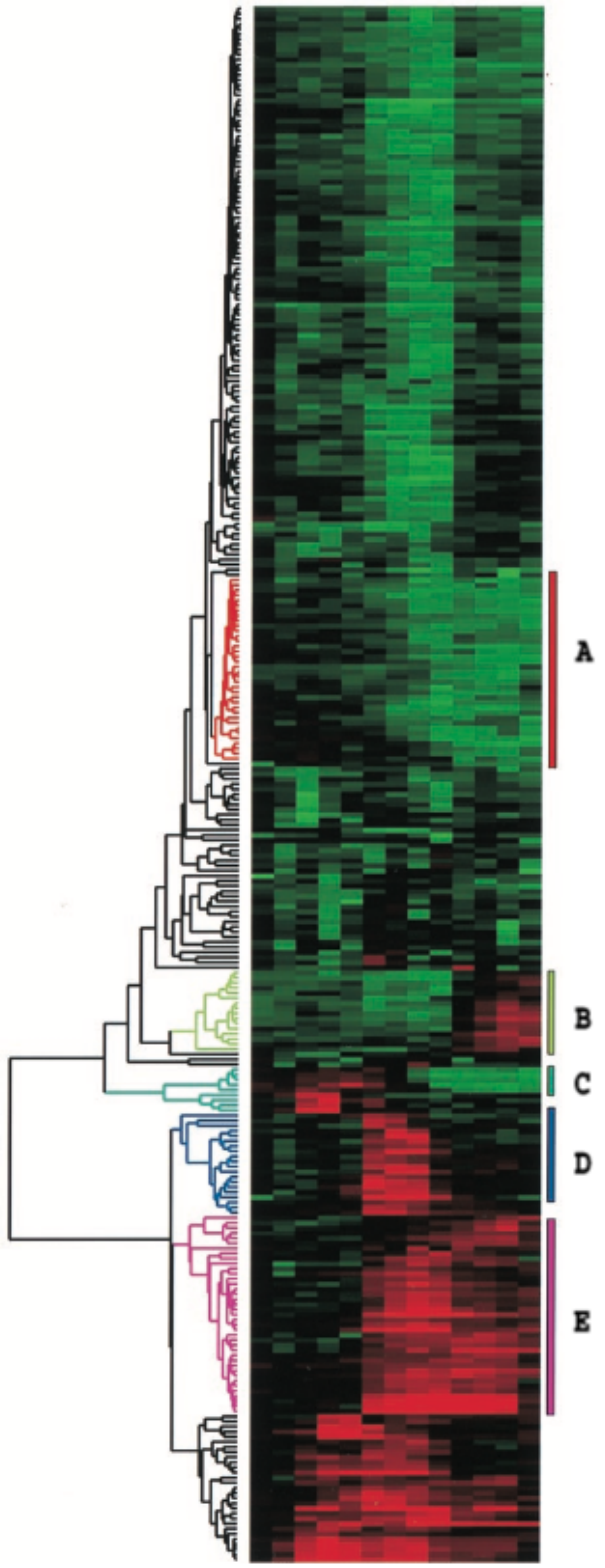
Two ways you can go:

agglomerative clustering

start with each node as a cluster and merge clusters together until you're happy with the cluster

divisive clustering

start with one cluster and split you're happy with the cluster



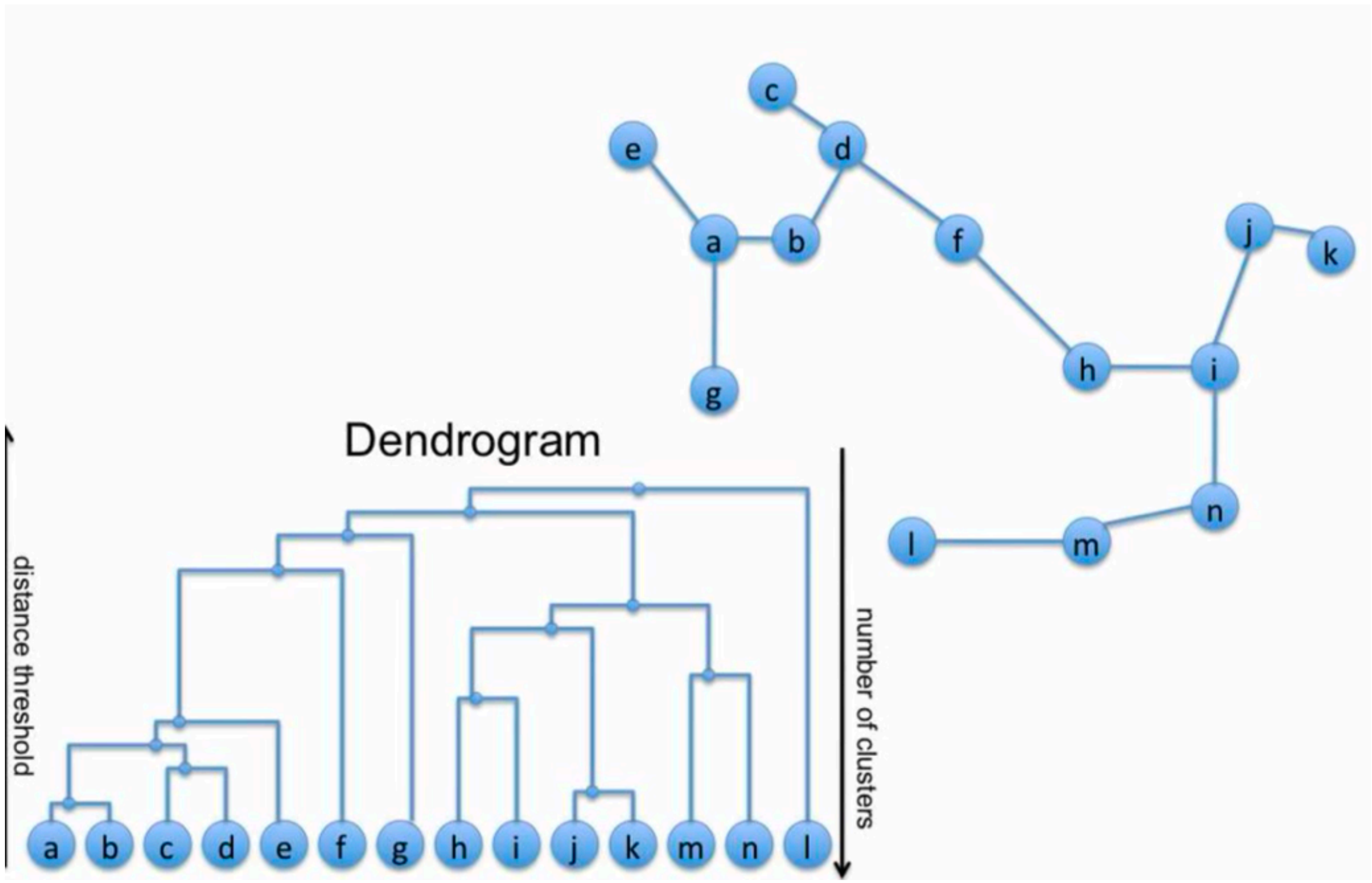
Agglomerative clustering

Start with each item as it’s own cluster.

Group together the two clusters that are ‘closest together’.

Continue this process until there is only one cluster.

Using the dendrogram plot, decide which clustering is best.



Linkage Criteria

How do you define similarity between two clusters (A and B) to be merged?

- Maximum linkage distance, $\max\{d(a,b) \colon a \in A, b \in B\}$
- Minimum linkage distance, $\min\{d(a,b) \colon a \in A, b \in B\}$
- Average linkage distance, $\frac{1}{|A||B|} \sum_{a \in B} \sum_{b \in B} d(a,b)$
- Centroid distance, if c_A and c_B are the centers of clusters A and B, then $d(c_A, c_B)$

Example measures of distance

- Euclidean distance, $d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$
- Manhattan distance, $d(x, y) = \sum_{i=1}^{d} |x_i - y_i|$
- Correlation
- If A and B are two sets, we define the *Jaccard similarity coefficient*,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

  We always have that $0 \leq J(A, B) \leq 1$. We then define the *Jaccard distance* as

$$d(A, B) = 1 - J(A, B).$$