LASSO-type estimators for semiparametric nonlinear mixed-effects models estimation

Ana Arribas-Gil, Karine Bertin, Cristian Meza, Vincent Rivoirard

Statistics and Computing manuscript No.
(will be inserted by the editor)

LASSO-type estimators for Semiparametric Nonlinear Mixed-Effects Models Estimation

Ana Arribas-Gil · Karine Bertin · Cristian Meza · Vincent Rivoirard

Received: date / Accepted: date

Abstract Parametric nonlinear mixed effects models (NLMEs) are now widely used in biometrical studies, especially in pharmacokinetics research and HIV dynamics models, due to, among other aspects, the computational advances achieved during the last years. However, this kind of models may not be flexible enough for complex longitudinal data analysis. Semiparametric NLMEs (SNMMs) have been proposed as an extension of NLMEs. These models are a good compromise and retain nice features of both parametric and nonparametric models resulting in more flexible models than standard parametric NLMEs. However, SNMMs are complex models for which estimation still remains a challenge. Previous estimation procedures are based on a combination of log-likelihood approximation methods for parametric estimation and smoothing splines techniques for nonparametric estimation. In this work, we propose new estimation strategies in SNMMs. On the one hand, we use the Stochastic Approximation version of EM algorithm (SAEM) to obtain exact ML and REML estimates of the fixed effects and variance components. On the other hand, we propose a LASSO-type method to estimate the unknown nonlinear function. We derive oracle inequalities for this nonparametric estimator. We combine the two approaches in a general estimation procedure that we illustrate with simulations and through the analysis of a real data set of price evolution in on-line auctions.

Keywords LASSO · Nonlinear mixed-effects model · On-line auction · SAEM algorithm · Semiparametric estimation

Ana Arribas-Gil is supported by projects MTM2010-17323 and ECO2011-25706, Spain.

Karine Bertin is supported by projects FONDECYT 1090285 and ECOS/CONICYT C10E03 2010, Chile.

Cristian Meza is supported by project FONDECYT 11090024, Chile.

Ana Arribas-Gil
Departamento de Estadística, Universidad Carlos III de Madrid,
Calle Madrid 126, 28903 Getafe, Spain.
E-mail: ana.arribas@uc3m.es

Karine Bertin · Cristian Meza
CIMFAV-Facultad de Ingeniería, Universidad de Valparaíso,
Valparaíso, Chile.
E-mail: karine.bertin@uv.cl, E-mail: cristian.meza@uv.cl

Vincent Rivoirard
CEREMADE, CNRS-UMR 7534, Université Paris Dauphine,
Paris and INRIA Paris-Rocquencourt, Classic-team, France.
E-mail: vincent.rivoirard@dauphine.fr

1 Introduction

We consider the semiparametric nonlinear mixed effects model (SNMM) as defined by Ke and Wang (2001) in which we have n individuals and we observe:

$$y_{ij} = g(x_{ij}, \phi_i, f) + \varepsilon_{ij}, \qquad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d.}, \quad i = 1, \dots, N,\ j = 1, \dots, n_i, \tag{1}$$

where $y_{ij} \in \mathbb{R}$ is the jth observation in the ith individual, $x_{ij} \in \mathbb{R}^d$ is a known regression variable, g is a common known function governing within-individual behaviour and f is an unknown nonparametric function to be estimated. The random effects $\phi_i \in \mathbb{R}^p$ satisfy

$$\phi_i = A_i\beta + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \Gamma) \text{ i.i.d.}$$

where $A_i \in \mathcal{M}_{p,q}$ are known design matrices, $\beta \in \mathbb{R}^q$ is the unknown vector of fixed effects and we suppose that ε_{ij} and η_i are mutually independent.
The parameter of the model is (θ, f), where $\theta = (\beta, \Gamma, \sigma^2)$ belongs to a finite dimensional space whereas f belongs to an infinite dimensional space of functions denoted $\mathcal{H}$.

Ke and Wang (2001) consider the most common type of SNMM in practice, in which g is linear in f conditionally

on ϕ_i,

$$g(x_{ij},\phi_i,f) = a(\phi_i;x_{ij}) + b(\phi_i;x_{ij}) f(c(\phi_i;x_{ij})), \tag{2}$$

where a, b and c are known functions which may depend on i.

Different formulations of SNMM's have been recently used to model HIV dynamics (Wu and Zhang, 2002; Liu and Wu, 2007, 2008), time course microarray gene expression data (Luan and Li, 2004), circadian rhythms (Wang and Brown, 1996; Wang et al, 2003), as in the following example, or to fit pharmacokinetic and pharmacodynamic models (Wang et al, 2008), among many other applications.

Example 1 The following model was proposed by Wang and Brown (1996) to fit human circadian rhythms:

$$\begin{aligned} y_{ij} = \mu + \eta_{1i} + \exp(\eta_{2i})\, f\left(x_{ij} - \frac{\exp(\eta_{3i})}{1+\exp(\eta_{3i})}\right) + \varepsilon_{ij}, \\ \varepsilon_{ij} \sim \mathcal{N}(0,\sigma^2) \text{ i.i.d.} \\ \eta_i \sim \mathcal{N}(0,\Gamma) \text{ i.i.d.} \end{aligned}$$

for $i = 1,\ldots,N$, $j = 1,\ldots,n_i$, where y_{ij} is the physiological response of individual i at the jth time point x_{ij}. This model can be written in the general form (1) as:

$$\begin{aligned} y_{ij} &= g(x_{ij},\phi_i,f) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0,\sigma^2) \text{ i.i.d.}, \\ g(x_{ij},\phi_i,f) &= \phi_{1i} + \exp(\phi_{2i})\, f\left(x_{ij} - \frac{\exp(\phi_{3i})}{1+\exp(\phi_{3i})}\right) \\ \phi_i &= (1,0,0)'\mu + \eta_i, \quad \eta_i \sim \mathcal{N}(0,\Gamma) \text{ i.i.d.} \end{aligned}$$

where $\phi_i = (\phi_{1i},\phi_{2i},\phi_{3i})'$ and $\eta_i = (\eta_{1i},\eta_{2i},\eta_{3i})'$. In this example f represents the common shape of the observed curves, and ϕ_{1i}, $\exp(\phi_{2i})$, and $\exp(\phi_{3i})/(1+\exp(\phi_{3i}))$ stand for the individual vertical shift, individual amplitude and individual horizontal shift respectively. Here $d = 1$, $p = 3$, $q = 1$ and the parameter of the model is (μ,Γ,σ^2,f). This model was also used by Ke and Wang (2001) for modeling Canadian temperatures at different weather stations.

Let us introduce the following vectorial notation: $y_i = (y_{i1},\ldots,y_{in_i})'$, $y = (y_1',\ldots,y_N')'$, $\phi = (\phi_1',\ldots,\phi_N')'$, $\eta = (\eta_1',\ldots,\eta_N')'$, $g_i(\phi_i,f) = (g(x_{i1},\phi_i,f),\ldots,g(x_{in_i},\phi_i,f))'$, $g(\phi,f) = (g_1(\phi_1,f)',\ldots,g_N(\phi_n,f)')'$, $A = (A_1',\ldots,A_N')'$, $\widetilde{\Gamma} = \mathrm{diag}(\Gamma,\ldots,\Gamma)$ and $n = \sum_{i=1}^N n_i$. Then, model (1) can be written as:

$$\begin{aligned} y|\phi &\sim \mathcal{N}(g(\phi,f),\sigma^2 I_n) \\ \phi &\sim \mathcal{N}(A\beta,\widetilde{\Gamma}) \end{aligned} \tag{3}$$

where I_n is the identity matrix of dimension n, and the likelihood of observations y is:

$$\begin{aligned} p(y;(\theta,f)) &= \int p(y|\phi;(\theta,f))p(\phi;(\theta,f))d\phi \\ &= \int \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{\frac{-1}{2\sigma^2}\|y-g(\phi,f)\|^2\right\} \\ &\times \frac{1}{(2\pi)^{\frac{Np}{2}}|\Gamma|^{\frac{N}{2}}} \exp\left\{\frac{-1}{2}\|\widetilde{\Gamma}^{-1/2}(\phi-A\beta)\|^2\right\} d\phi \\ &= \frac{1}{(2\pi)^{\frac{n+Np}{2}}(\sigma^2)^{\frac{n}{2}}|\Gamma|^{\frac{N}{2}}} \\ &\times \int \exp\left\{\frac{-1}{2}\left(\frac{1}{\sigma^2}\|y-g(\phi,f)\|^2 + \|\widetilde{\Gamma}^{-1/2}(\phi-A\beta)\|^2\right)\right\} d\phi, \end{aligned} \tag{4}$$

where $\|\cdot\|$ is the L_2 norm. In their seminal paper, Ke and Wang consider a penalized maximum likelihood approach for the estimation of (θ,f). That is, they propose to solve

$$\max_{\theta,f} \{\ell(y;(\theta,f)) - n\lambda J(f)\} \tag{5}$$

where $\ell(y;(\theta,f))$ is the marginal log-likelihood, $J(f)$ is some roughness penalty and λ is a smoothing parameter. Moreover, they assume that f belongs to some reproducing kernel Hilbert space (RKHS) $\mathscr{H} = \mathscr{H}_1 \oplus \mathscr{H}_2$, where $\mathscr{H}_1$ is a finite dimensional space of functions, $\mathscr{H}_1 = span\{\psi_1,\ldots,\psi_M\}$, and $\mathscr{H}_2$ is a RKHS itself. Since the nonlinear function f interacts in a complicated way with the random effects and the integral in (4) is intractable, they replace $\ell(y;(\theta,f))$ by a first-order linearization of the likelihood with respect to the random effects. Then, they propose to estimate (θ,f) by iterating the following two steps:

i) given an estimate of f, get estimates of θ and ϕ by fitting the resultant nonlinear mixed model by linearizing the log-likelihood (replacing ℓ by $\tilde{\ell}$). In practice they use the R-function `nlme` (Pinheiro and Bates, 2000) to solve this step.

ii) given an estimate of θ, $\hat{\theta}$, estimate f as the solution to

$$\max_{f\in\mathscr{H}} \left\{\tilde{\ell}(y;(\hat{\theta},f,\tilde{\phi})) - n\lambda J(f)\right\}.$$

Since in ii) the approximated log-likelihood involves a bounded linear functional, the maximizer in $\mathscr{H}$ of $\tilde{\ell}(y;(\hat{\theta},f,\tilde{\phi})) - N\lambda J(f)$ given $\hat{\theta}$ and $\tilde{\phi}$ belongs to a finite dimensional space and it is estimated as a linear combination of functions from $\mathscr{H}_1$ and $\mathscr{H}_2$. Conceptually, the whole approach is equivalent to solving (5) not on $\mathscr{H}$ but on a finite-dimensional approximation space of $\mathscr{H}$ at each iteration. As it is discussed in that article, despite of the lack of an exact solution, the spline smoothing method provides good results and its use in this framework is largely justified. However, the method relies on prior knowledge of the nonlinear function f and provides

better results when this kind of information is available.

In practice, the Ke and Wang's method is implemented in the R package *assist* (Wang and Ke (2004)) and in particular in the *snm* function which is directly related with the *nlme* function.
As for the parametric estimation, it is important to point out some drawbacks of the approximated methods based on linearization of the log-likelihood, such as the first-order linearization conditional estimates (FOCE) algorithm used in the *snm* funtion (Wang and Ke (2004)). It has been shown that they can produce inconsistent estimates of the fixed effects, in particular when the number of measurements per subject is not large enough (Ramos and Pantula (1995); Vonesh (1996); Ge et al (2004)). Furthermore, simulation studies have shown unexpected increases in the type I error of the likelihood ratio and Wald tests based on these linearization methods (Ding and Wu (2001)). In addition, from of statistical point of view, the theoretical basis of this linearization-based method is weak.

Since estimation in SNMMs is an important problem and a difficult task from which many challenging aspects arise, in this paper we propose an alternative estimation procedure to tackle some of these points. On the one hand, for the parametric step we will focus on the maximization of the exact likelihood. We propose to use a stochastic version of the EM algorithm, the so-called SAEM algorithm introduced by Delyon et al (1999) and extended by Kuhn and Lavielle (2005) for nonlinear mixed models, to estimate θ without any approximation or linearization. This stochastic EM algorithm replaces the usual E step of EM algorithm (Dempster et al, 1977) by a simulation step and a stochastic procedure, and converges to a local maximum of the likelihood. The SAEM has been proved to be computationally much more efficient than other stochastic algorithms as for example the classical Monte Carlo EM (MCEM) algorithm (Wei and Tanner, 1990) thanks to a recycling of the simulated variables from one iteration to the next (see Kuhn and Lavielle (2005)). Indeed, previous attempts to perform exact ML estimation in SNMMs have been discarded because of the computational problems related to the use of an MCEM algorithm (see Liu and Wu (2007, 2008, 2009)). Moreover we use a Restricted Maximum Likelihood (REML) version of the SAEM algorithm to correct bias estimation problems of the variance parameters following the same strategy as Meza et al (2007).
On the other hand, for the nonparametric step we will propose a LASSO-type method for the estimation of f. The popular LASSO estimator (least absolute shrinkage and selection operator, Tibshirani (1996)) based on ℓ_1 penalized least squares, has been extended in the last years to nonparametric regression (see for instance Bickel et al (2009)). It has been also used by Schelldorfer et al (2011) in high-dimensional linear mixed-effects models. In the nonparametric context, the idea is to reconstruct a sparse approximation of f with linear combinations of elements of a given set of functions $\{f_1, \ldots, f_M\}$, called dictionary. That is, we are implicitly assuming that f can be well approximated with a small number of those functions. In practice, for the nonparametric regression problem, the dictionary can be a collection of basis functions from different bases (splines with fixed knots, wavelets, Fourier, etc.). The difference between this approach and the smoothing splines, is that the selection of the approximation function space is done automatically and based on data among a large collection of possible spaces spanned by very different functions. This is particularly important in situations in which little knowledge about f is available. This approach allows us to construct a good approximation of the nonparametric function which is sparse thanks to the large dictionary. The sparsity of the approximation gives a model more interpretable and since few coefficients have to be estimated, this minimizes the estimation error. The LASSO algorithm allows to use the dictionary approach to select a sparse approximation, unlike to wavelet thresholding or ℓ_0- penalization. Moreover the LASSO algorithm has a low computational cost since it is based on a convex penalty.

We can summarize our iterative estimation procedure as:

i) given $\hat{f}$, an estimate of f, get estimates of θ and ϕ by fitting the resulting NLME with the SAEM algorithm (using either ML or REML).
ii) given estimates of θ and ϕ, solve the resulting nonparametric regression problem using a LASSO-type method.

The rest of the article is organized as follows. In Section 2.1 we describe the SAEM algorithm and its REML version in the framework of SNMMs. In Section 3 we propose a LASSO-type method for the estimation of f in the resulting nonparametric regression problem after estimation of θ and ϕ. Oracle inequalities and subset selection properties for the proposed estimator are provided in the Supplementary Material. In Section 4, we describe the algorithm that combines both procedures to perform joint estimation of (θ, f) in the SNMM. Finally, in Section 5, we illustrate our method through a simulation study and the analysis of price dynamics in on-line auction data. We conclude the article in Section 6.

2 Estimation of the finite-dimensional parameters

2.1 SAEM estimation of θ and ϕ

In this subsection we consider that we have an estimate of f, $\hat{f}$, obtained in the previous estimation step that does not change during the estimation of θ. Thus, we can proceed

as if f was a known nonlinear function and we fall into the SAEM estimation of nonlinear mixed-effects model framework (see Kuhn and Lavielle (2005)). In this setting, convergence of the algorithm to a local maximum of the likelihood is guaranteed. In fact, note that since the estimation of f is performed by solving a nonparametric regression problem with regression variables $c(\hat{\phi}_i; x_{ij}), i = 1,\dots,N,\ j = 1,\dots,n_i$ (see Section 3), it will depend on the estimated value of ϕ at the precedent iteration. Then, we will note $\hat{f}_-$ the current estimated function.

The complete likelihood for model (1) is:

$$\begin{aligned} p(y,\phi;\theta) &= p(y|\phi;\theta)p(\phi;\theta) \\ &= \frac{1}{(2\pi)^{\frac{n+Np}{2}}(\sigma^2)^{\frac{n}{2}}|\Gamma|^{\frac{N}{2}}} \exp\left\{ \frac{-1}{2}\left(\frac{1}{\sigma^2}\|y - g(\phi,\hat{f}_-)\|^2 \right.\right. \\ &\qquad \left.\left. + \|\widetilde{\Gamma}^{-1/2}(\phi - A\beta)\|^2 \right)\right\} \end{aligned}$$

where $n = \sum_{i=1}^{N} n_i$. Then, the complete log-likelihood is:

$$\begin{aligned} \log p(y,\phi;\theta) = -\frac{1}{2}\Big\{ & C + n\log\sigma^2 + N\log|\Gamma| \\ & + \frac{1}{\sigma^2}\|y - g(\phi,\hat{f}_-)\|^2 + \sum_{i=1}^{N}(\phi_i - A_i\beta)'\Gamma^{-1}(\phi_i - A_i\beta) \Big\} \end{aligned} \tag{6}$$

where C is a constant that does not depend on θ.

The distribution of the complete-data model belongs to the exponential family, that is $\log p(y,\phi;\theta) = -\Psi(\theta) + \langle S(y,\phi), \Phi(\theta)\rangle$, where $\langle\cdot,\cdot\rangle$ stands for the scalar product and $S(y,\phi)$ is the sufficient statistics. The EM algorithm in this framework would involve the computation of $\mathbb{E}[S(y,\phi)|y;\theta^{(k)}]$ in the E step, which in our case is intractable. The SAEM algorithm replaces, at each iteration, the step E by a simulation step (S) of the missing data (ϕ) and an approximation step (A). Then, iteration k of the SAEM algorithm writes:

- S step: simulate m values of the random effects, $\phi^{(k+1,1)},\dots,\phi^{(k+1,m)}$, from the conditional law $p(\cdot|y;\theta^{(k)})$.
- A step: update s_{k+1} according to:

$$s_{k+1} = s_k + \chi_k\left[\frac{1}{m}\sum_{l=1}^{m} S(y,\phi^{(k+1,l)}) - s_k\right].$$

- M step: update the value of θ:

$$\theta^{(k+1)} = \arg\max_{\theta}\{-\Psi(\theta) + \langle s_{k+1}, \Phi(\theta)\rangle\} \tag{7}$$

where $(s_k)_k$ is initialized at s_0 and $(\chi_k)_k$ is a decreasing sequence of positive numbers which accelerates the convergence (Kuhn and Lavielle, 2004). The role of the sequence $(\chi_k)_k$ is crucial in the SAEM algorithm since it performs a smoothing of the calculated likelihood values from one iteration to another. In practice, this smoothing parameter is defined as follows. During the first L iterations, $\chi_k = 1$, and from iteration $(L+1)$ the smoothing parameter starts to decrease in order to stabilize the estimates and provide a faster convergence towards the true ML estimates. For example, Kuhn and Lavielle (2005) recommend to take $\chi_k = (k-L)^{-1}$ for $k \geq (L+1)$. The choices of the total number of iterations, K, and of L are then crucial. In order to define these constants, following Jank (2006) and Meza et al (2009), we may use a graphical approach based on the likelihood difference from one iteration to the next one and monitor SAEM by estimating its progress towards θ_{ML} by using the property of increasing likelihood of the EM algorithm (see for more details (Meza et al, 2009)). Then, the total number of iterations can be fixed and the smoothing step can be defined. However, it is important to note that this procedure implies to run the SAEM algorithm twice. Furthermore, as all EM-type algorithms, SAEM is sensitive to the choice of the initial values.

From (6), the sufficient statistics for the complete model are given by

$$\begin{aligned} s_{1,i,k+1} &= s_{1,i,k} + \chi_k\left[\frac{1}{m}\sum_{l=1}^{m}\phi_i^{(k+1,l)} - s_{1,i,k}\right], \quad i = 1,\dots,N \\ s_{2,k+1} &= s_{2,k} + \chi_k\left[\frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{N}\phi_i^{(k+1,l)}\phi_i^{(k+1,l)'} - s_{2,k}\right] \\ s_{3,k+1} &= s_{3,k} + \chi_k\left[\frac{1}{m}\sum_{l=1}^{m}\|y - g(\phi^{(k+1,l)},\hat{f}_-)\|^2 - s_{3,k}\right]. \end{aligned}$$

Now, $\theta^{(k+1)}$ is obtained in the maximization step as follows:

$$\begin{aligned} \beta^{(k+1)} &= \left(\sum_{i=1}^{N} A_i'\Gamma^{(k)^{-1}}A_i\right)^{-1}\sum_{i=1}^{N} A_i'\Gamma^{(k)^{-1}}s_{1,i,k+1} \\ \Gamma^{(k+1)} &= \frac{1}{N}\left(s_{2,k+1} - \sum_{i=1}^{N} A_i\beta^{(k+1)}s_{1,i,k+1}' - \sum_{i=1}^{N} s_{1,i,k+1}\left(A_i\beta^{(k+1)}\right)' \right. \\ &\qquad \left. + \sum_{i=1}^{N} A_i\beta^{(k+1)}\left(A_i\beta^{(k+1)}\right)'\right) \\ \sigma^{2^{(k+1)}} &= \frac{s_{3,k+1}}{n}. \end{aligned}$$

When the simulation step cannot be directly performed, Kuhn and Lavielle (2004) propose to combine this algorithm with a Markov Chain Monte Carlo (MCMC) procedure. Then, the simulation step becomes:

- S step: using $\phi^{(k,l)}$, draw $\phi^{(k+1,l)}$ with transition probability $\Pi_{\theta^{(k)}}(\cdot|\phi^{(k,l)})$, $l = 1,\dots,m$,

that is, $(\phi^{(k+1,1)}),\dots,(\phi^{(k+1,m)})$ are m Markov chains with transition kernels $\left(\Pi_{\theta^{(k)}}\right)$. In practice, these Markov chains are generated using a Hastings-Metropolis algorithm (see Kuhn and Lavielle (2005) for details).

With respect to the number of chains, the convergence of the whole algorithm to a local maximum of the likelihood is

granted even for $m = 1$. Greater values of m can accelerate the convergence, but in practice m is always lower than 10. This is the main difference with the MCEM algorithm, in which very large samples of the random effects have to be generated to obtain convergence of the algorithm.

2.2 REML estimation of variance components

It is well known that the maximum likelihood estimator of variance components in mixed effects models can be biased downwards because it does not adjust for the loss of degrees of freedom caused by the estimation of the fixed effects. This is also true in the context of SNMMs as Ke and Wang (2001) point out in their paper.

To overcome this problem we consider restricted maximum likelihood (REML) estimation. REML, as originally formulated by Patterson and Thompson (1971) in the context of linear models, is a method that corrects this problem by maximizing the likelihood of a set of linear functions of the observed data that contain none of the fixed effects of the model. But this formulation does not directly extend beyond linear models, where in general it is not possible to construct linear functions of the observed data that do not contain any of the fixed effects. However, in the case of nonlinear models, other alternative formulations of REML have been proposed. Here, we will consider the approach of Harville (1974), that consists in the maximization of the likelihood after integrating out the fixed effects. To perform this integration we follow Foulley and Quaas (1995) and consider the fixed effects as random with a flat prior. The combination of this REML approach with the SAEM algorithm in the context of nonlinear mixed effects models has been studied recently by Meza et al (2007). The authors showed the efficiency of the method against purely ML estimation performed by SAEM and against REML estimation based on likelihood approximation methods.

Following these ideas we note $z = (\phi, \beta)$ the random effects and $\tilde{\theta} = (\Gamma, \sigma^2)$ the new parameter of the model. As in the general case, the simulation step is performed through an MCMC procedure. Here, since we have to draw values from the joint distribution of $(\phi, \beta)|y; \tilde{\theta}^{(k)}$, we use a Gibbs scheme, i.e., we iteratively draw values from the conditional distributions of $\phi|y, \beta^{(k)}; \tilde{\theta}^{(k)}$ and $\beta|y, \phi^{(k)}; \tilde{\theta}^{(k)}$. Then, we use again a Hastings-Metropolis algorithm to obtain approximations of these conditional distributions.

Finally, iteration k of the SAEM-REML algorithm for model (3) writes:

- S step: using $z^{(k,l)} = (\phi^{(k,l)}, \beta^{(k,l)})$, simulate $z^{(k+1,l)} = (\phi^{(k+1,l)}, \beta^{(k+1,l)})$, $l = 1, \ldots, m$ with a Metropolis-within-Gibbs scheme.
- A step: update $\tilde{s}_{k+1}$ by $\tilde{s}_{k+1} = \tilde{s}_k + \chi_k \left[\frac{1}{m} \sum_{l=1}^{m} \tilde{S}(y, z^{(k+1,j)}) - \tilde{s}_k \right]$, namely:

$$\tilde{s}_{1,k+1} = \tilde{s}_{1,k} + \chi_k \left[\frac{1}{m} \sum_{l=1}^{m} \sum_{i=1}^{N} \eta_i^{(k+1,l)} \eta_i^{(k+1,l)'} - \tilde{s}_{1,k} \right]$$

$$\tilde{s}_{2,k+1} = \tilde{s}_{2,k} + \chi_k \left[\frac{1}{m} \sum_{l=1}^{m} \|y - g(z^{(k+1,l)}, \hat{f}_-)\|^2 - \tilde{s}_{2,k} \right]$$

  where $\eta_i^{(k+1,l)} = \phi_i^{(k+1,l)} - A_i \beta^{(k+1,l)}$.
- M step: update $\tilde{\theta}$ by $\tilde{\theta}^{(k+1)} = \arg\max_{\tilde{\theta}} \{-\Psi(\tilde{\theta}) + \langle \tilde{s}_{k+1}, \Phi(\tilde{\theta}) \rangle\}$, namely:

$$\Gamma^{(k+1)} = \frac{\tilde{s}_{1,k+1}}{N} \qquad \text{and} \qquad \sigma^{2^{(k+1)}} = \frac{\tilde{s}_{2,k+1}}{n}. \tag{8}$$

In many situations, it is important to obtain inference on the fixed effects in the context of REML estimation of variance components. Following Meza et al (2007), estimation of fixed effects can be directly obtained as a by-product of the SAEM-REML algorithm via the expectation of the conditional distribution of the fixed effects given the observed data, the estimate, $\hat{f}$, of the unknown function f and the REML estimates of the variance-covariance components. This estimator makes sense in an Empirical Bayes framework.

3 Estimation of the function f using a LASSO-type method

In this part, our objective is to estimate f in the model (1) using the observations $y_{i,j}$ and assuming that for $i = 1, \ldots, N$ we have $\phi_i = \hat{\phi}_i$ and $\sigma^2 = \hat{\sigma}^2$ where the estimates $\hat{\phi}_i$ and $\hat{\sigma}^2$ have been obtained in the precedent SAEM step. Since g satisfies (2), model (1) can be rewritten as

$$\tilde{y}_{ij} = b(\phi_i; x_{ij}) f(\tilde{x}_{ij}) + \varepsilon_{ij}, \quad i = 1 \ldots, N, \ j = 1, \ldots, n_i$$

with $\tilde{y}_{ij} = y_{ij} - a(\phi_i; x_{ij})$ and $\tilde{x}_{ij} = c(\phi_i; x_{ij})$. Of course, since the $\hat{\phi}_i$'s and $\hat{\sigma}^2$ depend on the observations, the distribution of $\hat{\sigma}^{-1}\tilde{y}_{ij}$ is no longer Gaussian and the ε_{ij}'s are not i.i.d. but dependent. But in the sequel, to be able to derive theoretical results, we still assume that

$$\varepsilon_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \tag{9}$$

where the value of σ^2 is given by $\hat{\sigma}^2$. Simulation studies of Section 5 show that this assumption is reasonable. However, note that (9) is true at the price of splitting the data set into two parts: the first part for estimating θ and ϕ, the second part for estimating f. Now, reordering the observations, it is

equivalent to observing $(y_1,\dots,y_n)$ with $n=\sum_{i=1}^N n_i$, such that

$$y_i = b_i f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0,\sigma^2) \text{ i.i.d.} \tag{10}$$

where the b_i's and the design $(x_i)_{i=1,\dots,n}$ are known and depend on the estimators of the precedent SAEM step and the ε_i's are random variables with variance σ^2 estimated by $\hat{\sigma}^2$. Note that the notation y_i, $i=1,\dots,n$, does not correspond to the original observations in the SNMM or to any of the values introduced in the previous sections, and it is used in this section for the sake of simplicity. Without loss of generality, we suppose that $b_i \neq 0$ for all $i=1,\dots,n$.

In the sequel, our objective is then to estimate f nonparametrically in model (10). A classical method would consist in decomposing f on an orthonormal basis (Fourier basis, wavelets,...) and then to use a standard nonparametric procedure to estimate the coefficients of f associated with this basis (ℓ_0-penalization, wavelet thresholding,...). In the same spirit as Bertin et al (2011) who investigated the problem of density estimation, we wish to combine a more general dictionary approach with an estimation procedure leading to fast algorithms. The dictionary approach consists in proposing estimates that are linear combinations of various types of functions. Typically, the dictionary is built by gathering together atoms of various classical orthonormal bases. This approach offers two advantages. First, with a more wealthy dictionary than a classical orthonormal basis, we aim at obtaining sparse estimates leading to few estimation errors of the coefficients. Secondly, if the estimator is sparse enough, interesting interpretations of the results are possible by using the set of the non-zero coefficients, which corresponds to the set of functions of the dictionary "selected" by the procedure. For instance, we can point out the frequency of periodic components of the signal if trigonometric functions are selected or local peaks if some wavelets are chosen by the algorithm. Both aspects are illustrated in the next sections. ℓ_0-penalization or thresholding cannot be combined with a dictionary approach if we wish to obtain fast and good algorithms. But LASSO-type estimators based on ℓ_1-penalization, leading to minimization of convex criteria, constitute a natural tool for the dictionary approach. Furthermore, unlike ridge penalization or more generally ℓ_p-penalization with $p>1$, ℓ_1-penalization leads to sparse solutions for the minimization problem, in the sense that if the tuning parameter is large enough some coefficients are exactly equal to 0 (see Tibshirani (1996)).

There is now huge literature on LASSO-type procedures. From the theoretical point of view and in the specific context of the regression model close to (10), we mention that LASSO procedures have already been studied by Bunea et al (2006), Bunea et al (2007a), Bunea et al (2007b), Bunea (2008), Bickel et al (2009), van de Geer (2010), and Bühlmann and van de Geer (2011) among others.

In our setting, the proposed procedure is the following. For $M \in \mathbb{N}^*$, we consider a set of functions $\{\varphi_1,\dots,\varphi_M\}$, called the *dictionary*. We denote for $\lambda \in \mathbb{R}^M$,

$$f_\lambda = \sum_{j=1}^M \lambda_j \varphi_j.$$

Our objective is to find good candidates for estimating f which are linear combinations of functions of the dictionary, i.e. of the form f_λ. We consider, for $\lambda \in \mathbb{R}^M$

$$\text{crit}(\lambda) = \frac{1}{n}\sum_{i=1}^n (y_i - b_i f_\lambda(x_i))^2 + 2\sum_{j=1}^M r_{n,j}|\lambda_j|,$$

where $r_{n,j} = \sigma\|\varphi_j\|_n\sqrt{\frac{\tau \log M}{n}}$ with $\tau>0$ and for a function h

$$\|h\|_n^2 = \frac{1}{n}\sum_{i=1}^n b_i^2 h^2(x_i).$$

We call the LASSO estimator $\hat{\lambda}$ the minimizer of $\lambda \longmapsto \text{crit}(\lambda)$ for $\lambda \in \mathbb{R}^M$ and we denote $\hat{f} = f_{\hat{\lambda}}$.

The function $\lambda \longmapsto \text{crit}(\lambda)$ is the sum of two terms: the first one is a goodness-of-fit criterion based on the ℓ_2-loss and the second one is a penalty term that can be viewed as the weighted ℓ_1-norm of λ.

Before going further, let us discuss the important issue of tuning. In our context, the tuning parameter is the constant τ. From a theoretical point of view (see Theorem 1 in the supplementary material), the benchmark value for τ is 2. In the sequel, τ will be chosen satisfying two criteria: to be as close as possible to this benchmark value and allowing the stability of the SAEM algorithm. In Section 5, we will see that sometimes we choose values of τ smaller than 2 but relatively close of it, in particular to obtain the convergence of the variance components estimates, which is always challenging in NLME models.

Once we have chosen a value for τ satisfying these two criteria, the numerical scheme of the nonparametric step is the following:

- Using the estimates of the ϕ_i's and of σ^2 obtained in the previous iteration of SAEM, compute for $i=1,\dots,n$, the observations y_i, the constants b_i and the design x_i.
- Evaluate the dictionary $\{\varphi_1,\dots,\varphi_M\}$ at the design and calculate $r_{n,j}$.
- Obtain the LASSO estimates $\hat{\lambda}$ and $f_{\hat{\lambda}}$.

In practice, there exist many efficient algorithms to tackle this third point, namely, the minimization on λ of $\text{crit}(\lambda)$. For the implementation of our estimation procedure we have considered the approach used by Bertin et al (2011) which consists in using the LARS algorithm.

Numerical results of our procedure are presented in next sections but we also validate our approach from a theoretical point of view. Theoretical results are presented in the supplementary material. We prove oracle inequalities and properties of support for sparse functions under the mild assumption $\log(M) = o(n)$. Oracle inequalities ensure that the LASSO estimator of f behaves as well as the best linear combination of functions of the dictionnary. Moreover, we obtain that if the function f is a sparse linear combination of functions from the dictionnary, then the support of the LASSO estimator (functions of the dictionary selected in the LASSO estimator) is included in the support of the function f. These results are generalizations of the results of Bunea et al (2006), Bunea et al (2007a), Bunea et al (2007b), van de Geer (2010) and Bunea (2008) and they are obtained under more general assumptions on the dictionnary. In particular, in our results, the functions of the dictionary do not need to be bounded independently of n and M, which allow us to take wavelet functions.

4 Estimation algorithm and inferences

We propose the following estimation procedure for semiparametric estimation of (θ, f) in model (3), combining the algorithms described in sections 2.1 and 3:

Estimation Algorithm - ML version: at iteration k,

- Given the current estimate of θ, $\theta^{(k)} = (\beta^{(k)}, \Gamma^{(k)}, \sigma^{2(k)})$, and m sampled values of the random effects $\phi^{(k,l)}$, $l = 1,\dots,m$, update the estimates of f, $f^{(k,l)}$, $l = 1,\dots,m$, with the algorithm described in Section 3.
- Given the current estimates of f, $f^{(k,l)}$, $l = 1,\dots,m$, sample m values of the random effects $\phi^{(k,l)}$, $l = 1,\dots,m$, and update the value of θ, $\theta^{(k+1)} = (\beta^{(k+1)}, \Gamma^{(k+1)}, \sigma^{2(k+1)})$ with algorithm (7). (11)

Estimation Algorithm - REML version: at iteration k,

- Given the current estimate of $\tilde{\theta}$, $\tilde{\theta}^{(k)} = (\Gamma^{(k)}, \sigma^{2(k)})$, and m sampled values of the missing data $z^{(k,l)} = (\phi^{(k,l)}, \beta^{(k,l)})$, $l = 1,\dots,m$, update the estimates of f, $f^{(k,l)}$, $l = 1,\dots,m$, with the algorithm described in Section 3.
- Given the current estimates of f, $f^{(k,l)}$, $l = 1,\dots,m$, sample m values of the missing data $z^{(k+1,l)} = (\phi^{(k+1,l)}, \beta^{(k+1,l)})$, $l = 1,\dots,m$, and update the value of $\tilde{\theta}$, $\tilde{\theta}^{(k+1)} = (\Gamma^{(k+1)}, \sigma^{2(k+1)})$ with algorithm (8). (12)

As it is explained in Section 2.1, for parametric estimation (SAEM or SAEM-REML algorithms alone) the number of chains, m, can be set to 1, which still guarantees the convergence towards a local maximum of the log-likelihood. Higher values of m, may accelerate the convergence of the algorithms (but in practice, m is always lower than 10).
For the global semiparametric estimation procedure, we extend this idea of "parallel chains" of values to the estimation of f. Indeed, at iteration k, the estimation of f depends on the value of the missing data, and thus, from m sampled values $z^{(k,1)},\dots,z^{(k,m)}$ we obtain m estimates of f, $f^{(k,1)},\dots,f^{(k,m)}$ (see Section 3). Then, in the second step, we use each one of these different estimates of f in parallel to perform parametric estimation (using $f^{(k,l)}$ to sample $z^{(k+1,l)}$ and replacing $\hat{f}_-$ by $f^{(k,l)}$ in (8) for the estimation of $\tilde{\theta}$). This is in the case of the REML version of the algorithm, but the same idea underlies the ML version.

Inferences on model and individual parameters, β, Γ, σ^2 and ϕ, are performed as in NLMEs (see Kuhn and Lavielle (2005) and Meza et al (2007)). For inferences on the nonlinear function f, we propose an empirical approach based on the fact that our algorithm automatically provides large samples of estimates of f. Indeed, at each iteration of algorithms (11) and (12) we obtain m estimates of f. The last iterations of the algorithms typically correspond to small values of χ_k in algorithms (7) and (8), see Section 5 for the details. This can be seen as a phase in which the estimates of parameters are stabilized since we assume that convergence has been reached. Let us note by K and $L < K$ the total number of iterations and the number of iterations in the "stabilization phase" of the algorithm. Then, by considering the last $L_0 < L$ iterations of the algorithm, we get a large sample of estimates of f: $f^{(k,l)}$, $l = 1,\dots,m$, $k = K - L_0 + 1,\dots,K$. These $m \times L_0$ estimates of f are obtained conditionally on values of θ which are supposed to be close to the corresponding ML or REML estimates. Then, we obtain a point estimate for f as:

$$\hat{f} = \frac{1}{m \times L_0} \sum_{k=K-L_0+1}^{K} \sum_{l=1}^{m} f^{(k,l)}. \tag{13}$$

We think that it will be interesting to study how to exploit the estimates $f^{(k,l)}$ to obtain pointwise confidence intervals for $f(x)$. An intuitive empirical pointwise $(1-\alpha)100\%$ confidence interval for $f(x)$ could be dfined as follows:

$$\left(\hat{f}(x) - z_{\frac{\alpha}{2}} \sqrt{\frac{S^2_{f(x)}}{m \times L_0}} \, , \, \hat{f}(x) + z_{\frac{\alpha}{2}} \sqrt{\frac{S^2_{f(x)}}{m \times L_0}} \right). \tag{14}$$

where $S^2_{f(x)} = \frac{1}{m \times L_0 - 1} \sum_{k=K-L_0+1}^{K} \sum_{l=1}^{m} (f^{(k,l)}(x) - \hat{f}(x))^2$ and $z_{\frac{\alpha}{2}}$ is the $1 - \frac{\alpha}{2}$ percentile of a standard normal distribution. This interval is of course not a true $(1-\alpha)100\%$ confidence interval for $f(x)$ but constitutes an approximation of it. It provides a starting point for further research on how function samples generated by semiparametric stochastic approximation algorithms, such us *saem-lasso*, can be used for inference.

5 Application to synthetic and real data

Since our procedure consists in the combination of a parametric and a nonparametric estimation algorithm, one may be interested in evaluating the performance of both components separately. In Section 5.1 we provide a simulation study to compare only the parametric versions of our method and Ke and Wang's procedure. In Section 5.2 we compare both procedures in the whole semiparametric setting.

5.1 Simulation study: parametric estimation

As a first step, we want to validate through simulation our parametric estimation strategy alone, based on the SAEM algorithm, and to compare it, in the framework of SNMMs, to the FOCE method implemented in Ke and Wang (2001) via the *nlme* function. In order to be able to assess only the differences induced by the use of different parametric estimation algorithms, we will use the same nonparametric estimation algorithm for the estimation of f, namely the procedure proposed by Ke and Wang (2001). In Section 5.2, we compare the whole versions, including nonparametric estimation, of both approaches.
To this end, we performed the following simulation study based in Ke and Wang (2001) where data were generated from the model:

$$y_{ij} = \phi_{1i} + \exp(\phi_{2i}) 2f\left(\frac{j}{N} - \frac{\exp(\phi_{3i})}{1+\exp(\phi_{3i})}\right) + \varepsilon_{ij},\ i = 1\ldots,N,\ j = 1,\ldots,J,$$

where $\varepsilon_{ij} \sim \mathcal{N}(0,\sigma^2)$ and $\phi_i = (\phi_{1i},\phi_{2i},\phi_{3i})' \sim \mathcal{N}(\mu,\Gamma)$ with $\mu = (\mu_1,\mu_2,\mu_3)'$. The nonlinear function was set to $f(t) = \sin(2\pi t)$. As in the original setting, we choose a complex scenario with *small* sizes of individuals and observations and with *high* variance values: $N = J = 10$, $\mu = (1,0,0)'$, $\sigma^2 = 1$ and Γ is diagonal with $diag(\Gamma) = (1,0.25,0.16)$.

These data were analyzed using two semiparametric procedures: our SAEM based method combined with the nonparametric algorithm of Ke and Wang's (called *semi-SAEM*) and Ke and Wang's procedure for semiparametric models (called *snm*). For the SAEM algorithm, we used 80 iterations and the following sequence (χ_k): $\chi_k = 1$ for $1 \le k \le 50$ and $\chi_k = 1/(k-50)$ for $51 \le k \le 80$. We also considered $m = 5$ chains in each iteration. For the nonparametric estimation algorithm common to both procedures, following Ke and Wang (2001) we considered that f is periodic with period equal to 1 and $\int_0^1 f = 0$, i.e. $f \in W_2^0(per) = W_2(per) \ominus span\{1\}$ where $W_2(per)$ is the periodic Sobolev space of order 2 in L^2 and $span\{1\}$ represents the set of constant functions. The same initial values were used for both methods: $\mu_0 = (1,0,0)$, $\sigma_0^2 = 2$ and $\mathrm{diag}(\Gamma_0) = (\gamma_1^0,\gamma_2^0,\gamma_3^0) = (1,0.3,0.1)$.

Tables 1 and 2 summarize the performance of both methods over 100 simulated data sets. For each parameter we show the sample mean, the mean squared error ($MSE(\hat{\theta}) = \frac{1}{100}\sum_{i=1}^{100}(\theta - \hat{\theta}_i)^2$), and a 95% confidence interval computed over the total number of simulations.

We also compared the REML estimates obtained with our method and with *snm* (using the REML version of *nlme*) for the same simulated data sets. The results are summarized in Tables 3 and 4. It can be seen that the mean values for the REML estimates obtained with both procedures were closer to the simulated values, especially for the parameter γ_1. Moreover, the individual confidence intervals of REML estimates of this parameter, at a 95% level, include the true value for these parameters on the contrary to the ML estimates, showing that REML versions of the algorithms were able to correct the bias observed with ML. If we compare our method and *snm*, for both procedures ML and REML, we obtained results that are similar but it seems that our REML estimates are closer to the simulated values than those obtained with Ke and Wang's method. Furthermore, we can observe that our REML version, in comparison with our ML method, allows to reduce the bias of estimation of variance components in a better way. For instance, in Tables 2 and 4, we see that, for γ_1, we reduce the bias in almost 93% with our REML method whereas with Ke and Wang's REML method this reduction is only of 27%. Finally, let us point out that fixed effects estimates are more accurate with our REML method than with Ke and Wang's one. Let us remind that for SAEM-REML these estimates are the expectation of the conditional distribution of fixed effects given the observed data and the REML estimates of the variance-covariance parameters.

An important issue to discuss is the convergence of estimates with this kind of iterative maximization algorithms. It is well known that approximate methods for maximum likelihood estimation often present numerical problems and even fail to converge in the framework of NLME estimation (see (Hartford and Davidian, 2000) for instance). An advantage of the exact likelihood method is exactly to avoid those convergence problems as it was established by Kuhn and Lavielle (2005). In this simulation study, we have to say that both *semi-SAEM* and *snm* achieved convergence for all the data sets. However, we also tried to fit a nonlinear mixed effects model to the simulated data, that is, assuming that f was known and estimating only the fixed and random effects with *SAEM* and *nlme*, and in that case the second algorithm failed to converge for several data sets. It seems that in this case the combination of *nlme* with a nonparametric algorithm to perform semiparametric estimation solves the numerical problems encountered by *nlme* on its own. However, this is not true in general as we will see in the next simulation study.

Table 1 ML procedure: Mean, MSE and 95% confidence interval of mean components.

	Method	μ_1	μ_2	μ_3
True Value		1	0	0
Mean	semi-SAEM	1.06	0.31	0.27
	snm	1.05	0.26	-0.01
MSE	semi-SAEM	0.12	0.16	0.10
	snm	0.12	0.11	0.01
95 % C.I.	semi-SAEM	[0.99;1.12]	[0.27;0.36]	[0.23;0.30]
	snm	[0.99;1.12]	[0.22;0.30]	[-0.02;0.01]

Table 2 ML procedure: Mean, MSE and 95% confidence interval of variance components obtained with *semi-SAEM* and *snm*.

	Method	γ_1	γ_2	γ_3	σ^2
True Value		1	0.25	0.16	1
Mean	semi-SAEM	0.86	0.24	0.16	0.95
	snm	0.89	0.19	0.14	0.99
MSE	semi-SAEM	0.22	0.02	0.01	0.03
	snm	0.22	0.02	0.01	0.03
95 % C.I.	semi-SAEM	[0.77;0.95]	[0.21;0.27]	[0.14;0.17]	[0.92;0.98]
	snm	[0.80;0.98]	[0.17;0.21]	[0.13;0.16]	[0.96;1.02]

Table 3 REML procedure: Mean, MSE and 95% confidence interval of mean components.

	Method	μ_1	μ_2	μ_3
True Value		1	0	0
Mean	semi-SAEM	1.04	-0.01	-0.01
	snm	1.05	0.26	-0.01
MSE	semi-SAEM	0.03	0.02	0.01
	snm	0.12	0.11	0.01
95 % C.I.	semi-SAEM	[1.01;1.07]	[-0.03;0.02]	[-0.02;0.01]
	snm	[0.99;1.12]	[0.22;0.30]	[-0.02;0.01]

5.2 Simulation study: semiparametric estimation

In order to test our LASSO-based estimator we consider the same general model of the previous section

$$y_{ij} = \phi_{1i} + \exp(\phi_{2i}) 2 f\left(\frac{j}{N} - \frac{\exp(\phi_{3i})}{1+\exp(\phi_{3i})}\right) + \varepsilon_{ij},\ i = 1\ldots,N,\ j = 1,\ldots,J,$$

where $\varepsilon_{ij} \sim \mathcal{N}(0,\sigma^2)$ and $\phi_i = (\phi_{1i},\phi_{2i},\phi_{3i})' \sim \mathcal{N}(\mu,\Gamma)$ with $\mu = (\mu_1,\mu_2,\mu_3)'$. Now, $f(\cdot)$ is supposed to be unknown and must be estimated. It is generated as a mixture of one trigonometric function and two Laplace densities (see Figure 1).

$$f(t) = 0.6 \times \sin(2\pi t) + 0.2 \times \left(\frac{e^{-40|t-0.75|}}{2 \times \int_0^1 e^{-40|t-0.75|}}\right) + 0.2 \times \left(\frac{e^{-40|t-0.8|}}{2 \times \int_0^1 e^{-40|t-0.80|}}\right).$$

Data were simulated using the following parameters: $N = 10$, $J = 20$, $\mu = (1,0,0)'$, $\sigma^2 = 0.4$ and Γ is diagonal with $diag(\Gamma) = (0.25, 0.16, 0.04)$.

The chosen function exhibits two sharp peaks that can not be clearly distinguished by only looking at the resulting data (Figure 2). We propose this setting in order to compare the performance of our method and *snm* in a situation in which the underlying function is not smooth. Indeed, the definition of Ke and Wang's method guarantees that it will achieve very good results if the function to be estimated is well approximated by combinations of spline functions. However, there might be practical situations in which assessing the smoothness of the underlying function might not be easy. It is then interesting to investigate the performance of both methods in such cases.

Data were analyzed using the two following semiparametric procedures: our SAEM and LASSO based method (called *LASSO-SAEM*) and Ke and Wang's procedure for semiparametric models, still denoted *snm*. For both methods we obtained the REML estimates of parameters.

It is necessary to specify several values in order to run our algorithm, such as the choice of the LASSO's tuning parameter τ and the inputs of the SAEM algorithm (initial values, step sizes χ_k, number of chains in the MCMC step, number of burn-in iterations, and total number of iterations). For the latter, we used again 80 iterations with $\chi_k = 1$ for $1 \leq k \leq 50$ and $\chi_k = 1/(k-50)$ for $51 \leq k \leq 80$, and we considered $m = 5$ chains in each iteration. The initial values, which were also used with *snm*, were: $\mu_0 = (1,0,0)$, $\sigma_0^2 = 2$ and $\text{diag}(\Gamma_0) = (\gamma_1^0,\gamma_2^0,\gamma_3^0) = (1, 0.3, 0.1)$.

The nonparametric LASSO step has been performed with $\tau = 1/3$. For some datasets, larger values of τ did not lead to the stabilization of the convergence of some parameters, in

Table 4 REML procedure: Mean, MSE and 95% confidence interval of variance components obtained with *semi-SAEM* and *snm*.

	Method	γ_1	γ_2	γ_3	σ^2
True Value		1	0.25	0.16	1
Mean	semi-SAEM	0.99	0.25	0.16	0.95
	snm	0.92	0.19	0.15	1.02
MSE	semi-SAEM	0.21	0.03	0.01	0.03
	snm	0.23	0.02	0.01	0.03
95 % C.I.	semi-SAEM	[0.89;1.08]	[0.22;0.28]	[0.14;0.18]	[0.92;0.98]
	snm	[0.83;1.02]	[0.17;0.22]	[0.13;0.17]	[0.98;1.05]

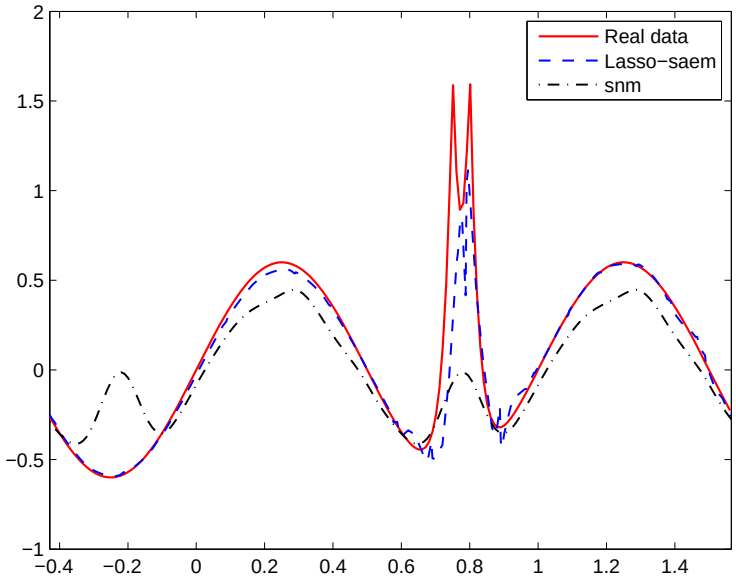

Fig. 1 True function f (solid line) and its estimates obtained with *LASSO-SAEM* (dashed line) and *snm* (dash-dotted line) for a particular data set in the semiparametric simulation study.

particular the variance γ_2, and smaller values of τ provided similar results to the one presented here. The dictionary chosen combined very different orthonormal families, namely Fourier functions with Haar wavelets, which ensured a sufficiently incoherent design in the spirit of Section 3. More precisely, our dictionary was composed by the following Fourier functions $\{t\mapsto 1, t\mapsto\cos(\pi t), t\mapsto\sin(\pi t), t\mapsto\cos(2\pi jt), t\mapsto\sin(2\pi jt), j=1,\cdots,5\}$ and by the Haar wavelet basis with resolution between 2^4 and 2^7, with a total size of 245 functions. Note that the data $\tilde{x}_{ij}=c(\phi_i;x_{ij})$ belongs approximately to $[-0.4,1.6]$. For *snm*, we took $f\in W_2^0(per)$. Of course, the true function does not belong to that space and a partial spline model with possible change points would be more appropriate for modeling it. However, we want to reflect the fact that in a real situation the only information available is the one provided by the observed data set. In this case the simulated data exhibit a clear periodic structure which we try to capture with a function in $W_2^0(per)$. In Figures 1 and 2, we can see the estimates of f compared with the true function and the fitted data with the two methods for a specific simulated data set. Results for REML estimates obtained with *LASSO-SAEM* and *snm* for 100 simulated data sets are summarized in Tables 5 and 6. We can see that the means of the estimates obtained with our method are close to their real values except for the variance of the error, σ^2, since our method tends to overestimate that parameter. However, we get overall better results than using the *snm* methodology (except for γ_1).

An important issue for this kind of problem is the estimation of the nonlinear function f. Ke and Wang's method based on splines works very well for regular functions. So, it is interesting to study its performance on less smooth functions, which is typically the case with the function f considered here. Then, to evaluate the accuracy of the estimation, we calculated the Integrated Square Error (ISE) of $\hat{f}$ for each simulated data set. Figure 3 provides a summary of estimates of f using *LASSO-SAEM* and *snm*. We computed the ISE for each estimate of f and plotted the estimates corresponding to (a) the minimum, (b) 1/4 quantile, (c) median, (d) 3/4 quantile and (e) maximum ISEs. We can see that our

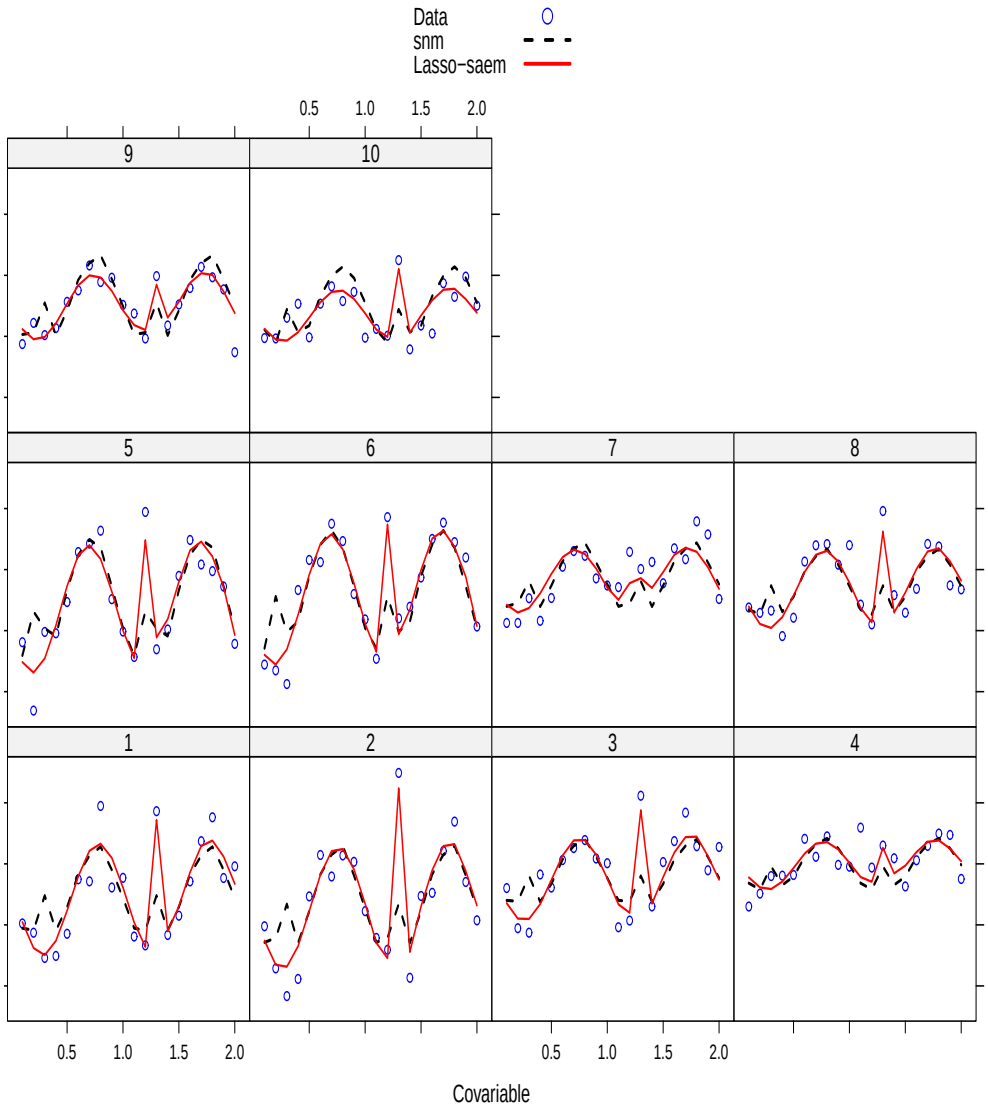

Fig. 2 Simulated data and fitted curves obtained with *LASSO-SAEM* (solid line) and *snm* (dashed line) for a particular data set in the semiparametric simulation study.

Table 5 REML procedure: Mean, MSE and 95% confidence interval of means components obtained with *LASSO-SAEM* and *snm*.

	Method	μ_1	μ_2	μ_3
True Value		1	0	0
Mean	LASSO-SAEM	0.97	0.02	0.01
	snm	1.09	1.39	-0.01
MSE	LASSO-SAEM	0.009	0.009	0.003
	snm	0.019	2.035	0.005
95 % C.I.	LASSO-SAEM	[0.949;0.984]	[0.005;0.041]	[-0.006;0.014]
	snm	[1.057;1.119]	[1.293;1.482]	[-0.025;0.015]

Table 6 REML procedure: Mean, MSE and 95% confidence interval of variance components obtained with *LASSO-SAEM* and *snm*.

	Method	γ_1	γ_2	γ_3	σ^2
True Value		0.25	0.16	0.04	0.4
Mean	LASSO-SAEM	0.18	0.14	0.03	0.69
	snm	0.21	0.11	0.03	0.90
MSE	LASSO-SAEM	0.01	0.01	4.0e-4	0.12
	snm	0.02	0.01	5.9e-4	0.27
95 % C.I.	LASSO-SAEM	[0.16;0.20]	[0.12;0.15]	[0.030;0.037]	[0.66;0.73]
	snm	[0.18;0.25]	[0.09;0.14]	[0.028;0.042]	[0.86;0.94]

method outperforms *snm* in the estimation of f, in the sense that our estimates are able to detect the presence of the peaks in the original function.
As for the functions of the dictionary selected with our LASSO method, it is interesting to note that the 100 linear combinations of functions of the dictionary obtained for each one of the 100 data sets have a length which varies between 10 and 32 functions, with an average length equal to 20. Furthermore, in 98% of the cases, the method selects the function $\sin(2\pi t)$ with the highest coefficient. For the remaining two data sets, the functions $\sin(6\pi t)$ and $\sin(10\pi t)$ are selected. For all the replicates, in addition to these sine functions, the rest of the selected functions are related to the Haar wavelets with smaller coefficients. So, our method is quite robust.

It is important to point out that the results obtained with *snm* are based only on 51 data sets since the function did not reach convergence in 46 data sets and in other 3 data sets we obtained incoherent estimation of the nonlinear function, when using the default setup of the *snm* algorithm (REML estimation and Generalized Cross Validation for the choice of the penalized parameter). By contrast, our method achieved convergence for all simulated data sets with the specific setup used here (choice of τ, initial values, number of chains, step sizes χ_k, number of iterations, etc ...).

To assess the robustness of the LASSO procedure, we have also performed an analysis of these data sets with a dictionary that is composed by the union of the dictionary defined above (the 245 functions) and the dictionary used in Section 5.3 (the 64 functions). The results obtained are very similar to those presented in Tables 5 and 6, so we have not included them here. Moreover, the estimates of f are also very similar. In particular, for 50% of the data sets, the estimates of f select only components in the old dictionary (with Fourier and wavelet functions) and for all the datasets, only 7% of the selected functions belongs to the dictionary defined in Section 5.3. Additionally, the function $\sin(2\pi t)$ is selected with the highest coefficient in 90% of the cases.

Finally, we compute the confidence intervals defined in (14) with $L_0 = 20$. We obtained very thin confidence intervals and a poor coverage (less to 40%) with these datasets. We think that it is a challenging issue to study if an appropriate choice of L_0 in (14) may yield to more robust intervals.

5.3 Application to on-line auction data

Modeling of price paths in on-line auction data has received a lot of attention in the last years (Shmueli and Jank, 2005; Jank and Shmueli, 2006; Shmueli et al, 2007; Liu and Müller, 2008). One of the reasons is the availability of huge amounts of data made public by the on-line auction and shopping website eBay.com, which has become a global market place in which millions of people worldwide buy and sell products. The price evolution during an auction can be thought as a continuous process which is observed discretely and sparsely only at the instants in which bids are placed. In fact, bids tend to concentrate at the beginning and at the end of the auction, responding to two typically observed phenomena, "early bidding" and "bid sniping" (a situation in which "snipers" place their bids at the very last moment).
To our knowledge, Reithinger et al (2008) provide the first attempt to model price paths taking into account the dependence among different auctions. This is an important consideration, since in practice bidders can participate in different auctions that take place simultaneously. They propose a semiparametric additive mixed model with a boosting estimation approach. In the same line, but considering a more complex interaction of the random effects and the unknown nonlinear function, we propose the following shape-invariant model for the price paths:

$$y_{ij} = \phi_{1i} + \exp(\phi_{2i}) f(t_{ij} - \phi_{3i}) + \varepsilon_{ij}, \qquad \begin{array}{l} i = 1, \cdots, N, \\ j = 1, \cdots, n_i, \end{array}$$

where $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$ and $\phi_i = (\phi_{1i}, \phi_{2i}, \phi_{3i})' \sim \mathcal{N}(\mu, \Gamma)$ with $\mu = (\mu_1, \mu_2, \mu_3)'$. We introduce an individual random horizontal shift, ϕ_{3i}, to model the possible delay of the price dynamics in some auctions with respect to the rest.
We analyzed a set of 183 eBay auctions for Palm M515 Personal Digital Assistants (PDA), of a fixed duration of seven days, that took place between March and May, 2003. This is the data set used in Reithinger et al (2008) and it is publicly available at `http://www.rhsmith.umd.edu/digits/statistics/data.aspx`. We were interested in modeling the live bids, that is, the actual prices that are shown by eBay during the live auction. Note that these are different from the bids placed by bidders during the auction, which are the prices recorded in the bid history published by eBay after the auction closes. Then, a transformation on the bid

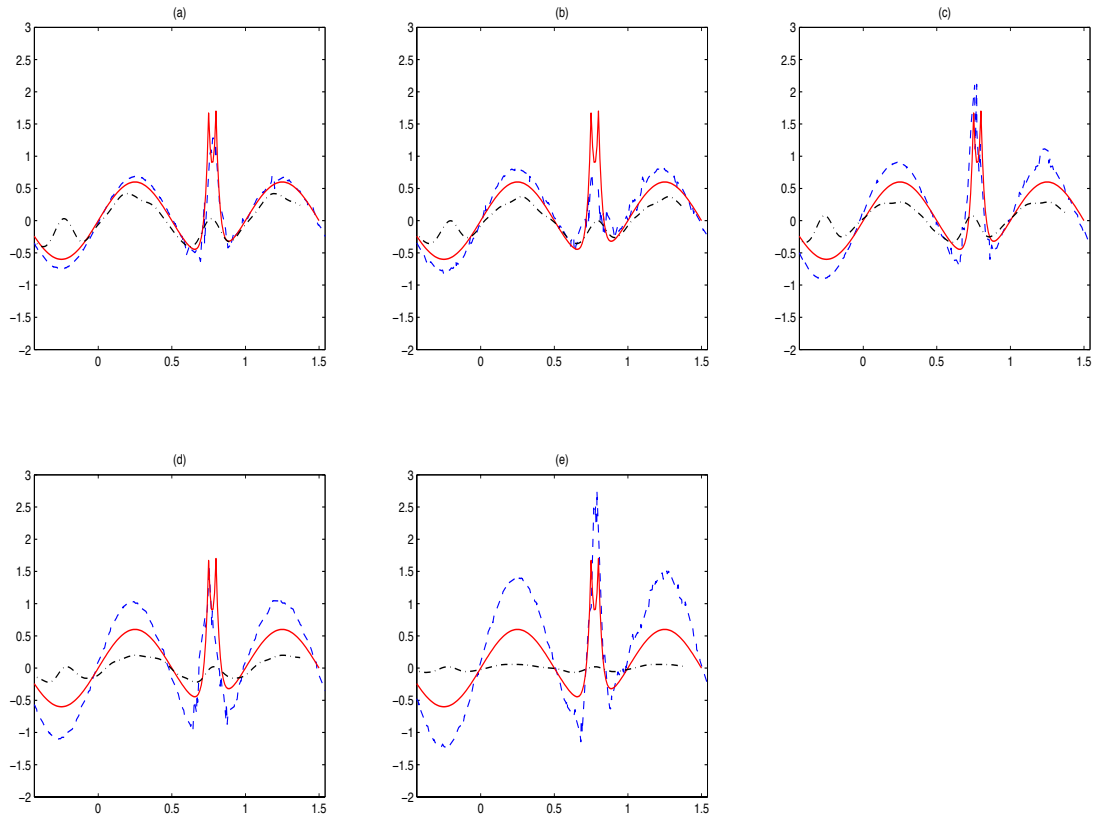

Fig. 3 Estimated functions corresponding to the five quantiles of ISE ((a) minimum, (b) 1/4 quantile, (c) median, (d) 3/4 quantile and (e) maximum) obtained with *LASSO-SAEM* (dashed line) and *snm* (dash-dotted line) compared to the true function f (solid line) for the total of the 100 simulated data sets in the semiparametric simulation study.

records is required to recover the live bids (see Shmueli and Jank (2005) for details).

The live bids range from \$0.01 to \$300 and form a sequence of non decreasing prices for each auction. We typically observe between 10 and 30 bids per auction, although there are auctions with only two bids. We have a total of 3280 bids for the 183 auctions. Following Reithinger et al (2008), we considered the square root of live bids to reduce the price variability. We run the REML version of our *LASSO-SAEM* algorithm, of which we performed 100 iterations with the following sequence of decreasing steps $(\chi_k)_k$: $\chi_k = 1$ for $1 \leq k \leq 60$ and $\chi_k = 1/(k-60)$ for $61 \leq k \leq 100$. We also considered $m = 3$ chains in each iteration. The dictionary for nonparametric estimation was composed by a combination of B-splines of degrees three and four, with 17 knots unequally spaced so that most of the knots were in those places with more data observed (at the beginning, at the end and at the middle of the interval), 10 power functions, 10 exponential functions and 5 logit functions, with a total size of 64 functions. The estimate of f is monotone, as expected by the nature of the data, and presents two steepest parts at the beginning and at the end of the interval. At each iteration of the algorithm the estimated function at the nonparametric step is a sparse combination of the functions of the dictionary. In fact, the set of functions selected by the LASSO method at the last iterations of the algorithm is almost constant, containing mainly two functions, $\varphi(x) = x^{0.35}$ and $\varphi(x) = \exp(0.9x)$, and in some iterations a small component of a cubic B-spline around the middle of the interval. In Figure 4 we present the last 24 estimates $f^{(k,l)}$ from which we have obtained $\hat{f}$ as in (13), and $\hat{f}$, together with a 95% pointwise confidence band. These results have been obtained with $\tau = 2$ as the value for the tuning parameter in the LASSO estimation step. The estimates for μ and Γ are presented in Table 7.

To assess the robustness of the LASSO procedure, we have also performed an analysis of this data set with a dictionary that is composed by the union of the dictionary defined above and the dictionary used in Section 5.2 to analyze the simulated data. That is, we have added Fourier and Haar

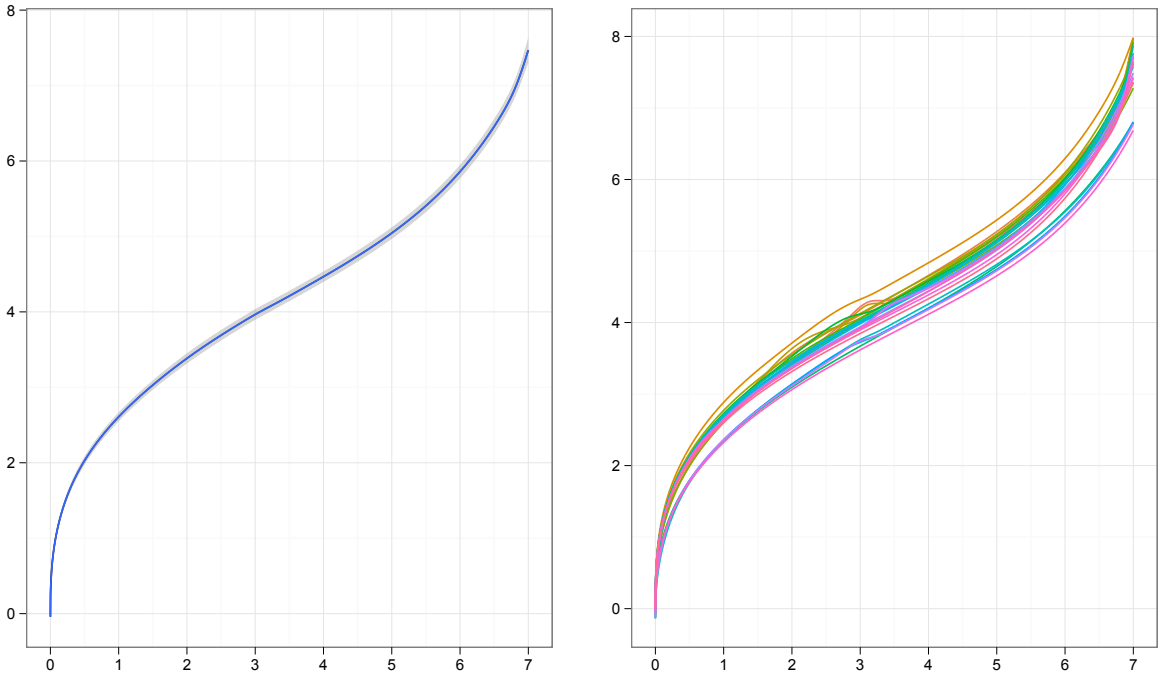

Fig. 4 Left: Estimated nonlinear function $\hat{f}$ (solid line) and 95% confidence band (gray shadow) in the on-line auction data set. Right: Last 24 LASSO estimates whose empirical mean provides $\hat{f}$.

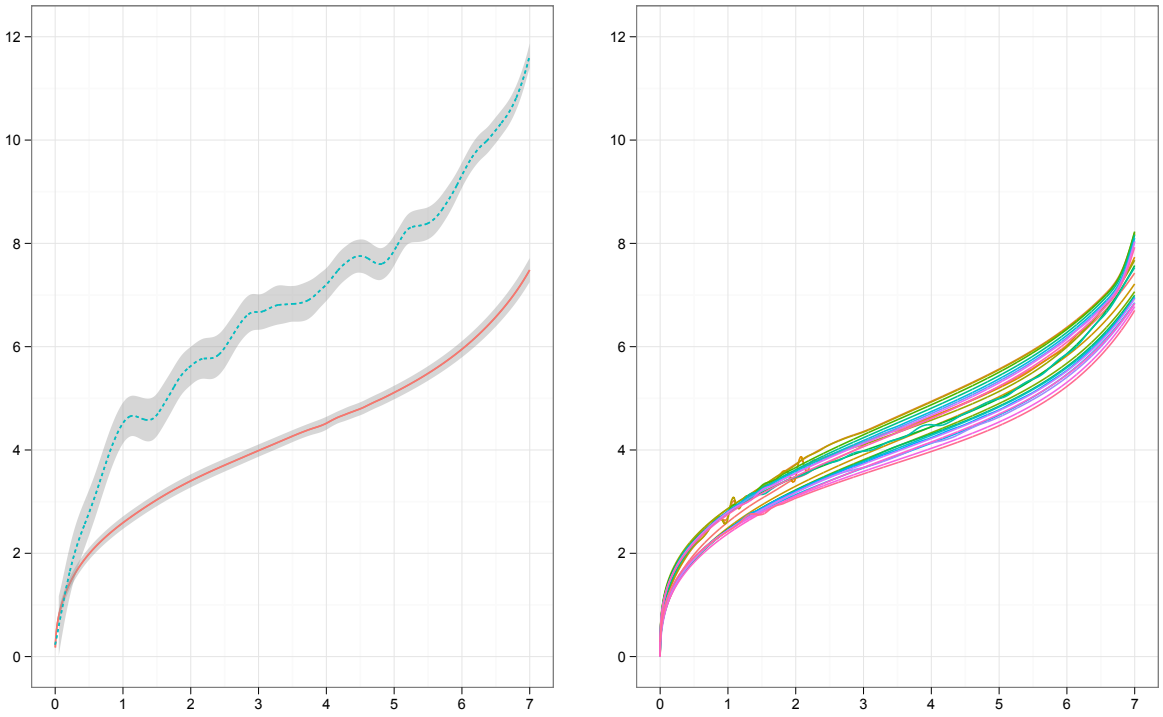

Fig. 5 Left: Estimates of f obtained with *snm* (dashed line) and *saem-lasso* with the large dictionary (solid lines) and 95% confidence bands (gray shadows). Right: Last 24 LASSO estimates in *saem-lasso*.

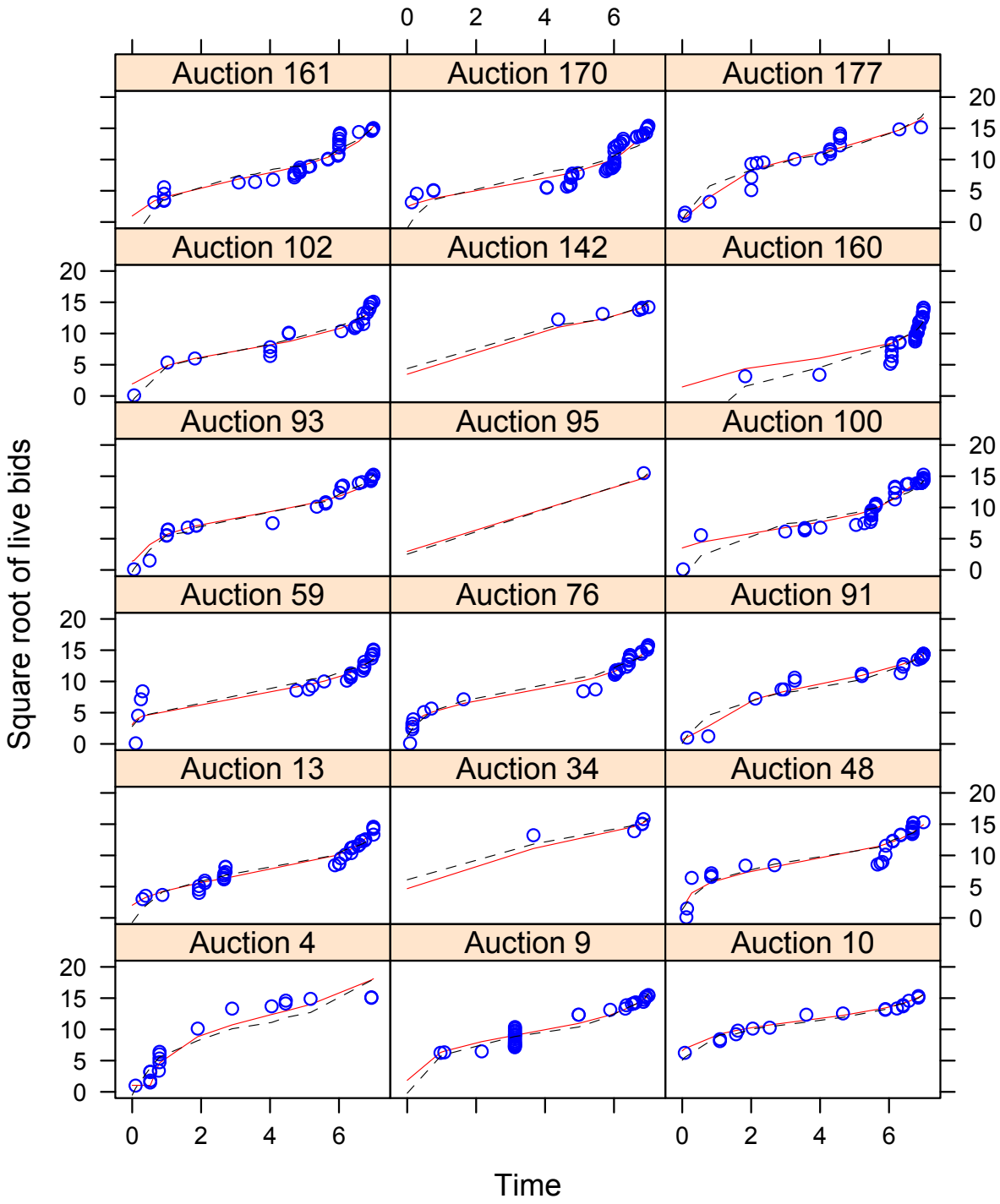

Fig. 6 Observed live bids (circles) and fitted price curves for a subset of 18 auctions obtained with *snm* (dahsed lines) and *saem-lasso* (solid lines) with the large dictionary.

Table 7 Estimated mean vector and covariance matrix of the random effects and estimated error variance in the on-line auction data set.

	ϕ_1	ϕ_2	ϕ_3	
Mean	1.04	0.18	-0.06	
Correlation	1 (7.68)	-0.02	0.41	ϕ_1
Matrix	-0.02	1 (0.19)	0.37	ϕ_2
(variances)	0.41	0.37	1 (0.23)	ϕ_3
σ^2		1.93		

wavelets bases to the dictionary initially chosen. The results are very similar to those obtained with the original dictionary. They are shown in Figure 5. In particular, the estimates of f are almost identical. Among the last 24 estimates of f, $f^{(k,l)}$, obtained with this new dictionary, only two estimates contain a significant component of functions not included in the original dictionary.

To compare our method to Ke and Wang's, in Figures 5 and 6 we also present the results of the analysis of this data set with *snm*. We have to mention that we have performed this analysis with five different function models for f and two different criteria for the estimation of the smoothing parameter, namely, general cross validation (GCV) and generalized maximum likelihood (GML). So, we ran *snm* with ten different specifications, among which we got convergence for only six specifications. None of the six estimates of f is strictly monotone and five of them are extremely rough. In Figure 5 we present the smoothest *snm* f-estimate, which is obtained by modeling f with cubic splines and by using the GLM criterion, together with the *saem-LASSO* estimate obtained with the largest dictionary. In Figure 6 we present the observed live bids and the model fits for 18 chosen auctions with different price profiles. We can appreciate how the fitted models provide in general an accurate fit of the final price, even in the cases when bid sniping is present. There are some differences between the two fits, mostly at the beginning of each auction, although the fitted curves are in general similar with the two methods. For the rest of the combinations of a function model and a smoothing estimation criterion used with *snm*, the fits of the data are suboptimal. Indeed, the fitted price curves produce almost perfect interpolation of the data.

As for the computation time, *saem-lasso* took 300 seconds to run on these data on a 2.5 GHz Mac OS X whereas the average time for *snm* over the six runs was about six hours on the same computer.

6 Conclusions and discussion

Semiparametric nonlinear mixed effects models cover a wide range of situations and generalize a large class of models, such as nonlinear mixed effects models or self-modelling nonlinear regression models among others. We have proposed a new method for estimation in SNMMs combining an exact likelihood estimation algorithm with a LASSO-type procedure. Our strategy relies on an iterative procedure to estimate θ conditioned on f and vice versa, which allow us to tackle the parametric and the nonparametric problem independently. This makes possible the use of fast algorithms providing an accurate and computationally efficient estimation method.

Concerning parametric estimation, our simulation results illustrate our method and point out some important advantages of using an exact likelihood estimation algorithm instead of likelihood approximation methods, such as convergence of the estimates. The REML version of our algorithm, corrects the estimation of variance components accounting for the loss of degrees of freedom from estimating the fixed effects and provide satisfactory results. However, as it was already pointed out in the comments to Ke and Wang (2001), it will be important to define a REML estimator that can also take into account the loss of degrees of freedom from estimating the nonlinear function. As for computational aspects, we have to mention that the SAEM algorithm avoids the convergence problems encountered by `nlme` based routines.

For nonparametric aspects, the dictionary approach based on LASSO algorithms shows, in some situations, some improvements when compared with Ke and Wangs' methodology. This is the case for instance for spiky or non-continuous functions to be estimated. Our dictionary method can adapt to different features of signals for wealthy enough dictionaries. Furthermore, our methodology allows us to obtain interesting interpretation with respect to the functions of the dictionary selected by the procedure. For instance, we can detect trends, frequencies of sinusoids or location and heights of peaks of the common shape represented by the estimated function f. We have observed that our LASSO estimate achieves good theoretical and numerical results if the dictionary is wealthy and incoherent enough. From the theoretical point of view, incoherence is expressed, in this paper, by Assumption A1(s) or by the quantity $\rho(S^*)$ defined in the Supplementary Material. These incoherence assumptions are hard to check in practice and we do not know if they can be relaxed in our setting.

We mention that our method can be non robust if the dictionary is not wealthy enough. That is, if the function to be estimated cannot be well approximated by linear combinations of the functions of the dictionary, the functions that are selected can vary from one simulation to another, which may lead to different estimates. However, it the main features of a signal (periodicity, smoothness, peaks,...) are included in the dictionary, our method is very robust to the enlarging of the dictionary with additional functions, as seen in Sections 5.2 and 5.3.

In Section 3, the particular structure of the observations (where we have n_i observations for each individual i) is not

used for applying the standard LASSO-procedure. But a natural and possible extension of this work would be to take into account this structure and then to apply a more sophisticated LASSO-type procedure inspired, for instance, by the group-LASSO proposed by Yuan and Lin (2006) to achieve better results. This is a challenging research axis we wish to investigate from a theoretical and practical point of view. The LASSO is a very popular algorithm, but Hybrid Adaptive Spline, MARS or BSML (see Sklar et al (2012)) could also be combined with the dictionary approach proposed in this paper. Since results of our paper show that the dictionary approach seems promising, results of our paper could be extended by using algorithms mentioned previously from both theoretical and practical points of view.

Among other possible extensions of this work, a very promising one would be the use of the nonparametric techniques herein described for density estimation (in the spirit of (Bertin et al, 2011)) of the random errors, assuming that they do not need to be normal. Indeed, the recent work of Comte and Samson (2012) deals with this problem in the case of a linear mixed effects model. Its generalization to NLMEs or even SNMMs is a real challenge.

Acknowledgements

The authors would like to thank the anonymous Associate Editor and two referees for valuable comments and suggestions.

Suppementary Material

In the supplement available on-line we provide theoretical results and proofs for the LASSO-type estimator of Section 3.

References

Bertin K, Le Pennec E, Rivoirard V (2011) Adaptive Dantzig density estimation. Annales de l'Institut Henri Poincaré 47:43–74

Bickel PJ, Ritov Y, Tsybakov AB (2009) Simultaneous analysis of lasso and Dantzig selector. Ann Statist 37(4):1705–1732

Bühlmann P, van de Geer S (2011) Statistics for high-dimensional data. Springer Series in Statistics, Springer, Heidelberg, methods, theory and applications

Bunea F (2008) Consistent selection via the Lasso for high dimensional approximating regression models. In: Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh, Inst. Math. Stat. Collect., vol 3, Inst. Math. Statist., Beachwood, OH, pp 122–137

Bunea F, Tsybakov AB, Wegkamp MH (2006) Aggregation and sparsity via l_1 penalized least squares. In: Learning theory, Lecture Notes in Comput. Sci., vol 4005, Springer, Berlin, pp 379–391

Bunea F, Tsybakov A, Wegkamp M (2007a) Sparsity oracle inequalities for the Lasso. Electronic Journal of Statistics 1:169–194

Bunea F, Tsybakov AB, Wegkamp MH (2007b) Aggregation for Gaussian regression. The Annals of Statistics 35(4):1674–1697

Comte F, Samson A (2012) Nonparametric estimation of random effects densities in linear mixed-effects model, DOI oai:hal.archives-ouvertes.fr:hal-00657052, URL http://hal.archives-ouvertes.fr/hal-00657052/fr/, unpublished manuscript. Available at http://hal.archives-ouvertes.fr/hal-00657052/fr/

Delyon B, Lavielle M, Moulines E (1999) Convergence of a stochastic approximation version of the em algorithm. The Annals of Statistics 27:94–128

Dempster AP, Laird NM, Rubin DB (1977) Maximum-likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society, Series B 39:1–38

Ding AA, Wu H (2001) Assessing antiviral potency of anti-HIV therapies in vivo by comparing viral decay rates in viral dynamic models. Biostatistics 2:13–29

Foulley JL, Quaas R (1995) Heterogeneous variances in gaussian linear mixed models. Genetics Selection Evolution 27:211–228

Ge Z, Bickel P, Rice J (2004) An approximate likelihood approach to nonlinear mixed effects models via spline approximation. Computational Statistics and Data Analysis 46:747–776

van de Geer S (2010) ℓ_1-regularization in high-dimensional statistical models. In: Proceedings of the International Congress of Mathematicians. Volume IV, Hindustan Book Agency, New Delhi, pp 2351–2369

Hartford A, Davidian M (2000) Consequences of misspecifying assumptions in nonlinear mixed effects models. Computational Statistics & Data Analysis 34:139–164

Harville D (1974) Bayesian inference for variance components using only error contrasts. Biometrika 61:383–385

Jank W (2006) Implementing and diagnosing the stochastic approximation em algorithm. Journal of Computational & Graphical Statistics 15(4):803–829

Jank W, Shmueli G (2006) Functional data analysis in electronic commerce research. Statistical Science 21:155–166

Ke C, Wang Y (2001) Semiparametric nonlinear mixed-effects models and their applications (with discussion). Journal of the American Statistical Association 96(456):1272–1298

Kuhn E, Lavielle M (2004) Coupling a stochastic approximation version of EM with an MCMC procedure. ESAIM: P&S 8:115–131

Kuhn E, Lavielle M (2005) Maximum likelihood estimation in nonlinear mixed effects models. Computational Statistics & Data Analysis 49(4):1020–1038

Liu B, Müller HG (2008) Functional data analysis for sparse auction data. In: Jank W, Shmueli G (eds) Statistical Methods in E-commerce research, Wiley, New York, pp 269–290

Liu W, Wu L (2007) Simultaneous inference for semiparametric nonlinear mixed-effects models with covariate measurement errors and missing responses. Biometrics 63:342–350

Liu W, Wu L (2008) A semiparametric nonlinear mixed-effects model with non-ignorable missing data and measurement errors for HIV viral data. Computational Statistics & Data Analysis 53:112–122

Liu W, Wu L (2009) Some asymptotic results for semiparametric nonlinear mixed-effects models with incomplete data. Journal of Statistical Planning and Inference Doi:10.1016j.jspi.2009.06.006

Luan Y, Li H (2004) Model-based methods for identifying periodically expressed genes based on time course microarray gene expression data. Bioinformatics 20(3):332–339

Meza C, Jaffrézic F, Foulley JL (2007) Estimation in the probit normal model for binary outcomes using the saem algorithm. Biometrical Journal 49(6):876–888

Meza C, Jaffrézic F, Foulley JL (2009) Reml estimation of variance parameters in nonlinear mixed effects models using the SAEM algorithm. Computational Statistics & Data Analysis 53(4):1350–1360

Patterson HD, Thompson R (1971) Recovery of inter-block information when block sizes are unequal. Biometrika 58:545–554

Pinheiro J, Bates D (2000) Mixed-Effects Models in S and S-PLUS. Springer-Verlag, New York

Ramos R, Pantula S (1995) Estimation of nonlinear random coefficient models. Statistics & Probability Letters 24:49–56

Reithinger F, Jank W, Tutz G, Shmueli G (2008) Modelling price paths in on-line auctions: smoothing sparse and unevenly sampled curves by using semiparametric mixed models. Applied Statistics 57:127–148

Schelldorfer J, Bühlmann P, van de Geer S (2011) Estimation for high-dimensional linear mixed-effects models using l1-penalization. Scandinavian Journal of Statistics 38:197–214

Shmueli G, Jank W (2005) Visualizing online auctions. Journal of Computational and Graphical Statistics 14:299–319

Shmueli G, Russo RP, Jank W (2007) The BARISTA: a model for bid arrivals in online auctions. The Annals of Applied Statistics 1:412–441

Sklar JC, Wu J, Meiring W, Wang Y (2012) Non-parametric regression with basis selection from multiple libraries. Technometrics, accepted

Tibshirani R (1996) Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society, Series B 58:267–288

Vonesh EF (1996) A note on the use of Laplace's approximation for nonlinear mixed-effects models. Biometrika 83:447–452

Wang Y, Brown MB (1996) A flexible model for human circadian rhythms. Biometrics 52:588–596

Wang Y, Ke C (2004) Assist: A suite of s functions implementing spline smoothing techniques. http://wwwpstatucsbedu/faculty/yuedong/assistpdf

Wang Y, Ke C, Brown MB (2003) Shape-invariant modeling of circadian rhythms with random effects and smoothing spline anova decompositions. Biometrics 59:804–812

Wang Y, Eskridge K, Zhang S (2008) Semiparametric mixed-effects analysis of PKPD models using differential equations. Journal of Pharmacokinetics and Pharmacodynamics 35:443–463

Wei GC, Tanner MA (1990) A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. Journal of the American Statistical Association 85:699–704

Wu H, Zhang J (2002) The study of longterm HIV dynamics using semi-parametric non-linear mixed-effects models. Statistics in Medicine 21:3655–3675

Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B 68(1):49–67

Statistics and Computing manuscript No.
(will be inserted by the editor)

Supplementary Material for "LASSO-type estimators for Semiparametric Nonlinear Mixed-Effects Models Estimation"

Ana Arribas-Gil · Karine Bertin · Cristian Meza · Vincent Rivoirard

the date of receipt and acceptance should be inserted later

1 Theoretical results for the LASSO-type estimator

1.1 Assumptions

As usual, assumptions on the dictionary are necessary to obtain oracle results for LASSO-type procedures. We refer the reader to van de Geer and Bühlmann (2009) for a good review of different assumptions considered in the literature for LASSO-type estimators and connections between them. The dictionary approach aims at extending results for orthonormal bases. Actually, our assumptions express the relaxation of the orthonormality property. To describe them, we introduce the following notation. For $l \in \mathbb{N}$, we denote

$$\nu_{\min}(l) = \min_{|J|\leq l} \min_{\substack{\lambda\in\mathbb{R}^M \\ \lambda_J\neq 0}} \frac{\|f_{\lambda_J}\|_n^2}{\|\lambda_J\|_{\ell_2}^2} \qquad \text{and} \qquad \nu_{\max}(l) = \max_{|J|\leq l} \max_{\substack{\lambda\in\mathbb{R}^M \\ \lambda_J\neq 0}} \frac{\|f_{\lambda_J}\|_n^2}{\|\lambda_J\|_{\ell_2}^2},$$

where $\|\cdot\|_{\ell_2}$ is the l_2 norm in $\mathbb{R}^M$. The notation λ_J means that for any $k \in \{1,\dots,M\}$, $(\lambda_J)_k = \lambda_k$ if $k \in J$ and $(\lambda_J)_k = 0$ otherwise. Previous quantities correspond to the "restricted" eigenvalues of the Gram matrix $G = (G_{j,j'})$ with coefficients

$$G_{j,j'} = \frac{1}{n}\sum_{i=1}^{n} b_i^2 \varphi_j(x_i)\varphi_{j'}(x_i).$$

Assuming that $\nu_{\min}(l)$ and $\nu_{\max}(l)$ are close to 1 means that every set of columns of G with cardinality less than l behaves like an orthonormal system. We also consider the restricted

Ana Arribas-Gil
Departamento de Estadística, Universidad Carlos III de Madrid,
E-mail: ana.arribas@uc3m.es

Karine Bertin · Cristian Meza
CIMFAV-Facultad de Ingeniería, Universidad de Valparaíso,
E-mail: karine.bertin@uv.cl, E-mail: cristian.meza@uv.cl

Vincent Rivoirard
CEREMADE, CNRS-UMR 7534, Université Paris Dauphine,
E-mail: vincent.rivoirard@dauphine.fr

correlations

$$\delta_{l,l'} = \max_{\substack{|J|\le l \\ |J'|\le l' \\ J\cap J'=\emptyset}} \max_{\lambda,\lambda'\in\mathbb{R}^M \atop \lambda_J\neq 0,\lambda'_{J'}\neq 0} \frac{\langle f_{\lambda_J}, f_{\lambda'_{J'}}\rangle}{\|\lambda_J\|_{\ell_2}\|\lambda'_{J'}\|_{\ell_2}},$$

where $\langle f,g\rangle = \frac{1}{n}\sum_{i=1}^n b_i^2 f(x_i)g(x_i)$. Small values of $\delta_{l,l'}$ means that two disjoint sets of columns of G with cardinality less than l and l' span nearly orthogonal spaces. We will use the following assumption considered in Bickel et al (2009).

Assumption 1 *For some integer $1 \le s \le M/2$, we have*

$$\nu_{\min}(2s) > \delta_{s,2s}. \tag{A1(s)}$$

Oracle inequalities of the Dantzig selector were established under this assumption in the parametric linear model by Candès and Tao (2007) and for density estimation by Bertin et al (2011). It was also considered by Bickel et al (2009) for nonparametric regression and for the LASSO estimate.

Let us denote

$$\kappa_s = \sqrt{\nu_{\min}(2s)}\left(1 - \frac{\delta_{s,2s}}{\nu_{\min}(2s)}\right) > 0, \quad \mu_s = \frac{\delta_{s,2s}}{\sqrt{\nu_{\min}(2s)}}.$$

We will say that $\lambda \in \mathbb{R}^M$ satisfies the Dantzig constraints if for all $j = 1,\dots,M$

$$\left|(G\lambda)_j - \hat\beta_j\right| \le r_{n,j}, \tag{1}$$

where

$$\hat\beta_j = \frac{1}{n}\sum_{i=1}^n b_i \varphi_j(x_i) Y_i.$$

We denote $\mathscr{D}$ the set of λ that satisfies (1). The classical use of Karush-Kuhn-Tucker conditions shows that the LASSO estimator $\hat\lambda \in \mathscr{D}$, so it satisfies the Dantzig constraint. Finally, we assume in the sequel

$$M \le \exp(n^\delta),$$

for $\delta < 1$. Therefore, if $\|\varphi_j\|_n$ is bounded by a constant independent of n and M, then $r_{n,j} = o(1)$ and oracle inequalities established below are meaningful.

1.2 Oracle inequalities

We obtain the following oracle inequalities.

Theorem 1 *Let $\tau > 2$. With probability at least $1 - M^{1-\tau/2}$, for any integer $s < n/2$ such that (A1(s)) holds, we have for any $\alpha > 0$,*

$$\|\hat f - f\|_n^2 \le \inf_{\lambda\in\mathbb{R}^M} \inf_{\substack{J_0\subset\{1,\dots,M\} \\ |J_0|=s}} \left\{ \|f_\lambda - f\|_n^2 + \alpha\left(1 + \frac{2\mu_s}{\kappa_s}\right)^2 \frac{\Lambda(\lambda, J_0^c)^2}{s} + 16 s\left(\frac{1}{\alpha} + \frac{1}{\kappa_s^2}\right) r_n^2 \right\} \tag{2}$$

where

$$r_n = \sup_{j=1,\dots,M} r_{n,j},$$

$$\Lambda(\lambda, J_0^c) = \|\lambda_{J_0^C}\|_{\ell_1} + \frac{\left(\|\hat{\lambda}\|_{\ell_1} - \|\lambda\|_{\ell_1}\right)_+}{2},$$

for any $x \in \mathbb{R}$ $x_+ := \max(x,0)$ *and* $\|\cdot\|_{\ell_1}$ *is the* l_1 *norm in* $\mathbb{R}^M$.

Theorem 2 *Let* $\tau > 2$. *With probability at least* $1 - M^{1-\tau/2}$, *for any integer* $s < n/2$ *such that (A1(s)) holds, we have for any* $\alpha > 0$,

$$\|\hat{f} - f\|_n^2 \leq \inf_{\lambda \in \mathscr{D}} \inf_{\substack{J_0 \subset \{1,\ldots,M\} \\ |J_0| = s}} \left\{ \|f_\lambda - f\|_n^2 + \alpha \left(1 + \frac{2\mu_s}{\kappa_s}\right)^2 \frac{\|\lambda_{J_0^C}\|_{\ell_1} + \|\hat{\lambda}_{J_0^C}\|_{\ell_1}}{s} + 32 s \left(\frac{1}{\alpha} + \frac{1}{\kappa_s^2}\right) r_n^2 \right\}. \tag{3}$$

Similar oracle inequalities were established by Bunea et al (2006), Bunea et al (2007a), Bunea et al (2007b), or van de Geer (2010). But in these works, the functions of the dictionary are assumed to be bounded by a constant independent of M and n. Let us comment the right-hand side of inequalities (2) and (3) of Theorems 1 and 2. The first term is an approximation term which measures the closeness between f and f_λ and that can vanish if f is a linear combination of the functions of the dictionary. The second term can be considered as a bias term. In both theorems, the term $\|\lambda_{J_0^C}\|_{\ell_1}$ corresponds to the cost of having λ with a support different of J_0. For a given λ, this term can be minimized by choosing J_0 as the set of largest coordinates of λ. Note that if the function f has a sparse expansion on the dictionary, that is $f = f_\lambda$ where λ is a vector with s non-zero coordinates, then by choosing J_0 as the set of the s non-zero coordinates, the approximation term and the term $\|\lambda_{J_0^C}\|_{\ell_1}$ vanish. In Theorem 1, the term $\left(\|\hat{\lambda}\|_{\ell_1} - \|\lambda\|_{\ell_1}\right)_+$ will be smaller as the ℓ_1-norm of the LASSO estimator is small and this term is equal to 0 if $\|\hat{\lambda}\|_{\ell_1} \leq \|\lambda\|_{\ell_1}$, which is frequently the case. In Theorem 2, given a vector λ such that f_λ approximates well f, the term $\|\hat{\lambda}_{J_0^C}\|_{\ell_1}$ will be small if the LASSO estimator selects the largest coordinates of λ. The last term can be viewed as a variance term corresponding to the estimation of f as linear combination of s functions of the dictionary (see Bertin et al (2011) for more details). Finally, the parameter α calibrates the weights given for the bias and variance terms.

The following section deals with estimation of sparse functions.

1.3 The support property of the LASSO estimate

Let $\tau > 2$. In this section, we apply the LASSO procedure with $\tilde{r}_{n,j}$ instead of $r_{n,j}$, with

$$\tilde{r}_{n,j} = \sigma \|\varphi_j\|_n \sqrt{\frac{\tilde{\tau} \log M}{n}}, \quad \tilde{\tau} > \tau.$$

We assume that the regression function f can be decomposed on the dictionary: there exists $\lambda^* \in \mathbb{R}^M$ such that

$$f = \sum_{j=1}^{M} \lambda_j^* \varphi_j.$$

We denote S^* the support of λ^*:

$$S^* = \left\{ j \in \{1,\ldots,M\} : \quad \lambda_j^* \neq 0 \right\},$$

and by s^* the cardinal of S^*. We still consider the LASSO estimate $\hat{\lambda}$ and, similarly, we denote $\hat{S}$ the support of $\hat{\lambda}$:

$$\hat{S} = \Big\{ j \in \{1,\dots,M\} : \quad \hat{\lambda}_j \neq 0 \Big\}.$$

One goal of this section is to show that with high probability, we have:

$$\hat{S} \subset S^*.$$

We have the following result.

Theorem 3 *We define*

$$\rho(S^*) = \max_{k \in S^*} \max_{j \neq k} \frac{| < \varphi_j, \varphi_k > |}{\|\varphi_j\|_n \|\varphi_k\|_n}$$

and we assume that there exists $c \in (0,1/3)$ *such that*

$$s^* \rho(S^*) \leq c.$$

If we have

$$\frac{\sqrt{\bar{\tau}} + \sqrt{\tau}}{\sqrt{\bar{\tau}} - \sqrt{\tau}} \leq \frac{1-c}{2c},$$

then

$$\mathbb{P}\big\{ \hat{S} \subset S^* \big\} \geq 1 - 2M^{1-\tau/2}.$$

A similar result was established by Bunea (2008) in a slightly less general model. However, her result is based on strong assumptions on the dictionary, namely each function is bounded by a constant L (see Assumption (A2)(a) in Bunea (2008)). This assumption is mild when considering dictionaries only based on Fourier bases. It is no longer the case when wavelets are considered and Bunea's assumption is satisfied only in the case where L depends on M and n on the one hand and is very large on the other hand. Since L plays a main role in the definition of the tuning parameters of the method, with too rough values for L, the procedure cannot achieve satisfying numerical results for moderate values of n even if asymptotic theoretical results of the procedure are good. In the setting of this paper, where we aim at providing calibrated statistical procedures, we avoid such assumptions.

Finally, we have the following corollary.

Corollary 1 *We suppose that* $A1(s^*)$ *is satisfied and that there exists* $c \in (0,1/3)$ *such that*

$$s^* \rho(S^*) \leq c.$$

If we have

$$\frac{\sqrt{\bar{\tau}} + \sqrt{\tau}}{\sqrt{\bar{\tau}} - \sqrt{\tau}} \leq \frac{1-c}{2c},$$

then, with probability at least $1 - 4M^{1-\tau/2}$,

$$\|\hat{f} - f\|_n^2 \leq \frac{32 s^* \tilde{r}_n^2}{\kappa_{s^*}},$$

where

$$\tilde{r}_n = \sup_{j=1,\dots,M} \tilde{r}_{n,j}.$$

This corollary is a simple consequence of Theorem 2 with $\lambda = \lambda^*$ and $J_0 = S^*$. Taking $\lambda = \lambda^*$ implies that the approximation term vanishes. Taking $J_0 = S^*$ implies that the bias term vanishes since the support of the LASSO estimator is included in the the support of λ^*. In this case, assuming that $\sup_j \|\varphi_j\|_n < \infty$, the rate of convergence is the classical rate $\frac{s^* \log M}{n}$.

2 The proofs

2.1 Preliminary lemma

Lemma 1 *For $1 \leq j \leq M$, we consider the event $\mathscr{A}_j = \left\{|V_j| < r_{n,j}\right\}$ where $V_j = \frac{1}{n}\sum_{i=1}^n b_i \varphi_j(x_i)\varepsilon_i$. Then,*

$$\mathbb{P}(\mathscr{A}_j) \geq 1 - M^{-\tau/2}.$$

Proof of Lemma 1: We have

$$\begin{aligned}
\mathbb{P}\left(\mathscr{A}_j^c\right) &\leq \mathbb{P}\left(\sqrt{n}|V_j|/(\sigma\|\varphi_j\|_n) \geq \sqrt{n} r_{n,j}/(\sigma\|\varphi_j\|_n)\right) \\
&\leq \mathbb{P}\left(|Z| \geq \sqrt{\tau \log M}\right) \\
&\leq M^{-\tau/2}
\end{aligned}$$

where Z is a standard normal variable. □

2.2 Proof of Theorem 1

Let $\lambda \in \mathbb{R}^M$ and J_0 such that $|J_0| = s$. We have

$$\|f_\lambda - f\|_n^2 = \|\hat{f} - f\|_n^2 + \|f_\lambda - \hat{f}\|_n^2 + \frac{2}{n}\sum_{i=1}^n b_i^2 \left(\hat{f}(x_i) - f(x_i)\right)\left(f_\lambda(x_i) - \hat{f}(x_i)\right).$$

We have $\|f_\lambda - \hat{f}\|_n^2 = \|f_\Delta\|_n^2$ where $\Delta = \lambda - \hat{\lambda}$. Moreover

$$A = \frac{2}{n}\sum_{i=1}^n b_i^2 \left(\hat{f}(x_i) - f(x_i)\right)\left(f_\lambda(x_i) - \hat{f}(x_i)\right) = 2\sum_{j=1}^M (\lambda_j - \hat{\lambda}_j)\left[(G\hat{\lambda})_j - \beta_j\right],$$

where

$$\beta_j = \frac{1}{n}\sum_{i=1}^n b_i^2 \varphi_j(x_i) f(x_i).$$

Since $\hat{\lambda}$ satisfies the Dantzig constraint, we have with probability at least $1 - M^{1-\tau/2}$, for any $j \in \{1,\ldots,M\}$,

$$|(G\hat{\lambda})_j - \beta_j| \leq |(G\hat{\lambda})_j - \hat{\beta}_j| + |\hat{\beta}_j - \beta_j| \leq 2r_{n,j}$$

and $|A| \leq 4r_n\|\Delta\|_1$. This implies that

$$\|\hat{f} - f\|_n^2 \leq \|f_\lambda - f\|_n^2 + 4r_n\|\Delta\|_1 - \|f_\Delta\|_n^2.$$

Moreover using Lemma 1 and Proposition 1 of Bertin et al (2011) (where the norm $\|\cdot\|_2$ is replaced by $\|\cdot\|_n$), we obtain that

$$\left(\|\Delta_{J_0^c}\|_{\ell_1} - \|\Delta_{J_0}\|_{\ell_1}\right)_+ \leq 2\|\lambda_{J_0^c}\|_{\ell_1} + \left(\|\hat{\lambda}\|_{\ell_1} - \|\lambda\|_{\ell_1}\right)_+ \tag{4}$$

and

$$\begin{aligned}
\|f_\Delta\|_n &\geq \kappa_s\|\Delta_{J_0}\|_{\ell_2} - \frac{\mu_s}{\sqrt{|J_0|}}\left(\|\Delta_{J_0^c}\|_{\ell_1} - \|\Delta_{J_0}\|_{\ell_1}\right)_+ \\
&\geq \kappa_s\|\Delta_{J_0}\|_{\ell_2} - 2\frac{\mu_s}{\sqrt{|J_0|}}\Lambda(\lambda, J_0^c).
\end{aligned}$$

Note that Proposition 1 of Bertin et al (2011) is obtained using Lemma 2 and Lemma 3 of Bertin et al (2011). In our context, Lemma 2 and Lemma 3 can be proved in the same way by replacing the norm $\|\cdot\|_2$ by $\|\cdot\|_n$ and by considering $P_{J_{01}}$ as the projector on the linear space spanned by $(\varphi_j(x_1),\dots,\varphi_j(x_n))_{j\in J_{01}}$.

Now following the same lines as Theorem 2 of Bertin et al (2011), replacing κ_{J_0} by κ_s and μ_{J_0} by μ_s, we obtain the result of the theorem.

2.3 Proof of Theorem 2

We consider $\hat{\lambda}^D$ defined by

$$\hat{\lambda}^D = \operatorname{argmin}_{\lambda\in\mathbb{R}^M}\|\lambda\|_{\ell_1} \quad \text{such that } \lambda \text{ satisfies the Dantzig constraint (1).}$$

Denote by $\hat{f}^D$ the estimator $f_{\hat{\lambda}^D}$. Following the same lines as in the proof of Theorem 1, it can be obtained that, with probability at least $1-M^{1-\tau/2}$, for any integer $s<n/2$ such that (A1(s)) holds, we have for any $\alpha>0$,

$$\|\hat{f}^D-f\|_n^2 \le \inf_{\lambda\in\mathbb{R}^M}\inf_{\substack{J_0\subset\{1,\dots,M\}\\|J_0|=s}}\left\{\|f_\lambda-f\|_n^2+\alpha\left(1+\frac{2\mu_s}{\kappa_s}\right)^2\frac{\Lambda(\lambda,J_0^c)^2}{s}+16s\left(\frac{1}{\alpha}+\frac{1}{\kappa_s^2}\right)r_n^2\right\},$$

where here

$$\Lambda(\lambda,J_0^c)=\|\lambda_{J_0^C}\|_{\ell_1}+\frac{\left(\|\hat{\lambda}^D\|_{\ell_1}-\|\lambda\|_{\ell_1}\right)_+}{2}.$$

If the infimum is only taken over the vectors λ that satisfy the Dantzig constraint, then, with the same probability we have

$$\|\hat{f}^D-f\|_n^2 \le \inf_{\lambda\in\mathscr{D}}\inf_{\substack{J_0\subset\{1,\dots,M\}\\|J_0|=s}}\left\{\|f_\lambda-f\|_n^2+\alpha\left(1+\frac{2\mu_s}{\kappa_s}\right)^2\frac{\|\lambda_{J_0^C}\|_{l_1}^2}{s}+16s\left(\frac{1}{\alpha}+\frac{1}{\kappa_s^2}\right)r_n^2\right\}. \tag{5}$$

Following the same lines as the proof of Theorem 1, replacing λ by $\hat{\lambda}^D$, we obtain, with probability at least $1-M^{1-\tau/2}$,

$$\|\hat{f}-f\|_n^2 \le \|\hat{f}^D-f\|_n^2+4r_n\|\Delta\|_1-\|f_\Delta\|_n^2,$$

with $\Delta=\hat{\lambda}-\hat{\lambda}^D$. Applying (4) where $\hat{\lambda}$ plays the role of λ and $\hat{\lambda}^D$ the role of $\hat{\lambda}$, the vector Δ satisfies

$$\left(\|\Delta_{J_0^C}\|_{\ell_1}-\|\Delta_{J_0}\|_{\ell_1}\right)_+ \le 2\|\hat{\lambda}_{J_0^C}\|_{\ell_1}.$$

Following the same lines as in the proof of Theorem 1, we obtain that for each $J_0\subset\{1,\dots,M\}$ such that $|J_0|=s$

$$\|\hat{f}-f\|_n^2 \le \left\{\|\hat{f}^D-f\|_n^2+\alpha\left(1+\frac{2\mu_s}{\kappa_s}\right)^2\frac{\|\hat{\lambda}_{J_0^C}\|_{l_1}^2}{s}+16s\left(\frac{1}{\alpha}+\frac{1}{\kappa_s^2}\right)r_n^2\right\}. \tag{6}$$

Finally, (5) and (6) imply the theorem.

2.4 Proof of Theorem 3

We first state the following lemma.

Lemma 2 *We have for any $u \in \mathbb{R}^M$,*

$$crit(\hat{\lambda}+u) - crit(\hat{\lambda}) \geq \left\| \sum_{k=1}^{M} u_k \varphi_k \right\|_n^2 .$$

Proof of Lemma 2: Since for any λ,

$$\mathrm{crit}(\lambda) = \frac{1}{n}\sum_{i=1}^{n} (y_i - b_i f_\lambda(x_i))^2 + 2\sum_{j=1}^{M} \tilde{r}_{n,j}|\lambda_j|,$$

$$\begin{aligned}
\mathrm{crit}(\hat{\lambda}+u) - \mathrm{crit}(\hat{\lambda}) &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - b_i\sum_{k=1}^{M}\hat{\lambda}_k\varphi_k(x_i) - b_i\sum_{k=1}^{M}u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}|\hat{\lambda}_j + u_j| \\
&\quad - \frac{1}{n}\sum_{i=1}^{n}\left(y_i - b_i\sum_{k=1}^{M}\hat{\lambda}_k\varphi_k(x_i)\right)^2 - 2\sum_{j=1}^{M}\tilde{r}_{n,j}|\hat{\lambda}_j| \\
&= \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M}u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}\left(|\hat{\lambda}_j + u_j| - |\hat{\lambda}_j|\right) \\
&\quad - \frac{2}{n}\sum_{i=1}^{n}\left(y_i - b_i\sum_{k=1}^{M}\hat{\lambda}_k\varphi_k(x_i)\right) b_i\sum_{k=1}^{M}u_k\varphi_k(x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M}u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}\left(|\hat{\lambda}_j + u_j| - |\hat{\lambda}_j|\right) \\
&\quad + \frac{2}{n}\sum_{i=1}^{n} b_i^2\sum_{j=1}^{M}\hat{\lambda}_j\varphi_j(x_i)\sum_{k=1}^{M}u_k\varphi_k(x_i) - \frac{2}{n}\sum_{i=1}^{n} b_i y_i\sum_{k=1}^{M}u_k\varphi_k(x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M}u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}\left(|\hat{\lambda}_j + u_j| - |\hat{\lambda}_j|\right) \\
&\quad + \frac{2}{n}\sum_{i=1}^{n}\sum_{k=1}^{M}u_k\varphi_k(x_i)\left(b_i^2\sum_{j=1}^{M}\hat{\lambda}_j\varphi_j(x_i) - b_i y_i\right).
\end{aligned}$$

Since $\hat{\lambda}$ minimizes $\lambda \longmapsto \mathrm{crit}(\lambda)$, we have for any k,

$$0 = \frac{2}{n}\sum_{i=1}^{n}\varphi_k(x_i)\left(b_i^2\sum_{j=1}^{M}\hat{\lambda}_j\varphi_j(x_i) - b_i y_i\right) + 2\tilde{r}_{n,k}s(\hat{\lambda}_k),$$

where $|s(\hat{\lambda}_k)| \leq 1$ and $s(\hat{\lambda}_k) = \mathrm{sign}(\hat{\lambda}_k)$ if $\hat{\lambda}_k \neq 0$. So,

$$\frac{2}{n}\sum_{i=1}^{n}\sum_{k=1}^{M}u_k\varphi_k(x_i)\left(b_i^2\sum_{j=1}^{M}\hat{\lambda}_j\varphi_j(x_i) - b_i y_i\right) = -2\sum_{k=1}^{M}u_k\tilde{r}_{n,k}s(\hat{\lambda}_k)$$

and

$$\begin{aligned}\mathrm{crit}(\hat{\lambda}+u)-\mathrm{crit}(\hat{\lambda}) &= \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M} u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}\left(|\hat{\lambda}_j+u_j|-|\hat{\lambda}_j|\right)\\ &\qquad -2\sum_{k=1}^{M} u_k\tilde{r}_{n,k}s(\hat{\lambda}_k)\\ &= \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M} u_k\varphi_k(x_i)\right)^2 + 2\sum_{j=1}^{M}\tilde{r}_{n,j}\left(|\hat{\lambda}_j+u_j|-|\hat{\lambda}_j|-u_j s(\hat{\lambda}_j)\right)\\ &\geq \frac{1}{n}\sum_{i=1}^{n} b_i^2\left(\sum_{k=1}^{M} u_k\varphi_k(x_i)\right)^2,\end{aligned}$$

which proves the result. □

Now, still with $s^* = \mathrm{card}(S^*)$, we consider for $\mu \in \mathbb{R}^{s^*}$

$$\mathrm{critS}^*(\mu) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - b_i\sum_{j\in S^*}\mu_j\varphi_j(x_i)\right)^2 + 2\sum_{j\in S^*}\tilde{r}_{n,j}|\mu_j|,$$

and

$$\tilde{\mu} = \arg\min_{\mu\in\mathbb{R}^{s^*}} \mathrm{critS}^*(\mu).$$

Then we set

$$\mathscr{S} = \bigcap_{j\notin S^*}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k\in S^*}\tilde{\mu}_k < \varphi_j,\varphi_k>\right| < \tilde{r}_{n,j}\right\}$$

and we state the following lemma.

Lemma 3 *On the set $\mathscr{S}$, the non-zero coordinates of $\hat{\lambda}$ are included into S^*.*

Proof of Lemma 3: Recall that $\hat{\lambda}$ is a minimizer of $\lambda \longmapsto \mathrm{crit}(\lambda)$. Using standard convex analysis arguments, this is equivalent to say that for any $1 \leq j \leq M$,

$$\begin{cases} \frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k=1}^{M}\hat{\lambda}_k < \varphi_j,\varphi_k > \;= \tilde{r}_{n,j}\mathrm{sign}(\hat{\lambda}_j) & \text{if } \hat{\lambda}_j \neq 0,\\ \left|\frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k=1}^{M}\hat{\lambda}_k < \varphi_j,\varphi_k >\right| \leq \tilde{r}_{n,j} & \text{if } \hat{\lambda}_j = 0.\end{cases}$$

Similarly, on $\mathscr{S}$, we have

$$\begin{cases} \frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k\in S^*}\tilde{\mu}_k < \varphi_j,\varphi_k > \;= \tilde{r}_{n,j}\mathrm{sign}(\tilde{\mu}_j) & \text{if } j\in S^* \text{ and } \tilde{\mu}_j \neq 0,\\ \left|\frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k\in S^*}\tilde{\mu}_k < \varphi_j,\varphi_k >\right| \leq \tilde{r}_{n,j} & \text{if } j\in S^* \text{ and } \tilde{\mu}_j = 0,\\ \left|\frac{1}{n}\sum_{i=1}^{n} y_i b_i\varphi_j(x_i) - \sum_{k\in S^*}\tilde{\mu}_k < \varphi_j,\varphi_k >\right| < \tilde{r}_{n,j} & \text{if } j\notin S^*.\end{cases}$$

So, on $\mathscr{S}$, the vector $\hat{\mu}$ such $\hat{\mu}_j = \tilde{\mu}_j$ if $j \in S^*$ and $\hat{\mu}_j = 0$ if $j \notin S^*$ is also a minimizer of $\lambda \longmapsto \mathrm{crit}(\lambda)$. Using Lemma 2, we have for any $1 \leq i \leq n$:

$$\sum_{k=1}^{M}(\hat{\lambda}_k - \hat{\mu}_k)\varphi_k(x_i) = 0.$$

So, for $j \notin S^*$,

$$\left| \frac{1}{n} \sum_{i=1}^{n} y_i b_i \varphi_j(x_i) - \sum_{k=1}^{M} \hat{\lambda}_k < \varphi_j, \varphi_k > \right| < \tilde{r}_{n,j}.$$

Therefore, on $\mathscr{S}$, the non-zero coordinates of $\hat{\lambda}$ are included into S^*. □

Lemma 3 shows that we just need to prove that

$$\mathbb{P}\{\mathscr{S}\} \geq 1 - 2M^{1-\tau/2}$$

$$\begin{aligned}
\mathbb{P}\{\mathscr{S}^c\} &\leq \sum_{j \notin S^*} \mathbb{P}\left\{ \left| \frac{1}{n} \sum_{i=1}^{n} y_i b_i \varphi_j(x_i) - \sum_{k \in S^*} \tilde{\mu}_k < \varphi_j, \varphi_k > \right| \geq \tilde{r}_{n,j} \right\} \\
&\leq A + B,
\end{aligned}$$

with

$$\begin{aligned}
A &= \sum_{j \notin S^*} \mathbb{P}\left\{ \left| \frac{1}{n} \sum_{i=1}^{n} [y_i b_i \varphi_j(x_i) - \mathbb{E}(y_i b_i \varphi_j(x_i))] \right| \geq r_{n,j} \right\} \\
&= \sum_{j \notin S^*} \mathbb{P}\left\{ \left| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i b_i \varphi_j(x_i) \right| \geq r_{n,j} \right\} \\
&= \sum_{j \notin S^*} \mathbb{P}\left\{ |V_j| \geq r_{n,j} \right\}
\end{aligned}$$

(see Lemma 1) and

$$\begin{aligned}
B &= \mathbb{P}\left[\bigcup_{j \notin S^*} \left\{ \left| \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(y_i b_i \varphi_j(x_i)) - \sum_{k \in S^*} \tilde{\mu}_k < \varphi_j, \varphi_k > \right| \geq \tilde{r}_{n,j} - r_{n,j} \right\} \right] \\
&= \mathbb{P}\left[\bigcup_{j \notin S^*} \left\{ \left| < \varphi_j, f_{\lambda^*} > - \sum_{k \in S^*} \tilde{\mu}_k < \varphi_j, \varphi_k > \right| \geq \tilde{r}_{n,j} - r_{n,j} \right\} \right] \\
&= \mathbb{P}\left[\bigcup_{j \notin S^*} \left\{ \left| \sum_{k \in S^*} (\lambda_k^* - \tilde{\mu}_k) < \varphi_j, \varphi_k > \right| \geq \tilde{r}_{n,j} - r_{n,j} \right\} \right] \\
&\leq \mathbb{P}\left[\bigcup_{j \notin S^*} \left\{ \rho(S^*) \|\varphi_j\|_n \sum_{k \in S^*} |\lambda_k^* - \tilde{\mu}_k| \, \|\varphi_k\|_n \geq \tilde{r}_{n,j} - r_{n,j} \right\} \right]
\end{aligned}$$

since

$$\rho(S^*) = \max_{k \in S^*} \max_{j \neq k} \frac{| < \varphi_j, \varphi_k > |}{\|\varphi_j\|_n \|\varphi_k\|_n}.$$

Using notation of Lemma 3, we have:

$$\begin{aligned}
\|f_{\lambda^*} - f_{\hat{\mu}}\|_n^2 &= \| \sum_{k \in S^*} (\lambda_k^* - \hat{\mu}_k) \varphi_k \|_n^2 \\
&= \sum_{k \in S^*} (\lambda_k^* - \hat{\mu}_k)^2 \|\varphi_k\|_n^2 + \sum_{k \in S^*} \sum_{j \in S^*, \; j \neq k} (\lambda_k^* - \hat{\mu}_k)(\lambda_j^* - \hat{\mu}_j) < \varphi_j, \varphi_k >,
\end{aligned}$$

and

$$\sum_{k\in S^*}(\lambda_k^*-\hat\mu_k)^2\|\varphi_k\|_n^2 \le \|f_{\lambda^*}-f_{\hat\mu}\|_n^2+\rho(S^*)\sum_{k\in S^*}\sum_{j\in S^*,\ j\neq k}|\lambda_k^*-\hat\mu_k|\|\varphi_k\|_n\times|\lambda_j^*-\hat\mu_j|\|\varphi_j\|_n$$
$$\le \|f_{\lambda^*}-f_{\hat\mu}\|_n^2+\rho(S^*)\left(\sum_{k\in S^*}|\lambda_k^*-\hat\mu_k|\|\varphi_k\|_n\right)^2.$$

Finally,

$$\left(\sum_{k\in S^*}|\lambda_k^*-\hat\mu_k|\|\varphi_k\|_n\right)^2 \le s^*\sum_{k\in S^*}(\lambda_k^*-\hat\mu_k)^2\|\varphi_k\|_n^2$$
$$\le s^*\left(\|f_{\lambda^*}-f_{\hat\mu}\|_n^2+\rho(S^*)\left(\sum_{k\in S^*}|\lambda_k^*-\hat\mu_k|\|\varphi_k\|_n\right)^2\right),$$

which shows that

$$\left(\sum_{k\in S^*}|\lambda_k^*-\hat\mu_k|\|\varphi_k\|_n\right)^2 \le \frac{s^*}{1-\rho(S^*)s^*}\|f_{\lambda^*}-f_{\hat\mu}\|_n^2.$$

Now,

$$\frac{1}{n}\sum_{i=1}^n\left(y_i-b_i\sum_{j\in S^*}\tilde\mu_j\varphi_j(x_i)\right)^2+2\sum_{j\in S^*}\tilde r_{n,j}|\tilde\mu_j| \le$$
$$\frac{1}{n}\sum_{i=1}^n\left(y_i-b_i\sum_{j\in S^*}\lambda_j^*\varphi_j(x_i)\right)^2+2\sum_{j\in S^*}\tilde r_{n,j}|\lambda_j^*|.$$

So,

$$\|\sum_{j\in S^*}\tilde\mu_j\varphi_j\|_n^2-\frac{2}{n}\sum_{i=1}^n b_iy_i\sum_{j\in S^*}\tilde\mu_j\varphi_j(x_i)+2\sum_{j\in S^*}\tilde r_{n,j}|\tilde\mu_j| \le$$
$$\|\sum_{j\in S^*}\lambda_j^*\varphi_j\|_n^2-\frac{2}{n}\sum_{i=1}^n b_iy_i\sum_{j\in S^*}\lambda_j^*\varphi_j(x_i)+2\sum_{j\in S^*}\tilde r_{n,j}|\lambda_j^*|,$$

and using previous notation,

$$\|f_{\hat\mu}\|_n^2-\frac{2}{n}\sum_{i=1}^n b_iy_i\sum_{j\in S^*}\tilde\mu_j\varphi_j(x_i)+2\sum_{j\in S^*}\tilde r_{n,j}|\tilde\mu_j| \le$$
$$\|f_{\lambda^*}\|_n^2-\frac{2}{n}\sum_{i=1}^n b_iy_i\sum_{j\in S^*}\lambda_j^*\varphi_j(x_i)+2\sum_{j\in S^*}\tilde r_{n,j}|\lambda_j^*|.$$

Therefore,

$$\begin{aligned}
\|f_{\lambda^*} - f_{\hat{\mu}}\|_n^2 &= \|f_{\hat{\mu}}\|_n^2 + \|f_{\lambda^*}\|_n^2 - 2 < f_{\hat{\mu}}, f_{\lambda^*} > \\
&\leq 2\|f_{\lambda^*}\|_n^2 - 2 < f_{\hat{\mu}}, f_{\lambda^*} > + \frac{2}{n}\sum_{i=1}^{n} b_i y_i \sum_{j\in S^*} (\tilde{\mu}_j - \lambda_j^*)\varphi_j(x_i) + 2\sum_{j\in S^*} \tilde{r}_{n,j}(|\lambda_j^*| - |\tilde{\mu}_j|) \\
&= \frac{2}{n}\sum_{i=1}^{n} b_i y_i (f_{\hat{\mu}}(x_i) - f_{\lambda^*}(x_i)) - \frac{2}{n}\sum_{i=1}^{n} b_i^2 f_{\lambda^*}(x_i)(f_{\hat{\mu}}(x_i) - f_{\lambda^*}(x_i)) \\
&\qquad + 2\sum_{j\in S^*} \tilde{r}_{n,j}(|\lambda_j^*| - |\tilde{\mu}_j|) \\
&= \frac{2}{n}\sum_{i=1}^{n} b_i (y_i - \mathbb{E}(y_i))(f_{\hat{\mu}}(x_i) - f_{\lambda^*}(x_i)) + 2\sum_{j\in S^*} \tilde{r}_{n,j}(|\lambda_j^*| - |\tilde{\mu}_j|) \\
&= \frac{2}{n}\sum_{i=1}^{n} b_i \varepsilon_i (f_{\hat{\mu}}(x_i) - f_{\lambda^*}(x_i)) + 2\sum_{j\in S^*} \tilde{r}_{n,j}(|\lambda_j^*| - |\tilde{\mu}_j|) \\
&= 2\sum_{j=1}^{M} V_j(\hat{\mu}_j - \lambda_j^*) + 2\sum_{j\in S^*} \tilde{r}_{n,j}(|\lambda_j^*| - |\tilde{\mu}_j|).
\end{aligned}$$

Now let us assume that for any $j \in S^*$, $V_j < r_{n,j}$. Then,

$$\begin{aligned}
\|f_{\lambda^*} - f_{\hat{\mu}}\|_n^2 &< 2\sum_{j\in S^*} (r_{n,j} + \tilde{r}_{n,j})|\hat{\mu}_j - \lambda_j^*| \\
&< 2\sigma\sqrt{\frac{\log M}{n}}(\sqrt{\tau} + \sqrt{\tilde{\tau}}) \sum_{j\in S^*} \|\varphi_j\|_n |\hat{\mu}_j - \lambda_j^*|.
\end{aligned}$$

So,

$$\sum_{k\in S^*} |\lambda_k^* - \hat{\mu}_k| \|\varphi_k\|_n < 2\sigma\sqrt{\frac{\log M}{n}}(\sqrt{\tau} + \sqrt{\tilde{\tau}}) \frac{s^*}{1 - \rho(S^*)s^*}$$

and for any $j \notin S^*$,

$$\begin{aligned}
\rho(S^*)\|\varphi_j\|_n \sum_{k\in S^*} |\lambda_k^* - \hat{\mu}_k| \|\varphi_k\|_n &< 2\sigma\sqrt{\frac{\log M}{n}} \|\varphi_j\|_n (\sqrt{\tau} + \sqrt{\tilde{\tau}}) \frac{\rho(S^*)s^*}{1 - \rho(S^*)s^*} \\
&< \frac{2\sigma c(\sqrt{\tau} + \sqrt{\tilde{\tau}})}{1 - c} \sqrt{\frac{\log M}{n}} \|\varphi_j\|_n \\
&< (\sqrt{\tilde{\tau}} - \sqrt{\tau})\sigma\sqrt{\frac{\log M}{n}} \|\varphi_j\|_n \\
&< \tilde{r}_{n,j} - r_{n,j}.
\end{aligned}$$

Therefore,

$$B \leq \sum_{j\in S^*} \mathbb{P}\left\{|V_j| \geq r_{n,j}\right\}$$

and using Lemma 1, since $\mathbb{P}\{\mathscr{S}^c\} \leq A + B$,

$$\mathbb{P}\{\mathscr{S}\} \geq 1 - 2M^{1-\tau/2}.$$

2.5 Proof of Corollary 1

First note that λ^* satisfies the Dantzig constraint (1) where $r_{n,j}$ is replaced by $\tilde{r}_{n,j}$ with probability larger than $1 - M^{1-\tilde{\tau}/2}$. On the event $\hat{S} \subset S^*$, we have $\lambda^*_{(S^*)^C} = \hat{\lambda}_{(S^*)^C} = 0$, then applying Theorem 2, we obtain that for any $\alpha > 0$

$$\|\hat{f} - f\|_n^2 \leq 32 s^* \left(\frac{1}{\alpha} + \frac{1}{\kappa_{s^*}^2} \right) \tilde{r}_n^2,$$

which implies the result of the theorem.

References

Bertin K, Le Pennec E, Rivoirard V (2011) Adaptive Dantzig density estimation. Annales de l'Institut Henri Poincaré 47:43–74

Bickel PJ, Ritov Y, Tsybakov AB (2009) Simultaneous analysis of lasso and Dantzig selector. Ann Statist 37(4):1705–1732

Bunea F (2008) Consistent selection via the Lasso for high dimensional approximating regression models. In: Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh, Inst. Math. Stat. Collect., vol 3, Inst. Math. Statist., Beachwood, OH, pp 122–137

Bunea F, Tsybakov AB, Wegkamp MH (2006) Aggregation and sparsity via l_1 penalized least squares. In: Learning theory, Lecture Notes in Comput. Sci., vol 4005, Springer, Berlin, pp 379–391

Bunea F, Tsybakov A, Wegkamp M (2007a) Sparsity oracle inequalities for the Lasso. Electronic Journal of Statistics 1:169–194

Bunea F, Tsybakov AB, Wegkamp MH (2007b) Aggregation for Gaussian regression. The Annals of Statistics 35(4):1674–1697

Candès EJ, Tao T (2007) The Dantzig selector: statistical estimation when p is much larger than n. The Annals of Statistics 35(6):2313–2351

van de Geer S (2010) ℓ_1-regularization in high-dimensional statistical models. In: Proceedings of the International Congress of Mathematicians. Volume IV, Hindustan Book Agency, New Delhi, pp 2351–2369

van de Geer SA, Bühlmann P (2009) On the conditions used to prove oracle results for the Lasso. Electronic Journal of Statistics 3:1360–1392