Statistics and Computing (2019) 29:43–51
https://doi.org/10.1007/s11222-017-9793-z

Deep Gaussian mixture models

Cinzia Viroli[1] · **Geoffrey J. McLachlan**[2]

Received: 18 November 2017 / Accepted: 25 November 2017 / Published online: 1 December 2017

Abstract
Deep learning is a hierarchical inference method formed by subsequent multiple layers of learning able to more efficiently describe complex relationships. In this work, deep Gaussian mixture models (DGMM) are introduced and discussed. A DGMM is a network of multiple layers of latent variables, where, at each layer, the variables follow a mixture of Gaussian distributions. Thus, the deep mixture model consists of a set of nested mixtures of linear models, which globally provide a nonlinear model able to describe the data in a very flexible way. In order to avoid overparameterized solutions, dimension reduction by factor models can be applied at each layer of the architecture, thus resulting in deep mixtures of factor analyzers.

Keywords Unsupervised classification · Mixtures of factor analyzers · Stochastic EM algorithm

1 Introduction

In the recent years, there has been an increasing interest on deep learning for supervised classification (LeCun et al. 2015). It is very difficult to give an exact definition of what it is due to its wide applicability in different contexts and formulations, but it can be thought of as a set of algorithms able to gradually learn a huge number of parameters in an architecture composed by multiple nonlinear transformations, called multilayer structure. Deep neural networks have achieved great success in supervised classification, and an important example of it is given by the so-called Facebook's DeepFace software: 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%, at the same level as the human visual capability (Taigman et al. 2014). Deep learning architectures are now widely used for speech recognition, object detection, pattern recognition, image processing and many other supervised classification tasks; for a comprehensive historical survey and its applications, see Schmidhuber (2015) and the references therein.

Despite the success of deep models for supervised tasks, there has been limited research in the machine learning and statistics community on deep methods for clustering. In this paper, we will present and discuss deep Gaussian mixtures for clustering purposes, a powerful generalization of classical Gaussian mixtures to multiple layers. Identifiability of the model is discussed, and an innovative stochastic estimation algorithm is proposed for parameter estimation. Despite the fact that in recent years research on mixture models has been intense and prolific in many directions, we will show how deep mixtures can be very useful for clustering in complex problems.

The paper is organized as follows. In the next section, classical Gaussian mixture models will be reviewed. In Sect. 3, deep Gaussian mixtures are defined and their main probabilistic properties presented. Identifiability is also discussed. In Sect. 4, dimensionally reduced deep mixtures are presented. Section 5 is devoted to the estimation algorithm for fitting the model. Experimental results on simulated and real data are presented in Sect. 6. We conclude this paper with some final remarks (Sect. 7).

✉ Cinzia Viroli
cinzia.viroli@unibo.it

Geoffrey J. McLachlan
g.mclachlan@uq.edu.au

[1] Department of Statistical Sciences, University of Bologna, Via Belle Arti 41, 40126 Bologna, Italy

[2] Department of Mathematics, University of Queensland, St. Lucia, Brisbane 4072, Australia

2 Gaussian mixture models

Finite mixture models (McLachlan and Peel 2000) have gained growing popularity in the last decades as a tool for

model-based clustering (Fraley and Raftery 2002). They are now widely used in several areas such as pattern recognition, data mining, image analysis, machine learning and many problems involving clustering and classification methods.

Let $\mathbf{y}_i$ be a p-dimensional random vector containing p quantitative variables of interest for the statistical unit ith, with $i = 1, \ldots, n$. Then, $\mathbf{y}_i$ is distributed as a Gaussian mixture model (GMM) with k components if

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

where the π_j are positive weights subject to $\sum_{j=1}^{k} \pi_j = 1$ and the μ_j, Σ_j are the parameters of the Gaussian components. Note an interesting property that will be very useful in defining our proposal: A Gaussian mixture model has a related factor analytic representation via a linear model with a certain prior probability as

$$\mathbf{y}_i = \mu_j + \Lambda_j \mathbf{z}_i + \mathbf{u}_i \quad \text{with prob. } \pi_j,$$

where $\mathbf{z}_i$ is a p-dimensional latent variable with a multivariate standard Gaussian distribution and $\mathbf{u}_i$ is an independent vector of random errors with $\mathbf{u}_i \sim N(0, \Psi_j)$, where the Ψ_j are diagonal matrices. The component-covariance matrices can then be decomposed as $\Sigma_j = \Lambda_j \Lambda_j^{\top} + \Psi_j$.

3 Deep mixture models

Deep learning is a hierarchical inference method organized in a multilayered architecture, where the subsequent multiple layers of learning are able to efficiently describe complex relationships. In the similar perspective of deep neural networks, we define a deep Gaussian mixture model (DGMM) as a network of multiple layers of latent variables. At each layer, the variables follow a mixture of Gaussian distributions. Thus, the deep mixture model consists of a set of nested mixtures of linear models that globally provide a nonlinear model able to describe the data in a very flexible way.

3.1 Definition

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

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

where $\mathbf{z}_i^{(h)} \sim N(\mathbf{0}, \mathbf{I}_p)$ $(i = 1, \ldots, n)$ and $\mathbf{u}_i^{(1)}, \ldots, \mathbf{u}_i^{(h)}$ are specific random errors that follow a Gaussian distribution with zero expectation and covariance matrices $\Psi_{s_1}^{(1)}, \ldots, \Psi_{s_h}^{(h)}$, respectively, $\eta_{s_1}^{(1)}, \ldots, \eta_{s_h}^{(h)}$ are vectors of length p, $\Lambda_{s_1}^{(1)}, \ldots, \Lambda_{s_h}^{(h)}$ are square matrices of dimension p. The specific random variables $\mathbf{u}$ are assumed to be independent of the latent variables $\mathbf{z}$. From this representation, it follows that at each layer the conditional distribution of the response variables given the regression latent variables is a (multivariate) mixture of Gaussian distributions.

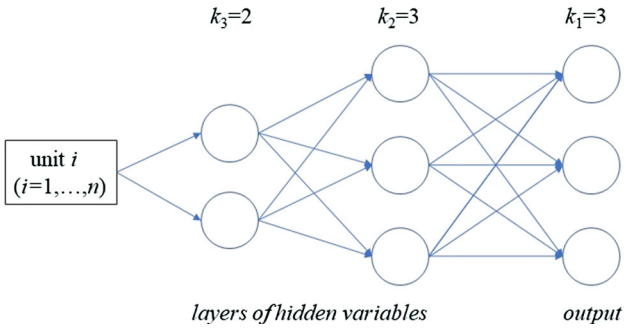

Fig. 1 Structure of a DGMM with $h = 3$ and number of layer components $k_1 = 3$, $k_2 = 3$ and $k_3 = 2$

To illustrate the DGMM, consider $h = 3$ and let the number of layer components be $k_1 = 3$, $k_2 = 3$ and $k_3 = 2$. The structure is shown in Fig. 1. Thus, at the first layer we have that the conditional distribution of the observed data given $\mathbf{z}^{(1)}$ is a mixture with three components and so on. More precisely, by considering the data as the zero layer, $\mathbf{y} = \mathbf{z}^{(0)}$, all the conditional distributions follow a first order Markov property that is $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)}, \mathbf{z}^{(l+2)}, \ldots, \mathbf{z}^{(h)}; \boldsymbol{\Theta}) = f(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)}; \boldsymbol{\Theta})$ for $l = 0, \ldots, h-1$. At each layer, we have

$$\begin{aligned}
f(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)}; \boldsymbol{\Theta}) = \sum_{i=1}^{k_{l+1}} \pi_i^{(l+1)} N(&\eta_i^{(l+1)}\\
&+ \Lambda_i^{(l+1)} \mathbf{z}^{(l+1)}, \Psi_i^{(l+1)}).
\end{aligned} \tag{2}$$

Moreover, with the DGMM with $k_1 = 3$, $k_2 = 3$ and $k_3 = 2$ will have a 'global' number of $M = 8$ sub-components ($M = \sum_{l=1}^{h} \pi_l$), but final $k = 18$ possible paths for the statistical units ($k = \prod_{l=1}^{h} \pi_l$) that share and combine the parameters of the M sub-components. Thanks to this tying, the number of parameters to be estimated is proportional to the number

of sub-components, thus reducing the computational cost to learning directly a model with $k = 18$ components.

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

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

The DGMM can be written as

$$f(\mathbf{y};\boldsymbol{\Theta}) = \sum_{s\in\Omega} \pi_s N(\mathbf{y};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s), \tag{3}$$

where

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

and

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

Thus, globally the deep mixture can be viewed as a mixture model with k components and a fewer number of parameters shared through the path. In a DGMM, not only the conditional distributions, but also the marginal distributions of the latent variables $\mathbf{z}^{(l)}$ are Gaussian mixtures. This can be established by integrating out the bottom latent variables, so that at each layer

$$f(\mathbf{z}^{(l)};\boldsymbol{\Theta}) = \sum_{\tilde{s}=(s_{l+1},\ldots,s_h)} \pi_{\tilde{s}} N(\mathbf{z}^{(l)};\tilde{\boldsymbol{\mu}}_{\tilde{s}}^{(l+1)},\tilde{\boldsymbol{\Sigma}}_{\tilde{s}}^{(l+1)}), \tag{4}$$

where $\tilde{\boldsymbol{\mu}}_{\tilde{s}}^{(l+1)} = \eta_{s_{l+1}}^{(l+1)} + \Lambda_{s_{l+1}}^{(l+1)}(\eta_{s_{l+2}}^{(l+2)} + \Lambda_{s_{l+2}}^{(l+2)}(\ldots(\eta_{s_{h-1}}^{(h-1)} + \Lambda_{s_{h-1}}^{(h-1)}\eta_h^{(h)})))$ and $\tilde{\boldsymbol{\Sigma}}_{\tilde{s}}^{(l+1)} = \Psi_{s_{l+1}}^{(l+1)} + \Lambda_{s_{l+1}}^{(l+1)}(\Lambda_{s_{l+2}}^{(l+2)}(\ldots(\Lambda_{s_h}^{(h)}\Lambda_{s_h}^{(h)\top} + \Psi_{s_h}^{(h)})\ldots)\Lambda_{s_{l+2}}^{(l+2)\top})\Lambda_{s_{l+1}}^{(l+1)\top}$.

A deep mixture model for modeling natural images has been proposed by van den Oord and Schrauwen (2014). However, this model suffers from serious identifiability issues as discussed in the next section.

3.2 Model-based clustering and identifiability

As previously observed in a DGMM, the total number of components (potentially identifying the groups) is given by the total number possible paths, k. In case the true number of groups, say k^*, is known, one could limit the estimation problem by considering only the models with $k_1 = k^*$ ($k_1 < k$) and perform clustering through the conditional distribution $f(\mathbf{y}|\mathbf{z}^{(1)};\boldsymbol{\Theta})$. This has the merit to have a nice interpretation: The remaining components of the bottom layers act as density approximations to the global non-Gaussian components. In this perspective, the model represents an automatic tool for merging mixture components (Hennig 2010; Baudry et al. 2010; Melnykov 2016) and the deep mixtures can be viewed as a special mixture of mixtures model (Li 2005).

However, in the general situation without further restrictions, the DGMM defined in the previous session suffers from serious identifiability issues related to the number of components at the different layers and the possible equivalent paths they could form. For instance, if $h = 2$, a DGMM with $k_1 = 2, k_2 = 3$ components may be indistinguishable from a DGMM with $k_1 = 3, k_2 = 2$ components, both giving a total number of possible $k = 6$ $(= k_1 \cdot k_2)$ paths. Notice that even if k^* is known and we fix $k_1 = k^*$, there is still non-identifiability for models with more than two layers.

Moreover, in all cases, there is a serious second identifiability issue related to parameter estimation.

In order to address the first issue, the we introduce an important assumption on the model dimensionality: The latent variables at the different layers have progressively decreasing dimension, $r_1, r_2, \ldots, r_h$, where $p > r_1 > r_2 > \cdots, > r_h \geq 1$. As a consequence, the parameters at the different levels will inherit different dimensionality as well. This constraint has also the merit to avoid overparameterized models, especially when p is high.

The second identifiability issue arises from the presence of latent variables, and it is similar in its nature to the identifiability issue that affects factor models. In particular, given an invertible matrix A of dimension $r \times r$, with $r < p$, the factor model $y = \eta + \Lambda z + u$, with $u \sim N(0,\Psi)$, and the transformed factor model $y = \eta + \Lambda AA^{-1}z + u$ are indistinguishable, where A is an orthogonal matrix and the factors have zero mean and identity covariance matrix. Thus, there are $r(r-1)/2$ fewer free parameters. This ambiguity can be avoided by imposing the constraint that $\Lambda^{\top}\Psi^{-1}\Lambda$ is diagonal with elements in decreasing order (see, for instance, Mardia et al. 1976).

Moving along the same lines, in the DGMM, at each layer from 1 to $h-1$, we assume that the conditional distribution of the latent variables $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l+1)};\boldsymbol{\Theta})$ has zero mean and identity covariance matrix and the same diagonality constraint on the parameters at each level.

4 Deep dimensionally reduced Gaussian mixture models

Starting from the model (1), dimension reduction is obtained by considering layers that are sequentially described by latent variables with a progressively decreasing dimension, $r_1, r_2, \ldots, r_h$, where $p > r_1 > r_2 > \ldots, > r_h \geq 1$. The dimension of the parameters in (1) changes accordingly.

Consider as an illustrative example a two-layer deep model ($h = 2$). In this case, the dimensionally reduced DGMM consists of the system of equations:

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

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

The model generalizes and encompasses several model-based clustering methods. Gaussian mixtures are trivially obtained in absence of any layer and dimension reduction. Mixtures of factor analyzers (McLachlan et al. 2003) may be considered as a one-layer deep model, where $\Psi_{s_1}^{(1)}$ are diagonal and $\mathbf{z}_i^{(1)} \sim N(\mathbf{0}, \mathbf{I}_{r_1})$. When $h = 2$ with $k_1 = 1$, $\Psi^{(1)}$ is diagonal, and $\Lambda_{s_2}^{(2)} = \{0\}$, the deep dimensionally reduced mixture coincides with mixtures of factor analyzers with common factor loadings (Baek et al. 2010) and heteroscedastic factor mixture analysis (Montanari and Viroli 2010). The so-called mixtures of factor mixture analyzers introduced by Viroli (2010) are a two-layer deep mixture with $k_1 > 1$, $\Psi_{s_1}^{(1)}$ diagonal and $\Lambda_{s_2}^{(2)} = \{0\}$. Under the constraints that $h = 2$, $\Psi_{s_1}^{(1)}$ and $\Psi_{s_2}^{(2)}$ are diagonal, the model is a deep mixture of factor analyzers (Tang et al. 2012). In this work, the authors propose to learn one layer at a time. After estimating the parameters at each layer, samples from the posterior distributions for that layer are used as data for learning the next step in a greedy layer-wise learning algorithm. Despite its computational efficiency, this multistage estimation process suffers from the uncertainty in the sampling of the latent variable generated values. A bias introduced at a layer will affect all the remaining ones and the problem grows with h, with the number of components and under unbalanced possible paths. In the next section, we will present a unified estimation algorithm for learning all the model parameters simultaneously.

5 Fitting deep Gaussian mixture models

Because of the hierarchical formulation of a deep mixture model, the EM algorithm represents the natural method for parameter estimation. The algorithm alternates between two steps, and it consists of maximizing (M-step) and calculating the conditional expectation (E-step) of the complete-data log-likelihood function given the observed data, evaluated at a given set of parameters, say $\boldsymbol{\Theta}'$:

$$
E_{\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(h)},\mathbf{s}|\mathbf{y};\boldsymbol{\Theta}'}\left[\log L_c(\boldsymbol{\Theta})\right]. \tag{5}
$$

This implies that we need to compute the posterior distributions of the latent variables given the data in the E-step of the algorithm. In contrast to the classical GMM, where this computation involves only the allocation latent variable $\mathbf{s}$ for each mixture component, in a deep mixture model the derivation of bivariate (or multivariate) posteriors is required, thus making the estimation algorithm very slow and not applicable to large data.

To further clarify this, consider the expansion of the conditional expectation in (5) as sum of specific terms. For a model with $h = 2$ layers, it takes the following form

$$
\begin{aligned}
&E_{\mathbf{z},s|\mathbf{y};\boldsymbol{\Theta}'}\left[\log L_c(\boldsymbol{\Theta})\right]\\
&= \sum_{s\in\Omega}\int f\left(\mathbf{z}^{(1)}, s|\mathbf{y};\boldsymbol{\Theta}'\right)\log f\left(\mathbf{y}|\mathbf{z}^{(1)}, s;\boldsymbol{\Theta}\right)d\mathbf{z}^{(1)}\\
&\quad + \sum_{s\in\Omega}\int\int f\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, s|\mathbf{y};\boldsymbol{\Theta}'\right)\\
&\quad \times \log f\left(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, s;\boldsymbol{\Theta}\right)d\mathbf{z}^{(1)}d\mathbf{z}^{(2)}\\
&\quad + \int f\left(\mathbf{z}^{(2)}|\mathbf{y};\boldsymbol{\Theta}'\right)\log f\left(\mathbf{z}^{(2)}\right)d\mathbf{z}^{(2)}\\
&\quad + \sum_{s\in\Omega} f\left(s|\mathbf{y};\boldsymbol{\Theta}'\right)\log f(s;\boldsymbol{\Theta}).
\end{aligned} \tag{6}
$$

A proper way to overcome these computational difficulties is to adopt a stochastic version of the EM algorithm (SEM) (Celeux and Diebolt 1985) or its Monte Carlo alternative (MCEM) (Wei and Tanner 1990). The principle underlying the handling of the latent variables is to draw observations (SEM) or samples of observations (MCEM) from the conditional density of the latent variables given the observed data, in order to simplify the computation of the E-step.

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 fixed, the variables at the upper level of the model for the current fit of parameters, where at the first layer $\mathbf{z}^{(0)} = \mathbf{y}$.

The conditional density $f(\mathbf{z}^{(l)}|\mathbf{z}^{(l-1)}, s; \boldsymbol{\Theta}')$ can be expressed as

$$f\left(\mathbf{z}^{(l)}|\mathbf{z}^{(l-1)}, s; \boldsymbol{\Theta}'\right) = \frac{f\left(\mathbf{z}^{(l-1)}|\mathbf{z}^{(l)}, s; \boldsymbol{\Theta}'\right) f\left(\mathbf{z}^{(l)}|s\right)}{f\left(\mathbf{z}^{(l-1)}|s; \boldsymbol{\Theta}'\right)}, \quad (7)$$

where the denominator does not depend on $\mathbf{z}^{(l)}$ and acts as a normalization constant, and the two terms in the numerator, conditionally on s, are Gaussian distributed according to Eqs. (4) and (2):

$$f\left(\mathbf{z}^{(l-1)}|\mathbf{z}^{(l)}, s; \boldsymbol{\Theta}'\right) = N\left(\eta_{s_l}^{(l)} + \Lambda_{s_l}^{(l)}\mathbf{z}^{(l)}, \Psi_{s_l}^{(l)}\right),$$
$$f\left(\mathbf{z}^{(l)}|s; \boldsymbol{\Theta}'\right) = N\left(\tilde{\boldsymbol{\mu}}_{s_l}^{(l+1)}, \tilde{\boldsymbol{\Sigma}}_{s_l}^{(l+1)}\right).$$

By substituting them in (7), after some simple algebra, it is possible to show that

$$f\left(\mathbf{z}^{(l)}|\mathbf{z}^{(l-1)}, s\right) = N\left(\boldsymbol{\rho}_{s_l}\left(\mathbf{z}^{(l-1)}\right), \boldsymbol{\xi}_{s_l}\right), \quad (8)$$

where

$$\boldsymbol{\rho}_{s_l}\left(\mathbf{z}^{(l-1)}\right) = \boldsymbol{\xi}_{s_l}\left(\left(\Lambda_{s_l}^{(l)}\right)^{\top}\left(\Psi_{s_l}^{(l)}\right)^{-1}\left(\mathbf{z}^{(l-1)} - \boldsymbol{\eta}_{s_l}^{(l)}\right) + \left(\tilde{\boldsymbol{\Sigma}}_{s_l}^{(l+1)}\right)^{-1}\tilde{\boldsymbol{\mu}}_{s_l}^{(l+1)}\right)$$

and

$$\boldsymbol{\xi}_{s_l} = \left(\left(\tilde{\boldsymbol{\Sigma}}_{s_l}^{(l+1)}\right)^{-1} + \left(\Lambda_{s_l}^{(l)}\right)^{\top}\left(\Psi_{s_l}^{(l)}\right)^{-1}\Lambda_{s_l}^{(l)}\right)^{-1}.$$

This is the core of the stochastic perturbation of the EM algorithm. Due to the sequential hierarchical structure of the random variable generation, the E and M steps of the algorithm can be computed for each layer. Considering the sample of n observations, at the layer $l = 1, \ldots, h$, we maximize

$$E_{\mathbf{z}^{(l)}, \mathbf{s}|\mathbf{z}^{(l-1)}; \boldsymbol{\theta}'}\left[\sum_{i=1}^{n} \log f(\mathbf{z}_i^{(l-1)}|\mathbf{z}_i^{(l)}, s; \boldsymbol{\Theta})\right] = \sum_{i=1}^{n}\int f(\mathbf{z}_i^{(l)}, s|\mathbf{z}_i^{(l-1)}; \boldsymbol{\Theta}') \log f(\mathbf{z}_i^{(l-1)}|\mathbf{z}_i^{(l)}, s; \boldsymbol{\Theta}) d\mathbf{z}_i \quad (9)$$

with respect to $\Lambda_{s_l}^{(l)}$, $\Psi_{s_l}^{(l)}$, and $\eta_{s_l}^{(l)}$. By considering $f(\mathbf{z}^{(l-1)}|\mathbf{z}^{(l)}, s) = N(\eta_{s_l}^{(l)} + \Lambda_{s_l}(l)\mathbf{z}^{(l)}, \Psi_{s_l}^{(l)})$, we can compute the score of (9) to derive the estimates for the new parameters given the provisional ones. Therefore, the complete stochastic EM algorithm can be schematized as follows. 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:

$$\hat{\Lambda}_{s_l}^{(l)} = \frac{\sum_{i=1}^{n} p\left(s|\mathbf{z}_i^{(l-1)}\right)\left(\mathbf{z}_i^{(l-1)} - \eta_{s_l}^{(l)}\right)E[\mathbf{z}_i^{(l)\top}|\mathbf{z}_i^{(l-1)}, s]E[\mathbf{z}_i^{(l)}\mathbf{z}_i^{(l)\top}|\mathbf{z}_i^{(l-1)}, s]^{-1}}{\sum_{i=1}^{n} p\left(s|\mathbf{z}_i^{(l-1)}\right)},$$

$$\hat{\Psi}_{s_l}^{(l)} = \frac{\sum_{i=1}^{n} p\left(s|\mathbf{z}_i^{(l-1)}\right)\left[(\mathbf{z}_i^{(l-1)} - \eta_{s_l})\left(\mathbf{z}_i^{(l-1)} - \eta_{s_l}\right)^{\top} - \left(\mathbf{z}_i^{(l-1)} - \eta_{s_l}\right)E\left[\mathbf{z}_i^{(l)\top}|\mathbf{z}_i^{(l-1)}, s\right]\hat{\Lambda}_{s_l}^{\top}\right]}{\sum_{i=1}^{n} p(s|\mathbf{z}_i^{(l-1)})},$$

$$\hat{\eta}_{s_l}^{(l)} = \frac{\sum_{i=1}^{n} p\left(s_l|\mathbf{z}_i^{(l-1)}\right)\left[\mathbf{z}_i^{(l-1)} - \Lambda_{s_l}E\left[\mathbf{z}_i^{(l)\top}|\mathbf{z}_i^{(l-1)}, s\right]\right]}{\sum_{i=1}^{n} p(s|\mathbf{z}_i^{(l-1)})},$$

$$\hat{\pi}_s^{(l)} = \sum_{i=1}^{n} f\left(s_l|\mathbf{y}_i\right),$$

where $f\left(s_l|\mathbf{y}_i\right)$ is the posterior probability of the allocation variable given the observed data that can be computed via Bayes' formula.

6 Simulated and real application

6.1 Smiley data

In this simulation experiment, we have generated $n = 1000$ observations from four classes in three-dimensional space. The first two variables are relevant for clustering and have been generated by using the R package `mlbench`. They are structured into two Gaussian eyes, a triangular nose and a parabolic mouth, as shown in Fig. 2. We have taken the standard deviation for eyes and mouth equal to 0.45 and 0.35, respectively. The third variable is a noise variable, independently generated from a Gaussian distribution with standard deviation 0.5.

Data have been independently generated 100 times. On each replicate, we applied DGMM with two layers with $r_1 = 2$, $r_2 = 1$, $k_1 = 4$ and k_2 ranging from 1 to 5. We fitted the models ten times in a multistart procedure, and we selected the best fit according to BIC.

We compared the DGMM results with several clustering methods by fixing the number of groups equal to the true $k = 4$ for all strategies. We fitted a Gaussian mixture model (GMM) by using the `R` package `Mclust` (Scrucca et al. 2016), skew-normal and skew-t mixture models (SNmm and STmm) by using the `R` package `EMMIXskew` (Wang et al. 2009), k-means, partition around medoids (PAM), and hierarchical clustering by Ward's method (Hclust). Clustering performance is measured by the Adjusted Rand Index (ARI) and the misclassification rate. The average of the two indicators across the 100 replicates together with their standard errors is reported in Table 1.

Figure 3 shows the box plots of the Adjusted Rand Indices and misclassification rates (m.r.'s) across the 100 replicates. The results indicate that DGMM achieves the best classification performance compared to the other methods (Table 2).

Table 1 Results on Smiley datasets: Average of Adjusted Rand Index and misclassification rates across the 100 replicated. Standard errors are reported in brackets

Method	ARI	m.r.
k-means	0.661 (0.003)	0.134 (0.001)
PAM	0.667 (0.004)	0.132 (0.001)
Hclust	0.672 (0.013)	0.141 (0.006)
GMM	0.653 (0.008)	0.178 (0.006)
SNmm	0.535 (0.006)	0.251 (0.006)
STmm	0.566 (0.006)	0.236 (0.004)
DGMM	0.788 (0.005)	0.087 (0.002)

6.2 Real data

In this section, we shall apply the deep mixture model to some benchmark data used by the clustering and classification community. We shall consider:

- *Wine Data* This dataset comes from a study (Forina et al. 1986) on 27 chemical and physical properties of three types of wine from the Piedmont region of Italy: Barolo (59), Grignolino (71) and Barbera (48). The clusters are well separated, and most clustering methods give high clustering performance on these data.
- *Olive Data* The dataset contains the percentage composition of eight fatty acids found by lipid fraction of 572 Italian olive oils (Forina and Tiscornia 1982). The data come from three regions: Southern Italy (323), Sardinia (98) and Northern Italy (151), and the aim is to distinguish between them. Also in this case, the clustering is not a very difficult task even if the clusters are not balanced.
- *Ecoli Data* Data consist of $n = 336$ proteins classified into their various cellular localization sites based on their amino acid sequences. There are $p = 7$ variables and $k = 8$ really unbalanced groups that make the clustering task rather difficult: cp cytoplasm (143),

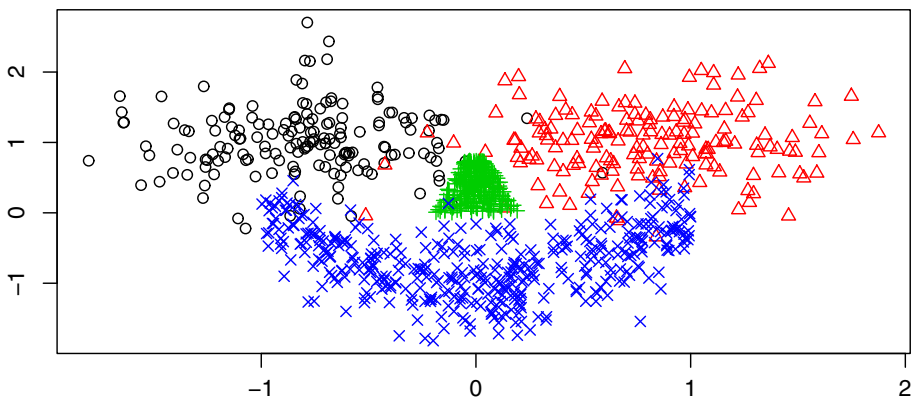

Fig. 2 Smiley Data

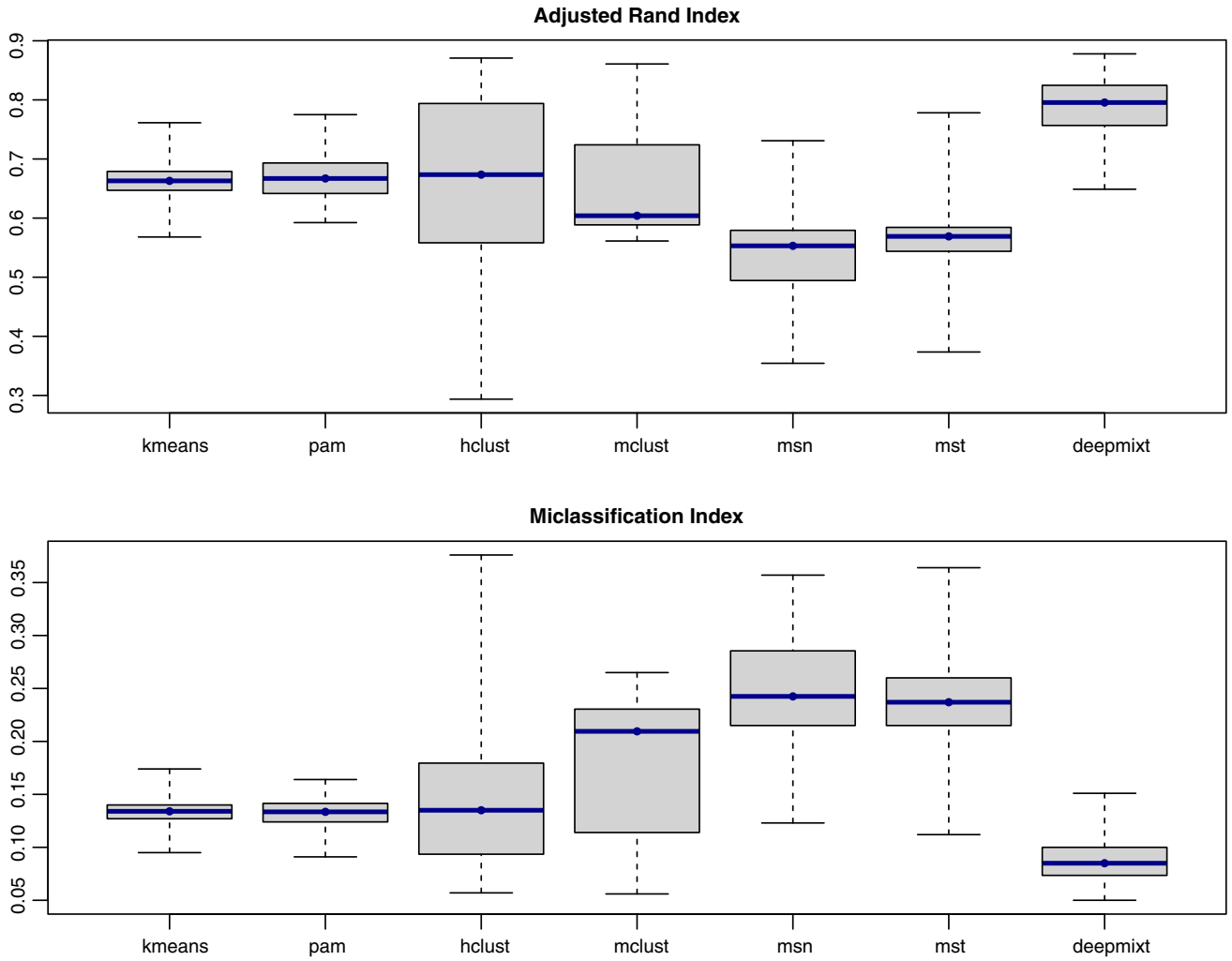

Fig. 3 Smiley data: box plots of the Adjusted Rand Indices and misclassification rates across the 100 replicates

Table 2 Results on real data: Adjusted Rand Index (ARI) and misclassification rates (m.r.)

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

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). These data are available from the UCI machine learning repository.

– *Vehicle Data* The dataset contains $k = 4$ types of vehicles: a double decker bus (218), Cheverolet van (199), Saab 9000 (217) and an Opel Manta 400 (212). The aim

is to cluster them on the basis of their silhouette represented from many different angles for a total of $p = 18$ variables. This is a difficult classification task. In particular, the bus, the van and the cars are distinguishable, but it is very difficult to distinguish between the cars. The data are taken from the R library `mlbench`.

– *Satellite Data* The data derive from multispectral, scanner images purchased from NASA by the Australian Centre for Remote Sensing. They consist of four digital images of the same scene in different spectral bands structured into 3×3 square neighborhood of pixels. Therefore, there are $p = 36$ variables. The number of images is $n = 6435$ coming from $k = 6$ groups of images: red soil (1533), cotton crop (703), gray soil (1358), damp gray soil (626), soil with vegetation stubble (707) and very damp gray soil (1508). This is notoriously a difficult clustering task not only because there are six unbalanced classes, but also because classical methods may suffer from the dimensionality $p = 36$. The data are available from the UCI machine learning repository.

On these data we compared the DGMM model 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). For all methods, we assumed the number of groups to be known. This assumption is made in order to compare the respective clustering performances. Note that in the case of an unknown number of groups, model selection for the DGMM can be done similarly to all the other mixture-based approaches by using information criteria. Therefore, we considered the DGMM with $h = 2$ and $h = 3$ layers, a number of sub-components in the hidden layers ranging from 1 to 5 (while $k_1 = k^*$) and all possible models with different dimensionality for the latent variables under the constraint $p > r_1 > \cdots > r_h \geq 1$. Moreover, we considered ten different starting points for all possible models. For the GMM, we considered all the possible submodels according to the family based on the covariance decomposition implemented in `mclust`. Finally, we fitted FMA and MFA by using the R package `MFMA` available from the first author's webpage with different starting points and different number of latent variables ranging from 1 to the maximum admissible number.

In all cases, we selected the best model according to BIC.

For the smaller dataset (*Wine*, *Olive*, *Ecoli*, *Vehicle*), the best DGMM suggested by BIC was the model with $h = 2$ layers, while $h = 3$ layers were suggested for the *Satellite* data. The *Wine* data are quite simple to classify. Most methods performed quite well. The best DGMM model was obtained with $r_1 = 3,\ r_2 = 2$ and $k_1 = 3,\ k_2 = 1$. The *Olive* data are not very well distinguished by classical methods such as k-means and hierarchical clustering, while model-based clustering strategies produce better performance. Here deep learning with $r_1 = 5,\ r_2 = 1$ and $k_1 = 3\ k_2 = 1$ suggested by BIC gives excellent results with only one misclassified unit.

The challenging aspect of a cluster analysis on *Ecoli* data is the high number of (unbalanced) classes. On these data, SNmm and STmm did not reach convergence due to their being unable to handle satisfactorily the presence of two variables that each took on only two distinct values. The best clustering method also in this case is given by the deep mixture with $r_1 = 2,\ r_2 = 1$ and $k_1 = 8,\ k_2 = 1$.

Deep mixtures performed better than the other methods also for the difficult task to distinguish between silhouettes of *vehicles* with progressively dimension reduction of $r_1 = 7,\ r_2 = 1$ and components $k_1 = 4,\ k_2 = 3$.

Finally, for the *Satellite* data a DGMM with $h = 3$ layers and $r_1 = 13,\ r_2 = 2,\ r_1 = 1$ and $k_1 = 6,\ k_2 = 2,\ k_1 = 1$ is preferred in terms of BIC. Results here are comparable with MFA with four factors; it is having slightly higher ARI, but with less corrected classified units in the total.

7 Final remarks

In this work, a deep Gaussian mixture model (DGMM) for unsupervised classification has been investigated. The model is a very general framework that encompasses classical mixtures, mixtures of mixtures models and mixture of factor analyzers as particular cases. Since DGMM is a generalization of classical model-based clustering strategies, it is guaranteed to work as well as these methods. We demonstrate the greater flexibility of DGMM with its higher complexity; for this reason, it is particularly suitable for data with large sample size.

We illustrated the model on simulated and real data. From the experimental study, we conducted the method works efficiently and it gives a good clustering performance with $h = 2$ and $h = 3$ layers where, as suggested, model choice can be undertaken according to information criteria.

References

Baek, J., McLachlan, G., Flack, L.: Mixtures of factor analyzers with common factor loadings: applications to the clustering and visualization of high-dimensional data. IEEE Trans. Pattern Anal. Mach. Intell. **32**(7), 1298–1309 (2010)

Baudry, J.-P., Raftery, A.E., Celeux, G., Lo, K., Gottardo, R.: Combining mixture components for clustering. J. Comput. Gr. Stat. **19**(2), 332–353 (2010)

Celeux, G., Diebolt, J.: The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. Comput. Stat. Q. **2**(1), 73–82 (1985)

Forina, M., Armanino, C., Castino, M., Ubigli, M.: Multivariate data analysis as a discriminating method of the origin of wines. Vitis **25**(3), 189–201 (1986)

Forina, M., Tiscornia, E.: Pattern-recognition methods in the prediction of Italian olive oil origin by their fatty-acid content. Anal. Chim. **72**(3–4), 143–155 (1982)

Fraley, C., Raftery, A.: Model-based clustering, discriminant analysis and density estimation. J. Am. Stat. Assoc. **97**, 611–631 (2002)

Hennig, C.: Methods for merging gaussian mixture components. Adv. Data Anal. Classif. **4**(1), 3–34 (2010)

LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

Li, J.: Clustering based on a multilayer mixture model. J. Comput. Gr. Stat. **14**(3), 547–568 (2005)

Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate Analysis. Academic Press, Oxford (1976)

McLachlan, G., Peel, D., Bean, R.: Modelling high-dimensional data by mixtures of factor analyzers. Comput. Stat. Data Anal. **41**(3), 379–388 (2003)

McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, Hoboken (2000)

Melnykov, V.: Merging mixture components for clustering through pairwise overlap. J. Comput. Gr. Stat. **25**(1), 66–90 (2016)

Montanari, A., Viroli, C.: Heteroscedastic factor mixture analysis. Stat. Model. **10**(4), 441–460 (2010)

Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)

Scrucca, L., Fop, M., Murphy, T.B., Raftery, A.E.: mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. R Journal **8**, 289–317 (2016)

Tang, Y., Hinton, G.E., Salakhutdinov, R.: Deep mixtures of factor analysers. In Langford, J., Pineau, J. (eds.) Proceedings of the 29th International Conference on Machine Learning (ICML-12), New York, NY, USA, pp. 505–512. ACM (2012)

Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1701–1708 (2014)

van den Oord, A., Schrauwen, B.: Factoring variations in natural images with deep gaussian mixture models. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.), Advances in Neural Information Processing Systems **27**, pp. 3518–3526. Curran Associates, Inc, Montreal, Quebec, Canada (2014)

Viroli, C.: Dimensionally reduced model-based clustering through mixtures of factor mixture analyzers. J. Classif. **27**(3), 363–388 (2010)

Wang, K., Ng, S.-K., McLachlan, G.J.: Multivariate skew t mixture models: applications to fluorescence-activated cell sorting data. In: Digital Image Computing: Techniques and Applications, 2009. DICTA'09., pp. 526–531. IEEE (2009)

Wei, G.C., Tanner, M.A.: A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. J. Am. Stat. Assoc. **85**(411), 699–704 (1990)