10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

Metamodels for mixed variables by multiple kernel regression

Manuel Herrera, Rajan Filomeno Coelho

BATir Department, Université libre de Bruxelles
Avenue Franklin Roosevelt, 50 (CP 194/2), 1050 Brussels, Belgium
{mherrera,rfilomen}@ulb.ac.be

1. Abstract
This paper is concerned with the development of metamodels specifically tailored for mixed variables, in particular continuous and categorical variables. Practically, we propose a surrogate model based on multiple kernel regression, and apply it to six benchmark test functions and a rigid frame structural analysis. When compared to other metamodels (support vector regression, ordinary least squares), the numerical results show the efficiency of the method, related to the flexible selection of different types of kernel functions. Further work will include the use of these metamodels for mixed-variable surrogate-based optimization involving computationally expensive simulations.

2. Keywords: surrogate models, support vector regression, multiple kernel regression, mixed variables, categorical variables.

3. 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 [1]. However, a majority of existing surrogate models 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. In particular, categorical variables can represent any non-numerical 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 [2, 3].

Preliminary studies by the authors [4, 5]–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). However, these approaches do not infer any *a priori* 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 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. Common kernel-based learning methods use an implicit mapping of the input data into a high dimensional feature space defined by a kernel function, i.e., a function returning the inner product $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ between the images of two data points $\boldsymbol{x}$, $\boldsymbol{x}'$ in the feature space. Since two distinct types of variables (i.e. continuous and categorical) have to be taken into account, the 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. This idea is explored in this study.

4. Multiple kernel regression
The use of kernel methods [6] lends itself well to the problem of data integration as it enables multiple distinct types of data to be converted into a common usable format. For example, the available data met in engineering catalogs of technological parameters like the choice of a material are typically categorical, while the geometrical variables (e.g. diameter, length) are generally continuous. If the interest of the study is the establishment of predictive models with respect to a set of inputs, the information from all of the sources should be utilized to allow for well informed and balanced predictions. By applying appropriate kernel functions it should be possible to generate individual kernel matrices for the different types of data.

Kernel selection in multiple kernel regression (MKr) is very important because its choice is highly dependent on the nature of the input data, and has a significant impact on the accuracy of predictions [7]. However, the determination of the most suitable kernel is a delicate task, and a poor choice can degrade the approximation. Instead of focusing on a single kernel, an optimal combination of a set of candidate kernels could be searched for, where each of the kernels represents a different type of data. The proposal is thus to apply appropriate kernel functions to generate individual kernel matrices for the different variable types; these kernels can be combined eventually with a weighted summation and used as training data for a classical support vector regression (SVR) [8].

Before describing the MKr approach for mixed variables, a few basic notions about SVR will be recalled. The key characteristic of SVR is that it allows to specify a margin, ε, within which errors in the sample data are accepted without affecting the quality of the prediction. The SVR predictor is defined by the points lying outside the region defined by the band of size $\pm\varepsilon$ around the regression (see Eq.(1)). Those vectors are the so-called *support vectors*.

$$\hat{f}(\boldsymbol{x}) = b + \sum_{i=1}^{n} \omega_i \phi(\boldsymbol{x}, x_i) \tag{1}$$

Considering a linear regression $\hat{f}(\boldsymbol{x}) = b + \boldsymbol{w}^T\boldsymbol{x}$, the goal is to find a function with at most deviates ε from the observed output for the regression at the same time that minimizes the model complexity (see Eq.(2)).

$$\begin{aligned} &\min \frac{1}{2}||\boldsymbol{w}||^2 \\ &\text{subject to } y_i - \langle \boldsymbol{w}, x_i \rangle - b \leq \varepsilon \\ &\qquad\qquad \langle \boldsymbol{w}, x_i \rangle + b - y_i \leq \varepsilon \end{aligned} \tag{2}$$

The constraints in Eq.(2) enforce that $\hat{f}(\boldsymbol{x})$ exists for all y_i with precision $\pm\varepsilon$. Nevertheless, the solution may actually not exist; moreover, it is often possible to achieve better predictions by allowing outliers. Consequently, *slack variables* ξ^+ and ξ^- are included such that:

$$\xi^+ = f(x_i) - y(x_i) > \varepsilon \tag{3}$$

$$\xi^- = y(x_i) - f(x_i) > \varepsilon \tag{4}$$

and the objective function and constraints for SVR are formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{w},b} \ \frac{1}{2}||\boldsymbol{w}||^2 + C\frac{1}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) \\ &\text{subject to } \ y_i - \langle \boldsymbol{w}, x_i \rangle - b \leq \varepsilon + \xi_i^+, \\ &\qquad\qquad \langle \boldsymbol{w}, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^-, \\ &\qquad\qquad \xi_i^+, \xi_i^- \geq 0 \ \ i = 1, \ldots, n \end{aligned} \tag{5}$$

where n is the number of training patterns and C is a parameter which gives a trade-off between model complexity and training error. As mentioned earlier, ξ^+ and ξ^- are slack variables allowing 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* [6].

The SVR method uses a single mapping function ϕ, and hence a single kernel function K. If a dataset has a locally varying distribution, using a single kernel may not catch up correctly the varying distribution. Kernel fusion can help to deal with this problem. Recent applications 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. Analogously, we make here the hypothesis that by the use of different kernels we can tackle problems with different data types.

The kernel fusion is straightforward using several mapping functions combined instead of one single mapping function. Assuming a basis of kernels:

$$\Phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})] \tag{6}$$

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

$$\Phi(\boldsymbol{x}) = [\sqrt{\mu_1}\phi_1(\boldsymbol{x}), \sqrt{\mu_2}\phi_2(\boldsymbol{x}), \ldots, \sqrt{\mu_M}\phi_M(\boldsymbol{x})] \tag{7}$$

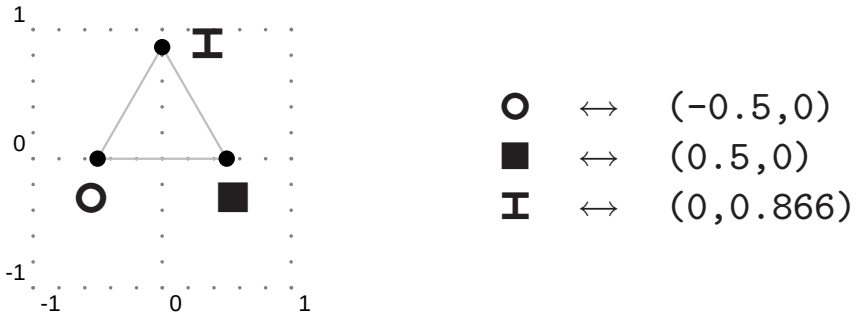

Figure 1: Representation of a categorical variable with three attributes in the 2D space by a standard regular simplex

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 $f(\boldsymbol{x})$ and the other part is the weight vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_M]$. The idea is to approach these two parts of the optimization process in one step, based on the parametric dependence idea.

The resulting multi-kernel is expressed by Eq.(8):

$$\begin{aligned}\tilde{K}(x_i, x_j) &= < \Phi(x_i), \Phi(x_j) > \\ &= \mu_1 < \phi_1(x_i), \phi_1(x_j) > + \mu_2 < \phi_2(x_i), \phi_2(x_j) > + \ldots \\ &+ \mu_M < \phi_M(x_i), \phi_M(x_j) > \\ &= \mu_1 K_1(x_i, x_j) + \mu_2 K_2(x_i, x_j) + \ldots + \mu_M K_M(x_i, x_j) \\ &= \sum_{s=1}^{M} \mu_s K_s(x_i, x_j)\end{aligned} \tag{8}$$

We can solve the regression hyperplane by plugging this multi-kernel on the equation defining the SVR regression surface, as Eq.(9) shows.

$$\hat{f}(\boldsymbol{x}) = b + \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) \tilde{K}(x_i, \boldsymbol{x}) \tag{9}$$

The specific nature of mixed variables has not been dealt with explicitly yet. In practice, regression models with mixed variables are often addressed by storing all the variables in a real design vector, including a conversion of the categorical inputs into (ordered) integers. While this conversion is straightforward for numerical and for ordinal parameters, nominal variables cannot be directly incorporated into a real vector representation without any loss of consistency. In order to avoid this issue, versatile methods employed to vectorizing these categorical cases have been proposed [9].

The *regular simplex* method is one of the simplest approaches to perform the conversion from categories to numbers. The basic idea is to assume that any two distinct levels of a categorical variable are separated by the same distance. 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 [3]. For simplicity, the distance between levels is assumed to be 1. For example, if x^{categ} can take value in a set of $n_{attr} = 3$ possible attributes $\{\mathbf{O}\,; \blacksquare\,; \mathbf{I}\}$, 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 [4].

In a rather similar way, *dummy coding* converts a 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. For the example described earlier, the conversion would be as follows: $\{\mathbf{O}\} \rightarrow (1, 0)$; $\{\blacksquare\} \rightarrow (0, 1)$; $\{\mathbf{I}\} \rightarrow (0, 0)$. 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 we should take into account in order to control undesirable effects in its use.

Anyway, despite making the correct conversions to vector of categorical variables, it still consist in data structured in some different way that the directly continuous inputs. Therefore, the proposal is to maintain the memory about this distinction by applying different kernel matrices depending on the nature of the variables. Thus, for straight numeric data, the Gaussian family of kernels usually fits better than others; nevertheless, for data resulting

from a conversion to numerics, other types of kernels (based on polynomials) present more accurate results. Table 1 summarizes some instances of kernels. The case of MKr is of special interest because it allows to separate via different kernels the straight numeric data and the converted nominal data.

Table 1: Short list of different kernel functions

Name	Expression
Gaussian	$K(x,y) = \exp\left(-\frac{\|\|x-y\|\|^2}{2\sigma^2}\right)$
ANOVA	$K(x,y) = \sum \exp\left(-\sigma(x^k - y^k)^2\right)^d$
Linear	$K(x,y) = x^T y + c$
Polynomial	$K(x,y) = (\alpha x^T y + c)^d$
Rational Quadratic	$K(x,y) = 1 - \frac{\|\|x-y\|\|^2}{\|\|x-y\|\|^2+c}$

Other kernels specific for structures such as trees, strings, and graphs, among others, have been proposed in the literature, but will not be considered in this paper, where only the most general-purpose kernels are investigated, and where the main concern is to address the vector conversion of categorical input and its posterior influence in the kernel selection for MKr modeling.

5. Numerical results

5.1 Analytical benchmark functions

The MKr for mixed variables approach is tested first on a set of six mixed-variable benchmark functions, with *ordinal variables*. The design variables consist of nz continuous and nc discrete variables (for all examples: $nz = nc$). The complete definition of the benchmarks is furnished in Table 2.

To analyze the evolution of the approximation/interpolation efficiency with respect to the dimension of the problem, the number of design variables ranges from $nz = nc = 2$ to $nz = nc = 20$. In all cases the data are composed of 5000 samples: 4000 for training, and 1000 reserved for the validation. In order to add statistical significance, each experiment is repeated 20 times.

Table 2: Definition of the six analytical benchmarks with mixed variables

Output		
$f_{\text{Ellipsoid,MV}}$	$=$	$\sum_{i=1}^{nz}\left(\beta^{\frac{i-1}{nz-1}} z_i\right) + \sum_{i=1}^{nc}\left(\beta^{\frac{i-1}{nc-1}} c_i\right) \qquad (\beta = 5)$
$f_{\text{Ackley,MV}}$	$=$	$-20e^{-0.2\sqrt{\frac{1}{nz}\sum_{i=1}^{nz} z_i^2}} - e^{\frac{1}{nz}\sum_{i=1} cos(2\pi z_i)}$ $-20e^{0.2\sqrt{\frac{1}{nc}\sum_{i=1}^{nc} c_i^2}} - e^{\frac{1}{nc}\sum_{i=1} cos(2\pi c_i)} + 20 + e$
$f_{\text{Rastrigin,MV}}$	$=$	$10(nz+nc) + \sum_{i=1}^{nz}\left[z_i^2 - 10cos(2\pi z_i^2)\right]$ $+\sum_{i=1}^{nc}\left[c_i^2 - 10cos(2\pi c_i^2)\right]$
$f_{\text{Rosenbrock,MV}}$	$=$	$\sum_{i=1}^{nz-1}\left[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2\right]$ $+\sum_{i=1}^{nc-1}\left[100(c_{i+1} - c_i^2)^2 + (c_i - 1)^2\right]$
$f_{\text{Sphere,MV}}$	$=$	$\sum_{i=1}^{nz} z_i^2 + \sum_{i=1}^{nz} c_i^2$
$f_{\text{Griewank,MV}}$	$=$	$\frac{1}{400}\sum_{i=1}^{nz} z_i^2 - \prod_{i=1}^{nz} cos\left(\frac{z_i}{\sqrt{i}}\right) + \frac{1}{400}\sum_{i=1}^{nc} c_i^2 - \prod_{i=1}^{nc} cos\left(\frac{c_i}{\sqrt{i}}\right)$
Input		
Cont. vars.		$z_i = 10^{-3} x_i^{cont}, \quad x_i^{cont} \in [-300, 700]$ for $i = 1, \ldots, nz$
Categ. vars.		$c_i \in [-3, -2, \ldots, 7]$ for $i = 1, \ldots, nc$

The six functions defined are coded by direct numeric conversion of their ordinal inputs; indeed, their direct numeric conversion has shown to work better than dummy coding, which is related to the inherent ranking of the ordinary inputs. A comparison between MKr and SVR based on RMSE (with respect to the number of input variables) is depicted in Figures 2 to 4; additionally, second-order ordinary least squares serve as reference method. Gaussian kernel has been implemented in SVR models. In the case of MKr, a Gaussian kernel is selected for the continuous kernels, while the categorical variables are modeled either by Polynomial (in Ellipsoid, Griewank, and Sphere functions) or by Rational Quadratic functions (in Ackley, Rastrigin, and Rosenbrock functions). These kernels were selected within a number of possible choices summarized in Table 1; the RMSEs of part of their possible combinations are shown in Table 3. This table is furnished by averaging the results obtained for all values of $nz = nc = 2, 5, 10, 15, 20$ and for each function and combination of kernels (Pol. stands for Polynomial, Gauss. for Gaussian, and R. Q. for Rational Quadratic).

From Figures 2 to 4, the following statements can be done. First, the special error behavior in Ellipsoid and

Table 3: RMSE (average for all $nz = nc$) for some kernel combinations in MKr

MKr composition	Ellipsoid	Ackley	Rastrigin	Rosenbrock	Sphere	Griewank
Pol. + Gauss.	**0.0035**	0.0174	0.0263	0.0623	**0.1831**	**0.0270**
R. Q. + Gauss.	0.0038	**0.0137**	**0.0182**	**0.0525**	0.2413	0.0667
Gauss. + Gauss.	0.0045	0.0229	0.0139	0.0683	0.1828	0.0504
Pol. + Pol.	0.0172	0.0431	0.0258	0.0744	0.2483	0.0435
R. Q. + R. Q.	0.0212	0.0398	0.0209	0.0697	0.3122	0.0494

Sphere functions are due to this similar construction to second order OLS expressions (e.g. theoretically, OLS can fit exactly the Ellipsoid function through a second-order polynomial). In the rest of the examples, OLS usually fits worst than the others, and MKr provides competitive results. We note that MKr is better than SVR in 4 over 5 cases, and in the sixth case (Sphere), both offer similar results. In the 4 cases in which MKr beats SVR, the models reveal a parallel trend in their behavior. The difference that separates both can be attributed to the choice of the kernel associated with the categorical input. In order to measure this difference we propose to use the ratio between the cumulative sum of RMSE_{SVR} over RMSE_{MKr}. As a result we see that MKr represents a gain–with respect to SVR–of 1.30, 1.69, 1.36, and 1.97 in the cases of Ellipsoid, Ackley, Rosenbrock, and Griewank functions, respectively. Finally, we also should note that if we choose Gaussian for both kernels in MKr we obtain similar (not exactly the same due to the parameter choices) results as SVR (see Table 3).

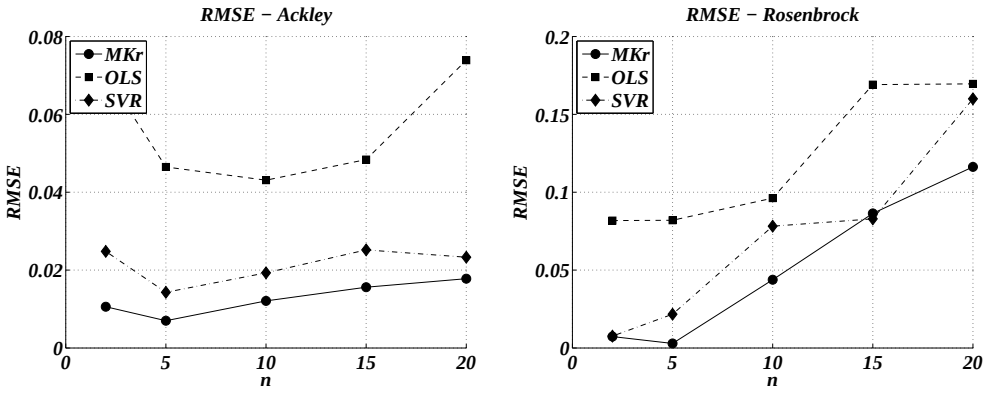

Figure 2: Comparison of regression models by the RMSE of the benchmark functions (Ackley & Rosenbrock)

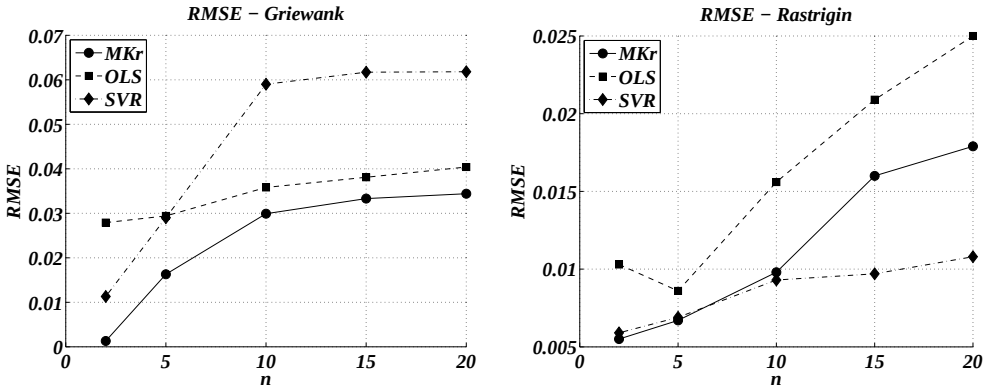

Figure 3: Comparison of regression models by the RMSE of the benchmark functions (Griewank & Rastrigin)

5.2 Structural analysis of a rigid frame

The second example is based on the structural design analysis of a 3D rigid frame [10] (see Figure 5). The loads of the structure are derived from Eurocode 3 [11], and consist in:

- the dead load of the beams and columns;

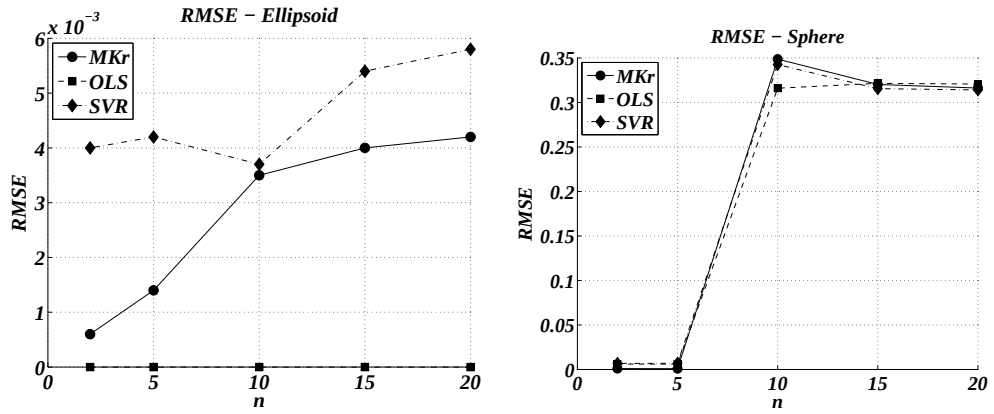

Figure 4: Comparison of regression models by the RMSE of the benchmark functions (Ellipsoid & Sphere)

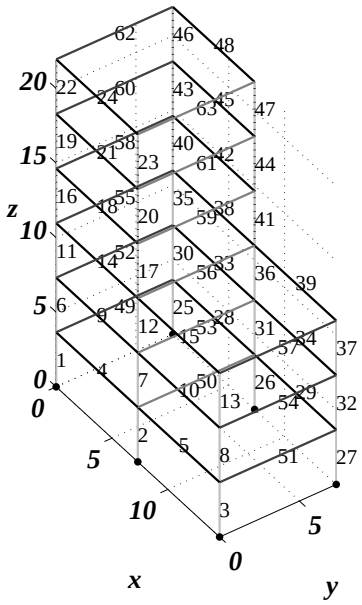

Figure 5: Rigid frame: geometrical configuration

- the gravity load on the floors (19.16 kPa);
- the lateral load due to the wind (110 kN).

The beams and columns are classified in five groups of common cross-sections. The quantities of interest are the total mass of the structure and the strain energy, the latter being assessed through a geometrically non-linear finite element analysis [12]. Seven design variables are necessary for the parametrization:

- for each of the five groups of profiles, a categorical variable defines the cross-section geometry among seven attributes { □ ; ○ ; I ; ■ ; ● ; ▯ ; ▮ };
- for all groups of profile, two continuous bounded variables define the maximum length l of the cross-section (either height or diameter), and the thickness t, with 0.09 m $\leq l \leq$ 0.11 m and 0.00225 $m \leq t \leq$ 0.00275 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.

The geometry of the cross-section is typically a nominal variable, since no ordering of the available cross-section 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. The data are composed by 5000 samples: 4000 for training and validating and 1000 reserved for test.

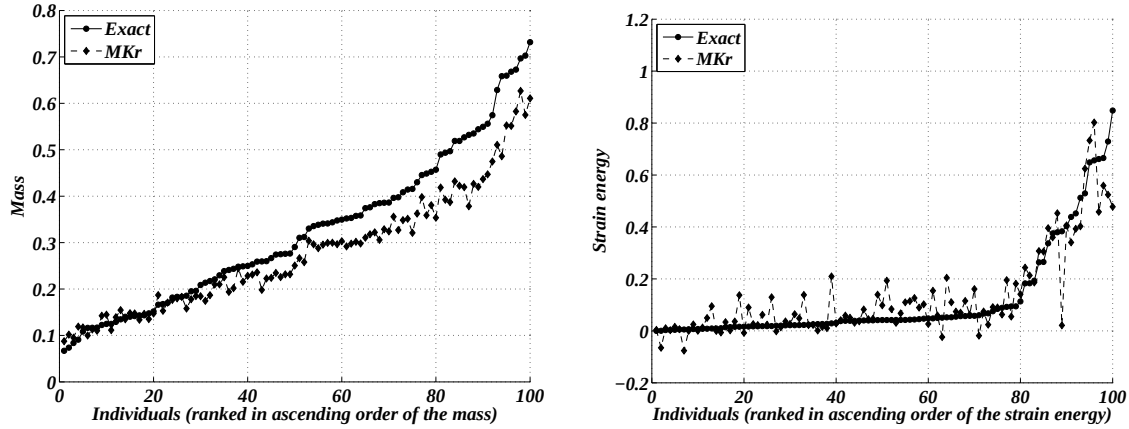

Figure 6: Observed quantity of interest vs. MKr prediction for the first 100 values of the validation database (left: mass; right: strain energy

For both quantities of interest (mass and strain energy), the direct numeric conversion is not consistent to represent the nominal (unordered) information of the database. Dummy coding is a more accurate proposal as demonstrated in Table 5.

Table 4: Dummy codes for the structural design case-study

□	→	(1, 0, 0, 0, 0, 0)
○	→	(0, 1, 0, 0, 0, 0)
⌶	→	(0, 0, 1, 0, 0, 0)
■	→	(0, 0, 0, 1, 0, 0)
●	→	(0, 0, 0, 0, 1, 0)
▫	→	(0, 0, 0, 0, 0, 1)
▮	→	(0, 0, 0, 0, 0, 0)

Table 5: RMSE for total mass and strain energy predictive models

Model	OLS	SVR	MKr	OLS	SVR	MKr
Output	Total mass			Strain energy		
RMSE (dummy coding)	0.0661	0.0625	**0.0440**	0.1442	0.1081	**0.0889**
RMSE (effect coding)	0.0682	0.0649	0.0539	0.1442	0.1128	0.0904
RMSE (real number conversion)	0.1001	0.0824	0.0588	0.1752	0.1604	0.1212

Figure 6 (left) shows the first 100 predictions respect the observed values of the total mass of the structure studied. A good agreement is observed, especially for lower values of the mass.

A parallel analysis has been carried out to predict the strain energy (see Figure 6, right). The results with the dummy coding, the effect coding, and the real number conversion are shown in Table 5. The MKr approach is the better predictive model again. We should note that other codings than dummy coding do not offer high variations with respect to the RMSE associated with MKr.

6. Conclusion

In this paper, a multiple kernel regression (MKr) method has been developed for continuous and categorical variables. MKr approaches attempt to improve the usual kernel methods in order to achieve a better adaptation to problems involving mixed-variable data.

From the numerical results obtained, it is demonstrated that MKr outperforms other methods in the structural design case study, and also provides promising results in the mixed-variable benchmark functions. MKr is

computationally more efficient than SVR, in terms of execution time, iterations, and number of support vectors. Additionally, in the numeric conversion from categorical nominal variables, dummy coding performs better than direct real number conversion for nominal inputs.

Future prospects include the application of these metamodels to surrogate-based optimization with mixed variables.

7. Acknowledgements

The authors are grateful to INNOVIRIS (Brussels-Capital Region, Belgium) for its support under a BB2B project entitled "Multicriteria optimization with uncertainty quantification applied to the building industry".

8. References

[1] P Breitkopf and R Filomeno Coelho, editors. *Multidisciplinary Design Optimization in Computational Mechanics*. ISTE/John Wiley & Sons, Chippenham, UK, April 2010. 1 volume, 549 pages.

[2] A Agresti. *An Introduction to Categorical Data Analysis*. John Wiley & Sons, New York, 1996.

[3] B McCane and M H Albert. Distance functions for categorical and mixed variables. *Pattern Recognition Letters*, 29(7):986–993, 2008.

[4] R Filomeno Coelho. 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, Austria, September 10-14*, 2012.

[5] R Filomeno Coelho. Metamodels for mixed variables based on moving least squares – Application to the structural analysis of a rigid frame. *Optimization and Engineering*, 2013. In press.

[6] B Schölkopf and A J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, USA, 2001.

[7] N Durrande, D Ginsbourger, O Roustant, and L Carraro. ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67, 2013.

[8] A J Smola and B Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[9] M Abramson, C Audet, and D E Dennis, Jr. Optimization using surrogates for engineering design, 2002.

[10] J Y Richard Liew, H Chen, N E Shanmugam, and W F Chen. Improved nonlinear plastic hinge analysis of space frame structures. *Engineering Structures*, 22(10):1324–1338, 2000.

[11] M Papadrakakis and N D Lagaros. Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 191:3491–3507, 2002.

[12] A J M Ferreira. *MATLAB Codes for Finite Element Analysis: Solids and Structures*. Solid Mechanics and Its Applications. Springer Science+Business Media B.V., 2009.