Dolomites Research Notes on Approximation
Proceedings of DWCAA12, Volume 6 · 2013 · Pages 130–139

Smoothing Spline ANOVA for Variable Screening

Laura Ricco[a] · Enrico Rigoni[a] · Alessandro Turco[a]

Abstract

Smoothing Spline ANOVA is a statistical modeling algorithm based on a function decomposition similar to the classical analysis of variance (ANOVA) decomposition and the associated notions of main effect and interaction. It represents a suitable screening technique for detecting important variables (Variable Screening) in a given dataset. We present the mathematical background together with possible industrial applications.

1 Introduction

Smoothing Spline ANOVA (SS-ANOVA) models are a family of smoothing methods suitable for both univariate and multivariate modeling/regression problems.

In this context *smoothing* means nonparametric function estimation in presence of stochastic data. Indeed SS-ANOVA belongs to the family of nonparametric or semiparametric models (more precisely it belongs to smoothing methods), which presents some peculiarities that distinguish them from the classical set of standard parametric models (polynomial models, etc.).

In fact in classical parametric regression analysis, the model has a given fixed form, known up to the parameters (the degrees of freedom of the model), which are estimated from the data: the model is said to have a set of rigid constraints on its functional form. Usually the number of unknown parameters (the dimension of the model space) is much smaller than the sample size. Data are affected by noise but can be considered as unbiased, while parametric models (viewed as a set of constraints) help in reducing noise but are responsible for the (possible) introduction of biases in the analysis (e.g. a possible model misspecification represents a typical bias).

On the contrary, nonparametric or semiparametric methods let the model vary in a high-dimensional (possibly infinite dimensional) function space. In this case so-called soft constraints are introduced, instead of the rigid constraints of parametric models: this leads to a more flexible function estimation. We will soon see how smoothing methods – a subset of such nonparametric models – can be derived by means of the so-called general penalized likelihood method. Smoothing methods are suited for regression with noisy data, given the assumption of Gaussian-type responses.

SS-ANOVA is a statistical modeling algorithm based on a function decomposition similar to the classical analysis of variance (ANOVA) decomposition and the associated notions of main effect and interaction (in fact higher-order interactions are typically excluded from the analysis, as we will see later). For this reason it presents an important fringe benefit over standard parametric models: the interpretability of the results. In fact each term – main effects and interactions – reveals an interesting measure: the percentage of its contribution to the global variance. That is in a statistical model the global variance is explained (decomposed) into single model terms.

For this reason SS-ANOVA represents a suitable screening technique for detecting important variables (Variable Screening) in a given dataset. Indeed, the industrial applications we are going to present rely on the implementation of this technique within the software framework modeFRONTIER [6], where SS-ANOVA is actually employed as Screening Analysis Tool for improving Response Surface Methodology (RSM) functions and it will soon exploited also for multi-objective optimization tasks. The mathematical formulation of the technique and some benchmark applications are included also in the documentation of the software [8, 9].

There exists a vast body of literature on Smoothing Spline ANOVA models: for example refer to [3, 11] for a detailed and complete treatment of the subject. In the following only the major theoretical and computational features will be provided: the interested reader could undoubtedly take advantage of these precious sources of information for a deeper technical analysis on this topic.

2 Mathematical Description

In the next subsections a general overview of SS-ANOVA models is given. Firstly the penalized likelihood method is introduced, then its application to a simple univariate (one-dimensional) case is shown: this leads to the formulation of cubic smoothing splines as smoothing methods. Subsequently the general multivariate case is shown: firstly the simple additive models are outlined, then the introduction of interaction effects leads to the general formulation of SS-ANOVA models.

[a] ESTECO s.p.a. - Area Science Park - Trieste

2.1 General penalized likelihood method

For the moment we will consider a univariate regression problem for the function $f(x) : [0,1] \subset \mathbb{R} \to \mathbb{R}$,

$$f_i = f(x_i) + \epsilon_i\,, \quad i = 1,\dots,n\,, \tag{1}$$

where n is the number of designs (sample set size), and $\epsilon_i \sim N(0,\sigma^2)$ represent independent random errors (normally distributed with mean 0 and variance σ^2 – Gaussian-type response). For convenience the input variable x is limited in the interval $[0,1]$, without losing generality.

In seeking a proper regression model $f(x)$ for the given stochastic data set, these two functionals can be taken into account:

$L(f)$ defined as minus log likelihood of the model $f(x)$ given the data (we will soon see its implementation as the usual least square function), that estimates the goodness of the fit (or better, the lack of fit),

$J(f)$ defined as a quadratic roughness functional, that measures in a proper way the roughness/smoothness of the model $f(x)$.

Therefore a suitable solution to the regression problem could be stated as a constrained minimization problem, in these terms

$$\min\ L(f)\,, \quad \text{subject to} \quad J(f) \le \rho\,, \tag{2}$$

where the minimization of L guarantees a good fit to the data, but the soft (i.e., inequality) constraint on J – limiting the admissible roughness – prevents overfitting.

By applying the Lagrange method, eq. (2) can be transformed in the following minimization problem:

$$\min\ L(f) + \frac{\lambda}{2} J(f)\,, \tag{3}$$

where λ is the Lagrange multiplier. By changing the value of the parameter λ, a whole family of different solutions f_λ to the regression problem can be obtained. The value of λ controls the trade-off in the resulting model between smoothness and fidelity to the data: the larger λ, the smoother the model, while smaller values imply rougher functions but with better agreement to the data.

The procedure of finding a proper function estimation by minimizing eq. (3) is called general penalized likelihood method: in fact the J term represents a penalty on the roughness. This methodology is also referred to as penalty smoothing (or smoothing method with roughness penalty), and λ takes the name of smoothing parameter.

A key issue for the success of such methodologies is the proper – and operatively practicable – selection of the value to be assigned to the smoothing parameter.

As a final note, we would like just to mention that the functionals L and J represent the two basic components of a statistical model: the deterministic part J, and the stochastic part L.

2.2 Univariate Cubic Smoothing Spline

The general method outlined in the previous subsection can be specialized as the minimization of the following penalized least square score:

$$\min\ \frac{1}{n}\sum_{i=1}^{n}(f_i - f(x_i))^2 + \lambda \int_0^1 [f''(x)]^2 dx\,, \tag{4}$$

where $f''(x) = d^2 f/dx^2$. With Gaussian-type responses, as in the case of eq. (1), the likelihood $L(f)$ takes the usual form of least square function (so we deal with a penalized least square problem): this term discourages the lack of fit of the model to the data. The penalty term is an overall measure of the magnitude of the curvature (roughness) of the function over the domain. Clearly we have assumed the existence of second-order derivative of f as well as of square integrable second-order derivative.

The minimizer f_λ of eq. (4) – the solution of the regression problem – is called the Cubic Smoothing Spline: in fact it can be demonstrated that it corresponds to a classical cubic spline (i.e., a piecewise cubic polynomial, with the knots at all of the distinct values of the sampling set $\{x_i\}$).

As $\lambda \to 0$, f_λ tends to the minimum curvature interpolant. On the contrary, as $\lambda \to +\infty$, f_λ becomes the simple linear regression, since linear polynomials form the null space of the roughness penalty functional.

2.3 Simple multivariate case: additive model

From now on we will consider a general multivariate regression problem for the function $f(\mathbf{x}) : [0,1]^N \subset \mathbb{R}^N \to \mathbb{R}$,

$$f_i = f(\mathbf{x}_i) + \epsilon_i\,, \quad i = 1,\dots,n\,, \tag{5}$$

where N is the number of input variables, and again we are in presence of Gaussian-type errors ϵ_i.

The simplest extension of smoothing splines to multivariate case is represented by the additive model,

$$f(\mathbf{x}) = \sum_{j=1}^{N} f_j(x_j)\,, \tag{6}$$

i.e., f is expressed as the sum of N independent components, each of which is function of one relevant variable. This means that only the main effects are taken into account.

The additive smoothing spline is then the minimizer of the expression

$$\min \frac{1}{n}\sum_{i=1}^{n}(f_i - f(\mathbf{x}_i))^2 + \sum_{j=1}^{N}\lambda_j \int_0^1 [f_j''(x_j)]^2 dx\,, \tag{7}$$

and in this case each of the components f_j takes the form of a natural cubic spline. Now there are N tuning parameters λ_j, one for each term.

2.4 General multivariate case

In general the minus log likelihood functional $L(f)$ – in presence of Gaussian-type response as that of eq. (5) – reduces to the usual least square functional. So eq. (3) – in general multivariate case – can be expressed as

$$\min \frac{1}{n}\sum_{i=1}^{n}(f_i - f(\mathbf{x}_i))^2 + \lambda J(f)\,, \tag{8}$$

An interesting fact is that the ANOVA decomposition can be built into the above penalized least square estimation through the proper construction of the roughness functional $J(f)$. The theory behind this formulation is based on the so-called reproducing kernel Hilbert space[1].

In seeking the solution of the penalized least square minimization problem, only smooth functions are taken into account: more specifically the solution lives in the reproducing kernel Hilbert space $\mathcal{H} \subseteq \{f : J(f) < \infty\}$.

$\mathcal{H}$ possesses the following tensor sum decomposition:

$$\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J = \oplus_{k=0}^{p}\mathcal{H}_k \tag{9}$$

where $\mathcal{N}_J = \mathcal{H}_0 = \{f : J(f) = 0\}$ is the null space of the roughness penalty functional $J(f)$. Let $\{\Phi_j\}_{j=1}^{m}$ be a basis of $\mathcal{N}_J$, where m is the dimension of $\mathcal{N}_J$ (Φ_j will be used in the following). The number of tensor sum terms p is determined only by the number and type (i.e., main/interaction) of effect terms introduced in the ANOVA decomposition model.

The penalty term then can be expressed as

$$\lambda J(f) = \lambda \sum_{k=1}^{p}\theta_k^{-1}\cdot(f,f)_k\,, \tag{10}$$

where $(f,g)_k$ is the inner product in $\mathcal{H}_k$, and let the function $R_k(x,y)$ be the relevant reproducing kernel (R_k will be used in the following).

λ and θ_k are the smoothing parameters. Their values are determined by a proper data-driven procedure: a suitable method is represented by the generalized cross validation (GCV), as described in [2, 4, 5].

Generally the computation of smoothing splines is of the order $O(n^3)$, where n is the sample size. In our implementation we adopted the procedure devised by [5], obtaining a more scalable computation of smoothing spline regression by means of certain low-dimensional approximations that are asymptotically as efficient as the standard procedure: their computation is of the order $O(nq^2)$, where $q \le n$. This result is achieved by selecting a suitable subset of the original sampling data: $\{\mathbf{z}_j\}_{j=1,\dots,q} \subseteq \{\mathbf{x}_i\}_{i=1,\dots,n}$.

In this context, the bilinear form $J(f,g) = \sum_{k=1}^{p}\theta_k^{-1}(f,g)_k$ is an inner product in $\mathcal{H}_J$, with a reproducing kernel $R_J(x,y) = \sum_{k=1}^{p}\theta_k R_k(x,y)$.

It can then be demonstrated that the minimizer f_λ of eq. (8) has this representation

$$f(\mathbf{x}) = \sum_{j=1}^{m} d_j \Phi_j(\mathbf{x}) + \sum_{j=1}^{q} c_j R_J(\mathbf{x},\mathbf{z}_j)\,, \tag{11}$$

where d_j and c_j are the unknown parameters to be determined by solving the penalized least square minimization problem. The estimation reduces to solving the following linear system expressed in matrix form

$$\left(\begin{array}{c|c} S^TS & S^TR \\ \hline R^TS & R^TR+Q \end{array}\right)\cdot\left(\begin{array}{c}\mathbf{d}\\ \hline \mathbf{c}\end{array}\right) = \left(\begin{array}{c} S^T\mathbf{f} \\ \hline R^T\mathbf{f}\end{array}\right), \tag{12}$$

which has dimension $(m+q)\times(m+q)$. S is a $n\times m$ matrix with entries $S_{ij} = \Phi_j(x_i)$, R is a $n\times q$ matrix with entries $R_{ij} = R_J(x_i,z_j)$, and Q is a $q\times q$ matrix with entries $Q_{ij} = \lambda R_J(z_i,z_j)$. $\mathbf{f}$ is the vector holding the n observations of eq. (5).

3 Screening Analysis

A key feature of Smoothing Spline ANOVA is the interpretability of the results. In fact – after the model has been trained over the given sampling data – different heuristic diagnostics are available for assessing the model quality and the significance of terms in the built-in multiple-term ANOVA decomposition. Indeed statistical modeling has two phases: model fitting and model checking. In the next subsections we provide all the information relevant to the use of SS-ANOVA as a Screening Analysis Tool.

[1] A complete treatment of the subject is far beyond the scope of the present work. For any detail, please refer e.g., to [3]

3.1 Curse of dimensionality

Usually in SS-ANOVA decomposition – compliantly with standard ANOVA models – only main effects and interaction effects are taken into account. In fact higher-order interactions are typically excluded from the analysis, mainly due to practical reasons: for limiting the model complexity, for improving the model interpretability, for reducing the required computational effort.

One related, practical, aspect that should be always taken into account when dealing with multivariate – high-dimensional – space is the so-called *curse of dimensionality*. This nasty reality affects all different aspects of multivariate analysis, and unfortunately it is an unavoidable evil. Its typical symptomatology involves the effect of sparsity of the space. This means that when the dimensionality (number of input variables) increases, the volume of the space increases so fast that the available data become sparse: in order to obtain a statistically significant and reliable result, the amount of needed sampling data grows exponentially with the dimensionality.

A major consequence of the curse of dimensionality on SS-ANOVA models is the explosive increase in the number of parameters (degrees of freedom) that would be required by the introduction in the model of higher-order effects in a high-dimensional space. So considering only main effects in building the model (i.e., additive model) – or possibly adding at most only interaction effects – helps in tackling the curse of dimensionality. If N is the number of input variables, the number of main effect terms is clearly equal to N, while the number of interaction effects is equal to $N(N-1)/2$, so the growth rate of second order models goes as $O(N^2)$

For this reason, in practical data analysis in a high-dimensional space, usually only the main effects are included. Interaction effects are taken into account only if the relevant computational demand is affordable.

3.2 Model checking

The function ANOVA decomposition of the trained model, evaluated at sampling points, can be written as

$$\mathbf{f}^* = \sum_{k=1}^{p} \mathbf{f}_k^* , \tag{13}$$

where $\mathbf{f}^*$ and $\mathbf{f}_k^*$ are column vectors of length n. The star symbol (*) denotes the fact that we got rid of the constant term[2].

The relative significance of the different terms composing the model can be assessed by means of the *contribution indices* π_k, defined as

$$\pi_k = \frac{(\mathbf{f}_k^*, \mathbf{f}^*)}{||\mathbf{f}^*||^2} , \tag{14}$$

where (u, v) denotes the usual scalar product of vectors, i.e., $(u, v) = u^T \cdot v$, and $||u|| = \sqrt{(u, u)}$ is the vector norm. In fact, applying the scalar product between $\mathbf{f}^*$ and the two terms of eq. (13) respectively – making use of eq. (14) – the following identity is obtained:

$$\sum_{k=1}^{p} \pi_k = 1 , \tag{15}$$

that can be interpreted as a decomposition of unity, although π_k can be negative. It follows that – in case $\mathbf{f}_k^*$ are nearly orthogonal to each other (see below) – the π_k can be interpreted as the "percentage decomposition of explained variance into model terms". In this sense each term – main effects and interactions – revealing the percentage of its contribution to the global variance, can be evaluated in terms of importance: higher values of π_k mean important terms that have to be kept, lower values mean negligible terms that can be dropped.

A useful diagnostics tool for assessing the model quality is represented by the *collinearity indices* κ_k. Defining the $p \times p$ matrix C as[3]

$$C_{ij} = \frac{(\mathbf{f}_i^*, \mathbf{f}_j^*)}{||\mathbf{f}_i^*|| \cdot ||\mathbf{f}_j^*||} \tag{16}$$

κ_k^2 can be identified as the diagonal elements of C^{-1}, so that

$$\kappa_k = \sqrt{C_{kk}^{-1}} . \tag{17}$$

Each index $\kappa_k \geq 1$: the equality $\kappa_j = 1$ holds if and only if a given $\mathbf{f}_j^*$ is orthogonal to all the other $\mathbf{f}_{k \neq j}^*$. So the ideal situation of all $\kappa_k \simeq 1$ holds only in case all the $\mathbf{f}_k^*$ are nearly orthogonal to each other.

In case two or more $\mathbf{f}_k^*$ are highly (linearly) correlated – a phenomenon referred to as *collinearity*[4] – can be easily detected since the relevant collinearity indices will be much greater than one ($\kappa_k \gg 1$). This unfortunate situation occurs when the chosen model decomposition is inadequately supported on the sampling points domain (that is the decomposition is bad defined on its domain, since its terms cannot be identified unambiguously). There can be many causes to this pathology: dependent input variables (explanatory variables of a model should be always be independent), bad sampling points (e.g., set of correlated designs, inappropriate DOE selection), too low sample size. As a severe consequence the values of the relevant π_k indices become unreliable.

[2] Technically, the full ANOVA decomposition has been projected on the space $\{\mathbf{1}\}^\perp = \{\mathbf{f} : \mathbf{f}^T \mathbf{1} = 0\}$ where the vector $\mathbf{1}$ is a vector of all ones generating the constant term space $\{\mathbf{1}\}$. Practically this operation is performed by centering the column vectors of the full decomposition, i.e., by subtracting their relevant element means.

[3] That is C_{ij} represents the cosine of the angle between the vectors $\mathbf{f}_i^*$ and $\mathbf{f}_j^*$: C is referred to as the cosines matrix.

[4] This situation can also be designated as *concurvity* or as *identifiability problem*.

4 Variable Screening Benchmarks

The performance of the screening method is tested on four different problems. It depends on the number of designs in the database, on the number of input variables and on the collinearity among inputs. For these reasons, the analysis for each problem is repeated using 500, 1000, 2000, 5000 designs, generated with Random DOE: a measure of their collinearity is express with *collinearity indices*. In section 4.4, we choose a scalable function in order to perform the same test for different number of variables as well as of designs. In particular, we consider 10, 20, 50 and 100 input variables.

The benchmark is performed on a computer mounting an Dual Core, 2.40 GHz, with 15 GB of RAM. The OS is Ubuntu 10.10 64 bit using modeFRONTIER 4.4.1.

For each test, we report the screening analysis execution time, the contribution indices and the collinearity indices. Since we know the explicit explicit model of all tests, we can summarize the performance in each test.

4.1 Problem 1

The first test is proposed by C. Gu in [3]. It involves three input variables between $[0,1]$.

$$x_1\ ,\ x_2\ ,\ x_3\ \in [0,1]$$

$$y = 10\sin(\pi x_2) + e^{3x_3} + 5\cos(2\pi(x_1 - x_2)) + 3x_1 \tag{18}$$

In Table 1 we report the results of the screening analysis: x_3 is the less important variable while x_1 and x_2 can be considered important. While the analysis with only "Main Effect" shows that x_2 could appear not important, the interaction effect analysis gives more informations about the single variables. In fact, the term $x_1 * x_2$ is important in the model, as it is possible to confirm by analysing the explicit formula of Eq. (18).

4.2 Problem 2 - Polynomial case

This test involves six variables, but only three of them give a contribution to the output (relative to indices $I = \{1,2,3\}$), as we can see in the following formula

$$x_1\ ,\ x_2\ ,\ x_3\ ,\ x_4\ ,\ x_5\ ,\ x_6 \in [0,1]$$

$$y = k_0 + \sum_{j\in I} k_j x_j + \sum_{i,j\in I} k_{ij} x_i x_j + \sum_{i,j,l\in I} k_{ijl} x_i x_j x_l$$

where:

$$\begin{array}{lllllllll}
k_0 & = & 0.5025 & k_1 & = & -0.4898 & k_2 & = & 0.0119 \\
k_3 & = & 0.3982 & k_{11} & = & 0.07818 & k_{12} & = & 0.4593 \\
k_{22} & = & 0.00944 & k_{13} & = & -0.3614 & k_{23} & = & -0.3507 \\
k_{33} & = & -0.04850 & k_{111} & = & 0.006814 & k_{112} & = & -0.04914 \\
k_{122} & = & 0.06286 & k_{222} & = & -0.005130 & k_{113} & = & 0.08585 \\
k_{123} & = & -0.03000 & k_{223} & = & -0.06068 & k_{133} & = & -0.04978 \\
k_{233} & = & 0.02321 & k_{333} & = & -0.000534 & & &
\end{array}$$

Thanks to this test, we notice different interesting behaviors. First of all, it seems that the results for the sensitivity indices do not depends on the number of designs used to perform the test. As expected, these results depend on the variable collinearity: indeed in such cases, the collinearity index is close to the value 1, which means that the input variables are not collinear.

We also notice that x_2 appears to be more important than x_3, since x_2 has bigger interaction effects.

Although in this test the interaction analysis provides relevant information about our model, we can consider the main effect analysis as a first study to understand the general behavior of variables. If the number of variables is too large to perform an interaction effect analysis, the main effect values are sufficient to determine a smaller set of input variables.

4.3 Problem 3 - Polynomial case

For this test, we consider the same model as Problem 2, but we change two coefficients. In particular:

$$k_{12} = 0.9 \quad k_{123} = 0.9$$

In this way, the importance of x2 increases, as also confirmed by the screening analysis. Also the main effect of x_3 increases, thanks to the high value of k_{123}.

4.4 Problem 4 - Uncontrained Problem 1 Cec 2009

The last test proposed is the first function of the Unconstrained Problem 1 reported in [12]. Thanks to its formulation independent of the number of variables, it has been possible to do the analysis with 10, 20, 50, 100 input variables.

$$x_0 \in [0,1], \quad x_i \in [-1,1]^n \quad i = 1,\dots,n$$
$$f_1 = x_0 + \frac{2}{|J|}\sum_{j\in J}\left(x_j - \sin\left(6\pi x_0 + \frac{j\pi}{n}\right)\right)^2$$

where:

$$J = \left\{j | j \text{ is odd and } 2 \le j \le n\right\}$$

Table 4 reports the time needed to do the screening and Table 5 summarizes the variable collinearity. The symbol (–) in the two Tables 4 and 5 represents unfinished analysis: this happens when the number of designs is less than the number of terms.

When at least one collinearity index is sensibly greater than 1, the main or interaction effects could be greater than 1. This means that the screening analysis does not represent the percentage decomposition of variance and the results might be far from reality. In this case there are several possible strategies. When these problems are due to the interaction effect analysis, a possible strategy is to divide the input variables by following the results of the main effect analysis and perform another sensitivity analysis. Also an increment of the number of designs should overcome the collinearity among terms. If all of these strategies are not possible, a preliminary screening of variables could be done by removing the variables which have the higher values of collinearity index.

5 Metamodeling

A metamodel (or response surface model, RSM) is a surrogate of a model, which has to maximizing the fidelity with the original one, while minimizing the computational effort needed to evaluate it. Smoothing Spline ANOVA is a metamodel itself, since it provides a piece-wise polynomial regression over a database generated using a "first order" model.

However, SS-ANOVA reveals its best utility when coupled with a different metamodeling technique. Indeed, the curse of dimensionality prevents the use of high order terms in the SS-ANOVA training, but thanks to the reliable variable number reduction obtained even with low order spline models, it is possible to train more sophisticated RSM reducing the effort and increasing the accuracy.

We will present the results of a severe benchmarking activity in a future work, but we can use the previously described Problem 2 as a toy model to explain a phenomenon observed in different industrial databases. If all the coefficients of the polynomial relative to x4, x5 or x6 are set to zero, the accuracy of any metamodelling technique depends strictly on its ability to recognize that only 3 out of 6 input variables are involved in the output.

Moreover, also the computational cost of the training phase depends on the number of variables considered, with often an exponential rate of growth. On the contrary, as shown in the previous benchmarks, the screening training is not so demanding. This is the reason why the implementation of SS-ANOVA in modeFRONTIER is proposed as a pre-RSM training tool, with several shortcuts to select the desired input variables depending on the analysis results.

6 Multi-Objective Optimization

The need for a screening variable technique is pressing when dealing with multi-objective optimization in industrial contexts. From one hand the already cited *curse of dimensionality* weakens also the most robust algorithms. On the other hand, the empirical but effective Pareto principle (the 80% of the effects is produced by the 20% of the causes) opens the door to great performance boosts.

Before describing a possible use of SS-ANOVA in this field, we recall some definitions. A multi-objective problem can be stated as:

$$\begin{cases} \min f_i(\mathbf{x}) & \text{for } i = 1\dots n, \\ g_j(\mathbf{x}) \le 0 & \text{for } j = 1\dots m_i, \\ h_k(\mathbf{x}) = 0 & \text{for } k = 1\dots m_e, \end{cases} \tag{19}$$

where $\mathbf{x}$ is an arbitrary large vector spanning an admissible configuration set called A; n is bigger than 1 and m_i, m_e express the number of constraints of the problem (which can also be zero).

The solution of the problem is the set P of optimal trade-off solution among all the objectives, called *Pareto Set*. It can be defined as the set of *non-dominated* points of A, where the Pareto dominance is a partial ordering defined as:

$$\mathbf{x} \text{ dominates } \mathbf{y} \Leftrightarrow \begin{cases} \forall i = 1\dots n & f_i(\mathbf{x}) \le f_i(\mathbf{y}) \text{ and} \\ \exists j \text{ such that} & f_j(\mathbf{x}) < f_j(\mathbf{y}). \end{cases} \tag{20}$$

6.1 Game Theory as a MOA

Multi-objective optimization is a very wide research field and literature reports many different approaches to tackle the problem 20. Here we focus on a heuristic algorithm referring to game theory and in particular on competitive games [1]. The idea is to have a number of players equal to the number of objectives. Each player uses a single objective algorithm (the Nelder-Mead Simplex [7]) to improve its objective function working on a subset of the input variables. The algorithm is iterative and at each iteration the variables are re-assigned to the players. The fundamental part of the proposed strategy is how this assignment is performed.

A first implementation of the algorithm relies on the Student test. At the first iteration the variables are assigned randomly. For each subsequent iteration, a player maintains the variables which are judged relevant by the Student test and refuses the non relevant ones, which are randomly reassigned. This algorithm, called *MOGT* (Multi Objective Game Theory), is part of modeFRONTIER [6] and it showed very good performances in finding -in the shortest time and requesting the minimum number of design evaluations- a point on the Pareto set or at least the Nash equilibrium point (which is not assured to lie on the Pareto front, but in many practical application it is close enough).

This assignment rule has the advantage of favoring the exchange of variables among the objective, but at the cost of loosing accuracy. Indeed, the Student test has to be performed on a biased database, since it is computed over the convergence histories of the Simplex algorithms representing the players.

Smoothing Spline ANOVA used as a sensitivity test, on the contrary, performs accurately even when applied on biased databases. The only drawback is that this technique is so precise that it may prevent the exchange of important variables among the players (in the case the same variable is important for more than one objective). Therefore we design two different assignment rules and we switch between them depending on the knowledge or the experience on each single problem.

One assignment rule can be named as *deterministic rule*. After each iteration, the SS-ANOVA coefficients are computed for each objective function. The values are normalized and then ranked. The assignments are done looking at the ranking from the top: the highest coefficient determines the first variable-objective couple. All other entries related to the same variable are excluded from the ranking and the procedure is iterated. At the end, if an objective remains without assigned variables, a repair mechanism reassign randomly one variable.

The *stochastic rule* is based on the roulette wheel operator taken from classical Genetic Algorithms. Once the coefficients are computed and normalized, the values for each variable are collected. A virtual roulette wheel is then created sizing each objective slot proportionally to the related coefficient. The variable-objective assignment is determined by a random number extraction, which simulates the stopping point of the roulette ball.

6.2 Working example

The resulting algorithm is being tested and we will present the results of the benchmark in a future work. However, this new implementation of MOGT is being used on demonstrative, but still challenging real-world optimization problem like the boomerang shape optimization [10].

The cited paper contains the details of the model and the proposed bi-level optimization problem: the shape of the boomerang is optimized under the constraint of a re-entrant launching loop. We are now trying to optimize also the trajectory of the boomerang, maximizing the launch range while minimizing the force applied (and, of course, maintaining the loop condition).

The new MOGT finds interesting configurations and it suggests a possible disconnected Pareto front, as shown in figure 1.

7 Conclusions

Smoothing Spline ANOVA algorithm is a smoothing method suitable for multivariate modeling/regression problems in presence of noisy data. Being a statistical modeling algorithm – based on a function decomposition similar to the classical analysis of variance (ANOVA) decomposition and the associated notions of main effect and interaction – it provides an important advantage over standard parametric models: the interpretability of the results. In fact the relative significance of the different terms – main effects and interactions – composing the model is properly assessed by means of the contribution indices (π_k). In this way the global variance can be explained (decomposed) into single model terms.

For this reason SS-ANOVA represents a suitable screening technique for detecting important variables (Variable Screening) in a given dataset.

The benchmark tests reported in this paper summarizes the performance and accuracy of the SS-ANOVA tool in the determination of the important variables of a problem. The results show the accuracy of this method in detecting important variables in a given dataset. An initial analysis can be refined by repeating the SS-ANOVA procedure on the reduced input variable set determined by the initial analysis.

The industrial applications of such a technique span a wide range of tasks: variable screening is a key ingredient for effective model building; metamodeling can benefit from a reduction of the input variable number; multi-objective optimization algorithms can exploit the information about most relevant variables in internal routines as shown in the MOGT case.

We expect to further enlarge this list, as well as to rigorously prove the effectiveness of the proposed approaches by severe benchmarking. The examples provided in this work, together with the mathematical description of the technique, are promising and plenty justify the choice of including Smoothing Spline ANOVA in the modeFRONTIER software.

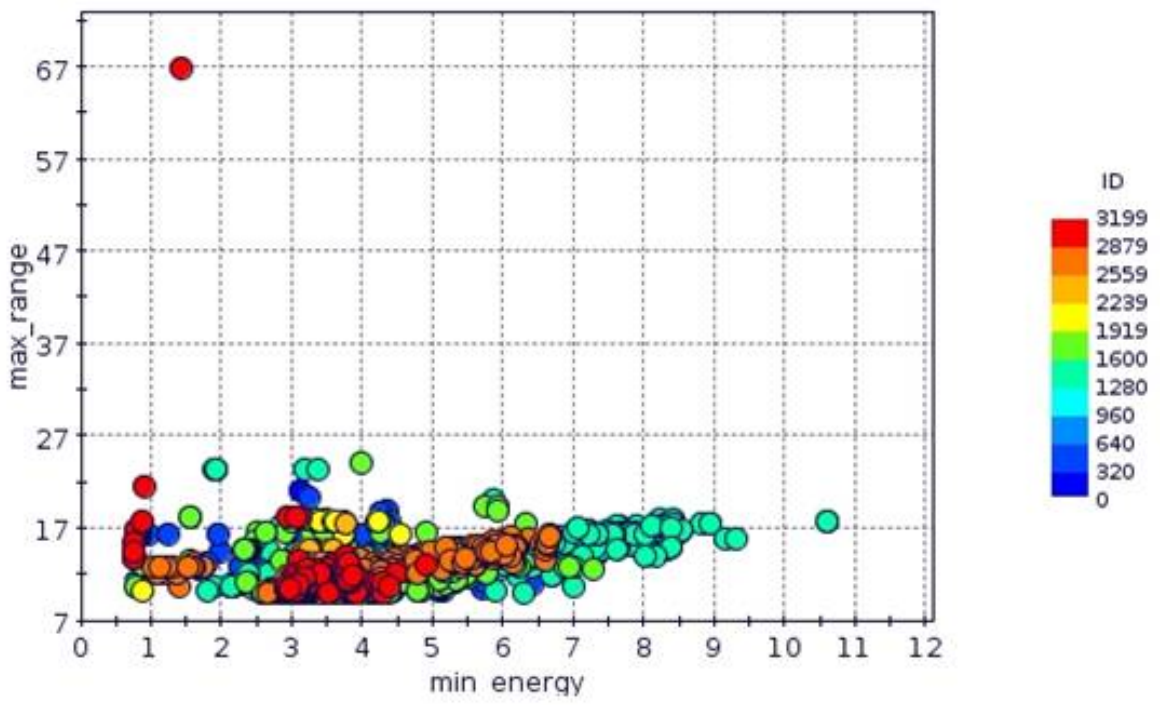

Figure 1: The optimization of the boomerang trajectory done by the new MOGT algorithm enhanced with SS-ANOVA variable screening.

References

[1] A. Clarich and C. Poloni. Multi-objective optimisation in modeFRONTIER for aeronautic applications, including cad parameterisation, evolutionary algorithms, game theory, metamodels and robust design. In *EUROGEN 2007*, Jyvaskyla, 2007.

[2] P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[3] C. Gu. *Smoothing Spline ANOVA Models*. Springer-Verlag, New York, 2002.

[4] C. Gu and G. Wahba. Minimizing GCV/GML scores with multiple smoothing parameters via the newton method. *SIAM Journal on Scientific and Statistical Computing*, 12:383–398, 1991.

[5] Y.-J. Kim and C. Gu. Smoothing spline gaussian regression: more scalable computation via efficient approximation. *Journal of the Royal Statistical Society, Ser. B*, 66:337–356, 2004.

[6] modeFRONTIER is a product of ESTECO s.p.a. www.esteco.com.

[7] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[8] L. Ricco and E. Rigoni. Variable screening tool benchmark tests. Technical Report 2011-008, Esteco, 2011.

[9] E. Rigoni and L. Ricco. Smoothing spline ANOVA for variable screening. Technical Report 2011-007, Esteco, 2011.

[10] R. Russo, A. Clarich, C. Poloni, and E. Nobile. Optimization of a boomerang shape using modeFRONTIER. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012.

[11] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[12] Zhang, Qingfu and Zhou, Aimin and Zhao, Shizheng and Suganthan, Ponnuthurai Nagaratnam and Liu, Wudong and Tiwari, Santosh. Multiobjective optimization test instances for the CEC 2009 special session and competition. *Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report, University of Essex, Colchester, UK and Nanyang Technological University, Singapore*, 2008.

Table 1: Contribution indices obtained by analysing Problem 1 with 500, 1000, 2000, 5000 designs. The last row reports analysis execution time (in seconds).

	500		1000		2000		5000	
Terms	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect
x_1	0.1424	0.0475	0.092	0.0089	0.0882	0.0375	0.0760	0.0326
x_2	0.8581	0.3294	0.8991	0.3511	0.9004	0.3656	0.9178	0.3680
x_3	-5.3327 E-4	-6.4887 E-4	0.0089	0.0037	0.0113	0.0042	0.0061	0.0034
$x_1 * x_2$	–	0.6202	–	0.6034	–	0.5933	–	0.5950
$x_1 * x_3$	–	-2.6677 E-4	–	0.0022	–	5.1315 E-4	–	3.2767 E-4
$x_2 * x_3$	–	0.0038	–	0.0035	–	-0.0013	–	5.2426 E-4
Time	0.924	0.629	0.517	1.546	1.622	3.468	7.714	12.494
	Collinearity Indices							
Max Value	1.0026	1.0305	1.0005	1.0177	1.0004	1.0123	1.0002	1.0080
Min Value	1.0004	1.0078	1.0001	1.0033	1.0001	1.0041	1.0001	1.0007
Mean Value	1.0015	1.0170	1.0003	1.0096	1.0003	1.0072	1.0001	1.0034

Table 2: Contribution indices obtained by analysing Problem 2 with 500, 1000, 2000, 5000 designs. The last row reports analysis execution time (in seconds).

	500		1000		2000		5000	
Terms	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect
x_1	0.9809	0.7189	0.9763	0.7126	0.9729	0.7168	0.9701	0.7140
x_2	0.0214	0.0223	0.0230	0.0183	0.0209	0.0170	0.0262	0.0192
x_3	-0.0028	-0.0035	-2.3527E-4	1.8991E-4	0.0049	0.0038	0.0036	0.0029
x_4	4.1501E-4	-5.2992E-6	3.8814E-4	-8.8781E-5	8.6698E-4	-1.6925E-6	1.6679E-4	4.1998E-6
x_5	9.3178E-4	-4.6284E-5	6.7718E-4	-2.3563E-5	6.1452E-4	-2.1965E-6	-1.9202E-5	-7.5136E-6
x_6	-9.0375E-4	-2.1165E-4	-2.5459E-4	-1.0819E-5	-3.6491E-4	1.2834E-5	-2.7545E-4	4.4288-6
$x_1 * x_2$	–	0.1118	–	0.1053	–	0.1072	–	0.1139
$x_1 * x_3$	–	0.0670	–	0.0748	–	0.0763	–	0.0609
$x_1 * x_4$	–	7.3585E-5	–	1.2678E-4	–	-2.9027E-5	–	-5.6391E-6
$x_1 * x_5$	–	-3.7609E-5	–	6.9811E-5	–	7.9851E-5	–	-1.8004E-4
$x_1 * x_6$	–	4.9782E-5	–	1.3315E-4	–	-1.5828E-4	–	6.1921E-5
$x_2 * x_3$	–	0.0836	–	0.0881	–	0.0789	–	0.0889
$x_2 * x_4$	–	-5.3684E-5	–	1.5795E-5	–	-1.7326E-5	–	-5.5969E-6
$x_2 * x_5$	–	-1.1205E-5	–	-7.0786E-5	–	-3.8717E-5	–	-3.5851E-5
$x_2 * x_6$	–	3.6886E-5	–	1.7460E-4	–	-7.4930E-5	–	-2.0645E-5
$x_3 * x_4$	–	-1.9821E-5	–	3.9820E-5	–	-4.3679E-5	–	-1.5173E-5
$x_3 * x_5$	–	-5.4946E-6	–	5.2976E-5	–	3.4419E-5	–	2.0064E-6
$x_3 * x_6$	–	5.8237E-5	–	4.4585E-5	–	1.8158E-5	–	1.5405E-5
$x_4 * x_5$	–	-1.5142E-4	–	2.2078E-4	–	-5.7626E-5	–	2.0620E-5
$x_4 * x_6$	–	7.9505E-5	–	-9.9617E-5	–	-2.0026E-6	–	1.7995E-6
$x_5 * x_6$	–	-6.6473E-5	–	-1.2678E-4	–	1.2366E-5	–	7.8857E-6
Time	0.34	2.353	0.791	5.491	1.929	14.255	9.188	43.589
	Collinearity Indices							
Max Value	1.0153	1.0657	1.0054	1.0416	1.0017	1.0264	1.0008	1.0260
Min Value	1.0051	1.0129	1.0013	1.0086	1.0003	1.0058	1.0002	1.0014
Mean Value	1.0094	1.0333	1.0035	1.0232	1.0009	1.0146	1.0005	1.0085

Table 3: Contribution indices obtained by analysing Problem 3 with 500, 1000, 2000, 5000 designs. The last row reports analysis execution time (in seconds).

	500		1000		2000		5000	
Terms	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect
x_1	0.7617	0.4274	0.7187	0.3968	0.6725	0.3825	0.6679	0.3777
x_2	0.0137	0.0146	0.0141	0.0102	0.0122	0.0084	0.0179	0.0109
x_3	0.2209	0.1460	0.2639	0.1639	0.3112	0.1836	0.3137	0.1826
x_4	-8.1804E-5	2.1058E-5	7.9491E-4	2.2947E-4	0.0017	1.7455E-4	2.7692E-4	2.5293E-5
x_5	0.0047	4.7546E-5	0.0025	1.8633E-5	0.0021	-2.2416E-5	2.9471E-4	1.8097E-5
x_6	-0.0011	-3.8401E-4	-2.2359E-4	-1.6152E-4	-1.2229E-5	-7.1246E-6	-1.9308E-4	1.7042E-6
$x_1 * x_2$	–	0.0605	–	0.0483	–	0.0519	–	0.0600
$x_1 * x_3$	–	0.1502	–	0.1746	–	0.1839	–	0.1629
$x_1 * x_4$	–	1.7163E-4	–	-1.6553E-6	–	-8.0553E-6	–	1.8182E-5
$x_1 * x_5$	–	2.3562E-5	–	-4.7009E-4	–	5.7116E-5	–	-1.8047E-4
$x_1 * x_6$	–	-7.3218E-4	–	3.5687E-4	–	6.7571E-5	–	2.0754E-4
$x_2 * x_3$	–	0.1996	–	0.2062	–	0.1891	–	0.2056
$x_2 * x_4$	–	-1.1441E-4	–	-2.0679E-5	–	2.2263E-5	–	-3.3959E-5
$x_2 * x_5$	–	0.0016	–	-2.3791E-5	–	3.2200E-5	–	-5.1379E-5
$x_2 * x_6$	–	-6.9754E-4	–	2.7571E-5	–	8.1896E-5	–	-4.7938E-5
$x_3 * x_4$	–	0.0011	–	-1.3185E-4	–	-1.5382E-4	–	-4.8759E-5
$x_3 * x_5$	–	3.0333E-5	–	-2.0252E-4	–	-1.8186E-5	–	-2.4602E-5
$x_3 * x_6$	–	5.1896E-4	–	1.3224E-4	–	9.4878E-5	–	2.1746E-5
$x_4 * x_5$	–	-2.0973E-4	–	1.2186E-4	–	9.1986E-6	–	7.8474E-5
$x_4 * x_6$	–	-9.2892E-5	–	-1.1611E-4	–	-3.9836E-6	–	-2.8478E-6
$x_5 * x_6$	–	7.8214E-5	–	-1.4455E-4	–	2.2336E-5	–	4.2879E-6
Time	0.341	2.303	0.736	5.378	2.118	13.71	8.951	43.255
	Collinearity Indices							
Max Value	1.0138	1.1001	1.0054	1.0438	1.0020	1.0339	1.0007	1.0207
Min Value	1.0042	1.0228	1.0014	1.0073	1.0004	1.0035	1.0002	1.0016
Mean Value	1.0087	1.0434	1.0036	1.0221	1.0010	1.0141	1.0005	1.0085

Table 4: Analysis execution time (in seconds) for Problem 4 with 10, 20, 30, 50 and 100 input variables and 500, 1000, 2000, 5000 designs.

Number	500		1000		2000		5000	
of Inputs	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect	Main Effect	Inter. Effect
10	0,45	6,406	1,115	15053	2,584	38,069	13,921	117,521
20	0,889	26,77	2,148	70,579	5,394	159,239	18,66	500,038
50	2,149	–	4,99	–	14,186	1350,639	45,598	4885,664
100	4,392	–	11,43	–	29,265	–	91,292	–

Table 5: Collinearity indices Problem 4 with 10, 20, 30, 50 and 100 input variables and 500, 1000, 2000, 5000 designs.

		500		1000		2000		5000	
Number	Collinearity	Main	Inter.	Main	Inter.	Main	Inter.	Main	Inter.
of Inputs	Indices	Effect	Effect	Effect	Effect	Effect	Effect	Effect	Effect
10	Max Value	1,0169	1,1254	1,0080	1,0699	1,0055	1,0751	1,0013	1,0609
	Min Value	1,0022	1,0468	1,0017	1,0192	1,0017	1,0121	1,0004	1,0058
	Mean Value	1,0078	1,0782	1,0049	1,0408	1,0030	1,0229	1,0008	1,0191
20	Max Value	1,0315	1,4910	1,0156	1,1994	1,0081	1,1586	1,0026	1,1084
	Min Value	1,0120	1,2626	1,0049	1,1161	1,0027	1,0481	1,0011	1,0205
	Mean Value	1,0192	1,3704	1,0096	1,1522	1,0049	1,0732	1,0018	1,0322
50	Max Value	1,0861	–	1,0389	–	1,0174	1,8250	1,0080	1,2747
	Min Value	1,0398	–	1,0149	–	1,0081	1,5787	1,0028	1,1435
	Mean Value	1,0541	–	1,0260	–	1,0126	1,6950	1,0048	1,1696
100	Max Value	1,1652	–	1,0854	–	1,0359	–	1,0137	–
	Min Value	1,0684	–	1,0388	–	1,0176	–	1,0067	–
	Mean Value	1,1134	–	1,0546	–	1,0255	–	1,0096	–