Struct Multidisc Optim (2014) 49:979–991
DOI 10.1007/s00158-013-1029-z

RESEARCH PAPER

Metamodel-assisted optimization based on multiple kernel regression for mixed variables

Manuel Herrera · Aurore Guglielmetti · Manyu Xiao · Rajan Filomeno Coelho

Received: 25 July 2013 / Revised: 19 November 2013 / Accepted: 20 November 2013 / Published online: 10 January 2014

Abstract While studies in metamodel-assisted optimization predominantly involve continuous variables, this paper explores the additional presence of categorical data, representing for instance the choice of a material or the type of connection. The common approach consisting in mapping them onto integers might lead to inconsistencies or poor approximation results. Therefore, an investigation of the best coding is necessary; however, to build accurate and flexible metamodels, a special attention should also be devoted to the treatment of the distinct nature of the variables involved. Consequently, a multiple kernel regression methodology is proposed, since it allows for selecting separate kernel functions with respect to the variable type. The validation of the advocated approach is carried out on six analytical benchmark test cases and on the structural responses of a rigid frame. In all cases, better performances are obtained by multiple kernel regression with respect to its single kernel counterpart, thereby demonstrating the potential offered by this approach, especially in combination with dummy coding. Finally, multi-objective surrogate-based optimization is performed on the rigid frame example, firstly to illustrate the benefit of dealing with mixed variables for structural design, then to show the reduction in terms of finite element simulations obtained thanks to the metamodels.

Keywords Multiple kernel regression · Mixed variables · Metamodels · Categorical variables · Dummy coding

1 Introduction

Metamodel-assisted optimization has greatly improved the design of mechanical components and civil engineering structures, due to their capacity to address physically complex problems through the use of inexpensive interpolation or regression models (Queipo et al. 2005; Forrester and Keane 2009). However, the majority of existing surrogates encountered in the literature focus on continuous inputs, viz. they do not take explicitly into account discrete, integer, or categorical values, although versatile practical engineering problems also involve non-continuous parameters (see the classification of variables in Fig. 1).

In this paper[1], the following terminology is advocated. *Discrete* variables are real variables available only from a

This work has been supported by Innoviris (Brussels-Capital Region, Belgium) through a BB2B project entitled "Multicriteria optimization with uncertainty quantification applied to the building industry".

M. Herrera · R. Filomeno Coelho (✉)
ULB–BATir Department, Université libre de Bruxelles,
Avenue F.D. Roosevelt, 50 (CP 194/2), B-1050 Brussels, Belgium
e-mail: rfilomen@ulb.ac.be

M. Herrera
e-mail: mherrera@ulb.ac.be

A. Guglielmetti · M. Xiao
NPU–Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072,
People's Republic of China

A. Guglielmetti
e-mail: wuhong_mathnpu@hotmail.fr

M. Xiao
e-mail: manyuxiao@nwpu.edu.cn

[1]This paper is based on a contribution presented at the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10), Orlando, Florida, USA, May 19-24, 2013.

finite set $I_d = \{d_1, \ldots, d_n\}$ where all $d_i \in \mathbb{R}$. *Integer* variables are defined over an interval $I_i \subseteq \mathbb{N}$ or $\mathbb{Z}$ (e.g., the number of stiffeners in a beam). In engineering problems, they differ from discrete variables by the fact that no intermediate values between two integer variables can be defined (e.g., a plate can contain two or three holes, but not 2.5), which has an impact both on the simulation and on the parametrization/optimization sides (note that *binary* variables are a particular case of integer variables). Finally, categorical variables can represent any non-numeric data, like a performance assessment ('low', 'medium', 'high'), or the choice of a material ('steel', 'titanium', 'aluminum'); in the former case, they are said to be ordered (*ordinal* variables), while they are unordered (*nominal* variables) in the latter case (Agresti 1996; McCane and Albert 2008).

To extend popular metamodels to categorical data, two main philosophies can be followed:

1. Using a *unified* model for all variables: after a conversion of the categorical variables into numeric values, the whole vector can be used as a real vector and undergoes the usual metamodeling approach for multivariate inputs (see Fig. 2).
2. Using a *separate model feature* according to the variable type.

Preliminary studies by the authors following philosophy 1–based on the development of moving least squares adapted to mixed (continuous and nominal) variables–have demonstrated their efficiency for a low number of nominal variables and a limited number of *attributes* (i.e., the possible values for the nominal variables) Filomeno Coelho (2012, 2013). However, these approaches do not infer any complex relationship between the inputs (e.g., in a structural design problem, the geometry of a beam cross-section: 'square', 'circle', 'I',...) and the outputs (e.g., the maximum deflection of the beam at mid-span). All attributes are implicitly considered as equally distant in the design space, while in practice clusters of attributes could be determined according to their corresponding influence on the outputs.

Therefore, the aim of this work, leaning on the second philosophy, is to propose a *multiple kernel regression* (MKr) alternative to develop efficient surrogate models which can handle continuous and categorical variables by a number of mapping functions combined, as initially proposed by the authors in Herrera and Filomeno Coelho (2013). Using multiple kernel regression improves prediction accuracy, reduces the number of support vectors, has a better ability to fit complex data, and is adaptive to more difficult problems. Nevertheless, it could be less interpretable and computationally expensive, since to evaluate the model output all basis kernels need to be evaluated. This computational effort has its counterpart by a smaller number of support vectors necessary for the regression than the more common approach of support vector regression (SVR) (Qiu and Lane 2005).

Common kernel-based learning methods use an implicit mapping of the input data into a high dimensional feature

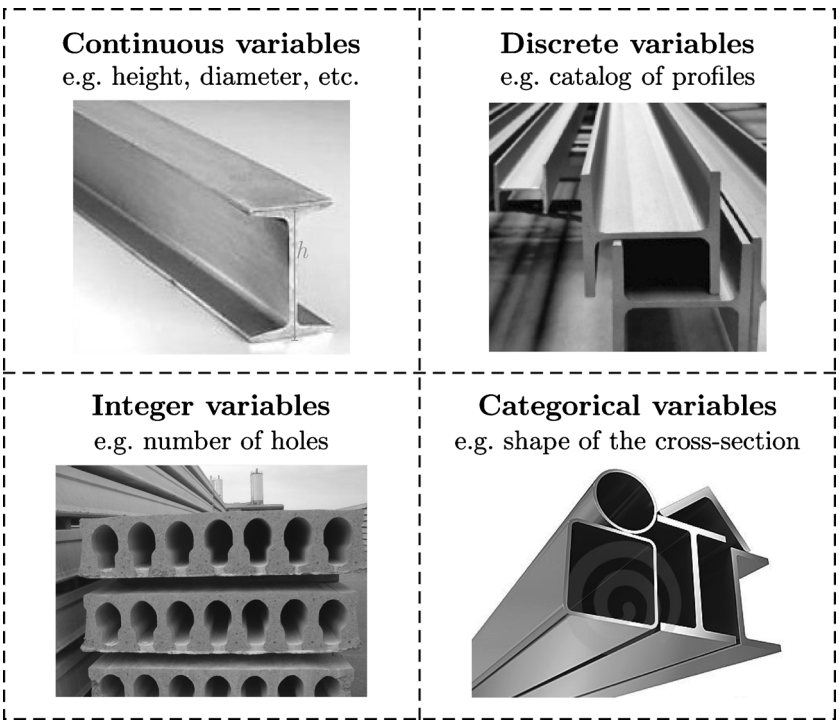

Fig. 1 Classification of the variables according to their type

space defined by a *kernel function*, i.e., a function returning the inner product $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ between the images of two data points $\mathbf{x}$, $\mathbf{x}'$ in the feature space (Schölkopf and Smola 2001; Shawe-Taylor and Cristianini 2006; Hofmann et al. 2008). In the mixed model case, problem specific information is required to adapt the kernel function(s) as much as possible to each particular problem. Furthermore, since two distinct types of variables have to be taken into account, intuition dictates that different kernel functions might provide a more flexible and adequate way to model the inherent behavior of the variables according to their nature (Purcell 2011).

Kernel selection in multiple kernel regression (MKr) (Qiu and Lane 2005; Huang et al. 2007) is very important because its choice is highly dependent on the nature of the input data, and can have a significant effect on the accuracy of predictions. However, when multiple distinct types of input data are involved, it may be difficult to determine the most suitable kernel. We could try instead to find the optimal combination of a set of candidate kernels, where each of the kernels represents a different type of data (Lanckriet et al. 2004). The proposal is thus to apply appropriate kernel functions to generate individual kernel matrices for each kind of variable. These can eventually be combined with a weighted summation and used as training data for a classical SVR (Smola and Schölkopf 2004).

Only a few studies are available on the impact of *coding* (i.e., the conversion from categories to actual numbers) in multiple regression, ANalysis Of VAriance (ANOVA), and other statistical methods (Hardy 1993; Agresti 1996; Cohen et al. 2003). Moreover, there is a minor part of them focused on distance in the case of categorical and/or mixed variables (McCane and Albert 2008), which are mandatory for our understanding of metrics for mixed data. The challenge now is to determine which suitable coding should be plugged in the framework of MKr models.

The outline of the paper is the following. Specific characteristics about dealing with mixed variables are introduced in Section 2. Then, MKr models are described in Section 3. Section 4 contains two experimental studies: first, six analytical benchmark functions are introduced and investigated; then, a structural design example is analyzed to predict the total mass and the compliance of a rigid frame. As an illustration of the benefit of handling mixed variables for design optimization, Section 5 shows the numerical results obtained for the sizing optimization of the rigid frame, along with the reduced number of calls to the finite element simulation allowed by the use of the metamodels proposed. Finally, Section 6 presents conclusions and future challenges of MKr models with mixed variables.

2 Dealing with mixed variables

In practice, regression models with mixed variables (continuous, discrete, and integer) can be addressed by storing all of them in a common vector of numeric values. However, this conversion process is not straightforward for non-numeric inputs, also referred to as *categorical* variables. These variables are divided in two families: while *ordinal* variables can still be ranked (e.g., the size of clothes: 'S', 'M', 'L', 'XL'), *nominal* categorical inputs are *a priori* unordered (e.g., materials: 'steel', 'titanium', 'aluminum'). Various methods are employed to vectorize these nominal cases (Abramson et al. 2004; Cohen et al. 2003; Davis 2010).

The most straightforward technique is the *direct* or *real number conversion*, which assigns a real value to each attribute, thereby leading to an implicit order of the categorical data. As a consequence, this solution may be adequate for ordinal data but might suffer from inconsistencies with nominal ones (Lee and Kim 2010).

To improve this behavior, the basic idea of the *regular simplex* method is to assume that any pair of levels of a categorical variable are separated by the same distance (see Fig. 3). To achieve this, each level of an n-level variable is associated with a distinct vertex of a regular simplex in $(n-1)$ dimensional space (McCane and Albert 2008; Mortier et al. 2006). For simplicity, the Euclidean distance between levels is assumed to be 1. For example, if x^{categ} can take values in a set of $n_{attr} = 3$ possible attributes {○ ; ■ ; 𝙸},

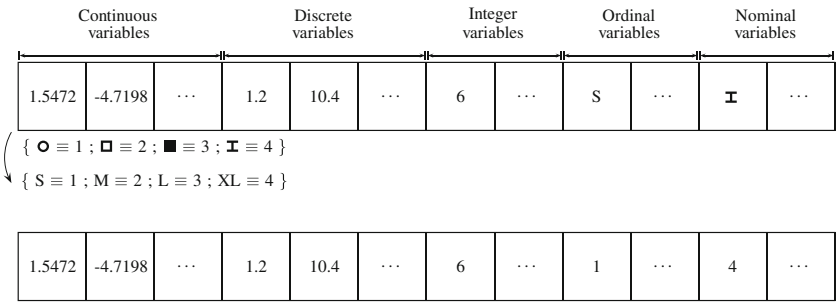

Fig. 2 Unified approach: all variables are converted into real values (either directly, or through a user-defined mapping). Then, the variables are concatenated in a real vector for further use in the metamodeling process

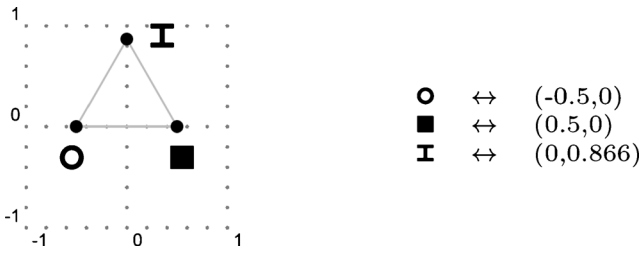

Fig. 3 Representation of a nominal variable with three attributes in the 2D space by a standard regular simplex

these attributes can be drawn in a $(n_{attr} - 1)$ space in such a way that each attribute is converted to the vertex coordinates of a standard regular simplex. By construction, all potential values are thus equally distant (Filomeno Coelho 2013).

At the contrary, *dummy coding* converts one variable into $(n - 1)$ binary variables, equal to 0 except for the value of the categorical variable, if n is the total number of possible values. In the abovementioned example, the conversion would be as follows: {O} → (1, 0); {■} → (0, 1); {I} → (0, 0), as depicted in Fig. 4. One of the strengths of this coding is that it is easily extensible for more complex cases. We should note that the implicit choice of a reference attribute (the vector conformed by all its components equal to zero) is a peculiarity of dummy coding that should be taken into account to control its undesirable effects.

Other coding possibilities are (Davis 2010):

- *contrast coding*: created as an extension of dummy coding to examine mean differences between groups, this coding is meaningless for this case study;
- *effect coding*: its structure maintains the same process as dummy coding except for the last group that receives a (−1) value on all bits (Wendorf 2004).

Nevertheless, whatever conversion procedure is applied, these data are still inherently structured in a different way than the continuous inputs; in other words, both ordinal and nominal information maintains some repeated information and/or structures along the whole database. Therefore, in this paper is suggested to consider this specific aspect through multiple kernel regression.

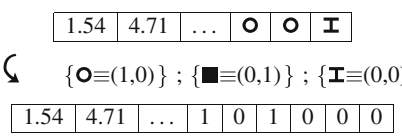

Fig. 4 Mapping from nominal variables to dummy coding values within a mixed design vector

3 Multiple kernel regression

As mentioned earlier, kernel-based learning methods (Schölkopf and Smola 2001, Shawe-Taylor and Cristianini 2006) use an implicit mapping of the input data into a high dimensional feature space defined by a kernel function, i.e., a function K returning the inner product $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ between the images of two data points $\mathbf{x}$, $\mathbf{x}'$ in the feature space. The choice of the map, ϕ, aims to convert the nonlinear relations into linear ones. The learning then takes place in the feature space and the learning algorithm can be expressed so that the data points only appear inside dot products with other points. This is often referred to as the "kernel trick" (Schölkopf and Smola 2001; Schölkopf 2000).

The use of kernel methods is well adapted to data integration as it enables multiple types of data to be converted into a common usable format. Kernels can be eventually combined with a weighted summation and used as training data for a classical support vector regression (SVR) scheme (Sonnenburg et al. 2006). The principles of SVR are summarized below.

3.1 Introduction to support vector regression

The key characteristic of SVR is that it allows to specify a margin ε within which errors are accepted in the sample data without affecting the approximation power. The SVR predictor is defined by the points lying outside the region formed by the band of size $\pm\varepsilon$ around the regression (see (1)). Those vectors are the so-called *support vectors*.

$$\hat{f}(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \tag{1}$$

where ϕ is the mapping function associated with the kernel and $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the parameters that regularize and control the regression hyperplane, respectively.

The goal is to find a function $\hat{f}(\mathbf{x})$ that deviates at most by ε from the observed output y_i from the training data set, while at the same time minimizing the model complexity (see (2)).

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to: } y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b \leq \varepsilon \\ &\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b - y_i \leq \varepsilon \end{aligned} \tag{2}$$

The constraints in (2) assume that $\hat{f}(\mathbf{x})$ exists for all y_i with precision $\pm\varepsilon$. Nevertheless, the solution may actually not exist, or it would be possible to achieve better predictions by allowing outliers. Those are the reasons to include *slack variables* (ξ^+ and ξ^-) on the regression. Thus, we have $\xi^+ = \hat{f}(\mathbf{x}_i) - y(\mathbf{x}_i)$ and $\xi^- = y(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)$ such that both

are greater than ε. The objective function and constraints for SVR are:

$$\min_{\mathbf{w},b} \ \frac{1}{2}||\mathbf{w}||^2 + C\frac{1}{n}\sum_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right)$$
$$\text{subject to: } \ y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b \leq \varepsilon + \xi_i^+,$$
$$\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b - y_i \leq \varepsilon + \xi_i^-,$$
$$\xi_i^+, \xi_i^- \geq 0 \ \ i = 1, \ldots, n \tag{3}$$

where n is the number of training patterns and C is a trade-off parameter between model complexity and training error. Additionally, ξ^+ and ξ^- are slack variables for exceeding the target value by more than ε and for being below the target value by more than ε, respectively. This method of tolerating errors is known as *ε-insensitive* (Schölkopf and Smola 2001).

3.2 Multiple kernel regression for mixed variables

The SVR method uses a single mapping function ϕ, and hence a single kernel function K. If a data set exhibits a local variation of its distribution, using a single kernel may not catch up correctly this behavior. Kernel fusion can help to deal with this problem (Christmann and Hable 2012). Recent applications (Lanckriet et al. 2004; Luts et al. 2012) and developments based on support vector machines have shown that using multiple kernels instead of a single one can enhance interpretation of the decision function and improve classifier performance (Sonnenburg et al. 2006). By the use of various kernels we can address problems from diverse data nature, since they have different measures of similarity corresponding to different kernels. In such cases, combining kernels is one way to combine information sources. It is also an advantage in the perspective of mixed variable programming (Hemker 2008; Abramson et al. 2004). The kernel fusion is straightforward using several mapping functions combined, instead of a single mapping function:

$$\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_M(\mathbf{x})]. \tag{4}$$

We adopt the weighted sum fusion with the following mapping functions:

$$\Phi(\mathbf{x}) = \left[\sqrt{\mu_1}\phi_1(\mathbf{x}), \sqrt{\mu_2}\phi_2(\mathbf{x}), \ldots, \sqrt{\mu_M}\phi_M(\mathbf{x})\right] \tag{5}$$

where $\mu_1, \mu_2, \ldots, \mu_M$ are weights of component functions. Now, the regression problem includes the optimization of two parts. One part is the regression hyperplane $f(\mathbf{x})$ and the other part is the weight vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_M]$. The idea is to address these two parts of the optimization process in one step, based on the parametric dependence idea.

Table 1 Short list of some common kernel functions

Name	Expression
Gaussian	$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
ANOVA	$K(\mathbf{x}, \mathbf{x}') = \sum \exp\left(-\sigma(x^k - x'^k)^2\right)^d$
Linear	$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}' + c$
Polynomial	$K(\mathbf{x}, \mathbf{x}') = (\alpha\mathbf{x}^T\mathbf{x}' + c)^d$
Rational Quadratic	$K(\mathbf{x}, \mathbf{x}') = 1 - \frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\|\mathbf{x}-\mathbf{x}'\|^2 + c}$

The resulting multi-kernel, expressed by (6),

$$\begin{aligned}\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) &= <\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)> \\ &= \mu_1 <\phi_1(\mathbf{x}_i), \phi_1(\mathbf{x}_j)> + \mu_2 <\phi_2(\mathbf{x}_i), \phi_2(\mathbf{x}_j)> + \ldots \\ &\quad + \mu_M <\phi_M(\mathbf{x}_i), \phi_M(\mathbf{x}_j)> \\ &= \mu_1 K_1(\mathbf{x}_i, \mathbf{x}_j) + \mu_2 K_2(\mathbf{x}_i, \mathbf{x}_j) + \ldots + \mu_M K_M(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{s=1}^{M} \mu_s K_s(\mathbf{x}_i, \mathbf{x}_j)\end{aligned} \tag{6}$$

is the weighted sum of M kernel functions constituting another kernel function (Shawe-Taylor and Cristianini 2006). We can solve the regression hyperplane by plugging this multi-kernel on the equation related to the SVR regression surface (Smola and Schölkopf 2004), as described by (7).

$$\hat{f}(\mathbf{x}) = b + \sum_{i=1}^{n}\left(\alpha_i^+ - \alpha_i^-\right)\tilde{K}(\mathbf{x}_i, \mathbf{x}) \tag{7}$$

where α_i^+ and α_i^- for $i = 1, \ldots, n$ are the Lagrange multipliers appearing when solving (3). Their combination (7) plays the role of coefficients in the regression surface.

The next step consists in selecting an appropriate coding for the categorical variables.

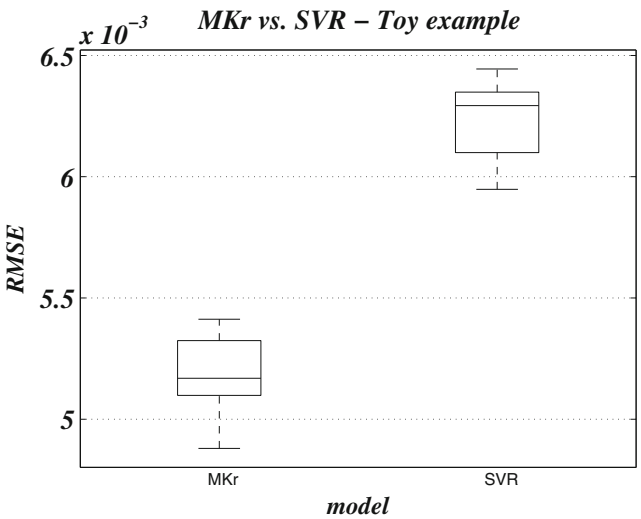

Fig. 5 RMSE results comparing MKr and single SVR for the toy example

Table 2 Definition of the six analytical benchmarks with mixed variables

Output		
$f_{Ellipsoid,MV}$	=	$\sum_{i=1}^{nz}\left(\beta^{\frac{i-1}{nz-1}} z_i\right) + \sum_{i=1}^{nc}\left(\beta^{\frac{i-1}{nc-1}} c_i\right) \quad (\beta = 5)$
$f_{Ackley,MV}$	=	$-20e^{-0.2\sqrt{\frac{1}{nz}\sum_{i=1}^{nz} z_i^2}} - e^{\frac{1}{nz}\sum_{i=1} cos(2\pi z_i)}$ $-20e^{0.2\sqrt{\frac{1}{nc}\sum_{i=1}^{nc} c_i^2}} - e^{\frac{1}{nc}\sum_{i=1} cos(2\pi c_i)} + 20 + e$
$f_{Rastrigin,MV}$	=	$10(nz+nc) + \sum_{i=1}^{nz}\left[z_i^2 - 10cos(2\pi z_i^2)\right]$ $+\sum_{i=1}^{nc}\left[c_i^2 - 10cos\left(2\pi c_i^2\right)\right]$
$f_{Rosenbrock,MV}$	=	$\sum_{i=1}^{nz-1}\left[100\left(z_{i+1} - z_i^2\right)^2 + (z_i - 1)^2\right]$ $+\sum_{i=1}^{nc-1}\left[100\left(c_{i+1} - c_i^2\right)^2 + (c_i - 1)^2\right]$
$f_{Sphere,MV}$	=	$\sum_{i=1}^{nz} z_i^2 + \sum_{i=1}^{nz} c_i^2$
$f_{Griewank,MV}$	=	$\frac{1}{4000}\sum_{i=1}^{nz} z_i^2 - \prod_{i=1}^{nz} cos\left(\frac{z_i}{\sqrt{i}}\right) + \frac{1}{4000}\sum_{i=1}^{nc} c_i^2 - \prod_{i=1}^{nc} cos\left(\frac{c_i}{\sqrt{i}}\right)$
Input		
Cont. vars.		$z_i = 10^{-3} x_i^{cont}, \quad x_i^{cont} \in [-300, 700]$ for $i = 1, \ldots, nz$
Categ. vars.		$c_i \in o_{permut}^{(i)}([-3, -2, \ldots, 7])$ for $i = 1, \ldots, nc$

3.3 Coding nominal variables for MKr

The proposal is to manage the continuous and categorical variables through a suitable coding as described above, while still maintaining memory about the origin of these distinct information. In the case of SVR, it results into more accurate regression surfaces if we apply different kernel matrices depending on this memory. Practically, it means that–as observed later in the numerical examples–for straight numeric data the Gaussian family of kernels usually fits better than others; nevertheless, for data derived from a conversion process, other types of kernels (based on polynomial bases) provide more accurate results. When using categorical features, the polynomial kernel function implies that the classifier considers not only the explicitly features, but also all available sets of size d of features (see Table 1). This way to represent the information is well adapted to handle binary data and is more informative than its transformation by the exponential function associated with Gaussian kernels (Goldberg and Elhadad 2008). Comparing (8) and (9) reveals the difference. While the use of single kernel merges all the data into a unique distance, the use of MKr allows to make differences between different data types. Thus, MKr combines different kernel matrices but does not use the same metric for heterogeneous data.

Table 1 summarizes typical kernel instances in which the corresponding generic parameters should be tuned to their best values for each regression. The case of MKr is of special interest because it manages in separated ways straight numeric data and data coded by different kernels, then merges all the information in one kernel matrix by their direct combination through a weighted sum (see (6)), thereby enhancing the results of any single SVR model.

We should remark that of course there are other kernels than the ones mentioned in Table 1. In addition, there also are other special kernels which are specific for peculiar structures such as trees, strings, and graphs, among others (Kondor and Lafferty 2002; Shawe-Taylor and Cristianini 2006; Hofmann et al. 2008). In this work

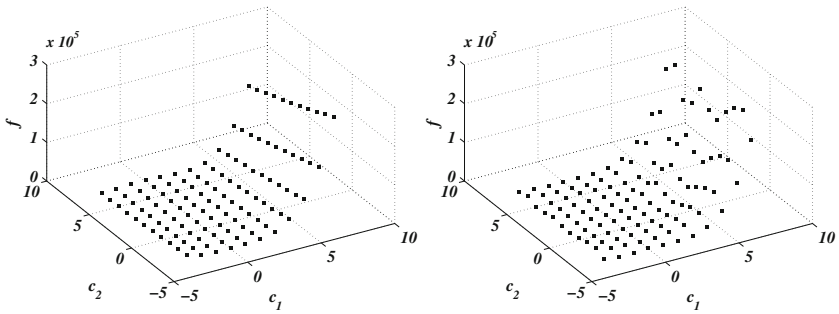

Fig. 6 Mixed Rosenbrock function: for illustration purposes, the continuous variables are fixed $\left(x_1^{cont} = x_2^{cont} = 200\right)$, and the categorical variables c_1 and c_2 vary with respect to ordered (for c_1 and c_2, *left* or mixed (ordered for c_1, unordered for c_2, *right*) attributes

Table 3 Sum of RMSE ($nc = nz = 2, \ldots, 20$) comparing dummy coding and real number conversion

Coding	Direct conversion (RMSE)			Dummy coding (RMSE)		
Model	OLS	SVR	MKr	OLS	SVR	MKr
Ellipsoid	< **1e–15**	0.0257	0.0143	< **1e-15**	0.0336	0.0155
Ackley	0.3818	0.2291	**0.1603**	0.3528	0.2631	0.1815
Rastrigin	0.4429	0.1787	**0.0474**	0.4540	0.1727	0.0495
Rosenbrock	0.3514	0.3763	0.2189	0.0703	0.0498	**0.0382**
Sphere	0.0233	0.0226	0.0091	0.0225	0.0242	**0.0077**
Griewank	0.0886	0.1398	0.0576	0.1337	0.1811	**0.0255**

(the values written in bold refer to the lowest error)

we only address the vector conversion of nominal inputs and its posterior influence in the kernel selection for MKr modeling. Understanding the kernel as a kind of distance (Schölkopf 2000; Tsang et al. 2003), we also propose the use of a metric into kernel expressions taking into account the binary nature of the dummy coding data, in order to achieve a better adequacy of the kernel matrix to dummy coding.

3.4 A toy example on MKr for mixed variables

In order to illustrate the difference in using either a single kernel or the proposed kernel combination, we consider a toy example featuring two inputs x_1 and x_2, corresponding to a continuous and a nominal variable, respectively. Gaussian and polynomial kernels are used on this example. The kernel expression for the single SVR follows (8):

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||(x_1, x_2) - (x_1', x_2')||^2}{2\sigma^2}\right) \quad (8)$$

while (9) is a multiple kernel variant:

$$\tilde{K}(\mathbf{x}, \mathbf{x}') = \mu_1 \exp\left(-\frac{||x_1 - x_1'||^2}{2\sigma^2}\right) + \mu_2 \left(\alpha x_2 x_2' + c\right)^d \quad (9)$$

It is clear that both expressions are different by construction, because multiple kernels (9) consider the mixed nature of the data, whereas (8) makes an implicit numeric conversion of the nominal part of the data, managing all the information in the same vector, with the same norm and distance definition. Figure 5 shows a comparison by box plots of MKr and SVR with respect to their root mean square error (RMSE) results. These are based on 800 training and 200 test sample data generated by a Latin Hypercube Sampling (LHS) adapted to mixed variables, on a simple case of Rastrigin function (see Table 2), just defined by x_1 and x_2. The parameters of both models have been tuned by a grid search algorithm. The results show that the RMSE related to MKr is inferior to the one related to SVR. Its difference measures the better treatment of the nominal data observed with the multiple kernel expression.

4 Numerical experiments

The development of two case studies will contribute in a better understanding and validation of the multiple kernel regression procedure for mixed variables. To summarize, we will address the following issues:

- comparing MKr and SVR. This comparison is of special interest in order to measure the profit of using MKr instead of SVR. A second order ordinary least squares (OLS) is added as a reference method for these comparisons;
- comparing the different coding options for categorical inputs.

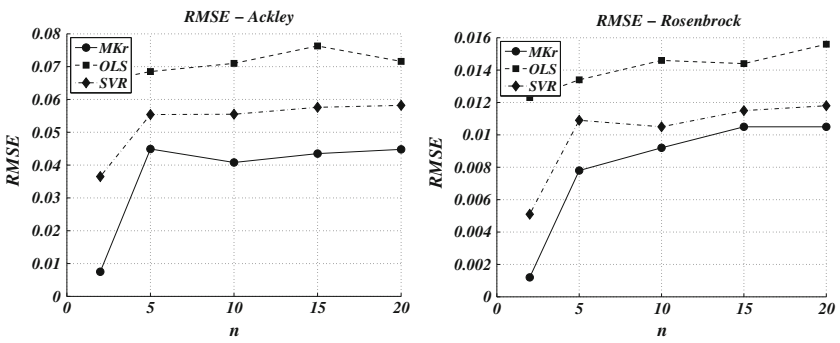

Fig. 7 Comparison of regression models with dummy coding by the RMSE of the benchmark functions (Ackley & Rosenbrock)

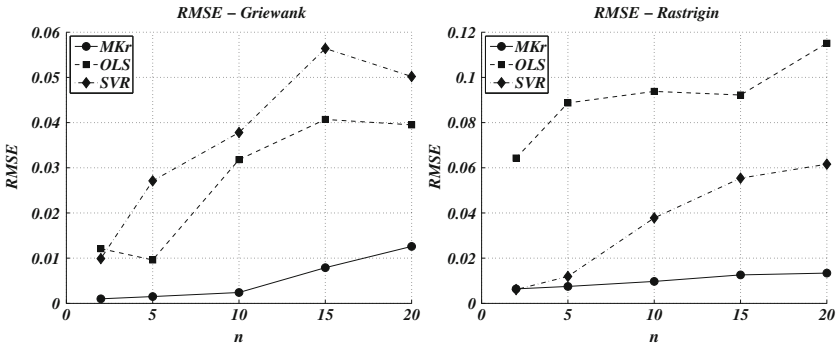

Fig. 8 Comparison of regression models with dummy coding by the RMSE of the benchmark functions (Griewank & Rastrigin)

The first case study is related to analytical mixed-variable benchmark functions, while the second one is an engineering problem, namely the structural design analysis of a 3D rigid frame.

4.1 Analytical test cases

The MKr for mixed variables is first tested on a set of six analytical mixed-variable benchmark functions. The design variables consist in nz continuous and nc categorical nominal variables (for all examples: $nz = nc$). To model nominal variables, whose central feature is their lack of intrinsic ordering, a random permutation operator $o^{(i)}_{permut}$ is applied to the set of attributes of each c_i. The complete definition of the benchmarks is available in Table 2.

As an illustration, Fig. 6 depicts the mixed version of Rosenbrock function for two continuous and two categorical variables. The continuous variables are set to the middle values between their bounds $\left(x_1^{cont} = x_2^{cont} = 200\right)$. *Ordinal* variables (Fig. 6, left) are ranked, thereby corresponding to ordered values of the attributes and therefore a smooth output function. On the contrary, if c_1 remains ordinal but c_2 becomes nominal, the responses become clearly more difficult to represent by a smooth surface, as exhibited in Fig. 6, right. Here, the nominal (hence: *a priori* unordered) variable c_2 is modeled through a random perturbation of its attributes. This intrinsic lack of ordering leads to a non-smooth behavior of the output function. The latter case mimics the unknown ordering of attributes for non-numerical variables.

To analyze the evolution of the approximation efficiency with respect to the dimension of the problem, the number of design variables ranges from $nz = nc = 2$ to $nz = nc = 20$. For each test case, a training set and a validation set are generated by a LHS adapted to mixed variables (i.e., a Latin Hypercube Sampling for the continuous variables, along with a simple random selection for the nominal variables). In all cases the data are composed of 5,000 samples: 4,000 for training and validation to obtain the best possible regression model, and 1,000 reserved for testing the regression (these test data are also sampled by LHS to be representative of the entire population). In order to add statistical significance, each experiment is repeated 20 times.

4.1.1 Numerical results

In this first case study, we test the six functions coded by direct numeric conversion and dummy coding of their categorical inputs (see Table 2). Table 3 shows a comparison between both codings. The best values (mentioned in bold) highlight that the minimum RMSE is associated with MKr in all cases except for the Ellipsoid function. In particular, dummy coding works better than real number conversion in Rosenbrock and Griewank functions, where the RMSE increases, in average, by a factor of 5 for the former example, and 2 in the latter.

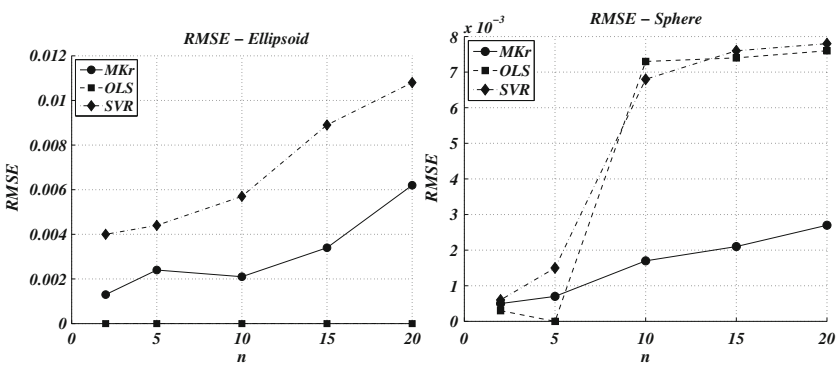

Fig. 9 Comparison of regression models with dummy coding by the RMSE of the benchmark functions (Ellipsoid & Sphere)

Table 4 RMSE (average for all $nz = nc$) for some kernel combinations in MKr

Kernel categ.	Kernel cont.	Ellipsoid	Ackley	Rastrigin	Rosenbrock	Sphere	Griewank
Polynomial	Gaussian	**0.0031**	0.0378	0.0108	0.0113	**0.0015**	**0.0051**
Rational Q.	Gaussian	0.0046	**0.0363**	**0.0099**	**0.0079**	0.0066	0.0231
Gaussian	Gaussian	0.0067	0.0526	0.0346	0.0100	0.0049	0.0363
Polynomial	Polynomial	0.0442	0.2452	0.2258	0.0835	0.1607	0.0335
Rational Q.	Rational Q.	0.0052	0.0667	0.1431	0.0712	0.1138	0.0329

Next to this initial point, a comparison between the MKr and SVR employed with respect to the number of input variables is shown in Figs. 7, 8 and 9. Second order OLS models have been chosen as a basic but robust method in order to add a reference model for the comparisons. Gaussian kernel has been implemented in SVR models. In the case of MKr, a Gaussian kernel is used for the continuous variables, while the categorical use either Polynomial kernels (in Ellipsoid, Griewank, and Sphere functions) or Rational Quadratic kernels (in Ackley, Rastrigin, and Rosenbrock functions). These kernels were selected from a predefined finite set, but increasing the number of candidates in an iterative manner by tuning their parameters for each choice by a grid search algorithm. Thus, the process of kernel selection focuses on modifications of basis kernels with better performance (Gönen and Alpaydin 2011). The RMSE of part of their possible combinations are shown in Table 4. This table is composed of average values for all values of $nz = nc = 2, 5, 10, 15, 20$ and for each function and combination of kernels: Polynomial, Gaussian, and Rational Quadratic (Rational Q., in Table 4).

From Figs. 7–9 some salient features can be extracted. Firstly, the special error behavior depicted in Ellipsoid functions (Fig. 9) is due to the fact that a second-order OLS approximation fits exactly an n-dimensional ellipsoid; in the rest of examples, OLS usually fits worst than the others. We note that MKr beats SVR in all cases. We also observe that MKr and SVR models are close to reveal a parallel trend in their behavior. The difference that distinguishes both can be attributed to the part of variability, explained by the kernel associated with the categorical inputs. In order to quantify this difference the ratio between the accumulative sum of RMSE_{SVR} over RMSE_{MKr} can be used, as shown in Table 5.

Finally, it is worth noticing that if Gaussian kernels are used for all variables within the MKr framework, results very similar to SVR are logically obtained, the slight discrepancy being related to the parameter determination. In case of polynomial or rational quadratic kernels for both types of variables, MKr also exhibits a noticeable loss in the accuracy of the prediction, increasing the RMSE in average for the majority of the cases (see Table 4).

4.2 Description of the structural design example

To analyze the efficiency of MKr combined to dummy coding for an engineering application, we introduce a structural design example consisting in the static analysis of a 3D rigid frame (Liew et al. 2000) (see Fig. 10). The loads of the structure are derived from Eurocode 3 (Papadrakakis et al. 2005):

- the dead load of the beams and columns;
- the gravity load on the floors (19.16kPa);
- the lateral load due to the wind (110kN).

The beams or columns are classified in five groups of common cross-sections. The quantities of interest are the total mass of the structure and the logarithm of the compliance (Ferreira 2009). Ten design variables are necessary to parametrize a design:

- for each of the five groups of beams, a categorical variable defines the cross-section geometry among seven attributes { □ ; ○ ; I ; ■ ; ● ; ▯ ; ▮ };
- additionally, for each group, a continuous bounded variable defines the maximum length l of the cross-section (either height or diameter) such that $0.09\ \text{m} \leq l \leq 0.11\ \text{m}$. For the rectangular cross-section, the width is defined as half of the height; for the I-section, the width is equal to the height. For hollow shapes and for the I-section, the thickness is equal to 2.5 mm.

The geometry of the cross-section is typically a nominal variable, since no ordering of the available cross-section

Table 5 Sum of RMSE ($nc = nz = 2, \ldots, 20$) for all the functions: SVR over MKr ratio

	Ellipsoid	Ackley	Rastrigin	Rosenbrock	Sphere	Griewank
SVR	0.0336	0.2631	0.1727	0.0498	0.0242	0.1811
MKr	0.0155	0.1815	0.0495	0.0382	0.0077	0.0255
Ratio	2.1677	1.4495	3.4888	1.3037	3.1428	7.1019

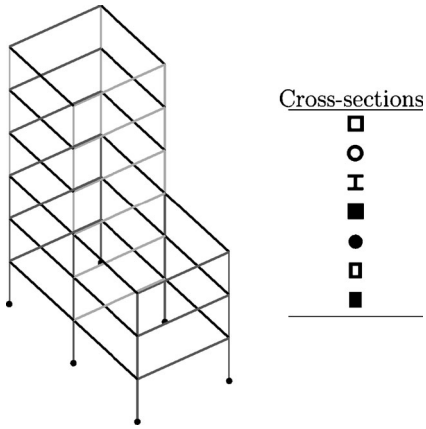

Fig. 10 Rigid frame design (boundary conditions: the lower nodes are fixed in all directions)

types can be made *a priori*. The choice of the cross-section has a direct impact on the calculation of the quantities (area, moments of inertia) necessary to get the normal efforts, shear forces, and bending moments, and the compliance. For each test case, a training set and a validation set are generated by a LHS adapted to mixed variables. The data are composed of 5,000 samples: 4,000 for training and 1,000 reserved for validation (also sampled by LHS).

4.2.1 Numerical results

In this example, a dummy coding of the nominal inputs is achieved, since it provides more accurate results as can be observed in Table 7. For all responses, MKr is the best option, but due to the variant of coding applied (depending on the choice of the "reference" vector set to 0 for all bits), RMSE ranges from 0.0438 to 0.0578. The selected coding, which corresponds to the minimum RMSE associated, is shown in Table 6.

Figure 11 illustrates the results for the 100 predictions, on a new fresh data set, with respect to the values of the total mass and the compliance of the structure provided by the "exact" simulation. A parallel analysis was done to predict the compliance. The results with the best dummy coding option, effect coding, and real number conversion are shown in Table 7. Again, the MKr approach is the best predictive model.

Table 6 Best dummy coding combination for the structural design case study

	Numeric conversion	Dummy coding
□	1	(1, 0, 0, 0, 0, 0)
○	2	(0, 1, 0, 0, 0, 0)
I	3	(0, 0, 1, 0, 0, 0)
■	4	(0, 0, 0, 1, 0, 0)
●	5	(0, 0, 0, 0, 1, 0)
□	6	(0, 0, 0, 0, 0, 1)
■	7	(0, 0, 0, 0, 0, 0)

5 Metamodel-assisted multi-objective optimization of the rigid frame

The final study of this work consists in the multi-objective design optimization of the rigid frame with respect to the categorical and continuous variables. The problem is formulated as follows:

$$\begin{cases} \min\limits_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{c} f_1 \equiv \text{mass} \\ \\ f_2 \equiv \log(\text{compliance}) \end{array} \right\} \\ \text{subject to: } \mathbf{x} = \{c_1, c_2, c_3, c_4, c_5, l_1, l_2, l_3, l_4, l_5\}, \\ \qquad c_i \in \{ \square ; \circ ; \text{I} ; \blacksquare ; \bullet ; \square ; \blacksquare \}, i = 1 \ldots, 5, \\ \qquad l_i \in [0.09, 0.11], i = 1 \ldots, 5. \end{cases} \tag{10}$$

Although the problem size is still reasonable, the number of categorical variables (= 5) and attributes by category (= 7) shows that an enumeration procedure would require $7^5 = 16,807$ combinations alone (without taking into account the continuous variables). Pure enumeration approaches are therefore unaffordable in this context.

While local optimality in single-objective optimization should be specifically redefined for mixed variables (as performed by Abramson et al. 2004),[2] the notion of Pareto optimality is directly applicable to the mixed-variable case, since only the information from the objectives is required.

On an algorithmic point of view, multi-objective implementations of metaheuristics (e.g., genetic algorithms, evolutionary strategies, particle swarms) have demonstrated their efficiency in finding Pareto sets in versatile engineering applications (Coello Coello et al. 2002). The multi-objective optimizer used in this study is the Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al. 2002), where the probabilities of simulated binary crossover and mutation are respectively set to 0.9 and 0.5, and the

[2]For a synthesis of single-objective optimization studies for mixed variables in engineering design, the reader is referred to Filomeno Coelho (2013).

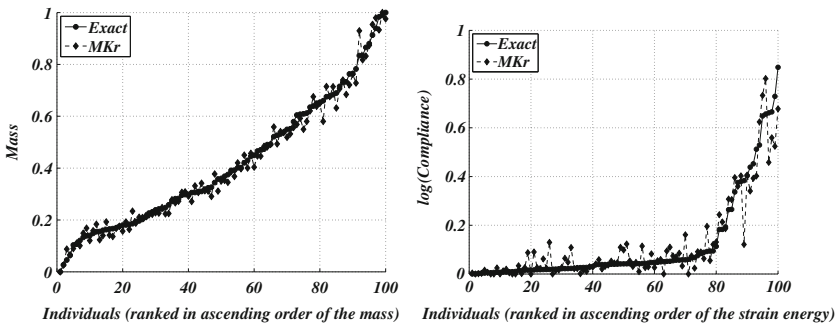

Fig. 11 Normalized observed quantity of interest vs. normalized MKr prediction for 100 new values (*left*: mass; *right*: log(compliance))

distribution index for simulated binary crossover (η_c) and mutation (η_m) are respectively set to 10 and 20.

The implementation has been modified to tackle nominal variables by adapting the evolutionary operators as follows:

- *crossover*: for each nominal variable and at the user-defined probability of crossover, the operation consists in swapping the values of the parents provided a randomly generated number is above 0.5;
- *mutation*: for each nominal variable and for the user-defined probability of mutation, the operation consists in changing the value of the variable randomly among the set of attributes.

Specific research work is required to improve and better adapt the evolutionary data structure to mixed variables, but these simple modifications are sufficient here to demonstrate the approximation power of the MKr-based metamodels.

Two calculations are performed:

- a reference multi-objective optimization using only the finite element simulation to assess the responses (i.e., no metamodels);
- a multi-objective optimization with MKr models replacing the high-fidelity simulation during the search for the Pareto front. To train the metamodel before the optimization process itself, the training set is composed of 2,000 individuals (generated by Latin Hypercube Sampling for the continuous variables, and random sampling for the categorical ones), each of them undergoing the finite element analysis to obtain the corresponding responses.

For both cases, the population size is set to 1,000 individuals. It is difficult to draw a general guideline for the choice of the training set size (N_{TS}) or the population size (N_{pop}), but our experience shows that a good practice consists in choosing $N_{TS} \approx N_{pop} \approx 100.var$, where N_{var} is the number of mixed variables. Of course these values also depend on the number of attributes by categorical variable, the output functions, etc. The non-dominated fronts are depicted in Fig. 12, showing that the solutions are clustered in two parts: for lower mass, a narrow set of solutions rapidly degrading the compliance performances for any slight mass reduction; then, for higher values of the mass, significantly lower compliance can be obtained. The approximation is more efficient for lower values of the mass, while the region characterized by lower compliance is more difficult to find. This can be explained by the relatively limited number of samples with low compliance present in the initial training set. While the relatively low accuracy of the surrogates is acknowledged by the authors, the following remarks can be drawn up:

- in practice, efficient metamodel-assisted optimization should consider an *online* procedure to add sampling points during the optimization phase, based on infill criteria related to the exploration of promising areas of the search space and to the accuracy of the metamodels (Forrester and Keane 2009). However, for this application, a good overview of the Pareto front is already demonstrated by the offline approach;

Table 7 RMSE for total mass and log(compliance) predictive models

Model	OLS	SVR	MKr	OLS	SVR	MKr
Output	Total mass			Log(Compliance)		
RMSE (dummy coding)	0.0661	0.0625	**0.0438**	0.1442	0.1081	**0.0766**
RMSE (effect coding)	0.0682	0.0649	0.0531	0.1442	0.1128	0.0858
RMSE (real number conversion)	0.1001	0.0824	0.0578	0.1752	0.1604	0.1103

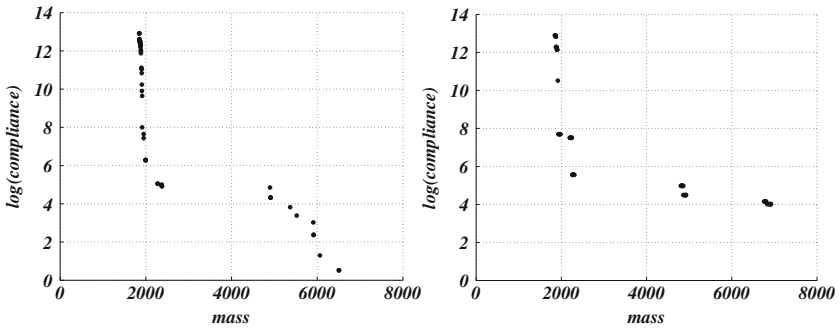

Fig. 12 Rigid frame: Pareto fronts obtained by multi-objective optimization coupled with the finite element simulation (*left*), or by an offline metamodel-assisted optimization (*right*)

– in any case, the "continuous" version of the surrogate models, when tested on mixed data, systematically show worse performances than its multiple kernel counterpart;
– finally, a richer "dictionary of kernels" could be used to increase the choice between the kernel functions; additionally, the tuning of the kernel function parameters is also a path of investigation to enhance the accuracy of the surrogates.

From an engineering design point of view, it is also interesting to observe that–as expected–the optimal shapes are mostly hollow profiles, as depicted in the values of the categories of the Pareto set (see Fig. 13).

Finally, the comparison between the number of simulations required by both optimization processes can be pointed out:

– without metamodels: 50,000 finite element analyzes are required (i.e., 50 generations, 1,000 individuals by generation) to reach convergence;
– with metamodels: only 2,000 finite element analyzes are necessary to obtain the non-dominated front.

6 Conclusions

This paper addresses meta-model assisted optimization for mixed variables, with an emphasis on continuous and categorical data. Intuition dictates that distinct variable types should be treated differently; a multiple kernel regression (MKr) paradigm has been chosen accordingly, consisting in improving the usual kernel methods through a separate treatment with respect to the nature of the inputs. Moreover, a comparison of several mapping schemes from categories to numeric values has been performed.

The numerical results obtained on six analytical benchmark test cases and on the structural analysis of a rigid frame reveal that MKr outperforms other methods in the structural design case study, and provides excellent results in the mixed-variable benchmark functions. MKr is computationally more efficient than support vector regression in terms of execution time, number of iterations, and number of support vectors. On the mapping side, dummy coding leads to better results than direct real number conversion for nominal inputs. Consequently, by setting a specific kernel for the inputs derived from the dummy coding conversion, MKr allows for combining the kernels for all types of variables into one global procedure.

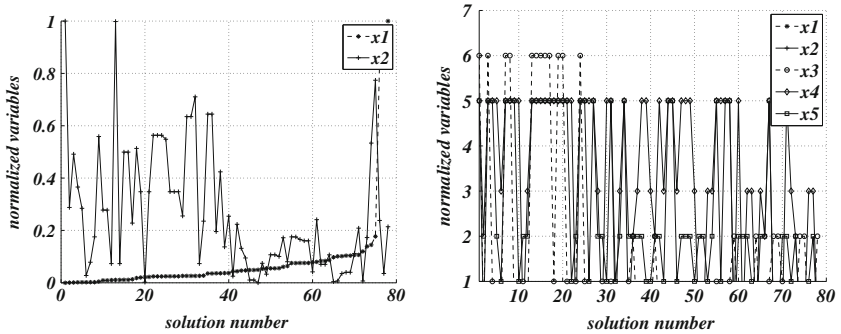

Fig. 13 Pareto optimal set obtained for the rigid frame example: the combinations of real variables (*left*) and categorical variables (*right*) found for each of the solutions are depicted. For the five categorical variables (representing the cross-section shapes for the five groups of beams), the y-axis is defined as follows: 1 ≡ ⌶, 2 ≡ □, 3 ≡ ○, 4 ≡ ▫, 5 ≡ ■, 6 ≡ ●, 7 ≡ ▮

Further improvements are currently under investigation, as using spectral clustering to zoom on interesting zones, developing self-tuning of the parameters of the method, and devising online surrogate-based optimization techniques.

Acknowledgments The authors would like to thank the Associate Editor and the Reviewers for their fruitful comments and suggestions.

The second and third authors also acknowledge support by the Basic Project Foundation of Northwestern Polytechnical University (GCKY1011).

References

Abramson M, Audet C, Dennis DEJ (2004) Filter pattern search algorithms for mixed variable constrained optimization problems. SIAM J Optim 11:573–594

Agresti A (1996) An introduction to categorical data analysis. Wiley, New York

Christmann A, Hable R (2012) Consistency of support vector machines using additive kernels for additive models. Comput Stat Data Anal 56(4):854–873

Coello Coello CA, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic/Plenum Publishers, New York

Cohen J, Cohen P, West SG, Aiken LS (2003) Applied multiple regression/correlation analysis for the behavioural sciences. Routledge, New York

Davis MJ (2010) Contrast coding in multiple regression analysis: strengths, weaknesses, and utility of popular coding structures. Data Sci 8:61–73

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Ferreira AJM (2009) MATLAB codes for finite element analysis. Solid mechanics and its applications. Springer, New York

Filomeno Coelho R (2012) Extending moving least squares to mixed variables for metamodel-assisted optimization. In: 6th European congress on computational methods in applied sciences and engineering (ECCOMAS 2012). Vienna

Filomeno Coelho R (2013) Metamodels for mixed variables based on moving least squares–application to the structural analysis of a rigid frame. Optim Eng. doi:10.1007/s11081-013-9216-8

Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(1–3):50–79

Goldberg Y, Elhadad M (2008) splitSVM: Fast, space-efficient, non-heuristic, polynomial kernel computation for NLP applications. In: 46st annual meeting of the association of computational linguistics (ACL)

Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. Mach Learn Res 12:2211–2268

Hardy M (1993) Regression with dummy variables. Sage, Newbury Park

Hemker T (2008) Derivative free surrogate optimization for mixed-integer nonlinear black box problems in engineering. PhD thesis, Technisen Universität Darmstad, Germany

Herrera M, Filomeno Coelho R (2013) Metamodels for mixed variables by multiple kernel regression. In: 10th world congress on structural and multidisciplinary optimization (WCSMO 10). Orlando

Hofmann T, Schölkopf B, Smola A (2008) Kernel methods in machine learning. Ann Stat 36(3):1171–1220

Huang CM, Lee YJ, Lin DK, Huang SY (2007) Model selection for support vector machines via uniform design. Comput Stat Data Anal 52(1):335–346

Kondor RI, Lafferty JD (2002) Diffusion kernels on graphs and other discrete input spaces. In: Proceedings of the nineteenth international conference on machine learning, ICML '02. Morgan Kaufmann Publishers Inc., San Francisco, pp 315–322

Lanckriet G, Cristianini N, Barlett P, El-Ghaoui L, Jordan MI (2004) Learning the kernel matrix with semi-definite programming. Mach Learn Res 5:27–72

Lee N, Kim JM (2010) Conversion of categorical variables into numerical variables via bayesian network classifiers for binary classifications. Comput Stat Data Anal 54(5):1247–1265

Liew R, Chen H, Shanmugam N, Chen W (2000) Improved nonlinear plastic hinge analysis of space frame structures. Eng Struct 22(10):1324–1338

Luts J, Molenberghs G, Verbeke G, Huffel SV, Suykens JA (2012) A mixed effects least squares support vector machine model for classification of longitudinal data. Comput Stat Data Anal 56(3):611–628

McCane B, Albert MH (2008) Distance functions for categorical and mixed variables. Pattern Recogn Lett 29(7):986–993

Mortier F, Robin S, Lassalvy S, Baril C, Bar-Hen A (2006) Prediction of Euclidean distances with discrete and continuous outcomes. Multivar Anal 97(8):1799–1814

Papadrakakis M, Lagaros N, Plevris V (2005) Design optimization of steel structures considering uncertainties. Eng Struct 27:1408–1418

Purcell R (2011) Machine learning with multiple kernel learning algorithms. Master's thesis, University of Bristol, UK

Qiu S, Lane T (2005) Multiple kernel learning for support vector regression. Tech. rep. Computer Science Department, University of New Mexico, Albuquerque

Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. Prog Aerosp Sci 41:1–28

Schölkopf B (2000) The kernel trick for distances. Tech. rep., Microsoft Research

Schölkopf B, Smola AJ (2001) Learning with kernels, support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

Shawe-Taylor J, Cristianini N (2006) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

Smola A, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Sonnenburg S, Rätsch G, Schäfer C (2006) A general and efficient multiple kernel learning algorithm. In: Weiss Y, Schölkopf B, Platt J (eds) Advances in neural information processing systems 2006. MIT Press, Cambridge, pp 1273–1280

Tsang IW, Kwok JT, Bay CW (2003) Distance metric learning with kernels. In: International conference on artificial neural networks 2003, pp 126–129

Wendorf CA (2004) Primer on multiple regression coding: common forms and the additional case of repeated contrasts. Underst Stat 3:47–57