

Projection Pursuit Mixture Density Estimation

Mayer Aladjem

Department of Electrical and Computer Engineering,
Ben-Gurion University of the Negev,
P.O.B. 653, 84105 Beer-Sheva, Israel

Abstract

In this paper we seek a *Gaussian mixture model* (GMM) of an n -variate probability density function. Usually the parameters of GMMs are determined in the original n -dimensional space by optimizing a *maximum likelihood* (ML) criterion. A practical deficiency of this method of fitting GMMs is its poor performance when dealing with high-dimensional data since a large sample size is needed to match the accuracy that is possible in low dimensions. We propose a method for fitting the GMM based on the *projection pursuit* (PP) strategy. This GMM is highly constrained and hence its ability to model structure in subspaces is enhanced, compared to a direct ML fitting of a GMM in high dimensions. Our method is closely related to recently developed *independent factor analysis* (IFA) mixture models. The comparisons with ML fitting of GMM in n -dimensions and IFA mixtures show that the proposed method is an attractive choice for fitting GMMs using small sizes of training sets.

Index Terms—Multivariate density estimation, Gaussian mixture models, Projection pursuit, Radial basis functions, Latent variable models, Probabilistic principal component analysis, Independent component analysis, Independent factor analysis, Blind source separation, Small sample size.

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. In this paper we model the density by a *projection pursuit* (PP) method proposed in [11]. The computational complexity of this method can be reduced by preliminary normalization of the data, called *sphering* [11, Section 2], [29]. In the remainder of the paper, all operations are performed on the *sphered* data. For the sphered X (1) the sample covariance matrix becomes the identity matrix and the sample mean vector is a zero vector. This implies that, for any directional vector \mathbf{a} having unit length ($\mathbf{a}^T \mathbf{a} = 1$) the projections $\mathbf{a}^T \mathbf{x}_i$ of the sphered \mathbf{x}_i have unit sample variance, which frees the PP algorithm from having to standardize the density estimates during the numerical optimization [11, Sections 2 and 5].

In this paper we seek a *Gaussian mixture model* (GMM) [22], [28] 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, ω_j are the mixing coefficients which are non-negative and sum to one, and $\phi_{\Sigma_j}(\mathbf{x} - \mathbf{m}_j)$ denotes the $N(\mathbf{m}_j, \Sigma_j)$ density in the vector \mathbf{x} .

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 Σ_j .

The problem of determining the number M of the clusters (Gaussian components) and the parameterization of Σ_j is known as *model selection*. Usually several models are considered and an appropriate one is chosen using some criterion [7], [8], [9],[10], [13]. The covariance matrices Σ_j are taken to be *full* (unrestricted), *diagonal* or *spherical* in the conventional GMMs [6], [22], [23], [28]. Recently the eigenvalue decomposition of Σ_j [9], [10] and the *latent variable models* [5], [7], [14], [16], [20], [21], [27] have been exploited to extend the conventional GMM. In Sections 4 and 5 we study some of the latter models. Usually the parameters of the conventional and latent GMMs are determined in the original n -dimensional space by optimizing

a *maximum likelihood* (ML) criterion. This leads to a large number of adjusted parameters and presents the risk of over-fitting for high-dimensional input space.

In this paper we propose a method for fitting GMMs based on *projection pursuit* (PP) density estimation [11, Section 4], [12]. We use the directions generated by the PP and then fit a mixture of univariate Gaussians onto the data. Thus we reduce the number of adjustable parameters and the risk of over-fitting significantly.

The paper is organized as follows. Section 2 introduces the PP density estimation [11], [12]. In Section 3 we present our method for fitting GMMs. Proof of an important identity is given in Appendix B. The results of a comparative study of our method, conventional GMMs and some latent variable models are discussed in Sections 4 and 5. The experimental results show that the proposed method is an attractive choice for fitting GMMs using small sizes of training sets. In Section 5 the relation of our method to the recently developed *independent factor analysis* IFA [5] is analyzed and illustrated experimentally. Some preliminary results of our work were presented in [4]. This paper contains a more thorough analysis and more complete results.

2 Projection Pursuit Density Estimation

Friedman, Stuetzle and Schroeder [12] proposed to estimate the density $p(\mathbf{x})$ by multiplication of K univariate *augmenting functions* $f_k(\cdot)$

$$\hat{p}(\mathbf{x}) = \phi(\mathbf{x}) \prod_{k=1}^K f_k(\mathbf{a}_k^T \mathbf{x}), \quad (3)$$

where $\phi(\mathbf{x})$ is an initial multivariate density estimate of $p(\mathbf{x})$ and \mathbf{a}_k for $k = 1, \dots, K$ are vectors specifying different directions in R^n . Usually $\phi(\mathbf{x})$ is taken to be $N(\mathbf{0}, \mathbf{1})$ (the standard normal n-variate probability density function) for *sphered X* (1).

The computation of the direction vectors \mathbf{a}_k and the augmenting functions $f_k(\cdot)$ has been discussed by many authors [11], [12], [18], [19, Section 9].

In this paper we compute \mathbf{a}_k and $f_k(\cdot)$ by the *projection pursuit* (PP) method of Friedman [11, Section 4]. We choose $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$ by solving a sequence of *nonlinear*

programming (NP) problems

$$\begin{aligned} \mathbf{a}_k &= \arg \max_{\mathbf{a}} \left\{ I(\mathbf{a}|X^{(k)}) \right\} \text{ for } k = 1, 2, \dots, K \\ &\text{subject to } \mathbf{a}^T \mathbf{a} = 1. \end{aligned} \tag{4}$$

Here $I(\mathbf{a}|X^{(k)})$ is an objective function, named the *PP index*. It depends implicitly on a specific data set, denoted by $X^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_N^{(k)}\}$. 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 below. For solving the NP problems (4) we employ a *hybrid optimization strategy* proposed in [11, Section 5] and extended in [26, Section 3].

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 $y_i^{(k)} = \mathbf{a}^T \mathbf{x}_i^{(k)}$. Then we approximate the density of these projections. We denote the obtained approximation $\hat{p}_{\mathbf{a}}^{(k)}(y)$. In principle any appropriate method for univariate density estimation can be used for $\hat{p}_{\mathbf{a}}^{(k)}(y)$. In [11] a J -term ($4 \geq J \leq 8$) Legendre polynomial approximation is used (we set $J = 6$ in our experiments). Obviously the shape of $\hat{p}_{\mathbf{a}}^{(k)}(y)$ depends on the direction of \mathbf{a} . Friedman [11] defined the PP index as the L_2 distance between $\hat{p}_{\mathbf{a}}^{(k)}(y)$ and $N(0,1)$. Thus the solution of the NP problem (4) defines \mathbf{a}_k for $k = 1, 2, \dots, K$ which manifest nongaussian projected densities $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$ as much as possible. Note that the directions \mathbf{a}_k are not constrained to be orthogonal.

The data sets $X^{(1)}, X^{(2)}, \dots, X^{(K)}$ are computed by the following successive transformations of the original training data set X (1). First we set $X^{(1)} = X$ and compute \mathbf{a}_1 solving NP (4) for $k = 1$. Then we transform $X^{(1)}$ into $X^{(2)}$. We require $X^{(2)}$ to have $N(0,1)$ onto \mathbf{a}_1 and the same data structure as $X^{(1)}$ into a $(n - 1)$ -dimensional subspace orthogonal to \mathbf{a}_1 . By this means we eliminate the maximum value of the PP index for $X^{(2)}$ at the point \mathbf{a}_1 ($I(\mathbf{a}_1|X^{(2)}) = 0$). $X^{(2)}$ is computed by a method called *structure removal* [11, Section 3]. We repeat this procedure K times. In [11, Section 7], [12, Section 5], [26, Section 3]) different model selection procedures for setting the number K have been proposed.

Finally, the augmenting function $f_k(\cdot)$ is

$$f_k(y) = \frac{\hat{p}_{\mathbf{a}_k}^{(k)}(y)}{\phi(y)}, \quad (5)$$

where $\phi(y)$ denotes $N(0,1)$ density in the variable y . The formula (5) is derived in [11, Section 4]. For completeness we give this derivation in Appendix A of this paper.

Thus we perform density estimation by finding the most nongaussian projections of the data. This is the same thing that is done in the recently developed *independent component analysis* (ICA) [13, Section 14.6], [14], [15, Section 8.5]. The relation to ICA is discussed in Section 5.

The PP density estimation method [11, Section 4] has been shown to possess excellent properties in simulations having $n = 2-5$ [18]. The computational complexity of the numerical (Monte-Carlo) evaluation of the *integrated squared error* (ISE) was the principal difficulty in running simulations for $n > 5$.

In the next section we show that the PP density estimation implies a GMM model for a specific setting of the augmenting functions (5). We employed the latter result for a closed-form solution of the ISE (Appendix C), which allowed us to run simulations for $n = 15$ (Section 4 and 5), which was impossible using the Monte Carlo evaluation used previously [12], [18].

Moreover our proposal allows the use of PP density estimation in important applications which currently employ GMMs, for example *blind source separation* (BSS) [5], the initialization of *radial basis function* networks [6] and other applications.

3 GMM Expansion of the PP Density Estimation

We propose to model $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$ in (5) by a mixture of univariate Gaussians

$$\hat{p}_{\mathbf{a}_k}^{(k)}(y) = \sum_{i=1}^{M_k} \omega_{ki} \phi_{\sigma_{ki}}(y - \mu_{ki}). \quad (6)$$

Here $\phi_{\sigma_{ki}}(y - \mu_{ki})$ denotes $N(\mu_{ki}, \sigma_{ki})$ density in the variable y and ω_{ki} are the mixing coefficients for $i = 1, 2, \dots, M_k$. After manipulations of (5) using (6) $f_k(y)$ takes the form of a Gaussian *radial basis function* (RBF) expansion [6, Section 5]

$$f_k(y) = \sum_{i=1}^{M_k} \tilde{\omega}_{ki} \phi_{\tilde{\sigma}_{ki}}(y - \tilde{\mu}_{ki}), \quad (7)$$

with

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

$$\tilde{\mu}_{ki} = \frac{\mu_{ki}}{1 - \sigma_{ki}^2}, \quad \tilde{\sigma}_{ki} = \frac{\sigma_{ki}}{\sqrt{1 - \sigma_{ki}^2}}. \quad (9)$$

Note that σ_{ki}^2 must be constrained to be $\sigma_{ki}^2 < 1$. The preliminary *sphering* of the data X (1) implies $\sigma_{ki}^2 < 1$ in most situations. In some special cases (long tails of $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$) a correction (enlarging the number M_k of the Gaussian components) is needed.

The formulas (8) and (9) for the computation of the RBF parameters $\tilde{\omega}_{ki}$, $\tilde{\mu}_{ki}$ and $\tilde{\sigma}_{ki}$ are for the specific augmenting functions $f_k(\cdot)$ (5) proposed in [11, Section 4]. In Section 2 we mentioned that other methods [11], [12], [18], [19, Section 9] for computation of $f_k(\cdot)$ exist. For these $f_k(\cdot)$ the values of $\tilde{\omega}_{ki}$, $\tilde{\mu}_{ki}$ and $\tilde{\sigma}_{ki}$ in (7) can be computed by a method for RBF function approximation [6, Section 5].

Substituting (7) into (3), we have

$$\hat{p}(\mathbf{x}) = \phi(\mathbf{x}) \prod_{k=1}^K \left[\sum_{i=1}^{M_k} \tilde{\omega}_{ki} \phi_{\tilde{\sigma}_{ki}}(\mathbf{a}_k^T \mathbf{x} - \tilde{\mu}_{ki}) \right]. \quad (10)$$

Finally, we employ the identity

$$\phi_{\Sigma}(\mathbf{x} - \mathbf{m}) \phi_{\sigma}(\mathbf{a}^T \mathbf{x} - \mu) = \alpha \phi_{\tilde{\Sigma}}(\mathbf{x} - \tilde{\mathbf{m}}), \quad (11)$$

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

$$\alpha = \frac{|\tilde{\Sigma}|^{\frac{1}{2}}}{\sqrt{2\pi\sigma} |\Sigma|^{\frac{1}{2}}} \exp\left\{ \frac{\mu^2}{2\sigma^2} \left(\frac{1}{\sigma^2} \mathbf{a}^T \tilde{\Sigma} \mathbf{a} - 1 \right) + \frac{1}{2} \mathbf{m}^T (\Sigma^{-1} \tilde{\Sigma} \Sigma^{-1} - \Sigma^{-1}) \mathbf{m} + \frac{\mu}{\sigma^2} \mathbf{a}^T \tilde{\Sigma} \Sigma^{-1} \mathbf{m} \right\}, \quad (12)$$

$$\tilde{\Sigma} = \Sigma - \frac{\frac{1}{\sigma^2} \Sigma \mathbf{a} \mathbf{a}^T \Sigma}{1 + \frac{1}{\sigma^2} \mathbf{a}^T \Sigma \mathbf{a}}, \quad \tilde{\mathbf{m}} = \tilde{\Sigma} \Sigma^{-1} \mathbf{m} + \frac{\mu}{\sigma^2} \tilde{\Sigma} \mathbf{a}. \quad (13)$$

The proof of formulas (11) - (13) is in Appendix B.

The identity (11) 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 α . After an iterative application of the identity (11) into (10), PP approximation (3) becomes the form of a GMM

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^{\tilde{M}} \tilde{\omega}_j \phi_{\tilde{\Sigma}_j}(\mathbf{x} - \tilde{\mathbf{m}}_j), \quad (14)$$

having $\tilde{M} = \prod_{k=1}^K M_k$ Gaussian components. We name (14) the GMM expansion of the PP density estimation (3). Here $\tilde{\omega}_j$, $\tilde{\Sigma}_j$ and $\tilde{\mathbf{m}}_j$ denote the parameter values implied by the iterative application of (11) - (13) into (10). The GMM expansion (14) can be simplified, i.e. the number \tilde{M} of the Gaussian components can be reduced by an *iterative pairwise replacement algorithm* (IPRA) proposed in [25]. IPRA refits the parameters of the mixture density locally, thus making it possible to simplify (14) by iteratively collapsing pairs of similar components.

For this scenario we must set the parameters M_k , ω_{kj} , μ_{kj} and σ_{kj} of the univariate mixture densities $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$ (6) for $k = 1, \dots, K$. Any method for univariate GMM density estimation can be used for this purpose. In our experiments (Sections 4 and 5) we computed the values of ω_{kj} , μ_{kj} and σ_{kj} by an *expectation-maximization* (EM) algorithm [23, Section 3] and selected M_k by the *Bayesian information criterion* (BIC) [9], [10]. In summary, 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 $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$ to the data points $y_i, i = 1, 2, \dots, N$ by the ML technique [23, Section 3]. The maximal number M_{max} of the components of $\hat{p}_{\mathbf{a}_k}^{(k)}(y)$ is set by the user (we set $M_{max} = 10$ in our experiments, Sections 4 and 5). For each M_k we compute the value of the *likelihood function* L_{M_k} ($L_{M_k} = \prod_{i=1}^N \hat{p}_{\mathbf{a}_k}^{(k)}(y_i)$) at the maximized values of the parameters ω_{kj} , μ_{kj} and σ_{kj} . Then we compute the values $BIC_{M_k} = 2 \log L_{M_k} - (3M_k - 1) \log(N)$ [9], [10] and plot them for $M_k = 1, 2, \dots, M_{max}$. Finally,

following [9], [10] we select the model having the number M_k which gives rise to a decisive first local maximum of the BIC values.

4 Simulation Studies

In this section we compare the performance of the conventional GMMs [6], the mixture of *probabilistic principal component analyzers* (PPCAs) [27, Section 6] and our PP method (Section 3). The PPCA allows controlling the restriction level of the covariance matrices Σ_j from a simple diagonal matrix (using *latent factor* $q = 1$) to full unrestricted covariances (for $q = n - 1$). We fitted the conventional GMMs and the mixture of PPCAs by the EM algorithm using the NETLAB procedures [23, Sections 3 and 7].

We draw the training samples from a 15-dimensional density

$$p_{\mathbf{JK}}(x_1, x_2, \dots, x_{15}) = \left[\sum_{j=1}^3 \alpha_j g_{\mathbf{J}_j}(x_1, x_2) g_{\mathbf{K}_j}(x_3, x_4) \right] \prod_{k=5}^{15} \phi_{\sigma_k}(x_k). \quad (15)$$

Here we set $\sigma_k = 1$ for $\phi_{\sigma_k}(x_k)$ ($N(0, \sigma_k)$ density in the variable x_k) and $\alpha_1 = \alpha_2 = \frac{9}{20}$ and $\alpha_3 = \frac{1}{10}$. In (15) $g_{\mathbf{J}_j}, g_{\mathbf{K}_j}$ for $j = 1, 2, 3$ denote bivariate normal densities, having parameters listed in Table 1. For easy of presentation, Table 1 uses the bivariate normal notations $N(\mu_1, \mu_2; \sigma_1^2, \sigma_2^2, \rho)$, where the marginal means and variances are μ_i and σ_i^2 for $i = 1, 2$ and the correlation coefficient is ρ . Figure 1 shows contour plots of the marginal densities of $p_{\mathbf{JK}}(x_1, x_2, \dots, x_{15})$ (15). The structure of $p_{\mathbf{JK}}(x_1, x_2, \dots, x_{15})$ lies in the first four variables x_1, x_2, x_3, x_4 . The remaining variables x_5, x_6, \dots, x_{15} only add noise (independent variables having $N(0, \sigma_k)$ densities). The density (15) has been carefully chosen because it combines the benchmarks widely used for comparison density estimation methods [10], [24], [30], [31].

We expect that our method will perform better than conventional methods for small sizes of the training samples. In order to check this we studied the influence of the training sample size on the performance for $N = 200, 300, 400, 500, 600, 700, 800, 1300, 1800, 2300, 2800, 3300$. We evaluated the performance by *percentage of*

Table 1: Parameters for 6 Example Bivariate Normal Densities from [30]

J -density	$g_{J_1}(x, y)$	$g_{J_2}(x, y)$	$g_{J_3}(x, y)$
parameters	$N(-\frac{6}{5}, 0; (\frac{3}{5})^2, (\frac{3}{5})^2, \frac{7}{10})$	$N(\frac{6}{5}, 0; (\frac{3}{5})^2, (\frac{3}{5})^2, \frac{7}{10})$	$N(0, 0; (\frac{3}{5})^2, (\frac{3}{5})^2, -\frac{7}{10})$
K -density	$g_{K_1}(x, y)$	$g_{K_2}(x, y)$	$g_{K_3}(x, y)$
parameters	$N(-1, 0; (\frac{3}{5})^2, (\frac{7}{10})^2, \frac{3}{5})$	$N(1, \frac{2\sqrt{3}}{3}; (\frac{3}{5})^2, (\frac{7}{10})^2, 0)$	$N(1, -\frac{2\sqrt{3}}{3}; (\frac{3}{5})^2, (\frac{7}{10})^2, 0)$

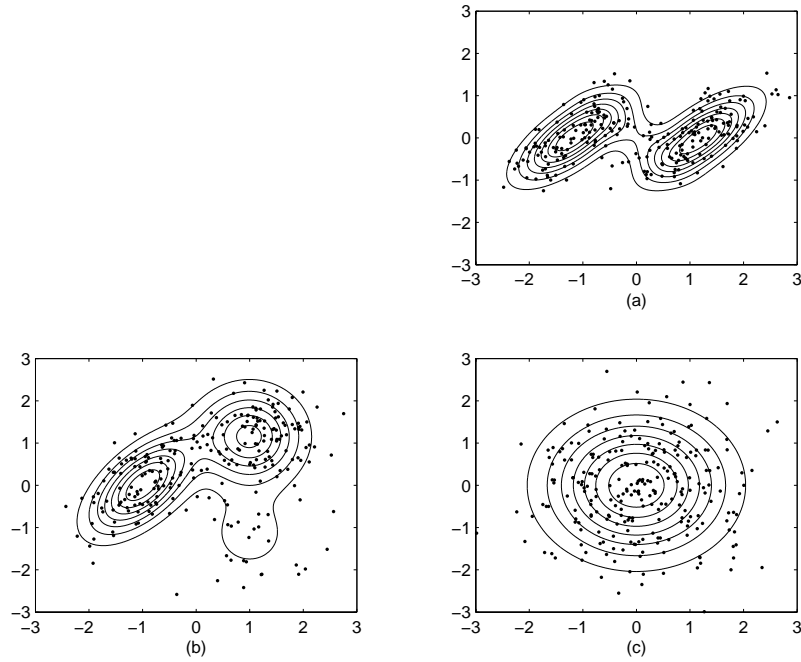


Figure 1: Contour plots of the marginal densities. (a) $p_J(x_1, x_2) = \sum_{j=1}^3 \alpha_j g_{J_j}(x_1, x_2)$. (b) $p_K(x_3, x_4) = \sum_{j=1}^3 \alpha_j g_{K_j}(x_3, x_4)$. (c) $\phi(x_5)\phi(x_6)$ There are 250 data points superimposed on the contours to show the locations of the training points drawn from these densities.

variance explained (PVE) [12, Section 7], which is a normalized version of the ISE (see Appendix C). We employed the result of Section 3, that the PP density estimation implies a GMM model for a specific setting of the augmenting functions, for simple exact computation of the PVE in our simulations. The exact calculation of the PVE (Appendix C) is carried out by direct matrix computations.

An experiment for a given size of the training sample consisted of 30 random replications of the following procedure. We drew a training sample of size N from $p_{\mathbf{JK}}(x_1, x_2, \dots, x_{15})$ (15). Then we normalized (*sphered* [11]) the data and rotated the coordinate system randomly. Using this data we fitted the GMMs for predefined values of the number M of the mixture components of the GMM (2), the number q of the latent factors of the mixture of the PPCAs [27, Section 6] and the number K of the augmenting functions of the PP density estimation (3). We set only four variations of the number K ($K = 1, 2, 3, 4$) for our method, while to the other methods we gave an advantage with respect to the number of the parameter settings - 13 settings for GMMs with diagonal covariance matrices ($M = 3, 4, \dots, 15$), 29 settings for GMMs with spherical covariance matrices ($M = 3, 4, \dots, 31$) and 48 settings for the mixture of the PPCAs ($q = 1, 2, \dots, 12, M = 3, 4, 5, 6$). We set $M = 3$ for the GMM with full covariance matrices, which is the best choice for the data drawn from (15). Methodologically the scenario of our experiments is the same as the experimental setup of previous studies of the projection pursuit density estimation [12], [18]. We employ the knowledge of the true (underlying) density (15) in the simulations and compute PVE (Appendix C) for the compared methods varying the parameter settings of the models. In the diagrams (Figure 2), we report the largest PVE for each method among the parameter variations. Thus we compare the performance of the compared methods for the best setting of their parameters. It is interesting to study the performance of the model selection methods (the methods for estimating the values of the parameters M , q and K from the training data). Such methods are explained in [7], [8], [9], [10], [11], [12], [13], [26]). The latter is somewhat beyond the scope of the present paper and is the object of our current research.

In Figure 2 we show the training sample size versus the mean of the PVE values

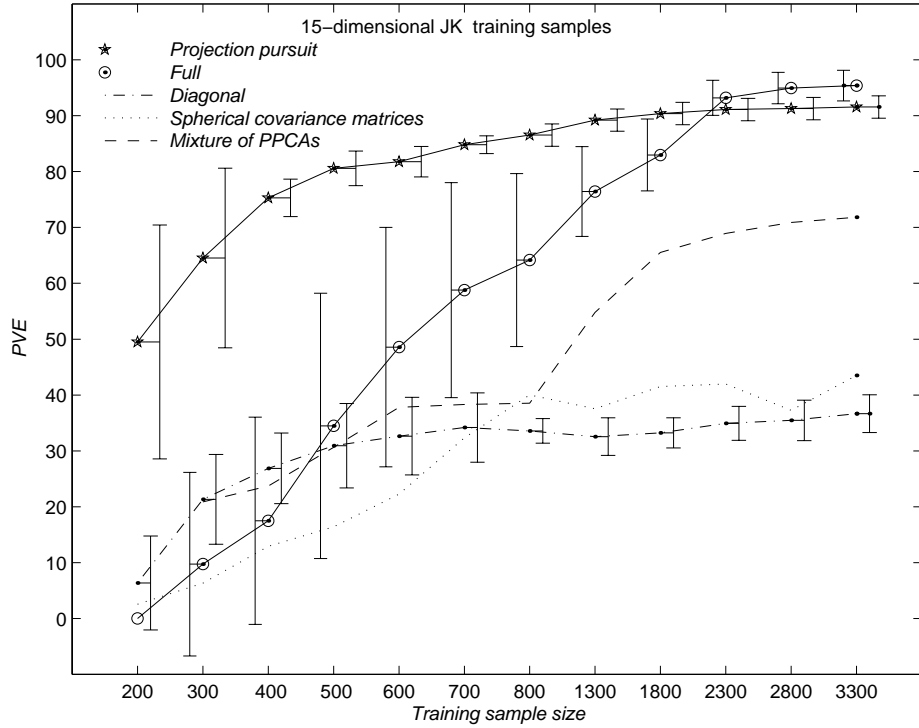


Figure 2: Estimation of $p_{\text{JK}}(x_1, x_2, \dots, x_{15})$ (15) having noise level $\sigma_k = 1$.

computed in the 30 random replications. The solid path with stars (\star) represents the PVE for our PP method, the solid path with circles (\circ) - the PVE of the GMMs with full covariance matrices, the dashed path ($-.-$) - the PVE of the GMMs with diagonal covariance matrices, the dotted path (\dots) - the PVE of the GMMs with spherical covariance matrices and the dashed path ($---$) - the PVE of the mixture of PPCAs. The bars represent the standard deviations of the PVE values. In order to simplify the graphical presentation we show the bars for our method (\star), GMMs with full (\circ) and diagonal ($-.-$) covariance matrices only. The standard deviations of the other methods were almost equal to those of the GMMs with diagonal covariance matrices.

We observe that our method (\star) outperforms the other methods for $N = 200 - 1300$. We succeeded in explaining 50-85% of the variance, while the other methods explain 0-70% for $N = 200 - 1300$. The standard deviations (the bars) of our method are smaller (for almost all N) than those of the other methods. This fact indicates that our PP method is less sensitive to training data variability.

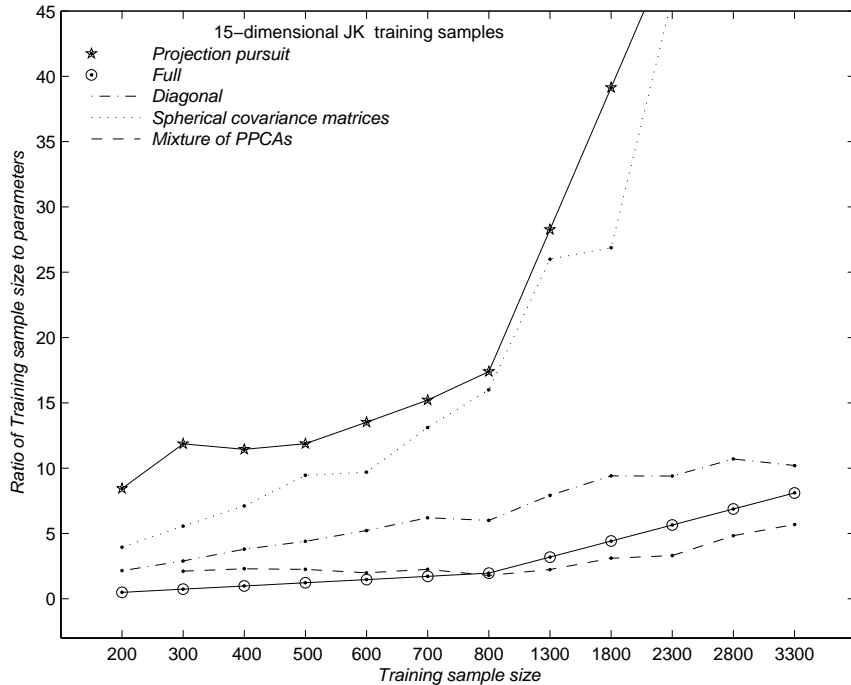


Figure 3: Ratio training sample size (N) to the number of the adjustable parameters in the estimation of $p_{\text{JK}}(x_1, x_2, \dots, x_{15})$ (15).

The GMM with full covariance matrices (\circ) has the largest PVE value for sufficiently large sample sizes: $N \geq 1800$. The latter is a quite obvious result because of the exact modeling of the underlying density (15) by the GMM with full covariance matrices. Finally, the mixture of the PPCAs ($-.-$) compromises between the best results of the standard GMMs.

In Appendix D we give the expressions for the number of the adjustable parameters of the GMMs. In Figure 3 we present the ratio of the training sample size to the mean of these numbers. Observing Figure 3 we conclude that our method explores relative few adjustable parameters, which is the reason for the better performance.

5 Relation to Independent Factor Analysis

Independent factor analysis (IFA) [5] was originally developed as a solution to *blind source separation* (BSS). It can be recast as data density modeling, which is explained below.

The IFA model is

$$\mathbf{x} = \mathbf{H}\mathbf{y} + \mathbf{u}. \quad (16)$$

Here \mathbf{x} is an observed n -dimensional data vector, $\mathbf{y} = (y_1, y_2, \dots, y_K)^T$, ($K \leq n$) are the unobserved (latent) mutually statistically independent variables, which are mixed by $(n \times K)$ -matrix \mathbf{H} . The resulting mixtures are corrupted by an n -dimensional noise signal \mathbf{u} having Gaussian density $p(\mathbf{u}) = \phi_{\Lambda}(\mathbf{u})$ with zero mean and unrestricted covariance matrix Λ . The densities of y_k for $k = 1, \dots, K$ are approximated by univariate GMMs

$$\hat{p}(y_k) = \sum_{i=1}^{M_k} \omega_{ki} \phi_{\sigma_{ki}}(y_k - \mu_{ki}). \quad (17)$$

Hence we have

$$\hat{p}(\mathbf{y}) = \prod_{k=1}^K \hat{p}(y_k) = \sum_{j=1}^{\tilde{M}} \tilde{\omega}_j \phi_{\mathbf{V}_{y_j}}(\mathbf{x} - \tilde{\mathbf{m}}_{y_j}), \quad \tilde{M} = \prod_{k=1}^K M_k. \quad (18)$$

Here $\tilde{\omega}$, \mathbf{V}_{y_j} , $\tilde{\mathbf{m}}_{y_j}$ denote the parameters obtained after multiplication of $\hat{p}(y_k)$ (17) for $k = 1, \dots, K$. In this scenario the model of the density of the observations is [5, Section 2.2]

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^{\tilde{M}} \tilde{\omega}_j \phi_{\mathbf{H}\mathbf{V}_{y_j}\mathbf{H}^T + \Lambda}(\mathbf{x} - \mathbf{H}\tilde{\mathbf{m}}_{y_j}). \quad (19)$$

In [5, Section 3.2] an EM algorithm is developed for ML fitting the parameters ω_{ki} , σ_{ki} , μ_{ki} of $\hat{p}(y_k)$ (17), \mathbf{H} and Λ to the training data (1). We must note the close relation of the IFA and the *independent component analysis* (ICA) methods [7], [20], [21].

The IFA, at least in this explanation is very close to our PP method:

(i) Comparing (19) and (14) we conclude that IFA and our method (Section 3) fit the same GMM to the training data (1).

(ii) Taking into account the ICA principle "*nongaussian is independent*" [13, Chapter 14.6], [15, Chapter 8] we conclude that the univariate GMMs of IFA (17) and our method (6) have the same nature.

(iii) During the EM iterations of the IFA the variances of the latent variables are constrained to $Var\{y_i\} = 1$ [5, Section 3.2], which relates to the preliminary *sphering*

of the training data in the projection pursuit method [11, Section 2].

The IFA and our method differ in the strategy of the fitting the GMM to the data.

The IFA optimizes an ML criterion on the $(n \times K)$ -mixing matrix \mathbf{H} , $(n \times n)$ -unrestricted covariance matrix $\mathbf{\Lambda}$ and the parameters $\omega_{ki}, \sigma_{ki}, \mu_{ki}$ for $i = 1, 2, \dots, M_k, k = 1, 2, \dots, K$ simultaneously [5, Section 3.2]. This simultaneous optimization causes the EM algorithm to become intractable [5, Section 6] for large $\tilde{M} = \prod_{k=1}^K M_k$. In order to make the algorithm practical (for large \tilde{M}) some approximations are proposed [5, Section 6]. Moreover the selection of the values of K and $M_k, k = 1, 2, \dots, K$ for IFA is a difficult problem and it is in active research [7].

Our method is based on a recursive approach [11], [17, Section 7]. We compute vector \mathbf{a}_1 defining the most nongaussian direction, remove the nongaussian structure and iterate the computations of $\mathbf{a}_2, \dots, \mathbf{a}_K$. For this purpose we optimize the PP index on n -components of \mathbf{a}_k (4). Then we select M_k using a standard model selection procedure for univariate GMM. Our method seems to be an attractive choice for large n . In some cases [17, Section 7] the recursive approach may miss structure that a direct K -dimensional search would find easily.

Here we present the result of a comparison of the IFA and our method. We used the same scenario as in Section 4. In order not to favor our method we set a low level of the noise $\sigma_k = 0.1$ for the underlying density (15). We ran the IFA using 300 iterations of the EM algorithm. Fig. 4 shows the obtained result. The solid path with stars (\star) shows the PVE values for our methods. IFA was studied for different methods for initialization of the EM algorithm. The dotted path (...) represents the PVE values for random initialization [5, Section 7], the solid path with circles (\circ) - initialization with our PP final solution, and the dashed path (--) - initialization with the solution of the coarse search of the PP *hybrid optimization strategy* [11, Section 5]. In the latter two initializations the starting matrix \mathbf{H} was set to be the pseudo-inverse of the matrix composed by the PP directional vectors into original observation space (before *sphering*). The starting $\mathbf{\Lambda}$ was set to be a diagonal matrix.

In Fig. 4 we observe the convergence of the paths with (\star) and (\circ). The latter

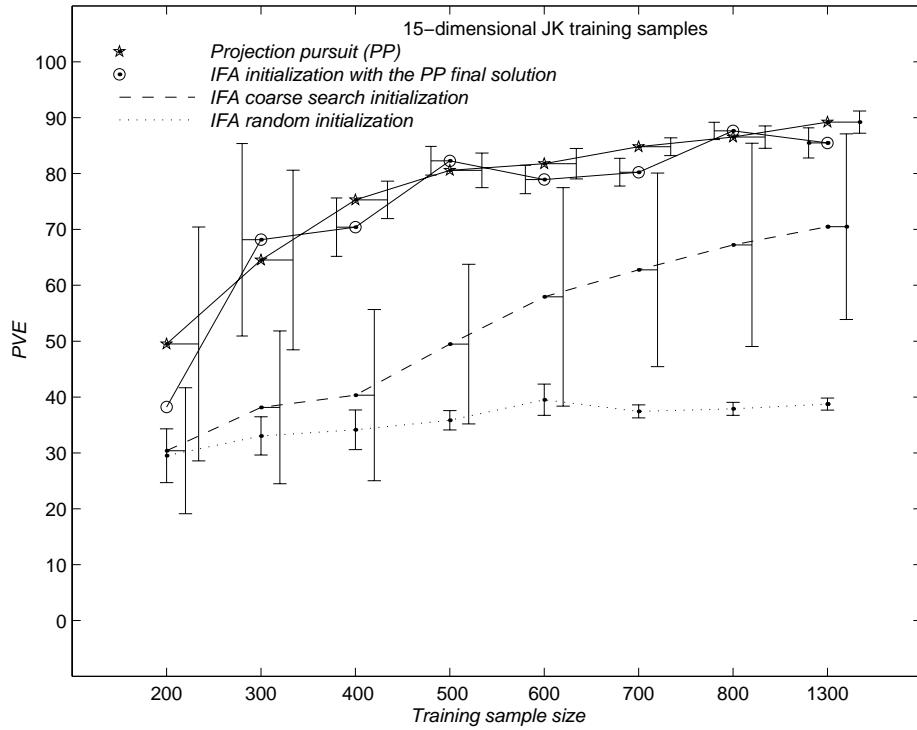


Figure 4: Estimation of $p_{\text{JK}}(x_1, x_2, \dots, x_{15})$ (15) having noise level $\sigma_k = 0.1$.

is not surprising. The goal of our method and the IFA is the fitting the same GMM (14) and (19) to the training data.

We observed small standard deviations for the dotted path (...) in Fig. 4. This means that different random initializations generally lead to the same solution. The latter observation is consistent with the result in [7, Section 6.1].

We must note that the initialization of the IFA and ICA is an open problem [7, Section 6.1], which is an area of active research. Based on the result obtained here it seems that a study of our method as the initializer of the EM algorithms of the IFA and ICA would be of interest.

6 Conclusion

We have proposed a method for fitting a *Gaussian mixture model* (GMM) based on the *projection pursuit* (PP) strategy proposed by Friedman, Stuetzle and Schroeder [11], [12]. In Section 3 we showed that the PP density estimation implies a GMM

model for a specific setting of the augmenting functions. The derived formulas (11)-(13) allow us to set the parameters of the GMM implied by the PP estimation. We employed the latter result for simple exact computation of the PVE performance in our simulations. The exact calculation of the PVE (Appendix C) is carried out by direct matrix computations instead of a complicated Monte-Carlo evaluation of the n -fold integrals of the PVE provided in [12], [18] for PP density estimation. The exact calculation of the PVE allowed us to study the performance of the PP density estimation in 15-dimensional input space, which was not possible previously because of the high computational complexity of the numerical evaluation of the PVE.

The simulation results (Section 4) show that for small sizes of the training sets, the PP strategy outperforms the GMM fitting based on the *maximum likelihood* (ML) criterion in the original high-dimensional input space.

In Section 5 we discussed the relationship between our method and the recently developed *independent factor analysis* (IFA) [5]. We concluded that a combination of our method and the IFA could be useful for the applications.

In our previous works we employed the PP strategy successfully in 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 training sets. Consequently PP density estimation may be used in important applications which currently employ GMMs, for example the *blind source separation* (BSS) [5], initialization of *radial basis function* (RBF) networks [6] and other applications.

Acknowledgments

The author wishes to thank associate editor Prof. Zixiang Xiong and the reviewers for their critical reading of the manuscript and helpful comments. This work was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev, Israel.

Appendices

A Derivation of the Formula (5)

Following Friedman [11, Section 4] we derive the formula (5) in its abstract version. That is, we imagine that the *projection pursuit* (PP) operates on the n -dimensional continuous probability density functions $p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_K(\mathbf{x})$. The first density $p_1(\mathbf{x})$ denotes the standardized underlying density $p(\mathbf{x})$ ($p_1(\mathbf{x}) = p(\mathbf{x})$ for $p(\mathbf{x})$ having zero mean vector and identity covariance matrix [11, Section 2]). The densities $p_2(\mathbf{x}), \dots, p_K(\mathbf{x})$ are the PP transformations of $p(\mathbf{x})$.

PP consists of finding directional vector \mathbf{a}_1 having the least normal univariate density $p_{\mathbf{a}_1}^{(1)}(\mathbf{a}_1^T \mathbf{x})$ under the joint density $p_1(\mathbf{x})$. Then the density $p_1(\mathbf{x})$ is transformed to $p_2(\mathbf{x})$ having standard normal density along \mathbf{a}_1 and unchanged conditional density given $\mathbf{a}_1^T \mathbf{x}$

$$p_1(\cdot | \mathbf{a}_1^T \mathbf{x}) = p_2(\cdot | \mathbf{a}_1^T \mathbf{x})^1. \quad (20)$$

Thus

$$p_2(\mathbf{x}) = p_1(\mathbf{x}) \frac{\phi(\mathbf{a}_1^T \mathbf{x})}{p_{\mathbf{a}_1}^{(1)}(\mathbf{a}_1^T \mathbf{x})}, \quad (21)$$

with $\phi(\mathbf{a}_1^T \mathbf{x})$ univariate standard normal density. The PP procedure is then repeated on the density $p_2(\mathbf{x})$, obtaining the second least normal solution $\mathbf{a}_2^T \mathbf{x}$. The density is again modified,

$$p_3(\mathbf{x}) = p_2(\mathbf{x}) \frac{\phi(\mathbf{a}_2^T \mathbf{x})}{p_{\mathbf{a}_2}^{(2)}(\mathbf{a}_2^T \mathbf{x})}. \quad (22)$$

Here $p_{\mathbf{a}_2}^{(2)}(\mathbf{a}_2^T \mathbf{x})$ is the univariate marginal density of $\mathbf{a}_2^T \mathbf{x}$ under the joint density $p_2(\mathbf{x})$.

At the K th iteration we have

$$p_K(\mathbf{x}) = p_1(\mathbf{x}) \prod_{k=1}^K \frac{\phi(\mathbf{a}_k^T \mathbf{x})}{p_{\mathbf{a}_k}^{(k)}(\mathbf{a}_k^T \mathbf{x})}, \quad (23)$$

where $p_{\mathbf{a}_k}^{(k)}(\mathbf{a}_k^T \mathbf{x})$ is the marginal density of $\mathbf{a}_k^T \mathbf{x}$ under the joint density $p_k(\mathbf{x})$, with $p_1(\mathbf{x}) = p(\mathbf{x})$.

At some point in the iterative process the PP procedure cannot find a direction

¹ $p_1(\cdot | \mathbf{a}_1^T \mathbf{x}) \approx p_2(\cdot | \mathbf{a}_1^T \mathbf{x})$ for the *structure removal* transformation [11, Section 3] used in this paper.

having a marginal density substantially different from normality. This indicates that $p_K(\mathbf{x})$ is approximately n -variate standard normal density ($p_K(\mathbf{x}) \approx \phi(\mathbf{x})$). After manipulation of (23) using $p_K(\mathbf{x}) \approx \phi(\mathbf{x})$ and $p_1(\mathbf{x}) = p(\mathbf{x})$ we obtain the approximation of $p(\mathbf{x})$ in the form

$$\hat{p}(\mathbf{x}) = \phi(\mathbf{x}) \prod_{k=1}^K f_k(\mathbf{a}_k^T \mathbf{x}), \quad (24)$$

$$f_k(\mathbf{a}_k^T \mathbf{x}) = \frac{p_{\mathbf{a}_k}^{(k)}(\mathbf{a}_k^T \mathbf{x})}{\phi(\mathbf{a}_k^T \mathbf{x})}. \quad (25)$$

Finally the formula (5) is obtained from (25) by changing $p_{\mathbf{a}_k}^{(k)}(\mathbf{a}_k^T \mathbf{x})$ to its approximation $\hat{p}_{\mathbf{a}_k}^{(k)}(\mathbf{a}_k^T \mathbf{x})$.

B Proof of the Identity (11)

According to the expressions of $N(\mathbf{m}, \Sigma)$ and $N(\mu, \sigma)$ we have for the left-hand part of the equality (11)

$$\phi_{\Sigma}(\mathbf{x} - \mathbf{m})\phi_{\sigma}(\mathbf{a}^T \mathbf{x} - \mu) = \frac{1}{\sqrt{2\pi}(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}} \sigma} e^{-\frac{1}{2}\gamma} \quad (26)$$

with

$$\gamma = \frac{(\mathbf{a}^T \mathbf{x} - \mu)^2}{\sigma^2} + (\mathbf{x} - \mathbf{m})^T \Sigma^{-1} (\mathbf{x} - \mathbf{m}). \quad (27)$$

First note that for $\mathbf{a}^T \mathbf{a} = 1$ we have

$$\gamma = (\mathbf{x} - \mathbf{a}\mu)^T \frac{\mathbf{a}\mathbf{a}^T}{\sigma^2} (\mathbf{x} - \mathbf{a}\mu) + (\mathbf{x} - \mathbf{m})^T \Sigma^{-1} (\mathbf{x} - \mathbf{m}). \quad (28)$$

Then we seek for $n \times n$ - matrices \mathbf{A} and \mathbf{B} , and a scalar γ^* that imply γ (28) in the form

$$\gamma = (\mathbf{x} - \tilde{\mathbf{m}})^T \tilde{\Sigma}^{-1} (\mathbf{x} - \tilde{\mathbf{m}}) - \gamma^* \quad (29)$$

for

$$\tilde{\mathbf{m}} = \mathbf{A}\mathbf{m} + \mathbf{B}\mathbf{a}\mu, \quad (30)$$

$$\tilde{\Sigma} = \left[\frac{\mathbf{a}\mathbf{a}^T}{\sigma^2} + \Sigma^{-1} \right]^{-1}. \quad (31)$$

Combining (28) and (29), (30), (31) we have

$$\gamma^* = \frac{\mu^2}{\sigma^2} \left[\frac{1}{\sigma^2} \mathbf{a}^T \tilde{\Sigma} \mathbf{a} - 1 \right] + \mathbf{m}^T \left[\Sigma^{-1} \tilde{\Sigma} \Sigma^{-1} - \Sigma^{-1} \right] \mathbf{m} + \frac{2\mu}{\sigma^2} \mathbf{a}^T \tilde{\Sigma} \Sigma^{-1} \mathbf{m}, \quad (32)$$

$$\mathbf{A} = \tilde{\Sigma} \Sigma^{-1}, \quad \mathbf{B} = \frac{1}{\sigma^2} \tilde{\Sigma} \mathbf{a} \mathbf{a}^T. \quad (33)$$

Then substituting γ (29) into the right-hand part of (26) and using γ^* (32) we obtain the identity (11) for α (12). Next, substituting \mathbf{A} and \mathbf{B} (33) into (30) we have $\tilde{\mathbf{m}}$ (13). Finally, applying the Sherman-Morrisson formula [32, page 325] on the right-hand part of (31) we obtain $\tilde{\Sigma}$ (13).

C Percentage of Variance Explained (PVE)

In Sections 4 and 5 we evaluated the performance of the GMMs by a performance criterion named *the percentage of variance explained* (PVE) [12, Section 7]

$$PVE = 100\left(1 - \frac{ISE}{var}\right), \quad (34)$$

where ISE is the *integrated squared error* of the GMM $\hat{p}(\mathbf{x})$

$$ISE = \int_{R^n} (\hat{p}(\mathbf{x}) - p(\mathbf{x}))^2 d\mathbf{x} \quad (35)$$

and *var* is a normalization

$$var = \int_E \left(p(\mathbf{x}) - \frac{1}{vol(E)}\right)^2 d\mathbf{x}. \quad (36)$$

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 $p(\mathbf{x})$.

The PVE of the GMM (2) can be calculated exactly by direct matrix calculations when the underlying density is a normal mixture

$$p(\mathbf{x}) = \sum_{j=1}^{M^*} \omega_j^* \phi_{\Sigma_j^*}(\mathbf{x} - \mathbf{m}_j^*). \quad (37)$$

First observe that

$$ISE = \int_{R^n} \hat{p}(\mathbf{x})^2 d\mathbf{x} - 2 \int_{R^n} \hat{p}(\mathbf{x})p(\mathbf{x})d\mathbf{x} + \int_{R^n} p(\mathbf{x})^2 d\mathbf{x} \quad (38)$$

and

$$var \approx \int_{R^n} p(\mathbf{x})^2 d\mathbf{x} - \frac{1}{vol(E)}. \quad (39)$$

From this we see that an explicit representation of PVE (34) requires a closed-form solution of the above n -fold integrals. We perform the required calculations for $\int_{R^n} \hat{p}(\mathbf{x})p(\mathbf{x})d\mathbf{x}$; the others can be performed analogously. Replacing $\hat{p}(\mathbf{x})$ and

$p(\mathbf{x})$ by the normal mixtures (2) and (37) we have

$$\int_{R^n} \hat{p}(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \sum_{k=1}^M \sum_{j=1}^{M^*} \omega_k \omega_j^* \int_{R^n} \phi_{\Sigma_k}(\mathbf{x} - \mathbf{m}_k) \phi_{\Sigma_j^*}(\mathbf{x} - \mathbf{m}_j^*) d\mathbf{x}. \quad (40)$$

The above integrals have a closed-form solution [31, page 101]

$$\int_{R^n} \phi_{\Sigma_k}(\mathbf{x} - \mathbf{m}_k) \phi_{\Sigma_j^*}(\mathbf{x} - \mathbf{m}_j^*) d\mathbf{x} = \phi_{\Sigma_k + \Sigma_j^*}(\mathbf{m}_k - \mathbf{m}_j^*). \quad (41)$$

This leads to a solution by direct matrix calculations

$$\int_{R^n} \hat{p}(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \mathbf{w}^T \mathbf{\Omega}_a \mathbf{w}^*, \quad (42)$$

where $\mathbf{w} = (\omega_1, \omega_2, \dots, \omega_M)^T$, $\mathbf{w}^* = (\omega_1^*, \omega_2^*, \dots, \omega_{M^*}^*)^T$ and $\mathbf{\Omega}_a$ denotes an $(M \times M^*)$ -matrix with (k, j) entry equal to $\phi_{\Sigma_k + \Sigma_j^*}(\mathbf{m}_k - \mathbf{m}_j^*)$.

The other integrals can be handled in the same way to obtain

$$\int_{R^n} p(\mathbf{x})^2 d\mathbf{x} = \mathbf{w}^{*T} \mathbf{\Omega}^* \mathbf{w}^* \quad (43)$$

$$\int_{R^n} \hat{p}(\mathbf{x})^2 d\mathbf{x} = \mathbf{w}^T \mathbf{\Omega} \mathbf{w}, \quad (44)$$

where $\mathbf{\Omega}^*$ is an $(M^* \times M^*)$ -matrix with (k, j) entry equal to $\phi_{\Sigma_k^* + \Sigma_j^*}(\mathbf{m}_k^* - \mathbf{m}_j^*)$ and $\mathbf{\Omega}$ is an $(M \times M)$ -matrix with entry $\phi_{\Sigma_k + \Sigma_j}(\mathbf{m}_k - \mathbf{m}_j)$.

Finally we obtain the formulas for the exact calculation

$$ISE = \mathbf{w}^T \mathbf{\Omega} \mathbf{w} - 2\mathbf{w}^T \mathbf{\Omega}_a \mathbf{w}^* + \mathbf{w}^{*T} \mathbf{\Omega}^* \mathbf{w}^* \quad (45)$$

$$var \approx \mathbf{w}^{*T} \mathbf{\Omega}^* \mathbf{w}^* - \frac{1}{vol(E)}. \quad (46)$$

In our simulations (Sections 4 and 5) we set $E = \{(-5 < x_i < 5), i = 1, 2, \dots, 15\}$ and employed (45) and (46) for the underlying density (15).

D Number of the adjustable parameters of the GMMs

The numbers $N_{\{\cdot\}}$ of the adjustable parameters of GMM (2) is:

$N_{\{full\}} = Mn(n + 3)/2 + M - 1$ for GMM with full covariance matrices;

$N_{\{diag\}} = M(2n + 1) - 1$ for GMM with diagonal covariance matrices;

$N_{\{sphered\}} = M(n + 2) - 1$ for GMM with spherical covariance matrices;

$N_{\{PPCA\}} = M[nq + 1 - q(q - 1)/2] + M - 1$ for the mixture of PPCAs [27];

$N_{\{PP\}} = nK + \sum_{k=1}^K(3M_k - 1)$ for the projection pursuit fitting of the GMM (Section 3).

References

- [1] M.E. Aladjem, "Linear discriminant analysis for two-classes via removal of classification structure", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, pp.187–192, 1997.
- [2] M.E. Aladjem, "Non-parametric discriminant analysis via recursive optimization of Patrick-Fisher distance", *IEEE Trans. on Syst., Man, Cybern.*, vol.28B, pp.292–299, 1998.
- [3] M.E. Aladjem, "Recursive training of neural networks for classification", *IEEE Trans. on Neural Networks*, vol.11, pp.488–503, 2000.
- [4] M.E. Aladjem, "Projection pursuit fitting Gaussian mixture models", Eds. T.Caelli, Amin A., Duin R.P.W., Kamel M. and Ridder D., *Advances in Statistical, Structural and Syntactical Pattern Recognition, Lecture Notes in Computer Science*, pp.380–388, 2002.
- [5] H. Attias, "Independent factor analysis", *Neural Computation*, vol.11, pp.803–851, 1999.
- [6] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press Inc., New York, 1995.
- [7] R.A. Choudrey and S.J.Roberts, "Variational mixture of Bayesian independent component analyzers", *Neural Computation*, vol.15, pp.213–252, 2003.

- [8] M.A.T. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models.", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, pp.381–396, 2002.
- [9] C. Fraley and A.E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis", *The Computer Journal*, Vol.41, pp.578–588, 1998.
- [10] C. Fraley and A.E. Raftery, "Model-based clustering, discriminant analysis, and density estimation", Technical Report no. 380, Department of Statistics, University of Washington, 2000.
- [11] J.H. Friedman, "Exploratory projection pursuit", *Journal of the American Statistical Association*, vol.82, pp.249–266, 1987.
- [12] J.H. Friedman, W. Stuetzle, and A. Schroeder, "Projection pursuit density estimation", *Journal of the American Statistical Association*, vol.79, pp.599–608, 1984.
- [13] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, Series: Springer Series in Statistics, 2001.
- [14] T. Hastie and R. Tibshirani, "Independent component analysis through product density estimation", Department of Statistics, Stanford University, *Technical Report*, 2002.
- [15] A. Hyvarinen, J. Karhunen and E. Oja, *Independent Component Analysis*, John Wiley and Sons, 2001
- [16] G.E. Hinton, P. Dayan and M. Rewov, "Modeling the manifolds of images of handwritten digits", *IEEE Transactions on Neural Networks*, Vol. 8, pp.65–74, 1997.
- [17] P.J. Huber, "Projection pursuit" (with discussion), *The Annals of Statistics*, Vol.13, pp.435–525, 1985.
- [18] J.N. Hwang, S.R. Lay and A. Lippman, "Nonparametric multivariate density estimation: A comparative study", *IEEE Trans. on Signal Processing*, Vol.42, pp.2795–2810, 1994.
- [19] A.J. Izenman, "Recent Developments in Nonparametric Density Estimation", *Journal of the American Statistical Association*, Vol. 86, pp.205–224, 1991.
- [20] T.W. Lee, M. Girolami, and T.J. Sejnowski, "Independent component analysis using an extended Infomax algorithm for mixed sub-Gaussian and super-Gaussian sources", *Neural Computation*, Vol. 11, pp. 417 – 441, 1999.
- [21] T.W. Lee, M.S. Lewicki and T.J. Sejnowski, "ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in

- blind signal separation”, *IEEE Trans. on Pattern Analysis and Machine Intelligence* Vol. 22, pp. 1078 – 1089, 2000.
- [22] G.J. McLachlan and K.E. Basford, *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker, 1988.
- [23] I.T. Nabney, *NETLAB, Algorithms for Pattern Recognition*, Springer, 2002.
- [24] K. Roeder and L. Wasserman, Practical Bayesian density estimation using mixtures of normals”, *Journal of the American Statistical Association*, vol. 92, 894 – 902, 1997.
- [25] D.W. Scott and W.F. Szewczyk, ”From kernels to mixtures”, *Technometrics*, Vol 43., pp. 323 – 335, 2002.
- [26] J. Sun, ”Some practical aspects of exploratory projected pursuit”, *SIAM J. Sci. Comput.*, vol.14, pp.68 – 80, 1993.
- [27] M.E. Tipping and C.M. Bishop, ”Mixtures of Probabilistic Principal Component Analyzers”, *Neural Computation*, vol. 11, pp.443-482, 1999.
- [28] D.M. Titterington, A.F.M. Smith and U.E. Makov, *The Statistical Analysis of Finite Mixture Distributions*, New York: Wiley, 1985.
- [29] P.A. Tukey and J.W. Tukey, ”Preparation: prechosen sequences of views”, in *Interpreting Multivariate Data*, ed. V. Barnett, London: John Wiley, pp.189-213, 1981.
- [30] M.P. Wand and M.C. Jones, ”Comparison of smoothing parameterizations in bivariate kernel density estimation”, *Journal of the American Statistical Association*, Vol.88, No.422, pp.520-528, 1993.
- [31] M.P. Wand and M.C. Jones, *Kernel Smoothing*, Charman & Hall/CRC, 1995.
- [32] A. Webb, *Statistical Pattern Recognition*, Oxford University Press Inc., 1999.