

Deep Gaussian Mixture Models

Cinzia Viroli
(University of Bologna, Italy)

joint with Geoff McLachlan (University of Queensland, Australia)

JOCLAD 2018, Lisbona, April 5th, 2018

Outline

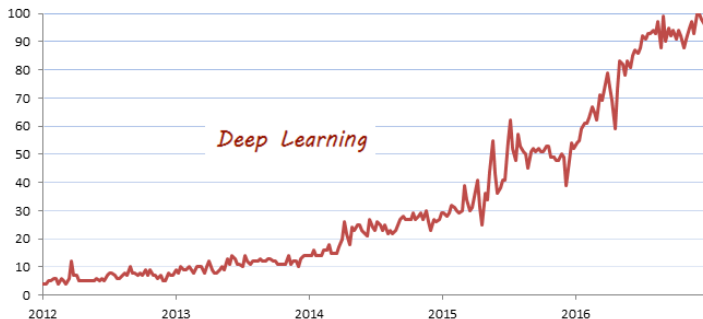
- Deep Learning
- Mixture Models
- Deep Gaussian Mixture Models

Deep Learning

Deep Learning

Deep Learning is a trendy topic in the machine learning community

Google Trend

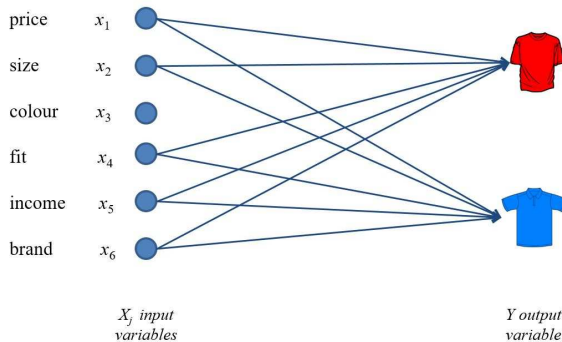


What is Deep Learning?

Deep Learning is a set of algorithms in machine learning able to gradually learning a huge number of parameters in an architecture composed by multiple non linear transformations (multi-layer structure)

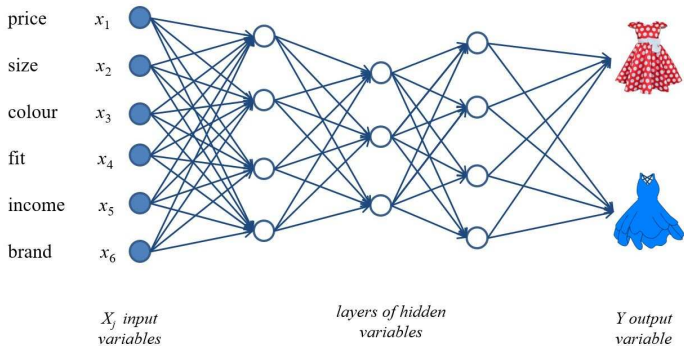
Example of Learning

Choice modeling



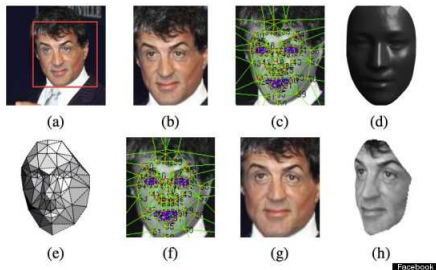
Example of Deep Learning

Deep(er) choice modeling



Facebook's DeepFace

DeepFace (Yaniv Taigman) is a deep learning facial recognition system that employs a nine-layer neural network with over 120 million connection weights.



It identifies human faces in digital images with an accuracy of 97.35%.

Mixture Models

Gaussian Mixture Models (GMM)

- In **model based clustering** data are assumed to come from a **finite mixture model** (McLachlan and Peel, 2000; Fraley and Raftery, 2002).

Gaussian Mixture Models (GMM)

- In **model based clustering** data are assumed to come from a **finite mixture model** (McLachlan and Peel, 2000; Fraley and Raftery, 2002).
- For quantitative data each mixture component is usually modeled as a multivariate Gaussian distribution:

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{j=1}^k \pi_j \phi^{(p)}(\mathbf{y}; \mu_j, \Sigma_j)$$

Gaussian Mixture Models (GMM)

- In **model based clustering** data are assumed to come from a **finite mixture model** (McLachlan and Peel, 2000; Fraley and Raftery, 2002).
- For quantitative data each mixture component is usually modeled as a multivariate Gaussian distribution:

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{j=1}^k \pi_j \phi^{(p)}(\mathbf{y}; \mu_j, \Sigma_j)$$

- Growing popularity, widely used.

Gaussian Mixture Models (GMM)

However, in the recent years, a lot of research has been done to address two issues:

- **High-dimensional data**: when the number of observed variables is large, it is well known that GMM represents an over-parameterized solution

Gaussian Mixture Models (GMM)

However, in the recent years, a lot of research has been done to address two issues:

- **High-dimensional data**: when the number of observed variables is large, it is well known that GMM represents an over-parameterized solution
- **Non-Gaussian data**: when data are not Gaussian, GMM could requires more components than true clusters thus requiring *merging* or alternative distributions.

High dimensional data

Some solutions (among the others):

Model based clustering

**Dimensionally reduced model
based clustering**

High dimensional data

Some solutions (among the others):

Model based clustering

- Banfield and Raftery (1993) and Celeux and Govaert (1995): proposed constrained GMM based on parameterization of the generic component-covariance matrix based on its spectral decomposition:

$$\Sigma_i = \lambda_i \mathbf{A}_i^\top \mathbf{D}_i \mathbf{A}_i$$

- Bouveyron et al. (2007): proposed a different parameterization of the generic component-covariance matrix

Dimensionally reduced model based clustering

High dimensional data

Some solutions (among the others):

Model based clustering

- Banfield and Raftery (1993) and Celeux and Govaert (1995): proposed constrained GMM based on parameterization of the generic component-covariance matrix based on its spectral decomposition:

$$\Sigma_i = \lambda_i \mathbf{A}_i^T \mathbf{D}_i \mathbf{A}_i$$

- Bouveyron et al. (2007): proposed a different parameterization of the generic component-covariance matrix

Dimensionally reduced model based clustering

- Ghahramani and Hilton (1997) and McLachlan et al. (2003): *Mixtures of Factor Analyzers (MFA)*
- Yoshida et al. (2004), Baek and McLachlan (2008), Montanari and Viroli (2010): *Factor Mixture Analysis (FMA) or Common MFA*
- McNicolas and Murphy (2008): eight parameterizations of the covariance matrices in MFA

Non-Gaussian data

Some solutions (among the others):

More components than clusters

Non-Gaussian distributions

Non-Gaussian data

Some solutions (among the others):

More components than clusters

Non-Gaussian distributions

- Merging mixture components (Hennig, 2010; Baudry et al., 2010; Melnykov, 2016)
- Mixtures of mixtures models (Li, 2005) and in the dimensional reduced space mixtures of MFA (Viroli, 2010)

Non-Gaussian data

Some solutions (among the others):

More components than clusters

- Merging mixture components (Hennig, 2010; Baudry et al., 2010; Melnykov, 2016)
- Mixtures of mixtures models (Li, 2005) and in the dimensional reduced space mixtures of MFA (Viroli, 2010)

Non-Gaussian distributions

- Mixtures of skew-normal, skew-t and canonical fundamental skew distributions (Lin, 2009; Lee and McLachlan, 2011-2017)
- Mixtures of generalized hyperbolic distributions (Subedi and McNicholas, 2014; Franczak et al., 2014)
- MFA with non-Normal distributions (McLachlan et al. 2007; Andrews and McNicholas, 2011; and many recent proposals by McNicholas, McLachlan and colleagues)

Deep Gaussian Mixture Models

Why Deep Mixtures?

A **Deep Gaussian Mixture Model** (DGMM) is a network of **multiple layers** of latent variables, where, at each layer, the variables follow a mixture of Gaussian distributions.

Gaussian Mixtures vs Deep Gaussian Mixtures

Given data \mathbf{y} , of dimension $n \times p$, the mixture model

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{j=1}^{k_1} \pi_j \phi^{(p)}(\mathbf{y}; \mu_j, \Sigma_j)$$

can be rewritten as a linear model with a certain prior probability:

$$\mathbf{y} = \mu_j + \Lambda_j \mathbf{z} + \mathbf{u} \quad \text{with probab } \pi_j$$

where

- $\mathbf{z} \sim N(0, I_p)$
- \mathbf{u} is an independent specific random errors with $\mathbf{u} \sim N(0, \Psi_j)$
- $\Sigma_j = \Lambda_j \Lambda_j^\top + \Psi_j$

Gaussian Mixtures vs Deep Gaussian Mixtures

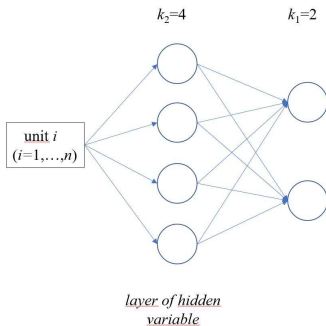
Now suppose we replace $\mathbf{z} \sim N(0, I_p)$ with

$$f(\mathbf{z}; \theta) = \sum_{j=1}^{k_2} \pi_j^{(2)} \phi^{(p)}(\mathbf{z}; \mu_j^{(2)}, \Sigma_j^{(2)})$$

This defines a **Deep Gaussian Mixture Model (DGMM)** with $h = 2$ layers.

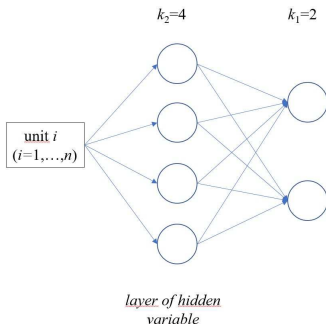
Deep Gaussian Mixtures

Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



Deep Gaussian Mixtures

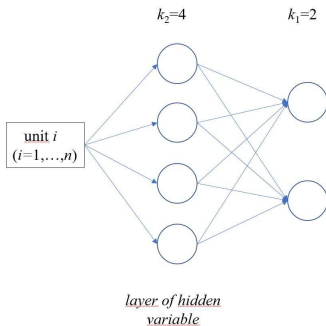
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



- $k = 8$ possible paths (total subcomponents)
- $M = 6$ real subcomponents (shared set of parameters)

Deep Gaussian Mixtures

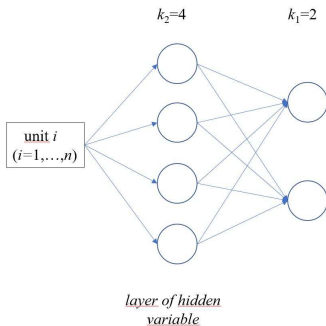
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



- $k = 8$ possible paths (total subcomponents)
- $M = 6$ real subcomponents (shared set of parameters)
- $M < k$ thanks to the tying

Deep Gaussian Mixtures

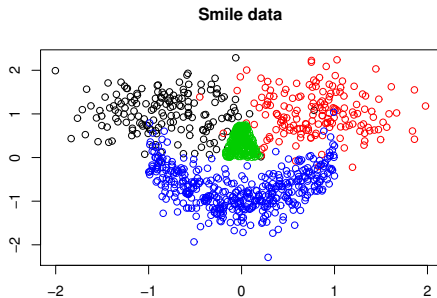
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



- $k = 8$ possible paths (total subcomponents)
- $M = 6$ real subcomponents (shared set of parameters)
- $M < k$ thanks to the tying
- Special mixtures of mixtures model (Li, 2005)

Do we really need DGMM?

Consider the $k = 4$ clustering problem

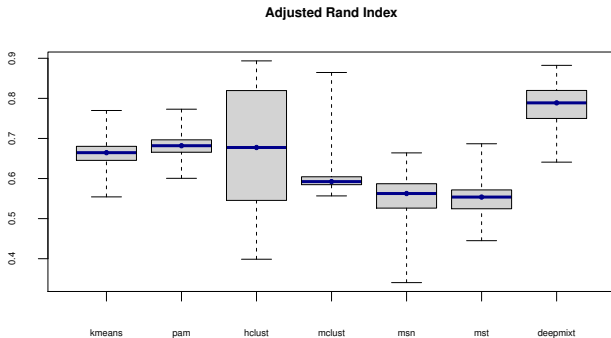


Do we really need DGMM?

- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)

Do we really need DGMM?

- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)



Do we really need DGMM?

- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)

Do we really need DGMM?

- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)
- In the DGMM we cluster data k_1 groups ($k_1 < k$) through $f(\mathbf{y}|\mathbf{z})$: the remaining components in the previous layer(s) act as density approximation of global non-Gaussian components

Do we really need DGMM?

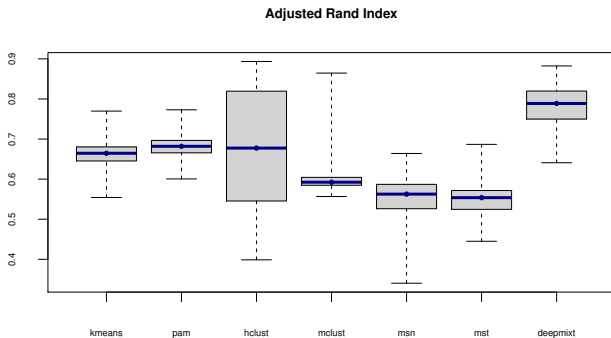
- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)
- In the DGMM we cluster data k_1 groups ($k_1 < k$) through $f(\mathbf{y}|\mathbf{z})$: the remaining components in the previous layer(s) act as density approximation of global non-Gaussian components
- Automatic tool for merging mixture components: merging is unit-dependent

Do we really need DGMM?

- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)
- In the DGMM we cluster data k_1 groups ($k_1 < k$) through $f(\mathbf{y}|\mathbf{z})$: the remaining components in the previous layer(s) act as density approximation of global non-Gaussian components
- Automatic tool for merging mixture components: merging is unit-dependent
- Thanks to its multilayered architecture, the deep mixture provides a way to estimate increasingly complex relationships as the number of layers increases.

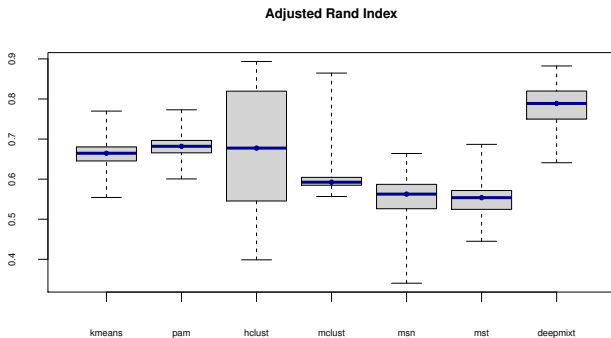
Do we really need DGMM?

Clustering on 100 generated datasets:



Do we really need DGMM?

Clustering on 100 generated datasets:



$n = 1000$, $p = 2$, # of param in DGMM $d = 50$

What about higher dimensional problems?

Dimensionally reduced DGMM

- Tang et al. (2012) proposed a deep mixture of factor analyzers with a stepwise greedy search algorithm: a separate and independent estimation for each layer (error propagation)
- A general strategy is presented and estimation is obtained in a unique procedure by a stochastic EM
- Fast for $h < 4$; computationally more demanding as h increases

Dimensionally reduced DGMM

Suppose h layers. Given \mathbf{y} , of dimension $n \times p$, at each layer a linear model describe the data with a certain prior probability as follows:

$$\begin{aligned}
 (1) \quad & \mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)} \text{ with prob. } \pi_{s_1}^{(1)}, s_1 = 1, \dots, k_1 \\
 (2) \quad & \mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)} \text{ with prob. } \pi_{s_2}^{(2)}, s_2 = 1, \dots, k_2 \\
 & \dots \\
 (h) \quad & \mathbf{z}_i^{(h-1)} = \eta_{s_h}^{(h)} + \Lambda_{s_h}^{(h)} \mathbf{z}_i^{(h)} + \mathbf{u}_i^{(h)} \text{ with prob. } \pi_{s_h}^{(h)}, t = 1, \dots, k_h
 \end{aligned}$$

where \mathbf{u} is independent on \mathbf{z} and layers that are sequentially described by latent variables with a progressively decreasing dimension, r_1, r_2, \dots, r_h , where $p > r_1 > r_2 > \dots, > r_h \geq 1$.

Let Ω be the set of all possible paths through the network. The generic path $s = (s_1, \dots, s_h)$ has a probability π_s of being sampled, with

$$\sum_{s \in \Omega} \pi_s = \sum_{s_1, \dots, s_h} \pi_{(s_1, \dots, s_h)} = 1.$$

Let Ω be the set of all possible paths through the network. The generic path $s = (s_1, \dots, s_h)$ has a probability π_s of being sampled, with

$$\sum_{s \in \Omega} \pi_s = \sum_{s_1, \dots, s_h} \pi_{(s_1, \dots, s_h)} = 1.$$

The DGMM can be written as $f(\mathbf{y}; \Theta) = \sum_{s \in \Omega} \pi_s \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$, where

$$\begin{aligned} \boldsymbol{\mu}_s &= \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)}(\eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)}(\dots(\eta_{s_{h-1}}^{(h-1)} + \Lambda_{s_{h-1}}^{(h-1)}\eta_h^{(h)}))) \\ &= \eta_{s_1}^{(1)} + \sum_{l=2}^h \left(\prod_{m=1}^{l-1} \Lambda_{s_m}^{(m)} \right) \eta_{s_l}^{(l)} \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{\Sigma}_s &= \Psi_{s_1}^{(1)} + \Lambda_{s_1}^{(1)}(\Lambda_{s_2}^{(2)}(\dots(\Lambda_{s_h}^{(h)}\Lambda_{s_h}^{(h)\top} + \Psi_{s_h}^{(h)}))\dots)\Lambda_{s_2}^{(2)\top})\Lambda_{s_1}^{(1)\top} \\ &= \Psi_{s_1}^{(1)} + \sum_{l=2}^h \left(\prod_{m=1}^{l-1} \Lambda_{s_m}^{(m)} \right) \Psi_{s_l}^{(l)} \left(\prod_{m=1}^{l-1} \Lambda_{s_m}^{(m)} \right)^\top \end{aligned}$$

By considering the data as the zero layer, $\mathbf{y} = \mathbf{z}^{(0)}$, in a DGMM all the marginal distributions of the latent variables $\mathbf{z}^{(l)}$ and their conditional distributions to the upper level of the network are distributed as Gaussian mixtures.

By considering the data as the zero layer, $\mathbf{y} = \mathbf{z}^{(0)}$, in a DGMM all the marginal distributions of the latent variables $\mathbf{z}^{(l)}$ and their conditional distributions to the upper level of the network are distributed as Gaussian mixtures.

Marginals:

$$f(\mathbf{z}^{(l)}; \Theta) = \sum_{\tilde{s}=(s_{l+1}, \dots, s_h)} \pi_{\tilde{s}} N(\mathbf{z}^{(l)}; \tilde{\boldsymbol{\mu}}_{\tilde{s}}^{(l+1)}, \tilde{\boldsymbol{\Sigma}}_{\tilde{s}}^{(l+1)}), \quad (2)$$

Conditionals:

$$f(\mathbf{z}^{(l)} | \mathbf{z}^{(l+1)}; \Theta) = \sum_{i=1}^{k_{l+1}} \pi_i^{(l+1)} N(\eta_i^{(l+1)} + \Lambda_i^{(l+1)} \mathbf{z}^{(l+1)}, \Psi_i^{(l+1)}). \quad (3)$$

By considering the data as the zero layer, $\mathbf{y} = \mathbf{z}^{(0)}$, in a DGMM all the marginal distributions of the latent variables $\mathbf{z}^{(l)}$ and their conditional distributions to the upper level of the network are distributed as Gaussian mixtures.

Marginals:

$$f(\mathbf{z}^{(l)}; \Theta) = \sum_{\tilde{s}=(s_{l+1}, \dots, s_h)} \pi_{\tilde{s}} N(\mathbf{z}^{(l)}; \tilde{\boldsymbol{\mu}}_{\tilde{s}}^{(l+1)}, \tilde{\boldsymbol{\Sigma}}_{\tilde{s}}^{(l+1)}), \quad (2)$$

Conditionals:

$$f(\mathbf{z}^{(l)} | \mathbf{z}^{(l+1)}; \Theta) = \sum_{i=1}^{k_{l+1}} \pi_i^{(l+1)} N(\eta_i^{(l+1)} + \Lambda_i^{(l+1)} \mathbf{z}^{(l+1)}, \Psi_i^{(l+1)}). \quad (3)$$

To **assure identifiability**: at each layer from 1 to $h - 1$, the conditional distribution of the latent variables $f(\mathbf{z}^{(l)} | \mathbf{z}^{(l+1)}; \Theta)$ has zero mean and identity covariance matrix and $\Lambda^\top \Psi^{-1} \Lambda$ is diagonal.

Two-layer DGMM

- (1) $\mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)}$ with prob. $\pi_{s_1}^{(1)}$, $j = 1, \dots, k_1$
- (2) $\mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)}$ with prob. $\pi_{s_2}^{(2)}$, $i = 1, \dots, k_2$

where $\mathbf{z}_i^{(2)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r_2})$, $\Lambda_{s_1}^{(1)}$ is a (factor loading) matrix of dimension $p \times r_1$, $\Lambda_{s_2}^{(2)}$ has dimension $r_1 \times r_2$ and $\Psi_{s_1}^{(1)}, \Psi_{s_2}^{(2)}$ are squared matrices of dimension $p \times p$ and $r_1 \times r_1$ respectively. The two latent variables have dimension $r_1 < p$ and $r_2 < r_1$.

Two-layer DGMM

- (1) $\mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)}$ with prob. $\pi_{s_1}^{(1)}$, $j = 1, \dots, k_1$
- (2) $\mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)}$ with prob. $\pi_{s_2}^{(2)}$, $i = 1, \dots, k_2$

It includes:

- MFA: if $h = 1$ and $\Psi_{s_1}^{(1)}$ are diagonal and $\mathbf{z}_i^{(1)} \sim N(\mathbf{0}, \mathbf{I}_{r_1})$;

Two-layer DGMM

- (1) $\mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)}$ with prob. $\pi_{s_1}^{(1)}$, $j = 1, \dots, k_1$
- (2) $\mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)}$ with prob. $\pi_{s_2}^{(2)}$, $i = 1, \dots, k_2$

It includes:

- MFA: if $h = 1$ and $\Psi_{s_1}^{(1)}$ are diagonal and $\mathbf{z}_i^{(1)} \sim N(\mathbf{0}, \mathbf{I}_{r_1})$;
- FMA (or common MFA): $h = 2$ with $k_1 = 1$, $\Psi^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$;

Two-layer DGMM

- (1) $\mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)}$ with prob. $\pi_{s_1}^{(1)}$, $j = 1, \dots, k_1$
- (2) $\mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)}$ with prob. $\pi_{s_2}^{(2)}$, $i = 1, \dots, k_2$

It includes:

- MFA: if $h = 1$ and $\Psi_{s_1}^{(1)}$ are diagonal and $\mathbf{z}_i^{(1)} \sim N(\mathbf{0}, \mathbf{I}_{r_1})$;
- FMA (or common MFA): $h = 2$ with $k_1 = 1$, $\Psi^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$;
- Mixtures of MFA: $h = 2$ with $k_1 > 1$, $\Psi_{s_1}^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$;

Two-layer DGMM

$$(1) \quad \mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)} \text{ with prob. } \pi_{s_1}^{(1)}, j = 1, \dots, k_1$$

$$(2) \quad \mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)} \text{ with prob. } \pi_{s_2}^{(2)}, i = 1, \dots, k_2$$

It includes:

- MFA: if $h = 1$ and $\Psi_{s_1}^{(1)}$ are diagonal and $\mathbf{z}_i^{(1)} \sim N(\mathbf{0}, \mathbf{I}_{r_1})$;
- FMA (or common MFA): $h = 2$ with $k_1 = 1$, $\Psi^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$;
- Mixtures of MFA: $h = 2$ with $k_1 > 1$, $\Psi_{s_1}^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$;
- Deep MFA (Tang et al. 2012): $h = 2$, $\Psi_{s_1}^{(1)}$ and $\Psi_{s_2}^{(2)}$ diagonal.

Fitting the DGMM

Thanks to the hierarchical form of the architecture of the DGMM, the EM algorithm seems to be the natural procedure.

Conditional expectation for $h = 2$:

$$\begin{aligned}
 E_{\mathbf{z}, s | \mathbf{y}; \Theta'} [\log L_c(\Theta)] &= \sum_{s \in \Omega} \int f(\mathbf{z}^{(1)}, s | \mathbf{y}; \Theta') \log f(\mathbf{y} | \mathbf{z}^{(1)}, s; \Theta) d\mathbf{z}^{(1)} \\
 + \sum_{s \in \Omega} \int \int f(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, s | \mathbf{y}; \Theta') \log f(\mathbf{z}^{(1)} | \mathbf{z}^{(2)}, s; \Theta) d\mathbf{z}^{(1)} d\mathbf{z}^{(2)} \\
 + \int f(\mathbf{z}^{(2)} | \mathbf{y}; \Theta') \log f(\mathbf{z}^{(2)}) d\mathbf{z}^{(2)} + \sum_{s \in \Omega} f(s | \mathbf{y}; \Theta') \log f(s; \Theta),
 \end{aligned}$$

Fitting the DGMM

Thanks to the hierarchical form of the architecture of the DGMM, the EM algorithm seems to be the natural procedure.

Conditional expectation for $h = 2$:

$$\begin{aligned}
 E_{\mathbf{z}, s | \mathbf{y}; \Theta'} [\log L_c(\Theta)] &= \sum_{s \in \Omega} \int f(\mathbf{z}^{(1)}, s | \mathbf{y}; \Theta') \log f(\mathbf{y} | \mathbf{z}^{(1)}, s; \Theta) d\mathbf{z}^{(1)} \\
 + \sum_{s \in \Omega} \int \int f(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, s | \mathbf{y}; \Theta') \log f(\mathbf{z}^{(1)} | \mathbf{z}^{(2)}, s; \Theta) d\mathbf{z}^{(1)} d\mathbf{z}^{(2)} \\
 + \int f(\mathbf{z}^{(2)} | \mathbf{y}; \Theta') \log f(\mathbf{z}^{(2)}) d\mathbf{z}^{(2)} + \sum_{s \in \Omega} f(s | \mathbf{y}; \Theta') \log f(s; \Theta),
 \end{aligned}$$

Fitting the DGMM via a Stochastic EM

- Draw unobserved observations or samples of observations from their conditional density given the observed data
- SEM (Celeux and Diebolt, 1985)
- MCEM (Wei and Tanner, 1990)

Fitting the DGMM via a Stochastic EM

- Draw unobserved observations or samples of observations from their conditional density given the observed data
- SEM (Celeux and Diebolt, 1985)
- MCEM (Wei and Tanner, 1990)

The strategy adopted is to draw pseudorandom observations at each layer of the network through the conditional density $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l-1)}, \mathbf{s}; \Theta')$, starting from $l = 1$ to $l = h$, by considering, as known, the variables at the upper level of the model for the current fit of parameters, where at the first layer $\mathbf{z}^{(0)} = \mathbf{y}$.

Stochastic EM

For $l = 1, \dots, h$:

- **S-STEP** ($\mathbf{z}_i^{(l-1)}$ is known)
Generate M replicates $\mathbf{z}_{i,m}^{(l)}$ from $f(\mathbf{z}_i^{(l)} | \mathbf{z}_i^{(l-1)}, s; \Theta')$
- **E-STEP**
Approximate:

$$E[\mathbf{z}_i^{(l)} | \mathbf{z}_i^{(l-1)}, s; \Theta'] \cong \frac{\sum_{m=1}^M \mathbf{z}_{i,m}^{(l)}}{M}$$

and

$$E[\mathbf{z}_i^{(l)} \mathbf{z}_i^{(l)\top} | \mathbf{z}_i^{(l-1)}, s; \Theta'] \cong \frac{\sum_{m=1}^M \mathbf{z}_{i,m}^{(l)} \mathbf{z}_{i,m}^{(l)\top}}{M}$$

- **M-STEP**
Compute the current estimated for the parameters

Real examples

■ Wine Data

$p = 27$ chemical and physical properties of $k = 3$ types of wine from the Piedmont region of Italy: Barolo (59), Grignolino (71), and Barbera (48). Clusters are well separated and most clustering methods give high clustering performance on this data.

■ Olive Data

percentage composition of $p = 8$ fatty acids found by lipid fraction of 572 Italian olive oils coming from $k = 3$ regions: Southern Italy (323), Sardinia (98), and Northern Italy (151). Clustering is not a very difficult task even if the clusters are not balanced.

Real examples

Ecoli Data

- proteins classified into their various cellular localization sites based on their amino acid sequences
- $p = 7$ variables
- $n = 336$ units
- $k = 8$ unbalanced classes: cp cytoplasm (143), inner membrane without signal sequence (77), periplasm (52), inner membrane, uncleavable signal sequence (35), outer membrane (20), outer membrane lipoprotein (5), inner membrane lipoprotein (2), inner membrane, cleavable signal sequence (2)

Real examples

Vehicle Data

- silhouette of vehicles represented from many different angles
- $p = 18$ angles
- $n = 846$ units
- $k = 4$ types of vehicles: a double decker bus (218), Cheverolet van (199), Saab 9000 (217) and an Opel Manta 400 (212)
- difficult task: very hard to distinguish between the 2 cars.

Real examples

Satellite Data

- multi-spectral, scanner image data purchased from NASA by the Australian Centre for Remote Sensing
- 4 digital images of the same scene in different spectral bands
- 3×3 square neighborhood of pixels
- $p = 36$ variables
- $n = 6435$ images
- $k = 6$ groups of images: red soil (1533), cotton crop (703), grey soil (1358), damp grey soil (626), soil with vegetation stubble (707) and very damp grey soil (1508)
- difficult task due to both unbalanced groups and dimensionality

Results

- DGMM: $h = 2$ and $h = 3$ layers, $k_1 = k^*$ and $k_2 = 1, 2, \dots, 5$ ($k_3 = 1, 2, \dots, 5$), all possible models with $p > r_1 > \dots > r_h \geq 1$
- 10 different starting points
- Model selection by BIC
- Comparison with Gaussian Mixture Models (GMM), skew-normal and skew-t Mixture Models (SNmm and STmm), k -means and the Partition Around Medoids (PAM), hierarchical clustering with Ward distance (Hclust), Factor Mixture Analysis (FMA), and Mixture of Factor Analyzers (MFA)

Results

Model selection by BIC:

- Wine data: $h = 2$, $p = 27$, $r_1 = 3$, $r_2 = 2$ and $k_1 = 3$, $k_2 = 1$
- Olive data: $h = 2$, $p = 8$, $r_1 = 5$, $r_2 = 1$ and $k_1 = 3$, $k_2 = 1$
- Ecoli data: $h = 2$, $p = 7$, $r_1 = 2$, $r_2 = 1$ and $k_1 = 8$, $k_2 = 1$
- Vehicle data: $h = 2$, $p = 18$, $r_1 = 7$, $r_2 = 1$ and $k_1 = 4$, $k_2 = 3$.
- Satellite data: $h = 3$, $p = 36$, $r_1 = 13$, $r_2 = 2$, $r_1 = 1$ and $k_1 = 6$, $k_2 = 2$, $k_1 = 1$

Results

<i>Dataset</i>	<i>Wine</i>		<i>Olive</i>		<i>Ecoli</i>		<i>Vehicle</i>		<i>Satellite</i>	
	ARI	m.r.	ARI	m.r.	ARI	m.r.	ARI	m.r.	ARI	m.r.
<i>kmeans</i>	0.930	0.022	0.448	0.234	0.548	0.298	0.071	0.629	0.529	0.277
PAM	0.863	0.045	0.725	0.107	0.507	0.330	0.073	0.619	0.531	0.292
Hclust	0.865	0.045	0.493	0.215	0.518	0.330	0.092	0.623	0.446	0.337
GMM	0.917	0.028	0.535	0.195	0.395	0.414	0.089	0.621	0.461	0.374
SNmm	0.964	0.011	0.816	0.168	-	-	0.125	0.566	0.440	0.390
STmm	0.085	0.511	0.811	0.171	-	-	0.171	0.587	0.463	0.390
FMA	0.361	0.303	0.706	0.213	0.222	0.586	0.093	0.595	0.367	0.426
MFA	0.983	0.006	0.914	0.052	0.525	0.330	0.090	0.626	0.589	0.243
DGMM	0.983	0.006	0.997	0.002	0.749	0.187	0.191	0.481	0.604	0.249

Final remarks

'Deep' means a multilayer architecture.

Deep NN works very well in machine learning (supervised classification)

Our aim: unsupervised classification

Final remarks

'Deep' means a multilayer architecture.

Deep NN works very well in machine learning (supervised classification)

Our aim: unsupervised classification

Deep mixtures require high n !

Model selection is another issue

Computationally intensive for $h > 3$

But for $h = 2$ and $h = 3$ results are promising

Final remarks

'Deep' means a multilayer architecture.

Deep NN works very well in machine learning (supervised classification)

Our aim: unsupervised classification

Deep mixtures require high n !

Model selection is another issue

Computationally intensive for $h > 3$

But for $h = 2$ and $h = 3$ results are promising

It being a generalization of mixtures (and of MFA)

it is guaranteed to work as well as these methods

Final remarks

'Deep' means a multilayer architecture.

Deep NN works very well in machine learning (supervised classification)

Our aim: unsupervised classification

Deep mixtures require high n !

Model selection is another issue

Computationally intensive for $h > 3$

But for $h = 2$ and $h = 3$ results are promising

It being a generalization of mixtures (and of MFA)

it is guaranteed to work as well as these methods

Remember: for simple clustering problems DGMM is like to use

'sledgehammer to crack a nut'

References

- Celeux, G. and J. Diebolt (1985). *The SEM algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem*. Computational statistics
- Hennig, C. (2010). *Methods for merging gaussian mixture components*. ADAC
- Li, J. (2005). *Clustering based on a multilayer mixture model*. JCGS
- McLachlan, G., D. Peel, and R. Bean (2003). *Modelling high-dimensional data by mixtures of factor analyzers*. CSDA
- McNicholas, P. D. and T. B. Murphy (2008). *Parsimonious gaussian mixture models*. Statistics and Computing
- Tang, Y., G. E. Hinton, and R. Salakhutdinov (2012). *Deep mixtures of factor analysers*. Proceedings of the 29th International Conference on Machine Learning
- Viroli, C. (2010). *Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers*. Journal of Classification
- Viroli, C. and McLachlan, G. (2018), *Deep Gaussian Mixture Models*, Statistics and Computing