Chapter 6

Clustering

Contents

6.1 Clustering

Clustering is an unsupervised learning technique which automatically partitions unlabeled data into groups of similar datapoints. It is useful for:

Segmentation Segmenting a large set of cases into small subsets that can be treated similarly.

- e.g., image segmentation.

Compression Generate a more compact description of a dataset.

- e.g., handwritten digit recognition.

Representation Model an underlying process that generates the data as a mixture of different, localized processes.

6.2 Clustering Applications

- Cluster news articles or web pages or search results by topic.
- Cluster protein sequences by function of genes according to expression profile.
- Cluster users of social networks by interest.
- Cluster galaxies or nearby stars.

6.3 Clustering Algorithms

6.3.1 Flat clustering

No inter-cluster structure.

- k-means algorithm.
- Gaussian mixture models (GMM).
- Spectral clustering.

6.3.2 Hierarchical clustering

Clusters for a hierarchy.

- Bottom-up (agglomerative clustering).
- Top-down (divisive clustering).

6.3.3 Hard clustering

Items are assigned to a unique cluster.

- k-means algorithm.
- Spectral clustering.

6.3.4 Soft (fuzzy) clustering

Cluster membership is a real-valued function, distributed across several clusters.

- Soft k-means.
- Gaussian mixture models.

6.3.5 Centroid-based clustering

This type of clustering algorithm forms around the centroids of the data points. E.g., k-means, k-modes.

6.3.6 Distribution-based clustering

Clustering algorithm is modeled using statistical distributions. It assumes that the data points in a cluster are generated from a particular probability distribution, and the algorithm aims to estimate the parameters of the distribution. E.g., GMM.

6.3.7 Density-based clustering

This type of clustering algorithm groups together data points that are in high-density concentration and separates points in low-concentration regions. E.g., DBSCAN.

6.4 k-Means Clustering

- k-means algorithm is an iterative clustering algorithm, based on the Euclidean distance. It is a non-parametric learning algorithm.
- A greedy algorithm (Lloyd's algorithm) locally optimizes the cluster quality measure:
  - The cluster quality measure is computed based on the cluster centroid.
  - Find the closest cluster center for each item and assign it to that cluster.
  - Recompute the cluster centroid as the mean of items, for the newly-assigned items in the cluster.

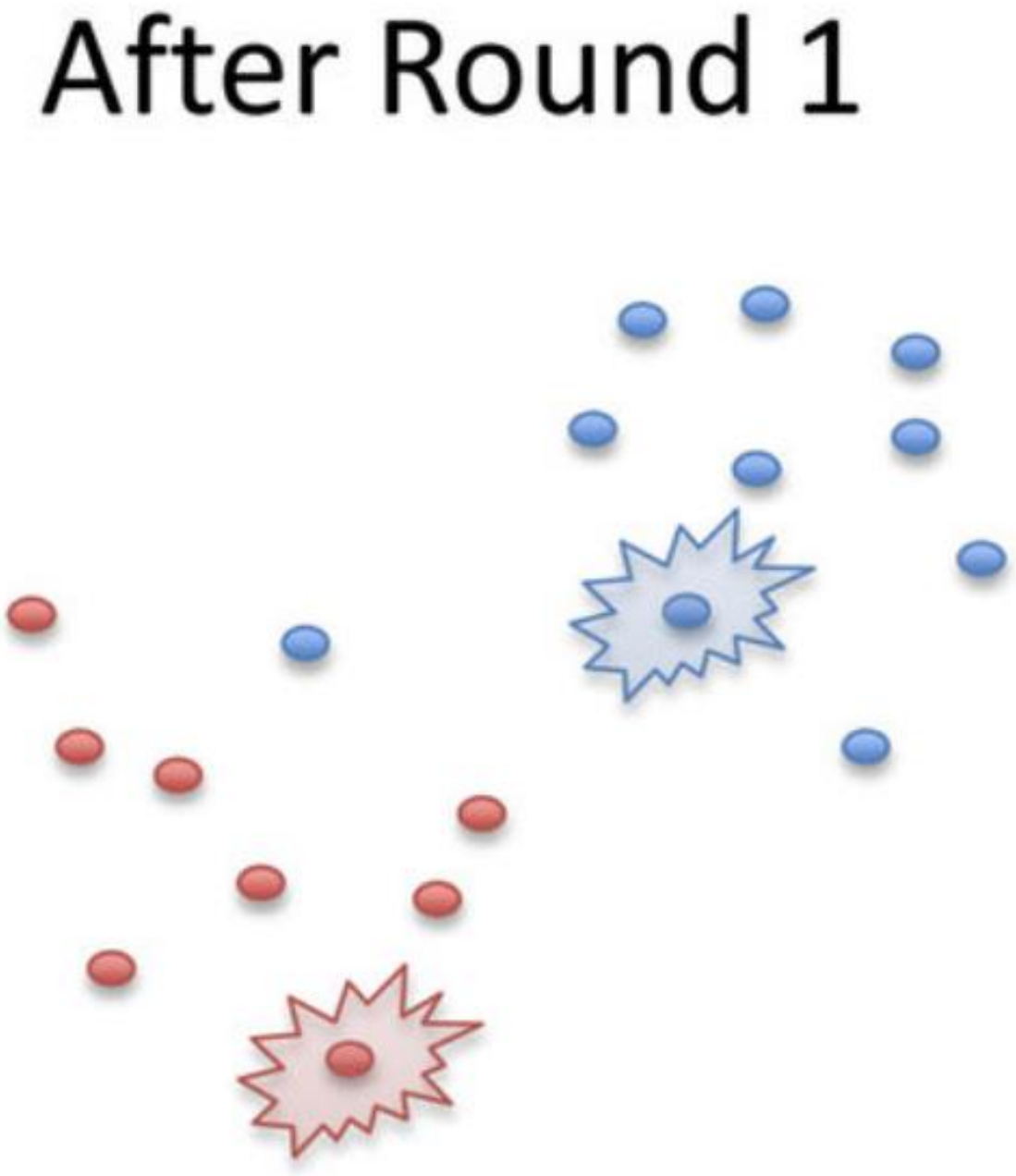


Figure 6.1: k-Means Clustering

- Initialize: Pick k data points as cluster centers.
- Repeat:
  - Assign data points to closest cluster center.
  - Change the cluster center to the average of its assigned points.
- Stop when the assignments of data points do not change.

- Input: A set of n datapoints $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ with k clusters.
- Output: k representatives $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$.
- Objective: choose $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ such that:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - c_j||^2 \tag{6.1}$$

- Initialize cluster centers $c_1, c_2, \ldots, c_k$ and clusters $C_1, C_2, \ldots, C_k$.
- Repeat until there is no further change:
  - For each $j : C_j \leftarrow \{x$ whose closest center is $c_j\}$.
  - For each $j : c_j \leftarrow$ mean of C_j.

$$\begin{aligned}\sum_{i=1}^{4} \sum_{j=1}^{2} ||x_i - c_j||^2 &= ||x_1 - c_1||^2 + ||x_1 - c_2||^2 + ||x_2 - c_1||^2 + ||x_2 - c_2||^2 \\ &+ ||x_3 - c_1||^2 + ||x_3 - c_2||^2 + ||x_4 - c_1||^2 + ||x_4 - c_2||^2 \\ &= 0^2 + 3^2 + 1^2 + 2^2 + 2^2 + 1^2 + 0^2 + 3^2 \\ &= 0 + 9 + 1 + 4 + 4 + 1 + 0 + 9 \\ &= 28\end{aligned}$$

With updated cluster centers c_1 and c_2, recompute the objective function:

$$\begin{aligned}0.5^2 + 2.5^2 + 0.5^2 + 1.5^2 + 1.5^2 + 0.5^2 + 2.5^2 + 0.5^2 &= 0.25 + 6.25 + 0.25 + 2.25 + 2.25 + 0.25 + 6.25 + 0.25 \\ &= 18\end{aligned}$$

6.4.1 k-Means Properties

- It is guaranteed to converge is a finite number of iterations, but it may converge at a local optimum that is different from the global optimum.
- Initialization is crucial as it decides how fast it converges as well as the quality of the solution output.
- Time complexity: $O(kdni)$

  n the number of d-dimensional data (to be clustered).

  k the number of clusters.

  i the number of iterations needed until convergence.

6.5 Furthest Point Initialization

- Choose c_1 arbitrarily at random.
- For $j = 2, \ldots, k$
  - Pick c_j among the datapoints $x_1, x_2, \ldots, x_n$ that is farthest from the previously chosen cluster centers $c_1, c_2, \ldots, c_{j-1}$.
- This method solves the issues with random initialization pertaining to well separated clusters.
- However, this method of initialization is sensitive to outliers.

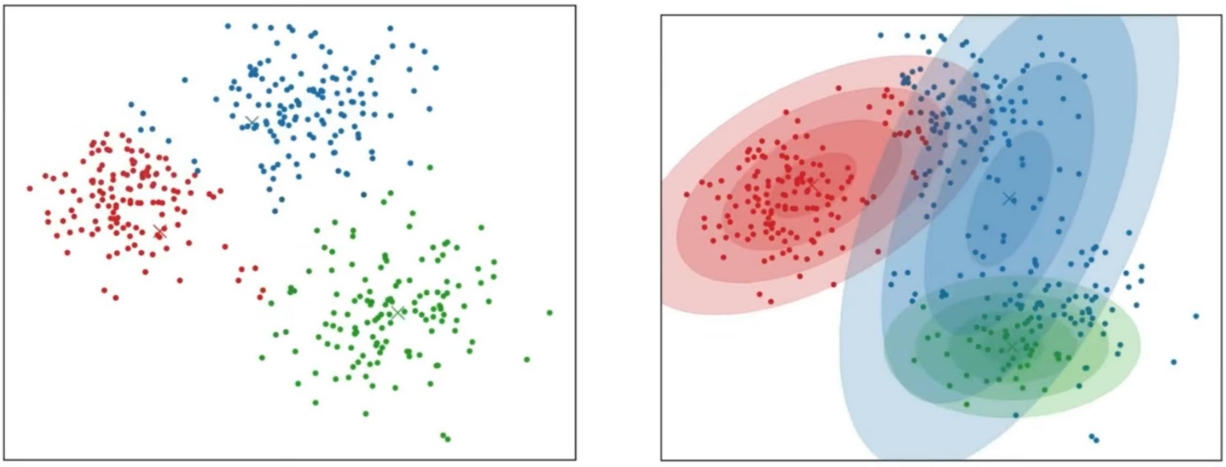
6.6 Pros and Cons

Pros:

- Easy to implement.
- Guarantees convergence.
- Generalized to clusters of different shapes and sizes.

Cons:

- Choosing a k manually.
- Final solution is dependent on initial values.
- Curse of dimensionality.



(a) k-means (b) GMM

Figure 6.2: k-Means vs. GMM

6.7 Gaussian Mixture Models (GMMs)

- Clusters are models as gaussian i.e., the data within a cluster follows the normal or gaussian distribution:

$$N(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma}\right)} \tag{6.2}$$

- The GMM clustering approach uses:
  - Parametric learning
  - Probabilistic learning
  - Generative learning
- Expectation-Maximation (EM) algorithm assigns data points to a cluster with some probability.

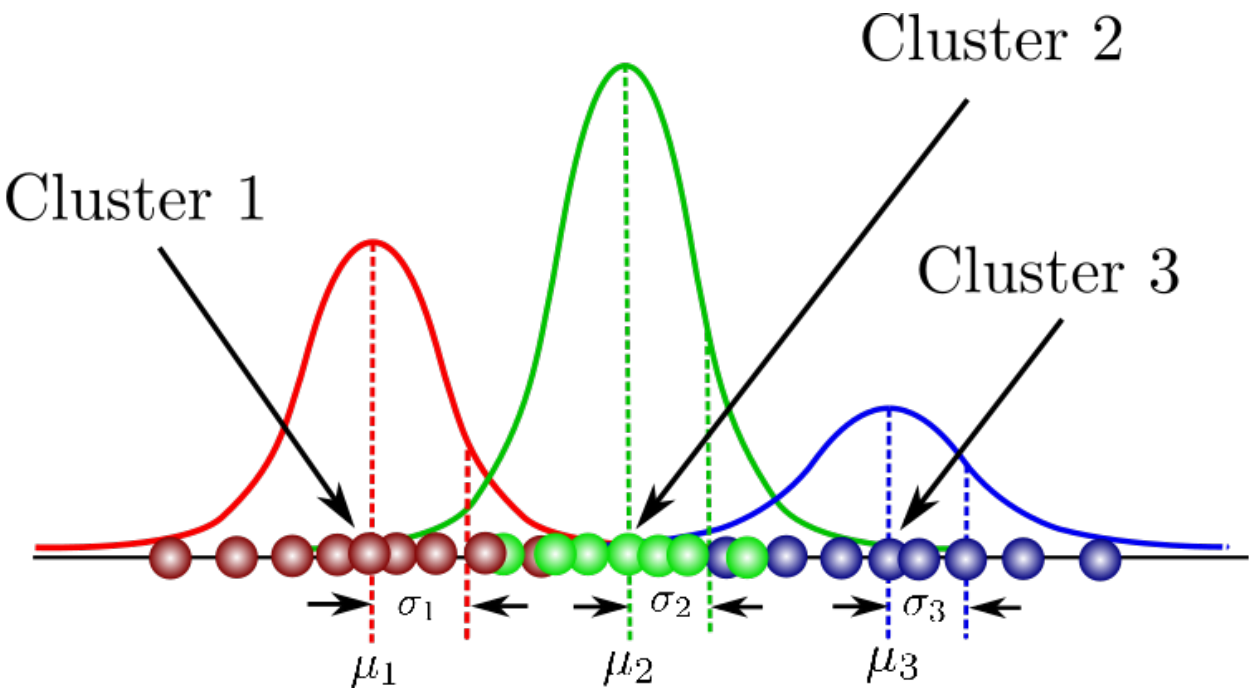


Figure 6.3: Gaussian Mixture Models

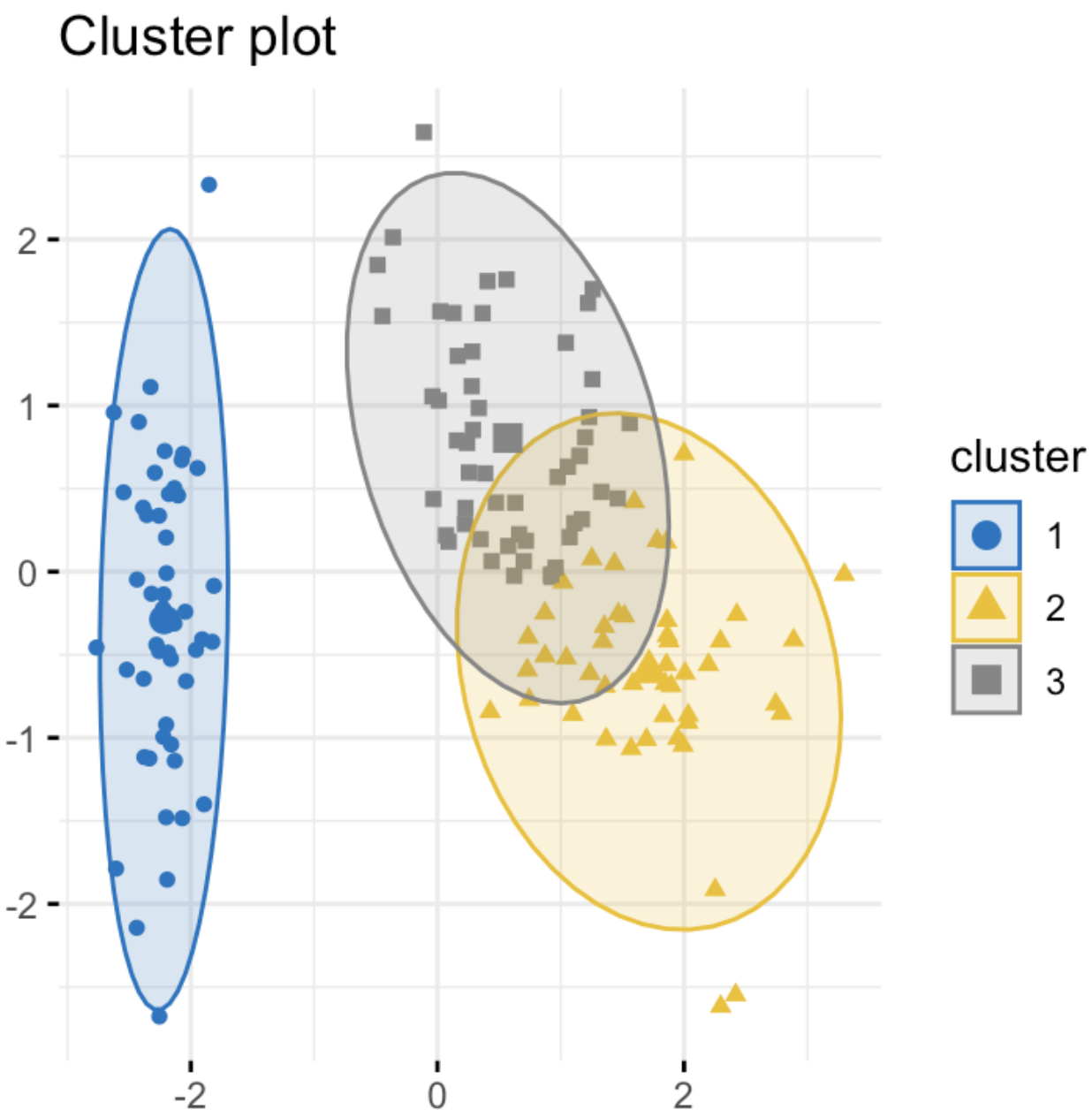


Figure 6.4: GMM Cluster Plot

- Likelihood is the probability of observing the data given the parameters of the model. In the EM algorithm, the goal is to find the parameters that maximize the likelihood.
- Latent Variables are unobserved variables in statistical models that can only by inferred indirectly through their effects on observable variables.
- Parameters (latent variables) describing a cluster c:

  μ_c Mean

  σ_c Covariance

π_c Weight

6.8 GMM Probability

$$P(x) = \sum_{c=1}^{k} \pi_c N(x|\mu_c, \sigma_c) \tag{6.3}$$

where $\sum_{c=1}^{k} \pi_c = 1$

6.9 EM Algorithm

- EM algorithm is an iterative optimization technique used for estimating the parameters (latent variables) of the gaussians in the GMM model.

Expectation step Calculate the expected value of the log-likelihood function given the current parameter estimates.

Maximization step Update the parameter estimates to maximize the expected log-likelihood calculated in the expectation step.

6.9.1 Expectation Step

- For each data point x_i, compute the probability it belongs in cluster c.

$$\gamma_{ic} = \frac{\pi_c N(x_i|\mu_c, \sigma_c)}{\sum_{c=1}^{k} \pi_c N(x_i|\mu_c, \sigma_c)} \tag{6.4}$$

- The denominator is the weighted sum of the probability that the data point x_i belongs to every gaussian.
- If x_i belongs to the c^{th} gaussian, corresponding weight π_c will be higher.

6.9.2 Maximization Step

For each cluster c, update the three parameters.

$$\pi_c = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ic}$$

$$\mu_c = \frac{\sum_{i=1}^{n} \gamma_{ic} x_i}{\sum_{i=1}^{n} \gamma_{ic}}$$

$$\sigma_c = \frac{\sum_{i=1}^{n} \gamma_{ic} (x_i - \mu_c)^2}{\sum_{i=1}^{n}}$$

6.10 Example Using 1-D data

- Initialize 2 gaussians with random latent variables.
- E-step: Points 1 and 2 will have higher γ for c_2, points 4 and 5 will have higher γ for c_1.
- M-step: Update π, μ, σ to generate new gaussians.
- Repeat EM until convergence.

6.11 Convergence

- Evaluate the log-likelihood and check for convergence of either the parameters or the log-likelihood.

$$\begin{aligned} \log(L) &= \log \prod_{i=1}^{n} P(x_i) \\ &= \sum_{i=1}^{n} \log \left(\sum_{c=1}^{k} \pi_c N(x_i | \mu_c, \sigma_c) \right) \end{aligned} \tag{6.5}$$

- Iterate over the EM algorithm until convergence is achieved.
- Each iteration increases the log-likelihood of the model.

6.12 Pros and Cons

Pros:

- Flexible i.e., can model a wide range of probability distributions.
- Robust to the outliers and can handle missing data.
- Converges quickly i.e., fast to fit a dataset.
- Easy to interpret since we obtain the latent variables.

Cons:

- Choosing a k manually.
- Sensitive to the initial values of the model parameters.
- Computationally expensive when working with high-dimensional data.

6.13 Hierarchical Clustering

Divisive clustering (top-down) –

- Partition data into 2-groups.
- Recursively cluster each group.

Agglomerative clustering (bottom-up) –

- Start with every point in its own cluster.
- Repeatedly merge the two closest clusters.

Hierarchical clustering produces not just one clustering, but a family of clustering represented by a dendogram.

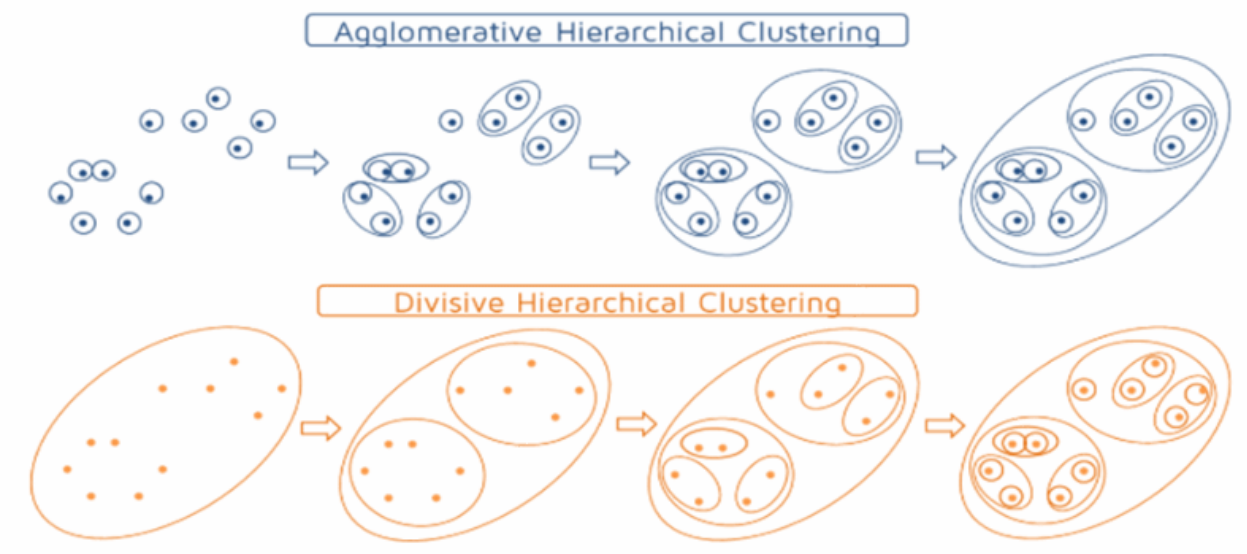


Figure 6.5: Hierarchical Clustering

6.14 Linkage

Single Linkage uses the smallest distance between all pairs of data points in two clusters:

$$d(C, C') = \min_{x \in C, x' \in C'} d(x, x')$$