University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 9.2 - Unsupervised Learning
K-Means Clustering

Erik Bekkers

(Bishop 9.1)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

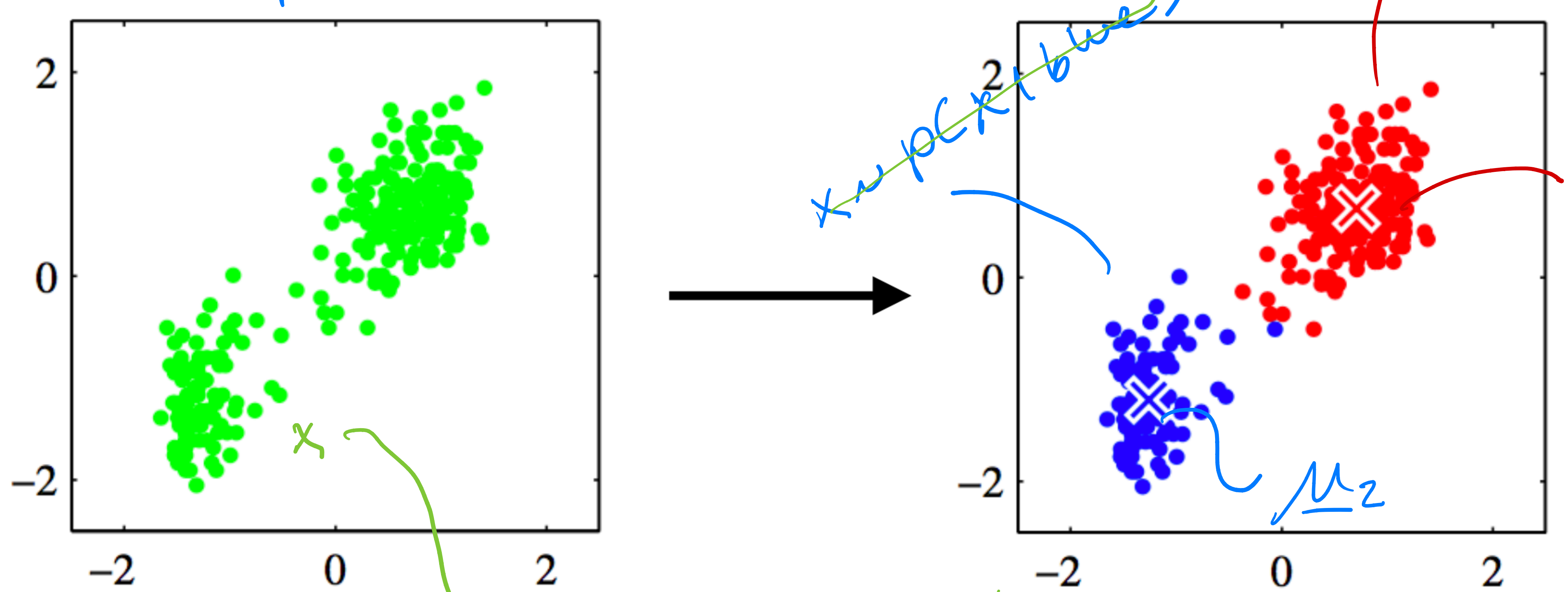
Clustering with K-means

No probabilistic interpretation for now

- Data: a sample of points $\boldsymbol{x}$ (*without* a target)

- Goal: every single data point is assigned to a cluster – **a discrete latent variable** $z = \{\text{blue}, \text{red}\}$

$x \sim p(x)$ $\qquad$ $x \sim p(x \mid \text{blue})$ $\qquad$ $x \sim p(x \mid \text{red})$



x $\qquad$ μ_2 $\qquad$ μ_1

Dataset $\qquad$ hard assignment blue vs red $\qquad$ Final clustering

K-means clustering as minimization problem

- Data: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_n \in \mathbb{R}^D$

$z_{nk} \in \{0,1\}$

$z_n = \begin{pmatrix} z_{n1} \\ z_{n2} \\ \vdots \\ z_{nK} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}$

- Goal: partition into K clusters by minimizing

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

- Find cluster assignments (latent var) $z_{nk} \in \{0, 1\}$

  and cluster means $\boldsymbol{\mu}_k \in \mathbb{R}^D$

- 1-hot-encoding: $z_{nk} = 1$ if and only if point n is assigned to cluster k

Minimize J (EM algorithm)

- Initialize with a random $\boldsymbol{\mu}_k \in \mathbb{R}^D$
- Repeat until convergence:
  - Find the assignment (fixed means) – **E-step**

"Expectation"

$$z_{nk} = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

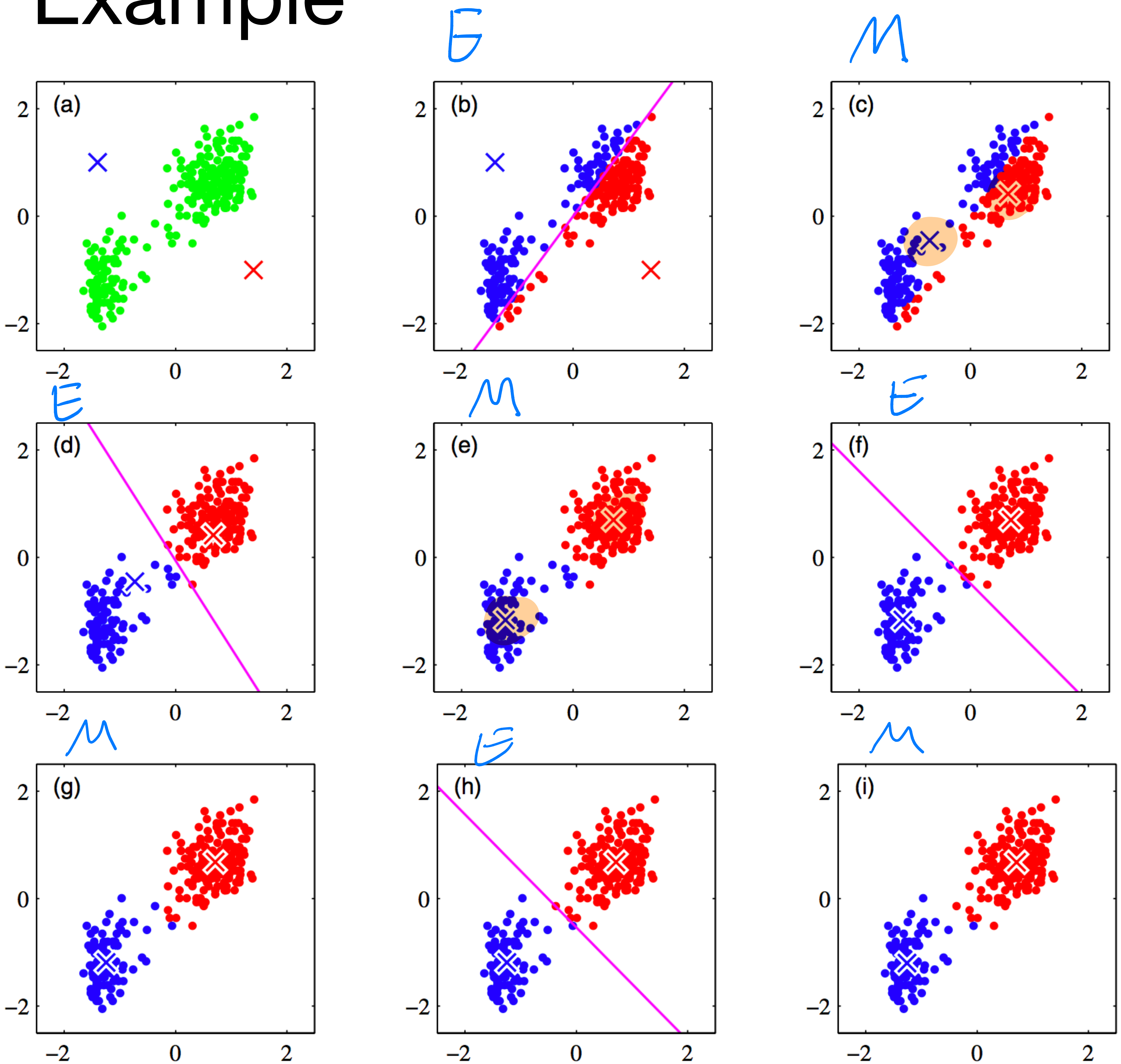
  - Find the means (fixed assignments) – **M-step**

"Maximization"

$$\boldsymbol{\mu}_k = \frac{\sum_n z_{nk} \boldsymbol{x}_n}{\sum_n z_{nk}}$$

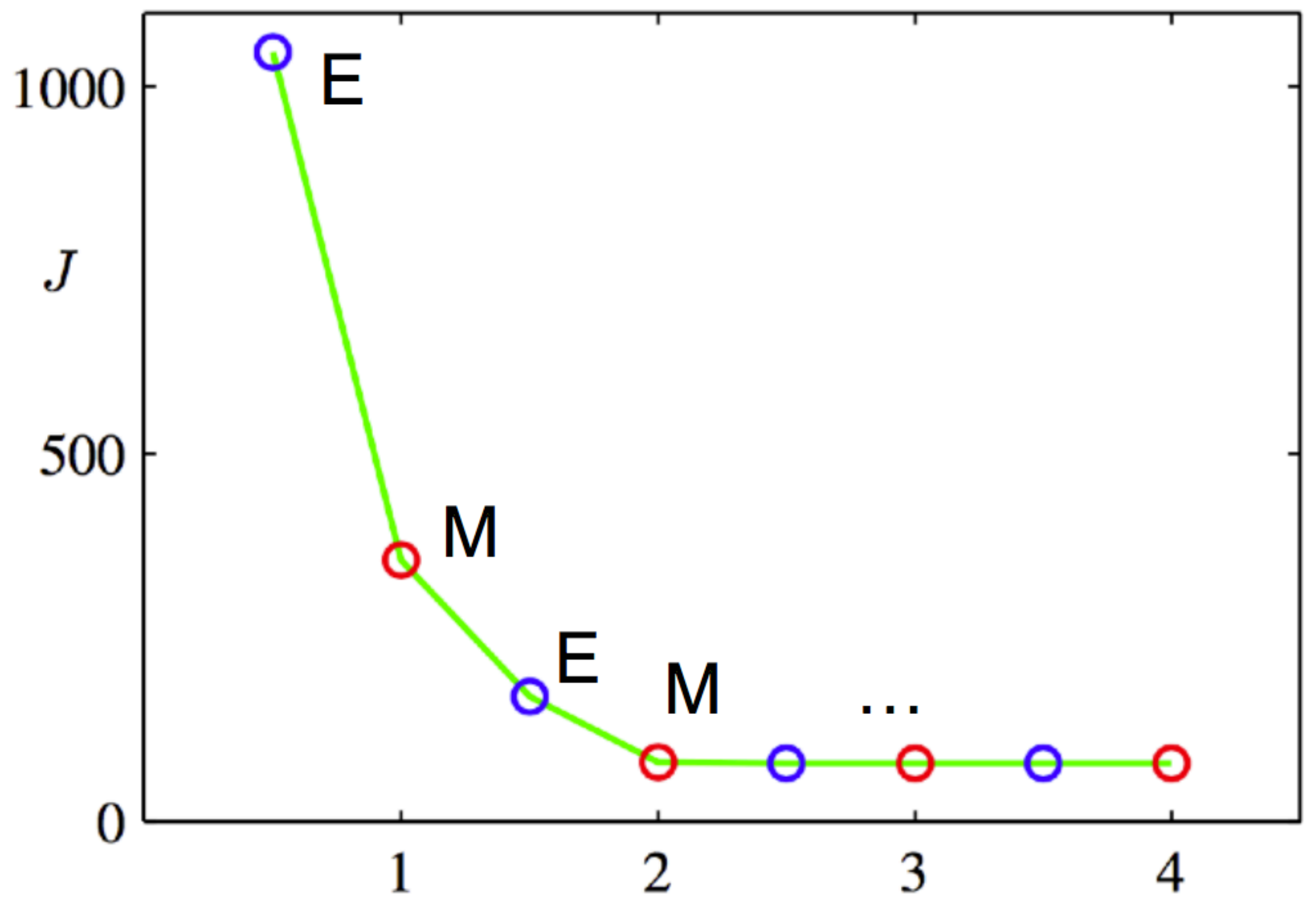
Example

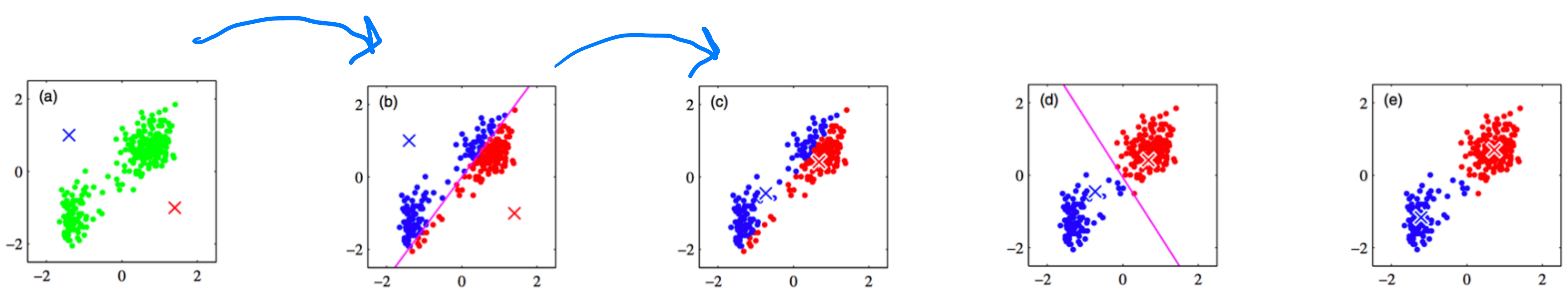
(assignment) (update means)



Convergence

$$J = \sum_n \sum_k z_{nk} \| \underline{x}_n - \underline{\mu}_k \|^2$$





Initialize E-step M-step E-step M-step ...

But ***global*** convergence?

- J is non-convex for $\boldsymbol{\mu}_k$ and z_{nk} together and k-means converges to a **local minimum**

- Can we do better? Random restarts with different initial cluster means and then select the clusters with minimal J.

Derivation of the M-step

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

- It is a convex function in $\boldsymbol{\mu}_k$ (fixed z_{nk})

$$\left(\frac{\partial}{\partial \mu_k} (x_n - \mu_k)^T (x_n - \mu_k) \right)$$

- Find the minimum by setting gradient = 0

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \left(\sum_{n=1}^{N} \sum_{l=1}^{K} z_{nl} \|\boldsymbol{x}_n - \boldsymbol{\mu}_l\|^2 \right) = \sum_{n=1}^{N} z_{nk} \frac{\partial}{\partial \boldsymbol{\mu}_k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

$\neq 0$ only when $l = k$

$$= -2 \sum_{n=1}^{N} z_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T = 0$$

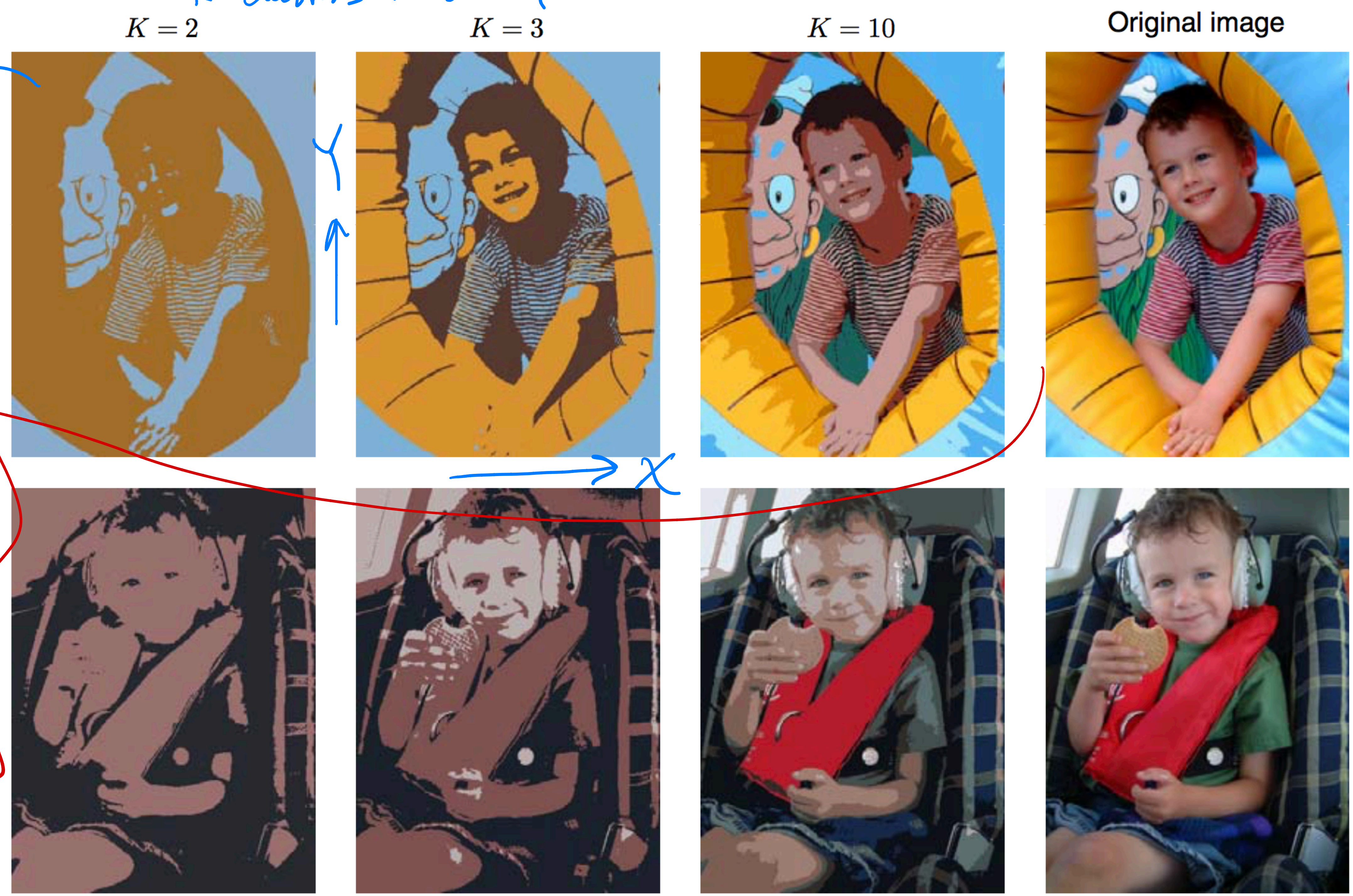
- Hence

$$\sum_{n=1}^{N} z_{nk} \boldsymbol{x}_n - \boldsymbol{\mu}_k \sum_{n=1}^{N} z_{nk} = 0 \implies \boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

Application: image compression

Data points : $x_n \in (R, G, B)$ pixel values

K clusters : color representation

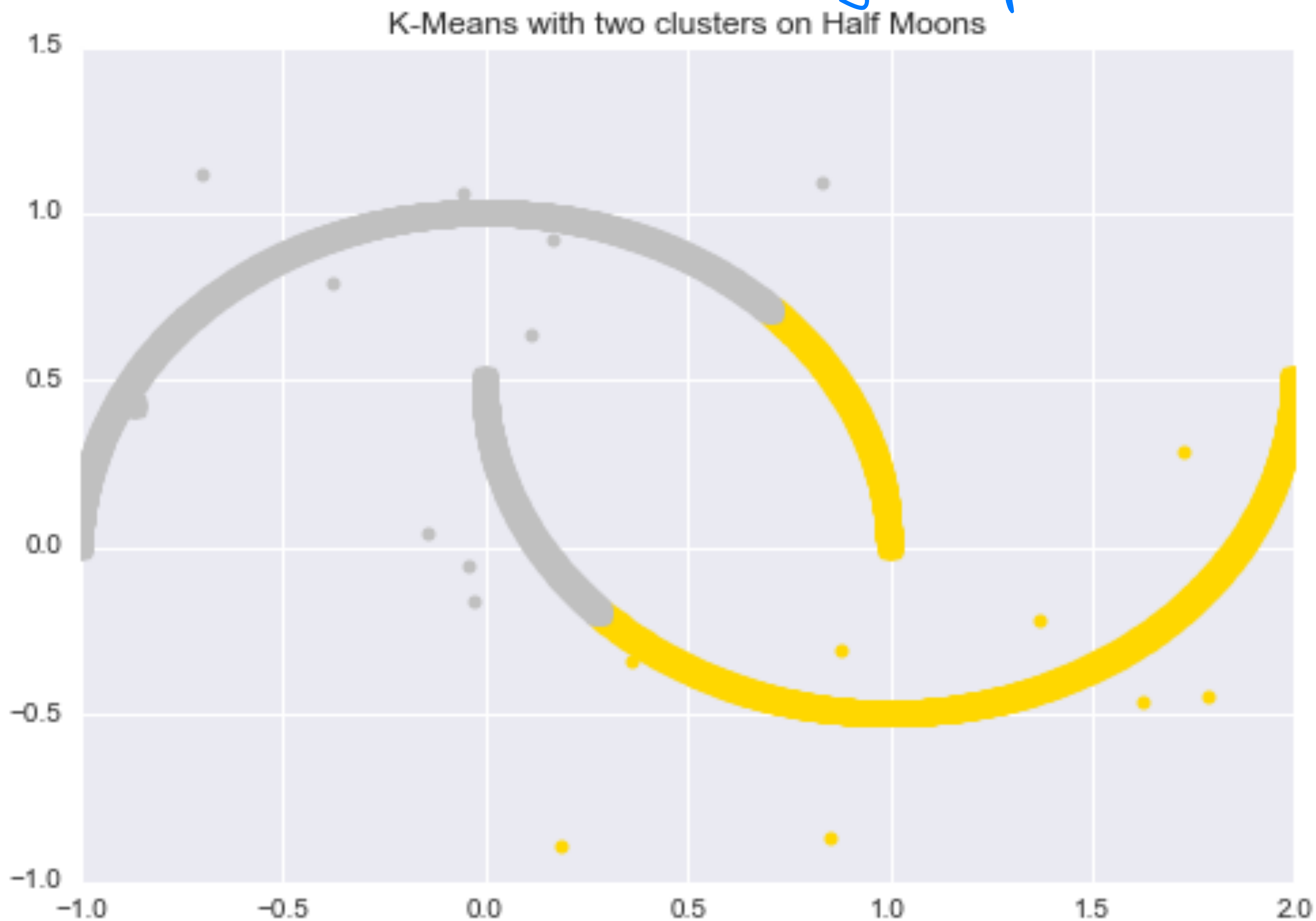


$X \cdot Y$ integers $\in \{1, 2\}$

$X \cdot Y \cdot 3$ values $\in [0, 255]$

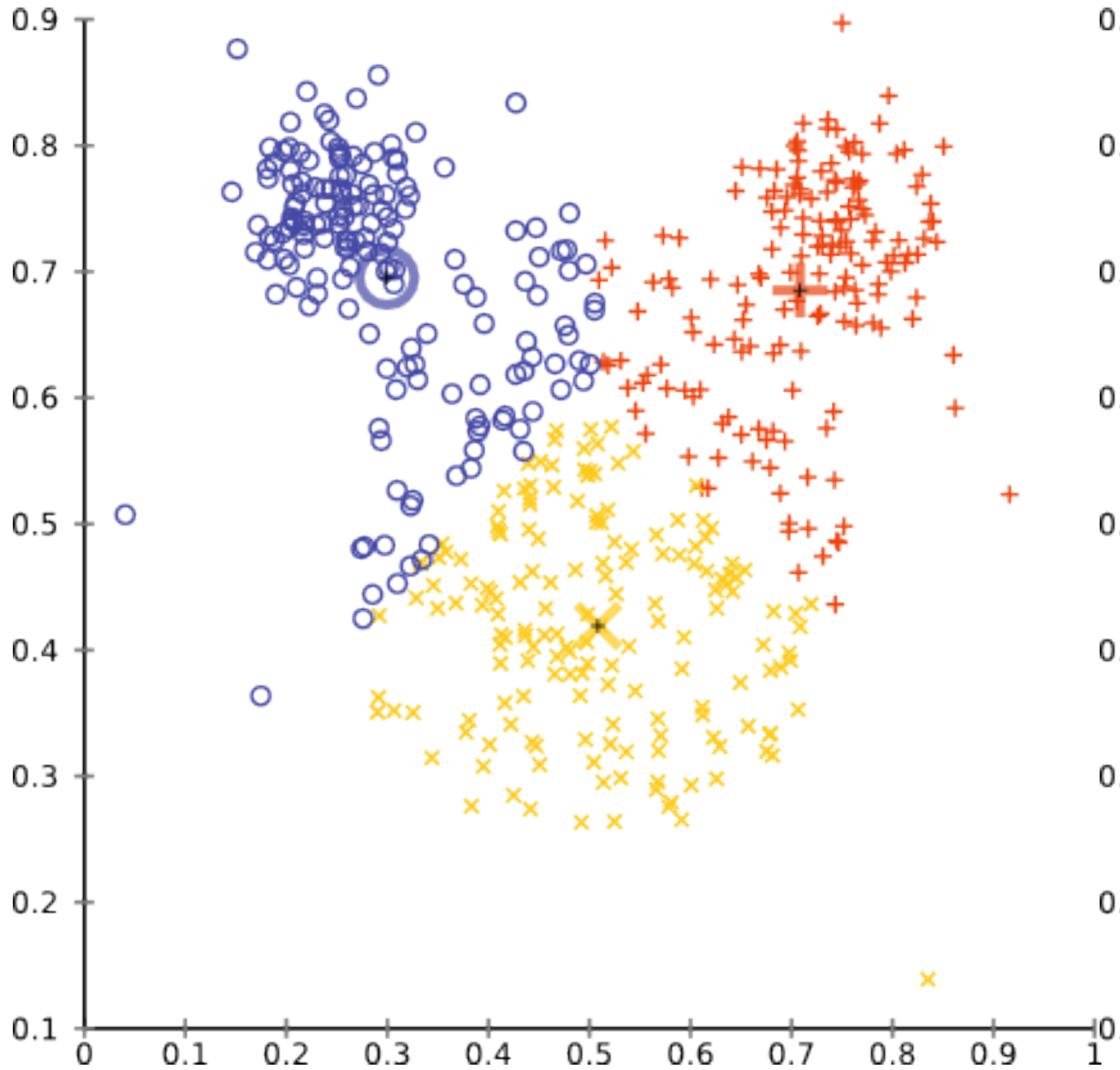
Failures of K-means

$\Longrightarrow$ only spherical clusters

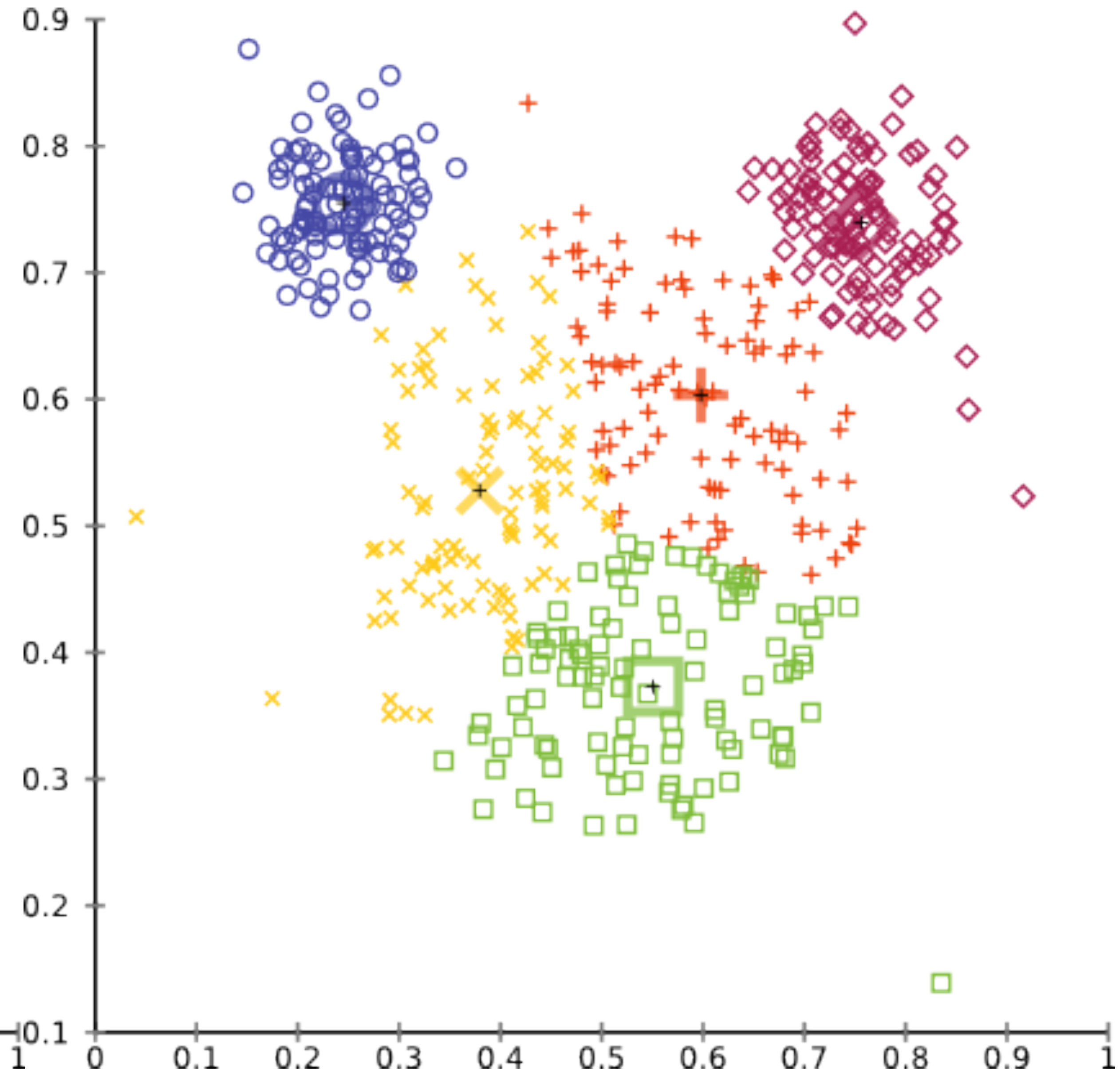


Failures of K-means (the mouse data!)

→ each cluster equal size



K = 3



K = 5

Improvements

$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \| x_n - \mu_k \|^2$

- Stochastic gradient for big data
  - For each datapoint, update nearest cluster mean:

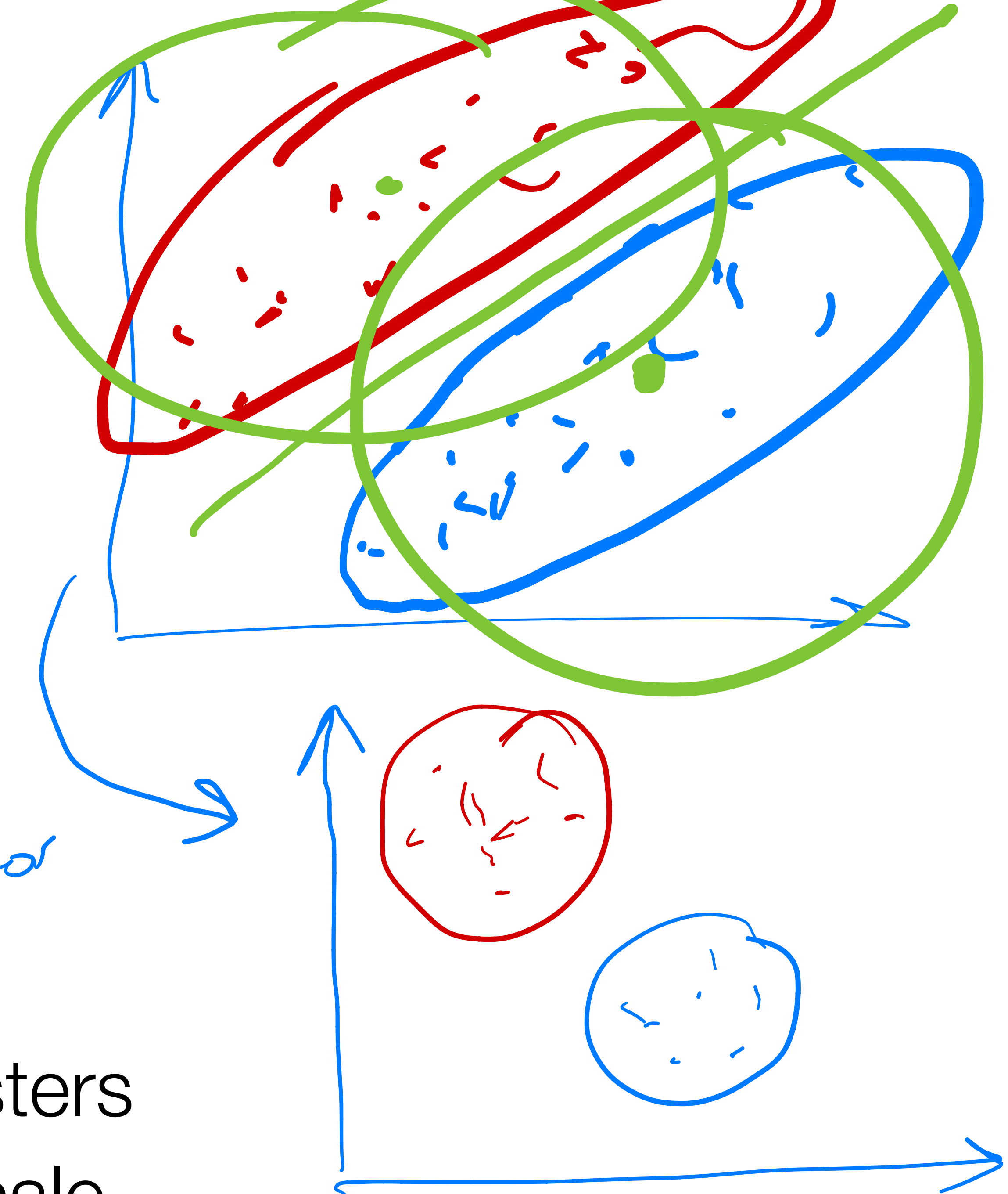
$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_k - \eta \left(\frac{\partial J}{\partial \boldsymbol{\mu}_k} \right)^T$$
$$= \boldsymbol{\mu}_k + 2\eta(\boldsymbol{x}_n - \boldsymbol{\mu}_k)$$

- Other distances between points (K- medoids)
  - Euclidean not always appropriate (discrete data), sensitive to outliers

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \mathcal{V}(\boldsymbol{x}_n, \boldsymbol{\mu}_k)$$

Pros & Cons

(*)



- Good
  - Simple to implement
  - Fast

via whitening operator

- Bad
  - Local minima
  - Model only “spherical” clusters
  - (*) Sensitive to the features scale
  - Number of clusters K to be chosen in advance
  - Cluster assignments are “hard”, not probabilistic => next topic, Gaussian Mixture Model