

Unsupervised Learning: K-Means and DBSCAN Clustering Algorithm

Panthadeep Bhattacharjee
Dept. of CSE, NIT Rourkela

panthadeep.edu@gmail.com



Clustering

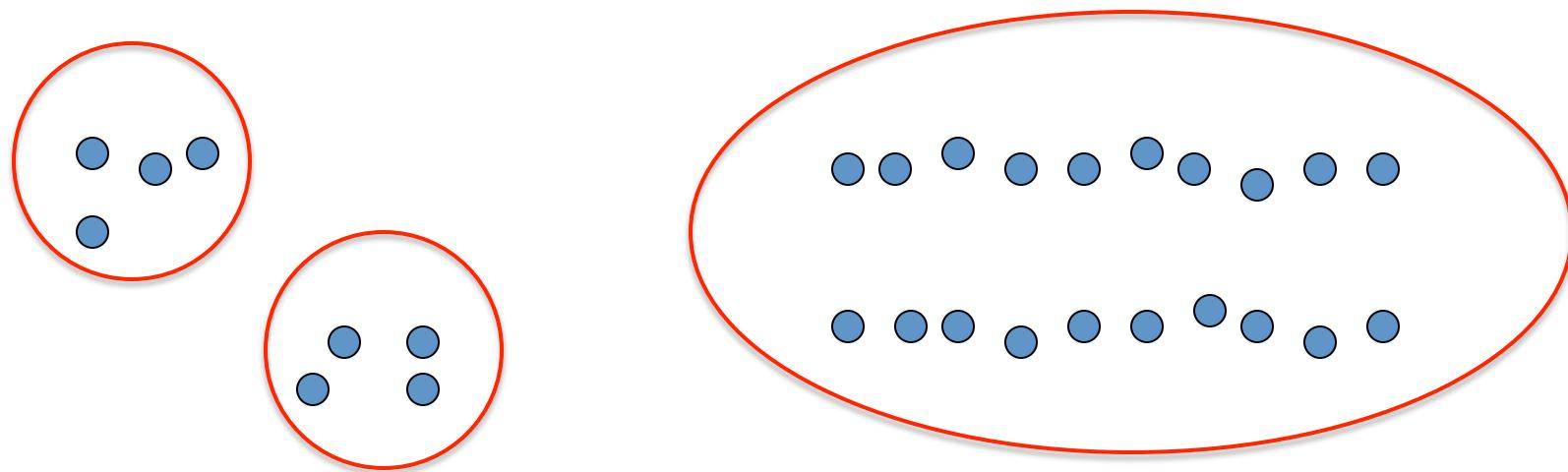
Clustering:

- Unsupervised learning
- Requires data, but no labels
- Detect patterns e.g. in
 - Group emails or search results
 - Customer shopping patterns
 - Regions of images
- Useful when don't know what you're looking for
- But: can get gibberish



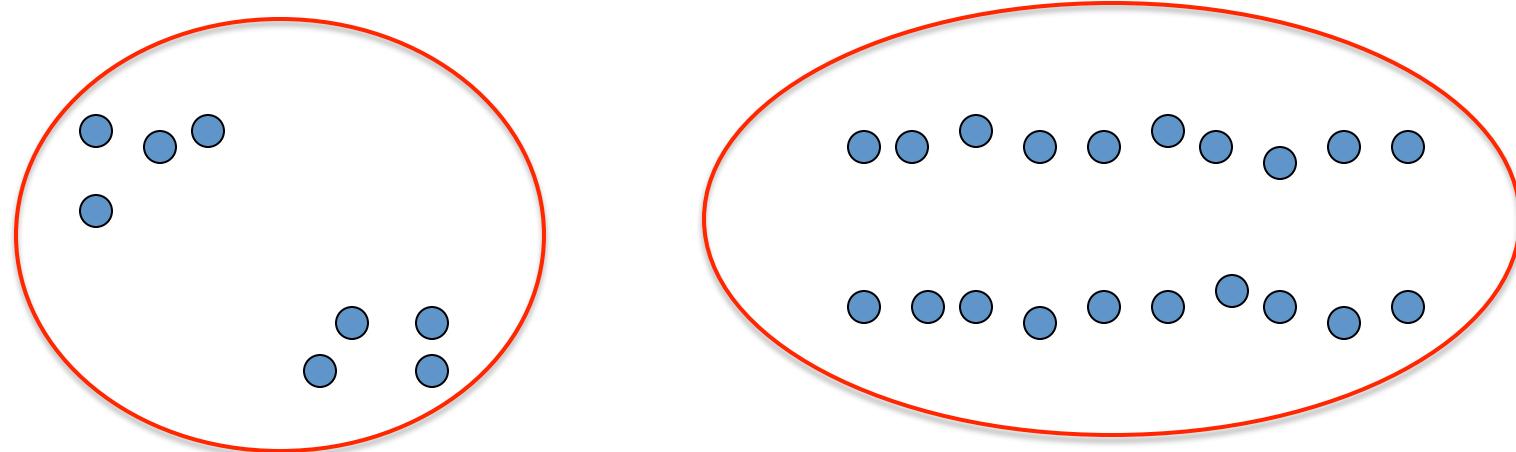
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



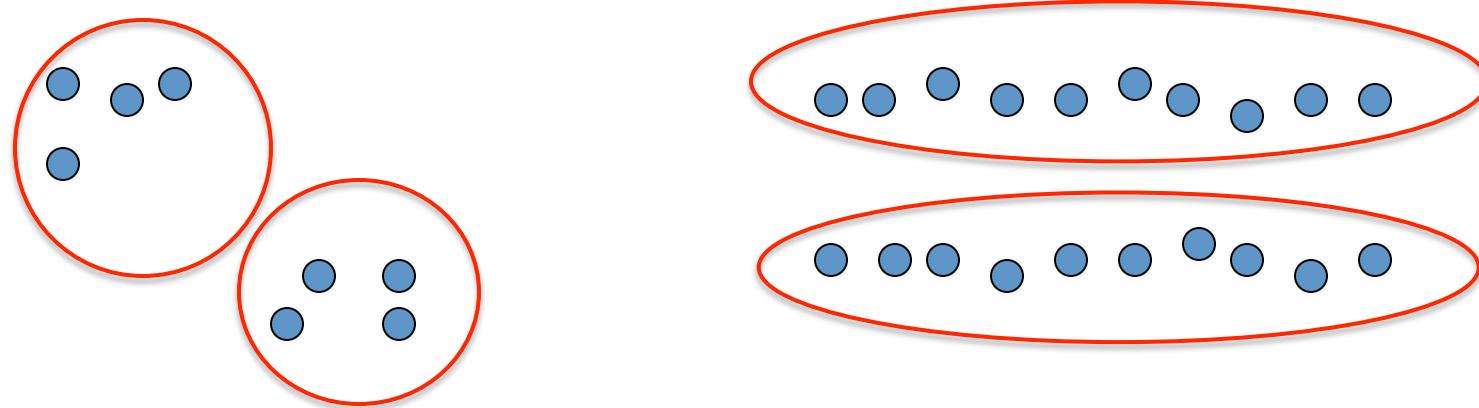
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



Clustering

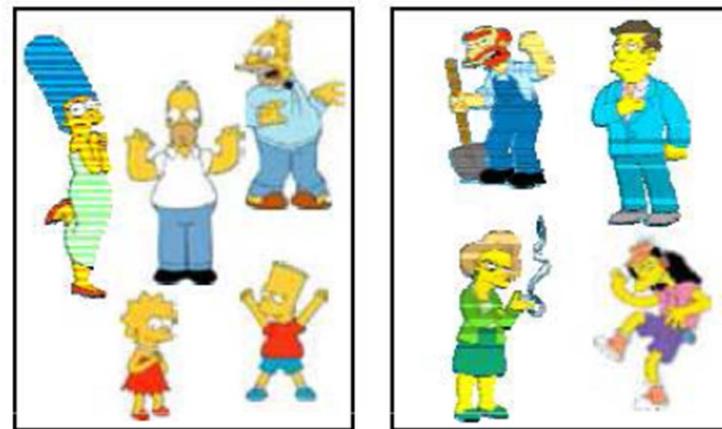
- Basic idea: group together similar instances
- Example: 2D point patterns



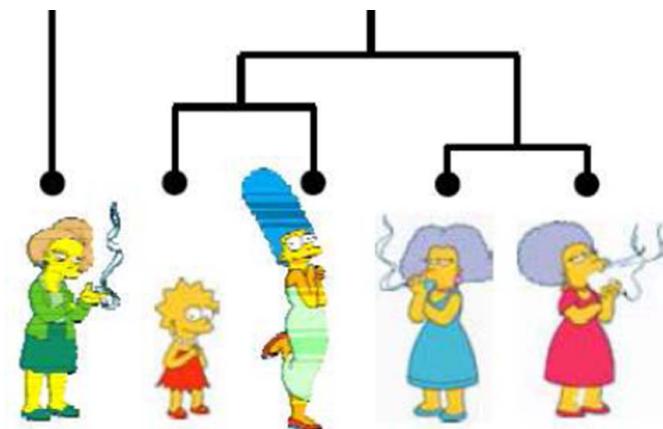
- What could “similar” mean?
 - One option: small Euclidean distance (squared)
$$\text{dist}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_2^2$$
 - Clustering results are crucially dependent on the measure of similarity (or distance) between “points” to be clustered

Clustering algorithms

- Partition algorithms (Flat)
 - K-means
 - Mixture of Gaussian
 - Spectral Clustering



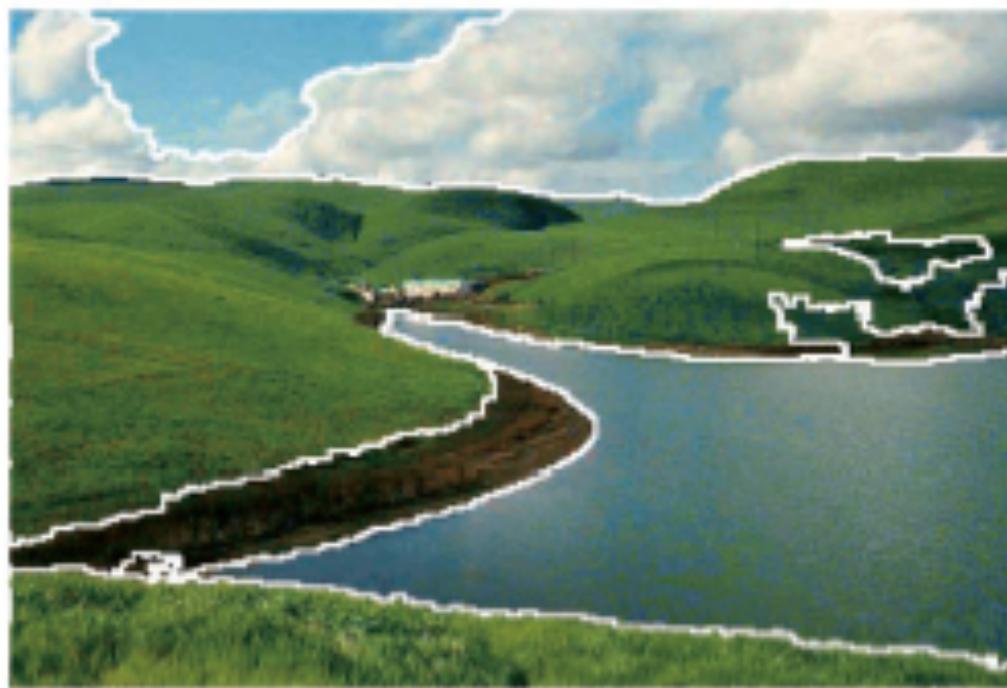
- Hierarchical algorithms
 - Bottom up – agglomerative
 - Top down – divisive



Clustering examples

Image segmentation

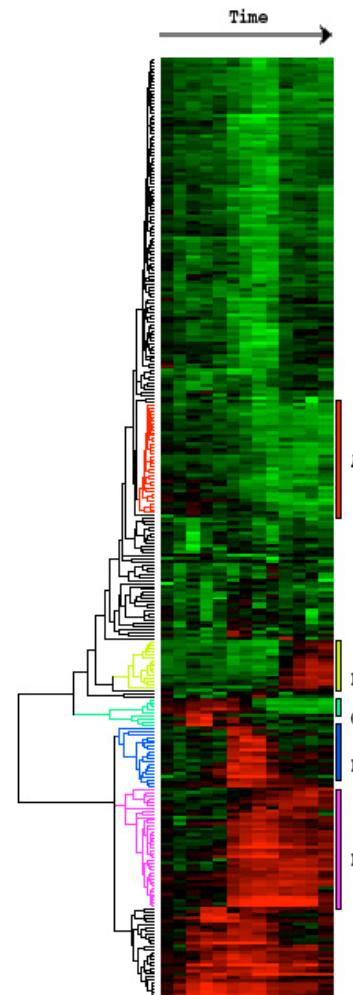
Goal: Break up the image into meaningful or perceptually similar regions



[Slide from James Hayes]

Clustering examples

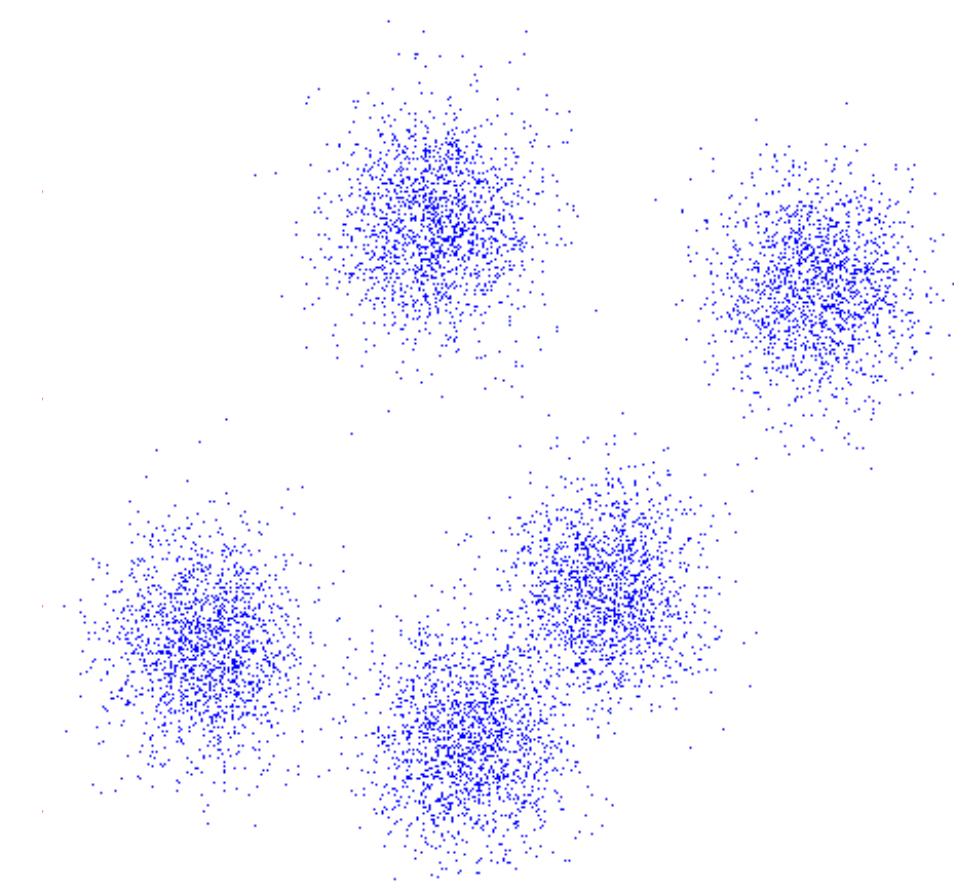
Clustering gene expression data



Eisen et al, PNAS 1998

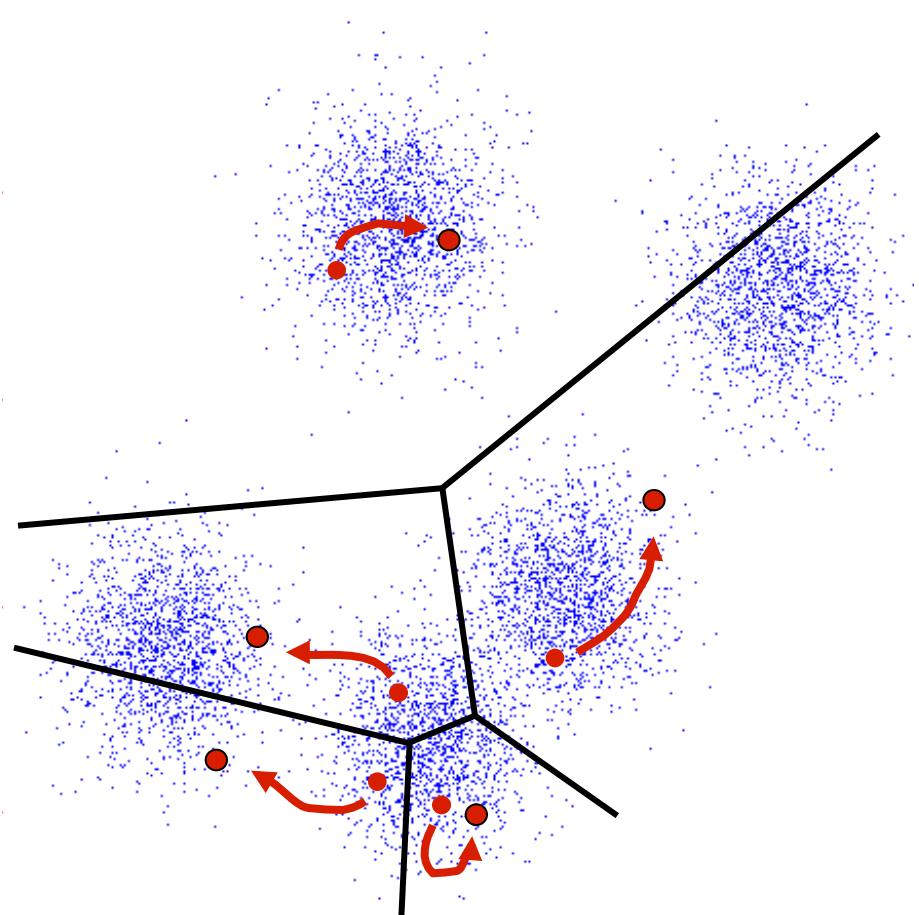
K-Means

- An iterative clustering algorithm
 - Initialize: Pick K random points as cluster centers
 - Alternate:
 1. Assign data points to closest cluster center
 2. Change the cluster center to the average of its assigned points
 - Stop when no points' assignments change

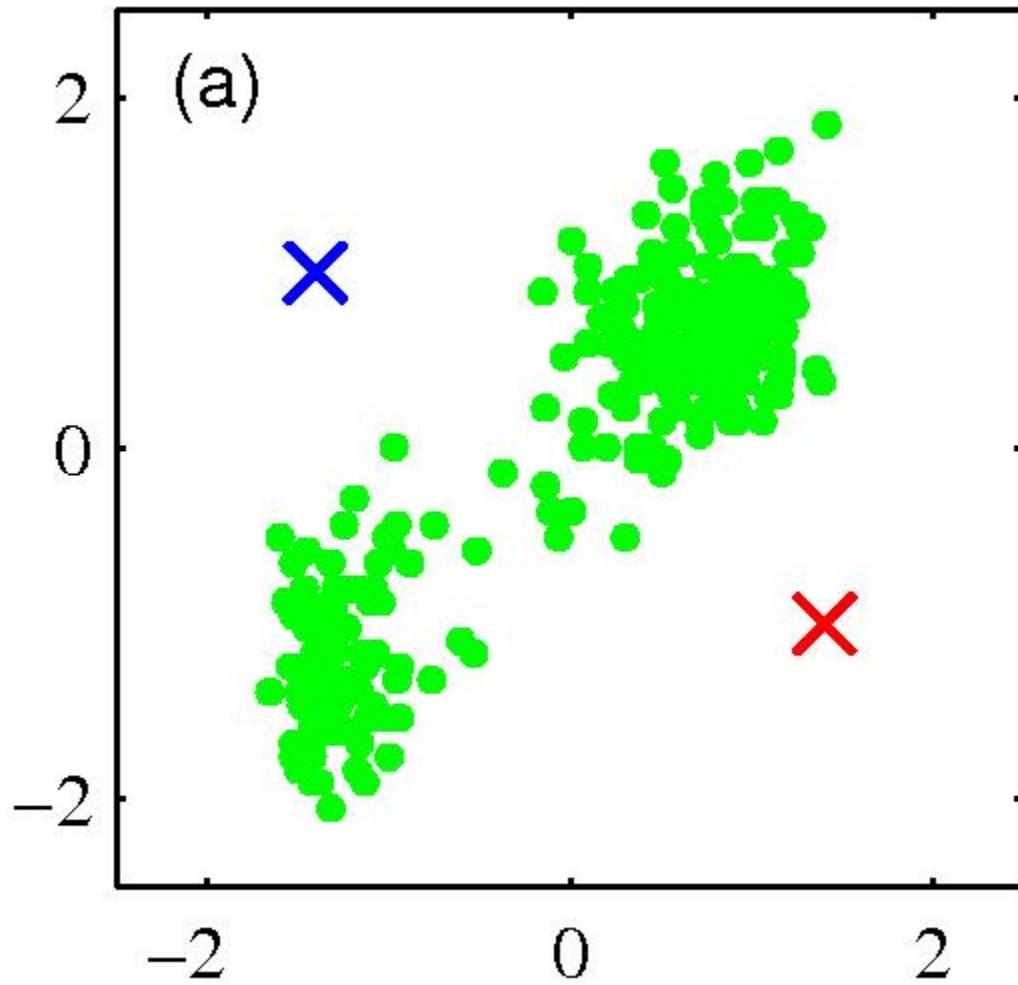


K-Means

- An iterative clustering algorithm
 - Initialize: Pick K random points as cluster centers
 - Alternate:
 1. Assign data points to closest cluster center
 2. Change the cluster center to the average of its assigned points
 - Stop when no points' assignments change



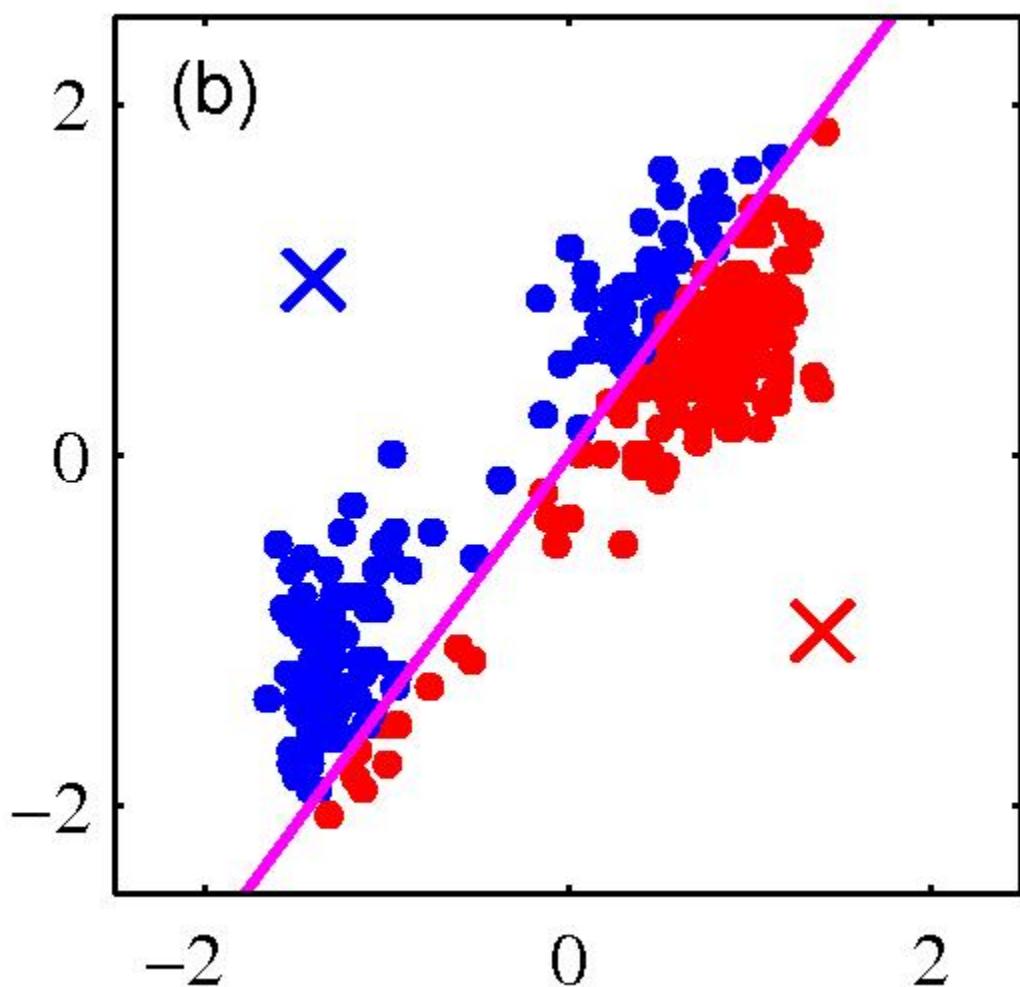
K-means clustering: Example



- Pick K random points as cluster centers (means)

Shown here for $K=2$

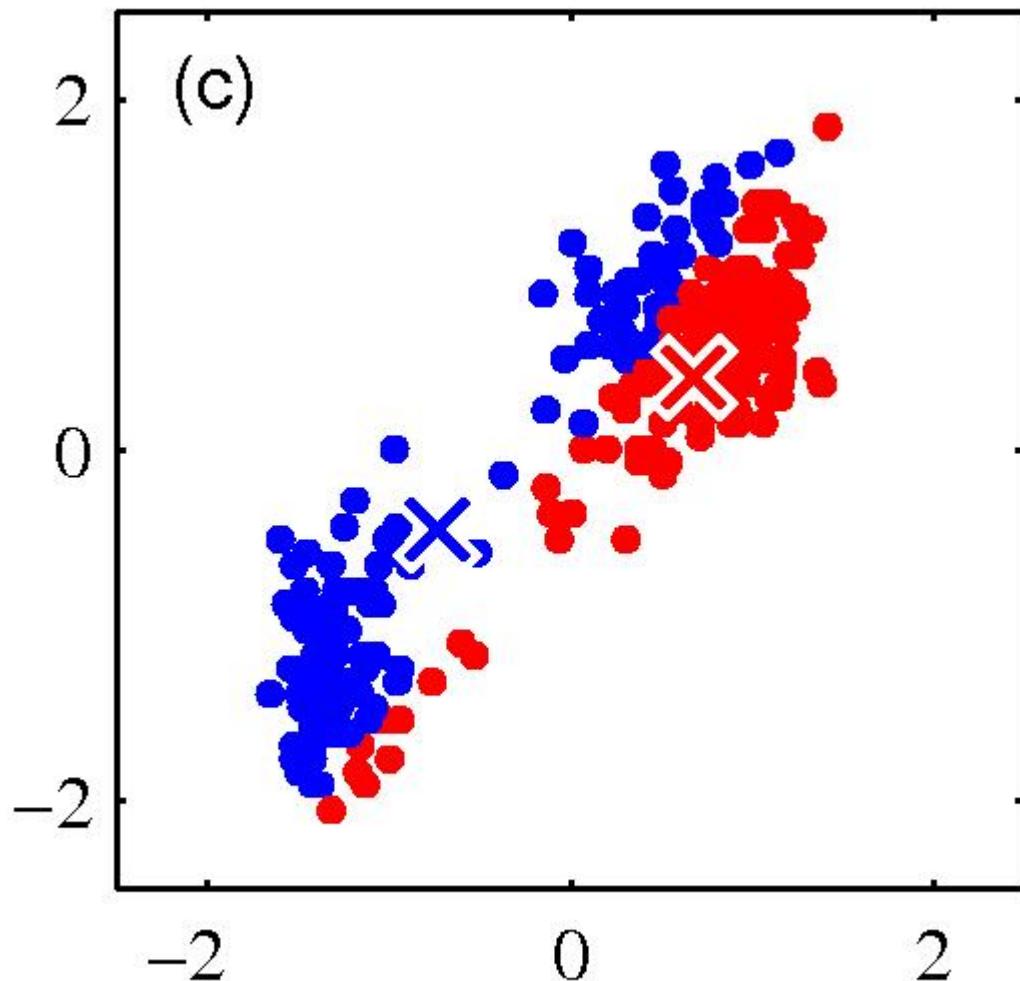
K-means clustering: Example



Iterative Step 1

- Assign data points to closest cluster center

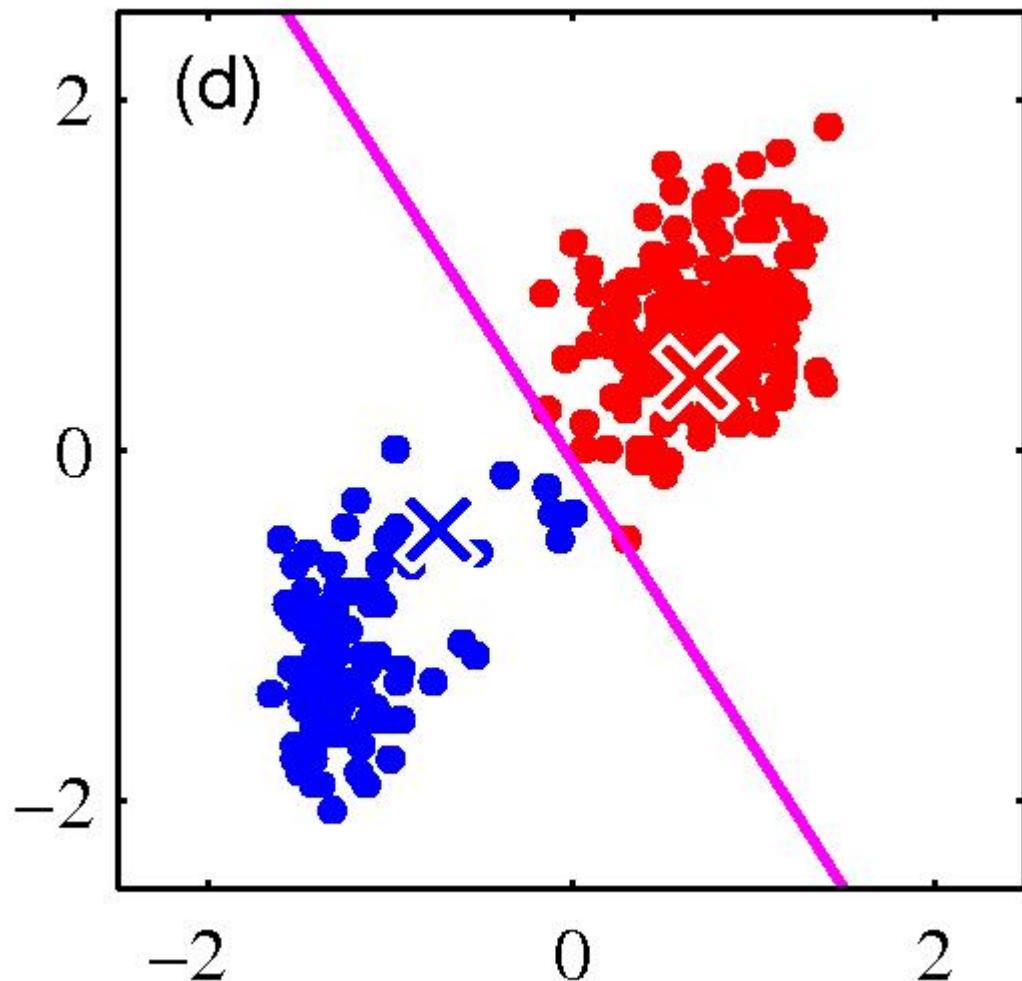
K-means clustering: Example



Iterative Step 2

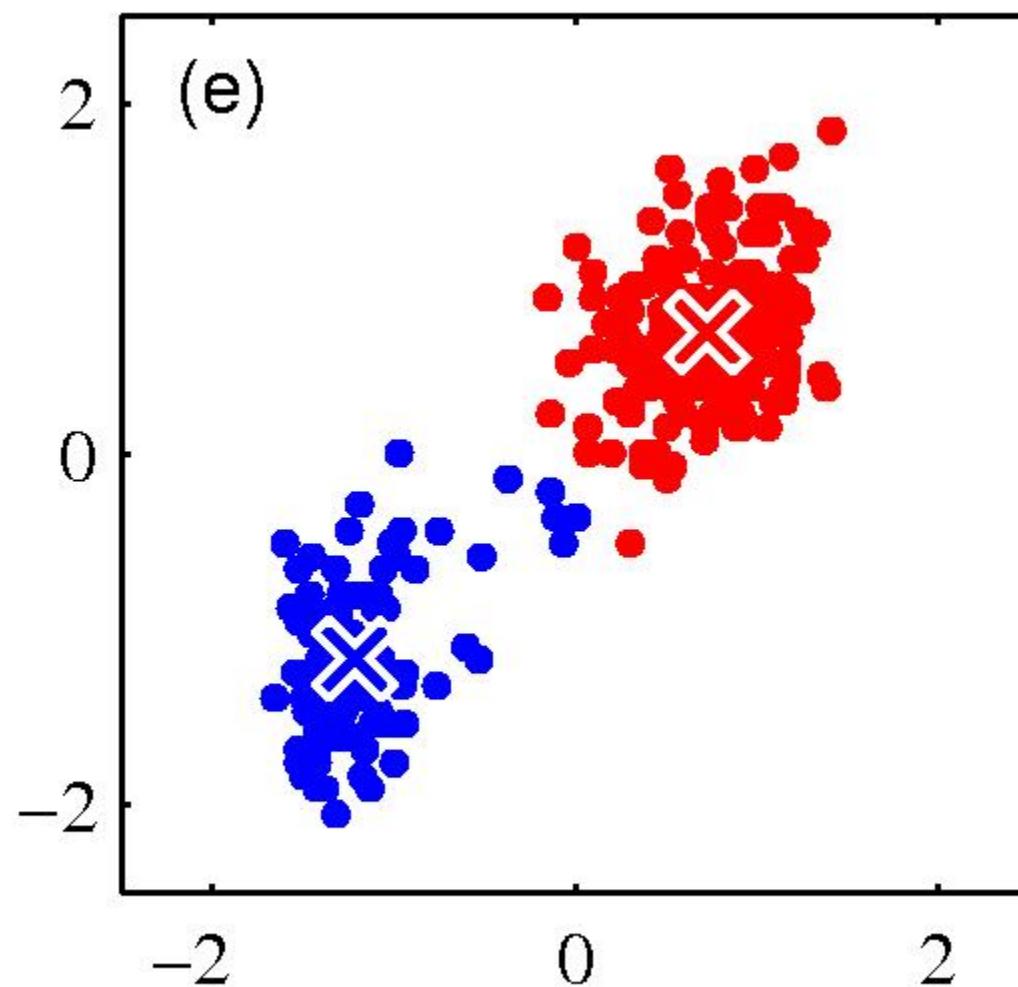
- Change the cluster center to the average of the assigned points

K-means clustering: Example

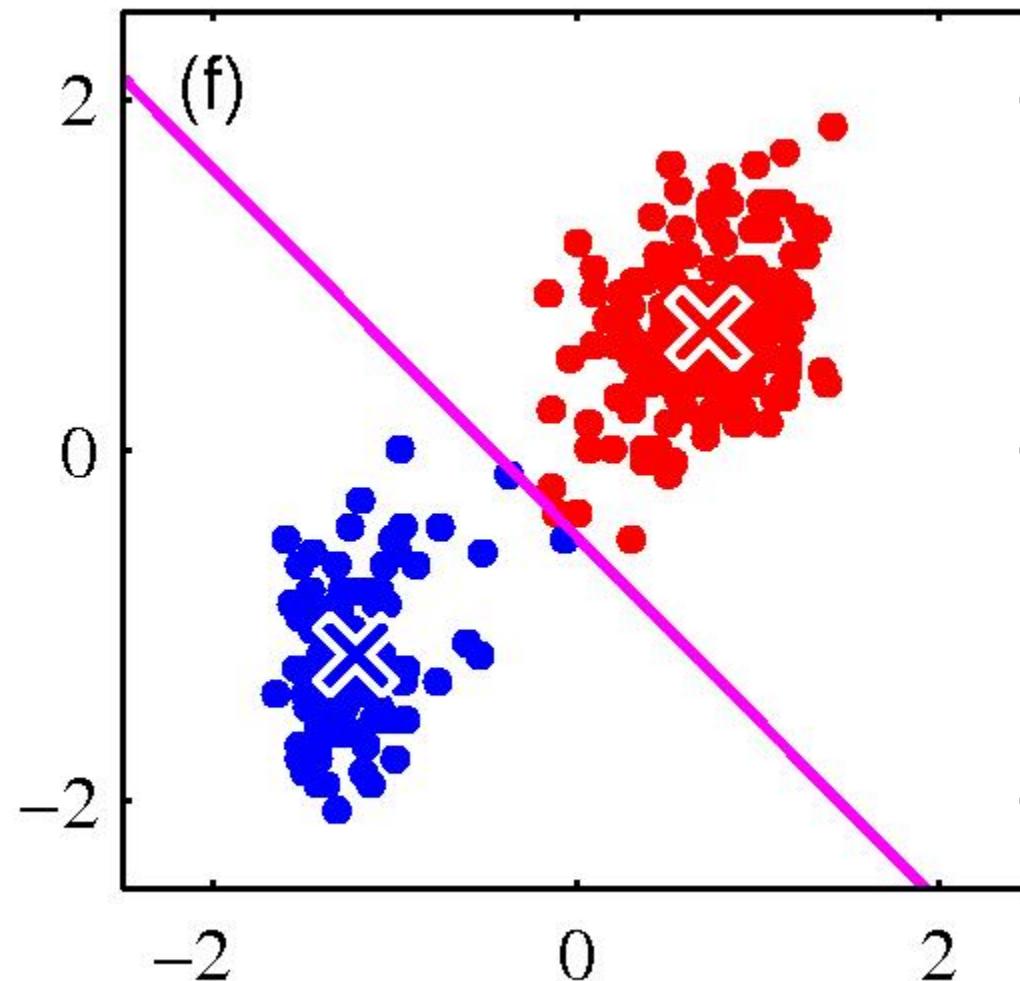


- Repeat until convergence

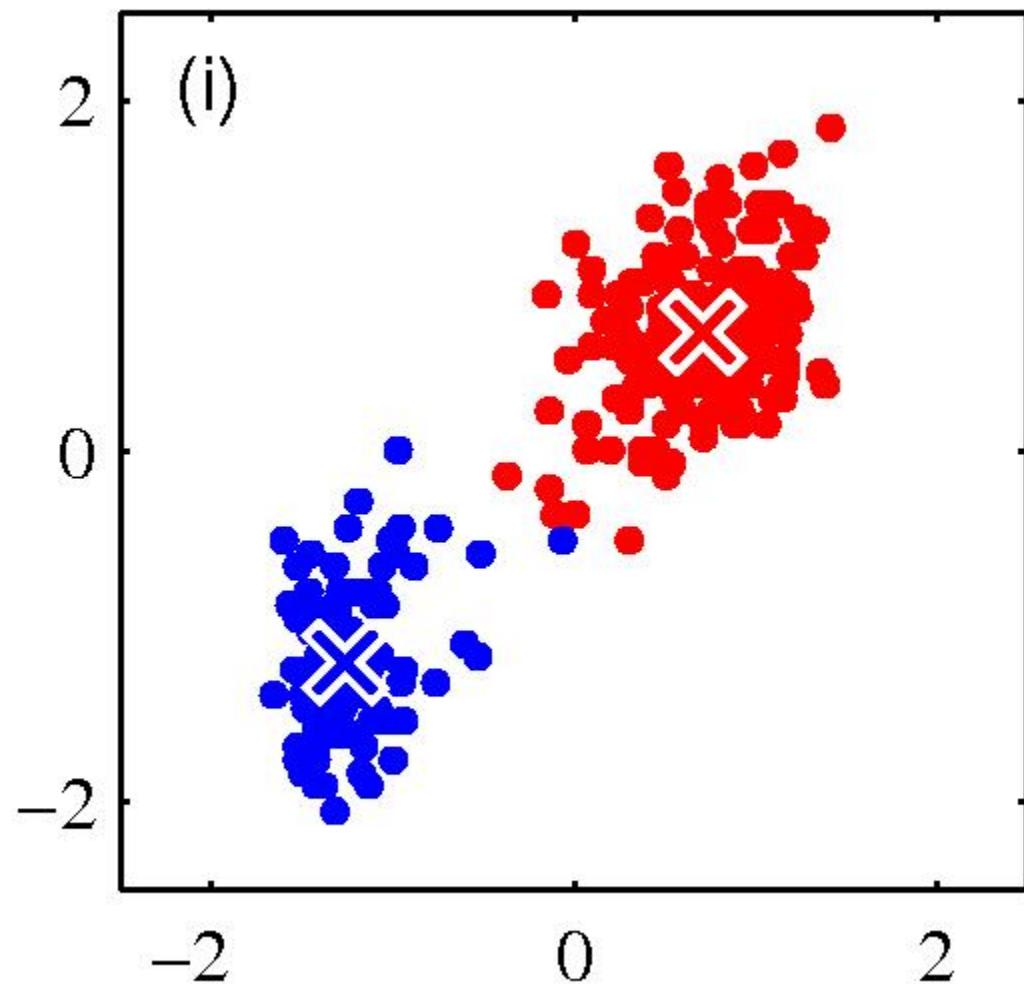
K-means clustering: Example



K-means clustering: Example



K-means clustering: Example



Properties of K-means algorithm

- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
 1. Assign data points to closest cluster center
 $O(KN)$ time
 2. Change the cluster center to the average of its assigned points
 $O(N)$

What properties should a distance measure have?

- Symmetric
 - $D(A,B)=D(B,A)$
 - Otherwise, we can say A looks like B but B does not look like A
- Positivity, and self-similarity
 - $D(A,B) \geq 0$, and $D(A,B)=0$ iff $A=B$
 - Otherwise there will be different objects that we cannot tell apart
- Triangle inequality
 - $D(A,B)+D(B,C) \geq D(A,C)$
 - Otherwise one can say “A is like B, B is like C, but A is not like C at all”

Kmeans Convergence

Objective

$$\min_{\mu} \min_C \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2$$

1. Fix μ , optimize C :

$$\min_C \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2 = \min_c \sum_i^n |x_i - \mu_{x_i}|^2$$

Step 1 of kmeans

2. Fix C , optimize μ :

$$\min_{\mu} \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2$$

- Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

Example: K-Means for Segmentation

K=2



Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

Original



Example: K-Means for Segmentation

K=2



K=3



Original



Example: K-Means for Segmentation



Example: Vector quantization

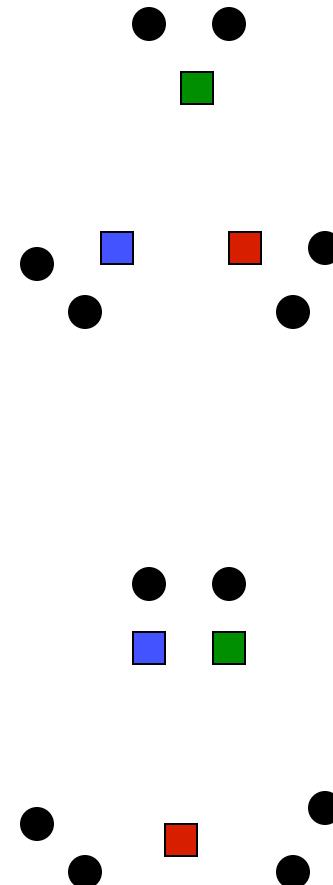


FIGURE 14.9. Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2×2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel

[Figure from Hastie et al. book]

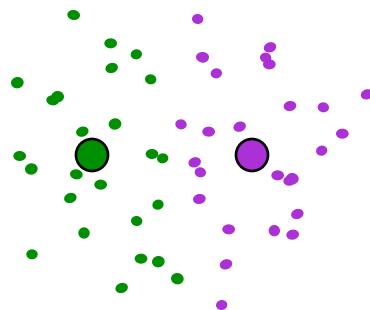
Initialization

- K-means **algorithm** is a heuristic
 - Requires initial means
 - It does matter what you pick!
 - What can go wrong?
 - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics

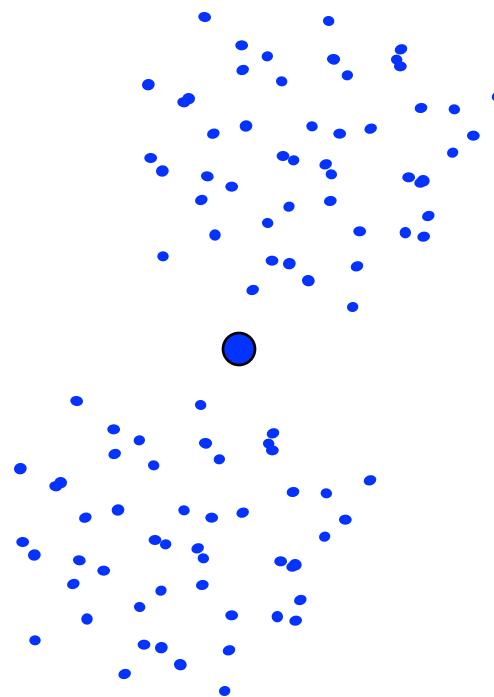


K-Means Getting Stuck

A local optimum:

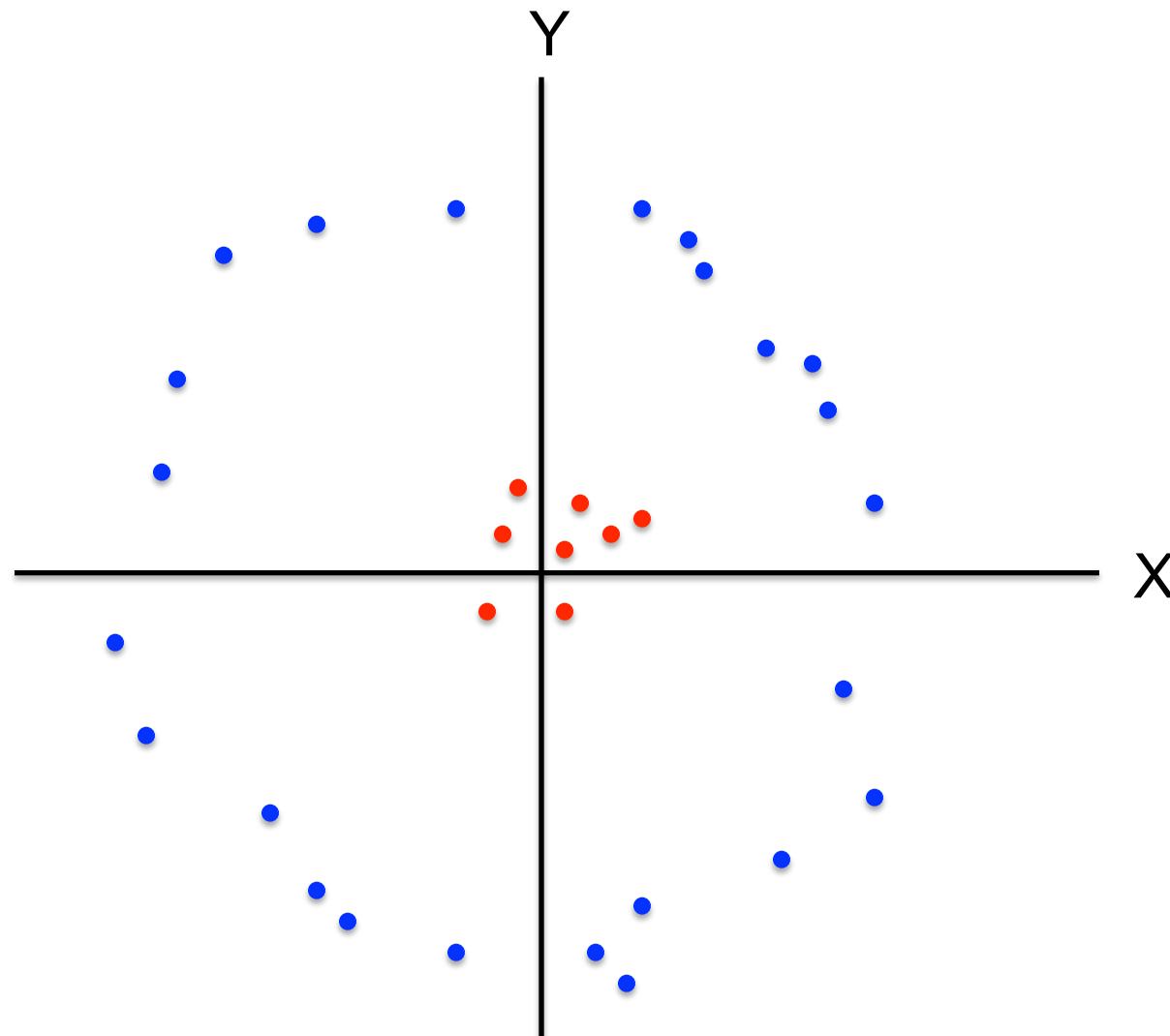


Would be better to have
one cluster here

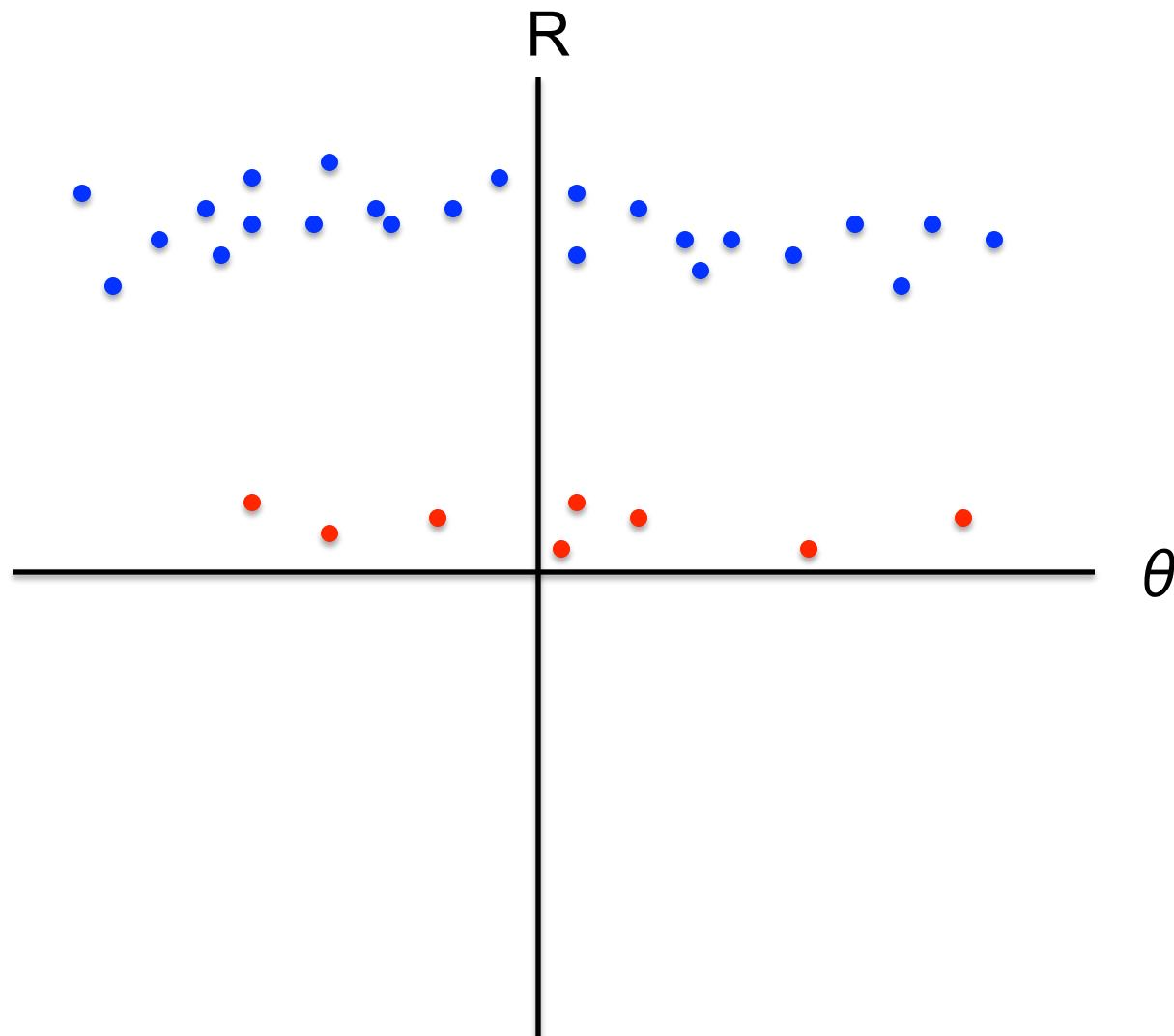


... and two clusters here

K-means not able to properly cluster



Changing the features (distance function)
can help



Fundamentals of AI

Clustering

Density-based and graph-based clustering*

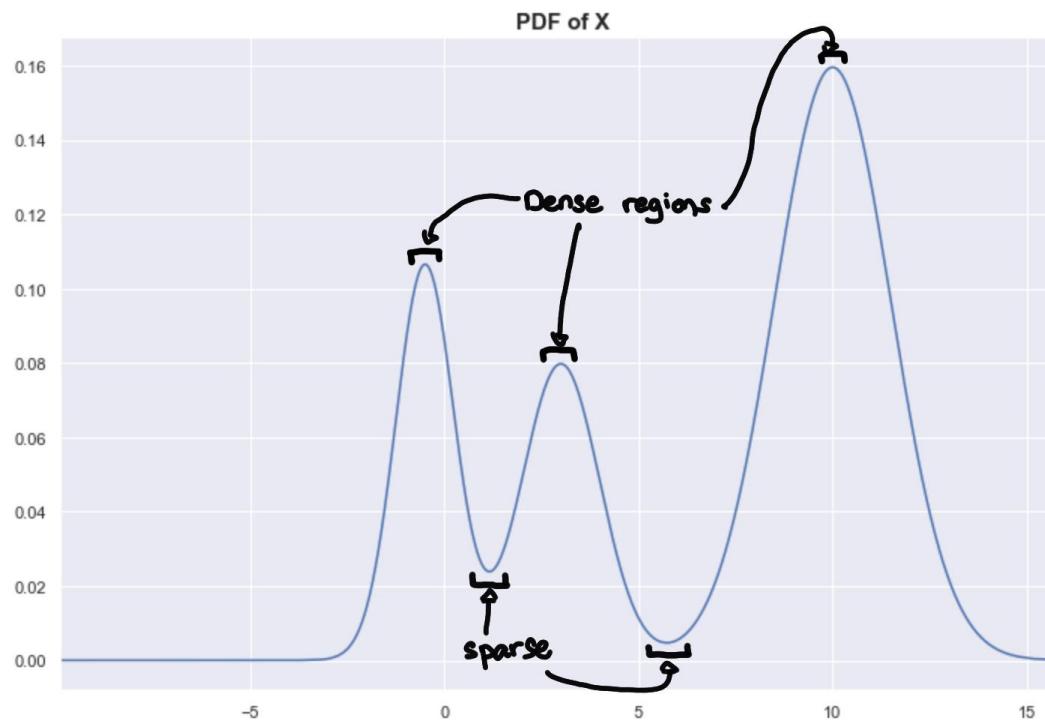
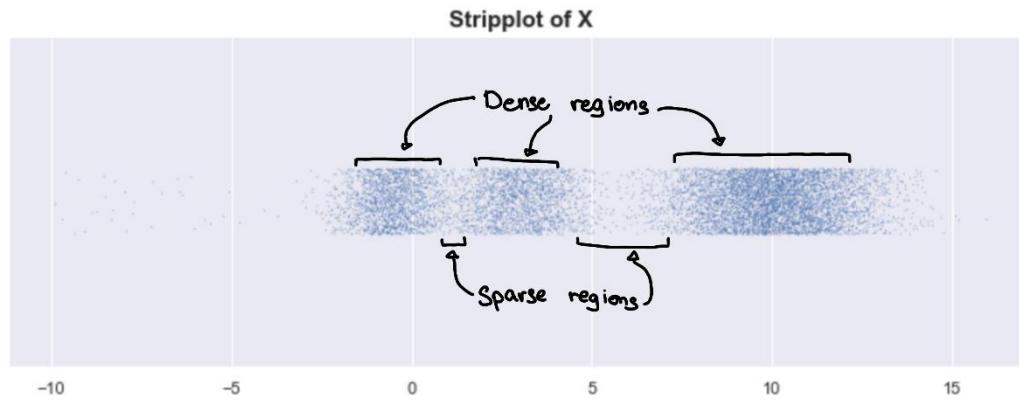
Some images in this lecture are used from: <https://www.kdnuggets.com/2020/02/understanding-density-based-clustering.html>

Distance-based clustering and its limitations

- Hard to find clusters with irregular shapes
- Hard to specify the number of clusters
- Some points are ‘in between’ clusters (outliers or background noise)

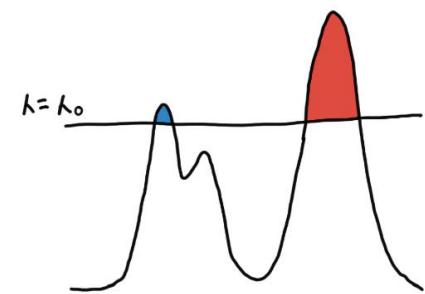
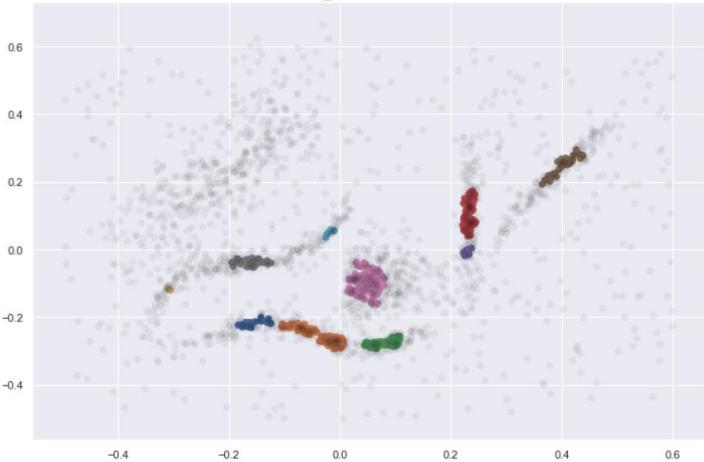


New concept: cluster as a probability density peak

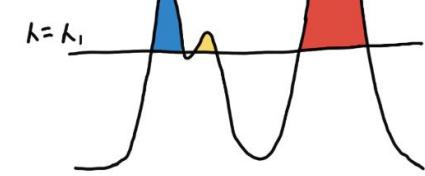
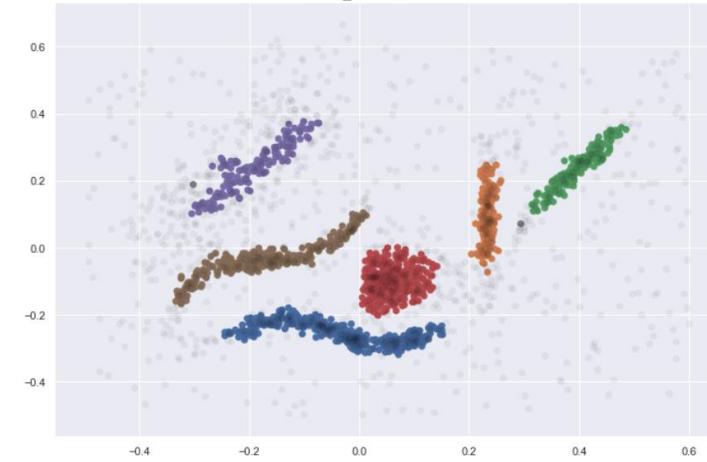


Cool, but how to define PDF in multi-dimensional space?
Expensive and better to avoid at all

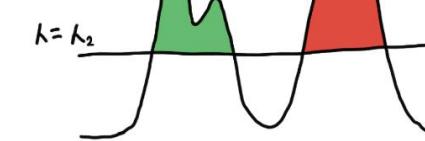
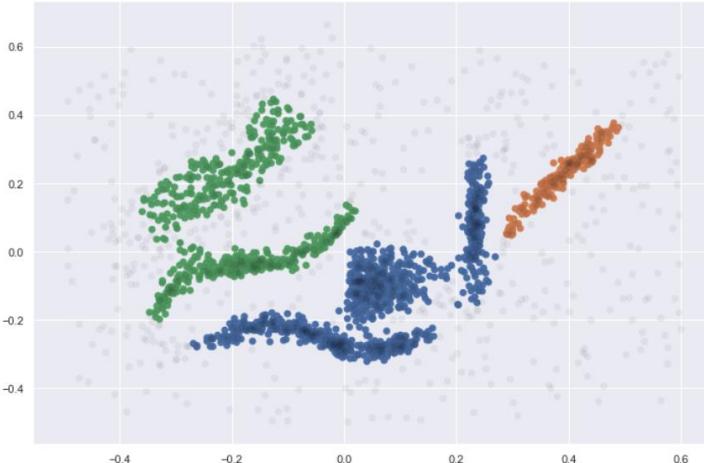
core_distance ≤ 0.015



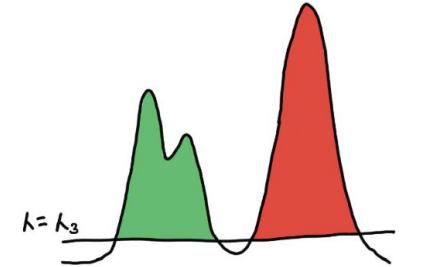
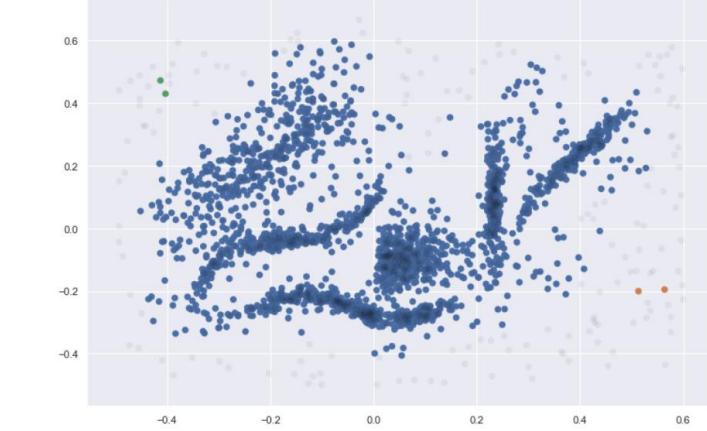
core_distance ≤ 0.030



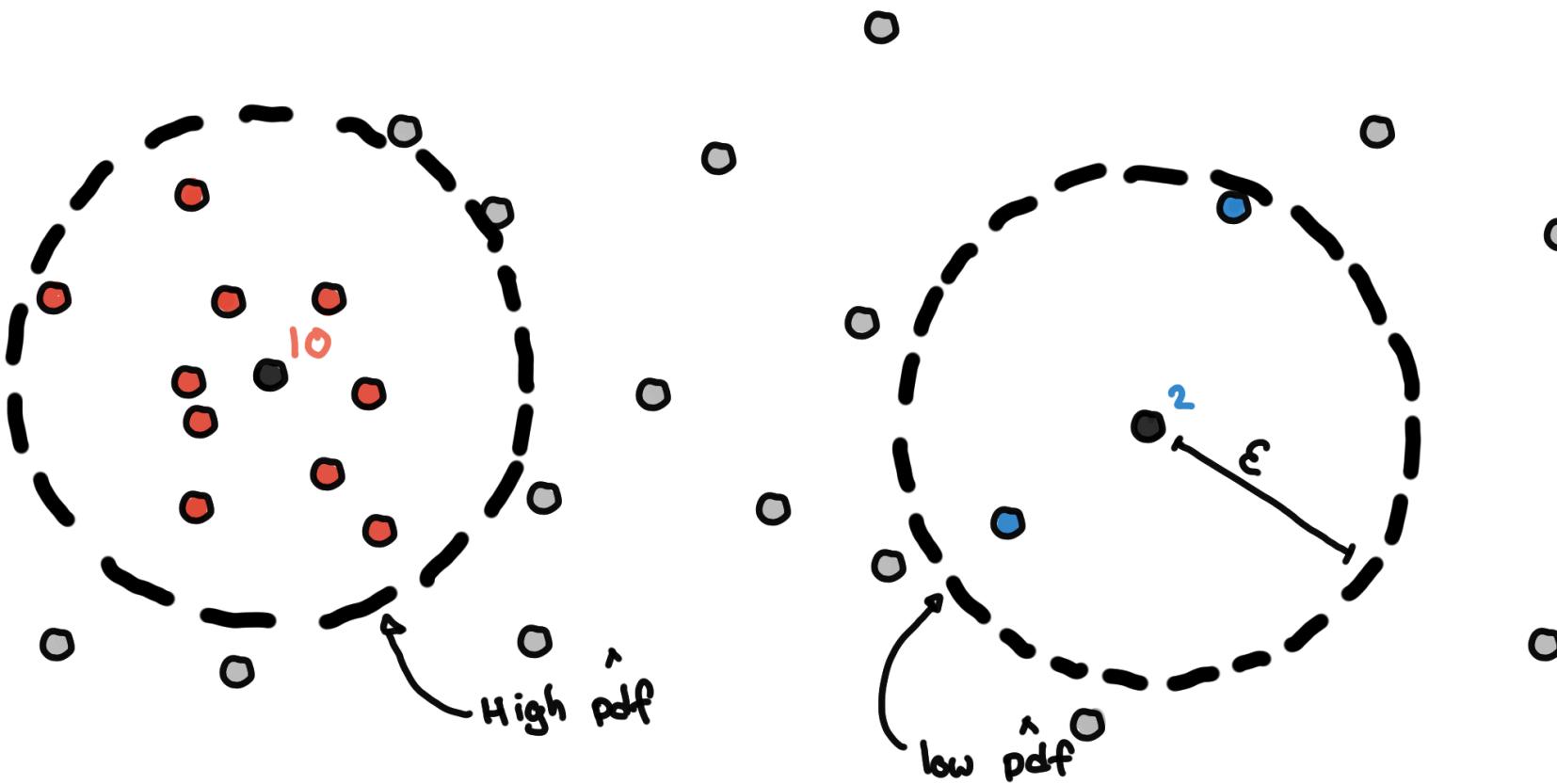
core_distance ≤ 0.040



core_distance ≤ 0.100



Trick: count neighbours within ε -radius

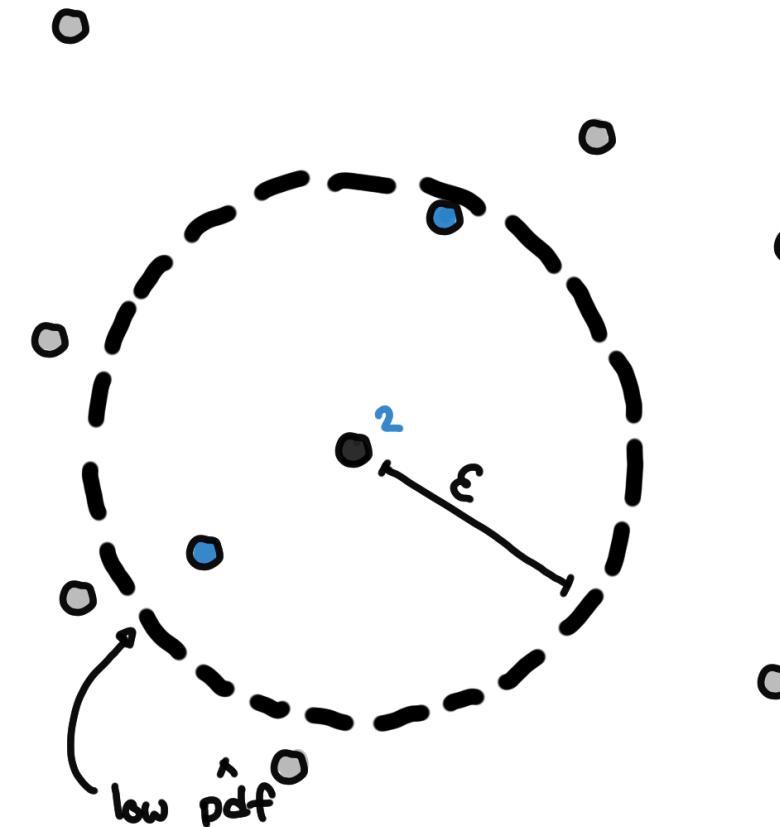
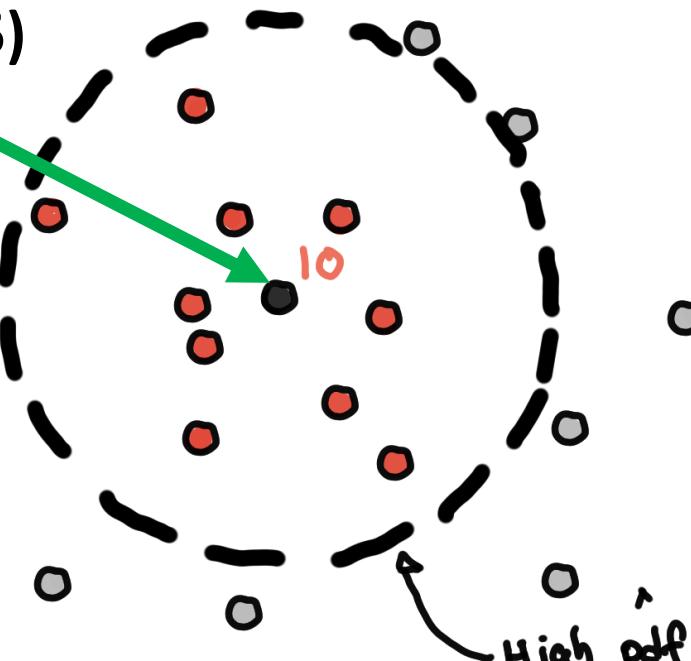


DBSCAN (Ester et al, 1996)

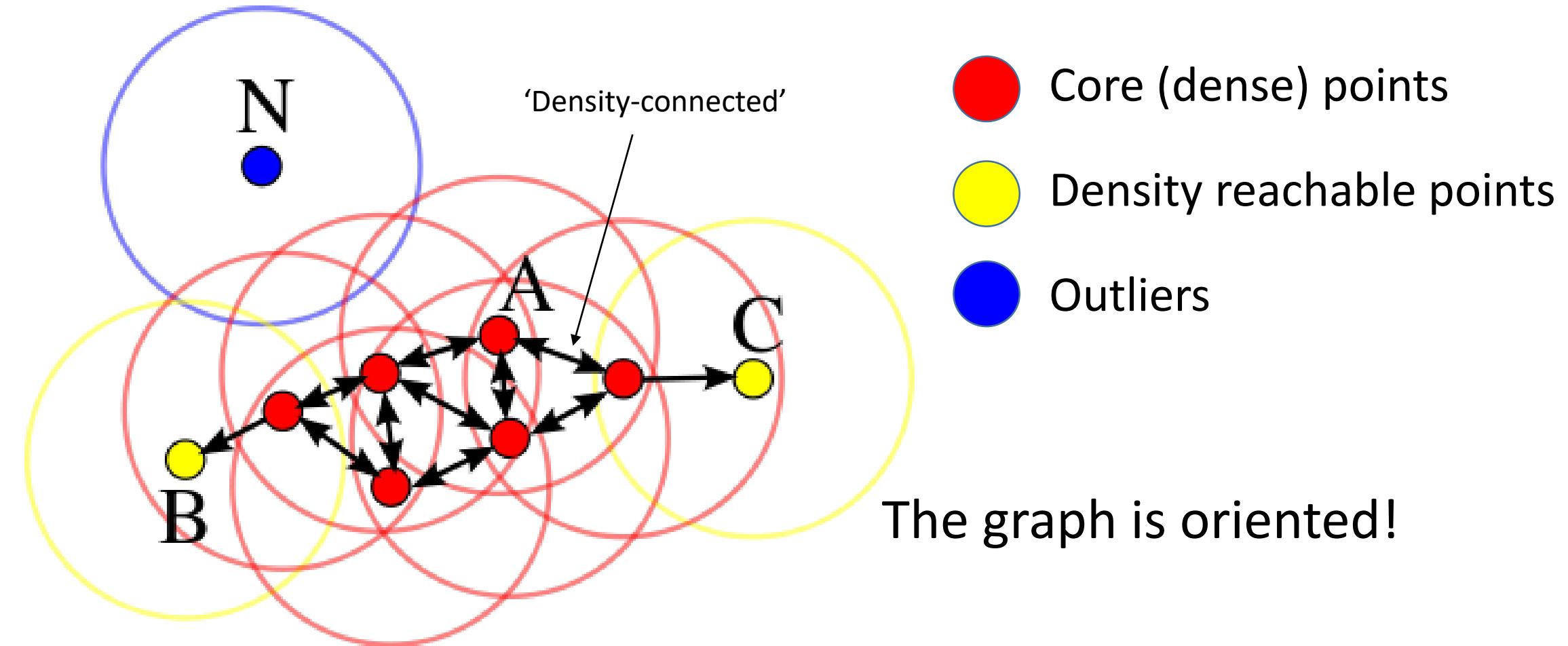
Parameters: ϵ and minPts

Core point

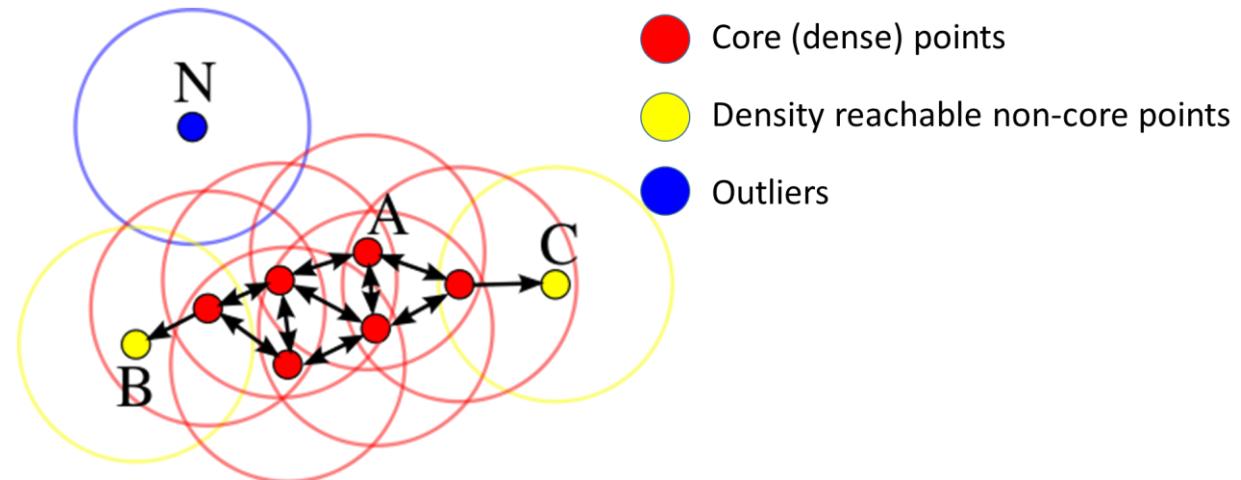
(e.g., minPts=5)



DBSCAN: graph of core points and density-reachable (peripheral) points



DBSCAN: graph of core points and density-reachable (peripheral) points



A cluster then satisfies two properties:

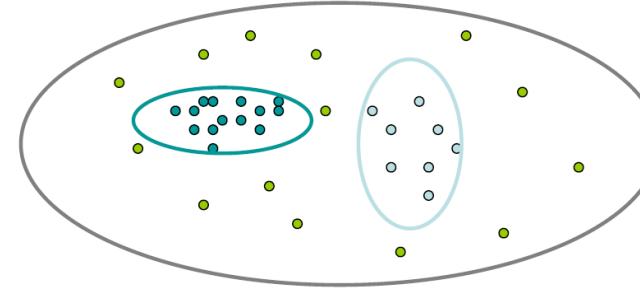
1. All points within the cluster are mutually density-connected.
2. If a point is density-reachable from some point of the cluster, it is part of the cluster as well.

DBSCAN: the Algorithm

- Arbitrary select a point p
- Retrieve all points density-reachable from p wrt Eps and MinPts
- If p is a core point, a cluster is formed
- If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed

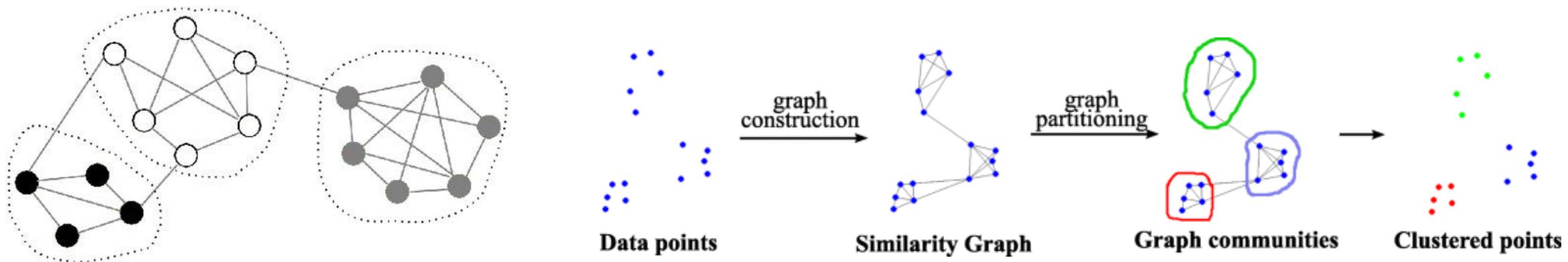
Comments on DBSCAN

- Complexity is $O(n \log n)$
- Unlike k-means and hierarchical, deal with the notion of noise
- Different clusters may have very different densities
- Very sensitive to the choice of ε
- Concentration of measures will spoil everything in high intrinsic dimensionalities
- Extensions: OPTICS, HDBSCAN, GDBSCAN
- Scikit learn implementation (arbitrary L_p metrics, accelerated neighbor search)



Graph-based clustering algorithms

- Cluster = tight community of the KNN graph



- The quality of communities is determined by **modularity**

Various definitions of the modularities

- E.g., **Louvain modularity**

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \quad \text{In range } [-0.5; 1]$$

where

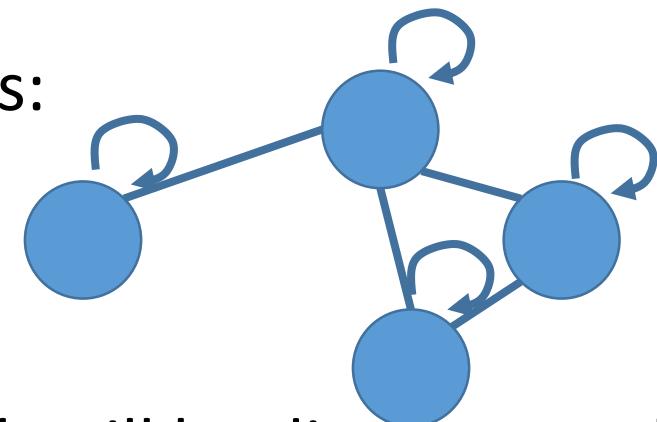
- A_{ij} represents the edge weight between nodes i and j ;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j , respectively;
- m is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is Kronecker delta function ($\delta(x, y) = 1$ if $x = y$, 0 otherwise).

Louvain clustering algorithm (a greedy one)

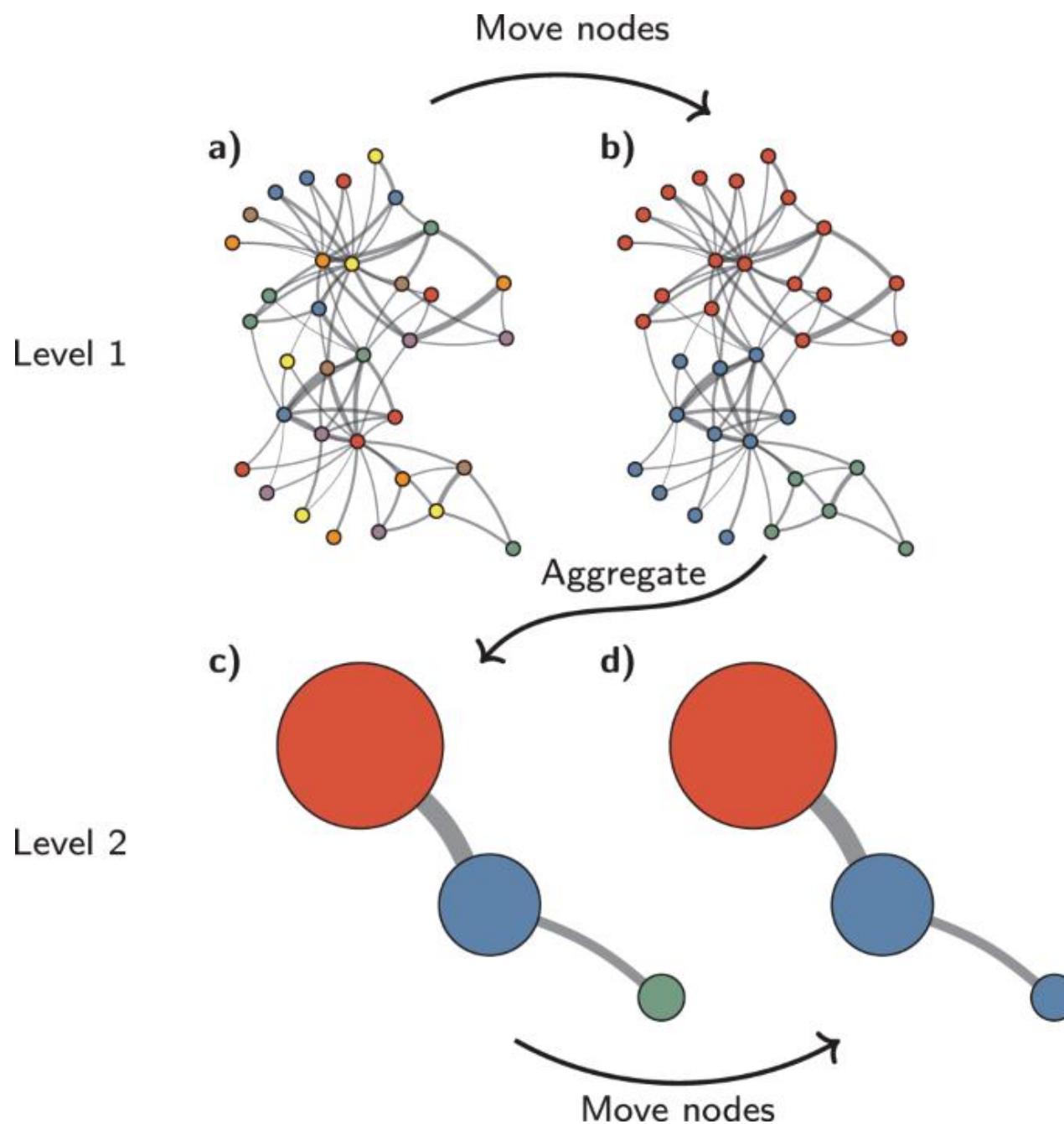
Base community search:

- At first, one node = one community
- We swap node i from its own community to the community of each of its neighbours
- For each such a swap, change in modularity is computed
- If no increase of modularity possible i remains in its own community

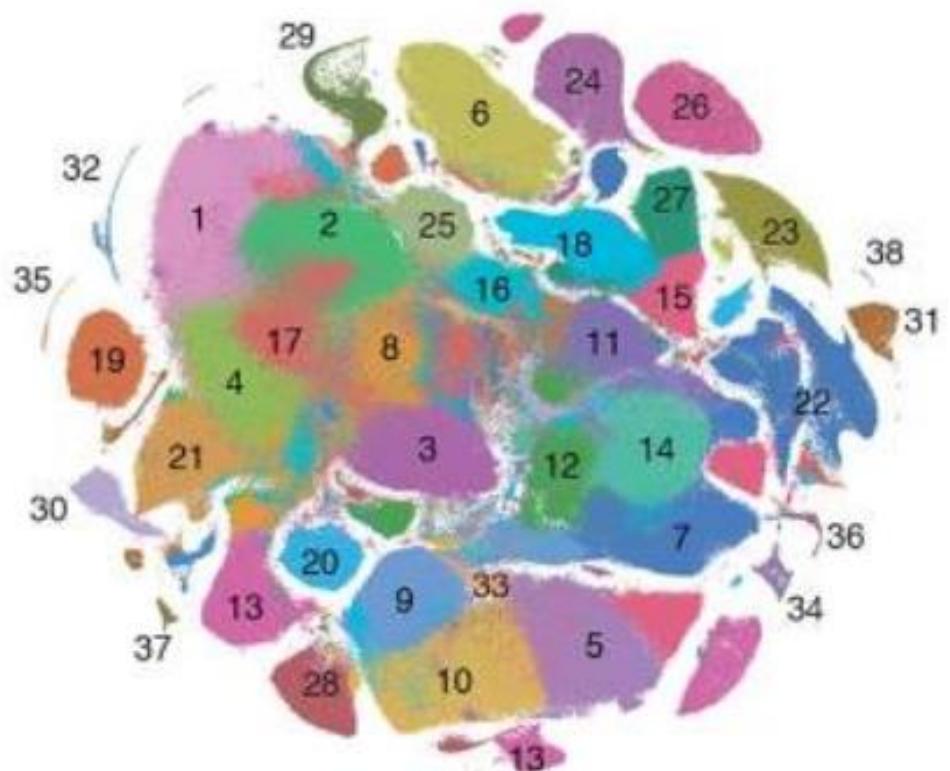
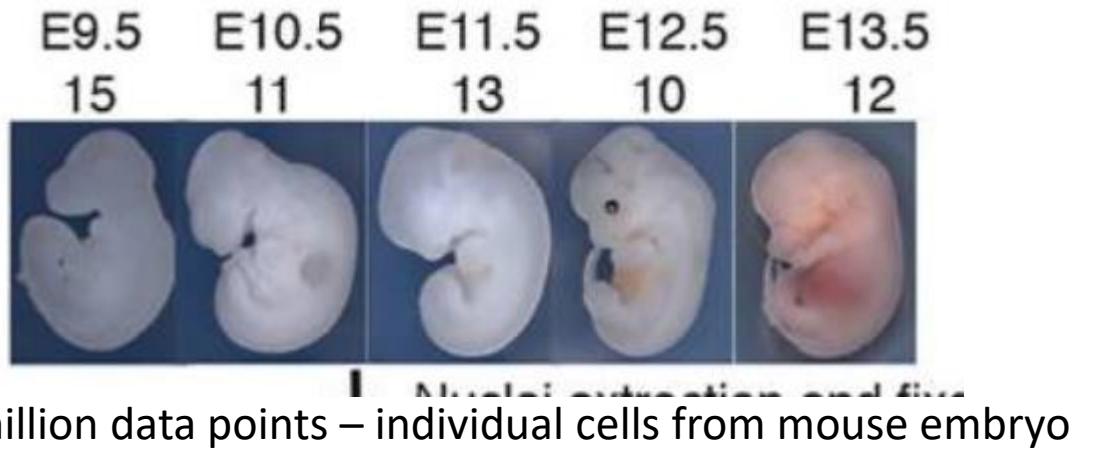
Finding communities in the graph of communities:



Cluster until one community remains or the graph will be disconnected



Graph-based
clustering became
new killer application
in life sciences,
replacing the
hierarchical clustering



(from Cao et al, Nature, 2019)