Clustering

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2023-2024

Giorgio Gambosi

Partitional clustering

Given a dataset $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, with $\mathbf{x}_i \in \mathbb{R}^d (i = 1, \ldots, n)$.

We wish to derive a set of **clusters** (i.e. a partition of $\mathbf{X}$ into subsets of "near" elements). Clusters are represented by their **prototypes** $(\mathbf{m}_1, \ldots, \mathbf{m}_k)$, with $\mathbf{m}_j \in \mathbb{R}^d, j = 1, \ldots, k$.

Representation of a clustering

1. Cluster prototypes $(\mathbf{m}_1, \ldots, \mathbf{m}_k)$, with $\mathbf{m}_j \in \mathbb{R}^d (j = 1, \ldots, k)$
2. Element assignment to clusters: for each $\mathbf{x}_i$, k binary flags $r_{ij} \in \{0, 1\}$, $j = 1, \ldots, k$. If $\mathbf{x}_i$ is assigned the t-th cluster, then $r_{it} = 1$ and $r_{ij} = 0$ for $j \neq t$

Clustering types

- Partitional clustering: Given a set of items (points) $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we wish to partition $\mathbf{X}$ by assigning each element to one out of k clusters $C_1, \ldots, C_k$ in such a way to maximize (or minimize) a given cost J. The number k of clusters could be given or should have to be computed.
- Hierarchical clustering: Given a set of items (points) $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we wish to derive a set of nested partitions of $\mathbf{X}$, from the partition composed by all singletons (one cluster for each node) to the one composed by a single item (the whole set).

Clustering cost

Sum of squares

Let us define the cost a clustering as follows:

$$J(R, M) = \sum_{i=1}^{k} \sum_{j=1}^{n} r_{ij} ||\mathbf{x}_j - \mathbf{m}_i||^2 = \sum_{i=1}^{k} \sum_{j=1}^{n} r_{ij} (\mathbf{x}_j - \mathbf{m}_i)^T (\mathbf{x}_j - \mathbf{m}_i)$$

where

- $R_{ij} = r_{ij}$, where $r_{is} = 1$ and $r_{ij} = 0$ for $j \neq s$ if x_i is assigned to cluster C_s
- $M_i = \mathbf{m}_i$, $i = 1, \ldots, k$ is the prototype (centroid) of cluster C_i,

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{j=1}^{n} r_{ij} \mathbf{x}_j$$

k-means clustering

Dataset $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, $\mathbf{x}_i \in \mathbb{R}^d$: we wish to derive k clusters with prototypes $\mathbf{m}_1, \ldots, \mathbf{m}_k$

Assignment of elements to cluster: for each $\mathbf{x}_i$, k binary flags r_{ij} $(j = 1, \ldots, k)$

- if $\mathbf{x}_i$ is assigned to cluster s, then $r_{is} = 1$, and $r_{ij} = 0$ for $j \neq k$

Cost: sum of the distances of each point from the prototype of the corresponding cluster

$$J(R, M) = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \, ||\mathbf{x}_i - \mathbf{m}_j||^2$$

Objective: finding r_{ij} and $\mathbf{m}_j$ $(i = 1, \ldots, n, j = 1, \ldots, k)$ to minimize $J(R, M)$

Algorithm

1. Given a set of prototypes $\mathbf{m}_{ij}$, minimize wrt r_{ij} (assigning elements to clusters).

   For each $\mathbf{x}_i$, minimize $\sum_{j=1}^{k} r_{ij} \, ||x_i - \mathbf{m}_j||^2$.

   The minimum is obtained for $r_{ik} = 1$ (and $r_{ij} = 0$ for $j \neq k$), where $||\mathbf{x}_i - \mathbf{m}_k||^2$ is the minimum distance. That is, each point is assigned to the cluster of the nearest prototype.

2. Given a set of assignments r_{ij}, minimize wrt $\mathbf{m}_{ij}$ (defining new cluster prototypes)

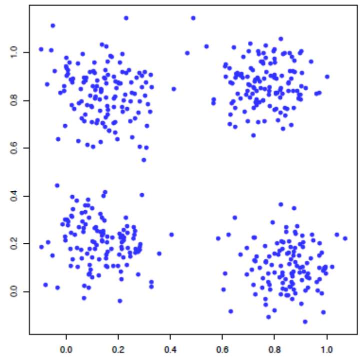
   For each $\mathbf{m}_k$, $J = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \, ||\mathbf{x}_i - \mathbf{m}_j||^2$ is a quadratic function of $\mathbf{m}_k$. By setting its derivative to zero, the values of $\mathbf{m}_k$ providing its minimum are obtained

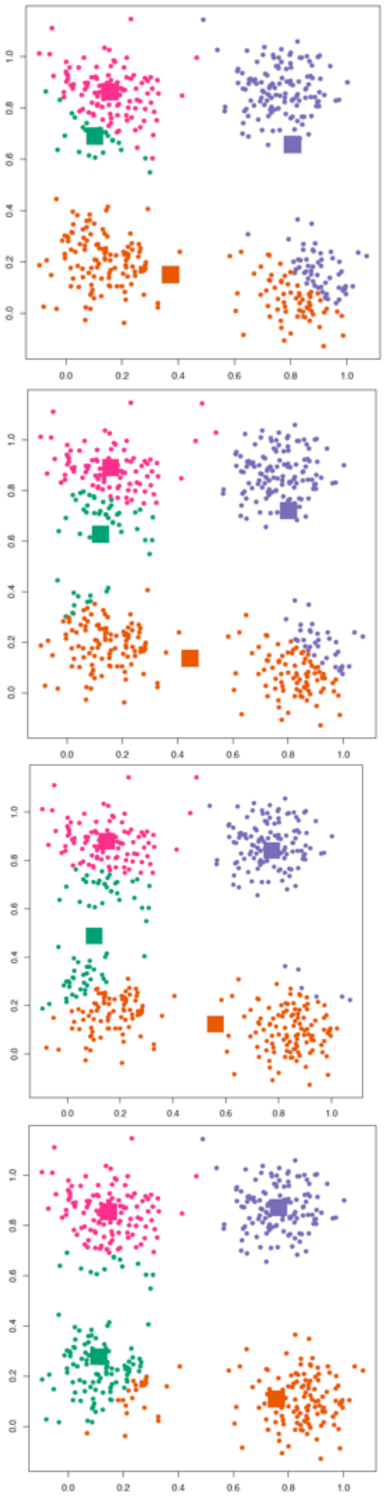
$$\frac{\partial J}{\partial \mathbf{m}_k} = 2 \sum_{i=1}^{n} r_{ik}(\mathbf{x}_i - \mathbf{m}_k) = 0 \Longrightarrow \mathbf{m}_k = \frac{\sum_{i=1}^{n} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{n} r_{ik}}$$

   That is, the new prototype is the mean of the elements assigned to the cluster

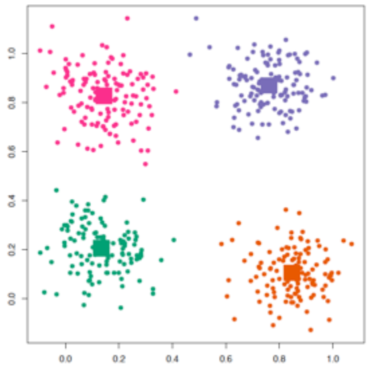
At each step, J does not increase. There is a convergence to a local minimum.

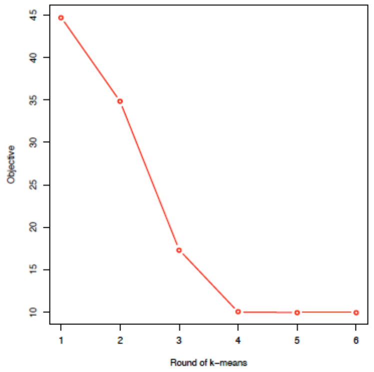
Example of application of k-means





0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0





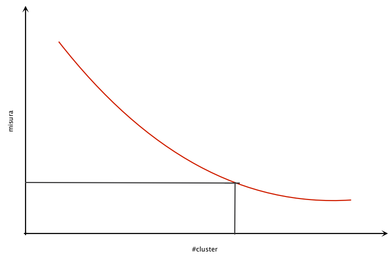
How to choose K

Cross validation

- Apply cross validation for different values of K, measuring the quality of the clustering obtained
- How to measure the quality of a clustering?
    1. mean distance of elements from the prototypes of their clusters
    2. log-likelihood of the elements wrt the resulting mixture model

Note

Measures improves as K increases (overfitting). A value such that further increases provide limited improvement should be found



Hierarchical clustering

Aim

Derivation of a binary tree. Node: cluster; arc: inclusion.

The tree specifies a set of pairwise merge of clusters.

- Aggregation, starting from n singleton clusters

- Separation, starting from a single cluster of size n

Requirements: k-means requires:

- a number K of clusters
- an initial assignment
- a distance function between elements

Hierarchical clustering requires:

- a similarity function between clusters

Algorithm

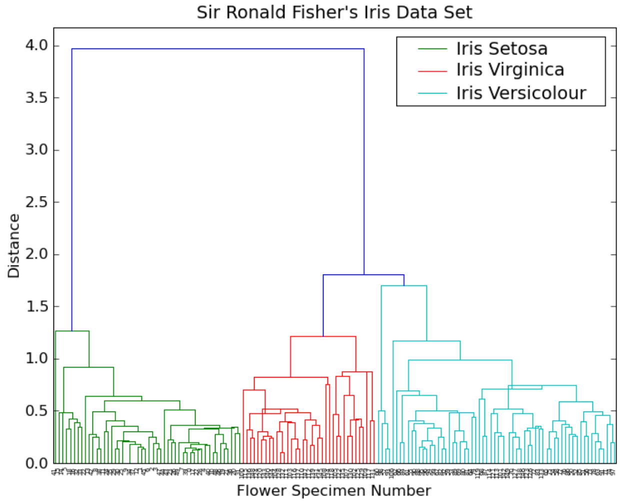
- define n clusters (singleton)
- repeat
    - compute the matrix of distances between clusters
    - merge the pair of clusters which are "nearest"
- until a single cluster has remained

Properties

- Each tree prefix is a partition of elements
- The algorithm provides a partial order of clusterings
- The best clustering has to be found
- Monotonicity: similarity between paired clusters decreases

Dendrogram

- Tree of cluster pairings
- The height of the nodes is inversely proportional to the similarity of the paired clusters



Cluster similarity

Many measures. Most frequent ones:

- Similarity between nearest nodes (**Single linkage**)

$$d_{SL}(C_1, C_2) = \min_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_i, \mathbf{x}_j)$$

- Similarity between farthest nodes (**Complete linkage**)

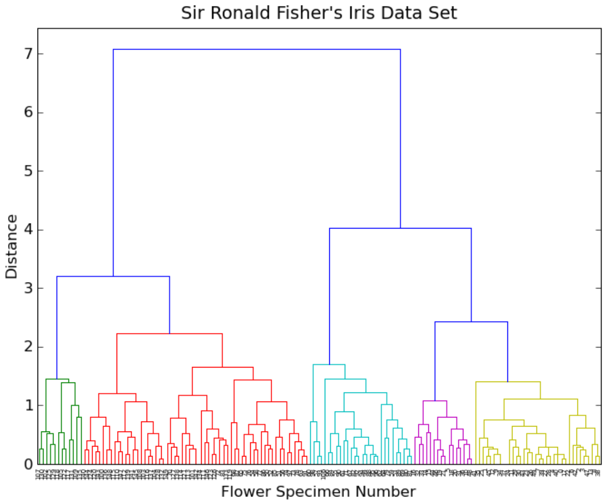
$$d_{CL}(C_1, C_2) = \max_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_i, \mathbf{x}_j)$$

- Mean similarity (**Group average**)

$$d_{GA}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{\mathbf{x}_1 \in C_1} \sum_{\mathbf{x}_2 \in C_2} d(\mathbf{x}_i, \mathbf{x}_j)$$

Different measures provide different dendrograms

Dendrogram with complete linkage



Mixtures of distributions

Linear combinations of probability distributions

- Same type of distributions $q(\mathrm{x}|\theta)$
- Differ by parameter values

$$p(\mathrm{x}|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k q(\mathrm{x}|\theta_k)$$

where

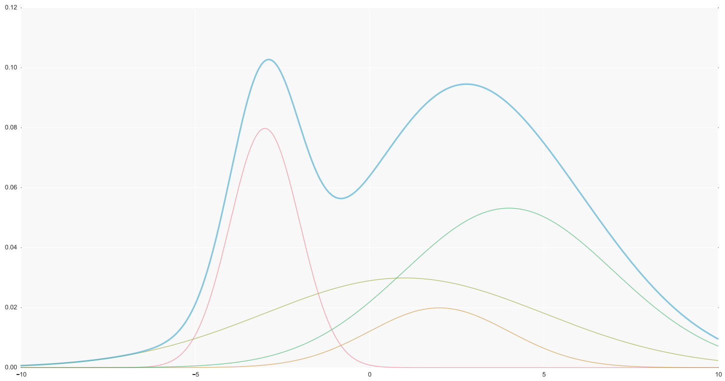
$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \qquad \boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$$

Mixing coefficients

$$0 \leq \pi_k \leq 1 \quad k = 1, \ldots, K \qquad \sum_{k=1}^{K} \pi_k = 1$$

Terms π_k have the properties of probability values

Provide extensive capabilities to model complex distributions. For example, almost all continuous distributions can be modeled by the linear combination of a suitable number of gaussians.



Mixture parameters estimation

Given a dataset $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, the parameters $\boldsymbol{\pi}, \boldsymbol{\theta}$ of a mixture can be estimated by maximum likelihood.

$$L(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X}) = p(\mathrm{X} | \boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} p(\mathrm{x}_i | \boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k q(\mathbf{x} | \theta_k)$$

or maximum log-likelihood

$$l(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X}) = \log p(\mathrm{X} | \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \log p(\mathrm{x}_i | \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i | \theta_k) \right)$$

Maximization is however constrained by the conditions $0 \leq \pi_i \leq 1$ for all i and $\sum_{i=1}^{K} \pi_i = 1$.

By applying the lagrangian multipliers method, we will maximize

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}, \lambda) = l(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X}) + \lambda(1 - \sum_{i=1}^{K} \pi_i)$$

Let us first consider the derivatives with respect to the weights $\boldsymbol{\pi}$, which we set to 0

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X})}{\partial \pi_j} = \frac{\partial l(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X})}{\partial \pi_j} - \lambda = 0$$

This is equivalent to

$$\lambda = \frac{\partial l(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X})}{\partial \pi_j} = \frac{\partial}{\partial \pi_j} \left[\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i | \theta_k) \right) \right] = \sum_{i=1}^{n} \frac{\partial}{\partial \pi_j} \left[\log \left(\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i | \theta_k) \right) \right]$$
$$= \sum_{i=1}^{n} \frac{q(\mathbf{x}_i | \theta_j)}{\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i | \theta_k)} = \sum_{i=1}^{n} \frac{\gamma_j(\mathbf{x}_i)}{\pi_j} = \frac{1}{\pi_j} \sum_{i=1}^{n} \gamma_j(\mathbf{x}_i)$$

where,

$$\gamma_k(\mathbf{x}) = \frac{\pi_k q(\mathbf{x} | \theta_k)}{\sum_{j=1}^{K} \pi_j q(\mathbf{x} | \theta_j)}$$

By setting the derivative wrt λ to 0

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X})}{\partial \lambda} = \frac{\partial}{\partial \lambda} \left(l(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathrm{X}) + \lambda(1 - \sum_{i=1}^{K} \pi_i) \right) = 0$$

we obtain

$$\sum_{i=1}^{K} \pi_i = 1$$

As a consequence, since, as shown above,

$$\pi_j = \frac{1}{\lambda} \sum_{i=1}^{n} \gamma_j(\mathbf{x}_i)$$

it results

$$\sum_{j=1}^{K} \pi_j = \frac{1}{\lambda} \sum_{j=1}^{K} \sum_{i=1}^{n} \gamma_j(\mathbf{x}_i) = 1$$

which implies

$$\lambda = \sum_{j=1}^{K} \sum_{i=1}^{n} \gamma_j(\mathbf{x}_i) = \sum_{i=1}^{n} \sum_{j=1}^{K} \gamma_j(\mathbf{x}_i) = \sum_{i=1}^{n} \sum_{j=1}^{K} \frac{\pi_j q(\mathbf{x}_i|\theta_j)}{\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i|\theta_k)} = \sum_{i=1}^{n} 1 = n$$

and, finally,

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} \gamma_k(\mathbf{x}_i)$$

For what concerns derivatives (or gradients) wrt distribution parameters $\boldsymbol{\theta}$, it results

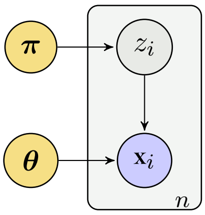
$$\begin{aligned}
\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}|\mathrm{X})}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \left[\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i|\theta_k) \right) \right] = \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \left[\log \left(\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i|\theta_k) \right) \right] \\
&= \sum_{i=1}^{n} \frac{\pi_j q(\mathbf{x}_i|\theta_j)}{\sum_{k=1}^{K} \pi_k q(\mathbf{x}_i|\theta_k)} \frac{\partial \log q(\mathbf{x}_i|\theta_j)}{\partial \theta_j} \\
&= \sum_{i=1}^{n} \gamma_j(\mathbf{x}_i) \frac{\partial \log q(\mathbf{x}_i|\theta_j)}{\partial \theta_j} = 0
\end{aligned}$$

Log likelihood maximization is intractable analytically: its solution cannot be given in closed form.

- $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ can be derived from $\gamma_k(\mathbf{x}_i)$
- Also, $\gamma_k(\mathbf{x}_i)$ can be derived from $\boldsymbol{\pi}$ e $\boldsymbol{\theta}$

Iterative techniques

- Given an estimation for $\boldsymbol{\pi}$ e $\boldsymbol{\theta}$...
- derive an estimation for $\gamma_k(\mathbf{x}_i)$, from which ...
- derive a new estimation for $\boldsymbol{\pi}$ e $\boldsymbol{\theta}$, from which ...
- derive a new estimation for $\gamma_k(\mathbf{x}_i)$ ...



Mixtures as generative processes

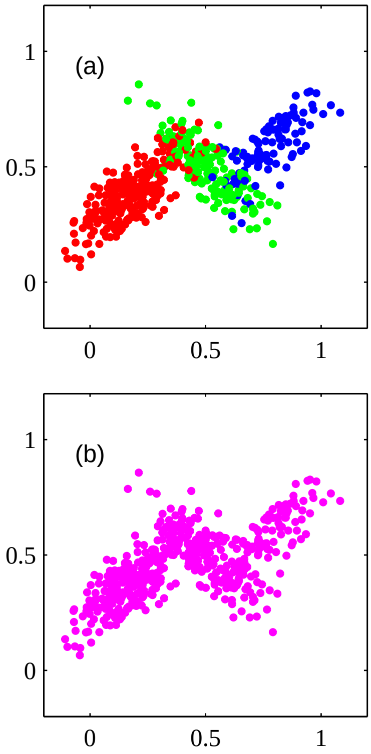
Graphical model representation of a mixture of distributions. Latent variables

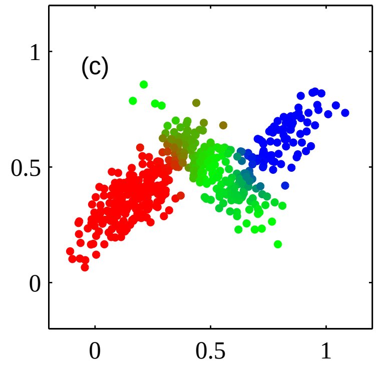
- Terms z_i are latent random variable with domain $z \in \{1, \dots, K\}$
- While $\mathbf{x}_i$ is observed, the value of z_i cannot be observed
- z_i denotes the component distribution $q(\mathbf{x}|\theta)$ responsible for the generation of $\mathbf{x}_i$

Generation process

1. Starting from the distribution $\pi_1, \dots, \pi_K$, the component distribution to apply to sample the value of $\mathbf{x}_i$ is sampled: its index is given by z_i. Hence z_i is dependent from $\boldsymbol{\pi}$
2. Let $z_i = k$: then, $\mathbf{x}_i$ is sampled from distribution $q(\mathbf{x}|\theta_k)$. That is, $\mathbf{x}_i$ is dependent from both z_i and $\boldsymbol{\theta}$ (through θ_k)

Example of generation of dataset from mixture of 3 gaussians





Distributions with latent variables

$$p(\mathbf{x}|z=k,\boldsymbol{\theta},\boldsymbol{\pi}) = p(\mathbf{x}|z=k,\boldsymbol{\theta}) = q(\mathbf{x}|\theta_k)$$

Marginalizing wrt z,

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\pi}) &= \sum_{k=1}^{K} p(\mathbf{x},z=k|\boldsymbol{\theta},\boldsymbol{\pi}) = \sum_{k=1}^{K} p(\mathbf{x}|z=k,\boldsymbol{\pi},\boldsymbol{\theta})p(z=k|\boldsymbol{\theta},\boldsymbol{\pi}) \\ &= \sum_{k=1}^{K} p(\mathbf{x}|z=k,\boldsymbol{\theta})p(z=k|\boldsymbol{\pi}) = \sum_{k=1}^{K} q(\mathbf{x}|\theta_k)p(z=k|\boldsymbol{\pi}) \end{aligned}$$

Since, by definition,

$$p(\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k q(\mathbf{x}_i|\theta_k)$$

it results

$$\pi_k = p(z=k|\boldsymbol{\pi})$$

Responsibilities

An interpretation for $\gamma_k(\mathbf{x})$ can be derived as follows

$$\begin{aligned} \gamma_k(\mathbf{x}) &= \frac{\pi_k q(\mathbf{x}|\theta_k)}{\sum_{j=1}^{K} \pi_j q(\mathbf{x}|\theta_j)} \\ &= \frac{p(z=k)p(\mathbf{x}|z=k)}{\sum_{j=1}^{K} p(z=j)p(\mathbf{x}|z=j)} = p(z=k|\mathbf{x}) \end{aligned}$$

Mixing coefficients and responsibilities

- A mixing coefficient $\pi_k = p(z=k)$ can be seen as the prior (wrt to the observation of the point) probability that the next point is generated by sampling the k-th component distribution
- A responsibility $\gamma_k(\mathbf{x}) = p(z=k|x)$ can be seen as the posterior (wrt to the observation of the point) probability that a point has been generated by sampling the k-th component distribution

In the case, of mixtures of gaussian distribution, we have $q(\mathbf{x}|\theta_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$. As a consequence,

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

and the likelihood is maximized for

$$\pi_j = \frac{1}{n}\sum_{i=1}^n \gamma_j(\mathbf{x}_i)$$

$$\sum_{i=1}^n \gamma_j(\mathbf{x}_i)\frac{\partial \log \mathcal{N}(\mathbf{x}_i|\mu_j, \Sigma_j)}{\partial \theta_j} = 0$$

Maximum likelihood

Data set

- Let $\mathrm{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the set of values of observed variables and let $\boldsymbol{Z} = (z_1, \ldots, z_n)$ be the set of values of the latent variables. Then $(\mathrm{X}, \boldsymbol{Z})$ is the complete dataset: it includes the values of all variables in the model
- X is the observed dataset (incomplete). It only includes “real” data, that is observed data.

Indeed, $\boldsymbol{Z}$ is unknown. If values have been assigned to model parameters, the only possible knowledge about $\boldsymbol{Z}$ is given by the posterior distribution $p(\boldsymbol{Z}|\mathrm{X}, \boldsymbol{\theta}, \boldsymbol{\pi})$.

Inferring parameters for gaussian mixtures

- If we assume that the complete dataset $(\mathrm{X}, \boldsymbol{Z})$ is known (that is the observed points together with their corresponding components) a maximum likelihood estimation of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ would be easy. In particular,
- For the mixing coefficients π_k it would result, as usual

  $$\pi_k = \frac{n_k}{n}$$

  where n_k is the number of elements of the set C_k such that $z = k$
- For component parameters $\theta_k = (\mu_k, \Sigma_k)$ the usual estimations for gaussians would provide

  $$\boldsymbol{\mu}_k = \frac{1}{n_k}\sum_{\mathbf{x}\in C_k}\mathbf{x}$$

  $$\Sigma_k = \frac{1}{n_k}\sum_{\mathbf{x}\in C_k}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T$$

Log likelihood of complete dataset

The above results derive from the maximimization, wrt π_k, μ_k, Σ_k, $(k = 1, \ldots, K)$ of the log likelihood

$$\begin{aligned} l(\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi}|\mathrm{X}, \boldsymbol{Z}) = \log p(\mathrm{X}, \boldsymbol{Z}|\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi}) &= \log \prod_{i=1}^n \prod_{k=1}^K \pi_k^{\zeta_{ik}} \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)^{\zeta_{ik}} \\ &= \sum_{i=1}^n \sum_{k=1}^K \zeta_{ik}(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)) \end{aligned}$$

where, ζ_{ik} is the k-component of the 1-to-K coding of z_i, that is, $\zeta_{ik} = 1$ iff $z_i = k$, and 0 otherwise

Dealing with latent variables

Unfortunately, since $\boldsymbol{Z}$ is unknown, the log-likelihood of the complete dataset cannot be defined (the sets C_k are not known).

Our approach will be to consider for maximization, instead of the log-likelihood where each z_i is specified,

- its expectation wrt to the conditional distribution $p(\boldsymbol{Z}|\mathrm{X})$, that is

$$\begin{aligned}E_{p(\boldsymbol{Z}|\mathrm{X})}[l(\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi}|\mathrm{X}, \boldsymbol{Z})] &= \sum_{i=1}^{n}\sum_{k=1}^{K} p(z_i = k|\mathbf{x}_i)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_k(\mathbf{x}_i)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k))\end{aligned}$$

Observe that this expectation can be derived if $p(\boldsymbol{Z}|\mathrm{X})$ (that is the set of all values $\gamma_k(\mathbf{x}_i)$) is known.

Maximization of expected log-likelihood

The maximization of $E_{p(\boldsymbol{Z}|\mathrm{X})}[l(\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi}|\mathrm{X}, \boldsymbol{Z})]$ wrt to π_k, μ_k, Σ_k results easily into

$$\begin{aligned}\pi_k &= \frac{1}{n}\sum_{i=1}^{n}\gamma_k(\mathbf{x}_j) \\ \mu_k &= \frac{1}{n_k}\sum_{i=1}^{n}\gamma_k(\mathbf{x}_i)\mathbf{x}_i \\ \Sigma_k &= \frac{1}{n_k}\sum_{i=1}^{n}\gamma_j(\mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\end{aligned}$$

this is named M-step (from "Maximization")

A new expectation

The computed values for the parameters result into new, different values for $\gamma_k(\mathbf{x}_i) = p(z_i = k|\mathbf{x}_i)$, and a different expectation $E_{p(\boldsymbol{Z}|\mathrm{X})}[l(\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi}|\mathrm{X}, \boldsymbol{Z})]$.

This is named E-step (from "Expectation")

ML and mixtures of gaussians: iterative approach

1. Assign an initial estimate to $\mu_j, \Sigma_j, \pi_j, j = 1, \ldots, K$
2. Repeat
   (a) Compute

$$\gamma_j(x_i) = \frac{1}{\gamma_i}\pi_j\mathcal{N}(x_i|\mu_j, \Sigma_j) \qquad \text{with} \qquad \gamma_i = \sum_{k=1}^{K}\pi_k\mathcal{N}(x_i|\mu_j, \Sigma_j)$$

   (b) Compute

$$\pi_j = \frac{n_j}{n} \qquad \text{with} \qquad n_j = \sum_{i=1}^{n}\gamma_j(x_i)$$

(c) Compute

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{n} \gamma_j(x_i) x_i$$

(d) Compute

$$\Sigma_j = \frac{1}{n_j} \sum_{i=1}^{n} \gamma_j(x_i)(x_i - \mu_j)(x_i - \mu_j)^T$$

3. until some convergence property is verified

The convergence test may refer to the the increase of log-likelihood in the last iteration

This algorithm is indeed the application of a general schema named **Expectation-Maximization**