ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Introduction to Machine Learning

Unsupervised Learning: Clustering

Prof. Andreas Krause
Learning and Adaptive Systems (las.ethz.ch)

Unsupervised learning

- “Learning without labels”
- Typically useful for exploratory data analysis (“find patterns”; visualization; …)
- Most common methods:
  - Clustering (unsupervised classification)
  - Dimension reduction (unsupervised regression)

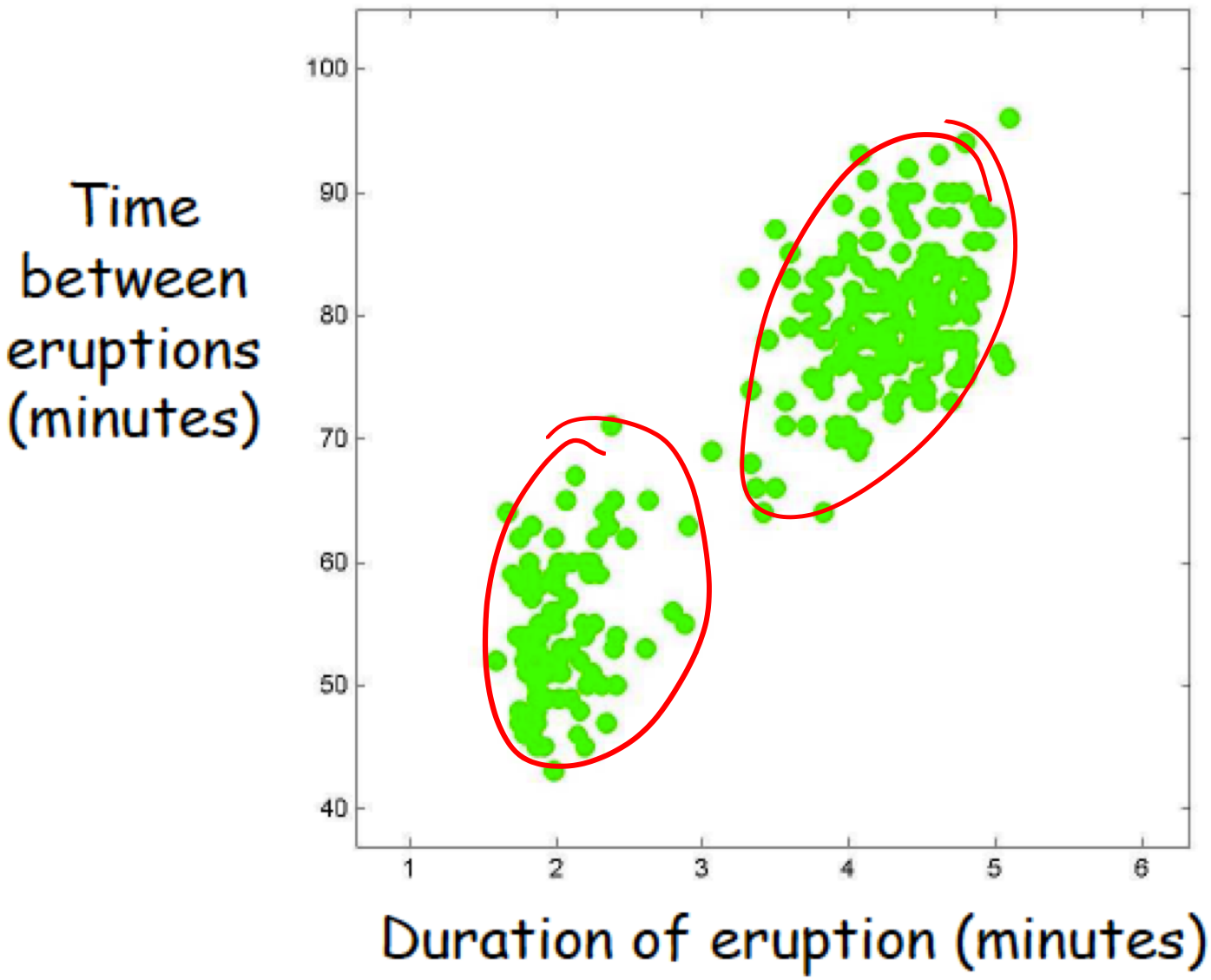
What is clustering?

- Given data points, group into clusters such that
  - *Similar* points are in the same cluster
  - *Dissimilar* points are in different clusters

- Points are typically represented either
  - in (high-dimensional) Euclidean space
  - with distances specified by a metric or kernel

- Related: Anomaly / outlier detection – Identification of points that “don’t fit well in any of the clusters”

Clustering example [Bishop]

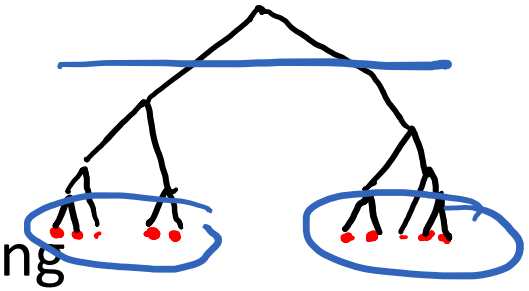


Examples of clustering tasks

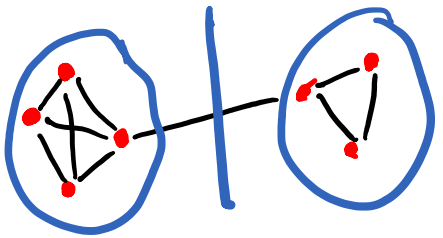
- Documents based on the words they contain
- Images based on image features
- DNA sequences based on “mutation” (edit) distance
- Products based on which customers bought them
- Customers based on their purchase history
- Web surfers based on their queries / sites they visit
- …

Standard approaches to clustering

- Hierarchical clustering
  - Build a tree (bottom-up or top-down), representing distances among data points
  - **Example**: single-, average- linkage clustering



- Partitional approaches
  - Define and optimize a notion of "cost" defined over partitions
  - **Example**: Spectral clustering, graph-cut based approaches

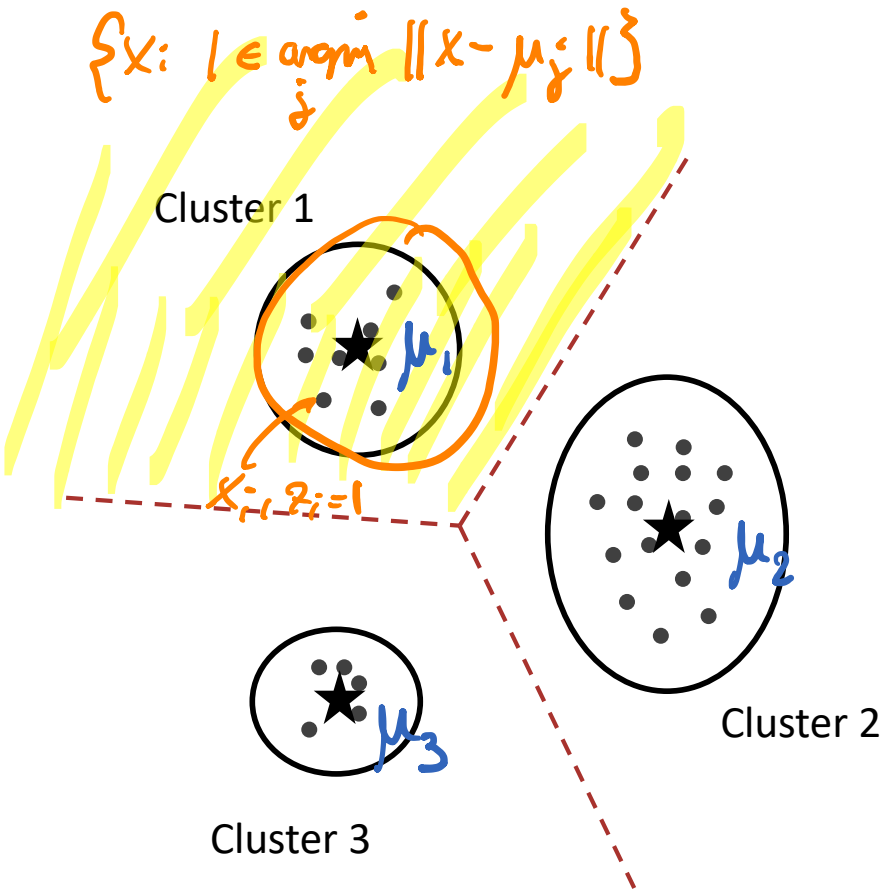


- Model-based approaches
  - Maintain cluster "models" and infer cluster membership (e.g., assign each point to closest center)
  - **Example**: k-means, Gaussian mixture models, …

We will

- Introduce basic, prototypical clustering problem:
  - k-means
- Illustrate the key challenges in unsupervised learning
- Much more details in
  - Computational Intelligence Lab
  - Statistical Learning Theory

k-Means clustering



- Represent each cluster by single point (center) $\mu_i \in \mathbb{R}^d$
- Assign points to closest center
- Induces Voronoi partition

The k-means problem

- Assumes points are in Euclidean space $\mathbf{x}_i \in \mathbb{R}^d$
- Represent clusters as centers $\mu_j \in \mathbb{R}^d$
- Each point is assigned to closest center

Goal: Pick centers to minimize average squared distance

$$\hat{R}(\mu) = \hat{R}(\mu_1, \ldots, \mu_k) = \sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} ||\mathbf{x}_i - \mu_j||_2^2$$

$d(\mu, x_i)$

$$\hat{\mu} = \arg\min_{\mu} \hat{R}(\mu)$$

- Non-convex optimization!
- NP-hard ➔ can't solve optimally in general

k-means algorithm (Lloyd's heuristic)

- Initialize cluster centers $\mu^{(0)} = [\mu_1^{(0)}, \ldots, \mu_k^{(0)}]$

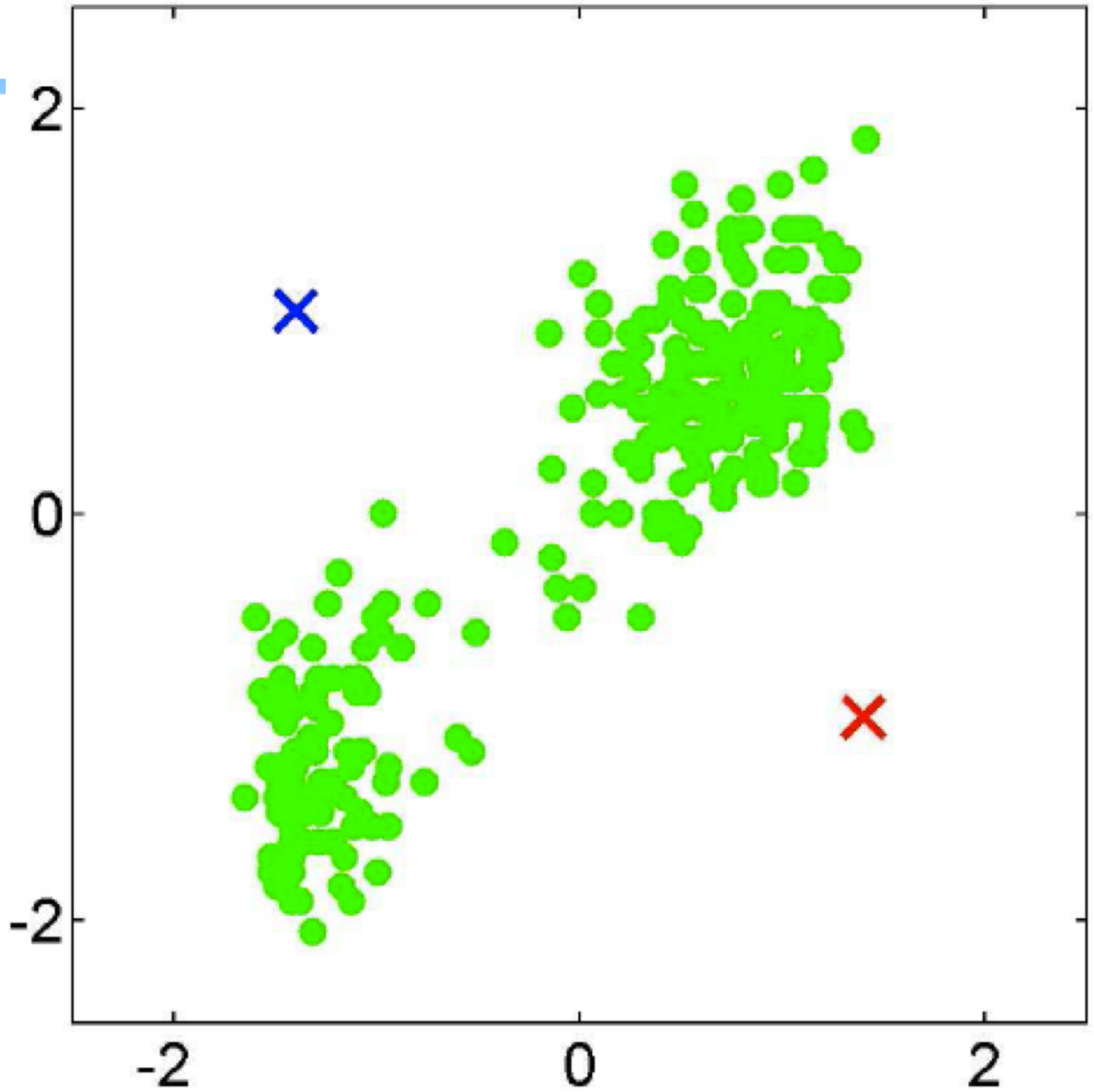
- While not converged
  - Assign each point $\boldsymbol{x}_i$ to closest center

$$z_i^{(t)} \leftarrow \arg\min_{j \in \{1,\ldots,k\}} ||\mathbf{x}_i - \mu_j^{(t-1)}||_2^2$$

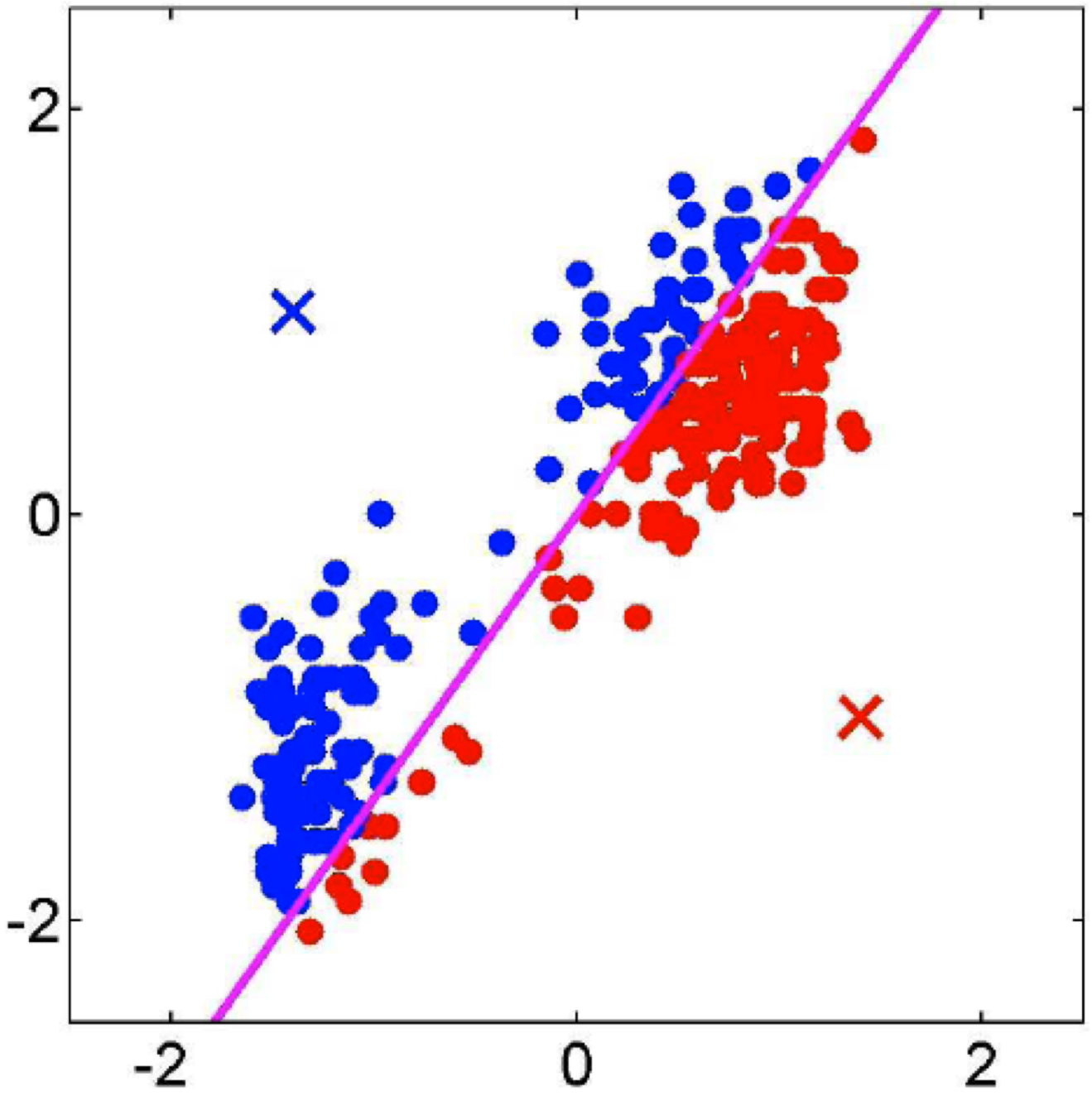
  - Update center as mean of assigned data points

$$\mu_j^{(t)} \leftarrow \frac{1}{n_j} \sum_{i: z_i^{(t)} = j} \mathbf{x}_i$$

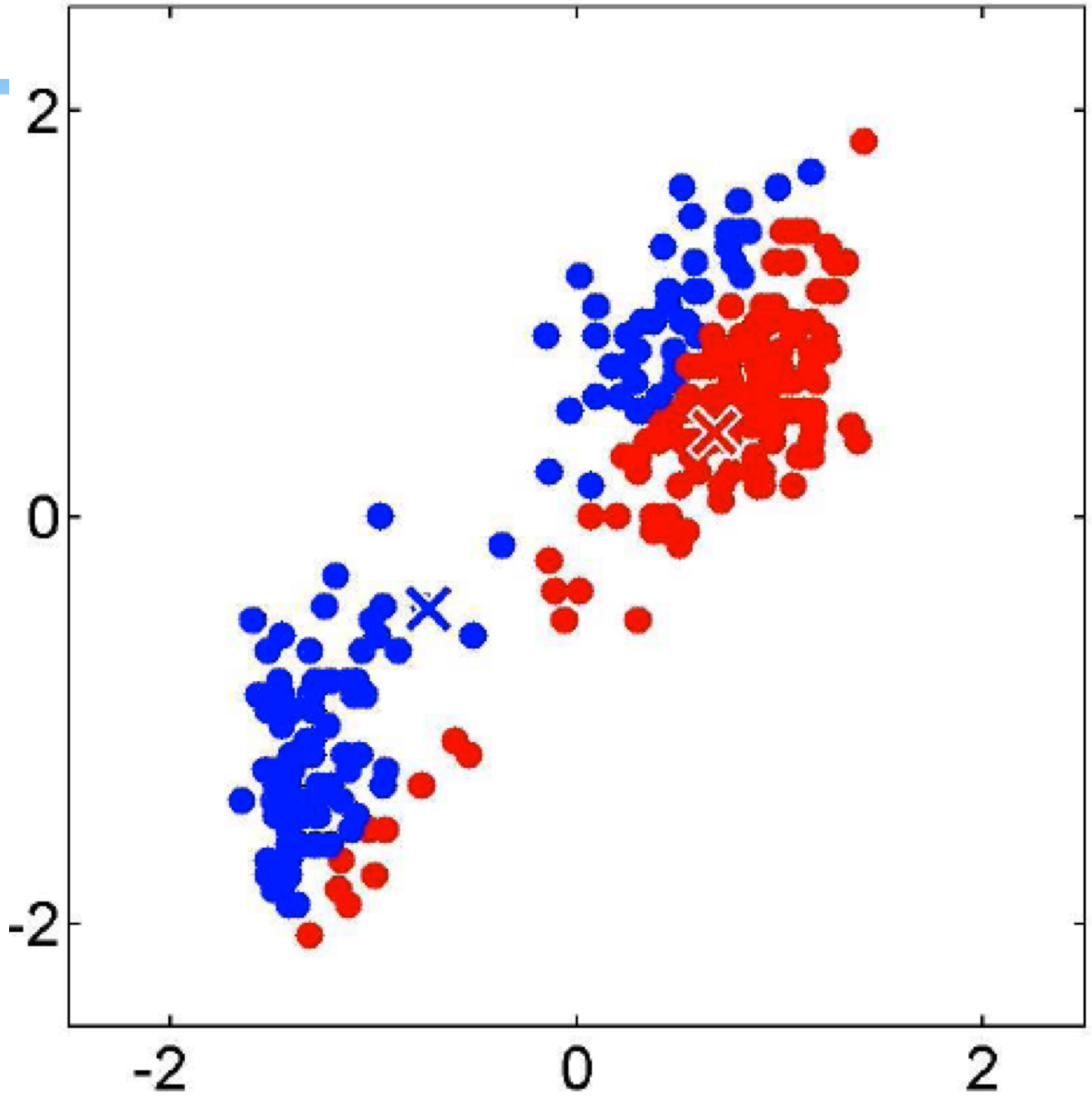
$n_j = |\{i : z_i^{(t)} = j\}|$



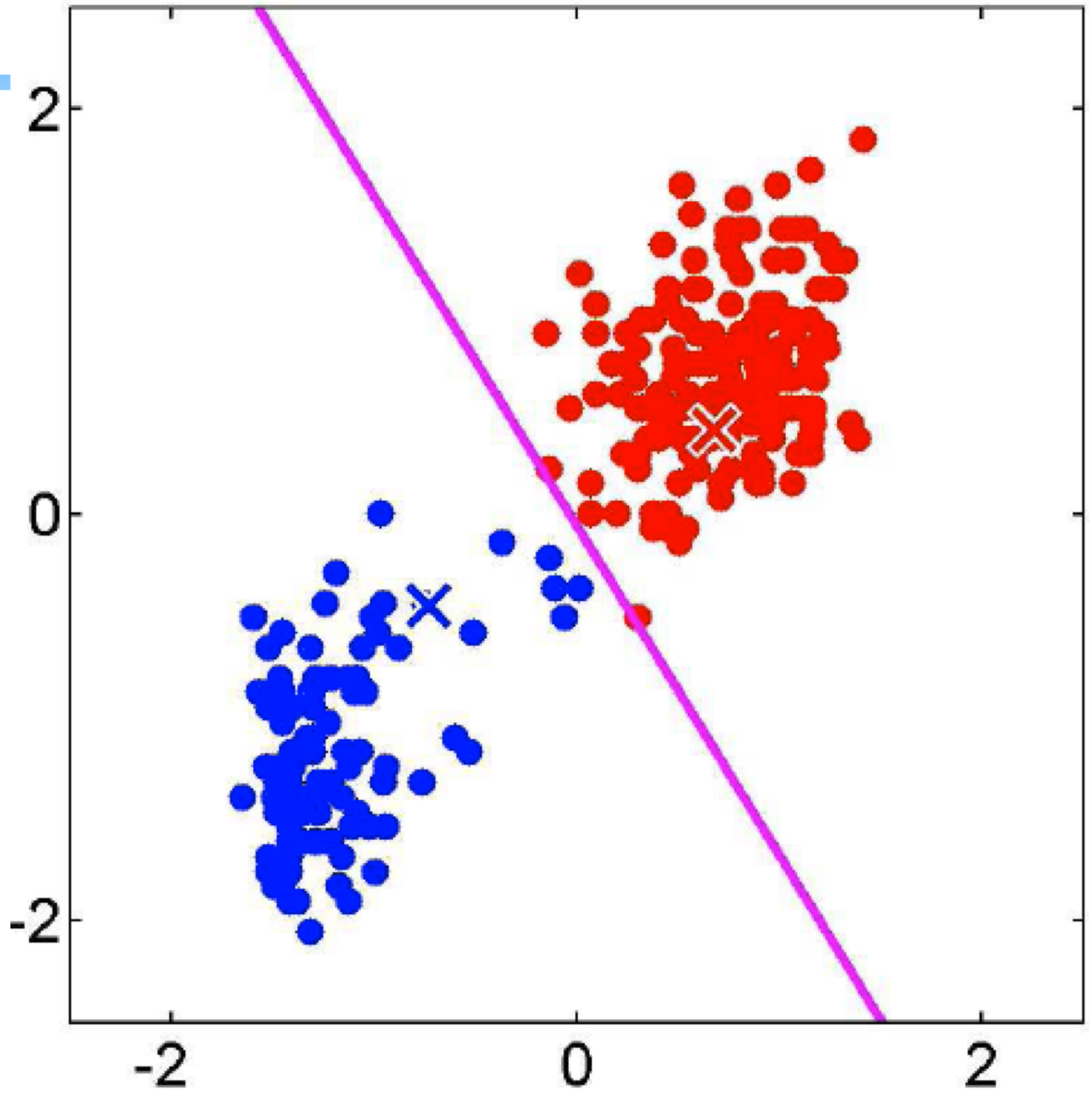
[Bishop]



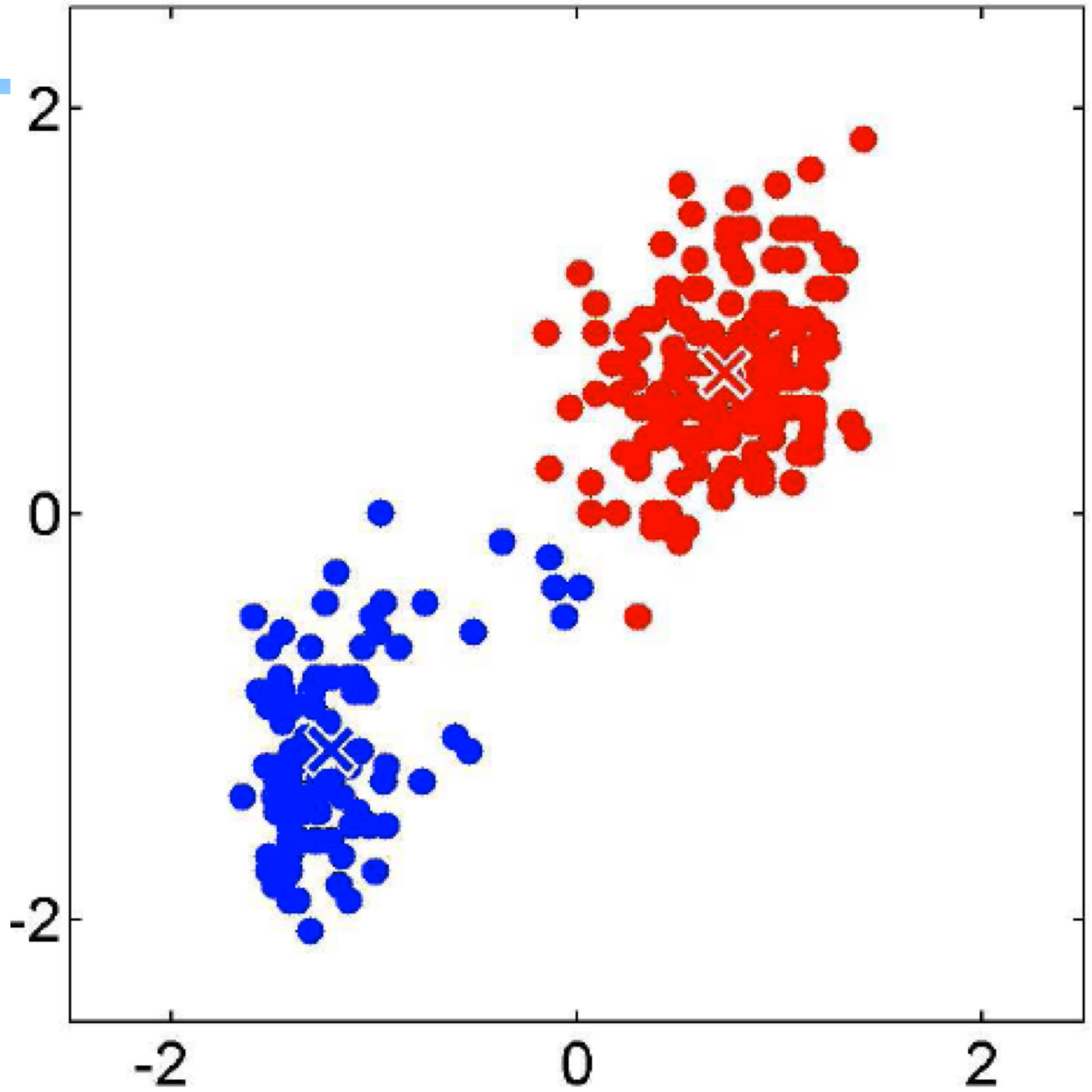
[Bishop]



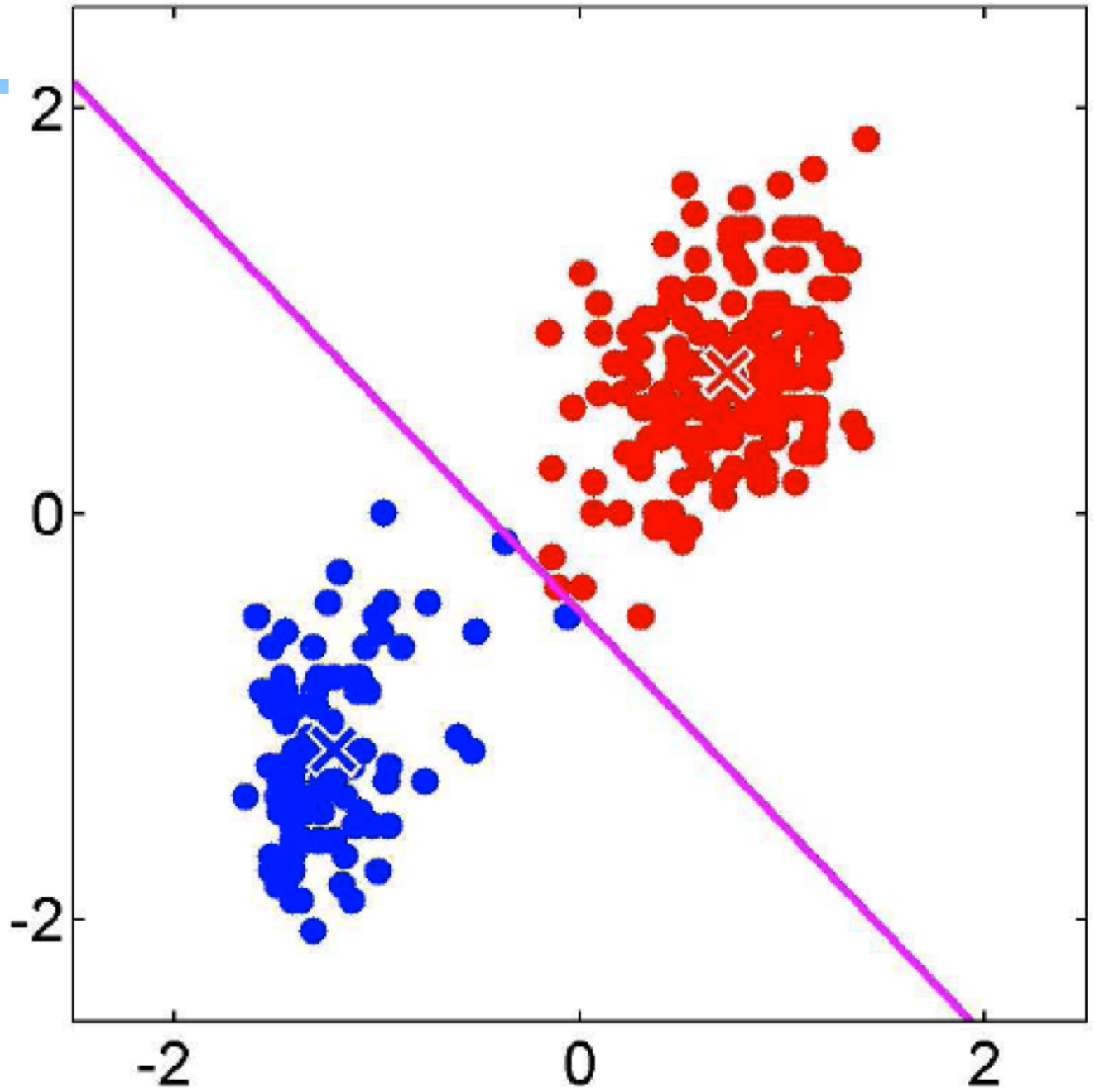
[Bishop]



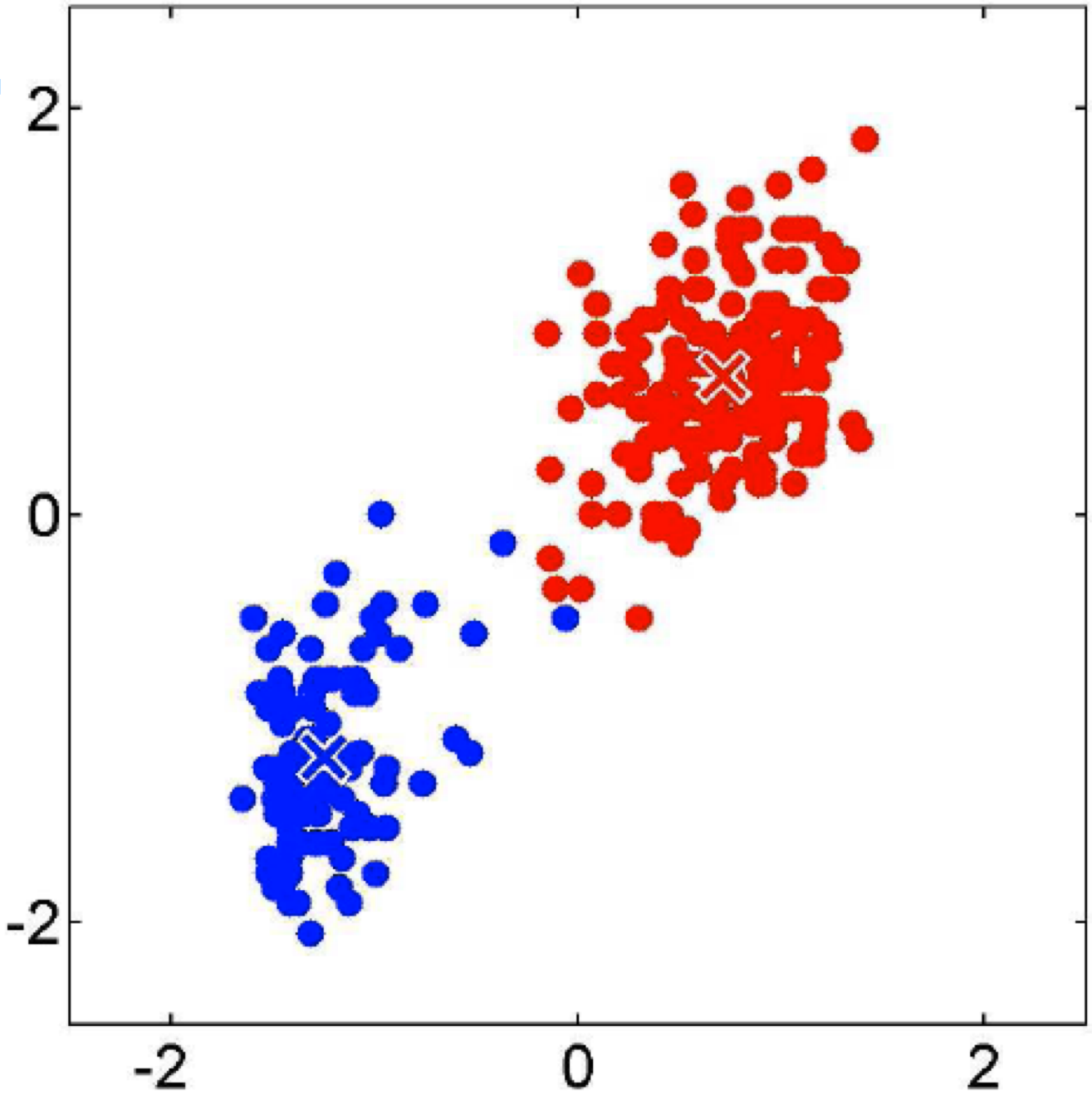
[Bishop]



[Bishop]



[Bishop]



[Bishop]

Properties of k-means

- Guaranteed to monotonically decrease average squared distance in each iteration: $\hat{R}(\mu^{(t)}) \geq \hat{R}(\mu^{(t+1)})$

$$\hat{R}(\mu^{(t)}) = \sum_{j=1}^{n} \min_{j \in \{1,\ldots,k\}} ||\mathbf{x}_i - \mu_j^{(t)}||_2^2$$

$\hat{R}(\mu) = \min_z \hat{R}(\mu, z)$

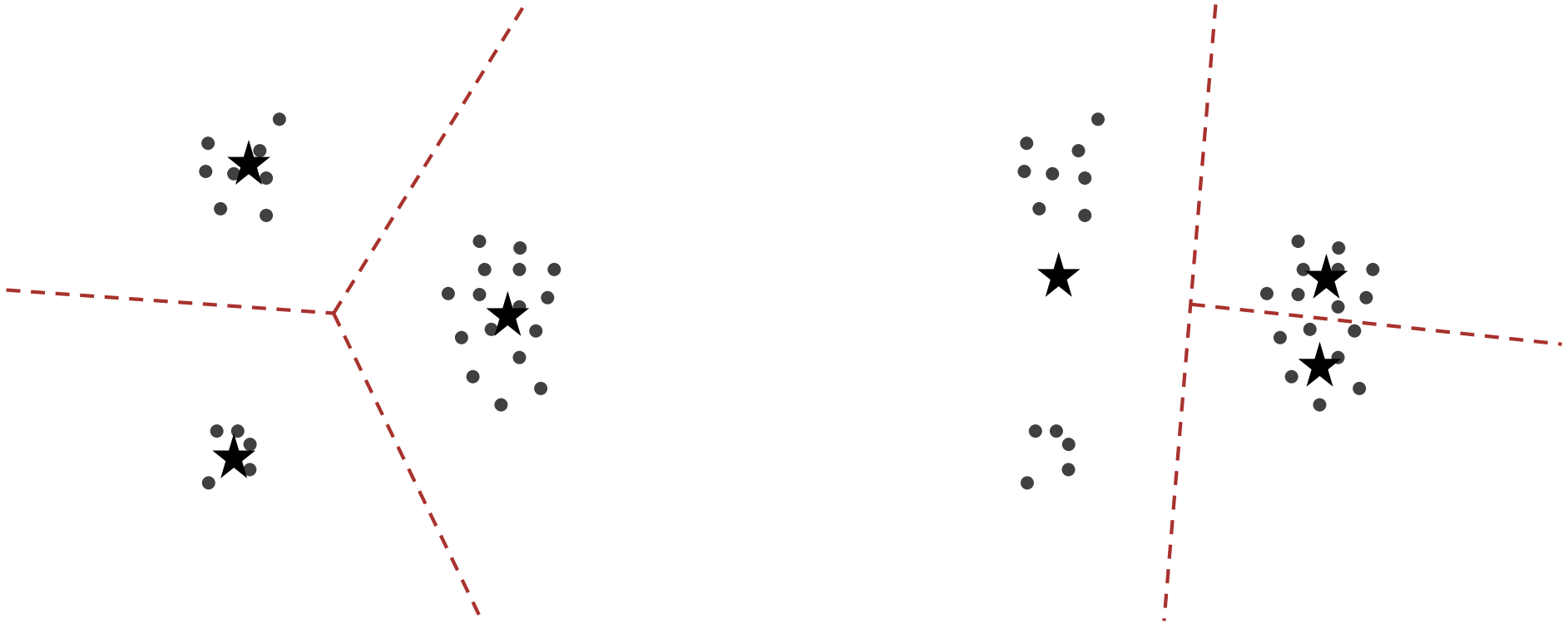
Why: $\hat{R}(\mu, z) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|_2^2$ ← distance² between x_i and cluster z_i (at μ_{z_i})

Thus: $\hat{R}(\mu^{(t)}, z^{(t)}) \geq \hat{R}(\mu^{(t)}, z^{(t+1)}) \geq \hat{R}(\mu^{(t+1)}, z^{(t+1)})$

since $z^{(t+1)} = \arg\min_z \hat{R}(\mu^{(t)}, z)$; $\mu^{(t+1)} = \arg\min_\mu \hat{R}(\mu, z^{(t+1)})$

- Converges to a local optimum
- Complexity:
  - Per iteration $O(n \cdot k \cdot d)$

Local minima



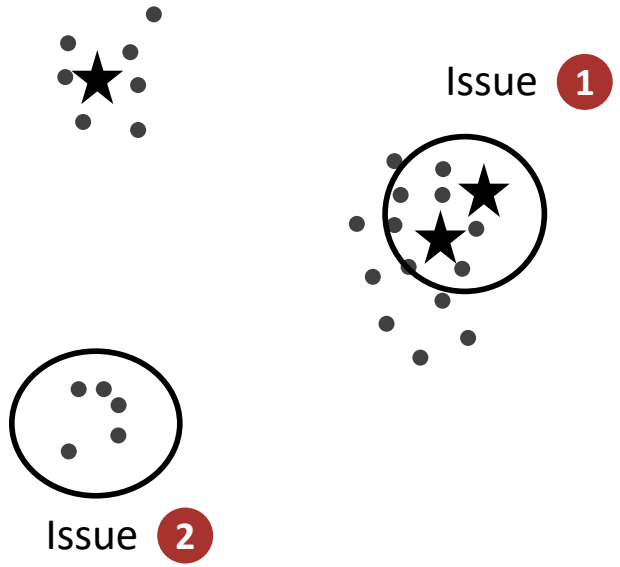
Challenges with k-Means

- Generally only converges to local optimum
  - Performance strongly dependent on initialization!
- Number of iterations required can be exponential (even in the plane!)
  - In practice however often converges very quickly
- Determining the number of clusters *k* is difficult
- Cannot well model clusters of arbitrary shape
  - (although Kernel-k-Means can be used to fix this, see later)

Initializing k-Means

- Lloyd's heuristic does not generally converge to the optimal solution
- Performance heavily depends on the initialization
- Approaches towards initialization:
  - Multiple random restarts
  - Farthest points heuristic (often works well, but prone to outliers)
  - Seeding with *k-Means++*
  - ...

How about random seeding?



K-Means++

- Start with random data point as center

$i_1 \sim \text{Uniform}(\{1..n\})$

$\mu_1^{(0)} = x_{i_1}$

- Add centers 2 to k randomly, proportionally to squared distance to closest selected center

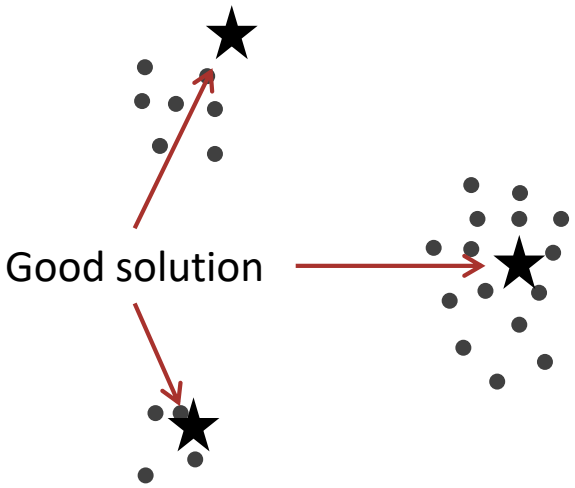
For $j = 2:k$

pick i_j with prob. $\frac{1}{Z} \cdot \underbrace{d(x_{i_j}, \mu_{1:j-1}^{(0)})}_{= \min_{\ell \in \{1..j-1\}} \|x_{i_j} - \mu_\ell^{(0)}\|_2^2}$

$\mu_j^{(0)} = x_{i_j}$

- Can show: Expected cost is *O(log k)* times that of optimal *k*-Means solution $\mathbb{E}[\hat{R}(\mu^{(0)})] \leq O(\log k) \min_\mu \hat{R}(\mu)$

Adaptive seeding using K-Means++

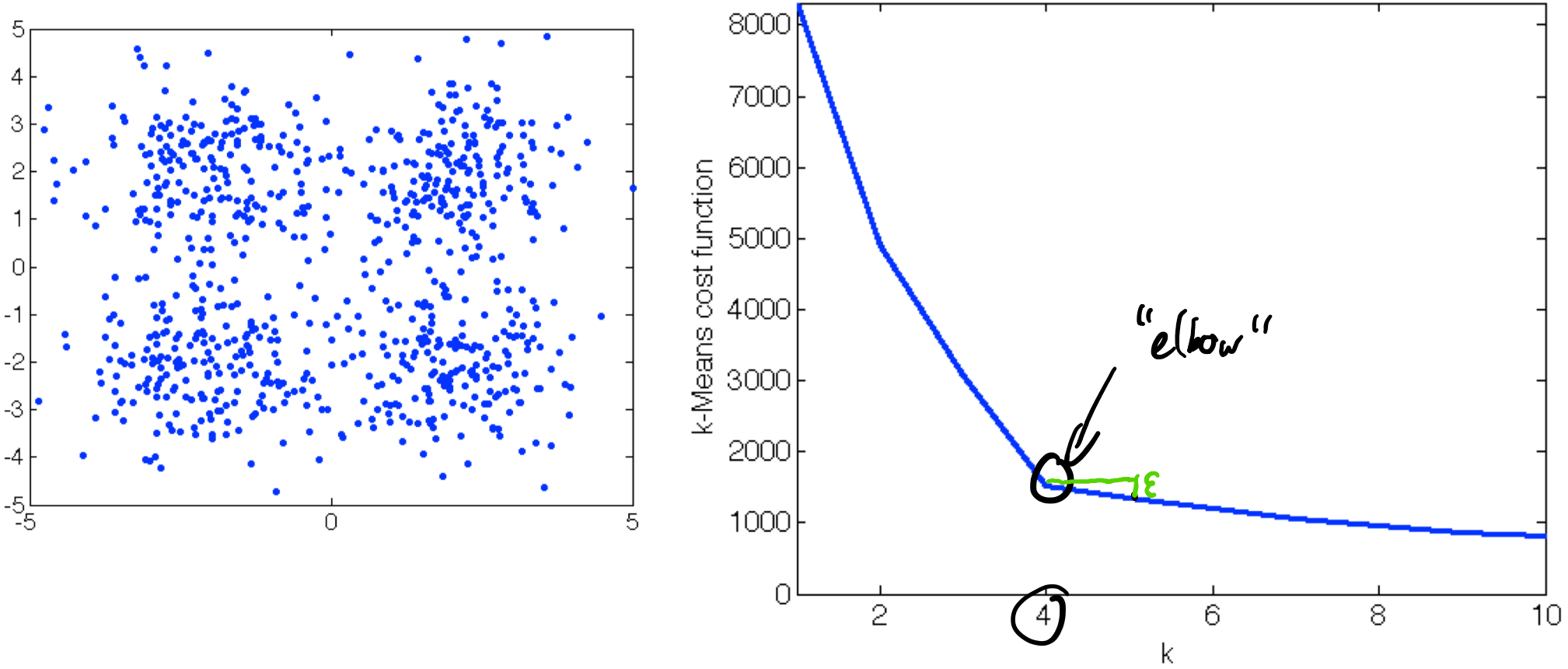


Model selection in clustering

- In general, model selection (e.g., determining the number of clusters) is very difficult

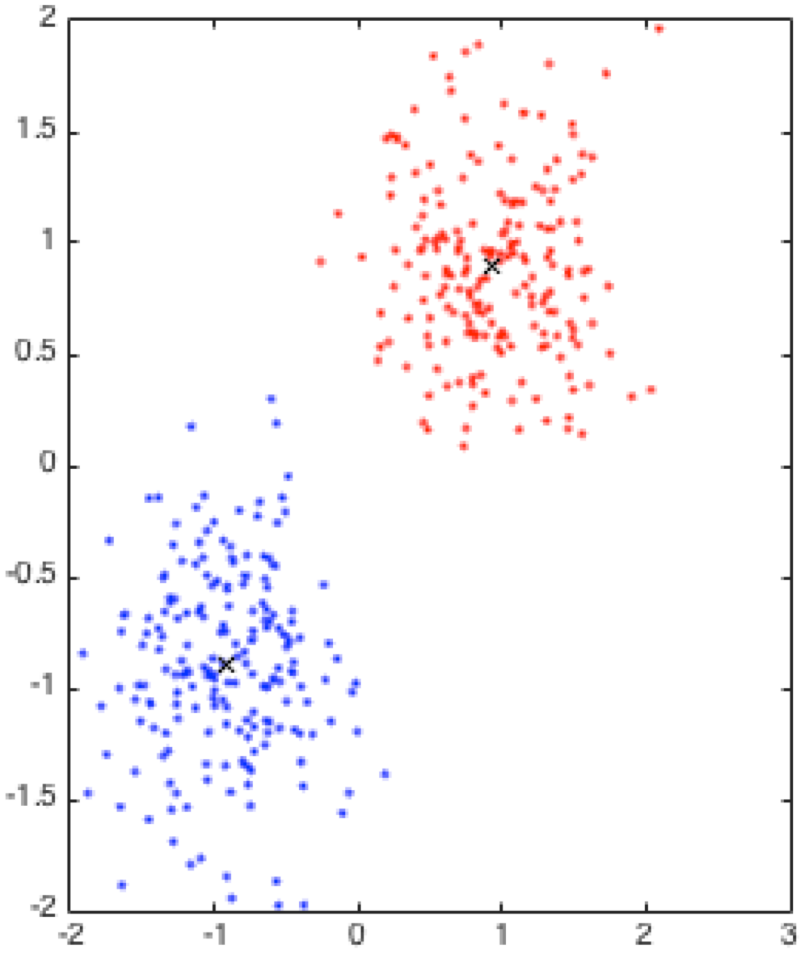
- Approaches:
  - Heuristic quality measures
  - Regularization (favor „simple“ models with few parameters by penalizing complex models)
  - Information theoretic basis (tradeoff between robustness (stability) and informativeness → Statistical Learning Theory)

Heuristic for determining *k*

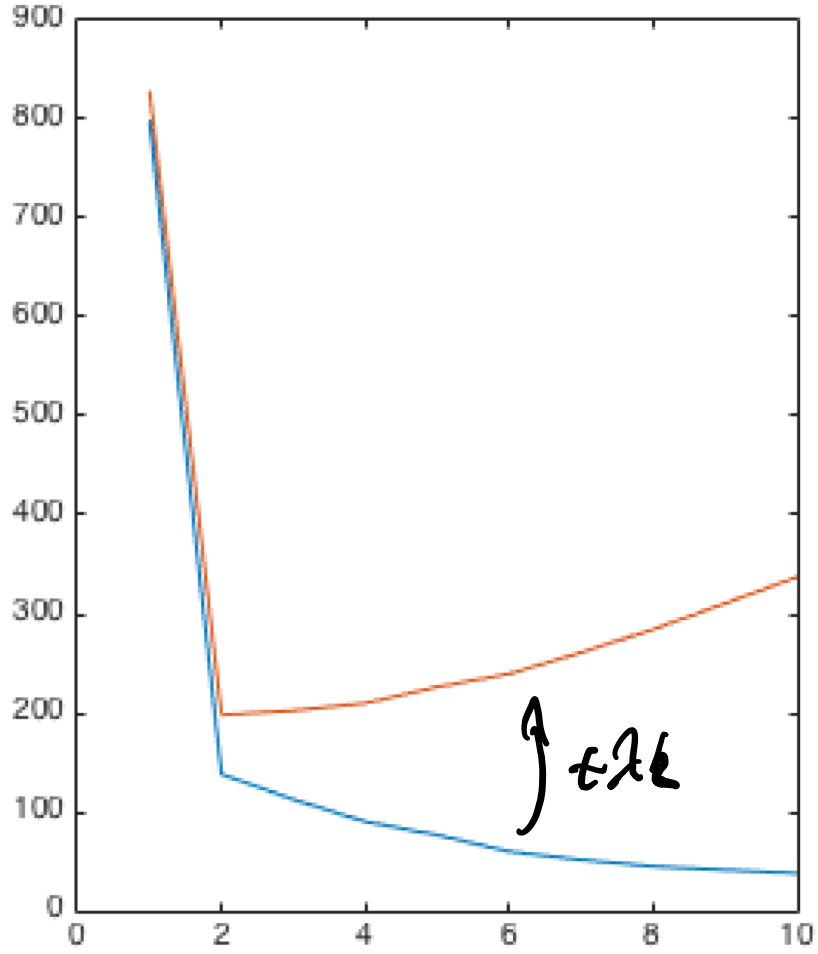


- „Diminishing returns“ in the loss function
- Pick *k* so that increasing k leads to negligible decrease in loss

Regularization

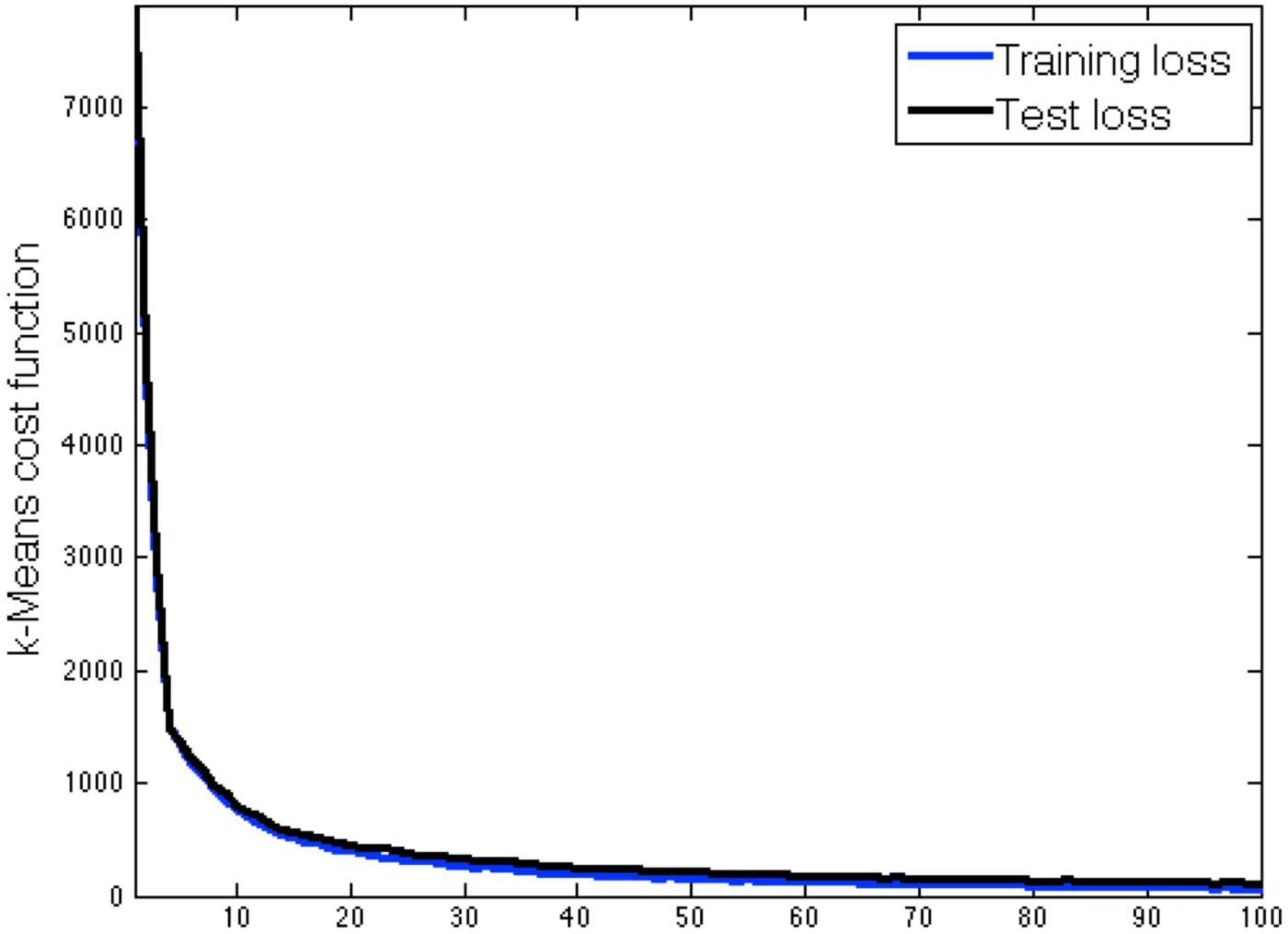


$$\min_{k,\,\mu_{1:k}} \hat{R}(\mu_{1:k}) + \lambda \cdot k$$



Equivalent to "elbow method" for $\varepsilon = 1$

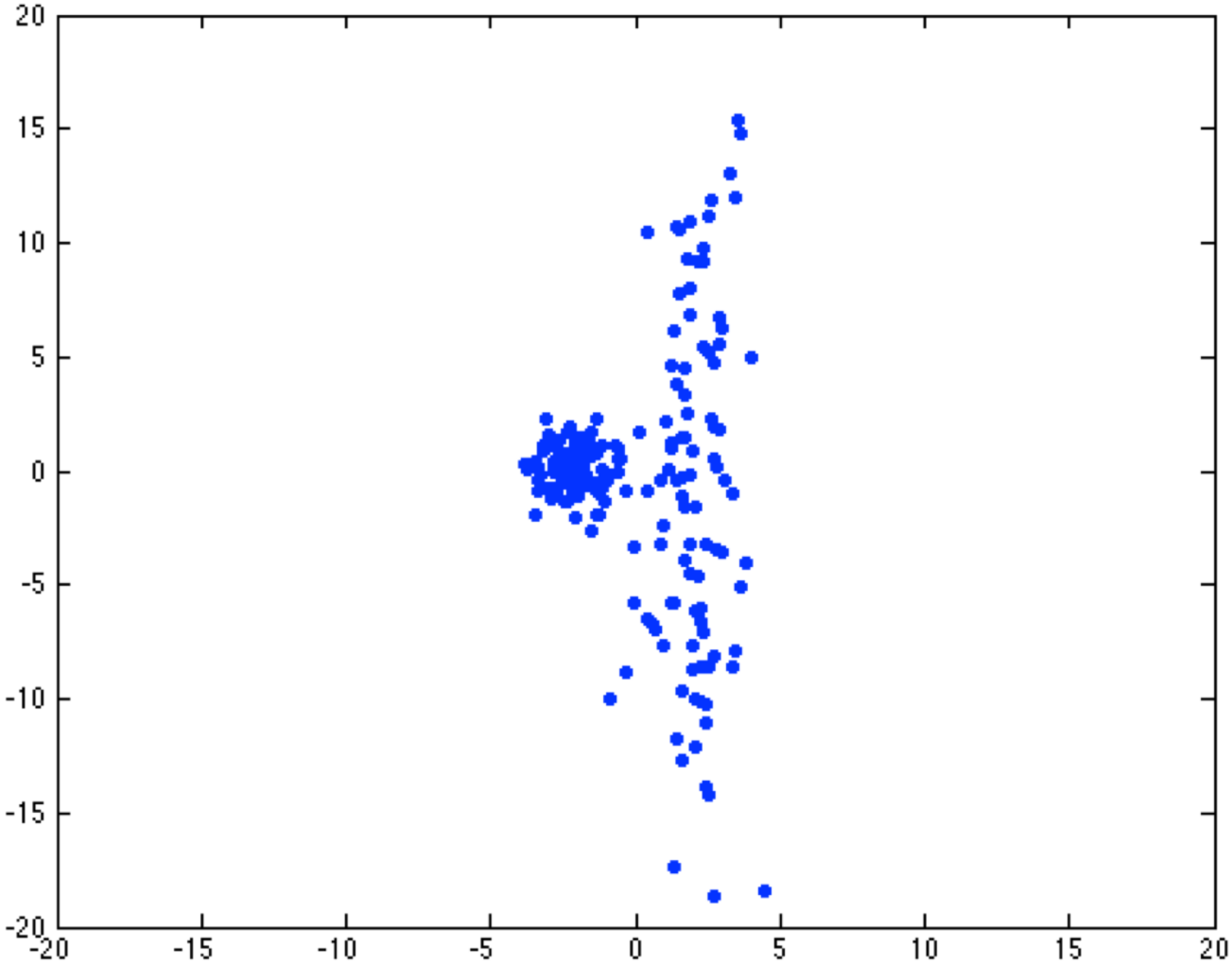
„Generalization“



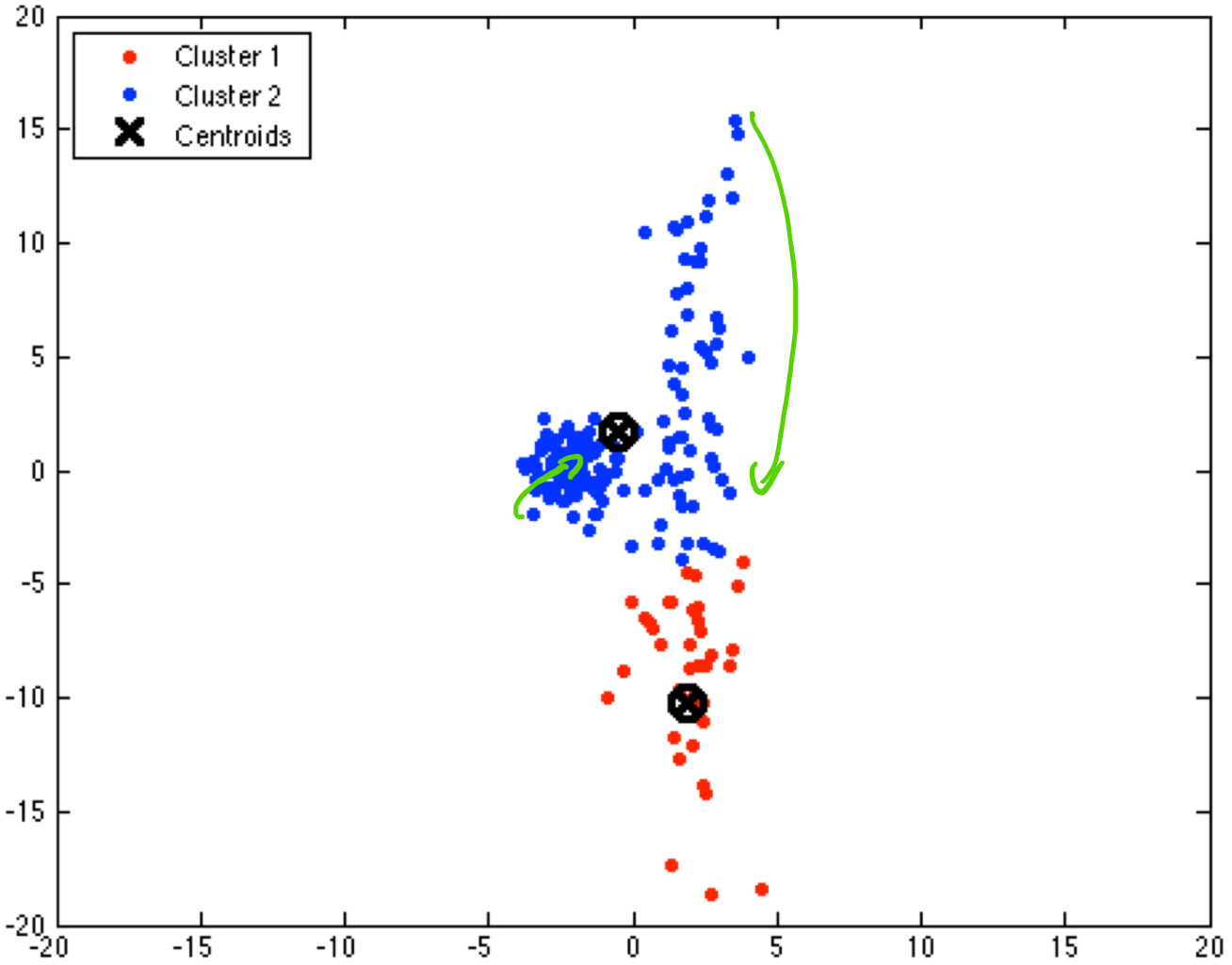
Test loss typically keeps decreasing

➔ Can‘t use cross-validation to determine #clusters

What will k-means do on this data?



Best solution found



Challenges with k-Means

- Generally only converges to local optimum
- Number of iterations required can be exponential

Practically not a big issue

- Cannot well model clusters of arbitrary shape

Can fix via kernel-k-means etc. (later)

- Determining the number of clusters *k* is difficult

Major (unsolved) practical problem!

What you need to be able to do

- Understand *k*-Means problem and apply Lloyd's algorithm for solving it
- Initialization with *k*-Means++
- Selecting *k* heuristically or via regularization
- Understand what kind of solutions *k*-means finds

Supervised AND unsupervised learning summary

Representation/ features	Linear hypotheses; nonlinear hypotheses with nonlinear feature transforms, kernels, learn nonlinear features via neural nets		
Model/ objective:	Loss-function Squared loss, 0/1 loss, Perceptron loss, Hinge loss, cost sensitive losses, multi-class hinge loss, reconstruction error	+	Regularization L^2 norm, L^1 norm, early stopping, dropout
Method:	Exact solution, Gradient Descent, (mini-batch) SGD, Reductions, Lloyd's heuristic		
Evaluation metric:	Mean squared error, Accuracy, F1 score, AUC, Confusion matrices, compression performance		
Model selection:	K-fold Cross-Validation, Monte Carlo CV		