COMPUTATIONAL & APPLIED MATHEMATICS

Volume 24, N. 2, pp. 209–229, 2005
Copyright © 2005 SBMAC
ISSN 0101-8205
www.scielo.br/cam

Adaptive basis selection for functional data analysis via stochastic penalization

CEZAR A.F. ANSELMO, RONALDO DIAS and NANCY L. GARCIA

Departamento de Estatística, IMECC, UNICAMP
Caixa Postal 6065 – 13081-970 Campinas, SP, Brazil
E-mails: cafa@ime.unicamp.br / dias@ime.unicamp.br / nancy@ime.unicamp.br

Abstract. We propose an adaptive method of analyzing a collection of curves which can be, individually, modeled as a linear combination of spline basis functions. Through the introduction of latent Bernoulli variables, the number of basis functions, the variance of the error measurements and the coefficients of the expansion are determined. We provide a modification of the stochastic EM algorithm for which numerical results show that the estimates are very close to the true curve in the sense of L_2 norm.

Mathematical subject classification: Primary: 62G05, Secondary: 65D15.

Key words: basis functions, SEM algorithm, functional statistics, summary measures, splines, non-parametric data analysis, registration.

1 Introduction

It is very common to have data that comes as samples of functions, that is, the data are curves. Such curves can be obtained either from an *on-line* measuring process, where we have the data collected continuously in time or from a smoothing process applied to discrete data. Functional Data Analysis (FDA) is a set of techniques that can be used to study the variability of functions from a sample as well as its derivatives. The major goal is to explain the variability within and among the functions. For an extensive discussion of such techniques see Ramsay and Silverman (2002).

#607/04. Received: 26/V/04. Accepted: 08/IX/04.

In this paper, we assume that each curve can be modeled as a linear combination of B-splines functions. The coefficients of the expansion can be obtained by the least square method. However, in doing so, we get a solution for which the number of basis functions equals the number of observations, achieving interpolation of the data. Interpolation is not desirable since we have noisy data. To avoid this problem we propose a new method to regularize the solution using stochastic penalization. This is possible through the introduction of latent Bernoulli random variables which indicate the subset of basis to be selected and consequently the dimension of the space.

Before applying the model we have to understand the different sources of variability for curves. Variability for functions can be of two types: range and phase. The range variability gives the common pattern to each individual or function. Phase variability, on the other hand, can mask the common pattern of the functions. The usual situation, where both sources of variability are present, require complex estimation techniques. As an example, we can think about height, it is well known that the growth velocity is very high for young children and slows down as the age increases. Different children have different growth velocities in scale (range variation) as well as in time (phase variation). The mean function – also called (cross-)sectional mean – is a descriptive statistic which is widely used to give a rough idea of the process generating the curves. Figures 1.1–1.3 show some examples of variability. Notice in Figure 1.3 that the cross-sectional mean can be very misleading in the presence of phase variability. For all simulated results in this section we used a low variance noise to better illustrate each individual curve.

Our approach, like most approaches in Functional Data Analysis can only be used for curves in the presence of range variability. If phase variability is present it is necessary to align the curves using a method introduced by Ramsay and Li (1998) called *registration*. After registration is done, we can find the mean of the registered curves – the structural mean. Figure 1.4 presents the same curves as Figure 1.3 after registration. It is clear that in this case the structural mean is much closer to the real curve than the arithmetic sectional mean of the original curves.

This paper is organized as follows. Section 2 presents the proposed model

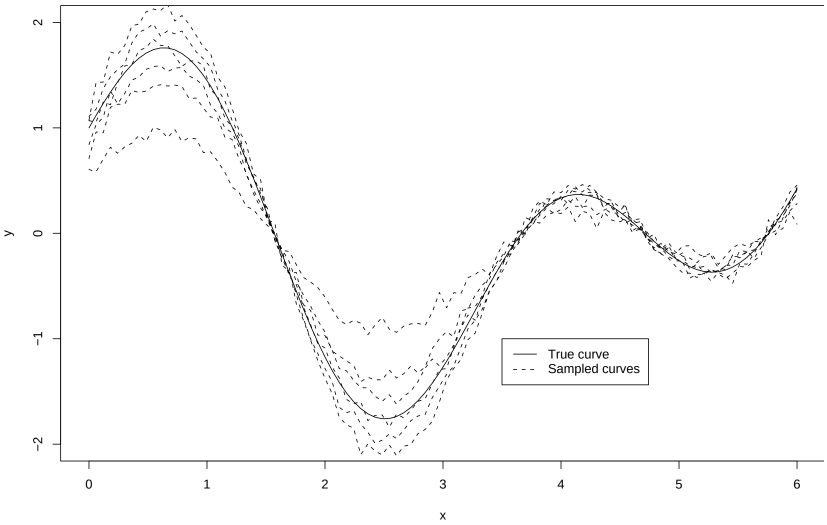

Figure 1.1 – Simulated curves under range variability.

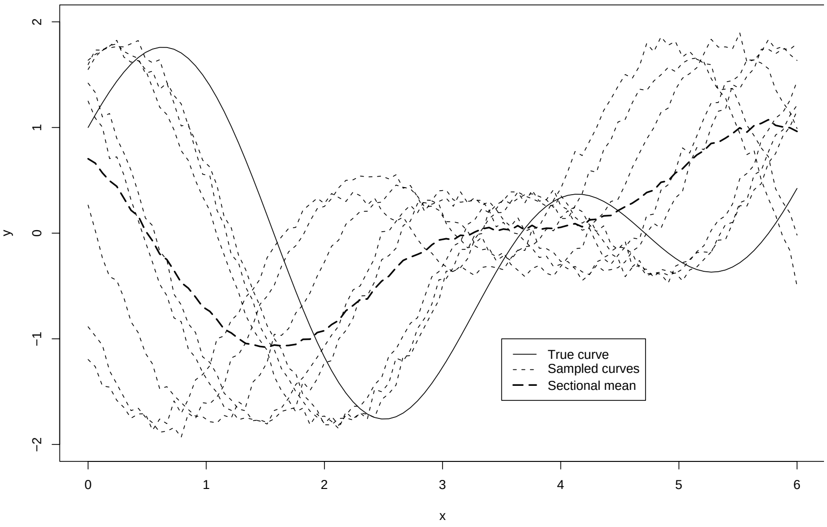

Figure 1.2 – Simulated curves under phase variability.

under range variability. An algorithm to estimate the curves based on a modification of the Stochastic EM algorithm is given in Section 3. The numerical results presented in Section 4 are based in small simulations and study two types of curves. The plots and the mean square errors (MSE) obtained show that the

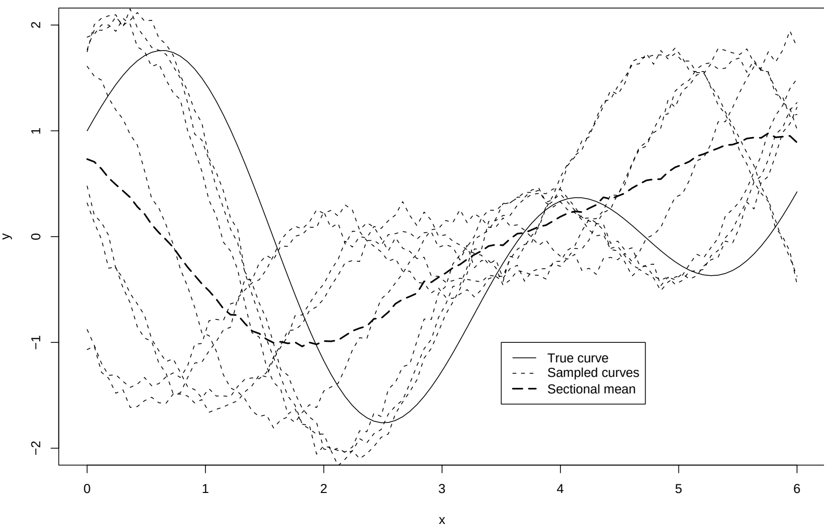

Figure 1.3 – Simulated curves under range and phase variability.

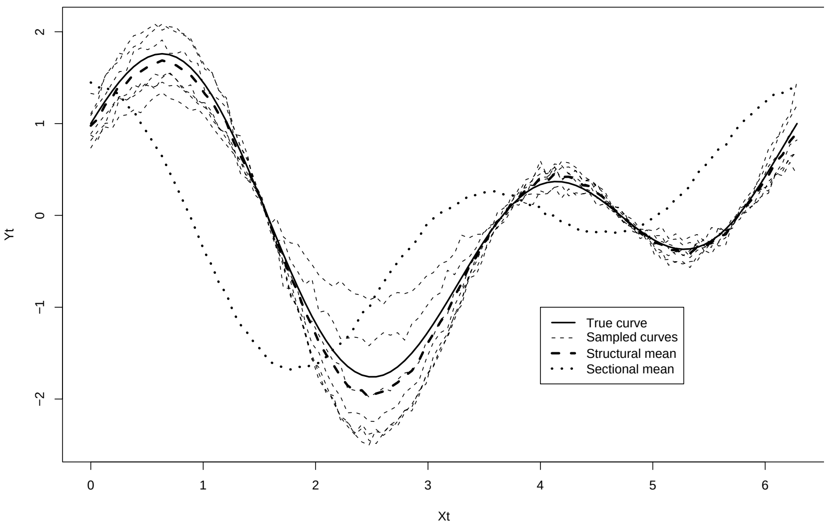

Figure 1.4 – Simulated curves under range variability.

techniques are highly successful and very adaptive. Moreover, increasing the number of curves, lead us to believe that the method is consistent. In Section 5 we provided a modification of the continuous registration algorithm (Ramsay and Li 1998) which has better performance than the original one in some cases.

2 Proposed model

Suppose we have m individuals with n_i observations at points $x_{ij} \in A \subseteq \mathbb{R}$, $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$. Let $Y_{ij} = Y(x_{ij})$ be the response obtained from the i-th individual at the point x_{ij}. Assuming that the observations are subjected only to range variability we have the following model

$$Y_{ij} = g(x_{ij}) + \varepsilon_{ij}, \tag{2.1}$$

where ε_{ij} are normally distributed with zero mean and constant variance σ^2.

Rice (2000) suggested that random coefficients should be included in the model to account for individual curve variation. In fact, in many applications, the observed curves differ not only due to experimental error but also due to random individual differences among subjects. In this paper we propose one possible way of accomplishing this random effect. Assume that g can be written as a random linear combination of spline functions as

$$g(x_{ij}) = \sum_{k=1}^{K} Z_{ki} \beta_{ki} B_k(x_{ij}) \tag{2.2}$$

where $B_k(\cdot)$ are the well known spline basis functions (cubic B-splines) and Z_{ki} are independent Bernoulli random variables with $Z_{ki} \in \{0, 1\}$ and $\mathbb{P}_\theta(Z_{ki} = 1) = \theta_{ki}$, for $k = 1, \ldots, K$ and $i = 1 \ldots, m$. That is, for $\mathbf{z}_i = (z_{1i}, \ldots, z_{Ki})$

$$f(\mathbf{z}_i) = \mathbb{P}_\theta(Z_{1i} = z_{1i}, \ldots, Z_{Ki} = z_{Ki}) = \prod_{k=1}^{K} \theta_k^{z_{ki}} (1 - \theta_{ki})^{1 - z_{ki}}. \tag{2.3}$$

To simplify the notation let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{1n_i})$, $\mathbf{Z}_i = (Z_{1i}, \ldots, Z_{Ki})$, $\beta_i^{(K)} = \beta_i = (\beta_{1i}, \ldots, \beta_{Ki})$, $\mathbf{X}_i^{(K)} = (B_1(\mathbf{x}_i), \ldots, B_K(\mathbf{x}_i))$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{in_i})$.

The conditional density of $(\mathbf{Y}_i | \mathbf{Z}_i = z_i)$ is given by:

$$f(\mathbf{y}_i | \mathbf{z}_i) = \phi\left(\frac{\mathbf{y}_i - \sum_{k=1}^{K} z_{ki} \beta_{ki} B_k(\mathbf{x}_i)}{\sigma}\right), \tag{2.4}$$

where $\phi(\cdot)$ denotes the standard multivariate normal density. Their joint density is

$$f(\mathbf{y}_i, \mathbf{z}_i) \ \propto \ (\sigma)^{-n_i} \exp\Bigg\{ -\frac{1}{2\sigma^2}\|\mathbf{y}_i - \sum_{k=1}^{K} z_{ki}\beta_{ki}B_k(\mathbf{x}_i)\|^2 + \sum_{k=1}^{K} z_{ki}\log\theta_{ki} + (1-z_{ki})\log(1-\theta_{ki})\Bigg\}.$$

Thus the joint log-density of $(\mathbf{Y}, \mathbf{Z})$ with respect to a dominant measure can be written as:

$$\begin{aligned}\log f(\mathbf{y}, \mathbf{z}|\sigma^2, \beta, \theta) \ &= \ \sum_{i=1}^{m} \log f(\mathbf{y}_i, \mathbf{z}_i)\\ &\propto \ \sum_{i=1}^{m} \Bigg\{ -n_i\log\sigma - \frac{1}{2\sigma^2}\|\mathbf{y}_i - \sum_{k=1}^{K} z_{ki}\beta_{ki}B_k(\mathbf{x}_i)\|^2 \\ &\quad + \sum_{k=1}^{K} \log(1-\theta_{ki}) + z_{ki}\log\left(\frac{\theta_{ki}}{1-\theta_{ki}}\right).\Bigg\}\end{aligned} \tag{2.5}$$

Note that maximizing the complete log-likelihood $f(\mathbf{y}, \mathbf{z}|\sigma^2, \beta, \theta)$ is equivalent to solve a stochastic penalized least square problem associated to (2.5). Since $\log(\theta/1-\theta) < 0$, increasing the number of variables $\sum_k z_{ki}$ decreases both the sum of squares and the last term in (2.5). Therefore, we can interpret the latent variables z_{ki} as regularization parameters or stochastic penalization.

In addition, the sum of z_{ki} random variables provides the number of basis functions needed to fit a model (the dimension of the space) and the values of z_{ki} indicate which variables should be included in the model $i = 1, \ldots, m$. However, this kind of representation leads to a limit solution with $z_{ki} = 1$ and $\theta_{ki} \to 1$ as $K \to \infty$, which causes a non-identifiable model. One possible way to avoid this problem is by using the following transformation:

$$\theta_{ki} = 1 - \exp\left\{ -\lambda_i \frac{|\beta_{ki}|}{\sum_r |\beta_{ri}|}\right\} \tag{2.6}$$

with $0 < \lambda_i < M$, for all i. Observe that θ_{ki} goes to $1 - e^{-\lambda_i}$ for large values of β_{ki}. That is, large values of β_{ki} indicate that the associated basis to this coefficient should be included in the model with high probability θ_{ki}. In order to avoid extreme values of θ_{ki} we suggest to use $M = 1$ limiting the inclusion probability to be $1 - e^{-1}$. Thus, the complete log-likelihood can be rewritten as:

$$\begin{aligned}\log f(\mathbf{y}, \mathbf{z}|\sigma^2, \beta, \theta) &= \sum_{i=1}^{m} \log f(\mathbf{y}_i, \mathbf{z}_i) \\ &\propto -n_i \log \sigma^2 - \frac{1}{2\sigma^2} \|\mathbf{y}_i - \sum_{k=1}^{K} z_{ki} \beta_{ki} B_k(\mathbf{x}_i)\|^2 \qquad (2.7) \\ &+ \sum_{k=1}^{K} z_{ki} \log\Big(\exp\Big\{\lambda \frac{|\beta_{ki}|}{\sum_r |\beta_{ri}|}\Big\} - 1\Big) - \sum_{k=1}^{K} \lambda \frac{|\beta_{ki}|}{\sum_r |\beta_{ri}|}.\end{aligned}$$

Without loss of generality suppose from now on that $n_i \equiv n$ for all $i = 1, \ldots, m$.

3 A variation of the stochastic EM algorithm

The EM algorithm was introduced by Dempster, Laird and Rubin (1977) to deal with estimation in the presence of missing data. In our model, the Z_{ki} are non-observable random variables and the EM algorithm could be applied. More specifically, the algorithm finds iteratively the value of θ that maximizes the complete likelihood where at each step the variables Z_{ki} are replaced by their expectations over the conditional density of $(\mathbf{Z}|\mathbf{Y})$ which is given by

$$f(\mathbf{z}|\mathbf{y}) = \frac{f(\mathbf{y}|\mathbf{z}) f(\mathbf{z})}{f(\mathbf{y})},$$

where

$$f(\mathbf{y}) = \sum_{\mathbf{z}} \prod_{i=1}^{m} f(\mathbf{y}_i, \mathbf{z}_i|\sigma^2, \beta, \theta),$$

$$f(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^{m} f(\mathbf{y}_i|\mathbf{z}_i) \quad \text{and} \quad f(\mathbf{z}) = \prod f(\mathbf{z}_i).$$

The appealing feature of the EM algorithm is that it increases the incomplete likelihood function at each iteration. However, it is well known that EM algorithm can reach a saddle point or a plateau. Moreover, there is a high computational cost to find $f(\mathbf{z}|\mathbf{y})$. The stochastic EM algorithm proposed by Celeux and Diebolt (1992) is an alternative to overcome EM algorithm limitations. At each iteration, the missing data are replaced by simulated values Z_{ki} generated according to $f(\mathbf{z}|\mathbf{y})$. To simulate these values we notice that

$$f(\mathbf{z}|\mathbf{y}) = \frac{f(\mathbf{y}|\mathbf{z})\,f(\mathbf{z})}{f(\mathbf{y})} \propto f(\mathbf{y}|\mathbf{z})\,f(\mathbf{z}) \tag{3.1}$$

and Metropolis-Hastings algorithm could be used. The theoretical convergence properties of SEM algorithms are difficult to assess since it involves the study of the ergodicity of the Markov chain generated by SEM algorithm and the existence of the corresponding stationary distribution. For particular cases and under regularity conditions, (Diebolt and Celeux 1993) proved convergence in probability to a local maximum. However, computational studies showed that SEM algorithm is even better than EM algorithm for several cases, for example censored data (Chauveau 1995), mixture case (Celeux, Chauveau and Diebolt 1996). A drawback of this procedure is that it requires thousands of simulations and the computational cost would be, again, very high.

In this work we propose a modification of the simulation step in the SEM algorithm. At each step, instead of generating Z_{ki} by the conditional density, we are going to generate them from their marginal Bernoulli distribution using estimates of θ obtained from the data. Notice that if Metropolis-Hastings were to be used to generate from the conditional distribution using the marginal as the proposal distribution, the acceptance probability would be

$$\alpha(\mathbf{z}_j, \mathbf{z}_{j+1}) \;=\; \min\left\{1,\; \frac{f(\mathbf{y}, \mathbf{z}_{j+1}|\theta)}{f(\mathbf{y}, \mathbf{z}_j|\theta)}\right\}.$$

Therefore, small changes at each step make the acceptance probability very high and the performances of the approximation and the SEM algorithm do not differ substantially.

Specifically, the algorithm can be described as follows.

Algorithm 3.1

1. Fix $\overline{K}$ the maximum number of basis to be used to represent each curve;

2. For each curve i, take $\theta_{ki}^{(0)} = 1/2$, $k = 1, \ldots, \overline{K}$;

3. At iteration l:

   (a) Simulate $Z_{ki}^{(l)}$ as a Bernoulli random variable with success probability $\theta_{ki}^{(l)}$ until $\sum_{k=1}^{\overline{K}} Z_{ki}^{(l)} \geq 3$;

   (b) Estimate $\hat{\beta}_i^{(l)}$ and $\hat{\sigma}^2$ using least squares;

   (c) Estimate $\hat{\lambda}$ by maximizing the complete likelihood subject to $\lambda < 1$;

   (d) Estimate $\hat{\beta}_i^{(l)}$ and $\hat{\sigma}^2$ using maximum likelihood;

   (e) Update $\hat{\theta}_{ki}^{(l+1)} = 1 - \exp\{-\hat{\lambda}_i^{(l)} |\hat{\beta}_{ki}^{(l)}| / \sum_r |\hat{\beta}_{ri}^{(l)}|\}$;

   (f) Save $\hat{\beta}_i^{(l)}$;

   (g) If the stopping criteria is satisfied, stop. If not, return to 3a.

4. Summarize all the obtained curves.

In order to apply Algorithm 3.1 we need to specify:

- The maximum number of basis $\overline{K}$;
- The summarization procedure of the curves;
- The stopping criterion.

The maximum number of basis $\overline{K}$. There is no consensus about a criteria to fix the maximum number of basis functions on any adaptive process. Dias and Gamerman (2002) suggest the use of at least $3b + 2$ as a starting point in the Bayesian non-parametric regression where b is the number of *bumps* of the curve. Particularly for our approach, numerical experiments give evidences that $\overline{K} = 4b + 3$ is large enough.

Summary measures. For each iteration l in Algorithm 3.1, vectors $\hat{\beta}_i^{(l)}$ are obtained for each curve i. These vectors contain the estimates of the coefficients for the basis selected (through the $\mathbf{Z}_i^{(l)}$) for the ith curve (the non-selected basis positions are filled with zeros) and the estimate of the ith curve is given by

$$\hat{Y}_i^{(l)} = \mathbf{X}_i \hat{\beta}_i^{(l)}. \tag{3.2}$$

The final estimate for the ith curve could be

$$\hat{Y}_i = \frac{1}{L} \sum_{l=1}^{L} \hat{Y}_i^{(l)}. \tag{3.3}$$

Although (3.3) provides a natural estimate for each curve, other summary measures can be proposed through weighted averages of the coefficients of the selected basis. For example, for $m = 1$ and taking $L = 3$ iterations with $K = 5$ basis, assume that for each iteration we selected the following sets of basis {1, 3, 4}, {2, 3, 5} and {3, 4, 5} respectively. We could use

$$\begin{gathered} \tilde{\beta}_1 = \frac{\hat{\beta}_1^{(1)}}{3}, \quad \tilde{\beta}_2 = \frac{\hat{\beta}_2^{(2)}}{3}, \quad \tilde{\beta}_3 = \frac{\hat{\beta}_3^{(1)} + \hat{\beta}_3^{(2)} + \hat{\beta}_3^{(3)}}{3}, \\ \tilde{\beta}_4 = \frac{\hat{\beta}_4^{(1)} + \hat{\beta}_4^{(2)}}{3}, \quad \tilde{\beta}_5 = \frac{\hat{\beta}_5^{(2)} + \hat{\beta}_5^{(3)}}{3} \end{gathered} \tag{3.4}$$

or

$$\begin{gathered} \tilde{\beta}_1 = \frac{\hat{\beta}_1^{(1)}}{1}, \, \tilde{\beta}_2 = \frac{\hat{\beta}_2^{(2)}}{2}, \, \tilde{\beta}_3 = \frac{\hat{\beta}_3^{(1)} + \hat{\beta}_3^{(2)} + \hat{\beta}_3^{(3)}}{3}, \\ \tilde{\beta}_4 = \frac{\hat{\beta}_4^{(1)} + \hat{\beta}_4^{(2)}}{2}, \, \tilde{\beta}_5 = \frac{\hat{\beta}_5^{(2)} + \hat{\beta}_5^{(3)}}{2}. \end{gathered} \tag{3.5}$$

There are several other ways to weight the coefficients. Notice that we can summarize the estimates $\hat{Y}_i^{(l)}$ along each iteration for computing the estimate $\hat{Y}_i$ for the ith curve in a completely analogous way that we can summarize each estimate $\hat{Y}_i$ to obtain the final estimate $\hat{Y}$. Therefore, we drop the l superscript and present here only summaries of $\hat{Y}_i$ to obtain the estimate $\hat{Y}$.

Observe that (3.4) and (3.5) are the unweighted and weighted versions of

$$\tilde{\beta}_k = \frac{1}{n(Z_k)} \sum_{i=1}^{m} \left[Z_{ki} \hat{\beta}_{ki} \right], \tag{3.6}$$

with

$$n(Z_k) = \begin{cases} m, & \text{unweighted case} \qquad \text{(3.6a)} \\ \max\left\{1, \sum_{i=1}^{'} Z_{ki}\right\}, & \text{weighted case.} \qquad \text{(3.6b)} \end{cases}$$

In this case, the weights take into account the number of times each basis was considered in the model.

We can do a similar summary measure by taking:

$$\tilde{\beta}'_k = \frac{1}{n'(Z_k)} \sum_{i=1}^{m} \left[Z_{ki} \hat{\beta}_{ki} \sum_{r=1}^{K} Z_{ri} \right], \tag{3.7}$$

with

$$n'(Z_k) = \begin{cases} \sum_{i=1}^{m} \sum_{r=1}^{K} Z_{ri}, & \text{unweighted case} \quad (3.7\text{a}) \\ \max\left\{1, \sum_{i=1}^{m} \left[Z_{ki} \sum_{r=1}^{K} Z_{ri} \right]\right\}, & \text{weighted case.} \quad (3.7\text{b}) \end{cases}$$

The weights proposed by 3.7 take into account two factors: the number of basis necessary to approximate each curve and the number of times each basis was selected. Another proposal is:

$$\tilde{\beta}''_k = \frac{1}{n''(Z_k)} \sum_{i=1}^{m} \left[\frac{Z_{ki} \hat{\beta}_{ki}}{\sum_{r=1}^{K} Z_{ri}} \right], \tag{3.8}$$

with

$$n''(Z_{ki}) = \begin{cases} \sum_{i=1}^{m} \frac{1}{\sum_{r=1}^{K} Z_{ri}}, & \text{unweighted case} \quad (3.8\text{a}) \\ \max\left\{1, \sum_{i=1}^{m} \left[\frac{Z_{ki}}{\sum_{r=1}^{K} Z_{ri}} \right]\right\}, & \text{weighted case.} \quad (3.8\text{b}) \end{cases}$$

This equation is analogous to (3.7), but considers as weight the inverse of the number of basis necessary to approximate each curve. That is, the bigger the number of basis necessary to approximate the curve, the smaller its weight.

After computing the summary coefficient's $\tilde{\beta}$ we can summarize the curve as

$$\hat{Y} = \mathbf{B}\tilde{\beta}, \tag{3.9}$$

where $\tilde{\beta}$ is obtained through (3.6), (3.7) or (3.8) and **B** is the design matrix given by the **X** variables.

In Section 4 we show that all these proposals provide very good estimates in terms of MSE.

The stopping criterion. We propose a flexible stopping criterion. Let $\delta > 0$ be such that we wish to stop the estimation process when the MSE between two successive estimates is smaller than δ. For each estimated curve, if the maximum number of iterations is attained (1000) and MSE $> \delta$ make $\delta \leftarrow c\delta$, $c > 1$ and begin again the estimation process. In the simulation study we used $c = 1.3$ and got good results.

4 Simulation

For all the simulations we used data with no outliers. Without loss of generality we took equally spaced observations for each curve and used the same sample grid for all individuals. Moreover, the curves are already registered and aligned by some registration method. How to register the curves is discussed in Section 5.

Simulations were run in bi-processed Athlon machine with 2.0 GHz processor and 1.5 Gb RAM memory.

The software used was Ox (`http://www.nuff.ox.ac/uk/users/Doornik`) and R (`http://www.cran.r-project.org`), operating in Linux platform.

The test curves used were

$$g_1(X_t) = \cos(X_t) + \sin(2X_t) \tag{4.1}$$

and

$$g_2(X_t) = 0.1\, X_t + 0.9\, e^{-(1/2)(X_t - \bar{X})^2}. \tag{4.2}$$

The observations Y_t are generated from the curves above plus a noise ε_t. The variables $\{\varepsilon_t, t = 1, \dots, n\}$ are iid normal random variables with zero mean and standard deviation σ. For comparison we run the simulations in three cases: small ($\sigma = 1/10$), moderate ($\sigma = 1/4$) and large ($\sigma = 1/2$) standard deviation.

Instead of using the raw data to estimate the β's we use a smoothed version of them called *the structural mean*. There are two ways of smoothing the data. The first one takes the average of data (discrete observations) and then smooths it. The second one first smooths each curve and then takes the average of the smoothed curve. Simulation studies showed that there are no difference between these methods in terms of mean square error (MSE). Therefore, from now on, we are going to use as input data the smoothed version of the curves by first averaging the raw data and then smoothing the obtained curve.

First, we analyze the MSE when we estimate the final curve using Equation (3.9) and weights given by (3.6), (3.7) and (3.8). We simulated 3 curves and added a noise with small variance ($\sigma = 1/10$). Figure 4.5 presents the estimated curves. Notice that all three estimates are practically the same and coincide with the true curve. Convergence was attained after 21, 141 and 159 iterations respectively. Figure 4.6 presents the same 3 curves but with a higher variance noise ($\sigma = 1/2$). In this case, for one of the curves convergence was not attained even after 1000 iterations. Using the flexible stopping criteria described before we just needed 8 extra iterations to achieve convergence.

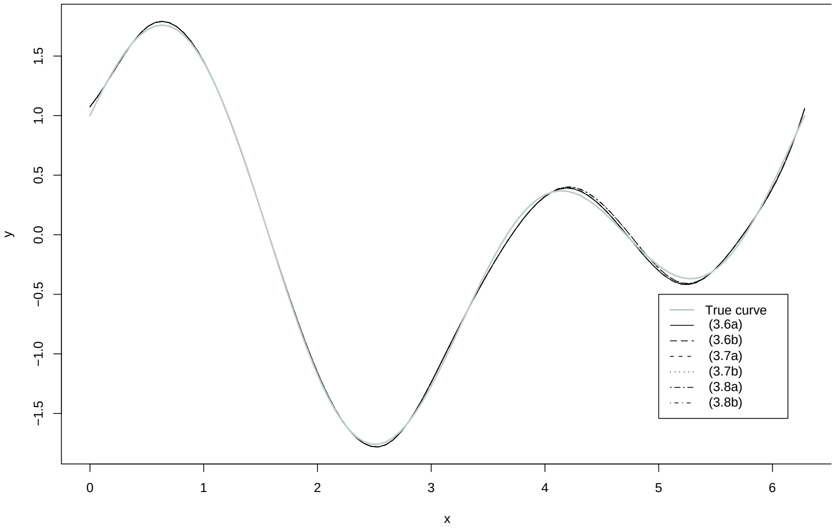

Figure 4.5 – Estimated curves for a sample with small variance.

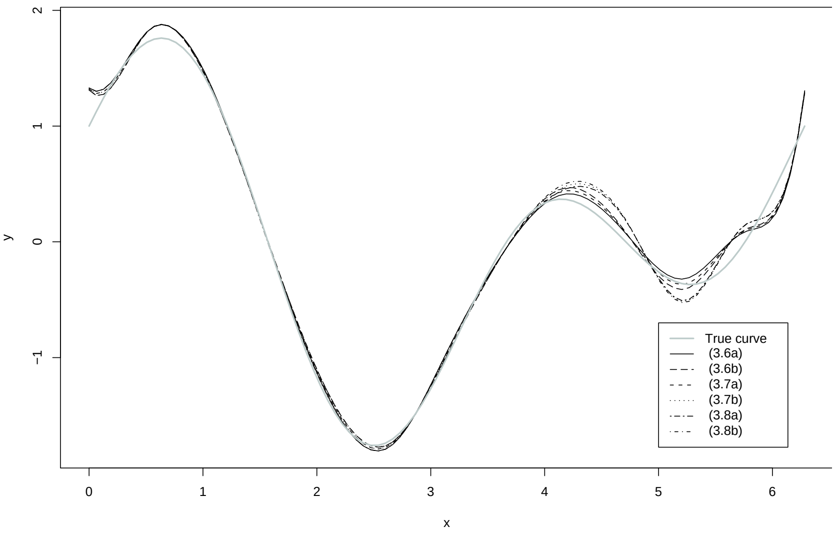

Figure 4.6 – Estimated curves for a sample with large variance.

To see the performance of Algorithm 3.1, we run several simulations with different values of m, different variances for the functions given by (4.1) and (4.2). As expected, the larger the variance, the lower the quality of the estimation. On the other hand, we have provided consistent estimators (the bigger the sample size the better the estimate). Table 4.1 summarizes these results. All summary measures give approximately the same resulting curve. Measure (3.7a) gives better results for high variance or bigger sample size, while (3.8a) is better for all other cases, except one where (3.6a) is better. The occurrence of several identical results is caused by the simplicity of the curves and small sample size reducing the number of different solutions.

	$g_1(X_t)$			$g_2(X_t)$			
noise	m			m			summary
sd (σ)	3	5	10	3	5	10	measure
1/10	4.37e-4	3.96e-4	3.88e-4	4.10e-4	2.62e-4	1.09e-4	(3.6a)
	4.29e-4	3.82e-4	3.87e-4	4.10e-4	2.62e-4	1.02e-4	(3.7a)
	4.49e-4	4.12e-4	3.91e-4	4.10e-4	2.62e-4	1.17e-4	(3.8a)
1/4	2.47e-3	1.53e-3	4.74e-4	1.12e-3	6.67e-4	3.73e-3	(3.6a)
	2.43e-3	1.56e-3	4.75e-4	1.24e-3	7.01e-4	3.58e-4	(3.7a)
	2.51e-3	1.53e-3	4.87e-4	1.02e-3	6.51e-4	3.94e-4	(3.8a)
1/2	1.02e-2	5.24e-3	2.89e-3	4.30e-3	1.48e-3	1.31e-3	(3.6a)
	1.03e-2	5.18e-3	2.78e-3	4.30e-3	1.64e-3	1.35e-3	(3.7a)
	1.03e-2	5.35e-3	3.03e-3	4.30e-3	1.39e-3	1.28e-3	(3.8a)

Table 4.1 – MSE between the estimate and the true curve.

5 Registration

Registration techniques can be found in Ramsay and Li (1998) and Ramsay (2003). The implementation of these techniques are available in R and Matlab. The package **fda** in R language has two available techniques *landmark* and *continuous registration*.

The landmark technique is appropriate when the curves to be registered have prominent features like valleys and peaks. Suppose we have curves $g_0, g_1, \ldots, g_m$ to be registered and they present Q of such properties at points t_{iq}, $q = 1, \ldots, Q$ and $i = 1, \ldots, m$. In fact, the beginning at t_{i0} and the ending at $t_{i,Q+1}$ of the ith curve are also considered and $Q+2$ properties are to be used in computing the *warping* function $h(t)$. This function h is such that the registration of the curve $g_i(t)$ is given by

$$g_i(h(t_{iq})) = g_0(t_{0q}), \quad \text{for } q = 0, \dots, Q+1. \tag{5.1}$$

The goal is to make a time transformation such that the $Q+2$ properties are aligned in time. In the R package, the user provides the placement of the Q properties for each curve which makes the process highly subject to error. For example, certain curves do not present one or more of the critical points or there is ambiguity where to place them. When there are too many curves and/or too many properties the marking is too tedious. However, Ramsay (2003) showed that automatic methods for mark identification can lead to serious mistakes.

The continuous registration method tries to solve these problems. It maximizes a measure of similarity among the curves,

$$F(h) = \int_a^b [g_i(h(t)) - g_0(t)]^2 \, dt, \tag{5.2}$$

where $[a, b]$ is the observation interval. This measure takes into account the whole curve and not only the critical points and it works well if h has good properties such as smoothness. However, it fails if g_i and g_0 differs also in range. This routine needs that the functions g_i and g_0 be given in functional form which can be obtained using Fourier series or B splines, for example.

In Figure 5.7 we present an example where we simulated $m = 3$ curves adding a low variance noise and shifting it through an uniform random variable in the interval [0,2]. In this case, since the functions are periodic we used Fourier expansion with 6 terms. Figure 5.8 presents the registered curves. Observe that there is a noticeable difference between the structural mean and the true curve caused by the failure of the registration of two of the curves.

Consider another example where the curves have a range and phase variation. To obtain this effect we fixed the horizontal axis as a reference and defined as “bumps” the pieces of the true curve between two zeroes. Following, for each bump generate q_{2iq}, $i = 1, \dots, m$, $q = 1, \dots, Q$ (Q = number of bumps) iid random samples from an uniform random variable in the interval (0.5, 1.5). Figure 5.9 presents this transformation for 3 curves adding a low variance noise

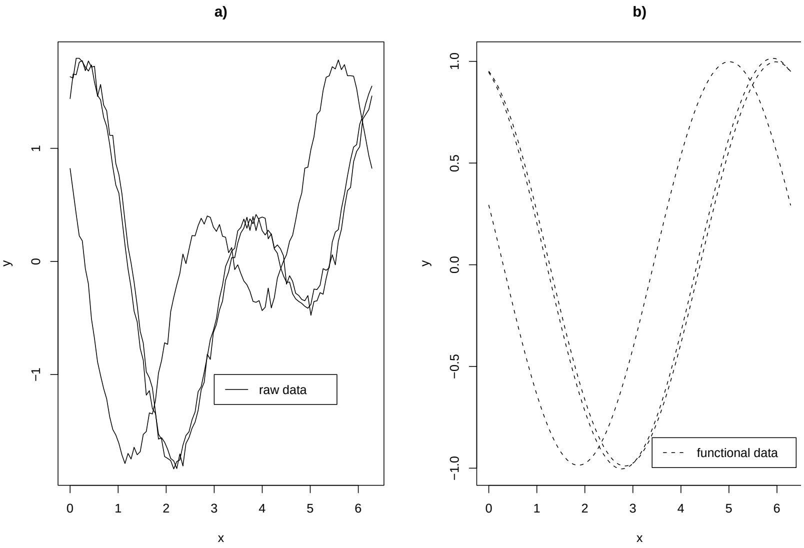

Figure 5.7 – A simple shifting case: a. Simulated curves; b. Curves obtained using `create.fourier.basis()`.

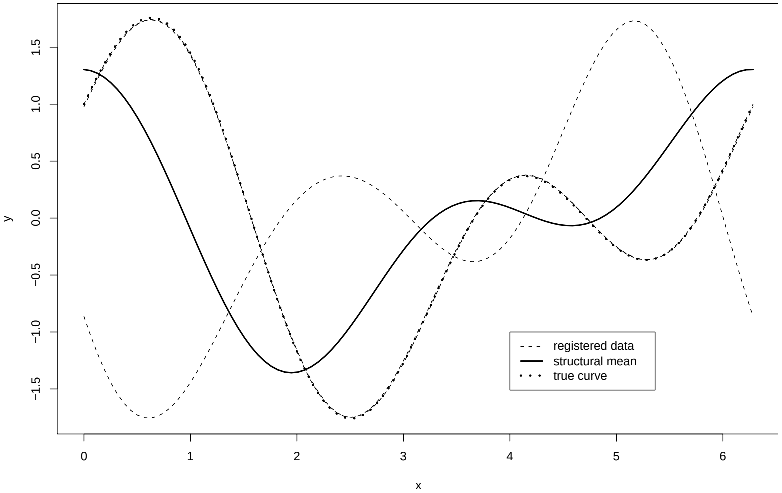

Figure 5.8 – A simple shifting case – Registered curves using `registerfd()`.

for the raw data and the functional data using Fourier transformation. Figure 5.10 presents the result of registration done by R routine `registerfd()`. As the registration process is not right, the structural mean differs very much from the true curve.

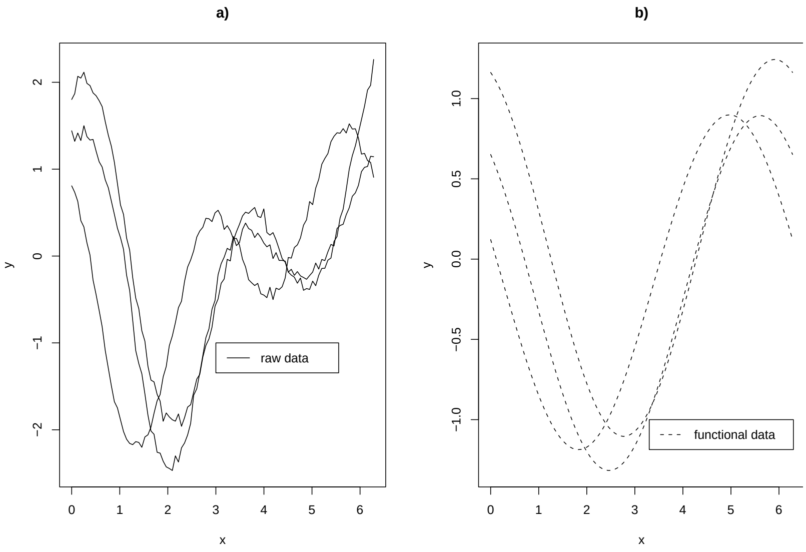

Figure 5.9 – A complex shifting case: a. Simulated curves; b. Curves obtained using `create.fourier.basis()`.

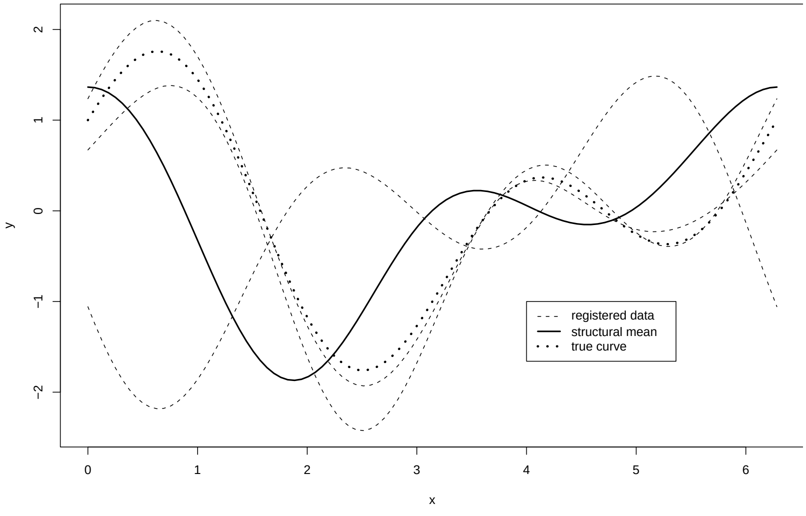

Figure 5.10 – A complex shifting case – Registered curves using `registerfd()`.

To overcome this problem we propose a modification in the continuous registration procedure. Based on (5.2) we try to minimize the cumulative difference between g_i and the reference curve g_0, however as each curve may have a different range we normalize them using

$$g_i^* = \frac{g_i - \min g_i}{\max g_i - \min g_i}. \tag{5.3}$$

From this point on, we can use an idea similar to Ramsay (2003) which suggests to substitute each curve by its ν-derivative, and our goal is to minimize

$$\tilde{\Delta}_i = \arg\min_{\Delta_i} \int_a^b \left\{ D^\nu \left[g_i^*(t + \Delta_i) - g_0^*(t) \right] \right\}^2 dt \tag{5.4}$$

for each curve i, $i = 1, \ldots, m$. In this work we decide to use $\nu = 0$. The procedure is described by Algorithm 5.1. We used the sample points as a grid to find $\tilde{\Delta}_i$.

Algorithm 5.1

1. Use Algorithm 3.1 to obtain the functional representation of each curve.
2. Normalize each curve using Equation (5.3).
3. Obtain $\tilde{\Delta}_i$ for each of the normalized curves.
4. Shift the non-normalized curves taking $g_i(t + \tilde{\Delta}_i)$.

Figure 5.11 presents a sample of $m = 7$ curves having phase and range variation plus a low variance noise. Figure 5.12 shows the registered curves after the application of Algorithm 5.1. All estimates using (3.6a), (3.7a) and (3.8a) were very similar and presented small MSE. Afterward we amplified the noise by taking a variance 25 times bigger, see Figure 5.13. Figure 5.12 shows the registered curves after the application of Algorithm 5.1. It seems that a good registration of the curves was achieved.

Acknowledgments. We would like to thank the referee for the comments and suggestions that made this work better and clearer. Also, we would like to thank Douglas Bates for making available the source code of the routine splineDesign() and Jim Ramsay for his ready response concerning the registration routines. This work was partially funded by CNPq 3000644/94-9, 475763/03-3 and 301054/93-2, FAPESP 01/00258-0 and 02/01554-5.

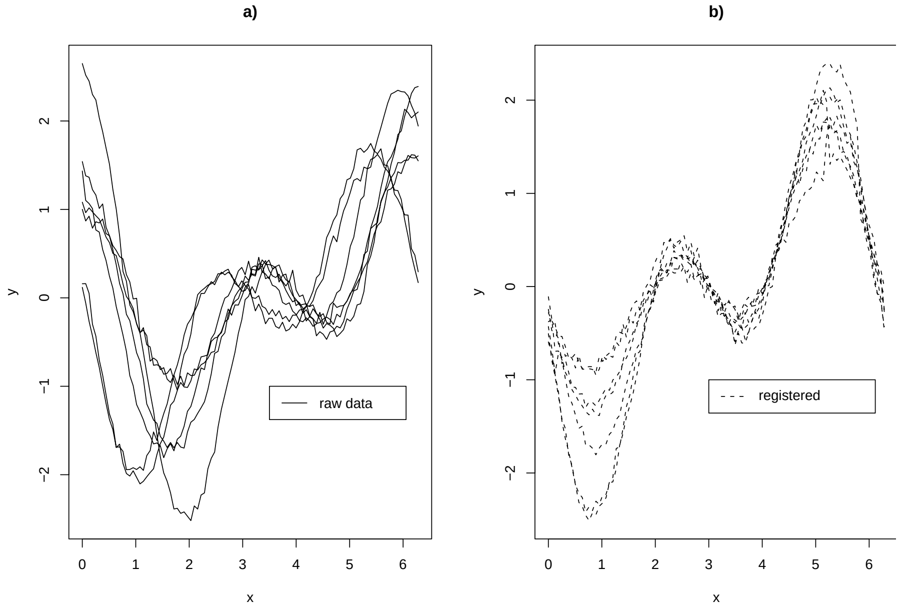

Figure 5.11 – A complex shifting case: a. Simulated curves; b. Curves obtained using Algorithm 5.1.

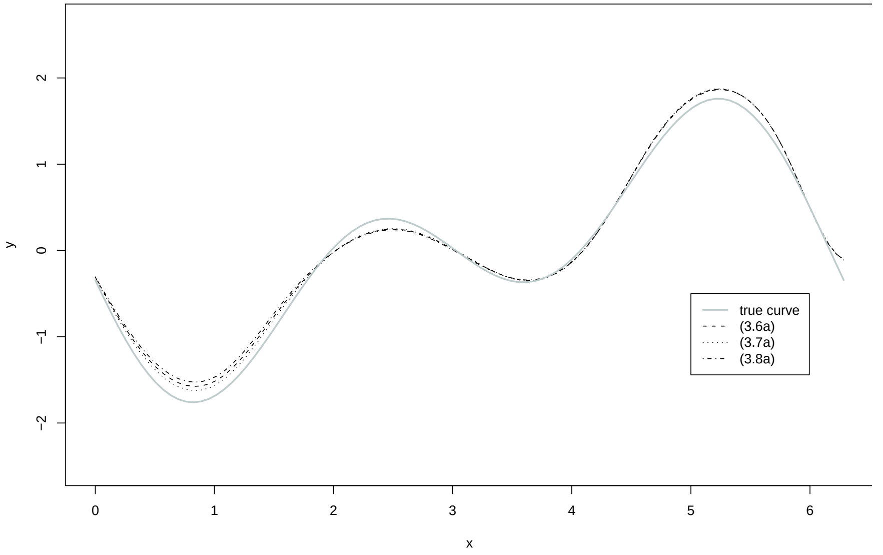

Figure 5.12 – A complex shifting case – Curves obtained using Algorithm 3.1.

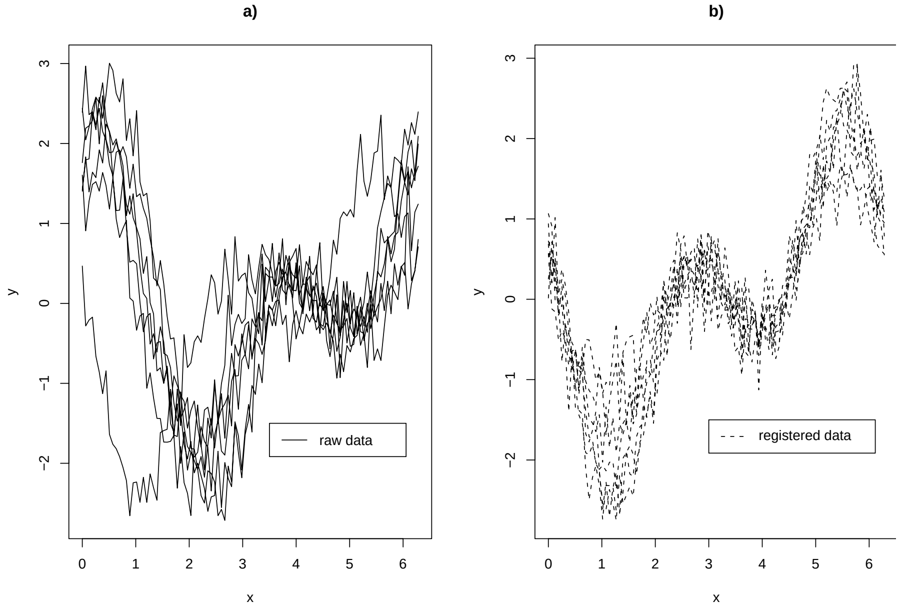

Figure 5.13 – A complex shifting case: a. Simulated curves plus a large variance noise; b. Curves registered using Algorithm 5.1.

REFERENCES

[1] G. Celeux and J. Diebolt, A stochastic approximation type EM algorithm for the mixture problem, *Stochastics Rep.*, **41** (1-2) (1992), 119–134.

[2] G. Celeux, D. Chauveau and J. Diebolt, Stochastic versions of the EM algorithm: an experimental study in the mixture case, *J. of Statist. Comput. Simul.*, **55** (1996), 287–314.

[3] D. Chauveau, A stochastic EM algorithm for mixtures with censored data, *J. Statist. Plann. Inference*, **46** (1) (1995), 1–25.

[4] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Statist. Soc. Ser. B*, **39** (1) (1977), 1–38. With discussion.

[5] R. Dias and D. Gamerman, A Bayesian approach to hybrid splines non-parametric regression, *Journal of Statistical Computation and Simulation*, **72** (4) (2002), 285–297.

[6] J. Diebolt and G. Celeux, Asymptotic properties of a stochastic EM algorithm for estimating mixing proportions, *Comm. Statist. Stochastic Models*, **9** (4) (1993), 599–613.

[7] J.O. Ramsay, Matlab, r and s-plus functions for Functional Data Analysis, *ftp://ego.psych.mcgill.ca/pub/ramsay/FDAfuns*, (2003).

[8] J.O. Ramsay and X. Li, Curve registration, *J.R. Stat. Soc. Ser. B Stat. Methodol.*, **60** (2) (1998), 351–363.

[9] J.O. Ramsay and B.W. Silverman, *Applied functional data analysis,* Springer Series in Statistics, Springer-Verlag, New York. Methods and case studies, (2002).

[10] J.A. Rice, Personal communication, (2000).