AStA Adv Stat Anal (2010) 94: 367–388
DOI 10.1007/s10182-010-0148-8

ORIGINAL PAPER

Using recursive algorithms for the efficient identification of smoothing spline ANOVA models

Marco Ratto · Andrea Pagano

Received: 4 September 2009 / Accepted: 26 July 2010

Abstract In this paper we present a unified discussion of different approaches to the identification of smoothing spline analysis of variance (ANOVA) models: (i) the "classical" approach (in the line of Wahba in Spline Models for Observational Data, 1990; Gu in Smoothing Spline ANOVA Models, 2002; Storlie et al. in Stat. Sin., 2011) and (ii) the State-Dependent Regression (SDR) approach of Young in Nonlinear Dynamics and Statistics (2001). The latter is a nonparametric approach which is very similar to smoothing splines and kernel regression methods, but based on recursive filtering and smoothing estimation (the Kalman filter combined with fixed interval smoothing). We will show that SDR can be effectively combined with the "classical" approach to obtain a more accurate and efficient estimation of smoothing spline ANOVA models to be applied for emulation purposes. We will also show that such an approach can compare favorably with kriging.

Keywords Smoothing spline ANOVA models · Recursive algorithms · Backfitting · Sensitivity analysis

1 Introduction

In the analysis of data from computer experiments, approximation models are built to emulate the behavior of large computational models. Smoothing spline models are

M. Ratto (✉) · A. Pagano
Joint Research Centre, European Commission, TP361 IPSC, Via Fermi 2749, 21027 Ispra (VA), Italy
e-mail: marco.ratto@jrc.it

A. Pagano
e-mail: andrea.pagano@cmcc.it

Present address:
A. Pagano
Euro-Mediterranean Centre for Climate Change, Milano, Italy

a useful tool for this kind of analysis (Levy and Steinberg 2010). In this paper we present a unified discussion of different approaches to the identification of smoothing spline analysis of variance (ANOVA) models. The "classical" approach to smoothing spline ANOVA models is along the same lines as the work of Wahba (1990) and Gu (2002). Recently, Storlie et al. (2011) presented the Adaptive COmponent Selection and Selection Operator (ACOSSO), "a new regularization method for simultaneous model fitting and variable selection in nonparametric regression models in the framework of smoothing spline ANOVA." This method is an improvement to COSSO (Lin and Zhang 2006), penalizing the sum of component norms, instead of the squared norm employed in the traditional smoothing spline method. In ACOSSO, an adaptive weight is used in the COSSO penalty which allows for more flexibility in estimating important functional components while giving a heavier penalty to unimportant functional components.

In a "parallel" stream of research, using the State-Dependent (Parameter) Regression (SDR) approach of Young (2001), Ratto et al. (2007) have developed a nonparametric approach, very similar to smoothing splines and kernel regression methods, based on recursive filtering and smoothing estimation (the Kalman filter combined with fixed interval smoothing). Such a recursive least-squares implementation has some key characteristics: (a) it is plugged with optimal maximum likelihood estimation, thus allowing for an objective estimation of the smoothing hyper-parameters, and (b) it provides greater flexibility in adapting to local discontinuities, heavy nonlinearity, and heteroscedastic error terms. The use of recursive algorithms in smoothing splines is not new in statistical literature: the works of Weinert et al. (1980) and Wecker and Ansley (1983) demonstrated the applicability of a stochastic framework for recursive computation of smoothing splines. However, such works were limited to the univariate case, while the subsequent history of tensor product smoothing splines developed in the "standard" nonrecursive form. The recursive approach of Young (2001) can be seen as an extension and generalization of such earlier papers, which is applicable to the multivariate case, and Ratto et al. (2007) discussed its possible extension for the estimation of interaction terms of the ANOVA.

The purposes of this paper are:

1. to develop a formal comparison and demonstrate equivalences between the "classical" tensor product smoothing spline with reproducing kernel Hilbert space (RKHS) algebra and the SDR, extending the results derived in Weinert et al. (1980) and Wecker and Ansley (1983) to the multivariate case;
2. to discuss the advantages and disadvantages of these approaches;
3. to propose a unified approach to smoothing spline ANOVA models that combines the best of the discussed methods: the use of the recursive algorithms in particular can be very effective in detecting the important functional components, adding valuable information in the ACOSSO framework; at the same time, such a combined approach improves the first extension of the SDR approach to interaction terms proposed by Ratto et al. (2007);
4. to compare results of analysis of computer experiments carried out using kriging-based techniques, e.g., Design and Analysis of Computer Experiments (DACE) and the proposed unified method.

Concerning item 1 above, for the sake of parsimony we concentrate here on the special case of cubic smoothing splines. This case is, in fact, the most widely applied spline method as well as the basic core in the ACOSSO framework. However, similarly to the discussion in Wecker and Ansley (1983), a full class of equivalences can be derived between the SDR recursive approach and polynomial splines of any order. In this context, we also note that Young and Pedregal (1999) already discussed the equivalence between the recursive and en bloc formulation of the smoothing problem, when the en bloc case is cast in the "discrete" form, as in the Hodrick–Prescott filter (Hodrick and Prescott 1981; Leser 1961). Here we extend such equivalence to the "continuous" integral form typical of the RKHS algebra.

2 State-dependent regressions and smoothing splines

2.1 Additive models

We denote the generic mapping as $z(\mathbf{X})$ and assume without loss of generality that $\mathbf{X} \in [0,1]^p$, where p is the number of input variables. The simplest example of smoothing spline mapping estimation of z is the additive model:

$$f(\mathbf{X}) = f_0 + \sum_{j=1}^{p} f_j(X_j). \tag{1}$$

To estimate f we can use a multivariate (cubic) smoothing spline minimization problem, that is, given $\lambda = (\lambda_1, \ldots, \lambda_p)$, find the minimizer $f(X_k)$ of:

$$\frac{1}{N}\sum_{k=1}^{N}\big(z_k - f(\mathbf{X}_k)\big)^2 + \sum_{j=1}^{p}\lambda_j \int_0^1 \big[f_j''(X_j)\big]^2 dX_j, \tag{2}$$

where a Monte Carlo (MC) sample of dimension N is assumed.

This statistical estimation problem requires the estimation of the p hyperparameters λ_j (also denoted as smoothing parameters). Various ways of doing that are available in the literature, by applying generalized cross-validation (GCV), generalized maximum likelihood (GML) procedures and so on (see, e.g., Wahba 1990; Gu 2002). Here we discuss the recursive estimation approach, where the additive model is put into the *State-Dependent Parameter Regression* (SDR) form of Young (2001) as applied to the estimation of ANOVA models by Ratto et al. (2007).

We highlight here the key features of Young's recursive algorithms of SDR by considering the case of $p = 1$ and $z(X) = g(X) + e$, with $e \sim N(0, \sigma^2)$; i.e., we rewrite the smoothing problem as $z_k = s_k + e_k$, where $k = 1, \ldots, N$ and s_k is the estimate of $g(X_k)$. To make the recursive approach meaningful, the MC sample needs to be sorted in ascending order with respect to X: i.e., k and $k-1$ subscripts are adjacent elements under such ordering (see Fig. 1), implying $0 \le X_1 < X_2 < \cdots < X_k < \cdots < X_N \le 1$.

To recursively estimate the s_k in SDR, it is necessary to characterize it in some stochastic manner, borrowing from nonstationary time series processes. In general

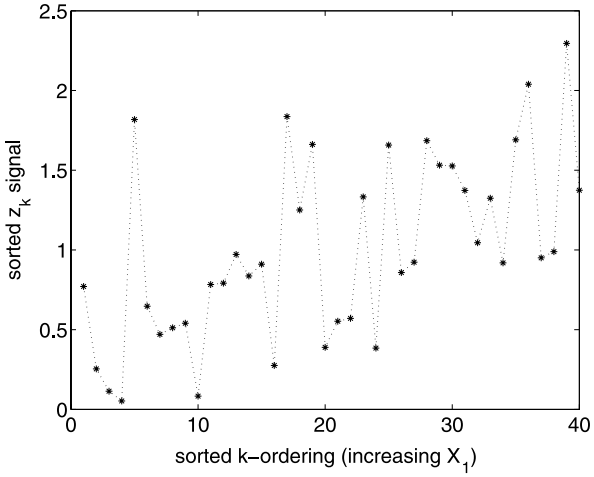

Fig. 1 Example of the sorted k-ordering: the recursive algorithm spans the output data z_k from the left to the right of the plot

this is accomplished by assuming that the evolution of s_k follows one member of the generalized random walk (GRW) class on nonstationary random sequences (see, e.g., Young and Ng 1989 and Ng and Young 1990).

In the present context, the integrated random walk (IRW) process provides the same smoothing properties of a cubic spline, in the overall state-space formulation:

$$\begin{array}{ll} \text{Observation Equation:} & z_k = s_k + e_k \\ \text{State Equations:} & s_k = s_{k-1} + d_{k-1} \\ & d_k = d_{k-1} + \eta_k \end{array} \tag{3}$$

where d_k is the "slope" of s_k, $\eta_k \sim N(0, \sigma_\eta^2)$ and η_k ("system disturbance" in systems terminology) is assumed to be independent of the "observation noise" $e_k \sim N(0, \sigma^2)$.

Within the framework of the analysis of computer experiments, it seems necessary to discuss the term e_k in (3). Normality and independence are strictly appropriate when there is observational error in the data but can be reasonable for smoothing observed data even in computer experiments because there can be applications where the "computed" value is produced with some error or variability, due to, e.g., convergence of numerical algorithms. Moreover, e_k descends naturally from the standard smoothing spline formulation (2): such a penalized least-squares regression rules out a perfect fit for $f(\mathbf{X}_k)$, thus implying an "observation noise" linked to the nonzero residual term $(z_k - f(\mathbf{X}_k))$. This residual reflects the fact that the present emulation approach is based on identifying a truncated ANOVA expansion that approximates $z(\mathbf{X})$. This is done by including a "small" subset of q ANOVA terms (main effects and interactions) that are statistically identifiable from the available MC sample. Thus, e_k can be seen as the sum of all the terms that are not included by a shrinkage procedure. This set of dropped ANOVA terms usually includes a very large number of elements (namely $2^p - q$, where $q \ll 2^p$), which are orthogonal (independent) by definition. It does not seem out of place to model the sum of a large number of independent variables in statistical terms (Central Limit Theorem). We will see that the inclusion of this "error" term, rather than being a drawback of this method, turns

out to be an advantage (see the examples in Sect. 3), since it implies that emulation (and therefore "prediction" at untried X values) is performed only using statistically significant ANOVA terms, enhancing the robustness of out-of-sample performances.

Given the ascending ordering of the MC sample, s_k can be estimated by using the recursive Kalman filter (KF) and the associated recursive fixed interval smoothing (FIS) algorithm (see, e.g., Kalman 1960; Young 1999 for details).

First, it is necessary to optimize the hyper-parameters associated with the state-space model (3), namely the white noise variances σ^2 and σ_η^2. In fact, by a simple reformulation of the KF and FIS algorithms, the IRW model can be entirely characterized by one noise variance ratio (NVR) hyper-parameter, where $\text{NVR} = \sigma_\eta^2/\sigma^2$. This NVR value is, of course, unknown *a priori* and needs to be optimized: for example, in the above references, this is accomplished by maximum likelihood (ML) optimization using prediction error decomposition (Schweppe 1965). The NVR plays the inverse role of a smoothing parameter: the smaller the NVR, the smoother the estimate of s_k (and at the limit $\text{NVR} = 0$, s_k will be a straight line). Given the NVR, the FIS algorithm then yields an estimate $\hat{s}_{k|N}$ of s_k at each data sample, and it can be seen that the $\hat{s}_{k|N}$ from the IRW process is the equivalent of $f(X_k)$ in the cubic smoothing spline model. At the same time, the recursive procedures provide, in a natural way, standard errors of the estimated $\hat{s}_{k|N}$ that allow for the testing of their relative significance.

We need to clarify here the meaning of the ML optimization in this recursive context. We first observe that, to avoid a perfect fit solution, a penalty term is used in the "classical" smoothing spline estimates. The "penalty" appears in the objective function (GCV, GML, etc.) which is used to optimize the λ's. In this way one may limit the "degrees of freedom" of the spline model.

For example, in GCV, we have to find a λ that minimizes

$$\text{GCV}_\lambda = 1/N \cdot \frac{\sum_k (z_k - f_\lambda(X_k))^2}{(1 - df(\lambda)/N)^2}, \tag{4}$$

where $df \in [0, N]$ denotes the "degrees of freedom" as a function of λ. Similarly, in the SDR notation, we look for an NVR minimizing

$$\text{GCV}_{\text{NVR}} = 1/N \cdot \frac{\sum_k (z_k - \hat{s}_{k|N})^2}{(1 - df(\text{NVR})/N)^2}, \tag{5}$$

where the "degrees of freedom" df depend on the NVR (by the above-mentioned equivalence).

Remark 1 Without the penalty term, the optimum would always be attained at $\lambda = 0$ (or $\text{NVR} \to \infty$), i.e., perfect fit.

A perfect fit "optimal" solution would never happen within the SDR recursive context. In this case penalty is directly associated with the estimate procedure, by the fact that the prediction error decomposition (ML) estimate is based on the *filtered* estimate $\hat{s}_{k|k-1} = s_{k-1} + d_{k-1}$ and not on the smoothed estimate $\hat{s}_{k|N}$. Namely, in ML

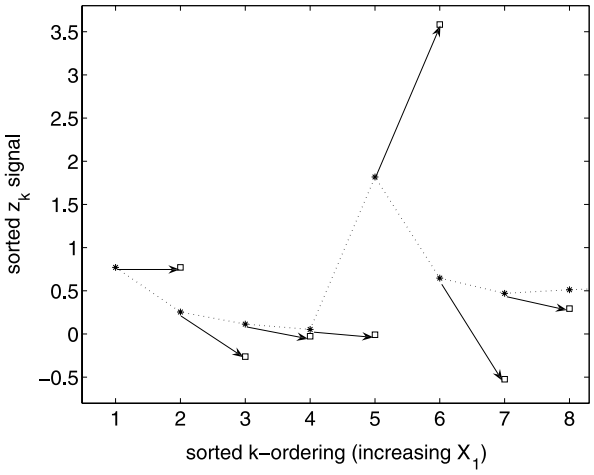

Fig. 2 Picture of the one step ahead predictions of the IRW process in the case of NVR $\to \infty$. The *stars/dotted line* denote the z_k data series, while the *squares* denote the one step ahead prediction $\hat{s}_{k|k-1}$

estimation, we find the NVR that maximizes the log-likelihood function L, where:

$$-2 \cdot \log(L) = \text{const} + \sum_{k=3}^{N} \log(1 + P_{k|k-1}) + (N-2) \cdot \log\left(\hat{\sigma^2}\right),$$

$$\hat{\sigma^2} = \frac{1}{N-2} \sum_{k=3}^{N} \frac{(z_k - \hat{s}_{k|k-1})^2}{(1 + P_{k|k-1})}, \tag{6}$$

where $\hat{\sigma^2}$ is the "weighted average" of the squared innovations (i.e., the prediction error of the IRW model), and $P_{k|k-1}$ is the one step ahead forecast error of the state $\hat{s}_{k|k-1}$ provided by the KF (both $P_{k|k-1}$ and $\hat{s}_{k|k-1}$ are functions of NVR). Since $\hat{s}_{k|k-1}$ is based only on the information contained in the sample values $[1, \ldots, k-1]$ (while smoothed estimates use the entire information set $[1, \ldots, N]$), it can be easily seen that the limit NVR $\to \infty$ ($\lambda \to 0$) is no longer a "perfect fit" situation, since a zero variance for e_k implies $\hat{s}_{k|k-1} = s_{k-1} + d_{k-1} = z_{k-1} + d_{k-1}$; i.e., the one step ahead prediction of z_k is given by the linear extrapolation of the adjacent value z_{k-1}, thus implying a nonzero prediction error in this limit case.

This is further exemplified in Fig. 2: the squares in the plots denote the one step ahead prediction $\hat{s}_{k|k-1}$ and the arrows show the linear extrapolation mechanism of the IRW process when NVR $\to \infty$. Such a prediction departs considerably not only from a "perfect fit" situation but also from a "decent fit," implying that the ML estimate will automatically penalize this kind of situation and provide the "right" value for the NVR (see also Wecker and Ansley 1983 for a discussion of this ML estimator in the smoothing spline context).

To complete the equivalence between the SDR and cubic spline formulations, we need to link the NVR estimated by the ML procedure to the smoothing parameters λ. It can be easily verified that by setting $\lambda = 1/(\text{NVR} \cdot N^4)$, and with evenly spaced X_k values, the $f(X_k)$ estimate in the cubic smoothing spline model equals the $\hat{s}_{k|N}$ estimate from the IRW process. The present results are also in line with the cited

work of Wecker and Ansley (1983), who assume, however, a different stochastic form for s_k, namely an ARIMA(0, 2, 2) process.

As mentioned in the Introduction, this is not the only possible equivalence between recursive algorithms and polynomial splines. For example, one can verify that assuming a random walk stochastic process for s_k with some ML optimized NVR(RW) is equivalent to estimating a linear spline with smoothing parameter $\lambda = 1/(\text{NVR(RW)} \cdot N^2)$ (see also Wecker and Ansley 1983).

The most interesting aspect of the SDR approach is that it is not limited to the univariate case, but can be effectively extended to the most relevant multivariate one. In the general additive case (1), for example, the recursive procedure needs to be applied, in turn, for each term $f_j(X_{j,k}) = \hat{s}_{j,k|N}$, requiring a different sorting strategy for each $\hat{s}_{j,k|N}$. Hence, the "backfitting" procedure, as described in Young (2000) and Young (2001), is exploited (see Appendix A.1). This procedure provides both ML estimates of all NVR_j's and the smoothed estimates of the additive terms $\hat{s}_{j,k|N}$. It can be easily verified that the equivalence between the λ's and NVRs presented for $p = 1$ also holds for the additive model with $p > 1$. So, the estimated NVR_j's can be converted into λ_j values using $\lambda_j = 1/(\text{NVR}_j \cdot N^4)$, allowing us to put the additive model into the standard cubic spline form.

2.2 ANOVA models with interaction functions

The additive model concept (1) can be generalized to include two-way (and higher) interaction functions via the functional ANOVA decomposition (Wahba 1990; Gu 2002). For example, we can let

$$f(\mathbf{X}) = f_0 + \sum_{j=1}^{p} f_j(X_j) + \sum_{j<i}^{p} f_{j,i}(X_j, X_i). \tag{7}$$

In the ANOVA smoothing spline context, corresponding optimization problems with interaction functions and their solutions can be obtained conveniently with the reproducing kernel Hilbert space (RKHS) approach (see Wahba 1990). In the SDR context, we propose here to formalize an interaction function as the product of two states $s_1 \cdot s_2$, each of them characterized by an IRW stochastic process. Hence, the estimation of a single interaction term $z^*(\mathbf{X}_k) = f(X_{1,k}, X_{2,k}) + e_k$ is expressed as

$$\begin{aligned} &\text{Observation Equation:} && z^*_k = s^I_{1,k} \cdot s^I_{2,k} + e_k \\ &\text{State Equations: } (j = 1, 2) && s^I_{j,k} = s^I_{j,k-1} + d^I_{j,k-1} \\ & && d^I_{j,k} = d^I_{j,k-1} + \eta^I_{j,k} \end{aligned} \tag{8}$$

where z^* is the model output after the main effects are taken out, $I = 1, 2$ is the multi-index denoting the interaction term under estimation, and $\eta^I_{j,k} \sim N(0, \sigma^2_{\eta^I_j})$. The two terms $s^I_{j,k}$ are estimated iteratively by running the recursive procedure in turn; i.e.,

– take an initial estimate of $s^I_{1,k}$ and $s^I_{2,k}$ by regressing z with the product of simple linear or quadratic polynomials $p_1(X_1) \cdot p_2(X_2)$ and set $s^{I,0}_{j,k} = p_j(X_{j,k})$;

– iterate $i = 1, 2$:
  – fix $s_{2,k}^{I,i-1}$ and estimate NVR_1^I and $s_{1,k}^{I,i}$ using the recursive procedure;
  – fix $s_{1,k}^{I,i}$ and estimate NVR_2^I and $s_{2,k}^{I,i}$ using the recursive procedure;
– the product $s_{1,k}^{I,2} \cdot s_{2,k}^{I,2}$ obtained after the second iteration provides the recursive SDR estimate of the interaction function.

The latter stopping criterion is a convenient choice to limit the computation time, and is due to the observation that the estimate of the interaction term never changed too much in any subsequent iteration.

Unfortunately, in the case of interaction functions we cannot derive an explicit and full equivalence between SDR and cubic splines of the type mentioned for first-order ANOVA terms. Therefore, in order to be able to exploit the estimation results in the context of a smoothing spline ANOVA model, we propose to take a different approach, similar to the ACOSSO case.

2.3 Very short summary of ACOSSO

We make the usual assumption that $f \in \mathcal{F}$, where $\mathcal{F}$ is an RKHS. The space $\mathcal{F}$ can be written as an orthogonal decomposition $\mathcal{F} = \{1\} \oplus \{\bigoplus_{j=1}^{q} \mathcal{F}_j\}$, where each $\mathcal{F}_j$ is itself an RKHS and $j = 1, \ldots, q$ spans ANOVA terms of various orders. Typically, q includes the main effects plus relevant interaction terms.

We reformulate (2) for the general case with interactions using the function f that minimizes

$$\frac{1}{N}\sum_{k=1}^{N}\big(z_k - f(\mathbf{X}_k)\big)^2 + \lambda_0 \sum_{j=1}^{q} \frac{1}{\theta_j} \big\| P^j f \big\|_{\mathcal{F}}^2, \tag{9}$$

where $P^j f$ is the orthogonal projection of f onto $\mathcal{F}_j$ and the q-dimensional vector θ_j of smoothing parameters needs to be optimized somehow. This is typically a formidable problem, and in the simplest case θ_j is set to one, with the single λ_0 estimated by GCV or GML.

Problem (9) also poses the issue of selection of $\mathcal{F}_j$ terms: this is tackled rather effectively within the COSSO/ACOSSO framework.

The COSSO (Lin and Zhang 2006) penalizes the sum of norms, using a Least Absolute Shrinkage and Selection Operator (LASSO)-type penalty (Tibshirani 1996) for the ANOVA model, which allows us to identify the informative predictor terms $\mathcal{F}_j$ with an estimate of f that minimizes

$$\frac{1}{N}\sum_{k=1}^{N}\big(z_k - f(\mathbf{X}_k)\big)^2 + \lambda \sum_{j=1}^{Q} \big\| P^j f \big\|_{\mathcal{F}} \tag{10}$$

using a single smoothing parameter λ, and where Q includes *all* ANOVA terms to be potentially included in f, e.g., with a truncation up to second- or third-order interactions.

It can be shown that the COSSO estimate is also the minimizer of

$$\frac{1}{N}\sum_{k=1}^{N}\big(z_k - f(\mathbf{X}_k)\big)^2 + \sum_{j=1}^{Q}\frac{1}{\theta_j}\big\|P^j f\big\|_{\mathcal{F}}^2 \tag{11}$$

subject to $\sum_{j=1}^{Q} 1/\theta_j < M$ (where there is a 1-1 mapping between M and λ). So we can think of the COSSO penalty as the traditional smoothing spline penalty plus a penalty on the Q smoothing parameters used for each component. The LASSO-type penalty has the effect of setting some of the functional components ($\mathcal{F}_j$'s) equal to zero (e.g., the variable X_j or the interaction (X_j, X_i) is not in the model); thus, it "automatically" selects the appropriate subset q of terms out of the Q "candidates." The key property of COSSO is that with one single smoothing parameter (λ or M) it provides proper estimates of all θ_j parameters: therefore, it considerably improves problem (9) with $\theta_j = 1$ (still with one single smoothing parameter λ_0) and is much more computationally efficient than the full problem (9) with optimized θ_j's.

In the adaptive COSSO (ACOSSO) of Storlie et al. (2011), $f \in \mathcal{F}$ minimizes

$$\frac{1}{N}\sum_{k=1}^{N}\big(z_k - f(\mathbf{X}_k)\big)^2 + \lambda\sum_{j=1}^{q} w_j\big\|P^j f\big\|_{\mathcal{F}}, \tag{12}$$

where $0 < w_j \leq \infty$ are weights that depend on an initial estimate of $\tilde{f}$, either using (9) with $\theta_j = 1$ or the COSSO estimate (10). The adaptive weights are obtained as $w_j = \|P^j\tilde{f}\|_{L_2}^{-\gamma}$, typically with $\gamma = 2$ and the L_2 norm $\|P^j\tilde{f}\|_{L_2} = (\int (P^j\tilde{f}(\mathbf{X}))^2 d\mathbf{X})^{1/2}$. The use of adaptive weights improves the predictive capability of ANOVA models with respect to the COSSO case.

2.4 Combining SDR and ACOSSO for interaction functions

There is an obvious way of exploiting the SDR identification and estimation steps in the ACOSSO framework: namely, the SDR estimates of additive and interaction function terms can be taken as the initial $\tilde{f}$ used to compute the weights in the ACOSSO. However, this would be a minimal approach, whereas the SDR identification and estimation provides more detailed information about f_j terms that is worth exploiting. We define $\mathcal{K}_{\langle j\rangle}$ to be the reproducing kernel (r.k.) of an additive term $\mathcal{F}_j$ of the ANOVA decomposition of the space $\mathcal{F}$. In the cubic spline case, this is constructed as the sum of two terms $\mathcal{K}_{\langle j\rangle} = \mathcal{K}_{01\langle j\rangle} \oplus \mathcal{K}_{1\langle j\rangle}$, where $\mathcal{K}_{01\langle j\rangle}$ is the r.k. of the parametric (linear) part and $\mathcal{K}_{1\langle j\rangle}$ is the r.k. of the purely nonparametric part. The second-order interaction terms are constructed as the tensor product of the first-order terms, for a total of four elements, i.e.,

$$\begin{aligned}\mathcal{K}_{\langle i,j\rangle} &= (\mathcal{K}_{01\langle i\rangle} \oplus \mathcal{K}_{1\langle i\rangle}) \otimes (\mathcal{K}_{01\langle j\rangle} \oplus \mathcal{K}_{1\langle j\rangle}) \\ &= (\mathcal{K}_{01\langle i\rangle} \otimes \mathcal{K}_{01\langle j\rangle}) \oplus (\mathcal{K}_{01\langle i\rangle} \otimes \mathcal{K}_{1\langle j\rangle}) \oplus (\mathcal{K}_{1\langle i\rangle} \otimes \mathcal{K}_{01\langle j\rangle}) \oplus (\mathcal{K}_{1\langle i\rangle} \otimes \mathcal{K}_{1\langle j\rangle}).\end{aligned} \tag{13}$$

In general, considering problem (9), one should attribute a specific coefficient $\theta_{\langle\cdot\rangle}$ to each single element of the r.k. of $\mathcal{F}_j$ (see, e.g., Gu 2002, Chap. 3), i.e. two θ's for each main effect, four θ's for each two-way interaction, and so on. In fact, each $\mathcal{F}_j$ would be optimally fitted by opportunely choosing weights in the sum of $\mathcal{K}_{\langle\cdot,\cdot\rangle}$ elements. This, however, makes the estimation problem rather complex, so, usually, the tensor product (13) is directly used, without tuning the weights of each element of the sum. This strategy is also applied in ACOSSO.

Instead, we propose to use SDR estimates of interaction to set the weights.

In particular, we can see that the SDR estimate of the interaction (8) is given by the product of two univariate cubic splines. So, one can easily decompose each estimated $\hat{s}_j^I$ into the sum of a linear ($\hat{s}_{01\langle j\rangle}^I$) and nonparametric term ($\hat{s}_{1\langle j\rangle}^I$). This provides a decomposition of the SDR interaction of the form

$$\hat{s}_i^I \cdot \hat{s}_j^I = \hat{s}_{01\langle i\rangle}^I \hat{s}_{01\langle j\rangle}^I + \hat{s}_{01\langle i\rangle}^I \hat{s}_{1\langle j\rangle}^I + \hat{s}_{1\langle i\rangle}^I \hat{s}_{01\langle j\rangle}^I + \hat{s}_{1\langle i\rangle}^I \hat{s}_{1\langle j\rangle}^I, \tag{14}$$

which can be thought of as a proxy of the four elements of the r.k. of the second-order tensor product cubic spline.

This suggests that a natural use of the SDR identification and estimation in the ACOSSO framework is to apply specific weights to each element of the r.k. $\mathcal{K}_{\langle\cdot,\cdot\rangle}$ in (13). In particular, the weights are the L_2 norms of each of the four elements estimated in (14). We will show in the examples that this choice leads to significant improvement in the accuracy of ANOVA models with respect to the original ACOSSO approach.

2.5 Kriging method: the DACE Matlab toolbox

DACE (Lophaven et al. 2002) is a Matlab toolbox used to construct kriging approximation models on the basis of data from computer experiments. Once we have this approximate model, we can use it as a surrogate model (emulator, meta-model).

We briefly highlight the main features of DACE.

Keeping wherever possible the same notation that we used previously, the kriging model can be expressed as a regression

$$\hat{z}(\mathbf{X}) = \beta_1 f_1(\mathbf{X}) + \cdots + \beta_q f_q(\mathbf{X}) + \zeta(\mathbf{X}), \tag{15}$$

where $f_i, i = 1, \ldots, q$ are deterministic regression terms, β_i are the related regression coefficients, and ζ is a zero-mean random process whose variance depends on the process variance ω^2 and on the correlation $\mathcal{R}(v, w)$ between $\zeta(v)$ and $\zeta(w)$. The toolbox provides a set of correlation functions defined as

$$\mathcal{R}(\theta, v, w) = \prod_{j=1:p} \mathcal{R}_j(\theta_j, w_j - v_j).$$

In particular, for the generalized exponential correlation function, used in the next section, one has

$$\mathcal{R}_j(\theta_j, w_j - v_j) = \exp\left(-\theta_j |w_j - v_j|^{\theta_{p+1}}\right).$$

Then, we can define R as the correlation matrix at the design points (i.e., $R_{i,j} = \mathcal{R}(\theta, \mathbf{X}_i, \mathbf{X}_j)$) and the matrix $r(\mathbf{X}) = [\mathcal{R}(\mathbf{X}_1, \mathbf{X}), \ldots, \mathcal{R}(\mathbf{X}_N, \mathbf{X})]$, $\mathbf{X}$ being an untried point. Similarly, we define $f = [f_1(\mathbf{X}) \cdots f_q(\mathbf{X})]'$ and $F = [f(\mathbf{X}_1) \cdots f(\mathbf{X}_N)]'$; i.e., F stacks in matrix form all values of f at the design points. Then, the regression problem $F\beta \approx \mathbf{Z}$ has a GLS solution given by

$$\beta^* = \left(F' R^{-1} F\right)^{-1} F' R^{-1} \mathbf{Z},$$

which gives the predictor at untried $\mathbf{X}$

$$\hat{z}(\mathbf{X}) = f(\mathbf{X})' \beta^* + r(\mathbf{X})' \gamma^*,$$

where γ^* is computed as $\gamma^* = R^{-1}(\mathbf{Z} - F\beta^*)$.

Of course, the proper estimation of the kriging emulator requires one to optimize the hyper-parameters θ in the correlation function: this is typically performed by maximum likelihood.

It is easy to check that the kriging predictor interpolates $\mathbf{X}_j$, if the latter is a design point. As far as regression models are concerned, the choice for f can be chosen from the following options:

Constant $q = 1,\ f_1 = 1$
Linear $q = p + 1,\ f_1 = 1,\ f_2 = X_1, \ldots, f_{p+1} = X_p$
Quadratic $q = \frac{1}{2}(p+1)(p+2),\ f_1 = 1,\ f_2 = X_1, \ldots, f_{p+1} = X_p,\ f_{p+2} = X_1^2,$ $f_{p+3} = X_1 X_2 \ldots$

It seems useful to underline that one major difference between DACE and ANOVA smoothing is the absence of any "observation error" in (15). This is a natural choice when analyzing computer experiments, and it aims to exploit the "zero-uncertainty" feature of this kind of data. This, in principle, makes the estimation of kriging emulators very efficient, as confirmed by the many successful applications described in the literature, and justifies the great success of this kind of emulator among practitioners. It also seems interesting to mention the "nugget" effect, which is also used in the kriging literature (see Wagner 2010 for an application of this in the present issue). This is nothing other than a "small" error term in (15), and it often reduces some numerical problems encountered in the estimation of the kriging emulator to the form of (15). The addition of a nugget term leads to kriging emulators that smooth, rather than interpolate, making them more similar to ANOVA models.

3 Examples

Storlie et al. (2008) performed an extensive analysis and comparison of metamodeling approaches for the estimation of total sensitivity indices. Their main conclusions were:

- simple models like quadratic regressions and additive smoothing splines can work very well, especially for small sample sizes;
- for larger sample sizes, more flexible approaches (MARS, ACOSSO, MLE GP in particular) can provide better estimations;

– GP does not outperform smoothing methods in estimating sensitivity indices.

The present paper does not substantially modify these results on sensitivity indices estimation, so we concentrate here on the forecast performance (out-of-sample R^2) of the different methods in predicting the function values at untried $\mathbf{X}$'s.

We compared the combined SDR-ACOSSO approach with ACOSSO and DACE on several examples (full details including Matlab routines are freely available on request):

- we checked the behavior of the SDR procedure proposed in Sect. 2.2 in identifying single two-way interaction functions;
- we performed full emulation exercises, considering multivariate analytic functions.

Concerning DACE, we always use the generalized exponential correlation function to estimate the emulator. Moreover, we include the nugget term in the single surface identification of Sect. 3.1, while the standard interpolating form of DACE is considered in the full emulation exercises of Sect. 3.2.

3.1 Single surface identification

First we checked the behavior of SDR in identifying single two-way interaction functions; i.e., we considered a number of surfaces $z(X_1, X_2) = g(X_1, X_2) + e$, with $e \sim N(0, \sigma^2)$, using different levels of signal-to-noise ratios $\text{SNR} = V(z)/V(e)$: very large ($\text{SNR} > 10$), medium ($\text{SNR} \sim 3$), very small ($\text{SNR} \sim 0.1$).

This kind of exercise is useful since it mimics the typical situation of the "backfitting" procedure adopted in the SDR method: in such a procedure each term of the ANOVA decomposition is identified and analyzed in turn, with the rest of the ANOVA terms acting as a sort of "noise." So $g(X_1, X_2)$ can be seen here as representing one single interaction term, to be identified among a number of ANOVA terms, represented by e. So, very large SNR represents the case of a predominant interaction term in the ANOVA model, and vice versa for very small SNR. We compared SDR results with standard GCV estimation and with DACE (extended to include observation noise/nugget) using a training MC sample $\mathbf{X}$ of 256 elements and tested the out-of sample performance of each method in predicting the "noise-free" signal $g(X_1, X_2)$ using a new validation sample $\mathbf{X}^*$ of dimension 256. We repeated this exercise on 100 random replicas for each function and each SNR.

We considered nine types of surfaces of increasing order of complexity (i.e., 27 different surface identifications, each replicated 100 times). The shapes of the surfaces are shown in Fig. 3, while their analytic expressions are shown in Table 1.

In Fig. 4 we can see that for only one out of the nine surfaces (f_9), DACE outperformed SDR or GCV estimation. This is probably to be expected since this is a pure smoothing problem, obviously more tailored for smoothing methods. In the other cases, SDR and GCV gave similar results, when the four terms in (14) have similar weights, while SDR was more efficient in identifying surfaces, where linear and nonparametric parts need to be better differentiated (f_6, f_7, f_9). These results demonstrate that the SDR identification step described in Sect. 2.2 is effective for identifying interaction functions.

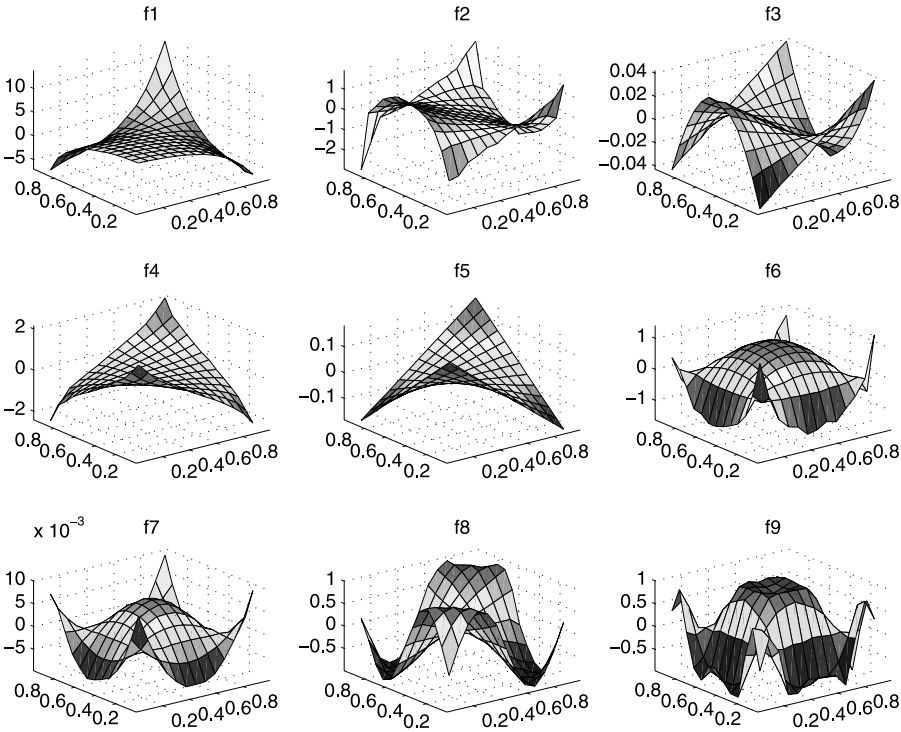

Fig. 3 Shape of the nine test surfaces considered for the test on identifying single interaction terms

Table 1 Analytic expressions for the test surfaces analyzed

Label	Expression	Support
f_1	$\prod_{i=1,2}(e^{(4X_1)} - (e^2 - e^{-2})/4)$	U[−0.5, 0.5]
f_2	$(X_2^2 - 1) \cdot X_1$	N(0, 1)
f_3	$(X_2^2 - 1/12) \cdot X_1$	U[−0.5, 0.5]
f_4	$X_1 \cdot X_2$	N(0, 1)
f_5	$X_1 \cdot X_2$	U[−0.5, 0.5]
f_6	$(X_1^2 - 1) \cdot (X_2^2 - 1)$	N(0, 1)
f_7	$(X_1^2 - 1/12) \cdot (X_2^2 - 1/12)$	U[−0.5, 0.5]
f_8	$\sin(6\pi X_1 \cdot X_2)$	U[−0.5, 0.5]
f_9	$\sin(100\pi (X_1^2 - 1/12) \cdot (X_2^2 - 1/12))$	U[−0.5, 0.5]

3.2 Full emulation exercises

Here we test the effectiveness of SDR in identifying full smoothing spline ANOVA models. We considered several test functions that have been used in the past to test sensitivity analysis and nonparametric regression methods. First, we used Sobol's G

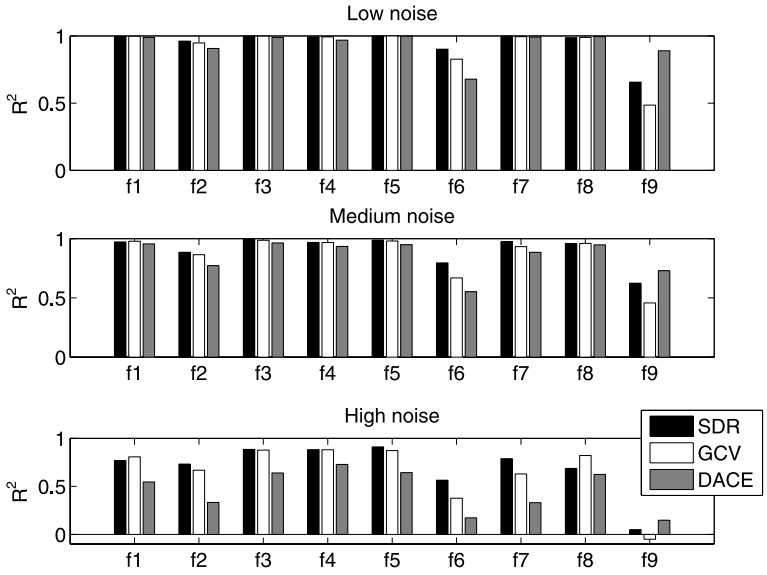

Fig. 4 Out-of-sample R^2 for the nine test surfaces of the three methods applied: SDR, ACOSSO, DACE

function (Archer et al. 1997), which can be used to generate test cases over a wide spectrum of difficulty and dimensionality p. It is defined as

$$G = G(X_1, X_2, \ldots, X_p, a_1, a_2, \ldots, a_p) = \prod_{i=1}^{p} \frac{|4X_i - 2| + a_i}{1 + a_i}. \tag{16}$$

The characteristics of the G-functions are driven both by the dimension (p) and by the spectrum of the coefficients a_i. Low values of a_i, such as $a_i = 0$, imply an important first-order effect. If more than one factor has low a_i's, then high interaction effects will also be present. The worst case for this function is where all a_i's are zero; i.e., all factors are equally important and all factors interact. If only a couple of a_i's are zero and all the others are large (e.g., $a_i \geq 9$), then we have a relatively easy test case, with just two important factors and a single two-way interaction term.

We considered the following four cases for the G-function, characterized by an increasing difficulty, due to the dimensionality of the problem (increasing p) or the degree of interactions (spectrum of a_i coefficients):

- $p = 4$, $a = [0, 1, 5, 99]$, labeled as "simple" in Table 2;
- $p = 4$, $a = [0, 0.01, 0.2, 0.5]$, labeled as "nasty" in Table 2;
- $p = 8$, $a = [0, 1, 4.5, 9, 99, 99, 99, 99]$, labeled as "simple" in Table 2;
- $p = 10$, $a = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$, labeled as "nasty" in Table 2.

We also use here a modified G-function, defined as

$$G^* = G^*\left(X_1, X_2, \ldots, X_p, a_1, a_2, \ldots, a_p, \delta_1, \delta_2, \ldots, \delta_p, \alpha_1, \alpha_2, \ldots, \alpha_p\right)$$
$$= \prod_{i=1}^{p} \frac{(1+\alpha_i)\cdot|2(X_i+\delta_i-[X_i+\delta_i])-1|^{\alpha_i}+a_i}{1+a_i}, \tag{17}$$

where $\delta_i \in [0, 1]$, $\alpha_i > 0$ are shift and curvature parameters, respectively, and $[X_i + \delta_i]$ is the integer part of $X_i + \delta_i$. If $\alpha_i = 1$ and $\delta_i = 0$, G^* degenerates to the G-function.

For this modified G^*-function, we analyzed the same cases listed previously for the standard G-function, but using $\alpha_i = 0.25$ and $\delta_i = 0$. These are essentially modified versions of the G-function examples, where we add more curvature in the model.

We also analyzed two test functions used in Storlie et al. (2011). In the first test function from Storlie et al. (2011), we have $\mathbf{X}$ uniform in $[0, 1]^{10}$ and the underlying function is nonadditive:

$$f_4(\mathbf{X}) = g_1(X_1) + g_2(X_2) + g_3(X_3) + g_4(X_4) + g_3(X_1X_2)$$
$$+ g_2\big((X_1 + X_3)/2\big) + g_1(X_3X_4) \tag{18}$$

and, therefore $X_5, \ldots, X_{10}$ are uninformative (dummy) variables.

In the second test function from Storlie et al. (2011), we have $\mathbf{X}$ uniform in $[0, 1]^{12}$ and the underlying regression function is additive:

$$f_{12}(\mathbf{X}) = g_1(X_1) + g_2(X_2) + g_3(X_3) + g_4(X_4)$$
$$+ 1.5\cdot g_1(X_5) + 1.5\cdot g_2(X_6) + 1.5\cdot g_3(X_7) + g_4(X_8)$$
$$+ 2\cdot g_1(X_9) + 2\cdot g_2(X_{10}) + 2\cdot g_3(X_{11}) + 2\cdot g_4(X_{12}). \tag{19}$$

The following four functions on $[0, 1]$ are used as building blocks in (18) and (19):

$$g_1(t) = t; \qquad g_2(t) = (2t-1)^2; \qquad g_3(t) = \frac{\sin(2\pi t)}{2-\sin(2\pi t)};$$
$$g_4(t) = 0.1\sin(2\pi t) + 0.2\cos(2\pi t) + 0.3\sin^2(2\pi t) + 0.4\cos^3(2\pi t)$$
$$+ 0.5\sin^3(2\pi t). \tag{20}$$

We considered training samples of growing dimension 64, 128, and 256 to estimate the emulators and used a new validation sample of the same dimension to check the out-of-sample performance. We repeated the analysis 100 times for each function and each method. We show detailed box-plots of the out-of-sample R^2 in Figs. 5 through 7, while Table 2 synthesizes the out-of-sample performance of the three methods, with stars highlighting the best performing method.

Considering small sample sizes ($N = 64$, Fig. 5), there is one test function (namely G^*, $p = 10$, "nasty") which is very difficult to predict. For that function, the only method which is able to give an out-of-sample fit is SDR-ACOSSO, while ACOSSO and especially DACE are not capable of providing any reasonable fit. For

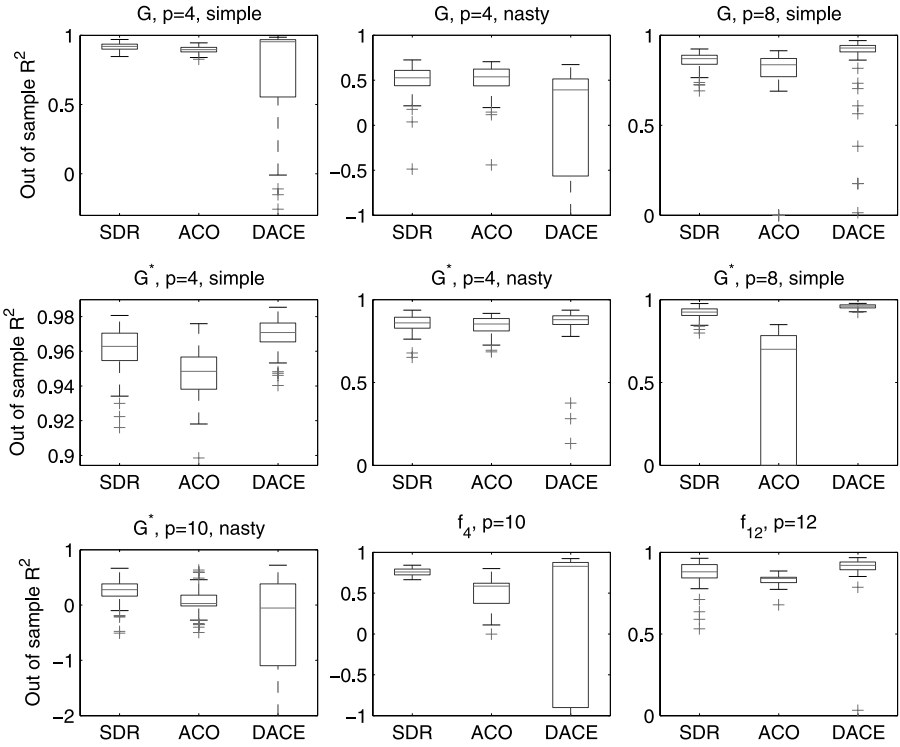

Fig. 5 Box-plots of the out-of-sample R^2 for the test models in Table 2, applying SDR, ACOSSO, DACE. Sample size is $N = 64$

all other test functions, both SDR-ACOSSO and DACE behave well, if the *median* of the R^2 distribution is only considered. However, for six of these test functions, the R^2 distribution of DACE presents "outliers" with very bad fit. This implies that, although the median of the R^2's is generally somewhat better for DACE, SDR-ACOSSO estimations are more robust and stable since bad outliers never occur. ACOSSO sometimes competes with SDR, in some other cases it performs a bit worse, but in one case (G^*, $p = 8$, "simple") it is clearly worse. Overall, we can say that, due to the occurrence of bad "outliers" for DACE, the SDR-ACOSSO seems overall the best method for small sample sizes.

Considering $N = 128$ in Fig. 6, we can see that DACE R^2 is affected by bad outliers for two test functions, making SDR preferable in terms of robustness, although in terms of median DACE is often better. ACOSSO almost always lags behind the other two methods. Two results are worth noting: (i) when DACE has the best performance, we can see that the SDR identification step is able to fill a large part of the gap between ACOSSO and DACE; (ii) for the additive f_{12} test function, SDR outperforms the other two methods (we recall that for additive models, SDR is not combined with ACOSSO, but directly provides the smoothing spline ANOVA model from the ML estimated NVRs).

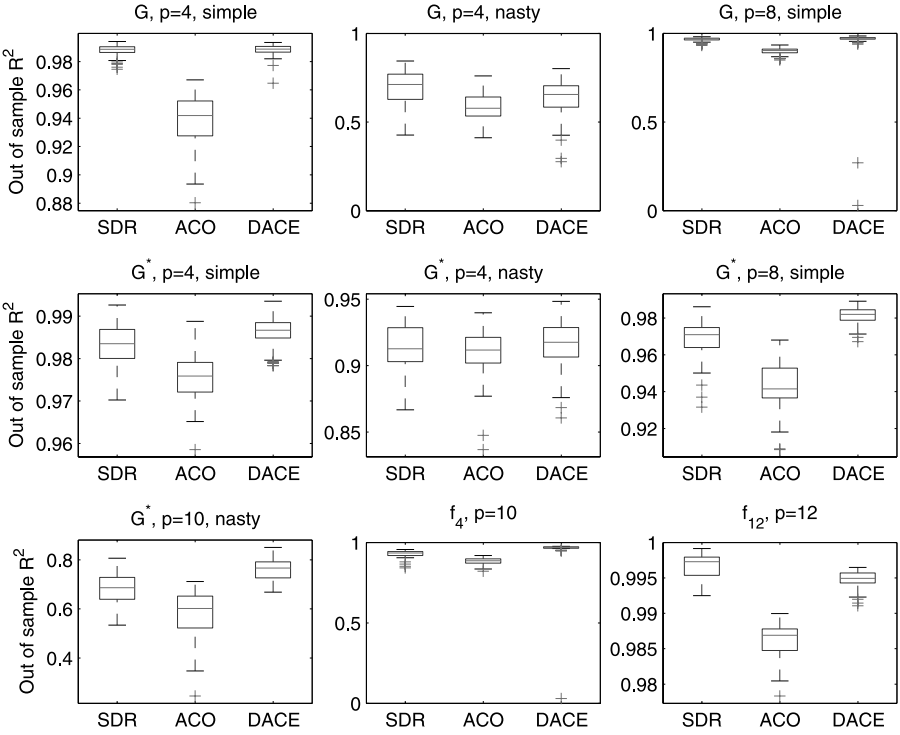

Fig. 6 Box-plots of the out-of-sample R^2 for the test models in Table 2, applying SDR, ACOSSO, DACE. Sample size is $N = 128$

Results for $N = 256$ in Fig. 7 are overall similar to those for $N = 128$, even if we can note an improvement of SDR-ACOSSO with respect to DACE. The latter, in fact, clearly outperforms the other methods only for the f_4 test function, while it still presents bad outliers for the G-function ($p = 8$, "simple"). Also, DACE is not able to provide any reasonable fit for the very difficult G-function ($p = 10$, "nasty"). SDR confirms its great efficiency in estimating the additive functions (see the f_{12} plot), with amazingly accurate predictions.

Overall, the results indicate that SDR identification provides a very significant added value in smoothing spline ANOVA models, allowing us to fill, in most cases, the gap between ACOSSO and DACE.

4 Discussion and conclusion

In general, it is not possible to identify a method (among ACOSSO, DACE, SDR-ACOSSO) which outperforms the others in all examples. SDR is extremely efficient and accurate in identifying *additive models*: the use of recursive algorithms avoids the inversion of large matrices and makes the computational cost of SDR linear both in p and N. This allows us to optimize as many as p smoothing parameters, at a

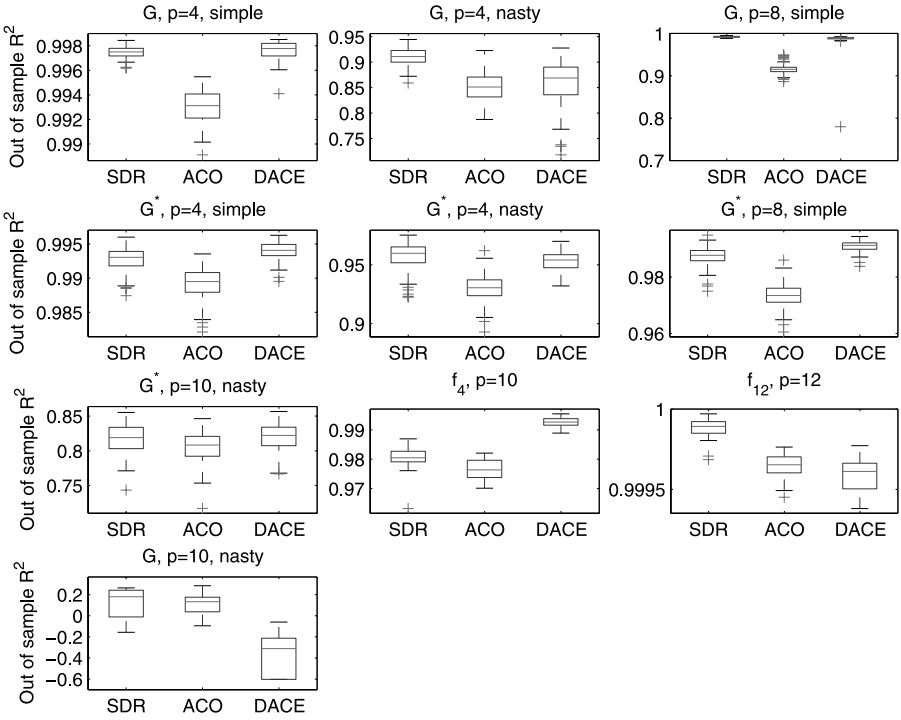

Fig. 7 Box-plots of the out-of-sample R^2 for the test models in Table 2, applying SDR, ACOSSO, DACE. Sample size is $N = 256$

similar cost of, e.g., ACOSSO in which one single smoothing parameter M is optimized. The computational cost of ACOSSO and DACE increases nonlinearly with p and N.

In the case of ANOVA models with interaction components, ACOSSO confirms entirely its good performances in terms of efficiency and low computational cost. When the model includes interactions, SDR combined with ACOSSO improves ACOSSO in many cases, although at the price of a higher computational cost. This is due to the SDR estimation of each single ANOVA term in the backfitting loop, prior to the final ACOSSO optimization of M. So, while for additive models the advantage of SDR is both low computational cost and accuracy, when interactions are included, the greater accuracy of SDR-ACOSSO has a cost, which does not exceed a few minutes in any cases (see Table 3). SDR-ACOSSO also compares very favorably with respect to DACE in many cases, even if there are cases where DACE outperforms SDR-ACOSSO in out-of-sample prediction. The main drawback of DACE seems to be the occurrence of very bad outliers in the distributions of R^2, implying some lack of robustness. The computational cost of DACE can be very sensitive to the underlying model. For each fixed dimension p and sample size N its variability can be of one order of magnitude (see Table 3). In terms of computational burden, we suggest that SDR (for additive models) and ACOSSO (for models with interactions) should be taken as the first choice for a *rapid and reliable* em-

Table 2 SDR-ACOSSO, ACOSSO, and DACE: average R^2 (out of sample) computed on 100 replicas for different types of test functions. Stars indicate the method with best out-of-sample performance

Model	Sample size	SDR-ACOSSO	ACOSSO	DACE
G, $p=4$ simple	64	0.9187	0.8960	0.5468 (0.9543[a])
G, $p=4$ simple	128	0.9881*	0.9392	0.9884*
G, $p=4$ simple	256	0.9973*	0.9926	0.9973*
G, $p=4$ nasty	64	0.5104	0.5159*	−0.5721 (0.3942[a])
G, $p=4$ nasty	128	0.6960*	0.5833	0.6008
G, $p=4$ nasty	256	0.9105*	0.8507	0.8609
G, $p=8$ simple	64	0.8603*	0.6906 (0.8367[a])	−0.7945 (0.9297[a])
G, $p=8$ simple	128	0.9679*	0.9008	0.9298 (0.9736[a])
G, $p=8$ simple	256	0.9924*	0.9164	0.9832
G, $p=10$ nasty	256	0.1922*	0.1963*	−0.0247
G^*, $p=4$ simple	64	0.9613	0.9476	0.9701*
G^*, $p=4$ simple	128	0.9831*	0.9759	0.9864*
G^*, $p=4$ simple	256	0.9928*	0.9892	0.9940*
G^*, $p=4$ nasty	64	0.8556*	0.8460	0.7354 (0.8804[a])
G^*, $p=4$ nasty	128	0.9142*	0.9103	0.9162*
G^*, $p=4$ nasty	256	0.9574*	0.9299	0.9534*
G^*, $p=8$ simple	64	0.9213	0.4393	0.9586*
G^*, $p=8$ simple	128	0.9689	0.9435	0.9813*
G^*, $p=8$ simple	256	0.9876	0.9736	0.9910*
G^*, $p=10$ nasty	64	0.2601*	0.0637	−21.4 (−0.0516[a])
G^*, $p=10$ nasty	128	0.6823	0.5845	0.7601*
G^*, $p=10$ nasty	256	0.8166*	0.8061	0.8205*
f_4, $p=10$	64	0.7621*	0.4815 (0.5859[a])	−6.763 (0.8303[a])
f_4, $p=10$	128	0.9241	0.8819	0.971*
f_4, $p=10$	256	0.9806	0.9764	0.9927*
f_{12}, $p=12$	64	0.8506 (0.8809[a])	0.8276 (0.8406[a])	0.8799* (0.9205[a])
f_{12}, $p=12$	128	0.9966*	0.9859	0.9947*
f_{12}, $p=12$	256	0.9999*	0.9996*	0.9996*

[a]We include the median of the R^2 sample here, due to the presence of "outliers" with very bad fit (see Figs. 5–7)

Table 3 Order of magnitude of computational costs (in seconds) of the three methods, at varying number of input factors p and sample size N. All three methods have been implemented in Matlab

Method	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$
$p = 4$ nonadditive (second-order ANOVA)					
SDR	8	16	32	100	350
ACOSSO	1	2	6	30	160
DACE	1	3	10	40	150
$p = 8$ nonadditive (second-order ANOVA)					
SDR	25	50	100	240	700
ACOSSO	2	3	9	50	240
DACE	5	20	40	160	650
$p = 10$ nonadditive (second-order ANOVA)					
SDR	38	70	140	–	–
ACOSSO	2	4	12	–	–
DACE	4	15	80	–	–
$p = 12$ additive (first-order ANOVA)					
SDR	4	8	16	–	–
ACOSSO	1	1	5	–	–
DACE	3	44	200	–	–

ulation exercise. Whenever ACOSSO is unable to explain a large part of the mapping, SDR-ACOSSO or DACE may be considered. We also noted that DACE is not necessarily the best choice when the model is supposed to be very complex with significant interactions. DACE, like any interpolation method, tries to exploit the "zero uncertainty" at observed samples of the mapping z. When the model is complex, this characteristic may lead one to wrongly identify spurious interaction terms involving unimportant X's, and therefore it may explain the occurrence of poorer performances in out-of-sample predictions with respect to smoothing methods. SDR-ACOSSO, on the other hand, can provide detailed information about the form of each additive and interaction term of a truncated ANOVA decomposition, often allowing very good out-of-sample predictions. Moreover, the component selection inherent in SDR-ACOSSO implies that the ANOVA model is only based on statistically significant terms, possibly enhancing the robustness of its out-of-sample performance. Finally, while we have seen that ACOSSO is overall less accurate than the other methods in out-of-sample predictions, we would like to stress that: (i) ACOSSO is the computationally cheapest method among the three considered for nonadditive models; and (ii) it provides a major methodological improvement to "classical" smoothing spline estimation (e.g., MARS), which is also a key characteristic of SDR-ACOSSO.

Appendix

A.1 The backfitting algorithm

We provide here a short summary of the backfitting algorithm, as described in Young (2000) and Young (2001). The basic reason why we need a backfitting procedure is to deal with the problems arising from the sorting strategy employed in the recursive algorithms. Once we have a *preliminary* estimate of the state variables, we define a "modified dependent variable" series obtained by subtracting from z_k all the other terms on the right-hand side of (1).

The backfitting algorithm takes the following form:

1. start from an initial estimate of states $\hat{s}^0_{i,k|N}$, $i = 1, \ldots, p$, $k = 1, \ldots, N$;
2. for backfitting iterations $b = 1, \ldots, B$
   (a) for $i = 1, \ldots, p$ define the modified dependent variable $\hat{z}^i_k = z_k - \sum_{j \neq i} \hat{s}^{b-1}_{j,k|N}$;
   (b) estimate NVR_i by the ML optimization;
   (c) get an updated estimate $\hat{s}^b_{i,k|N}$;
   (d) move to next b until no significant changes are detected in $\hat{s}^b_{i,k|N}$.
3. the final NVR_i's estimates are converted into the smoothing parameters λ_i's and the estimated model is finally cast in the standard smoothing spline ANOVA form.

Remark 2 It is not necessary to update the NVR_i estimates for all backfitting iterations $b = 1, \ldots, B$, but usually two backfitting iterations are sufficient for their effective estimates. This significantly speeds up the computational cost of subsequent backfitting iterations, that simply update, until convergence, $\hat{s}^b_{i,k|N}$. Usually $B < 10$.

Remark 3 Given the optimized NVRs, the backfitting algorithm provides estimates $\hat{s}^B_{i,k|N}$ that are equivalent to the smoothing spline estimate $f_i(X_{i,k})$ obtained with $\lambda_i = 1/(\text{NVR}_i N^4)$.

References

Archer, G., Saltelli, A., Sobol', I.: Sensitivity measures, ANOVA-like techniques and the use of bootstrap. J. Stat. Comput. Simul. **58**, 99–120 (1997)

Gu, C.: Smoothing Spline ANOVA Models. Springer, Berlin (2002)

Hodrick, T., Prescott, E.: Post-war US business cycles: an empirical investigation. Discussion paper n. 451, Northwestern University, Evanston, IL (1981)

Kalman, R.: A new approach to linear filtering and prediction problems. ASME Trans. J. Basic Eng. D **82**, 35–45 (1960)

Leser, C.: A simple method of trend construction. J. R. Stat. Soc. B **23**, 91–107 (1961)

Levy, S., Steinberg, D.: Computer experiments: A review. Adv. Stat. Anal. **94**(4), 311–324 (2010)

Lin, Y., Zhang, H.: Component selection and smoothing in smoothing spline analysis of variance models. Ann. Stat. **34**, 2272–2297 (2006)

Lophaven, S., Nielsen, H., Sondergaard, J.: DACE A MATLAB kriging toolbox, version 2.0. Technical Report IMM-TR-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark (2002). http://www.immm.dtu.dk/hbn/dace

Ng, C., Young, P.C.: Recursive estimation and forecasting of non-stationary time series. J. Forecast. **9**, 173–204 (1990)

Ratto, M., Pagano, A., Young, P.C.: State dependent parameter meta-modelling and sensitivity analysis. Comput. Phys. Commun. **177**, 863–876 (2007)

Schweppe, F.: Evaluation of likelihood functions for Gaussian signals. IEEE Trans. Inf. Theory **11**, 61–70 (1965)

Storlie, C., Bondell, H., Reich, B., Zhang, H.: Surface estimation, variable selection, and the nonparametric oracle property. Stat. Sin. (2011, in press). http://www3.stat.sinica.edu.tw/statistica/, preprint article SS-08-241

Storlie, C.B., Swiler, L., Helton, J.C., Sallaberry, C.: Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. SANDIA Report SAND2008-6570, Sandia Laboratories, Albuquerque, NM (2008)

Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B **58**(1), 267–288 (1996)

Wagner, T., Bröcker, C., Saba, N., Biermann, D., Matzenmiller, A., Steinhoff, K., Model of a thermomechanically coupled forming process based on functional outputs from a finite element analysis and from experimental measurements. Adv. Stat. Anal. **94**(4), 389–404 (2010)

Wahba, G.: Spline Models for Observational Data. CBMS-NSF Regional Conference Series in Applied Mathematics (1990)

Wecker, W.E., Ansley, C.F.: The signal extraction approach to non linear regression and spline smoothing. J. Am. Stat. Assoc. **78**, 81–89 (1983)

Weinert, H., Byrd, R., Sidhu, G.: A stochastic framework for recursive computation of spline functions: Part II, smoothing splines. J. Optim. Theory Appl. **30**, 255–268 (1980)

Young, P.C.: Nonstationary time series analysis and forecasting. Prog. Environ. Sci. **1**, 3–48 (1999)

Young, P.C.: Stochastic, dynamic modelling and signal processing: Time variable and state dependent parameter estimation. In: Fitzgerald, W.J., Walden, A., Smith, R., Young, P.C. (eds.) Nonlinear and Nonstationary Signal Processing, pp. 74–114. Cambridge University Press, Cambridge (2000)

Young, P.C.: The identification and estimation of nonlinear stochastic systems. In: Mees, AIea (ed.) Nonlinear Dynamics and Statistics. Birkhäuser, Boston (2001)

Young, P.C., Ng, C.N.: Variance intervention. J. Forecast. **8**, 399–416 (1989)

Young, P.C., Pedregal, D.J.: Recursive and en-bloc approaches to signal extraction. J. Appl. Stat. **26**, 103–128 (1999)