Unsupervised Learning

Where are we?

- Parametric vs. non-parametric
- Generative vs. Discriminative
- Supervised vs Unsupervised
  - Supervised:
    - Given X, predict Y
  - Unsupervised:
    - Given X… do something interesting…
    - assumes that X has some structure

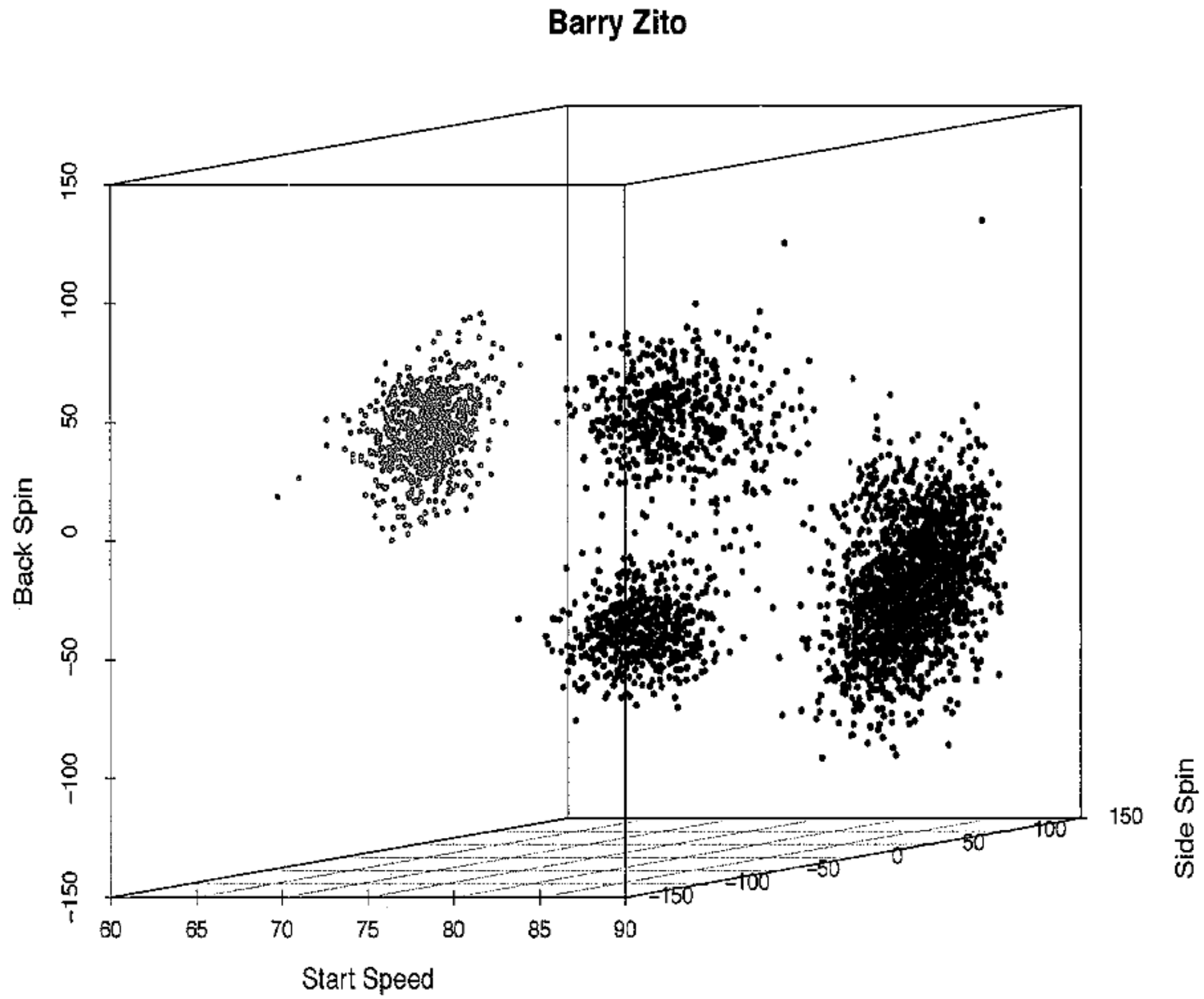
Discovering Structure

- Clustering
  - Discover a partitioning of the data into tight clusters
- Dimensionality Reduction
  - discover low-D manifolds
  - discover good features
- Mode seeking
  - Discover frequent patterns

Why unsupervised?

- Raw data cheap
- Want to discover something new:
  - Exploratory data analysis
  - Data mining
- Even if labels are available, they may be subjective (e.g. languagespecific), out-of-date, just wrong
  - better listen to the data

Clustering baseball pitches

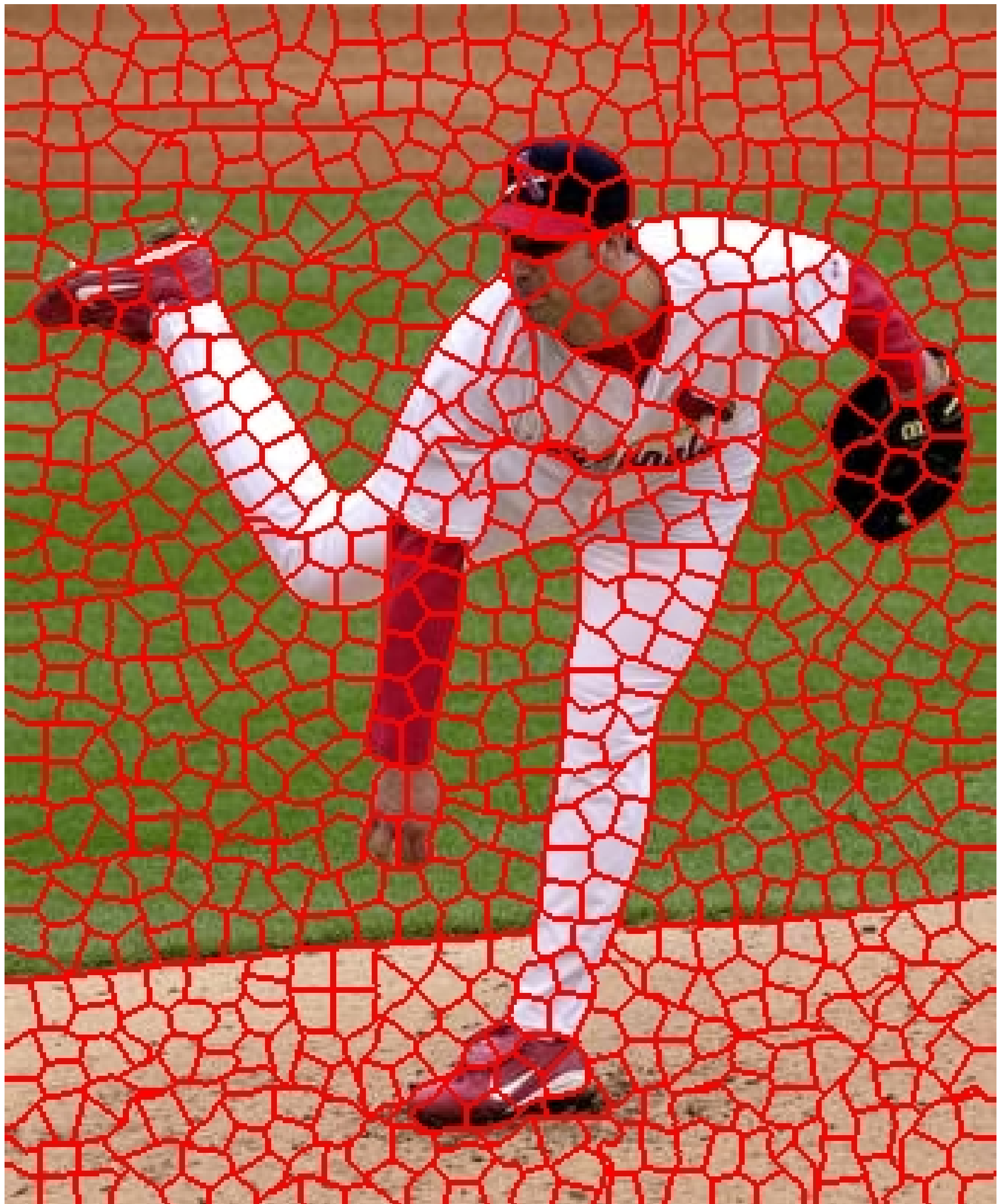


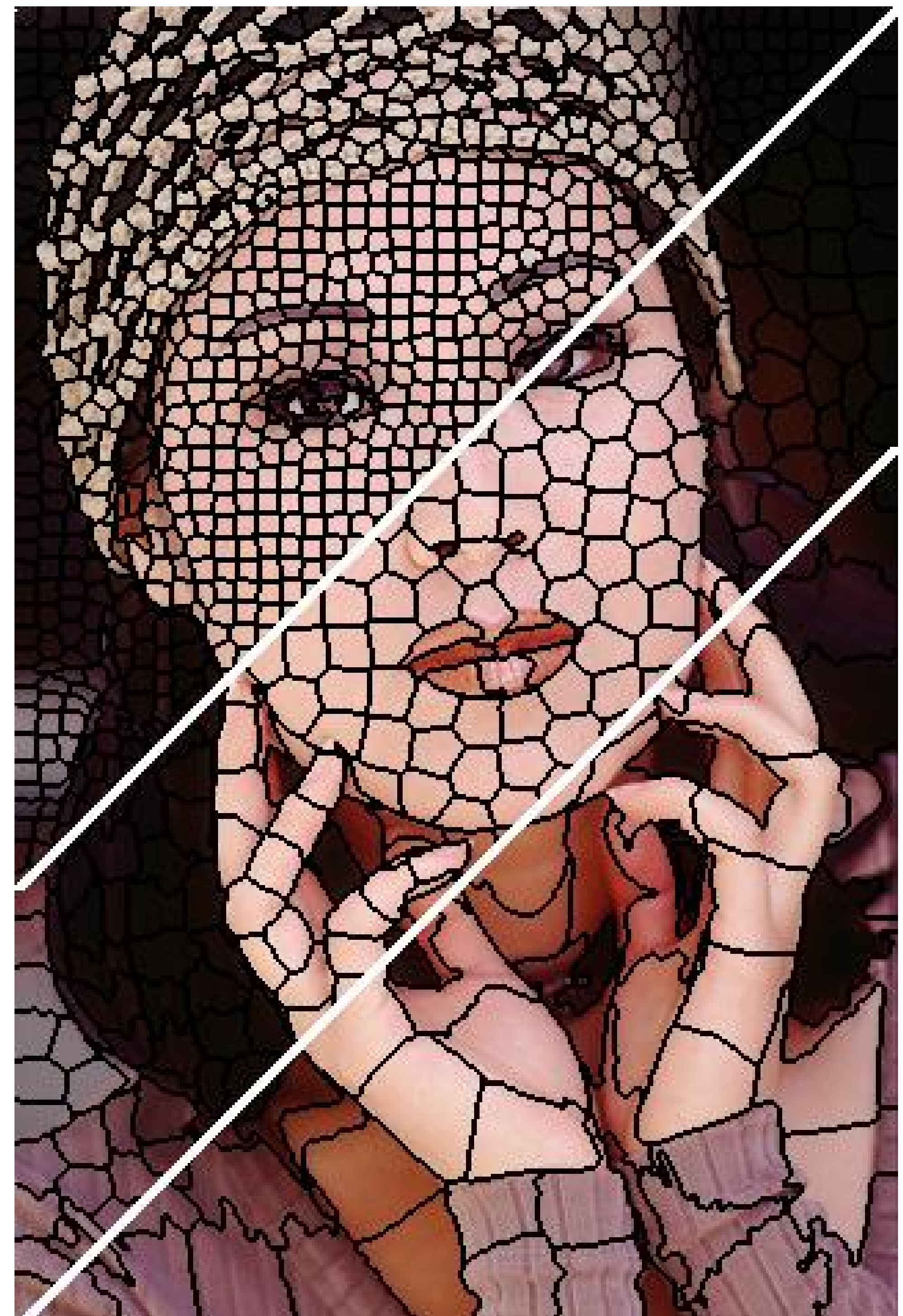
Inferred meaning of clusters: black – fastball, red – sinker, green – changeup, blue – slider, light blue – curveball

Image Segmentation

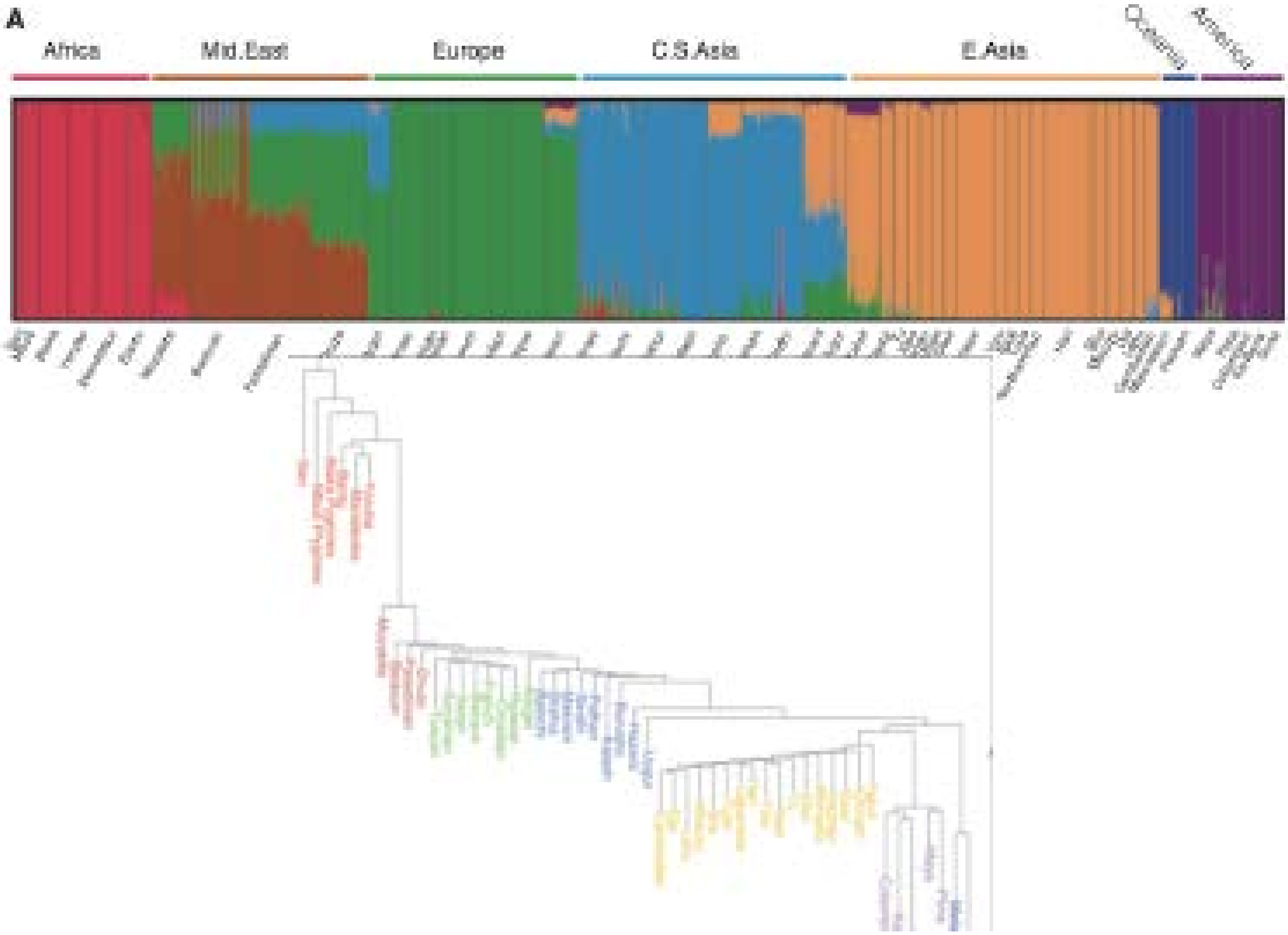
http://people.cs.uchicago.edu/ pff/segment

Super-pixels

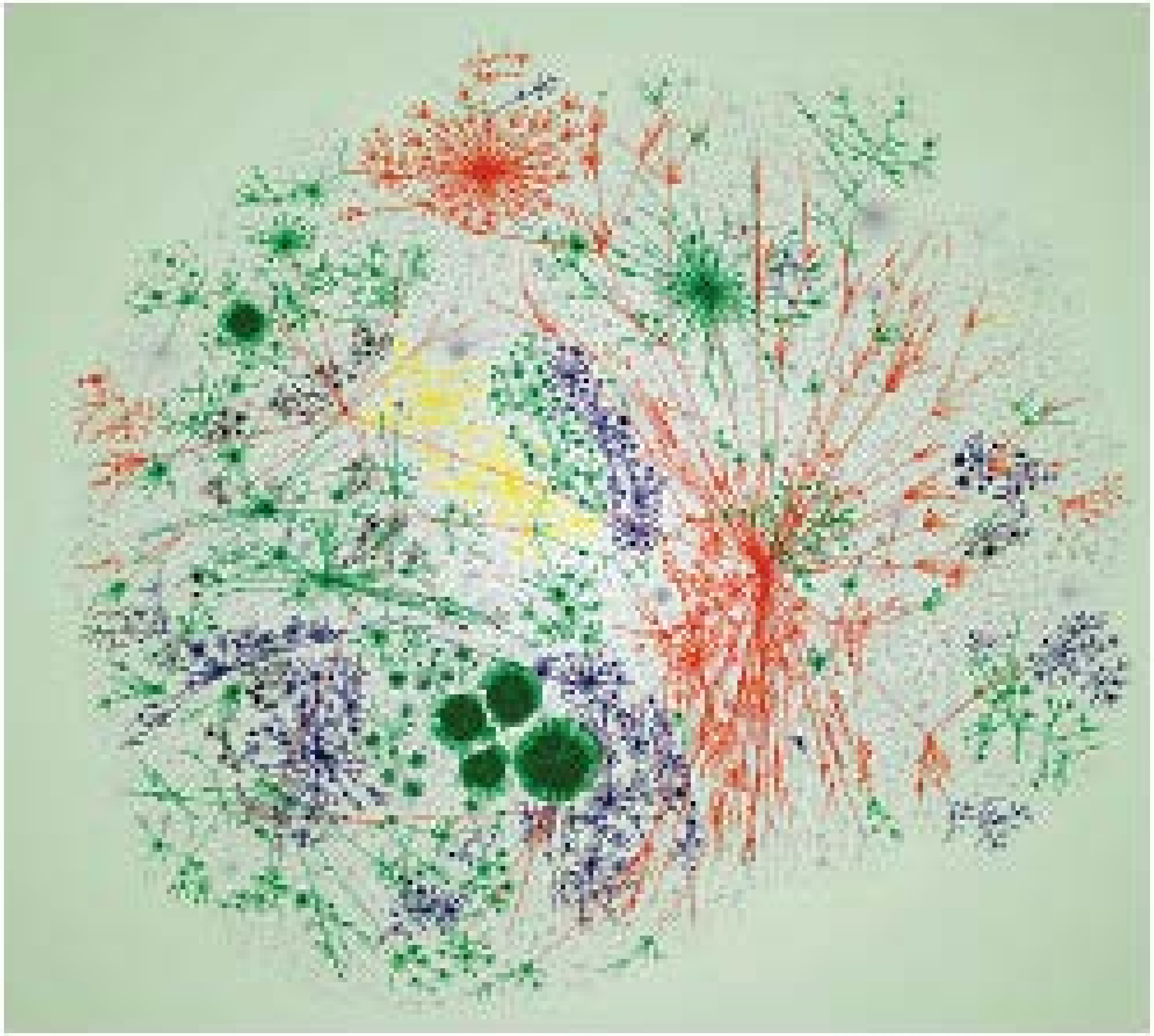




Human population structure



Clustering graphs



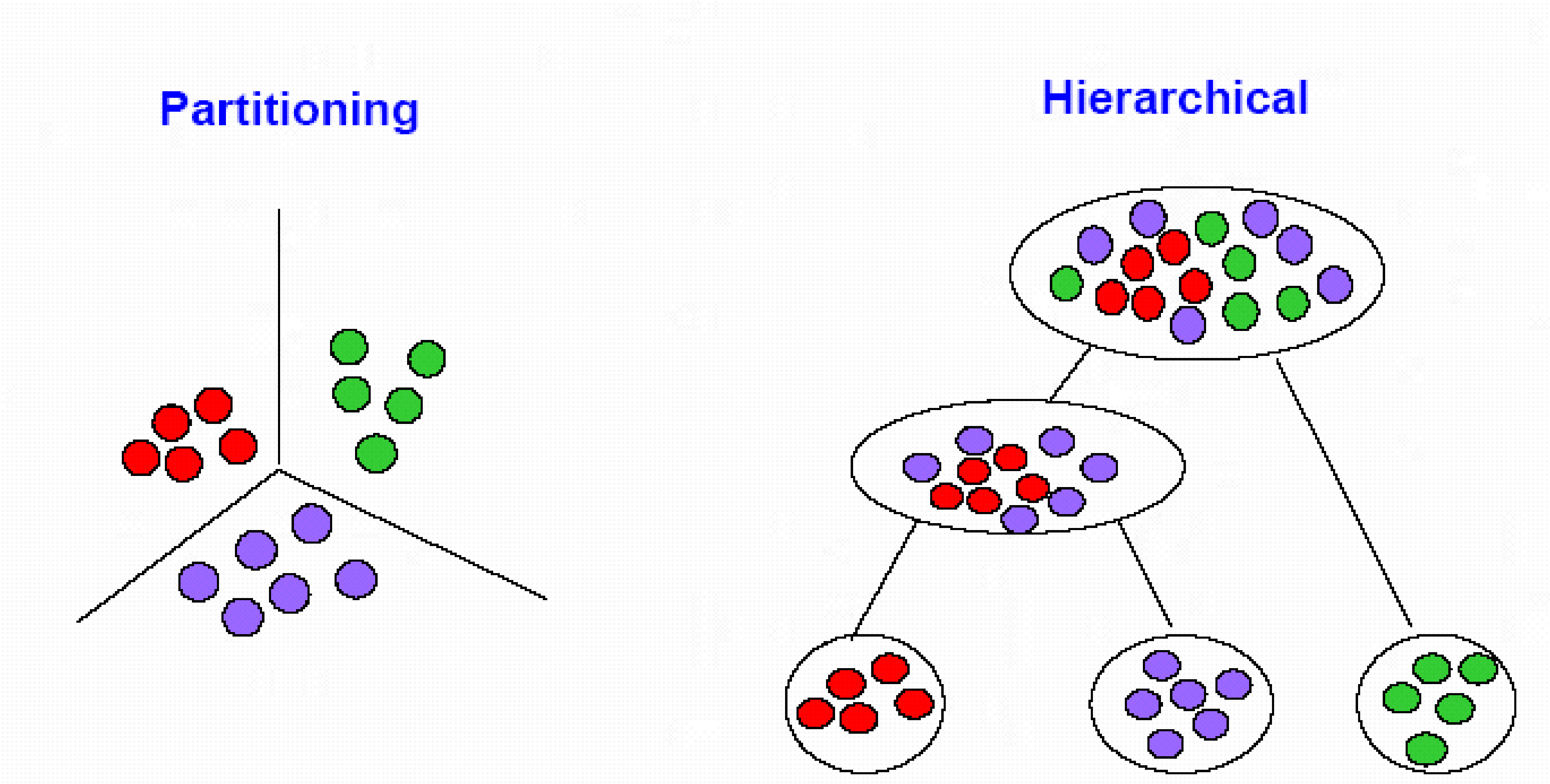
Newman, 2008

Market Segmentation

- hipsters
- “Soccer Moms”
- urban techies
- etc.

Clustering methods

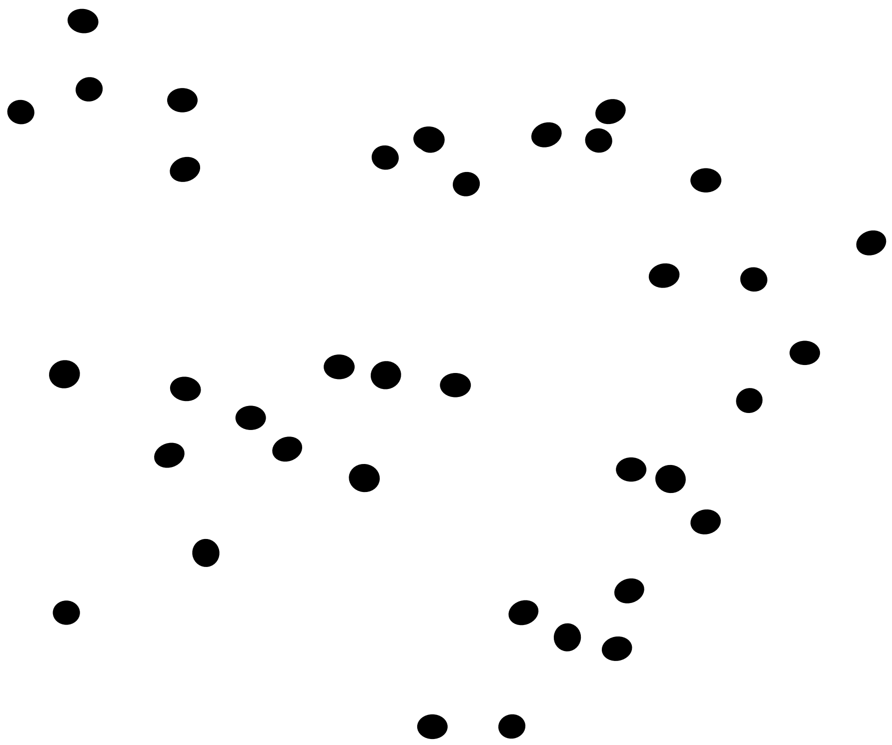
- Two popular flavors



Hierarchical clustering

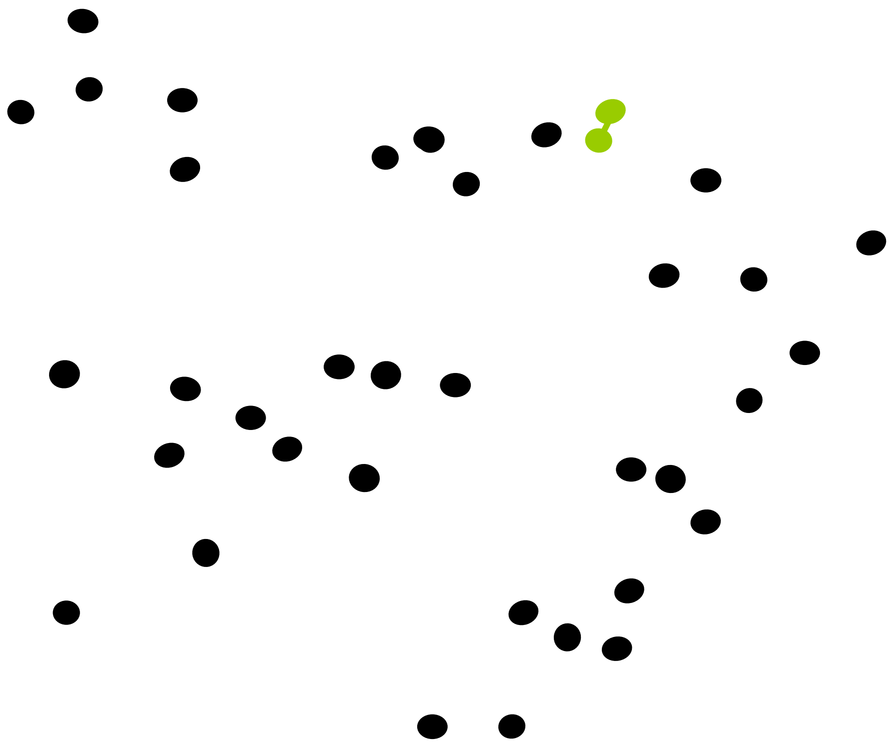
- Discovering a taxonomy (e.g. Linnaeus)
- Produce a tree or dendrogram.
- The tree can by built in two distinct ways
  - Bottom-up: agglomerative clustering (most used).
  - Top-down: divisive clustering.

Hierarchical Clustering

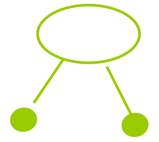


1. Say "Every point is its own cluster"

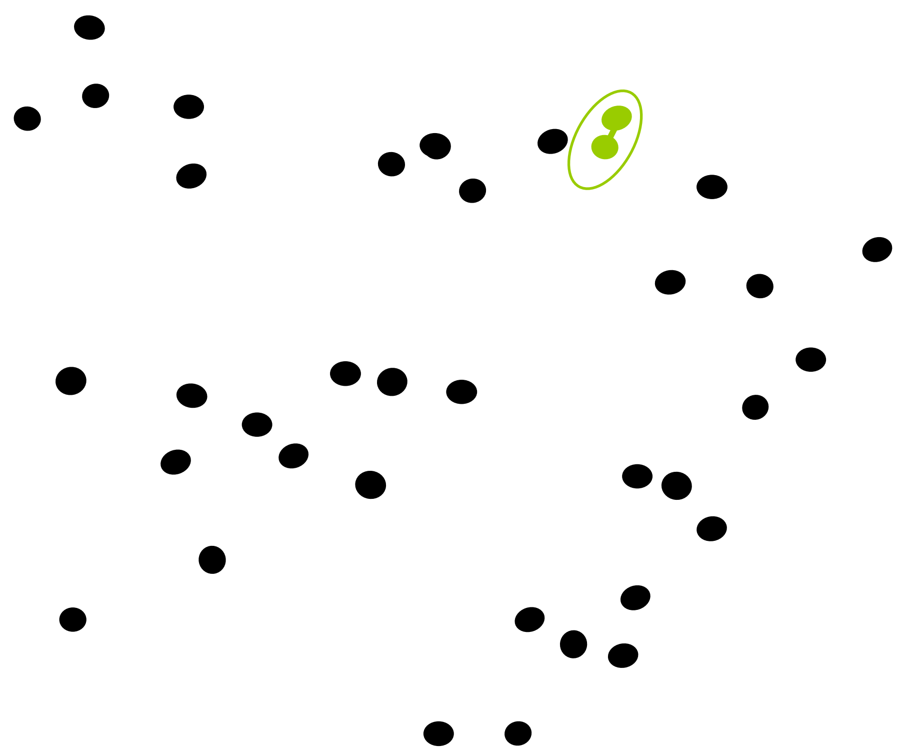
Hierarchical Clustering



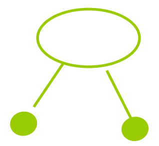
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters



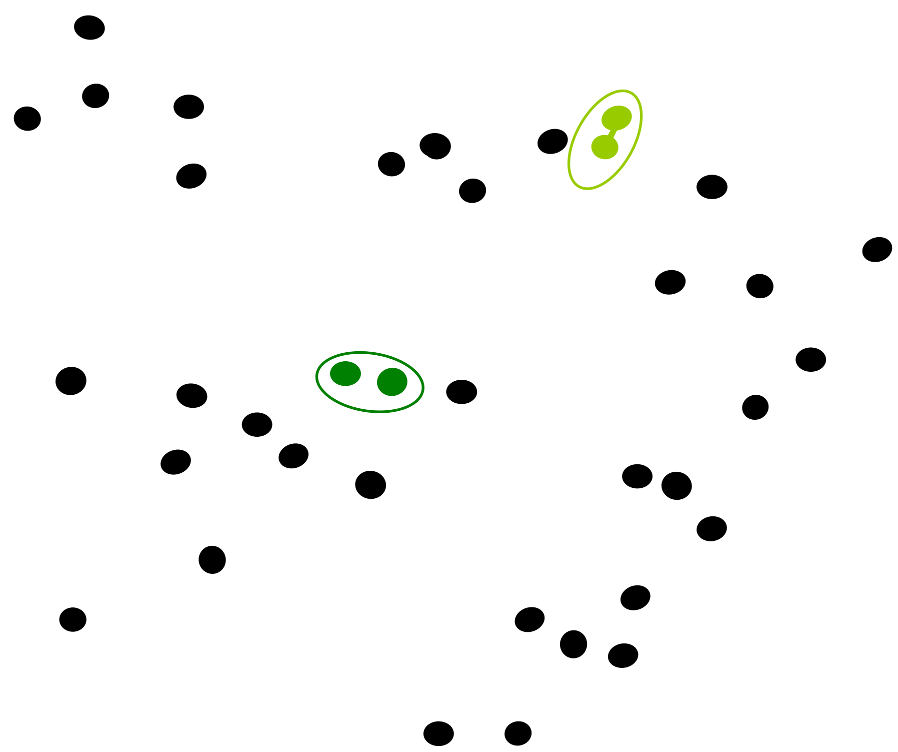
Hierarchical Clustering



1. Say “Every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster

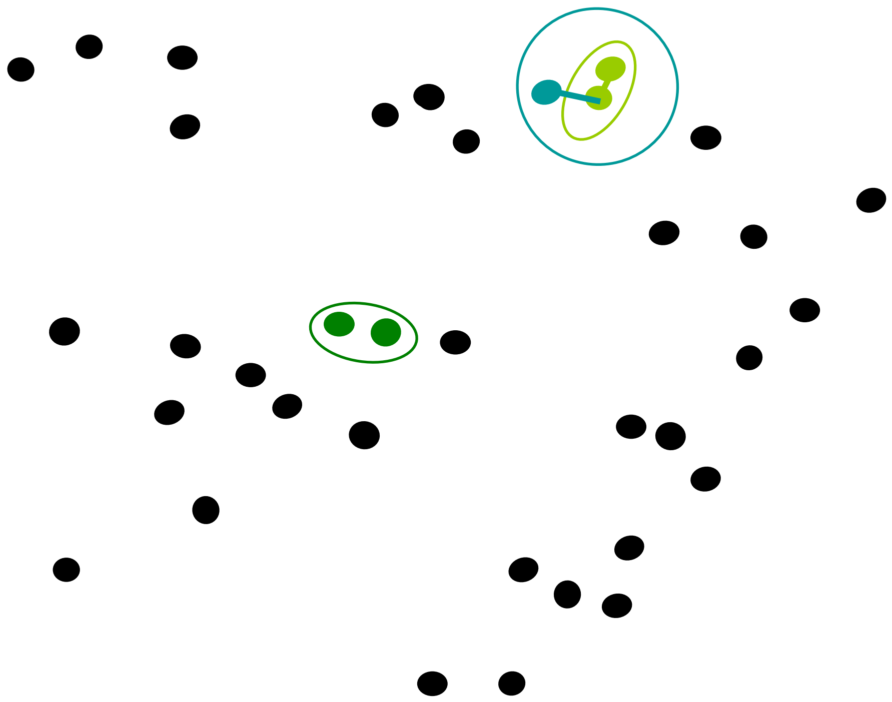


Hierarchical Clustering

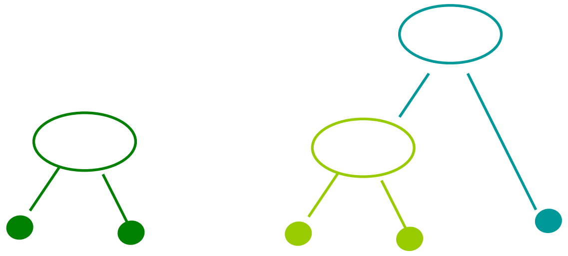


1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

Hierarchical Clustering



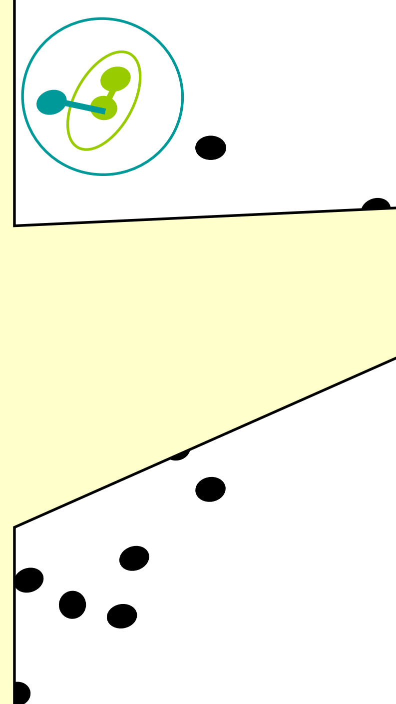
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



ical Clustering

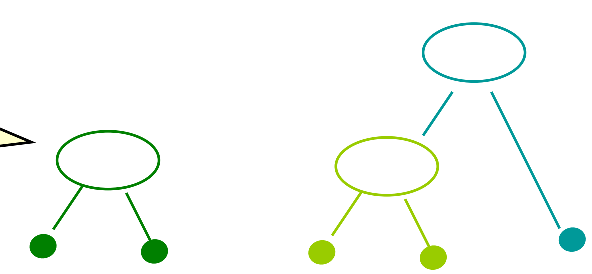
How do we define similarity between clusters?

- Minimum distance between points in clusters (in which case we're simply doing Euclidian Minimum Spanning Trees)
- Maximum distance between points in clusters
- Average distance between points in clusters



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat...until you've merged the whole dataset into one cluster

You're left with a nice dendrogram, or taxonomy, or hierarchy of datapoints (not shown here)

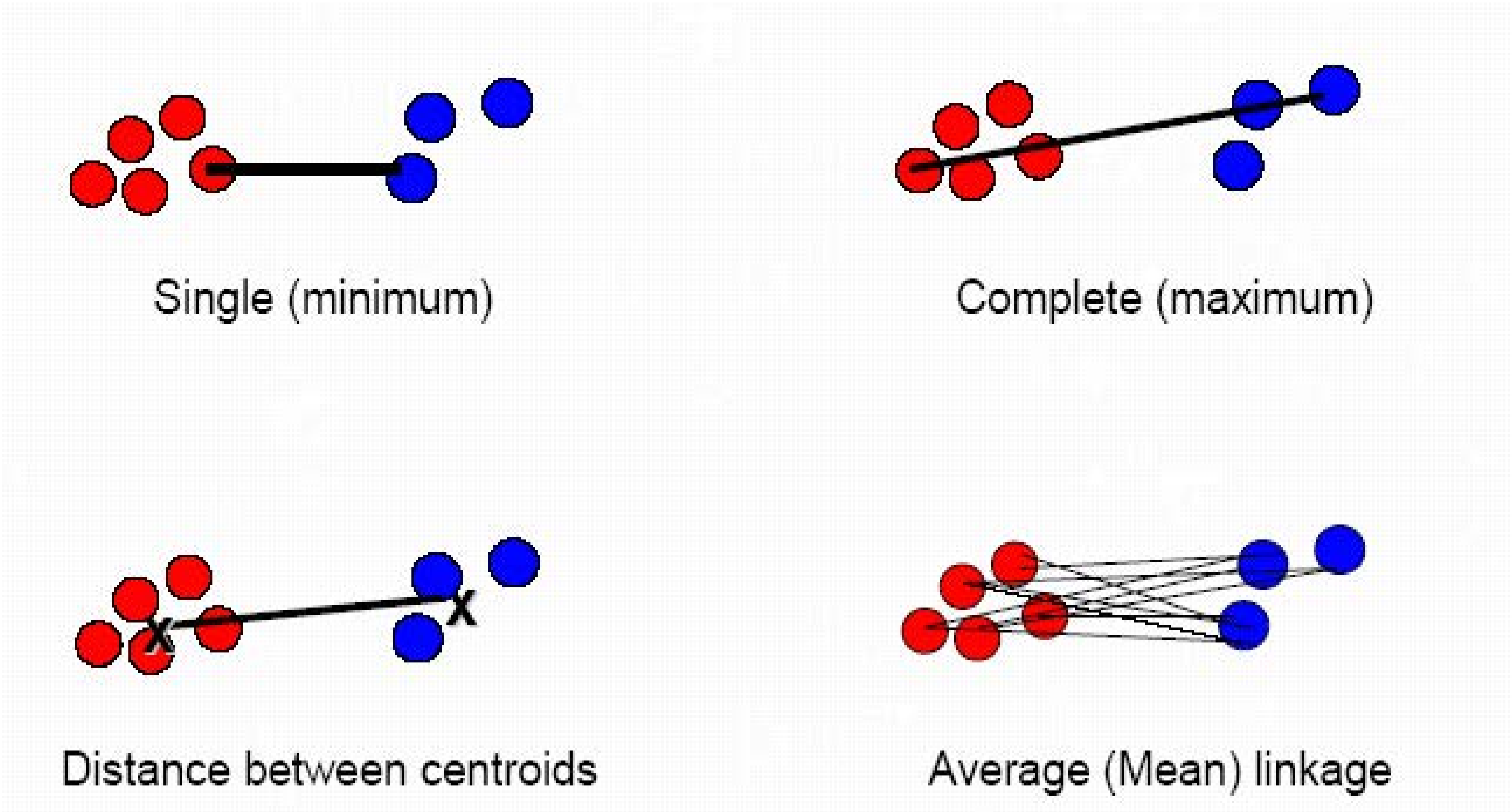


Distance is key

- Clustering is subjective:
  - e.g. “cat”, “chair”, “table”, “tiger”

- a) Need to define distance function (like before)
- b) Need to define distance function of groups

Choice of linkage



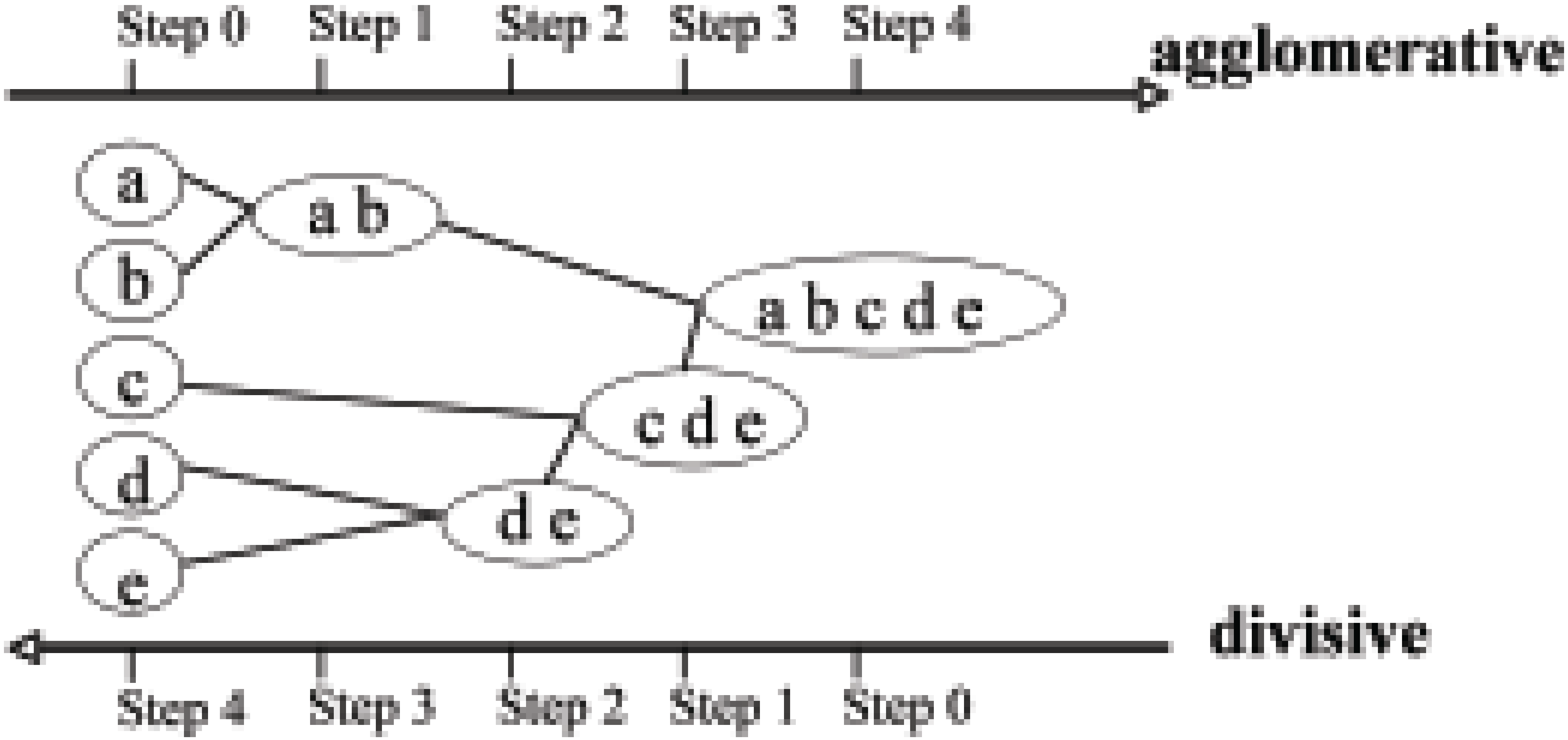
Group Similarities

- $dist\{S_a, S_b\}$:
  - –Single link: $dist\{S_a, S_b\} = \min_{i,j} d_{ij}$
  - –Complete link: $dist\{S_a, S_b\} = \max_{i,j} d_{ij}$
  - –Average link: $dist\{S_a, S_b\} = \sum_i \sum_j d_{ij}$
  - –Centroid link: $dist\{S_a, S_b\} = \left\|\overline{x_i} - \overline{x_j}\right\|$

Comparison of the three methods

- Single-link
  - Elongated clusters
  - Individual decision, sensitive to outliers
- Complete-link
  - Compact clusters
  - Individual decision, sensitive to outliers
- Average-link or centroid
  - “In between”
  - Group decision, insensitive to outliers.

Agglomerative vs. Divisive



Flat Partitioning methods

- Partition the data (size N) into a pre-specified number K of mutually excusive and exhaustive groups: a many-to-one mapping, or encoder k=C(i), that assigns the ith observation to the kth cluster.

- Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimization of a specific loss function

Loss Function

- A natural loss function would be the within cluster point scatter: $W(C)=\frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k}d\left(x_i,x_{i'}\right)$

For Euclidean distance…

- Choose the squared Euclidean distance as dissimilarity measure: $d(x_i, x_{i'}) = \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2$.
- Minimize the within cluster point scatter:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \left\| x_i - x_{i'} \right\|^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\| x_i - \bar{x}_k \right\|^2$$

- Where $\bar{x}_k = \left(\bar{x}_{1k}, \ldots, \bar{x}_{pk}\right)$ are cluster means.

Partitioning method

- In principle, we simply need to minimize W over all possible assignments of N objects to K clusters.
- However, the number of distinct assignment, $S(N,K) = \frac{1}{K!}\sum_{k=1}^{K}(-1)^{K-k}\binom{K}{k}k^N$ grows rapidly as N and K goes large.

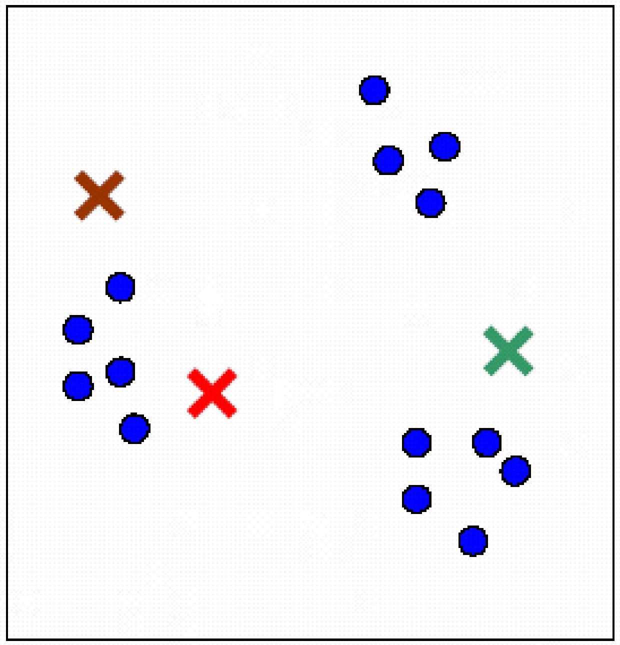
K-means

- In practice, we can only examine a small fraction of all possible encoders.
- Such feasible strategies are based on iterative greedy descent:
  - An initial partition is specified.
  - At each iterative step, the cluster assignments are changed in such a way that the value of the criterion is improved from its previous value.

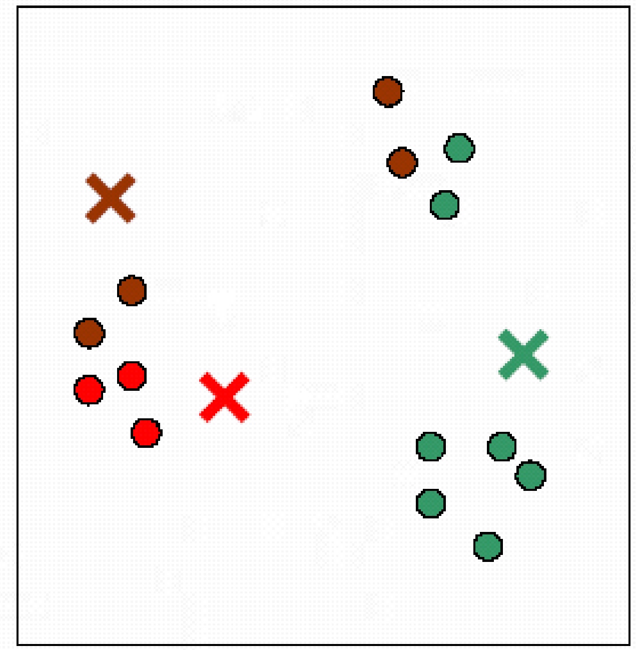
K-means Algorithm

1. Initialize cluster centroids $\mu_1, \ldots, \mu_K$ randomly
2. Repeat until convergence:
   – For every point i, assign it to the closest cluster k:
   $$C_i = \arg\min_k \|x_i - \mu_k\|^2$$
   – For every cluster k, recompute its mean μ_k:
   $$\mu_k = \frac{\sum_{C_i=k} x_i}{\sum_{C_i=k} 1}$$

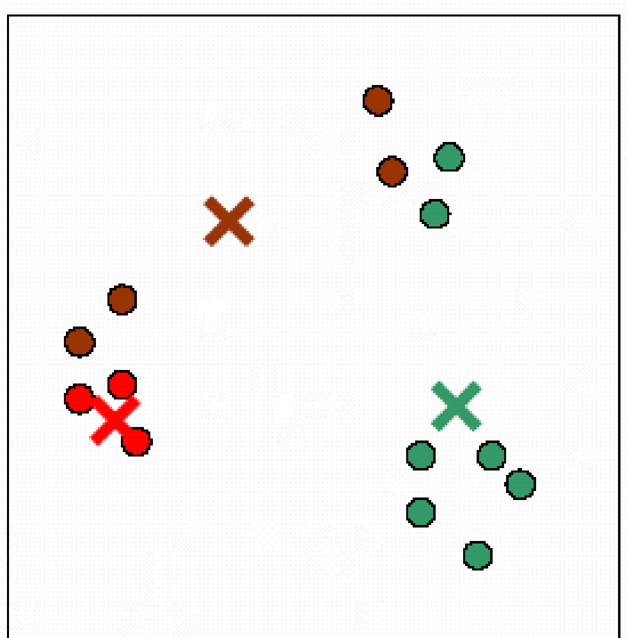
K-means example



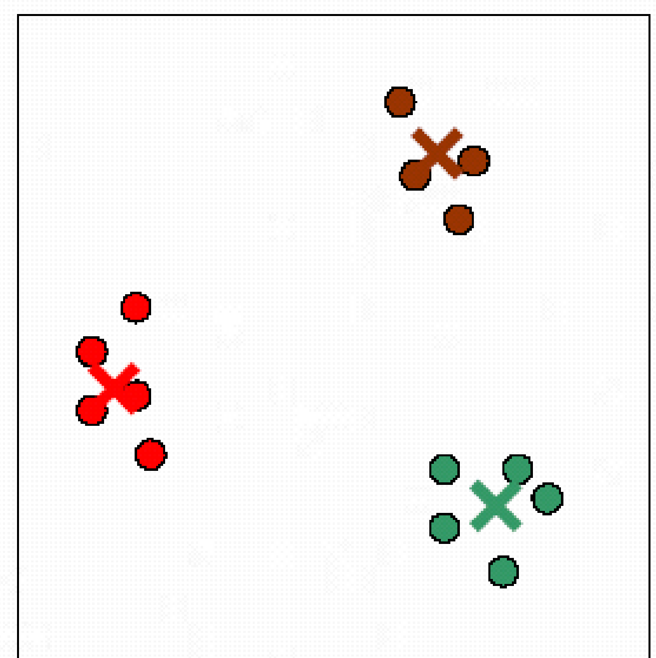
Iteration = 0



Iteration = 1



Iteration = 2



Iteration = 3

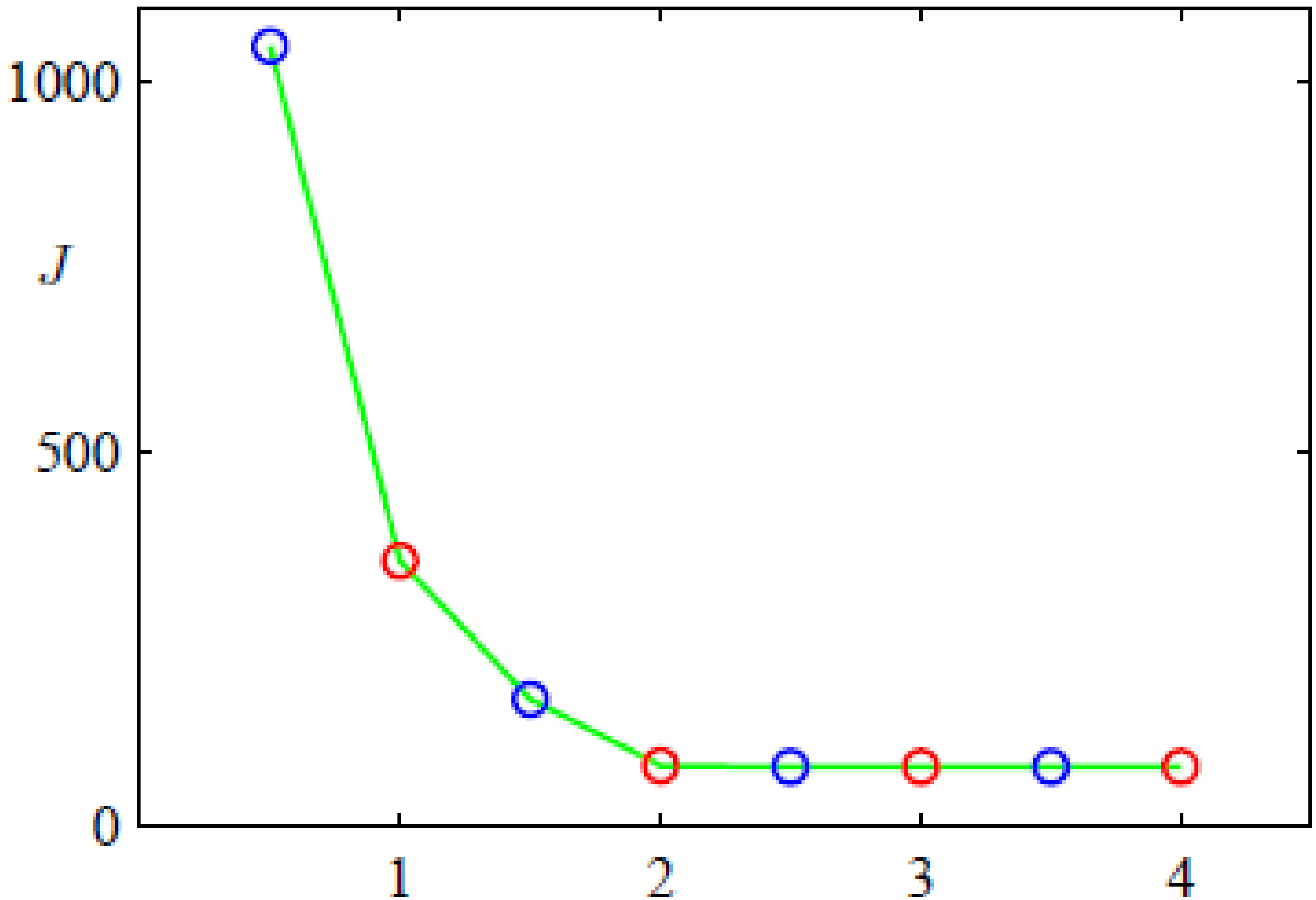
K-means Convergence

- Objective Function:

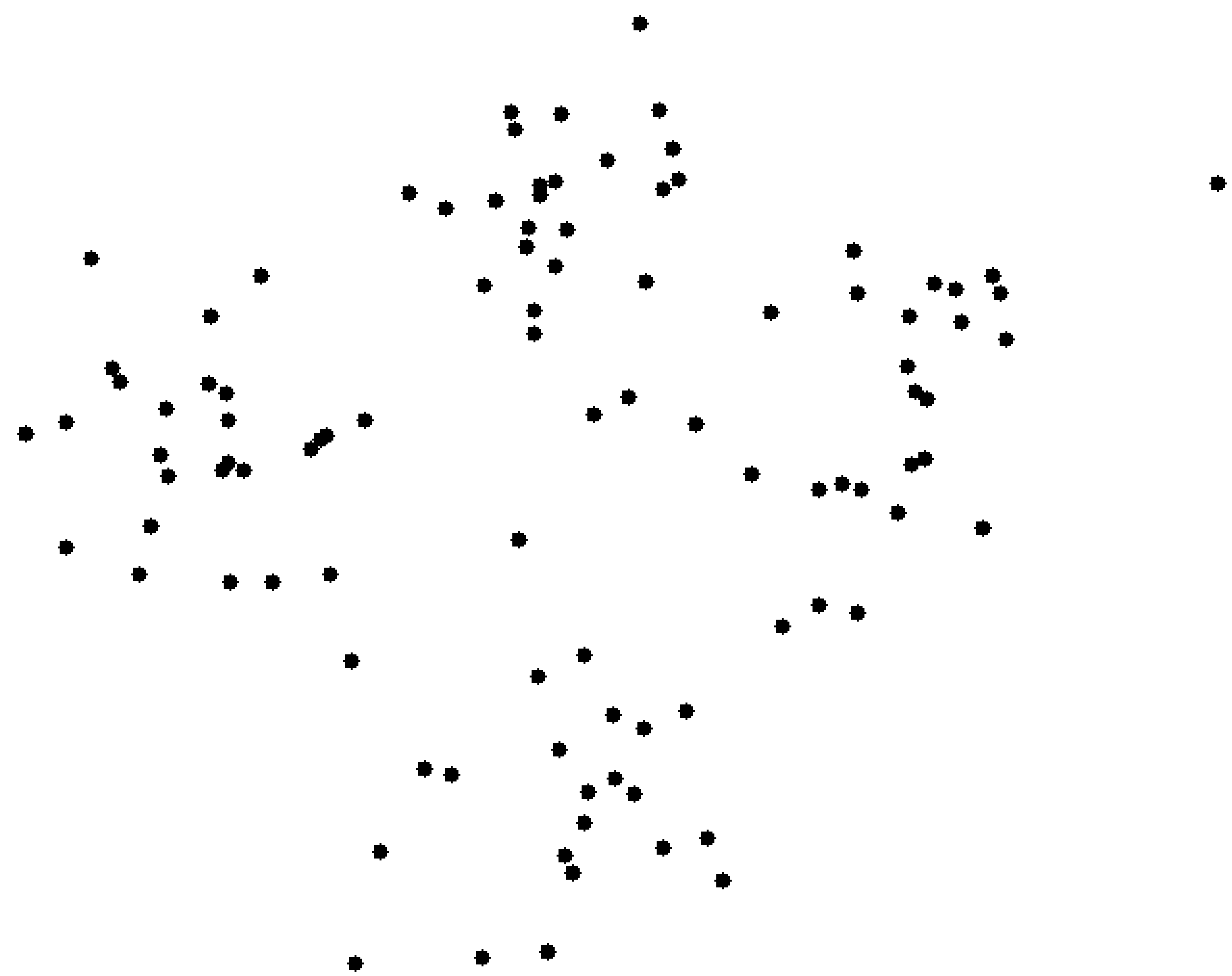
$$J(C,\mu) = \sum_{i=1}^{N} \left\| x_i - \mu_{C_i} \right\|^2$$

- K-means is coordinate descent on J, i.e. inner loop repeatedly:
  - –Minimize J with respect to C while holding μ fixed
  - –Minimize J with respect to μ while holding C fixed
- Thus J must monotonically descrease

Objective Function J after each iteration



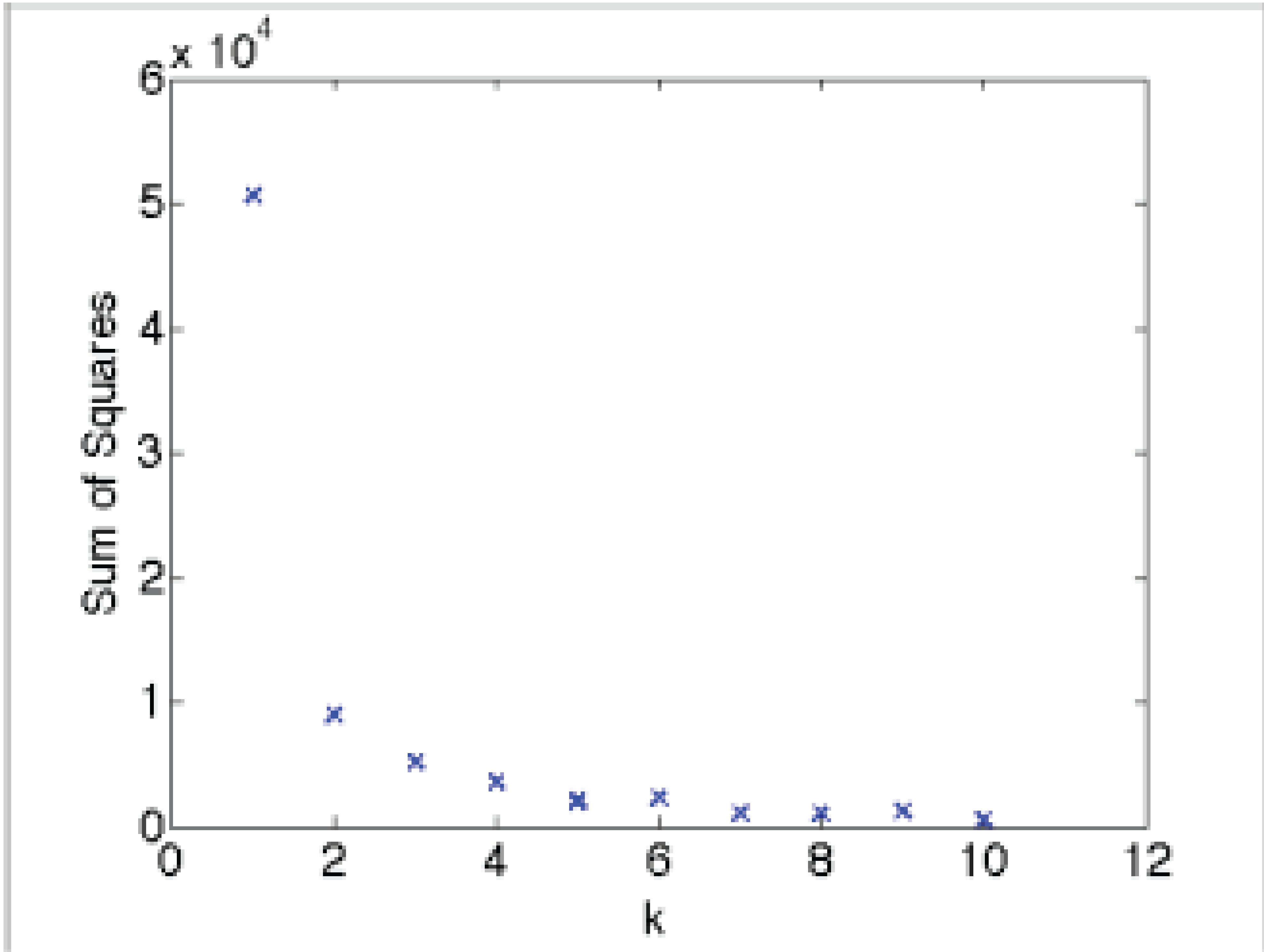
But only local minima



Avoiding bad local minima?

- Multiple restarts
- Better initialization
  - E.g. K-means++

How do we choose K?



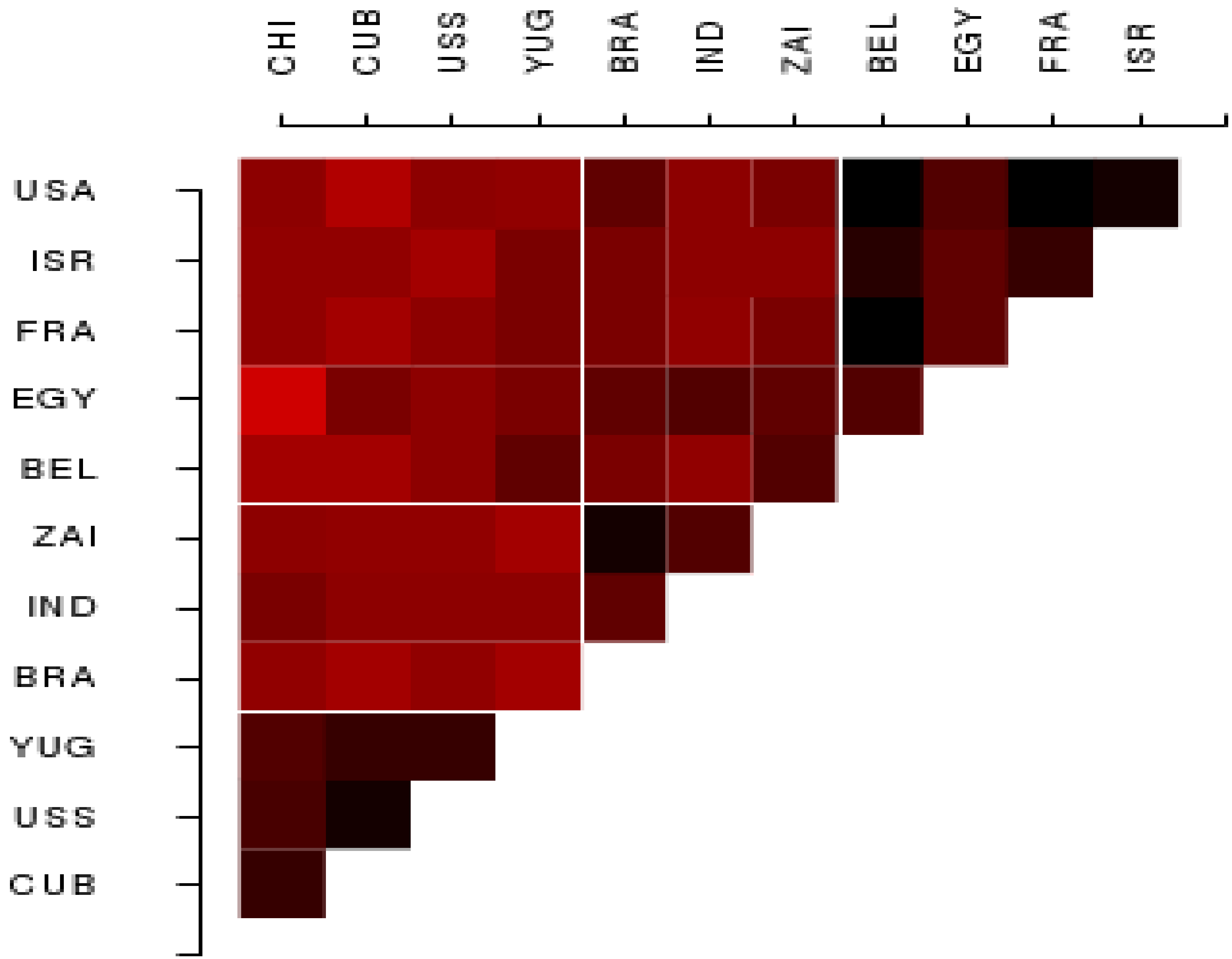
Beyond L2

- L2 distance is often not robust
- For some distance metrics, mean might not be well-defined / desirable
- Only dissimilarity matrix might be given

TABLE 14.3. *Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.*

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

K-Medoids Algorithm

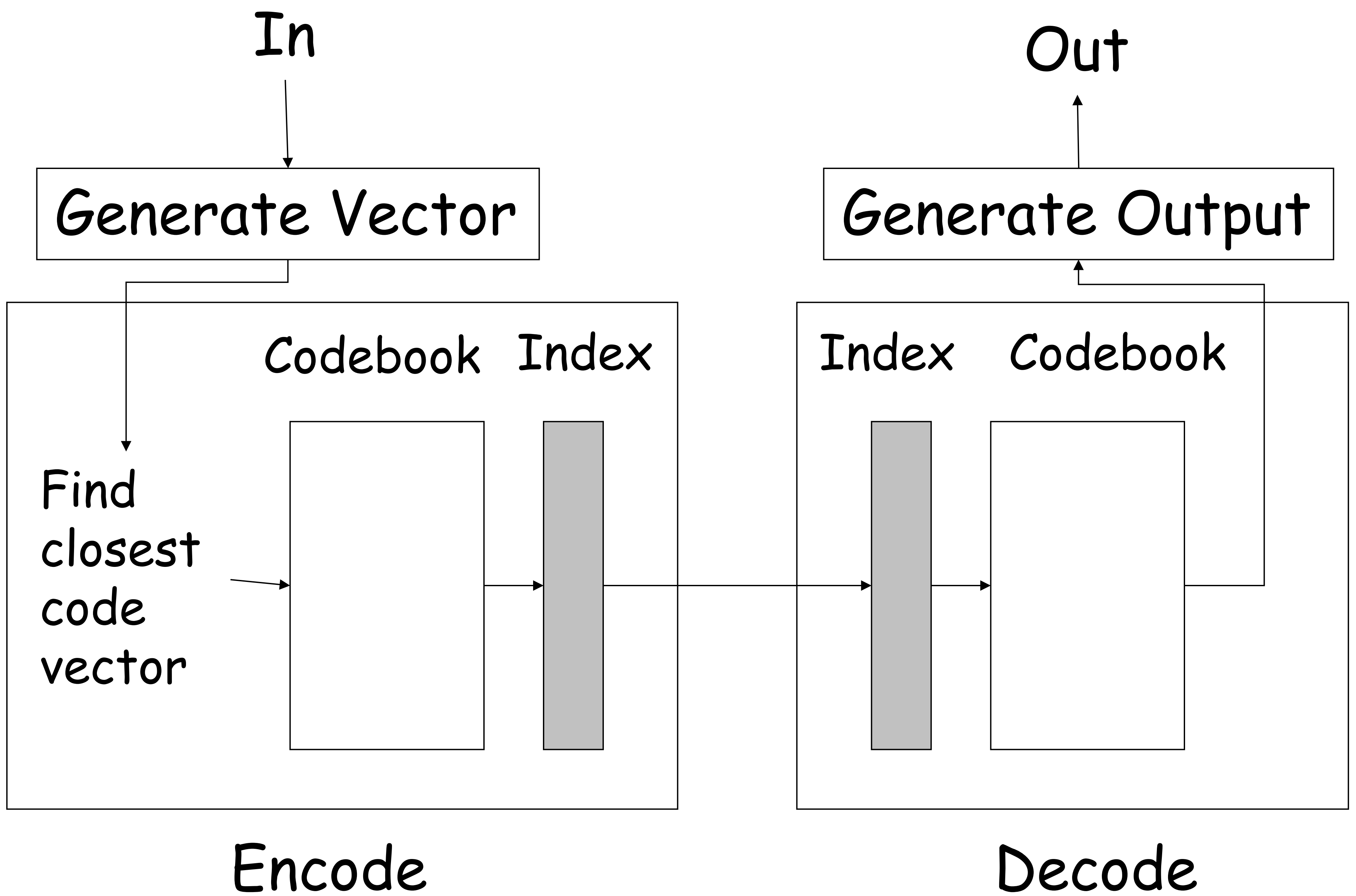


Reordered Dissimilarity Matrix

Online (streaming) K-means
(original Lloyds's algorithm)

- For each new point x
  - Find its closest center m
  - move m towards x “a little”
- Rinse and repeat

Vector Quantization



Vector Quantization

What do we use as vectors?

- Color (Red, Green, Blue)
  - Can be used, for example to reduce 24bits/pixel to 8bits/pixel
  - Used in some terminals to reduce data rate from the CPU (colormaps)
- K consecutive samples in audio
- Block of K pixels in an image

How do we decide on a codebook

- Typically done with **k-means**

Block Image Compression by VQ

8 bits/pixel

1.9 bits/pixel,
using 200 codewords

0.5 bits/pixel,
using 4 codewords

VQ doesn’t require “clustering”