PARAMETER ESTIMATION INTO A MODEL MIXTURE

THOMAS Edouard, RAGOT José

Centre de Recherche en Automatique de Nancy, UMR CNRS 7039
Institut National Polytechnique de Lorraine,
2*, avenue de la forêt de Haye,*
54516 *Vandœuvre-lès-Nancy, France*
{ethomas, jragot}@ensem.inpl-nancy.fr

Abstract:
We propose to study in this paper a new and general method to identify parameters in a mixture of models. We will emphasize applications of this method in the following examples: a mixture of Gaussian distributions, a mixture of affine models, a dynamic case. The limits of this method and, consequently, the prospects will be underlined.

Keywords: Parameter estimation, model mixture, EM algorithm, non-linear optimisation, differential calculus, allocation problem, number of models.

1. INTRODUCTION

We take here an interest in the particular problem of parameter identification in a model. Our approach goes together with the problem of allocation (assign a given data to a given model) and the problem of parameter estimation. These problematics are closely linked. The final goal would be to have a method such that, a set of data being given, we could decide which points belong to each model, and estimate the parameters of these models. From now on, this goal is not realistic, although a great amount of research has been achieved in these areas. Let us quote, for instance, the EM algorithm, the most famous for these purposes (Dempster, *et al.*, 1977), and its alternative forms (Roweis and Ghahramani, 2000; Karlis and Xekalaki, 2003; Biernacki, *et al.*, 2003; Verbeek, *et al.*, 2003; Hawkins, *et al.*, 2001; Zhang, *et al.*, 2004; Atkinson and Cheng, 2000; Santamaria-Caballero, *et al.*, 1996). Let's also mention the promising geometric approach from (Rousseeuw and Hubert, 1999), a deconvolution-based method (Santamaria-Caballero, *et al.*, 1999), an iterative dynamic finite algorithm (Likas, *et al.*, 2003), a method inspired by differential geometry (Pearson and Ragot, 2004). Our method is based upon non linear optimisation and differential calculus. It allows to identify, in an iterative way, the parameters of a model. The most obvious applications are the assignment of data to sets, or the determination of the number of models (each representing for instance the working order of a given system). This paper is structured as follows: presentation of the method, then application to mixture of Gaussian models, of affine models and of dynamic models. Finally we present some numerical results.

2. THE PROPOSED METHOD

Let d and r be integers greater than 2. For each $j \in [\![1; r]\!]$, let $f^{(j)} : \mathbb{R}^d \to \mathbb{R}$ be an almost everywhere positive measurable function such that $\int f^{(j)}(x)\lambda_d(x) = 1$. We suppose that the

functions $f^{(j)}, j \in [\![1;r]\!]$, depend on m parameters $\delta_i,\ i \in [\![1;m]\!], f^{(j)} = f^{(j)}_{(\delta_1 \dots \delta_m)}$. The purpose is to estimate these parameters. Let $(\alpha_j)_{j=1}^r \in]0;1[^r$ be real numbers such that $\sum_{j=1}^r \alpha_j = 1$. We consider the following mixed probability density function:

$$p(x_i) := \sum_{j=1}^{r} \alpha_j f^{(j)}(x_i) \tag{1}$$

where $x_i \in \mathbb{R}^d$. Hence there exists a probability space $(\Omega; A; P)$ and a random variable $X : (\Omega; A) \to (\mathbb{R}^d; B(\mathbb{R}^d))$ such that X admits p as a density function. We perform $N \geq 2$ measures of this random variable X; let $x := [x_1 \dots x_N]^T \in \mathbb{R}^{Nd}$ be the matrix of the N measures. For that set of measurements, we then consider the joint density:

$$\beta(x) = \prod_{i=1}^{N} \sum_{j=1}^{r} \alpha_j f^{(j)}(x_i). \tag{2}$$

We deduce from (2) the expression of the log-likelihood:

$$\nu(x) = \sum_{i=1}^{N} \log \sum_{j=1}^{r} \alpha_j f^{(j)}(x_i). \tag{3}$$

We then introduce the following Lagrangian:

$$\mathcal{L}(x; \lambda) = \sum_{i=1}^{N} \log \sum_{j=1}^{r} \alpha_j f^{(j)}(x_i) + \lambda(\sum_{j=1}^{r} \alpha_j - 1). \tag{4}$$

We will afterwards derive this function with respect to each of the parameters δ and the mixture coefficients α. They are considered as variables and, at their optimum (when the true value is reached), the different partial derivatives have to be equal to zero. We note that $f^{(j)}(x) = f^{(j)}_{(\delta_1 \dots \delta_m)}(x)$ where $j \in [\![1;r]\!], x \in \mathbb{R}^{Nd}$; each of the m parameters evolves in an open subset U_j of a Banach space E_j which is supposed to be metric and finite dimensional. We introduce this formal reference system of $E_1 \times ... \times E_m \times \mathbb{R}^{r+1}$: $\Re := \big(\delta_1 \dots \delta_m;\ \alpha_1 \dots \alpha_r;\ \lambda\big)$.

First assumption:
The gradient of the Lagrangian $\mathcal{L}$ exists.

Second assumption:
The functions $f^{(1)} \dots\ f^{(r)}$ are of class $\mathcal{C}^1$ over $\mathbb{R}^d$, and the r differential functions $y \mapsto d_y f^{(1)}_{(\delta_1 \dots \delta_m)}(x)$ $\dots$ $y \mapsto d_y f^{(r)}_{(\delta_1 \dots \delta_m)}(x)$ are invertible.

Let's notice that these two assumptions are not restricting as they are satisfied in practice. We shall evaluate the partial derivatives of $\mathcal{L} = \mathcal{L}(x; \lambda)$ thanks to (4):

$$\frac{\partial \mathcal{L}}{\partial \delta_k} = \sum_{i=1}^{N} \frac{\sum_{j=1}^{r} \alpha_j \frac{\partial f^{(j)}(x_i)}{\partial \delta_k}}{p(x_i)}, \quad k \in [\![1;m]\!], \tag{5a}$$

$$\frac{\partial \mathcal{L}(x;\lambda)}{\partial \alpha_k} = \sum_{i=1}^{N} \frac{f^{(j)}(x_i)}{p(x_i)} + \lambda, \quad k \in [\![1;r]\!], \tag{5b}$$

$$\frac{\partial \mathcal{L}(x;\lambda)}{\partial \lambda} = \sum_{j=1}^{r} \alpha_j - 1. \tag{5c}$$

We deduce from (5a), (5b) and (5c) the following optimality equations:

$$\sum_{i=1}^{N} \frac{1}{p_i(x)} \sum_{j=1}^{r} \alpha_j \frac{\partial f^{(j)}(x_i)}{\partial \delta_k} = 0,\ k \in [\![1;m]\!], \tag{6a}$$

$$\lambda = -\sum_{i=1}^{N} \frac{f^{(j)}(x_i)}{p_i(x)}, \qquad j \in [\![1;r]\!], \tag{6b}$$

$$\sum_{j=1}^{N} \alpha_j = 1. \tag{6c}$$

For sake of brievety, we will only deal with the optimality equations related to the m parameters δ in (6a). The implicit functions theorem applies to (6a) and (7) holds; for each index $k \in [\![1;m]\!]$, we have an implicit equation for δ_k (Ramis, *et al.*, 1998):

$$\begin{aligned} \delta_1 &= g^{(1)}_{(x;\delta_2 \dots \delta_m)}(\delta_1) \\ &\vdots \\ \delta_m &= g^{(m)}_{(x;\delta_1 \dots \delta_{m-1})}(\delta_m) \end{aligned} \tag{7}$$

Moreover, the m functions $g^{(j)}$ are U_j-valued, and of class $\mathcal{C}^1$ over U_j, $j \in [\![1;m]\!]$. Defining $\Delta = \big(\delta_1 \dots \delta_m\big)$ and $G = \big(g^{(1)} \dots g^{(m)}\big)$, we will deal with the formal equation (fixed-point problem): $\Delta = G(\Delta)$. We define as well $U := U_1 \times \dots \times U_m$ and $E = E_1 \times \dots \times E_m$, Banach space of which U is an open subspace. Therefore $\Delta \in U$ and we can suppose that $G : U \to U$, G of class C^1 over U.

Third assumption:
$G : F \to F$, where F is a non empty closed subspace of E included into U, is a $\mathcal{K}-$contracting application ($\mathcal{K} \in]0;1[$).

Then the Picard-Banach theorem holds: fixing $\Delta_0 \in U$, the sequence defined by $\Delta_{n+1} = G(\Delta_n), n \in \mathbb{N}$, is convergent, and its limit is the unique fixed point $\hat{\Delta} \in U$ of G. In some applications, the third assumption is however very strong.

3. APPLICATION

As mentioned in the introduction, we now present concrete applications of the proposed method

(mixture of gaussians, mixture of static systems, dynamic system with outliers) and discuss its relevance.

3.1 The case of r Gaussian distributions

The notations are strictly the same as in the previous section. We have in this case the distribution between r mixed Gaussian densities

$$p(x_i) = \sum_{j=1}^{r} \alpha_j p_j(x_i), \tag{8a}$$

$$p_j(x_i) := \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - m_j}{\sigma_j}\right)^2}. \tag{8b}$$

The joint density of the measures is given by:

$$\beta(x) = \frac{1}{(2\pi)^{N/2}} \prod_{i=1}^{N} \sum_{j=1}^{r} \alpha_j p_j(x_i). \tag{9}$$

Then we consider the Lagrangian $\mathcal{L} = \mathcal{L}(x;\lambda)$:

$$\mathcal{L} = \sum_{i=1}^{N} \log p(x_i) - \frac{N}{2}\log 2\pi + \lambda\left(\sum_{k=1}^{r}\alpha_k - 1\right). \tag{10}$$

As in the previous section, the processes is the following: we derive the Lagrangian $\mathcal{L}$ with respect to each of its parameters, we deduce the optimality equations and set up methods to solve those equations. The derivatives of the Lagrangian are evaluated below ($1 \le k \le r$):

$$\frac{\partial\mathcal{L}(x;\lambda)}{\partial m_k} = \sum_{i=1}^{N} \frac{x_i - m_k}{p(x_i)\sqrt{2\pi}} \frac{\alpha_k}{\sigma_k^3} e^{-\frac{1}{2}\left(\frac{x_i - m_k}{\sigma_k}\right)^2},$$

$$\frac{\partial\mathcal{L}(x;\lambda)}{\partial \sigma_k} = \sum_{i=1}^{N} \frac{1}{p(x_i)} \frac{\alpha_k}{\sigma_k^2} \left(\left(\frac{x_i - m_k}{\sigma_k}\right)^2 - 1\right) e^{-\frac{1}{2}\left(\frac{x_i - m_k}{\sigma_k}\right)^2},$$

$$\frac{\partial\mathcal{L}(x;\lambda)}{\partial \alpha_k} = \sum_{i=1}^{N} \frac{1}{p(x_i)} p_k(x_i) + \lambda,$$

$$\frac{\partial\mathcal{L}(x;\lambda)}{\partial \lambda} = \sum_{k=1}^{r}\alpha_k - 1. \tag{11}$$

At the optimum, these derivative are set to zero and then we deduce from (11):

$$\lambda = -N, \tag{12a}$$

$$m_k = \frac{1}{N}\sum_{i=1}^{N} \frac{x_i p_k(x_i)}{p(x_i)}, \quad 1 \le k \le r, \tag{12b}$$

$$\sigma_k^2 = \frac{1}{N}\sum_{i=1}^{N} \frac{p_k(x_i)}{p(x_i)}(x_i - m_k)^2, \ 1 \le k \le r. \tag{12c}$$

For simplicity, we will not deal with the evaluation of the mixture coefficients α_j. Let's put the problem of the estimation of the parameters m_k, σ_k in the form of a fixed-point problem. For this purpose, let's define the following vectors:

$$\theta := \left[\, m_1 \ \ldots \ m_r;\ \sigma_1^2 \ \ldots \ \sigma_r^2 \,\right]^T \in \mathbb{R}^{2r},$$

$$F(\theta) := \frac{1}{N}\begin{bmatrix} \sum_{i=1}^{N} \frac{x_i p_1(x_i)}{p(x_i)} \\ \vdots \\ \sum_{i=1}^{N} \frac{x_i p_r(x_i)}{p(x_i)} \\ \vdots \\ \sum_{i=1}^{N} \frac{p_1(x_i)}{p(x_i)}(x_i - m_1)^2 \\ \vdots \\ \sum_{i=1}^{N} \frac{p_r(x_i)}{p(x_i)}(x_i - m_r)^2 \end{bmatrix} \in \mathbb{R}^{2r}. \tag{13}$$

The $2r$ relations (12b) and (12c) are clearly equivalent to $\theta = F(\theta)$. It is important to note that the fixed-point theorems only apply to functions of class C^1, which is not necessarily the case of a function such as F. All we can hope is to obtain a condition upon the gradient of F (when this has a meaning) such as $\| F \| < 1$ (the norm $\|\bullet\|$ is to be settled) to ensure, at least locally speaking, the convergence of the fixed point algorithm towards an effective fixed point of F.

3.2 The case of two affine models

We apply the proposed method in the case of two affine models. The processes is the same for several affine models. The equations of two affine models are determined by four parameters a_1, a_2, b_1, b_2:

$$\begin{cases} y_i = a_1 x_i + b_1 \\ y_i = a_2 x_i + b_2 \end{cases} \tag{14}$$

where $i \in [\![1; N]\!]$, $N \ge 2$. We introduce the following quantities ($1 \le i \le N$):

$$\varepsilon_{1;i} = y_i - a_1 x_i - b_1, \qquad \varepsilon_{2;i} = y_i - a_2 x_i - b_2 \tag{15}$$

which correspond to the measurement errors. As in the case of the Gaussian models, we consider the distribution of the errors, then the functions:

$$p_k(x_i) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp -\frac{\varepsilon_{k;i}^2}{2\sigma_k^2}, \quad k \in \{1;2\}, \tag{16a}$$

$$p(x_i) = \alpha_1 p_1(x_i) + \alpha_2 p_2(x_i). \tag{16b}$$

The mixing coefficients $\alpha_1,\ \alpha_2$ will be taken equal to 0.5 and therefore will not be estimated. The processes is the same as in the case of Gaussian models: we work out the Lagrangian, derive it with respect to each parameter to be estimated, then set the derivatives to zero and finally deduce

from these an implicit expression of the parameters. Here are the relations that we obtain:

$$a_1 = \frac{\sum_{i=1}^{N}(y_i - b_1)x_i \frac{p_1(x_i)}{p(x_i)}}{\sum_{i=1}^{N} x_i^2 \frac{p_1(x_i)}{p(x_i)}}, \tag{17a}$$

$$b_1 = \frac{1}{N}\sum_{i=1}^{N}(y_i - b_1)\frac{p_1(x_i)}{p(x_i)}, \tag{17b}$$

$$a_2 = \frac{\sum_{i=1}^{N}(y_i - b_2)x_i \frac{p_1(x_i)}{p(x_i)}}{\sum_{i=1}^{N} x_i^2 \frac{p_1(x_i)}{p(x_i)}}, \tag{17c}$$

$$b_2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - b_2)\frac{p_1(x_i)}{p(x_i)}. \tag{17d}$$

Simulations and results will be discussed of in the fourth section of this paper.

3.3 The case of dynamic systems

In the two previous examples 3.1 and 3.2, the equations were related to static systems. We here deal with a dynamic case, with the same method. The equations of the model are the following:

$$y_0 \text{ is given,}$$
$$y_{i+1} = ay_i + bx_i, \qquad i \in [\![1; N]\!]. \tag{18}$$

The probability distribution is the following:

$$p_i(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y_{i+1} - ay_i - bx_i}{\sigma}\right)^2}. \tag{19}$$

The notations are given here: x_i stands for the input of the system and N for the number of measurements. We will focus only on the problem of the identification in the case of a single model. The parameters a and b satisfy the relations below:

$$a = \frac{\sum_{i=1}^{N}(y_{i+1} - bx_i)y_i p_i(x)}{\sum_{i=1}^{N} y_i^2 p_i(x)}, \tag{20a}$$

$$b = \frac{\sum_{i=1}^{N}(y_{i+1} - ay_i)x_i p_i(x)}{\sum_{i=1}^{N} y_i^2 p_i(x)}. \tag{20b}$$

Here, the algorithm is iterative and two-levelled (a and b are alternatively estimated). The results of the simulations are shown in the next part of the document.

4. NUMERICAL RESULTS

In this part we present some figures et numerical results that allow us to evaluate the relevance of the proposed method.

4.1 The case of r Gaussian distributions, r > 1

The processes followed in the section 3.1 can be generalised in the case of multidimensional Gaussian distributions. The example below shows a mixture of two bidimensional distributions. A zero mean and normalised Gaussian noise has been added to each data. We have generated two sets of points in the plane according to Gaussian distributions, which respective parameters are A(1; 2) and B(4; 4) for the means, 1.25 and 0.75 for the variances. See Fig. 1.

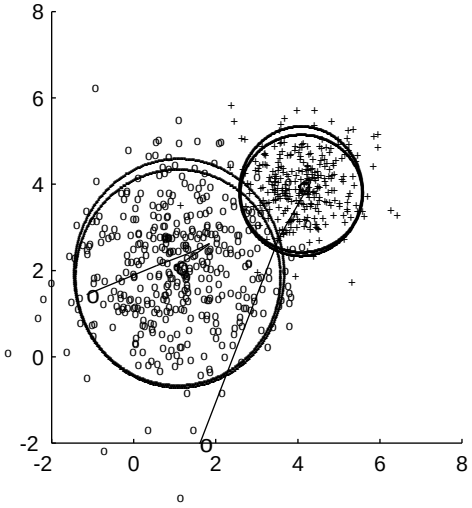

Fig. 1. visualization of convergence

For the algorithm, the initial values have been set equal to $(-1;\ 1.5)$ and $(2.2;\ -2)$, 1 and 1 respectively. Convergence towards the true values is obtained after about one hundred steps; we get for A and B the respective estimations $\tilde{A}$ (0.9; 1.93) and $\tilde{B}$ (3.97; 3.97). We have drawn on Fig. 1 the two circles centred on $\tilde{A}$ and $\tilde{B}$, with radius the respective variances estimated (1.17 and 0.77) on one hand. On the other hand, we have drawn the two circles centred on A and B with radius the true respective variances (1.25 and 0.75). They merely almost coincide.

As previously mentioned, the mixing coefficient is not estimated, it has been set to 0.5; however it is possible to estimate its value when deriving the Lagrangian with respect to α. The values that are to be estimated and their respective estimations are summarized in the Table 1 that way for two cases: $m_{k;1}$ stands for the abscissa of the kth Gaussian model, $m_{k;2}$ its ordinate, and σ_k its variance ($k = 1$ or 2). The number of points considered for the first Gaussian model is 100 (and 200 for the second one). Twenty experiences have been achieved for each of the simulations (the average of the results has been rounded), and about one hundred steps are needed to observe convergence.

	m_{11}	m_{12}	σ_1	m_{21}	m_{22}	σ_2
True values	2.0	3.0	1.0	4.0	5.0	3.0
Est. values	1.7	2.6	0.9	3.8	4.8	3.0
True values	2.0	3.0	2.0	4.0	5.0	2.0
Est. values	2.8	4.1	2.0	4.7	5.4	2.2

Table 1. Two bidimensional Gaussian models

These results are satisfactory insofar as we content ourselves with an order of magnitude of the parameters. Let's remark that the limitation to 20 simulations is a minimum: the results are excellent beyond 30 simulations (less than 5 per-cent of error is observed in the examples that we have treated). We can notice that the noise does not have a significance influence on the results. On the other hand, the results are very reliable when one of the variances at least is small.

4.2 The case of affine models

Four parameters have to be estimated with the described method: the iterative algorithm used is two-levelled (the slopes and the intercepts are alternatively estimated). The results are presented further. An example is reproduced in Fig. 2 where we see the superposition of two true straight lines and the two estimations that we get. A zero-mean normalized Gaussian additive noise has been added, and the lines have been perturbed by aberrant values. For all that, the results are satisfactory. The method that we propose yields in this case excellent results that are almost not sensitive to noisy data.

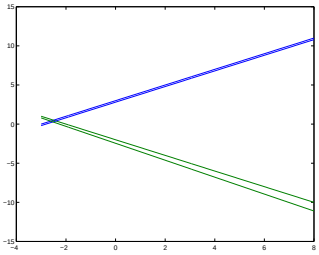

Fig. 2. Superposition of the straight lines

Fig. 3 emphasizes the noisy data that are used, they are moreover perturbed by some punctual aberrant measures around the straight line $y = 3x - 1$.

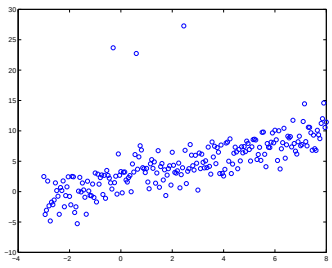

Fig. 3. Noisy data available

In the next simulations (10 experiences made for each one), we consider a constant number of iterations (100) and data that are strongly perturbed (by seven aberrant values equal to 50) and noisy (the slopes and the intercepts being submitted to a zero-mean normalized Gaussian noise). We choose 45 dots for each straight line, with an additive noise which is zero-mean, Gaussian and with variance 1.0. The Table 2 sums up some simulations.

True values				Estimated values			
a_1	b_1	a_2	b_2	a_1	b_1	a_2	b_2
1	-1	1	3	0.9	-1.2	1.2	2.5
1	-1	1	-1	1.1	-0.9	1.2	-1.2
1	-1	3	-1	1.1	-0.8	3.0	-1.0
1	3	-1	-2	1.0	3.2	-1.0	-2.1

Table 2. Two affine models

The applications presented here are particularly unfavourable to a reliable estimation. However, we get thanks to this method results that are of good accuracy. The estimations for the intercepts are less precise that for the slopes, this is understandable in the presence of aberrant values. The algorithm is not much sensitive to noise and aberrant values. The results are excellent when the parameters are all distinct. Finally, about one hundred iterations suffice to obtain convergence.

4.3 The case of dynamic models

We will emphasize the following phenomena: the number of dots considered and the number of iterations in the algorithm do not have much influence on the results; a is in general estimated with a satisfactory accuracy, whereas the estimation of b can be very different from the true value; the noise (additive noise for a, additive noise for b, and extra additive noise) has a strong influence on the results; the initialization step is a fundamental stage. When convergence arises (weak noise), the results are generally excellent. The use of a noisy output perturbed by aberrant values instead of the measured output does not have an influence on the results, except in the case of coarse initialization. In such a case, the algorithm quickly diverges or produces estimations that are very different from the true values. The aim of this part is to highlight the fundamental role played by the initialization step. Let's consider a very unfavourable case: additive noise upon the first parameter a, additive noise upon the second parameter b, additive noise upon the measure. The number of dots constituting the input x is weak, and so is the number of iterations used. We also systematically use the perturbed value of the output.

Parameters of the algorithm: each noise is a zero-mean Gaussian noise with variance 0.2; the number of measures of the input is 140; the number of aberrant values in the output is 2 (maximal magnitude: 10). The results are presented in the Table 3.

True values		Initial values		Estimated values	
a	b	a	b	a	b
0.9	6.1	0.88	5.90	0.88	5.98
0.9	6.1	0.20	0.11	0.78	8.14
0.2	0.2	0.70	5.00	0.26	0.17

Table 3. Dynamic model

To conclude, the results are quite satisfactory, even after modification of the additive noise or the noise upon the parameters a and b. In every situation, the algorithm is noise sensitive and produces excellent results when we consider the average of at least 20 simulations.

5. CONCLUSION

In this paper, we have proposed a new practical method dedicated to identification of parameters into mixtures. We have explained in detail the processes, based upon non linear optimization and involving some concepts from differential calculus. This method fits the case of several mixed Gaussian distributions, and even more general distributions. In the fourth part of this document, where some pieces of the method have been implemented, we have highlighted some advantages (instantaneous execution, straightforwardness of the code) and limitations (influence of the initialization, noise sensitivity).
This study can be extended in several ways: theoretical study of the validity of the processes, test of the influence of every parameter, generalization to non linear models, generalization to more abstract spaces than Banach spaces, study of the influence of the nature of the noise.

REFERENCES

Atkinson, A.C. and T.-C. Cheng (2000). On robust linear regression with incomplete data. In: *Computational Statistics and Data Analysis*, Vol. **33**, pp. 361-380

Biernacki C., G. Celeux and G. Govaert (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. In: *Computational Statistics and Data Analysis*, Vol. **41**, pp. 561-575

Hawkins D.S., D.M. Allen and A.J. Stromberg (2001). Determining the number of components in mixtures of linear models. In: *Computational Statistics and Data Analysis*, Vol. **38**, pp. 15-48

Karlis D. and E. Xekalaki (2003). Choosing initial values for the EM algorithm for finite mixtures. In: *Computational Statistics and Data Analysis*, Vol. **41**, pp. 577-590.

Likas A., N. Vlassis and J.J. Verbeek (2003). The global k-means clustering algorithm. In: *Pattern Recognition*, Vol. **36**, pp. 451-461

Pearson D.W. and J. Ragot (2004). Identifying parameters of local switching models: a geometrical approach. In: *International Workshop on Systems, Signals and Image Processing* (IWSSIP)

Ramis E. and C. Deschamps (1998). *3.Topologie et éléments d'analyse*, chapitre 8. Dunod, Paris

Rousseeuw P.J. and M. Hubert (1999). Regression depth. In: *Journal of the American Statistical Association*, Vol. **94**, pp. 388-402

Roweis S. and Z. Ghahramani (2000). An EM algorithm for identification of nonlinear dynamical systems. In: *Kalman Filtering and Neural Networks*. (S. Haykin (Ed.)). To appear.

Santamaria-Caballero I., C.J. Pantaleon-Prieto and A. Artès-Rodriguez (1996). Sparse deconvolution using adaptative mixed-Gaussian models. In: *Signal Processing*, Vol. **54**, pp. 161-172

Santamaria-Caballero I., C.J. Pantaleon-Prieto, J. Ibanez and A. Artès-Rodriguez (1999). Deconvolution of seismic data using adaptive Gaussian mixtures. In: *IEEE Transactions on Geoscience and Remote Sensing*, Vol. **37**

Verbeek J.J., J.R.J. Nunnink and N. Vlassis. Accelerated variants of the EM algorithm for Gaussian mixtures (2003). To appear. http://carol.science.uva.nl/ vlassis/accelem.pdf

Verbeek J.J., N. Vlassis and B. Krose (2003). Efficient Greedy Learning of Gaussian Mixture Models. In: *Neural Computation*, Vol. **15**, pp. 469-485

Zhang B., C. ZHANG and X. Yi (2004). Competitive EM algorithm for finite mixture models. In: *Pattern Recognition*, Vol. **37**, pp. 131-144