Machine Learning for Networking
ML4N

Luca Vassio
Gabriele Ciravegna
Zhihao Wang
Tailai Song

The three components of ML

- …
- …
- ...

ML for supervised

- Learn to predict the label y of a data point from its features **x**
- ML model = learn a hypothesis $h \in \mathcal{H}$ $h : \mathcal{X} \rightarrow \mathcal{Y}$

  such that h(**x**) ≈ y
- Loss function: how to quantify/weight error between y and h(**x**)

ML for supervised

- Learn to predict the label y of a data point from its features **x**
- ML model = learn a hypothesis $h \in \mathcal{H}$ $h : \mathcal{X} \rightarrow \mathcal{Y}$

  such that h(**x**) ≈ y
- Loss function: how to quantify/weight error between y and h(**x**)

Still valid for unsupervised ML, with minor modifications

Clustering

https://scikit-learn.org/stable/modules/clustering.html

Learning goals

- **Hard** clustering and **soft** clustering
- **k-Means** method for hard clustering
- **Gaussian Mixture Models** for soft clustering
- Other clustering techniques
- How to **compare** clustering techniques

What is a Cluster?

Noun [edit]

cluster (*plural* **clusters**)

1. A group or bunch of several discrete items that are close to each other. [quotations ▼]

   *a **cluster** of islands*

   *A **cluster** of flowers grew in the pot.*

   *A **leukemia** cluster has developed in the town.*

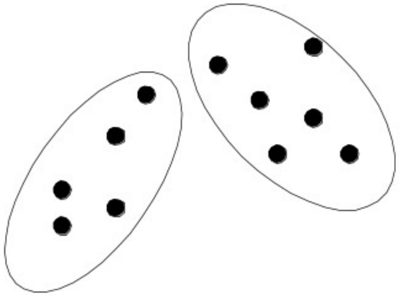
https://en.wiktionary.org/wiki/cluster

Unsupervised learning: Clustering

- Detecting **groups of similar data points**
- **Unsupervised**: no labelled data points, no ground truth to compare with, no y

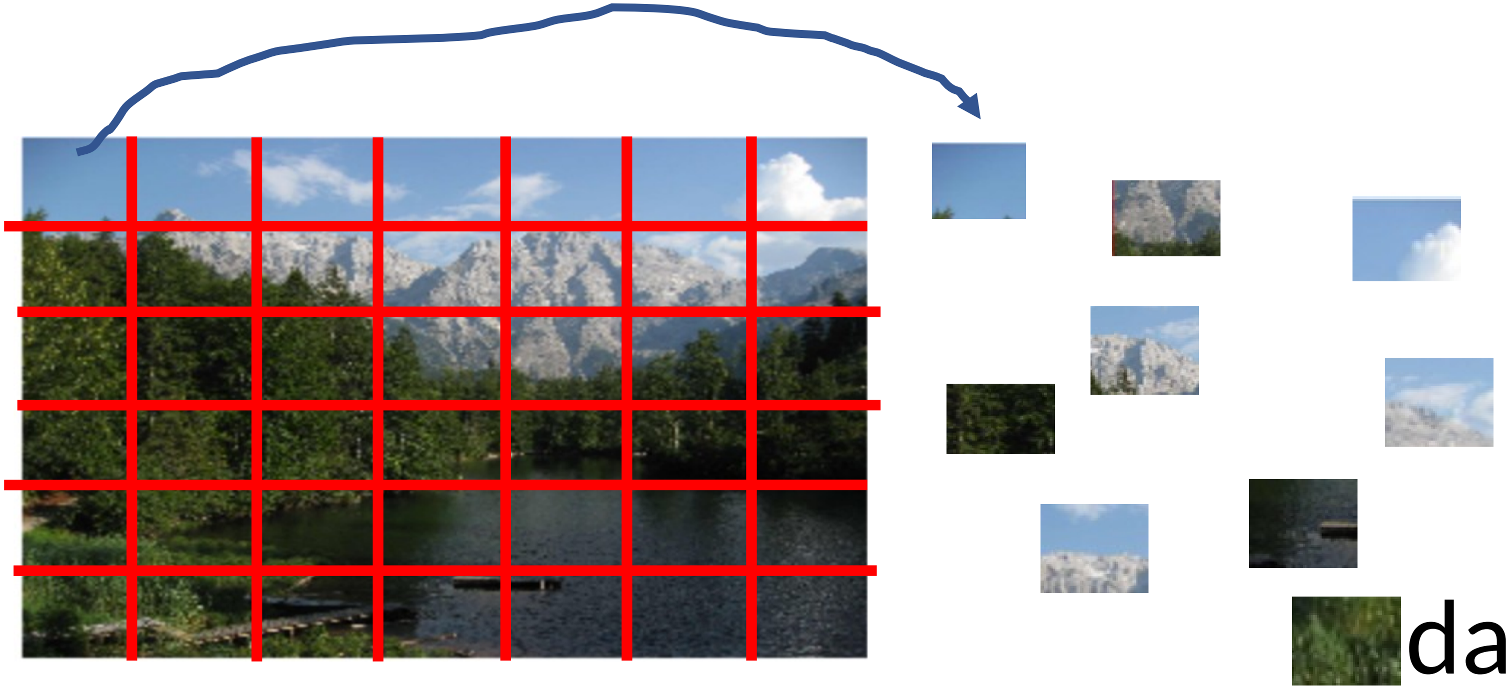
Unsupervised learning: Clustering

- A **cluster** corresponds to a **subset of data points** that are in some sense **homogeneous or similar**
- At the same time **different from** (or unrelated to) the data points in **other subsets**
- Plethora of different definitions for “homogeneous” and “similar/different”



Clustering for image segmentation

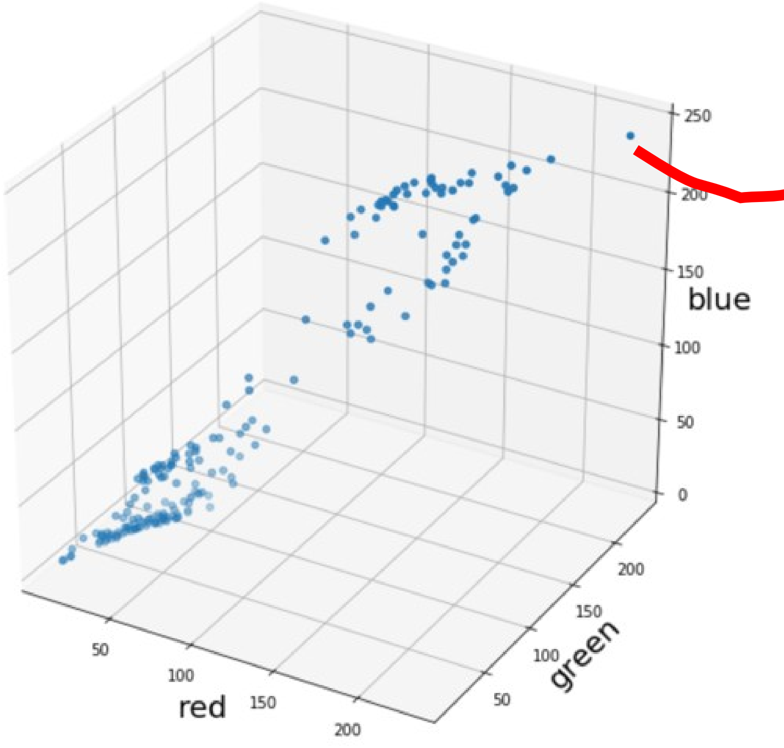
- dataset=patches of an image



data point

Clustering for image segmentation

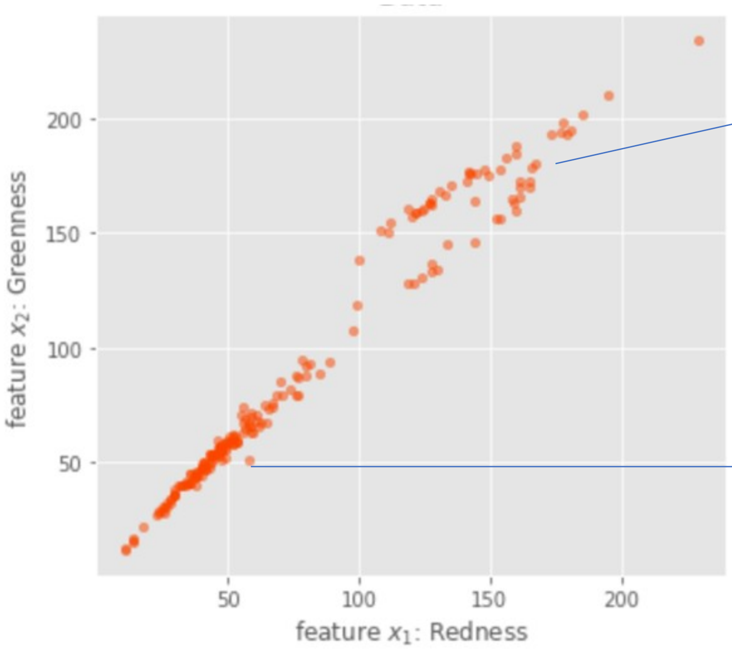
- Using three features

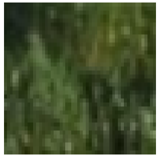


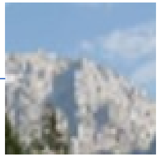
three features:
average red, green and blue component

Clustering for image segmentation

- Using two features (Red+Green)



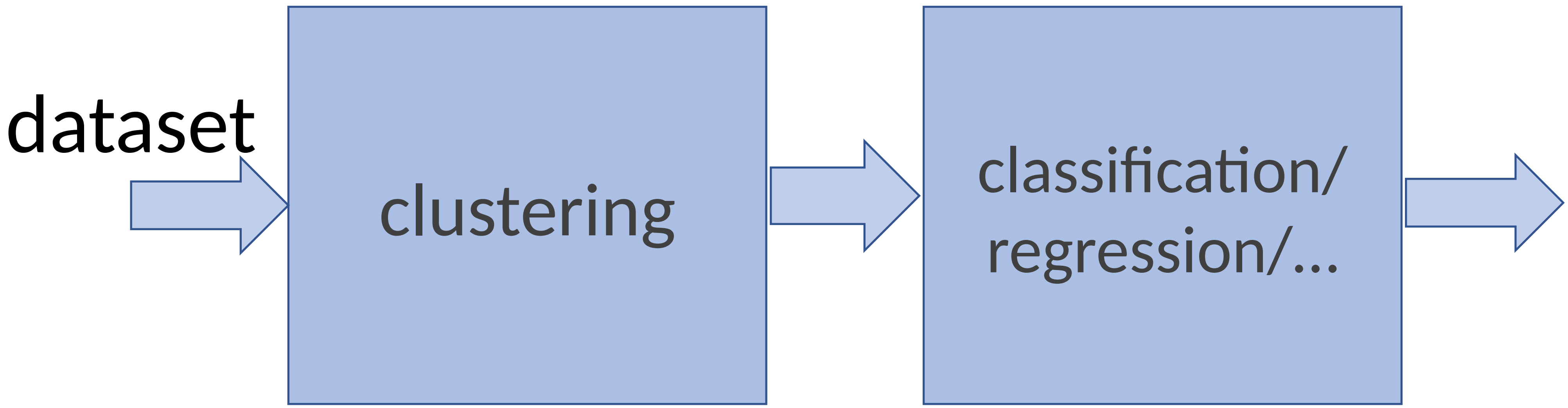




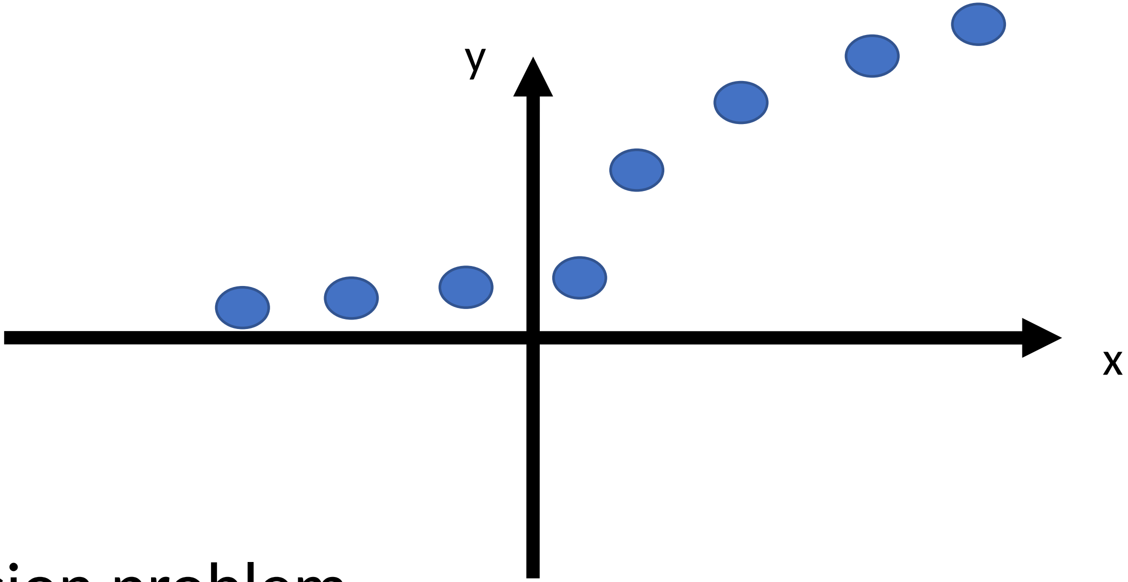
Clustering for image segmentation

- The two clusters segment the image

Clustering as pre-processing

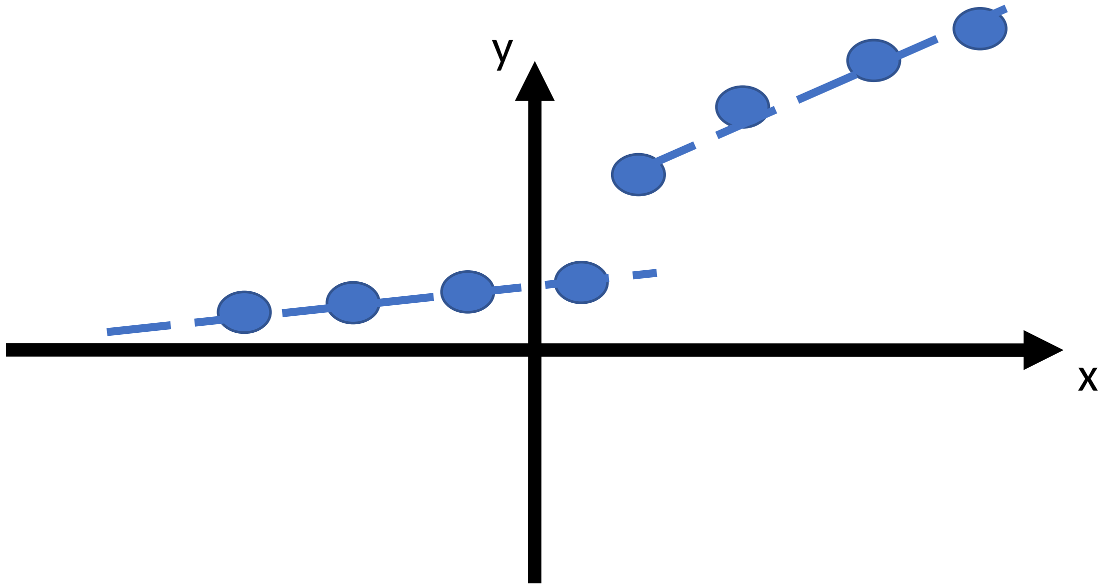


Clustering as pre-processing



- Regression problem
- Use linear regression

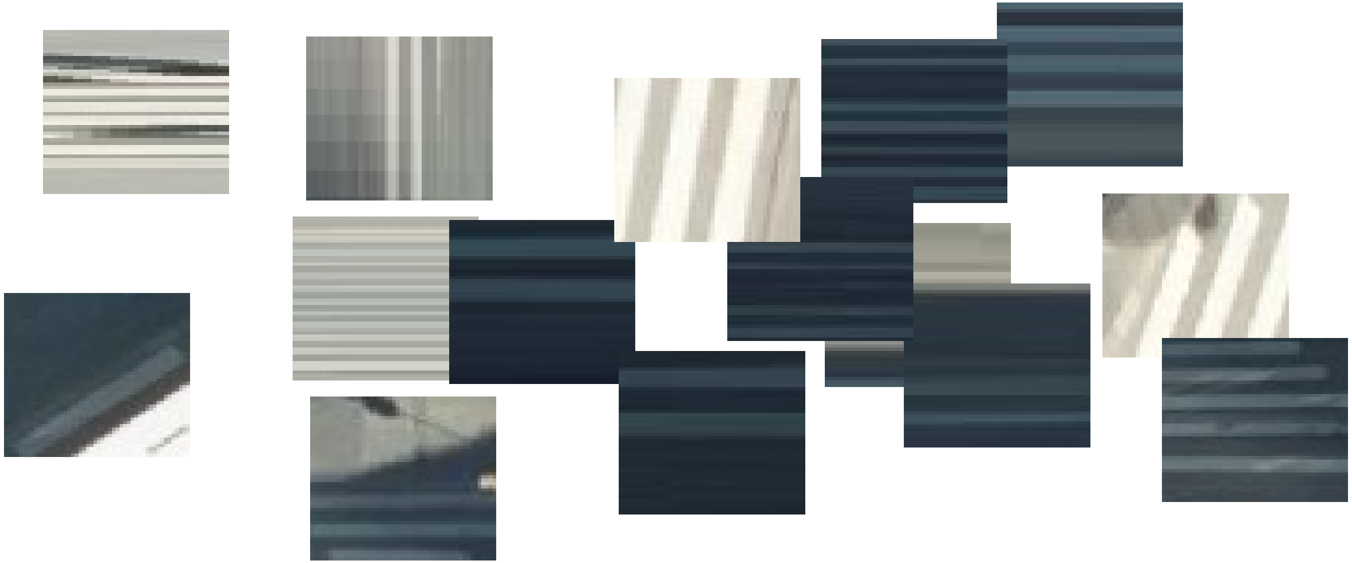
Clustering as pre-processing



- First partition into two clusters
- Then apply linear regression separately to each cluster

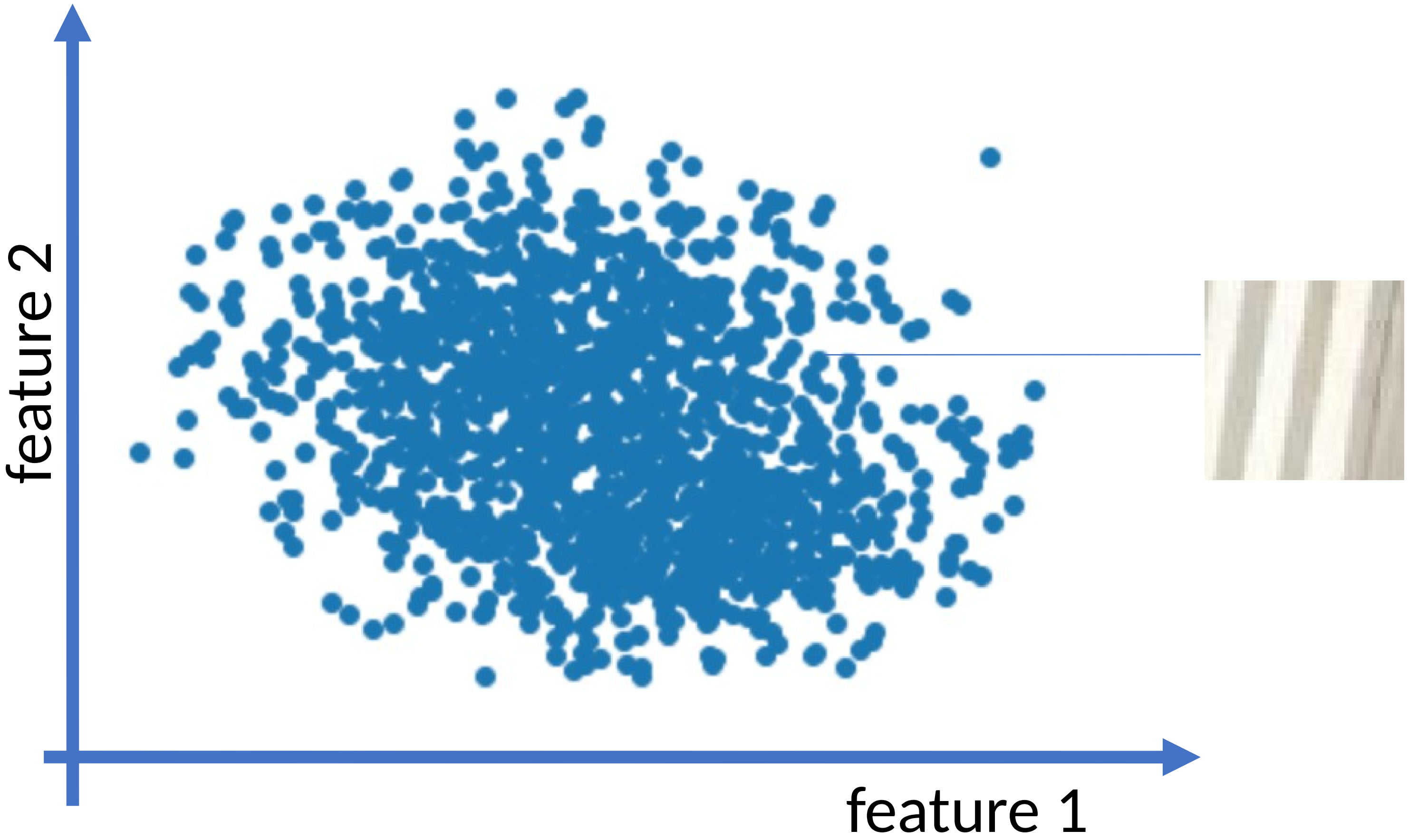
Clustering for outlier detection

- Dataset=”some images”

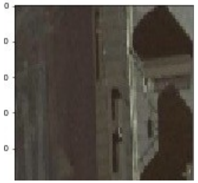


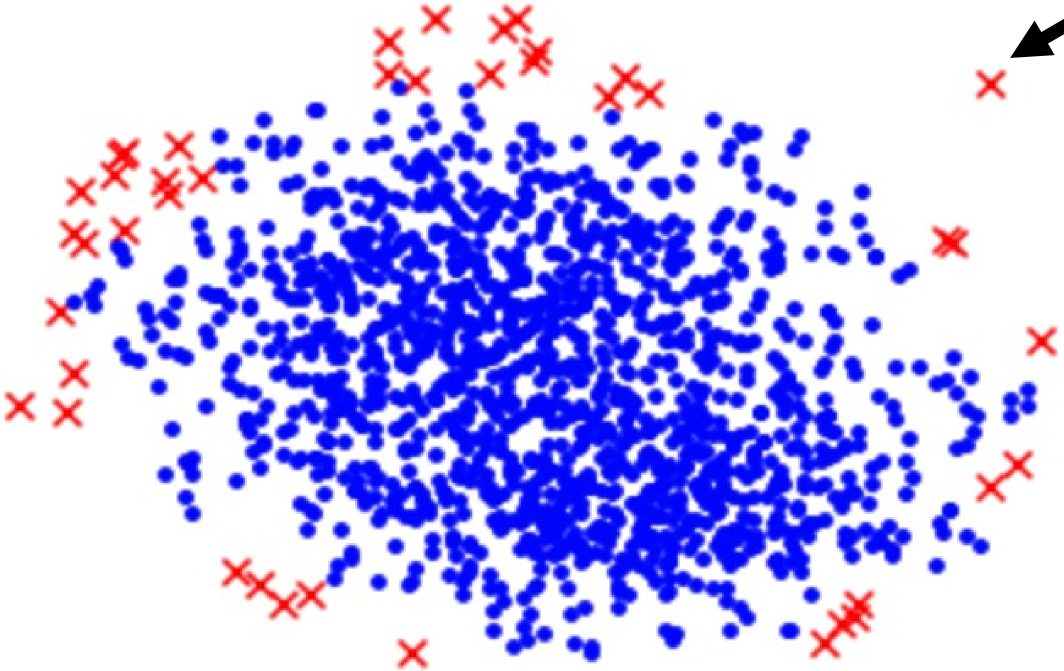
https://kartta.hel.fi/

Clustering for outlier detection

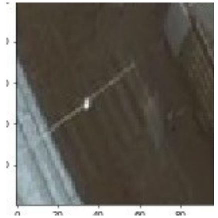


Clustering for outlier detection





“outlier”



Clustering: ambiguity

- Notion of a cluster can be ambiguous

How many clusters?

Clustering: ambiguity

- Notion of a cluster can be ambiguous

How many clusters?

Four Clusters

Two Clusters

Six Clusters

Hard Clustering

Hard clustering

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}.$$

Data points characterized by n features

- Features of i-th data

$$\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$$

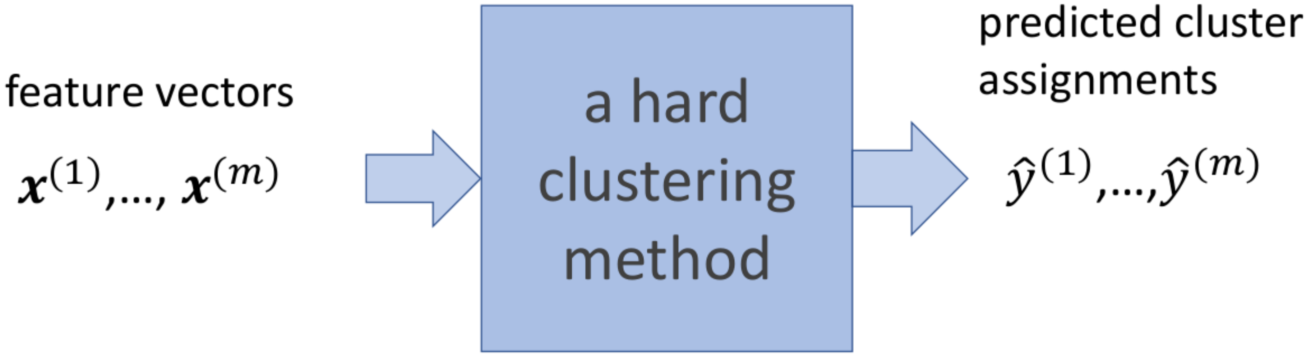
- i-th data point belongs to one of k clusters
- Cluster index of i-th datapoint is $y^{(i)} \epsilon \{1, \ldots, k\}$

Hard clustering

- Cluster index of i-th datapoint is $y^{(i)} \epsilon \{1, \ldots, k\}$
- Hard clustering methods compute predicted cluster indices $\hat{y}^{(i)}$ based solely on features $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$
- Does not require true cluster index $y^{(i)}$ of any data point

Hard clustering

- Cluster index of i-th datapoint is $y^{(i)} \epsilon \{1, \ldots, k\}$
- Hard clustering methods compute predicted cluster indices $\hat{y}^{(i)}$ based solely on features $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$
- Does not require true cluster index $y^{(i)}$ of any data point

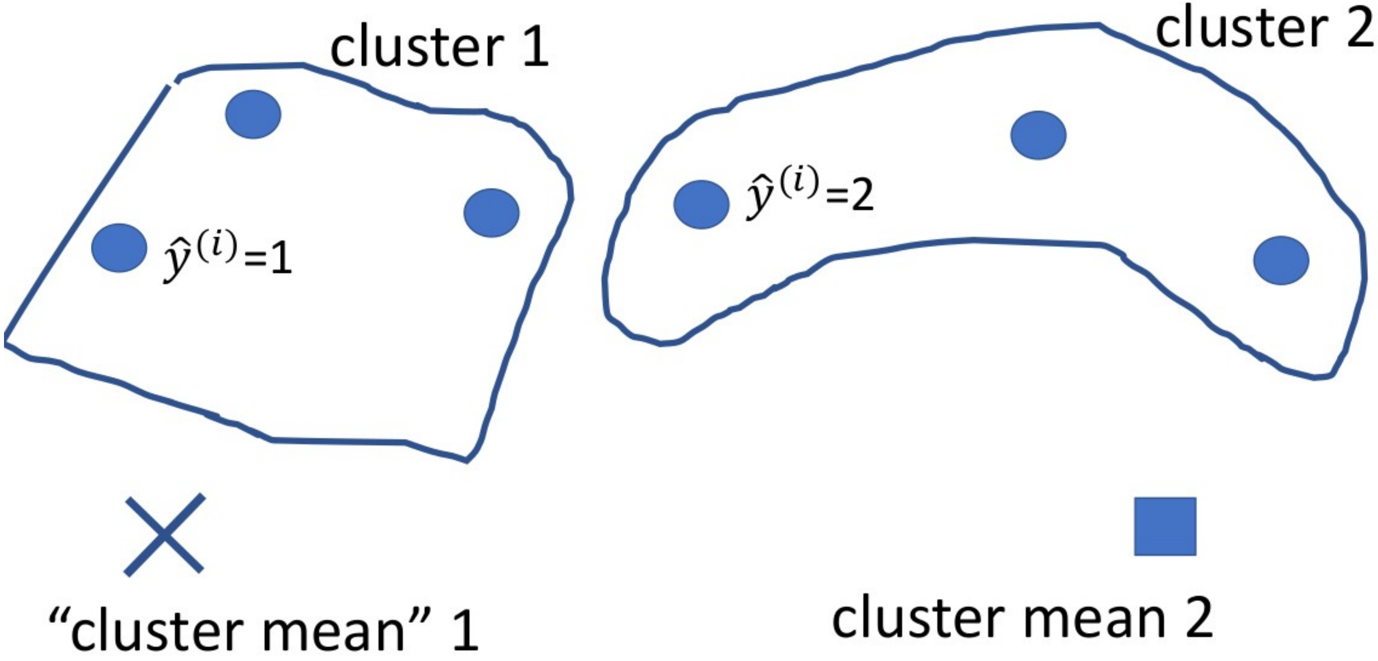


Hard Clustering with k-Means

k-Means clustering algorithm

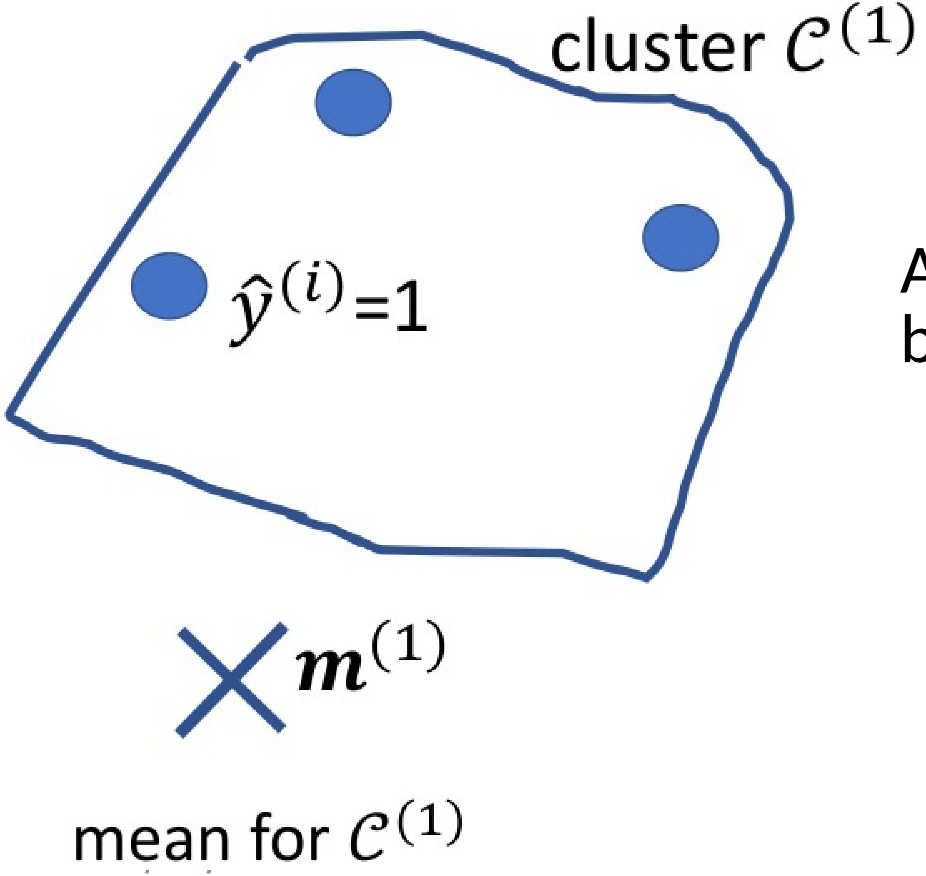
- It is characterized by one important hyper-parameter: the number of clusters k
- Each cluster is associated with a point (mean or centroid)

Representing a cluster by a point



Representative point: prototype feature vector for the cluster
Let’s call it “mean”

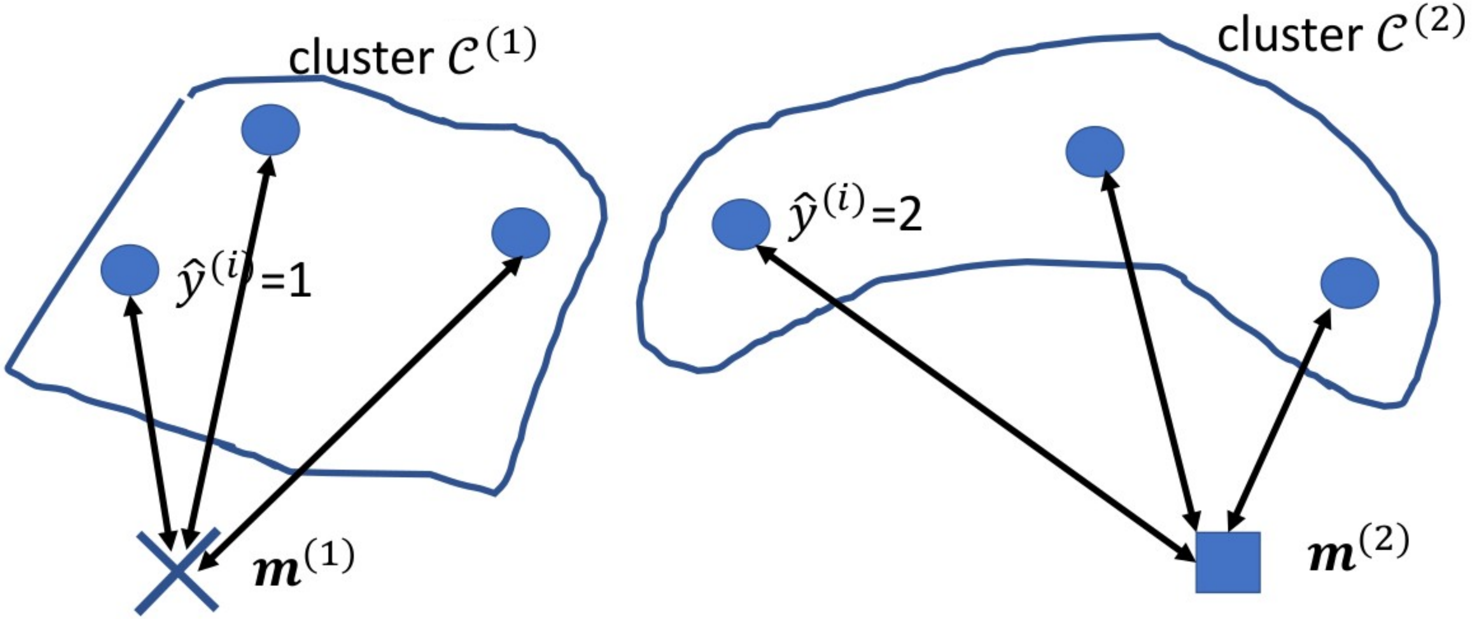
Cluster spread



Average squared Euclidean distance between points and mean of cluster:

$$(1/|\ \mathcal{C}^{(1)}\ |)\sum_{i\in\mathcal{C}^{(1)}}\left\|\boldsymbol{m}^{(1)}-\boldsymbol{x}^{(i)}\right\|^2$$

Clustering Error



$$\frac{1}{m}\sum_{c=1}^{2}\sum_{i\in\mathcal{C}^{(c)}} ||\mathbf{m}^{(c)} - \mathbf{x}^{(i)}||^2$$

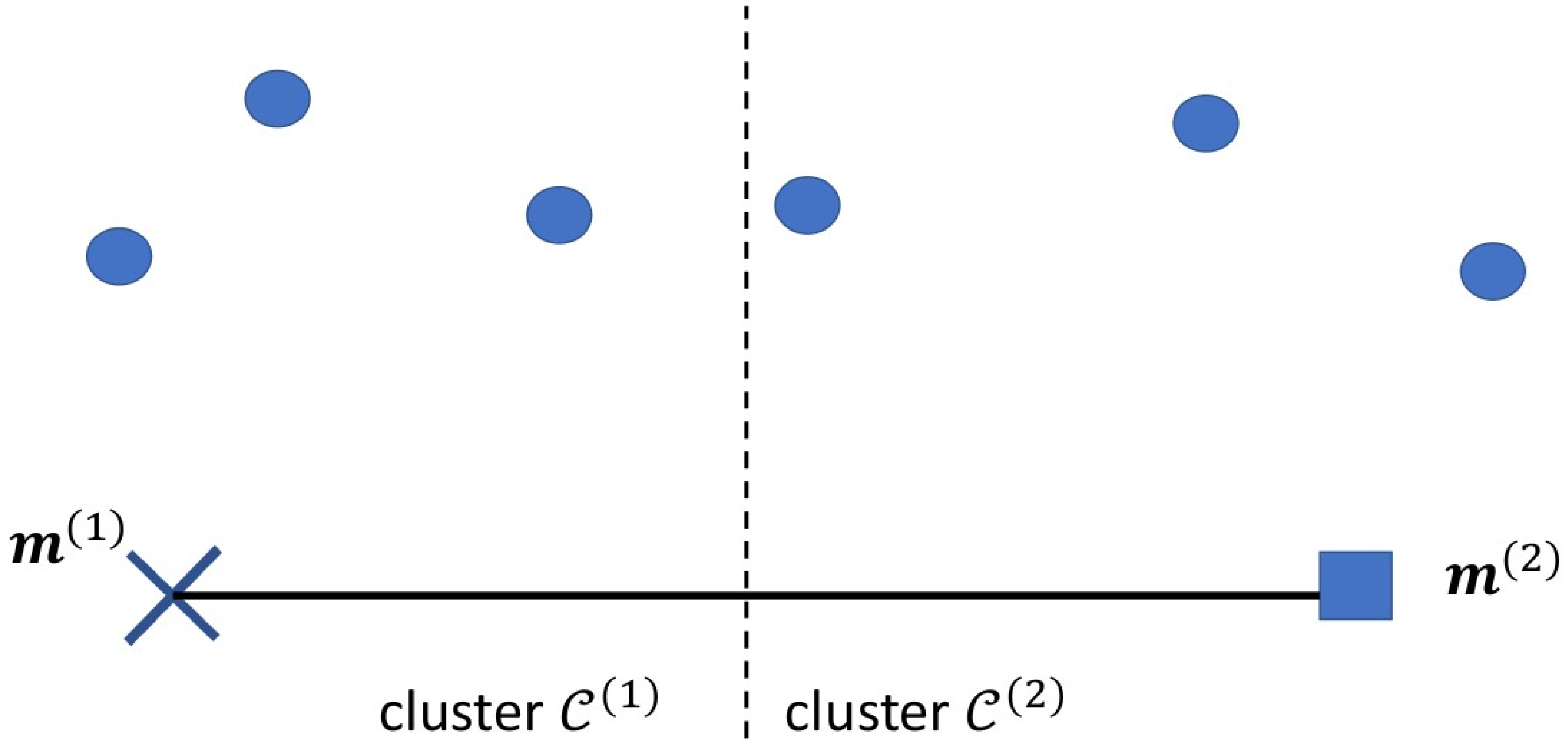
Update cluster assignments

- For given cluster means, clustering error is minimized by assigning i-th data point to cluster with **nearest cluster mean**

$$\hat{y}^{(i)} := c$$

with $$\left\|\boldsymbol{m}^{(c)} - \boldsymbol{x}^{(i)}\right\|^2 = \min_{c'=1,\dots,k} \left\|\boldsymbol{m}^{(c')} - \boldsymbol{x}^{(i)}\right\|^2$$

Update cluster assignment



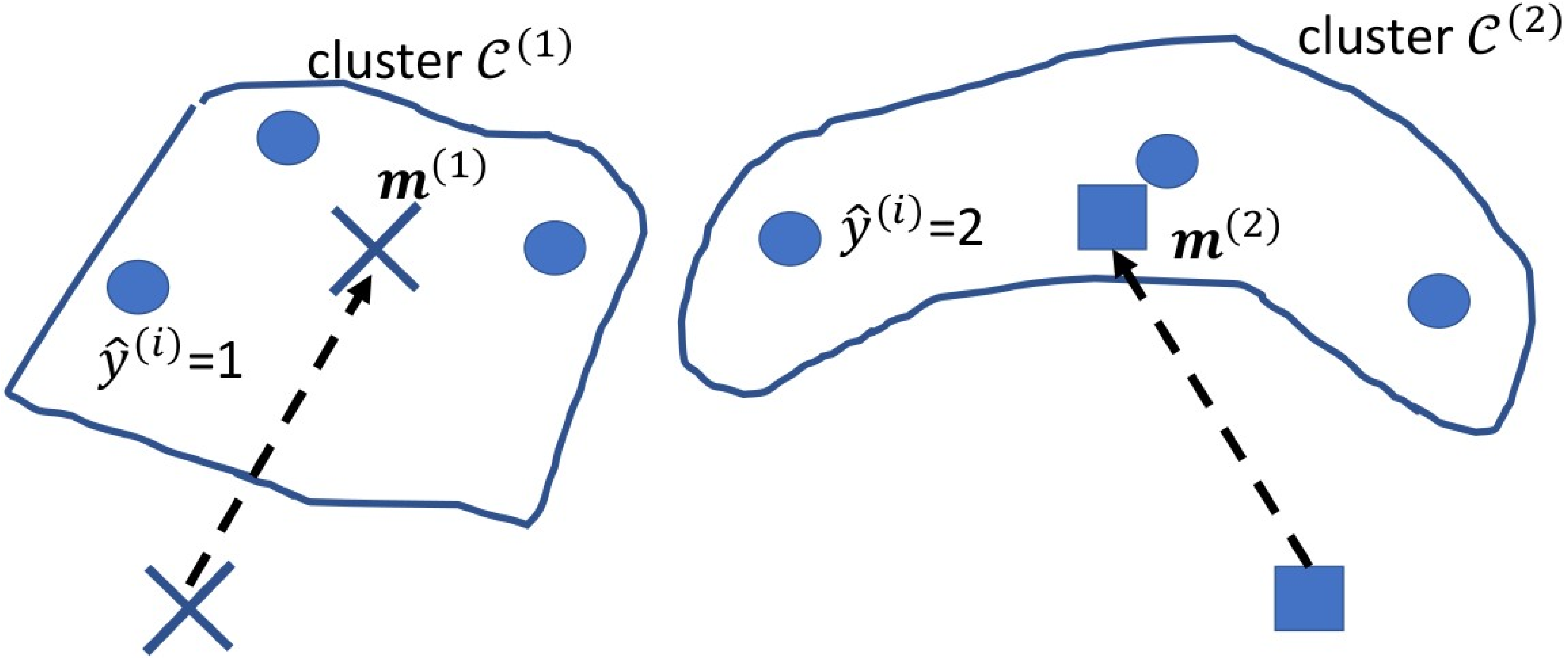
Update cluster means

- For given cluster assignments, clustering error is minimized by representing c-th cluster by the cluster mean of feature vectors of its data points

$$m^{(c)} := \frac{1}{|\mathcal{C}^{(c)}|} \sum_{i \in \mathcal{C}^{(c)}} \boldsymbol{x}^{(i)}$$

with cluster $\mathcal{C}^{(c)} = \{i : \hat{y}^{(i)} = c\}$

Update cluster means



Minimizing the clustering error

- Clustering error, to be minimized

$$\mathcal{E}\left(\{m^{(c)}\},\{\hat{y}^{(i)}\}\right) := \frac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{m}^{(\hat{y}^{(i)})} - \boldsymbol{x}^{(i)}\right\|^2$$

Minimizing the clustering error

- Clustering error, to be minimized

$$\mathcal{E}\big(\{m^{(c)}\},\{\hat{y}^{(i)}\}\big) := \frac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{m}^{(\hat{y}^{(i)})} - \boldsymbol{x}^{(i)}\right\|^2$$

- This is basically an empirical risk minimization problem

$$\widehat{L}(h|\mathcal{D}) = (1/m)\sum_{i=1}^{m}\left\|\mathbf{x}^{(i)} - \frac{\sum_{i'\in\mathcal{D}^{(i)}}\mathbf{x}^{(i')}}{\left|\mathcal{D}^{(i)}\right|}\right\|^2 \qquad \text{with } \mathcal{D}^{(i)} := \left\{i' : h\big(\mathbf{x}^{(i)}\big) = h\big(\mathbf{x}^{(i')}\big)\right\}$$

Minimizing the clustering error

- Clustering error, to be minimized

$$\mathcal{E}\left(\{m^{(c)}\},\{\hat{y}^{(i)}\}\right) := \frac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{m}^{(\hat{y}^{(i)})} - \boldsymbol{x}^{(i)}\right\|^2$$

- Simultaneously finding cluster means $\boldsymbol{m}^{(c)}$ and assignments $\hat{y}^{(i)}$ that minimize clustering error is difficult (“NP-hard”)

https://cseweb.ucsd.edu/~avattani/papers/kmeans_hardness.pdf

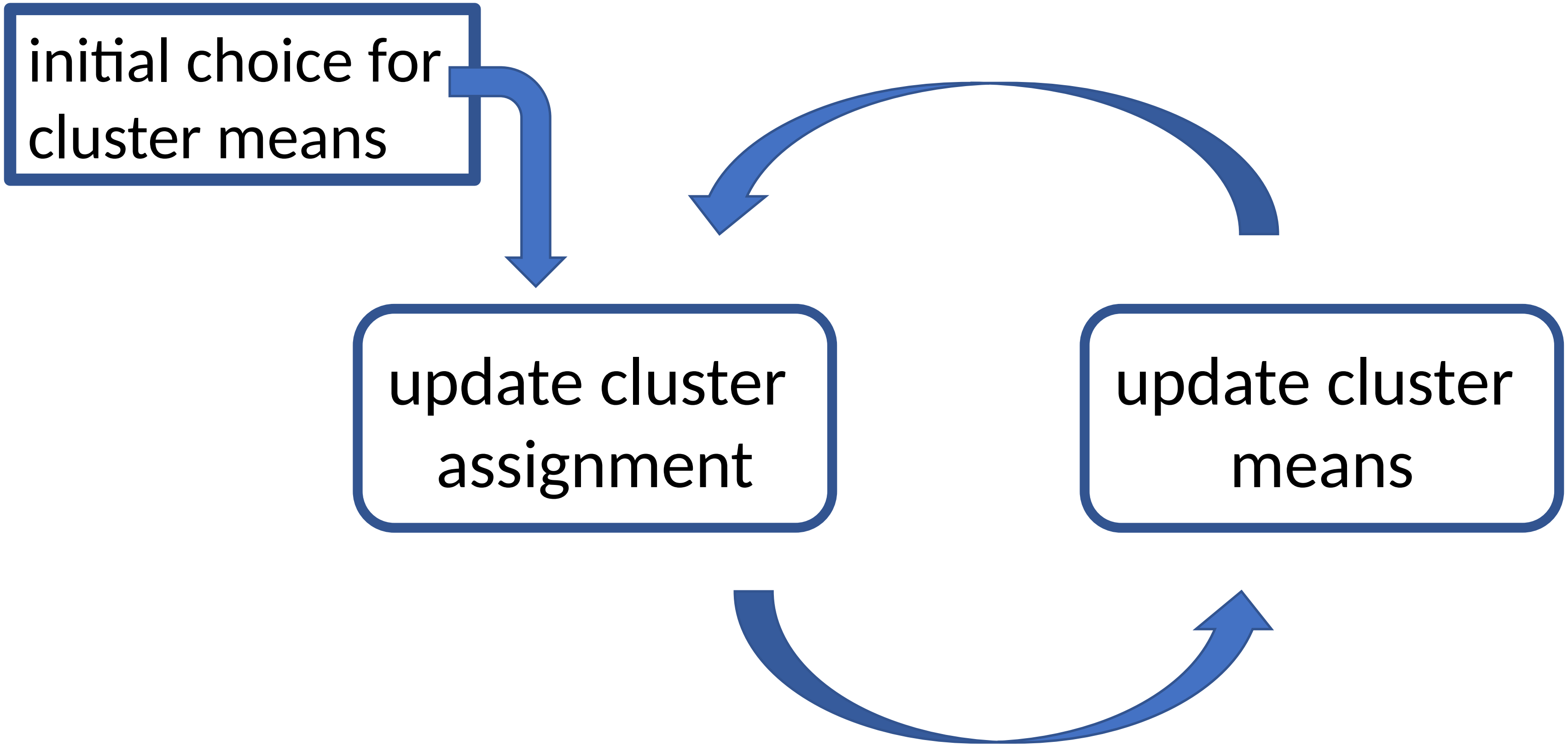
Alternating minimization

- Clustering error, to be minimized

$$\mathcal{E}\left(\left\{m^{(c)}\right\}, \left\{\hat{y}^{(i)}\right\}\right) := \frac{1}{m} \sum_{i=1}^{m} \left\| \boldsymbol{m}^{(\hat{y}^{(i)})} - \boldsymbol{x}^{(i)} \right\|^2$$

- For **given assignments**, finding cluster means that **minimize clustering error is easy**
- For **given cluster means**, finding assignments that **minimize clustering error is easy**

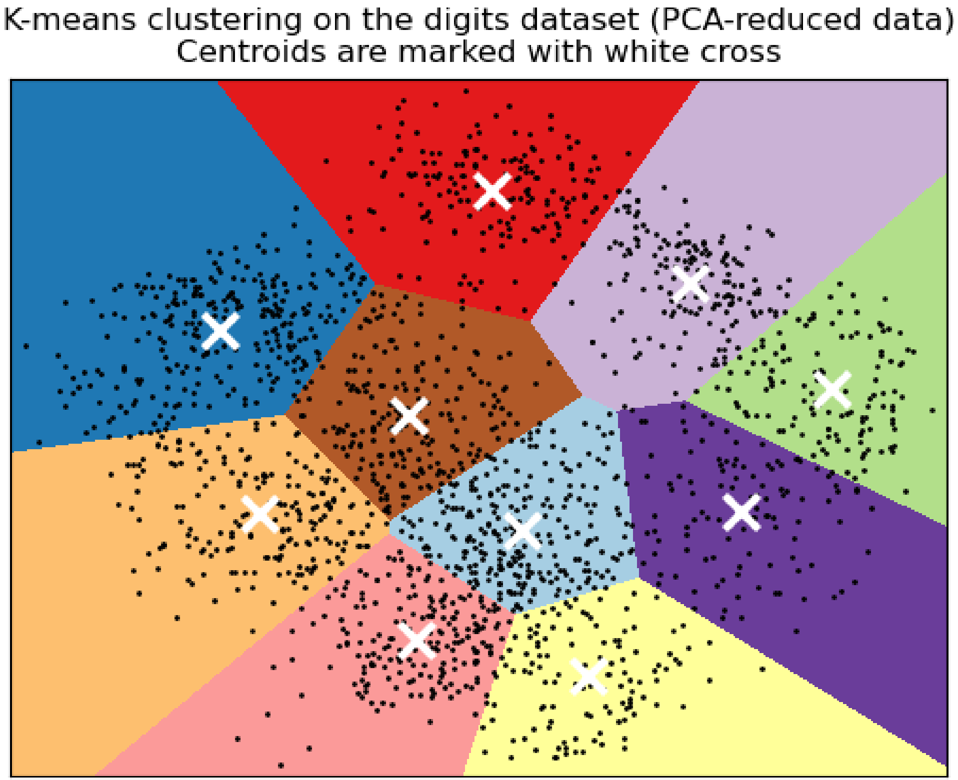
k-Means


initial choice for cluster means
update cluster assignment
update cluster means

k-Means

- **Input:** number k of clusters, initial cluster means
- Step 1: update cluster assignments
- Step 2: update cluster means
- Go to Step 1 unless “Finished”
- **Output**: final cluster means

Cluster shape of k-means



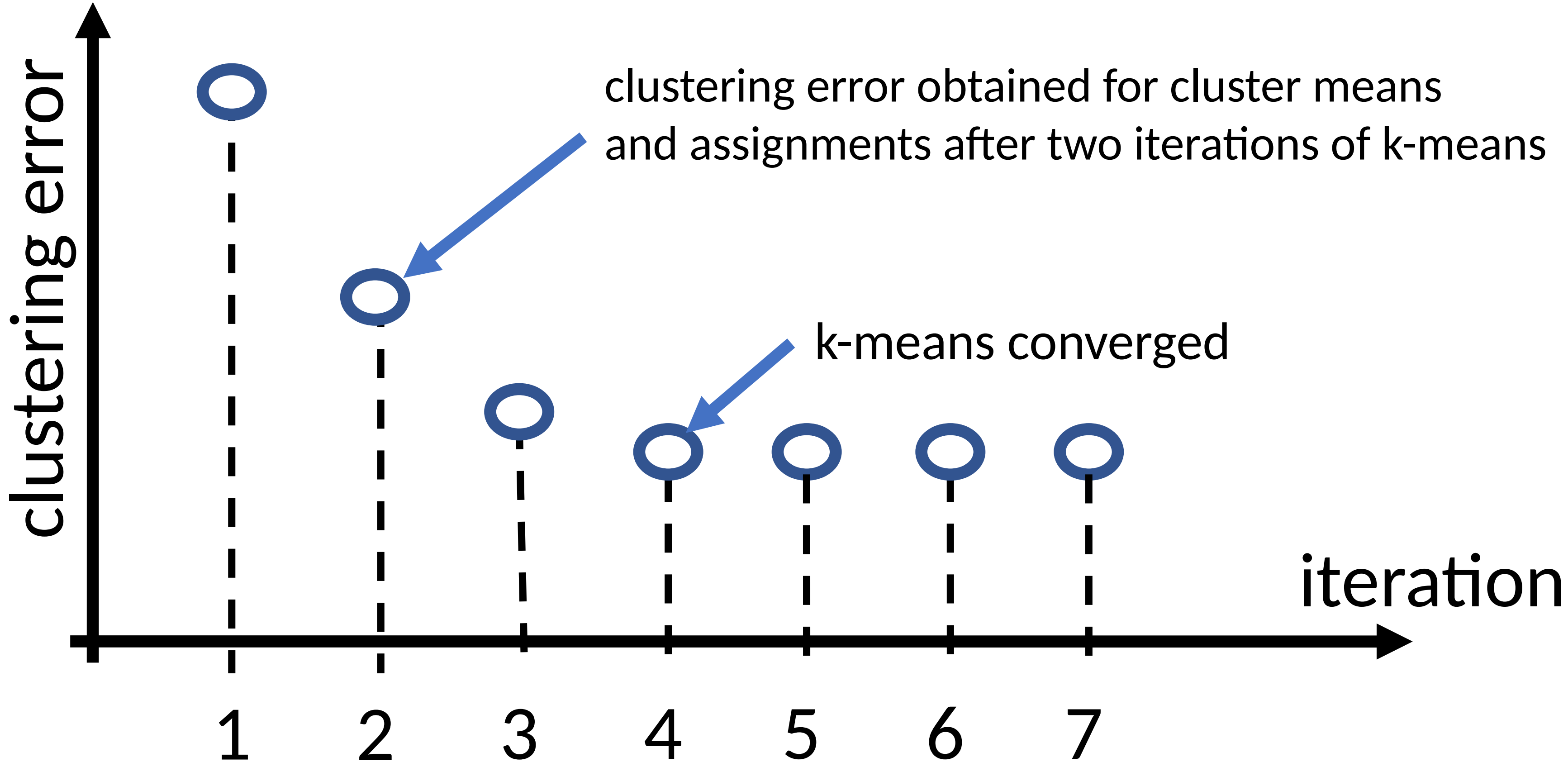
https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html

k-Means never increases clustering error

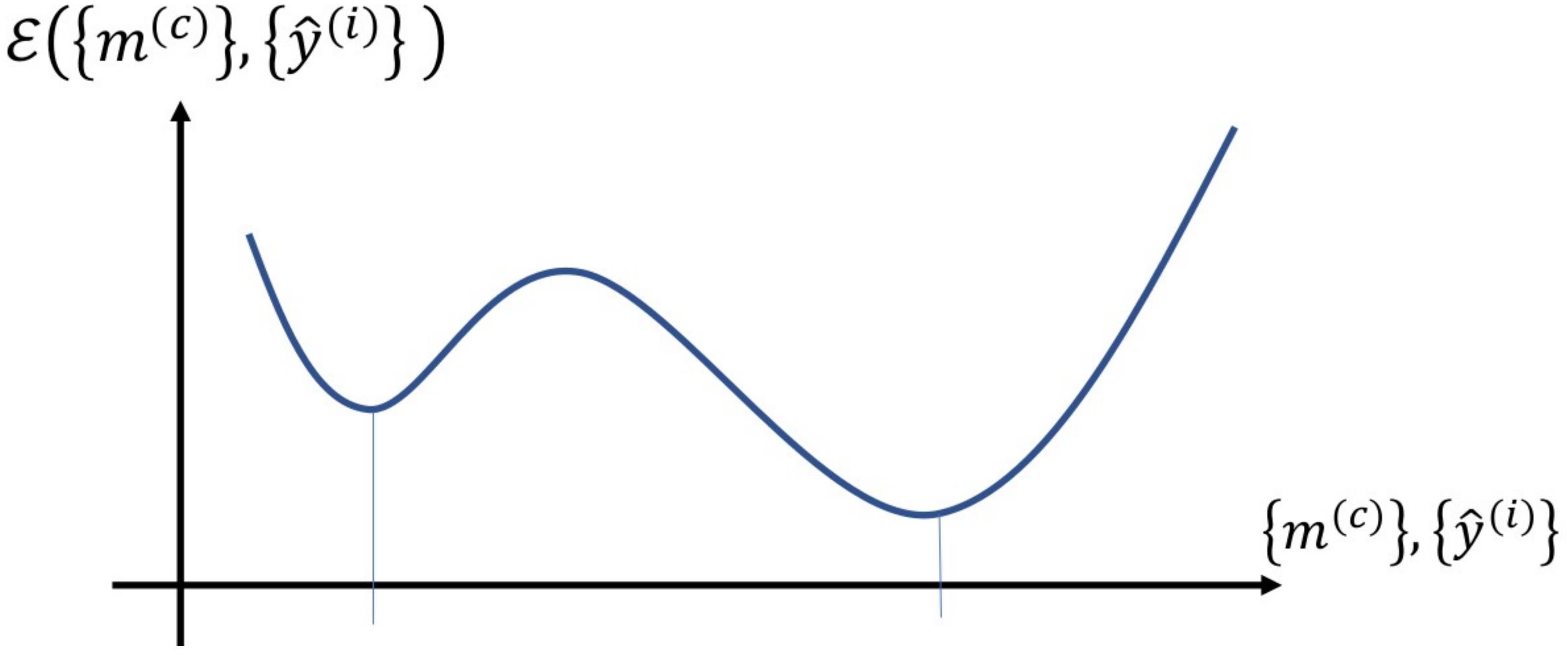
- Consider cluster means $m^{(c)}$ and assignments $\hat{y}^{(i)}$
- Run one iteration of k-Means
- Results in new cluster means $\widetilde{m}^{(c)}$ and assignments $\tilde{y}^{(i)}$

$$\mathcal{E}\left(\{\widetilde{m}^{(c)}\},\{\tilde{y}^{(i)}\}\right) \leq \mathcal{E}\left(\{m^{(c)}\},\{\hat{y}^{(i)}\}\right)$$

k-Means as iterative optimization method



Non-convexity of clustering error



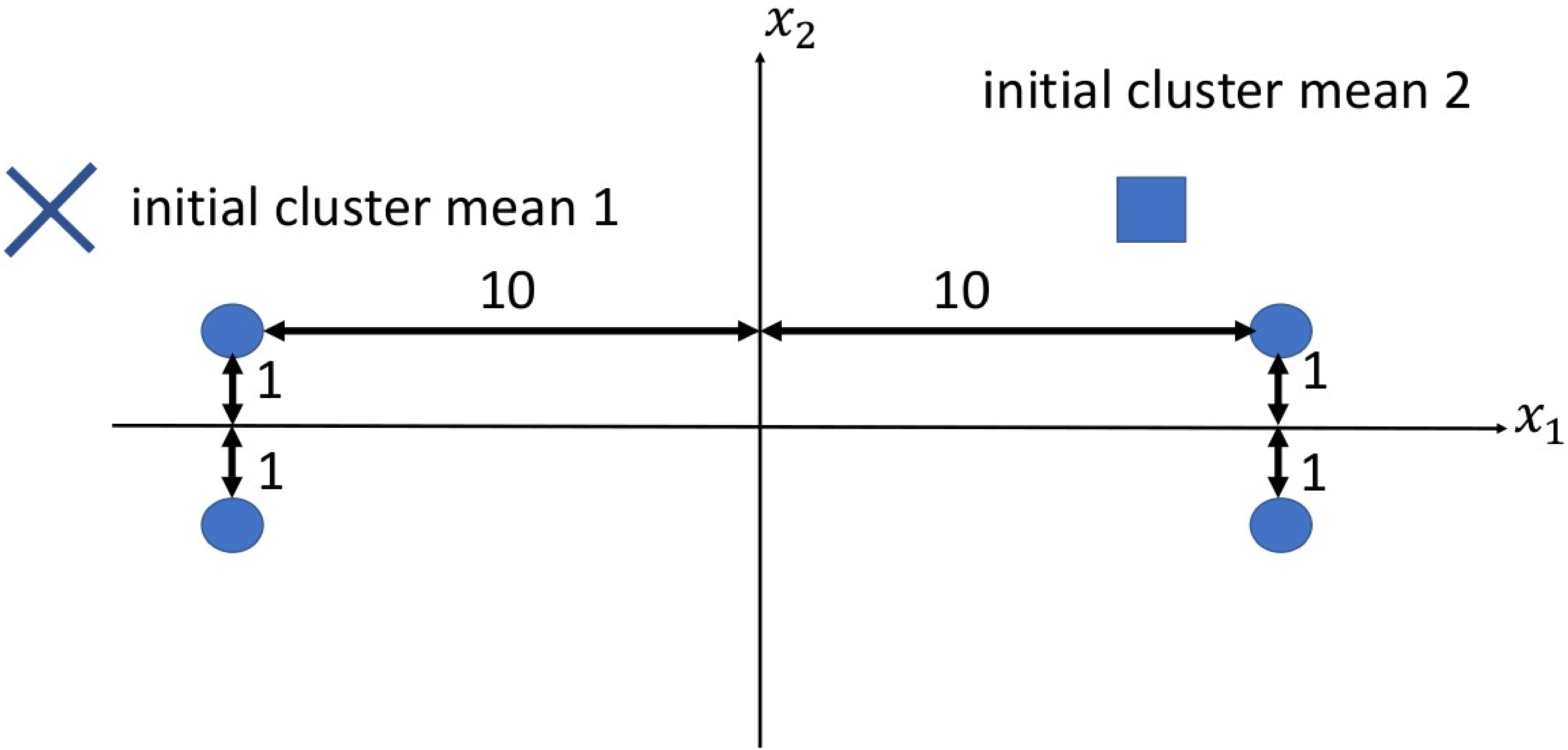
local optimum

optimal clustering

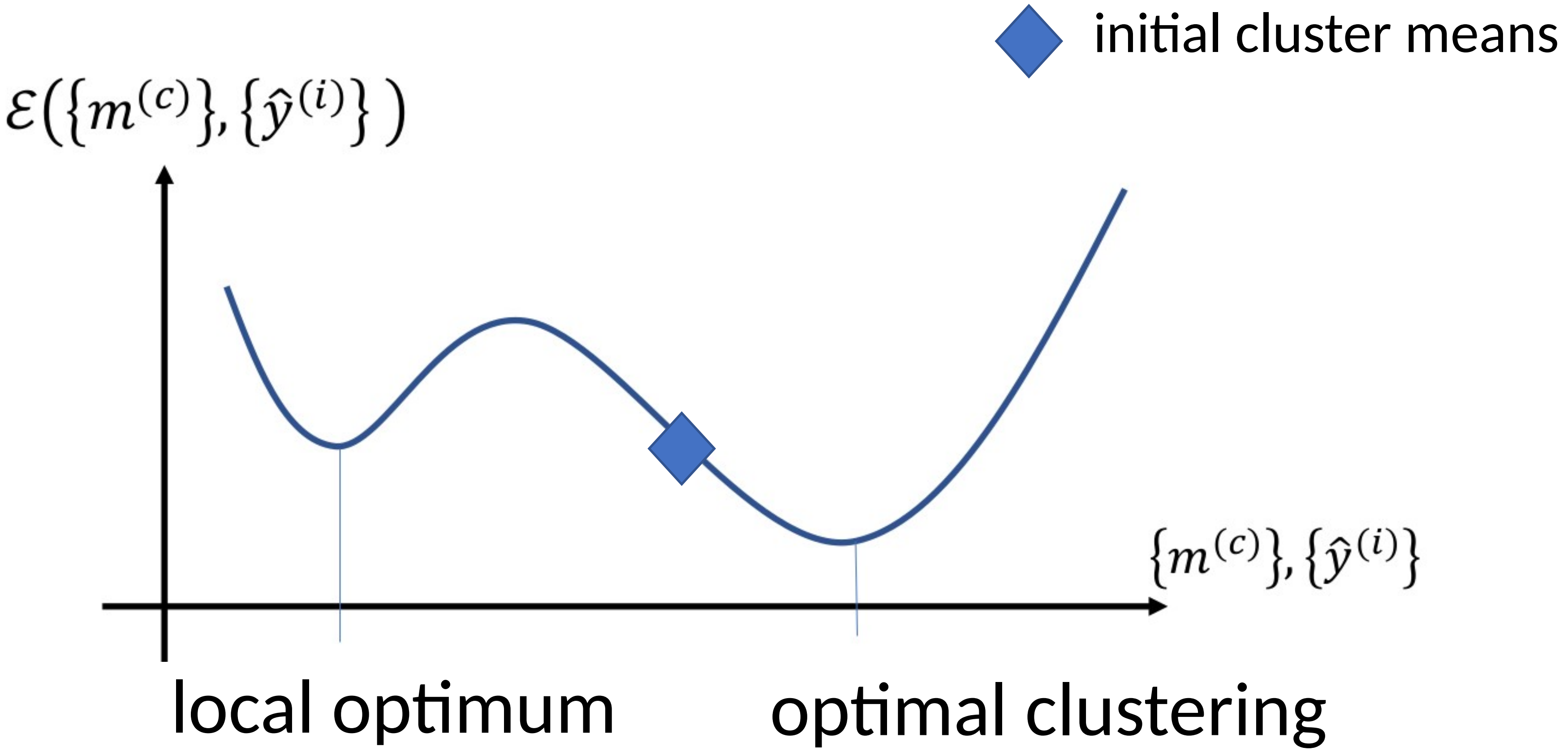
Initialization is crucial

- k-Means requires initial cluster means as inputs
- k-Means result depends crucially on initial means
- **Repeat k-Means several times with different initializations**

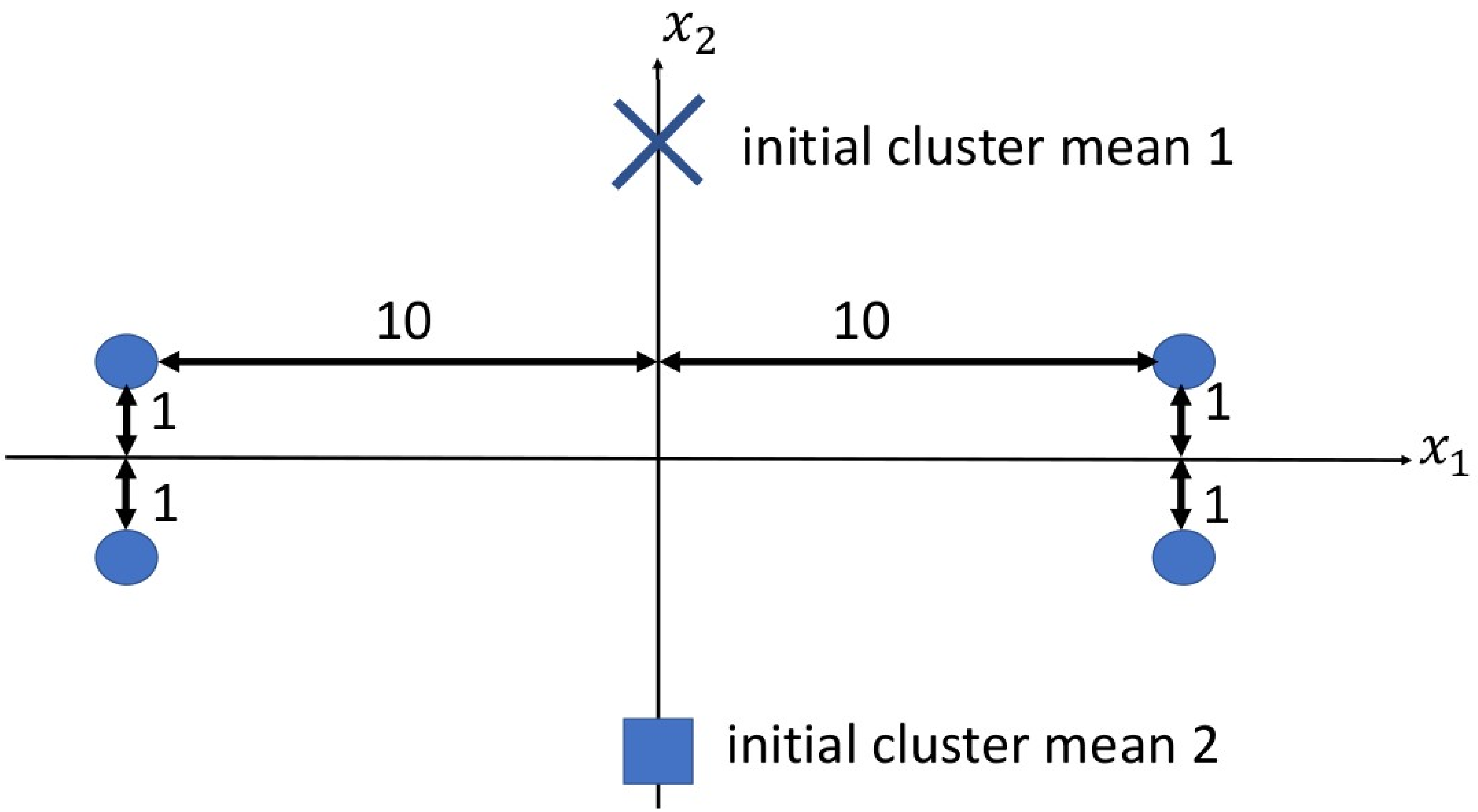
Good initialization



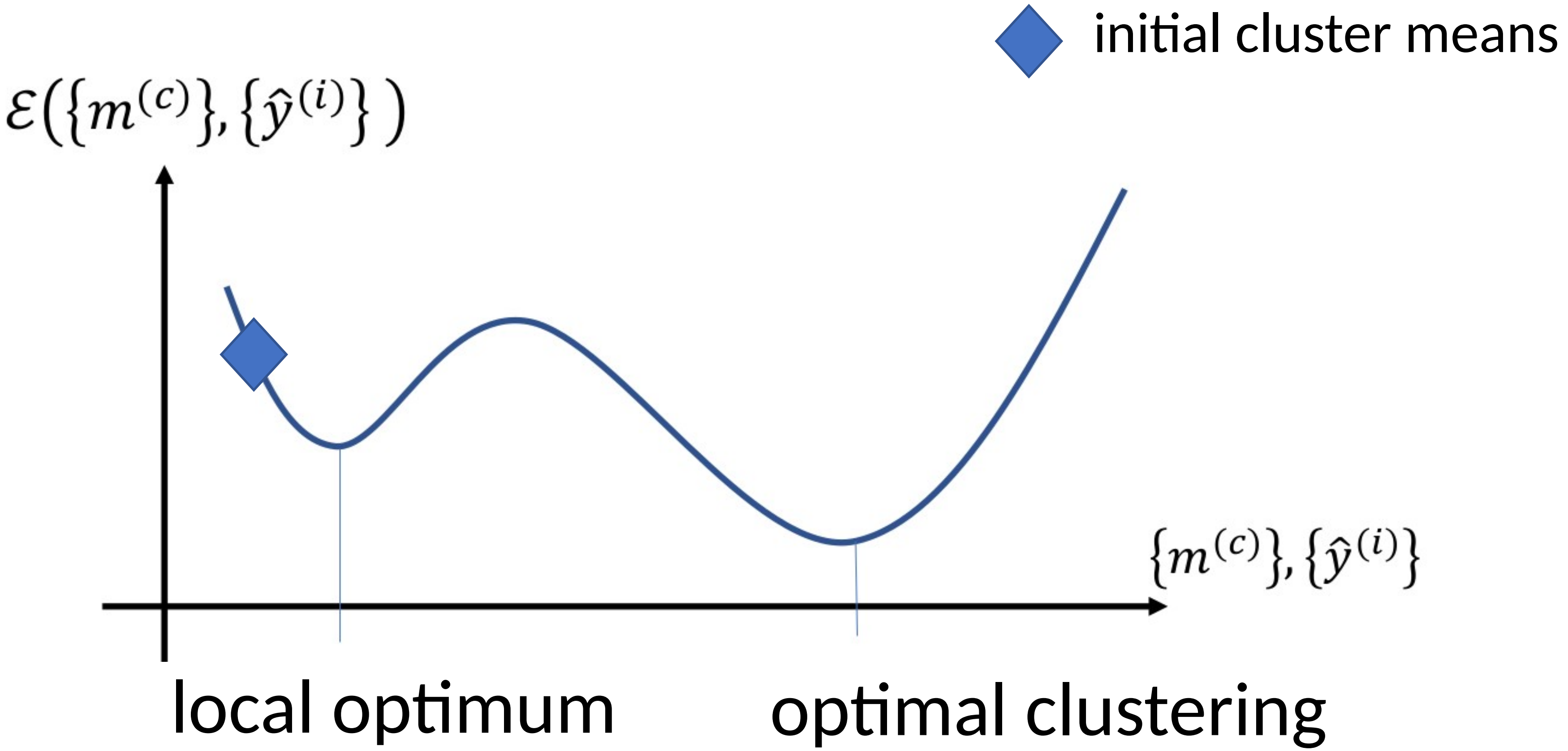
Good initialization



Bad initialization

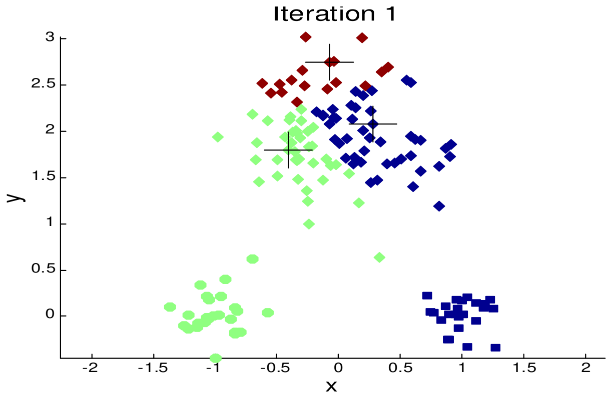


Bad initialization

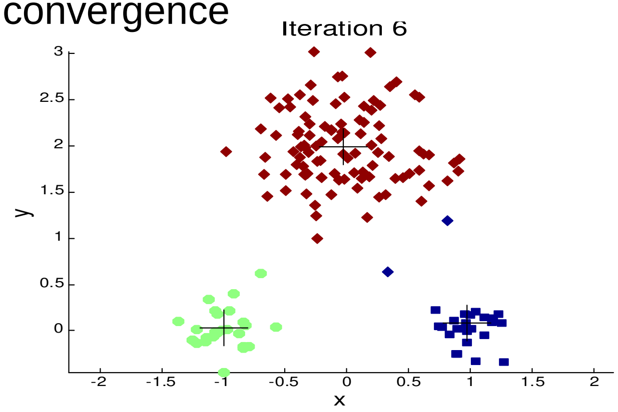


Good vs. bad initializations

- Case 1:

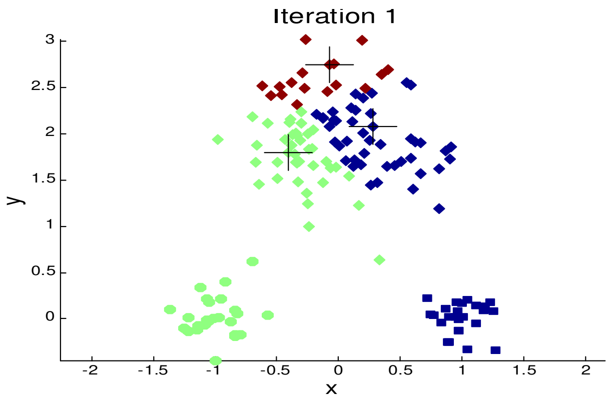


At convergence

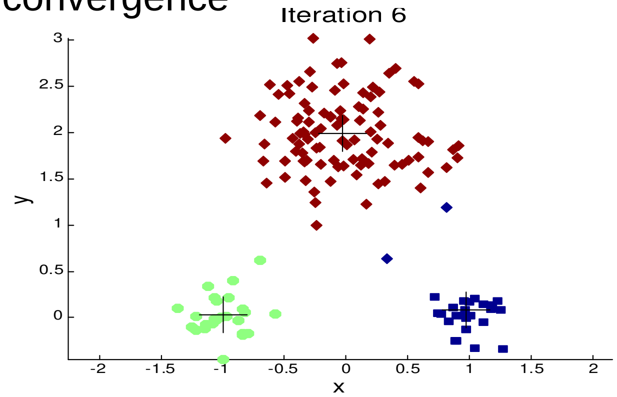


Good vs. bad initializations

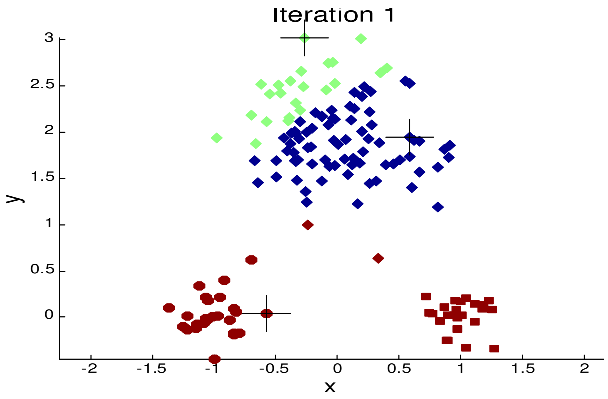
- Case 1:



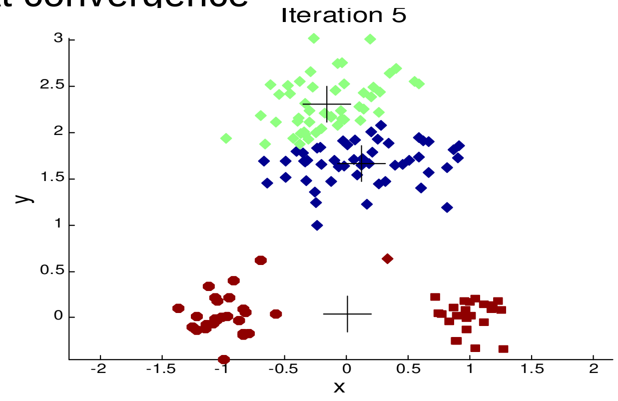
At convergence



- Case 2:



At convergence



How to choose number k of clusters?

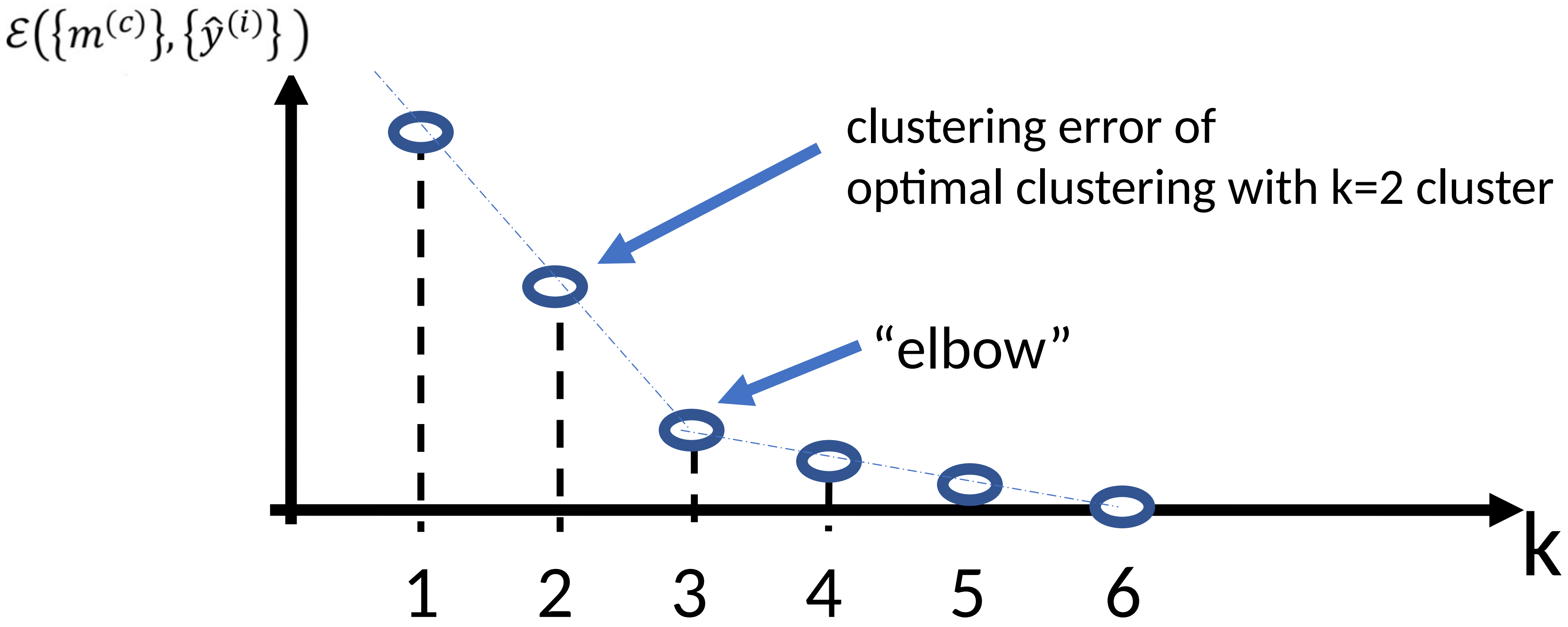
- Increasing k will decrease error (up to 0 when k=m)
- Hence:
  - Defined by application (as in image segmentation)
  - Desired compression rate
  - Elbow (or knee) method
  - Validation error

How to choose number k of clusters?

- For background segmentation k=2
- Cluster 1 = Background, Cluster 2=Foreground

Elbow method

- The gain from adding a cluster becomes negligible

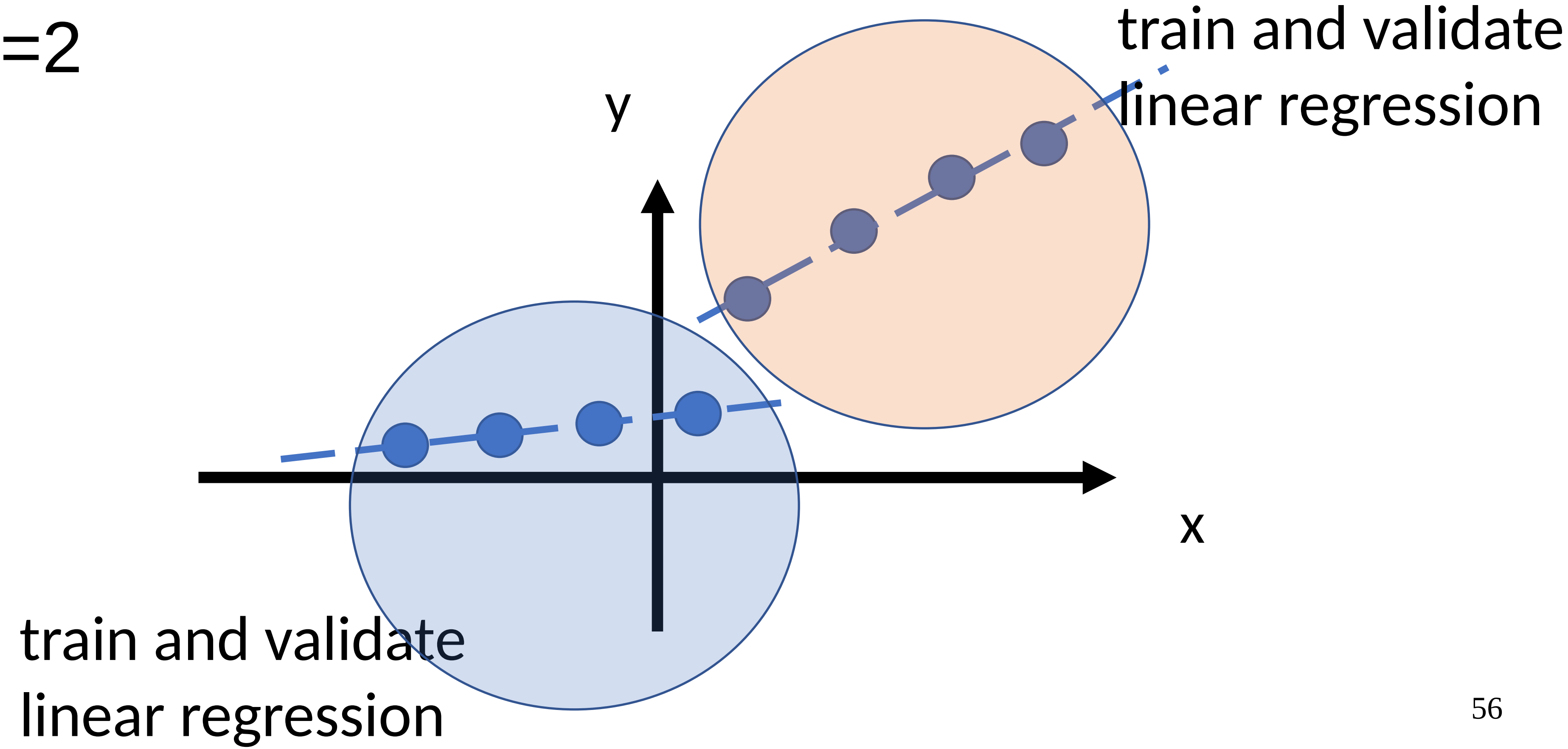


Choose k by validation error

- Clustering an be used as pre-processing for follow-up supervised (regression or classification) problem
- Try different values of k and pick the one resulting in smallest validation error in the supervised problem

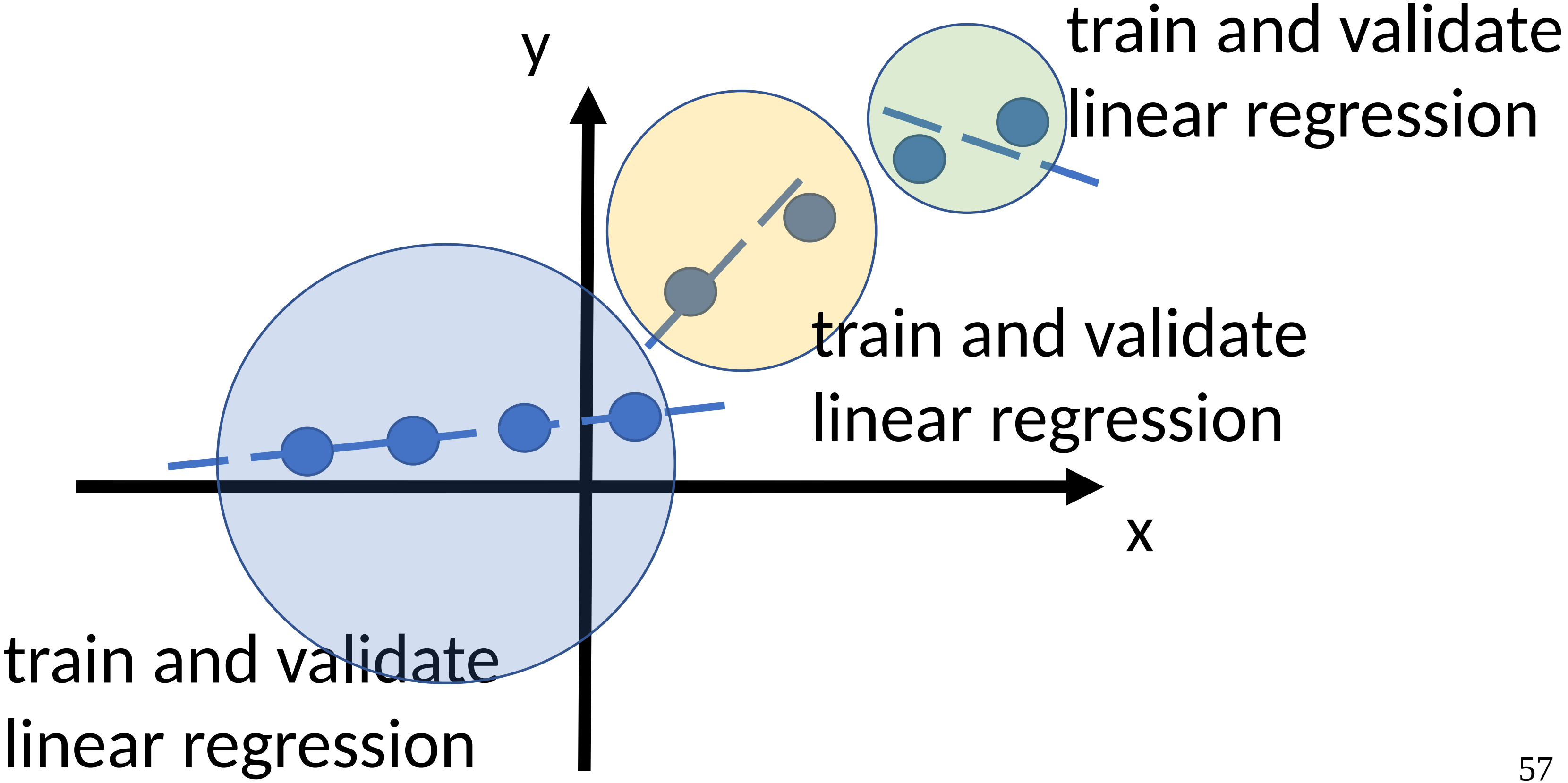
Choose k by validation error

- k=2



Choose k by validation Error

- k=3



k-Means in Python

sklearn.cluster.KMeans

class `sklearn.cluster.KMeans`(*n_clusters=8, *, init='k-means++', n_init='warn', max_iter=300, tol=0.0001, verbose=0, random_state=None, copy_x=True, algorithm='lloyd'*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

K-Means – recap

- k-Means partitions dataset into k clusters
- Number k of clusters needs to be given
- k-Means iteratively minimizes clustering error
- k-Means might deliver sub-optimal clustering: repeat it with different initial cluster means

Soft Clustering

Soft clustering

$$\mathcal{D} = \left\{\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}.$$

Data points characterized by n features

- Features of i-th data

$$\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$$

- i-th data point **characterized by k numerical label values**

$$\boldsymbol{y}^{(i)} = \left(y_1^{(i)}, \ldots, y_k^{(i)}\right)$$

Degree of belonging

- i-th data point **characterized by k numerical label values**

$$\boldsymbol{y}^{(i)} = \left(y_1^{(i)}, \dots, y_k^{(i)}\right)$$

- $y_1^{(i)}$ degree of i-th datapoint belonging to cluster 1
- $y_2^{(i)}$ degree of i-th datapoint belonging to cluster 2
- ...
- $y_k^{(i)}$ degree of i-th datapoint belonging to cluster k

Probabilistic interpretation

- $y_c^{(i)}$ degree of i-th datapoint belonging to cluster c
- Interpret $y_c^{(i)}$ as **probability**:

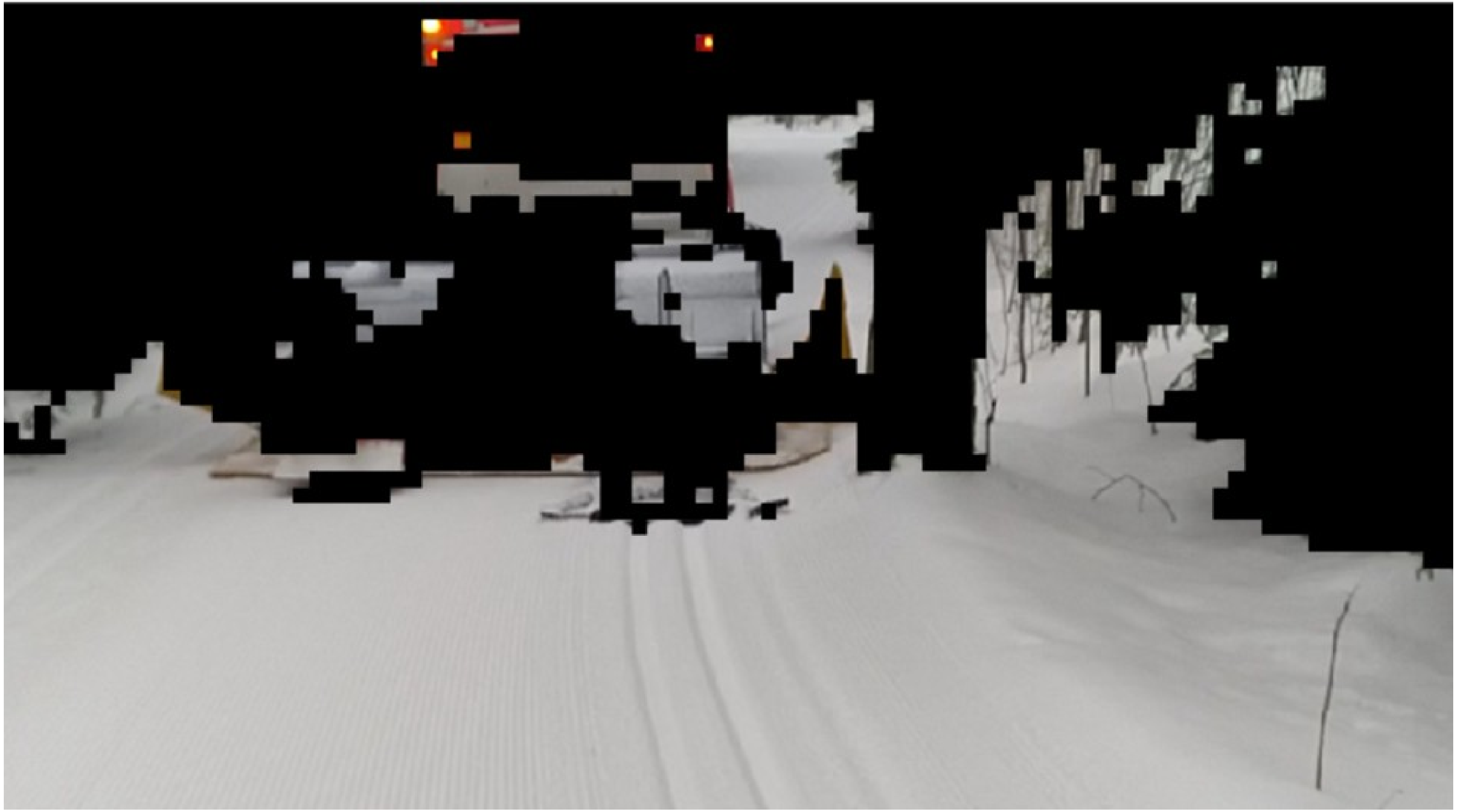
  p("i-th datapoint belongs to cluster c")
- $y_c^{(i)}$ can be any number between 0 and 1
- i-th datapoint must belong to some cluster $\sum_{c=1}^{k} y_c^{(i)} = 1$
- Hard clustering requires $y_c^{(i)}$ is either 0 or 1

Soft clustering methods

Dataset = Set of image patches

Output of Hard Clustering (k-Means)

k=2



Output of Soft Clustering (GMM)

k=2

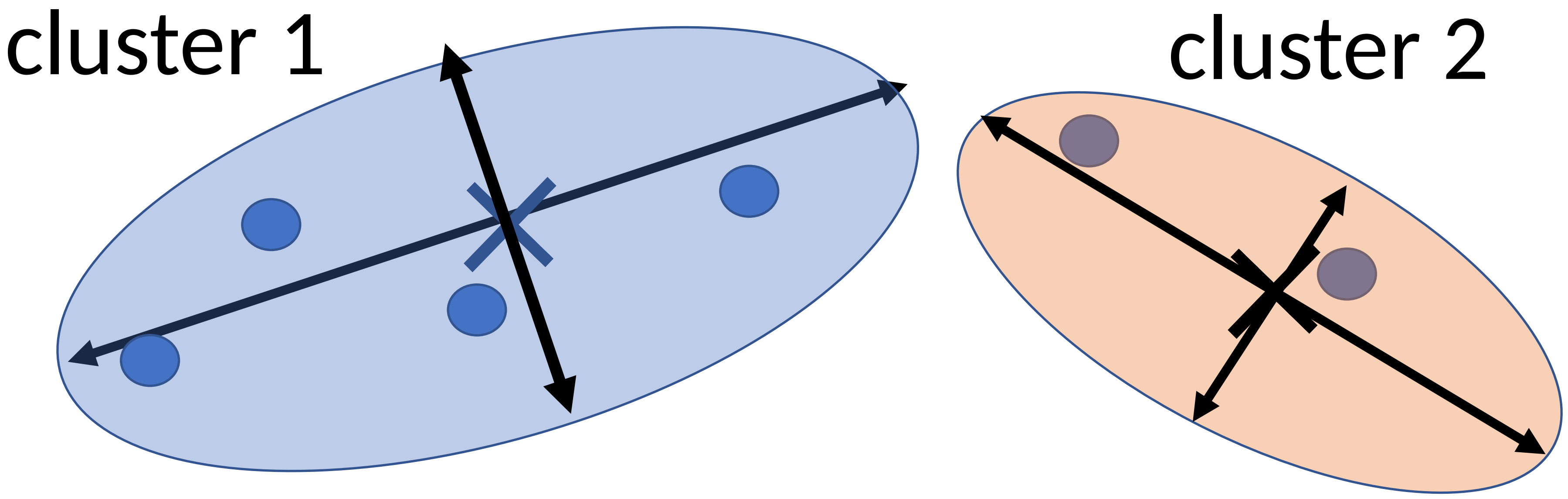
Soft Clustering with GMM

Gaussian mixture model

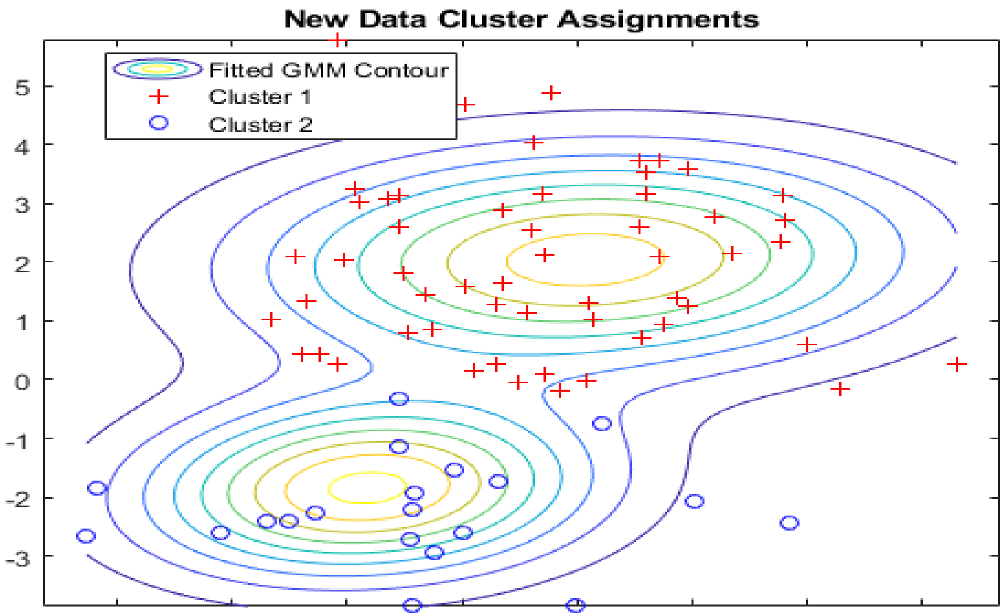
- Each cluster produces data based upon random draws from a (multi-dimensional) Gaussian distribution
- Clusters should be less likely to have data at the edge
- Each Gaussian cluster has its own mean and standard deviation

Represent clusters by Gaussians

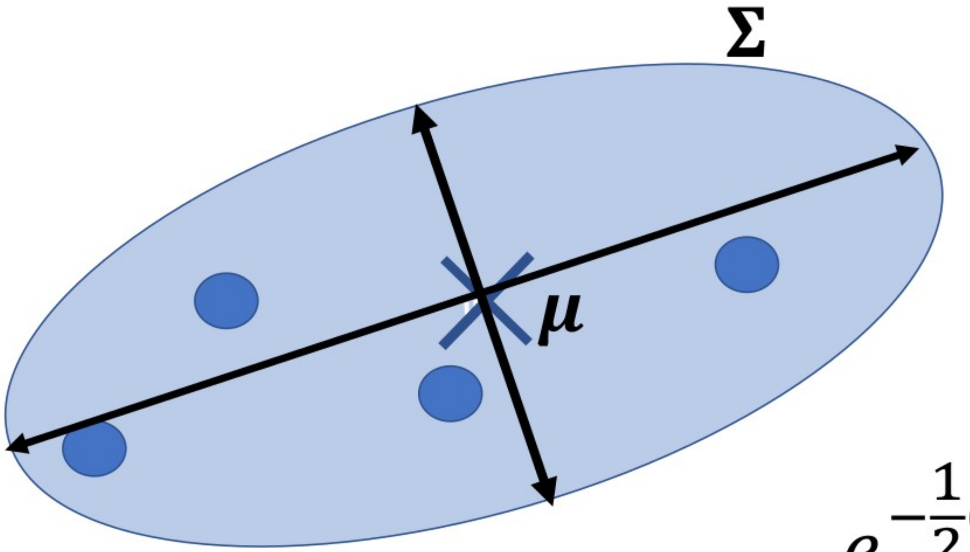
- Gaussian mixture model (GMM)



Sum of the Gaussians: GMM



Gaussian distribution

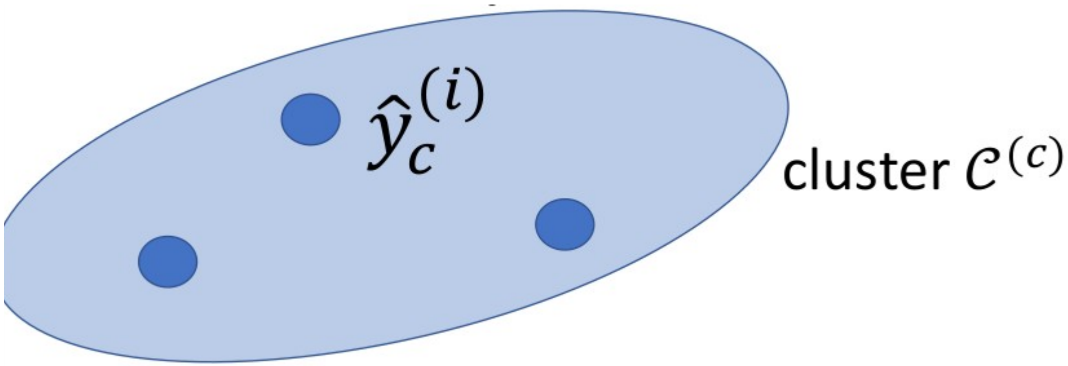


mean vector $\boldsymbol{\mu}$

covariance matrix $\boldsymbol{\Sigma}$

$$p(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma})}}$$

Cluster spread



$$\frac{1}{m^{(c)}} \sum_{i=1}^{m} \hat{y}_c^{(i)} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c)}\right)^T \left(\boldsymbol{\Sigma}^{(1)}\right)^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c)}\right)$$

Effective cluster size:

$$m^{(c)} := \sum_{i=1}^{m} \widehat{y}_c^{(i)}$$

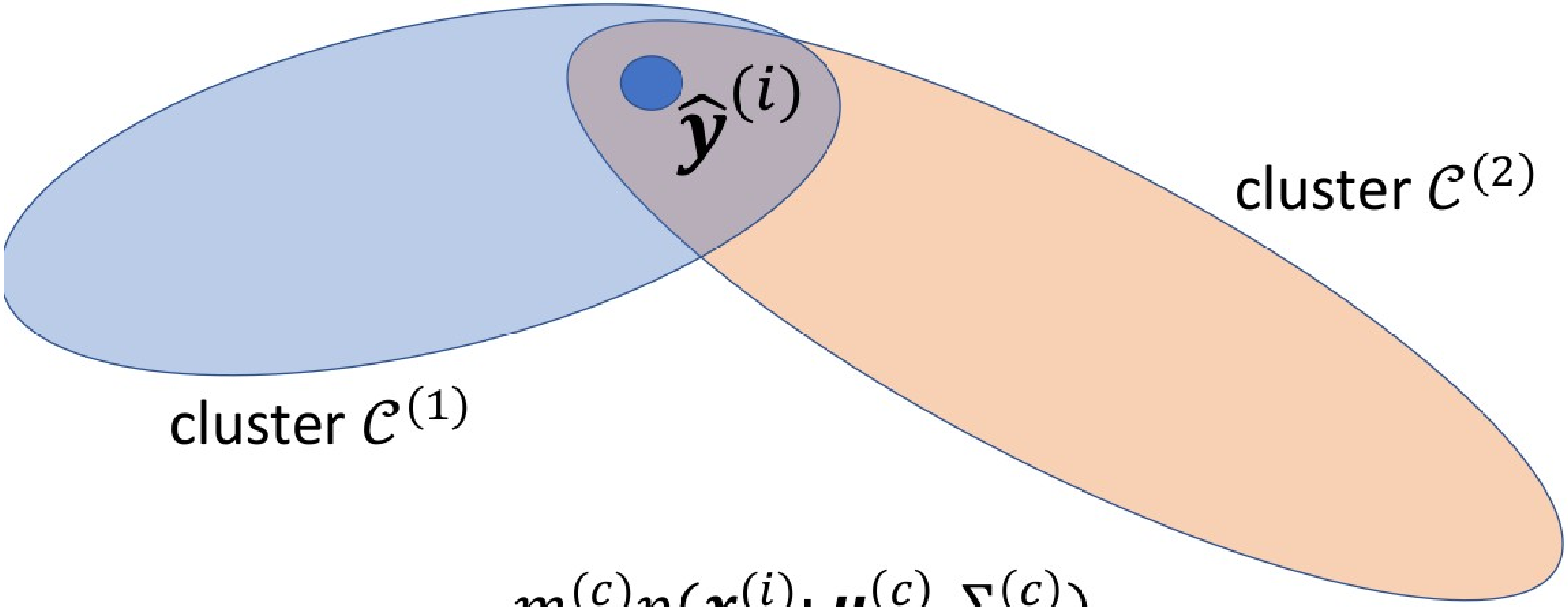
Update cluster mean and covariance

- For given (soft) cluster assignments chose cluster **means and covariance** to **minimize cluster spreads**

$$\boldsymbol{\mu}^{(c)} := \frac{1}{m^{(c)}} \sum_{i=1}^{m} \hat{y}_c^{(i)}\, \boldsymbol{x}^{(i)} \qquad \text{for all c =1,…,k}$$

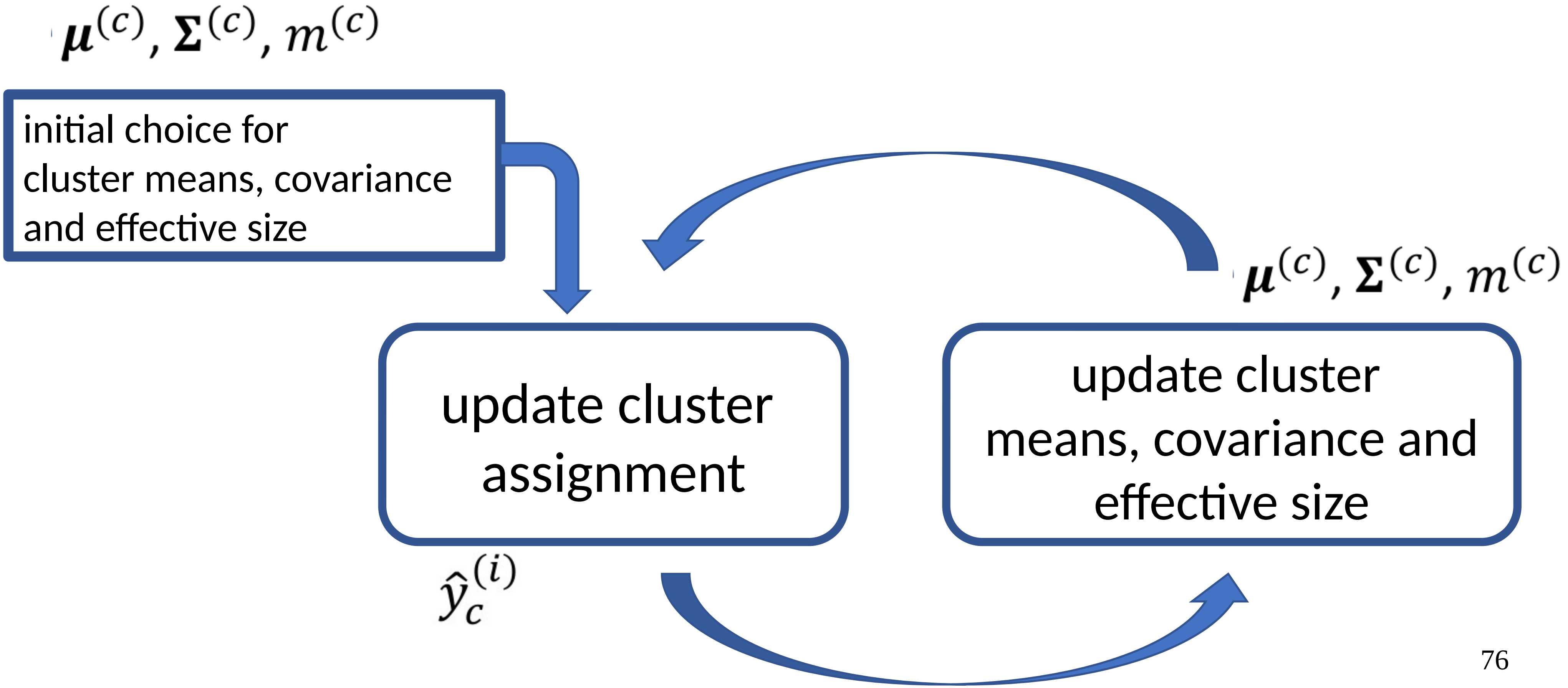
$$\boldsymbol{\Sigma}^{(c)} := \frac{1}{m^{(c)}} \sum_{i=1}^{m} \hat{y}_c^{(i)} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c)}\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c)}\right)^T$$

Cluster assignment update



$$\hat{y}_c^{(i)} := \frac{m^{(c)} p(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}^{(c)}, \Sigma^{(c)})}{\sum_{c'=1}^{k} m^{(c')} p(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}^{(c')}, \Sigma^{(c')})}$$

GMM Algorithm



GMM Algorithm

- **Input**: $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}, k, \{\boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}, m^{(c)}\}$
  1. Update soft cluster assignments $\hat{y}_c^{(i)}$
  2. Update cluster params $\boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}, m^{(c)}$
  3. Go to 1. unless "finished"
- **Output**: $\hat{y}_c^{(i)}, \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}, m^{(c)}$

Falls in the category of expectation–maximization (EM) algorithms (as k-means)

https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

Soft-Clustering Error

$$\mathcal{E}\left(\{\boldsymbol{\mu}^{(c)}\},\{\boldsymbol{\Sigma}^{(c)}\},\{m^{(c)}\}\ \right) :=$$
$$-\sum_{i=1}^{m} \log \sum_{c=1}^{k} \frac{m^{(c)}}{m} p\left(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}^{(c)}, \boldsymbol{\Sigma}^{(c)}\right)$$

- This is negative logarithm of probability to sample data points under Gaussian mixture model
- Minimize error equal to maximize (log) likelihood

Soft-Clustering Error

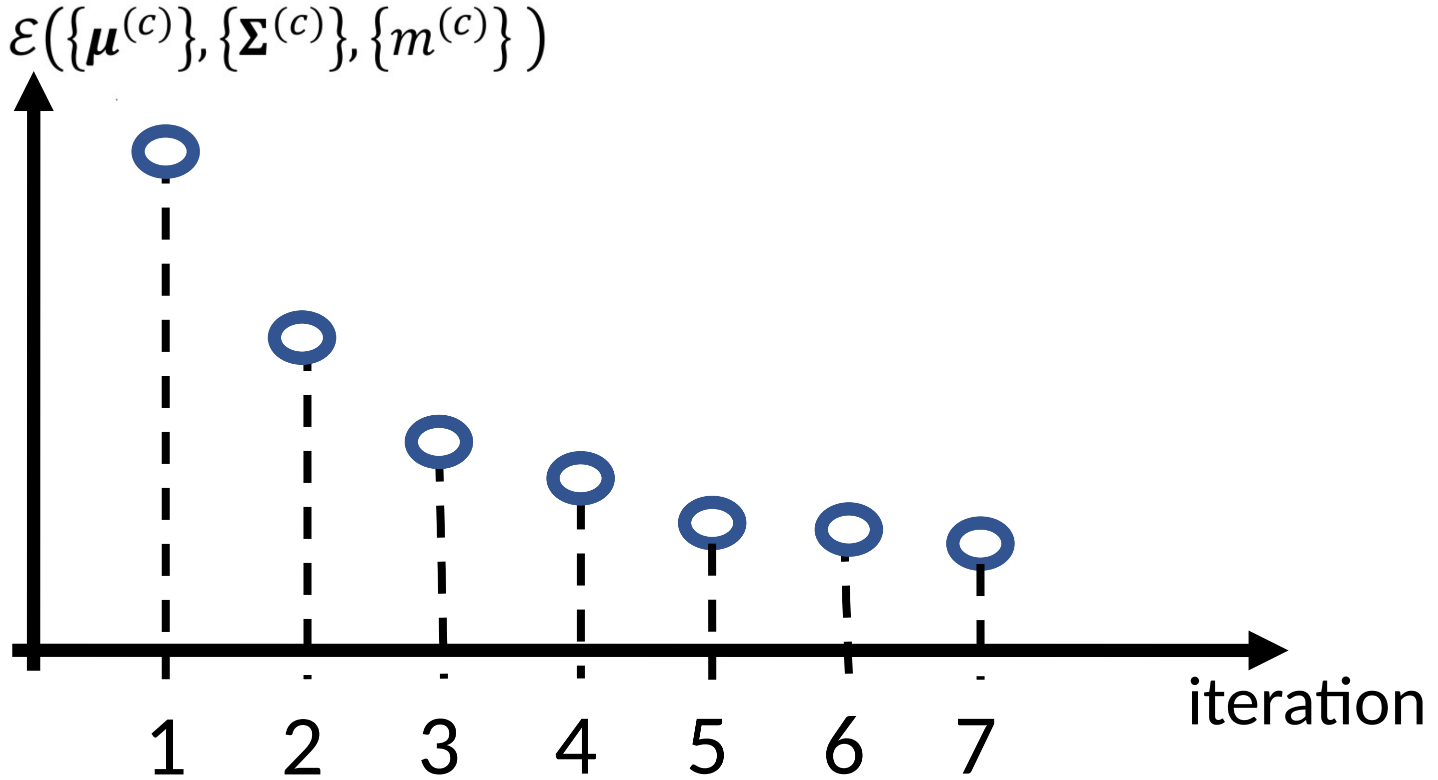
$$\mathcal{E}\Big(\{\boldsymbol{\mu}^{(c)}\},\{\boldsymbol{\Sigma}^{(c)}\},\{m^{(c)}\}\Big) :=$$
$$-\sum_{i=1}^{m} \log \sum_{c=1}^{k} \frac{m^{(c)}}{m} p\Big(\boldsymbol{x}^{(i)};\boldsymbol{\mu}^{(c)},\boldsymbol{\Sigma}^{(c)}\Big)$$

- Again, this is an Empirical Risk Minimization problem:

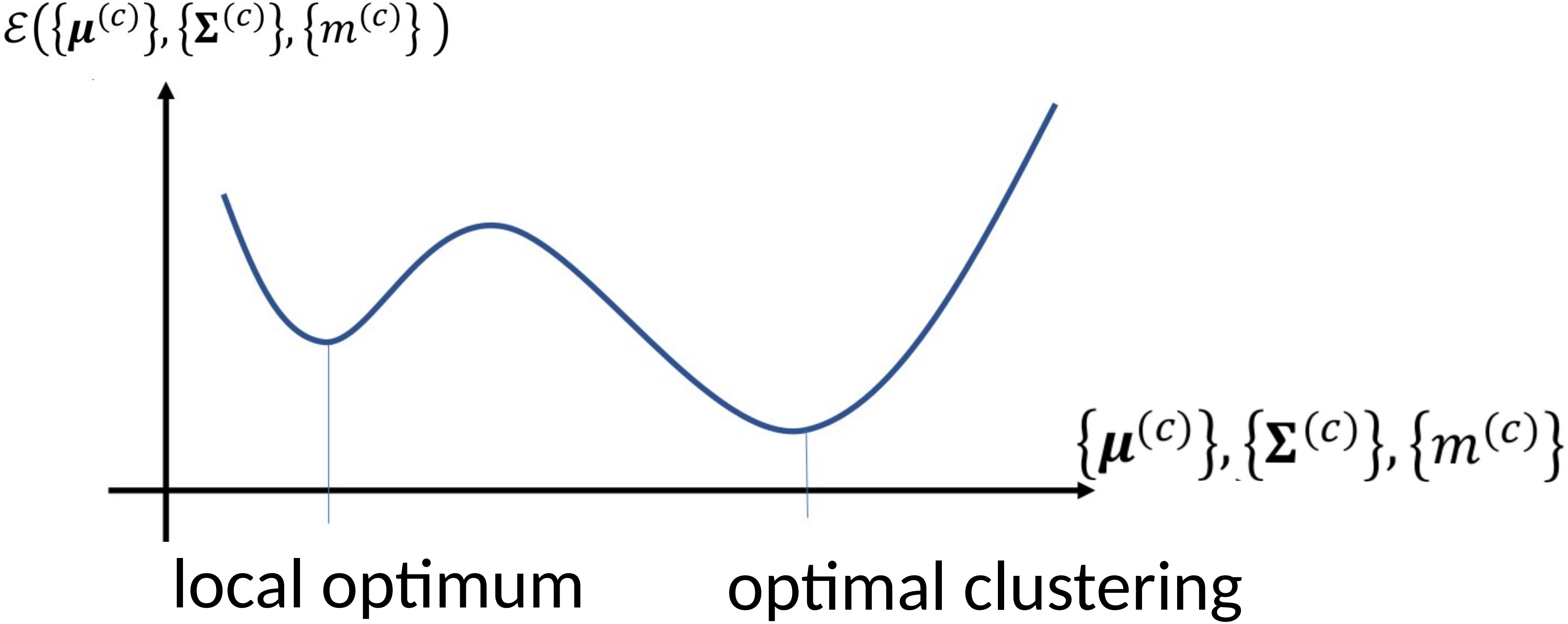
$$\widehat{L}(\boldsymbol{\theta} \mid \mathcal{D}) := -\log p\big(\mathcal{D};\boldsymbol{\theta}\big) \text{ with GMM parameters } \boldsymbol{\theta} := \{\boldsymbol{\mu}^{(c)},\boldsymbol{\Sigma}^{(c)},p_c\}_{c=1}^{k}$$

When to stop?

- Stop when decrease too small
- After fixed number of iteration



Non-convexity of soft-clustering error



Initialization is crucial

- Soft clustering depends crucially on initialization means
- Repeat several times with different initializations

How to choose number k of clusters?

- As in k-Means:
  - Defined by application
  - Desired compression rate
  - Elbow (knee) method
  - Validation error

GMM in Python

2.1. Gaussian mixture models

`sklearn.mixture` is a package which enables one to learn Gaussian Mixture Models (diagonal, spherical, tied and full c
matrices supported), sample them, and estimate them from data. Facilities to help determine the appropriate number o
nents are also provided.

https://scikit-learn.org/stable/modules/mixture.html#gmm

sklearn.mixture.GaussianMixture

class `sklearn.mixture.`**`GaussianMixture`**(*n_components=1, *, covariance_type='full', tol=0.001, reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans', weights_init=None, means_init=None, precisions_init=None, random_state=None, warm_start=False, verbose=0, verbose_interval=10*) [source]

https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture

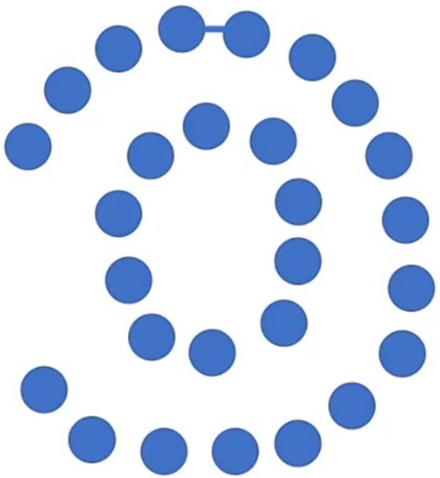
GMM – recap

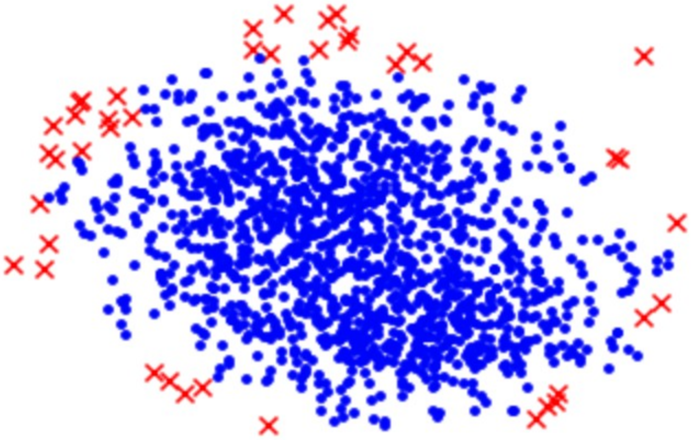
- Represent clusters by Gaussian distributions
- Soft clustering algorithm fits GMM
- Iterative optimization of soft-clustering error
- Trapped in local minimum for bad initialization

Other clustering approaches

Connectivity based clustering

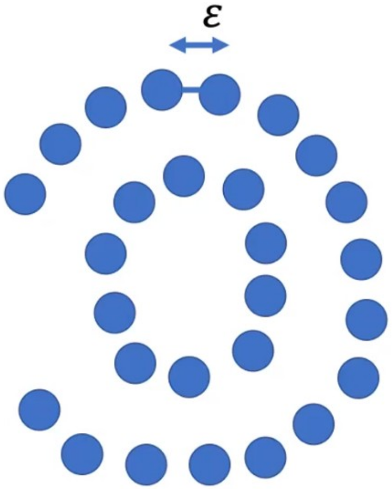
- k-Means and GMM fails on non-Euclidean cluster structure
- k-Means and GMM cannot recognize outliers/noise



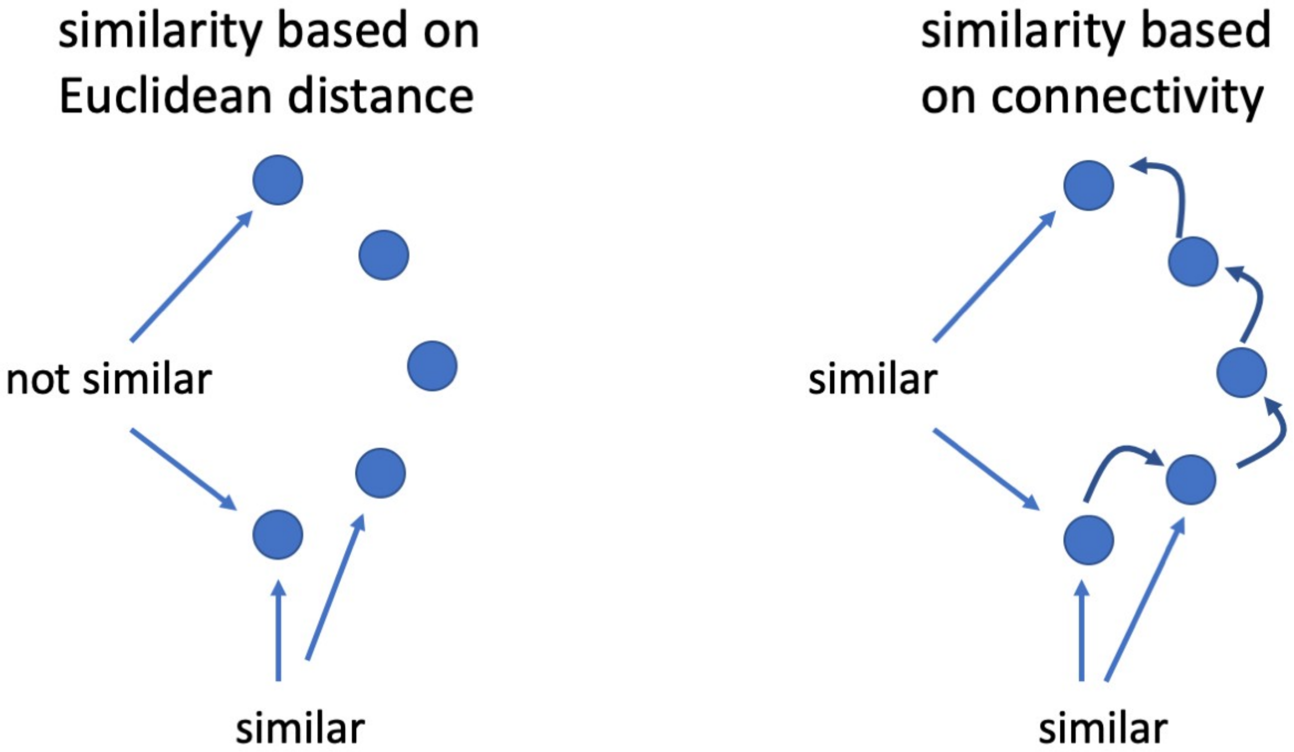


Connectivity based clustering

- Notions of connectivity between nodes
- Connect close-by data points obtaining an **empirical graph**
- Cluster ≈ **connected graph component**



Connectivity based clustering



E.g.: K-means,
Gaussian mixture models,...

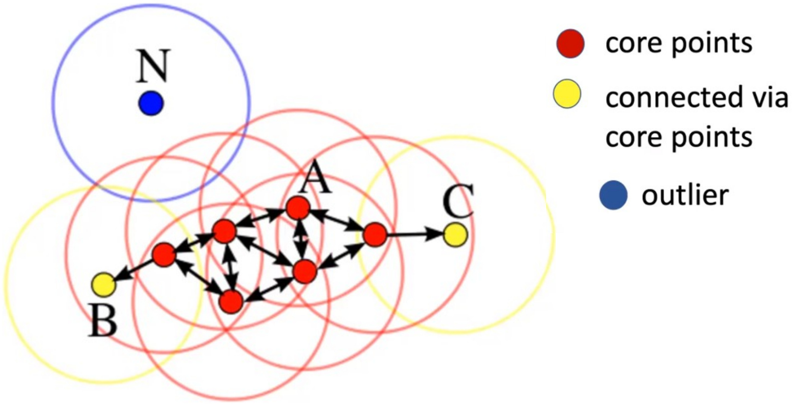
E.g.: DBSCAN

Connectivity based clustering

- Hard clustering
  - Spectral clustering
    - Eigenvectors of graph Laplacian matrix to measure connectivity between nodes
  - DBSCAN
    - Density based spatial clustering with noise

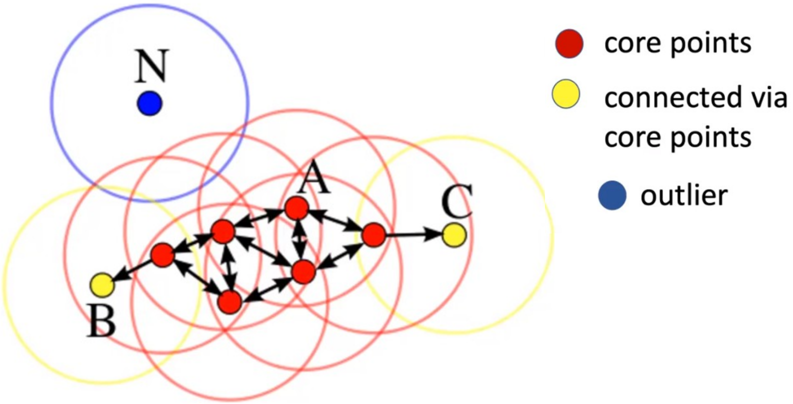
DBSCAN

- Data points need to be connected via **core points**
- Core points are the ones with a minimum number of neighbors
- Automatically determines number of k clusters



DBSCAN

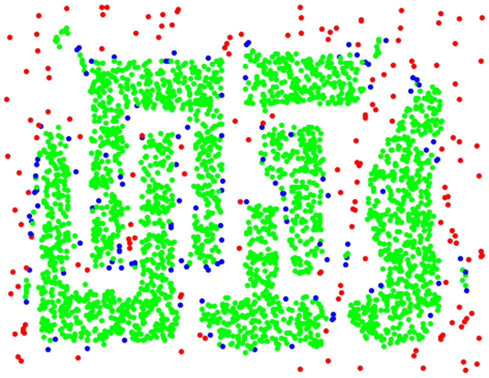
- Parameters:
  - Epsilon: Maximum distance to be connected
  - MinPts: Minimum number of points to be a core point



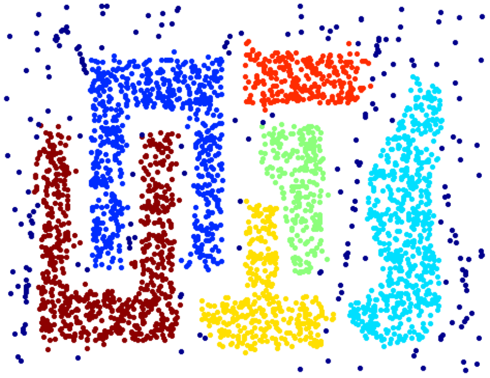
DBSCAN

Eps = 10, MinPts = 4

Original points



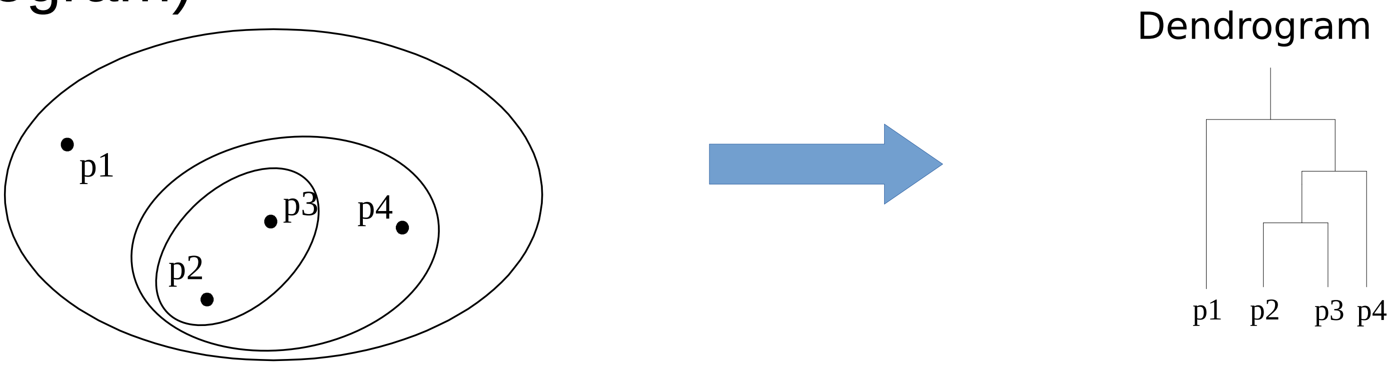
Core, border and outlier/noise

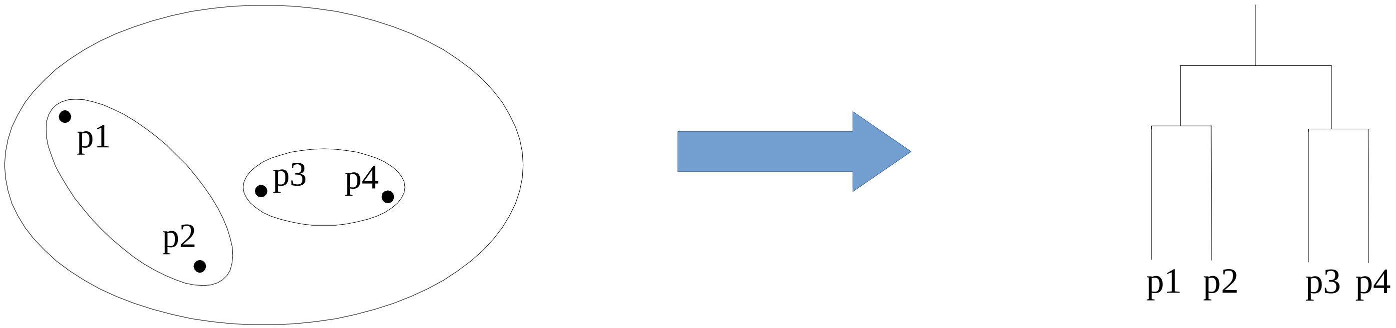


Clusters

Hierarchical clustering

- A set of nested clusters organized as a hierarchical tree (dendogram)

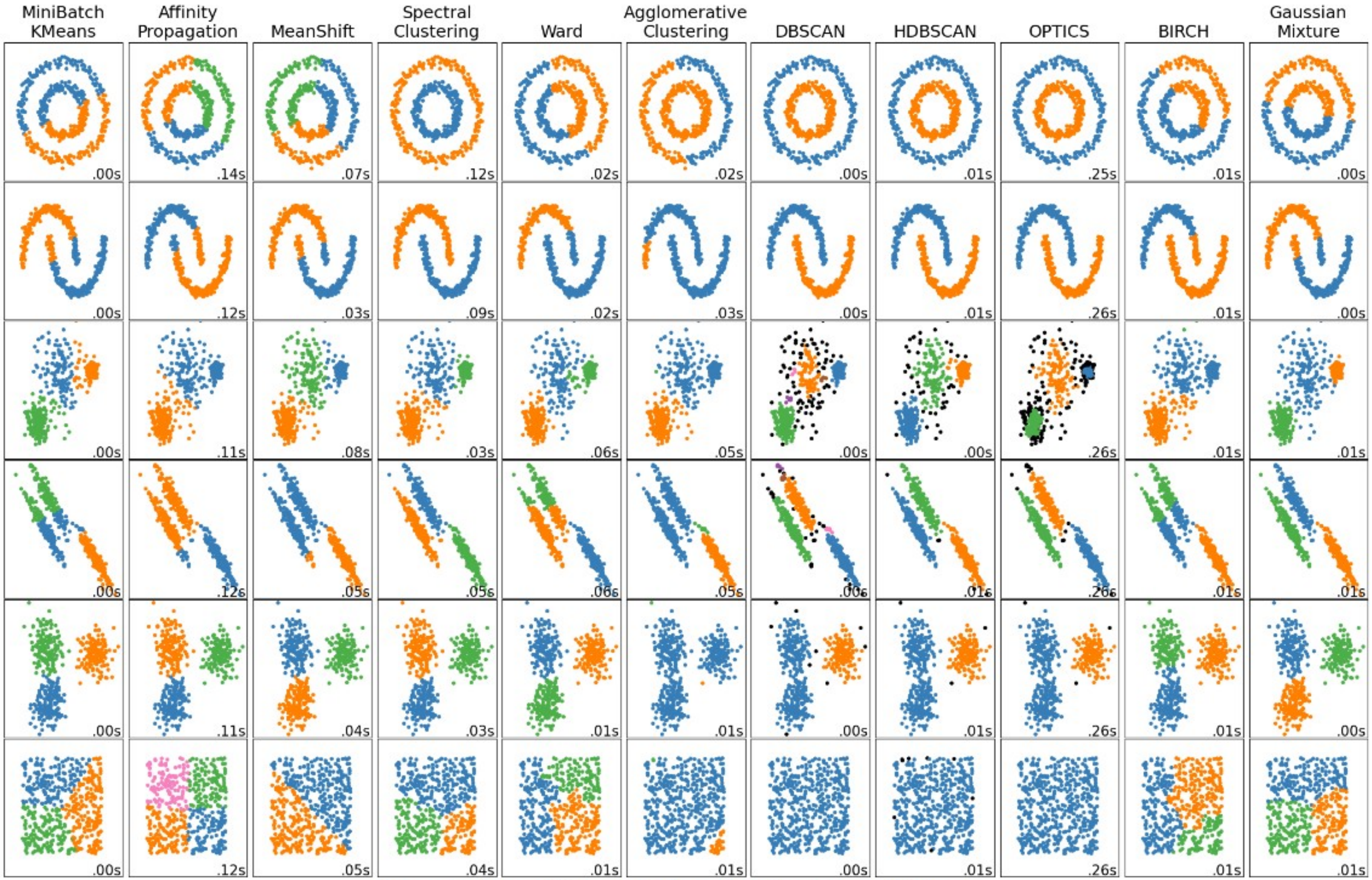




https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

Hierarchical clustering

- A set of nested clusters organized as a hierarchical tree (dendogram)
- Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- Different levels may correspond to meaningful taxonomies
- Key operation is the computation of the proximity of two clusters



A comparison of the clustering algorithms in scikit-learn

https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods

Evaluating/comparing clustering

https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation

Measures of cluster validity

- **Internal Index**: goodness of a clustering structure without external information
  - e.g., Silhouette index, sum of squared error (k-Means), log-likelihood (GMM).
- **External or Relative Index**: extent to which cluster labels match externally supplied class labels or labels from another clustering
  - e.g., entropy, purity, rand-index, adjusted rand-index, mutual information, adjusted mutual information.

https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation

Silhouette

- Silhouette measures consistency within clusters of data: how similar a data point is to its own cluster (cohesion) compared to other clusters (separation)
- The Silhouette score is defined for each sample and is composed of two scores:
  - The mean distance between a sample and all other points in the same cluster (a)
  - The mean distance between a sample and all other points in the next nearest cluster (b)

$$s = \frac{b - a}{max(a, b)}$$

- The Silhouette for a set of samples is given as the mean of the Silhouette for each sample

Silhouette

$$s = \frac{b - a}{max(a, b)}$$

- It ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters
- Scores around zero indicate overlapping clusters
- If most objects have a high value, then the clustering configuration is appropriate
- The silhouette can be calculated with any distance metric
- The **average silhouette over all data of a cluster** measures how tightly grouped all the data in the cluster are
- The **average silhouette over all data of the dataset** measures how appropriately the data has been clustered

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html
https://en.wikipedia.org/wiki/Silhouette_(clustering)

Rand Index

- Rand Index (RI) measures the similarity of two assignments
- Given a dataset D and two partitions S and R

$$RI(S,R) = \frac{a+b}{\binom{m}{2}}$$

- a is the number of pairs of elements in D that are in the same subset in S and in the same subset in R
- b is the number of pairs of elements in D that are in a different subset in S and in a different subset in R
- All possible pairs or element if D is the binomial coefficient $\binom{m}{2} = \frac{m(m-1)}{2}$

https://en.wikipedia.org/wiki/Rand_index

Adjusted Rand Index

- Rand Index can be interpreted as accuracy
- The Rand index does not ensure to obtain a value close to 0.0 for a random labelling
- The adjusted Rand index corrects for chance and will give such a baseline

$$ARI(S, R) = \frac{RI(S, R) - E[RI]}{\max(RI) - E[RI]}$$

- Max and expectation of RI are computed considering the number and size of clusters as in S and R, and all random clusterings are generated by shuffling the elements

https://en.wikipedia.org/wiki/Rand_index

Adjusted Rand Index

- Two example clusterings for a dataset. The calculated Adjusted Rand index for these two clusterings is 0.94

Final considerations: ML for clustering

- Learn to assign the cluster y of a data point from its features **x**
- ML model = learn a hypothesis $h \in \mathcal{H}$ $h : \mathcal{X} \to \mathcal{Y}$

  such that h(**x**) minimize an empirical risk over data points
- Loss function: how to quantify how good is h(**x**) with respect to the whole data D → clustering error

- Notice: even if not common, it is still possible to use a validation/test set for assessing how data generalize, stability of the clusters and tune hyper-parameters

https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1444
https://arxiv.org/abs/0909.3052
https://amueller.github.io/aml/04-model-evaluation/17-cluster-evaluation.html
https://www.jstor.org/stable/29777144

Any questions?

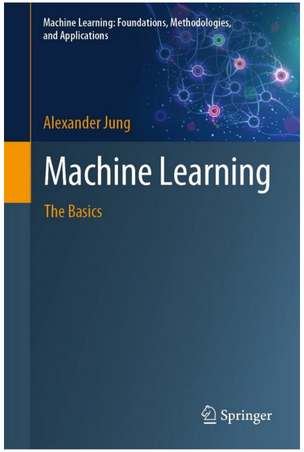
Self-assessment quiz

- Given the following dataset with 9 samples and 1 feature, provide the possible outputs for a hard-clustering method and for a soft-clustering method, both with 3 clusters
- Perform up to 5 iterations of k-Means, with k=2, on the same dataset
- Compute the Rand Index of the previous clustering with respect to the following clustering assignment: [0, 0, 0, 0, 0, 0, 0, 1, 1]

Feature 1
10
7
7
5
-1
10
2
-3
0

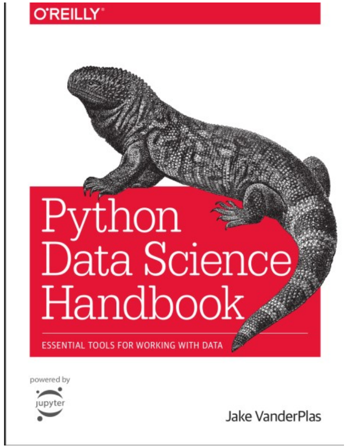
References: readings

- Chapter 8

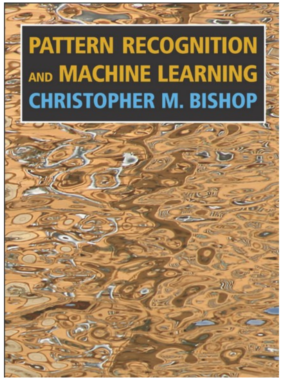


https://scikit-learn.org/stable/modules/clustering.html

- Chapter 5



- Chapter 9



Slide acknowledgments

- Alexander Jung – Aalto University
- Elena Baralis, Tania Cerquitelli - Politecnico di Torino