

# Projection Pursuit Fitting Gaussian Mixture Models\*

Mayer Aladjem

Department of Electrical and Computer Engineering  
Ben-Gurion University of the Negev  
P.O.B. 653, 84105 Beer-Sheva, Israel  
<http://www.ee.bgu.ac.il/~aladjem/>

**Abstract.** *Gaussian mixture models* (GMMs) are widely used to model complex distributions. Usually the parameters of the GMMs are determined in a *maximum likelihood* (ML) framework. A practical deficiency of ML fitting of the GMMs is the poor performance when dealing with high-dimensional data since a large sample size is needed to match the numerical accuracy that is possible in low dimensions. In this paper we propose a method for fitting the GMMs based on the *projection pursuit* (PP) strategy. By means of simulations we show that the proposed method outperforms ML fitting of the GMMs for small sizes of training sets.

## 1 Introduction

We consider the problem of modeling an  $n$ -variate probability density function  $p(\mathbf{x})$  ( $\mathbf{x} \in R^n$ ) on the basis of a training set

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}. \quad (1)$$

Here  $\mathbf{x}_i \in R^n$ ;  $i = 1, 2, \dots, N$  are data points drawn from that density. We require a normalization of the data, called *sphering* [7] (or *whitening* [4]). For the sphered  $X$  the sample covariance matrix becomes the identity matrix and the sample mean vector is a zero vector. In the remainder of the paper, all operations are performed on the sphered data.

In this paper we seek a *Gaussian mixture model* (GMM) of  $p(\mathbf{x})$ , which is a linear combination of  $M$  Gaussian densities

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^M \omega_j \phi_{\Sigma_j}(\mathbf{x} - \mathbf{m}_j). \quad (2)$$

Here,  $\omega_j$  are the mixing coefficients which are non-negative and sum to one, and  $\phi_{\Sigma_j}(\mathbf{x} - \mathbf{m}_j)$  denotes  $N(\mathbf{m}_j, \Sigma_j)$  density in the vector  $\mathbf{x}$ .

---

\* This work was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev, Israel.

The mixture model is widely applied due to its ease of interpretation by viewing each fitted Gaussian component as a distinct cluster in the data. The clusters are centered at the means  $\mathbf{m}_j$  and have geometric features (shape, volume, orientation) determined by the covariances  $\Sigma_j$ .

The problem of determining the number  $M$  of the clusters (Gaussian components) and the parameterization of  $\Sigma_j$  is known as model selection. Usually several models are considered and an appropriate one is chosen using some criterion, such as the *Bayesian information criterion* (BIC) [6].

In this paper we study GMMs with *full* (unrestricted) covariance matrices  $\Sigma_j$ , *spherical*  $\Sigma_j$  with a single parameter for the whole covariance structure and *diagonal*  $\Sigma_j$ . The use of GMMs with full covariance matrices leads to a large number of parameters for high-dimensional input vectors and presents the risk of over-fitting. Therefore  $\Sigma_j$  are often constrained to be spherical and diagonal. The latter parameterizations do not capture correlation of the variables and cannot match the numerical accuracy that is possible using unrestricted  $\Sigma_j$ . Additionally, the diagonal GMMs are strongly dependent on the rotation of the data. An attractive compromise between these parameterizations is the recently introduced mixture of latent variable models. In this paper we study a latent variable model, called a mixture of *probabilistic principal component analyses* (PPCA) [5]

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^M \omega_j \phi_{(\sigma_j^2 \mathbf{I} + \mathbf{W}_j \mathbf{W}_j^T)}(\mathbf{x} - \mathbf{m}_j), \tag{3}$$

where  $W_j$  is a  $(n \times q)$  matrix. The dimension  $q$  is called the latent factor. For  $q < n$  an unrestricted  $\Sigma_j$  (not spherical or diagonal) can be captured using only  $(1 + nq)$  parameters instead of the  $(n(n + 1)/2)$  parameters required for the full covariance matrix. Usually the parameters of the conventional and latent GMMs are determined in a *maximum likelihood* (ML) framework [4], [5].

In this paper we propose a method for fitting GMMs based on the *projection pursuit* (PP) density estimation [7], [8]. By means of simulations we show that our method outperforms the ML methods for small sizes of the training samples.

## 2 Projection Pursuit Fitting GMMs

We propose to set the parameters of GMM (2) using the *projection pursuit* (PP) density estimation [7], [8] proposed by Friedman. In Section 2.1 we summarize the original method of Friedman and in Section 2.2 we present our method for fitting GMMs.

### 2.1 Friedman’s PP Density Estimation

Friedman [7], [8] proposed to approximate the density  $p(\mathbf{x})$  by multiplication of  $K$  univariate functions  $f_k(\cdot)$

$$\hat{p}(\mathbf{x}) = \phi(\mathbf{x}) \prod_{k=1}^K f_k(\mathbf{a}_k^T \mathbf{x}), \tag{4}$$

where  $\phi(\mathbf{x})$  is  $N(\mathbf{0}, \mathbf{I})$  density in the vector  $\mathbf{x}$  (the standard normal n-variate probability density function) ,  $\mathbf{a}_k$  is an unit vector specifying a direction in  $R^n$  and  $f_k$  is

$$f_k(y) = \frac{\hat{p}_k(y)}{\phi(y)}. \tag{5}$$

Here  $\phi(y)$  denotes  $N(0, 1)$  density in the variable  $y$  and  $\hat{p}_k(y)$  is a density function along  $\mathbf{a}_k$ . Friedman approximate/estimate  $\hat{p}_k(y)$  using the Legendre polynomial expansion of the density along  $\mathbf{a}_k$ . The directional vectors  $\mathbf{a}_k$  are set by the projection pursuit strategy explained in Appendix A.

### 2.2 GMM Expansion of the PP Density Estimation

In order to expand (4) to the multivariate GMM we model  $\hat{p}_k(y)$  in (5) by a mixture of the univariate normals

$$\hat{p}_k(y) = \sum_{j=1}^{M_k} \omega_{kj} \phi_{\sigma_{kj}}(y - \mu_{kj}). \tag{6}$$

Here  $\phi_{\sigma_{kj}}(y - \mu_{kj})$  denotes  $N(\mu_{kj}, \sigma_{kj})$  density in the variable  $y$  and  $\omega_{kj}$  are the mixing coefficients for  $j = 1, 2, \dots, M_k$ . After manipulations of (5) using (6)  $f_k(y)$  becomes

$$f_k(y) = \sum_{j=1}^{M_k} \tilde{\omega}_{kj} \phi_{\tilde{\sigma}_{kj}}(y - \tilde{\mu}_{kj}), \tag{7}$$

with

$$\tilde{\omega}_{kj} = \omega_{kj} \sqrt{\frac{2\pi}{1 - \sigma_{kj}^2}} \exp\left(\frac{\mu_{kj}^2}{2(1 - \sigma_{kj}^2)}\right), \tag{8}$$

$$\tilde{\mu}_{kj} = \frac{\mu_{kj}}{1 - \sigma_{kj}^2}, \tag{9}$$

$$\tilde{\sigma}_{kj} = \frac{\sigma_{kj}}{\sqrt{1 - \sigma_{kj}^2}}. \tag{10}$$

Substituting (7) into (4), we have

$$\hat{p}(\mathbf{x}) = \phi(\mathbf{x}) \prod_{k=1}^K \left[ \sum_{j=1}^{M_k} \tilde{\omega}_{kj} \phi_{\tilde{\sigma}_{kj}}(\mathbf{a}_k^T \mathbf{x} - \tilde{\mu}_{kj}) \right]. \tag{11}$$

Finally, we employ the identity

$$\phi_{\Sigma}(\mathbf{x} - \mathbf{m}) \phi_{\sigma}(\mathbf{a}^T \mathbf{x} - \mu) = \alpha \phi_{\tilde{\Sigma}}(\mathbf{x} - \tilde{\mathbf{m}}), \tag{12}$$

with  $\mathbf{x}, \mathbf{m}, \mathbf{a} \in R^n$ ;  $\mathbf{a}^T \mathbf{a} = 1$  and

$$\tilde{\Sigma} = \Sigma - \frac{\frac{1}{\sigma^2} \Sigma \mathbf{a} \mathbf{a}^T \Sigma}{1 + \frac{1}{\sigma^2} \mathbf{a}^T \Sigma \mathbf{a}}, \tag{13}$$

$$\tilde{\mathbf{m}} = \tilde{\Sigma}\Sigma^{-1}\mathbf{m} + \frac{\mu}{\sigma^2}\tilde{\Sigma}\mathbf{a}, \tag{14}$$

$$\alpha = \frac{|\tilde{\Sigma}|^{\frac{1}{2}}}{\sqrt{2\pi\sigma}|\Sigma|^{\frac{1}{2}}} \exp\left\{\frac{\mu^2}{2\sigma^2}\left(\frac{1}{\sigma^2}\mathbf{a}^T\tilde{\Sigma}\mathbf{a} - 1\right) + \frac{1}{2}\mathbf{m}^T(\Sigma^{-1}\tilde{\Sigma}\Sigma^{-1} - \Sigma^{-1})\mathbf{m} + \frac{\mu}{\sigma^2}\mathbf{a}^T\tilde{\Sigma}\Sigma^{-1}\mathbf{m}\right\}. \tag{15}$$

The proof of formulae (12) - (15) will be included in an extended version of this paper.

The identity (12) shows that the multiplication of any  $n$ -variate normal density  $\phi_{\Sigma}(\mathbf{x} - \mathbf{m})$  by any univariate normal density  $\phi_{\sigma}(\mathbf{a}^T\mathbf{x} - \mu)$  along a directional vector  $\mathbf{a} \in R^n$  implies an  $n$ -variate normal density  $\phi_{\tilde{\Sigma}}(\mathbf{x} - \tilde{\mathbf{m}})$  scaled by a constant  $\alpha$ . After an iterative application of the identity (12) into (11), Friedman's approximation (4) becomes the form of an GMM

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^{\tilde{M}} \tilde{\omega}_j \phi_{\tilde{\Sigma}_j}(\mathbf{x} - \tilde{\mathbf{m}}_j) \tag{16}$$

having  $\tilde{M} = \prod_{i=1}^K M_i$  Gaussian components. We name (16) the GMM expansion of the PP density estimation (4). Here  $\tilde{\omega}_j$ ,  $\tilde{\Sigma}_j$  and  $\tilde{\mathbf{m}}_j$  denote the parameter values implied by the iterative application of (12) - (15) into (11). The GMM expansion (16) can be simplified, i.e. the number  $\tilde{M}$  of the Gaussian components can be reduced by suitable replacement of the similar components with a single normal. The latter is out of the scope of this paper and is subject of our current research.

### 2.3 Fitting Strategy

In the previous Section 2.2 we showed that Friedman's approximation (4) implies a GMM model (16) for the specific choice (6) of  $\hat{p}_k(y)$ . For this scenario the purpose of the PP fitting is to choose  $K$  and  $\mathbf{a}_k$  of the model (4), and to set the parameters  $M_k$ ,  $\omega_{kj}$ ,  $\mu_{kj}$  and  $\sigma_{kj}$  of the univariate mixture density  $\hat{p}_k(y)$  (6).

We compute  $K$  and  $\mathbf{a}_k$  by a method of Friedman, called *projection pursuit* (PP). We summarize the PP method in the Appendix A. The PP method computes each  $\mathbf{a}_k$  for a specific data set  $X^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_N^{(k)}\}$  (18). In the next explanation we refer to  $X^{(k)}$ .

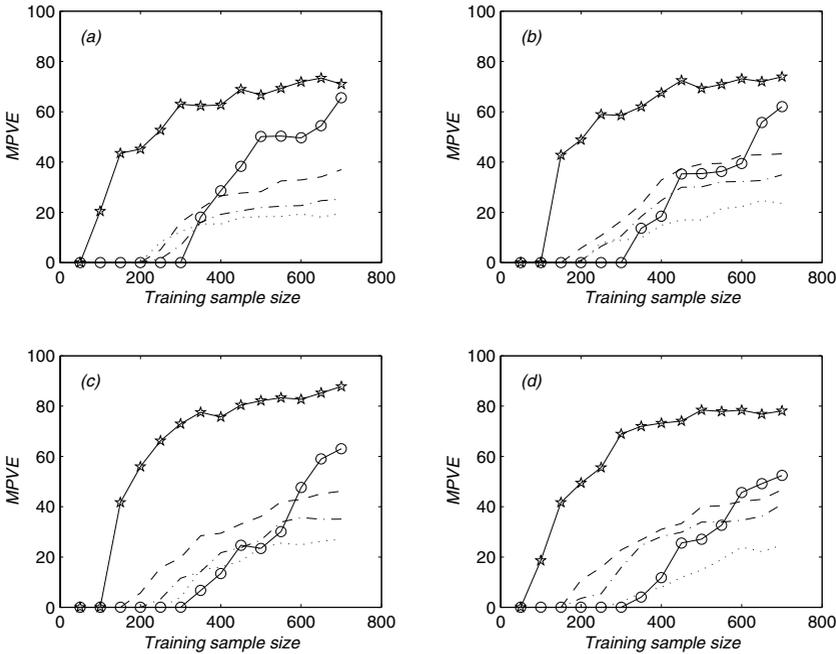
Our strategy for setting the parameters  $M_k$ ,  $\omega_{kj}$ ,  $\mu_{kj}$  and  $\sigma_{kj}$  of  $\hat{p}_k(y)$  (6) is based on the *maximum likelihood* (ML) technique [4, pages 65-72] and the *Bayesian information criterion* (BIC) [6]. In summary, it is as follows. First we project the data points  $\mathbf{x}_i^{(k)} \in X^{(k)}$ ,  $i = 1, 2, \dots, N$  onto  $\mathbf{a}_k$ . We denote the projections  $y_i = \mathbf{a}_k^T \mathbf{x}_i^{(k)}$ . Then for  $M_k = 1, 2, \dots, M_{max}$  we fit  $\hat{p}_k(y)$  to the data points  $y_i, i = 1, 2, \dots, N$  by the ML technique. The maximal number  $M_{max}$  of the components of  $\hat{p}_k(y_i)$  is set by the user (in our experiments described in Section 3 we set  $M_{max} = 10$ ). For each  $M_k$  we compute the value of the *log-likelihood function*  $L_{M_k}$  ( $L_{M_k} = \sum_{i=1}^N \ln \hat{p}_k(y_i)$ ) at the maximized values of the

parameter  $\omega_{kj}$ ,  $\mu_{kj}$  and  $\sigma_{kj}$ . Then we compute the values  $BIC_{M_k} = 2L_{M_k} - (3M_k - 1)\ln(N)$  [6] and plot them for  $M_k = 1, 2, \dots, M_{max}$ . Finally, following [6], we select the model having the number  $M_k$  giving rise to a decisive first local maximum of the BIC values. In the case of monotonically decreasing BIC values we drop  $\mathbf{a}_k$  from Friedman’s approximation (4).

### 3 Comparative Studies

In this section, we compare the performance of the *maximum likelihood* (ML) [4], [5] and the *projection pursuit* (PP) (Section 2) fittings the GMMs. We study a wide spectrum of situations in terms of the size  $N$  of the training samples (1) drawn from 15-dimensional trimodal densities  $p_{\mathbf{I}\mathbf{K}}(\mathbf{x})$ ,  $p_{\mathbf{J}\mathbf{K}}(\mathbf{x})$ ,  $p_{\mathbf{I}\mathbf{J}}(\mathbf{x})$  and  $p_{\mathbf{I}\mathbf{J}\mathbf{K}}(\mathbf{x})$  set in Appendix B. We ran experiments for  $N = 50, 100, 150, \dots, 700$ .

An experiment for a given combination of particular setting, density function and size of the training sample consisted of 10 replications of the following procedure. We generated training data of size  $N$  from an appropriate distribution. Then we normalized (sphered [7]) the data and rotated the coordinate system randomly in order not to favor the rotating dependent GMMs. Using this data



**Fig. 1.** The training sample size versus the mean percentage of variance explained (MPVE) for our method (\*), GMMs with full (o), diagonal (- -) and spherical (...) covariance matrices, and the mixture of PPCAs [5] (- -). Comparison on the 15-dimensional data sets drawn from densities: a)  $p_{\mathbf{I}\mathbf{J}\mathbf{K}}(\mathbf{x})$ , b)  $p_{\mathbf{I}\mathbf{K}}(\mathbf{x})$ , c)  $p_{\mathbf{J}\mathbf{K}}(\mathbf{x})$ , d)  $p_{\mathbf{I}\mathbf{J}}(\mathbf{x})$ .

we fitted the GMMs by our method (Section 2) and PPCA [5]. The number  $q$  of the latent factors for PPCA and the number  $M$  of the components of the GMMs were varied  $q = 1, 2, 3, 4$  and  $M = 3, 4, 5, 6, 7, 8$ . For the same data we fitted the GMMs with full, diagonal and spherical covariance matrices by the ML technique. The EM algorithm [4, pages 65-72] was used as a local optimizer of the likelihood of the GMMs for the training data. A k-mean clustering technique [4, page 187] was used to set the starting GMM parameter values for the EM algorithm. In order not to favor our method the starting point for the optimization (17) was set by the k-mean clustering, as well. The number  $M$  of the components of the GMM (2) was set  $M = 3$  for the GMMs with full covariance matrices, and  $M$  was varied for GMMs with diagonal ( $M = 3, 4, 5, 6, 7, 8$ ) and spherical ( $M = 3, 4, \dots, 14$ ) covariance matrices. For each GMM a performance criterion named the *percentage of variance explained (PVE)* (Appendix C) was computed. Finally we calculated the mean of the PVE values over the 10 replications and denoted it by *mean percentage of variance explained (MPVE)*. In Fig. 1 we show the largest *MPVE* values among the variation of  $q$  and  $M$ .

The results in Fig. 1 show that our method ( $\star$ ) outperforms all the methods for  $N = 150 - 700$ . We succeeded to explain 50-80% of the variance, while the other methods explain 0-40% only. The GMMs with full covariance matrices ( $\circ$ ) were highly sensitive to over-fitting ( $MPVE \approx 0\%$ ) for  $N = 50 - 300$ . The mixture of PPCAs (- -) was better than sphered and diagonal GMMs for all variations of  $N$ , and better than full GMMs for  $N < 400$ . The latter results are consistent with the observations in [10].

## 4 Summary and Conclusion

We proposed a method for fitting GMMs based on the *projection pursuit* (PP) strategy proposed by Friedman [7]. The results obtained by means of simulations (Section 3) show that the PP strategy outperforms the *maximal likelihood* (ML) fitting of the GMMs for small sizes of the training sets.

In Section 2.2 we showed that the PP density estimation implies a GMM model for a specific setting of the Friedman's approximation. The formulae (12)-(15) derived allow us to set the parameters of the GMM implied by the PP estimation. This allows simple exact computation of the performance (*PVE*, Appendix C) in the simulations with normal mixture densities. The exact calculation of the *PVE* of the GMMs is carried out by direct matrix computations instead of a complicated Monte-Carlo evaluation of the  $n$ -fold integrals of *PVE* provided in [8] and [9]. The exact computation of the *PVE* is possible for a high-dimensional input space  $n \gg 10$ , while the Monte-Carlo evaluation of the *PVE* is restricted to  $n < 10$ .

In our previous works we employed the PP strategy successfully in the discriminant analysis [1], [2] and for training neural networks for classification [3]. In this paper we showed that the PP strategy is an attractive choice for fitting GMMs using small sizes of the training sets.

## Appendices

### A Projection Pursuit

#### A.1 Computation the Directions $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$

Following Friedman [7] we choose  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$  by solving a sequence of *nonlinear programming* (NP) problems

$$\begin{aligned} \mathbf{a}_k &= \arg \max_{\mathbf{a}} \{I(\mathbf{a}|X^{(k)})\} \text{ for } k = 1, 2, \dots, K \\ &\text{subject to } \mathbf{a}^T \mathbf{a} = 1. \end{aligned} \tag{17}$$

Here  $I(\mathbf{a}|X^{(k)})$  is an objective function, named *projection pursuit* (PP) index (see Section A.2). It depends implicitly on a specific data set, denoted by

$$X^{(k)} = \{ \mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_N^{(k)} \}. \tag{18}$$

Here,  $\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_N^{(k)}$  are n-dimensional vectors. The data sets  $X^{(k)}, k = 1, 2, \dots, K$  are constructed in a sequential way, explained in Section A.3.

For solving the *nonlinear programming* (NP) problems (17) we employ a hybrid optimization strategy proposed in [11].

#### A.2 PP Index

The PP index  $I(\mathbf{a}|X^{(k)})$  is defined in the following way. We project the data points  $\mathbf{x}_i^{(k)} \in X^{(k)}$  onto  $\mathbf{a}$  (an arbitrary n-dimensional vector having unit length) and obtain the projections  $y_i^{(k)} = \mathbf{a}^T \mathbf{x}_i^{(k)}$ . Obviously the shape of the density of these projections depends on the direction of  $\mathbf{a}$ . Friedman [7] defined the PP index as a measure of the departure of that density from  $N(0,1)$ . He constructed the PP index based on an  $J$ -term Legendre polynomial expansion of the  $L_2$  distance between the densities [7, pages 250-252]

$$I(\mathbf{a}|X^{(k)}) = \sum_{j=1}^J \frac{2j+1}{2} \left[ \frac{1}{N} \sum_{i=1}^N P_j(r_i^{(k)}) \right]^2, \tag{19}$$

with

$$r_i^{(k)} = 2\Phi(y_i^{(k)}) - 1. \tag{20}$$

Here  $\Phi$  denotes the standard normal (cumulative) distribution function and the Legendre polynomials  $P_j$  are defined as follows:

$$\begin{aligned} P_0(r) &= 1, \quad P_1(r) = r, \quad P_2(r) = \frac{1}{2}(3r^2 - 1), \\ P_j(r) &= \frac{1}{j} \{ (2j-1)rP_{j-1}(r) - (j-1)P_{j-2}(r) \}, j = 3, 4, \dots \end{aligned} \tag{21}$$

If the projected density onto  $\mathbf{a}$  is  $N(0,1)$  then PP index (19) achieves its minimum value ( $\approx 0$ ). The solution of the NP problem (17) defines direction  $\mathbf{a}_k$  which manifests non-normal projected density as much as possible.

Following Friedman [7] we set  $J = 6$  in (19). We used the value of  $I(\mathbf{a}_k|X^{(k)})$  to set the number  $K$  in the approximation (4). If  $I(\mathbf{a}_k|X^{(k)}) < \epsilon$  then we dropped  $\mathbf{a}_k$  from (4). In our experiments in Section 3 we set  $\epsilon = 0.0001$ .

### A.3 Computation the Data Sets $X^{(1)}, X^{(2)}, \dots, X^{(K)}$

Following Friedman [7] we compute the data sets  $X^{(1)}, X^{(2)}, \dots, X^{(K)}$  by the following successive transformation of the original training data set  $X$  (1).

**For**  $k = 1, 2, \dots, K$

We assign  $X^{(k)} = X$  ( $X$  is the original data set (1) for  $k = 1$ ).

We compute  $\mathbf{a}_k$  solving (17).

We transform  $X^{(k)}$  into  $\tilde{X}^{(k)}$ . We require  $\tilde{X}^{(k)}$  to have  $N(0,1)$  onto  $\mathbf{a}_k$ , and the same data structure as  $X^{(k)}$  into an  $(n - 1)$ -dimensional subspace orthogonal to  $\mathbf{a}_k$ . By this means we eliminate the maximum value of the PP index for  $\tilde{X}^{(k)}$  at the point  $\mathbf{a}_k$  ( $I(\mathbf{a}_k|\tilde{X}^{(k)}) = 0$ ). The transformed data  $\tilde{X}^{(k)}$  is computed by a method [7, pages 253-254], called *structure removal*.

We assign  $X$  to be the transformed data  $\tilde{X}^{(k)}$  ( $X = \tilde{X}^{(k)}$ ) and continue.

**End**

## B Density Used to Generate the Training Data Sets

We generated training data sets from 15-dimensional density functions

$$\begin{aligned}
 p_{\mathbf{IK}}(\mathbf{x}) &= [\sum_{j=1}^3 \alpha_j g_{\mathbf{I}_j}(x_1, x_2) g_{\mathbf{K}_j}(x_3, x_4)] \prod_{k=5}^{15} \phi(x_k), \\
 p_{\mathbf{JK}}(\mathbf{x}) &= [\sum_{j=1}^3 \alpha_j g_{\mathbf{J}_j}(x_1, x_2) g_{\mathbf{K}_j}(x_3, x_4)] \prod_{k=5}^{15} \phi(x_k), \\
 p_{\mathbf{IJ}}(\mathbf{x}) &= [\sum_{j=1}^3 \alpha_j g_{\mathbf{I}_j}(x_1, x_2) g_{\mathbf{J}_j}(x_3, x_4)] \prod_{k=5}^{15} \phi(x_k), \\
 p_{\mathbf{IJK}}(\mathbf{x}) &= [\sum_{j=1}^3 \alpha_j g_{\mathbf{I}_j}(x_1, x_2) g_{\mathbf{J}_j}(x_3, x_4) g_{\mathbf{K}_j}(x_5, x_6)] \prod_{k=7}^{15} \phi(x_k).
 \end{aligned}$$

Here  $\mathbf{x} = (x_1, x_2, \dots, x_{15})^T$ ,  $\phi(x_k)$  is  $N(0,1)$  density in the variable  $x_k$  and  $g_{\mathbf{I}_j}(x_1, x_2)$ ,  $g_{\mathbf{J}_j}(x_3, x_4)$ ,  $g_{\mathbf{K}_j}(x_5, x_6)$  for  $j = 1, 2, 3$  are bivariate normal densities from [12, Table 1]. We set the mixing coefficients  $\alpha_1 = \alpha_2 = \frac{9}{20}$  and  $\alpha_3 = \frac{1}{10}$ . The structure of  $p_{\mathbf{IK}}(\mathbf{x})$ ,  $p_{\mathbf{JK}}(\mathbf{x})$  and  $p_{\mathbf{IJ}}(\mathbf{x})$  lies in the first four variables, and the structure of  $p_{\mathbf{IJK}}(\mathbf{x})$  lies in the first six variables. The remaining variables only add noise (variables having  $N(0,1)$  densities). Note that the data sets drawn from these densities were normalized (sphered [7]), and randomly rotated in the runs discussed in Section 3.

## C Percentage of Variance Explained (PVE)

In Section 3 we evaluated the performance of the GMMs by *percentage of variance explained*  $PVE = 100(1 - \frac{ISE}{var})$  [8], where  $ISE = \int_{R^n} (\hat{p}(\mathbf{x}) - p(\mathbf{x}))^2 d\mathbf{x}$  is the *integrated squared error* of the GMM  $\hat{p}(\mathbf{x})$  and  $var = \int_{R^n} (p(\mathbf{x}) - \frac{1}{vol(E)})^2 d\mathbf{x}$  is a normalization. Here  $p(\mathbf{x})$  is the true underlying density and  $vol(E)$  denotes the volume of a region  $E$  in space containing most of the mass of  $p(\mathbf{x})$ . We set  $E = \{(-5 < x_i < 5), i = 1, 2, \dots, 15\}$ . We employed a closed-form solution of the  $n$ -fold integrals  $ISE$  and  $var$ , which is available within the class of the normal mixture densities [13]. The latter allows us to compute the  $PVE$  for the densities  $p_{IJK}(\mathbf{x})$ ,  $p_{IK}(\mathbf{x})$ ,  $p_{JK}(\mathbf{x})$  and  $p_{IJ}(\mathbf{x})$  (Appendix B) exactly by direct matrix calculations. The formulae for the latter calculations will be included in an extended version of this paper.

## References

1. Aladjem, M. E.: Linear discriminant analysis for two-classes via removal of classification structure. *IEEE Trans. Pattern Anal. Mach. Intell.* **19** (1997) 187–192 [401](#)
2. Aladjem, M. E.: Non-parametric discriminant analysis via recursive optimization of Patrick-Fisher distance. *IEEE Trans. on Syst., Man, Cybern.* **28B** (1998) 292–299 [401](#)
3. Aladjem, M. E.: Recursive training of neural networks for classification. *IEEE Trans. on Neural Networks.* **11** (2000) 488–503 [401](#)
4. Bishop, C. M.: *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York (1995) [396](#), [397](#), [399](#), [400](#), [401](#)
5. Bishop, C. M.: Latent variable models. In: Jordan, M. I. (ed.): *Learning in Graphical Models*. The MIT Press, London (1999) 371–403 [397](#), [400](#), [401](#)
6. Fraley, C., Raftery, A. E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal.* **41** (1998) 578–588 [397](#), [399](#), [400](#)
7. Friedman, J. H.: Exploratory projection pursuit. *Journal of the American Statistical Association.* **82** (1987) 249–266 [396](#), [397](#), [400](#), [401](#), [402](#), [403](#)
8. Friedman, J. H., Stuetzle, W., Schroeder, A.: Projection pursuit density estimation. *Journal of the American Statistical Association.* **79** (1984) 599–608 [397](#), [401](#), [404](#)
9. Hwang, J. N., Lay, S. R., Lippman, A.: Nonparametric multivariate density estimation: A comparative study. *IEEE Trans. on Signal Processing.* **42** (1994) 2795–2810 [401](#)
10. Moerland, P.: A comparison of mixture models for density estimation. In: *Proceedings of the International Conference on Artificial Neural Networks* (1999) [401](#)
11. Sun, J.: Some practical aspects of exploratory projected pursuit. *SIAM J. Sci. Comput.* **14** (1993) 68–80 [402](#)
12. Wand, M. P., Jones, M. C.: Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association.* **88** (1993) 520–528 [403](#)
13. Wand, M. P., Jones, M. C.: *Kernel Smoothing*. Charman & Hall/CRC (1995) [404](#)