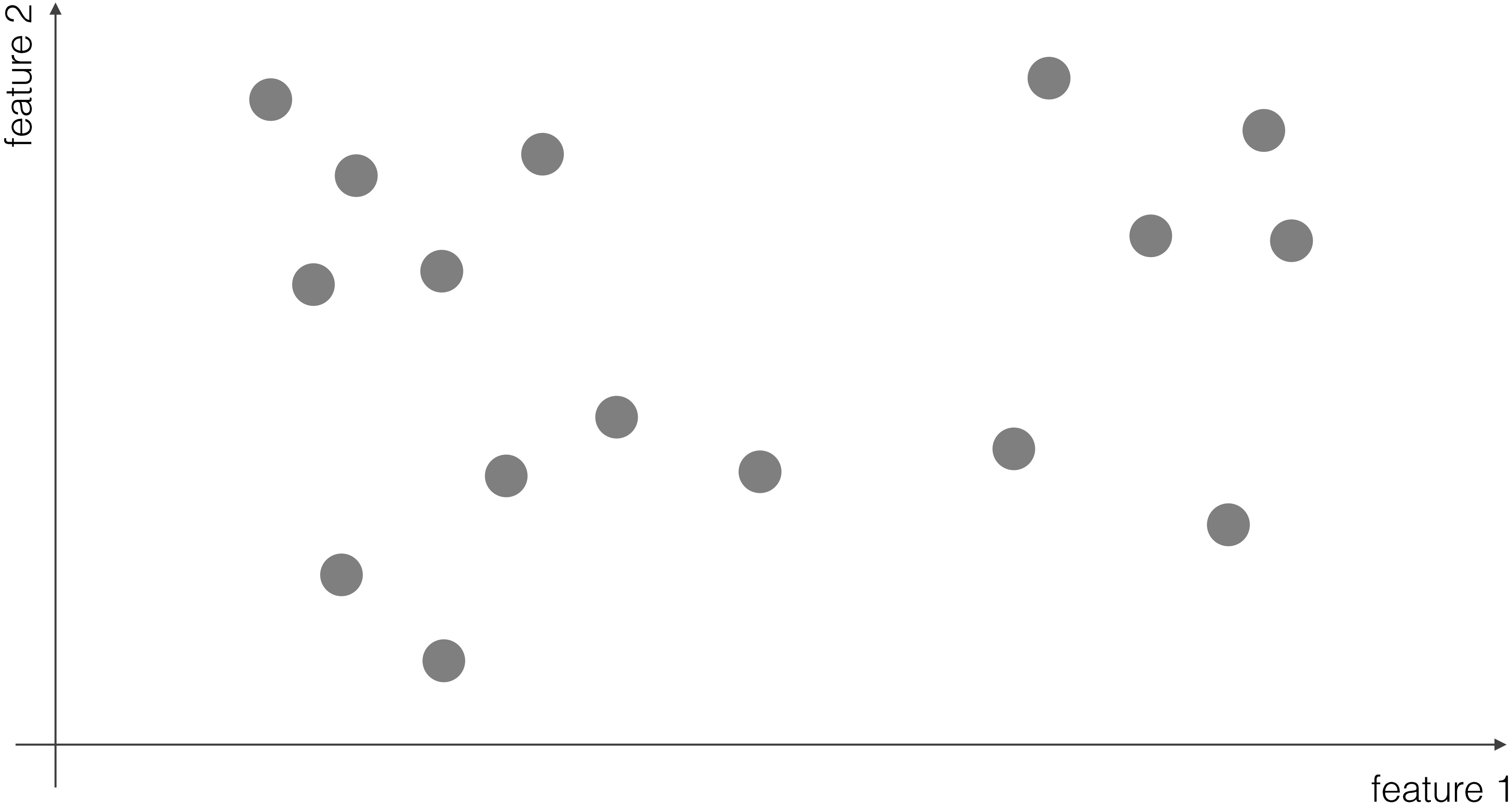
Clustering I

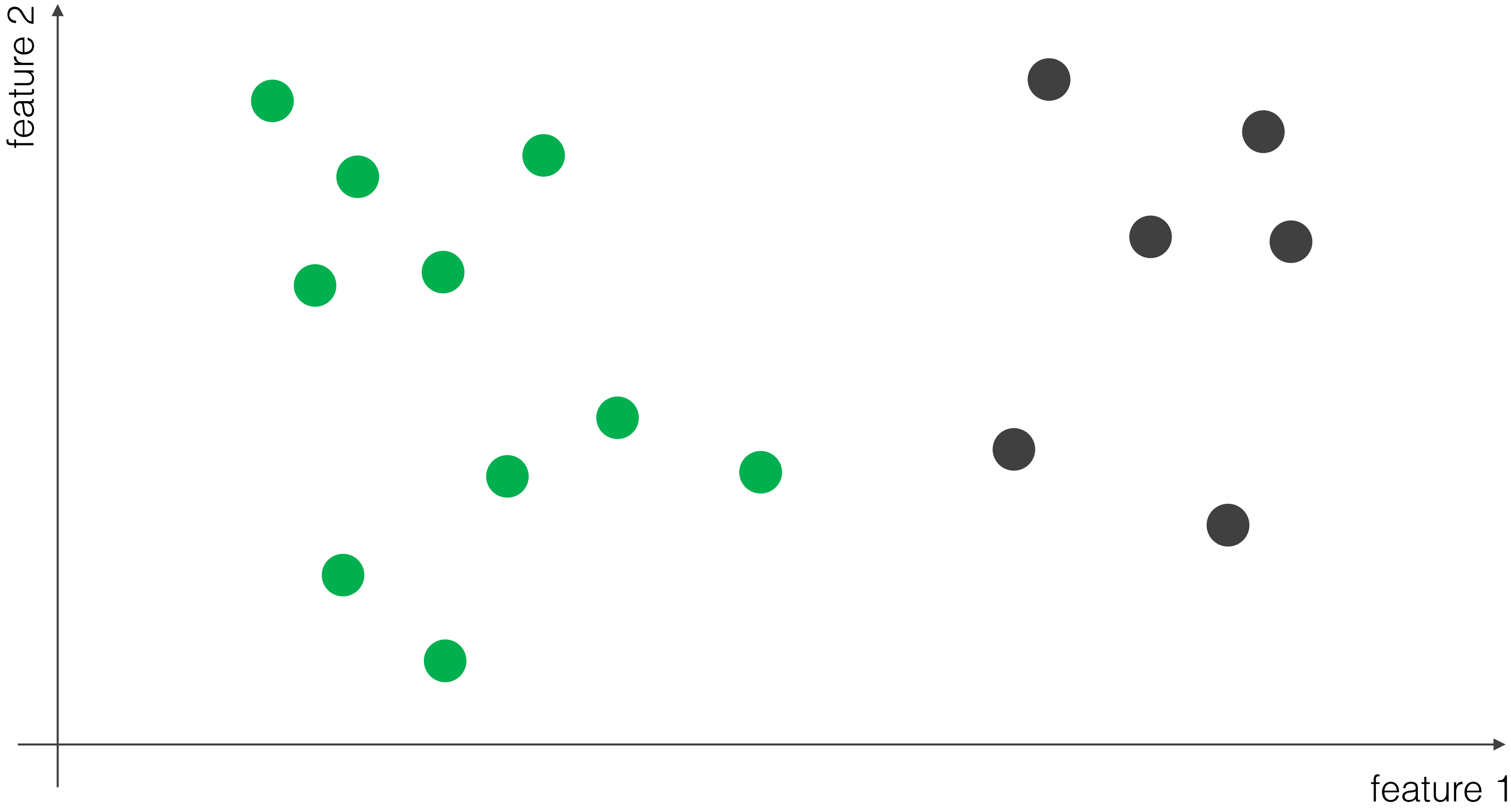
Lecture 14

Clustering



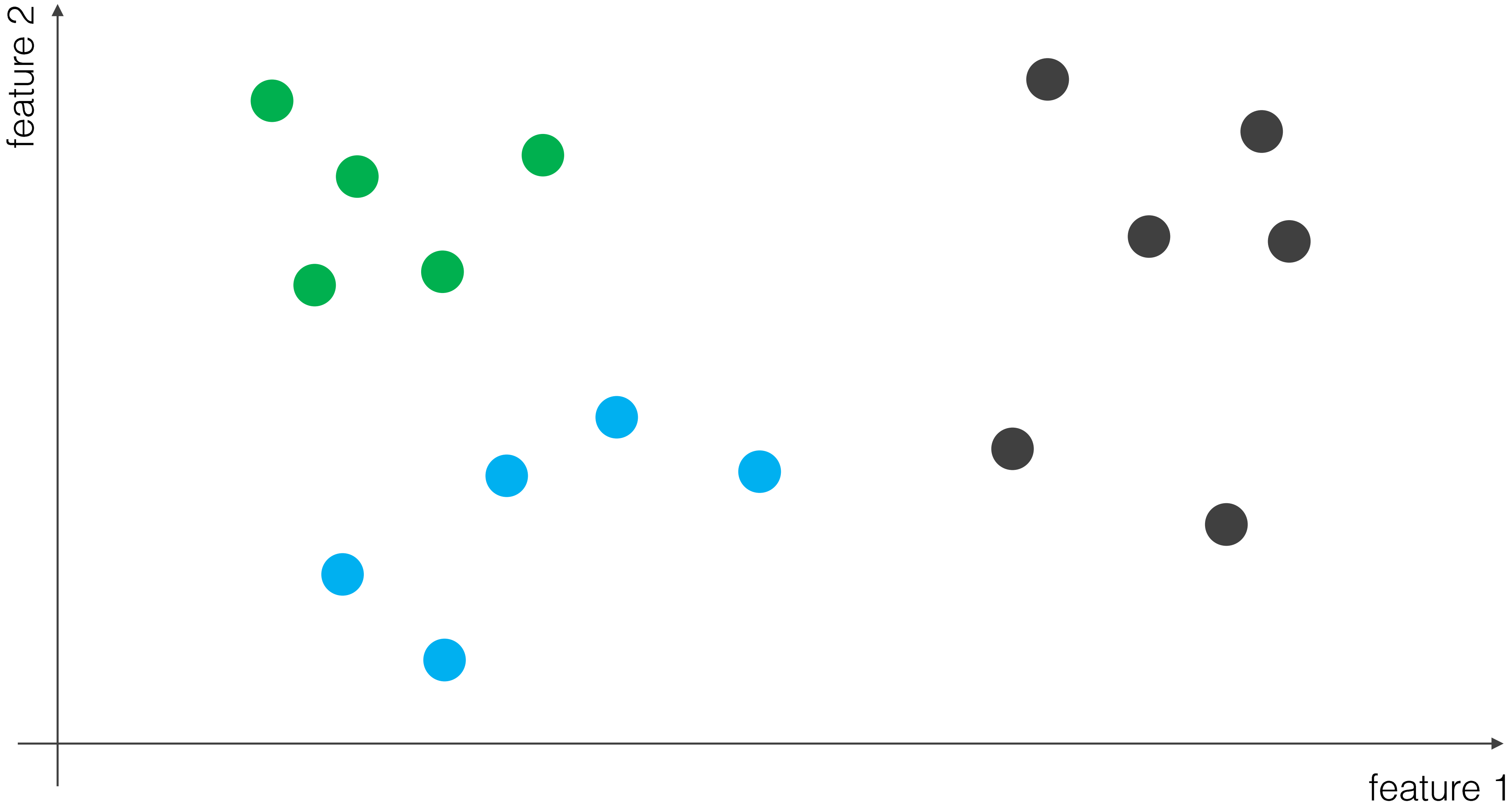
Clustering

Looks like 2 clusters…



Clustering

… or maybe 3?



Applications

Differentiating tissue types in PET scans

Customer segmentation for market research

Social network analysis and identifying communities

Crime tracking to identify hot spots for certain types of crimes

Types of clustering algorithms

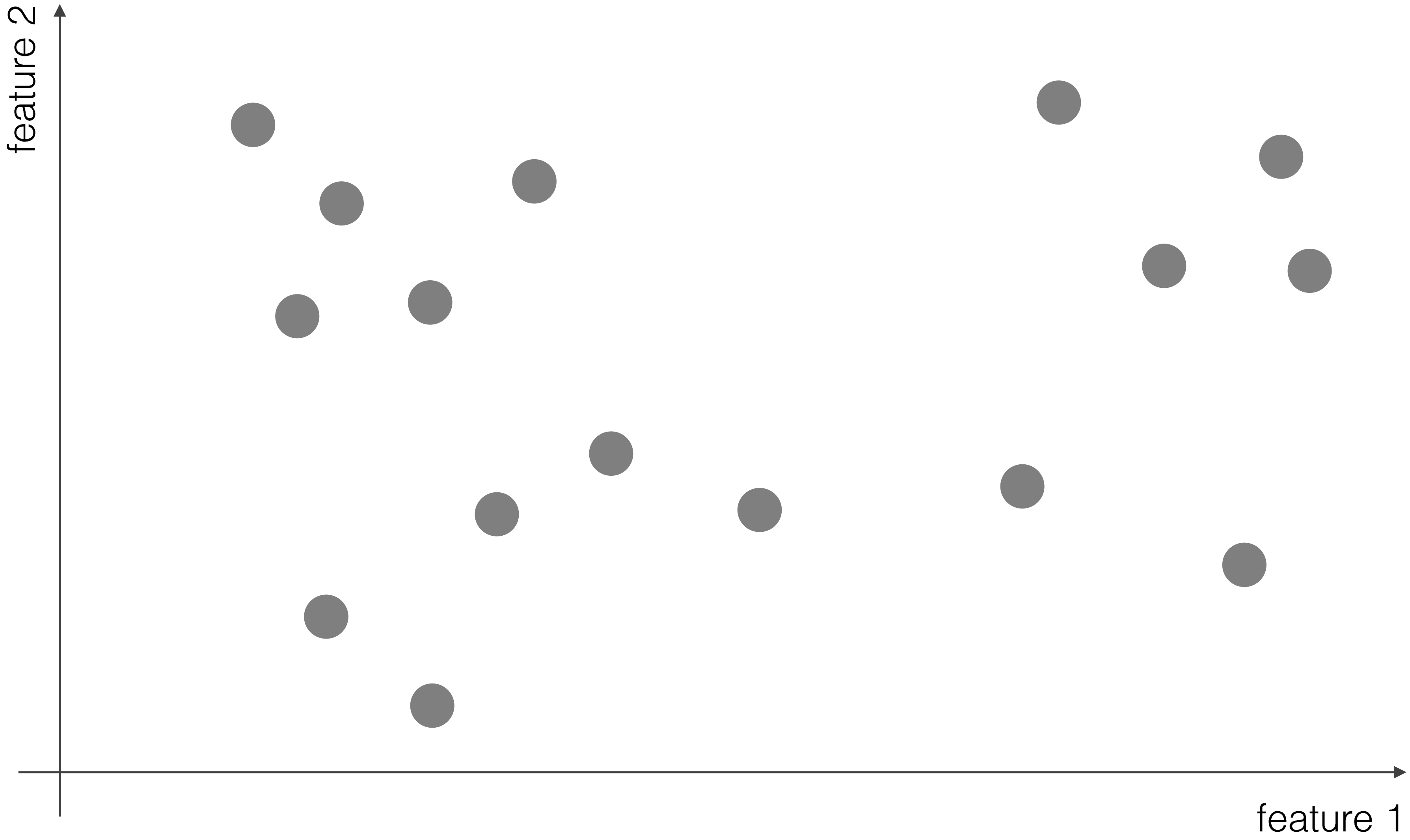
Methods

Centroid-based clustering (e.g. K-Means)
Distribution-based clustering (e.g. Gaussian mixture model)
Density-based clustering (e.g. DBSCAN)
Hierarchical clustering (e.g. agglomerative clustering)
a.k.a. connectivity-based clustering

Cluster assignment

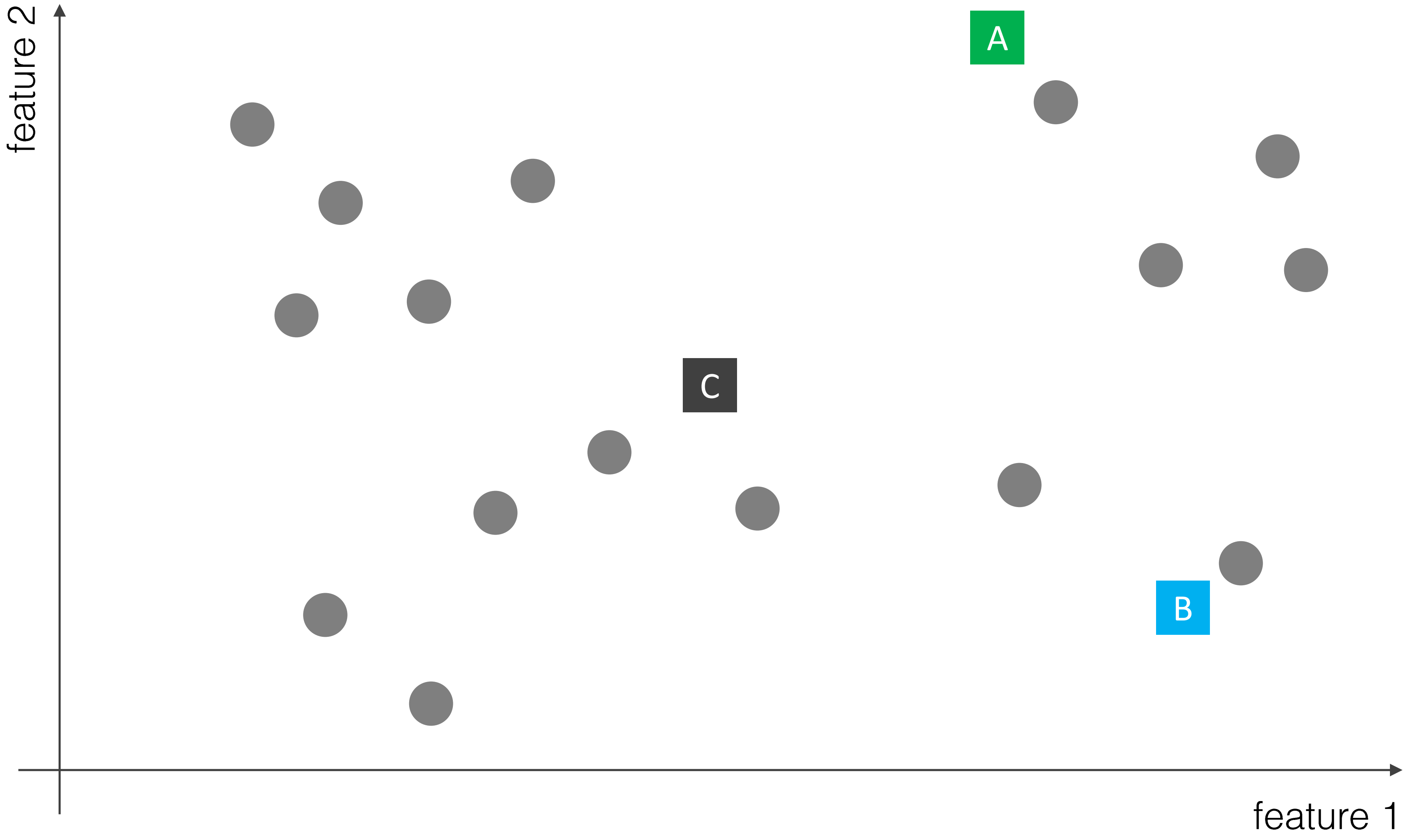
Hard clustering
Soft clustering (a.k.a. fuzzy clustering)

K-means clustering



K-means clustering

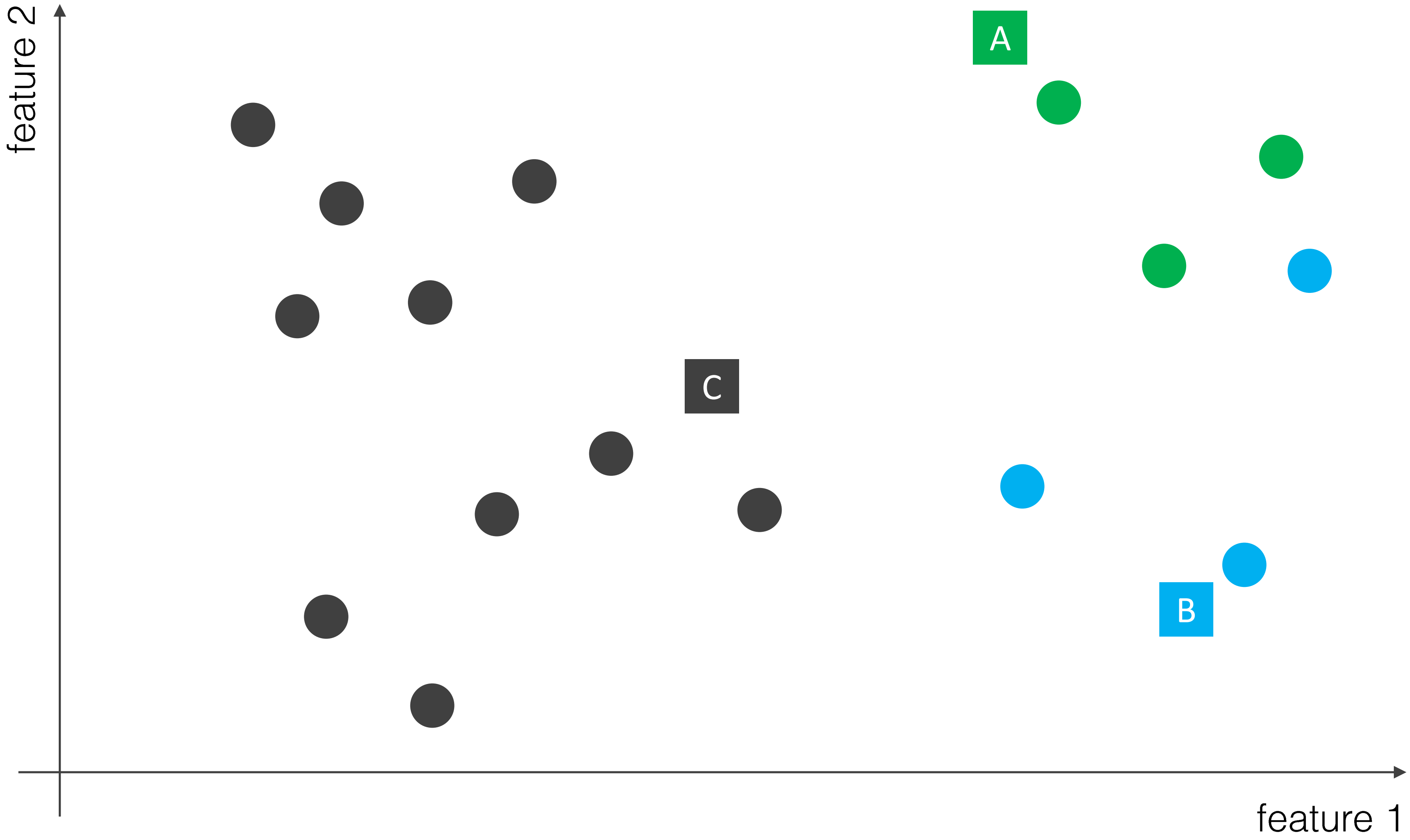
1 Select k and randomly initialize k mean values



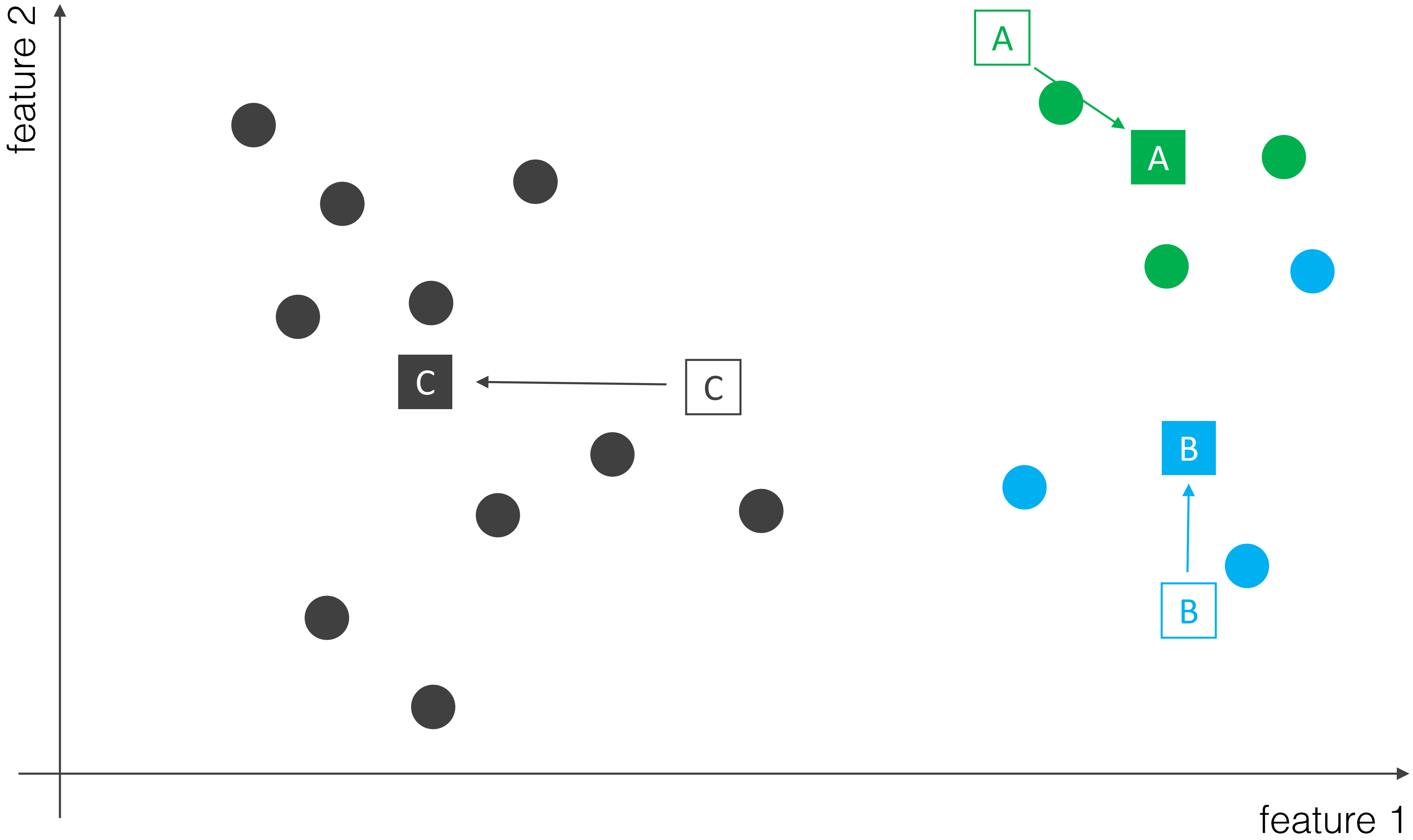
K-means clustering

1. Select k and randomly initialize k mean values

2. Assign observations to the nearest mean

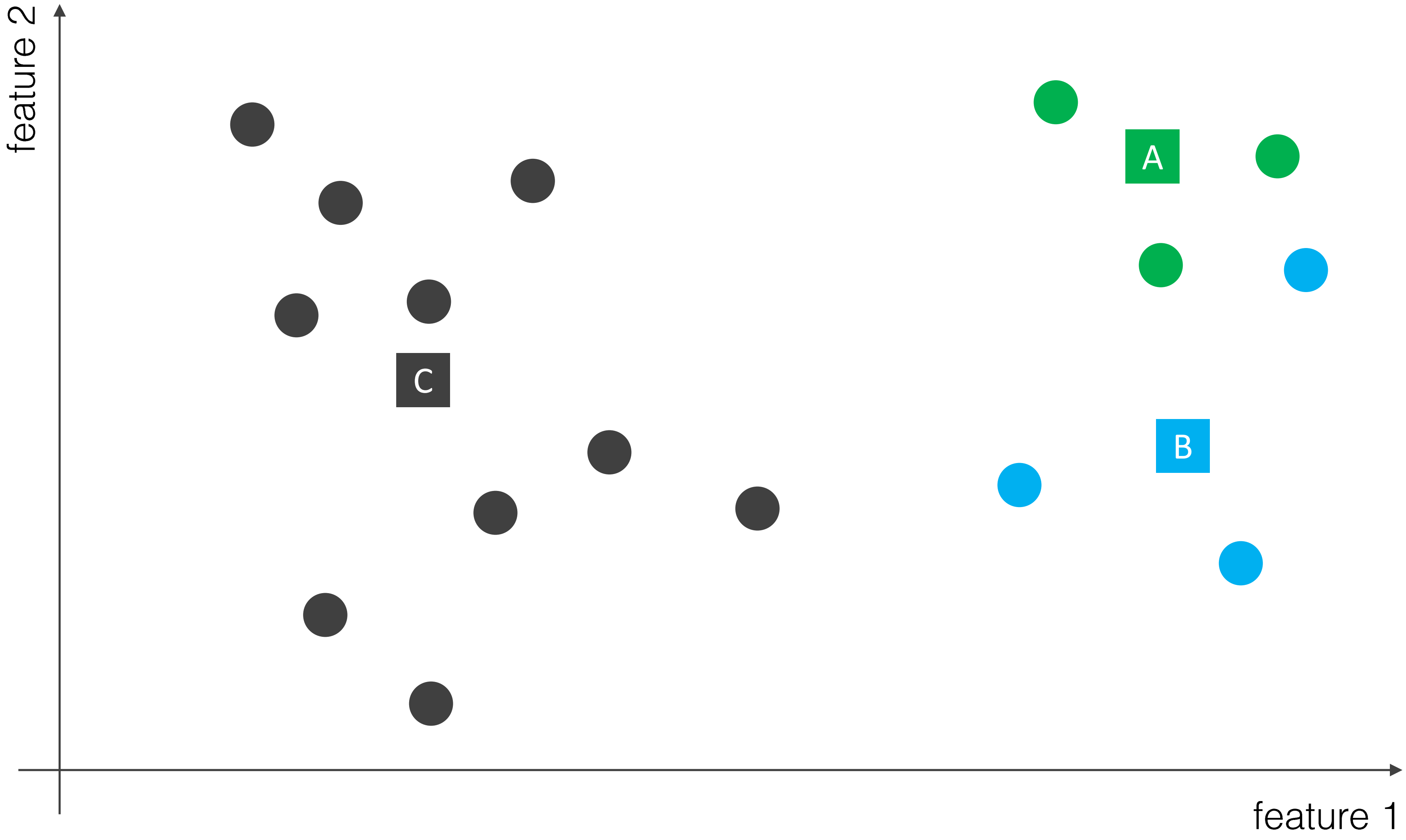


K-means clustering



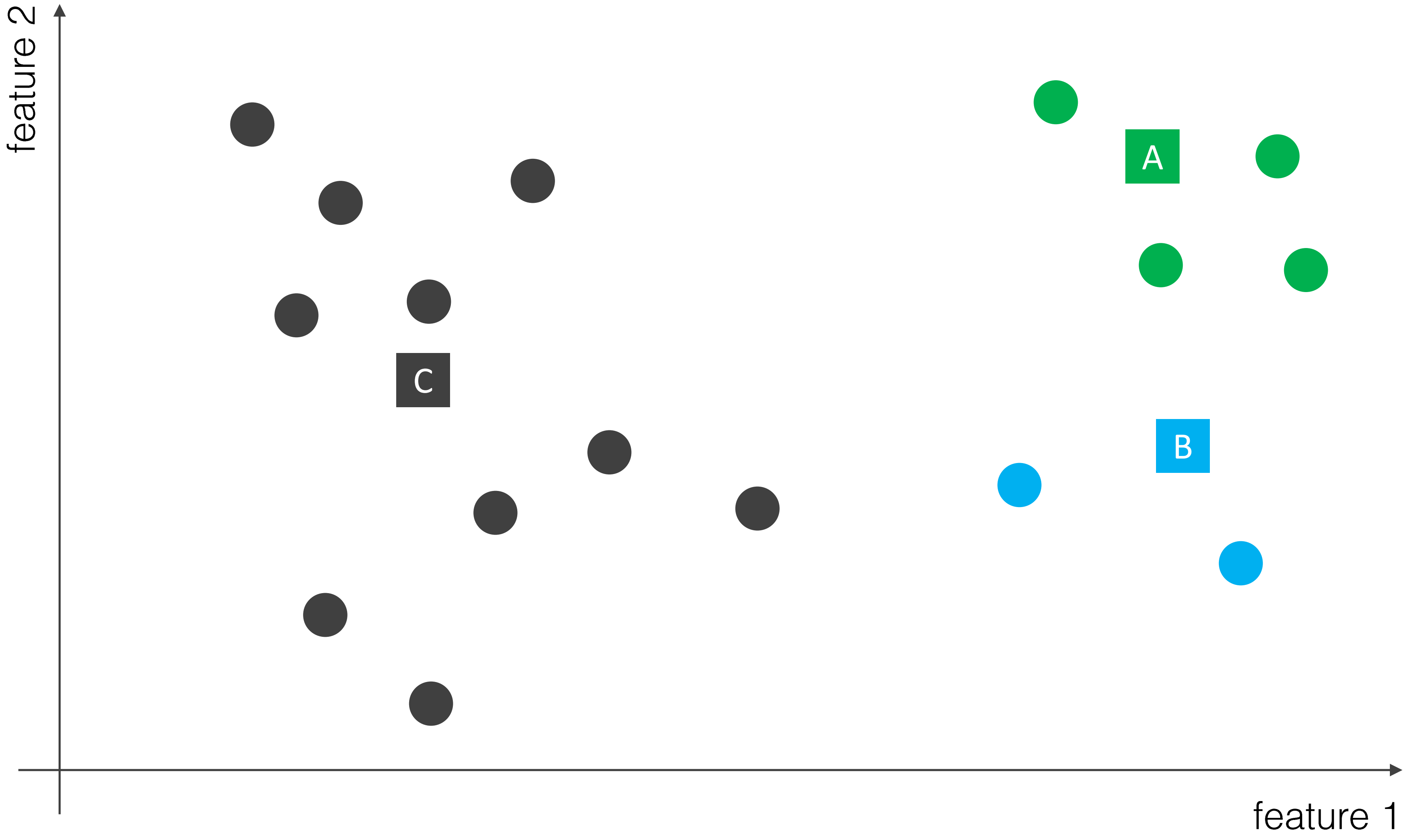
1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data

K-means clustering



1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

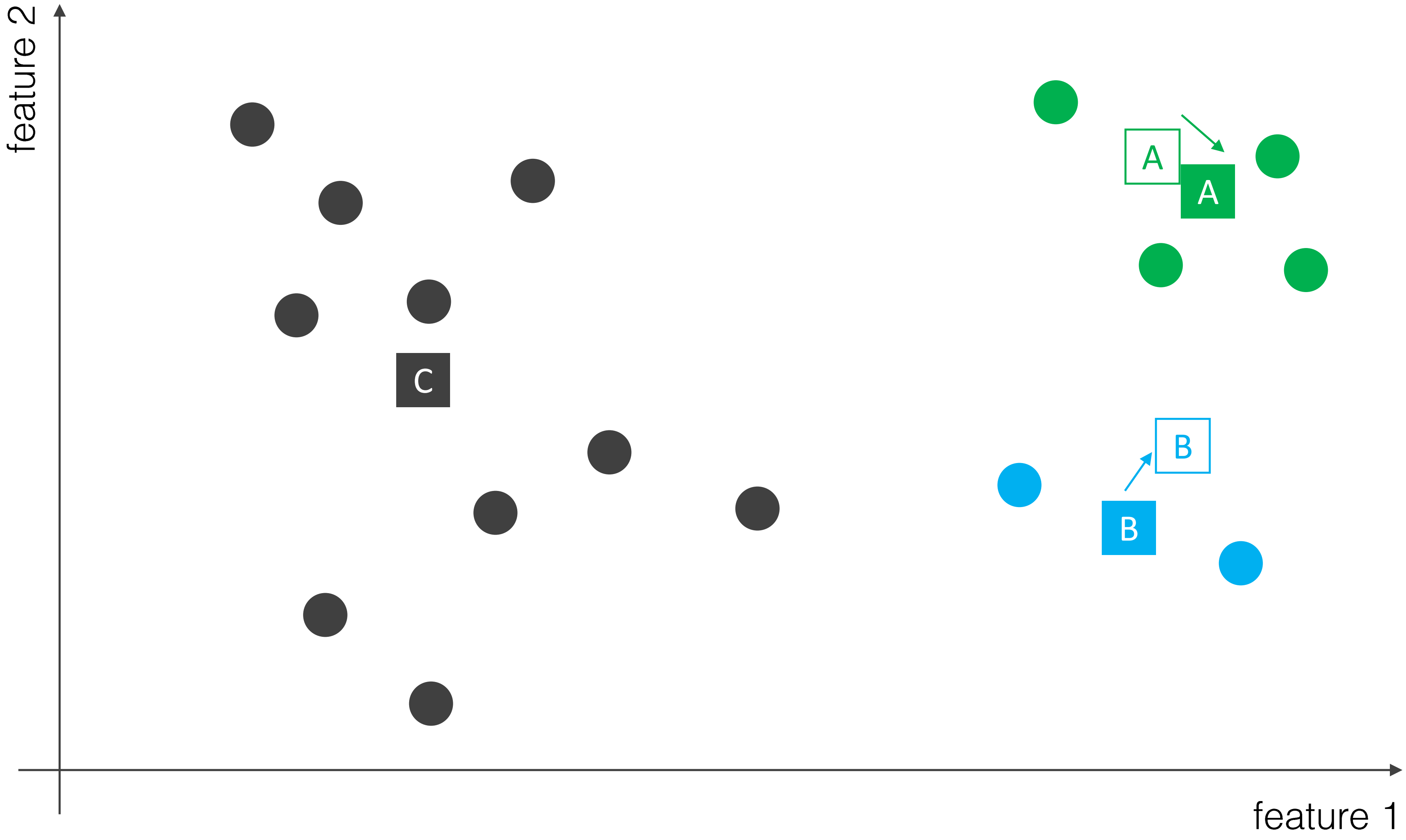
K-means clustering



1. Select k and randomly initial k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

…Iteration 2

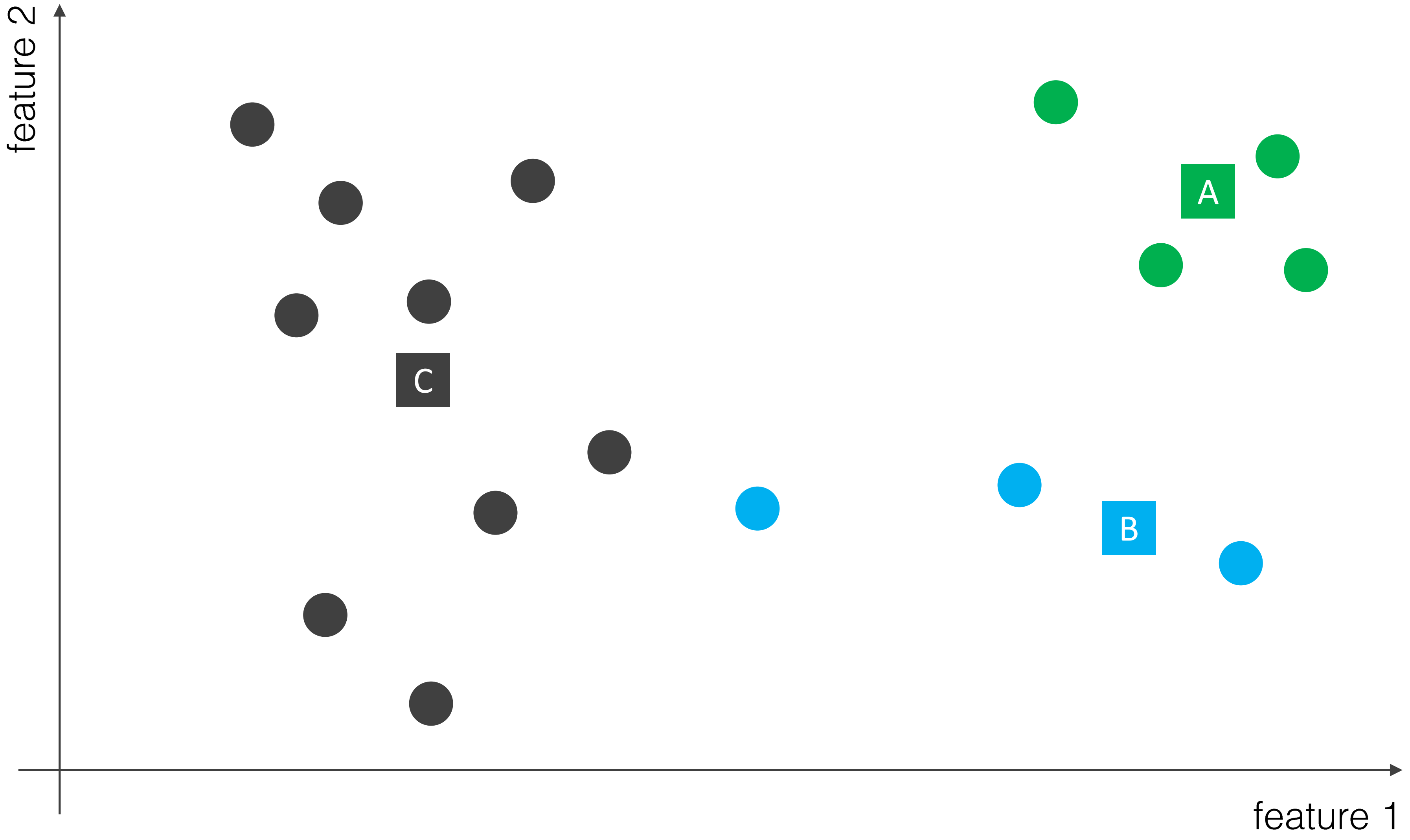
K-means clustering



1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

…Iteration 2

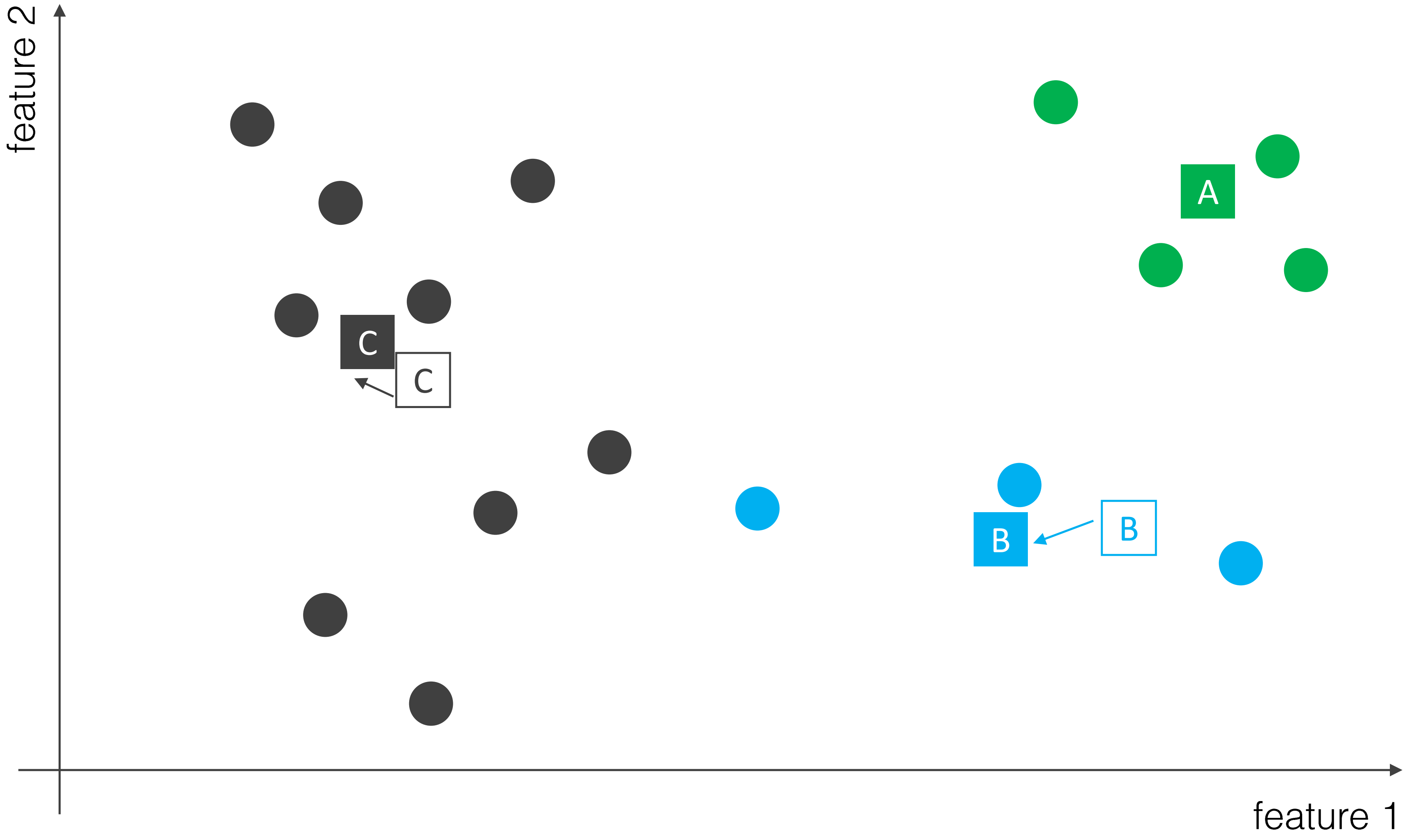
K-means clustering



1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

…Iteration 3

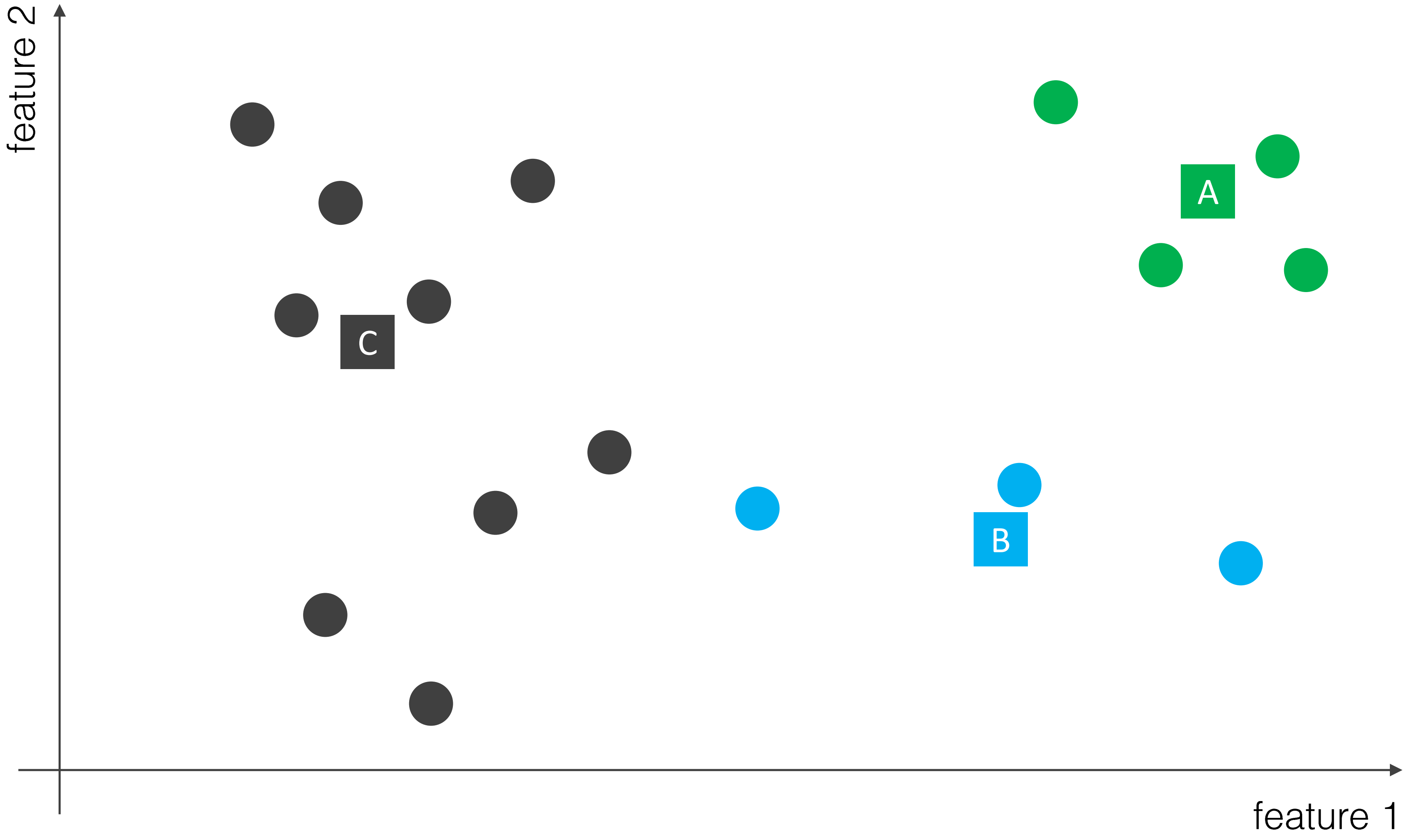
K-means clustering



1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

…Iteration 3

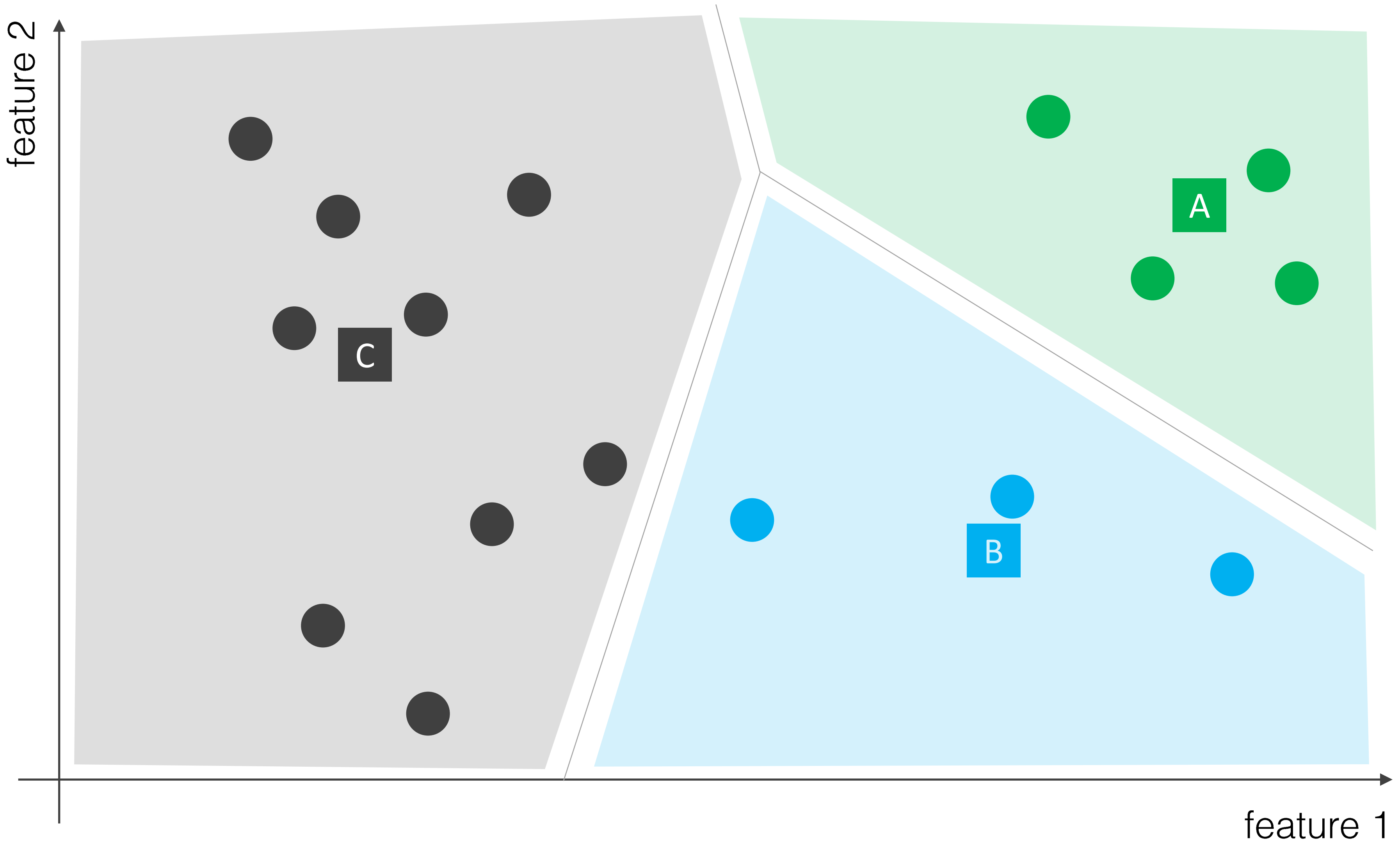
K-means clustering



1. Select k and randomly initialize k mean values
2. Assign observations to the nearest mean
3. Update the mean to be the centroid of the labeled data
4. Repeat steps 2 and 3 until convergence

…converged

K-means partitions the space into Voronoi cells



Under the hood, we minimize a cost function

Objective: identify K means, $\boldsymbol{\mu}_k$, such that the set of closest points in feature space are the minimum distance away.

$$r_{ik} = \begin{cases} 1 \text{ if } \boldsymbol{x}_j \text{ is closest to the kth mean } \boldsymbol{\mu}_k \\ 0 \text{ else} \end{cases}$$

L_2 norm

$$C(\boldsymbol{x}_i, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2$$

1. E-step

Re-evaluate r_{ik}

$$r_{ik} = \begin{cases} 1 \text{ if } \boldsymbol{x}_j \text{ is closest to the kth mean } \boldsymbol{\mu}_i \\ 0 \text{ else} \end{cases}$$

Assign new “expected” cluster labels

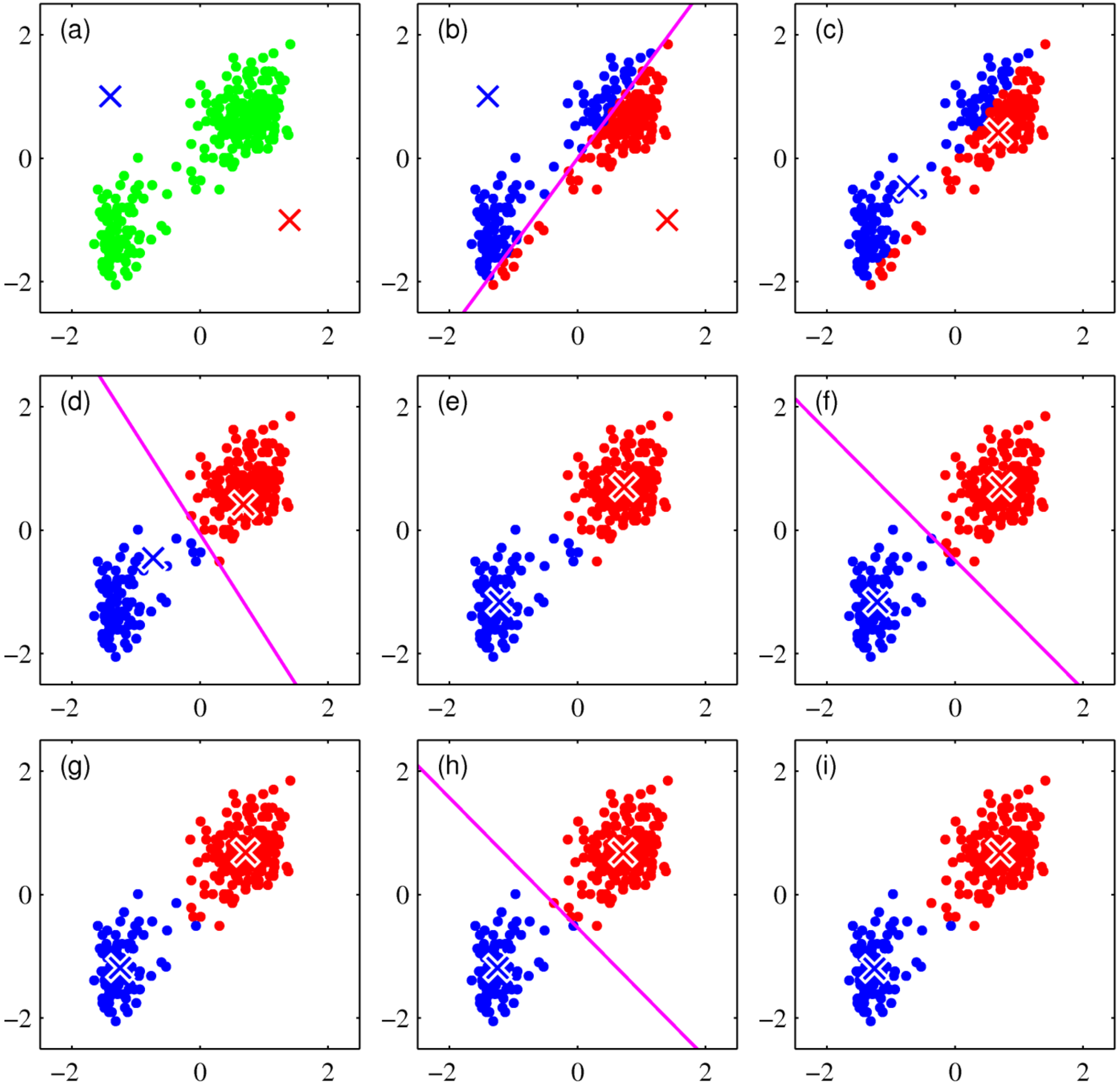
2. M-step

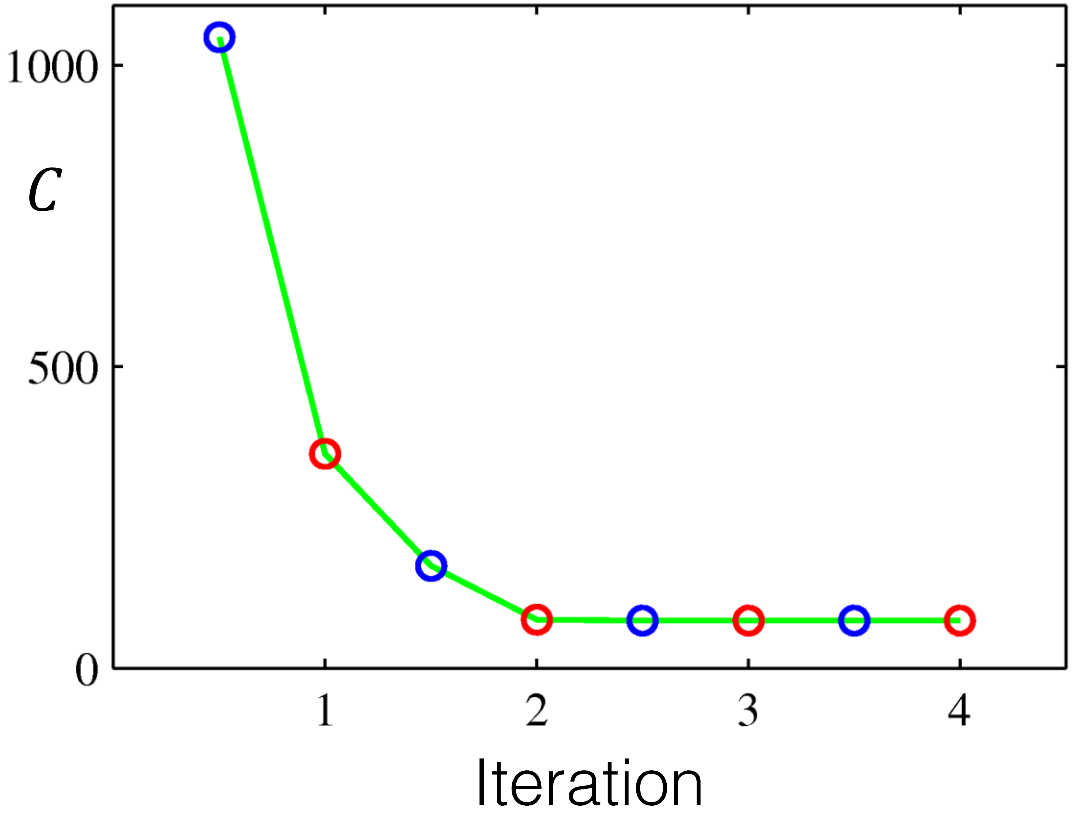
Minimize C wrt $\boldsymbol{\mu}_i$

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik} \, \boldsymbol{x}_i}{\sum_i r_{ik}}$$

Update the cluster means to maximize the likelihood

Convergence





Bishop, Pattern Recognition, 2006

How to choose k: Elbow method

Run k-means for various k

Choose the value of k at the "elbow" of the curve

Increasing k will improve the fit, but at the cost of potentially overfitting the data

Other approaches: silhouette (graphical approach to evaluating cluster fit), Akaike information criterion (AIC) and Bayesian information criterion (BIC) measure relative quality of models and factors in the number of parameters

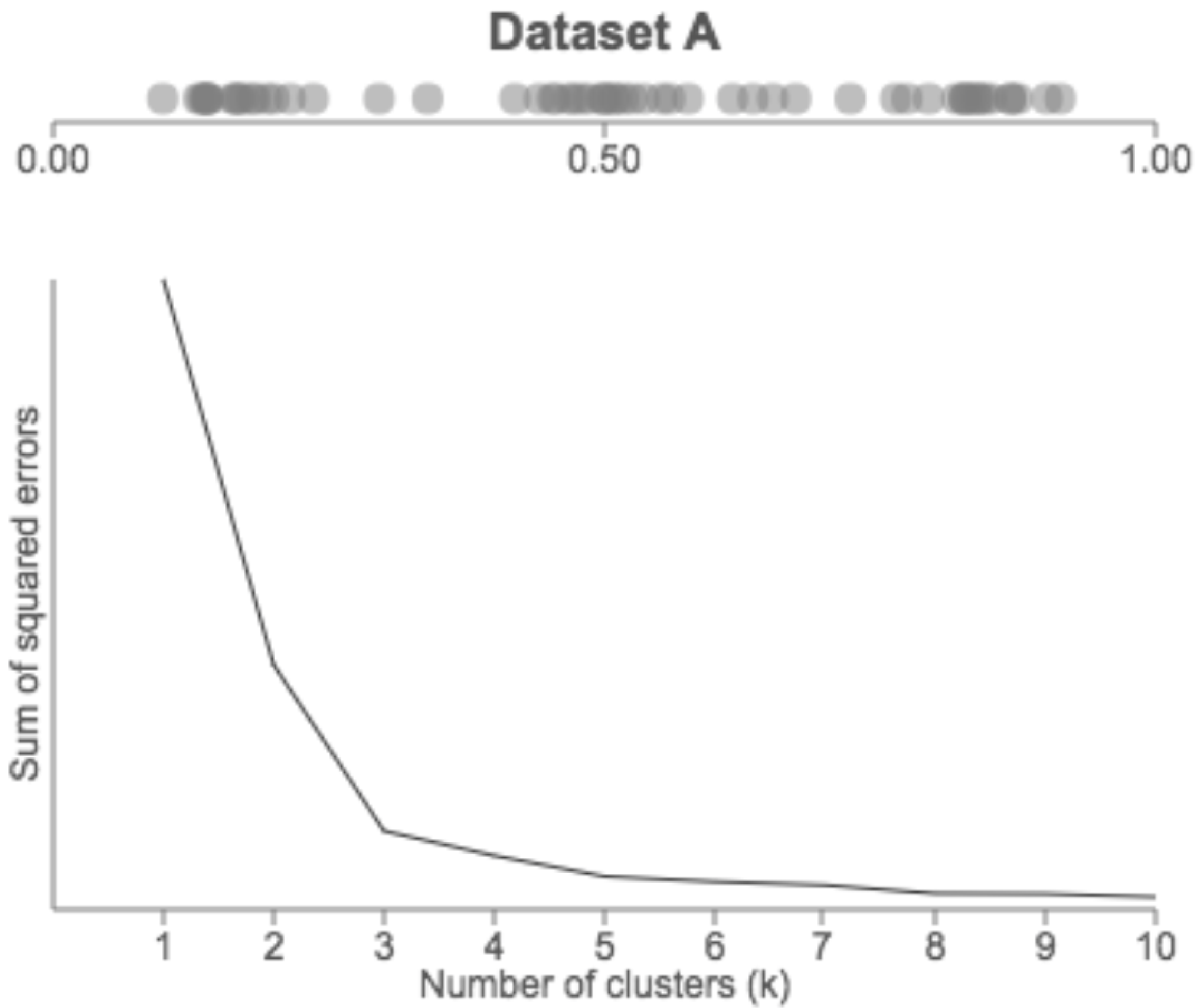


Image by Robert Gove: https://bl.ocks.org/rpgove/0060ff3b656618e9136b

Distance / dissimilarity measure

$$C(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} D(\boldsymbol{x}_i, \boldsymbol{\mu}_k)$$

Distance / dissimilarity measure

Distance measure (for distance from the mean):

Name of method:

L_2 norm $\quad D(\boldsymbol{x}_i, \boldsymbol{\mu}_k) = \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2 = \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^2$

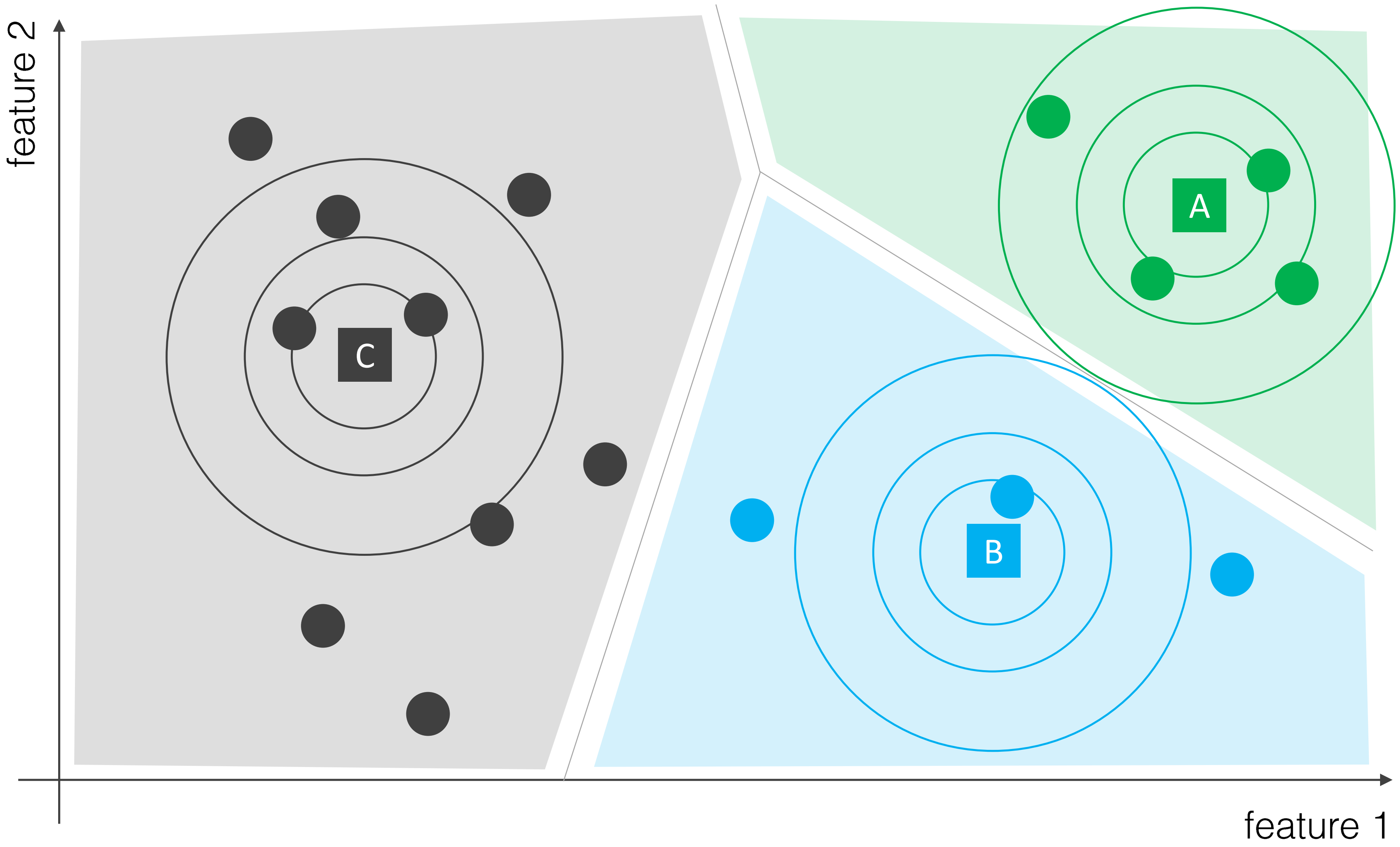
K-means

Generalization to **other distance measures**, e.g.:

K-mediods

L_1 norm $\quad D(\boldsymbol{x}_i, \boldsymbol{\mu}_k) = \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_1 = \sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{\mu}_k|$

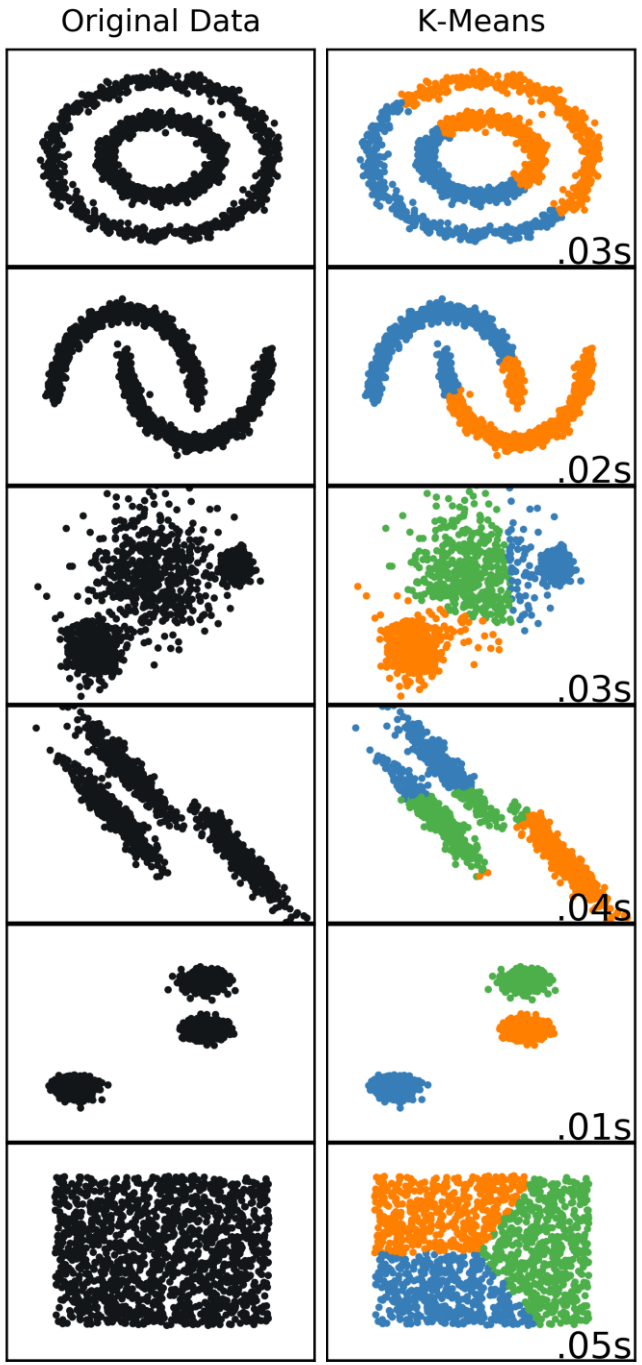
Relationship to Gaussian distributions



Assumes the clusters are **Gaussians** centered at the mean, each with **identical covariance matrices**, where all the features are independent:

$$\boldsymbol{\Sigma}_{\mathrm{k}} = \sigma^2 \boldsymbol{I} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

Examples: K-Means



Converges very quickly

Sensitive to initialization of means

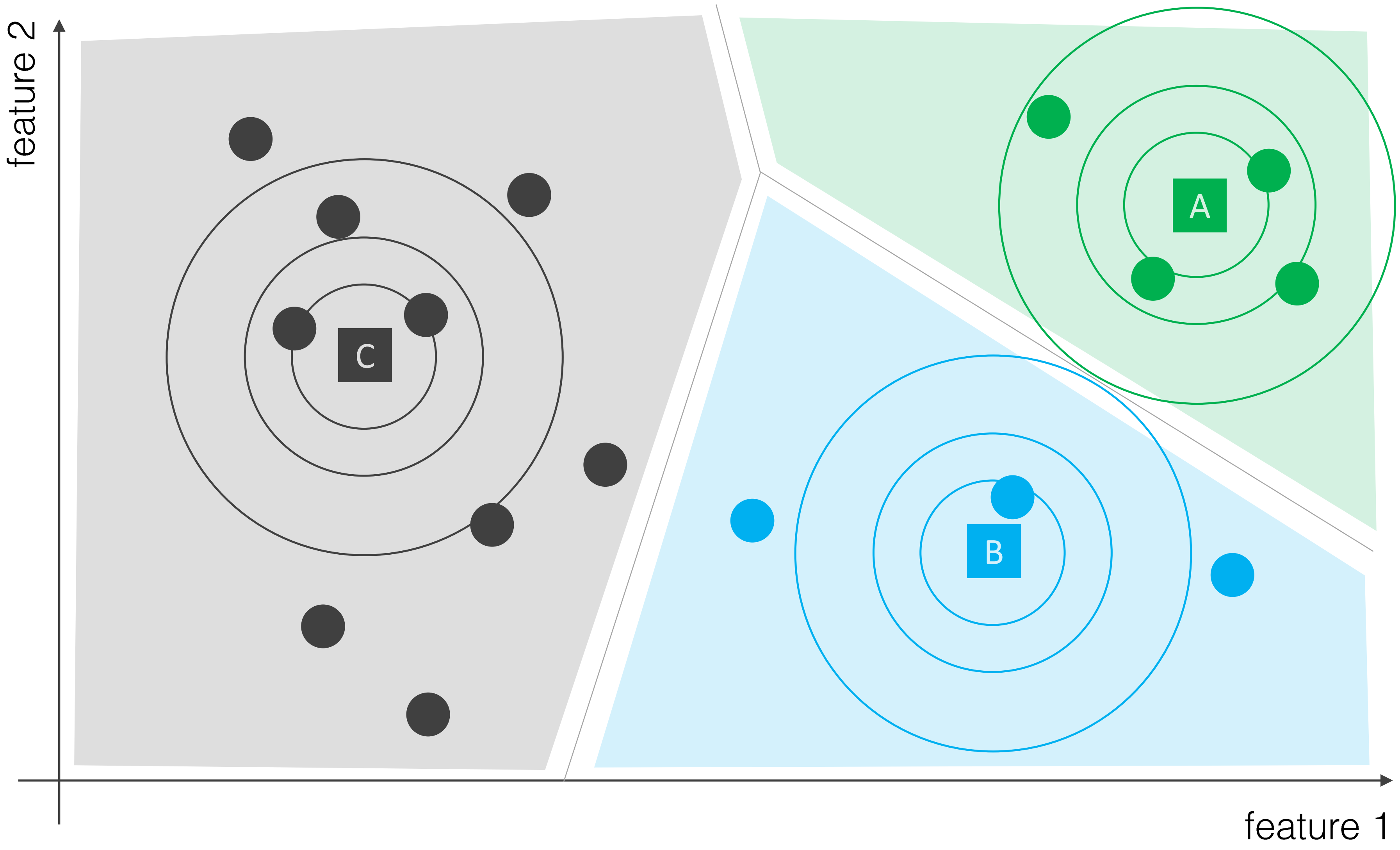
Struggles when there are **nonlinear** boundaries between clusters

Struggles in situations with **variation in cluster variance** and **correlation between features**

Excels with clusters of **equal variance**

Will divide into k clusters even when there are not k

Relaxing our assumptions on covariance…



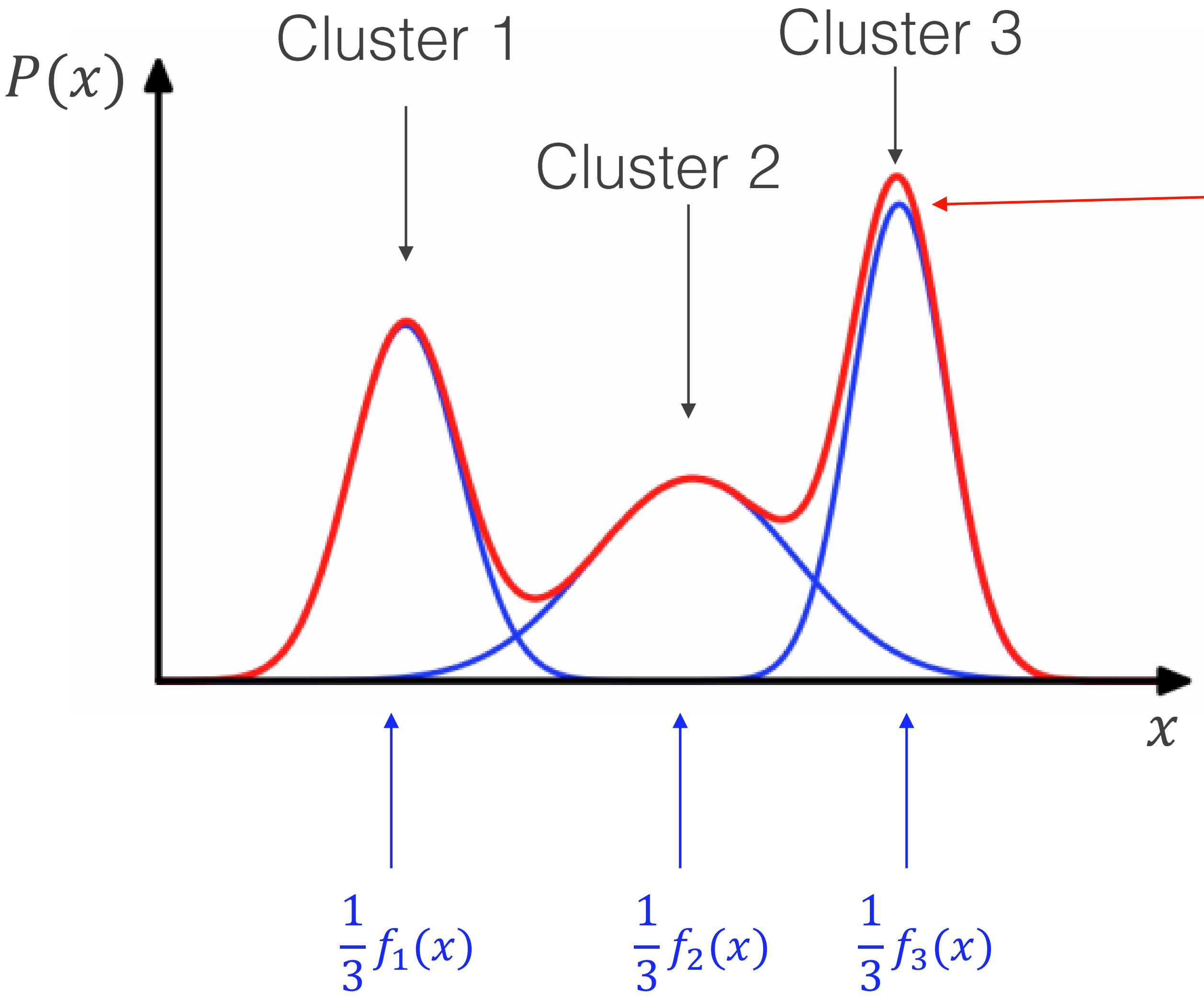
What if we **don't** assumes the clusters are **Gaussians** centered at the mean, each with **identical covariance matrices**, where all the features are independent:

$$\boldsymbol{\Sigma}_{\mathrm{k}} = \sigma^2 \boldsymbol{I} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

Gaussian Mixture Models

For clustering and density estimation

Mixture model



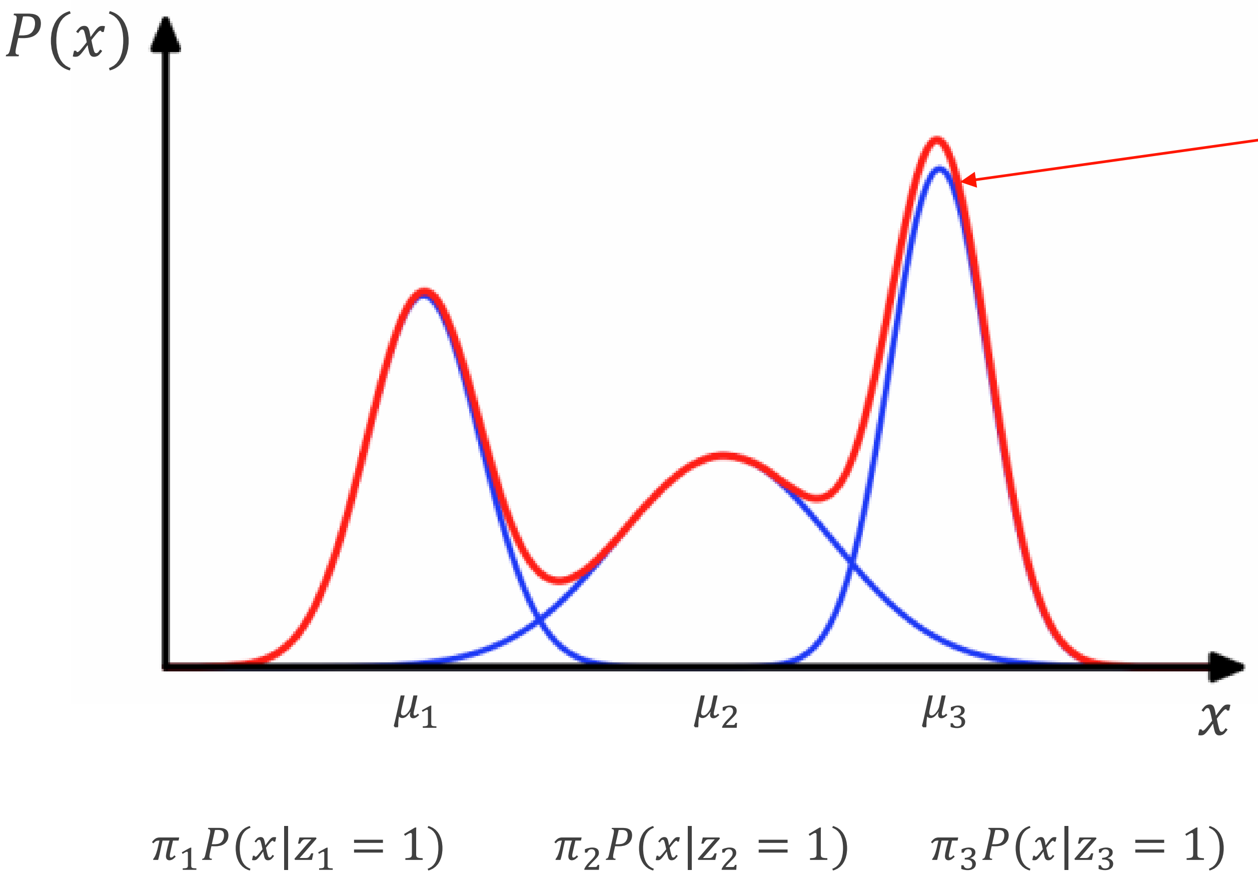
A weighted average of density functions

$$P(x) = \frac{1}{3}f_1(x) + \frac{1}{3}f_2(x) + \frac{1}{3}f_3(x)$$

1. Fit the model to the data
2. Use the model to assign clusters

Image from Shaun Dowling

Gaussian mixture model



A mixture model is represented as:

$$P(x) = \sum_{k=1}^{K} P(z_k = 1) P(x|z_k = 1)$$

If we assume this is Gaussian, it becomes a Gaussian mixture model (GMM)

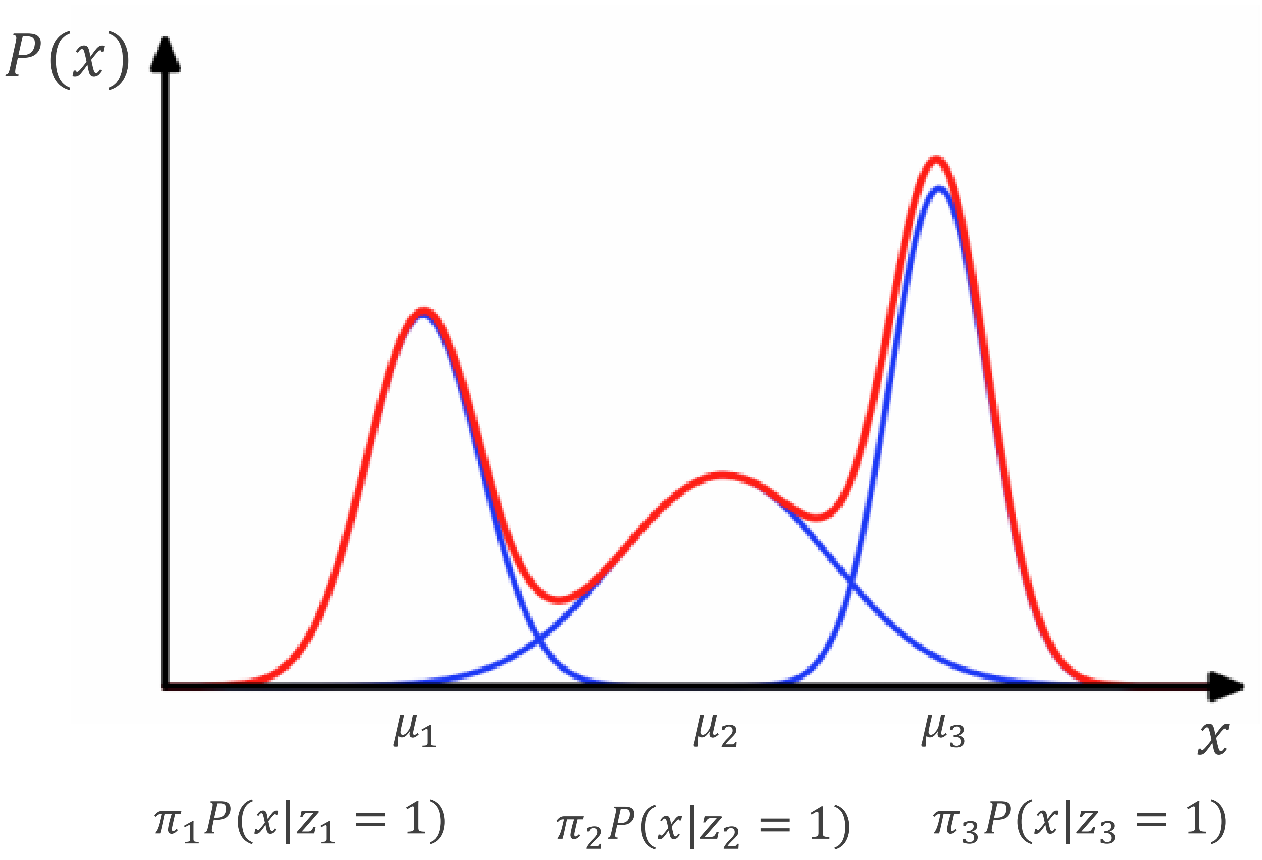
The mixing coefficients $\pi_k = P(z_k =$

$$\sum_{k=1}^{K} \pi_k = 1$$

z_k = binary variable that represents cluster membership

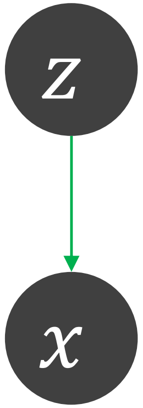
Image from Shaun Dowling

Gaussian mixture model



$$P(x) = \sum_{k=1}^{K} P(z_k = 1) P(x|z_k = 1)$$

Here we assume z is a **latent** (hidden / unobservable) variable



Hidden

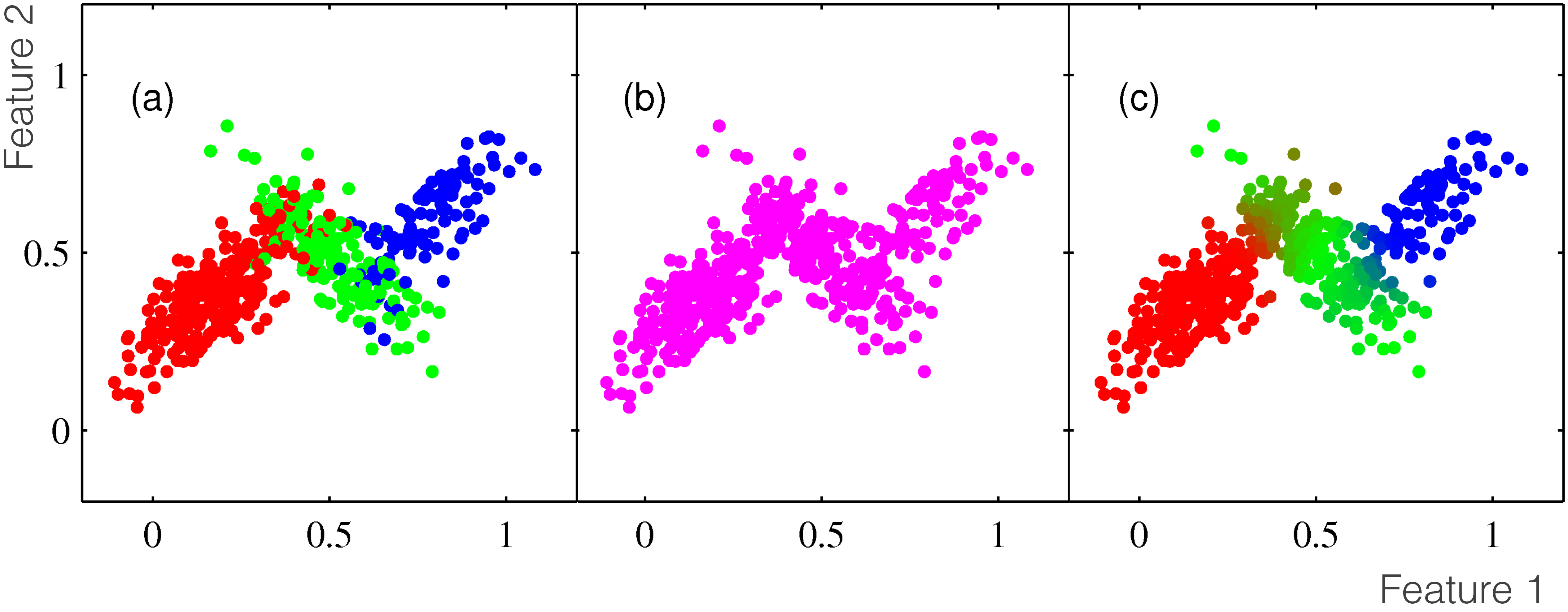
This variable controls which of the k mixture components a sample is drawn from

Observable

Given z, a sample is drawn from $P(x|z_k = 1)$

Image from Shaun Dowling

Gaussian Mixture Model Latent Variables



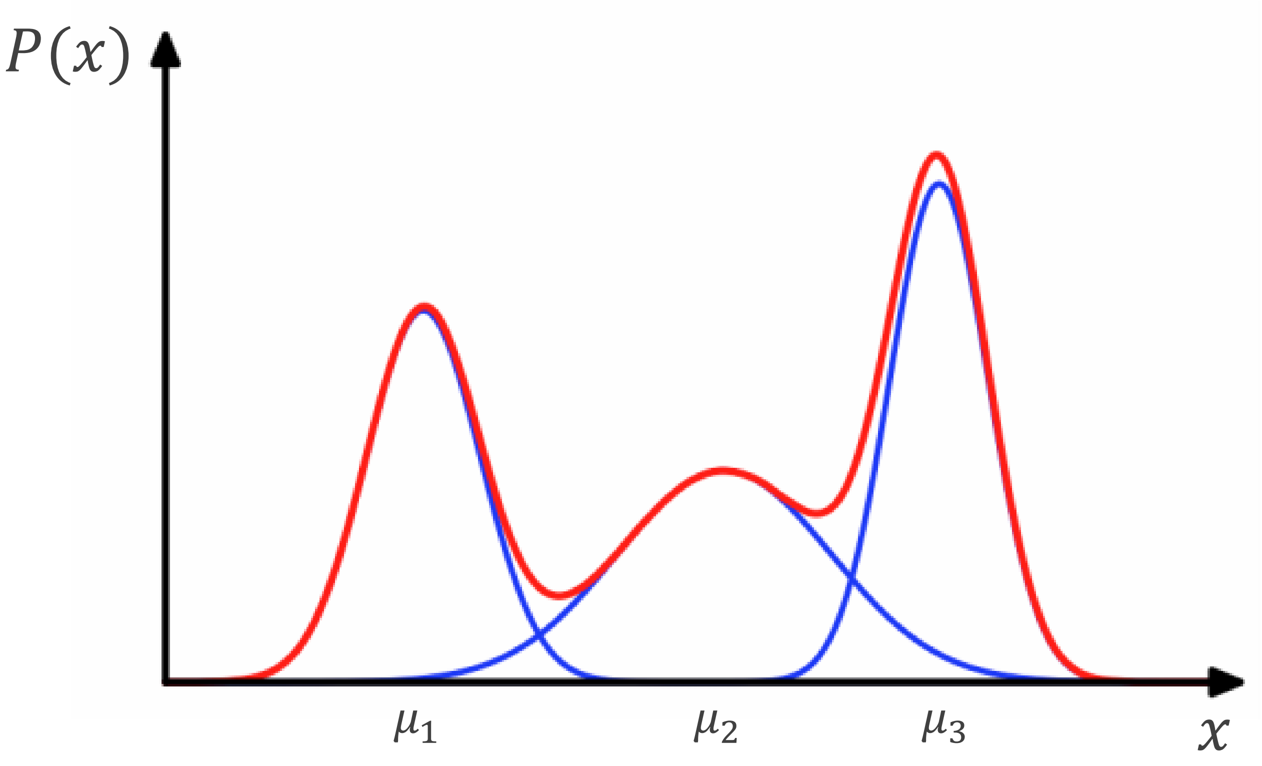
Complete data with latent variable labels z

Incomplete data without latent variable labels

Posterior probabilities, a.k.a. responsibilities

Image from Bishop, Pattern Recognition, 2006

Gaussian mixture model



$\pi_1 N(x|\mu_1, \sigma_1^2)$ $\pi_2 N(x|\mu_2, \sigma_2^2)$ $\pi_3 N(x|\mu_3, \sigma_3^2)$

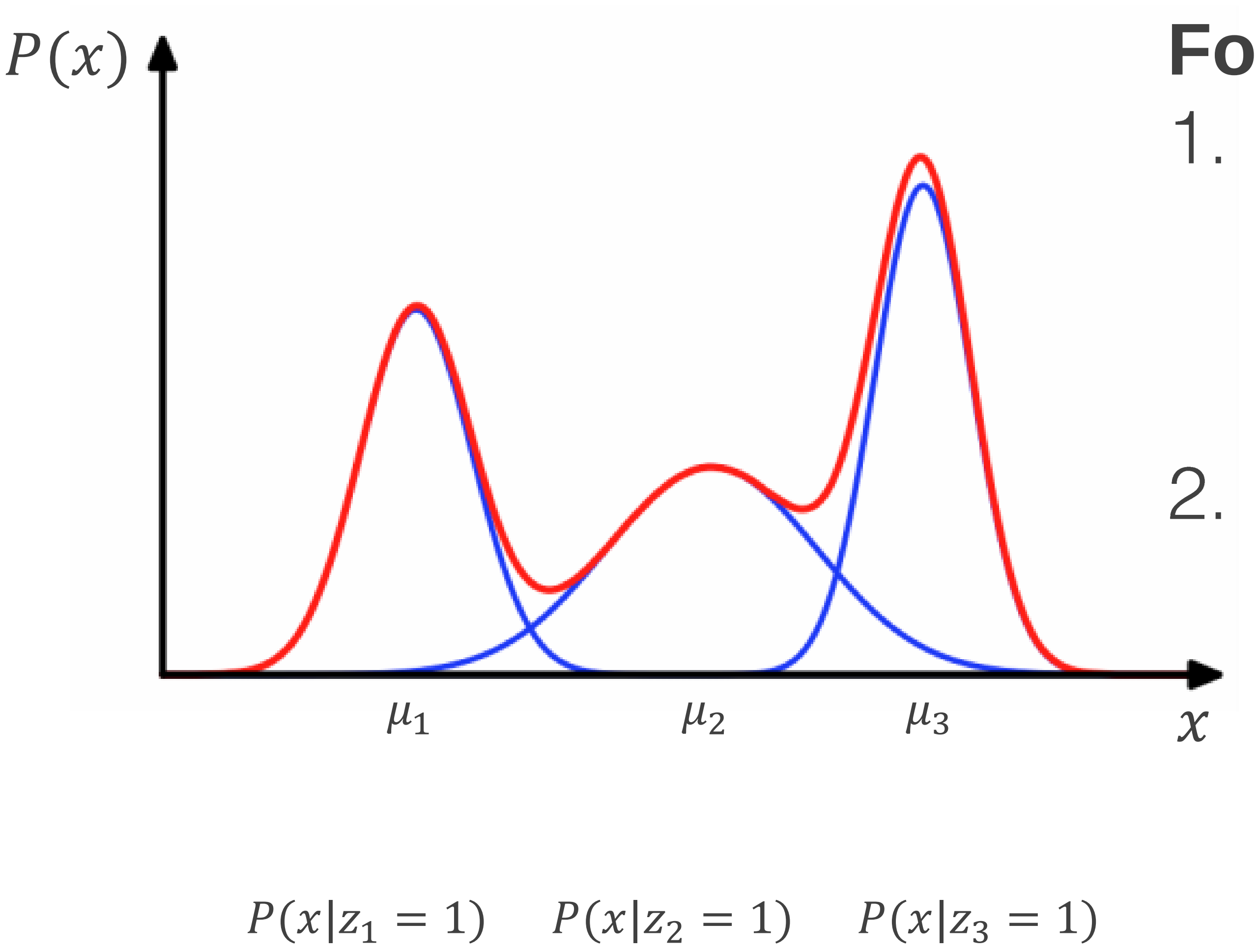
The Gaussian mixture model is represented as:

$$P(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k^2)$$

where $$\sum_{k=1}^{K} \pi_k = 1$$

Image from Shaun Dowling

Gaussian mixture model



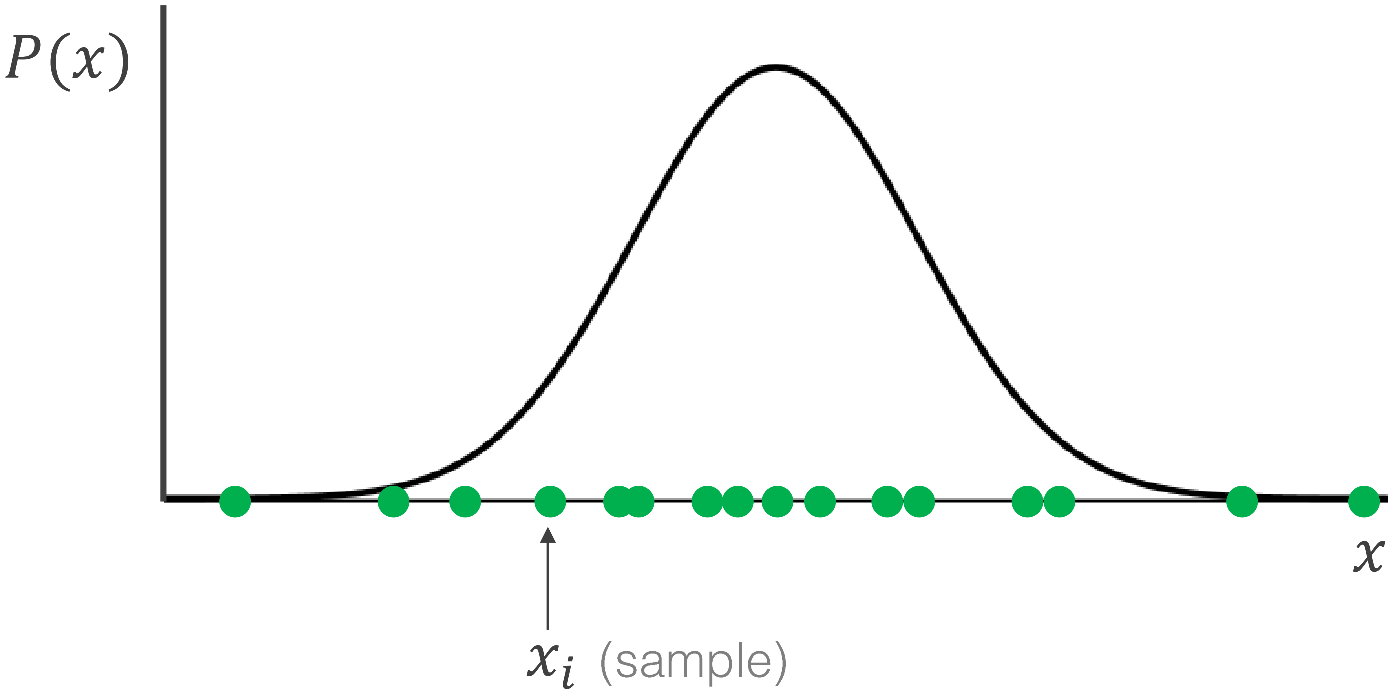
For clustering:

1. Fit a GMM to the data (estimate π_k, μ_k, σ_k^2 for $k = 1, \dots, K$ to maximize the likelihood of the data given the model)
2. Pick the cluster, z_k, that each data point was most likely to come from

Image from Shaun Dowling

Density estimation for a single mixture component

a.k.a. model fitting



Likelihood of one sample given the model

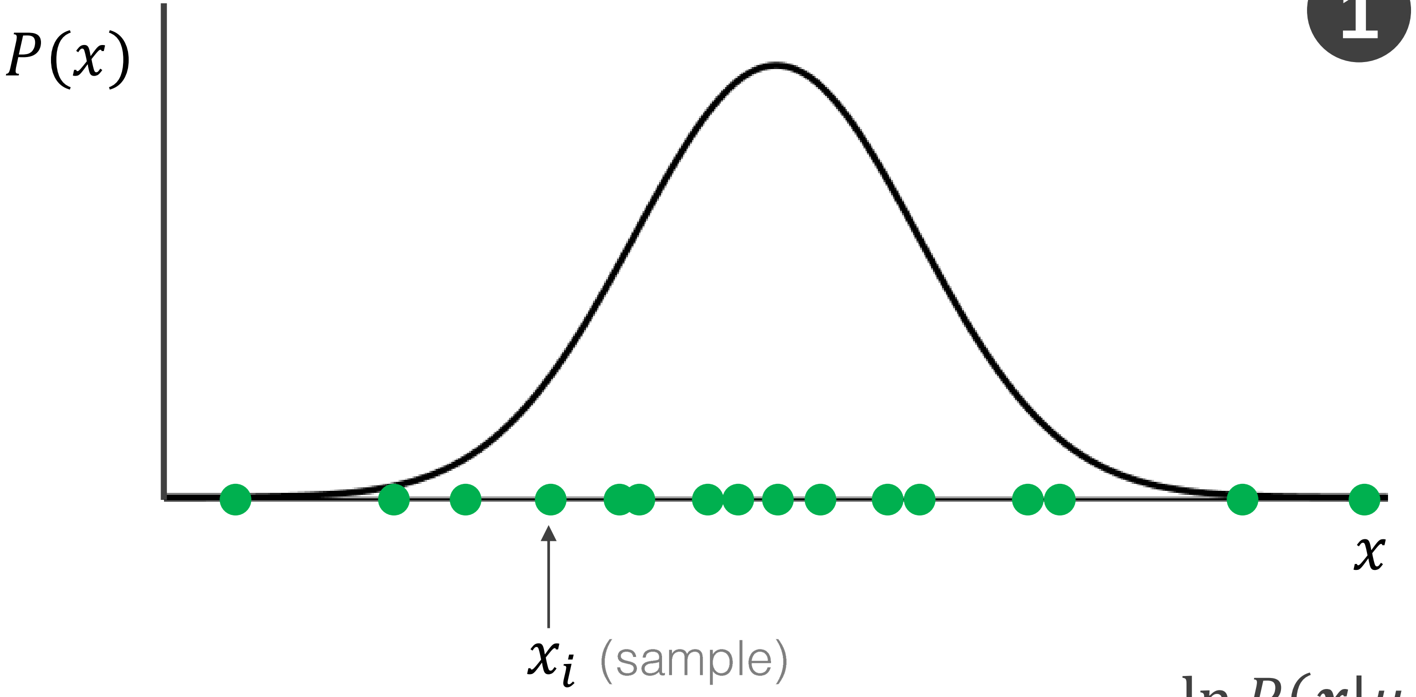
$$P(x_i|\mu,\sigma^2) = N(x_i|\mu,\sigma^2)$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Assuming independent samples, the likelihood of the data given the model is:

$$P(\boldsymbol{x}|\mu,\sigma^2)$$
$$= \prod_{i=1}^{N} P(x_i|\mu,\sigma^2)$$
$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Density estimation for a single mixture component

a.k.a. model fitting



1 We follow our familiar pattern: maximize the likelihood of the data by choosing our model parameters: μ, σ^2

$$P(\boldsymbol{x}|\mu,\sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

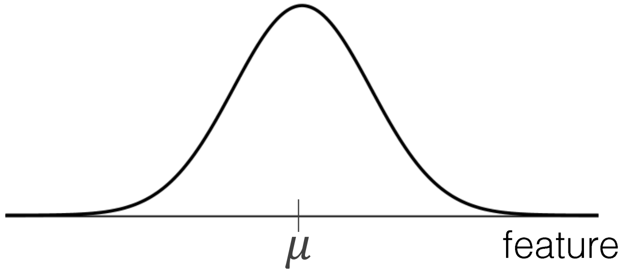
2 Calculate the log likelihood:

$$\ln P(\boldsymbol{x}|\mu,\sigma^2) = -\frac{N}{2}\ln 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2$$

3 Take the derivative of the log likelihood w.r.t. each parameter (μ, σ^2), set equal to zero, solve for μ, σ^2

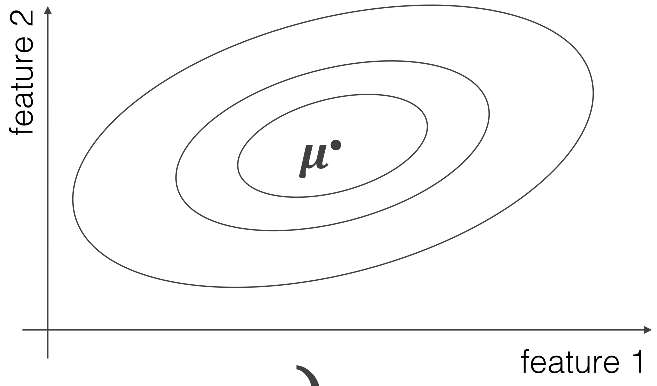
$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{\mu})^2$$

From a univariate to a multivariate Gaussian



Univariate Normal density

$$N(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}$$



Multivariate Normal density

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

From a univariate to a multivariate Gaussian

Univariate Normal MLE parameter estimates:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Multivariate Normal MLE parameter estimates:

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \qquad \widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T$$

Density estimation for a Gaussian mixture model

0 We define the likelihood of one observation given our model with parameters $\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k = 1, \dots, K$

$$P(\boldsymbol{x}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

1 We assume the observations are independent and calculate the likelihood for all our data

$$P(\boldsymbol{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

2 Calculate the log likelihood:

$$\ln P(\boldsymbol{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]$$

3 Take the derivative of the log likelihood w.r.t. each parameter ($\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k = 1, \dots, K$), set equal to zero, solve for the parameters

Density estimation for a Gaussian mixture model

Log likelihood of the data given the model parameters

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln\left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]$$

There is no **closed-form solution** that maximizes this.

We could use gradient descent BUT this approach can suffer from **severe overfitting**

Example: $k = 2$ mixture components

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln[\pi_1 N(\boldsymbol{x}_i|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) + \pi_2 N(\boldsymbol{x}_i|\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)]$$

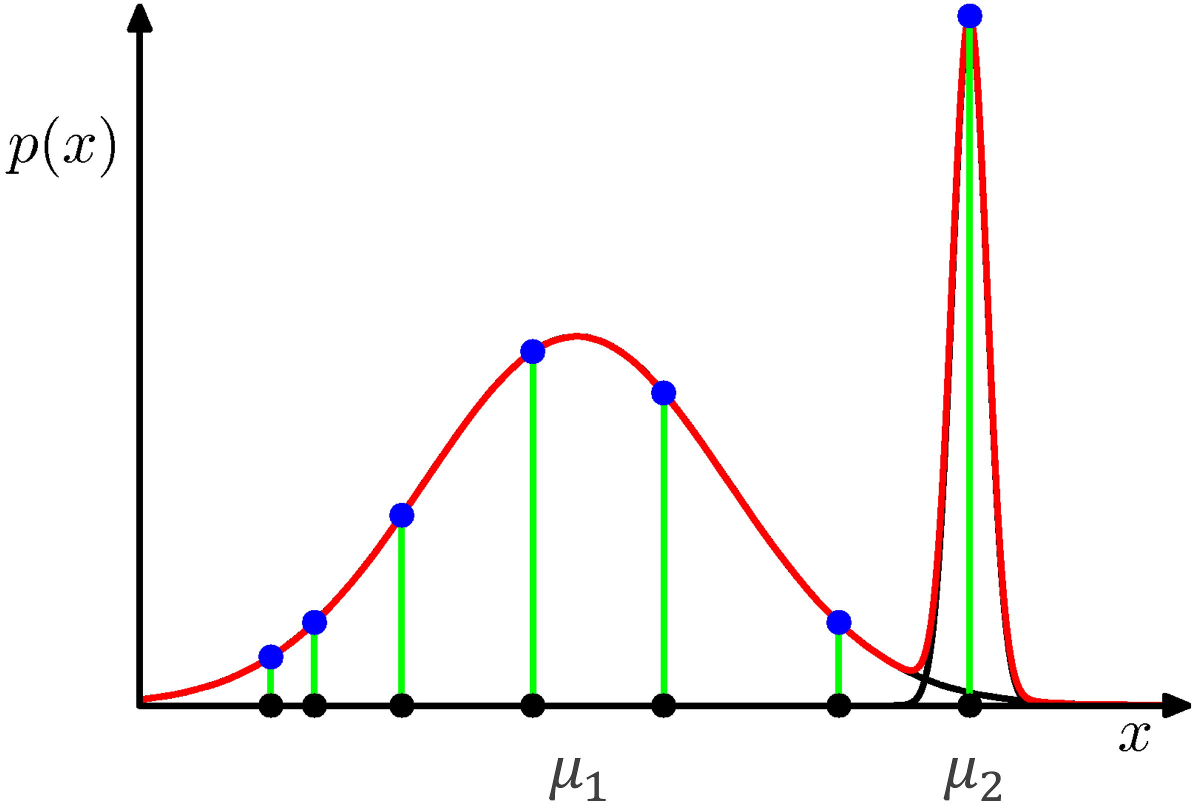
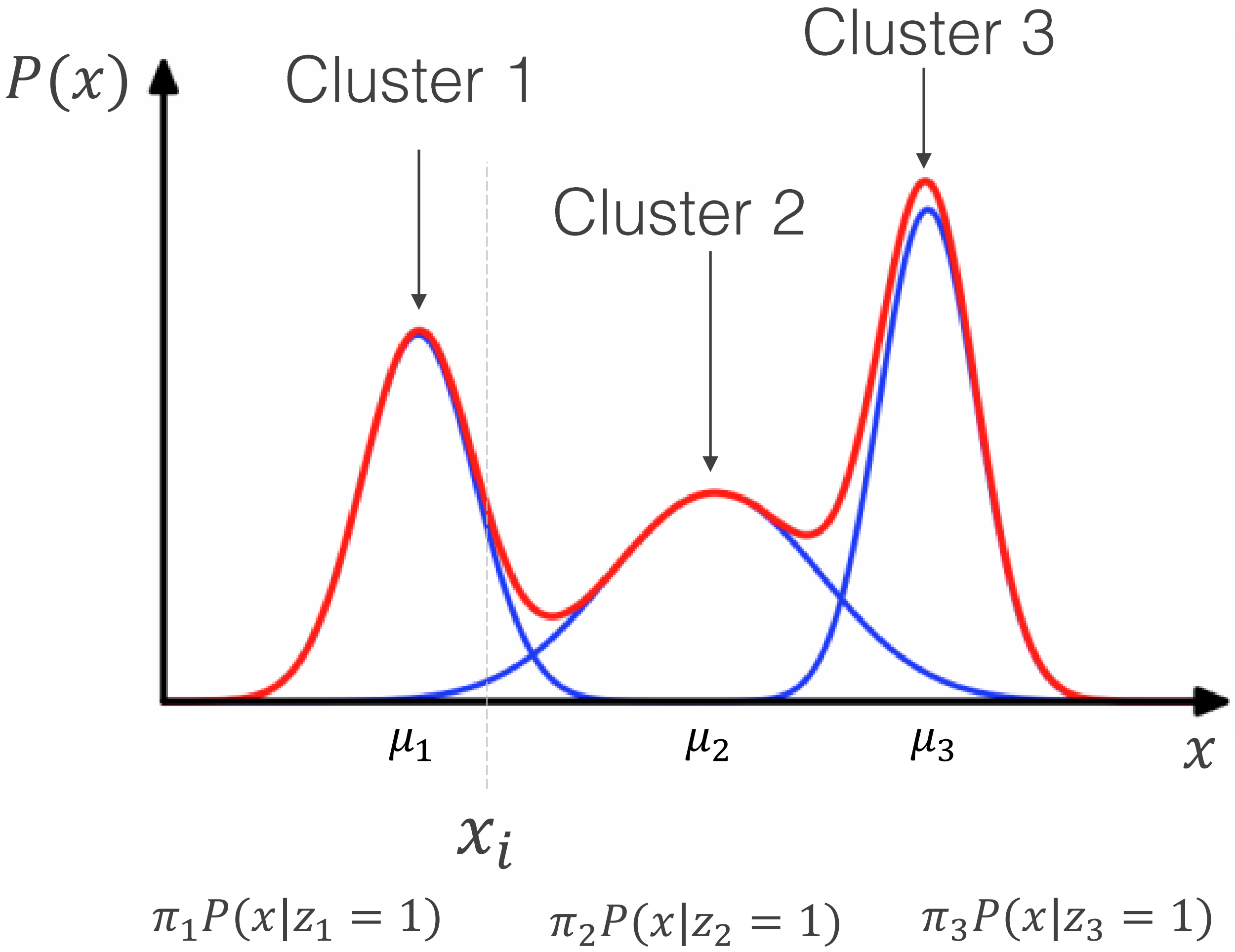


Image from Bishop, Pattern Recognition, 2006

How do we assign a cluster?

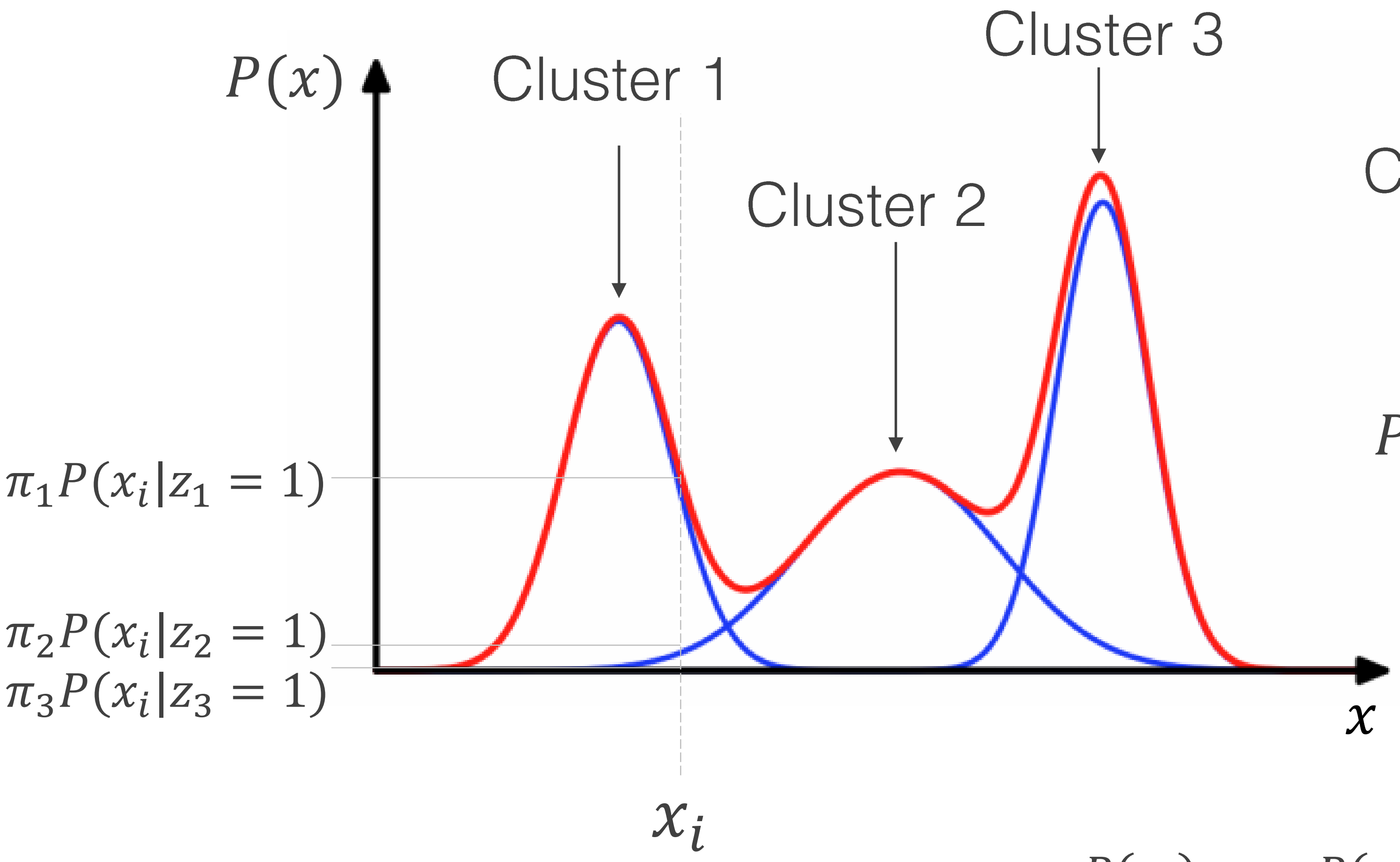


The probability of x_i is “explained” most by cluster 1, a little by cluster 2, and very little by cluster 3

We assign the cluster, z_k so that $P(z_k = 1|x)$ is the largest for all the k’s

We need an expression for: $P(z_k = 1|x)$

How do we assign a cluster?



Consider observation x_i

normal distribution
for the kth cluster

π_k

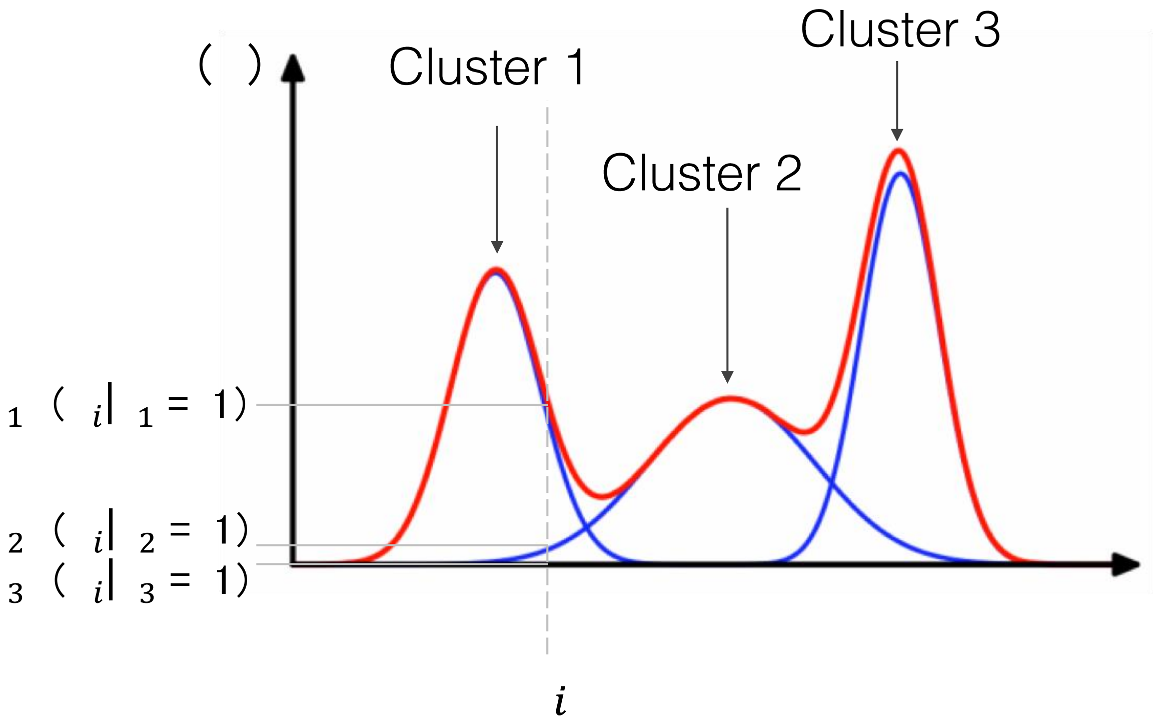
$$P(z_k = 1|x_i) = \frac{P(x_i|z_k = 1)P(z_k = 1)}{P(x_i)}$$

by Bayes' Rule

$$P(x_i) = \pi_1 P(x_i|z_1 = 1) + \pi_2 P(x_i|z_2 = 1) + \pi_3 P(x_i|z_3 = 1)$$

normalizes the probability, $P(z_k = 1|x_i)$, to add to one when summed over k

Posterior probabilities / "responsibilities"



Another interpretation of this quantity is what "fraction" of an observation is assigned to this cluster ("fuzzy" or "soft" clustering)

$N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ π_k

$$\gamma(z_{ik}) \triangleq P(z_k = 1|x_i) = \frac{P(x_i|z_k = 1)P(z_k = 1)}{\sum_{k=1}^{K} P(x_i|z_k = 1)P(z_k = 1)}$$

$$= \frac{\pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

Define $N_k = \sum_{i=1}^{N} \gamma(z_{ik})$

Expected number of samples per cluster

Expectation Maximization for a GMM

Goal: maximize the log likelihood of the data given the model parameters:

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln\left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]$$

0. Initialization

Initialize all the parameters
(often K-means is used for this purpose)

1. Expectation-step

Calculate the “responsibilities” based on the model parameters

$$\gamma(z_{ik}) \triangleq P(z_k = 1|x_i)$$

$$= \frac{\pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}$$

2. Maximization-step

Use the “responsibilities” to update the model parameters to maximize the log likelihood

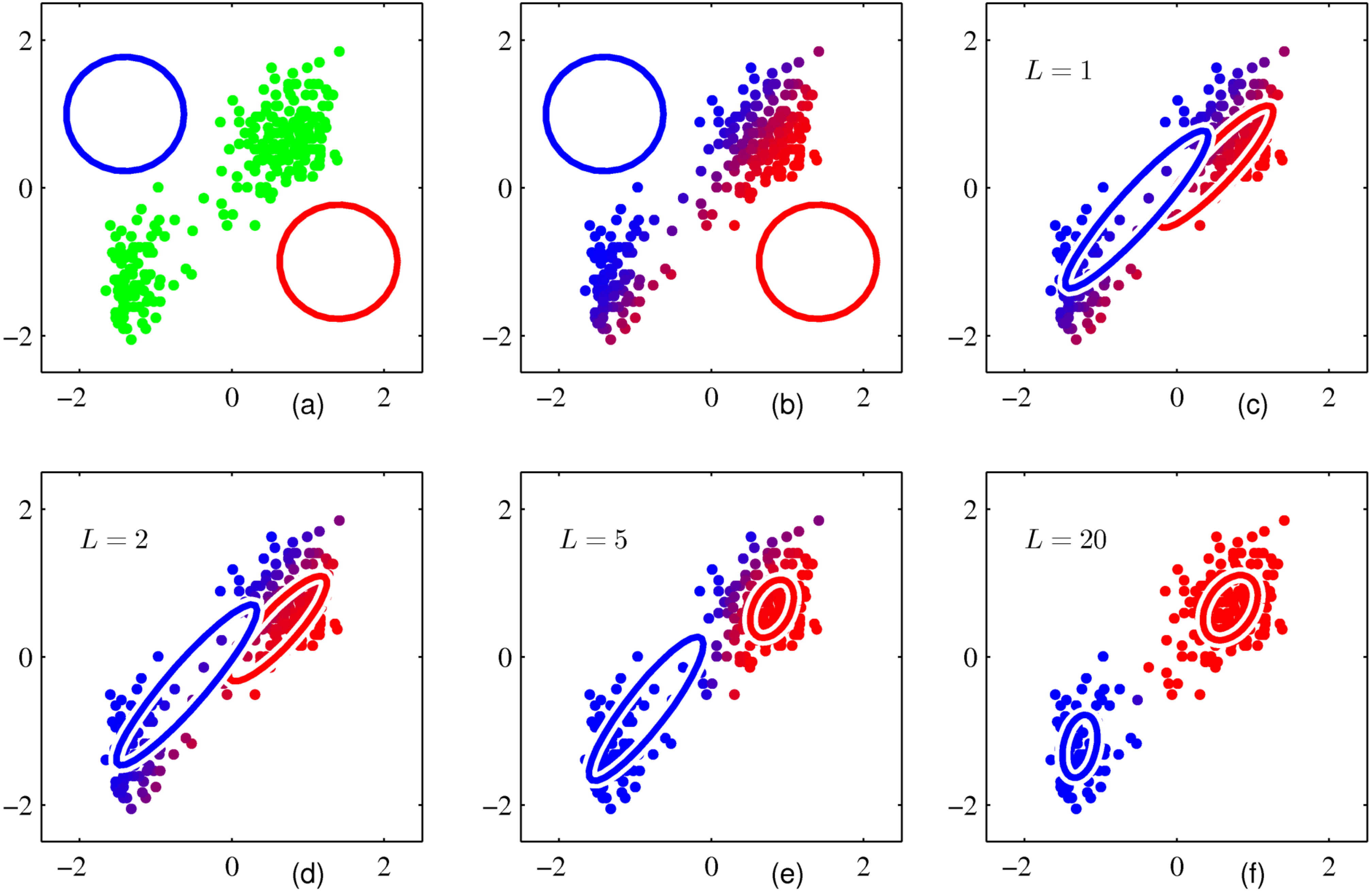
$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k}\sum_{i=1}^{N} \gamma(z_{ik})\boldsymbol{x}_i$$

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

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

Where $N_k = \sum_{i=1}^{N} \gamma(z_{ik})$

Expectation Maximization for GMM Example



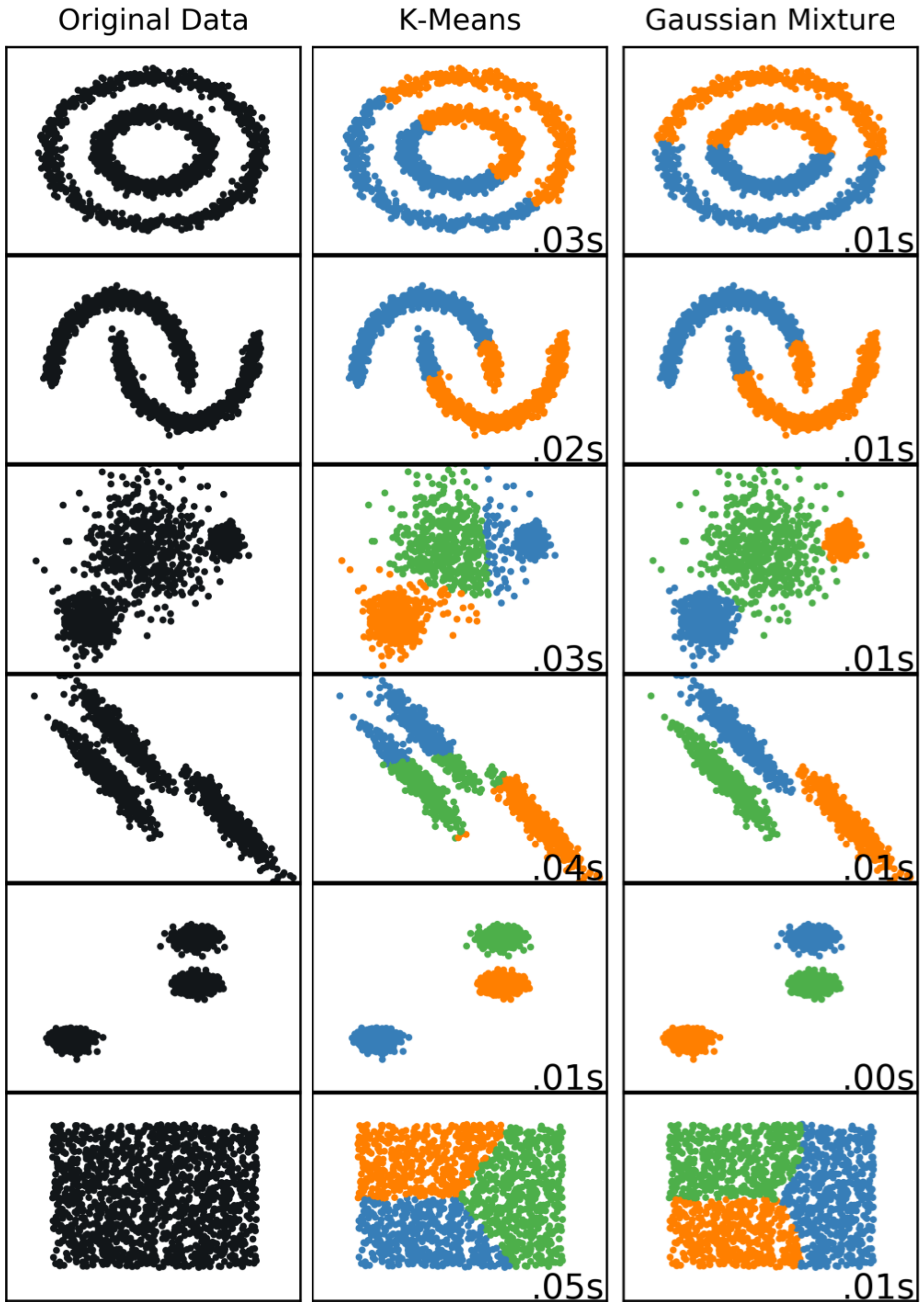
L = number of EM cycles

Image from Bishop, Pattern Recognition, 2006

Examples: GMM

Can produce soft clustering

Estimates the density / distribution of the data



Struggles when the clusters are not approximately Gaussian

Excels in situations with **variation in cluster variance** and **correlation between features**

Excels with clusters of **equal variance**

Will divide into k clusters even when there are not k

Gaussian Mixture Models

Generative models: model $P(\boldsymbol{X}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the model parameters

Very useful for density estimation

Produce hard or soft (fuzzy) clustering

When you restrict the covariance matrix to be diagonal and equal for all clusters, the GMM and K-means algorithm become the same

Expectation Maximization

Iterative method to find maximum likelihood parameter estimates when the model depends on unobserved latent variables, when this can’t be solved directly

The E-step updates the latent variable distribution estimates, so that we can calculate the likelihood function given the current parameter values

The M-step identifies the parameters that maximize the likelihood

Types of clustering algorithms

Methods

Centroid-based clustering (e.g. **K-Means**)
Distribution-based clustering (e.g. **Gaussian mixture model**)
Density-based clustering (e.g. DBSCAN)
Hierarchical clustering (e.g. agglomerative clustering)
 a.k.a. connectivity-based clustering

Cluster assignment

Hard clustering
Soft clustering (a.k.a. fuzzy clustering)