Clustering

Yi Ding, University of Chicago & JulyEdu

Clustering

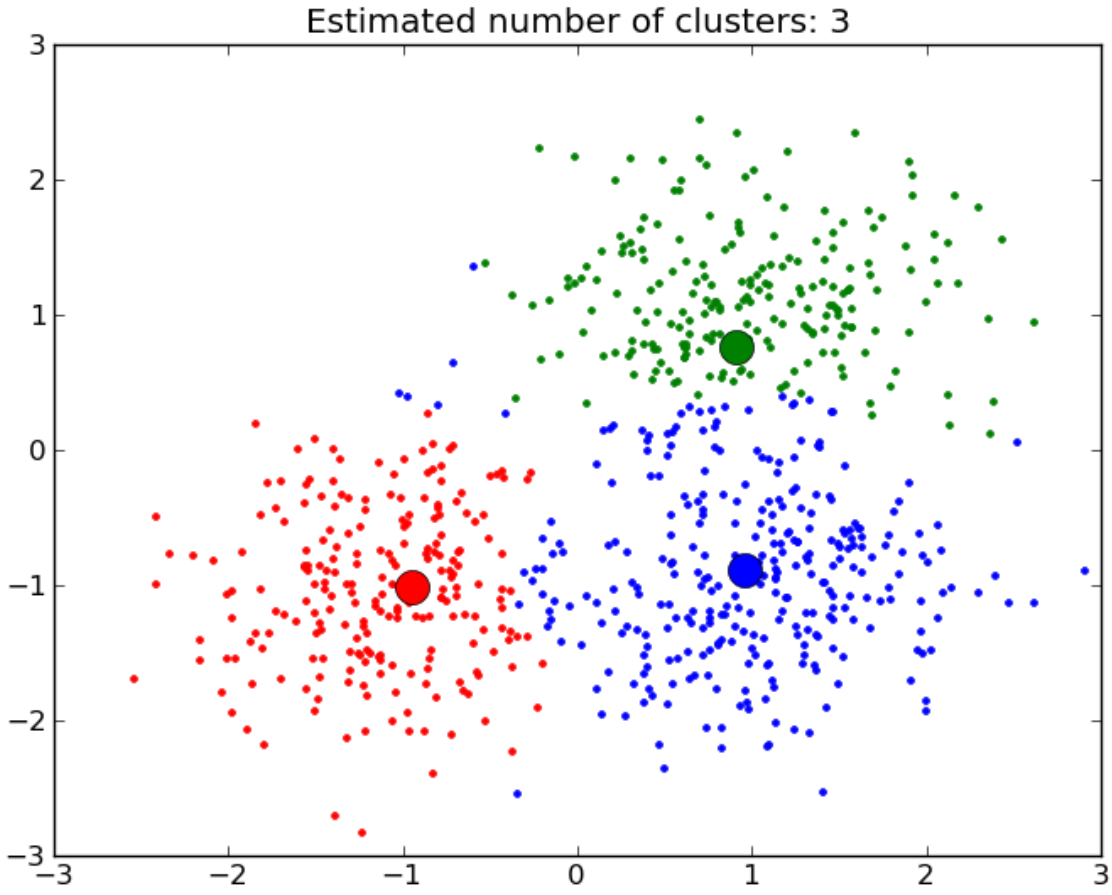
In modern ML, more often than not, the inputs are high dimensional real vectors:

$$\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d.$$

Each x_i is called a **feature** (**covariate** in Stats).
Example: $x_1 =$ age, $x_2 =$ weight, $x_3 =$ blood pressure,...
Example: $x_i =$ intensity of a pixel i in an image

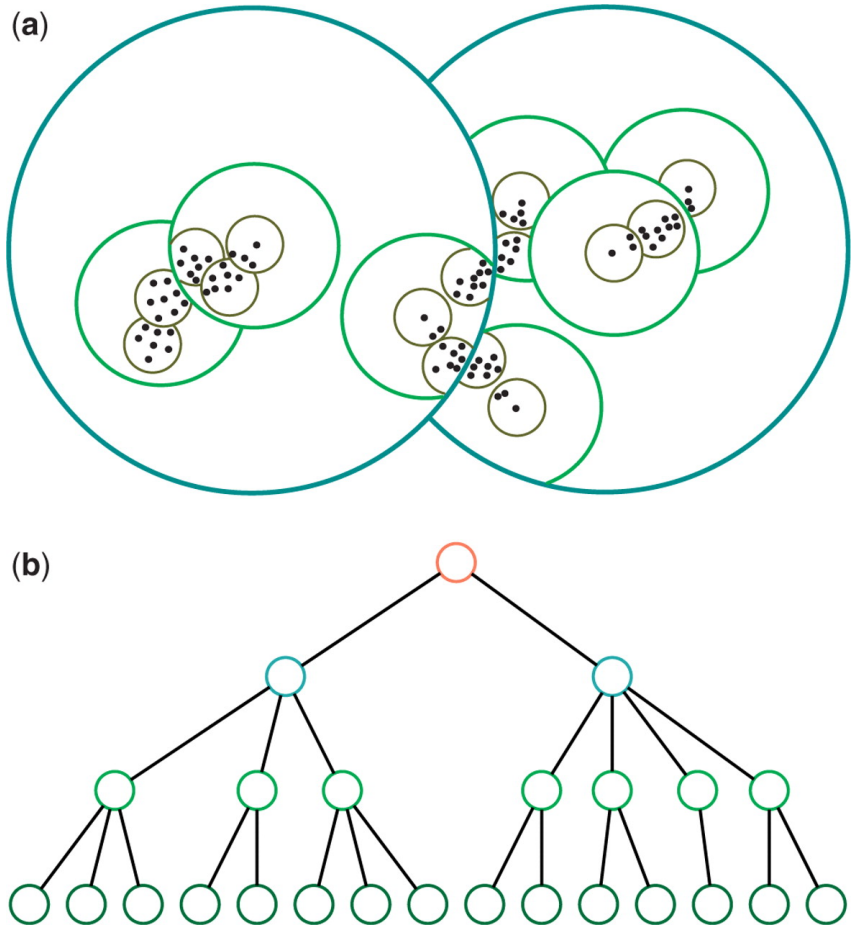
It often makes sense to ask whether a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ can be partitioned into a small number of **clusters** of similar datapoints.
$\rightarrow$ Clustering is a typical unsupervised learning problem.

Clustering



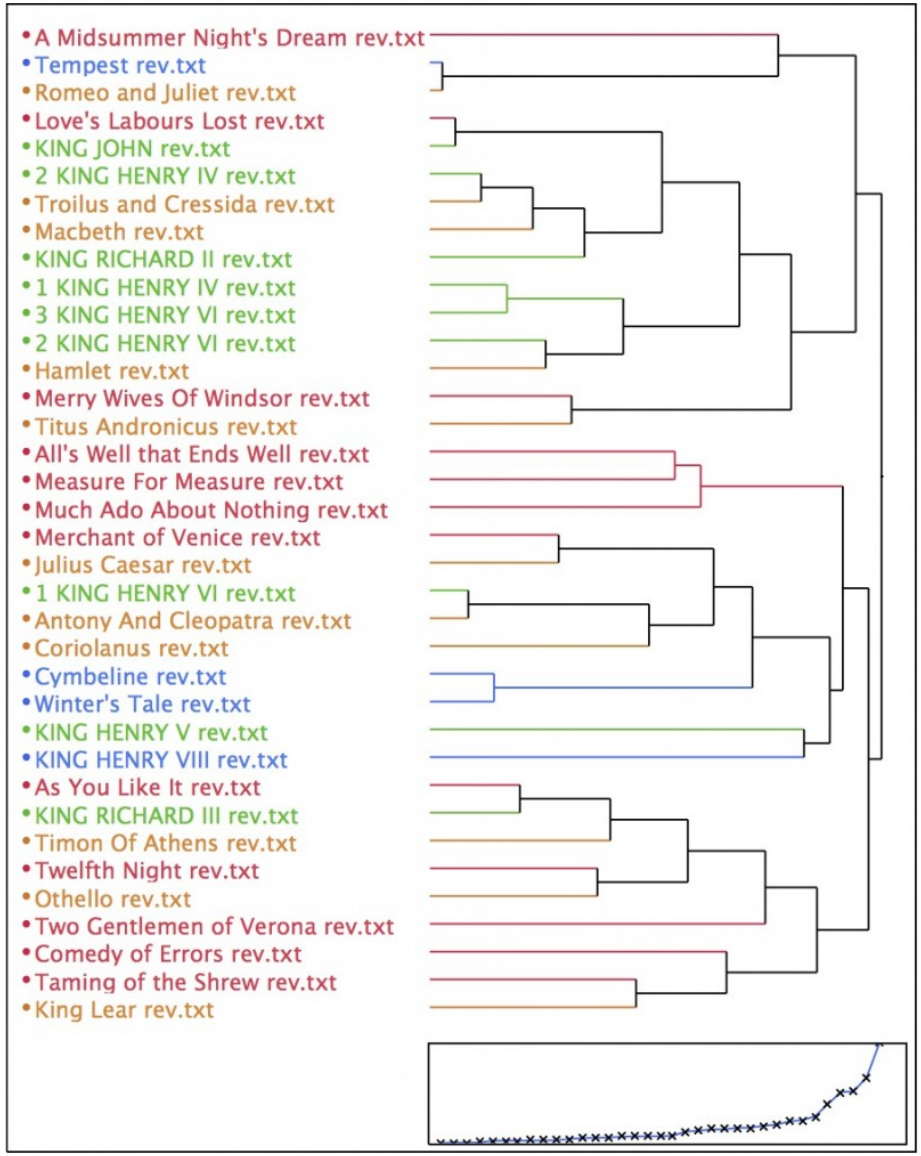
Cluster representatives indicated.

Hierarchical clustering

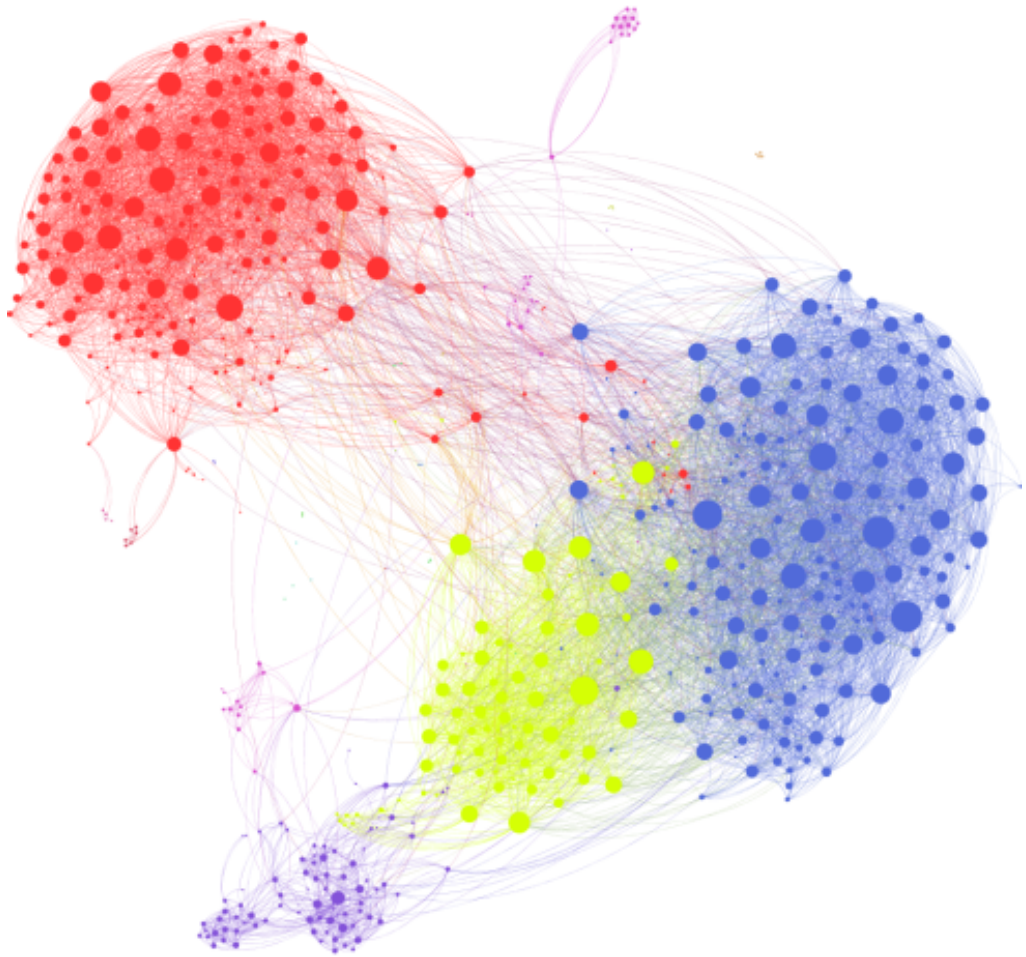


Cutting the tree at any level gives a flat clustering. Thanks to this freedom, don't have to decide the number of clusters in advance.

Hierarchical clustering



Clustering of nodes in a graph



Also known as **graph partitioning** (these are somebody's Facebook friends).

Clustering: the good

Clustering is important because

- It is a natural thing to want to do with large data.
- Can reveal a lot about the structure of data
  $\rightarrow$ exploratory data analysis.
  e.g., finding new types of stars, patients with similar disease profiles, ...
- Allows us to compress data by replacing points by their cluster representatives (called **vector quantization**).
- Key part of finding structure in large graphs & networks.

Clustering: the bad

- Unsupervised problem $\rightarrow$ always harder to formalize.
- Ill-defined: different objective functions possible, no clear winner. Even after we've clustered the data it's hard to say whether the clustering is good or bad $\rightarrow$ subjective.
- What is the "correct" number of clusters? Also subjective. Often data is very ambiguous in this regard.
- End users may attribute too much significance to the clusters with unforeseeable consequences.
- Compared to supervised ML, the theory is in its infancy.

Outline

- Flat clustering: k–means
- Hierarchical clustering: agglomerative clustering
- Model based clustering: mixture of Gaussians

Flat clustering

Flat clustering

Input: the datapoints $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$;
the desired number of clusters $k \in \mathbb{N}$.
Output: k disjoint sets $C_1, C_2, \ldots, C_k$ whose union is $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

Clustering is driven by a distance metric, d. In the simplest case it is just the Euclidean distance

$$d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| = \left(\sum_{i=1}^{d} (x_i - x_i')^2\right)^{1/2}.$$

Let's assign each cluster a representative point $\boldsymbol{m}_i$. Depending on context, we might or might not require $\boldsymbol{m}_i$ to be one of the $\mathbf{x}_1, \ldots, \mathbf{x}_n$ datapoints.

Cost functions

Start with a **cost function** (in this context also called **distortion**) that our algorithm tries minimize:

- Max distance to cluster center:
$$J_{\max} = \max_{i \in \{1,\ldots,k\}} \max_{\mathbf{x} \in C_i} d(\mathbf{x}, \boldsymbol{m}_i).$$
- Average distance to cluster center:
$$J_{\text{avg}} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \boldsymbol{m}_i).$$
- Average squared distance to cluster center:
$$J_{\text{avg}^2} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \boldsymbol{m}_i)^2.$$
- Sum of squared intra-cluster distances:
$$J_{\text{IC}} = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{x}' \in C_i} d(\mathbf{x}, \mathbf{x}')^2.$$

The k–means algorithm

Problem: find $C_1, C_2, \ldots, C_k$ and $\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_k \in \mathbb{R}^d$ that minimizes

$$J_{\text{avg}^2} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \boldsymbol{m}_i)^2.$$

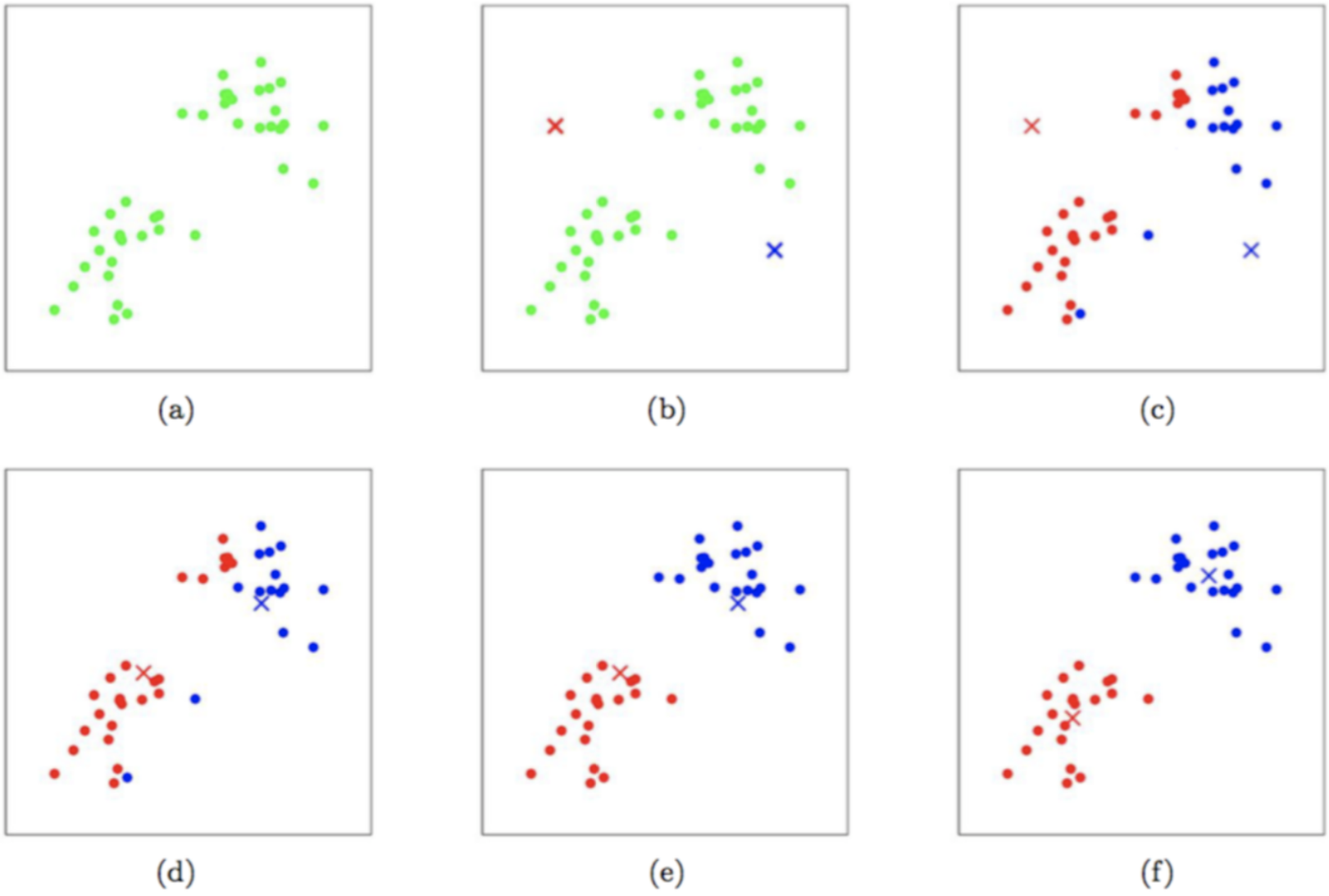
This is an **optimization problem**.

- Is it continuous? No. Is it combinatorial? No. $\rightarrow$ Mixed.
- Is it convex? No.
- How do we solve it? Alternating minimization strategy.

The k–means algorithm

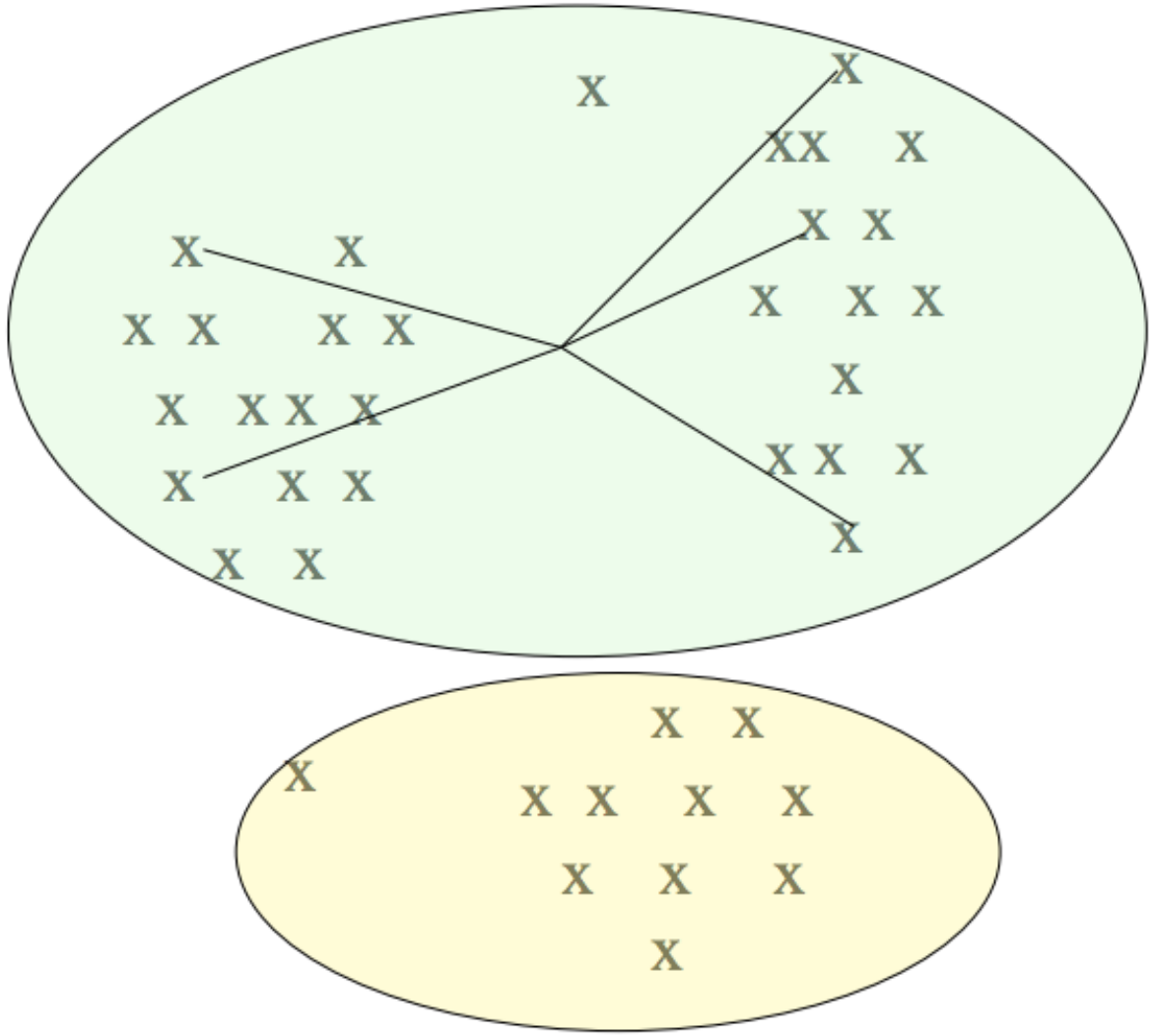
$\{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_k\} \leftarrow k$ random points in Ω ;
while(convergence){
  $C_1, C_2, \ldots, C_k \leftarrow \emptyset$;
  for $i = 1$ to n { // Assign each point to the closest center
    $\widehat{j} \leftarrow \arg\min_{j \in \{1,\ldots,k\}} d(\mathbf{x}_i, \boldsymbol{m}_j)$;
    $C_{\widehat{j}} \leftarrow C_{\widehat{j}} \cup \{\mathbf{x}_i\}$;
  }
  for $j = 1$ to k // Recompute cluster centers
    $\boldsymbol{m}_j \leftarrow \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} \mathbf{x}$;
}

The k–means algorithm



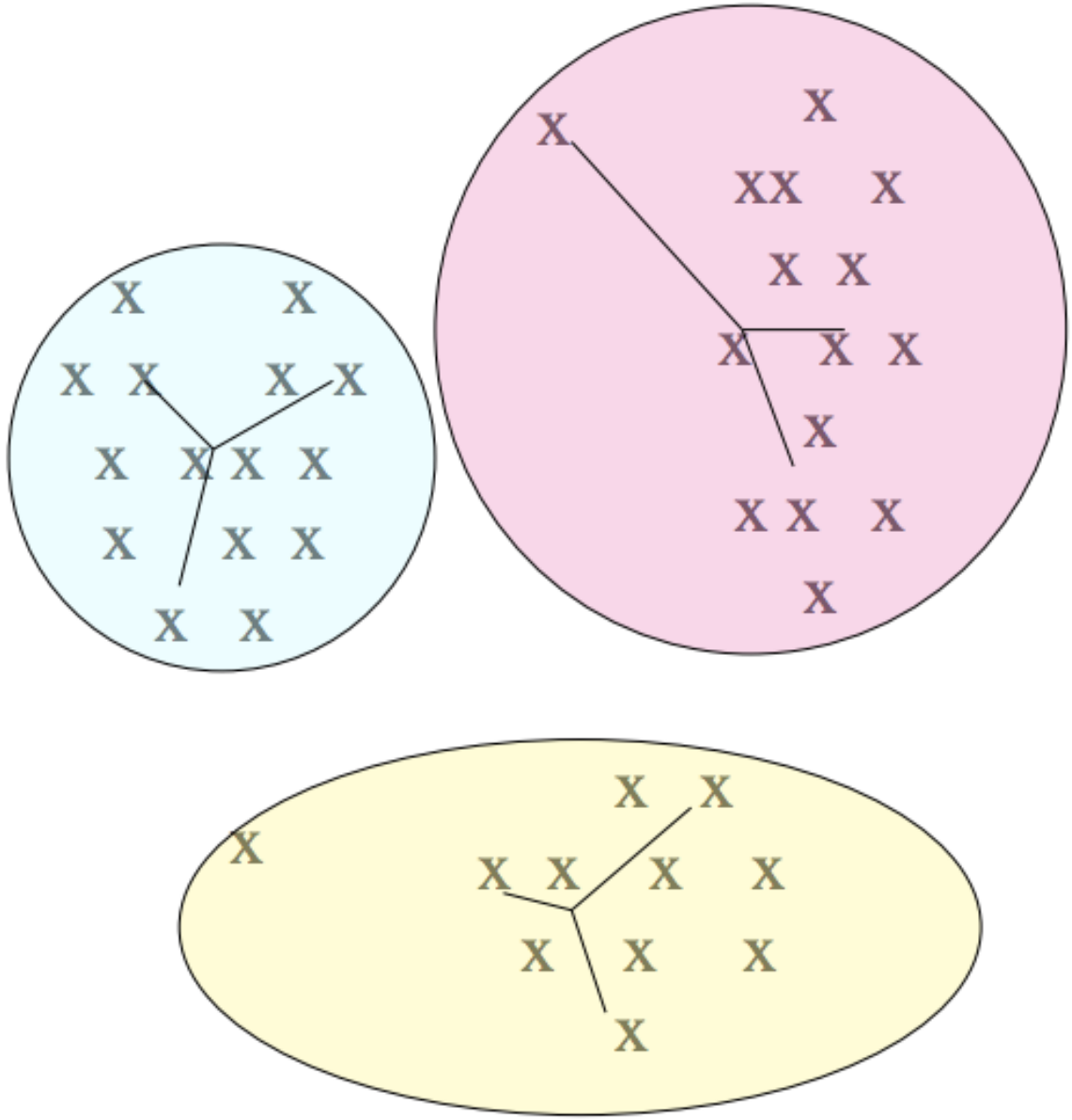
Pick *k*

Too few;
many long
distances
to centroid.



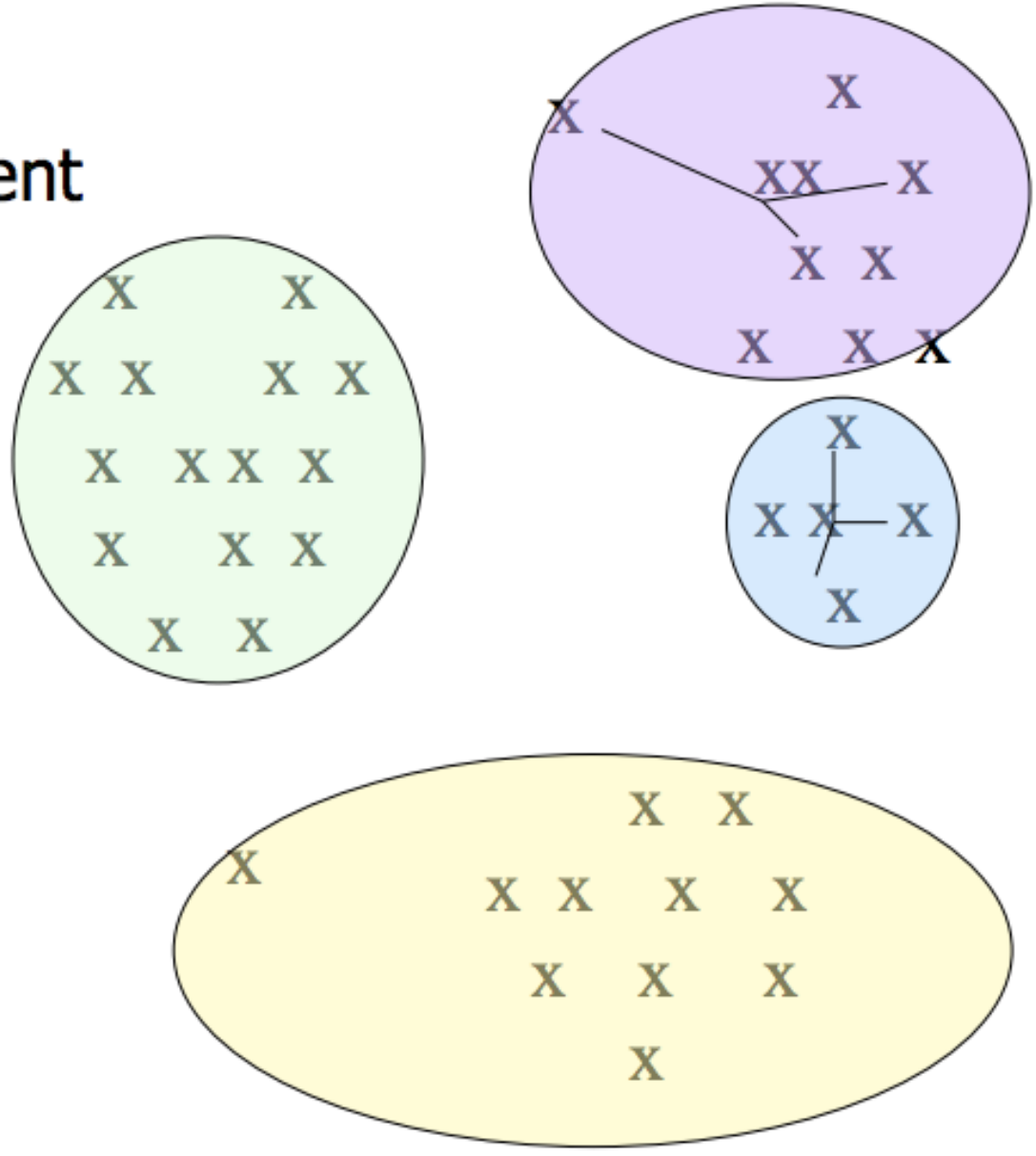
Pick *k*

Just right; distances rather short.



Pick *k*

Too many;
little improvement
in average
distance.



Hierarchical clustering

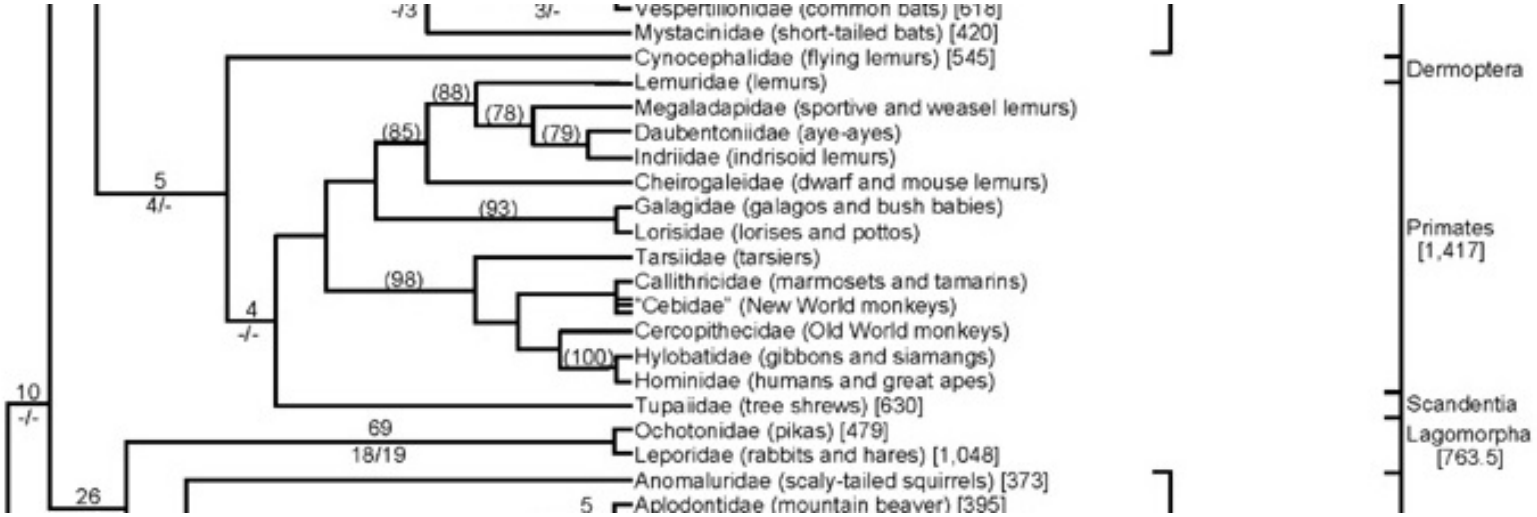
Hierarchical clustering

Input: the datapoints $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$
Output: a clustering tree (**dendrogram**)

Hierarchical clustering

Input: the datapoints $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$
Output: a clustering tree (**dendrogram**)

Advantages: Don't need to decide number of clusters in advance
Hierarchical structure is often very informative



Hierarchical clustering

- **Agglomerative:** start with n clusters containing one datapoint each, and then merge clusters pairwise until only one cluster is left.

Hierarchical clustering

- **Agglomerative:** start with n clusters containing one datapoint each, and then merge clusters pairwise until only one cluster is left.
- **Divisive:** Start with a single cluster containing all the datapoints and then split it into smaller and smaller clusters. $\rightarrow$ Recurisvely clustering clusters into smaller clusters.

Merging criteria for agglomerative

Agglomerative algorithms always merge the pair of clusters closest to each other according to some distance measure:

- Single linkage: $d(C_i, C_j) = \min_{\mathbf{x} \in C_i,\ \mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$
  $\rightarrow$ tends to generate long "chains"
- Complete linkage: $d(C_i, C_j) = \max_{\mathbf{x} \in C_i,\ \mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')$
  $\rightarrow$ tends to generate compact "round" clusters, k–center cost
- Average linkage:
  - $d(C_i, C_j) = \frac{1}{|C_i|} \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{x}' \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$
  - Ward's method $\rightarrow$ k–means cost of resulting clustering

Agglomerative clustering algorithm

$\mathcal{C} \leftarrow \emptyset$;
for $i = 1$ to n
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{\mathbf{x}_i\}\}$; // At first each point has its own cluster
while($|\mathcal{C}| > 1$){
  find the pair of clusters $C_1, C_2 \in \mathcal{C}$ for which $d(C_1, C_2)$ is smallest ;
  $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C_1, C_2\}) \cup \{C_1 \cup C_2\}$; // Merge C_1 and C_2
}

Model based clustering (flat case)

Model based clustering

- Regard each datapoint as consisting of two random quantities (**random variables**):
  - $\boldsymbol{X}_i \in \mathbb{R}^d$: the location of the i'th datapoint $\rightarrow$ **observed**
  - $Z_i \in \{1, \ldots, k\}$: the cluster assignment of the i'th datapoint $\rightarrow$ **hidden**
- Assume that each $(\mathbf{x}_i, z_i)$ pair is drawn independently from some probability distribution (model) with parameters θ:

$$(\mathbf{x}_i, z_i) \sim p_\theta.$$

Here θ can be any bunch of parameters, depends on the model.

Model based clustering

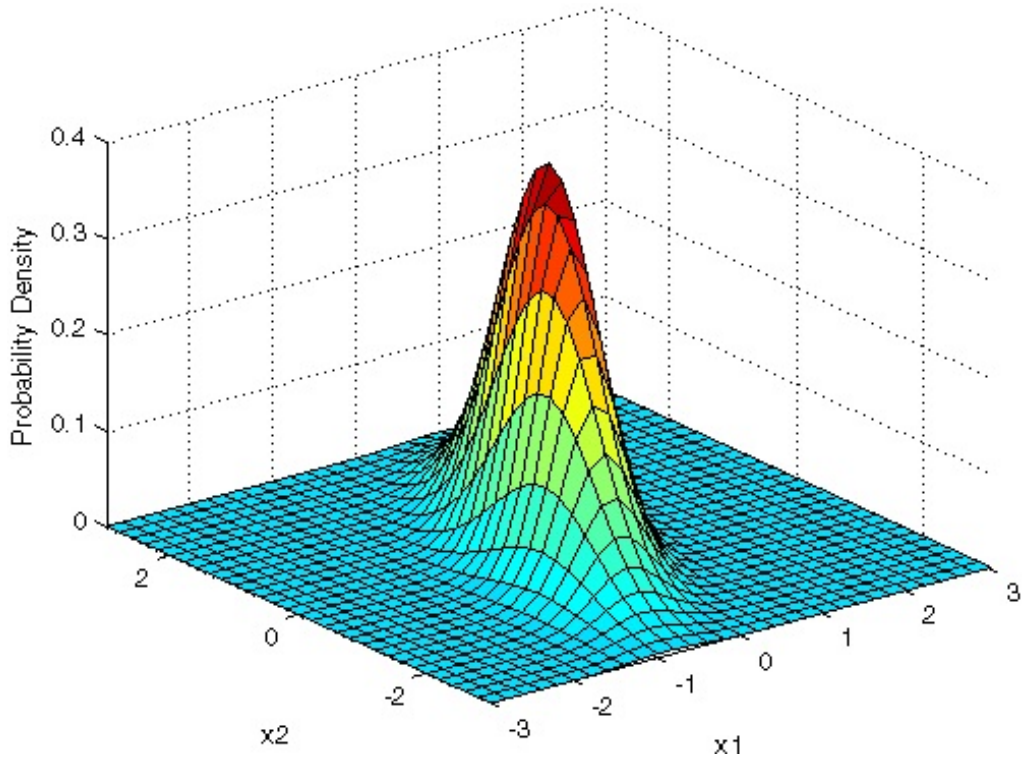
- Regard each datapoint as consisting of two random quantities (**random variables**):
  - $\boldsymbol{X}_i \in \mathbb{R}^d$: the location of the i'th datapoint $\rightarrow$ **observed**
  - $Z_i \in \{1, \ldots, k\}$: the cluster assignment of the i'th datapoint $\rightarrow$ **hidden**
- Assume that each $(\mathbf{x}_i, z_i)$ pair is drawn independently from some probability distribution (model) with parameters θ:

$$(\mathbf{x}_i, z_i) \sim p_\theta.$$

Here θ can be any bunch of parameters, depends on the model.

The probability distribution p_θ is said to **generate** the data.
$\rightarrow$ **generative modeling** (typical Bayesian idea)

The multivariate Gaussian (Normal)



$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{\Sigma}|}} \exp\left(-(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) := \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{\Sigma})$$

Mixture of Gaussians model

The most common generative model for clustering is a mixture of k Gaussians:

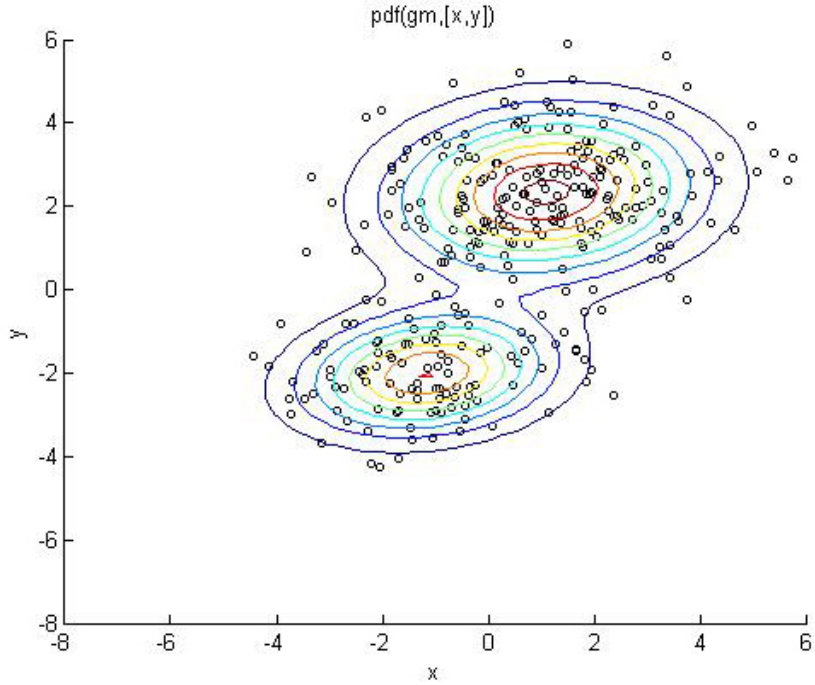
$$p_\theta(\mathbf{x}, z) = \pi_z \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$$

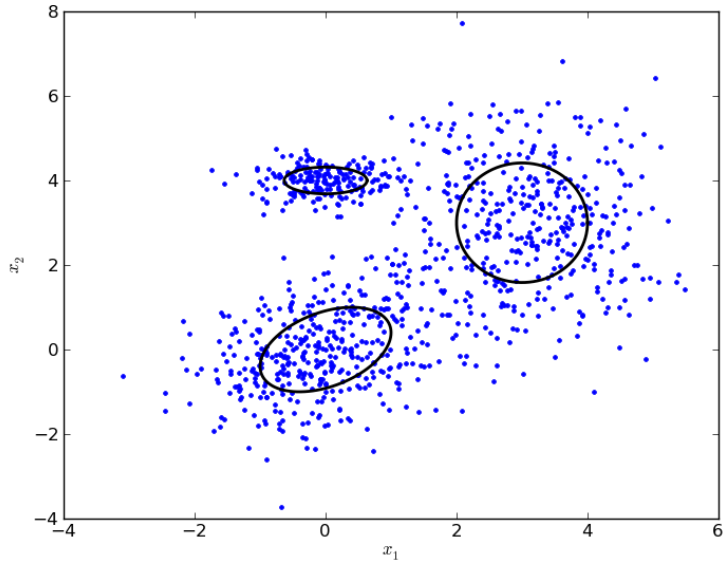
$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) = (2\pi)^{-d/2} \, |\boldsymbol{\Sigma}_z|^{-1/2} \exp(-(\mathbf{x} - \boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1} (\mathbf{x} - \boldsymbol{\mu}_z)/2).$$

The parameters $\theta = ((\pi_1, \mu_1, \Sigma_1), \ldots, (\pi_k, \mu_k, \Sigma_k))$ are:

- $\pi_z \in [0, 1]$: the prior probability of a new point coming from cluster z
- $\boldsymbol{\mu}_z \in \mathbb{R}^d$: the center of the z'th Gaussian
- $\boldsymbol{\Sigma}_z \in \mathbb{R}^{d \times d}$: the covariance matrix of the z'th Gaussian

Mixture of Gaussians





Big advantage: can capture clusters of different sizes and orientations. But how do we find the parameters? $\rightarrow$ statistical estimation.

The EM algorithm for clustering

Starting from random settings, iterate the following two steps:

- **"E-step":** Given the $\boldsymbol{\mu}_j$'s and $\boldsymbol{\Sigma}_j$'s update the assignments

$$p_{i,j} = p(\mathbf{x}_i \text{ belongs to cluster } j) \leftarrow \frac{\pi_j \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{j'} \pi_{j'} \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_{j'}, \boldsymbol{\Sigma}_{j'})}$$

- **"M-step":** Given the assignments, update π and the $\boldsymbol{\mu}_i$'s and Σ_i's

$$\pi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_{i,j} \qquad \boldsymbol{\mu}_i \leftarrow \frac{\sum_{i=1}^{n} p_{i,j} \mathbf{x}_j}{\sum_{i=1}^{n} p_{i,j}}$$

$$\boldsymbol{\Sigma}_i \leftarrow \frac{\sum_{i=1}^{n} p_{i,j} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top}{\sum_{i=1}^{n} p_{i,j}}$$

Summary

- Flat clustering: k–means
- Hierarchical clustering: agglomerative clustering
- Model based clustering: mixture of Gaussians

Thank you!