Recent advances in Deep Mixture Models

Cinzia Viroli
(University of Bologna, Italy)

joint with Geoff McLachlan (University of Queensland, Australia),
Laura Anderlucci (University of Bologna)

Wien, December 06, 2019

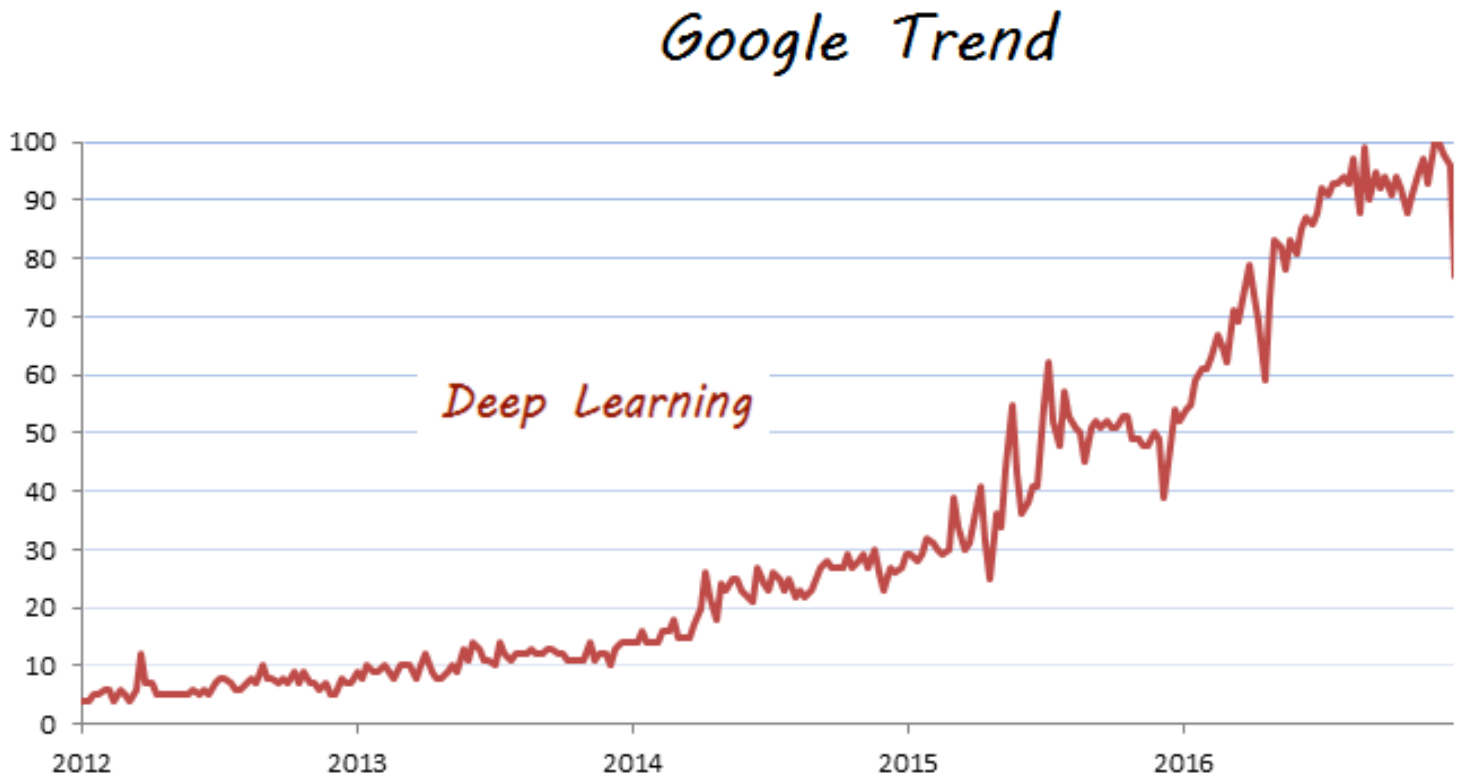
Outline of this talk

1. Deep Learning
2. Deep Gaussian Mixture Models
3. (i) Shallow vs (ii) Bayesian hierarchical vs
   (iii) Deeper mixtures (an example to textual data)
4. General Final remarks

Deep Learning

Deep Learning

Deep Learning is a trendy topic in the machine learning community



What is Deep Learning?

- Built over the framework of neural networks
- First formulations date back at least to the 1960s: neural networks with multiple layers
- The term "deep learning" became popular from the paper by Krizhevsky et al. in 2012

What is Deep Learning?

- Built over the framework of neural networks
- First formulations date back at least to the 1960s: neural networks with multiple layers
- The term "deep learning" became popular from the paper by Krizhevsky et al. in 2012
- Now they have attracted wide-spread attention; mainly for 'supervised' classification, where they very often outperform alternative learning methods

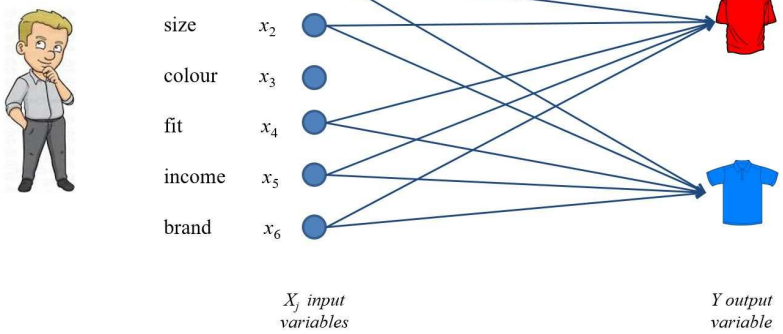
What is Deep Learning?

- Built over the framework of neural networks
- First formulations date back at least to the 1960s: neural networks with multiple layers
- The term "deep learning" became popular from the paper by Krizhevsky et al. in 2012
- Now they have attracted wide-spread attention; mainly for 'supervised' classification, where they very often outperform alternative learning methods

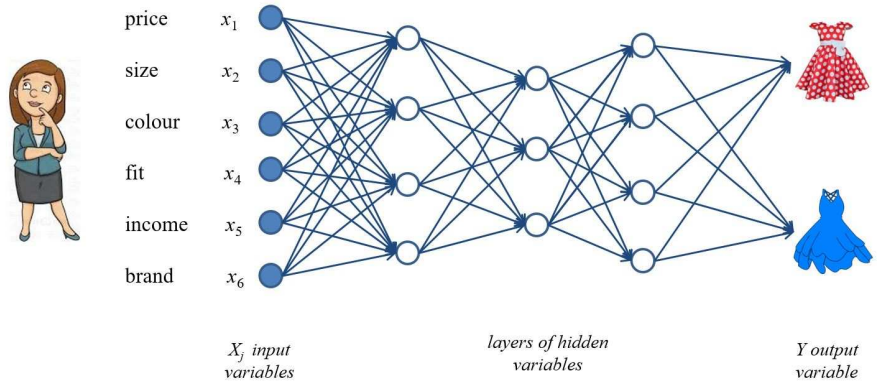
Deep Learning is a set of algorithms able to gradually learning a huge number of parameters in an architecture composed by multiple non linear transformations (multi-layer structure)

Learning vs Deep Learning

Choice modeling



Deep(er) choice modeling



Deep Learning

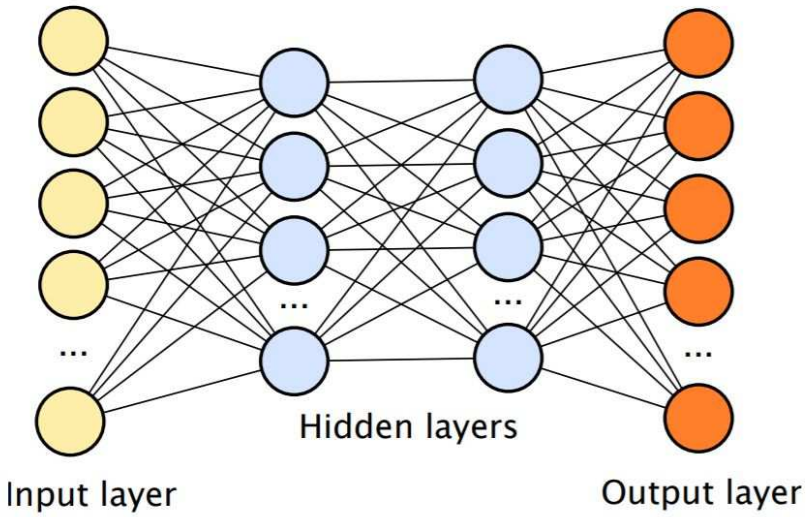
- The idea is to mimic the brain...

Deep Learning

- The idea is to mimic the brain... (in particular the women's brains!)

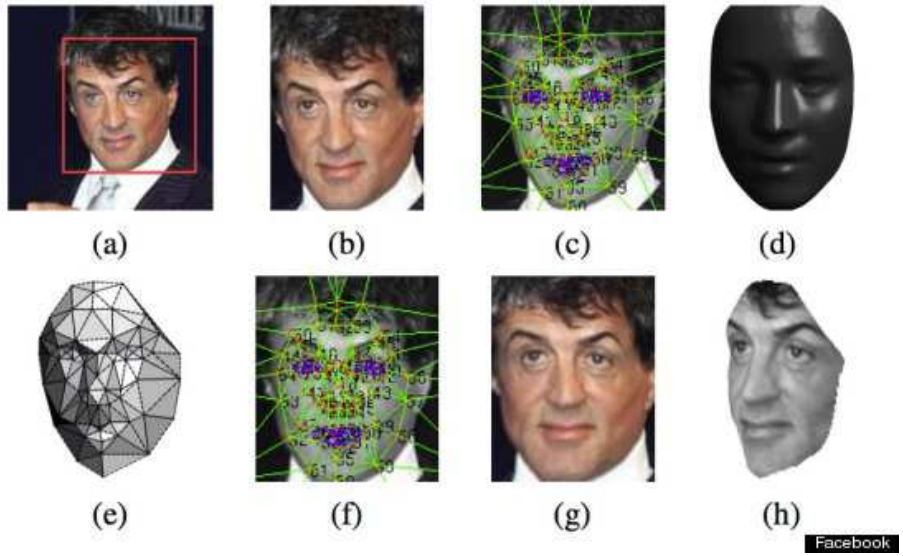
Deep Learning

- The idea is to mimic the brain... (in particular the women's brains!)
- The connections between nodes reflect connections between neurons
- A very flexible black-box able to learn whatever input-output mappings (provided that data are large enough)



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 can identify human faces in digital images with an accuracy of 97.35%.

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)
- Deep Generative Models (Rezende et al., 2014) are latent variable models with an hierarchical structure, which are computationally scalable to high dimensional data

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)
- Deep Generative Models (Rezende et al., 2014) are latent variable models with an hierarchical structure, which are computationally scalable to high dimensional data
- Causal effect inference via deep learning (Louizos et al., 2017)

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)
- Deep Generative Models (Rezende et al., 2014) are latent variable models with an hierarchical structure, which are computationally scalable to high dimensional data
- Causal effect inference via deep learning (Louizos et al., 2017)
- Combination of graphical models and deep learning (sum-product networks, Peharz, 2017)

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)
- Deep Generative Models (Rezende et al., 2014) are latent variable models with an hierarchical structure, which are computationally scalable to high dimensional data
- Causal effect inference via deep learning (Louizos et al., 2017)
- Combination of graphical models and deep learning (sum-product networks, Peharz, 2017)
- Deep learning for anomaly detection (Feng et al., 2017) and missing data (Mattei and Frellsen, 2018)

Examples of Deep Learning

- Artificial Neural Network with multiple layers and advanced neural network architectures (Feed Forward, Recurrent, Auto-encoder, Convolution NN). Most of them use the stochastic gradient descent algorithm. Mainly developed in supervised classification (for an historical overview: Schmidhuber, 2015)
- Deep Generative Models (Rezende et al., 2014) are latent variable models with an hierarchical structure, which are computationally scalable to high dimensional data
- Causal effect inference via deep learning (Louizos et al., 2017)
- Combination of graphical models and deep learning (sum-product networks, Peharz, 2017)
- Deep learning for anomaly detection (Feng et al., 2017) and missing data (Mattei and Frellsen, 2018)
- Multilayer Gaussian Mixture Models (Van den Oord and Schrauwen, 2014)

Deep Gaussian Mixture Models

Gaussian Mixture Models (GMM)

Growing popularity, widely used.

Many extensions for:
Dimensionally reduced parameterizations: Banfield and Raftery (1993), Celeux and Govaert (1995), Bouveyron et al. (2007) ...
High-dimensional data via factorial structures: Ghahrami and Hilton (1997), McLachlan et al. (2003), Yoshida et al. (2004), Baek and McLachlan (2008), McNicolas and Murphy (2008), Montanari and Viroli (2010), K. Murphy et al. (2019) ...
Non-Gaussian data: McLachlan et al. (2007), Lin (2009), Andrews and McNicholas (2011), Lee and McLachlan (2011-2018), Subedi and McNicholas (2014), Franczak et al. (2014), ...
Merging: Hennig (2010), Baudry et al. (2010), Melnykov (2016) ...
Mixtures of mixtures models: Li (2005), Viroli (2010), Malsiner-Walli et al. (2017)...
Model selection: Richardson and Green (1997), Biernacki et al. (2000), Raftery and Dean (2006), Malsiner-Walli et al. (2016)...

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

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

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

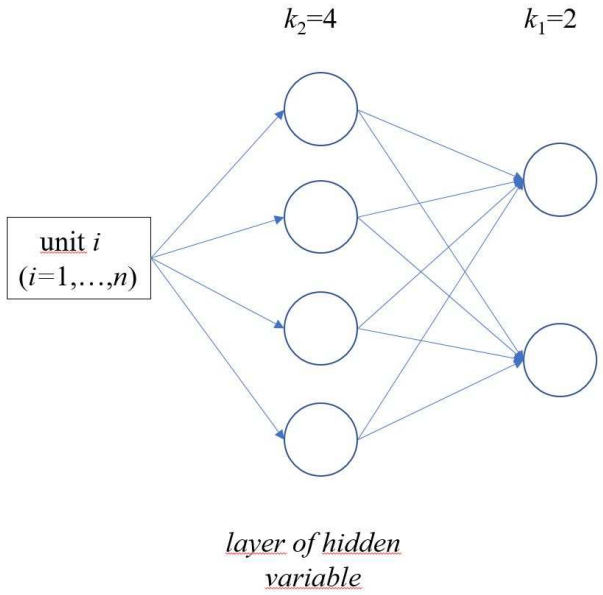
Now suppose we replace $\mathbf{z} \sim N(0, I_p)$ with

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

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

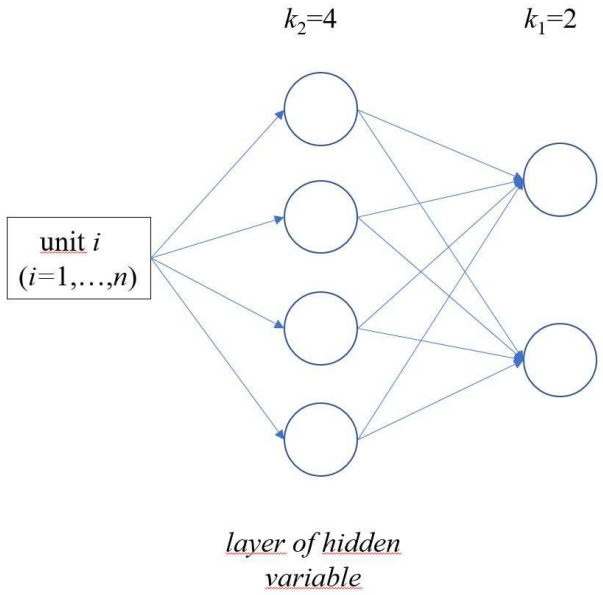
Deep Gaussian Mixtures

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



Deep Gaussian Mixtures

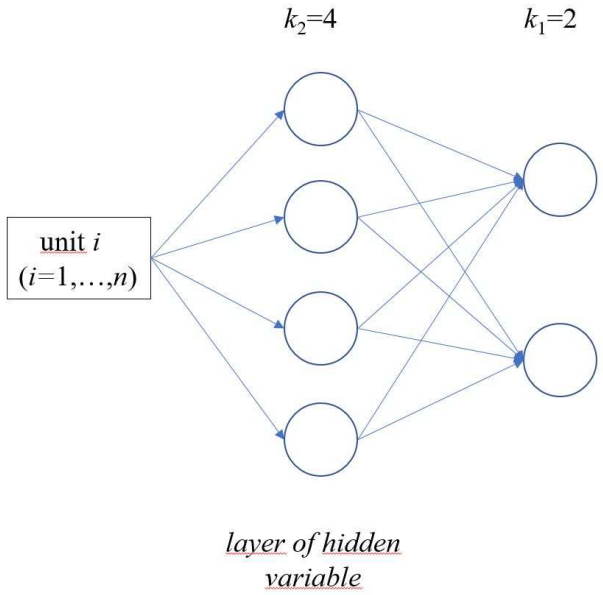
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



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

Deep Gaussian Mixtures

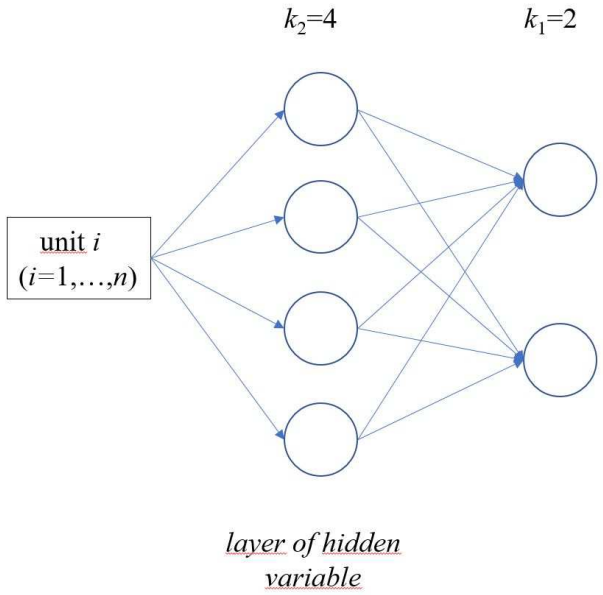
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



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

Deep Gaussian Mixtures

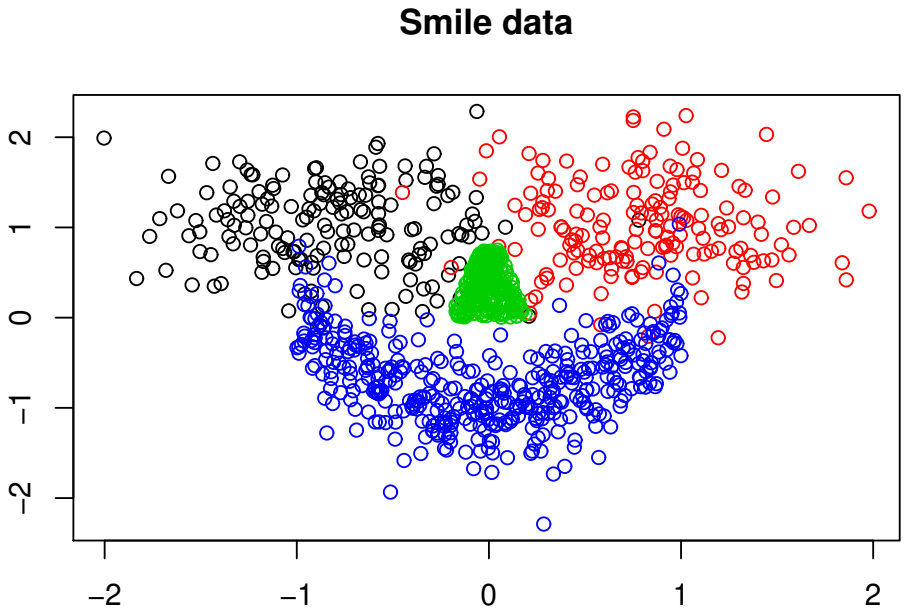
Imagine $h = 2$, $k_2 = 4$ and $k_1 = 2$:



- $k = 8$ possible paths (total subcomponents)
- $M = 6$ real subcomponents (shared set of parameters)
- $M < k$ thanks to the tying
- Proposed by Van den Oord and Schrauwen (2014). 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, $M = 6$)
- 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?

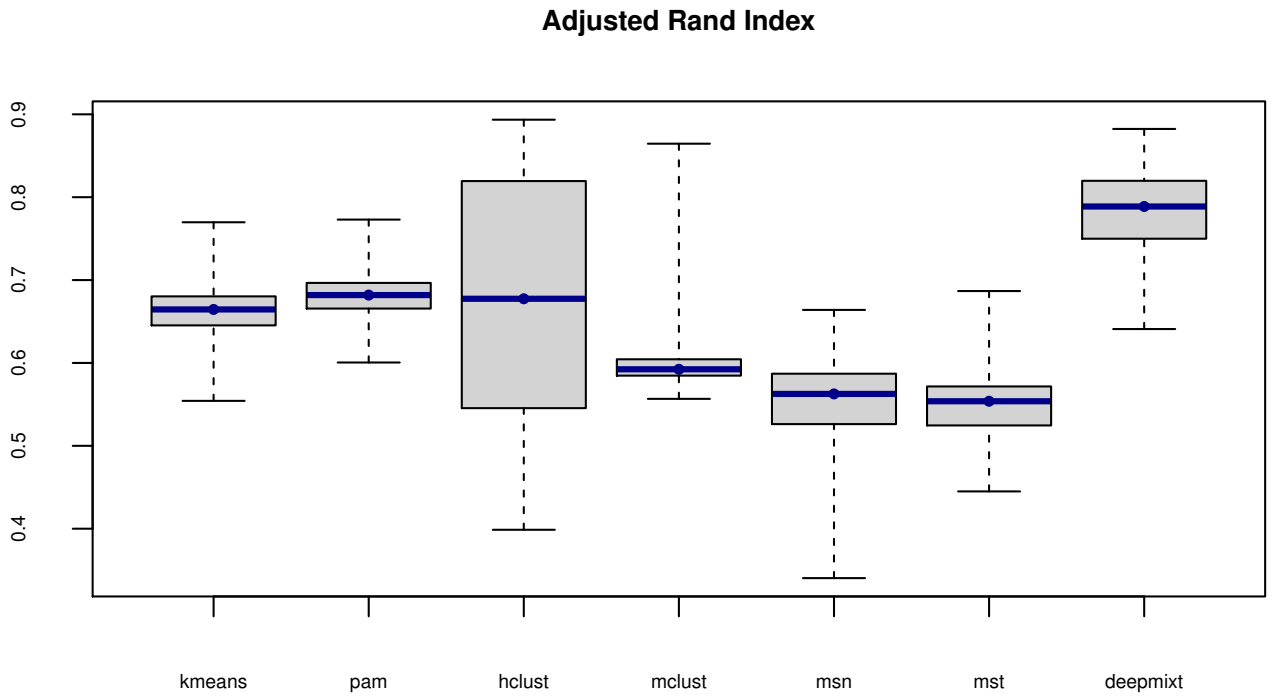
- A deep mixture with $h = 2, k_1 = 4, k_2 = 2$ ($k = 8$ paths, $M = 6$)
- In the DGMM we cluster data k_1 groups ($k_1 < k$) through $f(\mathbf{y}|\mathbf{z})$: the remaining components in the previous layer(s) act as density approximation of global non-Gaussian components
- Automatic tool for merging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

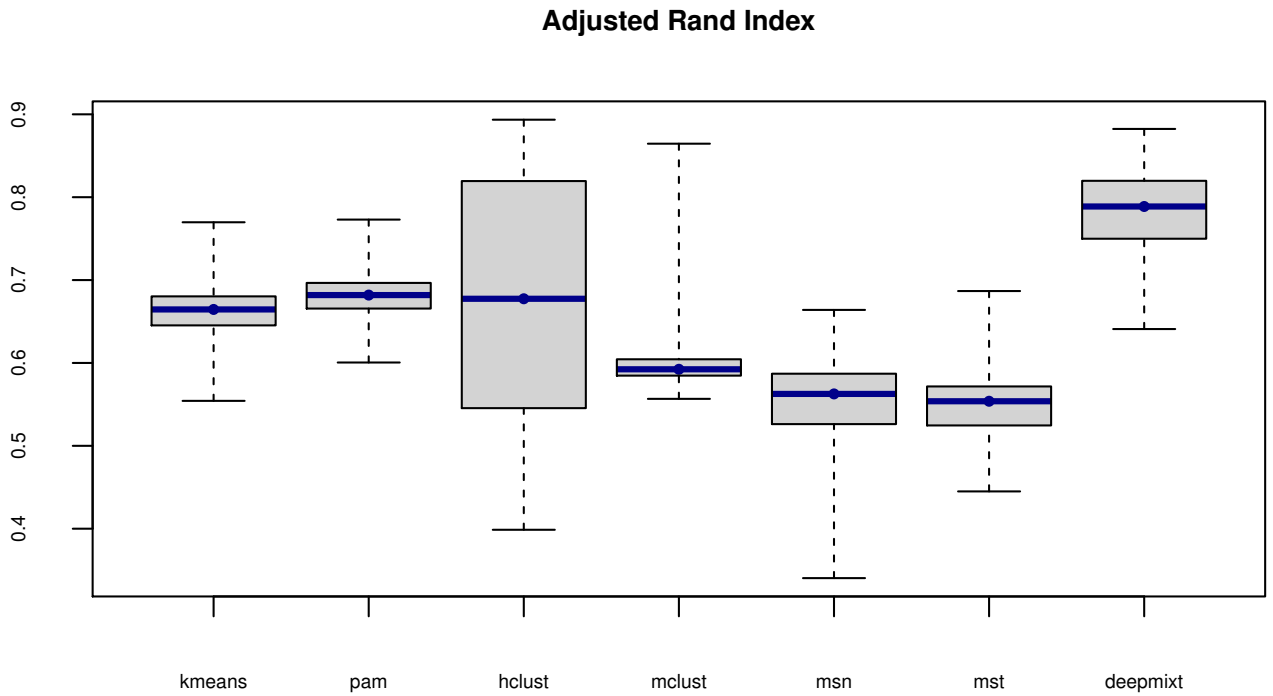
Do we really need DGMM?

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). Identifiability is not addressed.

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). Identifiability is not addressed.
- 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.
- Package `deepgmm` on CRAN R.

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:

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

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

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

$$\sum_{s \in \Omega} \pi_s = \sum_{s_1, \ldots, s_h} \pi_{(s1, \ldots, s_h)} = 1.$$

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

$$\sum_{s \in \Omega} \pi_s = \sum_{s_1, \ldots, s_h} \pi_{(s1, \ldots, s_h)} = 1.$$

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

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

and

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

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:

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

Conditionals:

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

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:

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

Conditionals:

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

To assure identifiability: at each layer from 1 to $h-1$, the conditional distribution of the latent variables $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)} = 0; \boldsymbol{\Theta})$ has zero mean and identity covariance matrix and $\Lambda^\top \Psi^{-1} \Lambda$ is diagonal.

Two-layer DGMM

$$(1) \quad \mathbf{y}_i = \eta^{(1)}_{s_1} + \Lambda^{(1)}_{s_1}\mathbf{z}^{(1)}_i + \mathbf{u}^{(1)}_i \text{ with prob. } \pi^{(1)}_{s_1},\ j = 1, \ldots, k_1$$

$$(2) \quad \mathbf{z}^{(1)}_i = \eta^{(2)}_{s_2} + \Lambda^{(2)}_{s_2}\mathbf{z}^{(2)}_i + \mathbf{u}^{(2)}_i \text{ with prob. } \pi^{(2)}_{s_2},\ i = 1, \ldots, k_2$$

where $\mathbf{z}^{(2)}_i \sim N(\mathbf{0}, \mathbf{I}_{r_2})$, $\Lambda^{(1)}_{s_1}$ is a (factor loading) matrix of dimension $p \times r_1$, $\Lambda^{(2)}_{s_2}$ has dimension $r_1 \times r_2$ and $\Psi^{(1)}_{s_1}, \Psi^{(2)}_{s_2}$ 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

$$
\begin{aligned}
&(1) \quad \mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)} \text{ with prob. } \pi_{s_1}^{(1)},\ j = 1, \ldots, k_1 \\
&(2) \quad \mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)} \text{ with prob. } \pi_{s_2}^{(2)},\ i = 1, \ldots, k_2
\end{aligned}
$$

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

$$
\begin{aligned}
&(1) \quad \mathbf{y}_i = \eta^{(1)}_{s_1} + \Lambda^{(1)}_{s_1}\mathbf{z}^{(1)}_i + \mathbf{u}^{(1)}_i \text{ with prob. } \pi^{(1)}_{s_1},\ j = 1, \ldots, k_1 \\
&(2) \quad \mathbf{z}^{(1)}_i = \eta^{(2)}_{s_2} + \Lambda^{(2)}_{s_2}\mathbf{z}^{(2)}_i + \mathbf{u}^{(2)}_i \text{ with prob. } \pi^{(2)}_{s_2},\ i = 1, \ldots, k_2
\end{aligned}
$$

It includes:

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

Two-layer DGMM

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

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

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

$$
\begin{aligned}
&(1) \quad \mathbf{y}_i = \eta_{s_1}^{(1)} + \Lambda_{s_1}^{(1)} \mathbf{z}_i^{(1)} + \mathbf{u}_i^{(1)} \text{ with prob. } \pi_{s_1}^{(1)},\ j = 1, \ldots, k_1 \\
&(2) \quad \mathbf{z}_i^{(1)} = \eta_{s_2}^{(2)} + \Lambda_{s_2}^{(2)} \mathbf{z}_i^{(2)} + \mathbf{u}_i^{(2)} \text{ with prob. } \pi_{s_2}^{(2)},\ i = 1, \ldots, k_2
\end{aligned}
$$

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 $h = 2$:

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

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 $h = 2$:

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

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 $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l-1)}, s; \boldsymbol{\Theta}')$, 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 $l = 1, \ldots, h$:

- S-STEP ($\mathbf{z}_i^{(l-1)}$ is known)

  Generate M replicates $\mathbf{z}_{i,m}^{(l)}$ from $f(\mathbf{z}_i^{(l)} | \mathbf{z}_i^{(l-1)}, s; \boldsymbol{\Theta}')$

- E-STEP

  Approximate:

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

  and

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

- 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), perisplasm (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 $k_2 = 1, 2, \ldots, 5$ ($k_3 = 1, 2, \ldots, 5$), all possible models with $p > r_1 > ... > r_h \geq 1$
- 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 $k_1 = 6,\ k_2 = 2,\ k_3 = 1$

Results

Dataset	*Wine*		*Olive*		*Ecoli*		*Vehicle*		*Satellite*	
	ARI	m.r.	ARI	m.r.	ARI	m.r.	ARI	m.r.	ARI	m.r.
*k*means	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

Some general comments

'Deep' refers to a technical, architectural property (multilayer structure)

Deep NNs work very well in machine learning (supervised classification)
Our aim: unsupervised classification

Some general comments

'Deep' refers to a technical, architectural property (multilayer structure)

Deep NNs work very well in machine learning (supervised classification)
Our aim: unsupervised classification

'Deep' is put in contrast to 'shallow' classical estimation methods.

Some general comments

'Deep' refers to a technical, architectural property (multilayer structure)

Deep NNs work very well in machine learning (supervised classification)
Our aim: unsupervised classification

'Deep' is put in contrast to 'shallow' classical estimation methods.

Not all methods and models in statistics are shallow. For instance:

1. Ensemble methods in classification are deep learning strategies (with non-interconnected layers)
2. Bayesian hierarchical models have similar structure and advantages of deep learners

(i) Shallow vs (ii) Bayesian hierarchical vs
(iii) Deeper mixtures (an example to textual data)

Empirical context

- Textual data collected by an important Italian mobile carrier company: phone calls received by the customer service (*tickets*): when a customer calls the assistance service, the company operator labels it as e.g. a complaint, a request of clarification, a request of information for specific services, deals or promotions.
- Examples:

Dov'è la modulistica per la disattivazione della linea?	*Where are the forms for the line deactivation?*
Cos'è il codice di sicurezza?	*What is the security code?*
Mi dite ke promozione ho?	*Can you tell me which promotion I have?*
...	...

Empirical context

The dataset contains $n = 2129$ tickets and $T = 489$ terms obtained after preprocessing. Tickets belong to $k = 5$ topics/groups:

Table: Number of tickets for each class.

Class	Frequencies
Activation of SIM, ADSL, new contracts	407
General information about current balance, consumption, etc.	471
Request of information about new offers and promotions	376
Top-up	435
Problems with password, top-up, internet connection, etc.	440

Empirical context

- Aim: to derive a clustering strategy that allows to automatically assign the tickets to their topic without the human judgment of an operator.
- Challenges: an elevate degree of sparsity (after a pre-processing step, the tickets exhibit indeed an average length of 5 words only and, thus, the document-term matrix contains zero almost everywhere).

Empirical context

Due to the challenges of the data, most conventional clustering strategies fail:

Table: Adjusted Rand Index (ARI) and Accuracy for different methods.

Method	ARI	Accuracy
k-means with cosine dissimilarity	0.233	0.516
k-means with Euclidean distance on Semantic	0.135	0.464
PAM with cosine dissimilarity	0.000	0.222
Mixture of Gaussians on Semantic	0.158	0.438
Hierarchical - Ward's method with cosine dissimilarity	-0.001	0.225
Hierarchical - Centroid method with cosine dissimilarity	0.000	0.223
Hierarchical - Single linkage with cosine dissimilarity	0.000	0.222
Hierarchical - Complete linkage with cosine dissimilarity	-0.001	0.224
Hierarchical - Average linkage with cosine dissimilarity	0.000	0.223
Latent Dirichlet Allocation	0.049	0.318

(i) The shallow model

- Mixtures of Unigrams (Nigam et al., 2000) are one of the simplest and most efficient tool for clustering textual data

(i) The shallow model

- Mixtures of Unigrams (Nigam et al., 2000) are one of the simplest and most efficient tool for clustering textual data
- Assumption: documents related to the same topic have similar distributions of terms

(i) The shallow model

- Mixtures of Unigrams (Nigam et al., 2000) are one of the simplest and most efficient tool for clustering textual data
- Assumption: documents related to the same topic have similar distributions of terms
- k topics with distribution naturally described by Multinomials

Mixtures of Unigrams: notation

- $\mathbf{X}$ is a document-term matrix of dimension $n \times T$ containing the word frequencies of each document
- k topics or groups
- $\mathbf{x}_d$ is the single document of length T, with $d = 1, \ldots, n$

Mixtures of Unigrams: notation

- $\mathbf{X}$ is a document-term matrix of dimension $n \times T$ containing the word frequencies of each document
- k topics or groups
- $\mathbf{x}_d$ is the single document of length T, with $d = 1, \ldots, n$

In MoU the distribution of each document has a specific distribution function conditionally on the values of a discrete latent variable z_d:

$$p(\mathbf{x}_d) = \sum_{i=1}^{k} \pi_i \frac{N_d!}{\prod_{t=1}^{T} x_{dt}!} \prod_{t=1}^{T} \omega_{ti}^{x_{dt}},$$

with $N_d = \sum_{t=1}^{T} x_{dt}$.

Fitting and results

The EM algorithm is usually applied for parameter estimation.
Clustering is improved:

Table: Adjusted Rand Index (ARI) and Accuracy for different methods.

Method	ARI	Accuracy
k-means with cosine dissimilarity	0.233	0.516
...	...	...
Mixtures of Unigrams	0.557	0.763

Easy interpretability; computational time (!)

(ii) Bayesian hierarchical approach

We add a layer by assuming the proportions $\boldsymbol{\omega}$ are realizations from a Dirichlet distribution of parameters $\boldsymbol{\theta}_i$:

$$p(\boldsymbol{\omega}|z=i) = \frac{\Gamma\left(\sum_{t=1}^{T}\theta_{it}\right)}{\prod_{t=1}^{T}\Gamma(\theta_{it})}\prod_{t=1}^{T}\omega_{ti}^{\theta_{it}-1}, \tag{4}$$

where Γ denotes the Gamma function.

(ii) Bayesian hierarchical approach

The latent variable $\boldsymbol{\omega}$ can be integrated out thus leading to the Dirichlet-Multinomial compound model.

The Dirichlet-Multinomial model is

$$\begin{aligned} p(\mathbf{x}|z=i) &= \int p(\mathbf{x}|z=i,\boldsymbol{\omega})p(\boldsymbol{\omega}|z=i)d\boldsymbol{\omega} \\ &= \frac{\sum_{t=1}^{T} x_t}{\prod_{t=1}^{T} x_t} \frac{B\left(\sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \theta_{it}\right)}{\prod_{t=1}^{T} B\left(x_t, \theta_{it}\right)}, \end{aligned} \tag{5}$$

where B denotes the Beta function.

Dirichlet-Multinomial mixtures

The mixture model

$$p(\mathbf{x}_d) = \sum_{i=1}^{k} \pi_i \frac{\sum_{t=1}^{T} x_t}{\prod_{t=1}^{T} x_t} \frac{B\left(\sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \theta_{it}\right)}{\prod_{t=1}^{T} B\left(x_t, \theta_{it}\right)},$$

can be easily estimated via a gradient descent algorithm or an EM algorithm or a Gibss sampling. Again: computational time (!)

Clustering results

Table: Adjusted Rand Index (ARI) and Accuracy for different methods.

Method	ARI	Accuracy
k-means with cosine dissimilarity	0.233	0.516
...	...	...
Mixtures of Unigrams	0.557	0.763
Mixtures of Dirichlet-Multinomials	0.722	0.871

(iii) A deeper mixture model

(Anderlucci and Viroli, 2019, submitted)

- A further layer in the probabilistic generative model is added.

(iii) A deeper mixture model

(Anderlucci and Viroli, 2019, submitted)

- A further layer in the probabilistic generative model is added.
- At the deepest latent layer, the documents can come from k_2 groups with different probabilities, say $\pi_j^{(2)} \quad j = 1, \ldots, k_2$.

(iii) A deeper mixture model

(Anderlucci and Viroli, 2019, submitted)

- A further layer in the probabilistic generative model is added.
- At the deepest latent layer, the documents can come from k_2 groups with different probabilities, say $\pi_j^{(2)} \quad j = 1, \ldots, k_2$.
- Conditionally to what happened at this level, at the top layer the documents can belong to k_1 groups with conditional probabilities $\pi_{i|j}^{(1)} \quad i = 1, \ldots, k_1$.

(iii) A deeper mixture model

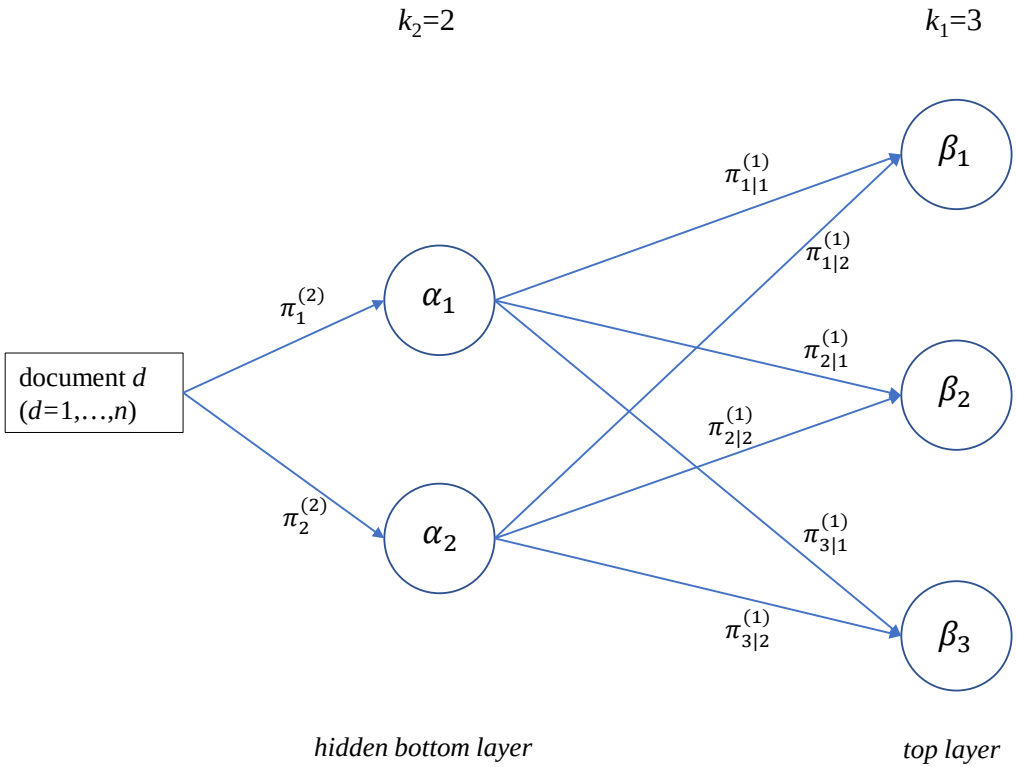


Figure: Structure of a Deep mixture with components $k_1 = 3$ and $k_2 = 2$.

Deep Mixtures of Dirichlet-Multinomials

The distribution of $\mathbf{x}$ conditionally to the two layers is a Dirichlet-Multinomial distribution with parameters $\boldsymbol{\theta}_{ij} = \boldsymbol{\beta}_i + \boldsymbol{\alpha}_j\boldsymbol{\beta}_i = \boldsymbol{\beta}_i(1 + \boldsymbol{\alpha}_j)$, that are cluster-specific:

$$p(\mathbf{x}|z^{(1)} = i, z^{(2)} = j, \boldsymbol{\theta}) = \frac{\sum_{t=1}^{T} x_t}{\prod_{t=1}^{T} x_t} \frac{B\left(\sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \theta_{ijt}\right)}{\prod_{t=1}^{T} B\left(x_t, \theta_{ijt}\right)},$$

Deep Mixtures of Dirichlet-Multinomials

The distribution of $\mathbf{x}$ conditionally to the two layers is a Dirichlet-Multinomial distribution with parameters $\boldsymbol{\theta}_{ij} = \boldsymbol{\beta}_i + \boldsymbol{\alpha}_j\boldsymbol{\beta}_i = \boldsymbol{\beta}_i(1 + \boldsymbol{\alpha}_j)$, that are cluster-specific:

$$p(\mathbf{x}|z^{(1)} = i, z^{(2)} = j, \boldsymbol{\theta}) = \frac{\sum_{t=1}^{T} x_t}{\prod_{t=1}^{T} x_t} \frac{B\left(\sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \theta_{ijt}\right)}{\prod_{t=1}^{T} B\left(x_t, \theta_{ijt}\right)},$$

where $z^{(1)}$ and $z^{(2)}$ are the allocation variables at the top and at the bottom layers, respectively. They are discrete latent variables that follow the following distributions:

$$p(z^{(2)} = j) = \pi_j^{(2)},$$

and

$$p(z^{(1)} = i|z^{(2)} = j) = \pi_{i|j}^{(1)}.$$

Interesting facts

The two sets of parameters are subjected to the restrictions:

$$\beta_{it} > 0$$

and

$$-1 < \alpha_{jt} < 1$$

so that, from the interpretative point of view, $\boldsymbol{\alpha}_j$ acts as a perturbation on the cluster-specific $\boldsymbol{\beta}_i$ parameters of the top layer.

Interesting facts

The two sets of parameters are subjected to the restrictions:

$$\beta_{it} > 0$$

and

$$-1 < \alpha_{jt} < 1$$

so that, from the interpretative point of view, $\boldsymbol{\alpha}_j$ acts as a perturbation on the cluster-specific $\boldsymbol{\beta}_i$ parameters of the top layer.

Under the constraint $-1 < \alpha_{jt} < 1$, the role of k_2 components at the deepest layer is confined to add flexibility to the model, while the real cluster-distribution is specified by the $\boldsymbol{\beta}$ parameters. Therefore k_1 corresponds to the number of clusters.

Interesting facts

The two sets of parameters are subjected to the restrictions:

$$\beta_{it} > 0$$

and

$$-1 < \alpha_{jt} < 1$$

so that, from the interpretative point of view, $\boldsymbol{\alpha}_j$ acts as a perturbation on the cluster-specific $\boldsymbol{\beta}_i$ parameters of the top layer.

Under the constraint $-1 < \alpha_{jt} < 1$, the role of k_2 components at the deepest layer is confined to add flexibility to the model, while the real cluster-distribution is specified by the $\boldsymbol{\beta}$ parameters. Therefore k_1 corresponds to the number of clusters.

The model encompasses the simple Mixture of Dirichlet-Multinomials, that can be obtained as special case when $k_2 = 1$ with $\boldsymbol{\alpha} = 0$.

Results

Table: Adjusted Rand Index (ARI) and Accuracy for different methods.

Method	ARI	Accuracy
k-means with cosine dissimilarity	0.233	0.516
...	...	...
Mixtures of Unigrams	0.557	0.763
Mixtures of Dirichlet-Multinomials ($k_2 = 1$)	0.722	0.871
Deep Mixtures of Dirichlet-Multinomials ($k_2 = 2$)	0.940	0.980
Deep Mixtures of Dirichlet-Multinomials ($k_2 = 3$)	0.910	0.970
Deep Mixtures of Dirichlet-Multinomials ($k_2 = 4$)	0.934	0.980
Deep Mixtures of Dirichlet-Multinomials ($k_2 = 5$)	0.925	0.975

Estimation: Gibbs sampling; the EM algorithm is strongly dependent on starting values.

General Final remarks

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

About DNNs it has been said:

"*In a world with infinite data, and infinite computational resources, there might be little need for any other technique.*" (G. Marcus, 2017)

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

About DNNs it has been said:

"*In a world with infinite data, and infinite computational resources, there might be little need for any other technique.*" (G. Marcus, 2017)

Different problems:

- interpretability

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

About DNNs it has been said:

"*In a world with infinite data, and infinite computational resources, there might be little need for any other technique.*" (G. Marcus, 2017)

Different problems:

- interpretability
- identifiability

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

About DNNs it has been said:

"*In a world with infinite data, and infinite computational resources, there might be little need for any other technique.*" (G. Marcus, 2017)

Different problems:

- interpretability
- identifiability
- computational effort

General Final remarks

Deep learning strategies seem to work surprisingly well for prediction

About DNNs it has been said:

"*In a world with infinite data, and infinite computational resources, there might be little need for any other technique.*" (G. Marcus, 2017)

Different problems:

- interpretability
- identifiability
- computational effort
- model selection against the Rashomon principle in machine learning (the multiplicity of good models)

Open issues

Model selection is an open issue

Open issues

Model selection is an open issue

DGMM offer good results for $h = 2$ and $h = 3$
Computationally intensive for $h > 3$: alternative and faster estimation methods?

Open issues

Model selection is an open issue

DGMM offer good results for $h = 2$ and $h = 3$
Computationally intensive for $h > 3$: alternative and faster estimation methods?

Deep mixtures require high n!
Old saying: '*If all a man has is a hammer, then every problem looks like a nail*'

Remember: for simple clustering problems DGMM
is like to use 'sledgehammer to crack a nut'

References

- Anderlucci L. and Viroli, C. (2019) *Going deep in clustering high-dimensional data: deep mixtures of unigrams for uncovering topics in textual data.* Under review.
- Celeux, G. and Diebolt J. (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. (2019), *Deep Gaussian Mixture Models*, Statistics and Computing