Chapter 5: **Clustering**

Jilles Vreeken

IRDM '15/16

10 Nov 2015

UNIVERSITÄT DES SAARLANDES

CLUSTER OF EXCELLENCE

max planck institut informatik

Question of the week

How can we discover
groups of objects
that are highly similar
to each other?

Clustering, where?

Biology

- creation of phylogenies (relations between organisms)
- inferring population structures from clusterings of DNA data
- analysis of genes and cellular processes (co-clustering)

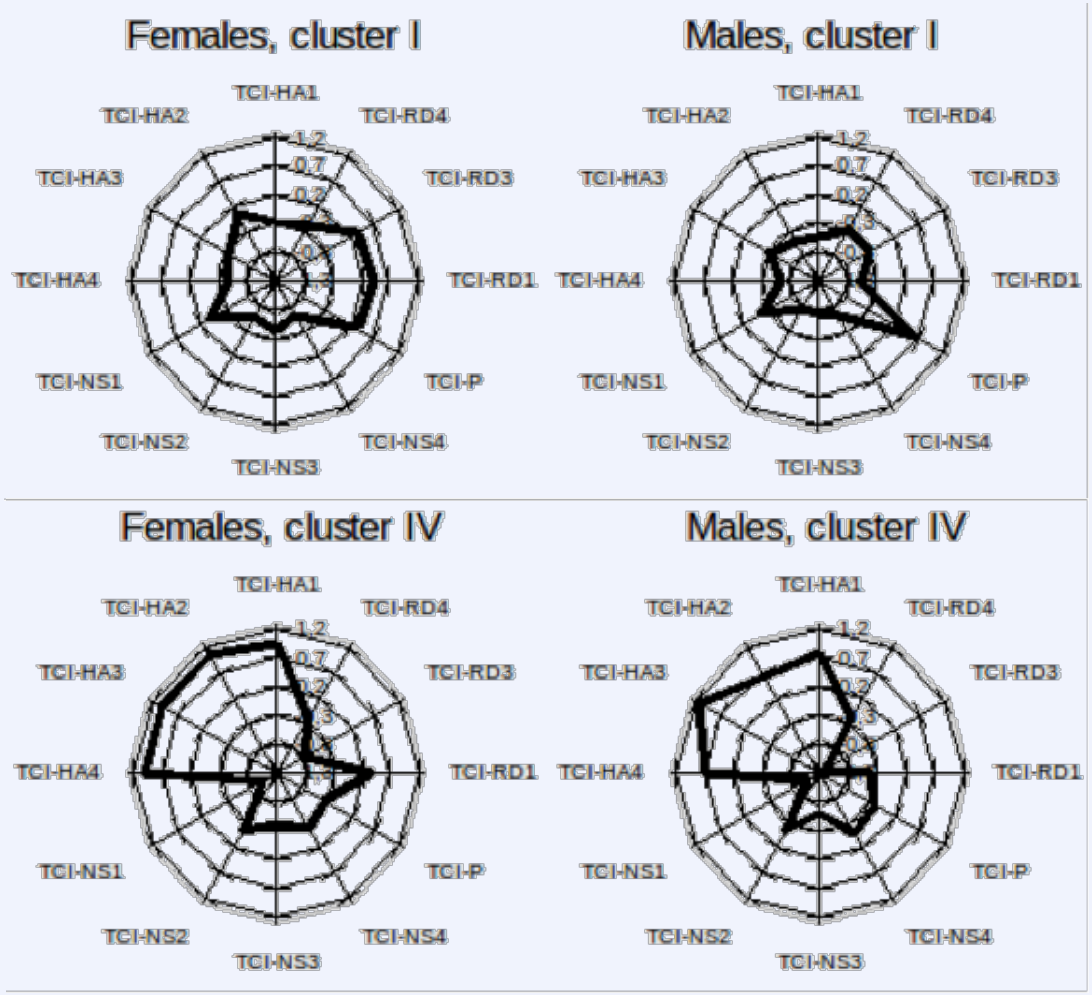
Business

- grouping of consumers into market segments

Computer science

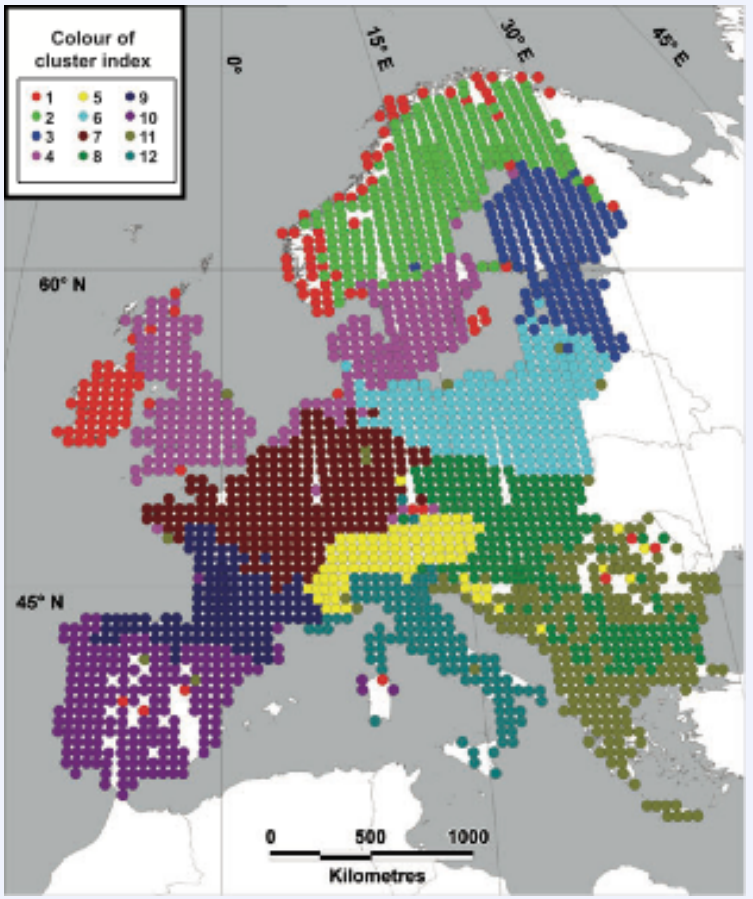
- pre-processing to reduce computation (representative-based methods)
- automatic discovery of similar items

Motivational Example



(Wessmann, 'Mixture Model Clustering in the analysis of complex diseases', 2012)

Even more motivation



(Heikinheimo et al., 'Clustering of European Mammals', 2007)

IRDM Chapter 5, overview

1. Basic idea
2. Representative-based clustering
3. Probabilistic clustering
4. Hierarchical clustering
5. Density-based clustering
6. Clustering high-dimensional data
7. Validation

You'll find this covered in
Aggarwal Ch. 6, 7
Zaki & Meira, Ch. 13—15

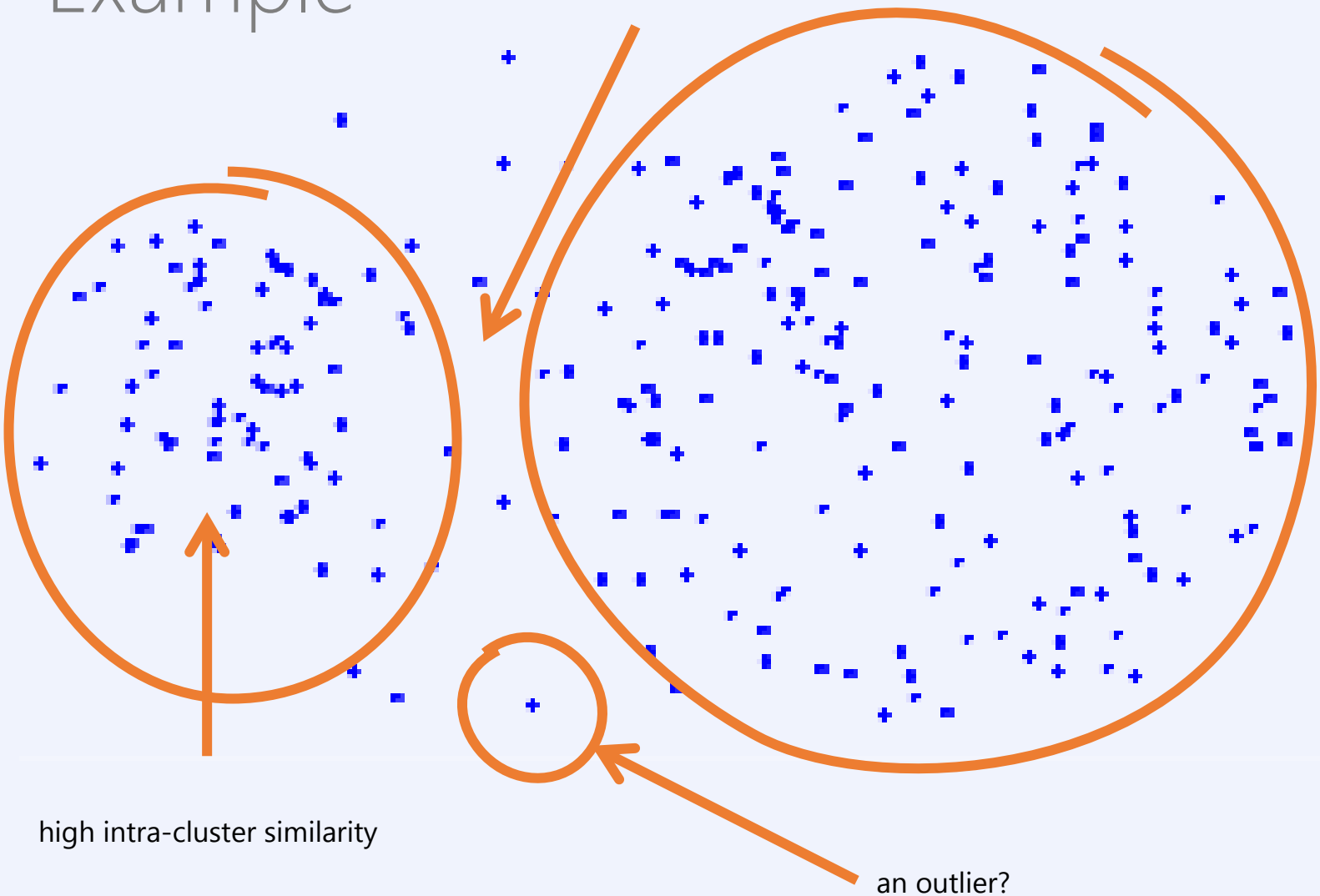
IRDM Chapter 5, today

1. Basic idea
2. Representative-based clustering
3. Probabilistic clustering
4. Hierarchical clustering
5. Density-based clustering
6. Clustering high-dimensional data
7. Validation

You'll find this covered in
Aggarwal Ch. 6, 7
Zaki & Meira, Ch. 13—15

Chapter 5.1: **Basics**

Example


low inter-cluster similarity
high intra-cluster similarity
an outlier?

The clustering problem

Given a set U of objects and a distance $d: U^2 \rightarrow R^+$ between objects, group the objects of U into **clusters** such that the distance **between points in the same cluster is low** and the distance **between the points in different clusters is large**

- **small** and **large** are not well defined
- a clustering of U can be
  - **exclusive** (each point belongs to exactly one cluster)
  - **probabilistic** (each point has a probability of belonging to a cluster)
  - **fuzzy** (each point can belong to multiple clusters)
- the number of clusters can be pre-defined, or not

On distances

A function $d: U^2 \rightarrow R^+$ is a **metric** if:

- $d(u,v) = 0$ if and only if $u = v$ — self-similarity
- $d(u,v) = d(v,u)$ for all $u, v \in U$ — symmetry
- $d(u,v) \leq d(u,w) + d(w,v)$ for all $u, v, w \in U$ — triangle-inequality

A metric is a **distance**; if $d: U^2 \rightarrow [0, \alpha]$ for some positive α then $a - d(u,v)$ is a **similarity** score

Common metrics include

- $L_p: \left(\sum_{i=1}^{d} |u_i - v_i|^p\right)^{\frac{1}{p}}$ for d-dimensional space
  - L_1 = Hamming = city-block; L_2 = Euclidean distance
- Correlation distance: $1 - \phi$
- Jaccard distance: $1 - |A \cap B| / |A \cup B|$

More distantly

For all-numerical data, the **sum of squared errors** (SSE) is the most common distance measure: $\sum_{i=1}^{d} |u_i - v_i|^2$

For all-binary data, either Hamming or Jaccard is typically used

For categorical data, we either

- first convert the data to binary by adding one binary variable per category label and then use Hamming distance; or
- count the agreements and disagreements of category labels with Jaccard

For mixed data, some combination must be used.

The distance matrix

$$\begin{pmatrix} 0 & d_{1,2} & d_{1,3} & & d_{1,n} \\ d_{1,2} & 0 & d_{2,3} & \cdots & d_{2,n} \\ d_{1,3} & d_{2,3} & 0 & & d_{3,n} \\ & \vdots & & \ddots & \vdots \\ d_{1,n} & d_{2,n} & d_{3,n} & \cdots & 0 \end{pmatrix}$$

A **distance** (or **dissimilarity**) **matrix** is

- n-by-n for n objects
- non-negative ($d_{i,j} \geq 0$)
- symmetric ($d_{i,j} = d_{j,i}$)
- Zero on diagonal ($d_{i,i} = 0$)

Chapter 5.2: Representative-based Clustering

Aggarwal Ch. 6.3

Partitions and Prototypes

Exclusive representative-based clustering

- the set of objects U is **partitioned** into k clusters $C_1, C_2, \ldots, C_k$
  - $\cup_i C_i = U$ and $C_i \cap C_j = \emptyset$ for $i \neq j$
- every cluster is **represented** by a prototype (aka centroid or mean) μ_i
- clustering quality is based on **sum of squared errors** between objects in a cluster and the cluster prototype

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2 = \sum_{i=1}^{k} \sum_{x_j \in C_i} \sum_{l=1}^{d} (x_{jl} - \mu_{il})^2$$

Partitions and Prototypes

Exclusive representative-based clustering

- the set of objects U is **partitioned** into k clusters $C_1, C_2, \ldots, C_k$
  - $\cup_i C_i = U$ and $C_i \cap C_j = \emptyset$ for $i \neq j$
- every cluster is **represented** by a prototype (aka centroid or mean) μ_i
- clustering quality is ba objects in a cluster and the cluster prototype

over all objects in the cluster

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2 = \sum_{i=1}^{k} \sum_{x_j \in C_i} \sum_{l=1}^{d} (x_{jl} - \mu_{il})^2$$

over all clusters

over all dimensions

The Naïve algorithm

The naïve algorithm goes like this

- one by one generate all possible clusterings
- compute the squared error
- select the best

Sadly, this is infeasible

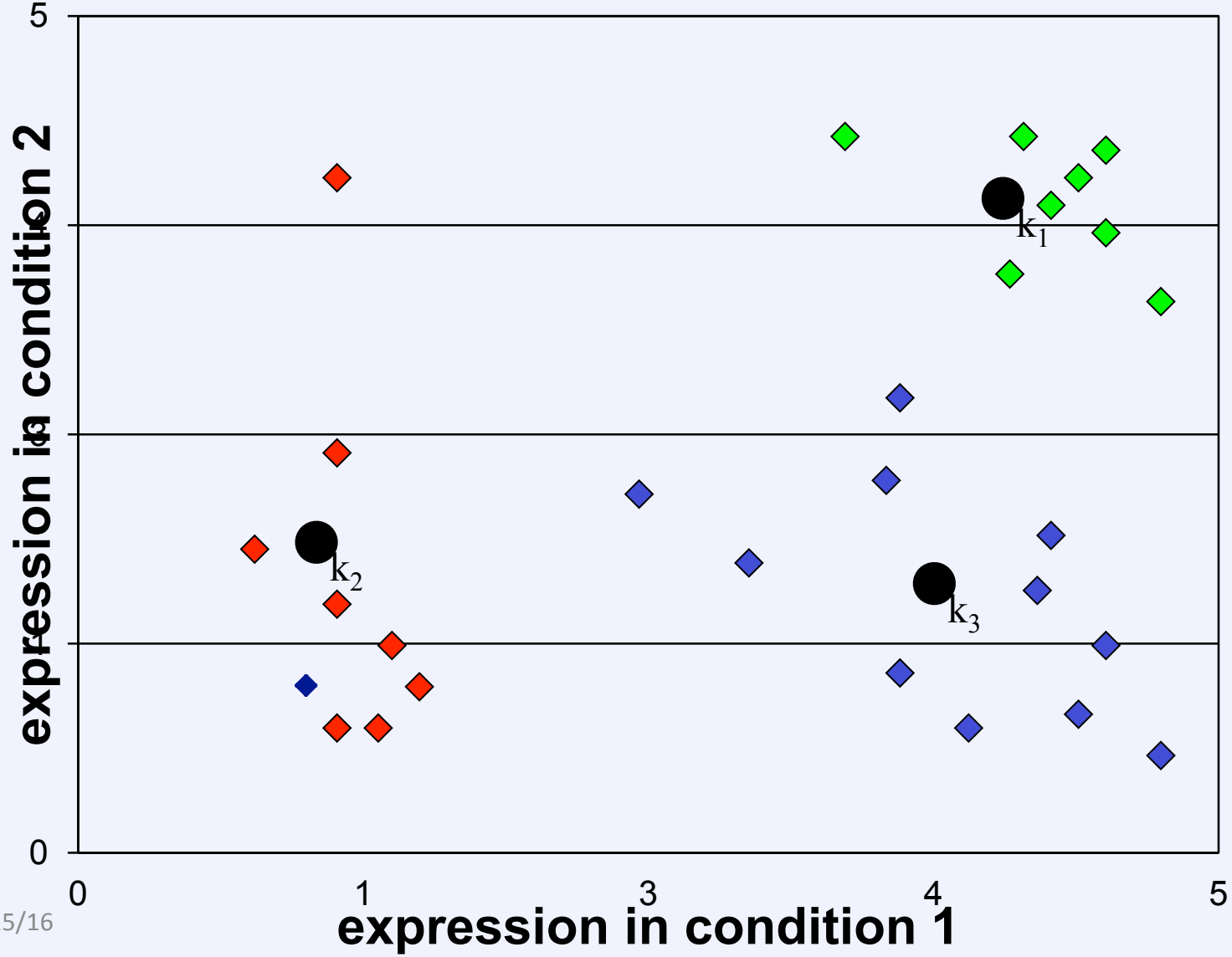
- there are too many possible clusterings to try
  - k^n different clusterings to k clusters (some possibly empty)
  - the number of ways to cluster n points in k non-empty clusters is the Stirling number of the second kind, $S(n,k)$,

$$S(n,k) = \left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \binom{k}{j} (k-j)^n$$

An iterative k-means algorithm

1. select k random cluster centroids
2. assign each point to its closest centroid
3. compute the error
4. **do**
    1. **for each** cluster C_i
        1. compute new centroid as $\mu_i = \frac{1}{|C_i|}\sum_{x_j \in C_i} x_j$
    2. **for each** element $x_j \in U$
        1. assign x_j to its closest cluster centroid
5. **while** error decreases

k-means Example



Some observations

Always converges, **eventually**

- on each step the error decreases
- only finite number of possible clusterings
- convergence to local optimum

At some point a cluster can become **empty**

- all points are closer to some other centroid
- some options include
  - split the biggest cluster
  - take the furthest point as a singleton cluster

Outliers can yield bad clusterings

Computational complexity

How long does iterative k-means take?

- computing the centroid takes $O(nd)$ time
  - averages over total of n points in d-dimensional space
- computing the cluster assignment takes $O(nkd)$ time
  - for each n points we have to compute the distances to all k clusters in d-dimensional space
- if the algorithm takes t iterations, the total running time is $O(tnkd)$
- how many iterations will we need?

How many iterations?

In practice the algorithm usually doesn't need many

- some hundred iterations is usually enough

Worst-case upper bound is $O(n^{dk})$

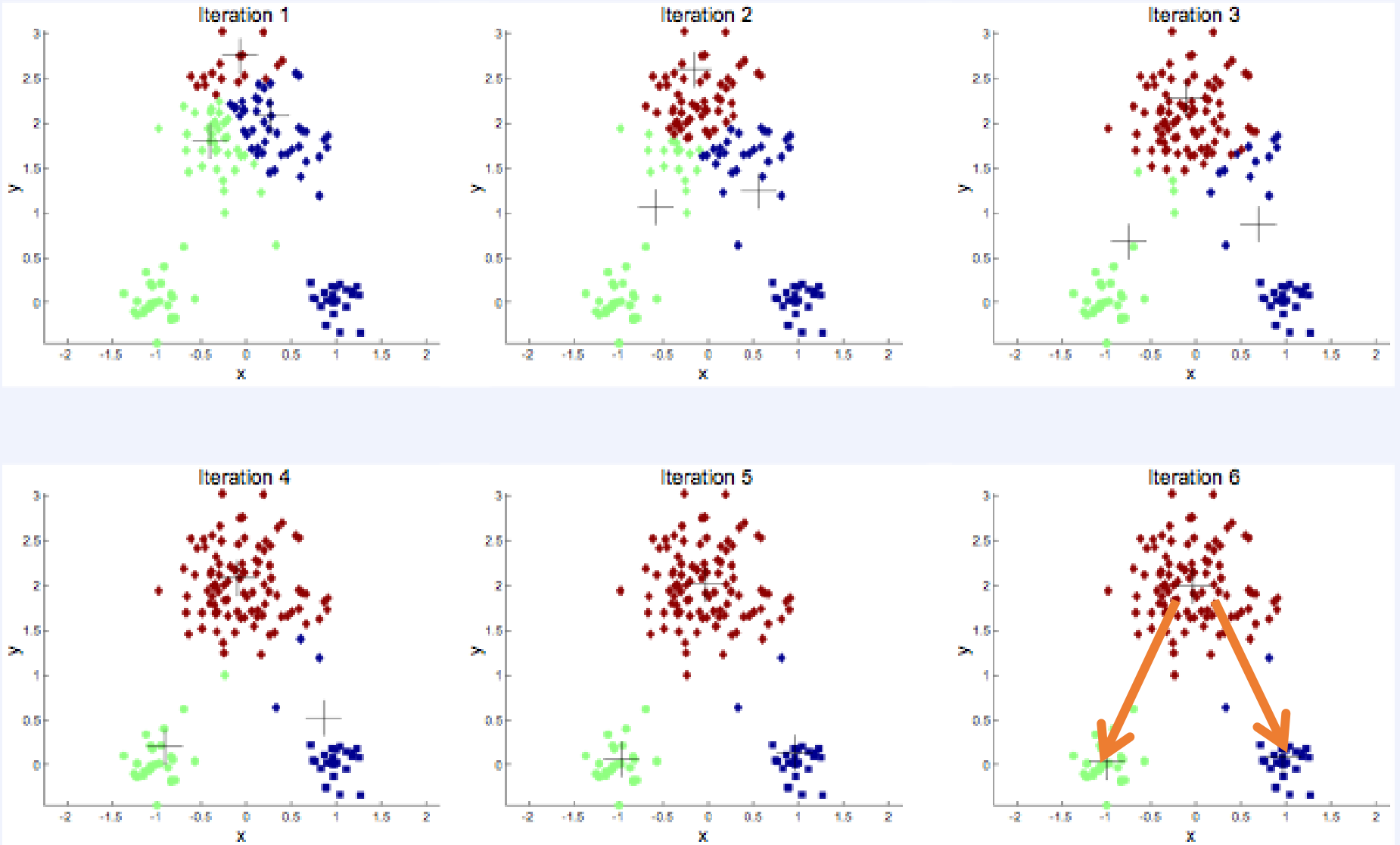
Worst-case lower bound is superpolynomial: $2^{\Omega(\sqrt{n})}$

The discrepancy between practice and worst-case analysis can be (somewhat) explained with some smoothed analysis

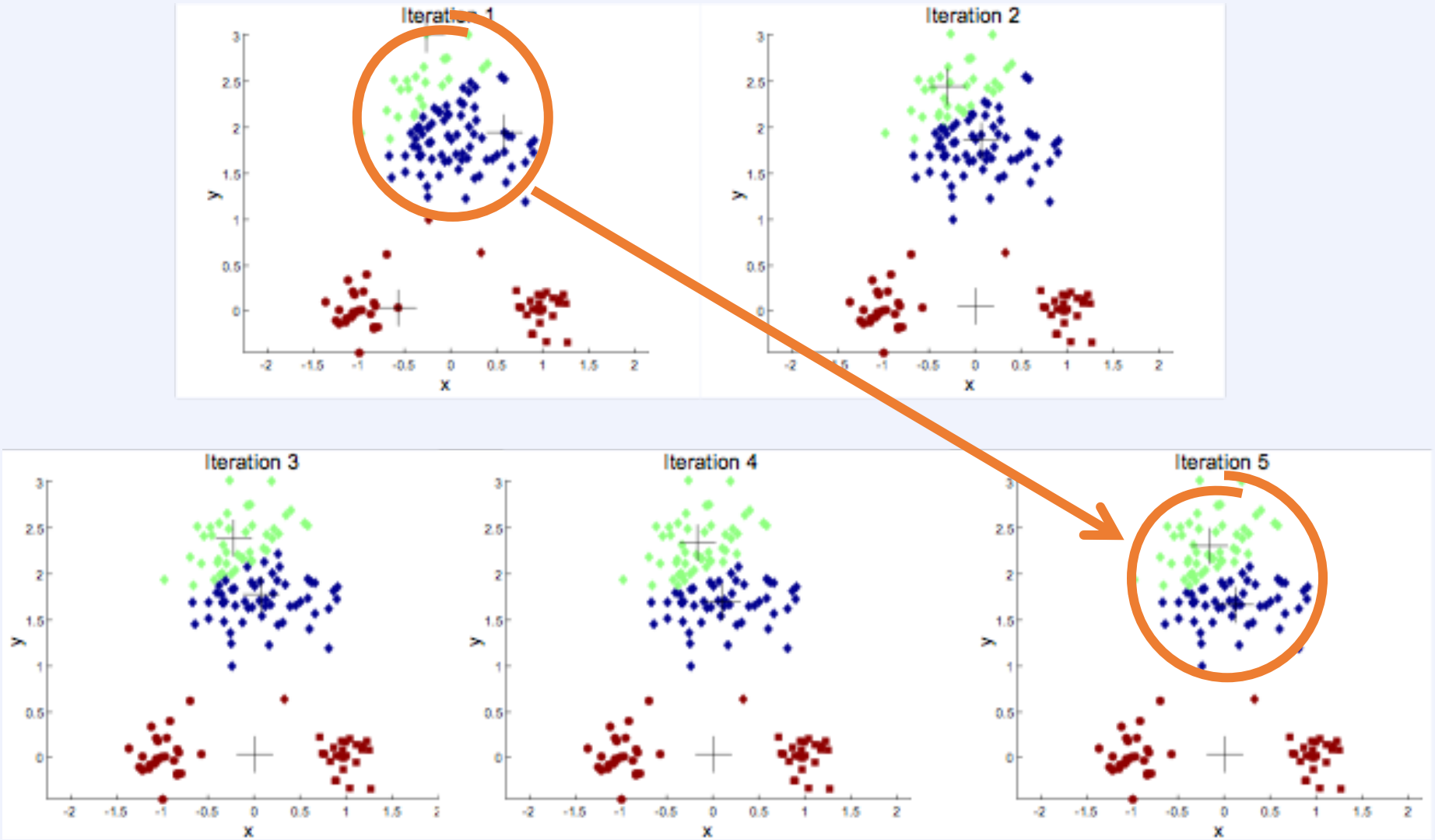
- if the data is sampled from independent d-dimensional normal distributions with same variance, iterative k-means will terminate in $O(n^k)$ time with high probability

(Arthur & Vassilvitskii, 2006)

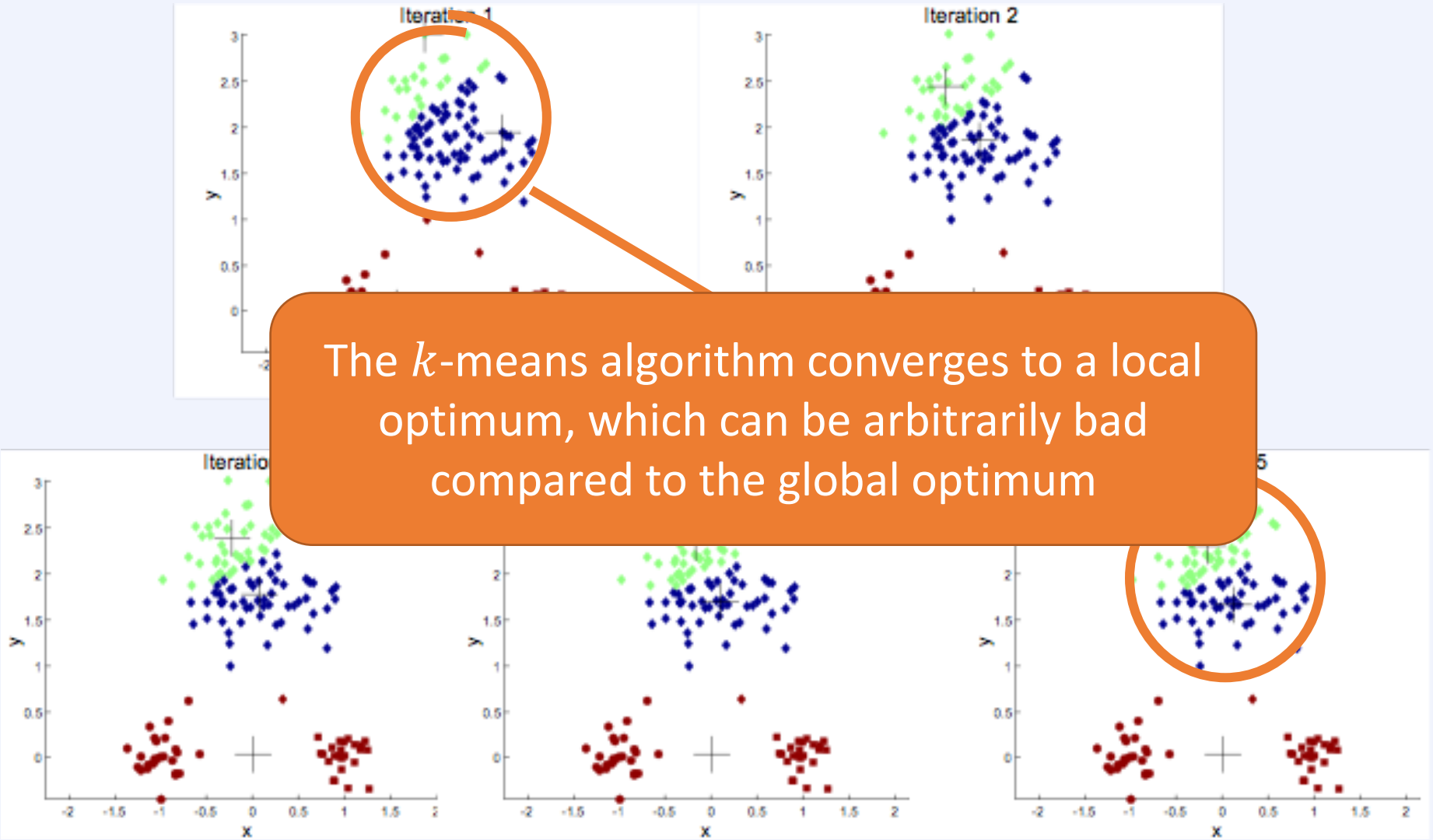
On the importance of starting well



On the importance of starting well



On the importance of starting well



The k-means++ algorithm

The Key Idea: **Careful initial seeding**

- choose first centroid u.a.r. from data points
- let $D(x)$ be the shortest distance from x to any already-selected centroid
- choose next centroid to be x' with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
  - points that are further away are **more probable** to be selected
- repeat until k centroids have been selected and continue as normal iterative k-means algorithm

The k-means++ algorithm achieves $O(\log k)$ approximation ratio on expectation

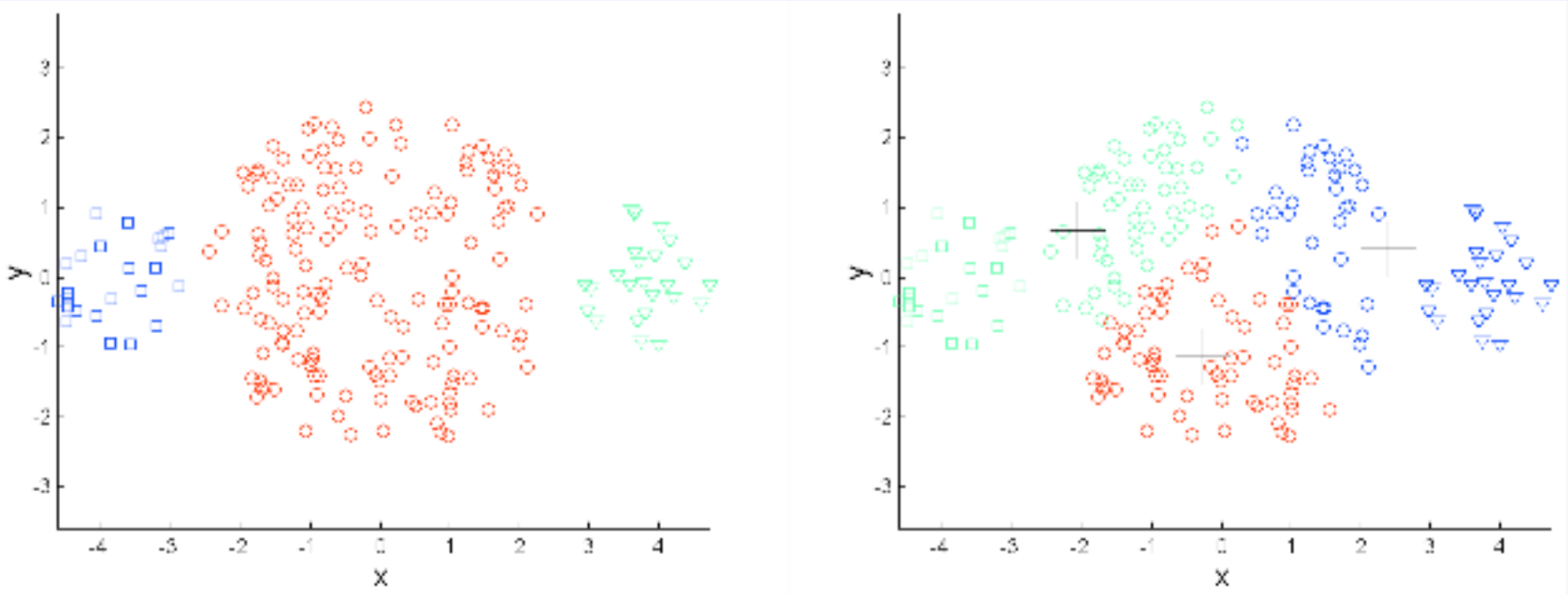
- $E[cost] = 8(\ln k + 2)\text{OPT}$

The k-means++ algorithm converges fast in practice

(Arthur & Vassilvitskii '07)

Limitations of k-means clusterings

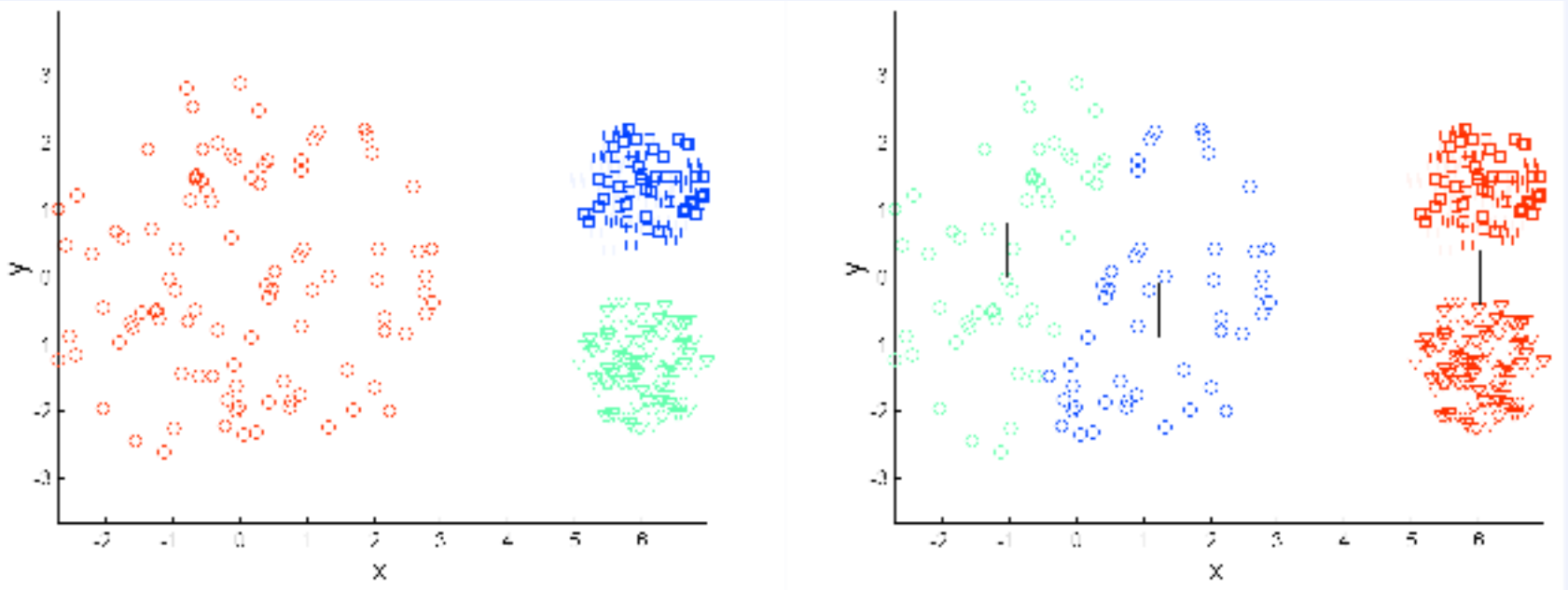
The clusters have to be of roughly equal size



Limitations of k-means clusterings

The clusters have to be of roughly equal size

The clusters have to be of roughly equal density

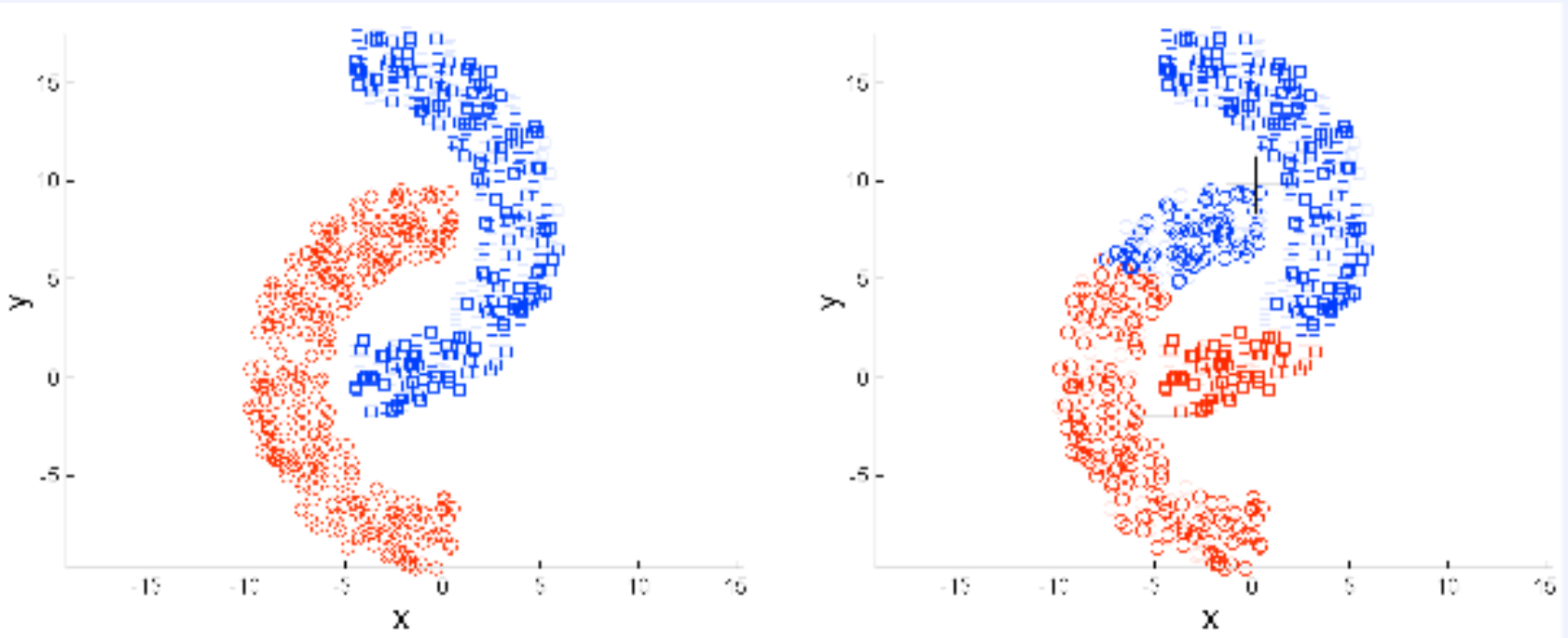


Limitations of k-means clusterings

The clusters have to be of roughly equal size

The clusters have to be of roughly equal density

The clusters have to be of roughly spherical shape



Chapter 5.3:
Probabilistic Model-based

Aggarwal Ch. 6.5

The EM clustering algorithm

Probabilistic clustering

- i.e. not exclusive
- every object has a certain probability (affinity) to every cluster

Representative, in a way

- each cluster is represented by some parameters, Θ
- the parameter may include (or specify) a cluster centroid

Requires us to assume a distribution of a cluster

- for now, each cluster is **independent Gaussian**

We use the **expectation-maximization** (EM) approach

The basics

We aim at finding model Θ, i.e. parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ for each d-dimensional Gaussian cluster, plus k mixture parameters $P(C_i)$

- pdf of an object $\boldsymbol{x}$ in cluster C_i is

$$f_i(\boldsymbol{x}) = f(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_i|^{-\frac{1}{2}} \exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)}{2}\right\}$$

- total pdf of x is a **mixture model** of the k cluster Gaussians

$$f(\boldsymbol{x}) = \sum_{i}^{k} f_i(\boldsymbol{x}) P(C_i) = \sum_{i}^{k} f(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) P(C_i)$$

- the log-likelihood of the data D given parameters Θ then is

$$\log(P(\boldsymbol{D} \mid \Theta) = \sum_{j=1}^{n} \log(\sum_{i}^{k} f(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) P(C_i)$$

The general EM clustering algorithm

Initialisation

- initialise parameters Θ randomly

Expectation (E) step

- compute the posterior probability $P(C_i \mid \boldsymbol{x}_j)$ per Bayes' theorem

$$P(C_i \mid \boldsymbol{x}_j) = \frac{P(\boldsymbol{x}_j \mid C_i)P(C_i)}{\sum_a^k P(\boldsymbol{x}_j \mid C_a)P(C_a)}$$

Maximisation (M) step

- re-estimate Θ given $P(C_i \mid \boldsymbol{x}_j)$

Repeate E and M steps until convergence

EM with 1d Gaussians

Pdf is: $f(x \mid \mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}$

Initialisation step

- mean μ is sampled u.a.r. from possible values, $\sigma^2 = 1$, and $P(C_i) = \frac{1}{k}$ (every cluster is equiprobable)

Expectation step

$$w_{ij} = P(C_i \mid x_j) = \frac{f(x_j \mid \mu_i, \sigma_i^2)P(C_i)}{\sum_a^k f(x_j \mid \mu_a, \sigma_a^2)P(C_a)}$$

Maximisation step

$$\mu_i = \frac{\sum_j^n w_{ij} x_j}{\sum_j^n w_{ij}}$$

Weighted mean

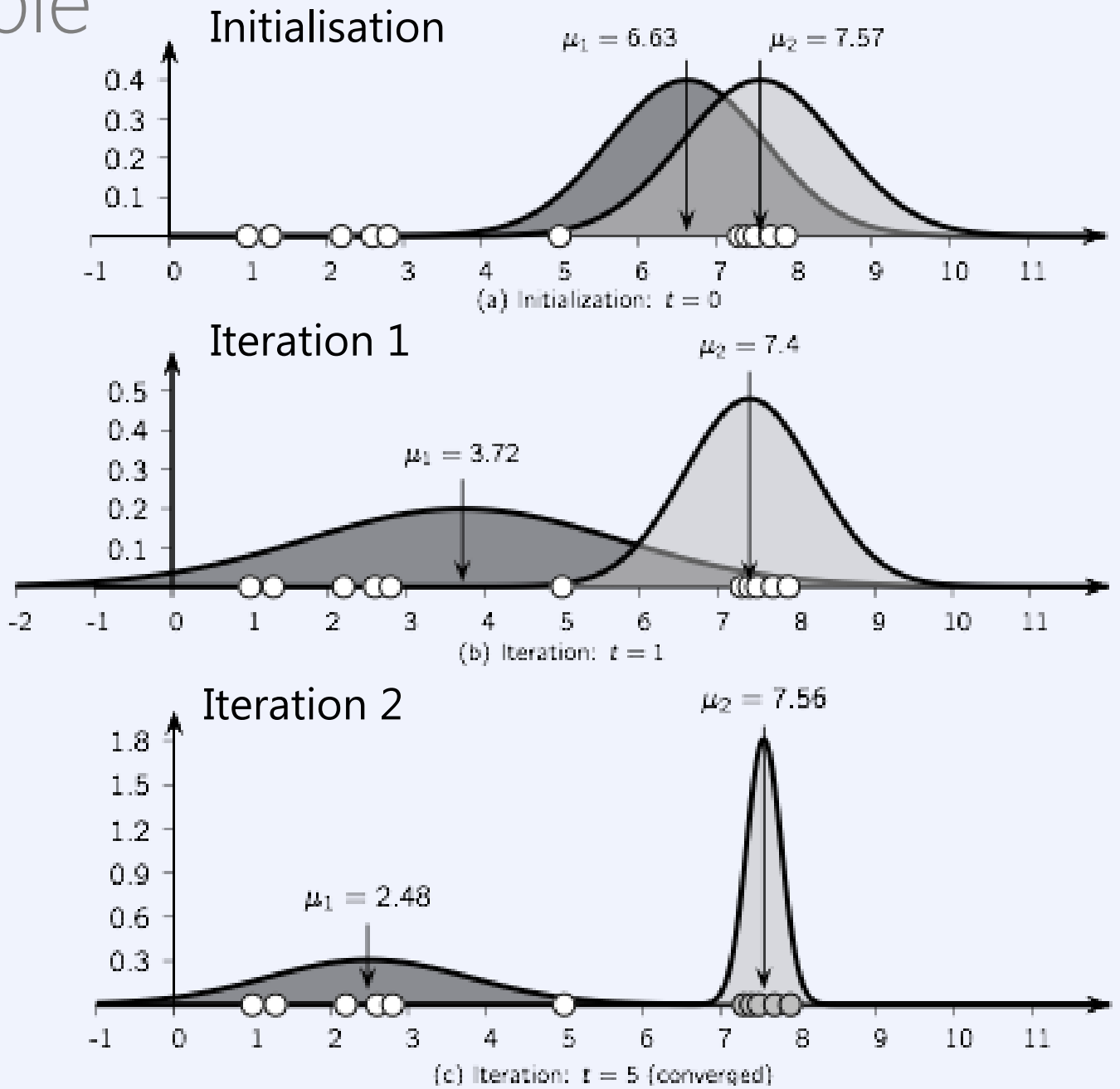
$$\sigma_i^2 = \frac{\sum_j^n w_{ij}(x_j - \mu_i)^2}{\sum_j^n w_{ij}}$$

Weighted variance

$$P(C_i) = \frac{\sum_j^n w_{ij}}{n}$$

Fraction of weight in cluster i

ExaMple


Initialisation
$\mu_1 = 6.63$
$\mu_2 = 7.57$
(a) Initialization: $t = 0$
Iteration 1
$\mu_2 = 7.4$
$\mu_1 = 3.72$
(b) Iteration: $t = 1$
Iteration 2
$\mu_2 = 7.56$
$\mu_1 = 2.48$
(c) Iteration: $t = 5$ (converged)

EM in d dimensions

If we generalise to d-dimensional Gaussians, we need to model the interactions between all dimensions – we need the covariance matrix.

In practice we need to estimate only the upper triangular matrix, which means estimating $\frac{d(d+1)}{2}$ parameters. That's a lot of parameters.

- hence, in practice often dimensions are **assumed** to be independent, yielding d parameters

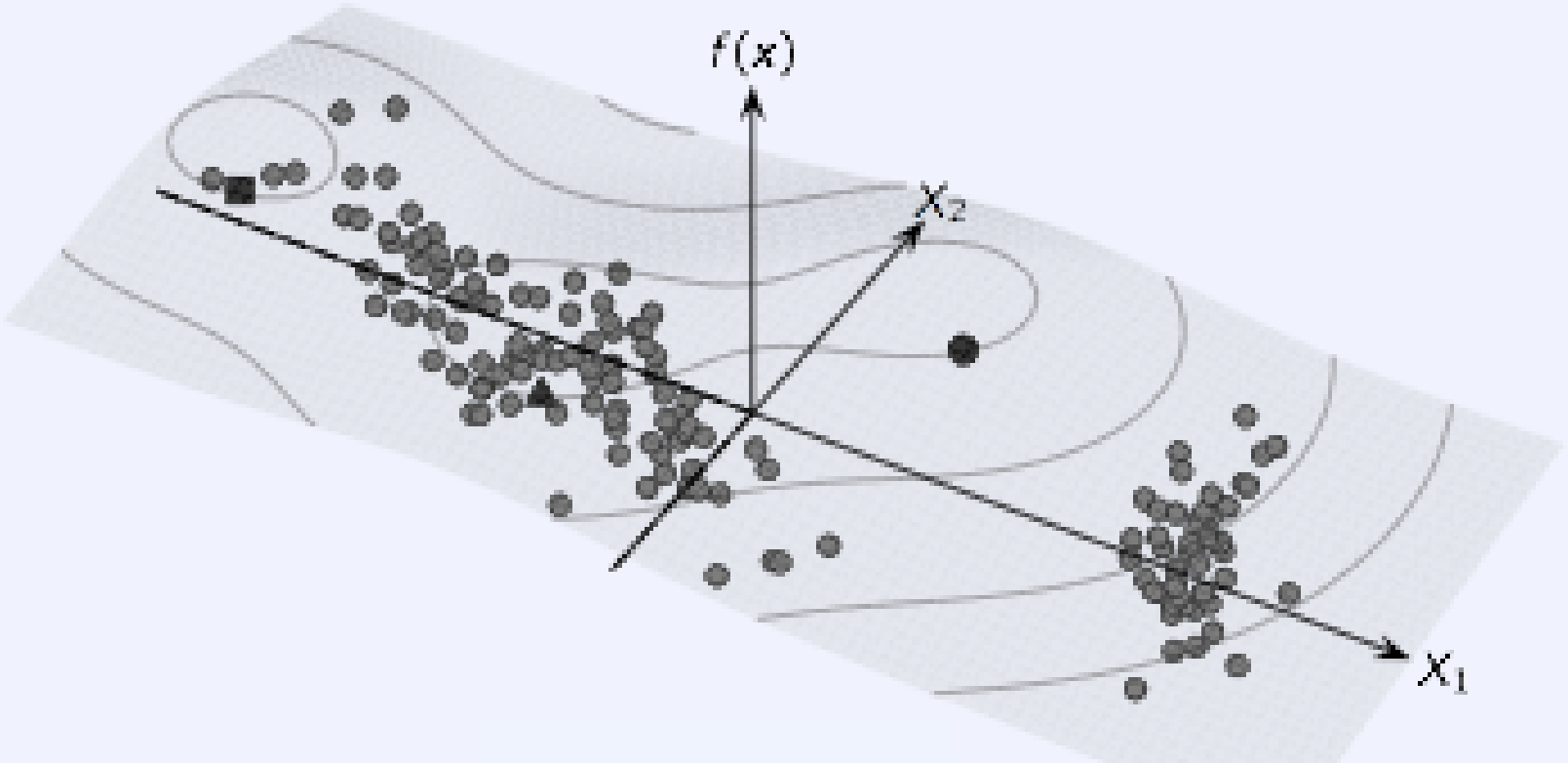
The expectation step is as in 1-D

The mean and prior $P(C_i)$ are estimated as in 1-D

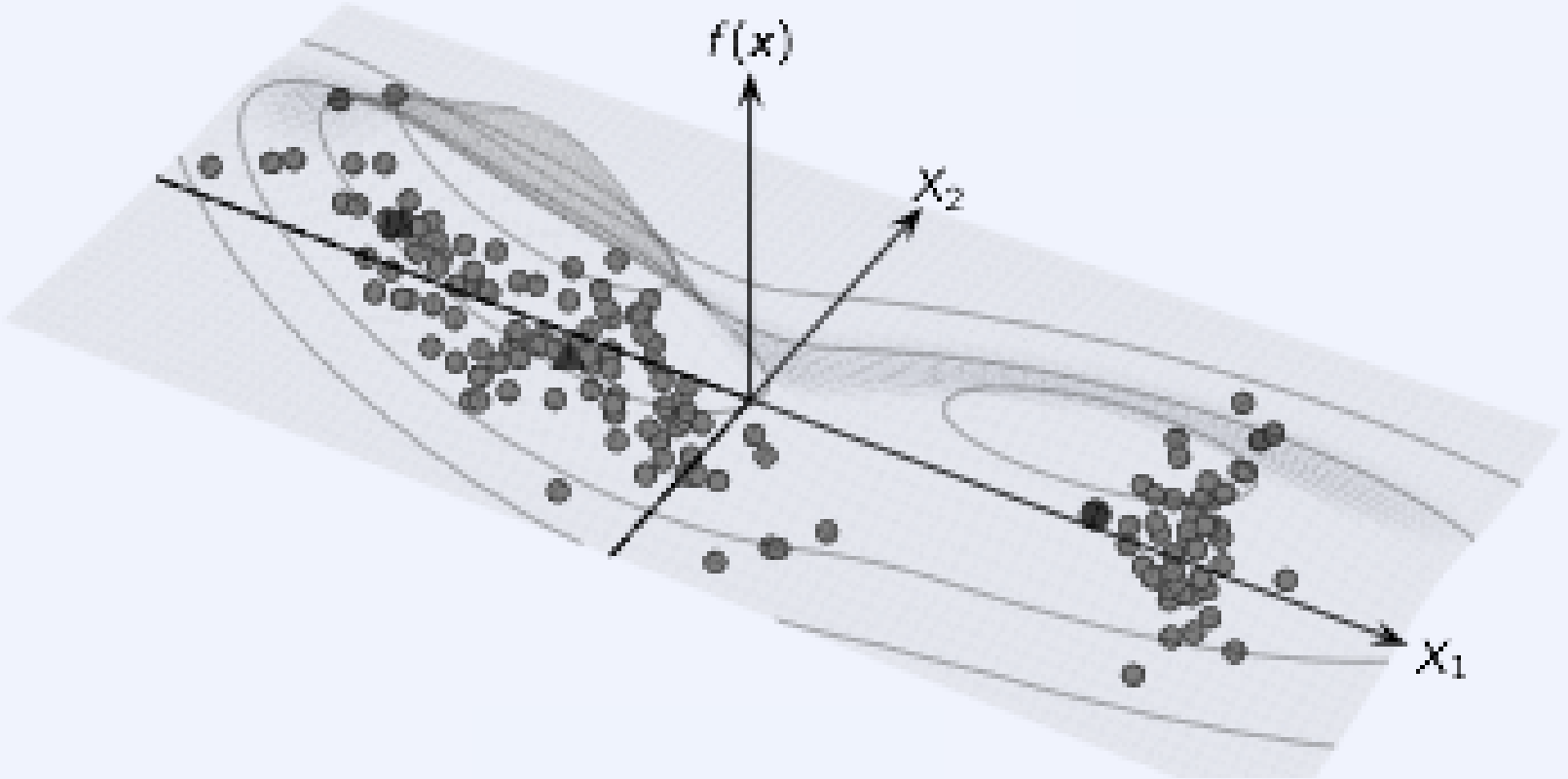
The variance of cluster C_i in dimension a is

$$\left(\sigma_{aa}^{i}\right)^2 = \frac{\sum_j^n w_{ij}\left(\boldsymbol{x}_{ja} - \boldsymbol{\mu}_{ia}\right)^2}{\sum_j^n w_{ij}}$$

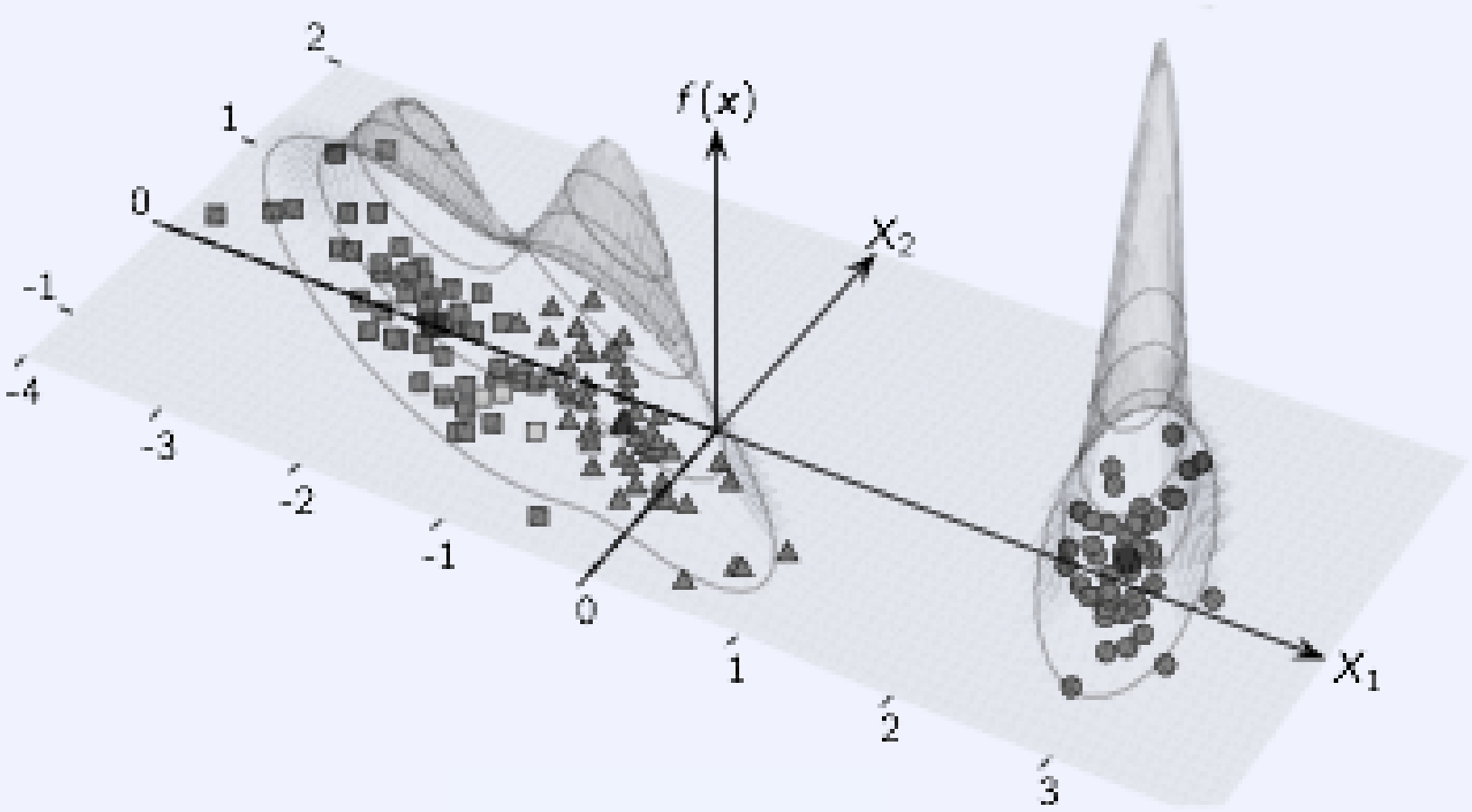
Example – initialisation



Example – iteration 1



Example – iteration 36



k-means as EM

Iterative k-means can be seen as a special case of EM,
i.e. with a different cluster density function

- $P(x_j \mid C_i) = 1$ iff centroid i is the closest to point x_j

The posterior probability is then

- $P(C_i \mid x_j) = 1$ iff point x_j belongs to cluster i

The parameters are the centroids and $P(C_i)$

- the co-variance matrix can be ignored

Chapter 5.4: Validation

Aggarwal Ch. 6.9

How to select k

Both k-means and EM require user to define k before the algorithm is run

- what if we don't know the number of clusters beforehand?

The larger the value of k,

- the smaller the error
- the more complex the model
- the higher the risk for over-fitting

Cross-validation

As with regression:

- hold out some random points (test set)
- run clustering on the remaining points (training set)
- compute the error with test set included
- re-iterate with different values of k and select the one with least overall error

Normally N-fold cross validation

- typically $N = 10$
- data is divided in N even sized sets
- cross-validation is run N times, each time keeping one set as the test set and rest $N-1$ sets together as the training set

AIC and BIC

Let $P_\Theta(D \mid C)$ be the maximized likelihood of clustering C (obtained e.g. via EM algorithm)

Let $l(C)$ be the number of parameters in Θ we need for C

- for Gaussian with independent dimensions, $q(C) = k \times (d + 2)$
  - k clusters, and per cluster 1 mixture parameter $P(C_i)$, d variances, and 1 mean (although d-dimensional, it only counts as one parameter)

Main idea: we pay for every parameter in the model

- in **Akaike's Information Criterion** (AIC) we select the k that minimizes $AIC = -\log P_\Theta(D \mid C) + l(C)$
- in **Bayesian Information Criterion** (BIC) we select the k that minimizes $BIC = -\log P_\Theta(D \mid C) + \frac{l(C)}{2} \log n$

(Akaike, 1974; Schwarz, 1978)

Today's Conclusions

Clustering is one of the most important and most used data analysis methods

There exist many different types of clustering

- so far we've seen representative and probabilistic clustering
- every type of clustering has its strengths and weaknesses

Choosing the number of clusters is often difficult

- cross-validation is a standard method
- AIC and BIC are principled general ways for **model selection**

Thank you!

Clustering is one of the most important and most used data analysis methods

There exist many different types of clustering

- so far we've seen representative and probabilistic clustering
- every type of clustering has its strengths and weaknesses

Choosing the number of clusters is often difficult

- cross-validation is a standard method
- AIC and BIC are principled general ways for **model selection**