Lecture 17. Cluster Analysis

COMP90051 Statistical Machine Learning

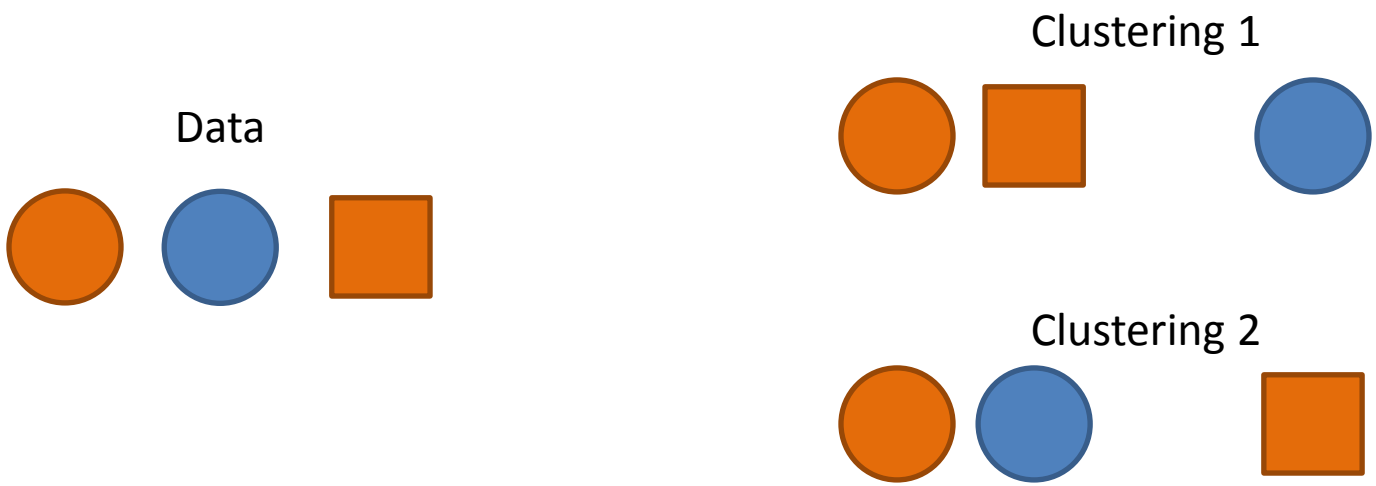
Semester 2, 2015

Lecturer: Andrey Kan

Content is based on slides provided by Jeffrey Chan

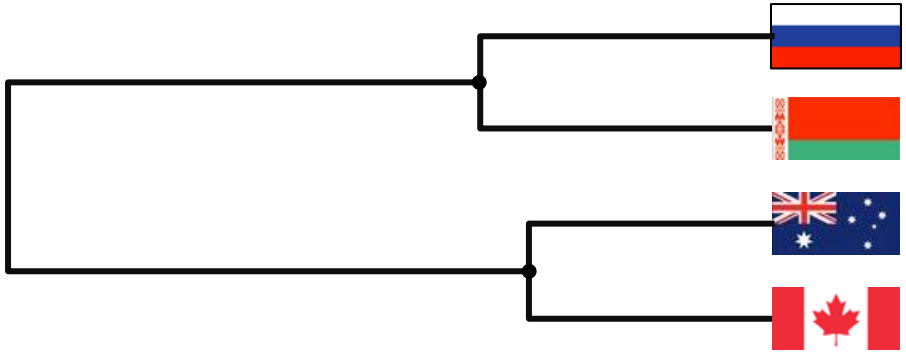
Cluster analysis (or clustering)

- The task of grouping a set of objects such that objects in the same group (cluster) are more *similar* than objects in different groups
- Similarity can be defined in many ways, hence there are many possible groupings of the same data



Major types of clustering

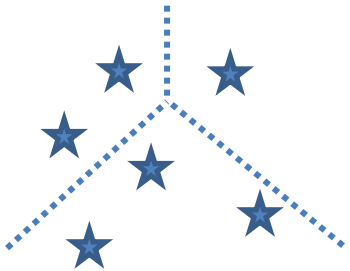
- Exclusive vs. overlapping
  - Exclusive: placing objects into disjoint subsets
  - Overlapping: allowing each object to be a member of more than one cluster
- Deterministic vs. probabilistic
  - Deterministic: each object either belongs to a particular cluster or not
  - Probabilistic: each object belongs to each cluster with some probability
- Hierarchical clustering
  - Aims to build a hierarchy of clusters usually represented with a *dendrogram* (a tree-like diagram)



Exclusive vs. overlapping clustering

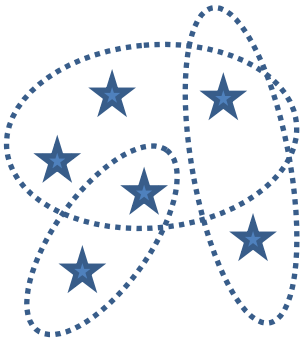
Exclusive

(hard clustering,
partition of a set)



Overlapping

(fuzzy clustering,
soft clustering)



Deterministic clustering is a combinatorial problem

Deterministic vs. probabilistic clustering

Overlapping deterministic

Object	Cluster membership
1	2
2	1, 3
3	4
…	

Exclusive deterministic

Object	Cluster membership
1	2
2	1
3	4
…	

Probabilistic

Object	Cluster			
	1	2	3	4
1	0.01	0.87	0.12	0.00
2	0.05	0.25	0.67	0.03
3	0.00	0.98	0.02	0.00
…				

Exclusive probabilistic: ?

Agglomerative hierarchical clustering: meta-algorithm

1. Initialisation: each point forms a separate cluster

2. Update:

   a) Choose *a pair of clusters*. E.g., two clusters with the smallest intergroup dissimilarity

   b) Merge the two clusters

   (the number of clusters decreases by one in each step)

3. Termination: stop when all points fall into a single cluster

4. Go to Step 2

Divisive hierarchical clustering: meta-algorithm

1. Initialisation: start with a single cluster

2. Update:

   a) Choose *a cluster* to split. E.g., so that it can produce two clusters with the largest between-group dissimilarity

   b) Split the cluster in two

   (the number of clusters increases by one in each step)

3. Termination: stop when each point fall into a separate cluster

4. Go to Step 2

K-means clustering (refresher)

- **Input**: N data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, where each point is a p-dimensional vector $\boldsymbol{x}_i = \left[x_1^{(i)}, \ldots, x_p^{(i)}\right]$; and the number of clusters K
- **Objective**: produce exclusive deterministic clustering that minimizes "within cluster" variation while maximizing "between-cluster" variation
- Minimize cost function $W = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)^2$
  - $\overline{\boldsymbol{x}}_k$ is the *centroid* for cluster k, i.e., the mean of all points assigned to cluster k
  - r_{ik} is the membership indicator: $r_{ik} = 1$ if point i is assigned to cluster k, and $r_{ik} = 0$ otherwise
  - <u>Constraints</u>: $\sum_{k=1}^{K} r_{ik} = 1$ and $\sum_{i=1}^{N} r_{ik} \geq 1$ for each i and k
- **Output**: exclusive deterministic clustering

K-means clustering (refresher)

- Combinatorial problem
  - Number of possible assignments grows exponentially with N
  - For only 19 points and 4 clusters there are already about 10^{10} possible assignments
  - NP-hard problem
- Old friend: iterative approach

K-means clustering is an iterative procedure

1. Initialisation: assign N points to K clusters randomly

2. Update:
   a) Compute centroid $\overline{\boldsymbol{x}}_k$ for each cluster
   b) Re-assign each point to the cluster with the nearest centroid

   (cost function W will monotonically decrease at this step!)

3. Termination: stop if assignment does not change any more (convergence)

4. Go to Step 2

Reflections on K-means clustering

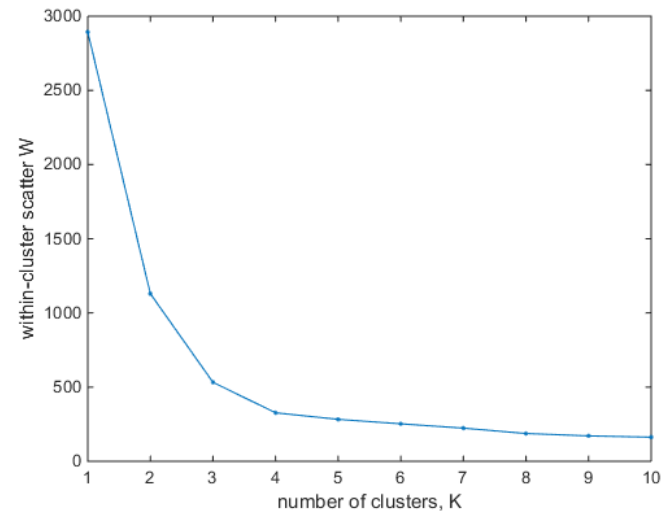
- Pros:
  - Simple and intuitive, intuition is supported by math
  - It works! One of the most cited algorithms in data mining
- Cons:

Problem	Solution
Often converges to a local minimum	Restart with a different initialization
Not able to find non-spherical clusters	Use kernelized PCA, MDS or Isomap first, then apply K-means
Requires point coordinates $\boldsymbol{x}_i = \left[x_1^{(i)}, \dots, x_p^{(i)}\right]$	
Requires number of clusters K	Next slide
Produces exclusive deterministic clustering	Slides after the next slide ☺

Choosing the number of clusters

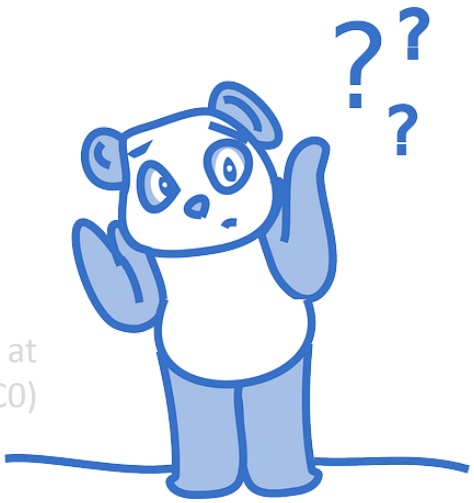
- Sometimes it is not needed (e.g., for hierarchical clustering)
- Sometimes it is know from the problem context:
  - E.g., clustering genetic sequences, and knowing from biology that there should be three types of cells in this dataset
- If it is not known, but needed:
  - Can't we just choose K that minimizes the cost function (within-cluster point scatter W)?
  - Can't we just use cross-validation?

$$W = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)^2$$



Checkpoint

- Which of the following statements is true?
  -  High-dimensional data can actually be represented with fewer dimensions because it lies on a manifold
  -  Isomap algorithm aims to approximate and preserve inter-point distances along the manifold
  -  Euclidean distance is an optimal way to combine features because it satisfies triangle inequality

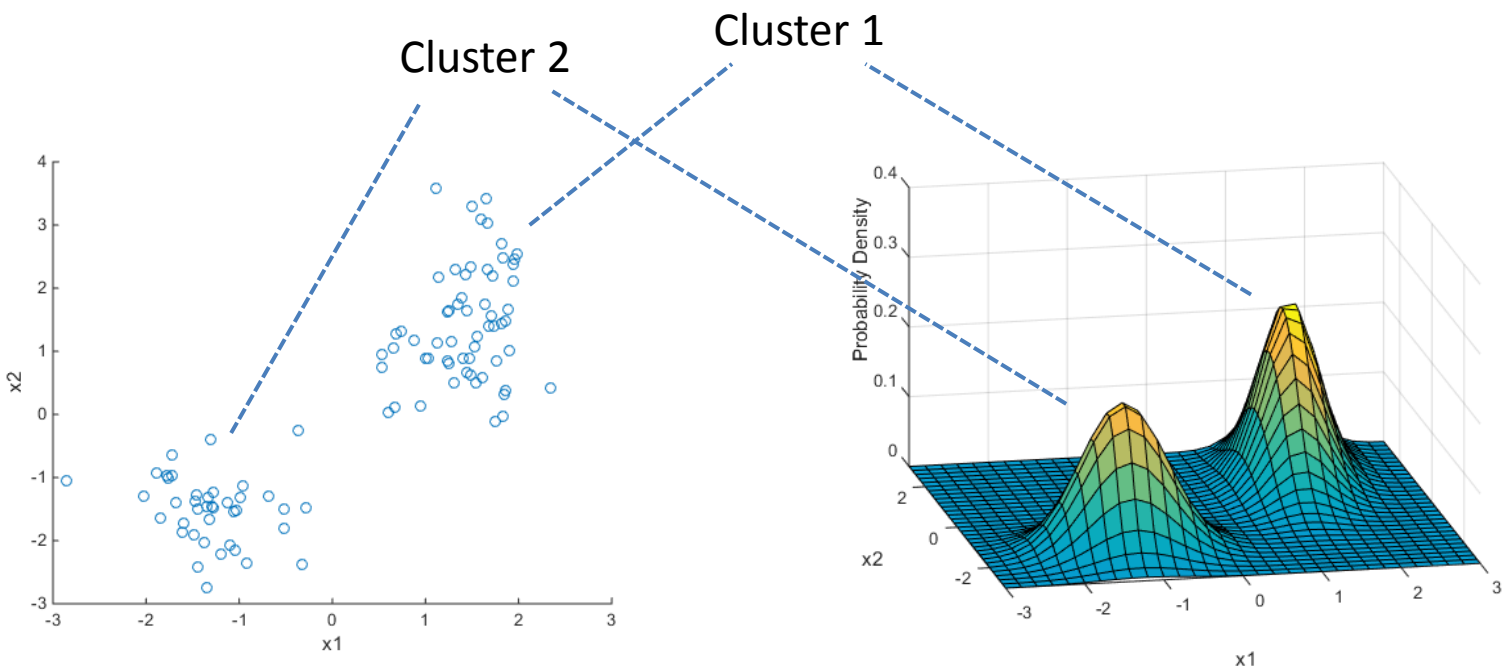


art: OpenClipartVectors at pixabay.com (CC0)

Probabilistic clustering: mixture models

Clustering can be viewed as identification of components in a mixture probability density function

Identifying cluster centroids can be viewed as finding modes of symmetric distributions



Gaussian distribution (refresher)

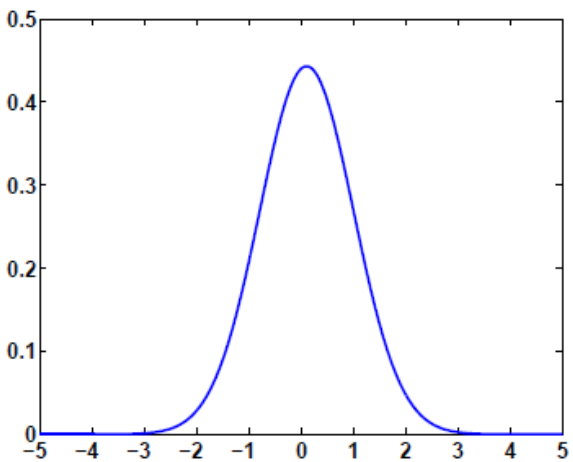
- Recall that a 1D Gaussian is

$$\mathcal{N}_{1D}(x|\mu,\sigma) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

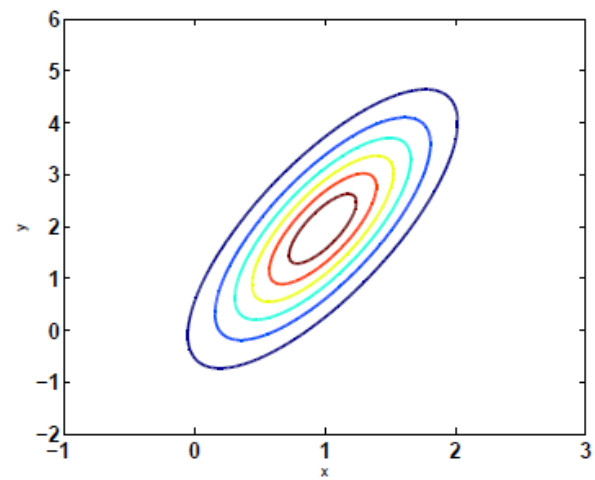
- And a p-dimensional Gaussian is

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \stackrel{\text{def}}{=} (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - $\boldsymbol{\Sigma}$ is positive definite, $|\boldsymbol{\Sigma}|$ denotes determinant



(a) 1-Dim



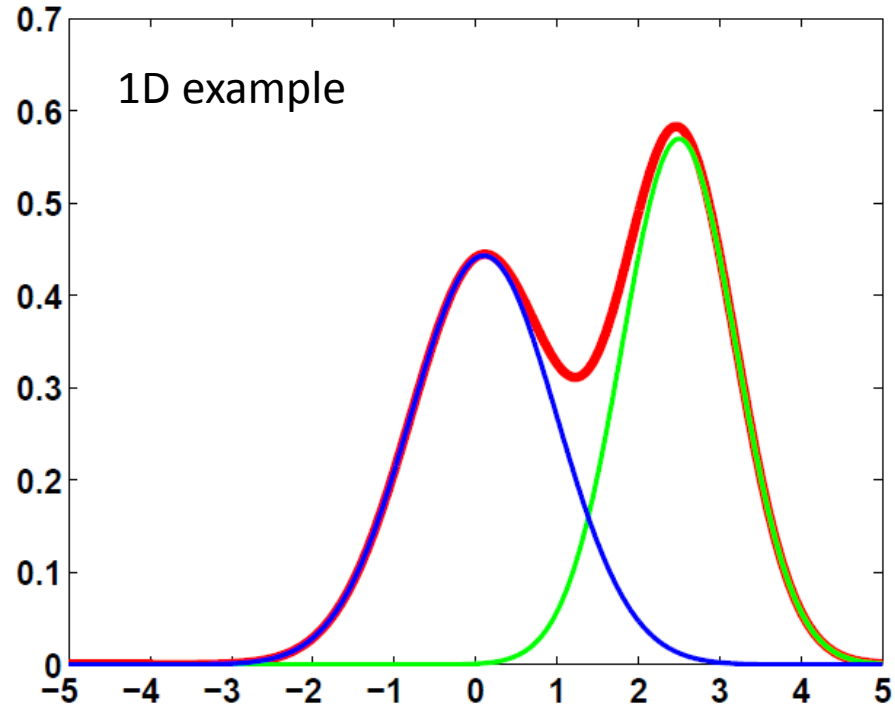
(b) 2-Dim

Gaussian mixture model

- Gaussian mixture distribution

$$f(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Here $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$
- That is, $w_1, \ldots, w_k$ is a *distribution* over components
- Parameters of the model are $w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, \ldots, K$



Another old friend: MLE

- We assume that data comes from a distribution defined by

$$f(\boldsymbol{x}|w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \stackrel{\text{def}}{=} \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Parameters of the model are

$$w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \text{ for } k = 1, \dots, K, \text{ where } K \text{ is fixed}$$

- MLE: find parameter values that maximize log-likelihood

$$\ln L(w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{i=1}^{N} \ln f(\boldsymbol{x}_i|w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

MLE for Gaussian mixtures

- Optimization problem is to maximize

$$\ln L\left(w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \sum_{i=1}^{N} \ln\left(\sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right)$$

- Difficult to optimize (no closed-form solution)
- Could use gradient descent, but there is a better way
- Yet another old friend: Expectation Maximization (EM) algorithm

Key idea: introduce latent variables

- Consider a dataset with N points
- Define a membership matrix $N \times K$ with an element $z_{ik} = 1$ if point $\boldsymbol{x}_i$ was drawn from component k, and $z_{ik} = 0$ otherwise
  - Of course, we don't know true memberships, but we'll iteratively make better and better guesses for z_{ik}
- It turns out that math becomes easier, if we pretend that we know memberships

$$\ln L = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} (\ln w_k + \ln \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$$

EM algorithm is an iterative procedure

1. Initialisation: choose random values for w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
2. Update:
   a) E-step updates an estimate for latent variables
   b) M-step updates an estimate for parameters
3. Termination: stop if parameter estimates don't change anymore (convergence)
4. Go to Step 2

EM for Gaussian mixtures

<u>E-step</u>

$$\tau_{ik} \stackrel{\text{def}}{=} Pr(z_{ik} = 1 | w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\tau_{ik} = \frac{w_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{s=1}^{K} \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}$$

<u>M-step</u>

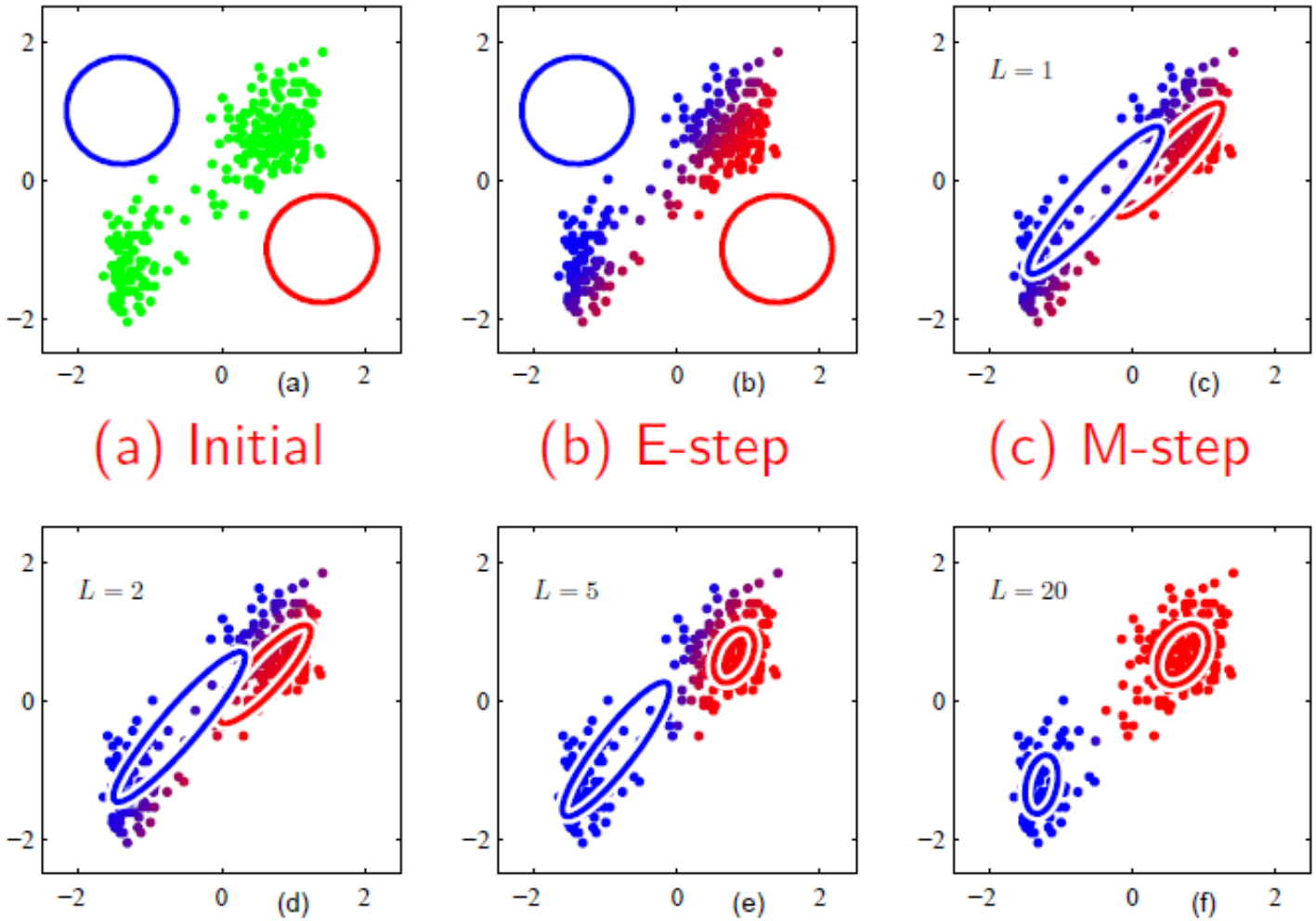
$$N_k = \sum_{i=1}^{N} \tau_{ik}$$

$$w_k^{new} = \frac{N_k}{N}$$

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} \tau_{ik} \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} \tau_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})^T$$

Example of fitting Gaussian Mixture model



(a) Initial (b) E-step (c) M-step

(d) 2 cycles (e) 5-cyclces (f) 20-cycles

K-means clustering is an iterative procedure

1. Initialisation: assign N points to K clusters randomly

2. Update:
   a) Compute centroid $\overline{\boldsymbol{x}}_k$ for each cluster (≈ M-step)
   b) Re-assign each point to the cluster with the nearest centroid (≈ E-step)

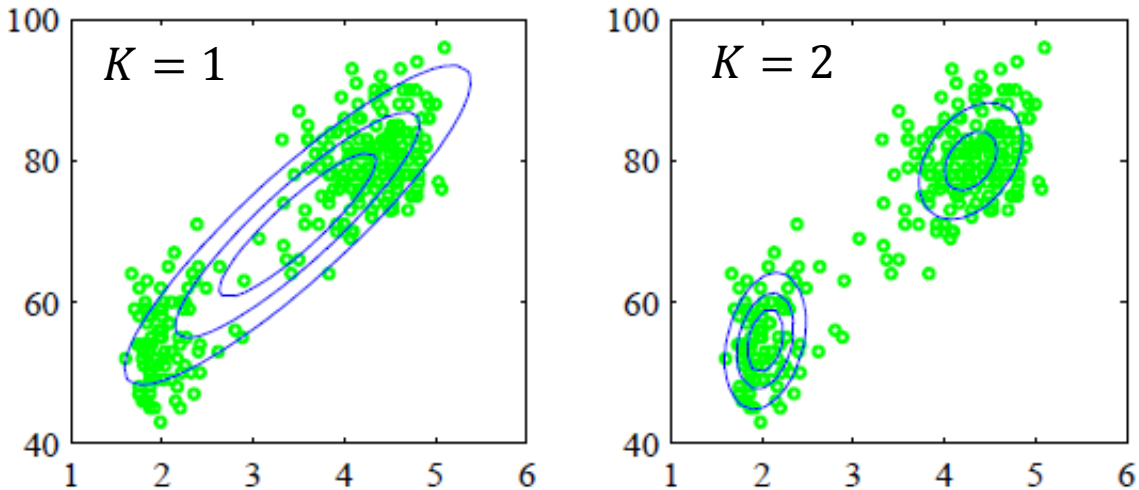
   (cost function W will monotonically decrease at this step!)

3. Termination: stop if assignment does not change any more (convergence)

4. Go to Step 2

A limitation of the MLE

Number of components (clusters) K is enforced



- MLE is applied once K is given
  - Note that given K the number of parameters is $O(K)$
- Can we use MLE to estimate K itself?

Plot from Bishop, Pattern recognition and machine learning, 2007

Choosing the number of components

- Suppose the data is generated by unknown distribution γ, and we have two guesses f_1 and f_2 of what that distribution might be
  - E.g., γ is the true mixture with some true number of components (clusters);
  - and f_1, f_2 are two Gaussian mixtures with numbers of clusters set to K_1 and K_2, respectively
- Each of f_1 and f_2 is a different approximation of γ, and there is a way of quantifying information loss incurred by approximation
  - Something called Kullback-Leiber divergence
  - A result from information theory
  - We denote KL divergence as $D(true\_model, approx\_model)$
- One should select f_1 (that is a mixture with K_1 clusters) if $D(\gamma, f_1) < D(\gamma, f_2)$, and f_2 otherwise

Akaike Information Criterion (AIC)

- The problem: we cannot compute KL divergence, because we don't know γ (in particular, we don't know the true number of clusters)
- Akaike showed that relative divergence can be estimated using the following criterion

$$AIC = 2N_{par} - 2\ln L^*$$

  - Here N_{par} is the total number of parameters, and $\ln L^*$ is the maximized log-likelihood
- Given a set of models (e.g., Gaussian mixture with 2 components, Gaussian mixture with 3 components, etc.), one should choose the model with the *smallest* AIC

AIC: counting parameters

- Akaike Information Criterion

$$AIC = 2N_{par} - 2\ln L^*$$

  - Here N_{par} is the total number of parameters, and $\ln L^*$ is the maximized log-likelihood

- Example: 1D Gaussian with 3 components.
  - $N_{par} = 2 + 3 + 3$, probability w_k for each component, mean and standard deviation for each component

- Example: 2D Gaussian with 3 components.
  - $N_{par} = 2 + 3 \cdot 2 + 3 \cdot 3$, probability w_k for each component, mean vector and a 2×2 covariance matrix for each component

- In our mixture models K does not count as a parameter
  - We only count free variables in MLE process

AIC: important notes

- For each candidate model, the log-likelihood should be maximized and evaluated on the same dataset
- Log-likelihood should be maximized for each model. That is,
  - For each model perform MLE
  - Use parameters found by MLE to compute log-likelihood
- AIC does not tell how good the model is, AIC tells that given a set of candidate models, the one with the smallest AIC is better than others

AIC in practice

- AIC is biased for finite sample sizes, thus a correction has been proposed
- AIC with a correction term

$$AICc = 2N_{par} + \ln L^* + \frac{2N_{par}\left(N_{par} + 1\right)}{N - N_{par} - 1}$$

- In practice, it is recommended to use $AICc$

Summary

- Major types of clustering
  - exclusive and overlapping, deterministic and probabilistic
- K-means clustering
  - Iterative approach to optimization
  - Pros and cons
- Gaussian mixtures as probabilistic clustering
- Party of old friends:
  - Iterative optimization
  - Maximum likelihood estimation
  - EM algorithm
- Estimating the number of clusters
  - Challenges
  - Akaike Information Criterion