Stoch Environ Res Risk Assess (2011) 25:51–66
DOI 10.1007/s00477-010-0405-0

ORIGINAL PAPER

Learning wind fields with multiple kernels

Loris Foresti · Devis Tuia · Mikhail Kanevski · Alexei Pozdnoukhov

Published online: 3 June 2010

Abstract This paper presents multiple kernel learning (MKL) regression as an exploratory spatial data analysis and modelling tool. The MKL approach is introduced as an extension of support vector regression, where MKL uses dedicated kernels to divide a given task into sub-problems and to treat them separately in an effective way. It provides better interpretability to non-linear robust kernel regression at the cost of a more complex numerical optimization. In particular, we investigate the use of MKL as a tool that allows us to avoid using ad-hoc topographic indices as covariables in statistical models in complex terrains. Instead, MKL learns these relationships from the data in a non-parametric fashion. A study on data simulated from real terrain features confirms the ability of MKL to enhance the interpretability of data-driven models and to aid feature selection without degrading predictive performances. Here we examine the stability of the MKL algorithm with respect to the number of training data samples and to the presence of noise. The results of a real case study are also presented, where MKL is able to exploit a large set of terrain features computed at multiple spatial scales, when predicting mean wind speed in an Alpine region.

L. Foresti (✉) · D. Tuia · M. Kanevski
Institute of Geomatics and Analysis of Risk, University of Lausanne, Amphipôle Building, 1015 Lausanne, Switzerland
e-mail: Loris.Foresti@unil.ch

A. Pozdnoukhov (✉)
National Centre for Geocomputation, National University of Ireland, John Hume Building, Maynooth, Ireland
e-mail: Alexei.Pozdnoukhov@nuim.ie

Keywords Multiple kernel learning · Support vector regression · Feature selection · Wind resource estimation · Topographic features/indices extraction

1 Introduction

In recent years, machine learning algorithms (Bishop 2006; Hastie et al. 2009) have gained significant importance as a set of tools for modeling geo- and environmental spatio-temporal data (Kanevski 2008). These algorithms derive functional dependencies directly from observations thus allowing the data to speak for themselves without having recourse to physical models. Physical models are often computationally heavy to run, difficult to calibrate and they need complex schemes for assimilating the growing amounts of empirical data (Evensen 2006). Meanwhile, machine learning algorithms are applicable to a wide range of situations and problems when the exploration of empirical dependencies hidden in data is needed to infer a computational model. Due to the increased accessibility of real-time data, data-driven techniques provide an interesting way to approach these challenges.

The present research explores the use of a contemporary data-driven machine learning method applied to the spatial predictions of the long term average wind speed. Wind speed mapping is a fundamental task for natural resources evaluation, optimal allocation of wind farms and single turbines, climatological analysis in general and, particularly, for understanding the local topography-related patterns of wind speeds (Whiteman 2000). The complex non-linear relations with topography make wind speed prediction mapping in rough terrains a challenging problem for physical models and an interesting case study for data-driven statistical methods.

Most state-of-the-art models for evaluating long term wind speeds are based on physical-dynamical equations (Ayotte 2008; Ayotte et al. 2001; Baines 1997; Eidsvik 2005, Eidsvik et al. 2004; Franck et al. 2001; Gravdahl 1998; Palma et al. 2008). However, statistical data-driven models are rapidly emerging thanks to the increased data availability (Beccali et al. in press; Cellura et al. 2008; Liston and Elder 2006; Pozdnoukhov et al. 2007; Schaffner and Remund 2005). The choice of the nature of the model is driven by the quantity and the quality of data, the complexity of topography and the scale of analysis. Physical-dynamical models are often used for meso-scale modeling and statistical ones for micro-scale modeling to account for topographic influences (Petersen et al. 1998). An overview of different approaches to wind speed mapping is given by Landberg (2003).

Topographic information is of crucial importance for both statistical and physical models used to spatialize wind fields, especially in mountainous regions. The state-of-the-art statistical model developed for the Alpine region takes into account the contribution of topographic indices (Schaffner and Remund 2005). Relying on prior physical knowledge, the effects of terrain curvature and slope, the presence of lakes or canyons, are introduced in the model by adding ad hoc corrections. A linear regression model is calibrated then to introduce the mutual impact of these correction terms to the observed mean wind speed. In operational modeling of wind-related phenomena at regional scales (such as snow deposition and redistribution) topographic corrections are widely accepted as a baseline factor to account for (Liston and Elder 2006). It is interesting to note that even more complicated situations such as channeling and deflection can be approached by generating specific terrain indices/features related to these effects (Lindsay and Rothwell 2008).

In this paper, we propose a strategy based on automatic data-driven generation of terrain features for their use in statistical regression techniques adopted from machine learning. In this framework, topographic features are computed from the digital elevation models (DEM) of the terrain and are directly used in predictive regression models for mapping of environmental variables such as temperature (Pozdnoukhov et al. 2009), wind speed (Pozdnoukhov et al. 2009) or precipitation (Foresti et al., in press). In these studies, non-parametric data driven models such as artificial neural networks (ANN, Haykin 1999) and support vector regression (SVR, Smola and Schölkopf 1998) have shown excellent performances. However, the dimensionality of the input space of predictors composed of the extensive set of topographic features can become very large. Even though using more information may seem appealing and potentially useful, it also poses some hard problems and new challenges. The high number of redundant features induces collinearity problems and provokes the well-known overfitting phenomenon (Hughes 1968).

To avoid these undesired effects, an application of feature selection techniques (Guyon et al. 2006) can be foreseen. Feature selection allows the reduction of data dimensionality and gives insights about the phenomenon thanks to the analysis of the relevance of each contributing factor. Moreover, by the automatic relevance determination these techniques hopefully enhance performance as a consequence of the reduction of noise in data. Finally, if applied operationally, the model with a reduced set of features is computationally faster.

There are three groups of feature selection methods: *filter* methods ranking the features according to predefined relevance criteria such as correlation coefficient, *wrapper* methods involving the predictor as a part of the selection process by scoring the predictive power of features (for example, recursive feature elimination is a particularly popular method for support vector machines (SVM, Guyon et al. 2002)) and *embedded* methods, which are algorithm-specific, performing feature selection as a part of the training process.

In this paper we present a solution combining the efficiency of kernel methods (Schölkopf et al. 2002), which are among the most successful machine learning algorithms, and feature selection through the use of multiple kernel learning (MKL, Bach et al. 2004; Lanckriet et al. 2004). The method consists in building a kernel as a convex combination of basis kernels built using a single feature or meaningful sets of features. By attributing a single feature (or features subset) to a dedicated kernel, the general problem can thus be divided into a set of sub-problems which are expected to be simpler. Since a large number of parameters is involved, exhaustive search is computationally heavy and therefore several efficient optimization schemes have been proposed in the machine learning literature (Gönen and Alpaydin 2008; Sonnenburg et al. 2006; Zien and Ong 2007).

In this study, an efficient optimization scheme based on the recently proposed *SimpleMKL* algorithm (Rakotomamonjy et al. 2008) is used with support vector regression as a wrapper method for finding the optimal weighting of M basis kernels. First applications of this scheme have been recently proposed for remote sensing images (Tuia et al., in press), wind speed mapping (Foresti et al. 2009) and speaker verification (Longworth and Gales 2008) with promising results. Moreover, prior to the development of the SimpleMKL scheme, the MKL framework was already applied to the extraction of relevant genes from biological sequences (Rätsch et al. 2006; Sonnenburg et al. 2006). A related kernel-based model was investigated in Pozdnoukhov and Kanevski (2008) for modeling multiscale environmental data. There, an individual weight was assigned to each kernel for all the N samples, resulting in an optimization problem of NM

weights. This allows considering spatially-varying mixtures of kernels, but the optimization problem becomes intractable for large number of features and large data sets. In the current study, weights are assigned to each kernel for all the samples that is, $N + M$ weights in total. Although by using this approach one can not introduce a multi-scale model where the scales vary spatially, the computational load in training is reduced significantly.

In the present research performances of the conventional SVR and its MKL extension are compared on both simulated and real data, and the use of MKL is analyzed as a feature exploratory tool.

The scheme in Fig. 1 summarizes the applied modeling methodology. The framework proposed has a twofold objective: first, the generation of supplementary data using expert knowledge, and second the modeling of high-dimensional data via nonparametric data-driven approaches accounting for the relevance of input information.

The paper is organized as follows. After a brief introduction to statistical learning from data, Sect. 2 discusses the methods and the algorithms, with particular emphasis on the proposed MKL-based scheme. Section 3 describes the computation of topographic features from DEM used in this study. The experiments are then presented in the two following sections, first considering simulated topographic patterns (Sect. 4) and then using real data (Sect. 5). Finally, Sects. 6 and 7 summarize the main findings and conclude the paper.

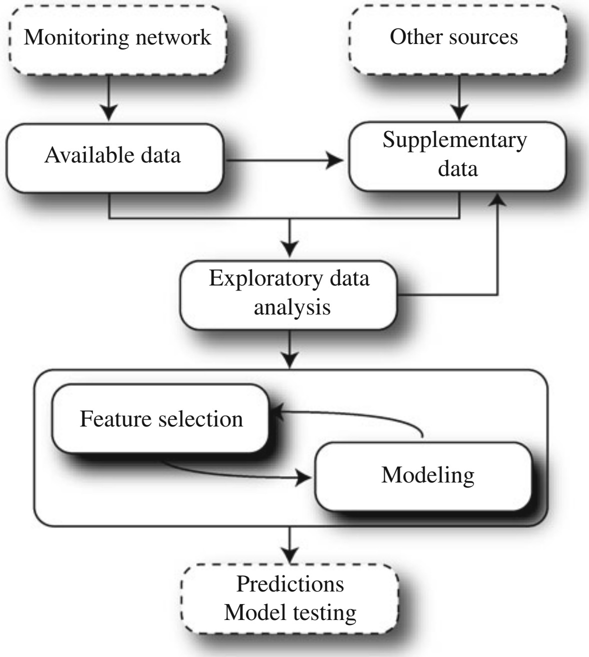


Fig. 1 Schematic outline of the proposed methodology

2 Machine learning algorithms

This section presents some basic concepts of machine learning and focuses on the support vector regression method used in the experiments. Afterwards, the general framework of multiple kernel learning and the optimization scheme used in this study are presented.

2.1 Statistical learning theory

Statistical learning theory (SLT) is a framework developed by Vapnik (1995) in order to assess and control the generalization capability of a statistical predictive model. SLT introduces the principle of structural risk minimization, which provides a constructive way for selecting models capable to generalize the observed dependence from empirical data. It consists in minimizing the bound on the (unknown) expected risk

$$R_{expected}(h) \leq R_{emp}(h) + R_{conf}(h) \qquad (1)$$

where R_{emp} is the empirical risk (computed using a loss function such as the mean squared error over training data), and R_{conf} is the confidence interval which penalizes excessively complex models. The generalization skills on new data are reached by controlling the model's complexity h. SLT has been introduced to work with finite datasets and does not need to take restrictive assumptions on the statistical distribution of data.

The aim of SLT is to find an optimum fit to training data and generalization capabilities (Fig. 2). Simple models (left side of Fig. 2) provide high empirical risk (they cannot fit to training data) but because of their simplicity they are not penalized. This situation is referred to as undertraining or underfitting. On the contrary, too complex models (right side of Fig. 2) result in low empirical risk. However, their expected risk will be high since they rely too much on the noisy and incomplete training set used resulting in a high generalization error. This situation is called overfitting or overtraining. Both overfitting and underfitting are not desirable because of their low generalization abilities. The optimal model lies in the middle of these two limit cases and corresponds to a compromise between model complexity and training error. A related notion depicting this situation is the bias-variance dilemma (Hastie et al. 2009).

2.2 Support vector regression

Support vector regression is a non-linear robust method for regression estimation (Smola and Schölkopf 1998). SVR intrinsically controls the complexity of the model according to SLT and provides accurate results when dealing with high-dimensional and noisy data.

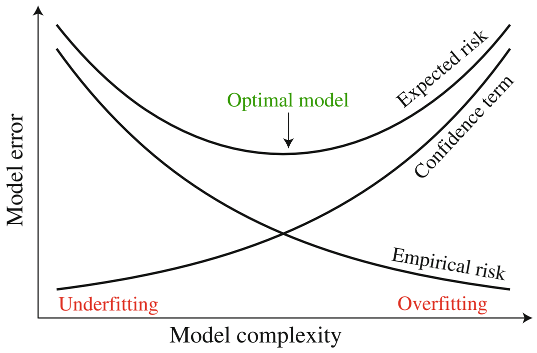


Fig. 2 Structural risk minimization: the optimal model is the one that minimizes the sum of the empirical risk and the confidence term

Given a set of N training data $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$ where $\mathbf{x}_i \in \mathcal{R}^D$ is the input feature vector and $y_i \in \mathcal{R}$ is the output, or target, SVR maps the data into a higher dimensional space where a linear regression $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ can be found. The linear regression problem is solved by defining the minimal width hyperplane that contains most of the observations within its margin (Fig. 3, right). This plane is uniquely defined by $\mathbf{w}$ and b and is determined geometrically by the samples lying on the borders of the margin. These points are called *support vectors* and are important training samples which are expected to give the most valuable information to solve the problem.

The mapping $\mathbf{x} \mapsto \varphi(\mathbf{x})$ into a higher dimensional space is achieved implicitly by applying a kernel function. These are symmetric positive-definite functions (Mercer 1905; Schölkopf 2001) representing dot products (a measure of similarity) between training pairs in a reproducing kernel Hilbert space (RKHS, feature space). This implicit mapping allows to find the SVR solution without computing the explicit mapping of data points. For an appropriately chosen kernel function there exists a linear regression in the related RKHS which translates into a non-linear solution in the original input space.

Since real data often contain outliers and highly noisy measurements, a "soft margin" version of SVR exists. During the optimization of the SVR weights, points far (at a distance of ξ_i) from the ε-tube can be penalized, providing a final regularized solution. The parameter which controls the degree of penalization of the solution is the C parameter. A high C means that the user is confident with the data and SVR will find complex solutions staying close to the observations. On the contrary, a low C leads to a function which remains as simple as possible and ignores data points that are far outside the ε-tube.

The soft margin SVR can be formulated as a constrained minimization problem:

$$\begin{aligned} \min_{f,b,\xi_i} \quad & \frac{1}{2}\|f\|^2 + C\sum_i(\xi_i + \xi_i^*) \\ s.t. \quad & y_i - f(\mathbf{x}_i) \leq \epsilon + \xi_i \quad \forall i, \\ & f(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i^* \quad \forall i, \\ & \xi_i \geq 0, \quad \xi_i^* \geq 0 \quad \forall i. \end{aligned} \tag{2}$$

The minimum is found by minimizing the squared norm $\|f\|^2$ of the regression function and the penalization term for data $\sum_i(\xi_i + \xi_i^*)$ lying outside the ε-tube (Fig. 3). The primal problem of Eq. 2 is solved in its dual form using Lagrangian multipliers. We present it directly substituting the dot products with a kernel function:

$$\begin{aligned} \max_{\alpha,\beta} \quad & \sum_i(\beta_i - \alpha_i)y_i - \epsilon\sum_i(\beta_i + \alpha_i) \\ & -\frac{1}{2}\sum_{i,j}(\beta_i - \alpha_i)(\beta_j - \alpha_j)K(\mathbf{x}_i, \mathbf{x}_j) \\ s.t. \quad & \sum_i(\beta_i - \alpha_i) = 0, \\ & 0 \leq \alpha_i \leq C, \quad 0 \leq \beta_i \leq C \quad \forall i. \end{aligned} \tag{3}$$

Finally, the SVR decision function is provided by the linear expansion of kernel functions $K(\mathbf{x}, \mathbf{x}_i)$:

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_i - \beta_i) \cdot K(\mathbf{x}, \mathbf{x}_i) + b, \tag{4}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are Lagrangian coefficients, nonzero only for support vectors. This way, the solution is sparse, because it does not depend on all the available training data, but only on important samples. Due to the convexity of the problem, as the kernel function is positive definite, the SVR solution is unique.

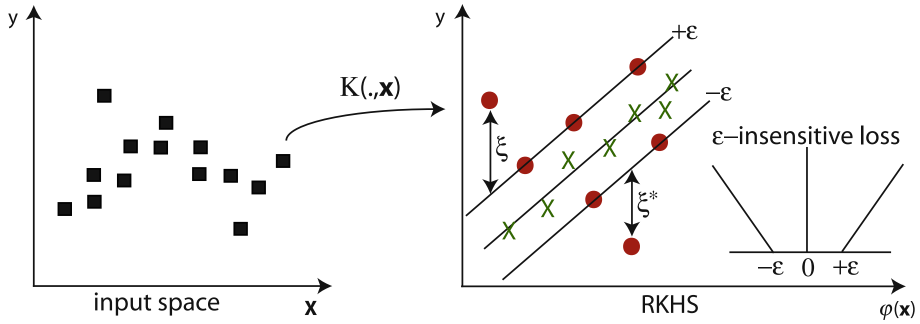


Fig. 3 SVR scheme. Support vectors are represented with dots; the noisy data inside the ε-tube (*crosses*) are not involved during the prediction part

Sparseness is a key feature of SVR and comes from the use of an ε-insensitive loss function (see Fig. 3). This property, together with linearity of the loss function, is partly responsible for the robustness of SVR model (Huber 1964) as only a reduced part of available data composed of support vectors is used.

To find the best function to model the data, at least three parameters have to be optimized: C, ε and the hyperparameters of the kernel. When large datasets are available, data can be split into training, validation and testing subsets. Training data are used to build the model, validation data are used to find optimal hyper-parameters (model selection) and testing data serve for a final evaluation of the generalization ability of the model (model assessment). Whenever data are scarce, cross-validation leave-k-out techniques are preferable in order to avoid problems of representativity.

2.3 Learning with multiple kernels

Often, the SVR problem of Eq. 3 is solved using closed form kernels such as the polynomial $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}\cdot\mathbf{x}' + 1)^p$ or the Gaussian (radial basis function, RBF) kernel $K(\mathbf{x},\mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}$ which maps the data into a potentially infinite dimensional space (Schölkopf et al. 2002). Such kernels are rigid representations of the data and may be replaced by more flexible and data-adapted kernels. The use of multiple kernels can enhance the performance of the model (described by Eq. 4) and, more importantly, the interpretability of the results. A multiple kernel in the sense of Lanckriet et al. (2004) is built by using a convex combination of basis kernels. In this case the kernel function $K(\mathbf{x}, \mathbf{x}')$ can be replaced by a convex linear combination of kernels

$$K(\mathbf{x},\mathbf{x}') = \sum_{m=1}^{M} d_m K_m(\mathbf{x},\mathbf{x}') \text{ with } d_m \geq 0 \text{ and } \sum_{m=1}^{M} d_m = 1 \tag{5}$$

where d_m are the weights associated to each kernel. For a given weight vector $\mathbf{d}$, the associated feature space is the sum of all feature spaces $\mathcal{H}_1, \ldots, \mathcal{H}_M$ for which $d_m > 0$. Multiple kernel learning aims at optimizing simultaneously the SVR coefficients α and β and the weights $\mathbf{d}$.

This formulation is very flexible and can be used in a variety of situations. For example, each kernel K_m can operate on the particular features predefined by the user, or a combination of features accounting for different properties of the dataset. Moreover kernels accounting for the same features, but using different kernel parameters, can be considered in order to model different length scales. When kernels are associated to single features MKL provides a basis for feature/kernel selection.

2.4 SimpleMKL for support vector regression

SimpleMKL (Rakotomamonjy et al. 2008) is a recently proposed efficient method for optimizing the weighted combination of kernels of Eq. 5. Similarly to Sonnenburg et al. (2006), SimpleMKL wraps an SVR solver considering the kernel of Eq. 5 as a fixed single kernel. A gradient descent on the SVR's objective function $J(\mathbf{d})$ in the space of kernel coefficients $\mathbf{d}$ is then iterated. The multiple kernel adaptation of the primal SVR problem Eq. 2 is:

$$\min_{\mathbf{d}} J(\mathbf{d}) \quad \text{such that} \quad \sum_{m=1}^{M} d_m = 1 \quad \text{and} \quad d_m \geq 0 \quad \forall m \tag{6}$$

$$J(\mathbf{d}) = \begin{cases} \min_{f,b,\xi_i} & \frac{1}{2}\sum_m \frac{1}{d_m}\|f_m\|^2_{H_m} + C\sum_i(\xi_i + \xi_i^*) \\ s.t. & y_i - \sum_m f_m(\mathbf{x}_i) - b \leq \epsilon + \xi_i \quad \forall i, \\ & \sum_m f_m(\mathbf{x}_i) + b - y_i \leq \epsilon + \xi_i^* \quad \forall i, \\ & \xi_i \geq 0, \xi_i^* \geq 0 \quad \forall i \end{cases} \tag{7}$$

This is basically the usual formulation of the SVR, except the function $f(\mathbf{x})$ which has been replaced by the linear combination of basis functions $\sum_m f_m(\mathbf{x})$. MKL optimization is done on two levels. The outer level optimizes the weights vector $\mathbf{d}$ (Eq. 6) while the inner level optimizes the SVR model function (Eq. 7). Sparseness of the final $\mathbf{d}$ vector is due to the l_1-norm regularization of the weights d_m which enhances feature/kernel selection skills.

The dual formulation of Eq. 7 can be derived and is similar to the dual formulation of the SVR problem in Eq. 3. The difference between the two formulations lies in the use of the linear combination of kernels $\sum_m d_m K_m(\mathbf{x}_i, \mathbf{x}_j)$. The update of the weights vector $\mathbf{d}$ to minimize $J(\mathbf{d})$ at the outer level is goverened by:

$$\frac{\partial J}{\partial d_m} = -\frac{1}{2}\sum_{i,j}(\beta_i^* - \alpha_i^*)(\beta_j^* - \alpha_j^*)K_m(\mathbf{x}_i, \mathbf{x}_j). \tag{8}$$

Using the per-component derivatives, a gradient direction is found for each component of the $\mathbf{d}$ vector. The final updating scheme for $\mathbf{d}$ is $\mathbf{d} \leftarrow \mathbf{d} + \gamma\mathbf{D}$, where γ is the step size and $\mathbf{D}$ is the descent direction computed using the reduced gradient algorithm (Faure 1965; Freund 2004). The reduced gradient used in Rakotomamonjy et al. (2008) allows to respect the equality and positiveness constraints of Eq. 6. The flowchart in Fig. 4 resumes the main SimpleMKL steps.

The magnitude of the coefficients d_m provides a criterion for feature selection and enhances the interpretability of the model. In the case where the basis kernels $K_m(\mathbf{x}_i, \mathbf{x}_j)$ operate only on predefined subsets or even individual input features, kernels with small (or null) d_ms, do not contribute to the solution. Then, in the sense of feature selection, the corresponding features can be omitted from the analysis.

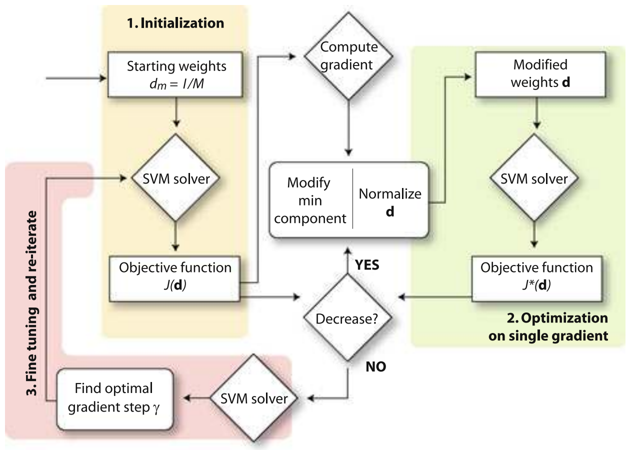


Fig. 4 Flow-chart of the SimpleMKL optimization

SVR and simpleMKL codes are freely available[1] (Canu et al. 2005).

3 Topographic feature extraction

The experiments presented in this paper deal with spatial predictions based on a small number of observations. Geostatistics provides a set of well-developed tools to approach such a task (Cressie 1993). However, X and Y coordinates[2] can fail at describing such complex phenomena. The dependencies to topography and the inherent nonlinearity of the phenomenon force an analyst to add knowledge to the model. This is usually done by introducing some primary (directly computed from digital elevation models) or secondary (process-specific values combining two or more primary attributes) topographic indices (Wilson and Gallant 2000) as predictors in a statistical regression model. A small number of fixed ad-hoc attributes are usually computed at the chosen spatial scale.

In this paper, we considered such information by adding topographic features extracted from the real terrain of the Swiss Alps. The DEM of Switzerland used in this study is available from the Swiss Federal Office of Topography. It has the resolution of 250 m. Topography-related features were computed from DEM using convolutional filters (Freeman and Adelson 1991) and stacked into a single vector following the three available features X, Y (coordinates) and Z (altitude). Three sets of features have been considered:

1. *Gaussian smoothing filters*. By subtracting two smoothed DEM surfaces obtained with different smoothing bandwidths, the ridges and canyons of different characteristic length scales are highlighted (Fig. 5). These features are referred to as Differences of Gaussians, DoG. The set of DoGs is generated by gradually increasing the widths of the smoothing kernels. The resulting set of features describes terrain convexity at different spatial scales.
2. *Terrain slopes*. The norm of the terrain gradient, which is proportional to slope, is computed at different scales on smoothed DEM surfaces.
3. *Directional derivatives*. The gradient is evaluated as for the slope, but only along specific directions.

One feature from each group is shown in Fig. 6. The resulting dataset is composed of 57 input features and 1 target variable: [X, Y, Z | 17 DoG | 21 Directional Derivatives | 16 Slopes | Wind Speed]. Consecutive features are correlated within each group since they are computed at close spatial scales.

4 Multiple kernel learning with simulated data

In the following section several experiments on simulated patterns providing a data-rich situation are led to highlight the properties of SimpleMKL and its behavior in different limit conditions. This study was aimed at investigating a simulated but realistic example of an environmental phenomenon hardly influenced by topographic features.

4.1 Preparation of the simulated datasets

The simulated patterns have been constructed by combining 3 among the 57 real terrain features described in Sect. 3. The particular features shown in Fig. 6 were used to compute the simulated target functions that reproduce an idealized topographic pattern useful for the MKL experiments.

The general formula used to compute the target function t is:

$$t = \tau(X, Y)[5 - 2 \cdot sig(f1) - 2 \cdot sig(f2) + 2 \cdot sig(f3)]$$

where $sig(f)$ is a sigmoid transformation of the feature values applied in order to have comparable value ranges and to reduce the effect of very high feature values.

$\tau(X, Y)$ is a function of X and Y coordinates accounting to the spatial variation of the magnitude of topography-induced relations. Four patterns of increasing complexity have been generated using the following τ:

[1] http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html.

[2] We refer to spatial coordinates when using uppercase X and Y; on the other hand, **x** is the input vector and y is the output.

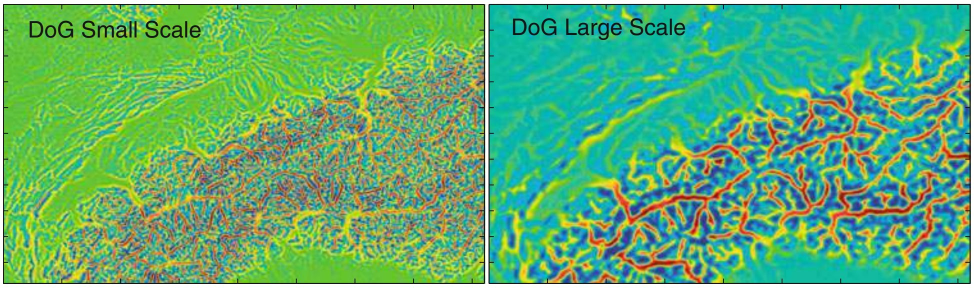


Fig. 5 Example of features computed at different scales (differences of Gaussians)

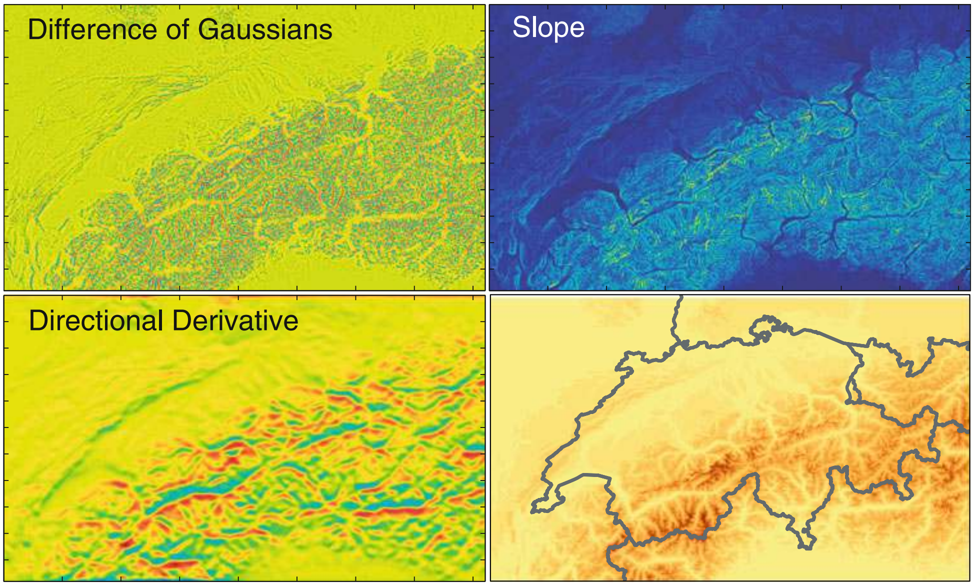


Fig. 6 Three topographic features that are combined to build the target function; *top left* (DoG, *f1*), *top right* (slope, *f2*), *bottom left* (directional derivative, *f3*). *Bottom-right*: Digital elevation model with country boundaries

$$\tau_{const}(X,Y) = 1$$

$$\tau_{lin}(X,Y) = \frac{8+X}{10}$$

$$\tau_{quad}(X,Y) = 1 - \frac{(X^2+Y^2)}{15}$$

$$\tau_{wave}(X,Y) = \frac{(2+(X+Y)e^{-(X^2+Y^2)})}{2}$$

The first pattern does not show variations in space, the second presents a linear trend in the East-West direction (X coordinate), the third one is characterized by a quadratic trend in both the X and Y coordinates and the fourth by a more complex non-linear trend. The aim of generating several patterns is to test MKL with increasing complexity of the input-output relationship. For the sake of convenience, we call the resulting patterns $\mathrm{Ptr}_{\mathrm{const}}$, $\mathrm{Ptr}_{\mathrm{lin}}$, $\mathrm{Ptr}_{\mathrm{quad}}$ and $\mathrm{Ptr}_{\mathrm{wave}}$ respectively. Figure 7 illustrates them.

A random spatial sampling is performed to extract training, validation and test (1000 samples) data subsets. In order to study the stability of MKL with respect to dataset size, training and validation sets of increasing size (10, 20, 50, 100 and 200 points) have been extracted. All the results report mean and standard deviation of testing performances based on 5 different splits of the training and validation sets.

In order to investigate the feature selection skills of MKL, three noisy features (Gaussian noise $N(0, 1)$) are added to the datasets. The final feature sets are detailed in Table 1. A total of $M = 8$ kernels applied on individual features are used in each experiment.

Four models are considered for each pattern. First, SVR with single linear ($\mathrm{SVR}_{\mathrm{lin}}$) and RBF ($\mathrm{SVR}_{\mathrm{RBF}}$) kernels and, secondly, MKL with linear ($\mathrm{MKL}_{\mathrm{lin}}$) and RBF ($\mathrm{MKL}_{\mathrm{RBF}}$) kernels. The corresponding kernels are:

$$K_{SVR-lin}(\mathbf{x},\mathbf{x}_i) = (\mathbf{x}\cdot\mathbf{x}_i)$$

$$K_{SVR-RBF}(\mathbf{x},\mathbf{x}_i) = e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}}$$

$$K_{MKL-lin}(\mathbf{x},\mathbf{x}_i) = \sum_{m=1}^{M} d_m(\mathbf{x}^m\cdot\mathbf{x}_i^m)$$

$$K_{MKL-RBF}(\mathbf{x},\mathbf{x}_i) = \sum_{m=1}^{M} d_m e^{-\frac{\|\mathbf{x}^m-\mathbf{x}_i^m\|^2}{2\sigma^2}}$$

where $\mathbf{x}_i^m$ is the mth component (feature) of the sample $\mathbf{x}_i$.

For the last pattern, an additional experiment named $\mathrm{MKL}_{\sigma XY}$ has been carried out: taking advantage of

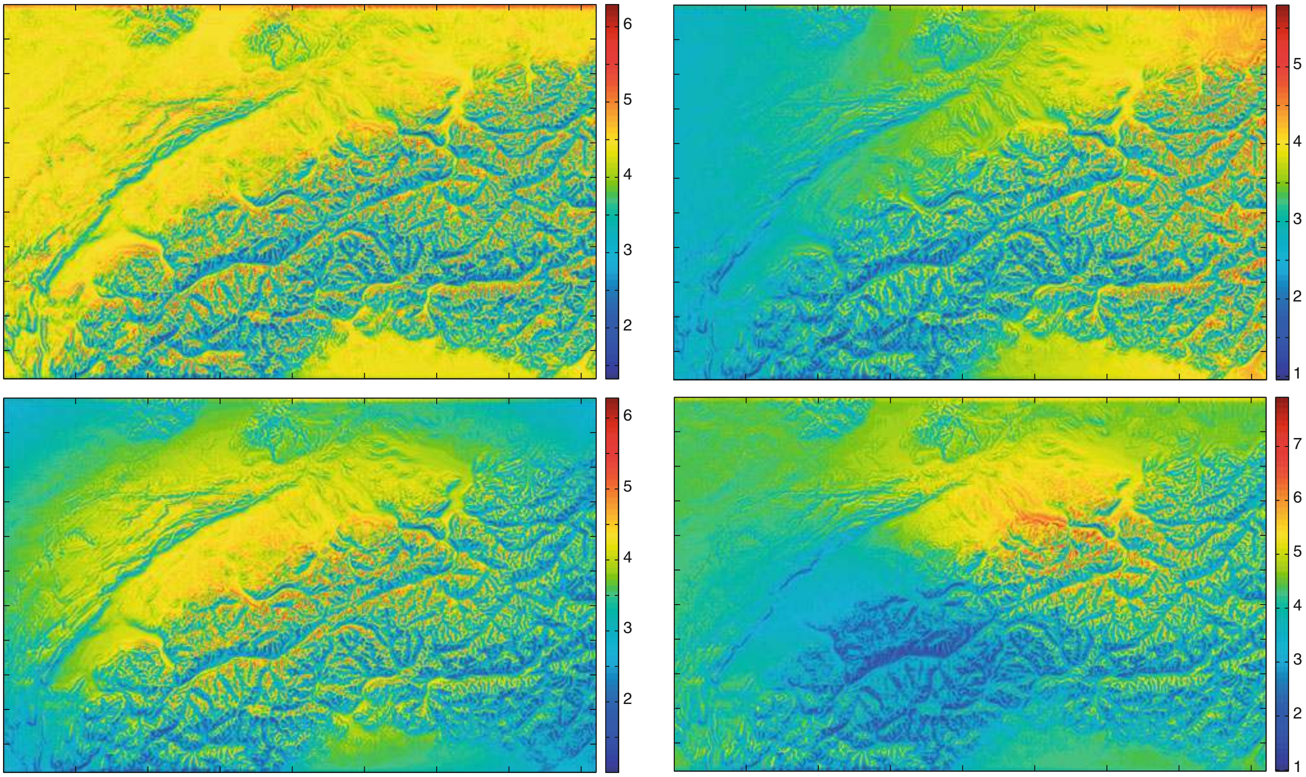


Fig. 7 Simulated patterns. *Top left*: $\mathrm{Ptr}_{\mathrm{const}}$, *top right*: $\mathrm{Ptr}_{\mathrm{lin}}$, *bottom left*: $\mathrm{Ptr}_{\mathrm{quad}}$ and *bottom right*: $\mathrm{Ptr}_{\mathrm{wave}}$

Table 1 Feature set considered in the simulated examples

Feature #	Description	Const	Lin	Quad	Wave
1	X	–	√	√	√
2	Y	–	–	√	√
3	**Noise**	–	–	–	–
4	DoG	√	√	√	√
5	**Noise**	–	–	–	–
6	Slope	√	√	√	√
7	**Noise**	–	–	–	–
8	Dir. derivative	√	√	√	√

Features # 3, 5 and 7 are noisy features that must be removed by MKL. √ = have to be selected by the MKL algorithm

knowledge about the pattern $\mathrm{Ptr}_{\mathrm{wave}}$, that depends jointly on the XY coordinates, this experiment optimizes a XY kernel. This kernel was built using both the spatial coordinates and encodes target dependencies on both the X and Y features:

$$K_{MKL_{\sigma XY}}(\mathbf{x}, \mathbf{x}_i) = d_{XY} e^{-\frac{(X-X_i)^2+(Y-Y_i)^2}{2\sigma_{XY}^2}} + \sum_{m=3}^{M} d_m e^{-\frac{\left\|\mathbf{x}^m - \mathbf{x}_i^m\right\|^2}{2\sigma^2}}$$

As explained in Sect. 2.2, model hyperparameters C, ε and σ for RBF kernel are optimized by minimizing the validation error.

4.2 MKL with increasing pattern complexity

Results obtained for the four simulated patterns are shown in Table 2. The decrease of performance with respect to increasing pattern complexity is clearly observable for all the models studied. Linear models perform well for the $\mathrm{Ptr}_{\mathrm{const}}$ and $\mathrm{Ptr}_{\mathrm{lin}}$ patterns, since they show no intrinsic nonlinearity. The improvements observed when using RBF kernels can be explained by a small level of non-linearity related to the sigmoid transformation applied to create the patterns. For the $\mathrm{Ptr}_{\mathrm{quad}}$ pattern, models using RBF kernels clearly outperform linear models, but MKL improves the SVR solution only slightly. The $\mathrm{Ptr}_{\mathrm{wave}}$ pattern shows the most interesting results: $\mathrm{SVR}_{\mathrm{RBF}}$ fails at describing the pattern, while the $\mathrm{MKL}_{\mathrm{RBF}}$ solution results in both lower RMSE and higher correlation. The $\mathrm{MKL}_{\sigma\ XY}$ experiment provides the best results for this pattern: although the striking result, we recall that this experiment is based on the integration of prior knowledge about the phenomenon that is not always available.

Table 3 shows the weights d_m after optimizing MKL algorithm, that is, it illustrates the features selected by the $\mathrm{MKL}_{\mathrm{RBF}}$. It selects the correct features (in Table 1) in every experiment: it ignores spatial coordinates for the $\mathrm{Ptr}_{\mathrm{const}}$ pattern, uses only the X coordinate for the $\mathrm{Ptr}_{\mathrm{lin}}$ and excludes noisy features in all the experiments.

Table 2 Test RMSE and correlation (ρ) for the four simulated patterns considered (training set size = 100)

Pattern	Method	Test RMSE		Test ρ	
		Mean	SD	Mean	SD
$\text{Ptr}_{\text{const}}$	SVR_{lin}	0.117	0.004	0.985	0.001
	SVR_{RBF}	0.074	0.006	0.994	0.001
	MKL_{lin}	0.115	0.006	0.986	0.002
	MKL_{RBF}	**0.019**	0.001	**0.9996**	3.8e-05
Ptr_{lin}	SVR_{lin}	0.124	0.004	0.980	0.001
	SVR_{RBF}	0.076	0.011	0.992	0.002
	MKL_{lin}	0.126	0.007	0.979	0.002
	MKL_{RBF}	**0.075**	0.013	**0.993**	0.002
Ptr_{quad}	SVR_{lin}	0.361	0.015	0.838	0.009
	SVR_{RBF}	0.084	0.010	0.991	0.002
	MKL_{lin}	0.354	0.013	0.842	0.007
	MKL_{RBF}	**0.072**	0.010	**0.994**	0.002
Ptr_{wave}	SVR_{lin}	0.475	0.013	0.851	0.006
	SVR_{RBF}	0.438	0.019	0.870	0.012
	MKL_{lin}	0.470	0.020	0.851	0.009
	MKL_{RBF}	**0.342**	0.028	**0.921**	0.014
	$\text{MKL}_{\sigma\text{XY}}$	**0.159**	0.019	**0.984**	0.004

Therefore, MKL is able to select the relevant features and to adapt the resulting kernel function to the specific problem.

4.3 MKL robustness to noise

In the previous section, the ability of MKL to perform feature selection was studied through the insertion of noisy features that the model should ignore. However, the three useful features in the dataset were noise free and the selection task was therefore facilitated. In the experiments presented below, noise has been added in the three features of Fig. 6. Three experiments accounting for increasing amounts of artificially generated noise have been considered: each variable has been contaminated with zero-mean Gaussian noise with standard deviation of 0.001, 0.01 and 0.1 respectively. Table 4 illustrates the numerical results. To avoid redundant tables, only the results of two patterns, the least ($\text{Ptr}_{\text{const}}$) and the most ($\text{Ptr}_{\text{wave}}$) complex, are presented. The numerical results of Table 2 are confirmed for both patterns, and the increasing noise does not strongly influence the test error. Thus, both SVR and MKL show a resistant behavior to noise.

Table 4 Test RMSE for the experiments with simulated noise (training set size = 100)

Pattern	Method	Noise = 0.001		Noise = 0.01		Noise = 0.1	
		Mean	SD	Mean	SD	Mean	SD
$\text{Ptr}_{\text{const}}$	SVR_{lin}	0.131	0.010	0.143	0.007	0.137	0.004
	SVR_{RBF}	0.073	0.004	0.084	0.007	0.123	0.004
	MKL_{lin}	0.127	0.006	0.141	0.008	0.133	0.002
	MKL_{RBF}	**0.032**	0.012	**0.038**	0.011	**0.084**	0.001
Ptr_{wave}	SVR_{lin}	0.449	0.010	0.451	0.011	0.432	0.015
	SVR_{RBF}	0.435	0.008	0.439	0.008	0.419	0.005
	MKL_{lin}	0.445	0.012	0.452	0.011	0.425	0.005
	MKL_{RBF}	**0.343**	0.013	**0.340**	0.017	**0.360**	0.034

Regarding feature selection, the first row of Fig. 8 shows the variability of feature weights for the three experiments using MKL_{RBF}: while an increase in the variance of the weights can be observed for higher levels of noise (dark boxes in the figure), MKL_{RBF} always selects the correct features for both patterns.

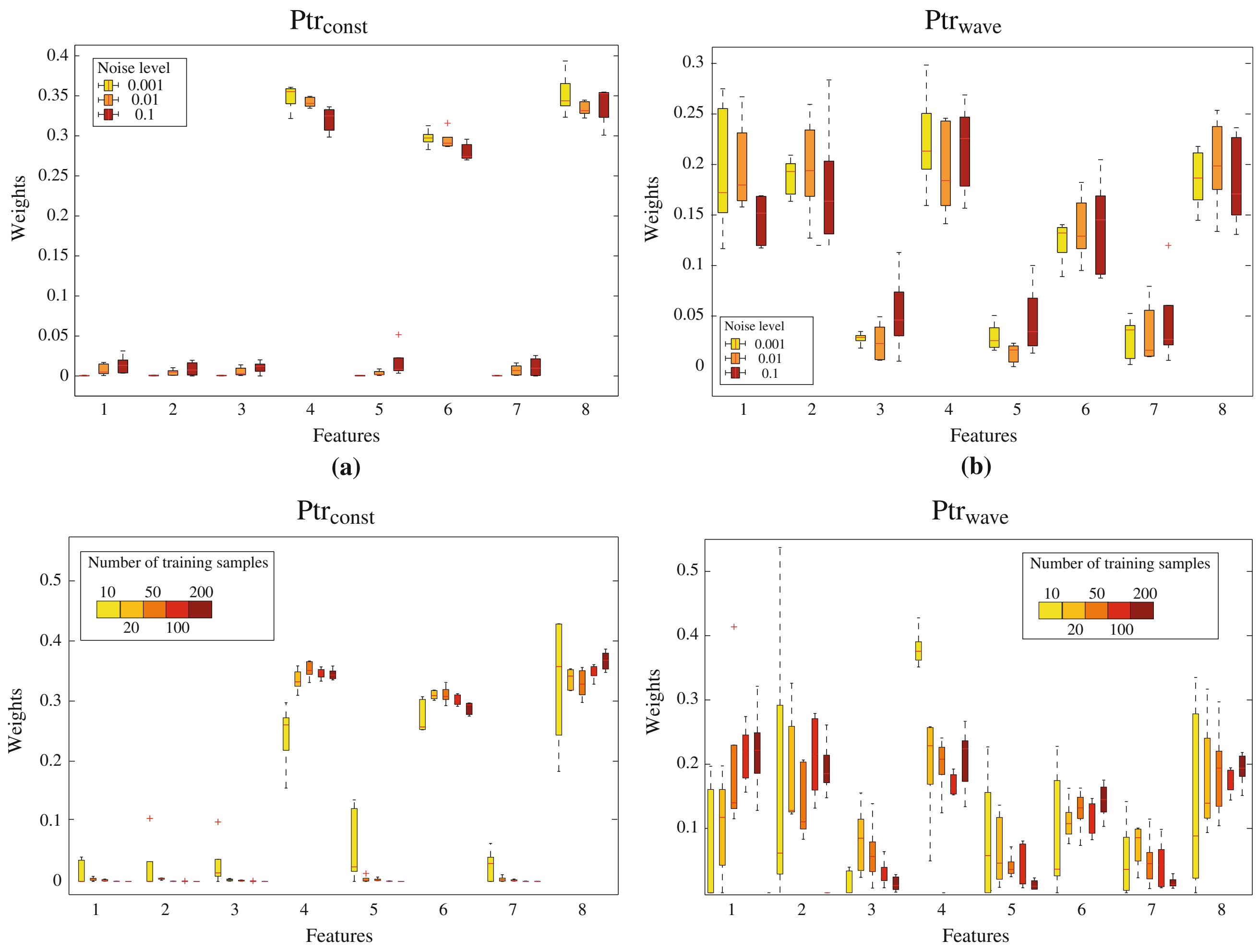
4.4 Dependence on dataset size

The quality and stability of the solution depends heavily on the number of training examples. This is important when MKL optimizes the weight vector **d** by gradient descent on the SVM decision function. Figure 9 illustrates the behavior of MKL trained on sets of different sizes for the

Table 3 Kernel weights for the const, lin, quad and wave experiments using the MKL_{RBF} approach (training set size = 100)

Pattern		Features							
		1	2	3	4	5	6	7	8
$\text{Ptr}_{\text{const}}$	Mean	0.0002	0.0004	0.0006	**0.3574**	0.0003	**0.2949**	0.0004	**0.3457**
	SD	0.0001	0.0002	0.0004	0.0145	0.0001	0.0130	0.0002	0.0038
Ptr_{lin}	Mean	**0.2753**	0.0191	0.0059	**0.2420**	0.0054	**0.2015**	0.0110	**0.2399**
	SD	0.0198	0.0154	0.0058	0.0155	0.0055	0.0202	0.0129	0.0059
Ptr_{quad}	Mean	**0.2252**	**0.2427**	0.0056	**0.1795**	0.0072	**0.1519**	0.0099	**0.1779**
	SD	0.0222	0.0208	0.0028	0.0055	0.0051	0.0121	0.0061	0.0088
Ptr_{wave}	Mean	**0.2378**	**0.2543**	0.0313	**0.1687**	0.0097	**0.1072**	0.0274	**0.1635**
	SD	0.0367	0.0487	0.0279	0.0294	0.0066	0.0121	0.0168	0.0488

In *bold* the features selected by SimpleMKL. Feature numbers are detailed in Table 1



(a)

(b)

(c)

(d)

Fig. 8 Boxplots of the optimized weights for different levels of noise (*top row*, training set size = 100) and different training set sizes (*bottom row*, noise level = 0.01) using $\mathrm{MKL}_{\mathrm{RBF}}$ on $\mathrm{Ptr}_{\mathrm{const}}$ (*left column*) and $\mathrm{Ptr}_{\mathrm{wave}}$ (*right column*). Variance of the weights is assessed over the 5 splits. Feature numbers are detailed in Table 1

$\mathrm{Ptr}_{\mathrm{const}}$ (Fig. 9a) and $\mathrm{Ptr}_{\mathrm{wave}}$ (Fig. 9b) patterns. For the two patterns considered, the mean RMSE of the five experiments decreases proportionally to the size of the training set. For both cases, 10 training points are not sufficient to obtain a stable solution and the observed standard deviation is very large. When using a larger training set, the problem is alleviated and from 100 training points on the solution becomes stable.

Considering the feature weights (second row of Fig. 8), the previous observations are confirmed: for 10 training points (light bars) the variance of the weights is stronger. If that does not affect the solution for the pattern $\mathrm{Ptr}_{\mathrm{const}}$, a strong confusion can be seen for the $\mathrm{Ptr}_{\mathrm{wave}}$, at the point that the algorithm starts selecting the noisy features. By increasing the number of training pixels (darker bars) the variance of the weights decreases and the solution is stabilized to the desired result.

5 MKL application to wind speed data

In this section, MKL is applied to the problem of spatial prediction of the mean wind speed in Switzerland. Data and exploratory data analysis are briefly presented in Sects. 5.1 and 5.2 respectively. The experiments are described in the following sections.

5.1 Preparation of the wind dataset

The mean wind speed above ground (period 1987–2006)[3] is sensed by 148 weather stations, either permanent or temporary at different heights above ground. To provide a coherent dataset, the measurements of the stations are

[3] More informations can be found on: The Swiss Wind Power Data Website, http://www.wind-data.ch/index.php.

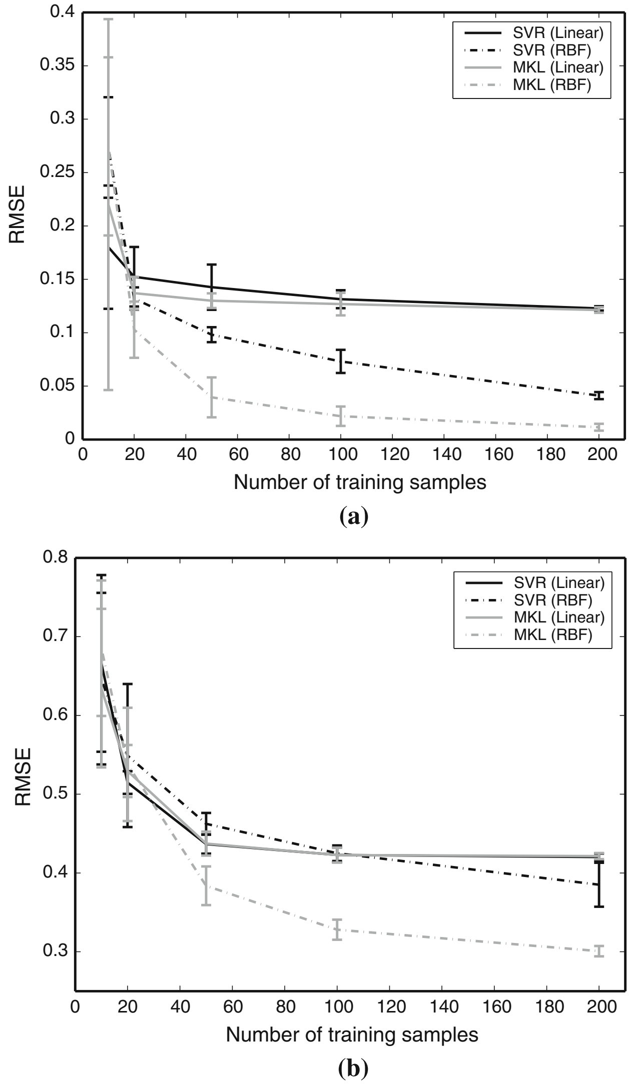


Fig. 9 Performance of SVR and MKL on **a** $\text{Ptr}_{\text{const}}$ and **b** Ptr_{wave} with respect to the size of the training set

extrapolated to the level at 50 m above ground. Such an extrapolation is obtained using a logarithmic wind profile according to the roughness length of different soil types (agricultural land, towns, bare soil, forests, etc). All these corrections were performed prior to the analysis. An interested reader can find additional details in Schaffner and Remund (2005).

The features used are the ones described in Sect. 3: coordinates, difference of Gaussians, slope and directional derivatives, for a total of 57 topographic features.

The data were split in two parts: a training set of 100 measurements and a test set of 48 measurements, used to estimate generalization performances of the model. Hyperparameters selection was carried out by 10-fold cross-validation.

5.2 Exploratory data analysis

Before building a model, the exploratory analysis of data (Andrienko and Andrienko 2006; Tuia and Kanevski 2008; Kanevski et al. 2009; Martinez 2004) allows to detect trends and extreme values, to estimate the amount of noise, etc. Summary statistics, variograms, dimensionality reduction techniques are necessary to have a first overview of the complexity of the data. Since machine learning methods are *data-driven*, the choice of the model must be done with respect to the complexity of the patterns and the size of the dataset. In the specific case of wind speed mapping, a linear model may fail because of the high dimensionality and non-linearity of the problem. Thus, support vector regression is a more adapted method for this task.

The postplot of wind speed values in the XY space (Fig. 10a) shows complex patterns and some high values, especially in the North-West regions. A variogram computed in geographical XY space (Fig. 11a) shows poor spatial structure, making the use of classical geostatistical and data-driven models in geographical space difficult. The introduction of topographic features brings additional useful structure to the data, as presented in both Figs. 10b and 11b showing respectively a post-plot and a variogram using the altitude (Z) and the feature #4 (a small-scale DoG). For instance, the plot of the data in this feature space shows a grouping of low wind speed values in the bottom-right corner (low elevation and positive convexity). Such structure was not visible using only XY coordinates and confirms the interest of using topographic features for wind mapping.

5.3 Mean wind speed prediction

Similarly to the experiments on simulated data, MKL has been compared to SVR. Models using only X, Y, Z coordinates have been considered in order to estimate a baseline performance and to compare them with the models built with the complete 57-dimensional dataset.

5.3.1 Numerical comparison

Numerical results of the experiments are given in Table 5. Comparison of the results gives rise to three main observations: first, none of the models using a linear kernel provide satisfying results. The presence of non-linearities in the wind-topography relationships can be the reason for such poor performances. Moreover, the difference observed when using the RBF kernels is significant. Second, the SVR and MKL performances on real data are almost equivalent: for all the experiments, SVR has performed slightly better than MKL, but the RMSE and ρ observed are in the same range for both models. Finally,

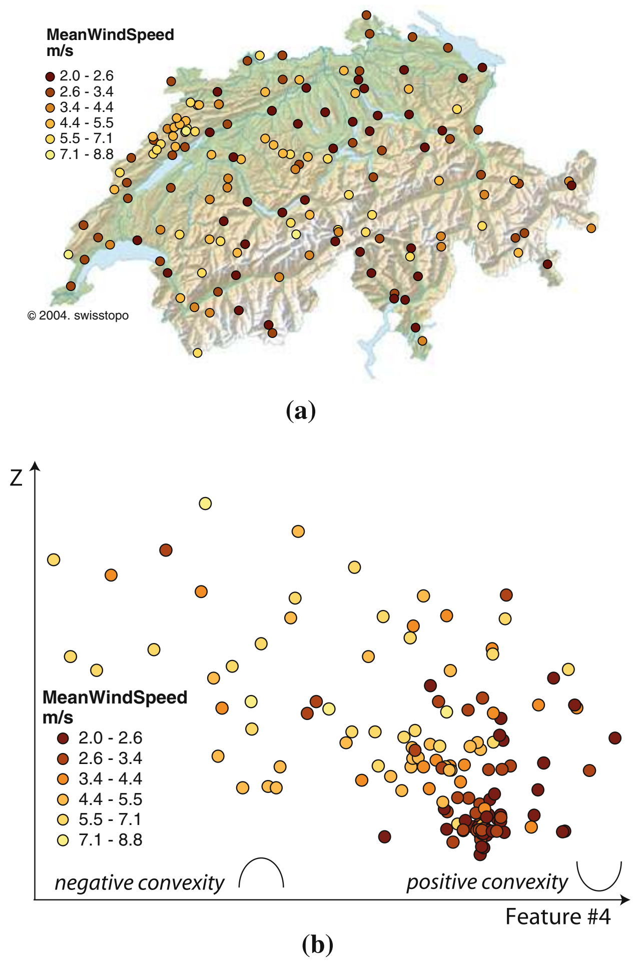


Fig. 10 Wind data postplot. **a** X and Y dimensions; **b** Z and Feature #4 (DoG, small scale)

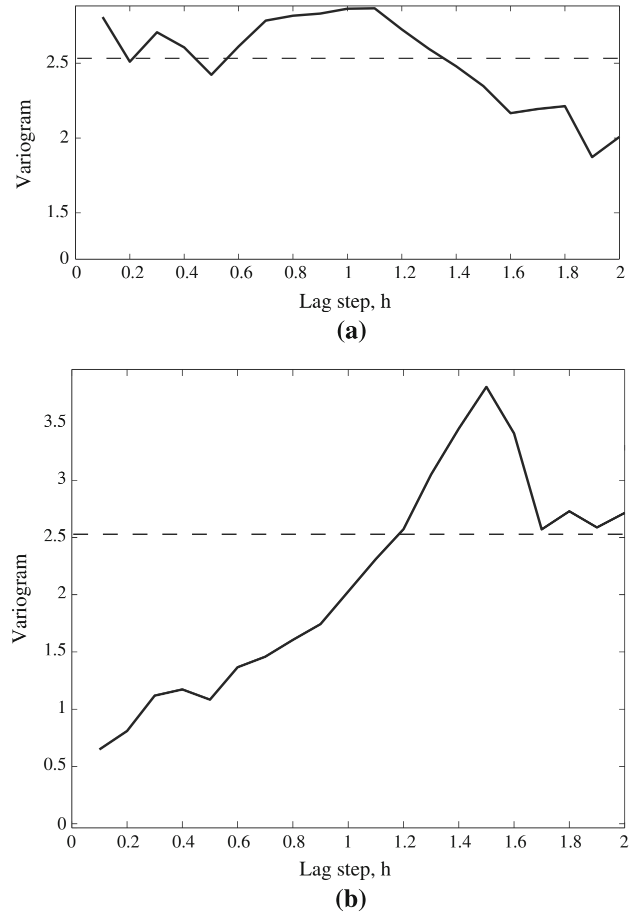


Fig. 11 Variograms in different dimensions. **a** X and Y dimensions; **b** Z and Feature #4 (DoG, small scale). The dashed line represents a-priori variance. Input data were normalized to N(0,1)

the use of topographic features strongly improves the quality of predictions in all the experiments. This is coherent with what was observed in Sect. 5.2.

5.3.2 MKL for feature selection

Figure 12 illustrates the mean weights $\bar{\mathbf{d}}$ averaged over 5 splits for the 57 features considered in the $\mathrm{MKL}_{\mathrm{RBF}}$ experiment. A total of 17 features has a mean weight greater than 0, while only 5 features show mean weights greater than 0.05. X and Z features are selected by each experiment, confirming the importance of these features. Surprisingly, the Y coordinate is never selected by the MKL algorithm. Among topographic features, the DoG are the most useful to model mean wind speed: DoG features are selected at each scale and 9 out of the 17 features with nonzero mean weights are of this kind. Slopes and directional derivatives are selected scarcely and at medium/large scale: these terrain features seem to be useful at the regional level only, while the DoG are used to model local relationships.

Table 5 Test RMSE and correlation (ρ) for the wind speed prediction considering different number of features (*= varies according to the number of features highlighted by $\mathrm{MKL}_{\mathrm{RBF}}$ in the single experiments). Results are averaged over 5 splits

Method	Feat.	Test RMSE		Test ρ	
		Mean	Std. dev.	Mean	Std. dev.
SVR_{RBF}	X, Y, Z	1.175	0.135	0.693	0.050
MKL_{RBF}	X, Y, Z	1.226	0.098	0.624	0.104
SVR_{lin}	57	1.278	0.114	0.651	0.027
SVR_{RBF}	57	**0.984**	0.106	0.782	0.061
MKL_{lin}	57	1.148	0.102	0.705	0.064
MKL_{RBF}	57	1.028	0.117	0.768	0.079
$\mathrm{SVR}_{RBF\text{-}0.01}$	*	1.009	0.139	0.770	0.061
$\mathrm{SVR}_{RBF\text{-}0.05}$	*	**0.984**	0.118	**0.789**	0.050

Two additional experiments, called $\mathrm{SVR}_{\mathrm{RBF}-0.01}$ and $\mathrm{SVR}_{\mathrm{RBF}-0.05}$, have been carried out using the features highlighted by $\mathrm{MKL}_{\mathrm{RBF}}$: in these experiments, SVR has been optimized using the features that received weights greater than 0.01 and 0.05 respectively after the MKL

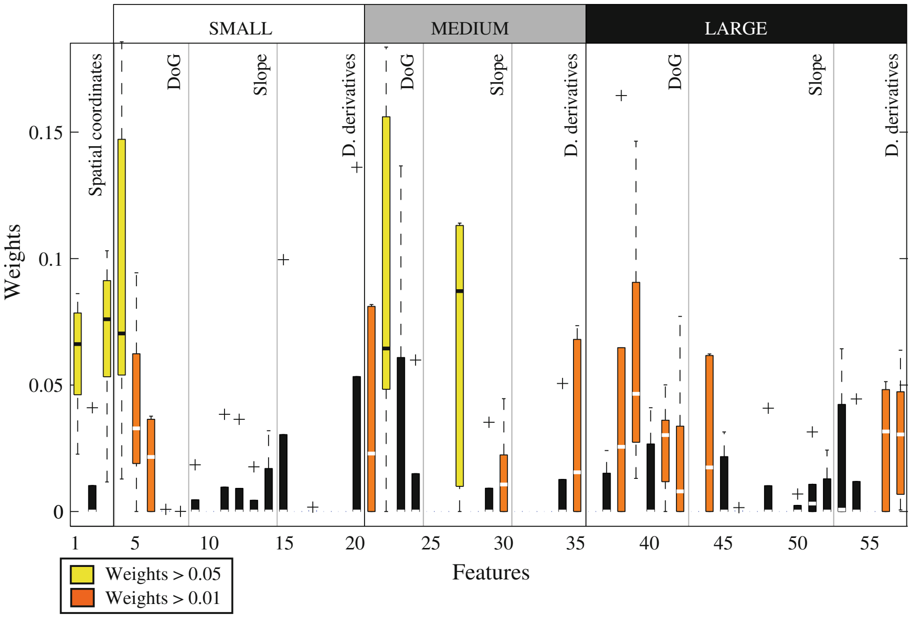


Fig. 12 Mean weights associated to each feature for the $\mathrm{MKL_{RBF}}$ experiments

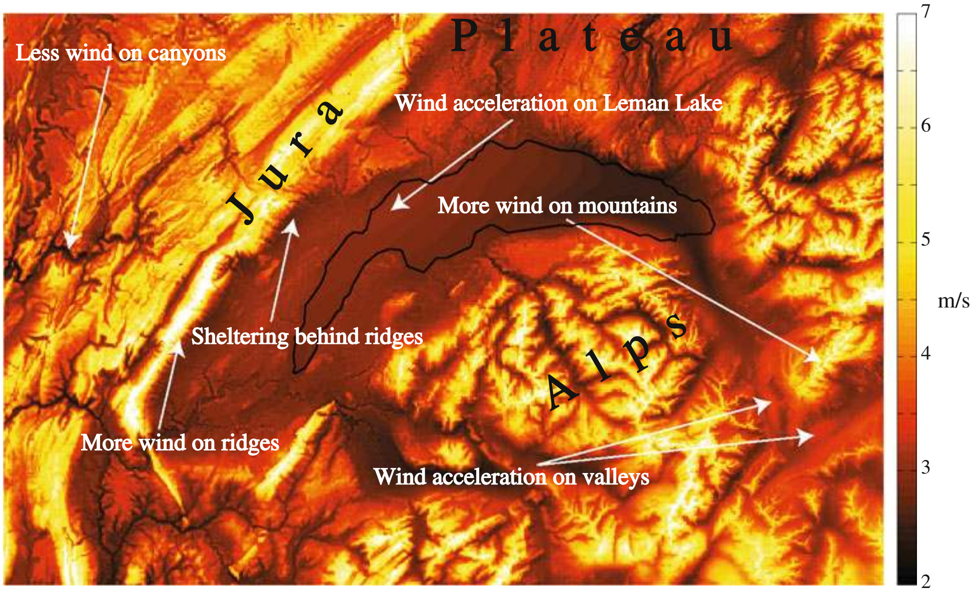


Fig. 13 $\mathrm{SVR_{RBF-0.01}}$ prediction of mean wind speed. In black is the edge of the Leman Lake

optimization. This way, MKL is used as a filter for feature selection. The aim of such study is to see if the subsets selected by MKL are indeed coherent to model the complex wind pattern. The mean results over the five runs are reported in the two last lines of Table 5. The test RMSE of the $\mathrm{SVR_{RBF-0.01}}$ shows a very small difference in the performance with respect to the model using the entire features set. Therefore, MKL has selected the relevant features that are used by SVR to model the wind pattern. The $\mathrm{SVR_{RBF-0.05}}$ experiment shows equal performance and this illustrates that the input space size can be reduced strongly without degrading the global quality of the prediction if the good features are highlighted.

Prediction map of the mean wind speed by the $\mathrm{SVR_{RBF-0.01}}$ model is shown in Fig. 13. A qualitative inspection of the resulting wind patterns provides useful insights to interpret the final model in terms of its physical consistency. Wind accelerations over ridges and mountains are well reproduced in the prediction map. The very low wind speed in narrow canyons is also visible. However, a more precise exploration allows to detect less evident wind patterns such as the wind acceleration in some valleys and the sheltering effects behind ridges (with respect to the predominant wind direction which is from west). Another surprising pattern is the wind acceleration over the west part of the Leman Lake. The channeling effect of the Swiss

Plateau is here braced because of the relative approaching of the Alps and the Jura Chain.

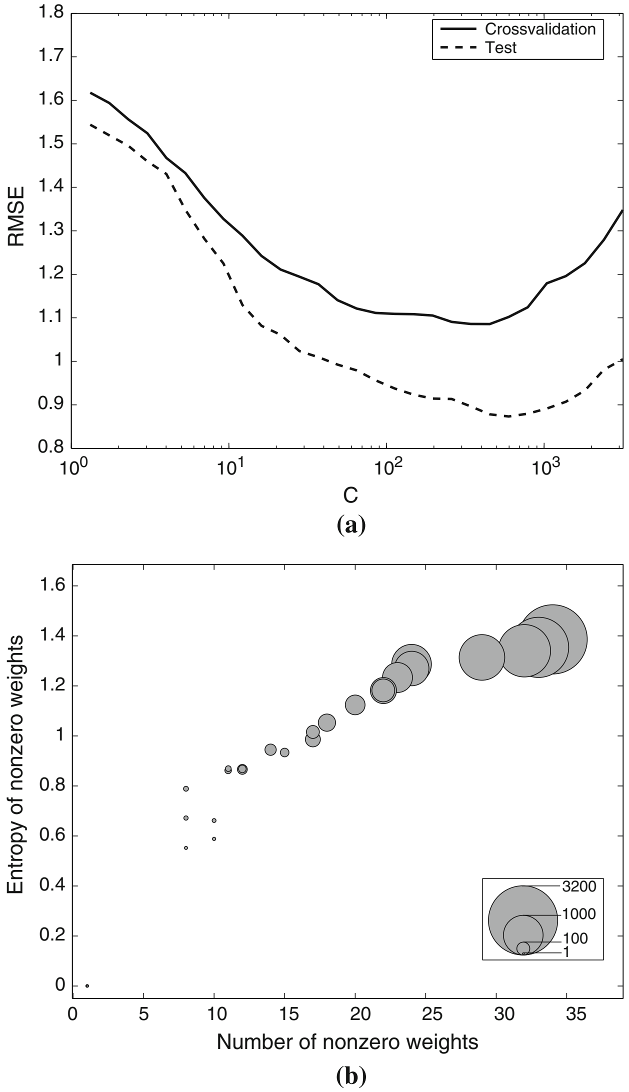
5.3.3 MKL and trade-off parameter C

In the experiments, the C parameter has been optimized by 10-fold cross-validation. The correct choice of this parameter is crucial for the regularization of the solution, as shown in Fig. 14a: the error is strongly influenced by the complexity of the solution that is regularized by this parameter. Alternatively, it also plays an important role on the control of the weight vector **d**. Figure 14b illustrates the effect of the parameter C on the sparsity of **d** and thus the number of features used in the prediction: the number of features selected by MKL is here plotted against the entropy of the weights attributed to their respective kernels and as a function of the value of the parameter *C* (shown by the size of the circles). This figure illustrates that the increase of the C value has a double effect: first, more variables are included in the final solution and second, the distribution of their weights tends to become more and more homogeneous, resulting in a higher entropy of the distribution of the weights. Thus, the value of C acts as a controller of the sparsity of the MKL solution, that is *de facto* a way of controlling the complexity of the model.

6 Advantages and limitations of multiple kernel learning

The flexibility of the MKL approach is by far its main strength. The user is allowed to divide the problem in several parts (features or feature groups) in order to have a better understanding about the contribution of each source of information. The model is efficient for medium-sized datasets and is resistant to noisy features. Finally, the biggest advantage of MKL compared with standard support vector regression is that the first gives insights about the importance of the features and, at the same time, provides a model, while the second works more like a black box.

Nonetheless, MKL also presents some limitations: the slowness in the optimization of the weights with respect to the size of the training set is the most striking. This is due to the repeated calls to the SVM solver during optimization. This problem can be approached with a more efficient implementation. Moreover, MKL has shown a tendency to overfit when only a few training data are available: it concentrated on short-scale features by missing global spatial relationships. Finally, the kernel combination proposed in this paper does not consider cross relations between inputs (cross-kernels) because each feature is mapped to an independent feature space: therefore, MKL should work at its best only when the target does not depend jointly on two or more features. The ideal example of improvement when using cross-kernels was the model MKL$_{\sigma XY}$ with simulated data (Ptr$_{\mathrm{wave}}$). In that case, instead of using two kernels, one for the X dimension and one for the Y dimension, the cross-kernel XY was used. The MKL$_{\sigma XY}$ model has shown an increase in performance in the complex pattern depending simultaneously on X and Y coordinates.



(a)

(b)

Fig. 14 Behavior of MKL while varying the trade-off parameter C. **a** Cross-validation and test RMSE for a single run of MKL; **b** number of nonzero weights and entropy of the weights in the **d** vector as a function of C (size of the *circles*, going from 1 (*smallest*) to 3200 (*biggest*))

7 Conclusions

Due to its robustness and suitability for working with high-dimensional input data for modeling non-linear dependencies, support vector regression provides good results in spatial prediction of the wind speed. In this paper we explored the use of the multiple kernel learning to enhance the interpretability of this kernel-based predictive model. MKL wraps an SVR trained with a linear combination of kernels and finds the optimal combination of input features. We applied it to the predictive mapping of wind speed aiming at detecting the optimal characteristic length scales of different topographic features influencing the phenomenon.

The empirical studies of the real data provided interesting insights about the use of the proposed approach for feature selection. MKL scheme was found to be successful in detecting meaningful features subsets. The sensitivity to hyper-parameters (particularly, the data fit vs. complexity trade-off parameter of SVR) in finding the optimal distribution of weights was investigated.

The definition of the optimal set of kernels (currently based on the prior knowledge) remains an open question and it is currently one of the limitations of the algorithm. Irrelevant kernel sets associated with difficult and small datasets may lead to overfitting as shown empirically in Lewis et al. (2006). Since the distance metric induced by real processes is often variable over the input space, the non-stationarity of kernel functions is also an important research issue. Future promising perspectives for environmental data modeling concern the use of MKL for integrating multisource data from monitoring networks, both for the modeling of joint multiscale physical processes and for automatic feature selection.

Acknowledgements The research is funded in part by the Swiss National Science Foundation projects “GeoKernels: kernel-based methods for geo- and environmental sciences (Phase II)” (No 200020-121835/1) and “Structured learning for remote sensing data analysis” (PBLAP2-127713/1). A. Pozdnoukhov acknowledges the support of Science Foundation Ireland under the National Development Plan, particularly through Stokes Award and Strategic Research Cluster grant (07/SRC/I1168). The authors would like to acknowledge Prof. S. Canu and Prof. A. Rakotomamonjy for the useful discussion and interesting comments.

References

Andrienko N, Andrienko G (2006) Exploratory data analysis of spatial and temporal data. Springer, NY

Ayotte KW (2008) Computational modelling for wind energy assessment. J Wind Eng Indus Aerodyn 96:1571–1590

Ayotte KW, Davy RJ, Coppin PA (2001) A simple temporal and spatial analysis of flow in complex terrain in the context of wind energy modeling. Boundary-Layer Meteorol 98:275–295

Bach FR, Lanckriet GRG, Jordan MI (2004) Multiple kernel learning, conic duality and the SMO algorithm. In: Proceedings of the 21th international conference on machine learning 69

Baines PG (1997) Topographic effects in stratified flows. Cambridge University Press, Cambridge

Beccali M, Cirrincione G, Marvuglia A, Serporta C (In press) Estimation of wind velocity over a complex terrain using the generalized mapping regressor. Applied Energy

Bishop C (2006) Pattern recognition and machine learning. Springer, NY

Canu S, Grandvalet Y, Guigue V, and Rakotomamonjy A (2005) SVM and kernel methods matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France

Cellura M, Cirrincione G, Marvuglia A, Miraoui A (2008) Wind speed spatial estimation for energy planning in Sicily: a neural kriging application. Renew Energy 33:1251–1266

Cressie N (1993) Statistics for spatial data, revised edn. Wiley, NY

Eidsvik KJ (2005) A system for wind power estimation in mountainous terrain. Prediction of Askervein hill data. Wind Energy 8:237–249

Eidsvik KJ, Holstad A, Lie I, Utnes T (2004) A prediction system for local wind variations in mountainous terrain. Boundary-Layer Meteorology 112:557–586

Evensen G (2006) Data assimilation: The ensemble Kalman filter. Springer, NY

Faure P, Huard P (1965) Résolution de programmes mathématiques à fonction non linéaire par la méthode du gradient réduit, Revue Française de Recherche Opérationnelle 36

Foresti L, Pozdnoukhov A, Tuia D and Kanevski M (In press) Extreme precipitation modelling using geostatistics and machine learning algorithms. Proceedings of the 7th international conference on geostatistics for environmental applications

Foresti L, Tuia D, Pozdnoukhov A, Kanevski M (2009) Multiple kernel learning of environmental data. Case study: analysis and mapping of wind fields. Proceedings of the 19th international conference on artificial neural networks, Part II, pp 933–943

Franck HP, Rathmann O, Mortensen NG, Landberg L (2001) The numerical wind atlas—the KAMM/WAsP method. Risoe National Laboratory publications, Danemark Risoe-R-1252(EN)

Freeman WT and Adelson EH (1991) The design and use of steerable filters. IEEE Trans Pattern Anal Mach Intel 13:891–906

Freund RM (2004) Solution methods for quadratic optimization. Technical report, Massachusetts Institute of Technology, MA

Gönen M, Alpaydin E (2008) Localized multiple kernel learning. Proceedings of the 25th international conference on machine learning, vol 307. pp 352–359

Gravdahl AR (1998) Meso scale modeling with a reynolds averaged navier-stokes solver: assessment of wind resources along the Norwegian coast. 31th IEA experts meeting. State of the Art on Wind Resource Estimation

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422

Guyon I, Gunn S, Nikravesh M, Zadeh LA (eds) (2006) Feature extraction: foundations and applications. Springer, NY

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer, NY

Haykin S (1999) Neural Networks. Prentice Hall, India

Huber PJ (1964) Robust estimation of a location parameter. Ann Math Stat 35(1):73–101

Hughes GF (1968) On the mean accuracy of statistical pattern recognition. IEEE Trans Inf Theory 14(1):55–63

Kanevski M (ed) (2008) Advanced mapping of environmental data. ISTE Wiley, NY

Kanevski M, Pozdnoukhov A, Timonin V (2009) Machine learning algorithms for spatial data analysis and modelling. EPFL Press, Lausanne

Lanckriet GRG, De Bie T, Cristianini N, Jordan MI, Noble WS (2004) A statistical framework for genomic data fusion. Bioinformatics 20(16):2626–2635

Landberg L, Myllerup L, Rathmann O, Petersen EL, Jorgensen BH, Badger J, Mortensen NG (2003) Wind resource estimation-an overview. Wind Energy 6:261–271

Lewis DP, Jebara T, Noble WS (2006) Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. Bioinformatics 22:2753–2760

Lindsay JB, Rothwell J (2008) Modelling channeling and deflection of wind by topography. In: Zhou Q, Lees B (eds) Advances in digital terrain analysis. Springer, NY, pp 383–406

Liston GE, Elder KA (2006) Meteorological distribution system for high-resolution terrestrial modeling (microMet). J Hydrometeorol 7:217–234

Longworth C, Gales MJF (2008) multiple kernel learning for speaker verification. IEEE conference on acoustic, speech and signal processing ICASSP, pp 1581–1584

Martinez WL (2004) Exploratory data analysis with matlab. Chapman & Hall/CRC, London

Mercer J (1905) Functions of positive and negative type and their connection with the theory of integral equations. Phil Trans R Soc CCIX:215–228

Palma JMLM, Castro FA, Ribeiro LF, Rodrigues AH, Pinto AP (2008) Linear and nonlinear models in wind resource assessment and wind turbine micro-siting in complex terrain. J Eng Indus Aerodyn 96:2308–2326

Petersen EL, Mortensen NG, Landberg L, Hojstrup J, Frank HP (1998) Wind power meteorology. Wind Energy 1:2–22

Pozdnoukhov A, Kanevski M (2008) Multi-scale support vector algorithms for hot spot detection and modelling. Stoch Environ Res Risk Assess 22(5):647–660

Pozdnoukhov A, Kanevski M, Timonin V (2007) Prediction of wind power density using machine learning algorithms. Proceedings of the 12th annual conference of international association for mathematical Geology

Pozdnoukhov A, Foresti L and Kanevski M (2009) Data-driven topoclimatic mapping with machine learning methods. Nat Haz 3(50):497–518

Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) Simple MKL. J Mach Learn Res 9:2491–2521

Rätsch G, Sonnenburg S, Schäfer C (2006) Learning interpretable SVMs for biological sequence classification. BMC Bioinformatics 7(Suppl 1):S9

Schaffner B, Remund J (eds) (2005) The alpine space wind map: modeling approach. Alpine Windharvest Report Series 7–2. Alpine windharvest partnership network

Schölkopf B (2001) The kernel trick for distances. In: Leen TK, Dietterich TG, and Tresp V (eds) NIPS. MIT Press, Cambridge, pp 301–307

Schölkopf B, Smola A (2002) Learning with Kernels. MIT Press, Cambridge

Smola A-J, Schölkopf B (1998) A Tutorial on support vector regression. NeuroCOLT2 technical report series, NC2-TR-1998-030

Sonnenburg S, Schaefer G, Rätsch G, Schölkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565

Tuia D, Kanevski M (2008) Environmental monitoring network characterization and clustering. In: Kanevski (ed) Advanced mapping of environmental data. ISTE Wiley, NY, pp 19–47

Tuia D, Camps-Valls G, Matasci G, Kanevski M (in press) Learning relevant image features with multiple kernel classification. IEEE Trans Geosci Remote Sens

Vapnik V (1995) The nature of statistical learning theory. Springer, NY

Whiteman CD (2000) Mountain meteorology: fundamentals and applications. Oxford University Press, Oxford

Wilson JP, Gallant JC (eds) (2000) Terrain analysis: principles and applications. Wiley, NY

Zien A, Ong CS (2007) Multiclass multiple kernel learning. Proceedings of the 24th international conference on machine learning, vol 227. pp 1191–1198