Mixture models for cluster analysis: from model-based inference to Bayesian non-parametrics

Faicel Chamroukhi
chamroukhi.univ-tln.fr

uLearnBio@ICML 2014, Beijing

26 June 2014

Outline

Model-Based Clustering

Context

- Unsupervised learning for cluster analysis
- A latent data modeling framework
- Observed data: $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ a sample of n multidimensional individuals $\mathbf{x}_i$ in $\mathbb{R}^d$.
- Hidden variables: $\mathbf{z} = (z_1, \ldots, z_n)$ the hidden cluster labels $(z_i \in \{1, \ldots, K\}$
- K possibly unknown number of clusters clusters

Context

- Unsupervised learning for cluster analysis
- A latent data modeling framework
- Observed data: $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ a sample of n multidimensional individuals $\mathbf{x}_i$ in $\mathbb{R}^d$.
- Hidden variables: $\mathbf{z} = (z_1, \ldots, z_n)$ the hidden cluster labels $(z_i \in \{1, \ldots, K\}$
- K possibly unknown number of clusters clusters

Objective

- Infer the hidden structure from the data ⇒ find a partition of the unlabeled dataset into a finite number of clusters
- ⇒ Learn a probabilistic generative model
- Chose the best possibly unknown number of clusters

Model-based clustering

Model-based clustering

- The aim of clustering in general is to find a partition of an unlabeled dataset into clusters (groups)
- the data within the same group tend to be more similar, in the sense of a chosen dissimilarity measure, to one another as compared to the data belonging to different groups
- Model-based clustering [a] generally used for multidimensional data, is based on the finite mixture model formulation [b]
- They are one of the most popular and successful approaches in cluster analysis.

[a] McLachlan and Basford (1988); Banfield and Raftery (1993a); Fraley and Raftery (2002)
[b] McLachlan and Peel. (2000)

Clustering via finite mixture models

- $\Rightarrow$ This approach is known as the *model-based clustering*
- The clustering problem is reformulated as a density estimation problem
- the data probability density function is assumed to be a mixture density, each component density being associated with a cluster.
- $\Rightarrow$ The problem of clustering becomes the one of estimating the parameters of the assumed mixture model (e.g, estimating the means and the covariances for Gaussian mixtures).

Mixture approach/Classification approach

Two main approaches are possible. The former is refereed to as *the mixture approach* or *the estimation approach* and the latter is known as *the classification approach.*

1. **The mixture approach** consists of two steps :
   1. The parameters of the mixture density are estimated by maximizing the *observed-data likelihood* generally via the EM algorithm
   2. After performing the probability density estimation, the posterior probabilities τ_{ik} are then used to determine the cluster memberships through the MAP principle.
2. **The classification approach**
   - consists in optimizing a classification likelihood function which is (can be) the *complete-data likelihood* by using the CEM algorithm Celeux and Govaert (1992).
   - The cluster memberships and the model parameters are estimated simultaneously as the learning proceeds.

Data modeling using finite mixture models

- Finite mixture models are an example of latent variable models
- widely used in probabilistic machine learning and pattern recognition.
- The finite mixture model decomposes the density of the observed data $\mathbf{x}$ into a weighted linear combination of K component densities.
- The mixture model allows for placing K component densities in the input space to approximate the true density.

  $\Rightarrow$ Mixtures provide a natural generalization of the simple parametric density model which is global, to a weighted sum of these models, allowing local adaptation to the density of the data in the input space.

Model definition

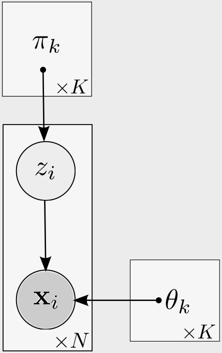
- Let z represent a discrete random variable which takes its values in the finite set $\mathcal{Z} = \{1, \ldots, K\}$.
- In a general setting, the mixture density of $\mathbf{x}$ is

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{k=1}^{K} p(z=k) f(\mathbf{x}|z=k; \boldsymbol{\Psi}_k)$$
$$= \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}; \boldsymbol{\Psi}_k),$$

  - $\pi_k = p(z=k)$: the probability that a randomly chosen data point was generated by component k. Referred to as *mixing proportions* $\pi_k \geq 0 \ \forall k$, and $\sum_{k=1}^{K} \pi_k = 1$.
  - $f_1, \ldots, f_K$ are the *component densities*.
  - Each f_k typically consists of a relatively simple parametric model $p(\mathbf{x}|z=k; \boldsymbol{\Psi}_k)$ (such as a Gaussian distribution with parameters $\boldsymbol{\Psi}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$).

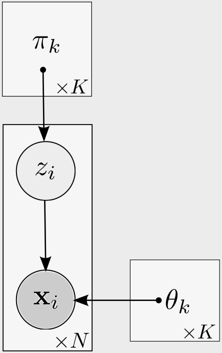
The finite mixture model

Graphical model representation of the finite mixture model



The finite mixture model

Graphical model representation of the finite mixture model

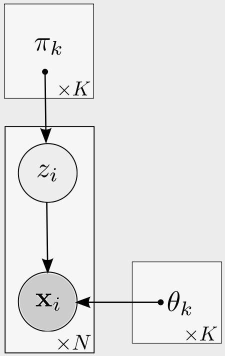


Generative model

$$z_i|\boldsymbol{\pi} \quad \sim \quad \text{Mult}(.|\boldsymbol{\pi})$$

The finite mixture model

Graphical model representation of the finite mixture model



Generative model

$$
\begin{aligned}
z_i|\boldsymbol{\pi} &\sim \text{Mult}(.|\boldsymbol{\pi}) \\
\mathbf{x}_i|z_i,\boldsymbol{\theta} &\sim f(.|\boldsymbol{\theta}_{z_i})
\end{aligned}
$$

Parameter estimation for the mixture model

Common estimation methods

- the *maximum likelihood* (ML) estimation approach
- the Maximum A Posteriori (MAP) (*Bayesian*) estimation where a prior distribution is assumed for the model parameters

- The ML approach maximizes the observed-data likelihood maximizing the observed data likelihood $p(\mathbf{X}|\boldsymbol{\theta})$

  ⇒ The used algorithm is the Expectation-Maximization (EM) algorithm [a].

- The MAP maximizes the posterior parameter distribution $p(\boldsymbol{\theta}|\mathbf{X}) = p(\boldsymbol{\theta})p(\mathbf{X}|\boldsymbol{\theta})$, $p(\boldsymbol{\theta})$ being a prior parameter distribution

  ⇒ The MAP can still be performed by EM in some cases [b] and is in general performed by Markov Chain Monte Carlo (MCMC) [c]

---

[a] Dempster et al. (1977); McLachlan and Krishnan (1997)

[b] Fraley and Raftery (2007)

[c] Richardson and Green (1997)Bensmail et al. (1997)Bensmail and Meulman (2003) Ormoneit and Tresp (1998)Stephens (1997, 2000)

Parameter estimation for the mixture model

- Assume we have an i.i.d sample $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- The observed-data log-likelihood of $\boldsymbol{\Psi}$ is:

$$\begin{aligned}\mathscr{L}(\boldsymbol{\Psi};\mathbf{X}) &= \log \prod_{i=1}^{n} p(\mathbf{x}_i;\boldsymbol{\Psi}) \\ &= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i;\boldsymbol{\Psi}_k).\end{aligned}$$

- the log-likelihood to be maximized results in a nonlinear function due to the logarithm of the sum
- very difficult to maximize in a closed form

  ⇒ maximize it (locally) using iterative procedures such as gradient ascent, a Newton Raphson procedure or the Expectation-Maximization (EM) algorithm

  ⇒ The EM algorithm is the widely technique for mixture models.

EM algorithm

- a broadly applicable approach to the iterative computation of maximum likelihood estimates in the framework of latent data models.
- In particular, the EM algorithm simplifies considerably the problem of fitting finite mixture models by maximum likelihood.

- an iterative algorithm where each iteration consists of two steps:
  1. the Expectation step (E-step): computes the expectation of the complete-data log-likelihood, given the observations $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and a current value $\boldsymbol{\Psi}^{(q)}$ of the model parameter
  2. the Maximization step (M-step): Maximize the expected complete-data log-likelihood over the parameter space

EM algorithm

- let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be a set of n i.i.d observations with $\mathbf{x}_i \in \mathbb{R}^d$
- $\mathbf{z} = (z_1, \ldots, z_n)$ denote the corresponding unobserved (missing) labels with $z_i \in \mathcal{Z} = \{1, \ldots, K\}$.
- The complete-data: $(\mathbf{X}, \mathbf{z}) = ((\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_n, z_n))$
- The complete-data log-likelihood:

$$\begin{aligned}\mathscr{L}_c(\boldsymbol{\Psi}; \mathbf{X}, \mathbf{z}) &= \log p((\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_n, z_n); \boldsymbol{\Psi}) = \log \prod_{i=1}^{n} p(\mathbf{x}_i, z_i; \boldsymbol{\Psi}) \\ &= \sum_{i=1}^{n} \log \prod_{k=1}^{K} \Big[p(z_i = k) p(\mathbf{x} | z_i = k; \boldsymbol{\Psi}_k) \Big]^{z_{ik}} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \pi_k f_k(\mathbf{x}_i; \boldsymbol{\Psi}_k),\end{aligned}$$

where $z_{ik} = 1$ if $z_i = k$ (i.e, when $\mathbf{x}_i$ is generated by the kth component density) and $z_{ik} = 0$ otherwise.

- this log-likelihood depends on the unobservable data $\mathbf{z}$!.

EM algorithm

- The EM algorithm starts with an initial parameter $\boldsymbol{\Psi}^{(0)}$ and iteratively alternates between the two following steps until convergence:
- **E-step (Expectation):** computes the expectation of the complete-data log-likelihood given the observations $\mathbf{X}$ and the current value $\boldsymbol{\Psi}^{(q)}$ of the parameter $\boldsymbol{\Psi}$ (q being the current iteration).

$$
\begin{aligned}
Q(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{(q)}) &= \mathbb{E}\Big[\mathscr{L}_c(\boldsymbol{\Psi}; \mathbf{X}, \mathbf{z}) | \mathbf{X}; \boldsymbol{\Psi}^{(q)}\Big] \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[z_{ik} | \mathbf{x}_i, \boldsymbol{\Psi}^{(q)}] \log \pi_k f_k(\mathbf{x}_i; \boldsymbol{\Psi}_k) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} p(z_{ik} = 1 | \mathbf{x}_i; \boldsymbol{\Psi}^{(q)}) \log \pi_k f_k(\mathbf{x}_i; \boldsymbol{\Psi}_k) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(q)} \log \pi_k f_k(\mathbf{x}_i; \boldsymbol{\Psi}_k)
\end{aligned}
$$

EM algorithm

- where

$$\tau_{ik}^{(q)} = p(z_i = k|\mathbf{x}_i; \mathbf{\Psi}^{(q)}) = \frac{\pi_k f_k(\mathbf{x}_i; \mathbf{\Psi}_k^{(q)})}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(\mathbf{x}_i; \mathbf{\Psi}_\ell^{(q)})}$$

  is the posterior probability that $\mathbf{x}_i$ originates from the kth component density.

- In $\mathbb{E}[z_{ik}|\mathbf{x}_i, \mathbf{\Psi}^{(q)}]$, we used the fact that conditional expectations and conditional probabilities are the same for the indicator binary-valued variables z_{ik}: $\mathbb{E}[z_{ik}|\mathbf{x}_i, \mathbf{\Psi}^{(q)}] = p(z_{ik} = 1|\mathbf{x}_i, \mathbf{\Psi}^{(q)})$.

$\Rightarrow$ From the expression of $Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)})$, we can see that this step simply requires the computation of the posterior probabilities $\tau_{ik}^{(q)}$.

EM algorithm

M-step (Maximization): updates the estimate of $\mathbf{\Psi}$ by the value $\mathbf{\Psi}^{(q+1)}$ of $\mathbf{\Psi}$ that maximizes the Q-function $Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)})$ with respect to $\mathbf{\Psi}$ over the parameter space $\mathbf{\Omega}$:

$$\mathbf{\Psi}^{(q+1)} = \arg\max_{\mathbf{\Psi} \in \mathbf{\Omega}} Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)}).$$

We can write

$$Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)}) = Q_\pi(\pi_1, \ldots, \pi_K, \mathbf{\Psi}^{(q)}) + \sum_{k=1}^{K} Q_{\mathbf{\Psi}_k}(\mathbf{\Psi}_k, \mathbf{\Psi}^{(q)})$$

where

$$Q_\pi(\pi_1, \ldots, \pi_K, \mathbf{\Psi}^{(q)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(q)} \log \pi_k$$

$$Q_{\mathbf{\Psi}_k}(\mathbf{\Psi}_k, \mathbf{\Psi}^{(q)}) = \sum_{i=1}^{n} \tau_{ik}^{(q)} \log f_k(\mathbf{x}_i; \mathbf{\Psi}_k)$$

M-Step

⇒ the maximization of the function $Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(q)})$ w.r.t $\boldsymbol{\Psi}$ can be performed by separately maximizing Q_π with respect to the mixing proportions $(\pi_1,\ldots,\pi_K)$ and $Q_{\boldsymbol{\Psi}_k}$ with respect to parameters $\boldsymbol{\Psi}_k$ for each of the K components densities.

- The function Q_π is maximized with respect to $(\pi_1,\ldots,\pi_K)\in[0,1]^K$ subject to the constraint $\sum_k \pi_k = 1$. This maximization is done in a closed using Lagrange multipliers form and leads to

$$\pi_k^{(q+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(q)}}{n} = \frac{n_k^{(q)}}{n},$$

- $n_k^{(q)}$ can be viewed as the expected cardinal number of the subpopulation k estimated at iteration q.
- The update of $\boldsymbol{\Psi}_k$ depends on the form of the density f_k (e.g., Gaussian)

EM for Gaussian mixture models (GMMs)

The finite Gaussian mixture density is defined as::

$$f(\mathbf{x}_i;\boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

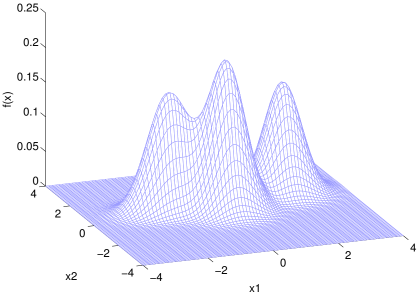
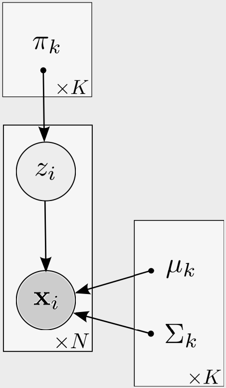


Figure : An example of a three-component Gaussian mixture density in $\mathbb{R}^2$.

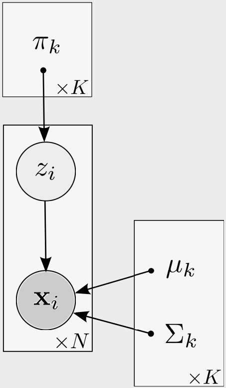
The finite Gaussian Mixture Model (GMM)

Graphical model representation of the finite GMM



The finite Gaussian Mixture Model (GMM)

Graphical model representation of the finite GMM

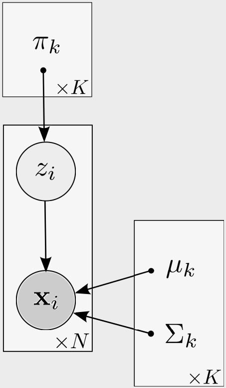


Generative model

$$z_i|\boldsymbol{\pi} \quad \sim \quad \text{Mult}(.|\boldsymbol{\pi})$$

The finite Gaussian Mixture Model (GMM)

Graphical model representation of the finite GMM



Generative model

$$
\begin{aligned}
z_i|\boldsymbol{\pi} &\sim \text{Mult}(.|\boldsymbol{\pi}) \\
\mathbf{x}_i|z_i = k,\boldsymbol{\theta} &\sim \mathcal{N}(.|\boldsymbol{\theta}_k)
\end{aligned}
$$

$\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

EM for GMMs

- The observed-data log-likelihood of $\mathbf{\Psi}$ for the Gaussian mixture model:

$$\mathscr{L}(\mathbf{\Psi};\mathbf{X}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k,\mathbf{\Sigma}_k).$$

- The complete-data log-likelihood of $\mathbf{\Psi}$ for the Gaussian mixture model:

$$\mathscr{L}_c(\mathbf{\Psi};\mathbf{X},\mathbf{z}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log \pi_k \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k\mathbf{\Sigma}_k).$$

EM:

- Starts with an initial parameter $\mathbf{\Psi}^{(0)} = (\pi_1^{(0)},\ldots,\pi_K^{(0)},\mathbf{\Psi}_1^{(0)},\ldots,\mathbf{\Psi}_K^{(0)})$ where $\mathbf{\Psi}_k^{(0)} = (\boldsymbol{\mu}_k^{(0)},\mathbf{\Sigma}_k^{(0)})$

E-Step for GMMs

- the expected complete-data log-likelihood:

$$\begin{aligned} Q(\boldsymbol{\Psi},\boldsymbol{\Psi}^{(q)}) &= \mathbb{E}\Big[\mathscr{L}_c(\boldsymbol{\Psi};\mathbf{X},\mathbf{z})|\mathbf{X};\boldsymbol{\Psi}^{(q)}\Big] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\pi_k + \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \end{aligned}$$

$\Rightarrow$ This step therefore computes the posterior probabilities

$$\tau_{ik}^{(q)} = p(z_i = k|\mathbf{x}_i,\boldsymbol{\Psi}^{(q)}) = \frac{\pi_k\mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k^{(q)},\boldsymbol{\Sigma}_k^{(q)})}{\sum_{\ell=1}^{K}\pi_\ell\mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_\ell^{(q)},\boldsymbol{\Sigma}_\ell^{(q)})}$$

that $\mathbf{x}_i$ originates from the kth component density.

M-Step for GMMs

- update the parameter $\mathbf{\Psi}$ by the value $\mathbf{\Psi}^{(q+1)}$ of $\mathbf{\Psi}$ that maximizes the function $Q(\mathbf{\Psi}, \mathbf{\Psi}^{(q)})$ w.r.t $\mathbf{\Psi}$ over the parameter space $\mathbf{\Omega}$.

$$\begin{aligned}\boldsymbol{\mu}_k^{(q+1)} &= \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{x}_i, \\ \mathbf{\Sigma}_k^{(q+1)} &= \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} \tau_{ik}^{(q)} (\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})(\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})^T.\end{aligned}$$

- The E- and M-steps are alternated iteratively until the change in the log likelihood value are less than some specified threshold.

Algorithm 1 Pseudo code of the EM algorithm for GMMs.

Inputs: Data set $(\mathbf{x}_1,\ldots,\mathbf{x}_n)$ and # of clusters K

fix a threshold $\epsilon > 0$; set $q \leftarrow 0$ (iteration)

Initialize: $\boldsymbol{\Psi}^{(0)} = (\pi_1^{(0)},\ldots,\pi_K^{(0)},\boldsymbol{\Psi}_1^{(0)},\ldots,\boldsymbol{\Psi}_K^{(0)})$ with $\boldsymbol{\Psi}_k^{(0)} = (\boldsymbol{\mu}_K^{(0)},\boldsymbol{\Sigma}_K^{(0)})$

while increment in log-likelihood $> \epsilon$ **do**

E-step:

for $k = 1,\ldots,K$ **do**

Compute $\tau_{ik}^{(q)} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k^{(q)},\boldsymbol{\Sigma}_k^{(q)})}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_\ell^{(q)},\boldsymbol{\Sigma}_\ell^{(q)})}$ for $i = 1,\ldots,n$

end for

M-step:

for $k = 1,\ldots,K$ **do**

Compute $\pi_k^{(q+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(q)}}{n}$

Compute $\boldsymbol{\mu}_k^{(q+1)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{x}_i$

Compute $\boldsymbol{\Sigma}_k^{(q+1)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} \tau_{ik}^{(q)} (\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})(\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})^T$

end for

$q \leftarrow q + 1$

end while

Outputs: $\widehat{\boldsymbol{\psi}} = \boldsymbol{\Psi}^{(q)}$; $\widehat{\tau}_{ik} = \tau_{ik}^{(q)}$ (a fuzzy partition of the data)

Initialization Strategies and stopping rules for EM

- The initialization of EM is a crucial point since it maximizes locally the log-likelihood.
- if the initial value is inappropriately selected, the EM algorithm may lead to an unsatisfactory estimation.
- The most used strategy: use several EM tries and select the solution maximizing the log-likelihood among those runs.
- For each run of EM, one can initialize it
  - randomly
  - by Computing a parameter estimate from another clustering algorithm such as K-means, Classification EM, Stochastic EM ...
  - with a few number of steps of EM itself.
- Stop EM when the relative increase of the log-likelihood between two iterations is below a fixed threshold $|\frac{\mathcal{L}^{(q+1)} - \mathcal{L}^{(q)}}{\mathcal{L}^{(q)}}| \leq \epsilon$ or when a predefined number of iterations is reached.

EM properties

- The EM algorithm always monotonically increases the observed-data log-likelihood.
- The sequence of parameter estimates generated by the EM algorithm converges toward at least a local maximum or a stationary value of the incomplete-data likelihood function.
- numerical stability
- simplicity of implementation
- reliable convergence
- In general, both the E- and M-steps will have particularly simple forms when the complete-data probability density function is from the exponential family;
- Some drawbacks: EM is sometimes very slow to converge especially for high dimensional data;

  in some problems, the E- or M-step may be analytically intractable (but this can be tackled by using EM extensions)

EM extensions

- The EM variants mainly aim at:
  1. increasing the convergence speed of EM and addressing the optimization problem in the M-step
  2. computing the E-step when it is intractable.
- In the first case, one can speak about deterministic algorithms :
  - e.g., Incremental EM (IEM)
  - Gradient EM
  - Generalized EM (GEM) algorithm
  - Expectation Conditional Maximization (ECM)
  - Expectation Conditional Maximization Either (ECME)
- In the second case, one can speak about stochastic algorithms:
  - e.g., Monte Carlo EM (MCEM)
  - Stochastic EM (SEM)
  - Simulated Annealing EM (SAEM)

Classification EM (CEM) algorithm

- we saw that EM computes the maximum likelihood (ML) estimate of a mixture model.
- The Classification EM (CEM) algorithm Celeux and Govaert (1992) estimates both the mixture model parameters and the classes' labels by maximizing the completed-data log-likelihood $\mathcal{L}_c(\boldsymbol{\Psi}; \mathbf{X}, \mathbf{z}) = \log p(\mathbf{X}, \mathbf{z}; \boldsymbol{\Psi})$
- start with an initial parameter $\boldsymbol{\Psi}^{(0)}$

1. **Step 1:** Compute the missing data $\mathbf{z}^{(q+1)}$ given the observations and the current estimated model parameters $\boldsymbol{\Psi}^{(q)}$:

$$\mathbf{z}^{(q+1)} = \arg\max_{\mathbf{z} \in \mathcal{Z}^n} \mathcal{L}_c(\boldsymbol{\Psi}^{(q)}; \mathbf{X}, \mathbf{z})$$

2. **Step 2:** Compute the model parameters update $\boldsymbol{\Psi}^{(q+1)}$ by maximizing the complete-data log-likelihood given the current estimation of the missing data $\mathbf{z}^{(q+1)}$:

$$\boldsymbol{\Psi}^{(q+1)} = \arg\max_{\boldsymbol{\Psi} \in \boldsymbol{\Omega}} \mathcal{L}_c(\boldsymbol{\Psi}; \mathbf{X}, \mathbf{z}^{(q+1)}).$$

CEM for GMMs

- the CEM algorithm, for the case of mixture models, is equivalent to integrating a classification step (C-step) between the E- and the M- steps of the EM algorithm.
- The C-step assigns the observations to the component densities by using the MAP rule:

1. **E-step:** Compute the conditional posterior probabilities $\tau_{ik}^{(q)}$ that the observation $\mathbf{x}_i$ arises from the kth component density.
2. **C-step:** Assign each observation $\mathbf{x}_i$ to the component maximizing the conditional posterior probability τ_{ik}:

$$z_i^{(q+1)} = \arg\max_{k \in \mathcal{Z}} \tau_{ik}^{(q)} \quad (i = 1, \ldots, n).$$

   $\Rightarrow$ this step provides a hard partition of the data
3. **M-step:** Update the mixture model parameters by maximizing the completed-data log-likelihood for the partition provided by the C-step.

Algorithm 2 Pseudo code of the CEM algorithm for GMMs.

Inputs: a data set $\mathbf{X}$ and the number of clusters K

fix a threshold $\epsilon > 0$; set $q \leftarrow 0$ (iteration)

Initialize: $\boldsymbol{\Psi}^{(0)} = (\pi_1^{(0)},\ldots,\pi_K^{(0)},\boldsymbol{\Psi}_1^{(0)},\ldots,\boldsymbol{\Psi}_K^{(0)})$ with $\boldsymbol{\Psi}_k^{(0)} = (\boldsymbol{\mu}_K^{(0)},\boldsymbol{\Sigma}_K^{(0)})$

while increment in the complete-data log-likelihood $> \epsilon$ **do**

E-step:

for $k = 1,\ldots,K$ **do**

Compute $\tau_{ik}^{(q)} == \frac{\pi_k \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_k^{(q)},\boldsymbol{\Sigma}_k^{(q)})}{\sum_{\ell=1}^{K}\pi_\ell \mathcal{N}(\mathbf{x}_i;\boldsymbol{\mu}_\ell^{(q)},\boldsymbol{\Sigma}_\ell^{(q)})}$

end for

C-step:

for $k = 1,\ldots,K$ **do**

Compute $z_i^{(q)} = \arg\max_{k\in\mathcal{Z}} \tau_{ik}^{(q)}$ for $i = 1,\ldots,n$

Set $z_{ik}^{(q)} = 1$ if $z_i^{(q)} = k$ and $z_{ik}^{(q)} = 0$ otherwise, for $i = 1,\ldots,n$

end for

M-step:

for $k = 1,\ldots,K$ **do**

Compute $\pi_k^{(q+1)} = \frac{\sum_{i=1}^{n} z_{ik}^{(q)}}{n}$

Compute $\boldsymbol{\mu}_k^{(q+1)} = \frac{1}{n_k^{(q)}}\sum_{i=1}^{n} z_{ik}^{(q)}\mathbf{x}_i$

Compute $\boldsymbol{\Sigma}_k^{(q+1)} = \frac{1}{n_k^{(q)}}\sum_{i=1}^{n} z_{ik}^{(q)}(\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})(\mathbf{x}_i - \boldsymbol{\mu}^{(q+1)})^T$

end for

$q \leftarrow q+1$

end while

CEM algorithm

- CEM is easy to implement, typically faster to converge than EM and monotonically improves the complete-data log-likelihood as the learning proceeds.
- converges toward a local maximum of the complete-data log-likelihood
- ! CEM provides biased estimates of the mixture model parameters. Indeed, CEM updates the model parameters from a truncated sample contrary to EM for which the model parameters are updated from the whole data through the fuzzy posterior probabilities and therefore the parameter estimations provided by EM are more accurate.
- **link with K-means**:
    - It can be shown that CEM which is formulated in a probabilistic framework, generalizes K-means
    - From a probabilistic point of view, K-means is equivalent to a particular case of the CEM algorithm for a mixture of K Gaussian densities with the same proportions $\pi_k = \frac{1}{K}\ \forall k$ and identical isotropic covariance matrices $\mathbf{\Sigma}_k = \sigma^2 \mathbf{I}\ \forall k$.

Parsimonious Gaussian mixture models

Parsimonious Gaussian mixtures

- Parsimonious Gaussian mixture models[1] are statistical models that allow for capturing a specific cluster shapes (e.g., clusters having the same shape or different shapes, spherical or elliptical clusters, etc).
- Eigenvalue decomposition of the cluster covariance matrices:

$$\mathbf{\Sigma}_k = \lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$$

where

- λ_k represents the volume of the kth cluster (the amount of space of the cluster).
- $\mathbf{D}_k$ is a matrix with columns corresponding to the eigenvectors of $\mathbf{\Sigma}_k$ that determines the orientation of the cluster.
- $\mathbf{A}_k$ is a diagonal matrix, whose diagonal entries are the normalized eigenvalues of $\mathbf{\Sigma}_k$ arranged in a decreasing order and its determinant is 1. This matrix is associated with the shape of the cluster.

[1] Banfield and Raftery (1993b); Celeux and Govaert (1995)

Parsimonious Gaussian mixtures

- This eigenvalue decomposition provides three main families of models: the spherical family, the diagonal family, and the general family

  and produces 14 different models, according to the choice of the configuration for the parameters λ_k, $\mathbf{A}_k$, and $\mathbf{D}_k$

Decomposition	Model-Type	Prior	Applied to
$\lambda\mathbf{I}$	Spherical	$\mathscr{IG}$	λ
$\lambda_k\mathbf{I}$	Spherical	$\mathscr{IG}$	λ_k
$\lambda\mathbf{A}$	Diagonal	$\mathscr{IG}$	$\mathrm{diag}(\lambda\mathbf{A})$
$\lambda_k\mathbf{A}$	Diagonal	$\mathscr{IG}$	$\mathrm{diag}(\lambda_k\mathbf{A})$
$\lambda\mathbf{DAD}^T$	General	$\mathscr{IW}$	$\boldsymbol{\Sigma} = \lambda\mathbf{DAD}^T$
$\lambda_k\mathbf{DAD}^T$	General	$\mathscr{IG}$ and $\mathscr{IW}$	λ_k and $\boldsymbol{\Sigma} = \mathbf{DAD}^T$
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	General	$\mathscr{IW}$	$\boldsymbol{\Sigma}_k = \lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$

Parsimonious GMMs

- In addition to providing flexible statistical models for the clusters, parsimonious Gaussian mixture can be viewed as techniques for reducing the number of parameters in the model.
- imposing constraints on the covariance matrices reduces the dimension of the optimization problem.
- The EM algorithms therefore provide more accurate estimations compared to the full mixture model.

Model selection

- The problem of choosing the number of clusters can be seen as a model selection problem.
- The model selection task consists of choosing a suitable compromise between flexibility so that a reasonable fit to the available data is obtained, and over-fitting.
- A common way is to use a criterion (score function) that ensure the compromise.
- In general, we choose an overall score function that is explicitly composed of two components: a component that measures the goodness of fit of the model to the data, and a penalty component that governs the model complexity:

$$\text{score}(\text{model}) = \text{error}(\text{model}) + \text{penalty}(\text{model})$$

which will be minimized.

Model selection

- The complexity of a model $\mathscr{M}$ is related to the number of its (free) parameters ν, the penalty function then involves the number of model parameters.
- Let $\mathscr{M}$ denote a model, $\mathscr{L}(\boldsymbol{\theta})$ its log-likelihood and ν the number of its free parameters. Consider that we fitted M different model structures $(\mathscr{M}_1,\ldots,\mathscr{M}_M)$, from which we wish to choose the "best" one (ideally the one providing the best prediction on future data).
- Assume we have estimated the model parameters $\widehat{\boldsymbol{\theta}}_m$ for each model structure $\mathscr{M}_m$ $(m=1,\ldots,M)$ from a sample of n observations $\mathbf{X}=(\mathbf{x}_1,\ldots,\mathbf{x}_n)$ and now we wish to choose among these fitted models.

Model selection

- Akaike Information Criterion (AIC):

$$\mathrm{AIC}(\mathcal{M}_m) = \mathcal{L}(\widehat{\boldsymbol{\theta}}_m) - \nu_m$$

- Bayesian Information Criterion (BIC):

$$\mathrm{BIC}(\mathcal{M}_m) = \mathcal{L}(\widehat{\boldsymbol{\theta}}_m) - \frac{\nu_m \log(n)}{2}$$

- Integrated Classification Likelihood (ICL):

$$\mathrm{ICL}(\mathcal{M}_m) = \mathcal{L}_c(\widehat{\boldsymbol{\theta}}_m) - \frac{\nu_m \log(n)}{2}$$

where $\mathcal{L}_c(\widehat{\boldsymbol{\theta}}_m)$ is the complete-data log-likelihood for the model $\mathcal{M}_m$ and ν_m denotes the number of free model parameters. For example, in the case of a d-dimensional Gaussian mixture model we have:

$$\nu = \underbrace{(K-1)}_{\pi_k\text{'s}} + \underbrace{K \times d}_{\{\boldsymbol{\mu}_k\}} + \underbrace{K \times \frac{d \times (d+1)}{2}}_{\{\boldsymbol{\Sigma}_k\}} = \frac{K \times (d+1) \times (d+2)}{2} - 1.$$

Examples

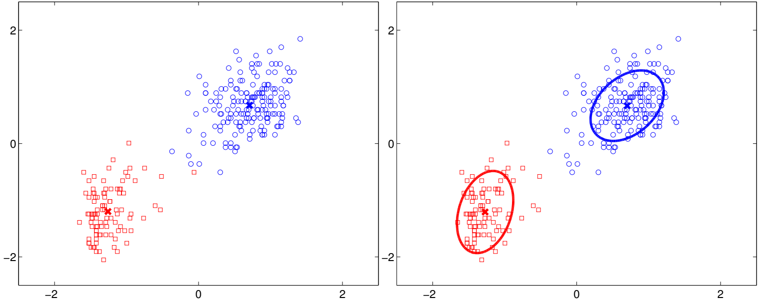


Figure : Clustering results obtained with K-means algorithm (left) with $K = 2$ and the EM algorithm (right). The cluster centers are shown by the red and blue crosses and the ellipses are the contours of the Gaussian component densities at level 0.4 estimated by EM. The number of clusters for EM have been chosen by BIC for $K = 1, \dots, 4$.

Examples

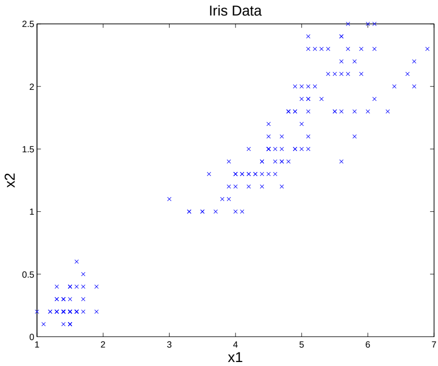


Figure : A three-class example of a real data set: Iris data of Fisher.

Examples

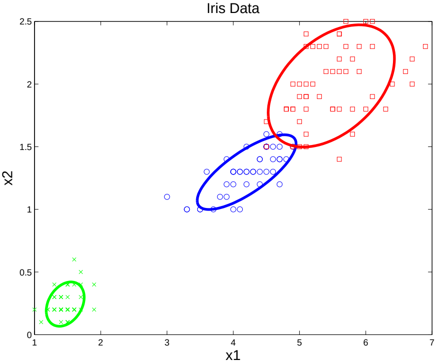


Figure : Iris data: Clustering results with EM for a GMM and AIC.

Examples

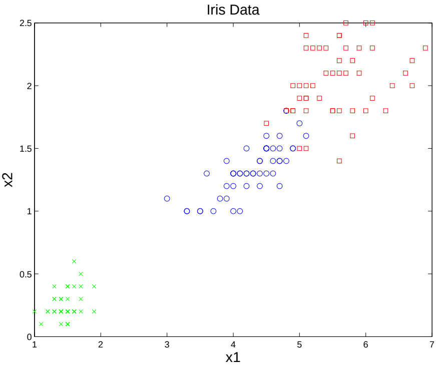
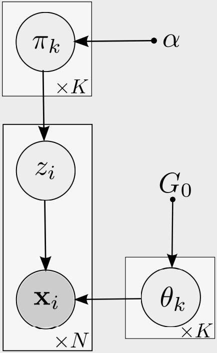


Figure : Iris data of Fisher: The data are colored according to the true partition.

Bayesian regularization of mixtures and Model-Based Clustering

The Bayesian finite mixture model

Graphical model representation of the Bayesian finite mixture model

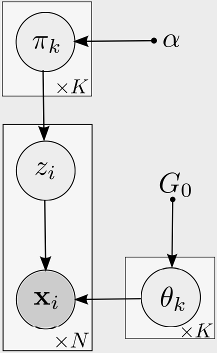


Generative model

$$\pi_1,\ldots,\pi_K|\boldsymbol{\alpha} \sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K)$$

The Bayesian finite mixture model

Graphical model representation of the Bayesian finite mixture model

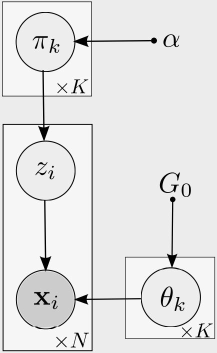


Generative model

$$\begin{aligned} \pi_1,\ldots,\pi_K|\boldsymbol{\alpha} &\sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K) \\ z_i|\boldsymbol{\pi} &\sim \mathscr{M}ult(.|\boldsymbol{\pi}) \end{aligned}$$

The Bayesian finite mixture model

Graphical model representation of the Bayesian finite mixture model

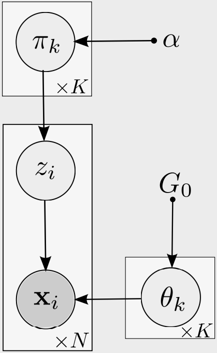


Generative model

$$
\begin{aligned}
\pi_1,...,\pi_K|\boldsymbol{\alpha} &\sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K) \\
z_i|\boldsymbol{\pi} &\sim \mathscr{M}ult(.|\boldsymbol{\pi}) \\
\boldsymbol{\theta}_{z_i}|G_0 &\sim \mathrm{G}(.|G_0)
\end{aligned}
$$

The Bayesian finite mixture model

Graphical model representation of the Bayesian finite mixture model



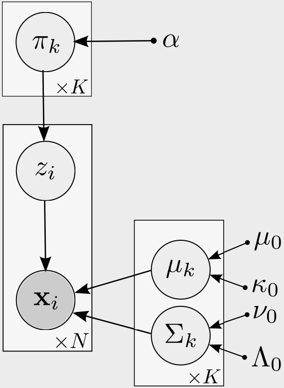
Generative model

$$
\begin{aligned}
\pi_1,\ldots,\pi_K|\boldsymbol{\alpha} &\sim \mathcal{D}ir\,|(\alpha_1,\ldots,\alpha_K)\\
z_i|\boldsymbol{\pi} &\sim \mathcal{M}ult(.|\boldsymbol{\pi})\\
\boldsymbol{\theta}_{z_i}|G_0 &\sim \mathrm{G}(.|G_0)\\
\mathbf{x}_i|z_i,\boldsymbol{\theta}_{z_i} &\sim f(.|\boldsymbol{\theta}_{z_i})
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\theta}_{z_i} &= (\boldsymbol{\mu}_{z_i},\boldsymbol{\Sigma}_{z_i})\\
\mathrm{G} &: \text{prior distribution}\\
\mathrm{G}_0 &: \text{hyperparameters}
\end{aligned}
$$

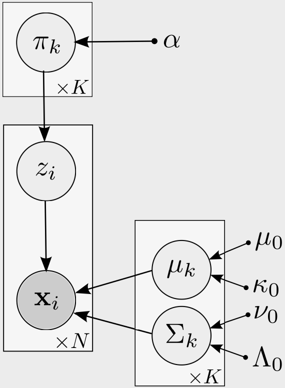
The Bayesian finite GMM

Graphical model representation of the Bayesian finite GMM



The Bayesian finite GMM

Graphical model representation of the Bayesian finite GMM

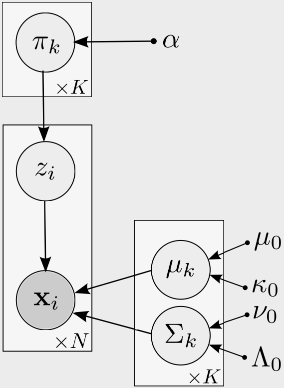


Generative model

$$\pi_1,...,\pi_K|\boldsymbol{\alpha} \sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K)$$

The Bayesian finite GMM

Graphical model representation of the Bayesian finite GMM

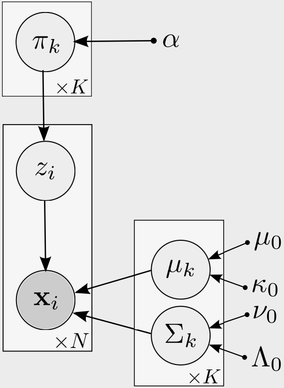


Generative model

$$\begin{aligned} \pi_1,...,\pi_K|\boldsymbol{\alpha} &\sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K) \\ z_i|\boldsymbol{\pi} &\sim \mathscr{M}ult(.|\boldsymbol{\pi}) \end{aligned}$$

The Bayesian finite GMM

Graphical model representation of the Bayesian finite GMM

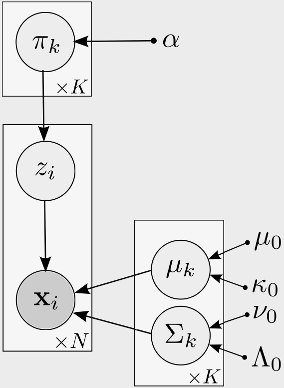


Generative model

$$\begin{aligned}\pi_1,\ldots,\pi_K|\boldsymbol{\alpha} &\sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K)\\ z_i|\boldsymbol{\pi} &\sim \mathscr{M}ult(.|\boldsymbol{\pi})\\ \boldsymbol{\theta}_{z_i}|\mathrm{G}_0 &\sim \mathrm{G}(.|G_0)\end{aligned}$$

The Bayesian finite GMM

Graphical model representation of the Bayesian finite GMM



Generative model

$$
\begin{aligned}
\pi_1,\ldots,\pi_K|\boldsymbol{\alpha} &\sim \mathscr{D}ir\,|(\alpha_1,\ldots,\alpha_K)\\
z_i|\boldsymbol{\pi} &\sim \mathscr{M}ult(.|\boldsymbol{\pi})\\
\boldsymbol{\theta}_{z_i}|\mathrm{G}_0 &\sim \mathrm{G}(.|G_0)\\
\mathbf{x}_i|z_i,\boldsymbol{\theta}_{z_i} &\sim f(.|\boldsymbol{\theta}_{z_i})
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\theta}_k &= (\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\\
\boldsymbol{\mu}_k &\sim \mathscr{N}(\boldsymbol{\mu}_0,\mathbf{V}_0)\\
\boldsymbol{\Sigma}_k &\sim \mathscr{IW}(\mathbf{S}_0,\nu_0)
\end{aligned}
$$

The Gibbs sampler for the Bayesian finite GMM

Bayesian sampling

$$\begin{aligned} z_i|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\theta} &\sim \mathscr{M}ult(.|\tau_{i1},...,\tau_{iK}) \\ \tau_{ik} &= p(z_i = k|\mathbf{x}_i,\boldsymbol{\pi},\boldsymbol{\theta} =)\frac{\pi_k\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K}\pi_l\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_l} \end{aligned} \quad (1)$$

The Gibbs sampler for the Bayesian finite GMM

Bayesian sampling

$$
\begin{aligned}
z_i|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\theta} &\sim \mathcal{M}ult(.|\tau_{i1},...,\tau_{iK}) && (1)\\
\tau_{ik} &= p(z_i = k|\mathbf{x}_i,\boldsymbol{\pi},\boldsymbol{\theta} =)\frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K}\pi_l \mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_l} \\
\pi_1,...,\pi_K|\mathbf{z} &\sim \mathcal{D}ir(.|\alpha_1+n_1,\ldots,\alpha_K+n_K) && (2)\\
n_k &= \sum_{i=1}^{n} z_{ik}
\end{aligned}
$$

The Gibbs sampler for the Bayesian finite GMM

Bayesian sampling

$$
\begin{aligned}
z_i|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\theta} &\sim \mathcal{M}ult(.|\tau_{i1},...,\tau_{iK}) &(1)\\
\tau_{ik} &= p(z_i = k|\mathbf{x}_i,\boldsymbol{\pi},\boldsymbol{\theta} =)\frac{\pi_k\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K}\pi_l\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_l} \\
\pi_1,...,\pi_K|\mathbf{z} &\sim \mathcal{D}ir(.|\alpha_1+n_1,\ldots,\alpha_K+n_K) &(2)\\
n_k &= \sum_{i=1}^{n} z_{ik} \\
\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k,\mathbf{z},\mathbf{X} &\sim \mathcal{N}(.|\mathbf{m}_k,\mathbf{V}_k) &(3)\\
\mathbf{V}_k^{-1} &= \mathbf{V}_0^{-1} + n_k\boldsymbol{\Sigma}_k^{-1} \\
\mathbf{m}_k &= \mathbf{V}_k(\boldsymbol{\Sigma}_k^{-1}\sum_{i=1}^{n} z_{ik}\mathbf{x}_i + \mathbf{V}_0^{-1}\mathbf{m}_0)
\end{aligned}
$$

The Gibbs sampler for the Bayesian finite GMM

Bayesian sampling

$$
\begin{aligned}
z_i|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\theta} &\sim \mathscr{M}ult(.|\tau_{i1},...,\tau_{iK}) & (1)\\
\tau_{ik} &= p(z_i = k|\mathbf{x}_i,\boldsymbol{\pi},\boldsymbol{\theta} =)\frac{\pi_k\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K}\pi_l\mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_l} \\
\pi_1,...,\pi_K|\mathbf{z} &\sim \mathscr{D}ir(.|\alpha_1+n_1,\ldots,\alpha_K+n_K) & (2)\\
n_k &= \sum_{i=1}^{n} z_{ik} \\
\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k,\mathbf{z},\mathbf{X} &\sim \mathcal{N}(.|\mathbf{m}_k,\mathbf{V}_k) & (3)\\
\mathbf{V}_k^{-1} &= \mathbf{V}_0^{-1} + n_k\boldsymbol{\Sigma}_k^{-1} \\
\mathbf{m}_k &= \mathbf{V}_k(\boldsymbol{\Sigma}_k^{-1}\sum_{i=1}^{n} z_{ik}\mathbf{x}_i + \mathbf{V}_0^{-1}\mathbf{m}_0) \\
\boldsymbol{\Sigma}_k|\boldsymbol{\mu}_k,\mathbf{z},bx &\sim \mathscr{IW}(\mathbf{S}_0 + \sum_{i=1}^{n} z_{ik}(\mathbf{x}_i-\boldsymbol{\mu}_k)(\mathbf{x}_i-\boldsymbol{\mu}_k)^T,\nu_0+n_k) & (4)
\end{aligned}
$$

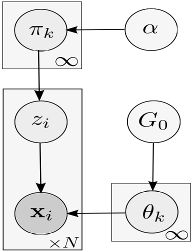
Infinite Gaussian Mixture Model and Dirichlet Process Mixtures

- Infinite GMM: $p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_k \; \mathcal{N}_k(\mathbf{x}_i|\theta_k)$ Rasmussen (2000)
- Parameters: $\boldsymbol{\theta} = \{\pi_k, \; \boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \; \boldsymbol{\Sigma}_k)\}_{k=1}^{\infty}$
- Prior: add a distribution over the parameters distribution: a Dirichlet Process Antoniak (1974)
- Generative model:

$$\begin{aligned} G|\alpha, G_0 &\sim \mathrm{DP}(\alpha, G_0) \\ \boldsymbol{\theta}_i|G &\sim G \\ \mathbf{x}_i|\boldsymbol{\theta}_i &\sim p(.|\boldsymbol{\theta}_i) \end{aligned}$$

equivalent to

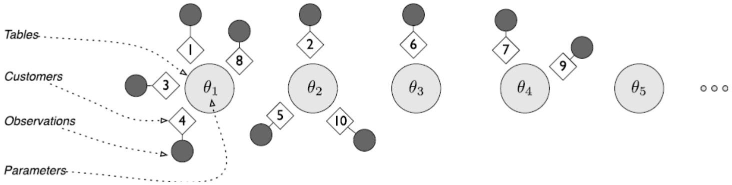
$$\begin{aligned} z_i|\alpha &\sim \mathrm{CRP}(\mathbf{z}_{\backslash i}; \alpha) \\ \boldsymbol{\theta}_{z_i}|G_0 &\sim G_0 \\ \mathbf{x}_i|\boldsymbol{\theta}_{z_i} &\sim p(.|\boldsymbol{\theta}_{z_i}) \end{aligned}$$



Chinese Restaurant Process (CRP)

Imagine a Restaurant with an infinite number of tables and in which customers are entering and sitting at these tables.

1. The first customer sits at table 1
2. The second customer may sit at table with probability $\frac{1}{1+\alpha}$ or chose another table with probability $\frac{\alpha}{1+\alpha}$
3. ...
4. ith customer sits at table k with probability proportional to the number of already seated customers n_k and may choose a new table with a probability proportional to a small positive real number α



Chinese Restaurant Process (CRP)

Chinese Restaurant Process (CRP) Wood and Black (2008); Samuel and Blei (2012):

- The CRP provides a distribution on the infinite partitions of the data: $p(\mathbf{z}) = p(z_1)p(z_2|z_1)\ldots p(z_n|z_{n-1})$:

$$\begin{aligned} p(z_i = k|z_1,...,z_{i-1}) &= \text{CRP}(z_1,...,z_{i-1};\alpha) \\ &= \begin{cases} \frac{n_k}{i-1+\alpha} & \text{if} \quad k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{if} \quad k > K_+ \end{cases} \end{aligned}$$

- α represents the CRP concentration parameter
- K_+ : nbr. of tables for which the nbr. of customers sitting in is $n_k > 0$ (active clusters)
- $k \leq K_+$ means that k is a previously occupied table and $k > K_+$ means k is a new table to be occupied.

Gibbs sampler for the Infinite Parsimonious GMM

Algorithm 3 Gibbs sampler for the Infinite Parsimonious GMM

Entrees: Data $\mathbf{x}_i$, nbr of Gibbs samples n_s.

Initialization: $q \leftarrow 0$; hyper-parameters $G_0^{(q)}$, $\boldsymbol{\alpha}$; nbr of clusters $K_+ = 1$.

for $i = 1, \ldots, n$ **do**

  sample the cluster labels $z_i^{(t)} \sim p(\mathbf{x}_i | z_i, \boldsymbol{\theta}_k)\ \mathrm{CRP}(\{z_1, \ldots, z_n\}_{\backslash z_i}; \boldsymbol{\alpha}^{(t)})$

  if $z_i^{(t)} = K_+ + 1$ create a new cluster $K_+ = K_+ + 1$, sample $\boldsymbol{\theta}_{z_i}^{(t)}$ from the prior

end for

for $k = 1, \ldots, K_+$ **do**

  sample$\boldsymbol{\theta}_k^{(t)}$ from the posterior

end for

sample $\boldsymbol{\alpha}^{(t)}$

$\mathbf{z}^{(t+1)} \leftarrow \mathbf{z}^{(t)}$

$\boldsymbol{\alpha}^{(t+1)} \leftarrow \boldsymbol{\alpha}^{(t)}$

$t \leftarrow t + 1$

Outputs: $\{\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{z}}, \widehat{K} = K_+\}$

Seven included models in the non-parametric approach:

Decomposition	Model-Type	Prior	Applied to
$\lambda\mathbf{I}$	Spherical	$\mathscr{IG}$	λ
$\lambda_k\mathbf{I}$	Spherical	$\mathscr{IG}$	λ_k
$\lambda\mathbf{A}$	Diagonal	$\mathscr{IG}$	$\mathrm{diag}(\lambda\mathbf{A})$
$\lambda_k\mathbf{A}$	Diagonal	$\mathscr{IG}$	$\mathrm{diag}(\lambda_k\mathbf{A})$
$\lambda\mathbf{DAD}^T$	General	$\mathscr{IW}$	$\mathbf{\Sigma} = \lambda\mathbf{DAD}^T$
$\lambda_k\mathbf{DAD}^T$	General	$\mathscr{IG}$ and $\mathscr{IW}$	λ_k and $\mathbf{\Sigma} = \mathbf{DAD}^T$
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	General	$\mathscr{IW}$	$\mathbf{\Sigma}_k = \lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$

Bayesian learning

Gibbs sampler: e.g for $\lambda_k\mathbf{DAD}^T$: Normal Inverse-Wishart (prior/posterior)

$$\boldsymbol{\mu}_k|. \sim \mathscr{N}(\mu_0, \lambda_k\mathbf{\Sigma}_0/\kappa_0) \;\dots\; \mathbf{\Sigma}_0|. \sim \mathscr{IW}(\nu_0, \Lambda_0) \;\dots \lambda_k|. \sim \mathscr{IG}(r_0/2, s_0/2)$$

$$\boldsymbol{\mu}_k|\mathbf{X},. \sim \mathscr{N}\left(\frac{n_k\overline{\mathbf{x}}_k + \kappa_0\mu_0}{n_k+\kappa_0}, \frac{\lambda_k\mathbf{\Sigma}_0}{n_k+\kappa_0}\right)$$

$$\mathbf{\Sigma}_0|\mathbf{X},. \sim \mathscr{IW}\left(\nu_0 + n, \Lambda_0 + \sum_{k=1}^{K}\left\{\frac{W_k}{\lambda_k} + \frac{n_k\kappa_0}{\lambda_k(n_k+\kappa_0)}(\overline{\mathbf{x}}_k - \mu_0)(\overline{\mathbf{x}}_k - \mu_0)^T\right\}\right)$$

$$\lambda_k|\mathbf{X},. \sim \mathscr{IG}\left(\frac{r_0 + n_k d}{2}, \frac{1}{2}\left\{s_0 + \mathrm{tr}(W_k\mathbf{\Sigma}_0^{-1}) + \frac{n_k\kappa_0}{n_k+\kappa_0}(\overline{\mathbf{x}}_k - \boldsymbol{\mu}_0)^T\mathbf{\Sigma}_0^{-1}(\overline{\mathbf{x}}_k - \boldsymbol{\mu}_0)\right\}\right)$$

Markovian Extension

The infinite GMMs has been extended to the infinite HMM for sequential data modeling: This is the Hierarchical Dirichlet Process for Hidden Markov Model (HDP-HMM)

Two main inference approaches approaches: the Gibbs sampling [2] [3] and the Beam sampling [4].

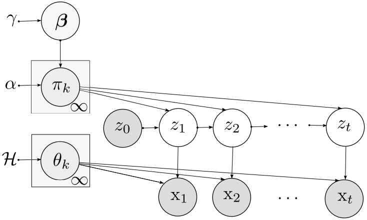


Figure : Infinite Hidden Markov Model (IHMM) graphical representation

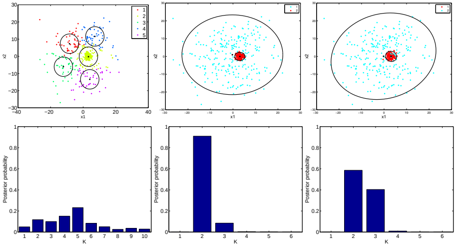
2 Fox et al. (2008)

3 Teh et al. (2006)

4 Van Gael et al. (2008)

Experiment on simulated data

1. A two-clusters data set
2. $n = 500$ observation in $\mathbb{R}^2$
3. $\boldsymbol{\pi} = [.5\ .5]$; $\boldsymbol{\mu}_1 = [0\ 0]^T$; $\boldsymbol{\mu}_2 = [3\ 0]^T$; $\boldsymbol{\Sigma}_1 = 100 * \mathbf{I}_2; \boldsymbol{\Sigma}_2 = \mathbf{I}_2$



$\lambda \mathbf{I}$ $\lambda_k \mathbf{I}$ $\lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$

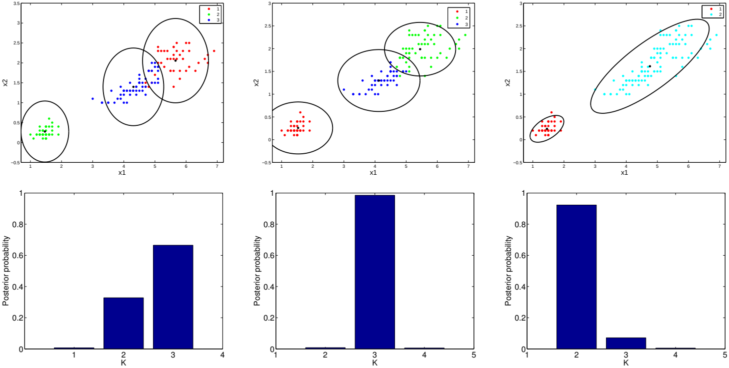
Experiments on benchmarks

Dataset	n	d	True K
Iris	150	4	3
Old Faithful Geyser	272	2	?
Trees	31	3	?
Wine	178	13	3
Diabetes	145	3	3

Model	Iris	Geyser	Trees	Wine	Diabetes
$\lambda\mathbf{I}$	4	2	1	1	3
$\lambda_k\mathbf{I}$	3	2	1	2	5
$\lambda\mathbf{A}$	3	3	2	3	3
$\lambda_k\mathbf{A}$	3	2	2	1	5
$\lambda\mathbf{DAD}^T$	4	2	2	3	5
$\lambda_k\mathbf{DAD}^T$	2	2	2	3	3
$\lambda_k\mathbf{D}_k\mathbf{A}_k\mathbf{D}_k^T$	2	2	2	3	3

Table : Estimated K by the infinite parsimonious GMM

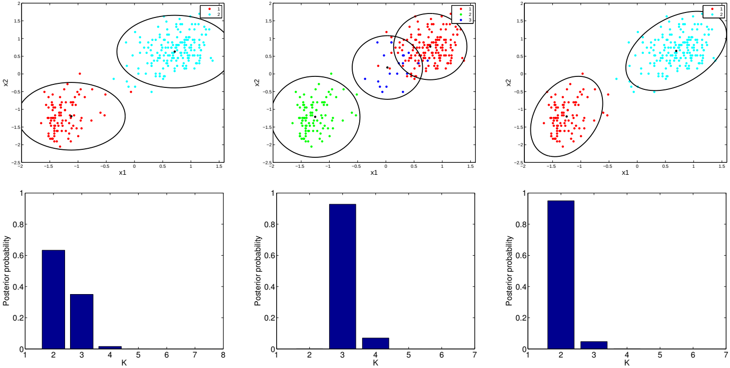
Experiments on Benchmarks



$\lambda_k \mathbf{I}$ $\lambda_k \mathbf{A}$ $\lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$

Figure : Graphical results for Iris data

Experiments on Benchmarks



$\lambda_k \mathbf{I}$ $\lambda \mathbf{A}$ $\lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$

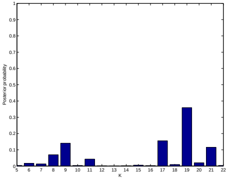
Figure : Graphical results for Old Faithful Geyser data

Whale song decomposition

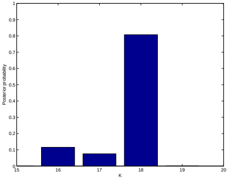
- In this experiment, we apply the proposed approach to a challenging problem of humpback whale song decomposition.
- The analysis is unsupervised and aims at discovering the call units (which can be considered as a kind of whale vocabulary),
- this can be seen as a problem of unsupervised call units classification as in[5].
- We therefore reformulate the problem of whale song decomposition as a clustering problem.
- We apply our proposed IPGMM to find a partition of the whale song into clusters, and automatically infer the number of clusters from the data.
- We then extend the Infinite GMM to the Markovian case as in
- The data consist of MFCC parameters of 8.6 minutes of a Humpback whale song recordings

[5] Pace et al. (2010)

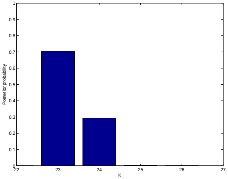
whale song decomposition problem



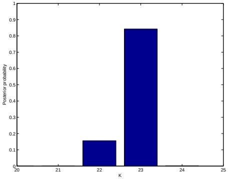
$\lambda \mathbf{I}$



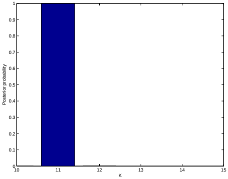
$\lambda \mathbf{A}$



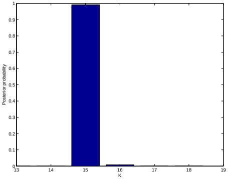
$\lambda_k \mathbf{D}\mathbf{A}\mathbf{D}^T$



$\lambda_k \mathbf{I}$

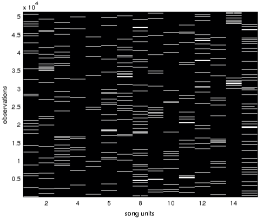


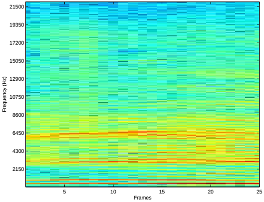
$\lambda_k \mathbf{A}$



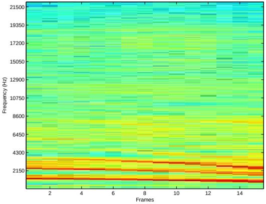
$\lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$

whale song decomposition problem

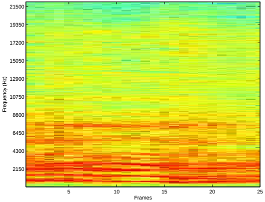




song unit 1



song unit 9



song unit 14

State sequence obtained by using a HDP-HMM [6].

"sparse" approach: the number of estimated states equals $K = 6$

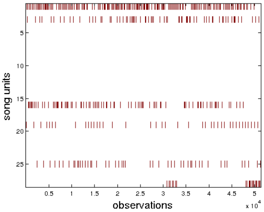
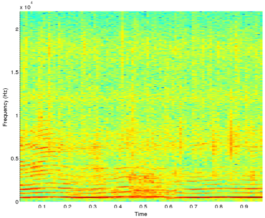
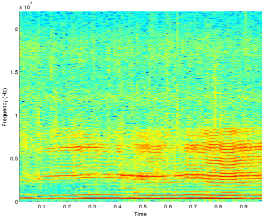


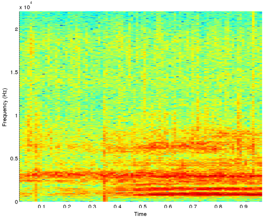
Figure : State sequence obtained by an Infinite HMM.



song unit 16



song unit 19



song unit 30

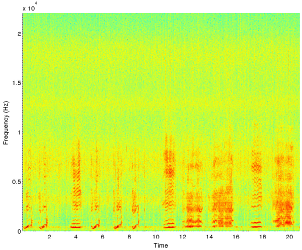


Figure : The state sequences (left) and the spectrogram of the ninth unit (right) obtained by the HDP-HMM (Beam sampling)

Conclusion and Perspectives

- Mixtures are very flexible for cluster analysis namely via Parsimonious mixture modeling
- The Dirichlet Process mixture approach is a Bayesian non-parametric alternative
- It avoids the problem of model selection encountered in maximum likelihood and Bayesian learning of parametric GMMs
-
- The parsimonious version allows to have several flexible models adapted for several clusters configurations
- Perspectives :
  - More comparisons between the different modes (e.g. using Bayes Factors)
  - More experiments for the Markovian extension for sequential data
  - Infinite block mixture
  - Other MCMC techniques

References I

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. The Annals of Statistics, 2(6):1152–1174, 1974.

J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. Biometrics, 49(3):803–821, 1993a.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based Gaussian and non-Gaussian clustering. Biometrics, 49(3):803–821, 1993b.

H. Bensmail and J. J. Meulman. Model-based clustering with noise: Bayesian inference and estimation. J. Classification, 20(1): 049–076, 2003.

H. Bensmail, G. Celeux, A. E. Raftery, and C. P. Robert. Inference in model-based cluster analysis. Statistics and Computing, 7 (1):1–10, 1997.

G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis, 14:315–332, 1992.

G. Celeux and G. Govaert. Gaussian parsimonious clustering models. Pattern Recognition, 28(5):781–793, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of The Royal Statistical Society, B, 39(1):1–38, 1977.

Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. An HDP-HMM for systems with state persistence. In ICML 2008: Proceedings of the 25th international conference on Machine learning, pages 312–319, New York, NY, USA, 2008. ACM.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association, 97:611–631, 2002.

C. Fraley and A. E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. Journal of Classification, (2):155–181, 2007.

G. J. McLachlan and K. E. Basford. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York, 1988.

G. J. McLachlan and T. Krishnan. The EM algorithm and extensions. New York: Wiley, 1997.

References II

G. J. McLachlan and D. Peel. Finite mixture models. New York: Wiley, 2000.

R. M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

D. Ormoneit and V. Tresp. Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates. IEEE Transactions on Neural Networks, 9(4):639–650, 1998.

Federica Pace, Frederic Benard, Herve Glotin, Olivier Adam, and Paul White. Subunit definition and analysis for humpback whale call classification. Applied Acoustics, 71(11):1107 – 1112, 2010.

C. Rasmussen. The infinite gaussian mixture model. Advances in neuronal Information Processing Systems, 10:554 – 560, 2000.

Sylvia Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components. Journal of the Royal Statistical Society, 59(4):731–792, 1997.

J. Gershman Samuel and David M. Blei. A tutorial on bayesian non-parametric model. Journal of Mathematical Psychology, 56: 1–12, 2012.

M. Stephens. Bayesian Methods for Mixtures of Normal Distributions. PhD thesis, University of Oxford, 1997.

M. Stephens. Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods. Annals of Statistics, 28(1):40–74, 2000.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. Journal of the American Statistical Association, 101(476):1566–1581, 2006.

J. Van Gael, Y. Saatci, Y.W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In Proceedings of the 25th international conference on Machine learning, pages 1088–1095. ACM New York, NY, USA, 2008.

F. Wood and M. J. Black. A nonparametric bayesian alternative to spike sorting. Journal of Neuroscience Methods, 173(1): 1–12, 2008.

Thank you!