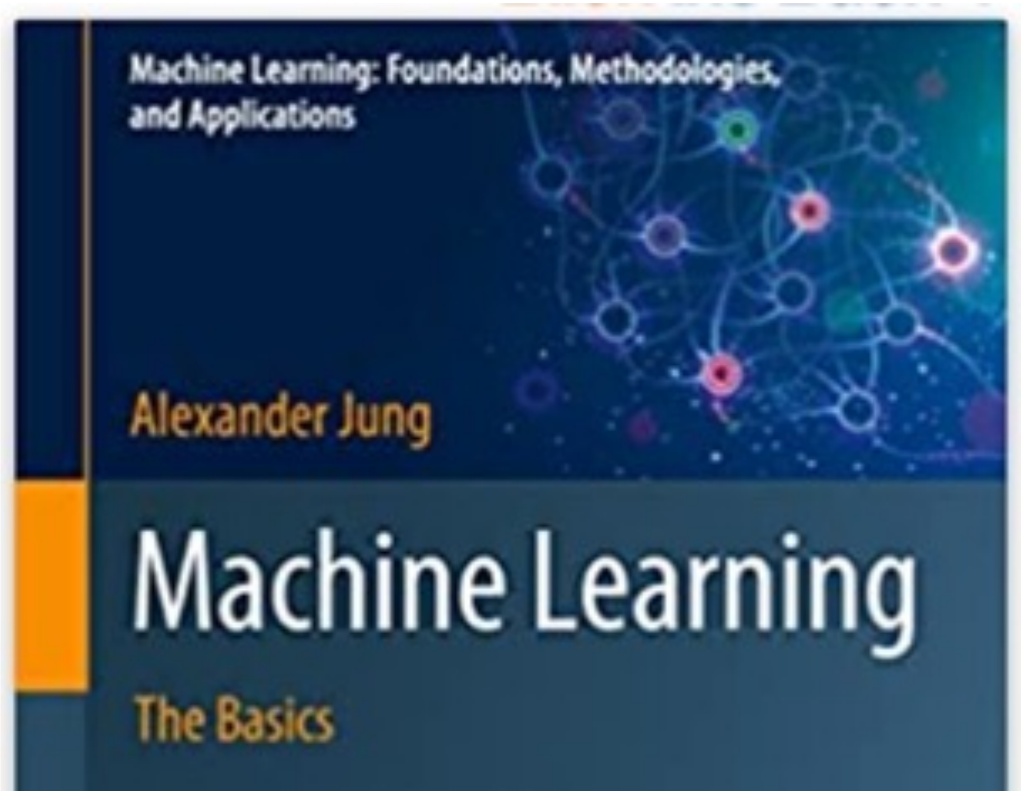
Hard Clustering

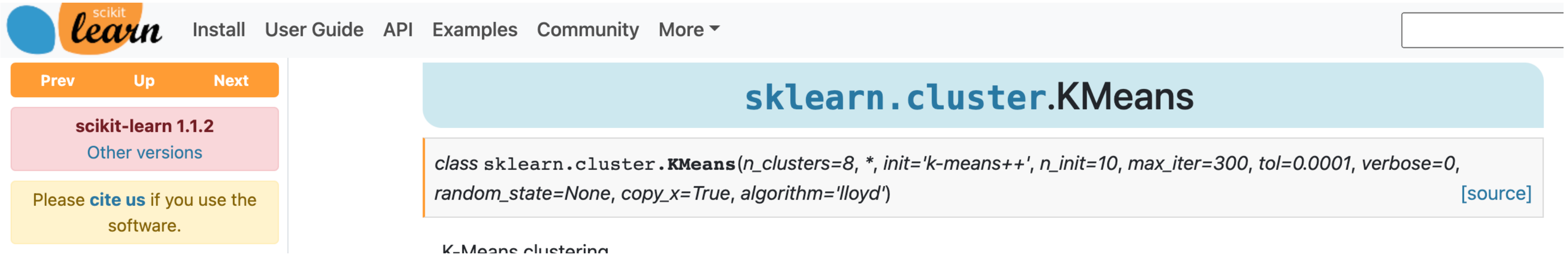
Alex(ander) Jung
Assistant Professor for Machine Learning
Department of Computer Science
Aalto University

Reading.

Sec. 8.1 of https://mlbook.cs.aalto.fi



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


scikit learn Install User Guide API Examples Community More

Prev Up Next

scikit-learn 1.1.2
Other versions

Please cite us if you use the software.

sklearn.cluster.KMeans

class sklearn.cluster.**KMeans**(*n_clusters=8, *, init='k-means++', n_init=10, max_iter=300, tol=0.0001, verbose=0, random_state=None, copy_x=True, algorithm='lloyd'*) [source]

K-Means clustering

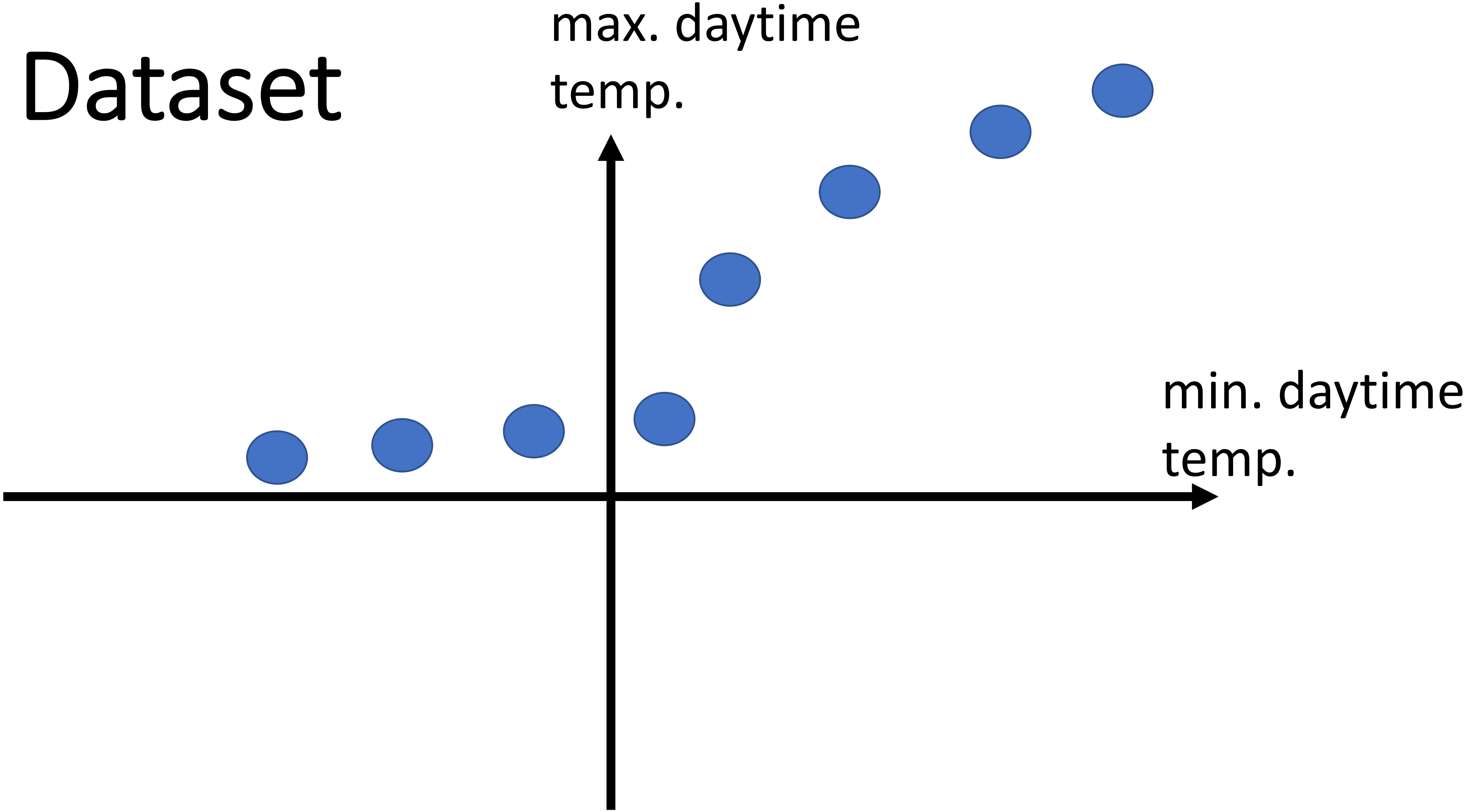
What I want to teach you today:

- basic idea of hard clustering
- k-means method for hard clustering
- optimization problem underlying k-means
- how to choose number of clusters

First things First

What are three main components of Machine Learning ?

A Dataset



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.*

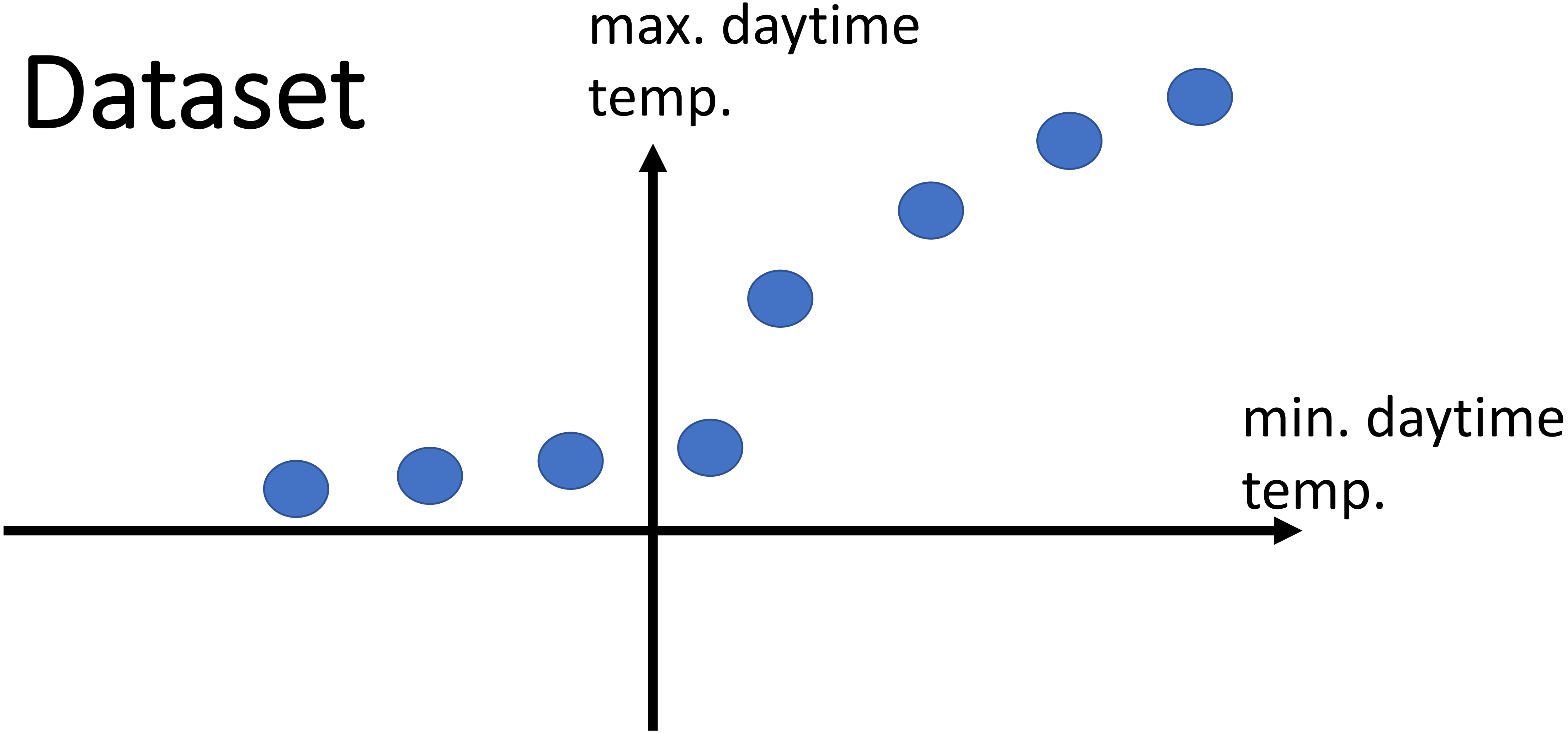
https://en.wiktionary.org/wiki/cluster

Informal Definition

a cluster corresponds to a subset of datapoints that are in some sense homogeneous or similar

plethora of different definitions for "homogeneous" and "similar"

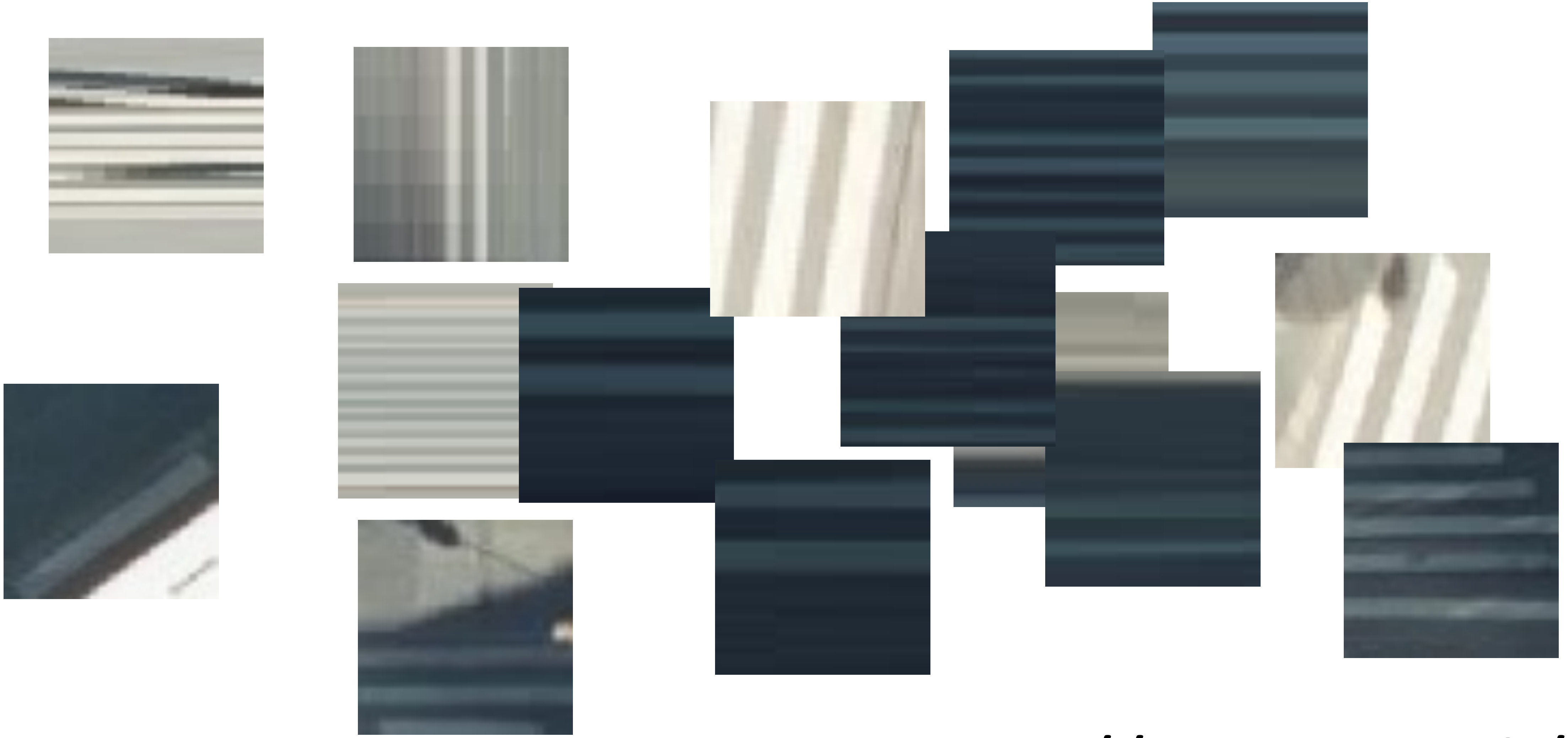
A Dataset



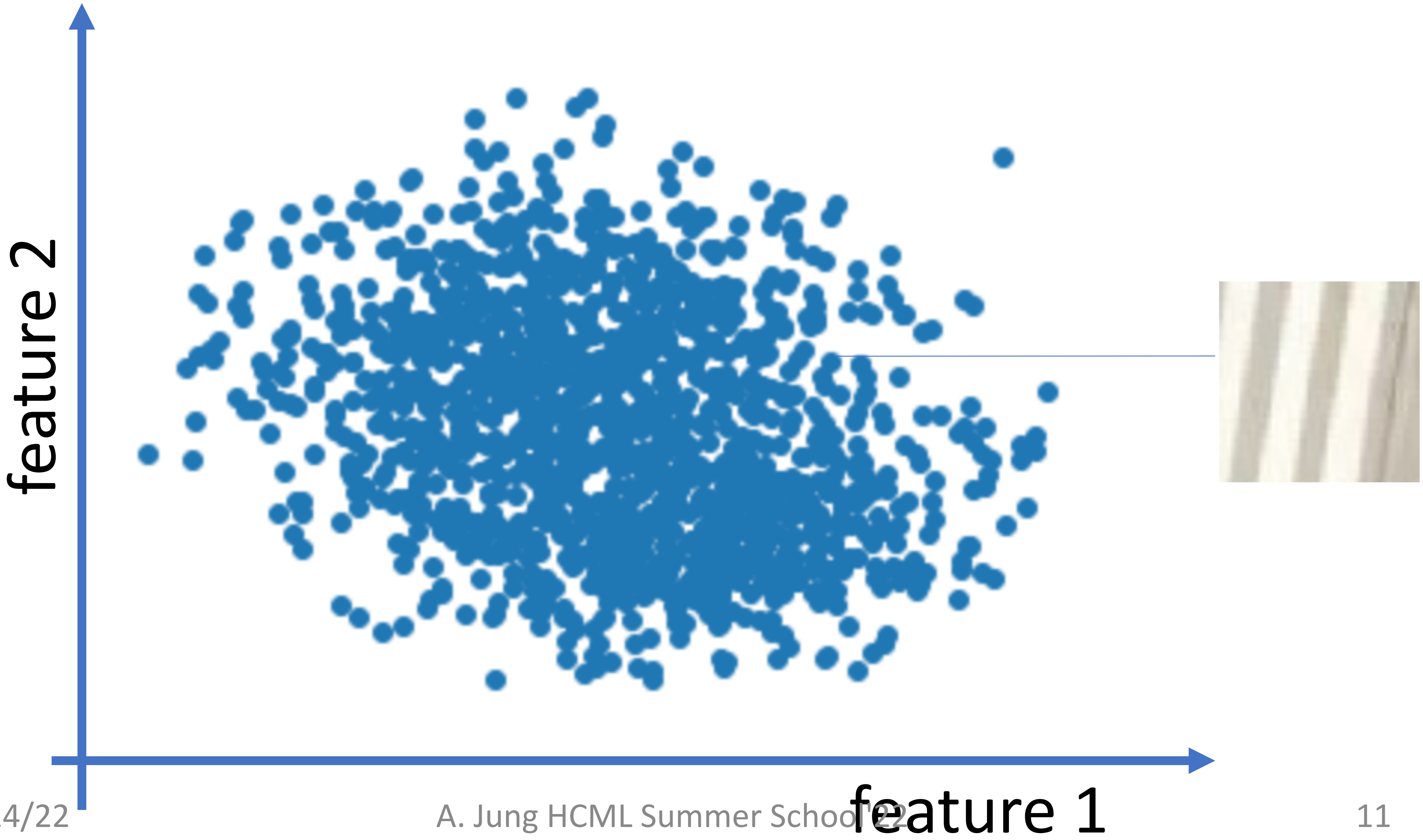
How many clusters do you see ?

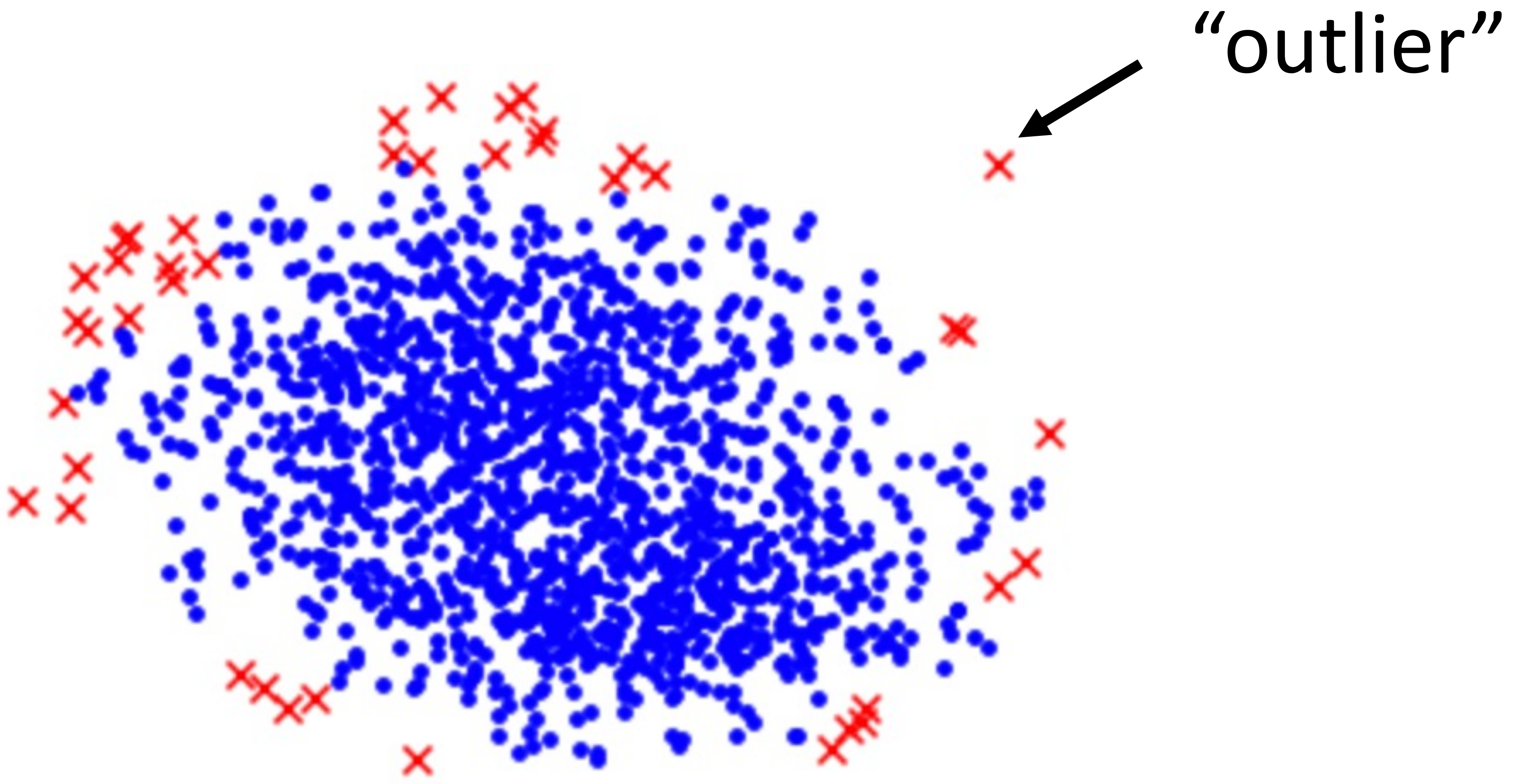
Clustering for Outlier/Anomaly Detection

Dataset = “Bunch of Images”

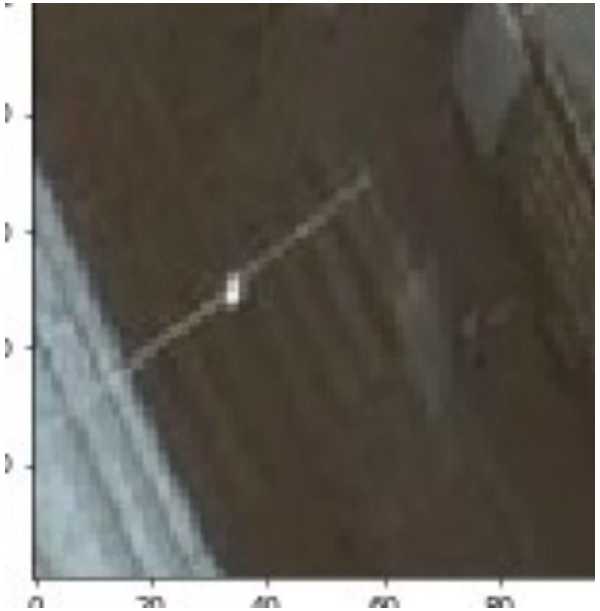


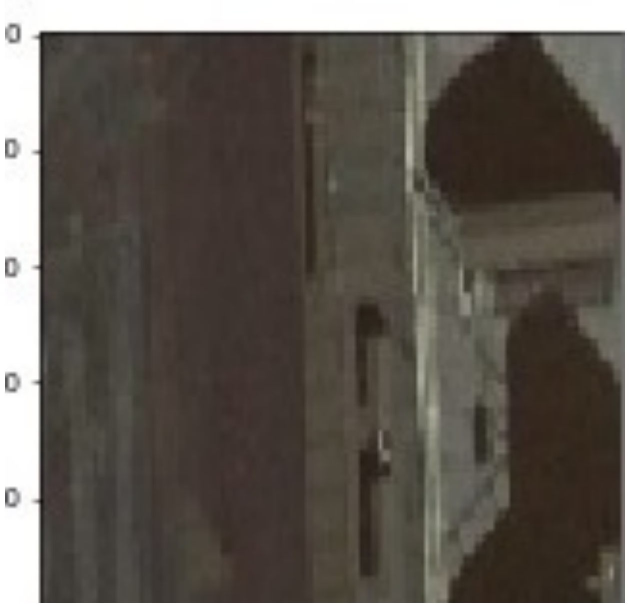
https://kartta.hel.fi/


feature 2
feature 1


"outlier"

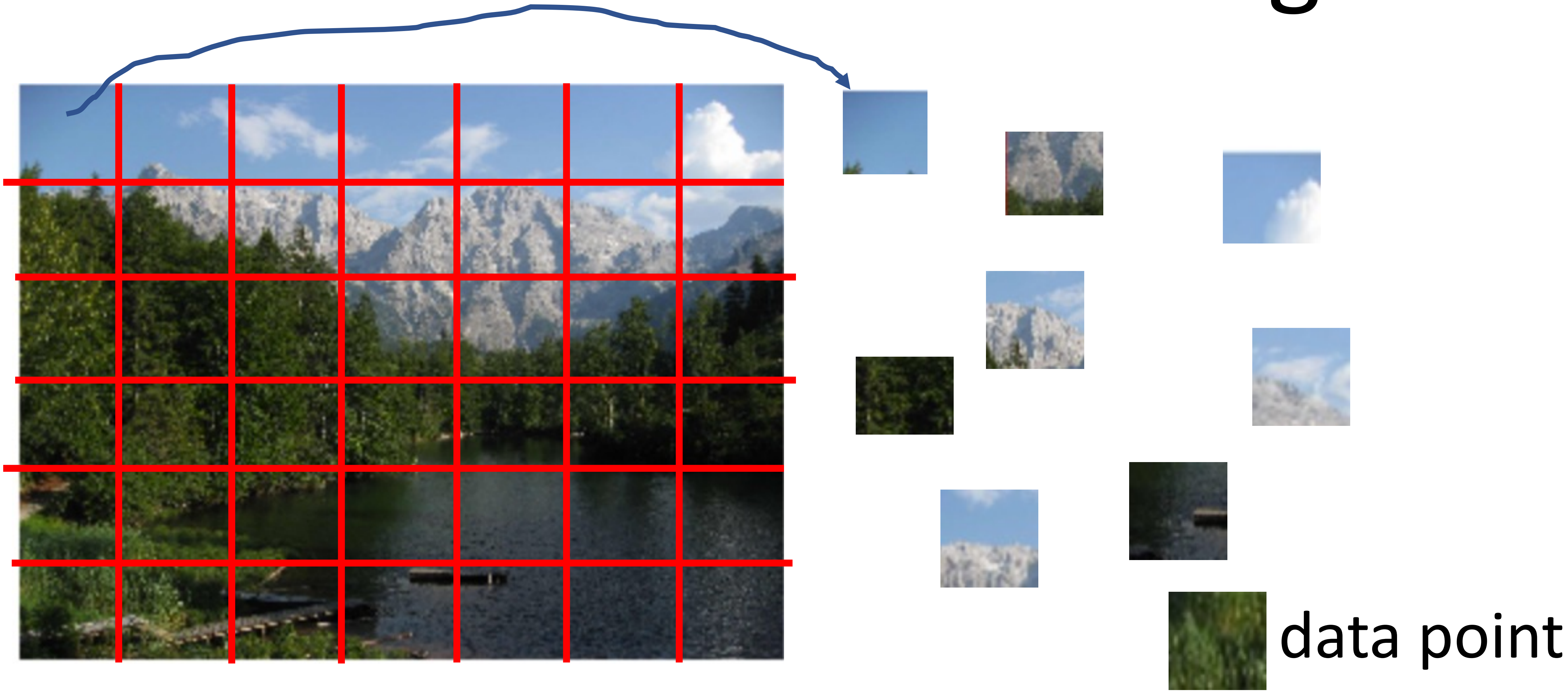
some outliers



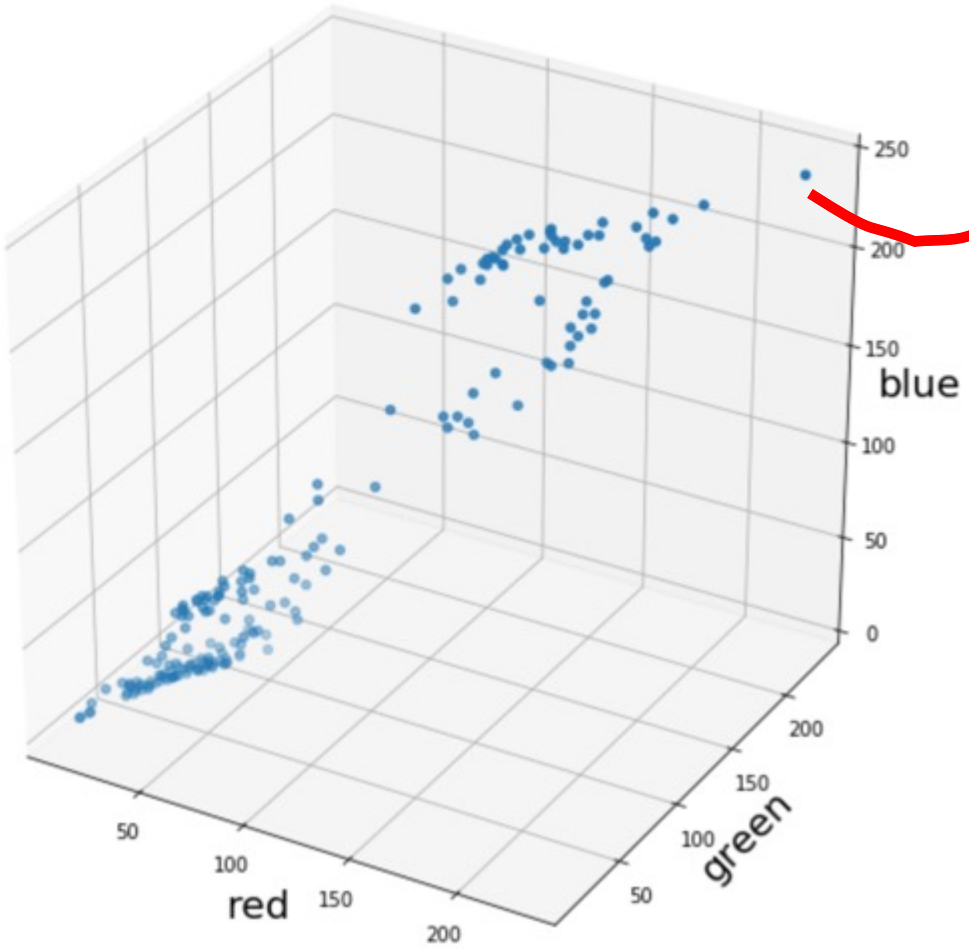


Clustering for Image Segmentation

Dataset = Patches of Image

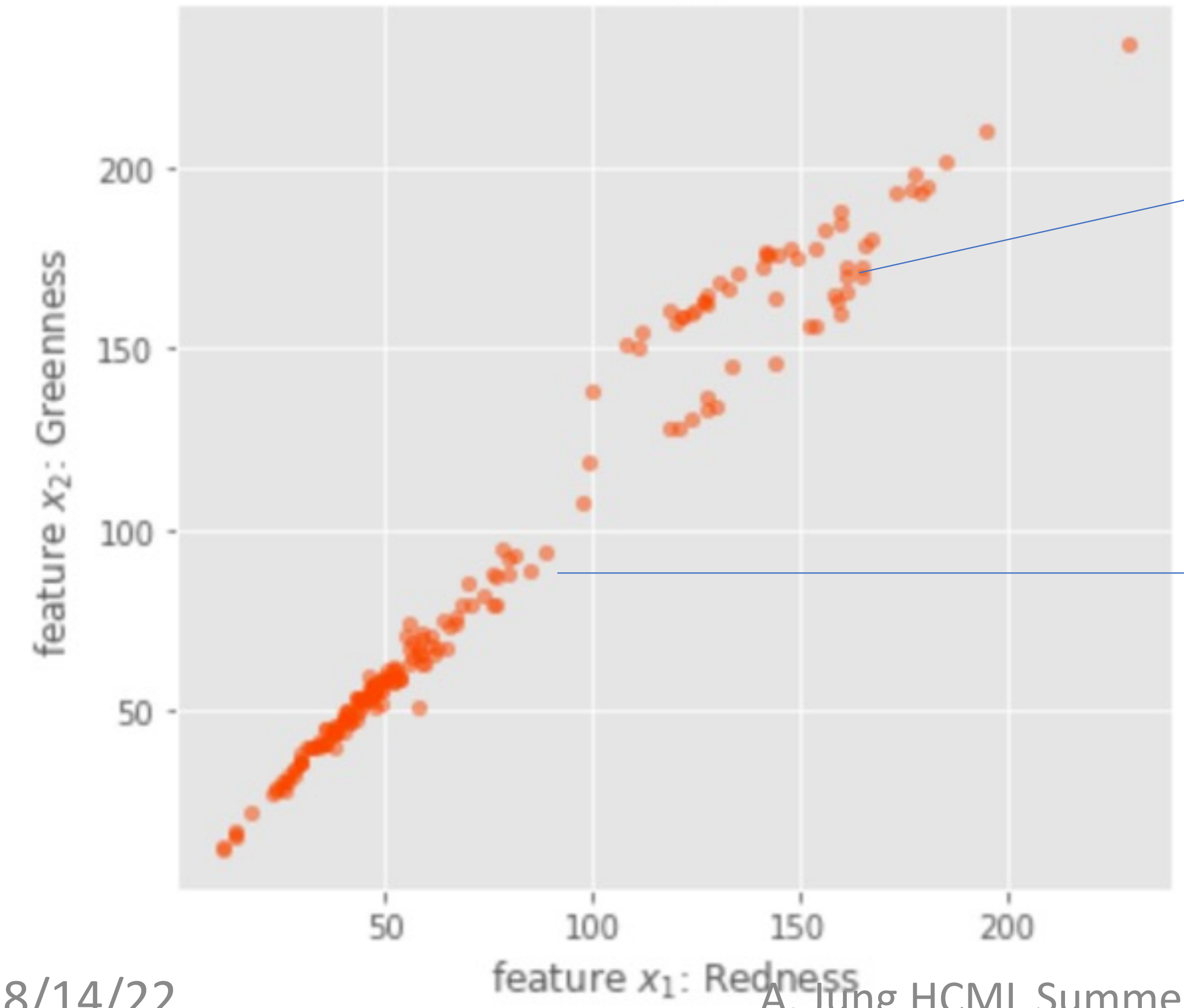


Using Three Features



three features:
average red, green and blue component

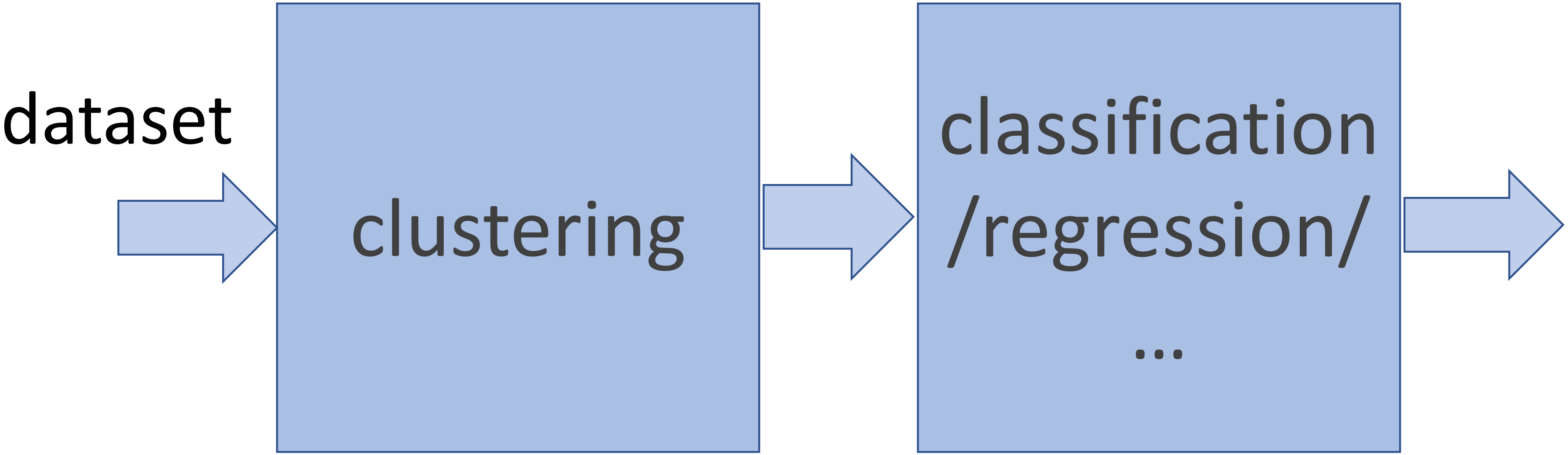
Using Two Features (Red+Green)

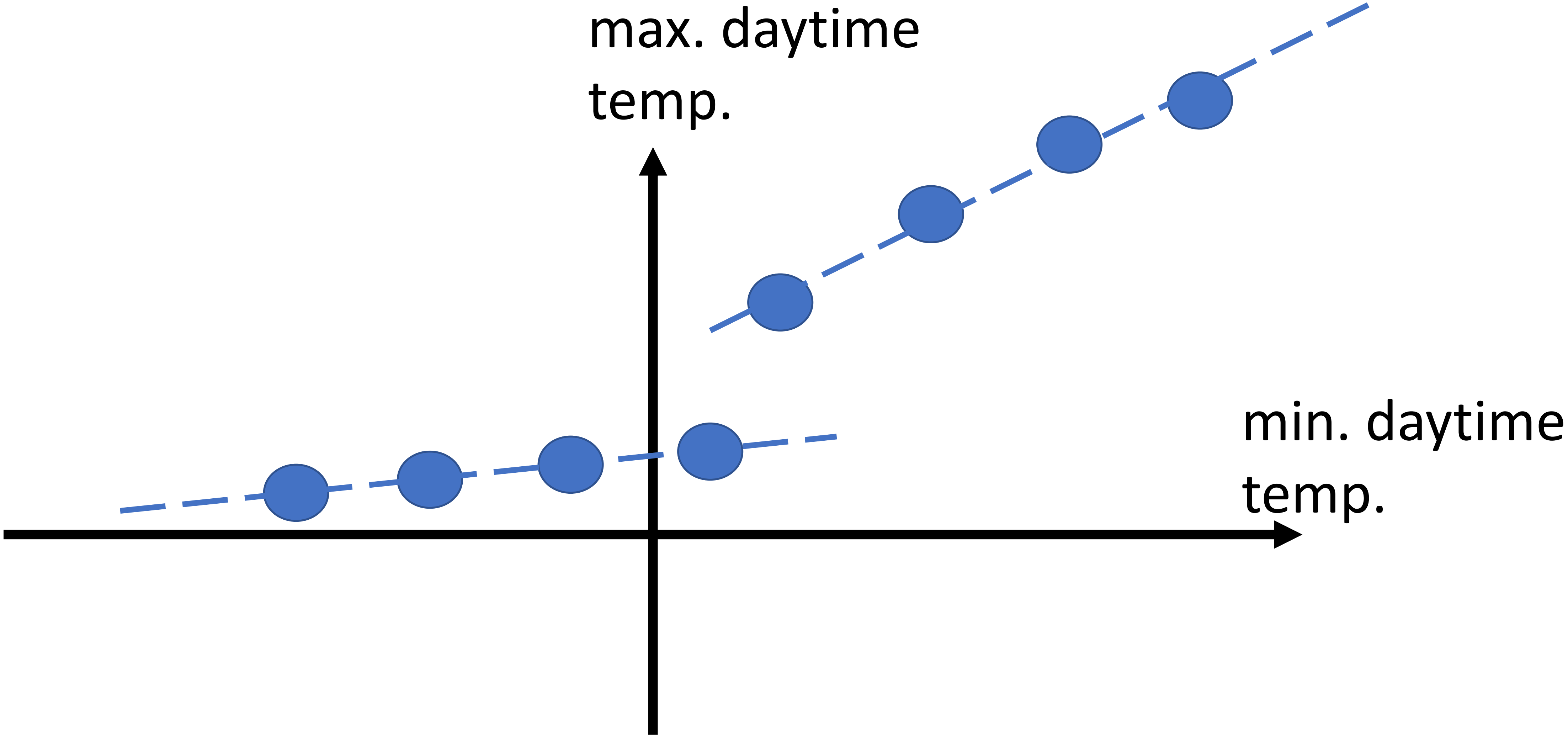


Use Clustering For Image Segmentation

Pre-Processing

Clustering as Pre-Processing





first partition into two clusters. then apply linear regression separately to each cluster

Hard Clustering

- datapoints $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right)$,…, $\left(\boldsymbol{x}^{(m)}, y^{(m)}\right)$
- i-th datapoint characterized by n features

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

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

Hard Clustering Methods

- datapoints $\left(\boldsymbol{x}^{(1)}, y^{(1)}\right)$,…, $\left(\boldsymbol{x}^{(m)}, y^{(m)}\right)$
- 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
- does not require true cluster index $y^{(i)}$ of any datapoint

Hard Clustering Methods

feature vectors

$\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}$

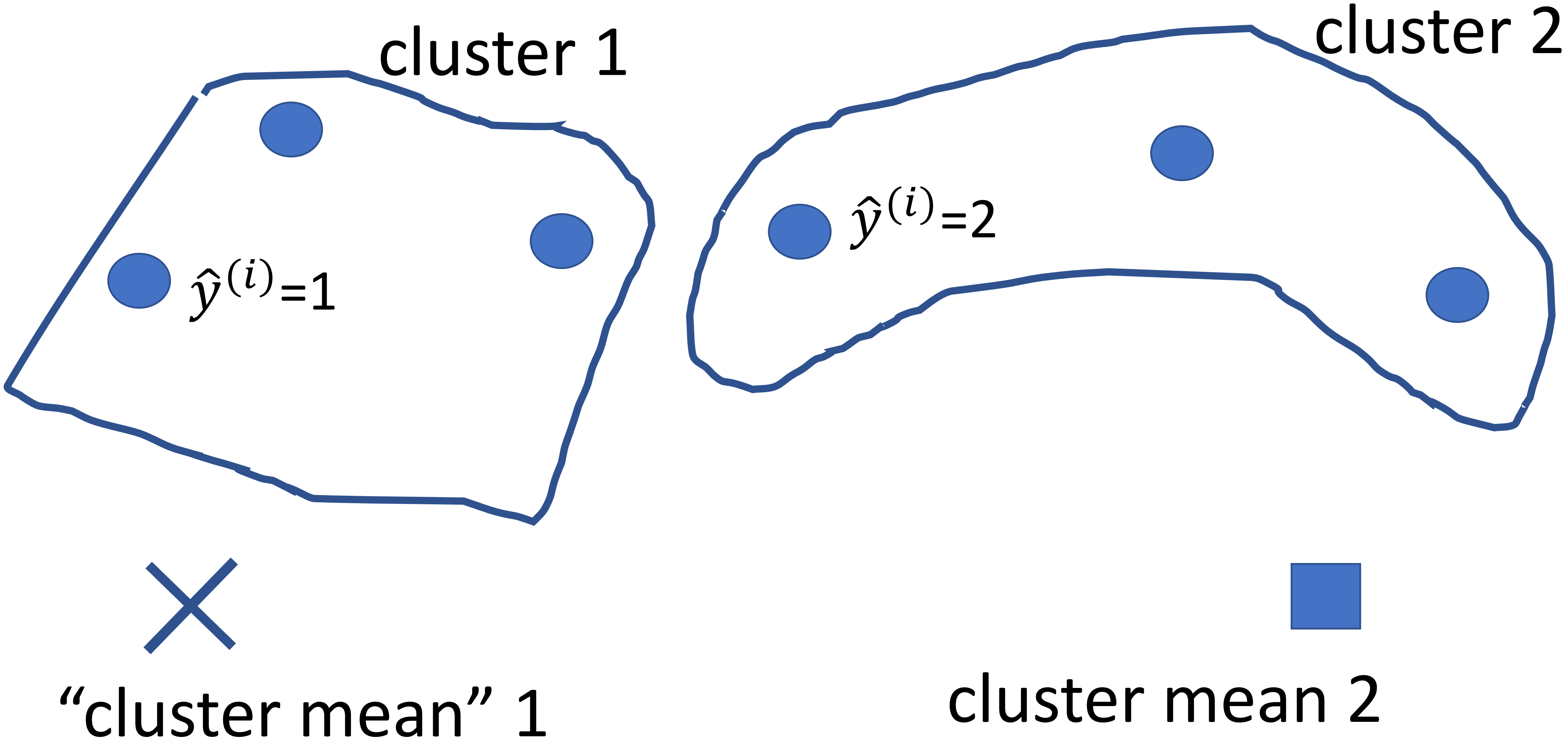
a hard clustering method

predicted cluster assignments

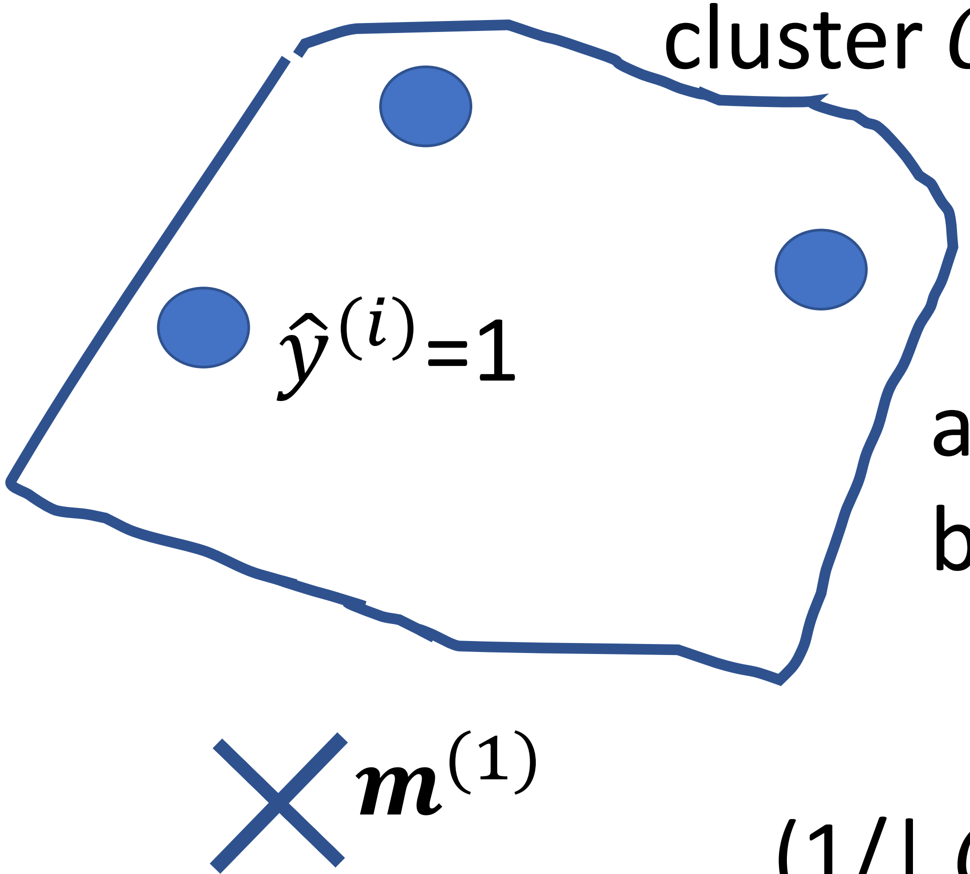
$\hat{y}^{(1)},\ldots,\hat{y}^{(m)}$

Hard Clustering with k-Means

Representing a Cluster by a Mean



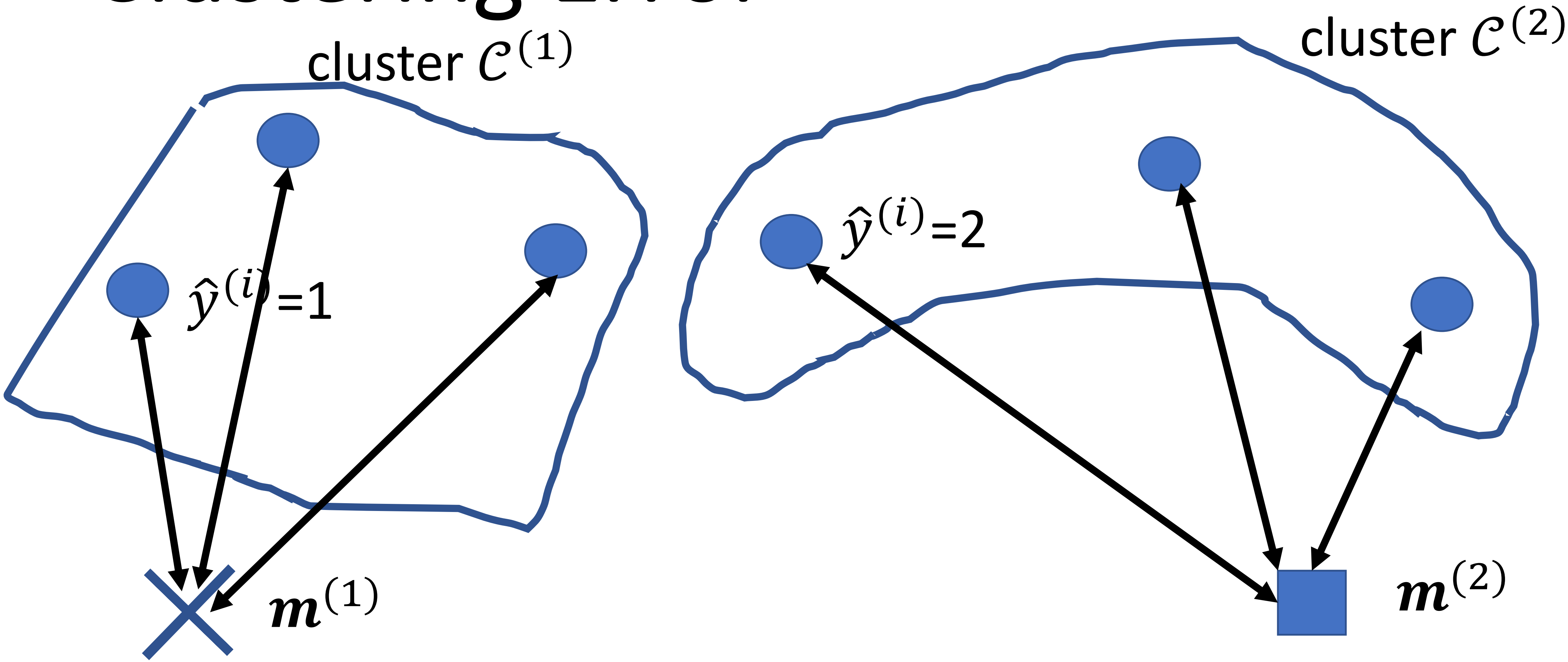
Cluster Spread



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



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

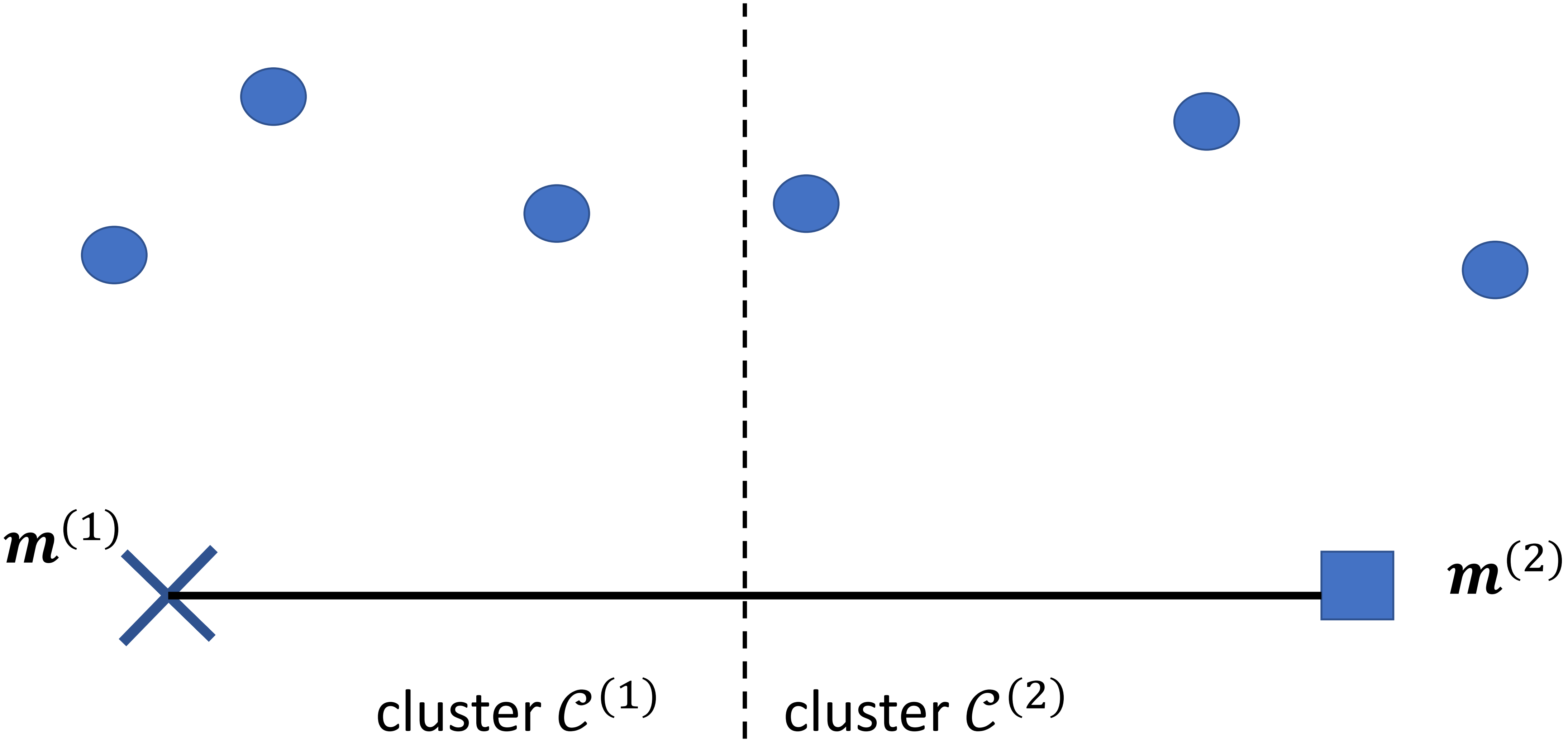
Update Cluster Assignments

for given cluster means, clustering error is minimized by assigning i-th datapoint to cluster with nearest cluster mean

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

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

Update Cluster Assignment



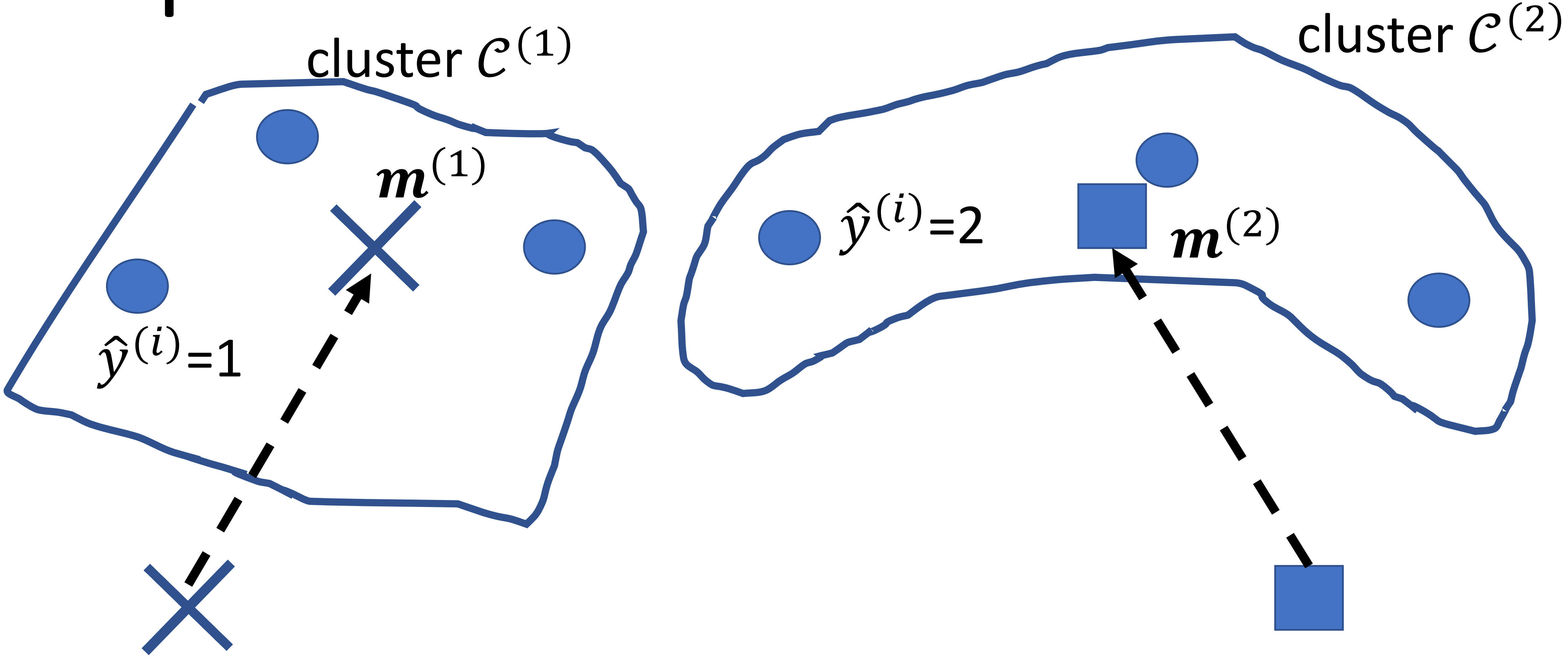
Update Cluster Means

for given cluster assignments, clustering error is minimized by representing c-th cluster by the cluster mean

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

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

Update Cluster Means



Minimizing the Clustering Error

clustering error

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

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

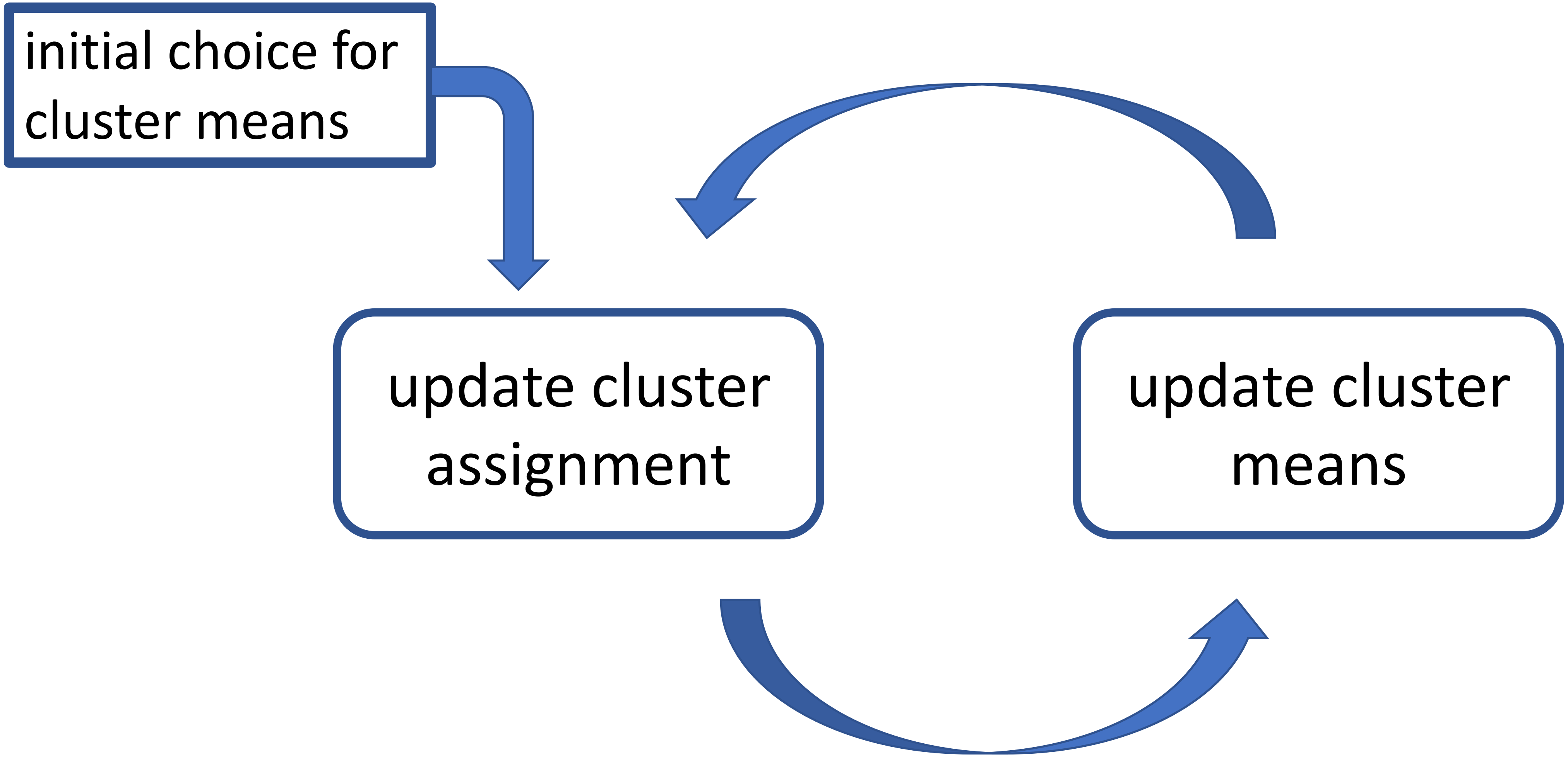
clustering error

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

for given assignments $\hat{y}^{(i)}$, finding cluster means $\boldsymbol{m}^{(c)}$ that minimize clustering error is easy

for given cluster means $\boldsymbol{m}^{(c)}$, finding assignments $\hat{y}^{(i)}$ that minimize clustering error is easy

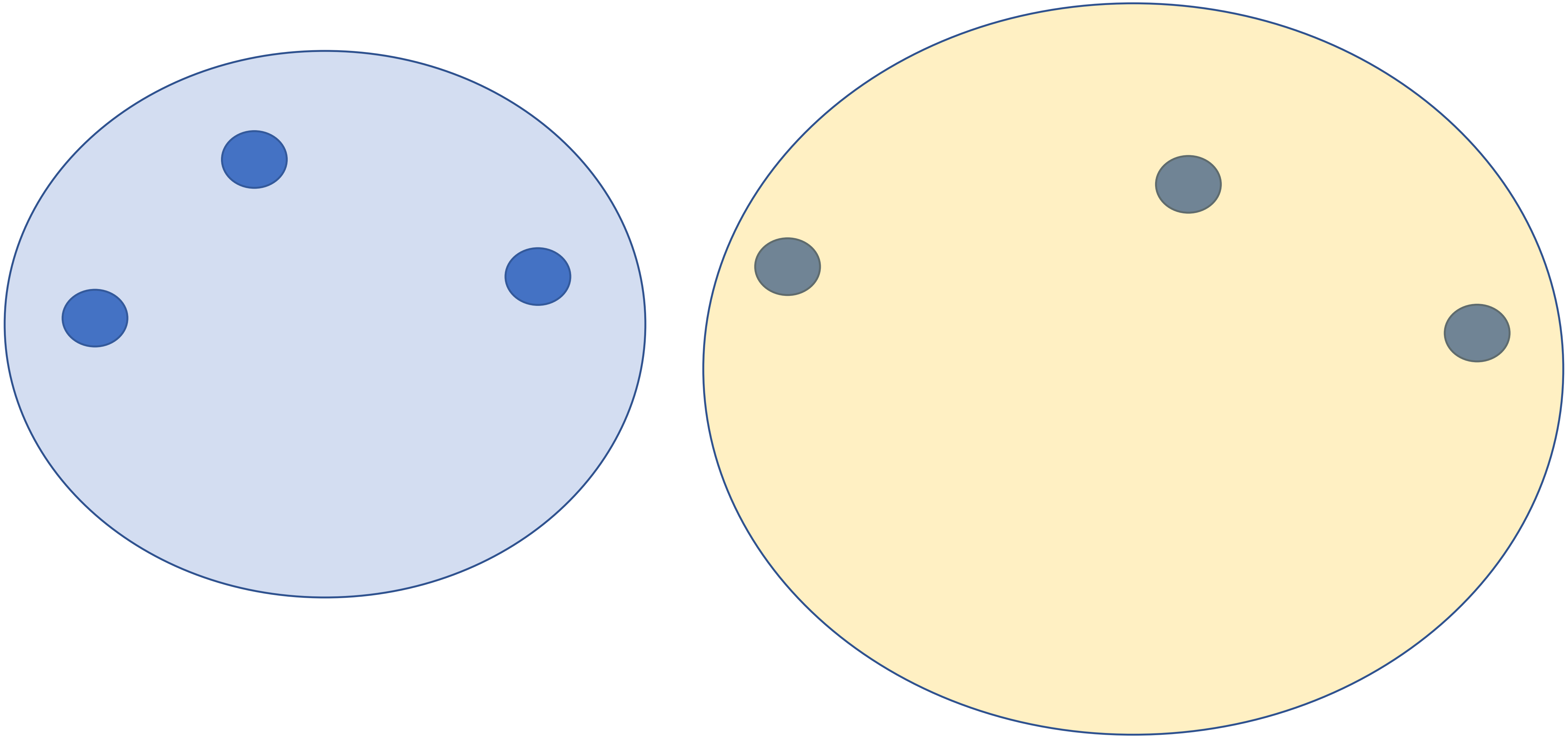
“k-Means”


initial choice for cluster means
update cluster assignment
update cluster means

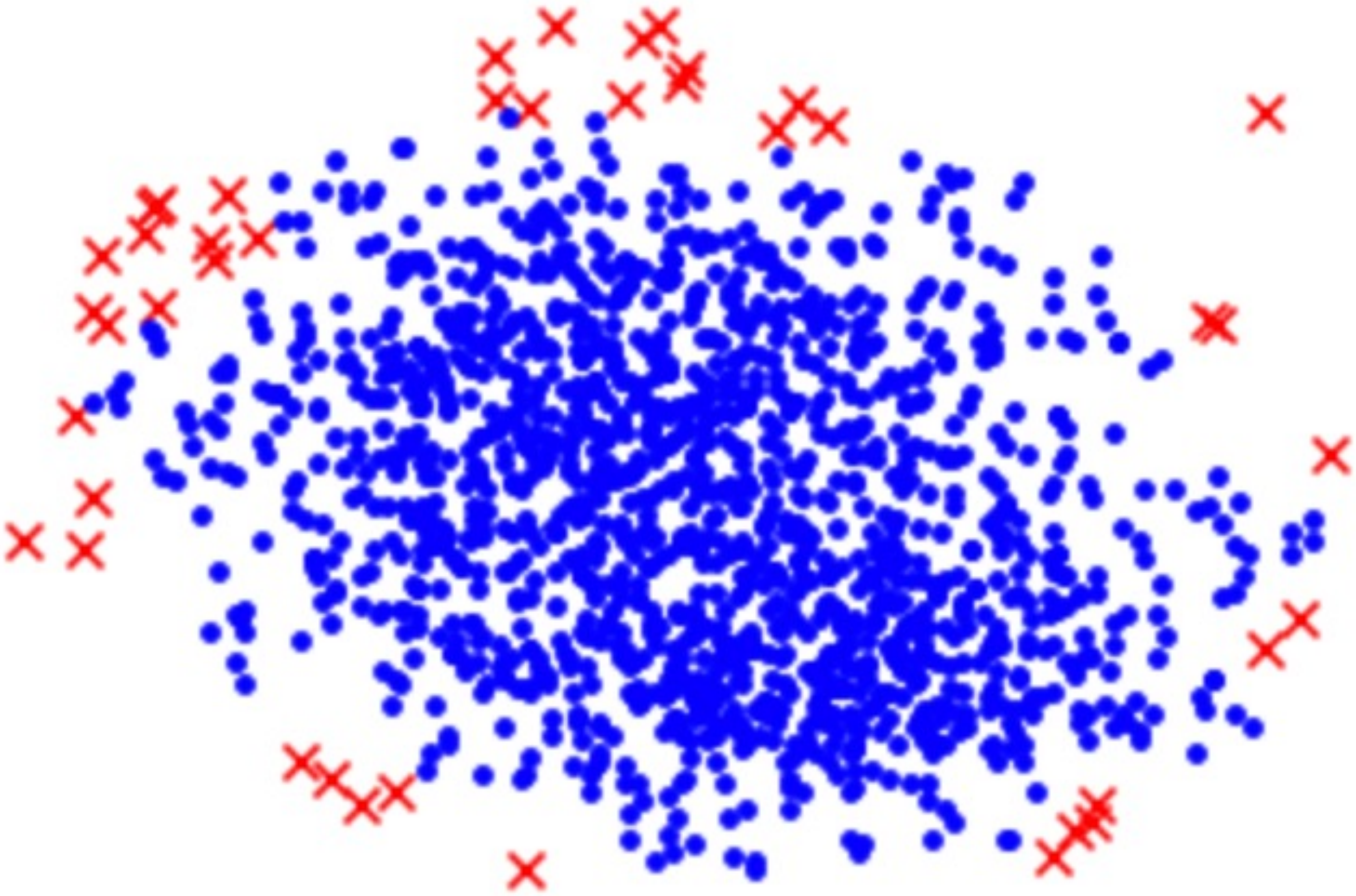
“k-Means” (Algorithm 8 mlbook.cs.aalto.fi)

- 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 Result



Clustering by k-means?



k-Means never increases Clustering Error !

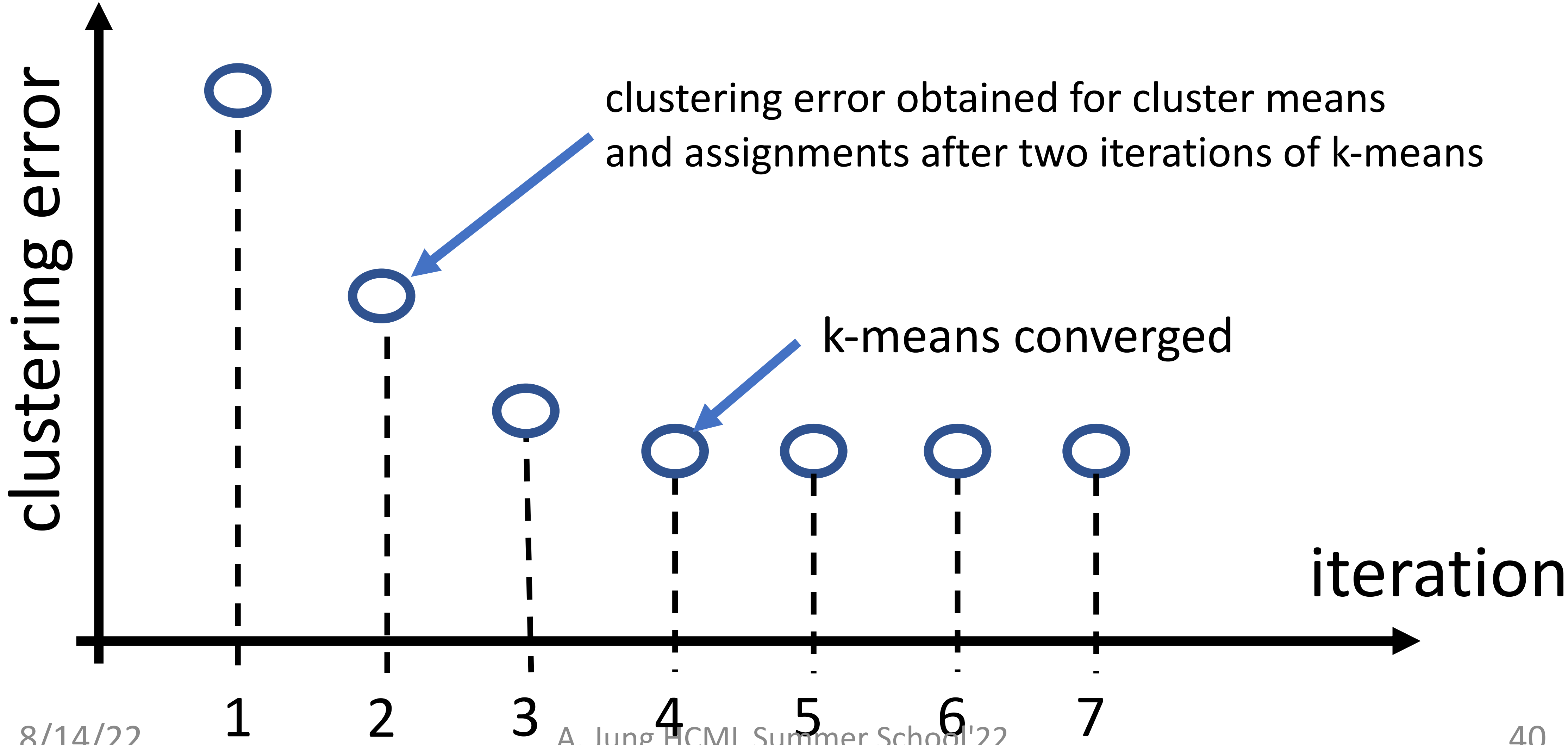
consider cluster means $m^{(c)}$ and assignments $\hat{y}^{(i)}$

run one iteration of k-means

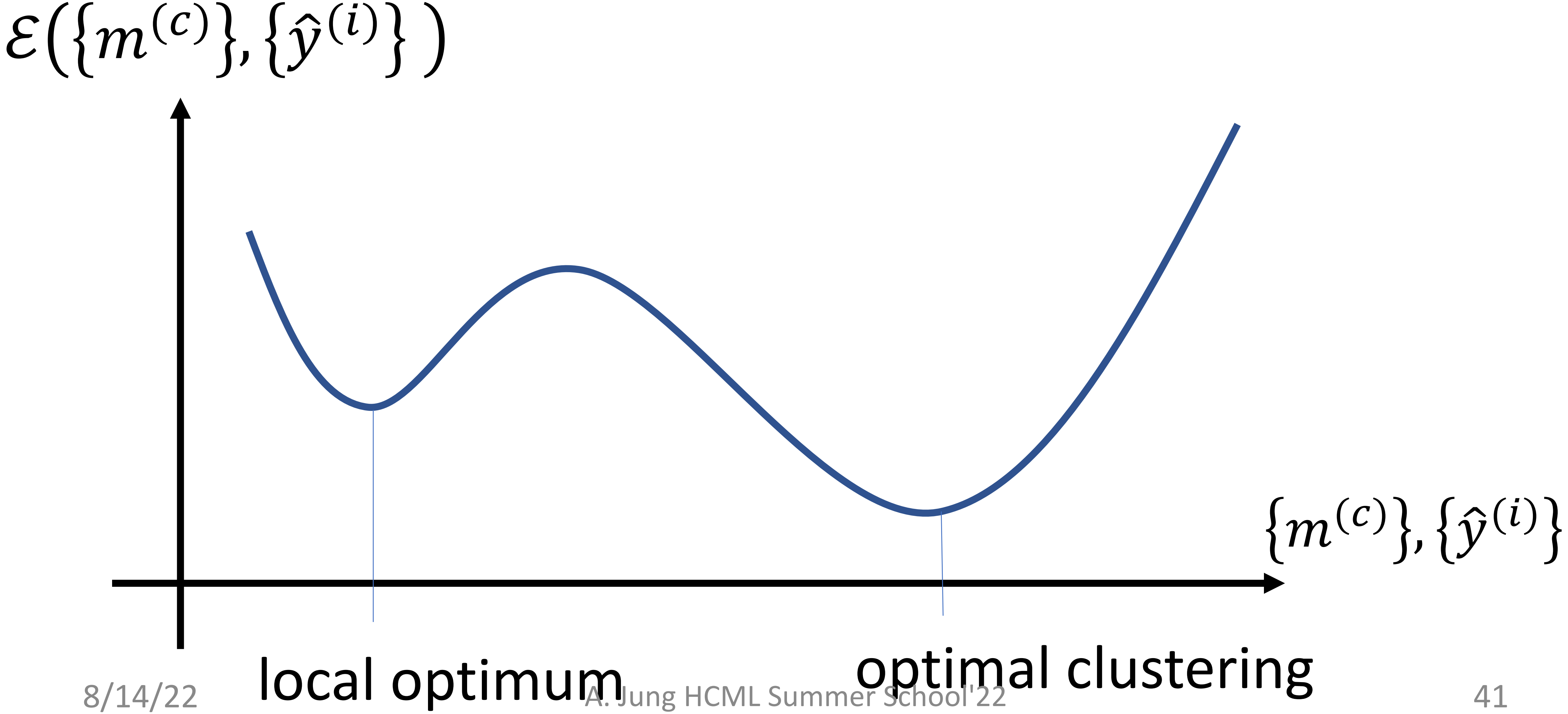
results in new cluster means $\widetilde{m}^{(c)}$ and assignments $\tilde{y}^{(i)}$

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

k-Means as Iterative Optimization Method



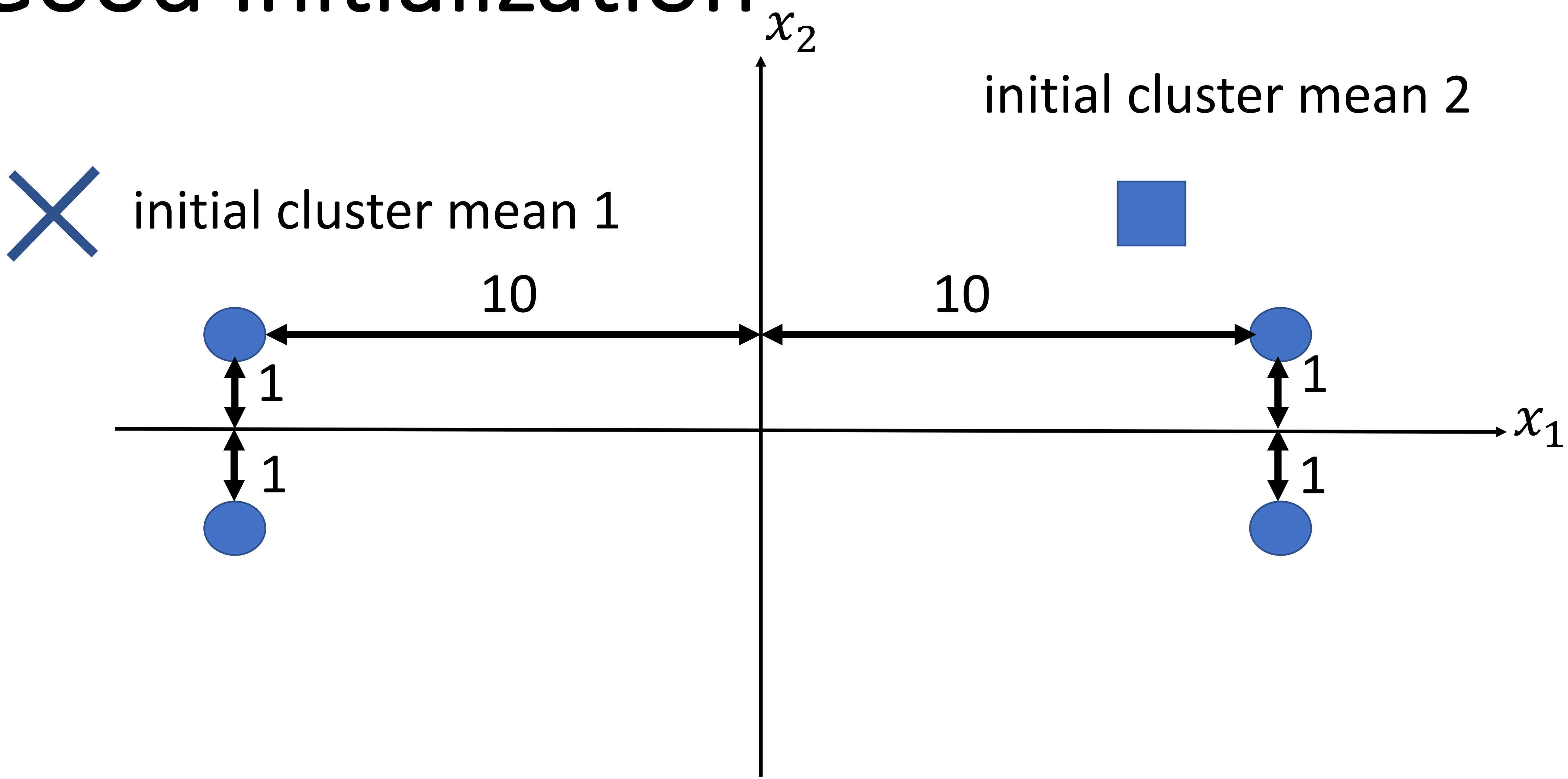
Non-Convexity of Clustering Error



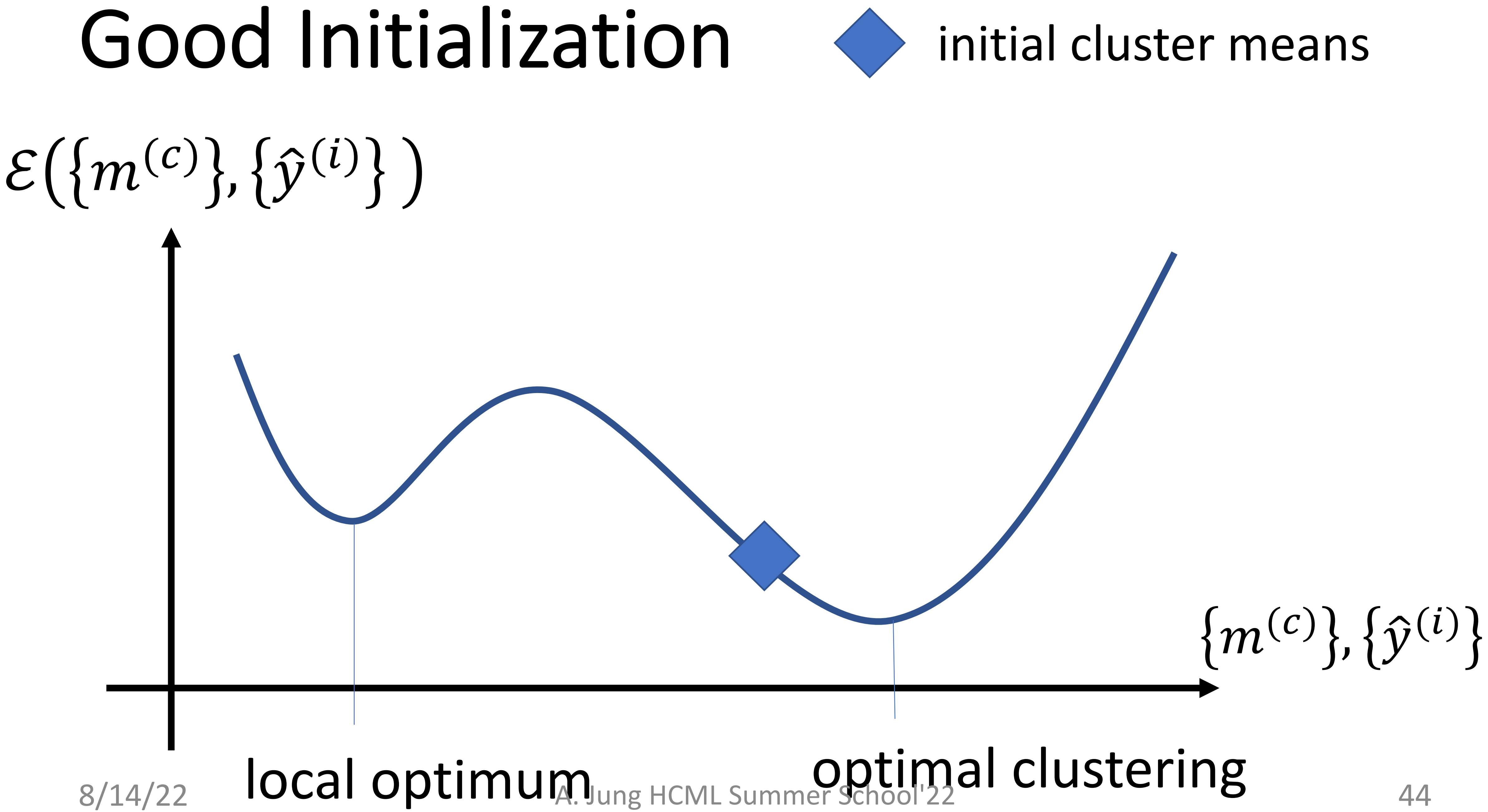
Initialization is Crucial

- k-means requires initial cluster means as inputs
- k-means result depends crucially on init. means
- repeat k-means several times with different init.

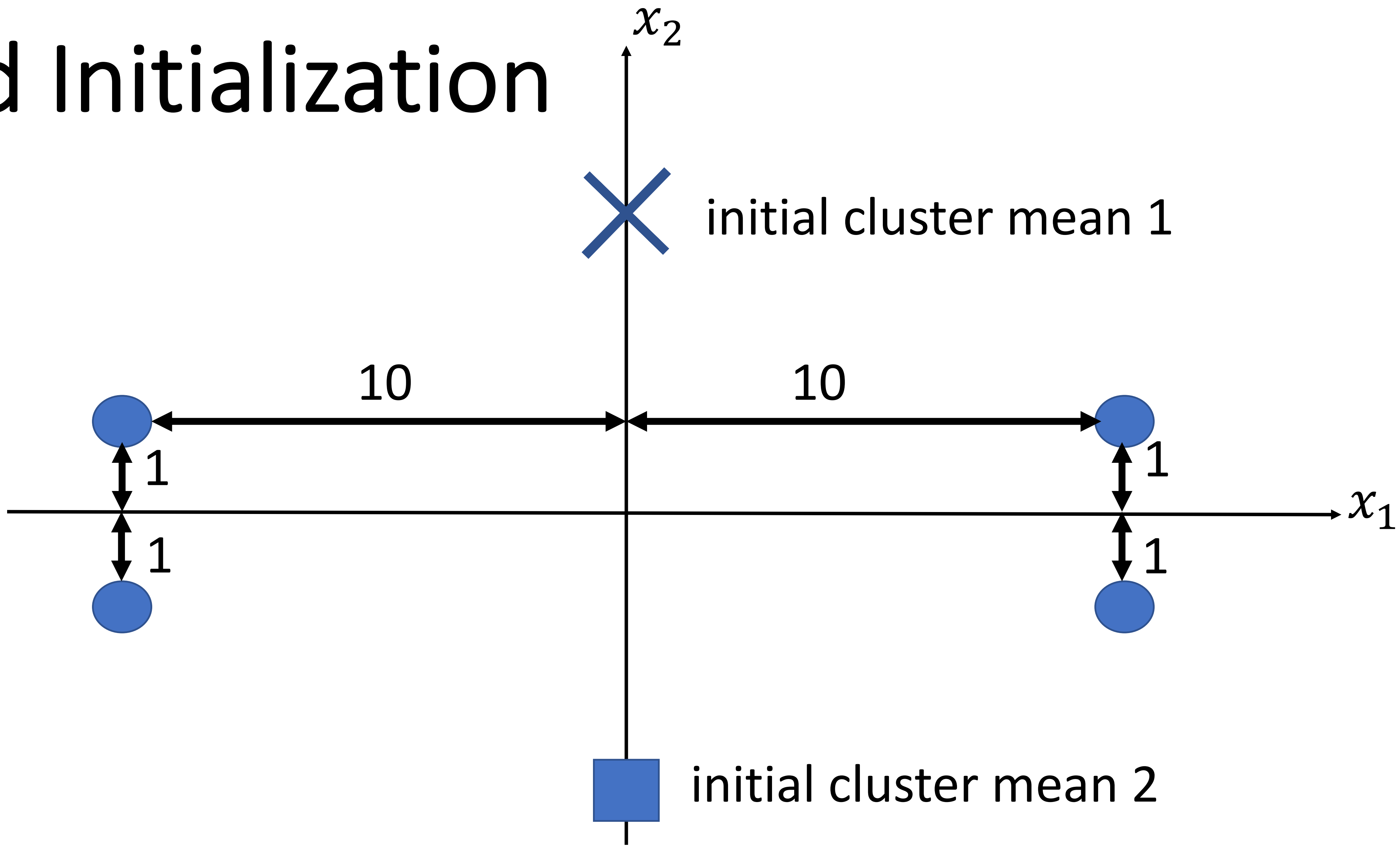
Good Initialization



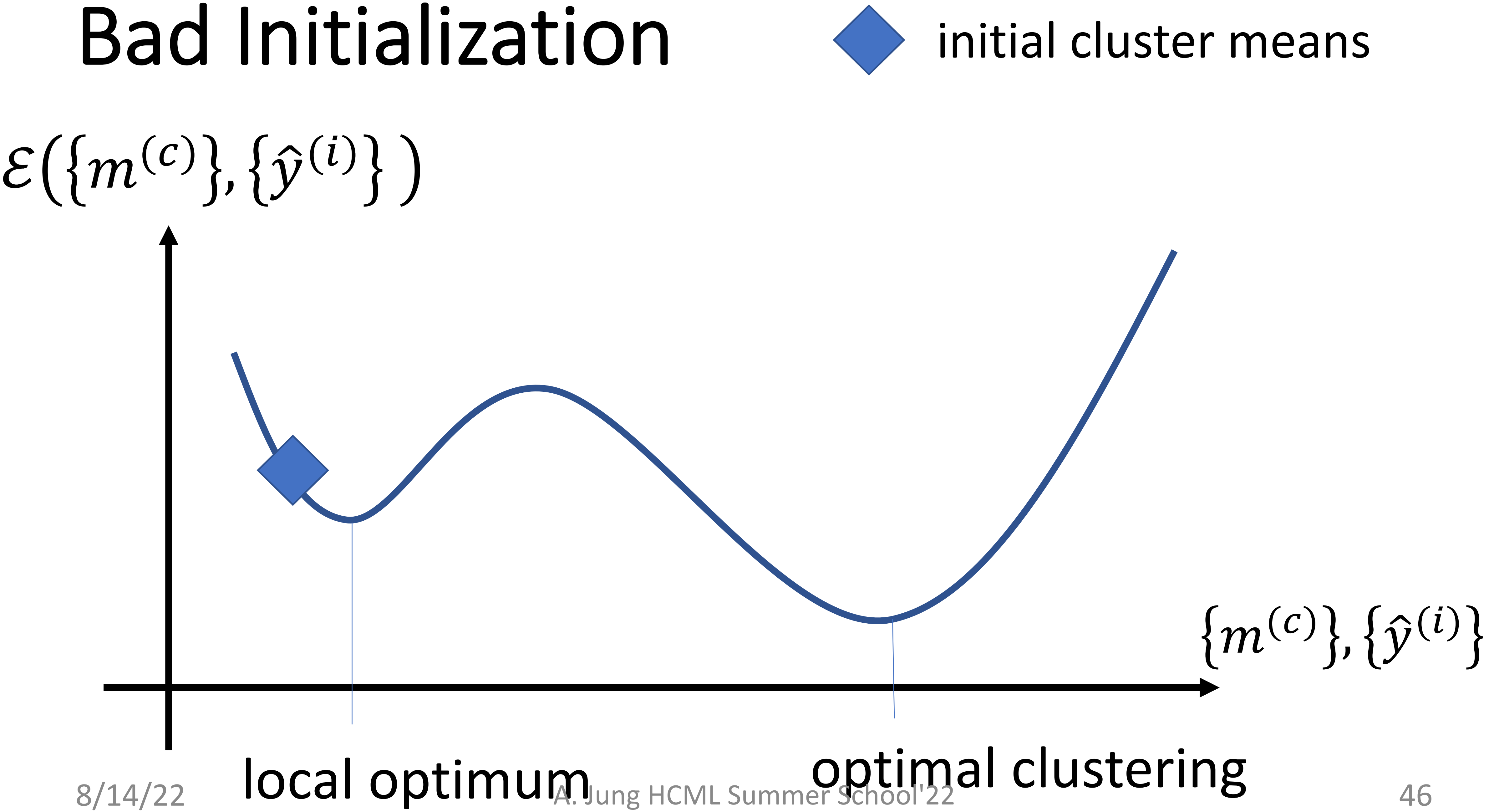
Good Initialization



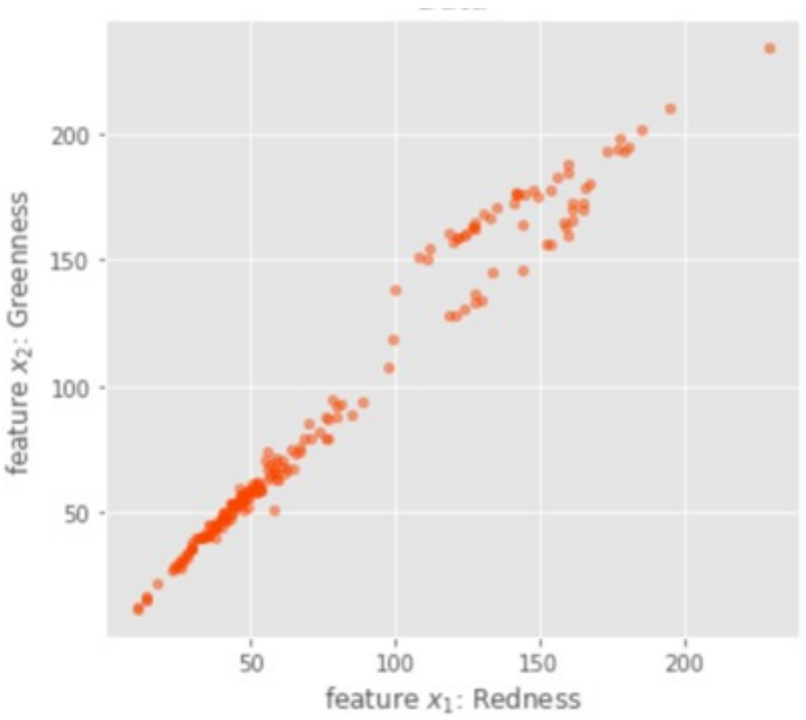
Bad Initialization



Bad Initialization



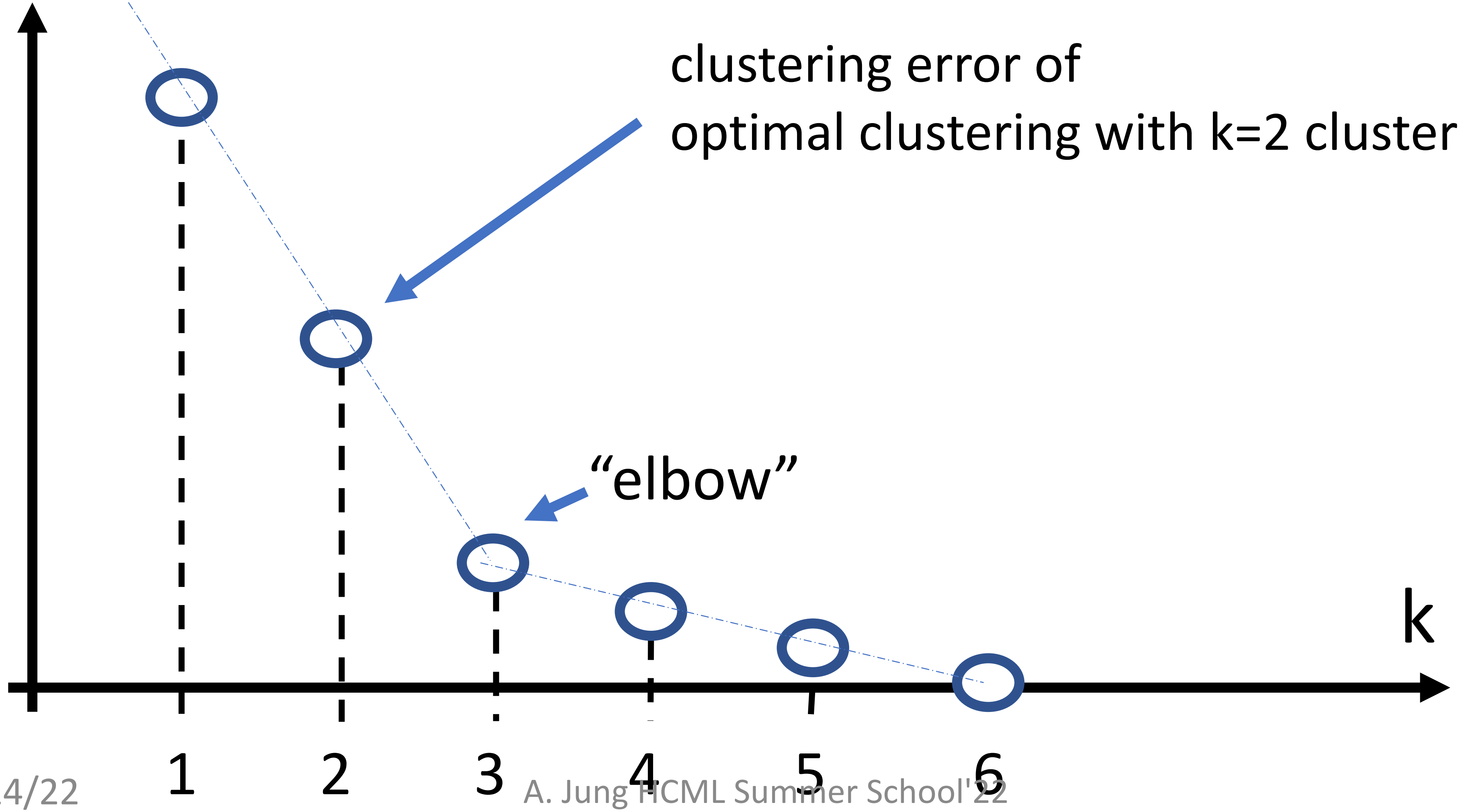
How to choose number k of clusters?



- defined by application (img. seg.)
- desired compression rate
- “elbow-method”

For/Background Segmentation k=2
Cluster 1 = Background, Cluster 2=Foreground

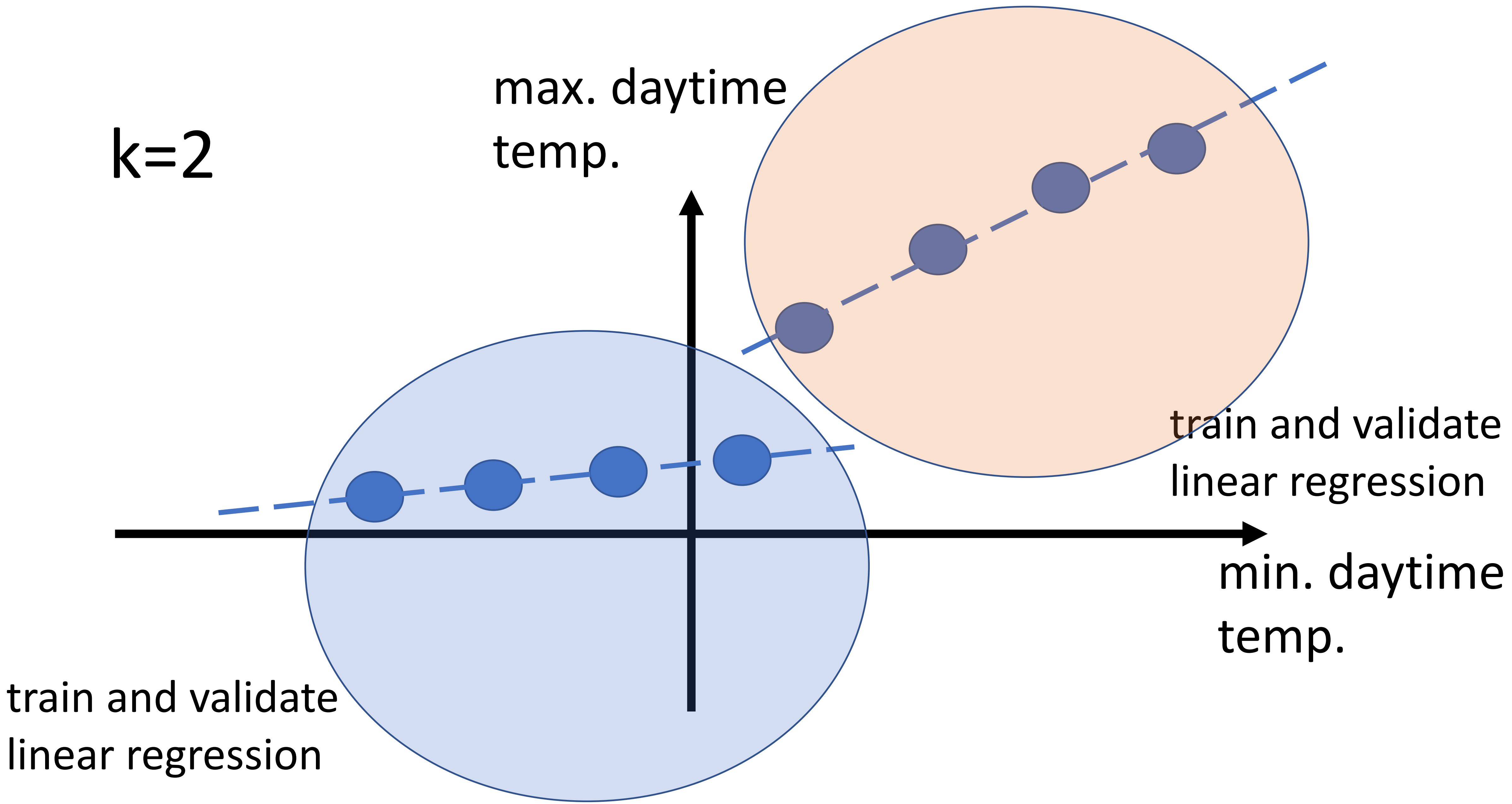
Elbow Method

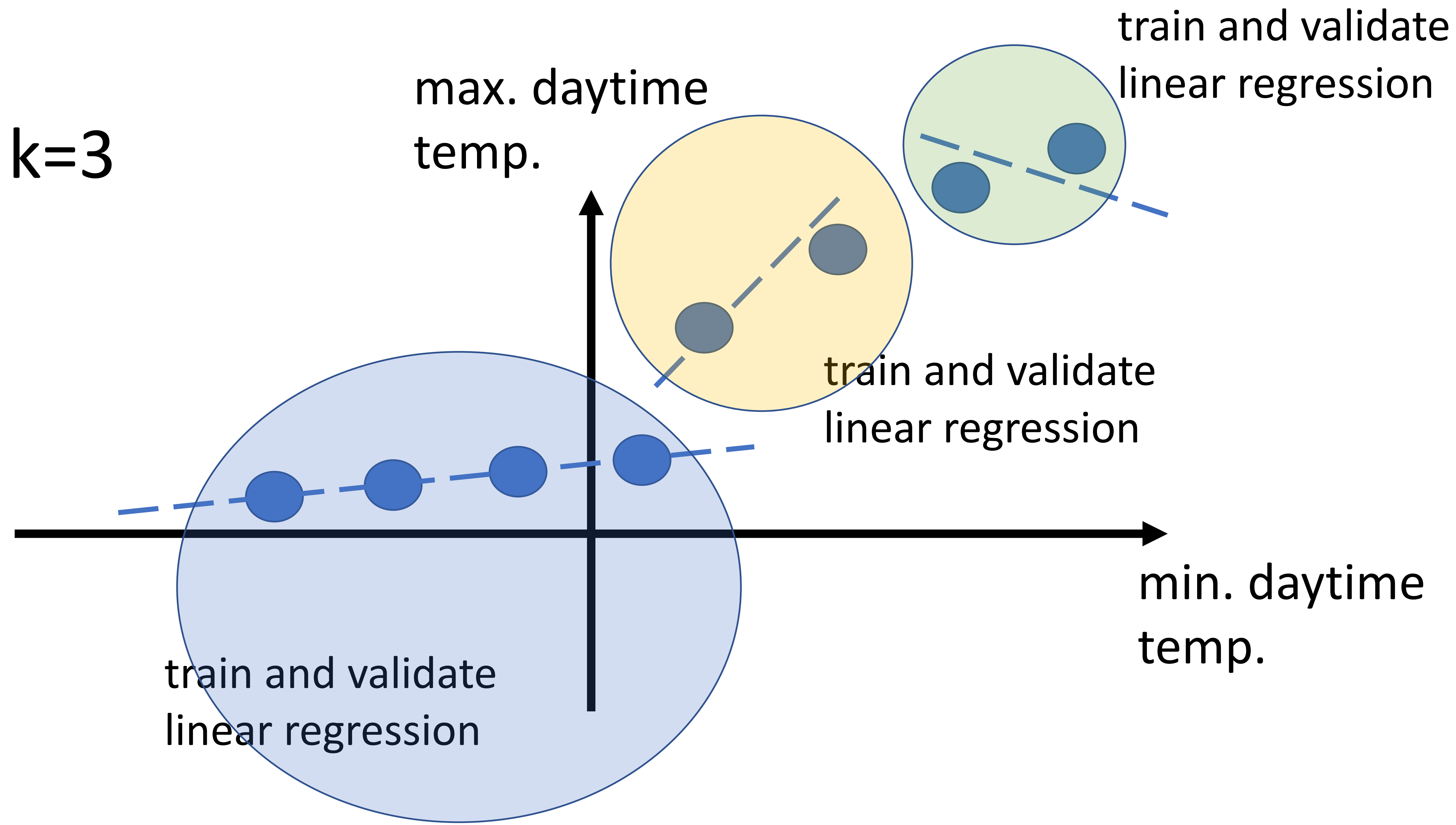


Choose k by Validation Error

- clustering an be used as pre-processing for follow-up regression method

- try different values of k and pick the one resulting in smallest validation error k=2


max. daytime
temp.
train and validate
linear regression
min. daytime
temp.
train and validate
linear regression



k=3
max. daytime
temp.
train and validate
linear regression
train and validate
linear regression
train and validate
linear regression
min. daytime
temp.

To Sum Up

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