CHAPTER 7:

CLUSTERING

Stella Grasshof

Overview

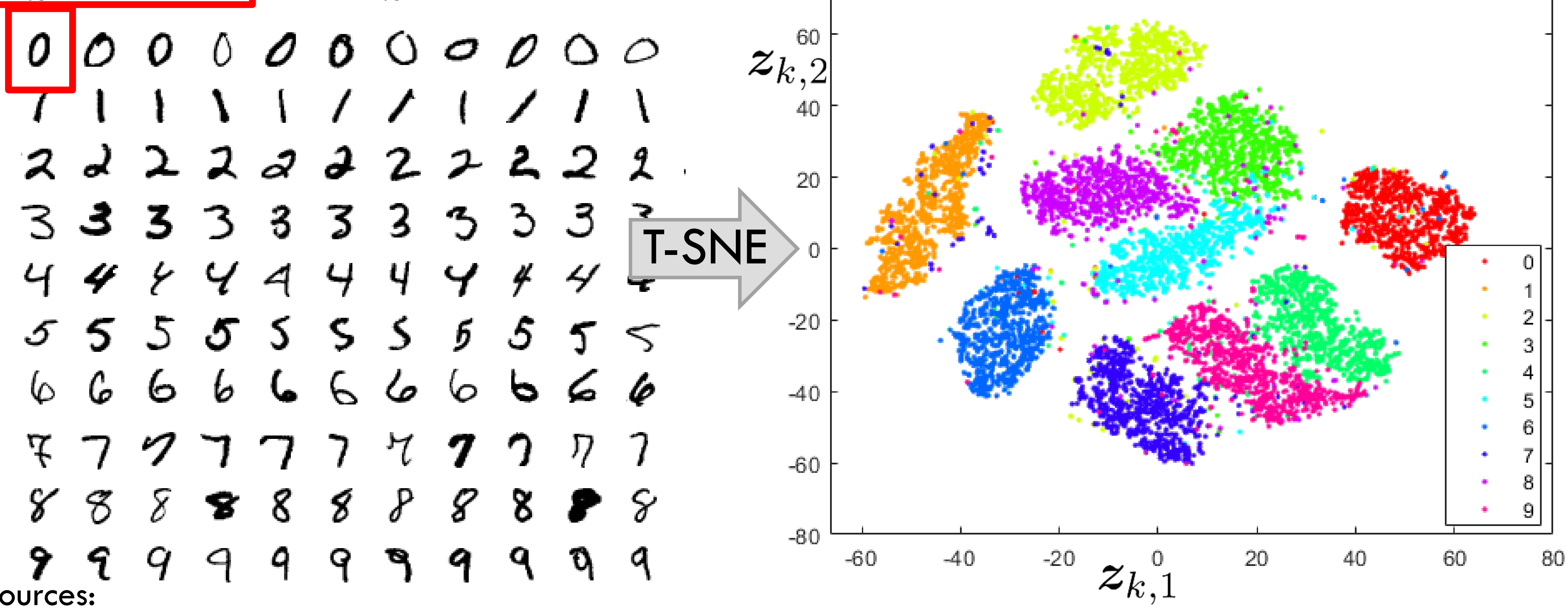
1) Intro
2) Hierarchical Clustering
3) K-means
4) Spectral Clustering
5) Expectation-Maximization Algorithm (EM)
6) Considerations
7) Mean-Shift
8) Outro

Feature Extraction

Consider high-dimensional data is given, we want:

- Compact representation of the data
- Extract **most relevant** information

$\boxed{\boldsymbol{x}_k \in \mathbb{R}^{64\cdot 64}} \mapsto \boldsymbol{z}_k \in \mathbb{R}^2$



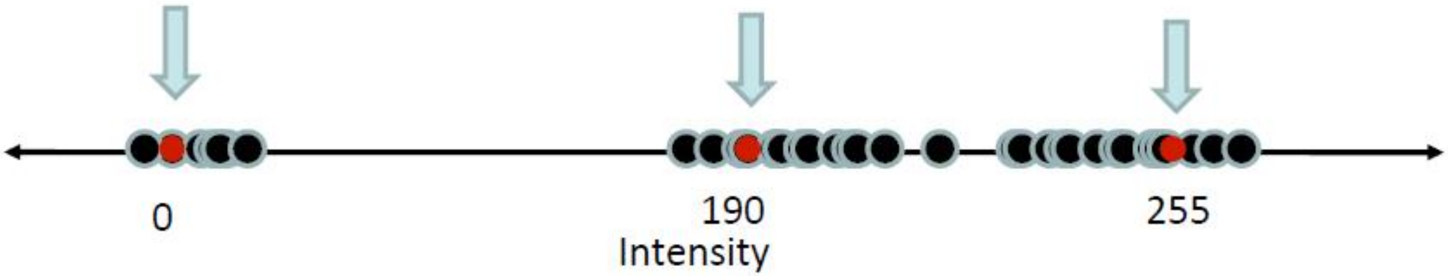
Sources:
[1] MNIST, wikpedia.org, Josef Steppan [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)]
[2] MathWorks, https://se.mathworks.com/help/stats/visualize-high-dimensional-data-using-t-sne.html

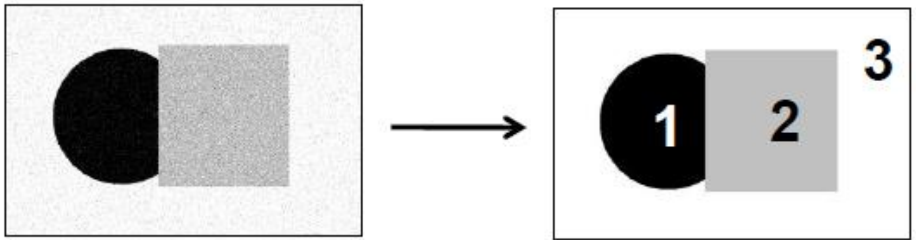
Why Clustering?

- Classification: $\boldsymbol{x}_n \mapsto C_k$
- Dimensionality Reduction:

$$\boldsymbol{x}_n \in \boldsymbol{X} \in \mathbb{R}^{N \times D} \rightsquigarrow \boldsymbol{z}_n \in \boldsymbol{Z} \in \mathbb{R}^{N \times E}, \;\; E < D$$

- Clustering:
  - "estimate unknown class labels"
  - Analyze structure in data





http://vision.stanford.edu/teaching/cs131_fall1314_nope/lectures/lecture13_kmeans_cs131.pdf

Semiparametric Density Estimation

Data probability density function (pdf): $p(\boldsymbol{x})$

- **Parametric:** one model, (Ch. 4 and 5) e.g.

$$p(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

- **Semiparametric:** a **mixture** of densities

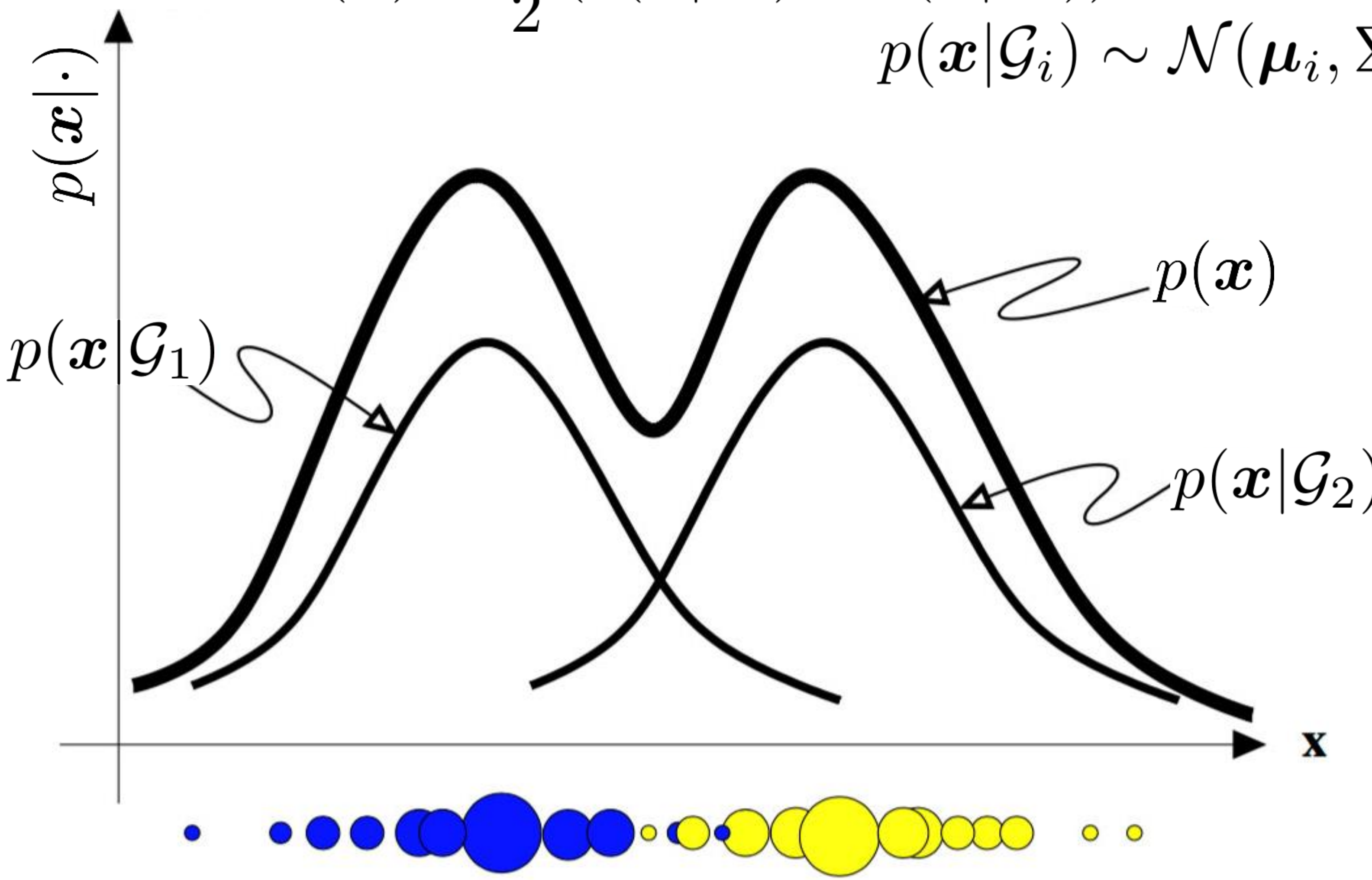
$$p(\boldsymbol{x}) = \sum_i P(\mathcal{G}_i)\ p(\boldsymbol{x}|\mathcal{G}_i)$$

- **Nonparametric: No model** (Ch. 8)

Mixture of Gaussians

$$p(\boldsymbol{x}) = \frac{1}{2}\left(p(\boldsymbol{x}|\mathcal{G}_1) + p(\boldsymbol{x}|\mathcal{G}_2)\right)$$

$$p(\boldsymbol{x}|\mathcal{G}_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$$



https://stackoverflow.com/questions/37320025/mixture-of-gaussian-distribution-in-c

Classes vs. Clusters

Supervised: data with class labels $\boldsymbol{x}_n \in C_k$

$$\mathbb{1}\left\{\boldsymbol{x}_n \in C_k\right\} = \begin{cases} 1 & , \ \boldsymbol{x}_n \in C_k \\ 0 & , \ \text{else} \end{cases}, \ N_k := \sum_{n=1}^{N} \mathbb{1}\left\{\boldsymbol{x}_n \in C_k\right\},$$

$$p\left(\boldsymbol{x}\right) = \sum_{k=1}^{K} p\left(\boldsymbol{x}|C_k\right) P(C_k)$$

$$p\left(\boldsymbol{x}|C_k\right) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\boxed{\boldsymbol{\theta} = \left\{P(C_k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right\}_{k=1}^{K}}$$

$$\widehat{P}(C_k) = \frac{N_k}{N}$$

$$\widehat{\boldsymbol{\mu}}_k = \boldsymbol{m}_k = \frac{1}{N_i} \sum_{n=1}^{N} \mathbb{1}\left\{\boldsymbol{x}_n \in C_k\right\} \boldsymbol{x}_n$$

$$\widehat{\boldsymbol{\Sigma}}_k = \boldsymbol{S}_k = \frac{1}{N_i} \sum_{n=1}^{N} \mathbb{1}\left\{\boldsymbol{x}_n \in C_k\right\} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^{\mathrm{T}}$$

Classes vs. Clusters

Unsupervised: data **without** class labels $\boldsymbol{x}_n$
but assume there are groupings in the data $\mathcal{G}_k$ => Labels?

$$\mathbb{1}\left\{\boldsymbol{x}_n \in \mathcal{G}_k\right\} = \begin{cases} 1 & , \ \boldsymbol{x}_n \in \mathcal{G}_k \\ 0 & , \ \text{else} \end{cases} ,$$

$$p\left(\boldsymbol{x}\right) = \sum_{k=1}^{K} p\left(\boldsymbol{x}|\mathcal{G}_k\right) P(\mathcal{G}_k)$$

$$p\left(\boldsymbol{x}|\mathcal{G}_k\right) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

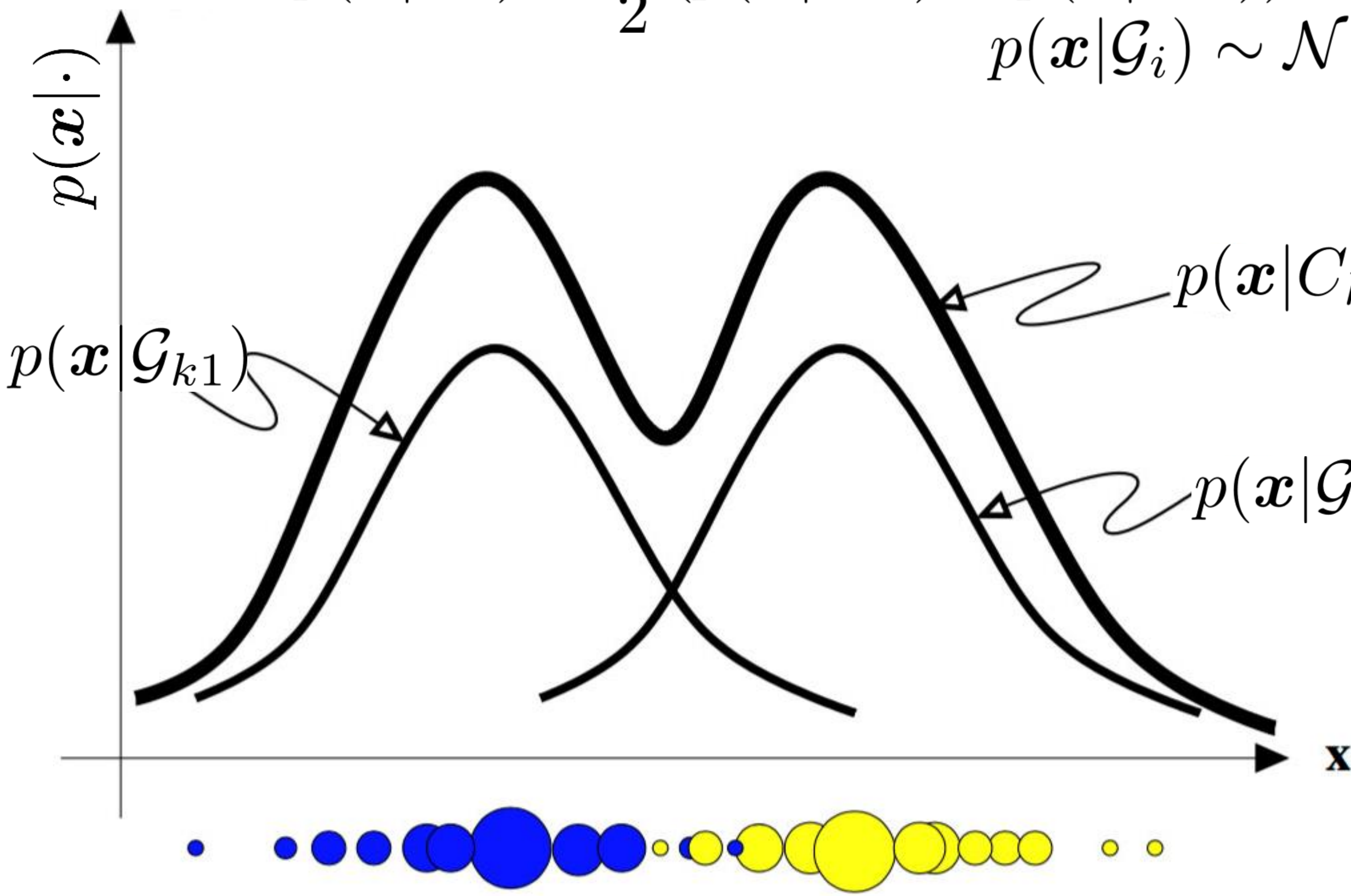
$$\boxed{\boldsymbol{\theta} = \{P(\mathcal{G}_k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}}$$

Classes: prior known category
Cluster: e.g. sub-groups per class

Mixture of Mixtures

$$p(\boldsymbol{x}|C_k) = \frac{1}{2}\left(p(\boldsymbol{x}|\mathcal{G}_{k1}) + p(\boldsymbol{x}|\mathcal{G}_{k2})\right)$$

$$p(\boldsymbol{x}|\mathcal{G}_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$$



https://stackoverflow.com/questions/37320025/mixture-of-gaussian-distribution-in-c

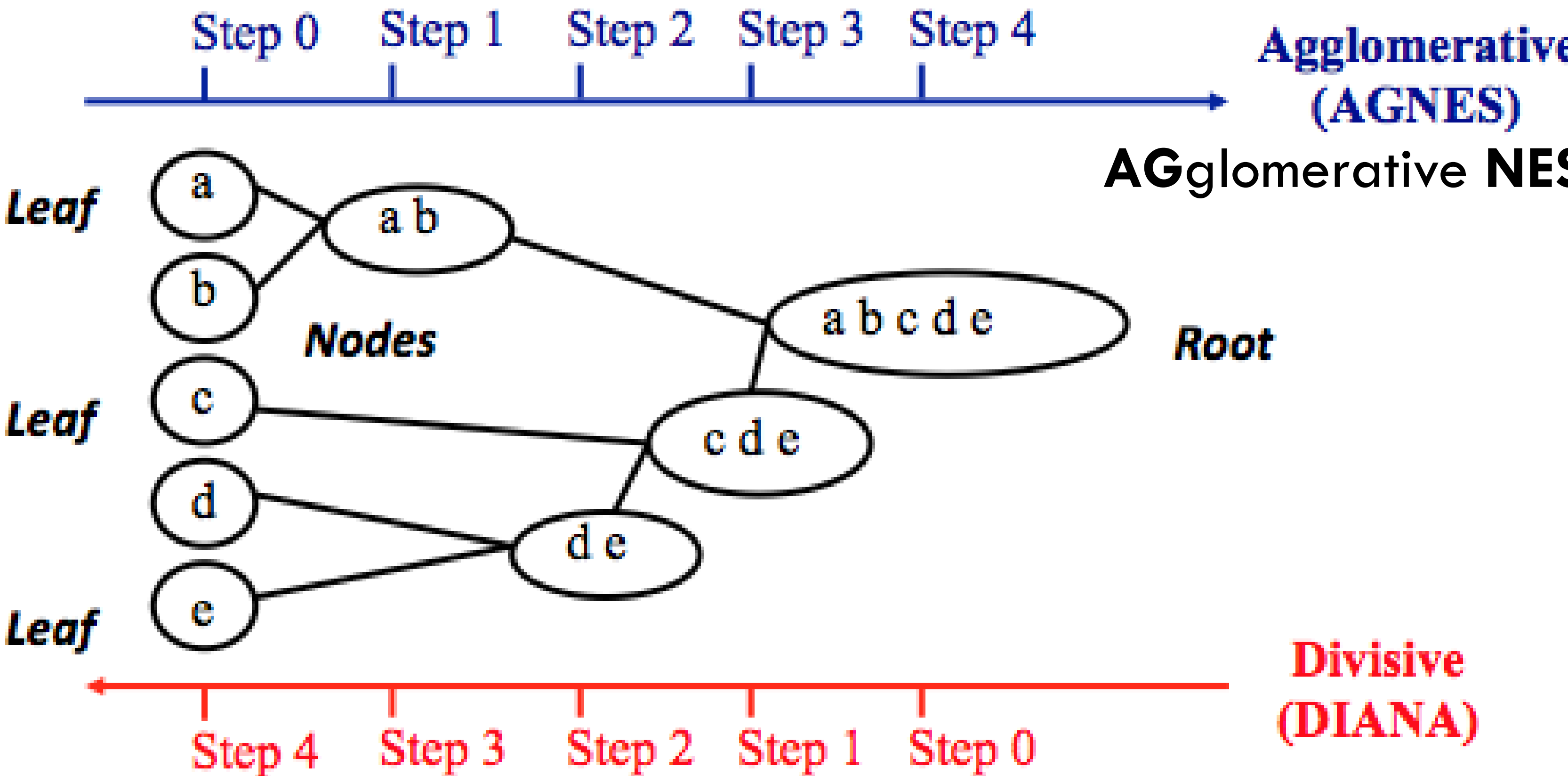
Hierarchical Clustering

- Remember: Feature Selection

- **Agglomerative Clustering:**
  - Start with *N* cluster: each point is one cluster
  - Merge several cluster

- **Divisive Clustering:**
  - Start with 1 cluster: All points are 1 cluster
  - Divide cluster

Hierarchical Clustering



AGglomerative **NES**ting

DIvisive **ANA**lysis

https://uc-r.github.io/hc_clustering

Hierarchical Clustering

How to decide merging and splitting?

- Cluster based on similarities/distances
- Distance between single points $\boldsymbol{x}_i, \boldsymbol{x}_j$
- Distance between cluster/groups $\mathcal{G}_k, \mathcal{G}_l$ containing several points

Hierarchical Clustering

- Distance between points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^D$
  $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iD})^{\mathrm{T}}$

  Minkowski (L_p)

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left[\sum_{d=1}^{D} |x_{id} - x_{jd}|^p \right]^{1/p}$$

  - p=2: Euclidean norm
  - p=1: City-block distance (Manhattan distance)
  - …

Hierarchical Clustering

Distance between two groups $\mathcal{G}_k, \mathcal{G}_l$

- Single-link:

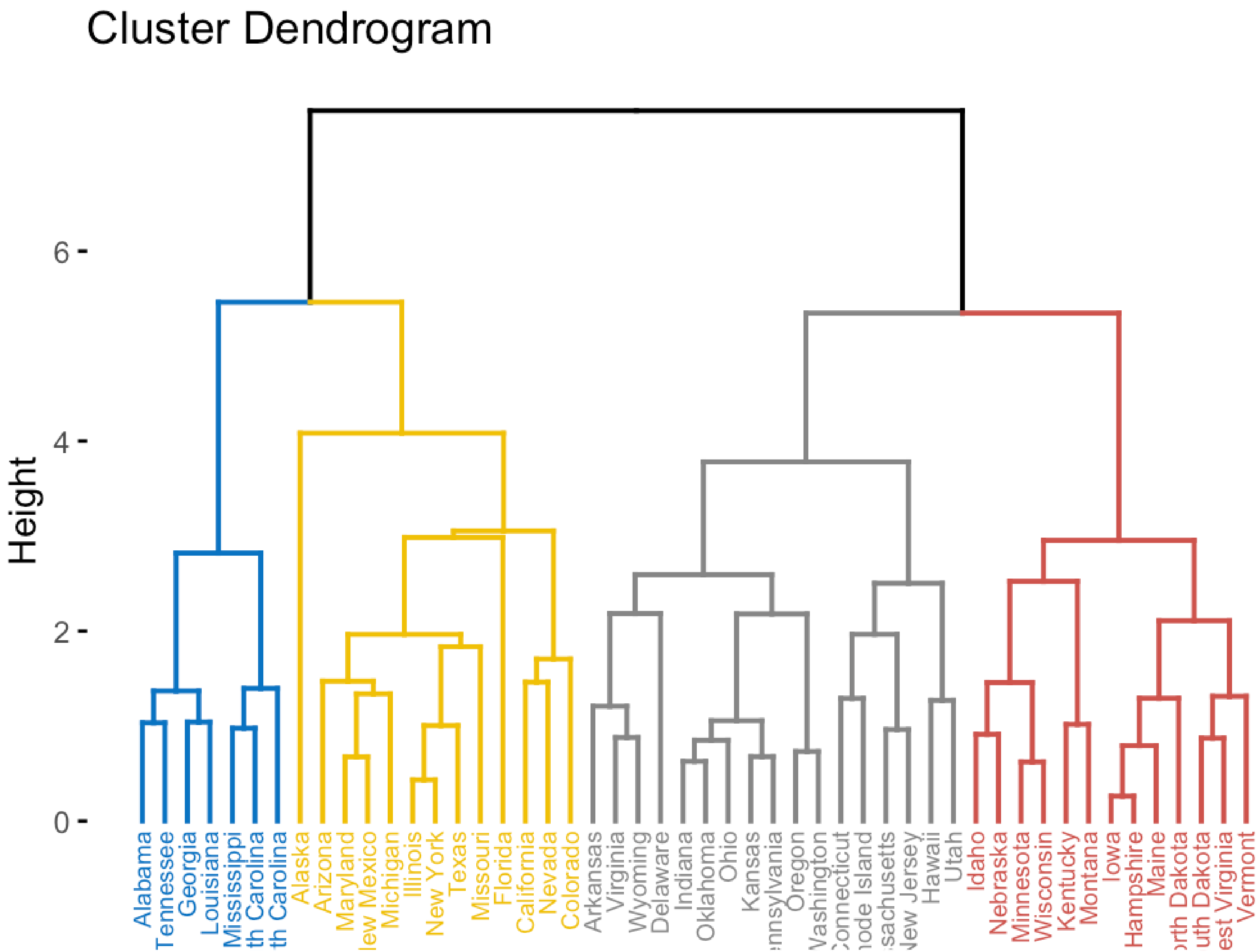
$$d(\mathcal{G}_k, \mathcal{G}_l) = \min_{\boldsymbol{x}_i \in \mathcal{G}_k, \boldsymbol{x}_j \in \mathcal{G}_l} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- Complete-link:

$$d(\mathcal{G}_k, \mathcal{G}_l) = \max_{\boldsymbol{x}_i \in \mathcal{G}_k, \boldsymbol{x}_j \in \mathcal{G}_l} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

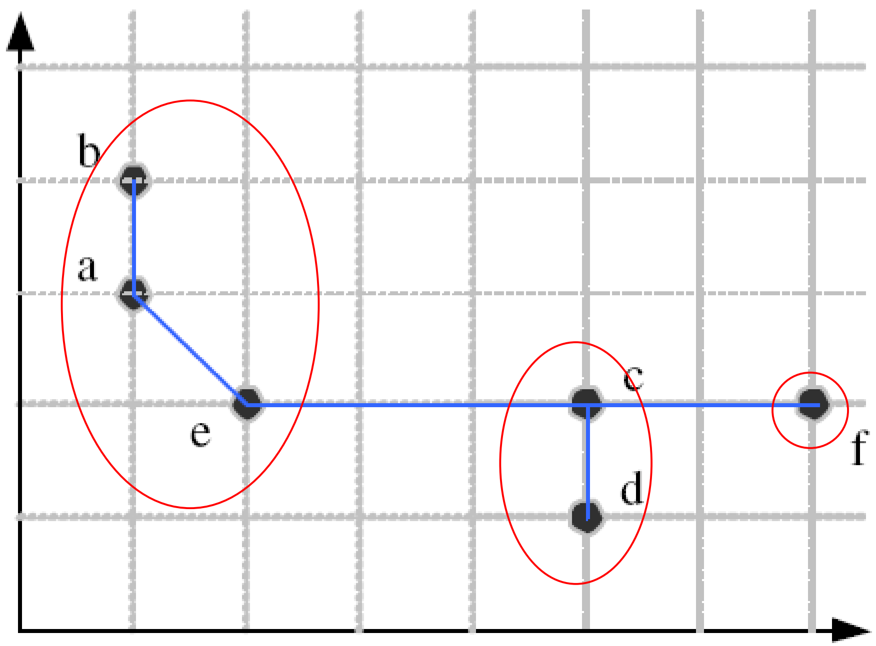
- Average-link, centroid

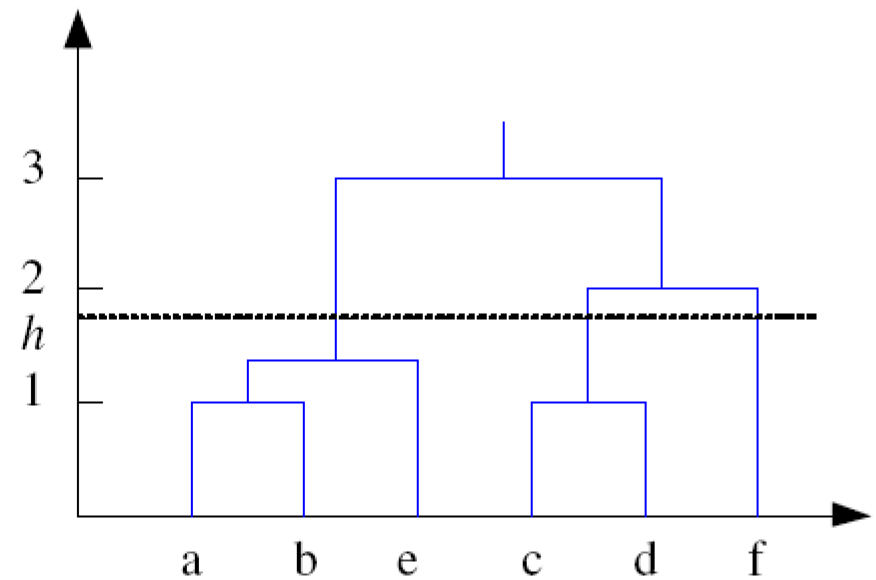
$$d(\mathcal{G}_k, \mathcal{G}_l) = \operatorname*{ave}_{\boldsymbol{x}_i \in \mathcal{G}_k, \boldsymbol{x}_j \in \mathcal{G}_l} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$$



https://www.datanovia.com/en/lessons/divisive-hierarchical-clustering/

Example: Single-Link Clustering





Dendrogram

How to decide number of clusters?

Hierarchical Clustering

- Pro:
  - Flexibility: free choice of distance measure
  - Easy to implement and widespread
  - Provides hierarchy of points and clusters
- Contra:
  - Runtime
  - No model
  - Imbalaced: outlier will end up in separate small cluster

K-Means Clustering

Input:

- *N* data points $\boldsymbol{x}_n \in \mathbb{R}^D, n = 1, \ldots, N$
- parameter K : number of classes

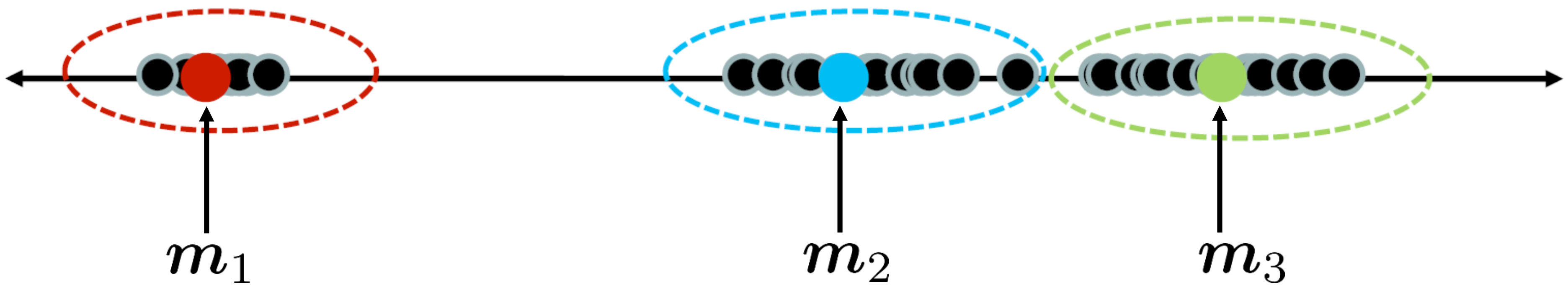
Output:

- labels for each point $\boldsymbol{x}_n \mapsto \boldsymbol{m}_k$
- Codebook of reference vectors $\boldsymbol{m}_k, \;\; k = 1, \ldots, K$
  Template Matching, Vector Quantization (VQ), …

=> N points are matched on K references
Compression from N to K

K-Means Clustering: K=3

- Assuming known $\boldsymbol{m}_k, \; k = 1, \ldots, K$
  cluster centers / templates / references
- Assign closest center to each data point: $\boldsymbol{x}_n \mapsto \boldsymbol{m}_k$



- Problem:
  - No class labels given
  - $\boldsymbol{m}_k$ are unknown

http://vision.stanford.edu/teaching/cs131_fall1314_nope/lectures/lecture13_kmeans_cs131.pdf

K-Means Clustering

1) Initialize K cluster centers $\boldsymbol{m}_k, \quad k = 1, \ldots, K$
2) Assign each point to closest cluster center

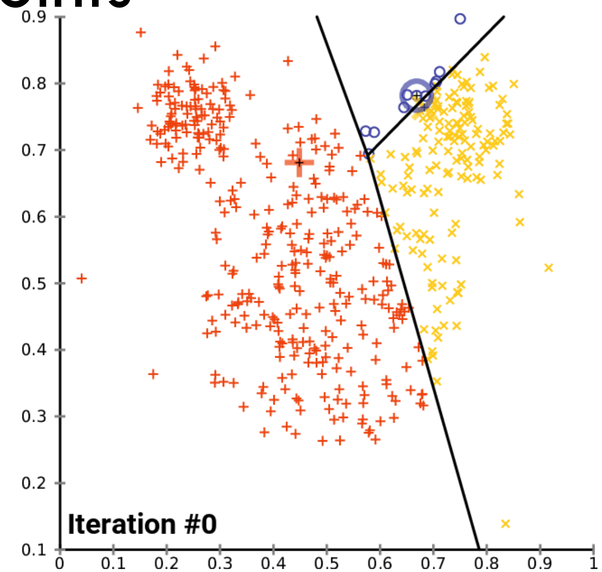
$$1(\boldsymbol{x}_n \mapsto \boldsymbol{m}_k) = \begin{cases} 1 & , \|\boldsymbol{x}_n - \boldsymbol{m}_k\|_2 = \min_i \|\boldsymbol{x}_n - \boldsymbol{m}_i\|_2 \\ 0 & , \ \text{else} \end{cases}$$

3) Update cluster center as mean of points

$$\boldsymbol{m}_k = \frac{1}{N_k} \sum_{n=1}^{N} \boldsymbol{x}_n 1(\boldsymbol{x}_n \mapsto \boldsymbol{m}_k)$$

$$N_k = \sum_{n=1}^{N} 1(\boldsymbol{x}_n \mapsto \boldsymbol{m}_k)$$

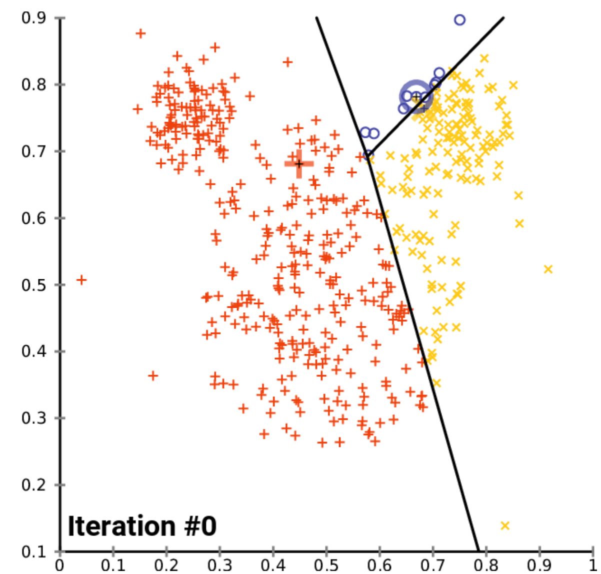
4) Repeat 2)-3) until convergence



https://en.wikipedia.org/wiki/K-means_clustering

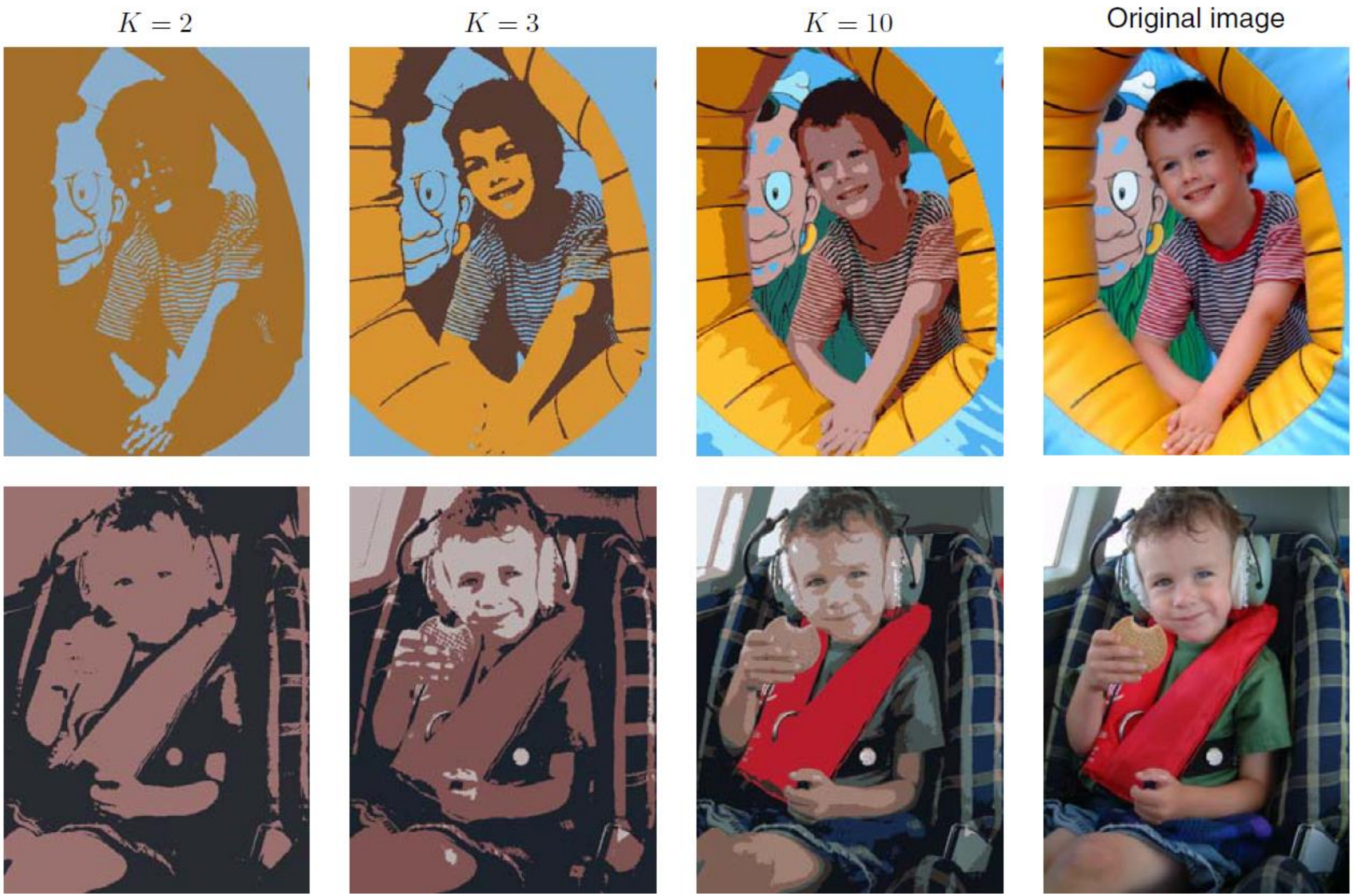
K-Means Clustering

- **Initialization**
  - Randomly selected K data points
  - Divide data space in intervals: choose mean of these
  - …
- **Convergence**
  - If cluster centers do not change anymore
  - …
- How to choose K?
  - Depends on application
  - Re-run and inspect the results
  - look at data !



https://en.wikipedia.org/wiki/K-means_clustering

K-Means: Image Segmentation



Bishop, "Pattern Recognition and Machine Learning", 2006, Fig. 9.3

K-Means Clustering

- Different variants, e.g.
  - assign cluster, such that variance is minimzed
  - Fuzzy K-means: assign probabilty instead of one cluster
- **Pro**
  - easy to implement
- **Contra**
  - Need to choose K
  - Sensitive to initial cluster centers
  - Local minima
  - Spherical structure of clusters (euclidean norm)

Spectral Clustering

- First process data, then clustering
- Given pairwise similarities (or " minus" distance)

$$w_{ij} = w_{ji} = \mathrm{similarity}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- Steps:
  I. Use these values for Laplacian Eigenmaps (Ch. 6) to map $\boldsymbol{x}_n$ to a new space $\boldsymbol{z}_n$
  II. Use *k*-means on $\boldsymbol{z}_n$ for clustering

Expectation-Maximization (EM)

Probability density function (pdf) vs. Likelihood

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad p(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$$

- Pdf: evaluate on data **given parameters**

$$p(\boldsymbol{x}|\boldsymbol{\theta}),\ \boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Likelihood: evaluate on parameters **given data**

$$l(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{x}),\ \boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Given data:
  - maybe missing values, no labels
  - Unknown parameters of pdf

=> Complete likelihood unknown

Expectation-Maximization (EM)

- Log likelihood with iid samples $\boldsymbol{x}_n \in \boldsymbol{X}$

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}) = \log \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta})$$

$$= \sum_{n=1}^{N} \log \sum_{k=1}^{K} p(\boldsymbol{x}_n|\mathcal{G}_k) P(\mathcal{G}_k)$$

- Assume hidden variables $\mathcal{Z}$
  when known, make optimization much simpler
- Complete likelihood $\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{X}, \mathcal{Z})$
- Incomplete likelihood $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X})$

Expectation-Maximization (EM)

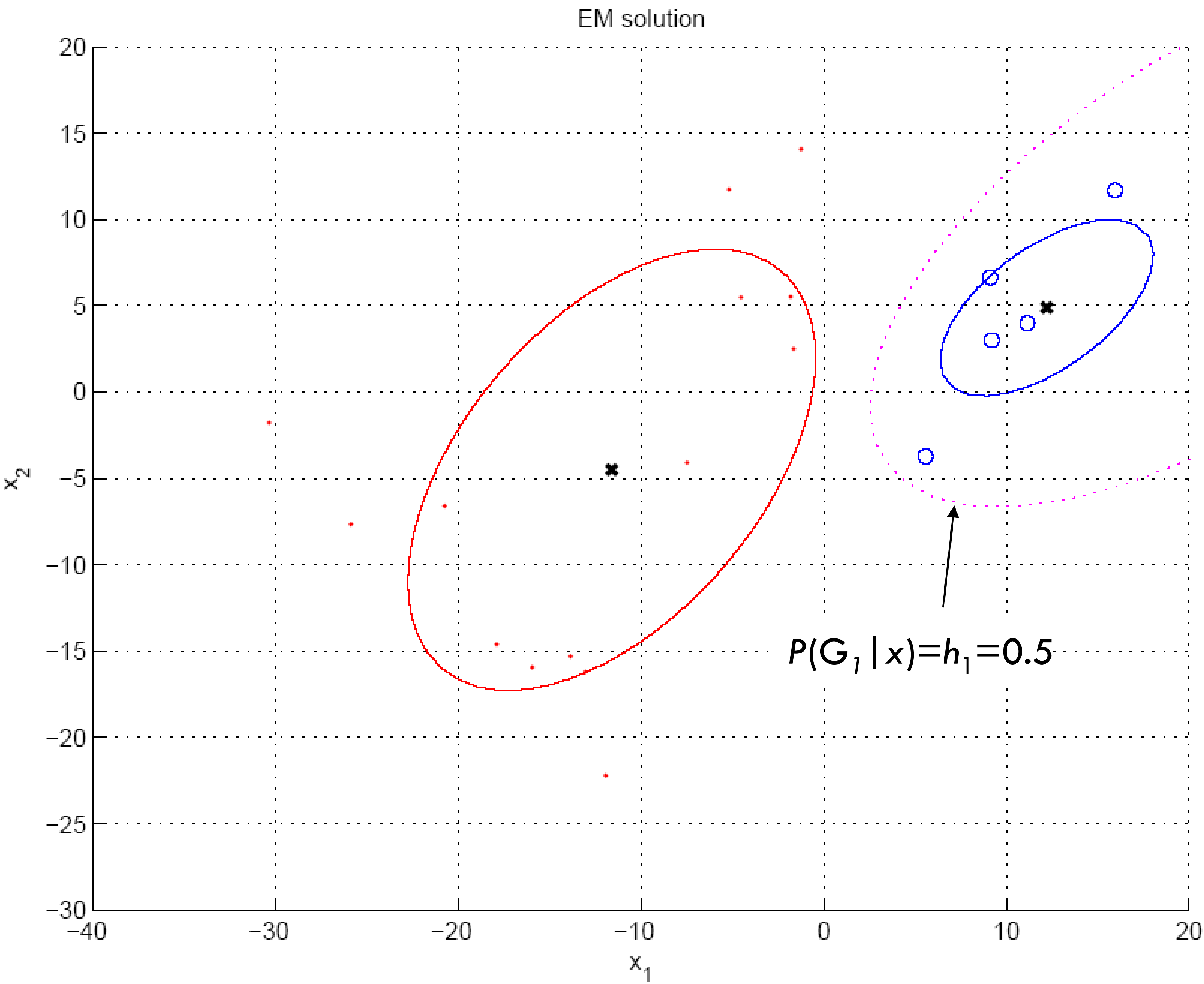
- **Initialize** parameters, set: $\boldsymbol{\theta}^0$
- **Expectation**: given parameters: $\boldsymbol{\theta}^l$
  - Estimate hidden variables $\mathcal{Z}$
  - Get expected value of likelihood

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^l) = E[\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{X},\mathcal{Z})|\boldsymbol{\theta}^l]$$

- **Maximization**:
  - Estimate parameters which maximize likelihood

$$\boldsymbol{\theta}^{l+1} = \mathrm{argmax}_\theta \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^l)$$

- Repeat E-step and M-step


EM solution
x2
x1
P(G1 | x)=h1=0.5
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−40
−30
−20
−10
0
10
20

Mixtures of Latent Variable Models

Regularize clusters

1. Assume shared/diagonal covariance matrices
2. Use PCA/FA to decrease dimensionality: Mixtures of PCA/FA

FA: $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{z} + \boldsymbol{\epsilon}$ $\qquad \mathrm{Cov}(\boldsymbol{z}) = \boldsymbol{I},\ \mathrm{Cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Psi}$

$$p(\boldsymbol{x}|\mathcal{G}_k) = \mathcal{N}(\boldsymbol{m}_k, \boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}} + \boldsymbol{\Psi}_k)$$

Choosing *K*

- Defined by the application, e.g., image quantization
- Plot data (after e.g. PCA) and check for clusters
- Incremental (leader-cluster) algorithm:
  Add one at a time until large change
- Manually check for meaning

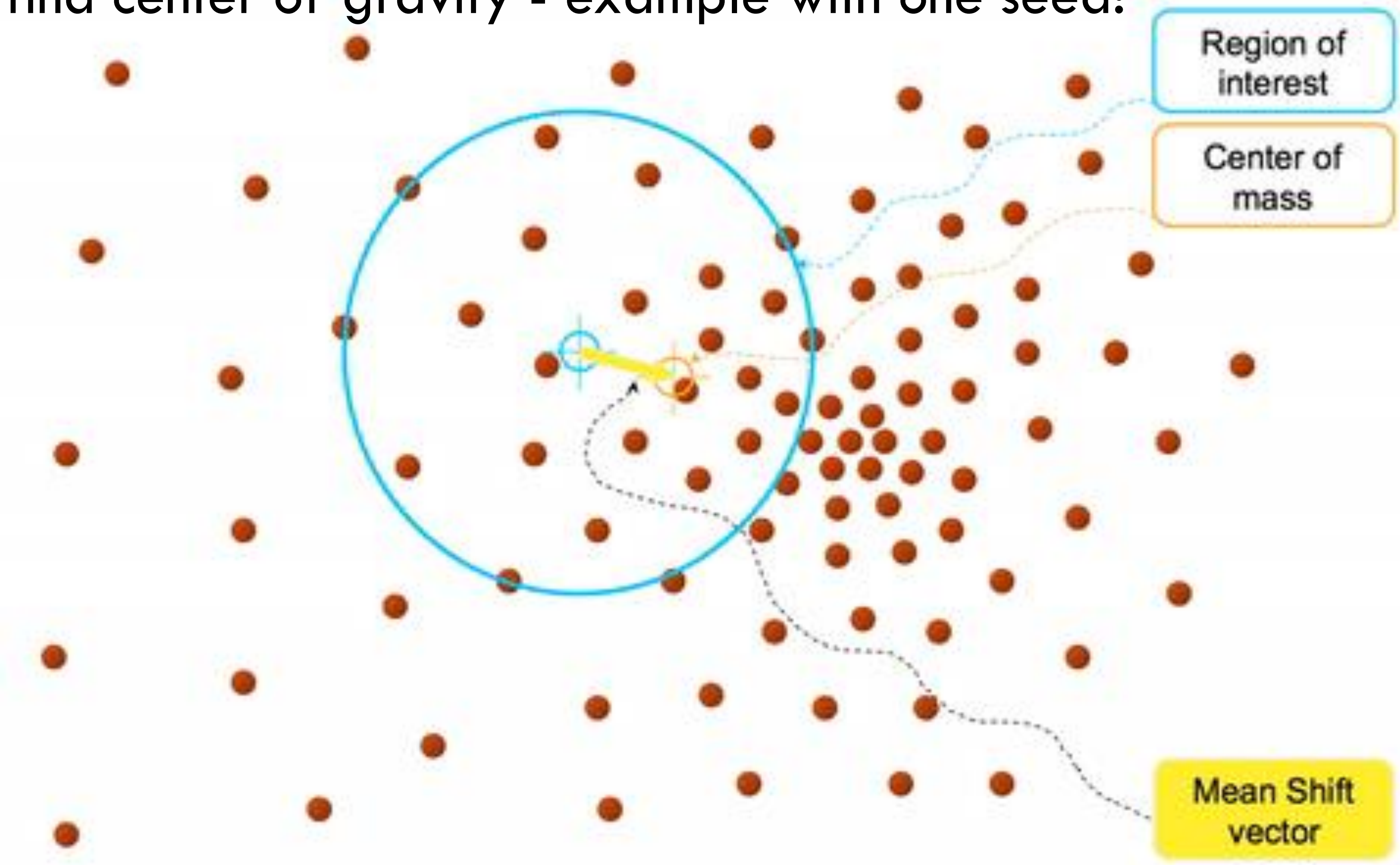
Mean Shift: no need to choose K

Mean Shift instead of K-means
Parameter: window function and width
=> locates maxima

1) Initialize random seed $\boldsymbol{m}_{(0)}$ and window W
2) Calculate new center of gravity (mean) $\boldsymbol{m}_{(l)}$ as (weighted) mean of all points in window $\boldsymbol{x}_n \in W(\boldsymbol{m}_{(l-1)})$ centered at $\boldsymbol{m}_{(l-1)}$
3) Shift search window to new mean $\boldsymbol{m}_{(l)}$
4) Repeat 2-3

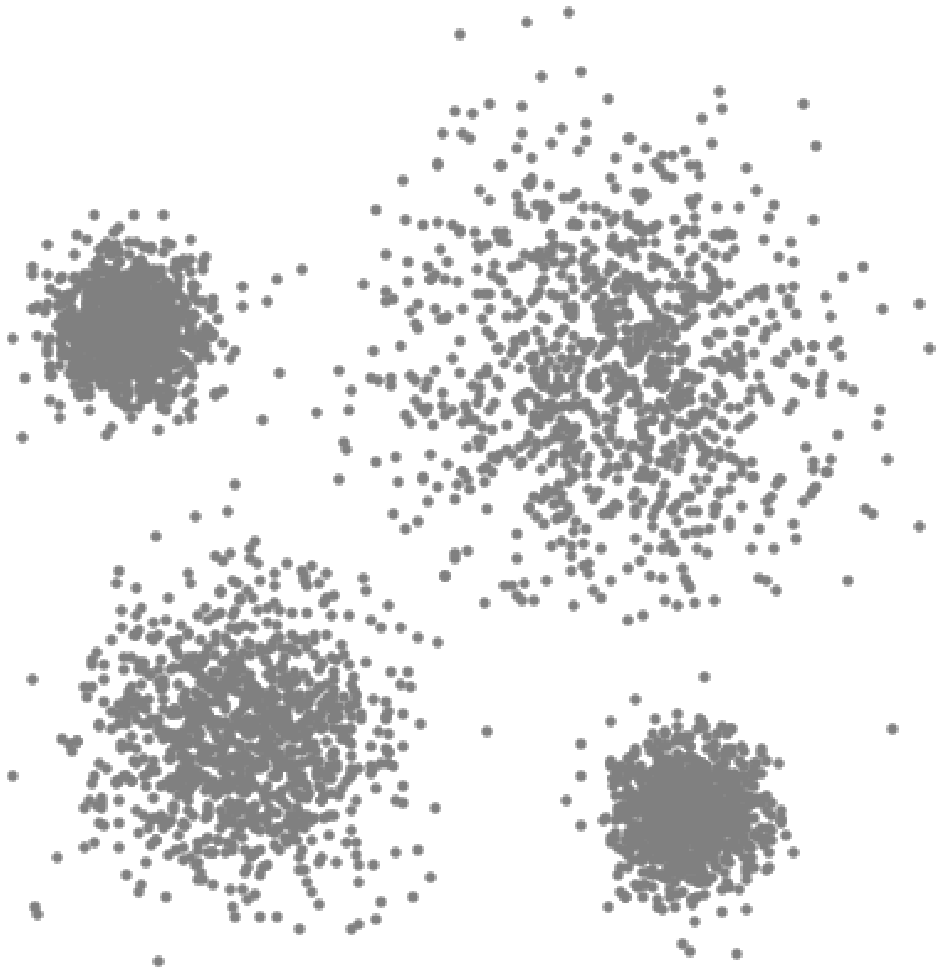
Mean Shift

find center of gravity - example with one seed:



https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html

Mean Shift



Initialize:

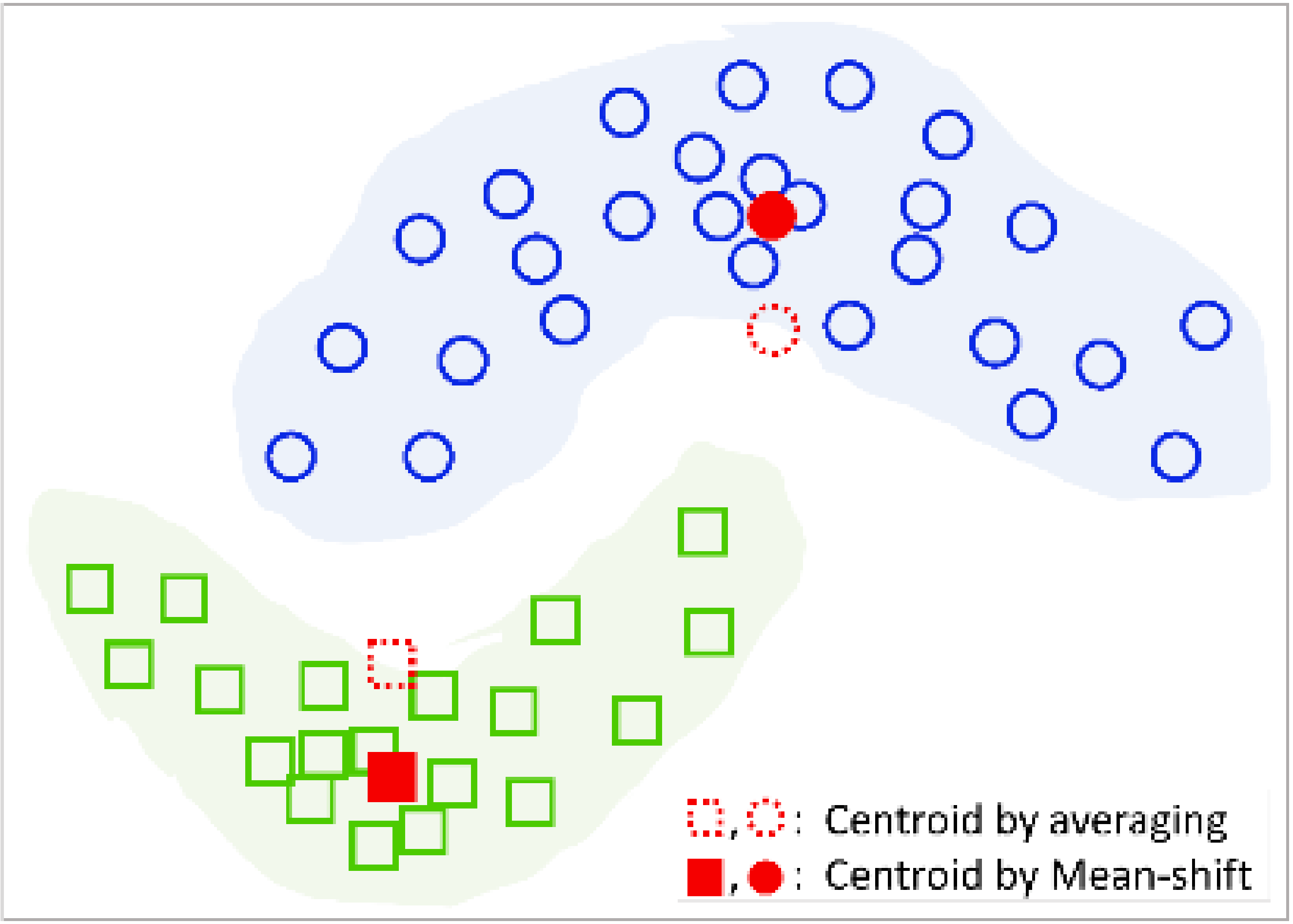
- random starting points
- Each data point

Mean Shift:

- for start each point: where does it end up?
- Different points share the same center of gravity = cluster center

https://giphy.com/gifs/shift-dashee87githubio-scikit-1p9odvZq5lpxK

Mean Shift



https://www.mdpi.com/1996-1073/11/9/2397/htm

Mean Shift:

Pro:

- No need to choose number of clusters K
  => finds variable number of clusters
- Only one parameter windows size:
  has physical meaning
- Model-free
- Robust to outliers

Contra:

- Result depends on window size (not trivial)
- Computationally expensive
- Attention in high dims!

After Clustering

- **Dimensionality reduction**
  remove redundant information, e.g.
  - correlations between features and group features
- **Clustering methods** find similarities between instances and group instances
  => Analysis
- knowledge extraction through
  - number of clusters
  - prior probabilities
  - cluster parameters, i.e., center, range of features.

Example: CRM, customer segmentation

Clustering as Preprocessing

$$\boxed{b_k^n := \mathbb{1}\left\{\boldsymbol{x}_n \in \mathcal{G}_k\right\} = \begin{cases} 1 & , \ \boldsymbol{x}_n \in \mathcal{G}_k \\ 0 & , \ \text{else} \end{cases}} \text{ vs. } \boxed{h_i^k := P(\mathcal{G}_k | \boldsymbol{x}_i)}$$

- Estimated group labels: hard vs. soft dimensions of a new K-dim. space => learn discriminant or regressor
- **Local** representation
  - only one b_i is 1, all others are 0
  - only few h_i are nonzero

vs

Distributed representation, e.g.: after PCA all z_i are nonzero

Summary

- Dimensionality reduction as preprocessing
- Clustering to:
  - analyze data (subgroups, etc.)
  - Estimate classes
- Classification
  - Given class labels

APPENDIX