03.09.2024

Statistical Methods in AI (CS7.403)

Lecture-9: Clustering (Gaussian Mixture Models, Hierarchical Clustering)

Ravi Kiran (ravi.kiran@iiit.ac.in)

https://ravika.github.io

@vikataravi

Center for Visual Information Technology (CVIT)

IIIT Hyderabad

swyx: wukong
@swyx

I saw an AI engineer today

No cursor.
No claude 3.5 sonnet.
No aider.

He just sat there.
Coding without running an LLM every other minute.
Like a psychopath.

1:31 AM · Sep 2, 2024 · **608K** Views

Recap: k-means

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set

   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

Mixture Models

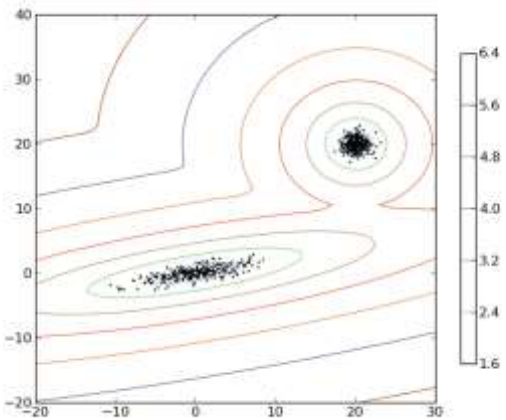
Most common mixture model: Gaussian mixture model (GMM)

- A GMM represents a **distribution** as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

with π_k the mixing coefficients, where:

$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$



http://scikit-learn.sourceforge.net/0.5/auto_examples/gmm/plot_gmm_pdf.html

- Can think of the data $\{x_1, x_n, \ldots, x_N\}$ using a "generative story"
  - For each example x_n, first choose its cluster assignment $z_n \in \{1, 2, \ldots, K\}$ as

$$z_n \sim \text{Multinoulli}(\pi_1, \pi_2, \ldots, \pi_K) \quad \text{aka "categorical"}$$

  - Now generate x from the Gaussian with id z_n

$$x_n | z_n \sim \mathcal{N}(\mu_{z_n}, \Sigma_{z_n})$$

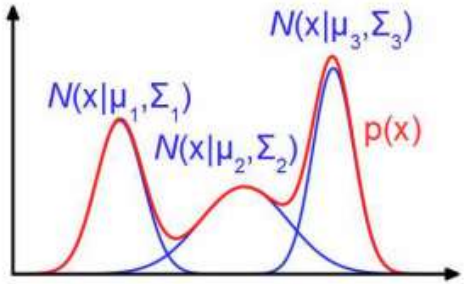
https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec7.pdf

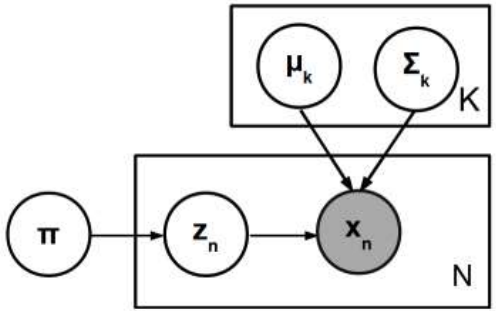
- Can think of the data $\{x_1, x_n, \ldots, x_N\}$ using a "generative story"
  - For each example x_n, first choose its cluster assignment $z_n \in \{1, 2, \ldots, K\}$ as

$$z_n \sim \text{Multinoulli}(\pi_1, \pi_2, \ldots, \pi_K)$$

  - Now generate x from the Gaussian with id z_n

$$x_n | z_n \sim \mathcal{N}(\mu_{z_n}, \Sigma_{z_n})$$





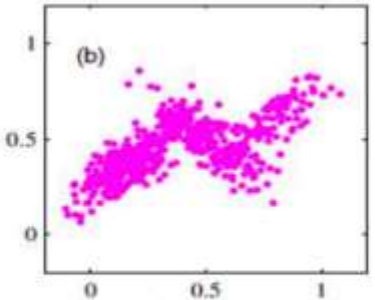
Shaded nodes: Observed

White nodes: Unknowns

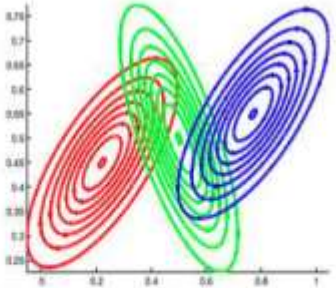
- Note: $p(z_{nk} = 1) = \pi_k$ is the prior probability of x_n going to cluster k and

$$p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$$

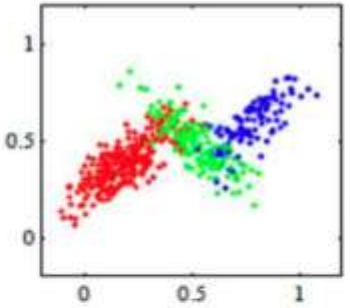
https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec7.pdf



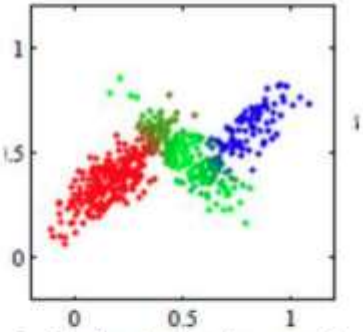
Samples from p(x)



Component Distributions



Samples labeled using their true component



Soft clustering learned by a Gaussian mixture model

- Joint distribution of data and cluster assignments

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$$

- Marginal distribution of data

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

Notice the "mixed" colored points in the overlapping regions

https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec7.pdf

GMM – two scenarios

- Joint distribution of data and cluster assignments

$$p(x, z) = p(z)p(x|z)$$

- Marginal distribution of data

$$p(x) = \sum_{k=1}^{K} p(z_k = 1)p(x|z_k = 1) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

X and Z are both observed and Θ is known
(Idealized scenario)

Only X is observed
(Realistic scenario)

Parameter Estimation in GMMs

Most common mixture model: Gaussian mixture model (GMM)

- A GMM represents a **distribution** as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

with π_k the mixing coefficients, where:

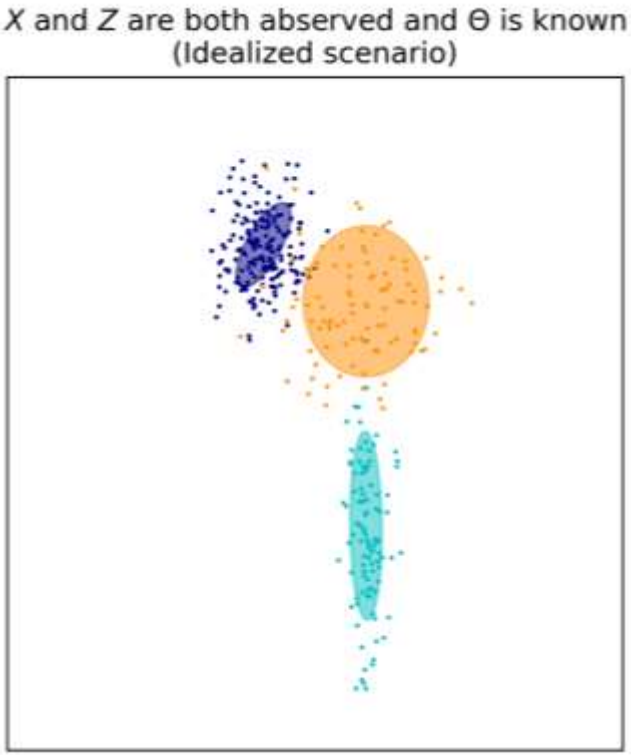
$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$

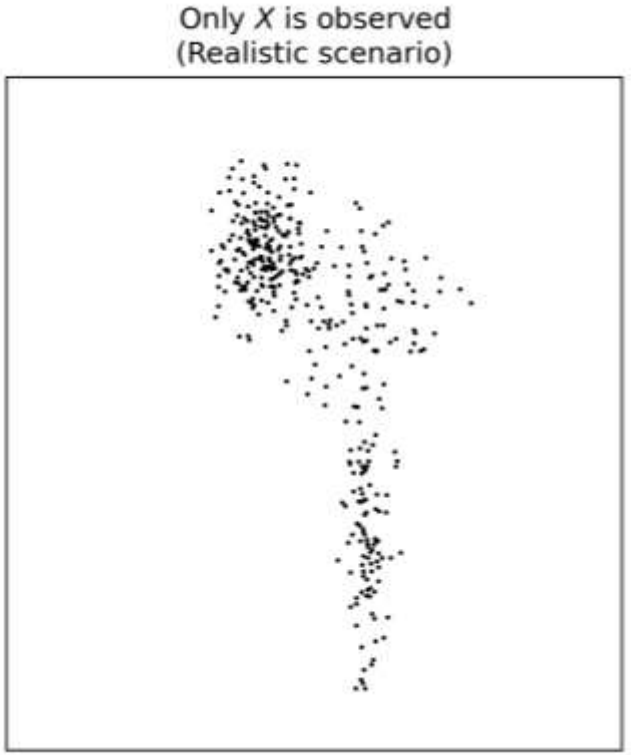
- Maximum likelihood maximizes

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k) \right)$$

w.r.t $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

GMM – two scenarios



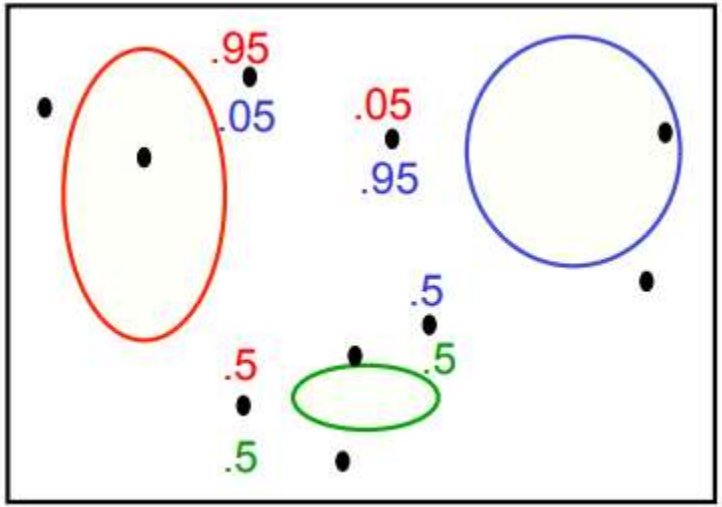


Parameter Estimation in GMMs using EM

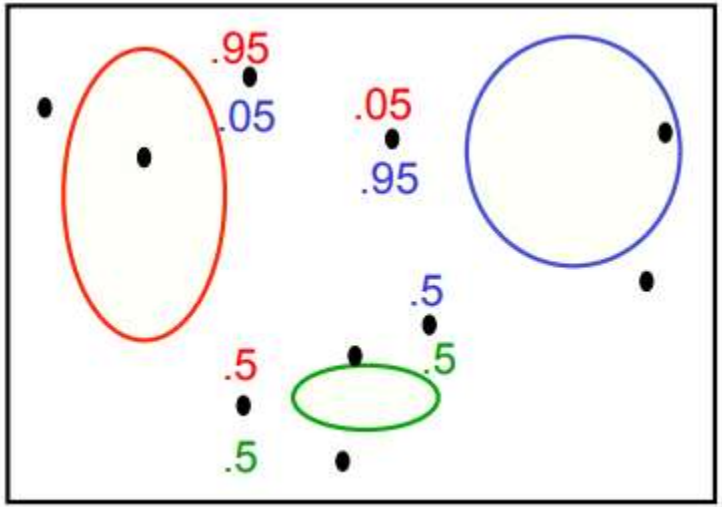
Parameter Estimation in GMMs using EM

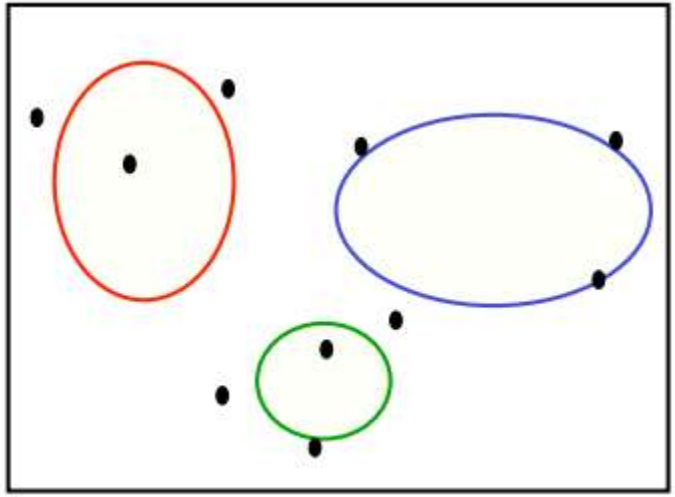
Strategy: Adjust the model parameters to maximize the observed ‘data’ probability

- Optimization uses the Expectation Maximization algorithm, which alternates between two steps:
  1. E-step: Compute the posterior probability over z given our current model - i.e. how much do we think each Gaussian generates each datapoint.

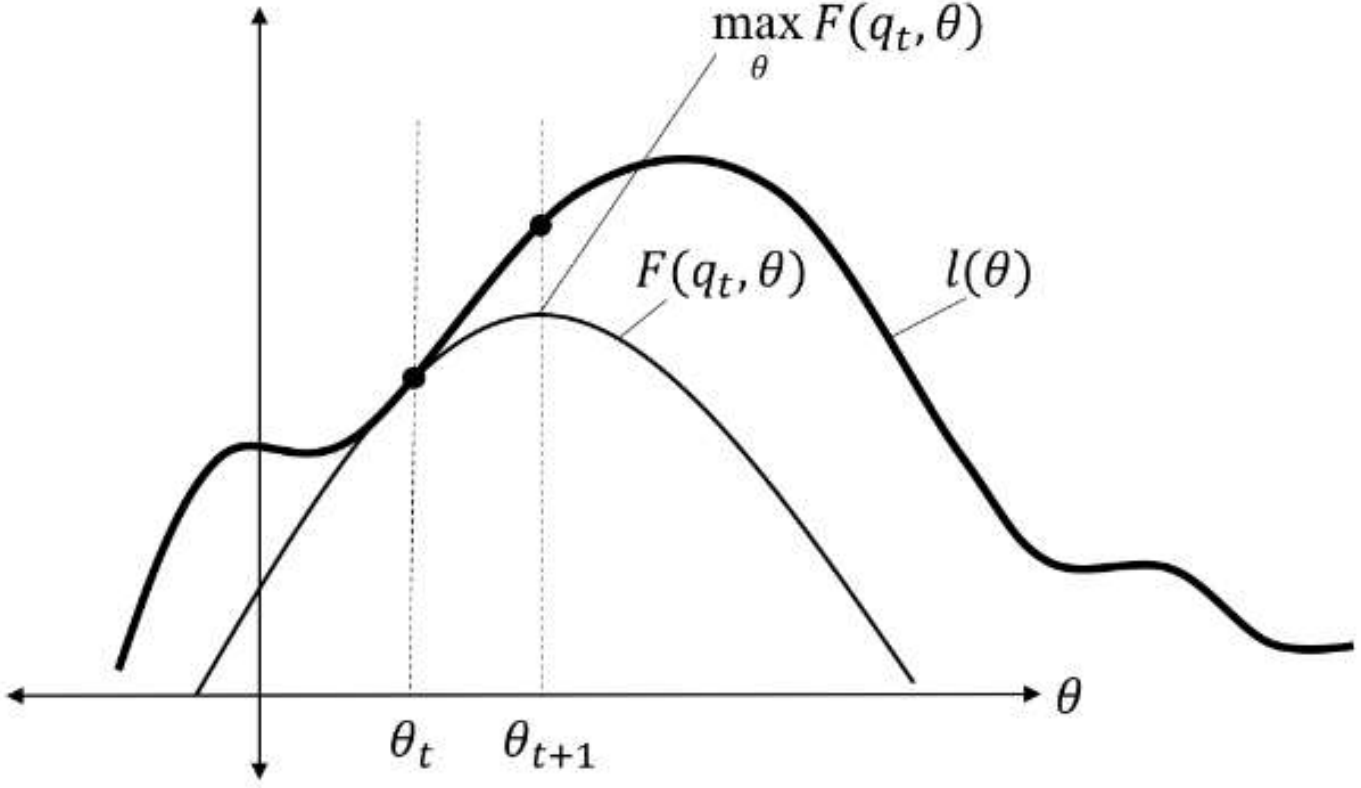


- Optimization uses the Expectation Maximization algorithm, which alternates between two steps:
  1. E-step: Compute the posterior probability over z given our current model - i.e. how much do we think each Gaussian generates each datapoint.
  2. M-step: Assuming that the data really was generated this way, change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for.



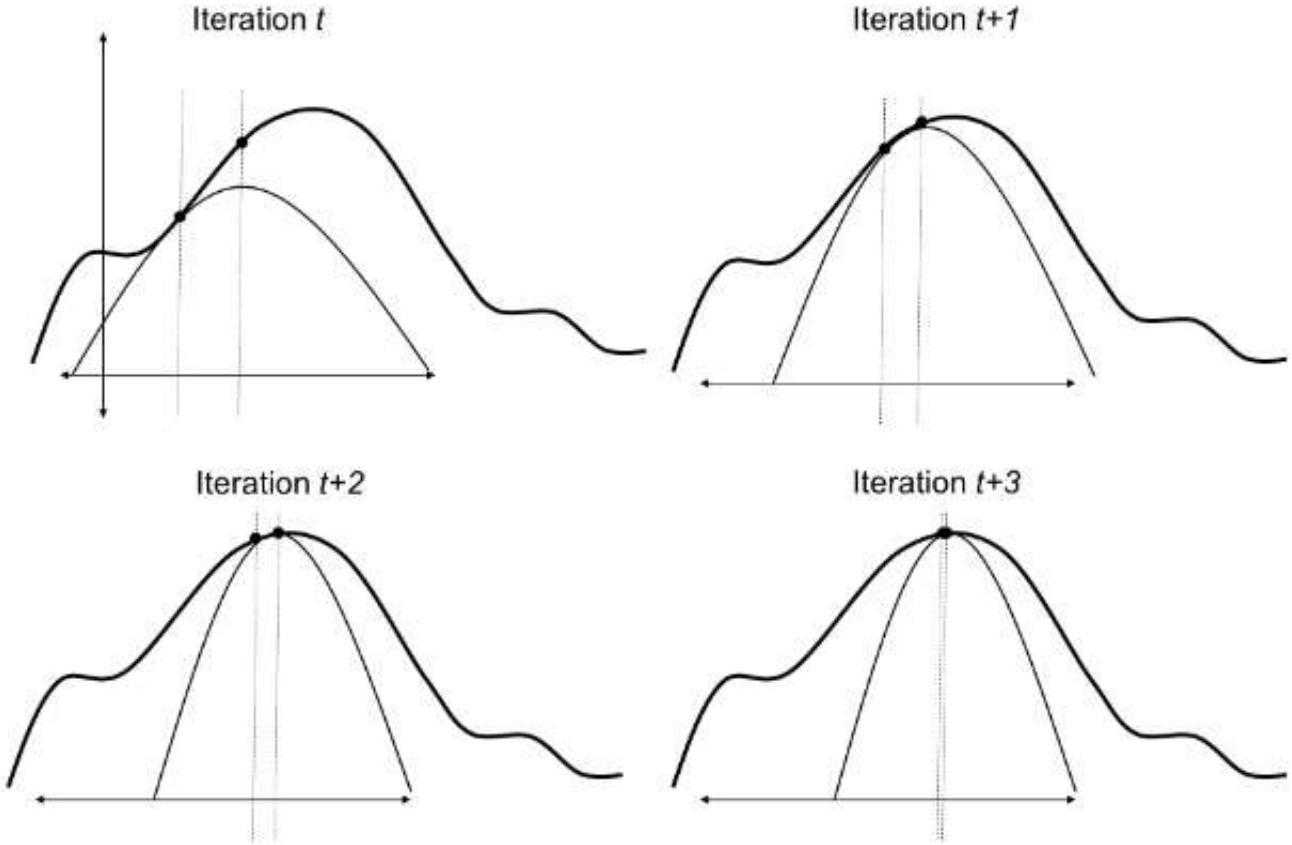


EM visualized



https://mbernste.github.io/posts/gmm_em/

EM visualized



https://mbernste.github.io/posts/gmm_em/

GMM - Algorithm

- Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k

GMM - Algorithm

- Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k
- Iterate until convergence:
  - E-step: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = p(z^{(n)}|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \Sigma_j)}$$

GMM - Algorithm

- Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k
- Iterate until convergence:
  - E-step: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = p(z^{(n)}|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \Sigma_j)}$$

  - M-step: Re-estimate the parameters given current responsibilities

$$\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)} \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\mathbf{x}^{(n)} - \mu_k)(\mathbf{x}^{(n)} - \mu_k)^T \\
\pi_k &= \frac{N_k}{N} \quad \text{with} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)}
\end{aligned}$$

GMM - Algorithm

- Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k
- Iterate until convergence:
  - E-step: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = p(z^{(n)}|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \Sigma_j)}$$

  - M-step: Re-estimate the parameters given current responsibilities

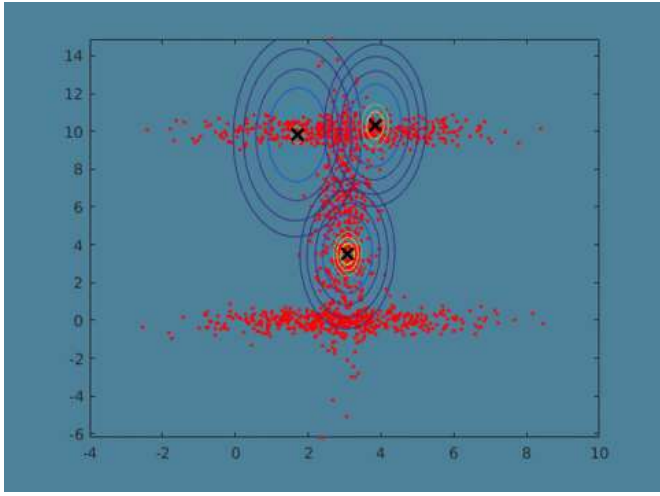
$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\mathbf{x}^{(n)} - \mu_k)(\mathbf{x}^{(n)} - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N} \quad \text{with} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)}$$

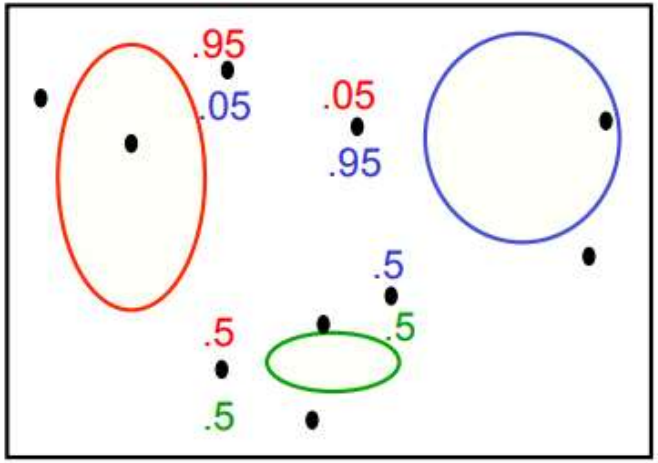
  - Evaluate log likelihood and check for convergence

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k) \right)$$



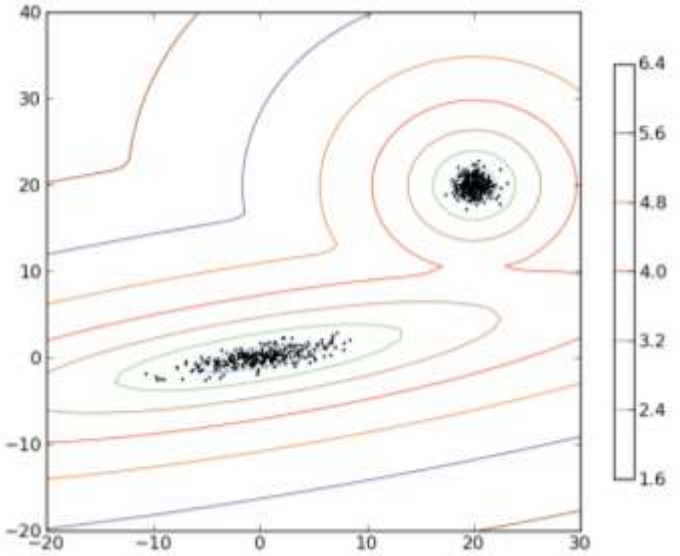
GMM - Advantages

- **Flexibility in modelling the data**
  - k-Means assumes that clusters are spherical
- **Uncertainty Estimation of data assignments**
  - Soft assignments

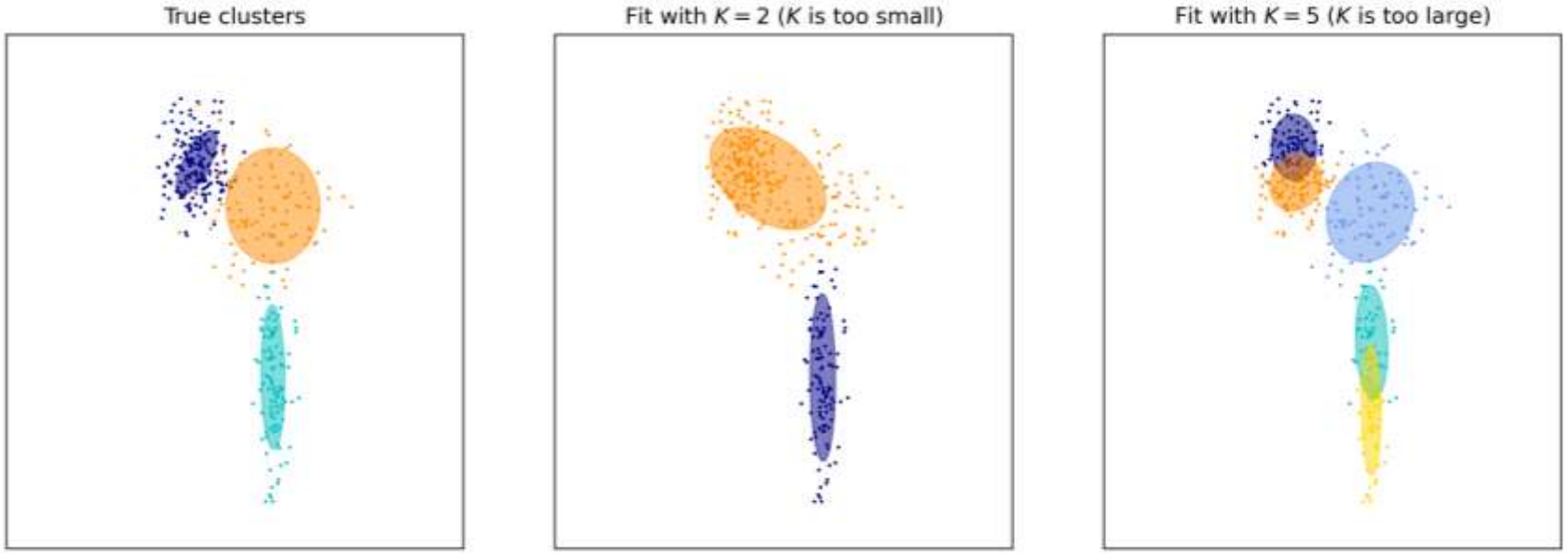


GMM - Advantages

- **Density Estimation**
  - Useful for identifying outliers, anomalies (low p(x))
  - Useful as a generative model
- **Less sensitive to initializa**
  - Soft assignments
    - In E-step
    - In M-step



Choosing k properly is crucial



https://mbernste.github.io/posts/gmm_em/

How to choose k ?

- Simple: Pick a ‘k’ which generates maximum likelihood for a ‘hold out’ set
- Log-likelihood:

$$\ell(\boldsymbol{\pi}, \mu, \Sigma) = \ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln p(\mathbf{x}^{(n)}|\pi, \mu, \Sigma)$$

$$= \sum_{n=1}^{N} \ln \sum_{z^{(n)}=1}^{K} p(\mathbf{x}^{(n)}|\, z^{(n)}; \mu, \Sigma) p(z^{(n)}|\, \boldsymbol{\pi})$$

Alternative criteria exist → Cross-validation, Information-Theoretic (AIC, BIC)

Comparing GMM and k-means

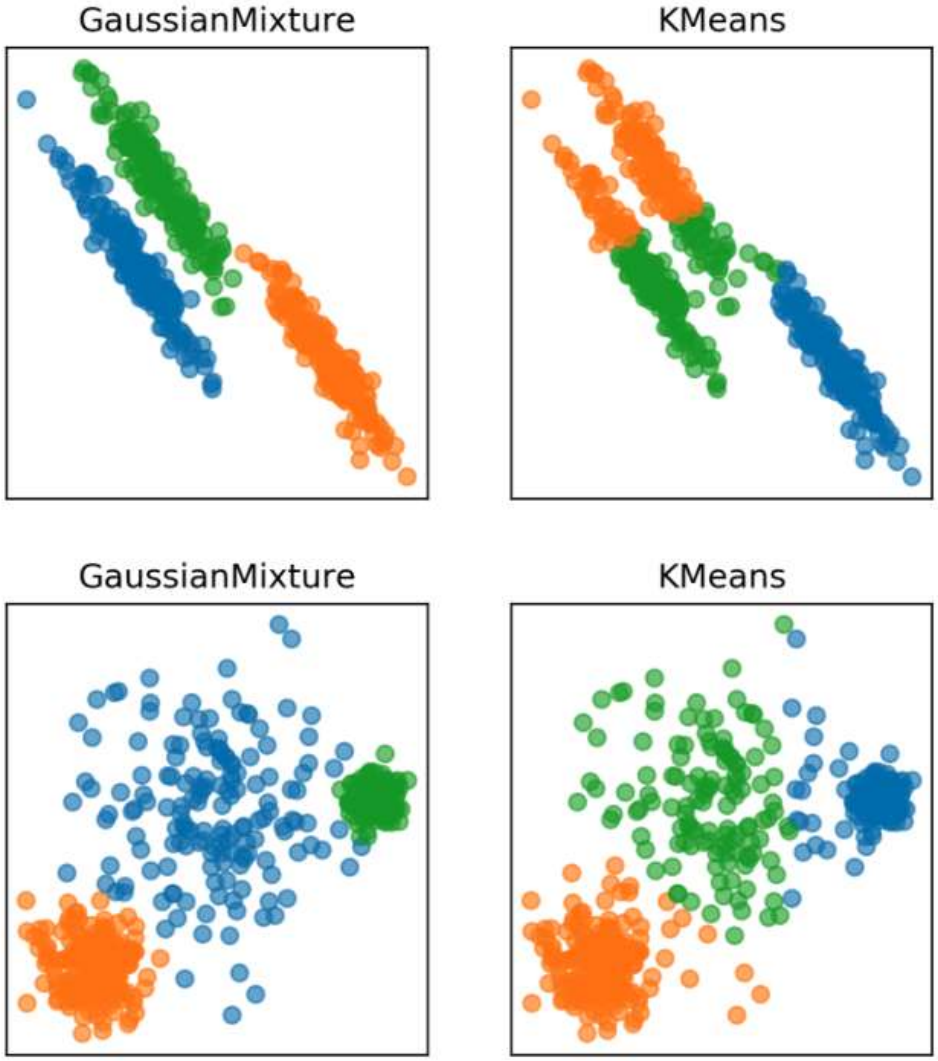
- The K-Means Algorithm:
  1. Assignment step: Assign each data point to the closest cluster
  2. Refitting step: Move each cluster center to the center of gravity of the data assigned to it
- The EM Algorithm:
  1. E-step: Compute the posterior probability over z given our current model
  2. M-step: Maximize the probability that it would generate the data it is currently responsible for.

Comparing GMM and k-means

- The K-Means Algorithm:
  1. Assignment step: Assign each data point to the closest cluster
  2. Refitting step: Move each cluster center to the center of gravity of the data assigned to it
- The EM Algorithm:
  1. E-step: Compute the posterior probability over z given our current model
  2. M-step: Maximize the probability that it would generate the data it is currently responsible for.

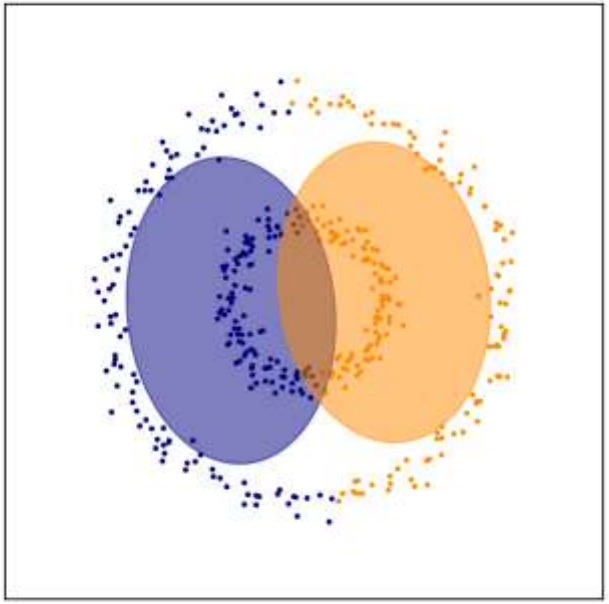
- E-step in GMM a soft version of K-means. $r_{ik} \in [0, 1]$ instead of $\{0, 1\}$.
- M-step in GMM estimates the probabilities and the covariance matrix of each cluster in addition to the means.
- All π_k are equal. $\Sigma_k = \delta^2 I$. As $\delta^2 \to 0$, $r_{ik} \to \{0, 1\}$, and the two methods coincide.

K-means v/s GMM



https://amueller.github.io/aml/03-unsupervised-learning/02-clustering-mixture-models.html

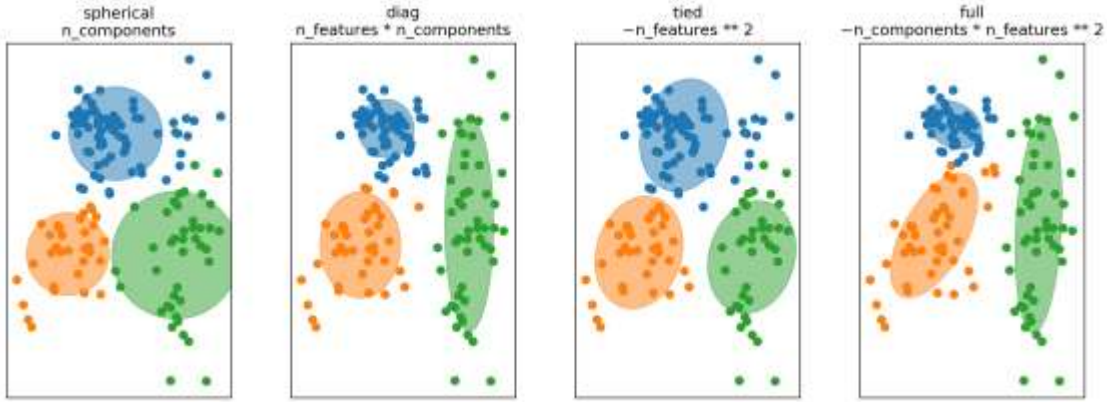
GMM: A failure case



https://mbernste.github.io/posts/gmm_em/

GMM: Problematic scenarios

- Higher dimensions
  - Numerical instability
- Insufficient data (N vs #params)
- Restricting Σ



Resources

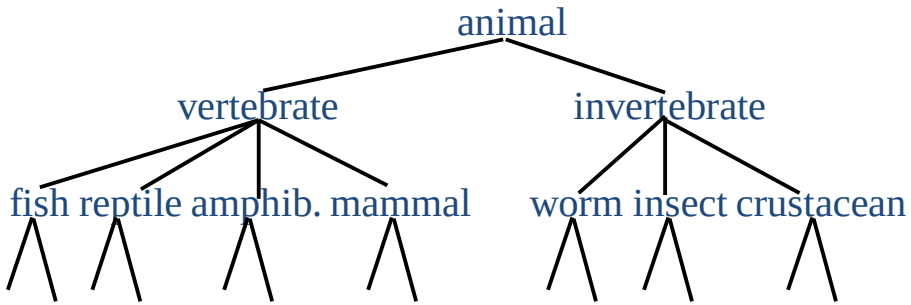
- Textbook
  - PRML (Bishop) – Chapter 9: 9.1,9.2,9.3.2
  - Pattern Classification (Duda, Hart, Stork)
    - 10.4.3,10.6.1,10.7.1,10.7.2,10.8,10.10
- Videos
  - https://www.youtube.com/watch?v=REypj2sy_5U&list=PLBv09BD7ez_4e9LtmK626Evn1ion6ynrt
  - https://www.youtube.com/watch?v=rVfZHWTwXSA
- Blog posts/Lecture Notes
  - https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec7.pdf
  - https://see.stanford.edu/materials/aimlcs229/cs229-notes8.pdf
  - https://www.cs.toronto.edu/~jlucas/teaching/csc411/lectures/lec15_16_handout.pdf
  - https://mbernste.github.io/posts/gmm_em/
  - https://www.ritchievink.com/blog/2019/05/24/algorithm-breakdown-expectation-maximization/

Hierarchical Clustering

Adapted from Slides by Prabhakar Raghavan, Christopher Manning, Ray Mooney and Soumen Chakrabarti

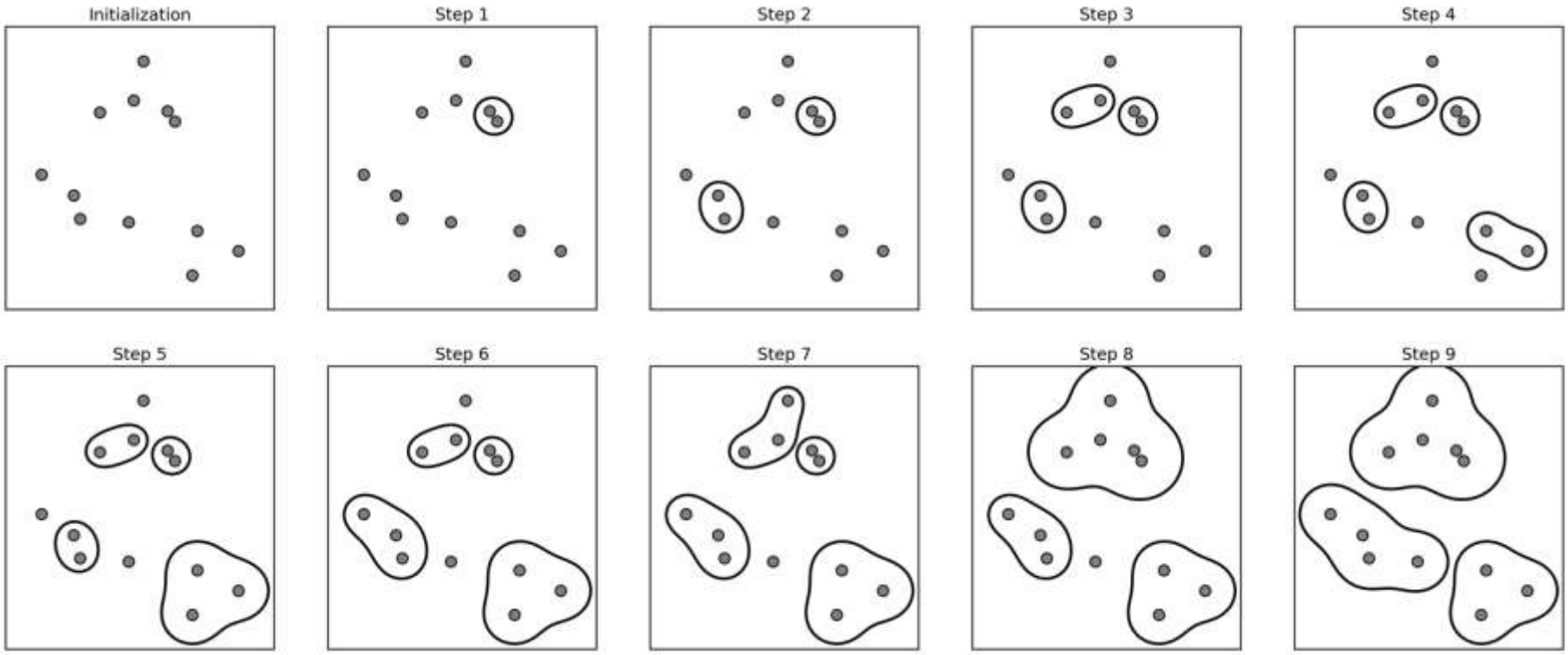
Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of samples.



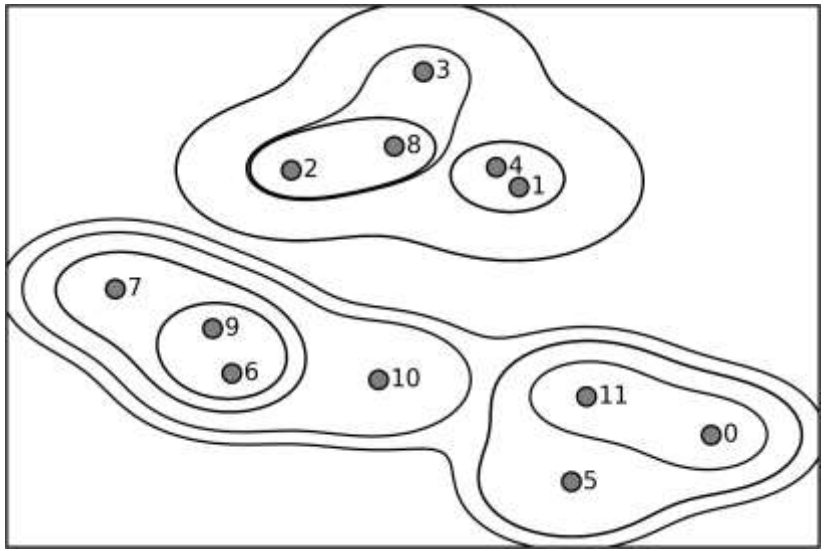
- *One approach*: recursive application of a partitional clustering algorithm.

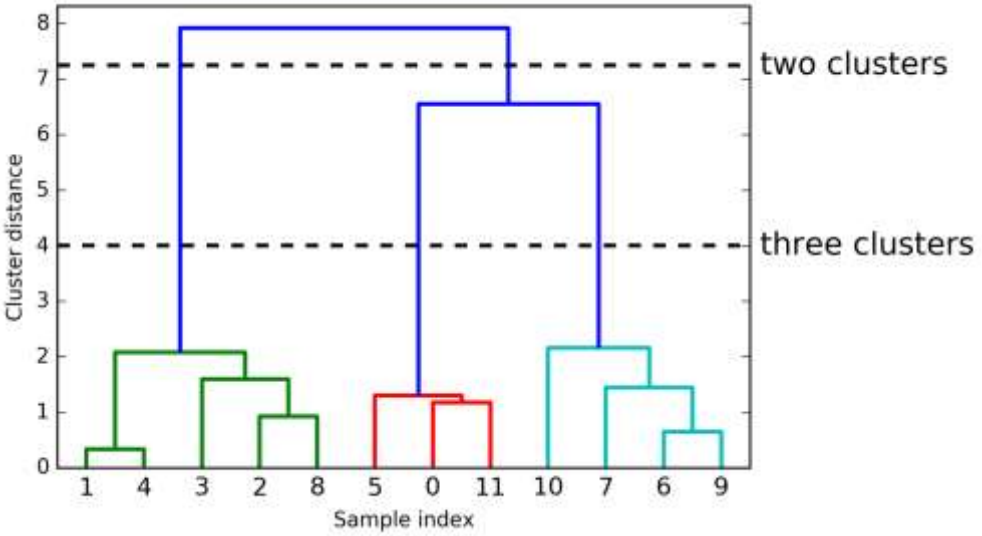
Bottom-up (Agglomerative) Clustering



https://amueller.github.io/aml/03-unsupervised-learning/02-clustering-mixture-models.html

Dendrogram: A cluster visualization



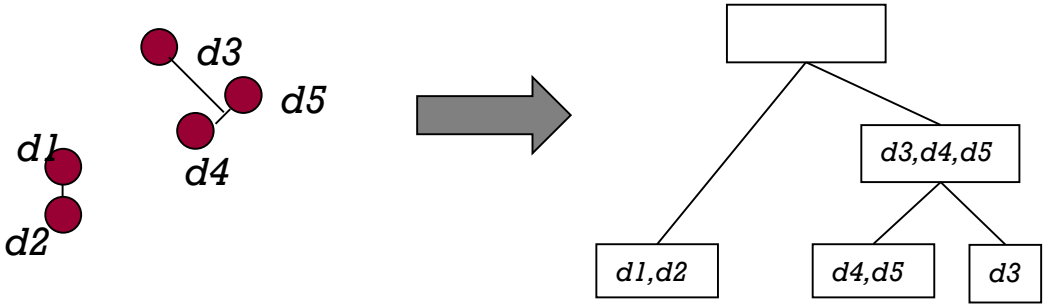


Hierarchical Clustering algorithms

- **Agglomerative (bottom-up):**
  - Start with each sample being a single cluster.
  - Eventually all samples belong to the same cluster.
- **Divisive (top-down):**
  - Start with all samples belonging to same cluster.
  - Eventually each node forms a cluster on its own.
- Does not require the number of clusters ***k*** in advance
- Needs a termination condition

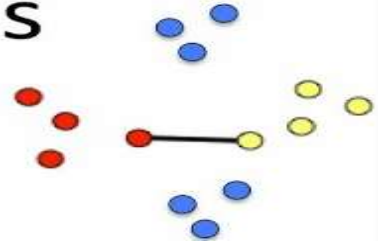
Dendrogram: Example

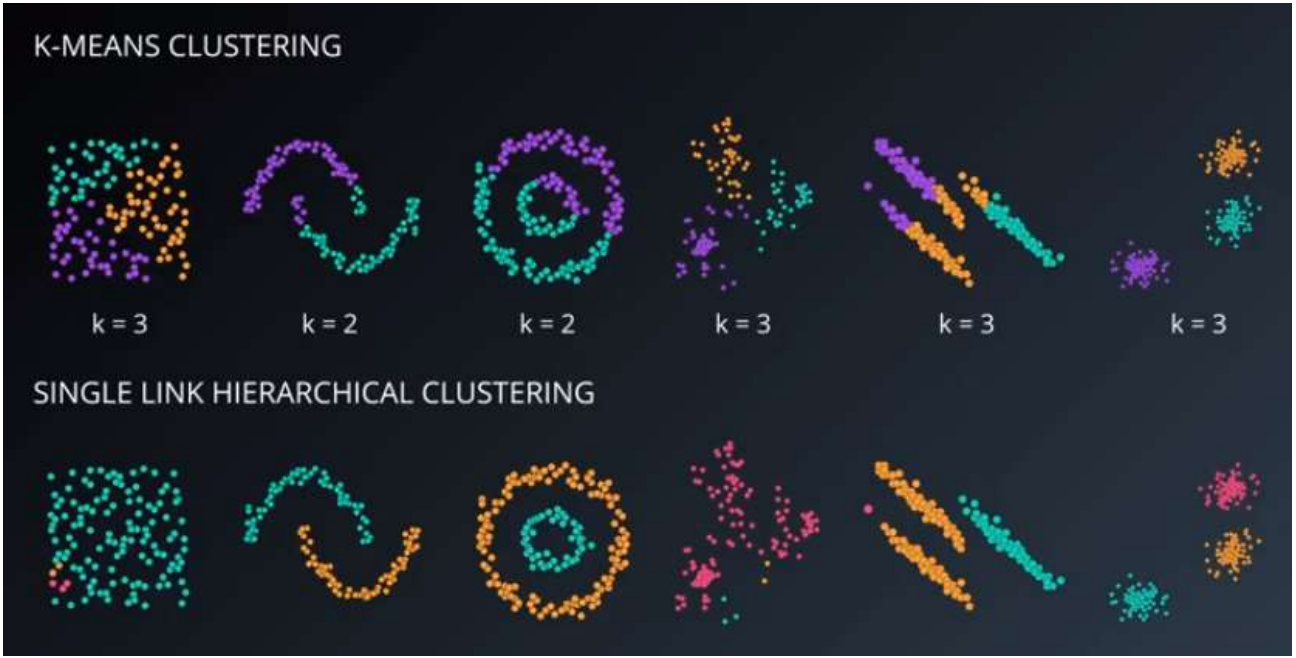
- As clusters *agglomerate*, samples likely to fall into a hierarchy of “topics” or concepts.

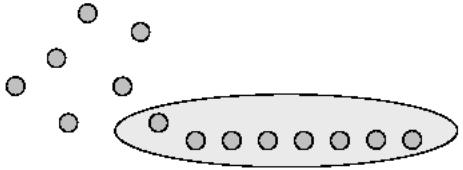


Cluster distance measures

- **Single link:** $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z





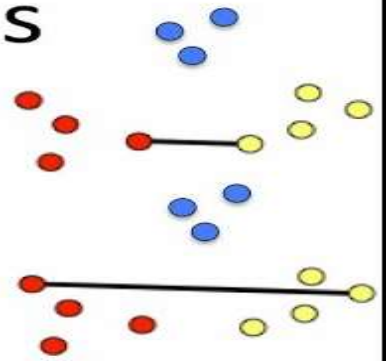


https://www.youtube.com/watch?app=desktop&v=vg1w5ZUF5lA

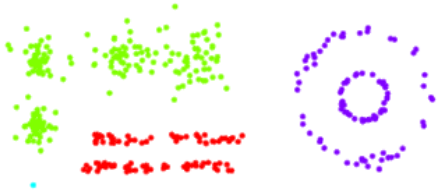
https://ryanwingate.com/intro-to-machine-learning/unsupervised/hierarchical-and-density-based-clustering/

Cluster distance measures

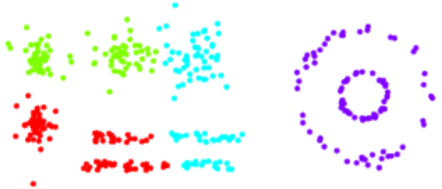
- **Single link:** $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- **Complete link:** $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"



single linkage,4 clusters



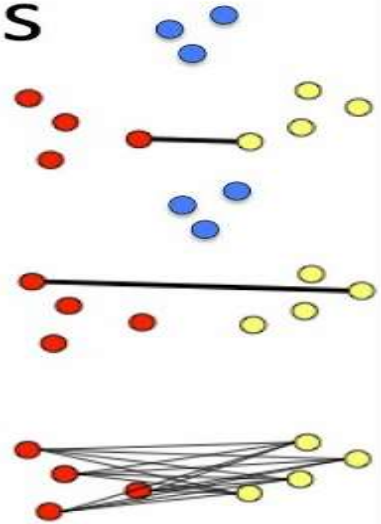
complete linkage, 4 clusters



https://www.youtube.com/watch?app=desktop&v=vg1w5ZUF5lA

Cluster distance measures

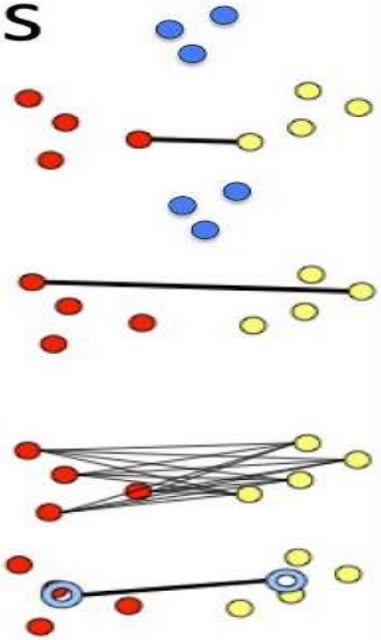
- **Single link:** $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- **Complete link:** $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"
- **Average link:** $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  - average of all pairwise distances



https://www.youtube.com/watch?app=desktop&v=vg1w5ZUF5lA

Cluster distance measures

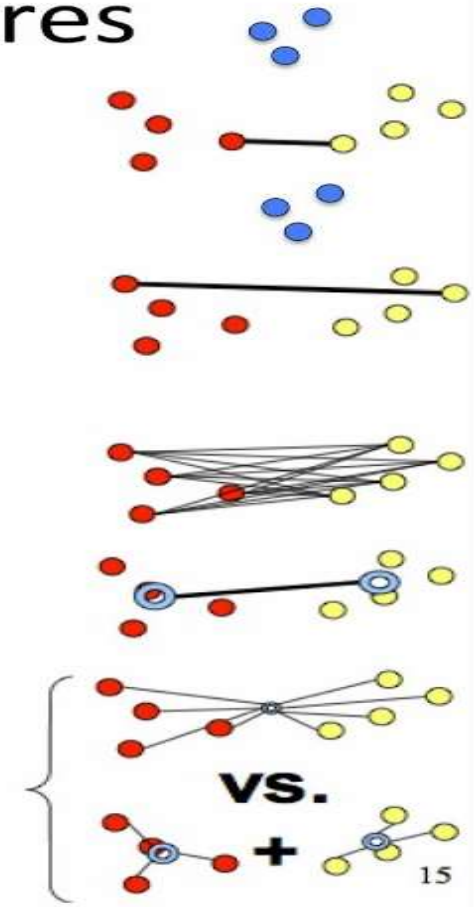
- **Single link:** $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- **Complete link:** $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"
- **Average link:** $D(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1, x_2)$
  - average of all pairwise distances
  - less affected by outliers
- **Centroids:** $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x \in c_1}\vec{x}\right), \left(\frac{1}{|c_2|}\sum_{x \in c_2}\vec{x}\right)\right)$
  - distance between centroids (means) of two clusters



https://www.youtube.com/watch?app=desktop&v=vg1w5ZUF5lA

Cluster distance measures

- **Single link:** $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- **Complete link:** $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"
- **Average link:** $D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
  - average of all pairwise distances
  - less affected by outliers
- **Centroids:** $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x \in c_1} \vec{x}\right), \left(\frac{1}{|c_2|}\sum_{x \in c_2} \vec{x}\right)\right)$
  - distance between centroids (means) of two clusters
- **Ward's method:** $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$
  - consider joining two clusters, how does it change the total distance (TD) from centroids?



Major issue - labeling

- After clustering algorithm finds clusters - how can they be useful to the end user?
- Need pithy label for each cluster
  - In search results, say “Animal” or “Car” in the ***Jaguar*** example.
  - In topic trees, need navigational cues.
    - Often done by hand, a posteriori.

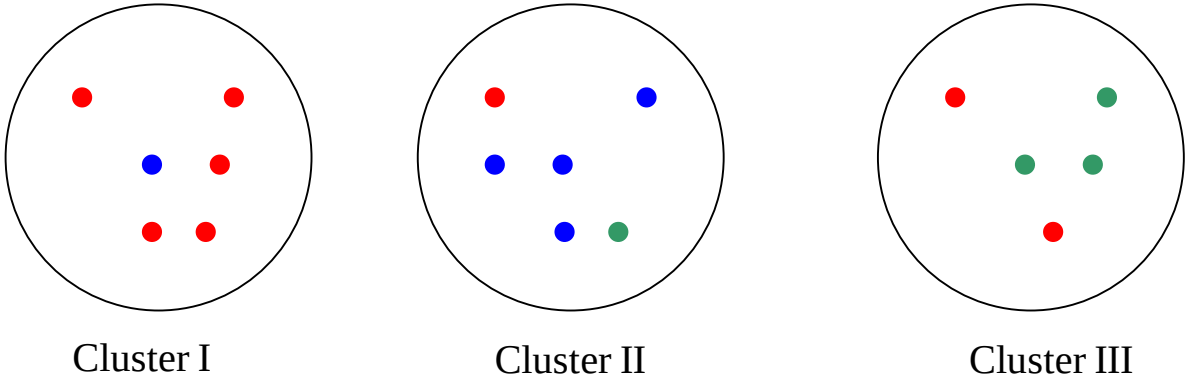
What is a Good Clustering?

- *Internal criterion*: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the feature representation and the similarity measure used

External criteria for clustering quality

- Assesses a clustering with respect to ground truth
- Assume samples with *C* gold standard classes, while our clustering algorithms produce *K* clusters, ω_1, ω_2, …, ω_K with n_i members.

Purity example



Cluster I Cluster II Cluster III

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5

External Evaluation of Cluster Quality

- *Simple measure*: purity, the ratio between the dominant class in the cluster π_i and the size of cluster ω_i

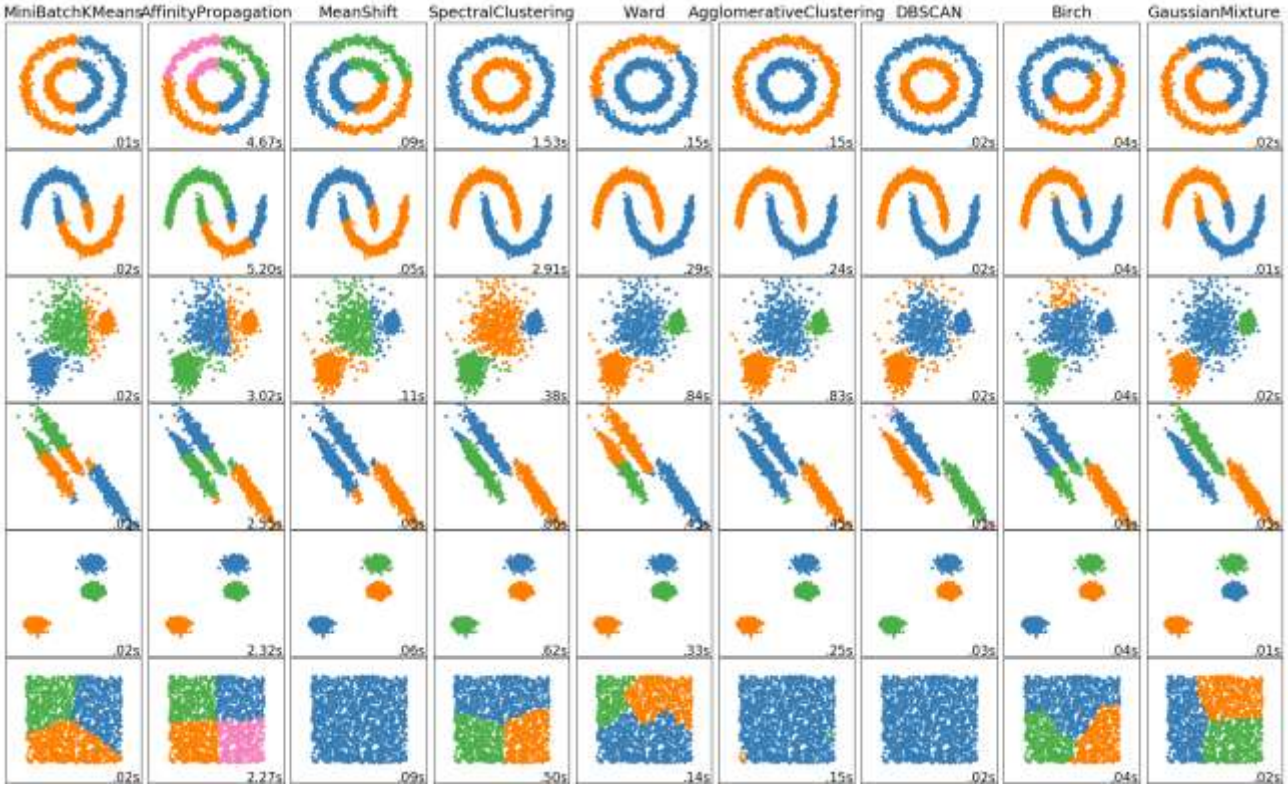
$$Purity(\omega_i) = \frac{1}{n_i} \max_j (n_{ij}) \quad j \in C$$

- Others are entropy of classes in clusters (or mutual information between classes and clusters)

Evaluation of clustering

- Perhaps the most substantive issue in ML:
  - how do you measure goodness?
- Most measures focus on computational efficiency
  - Time and space
- For application of clustering to search:
  - Measure retrieval effectiveness

Comparison of clustering methods



http://scikit-learn.org/dev/auto_examples/cluster/plot_cluster_comparison.html

References

- Pattern Classification (Duda, Hart, Stork)
  - 10.9 (Hierarchical Clustering)