Structured nonlinear variable selection

Magda Gregorová **Alexandros Kalousis**
Geneva School of Business Administration, HES-SO, Switzerland
University of Geneva, Switzerland

Stéphane Marchand-Maillet
University of Geneva
Switzerland

Abstract

We investigate structured sparsity methods for variable selection in regression problems where the target depends nonlinearly on the inputs. We focus on general nonlinear functions not limiting a priori the function space to additive models. We propose two new regularizers based on partial derivatives as nonlinear equivalents of group lasso and elastic net. We formulate the problem within the framework of learning in reproducing kernel Hilbert spaces and show how the variational problem can be reformulated into a more practical finite dimensional equivalent. We develop a new algorithm derived from the ADMM principles that relies solely on closed forms of the proximal operators. We explore the empirical properties of our new algorithm for Nonlinear Variable Selection based on Derivatives (NVSD) on a set of experiments and confirm favourable properties of our structured-sparsity models and the algorithm in terms of both prediction and variable selection accuracy.

1 INTRODUCTION

We are given a set of n input-output pairs $\{(\mathbf{x}^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) : \mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} \subseteq \mathbb{R}, i \in \mathbb{N}_n\}$ sampled i.i.d. according to an unknown probability measure ρ. Our task is to learn a regression function $f : \mathcal{X} \to \mathcal{Y}$ with minimal expected squared error loss $\mathcal{L}(f) = \mathbb{E}\,(y - f(\mathbf{x}))^2 = \int (y - f(\mathbf{x}))^2\, d\rho(\mathbf{x}, y)$.

We follow the standard theory of regularised learning where $\widehat{f}$ is learned by minimising the regularised empirical squared error loss $\widehat{\mathcal{L}}(f) = \frac{1}{n}\sum_i^n \left(y^i - f(\mathbf{x}^i)\right)^2$

$$\widehat{f} = \operatorname*{argmin}_f \ \widehat{\mathcal{L}}(f) + \tau \mathcal{R}(f) \ . \qquad (1)$$

In the above, $\mathcal{R}(f)$ is a suitable penalty typically based on some prior assumption about the function space (e.g. smoothness), and $\tau > 0$ is a suitable regularization hyper-parameter. The principal assumption we consider in this paper is that the function f is sparse with respect to the original input space $\mathcal{X}$, that is it depends only on $l \ll d$ input variables.

Learning with variable selection is a well-established and rather well-explored problem in the case of linear models $f(\mathbf{x}) = \sum_a^d x_a w_a$, e.g. Hastie et al. (2015). The main ideas from linear models have been successfully transferred to additive models $f(\mathbf{x}) = \sum_a^d f_a(x_a)$, e.g. Ravikumar et al. (2007), Bach (2009), Koltchinskii and Yuan (2010), and Yin et al. (2012), or to additive models with interactions $f(\mathbf{x}) = \sum_a^d f_a(x_a) + \sum_{a<b}^d f_{a,b}(x_a, x_b)$, e.g Lin and Zhang (2006) and Tyagi et al. (2016).

However, sparse modelling of general non-linear functions is more intricate. A promising stream of works focuses on the use of non-linear (conditional) cross-covariance operators arising from embedding probability measures into Hilbert function spaces, e.g. Yamada et al. (2014) and Chen et al. (2017).

In this work, we follow an alternative approach proposed in Rosasco et al. (2013) based on partial derivatives and develop new regularizers to promote structured sparsity with respect to the original input variables. We stress that our objective here is not to learn new data representations nor learn sparse models in some latent feature space, e.g. Gurram and Kwon (2014). Nor is it to learn models sparse in the data instances (in the sense of support vectors, e.g. Chan et al. (2007)). We aim at selecting the relevant input variables, the relevant dimensions of the input vectors $\mathbf{x} \in \mathbb{R}^d$.

After a brief review of the regularizers used in Rosasco et al. (2013) for individual variable selection in non-linear model learning (similar in spirit to lasso Tibshirani (1996)) we propose two extensions motivated by the linear structured-sparsity learning literature. Using suitable norms of the partial derivatives we propose the non-linear versions of the group lasso Yuan and Lin (2006) and the elastic net Zou and Hastie (2005).

We pose our problem into the framework of learning in the reproducing kernel Hilbert space (RKHS). We extend the representer theorem to show that the minimiser of (1) with our new regularizers $\mathcal{R}(f)$ can be conveniently written as a linear combination of kernel functions and their partial derivatives evaluated over the training set.

We further propose a new reformulation of the equivalent finite dimensional learning problem, which allows us to develop a new algorithm (NVSD) based on the Alternating Direction Method of Multipliers (ADMM) Boyd (2010). This is a generic algorithm that can be used (with small alterations) for all regularizers we discuss here. At each iteration, the algorithm needs to solve a single linear problem, perform a proximal step resulting in a soft-thresholding operation, and do a simple additive update of the dual variables. Unlike Rosasco et al. (2013), which uses approximations of the proximal operator, our algorithm is based on proximals admitting closed forms for all the discussed regularizers, including the one suggested previously in Rosasco et al. (2013). Furthermore, by avoiding the approximations in the proximal step, the algorithm directly provides also the learned sparsity patterns over the training set (up to the algorithmic convergence precision).

We explore the effect of the proposed regularizers on model learning on synthetic and real-data experiments, and confirm the superior performance of our methods in comparison to a range of baseline methods when learning structured-sparse problems. Finally, we conclude by discussing the advantages and shortcomings of the current proposal and outline some directions for future work.

2 REGULARIZERS FOR VARIABLE SELECTION

In Rosasco et al. (2013) the authors propose to use the partial derivatives of the function with respect to the input vector dimensions $\{\partial_a f : a \in \mathbb{N}_d\}$ to construct a regularizer promoting sparsity. The partial derivative evaluated at an input point $\partial_a f(\mathbf{x})$ is the rate of change of the function at that point with respect to x_a holding the other input dimensions fixed. Intuitively, when the function does not dependent on an input variable (input dimension a), its evaluations do not change with changes in the input variable: $\partial_a f(\mathbf{x}) = 0$ at all points $\mathbf{x} \in \mathcal{X}$. A natural measure of the size of the partial derivatives across the space $\mathcal{X}$ is the L_2 norm

$$||\partial_a f||_{L_2} = \sqrt{\int_{\mathcal{X}} |\partial_a f(\mathbf{x})|^2 \, d\rho_x(\mathbf{x})} \qquad (2)$$

Remark 1. *At this point we wish to step back and make a link to the linear models $f(\mathbf{x}) = \sum_a^d x_a w_a$. The partial derivatives with respect to any of the d dimensions of the input vector $\mathbf{x}$ are the individual elements of the d-dimensional parameter vector $\mathbf{w}$, $\partial_a f(\mathbf{x}) = w_a$, and this at every point $\mathbf{x} \in \mathcal{X}$. For the linear model we thus have $||\partial_a f||_{L_2} = |w_a|$. Sparsity inducing norms or constraints operating over the parameter vectors $\mathbf{w}$ can therefore be seen as special cases of the same norms and constraints imposed on the partial-derivative norms* (2).

2.1 SPARSITY INDUCING NORMS

The sparsity objective over a vector $\mathbf{v} \in \mathbb{R}^d$ can be cast as the minimization of the ℓ_0 norm $||\mathbf{v}||_0 = \#\{a = 1, \ldots, d : v_a \neq 0\}$ which counts the number of non-zero elements of the vector. Since it is well known from the linear sparse learning literature that finding the ℓ_0 solutions is computationally difficult in higher dimensions (NP-hard, Weston et al. (2003)), the authors in Rosasco et al. (2013) suggest to use its tightest convex relaxation, the ℓ_1 norm $||\mathbf{v}||_1 = \sum_a^d |v_a|$. They apply the ℓ_1 norm over the partial-derivative norms (2) so that the lasso-like sparsity regularizer in (1) is

$$\mathcal{R}^L(f) = \sum_{a=1}^d ||\partial_a f||_{L_2} \ . \qquad (3)$$

In this paper we explore two extensions inspired by the linear sparse learning, opening the doors to many of the other sparsity and structured sparsity inducing norms that have been proposed in the abundant literature on this topic. Namely, we focus here on the structured sparsity induced by the mixed ℓ_1/ℓ_2 norm known in the context of linear least squares as the group lasso Yuan and Lin (2006). For a vector $\mathbf{v}$ composed of G groups $\mathbf{v}_g$ (non-overlapping but not necessarily consecutive) with p_g number of elements each, the mixed ℓ_1/ℓ_2 norm is $||\mathbf{v}||_{1,2} = \sum_g^G p_g \, ||\mathbf{v}_g||_2$. The corresponding group-lasso-like regularizer to be used in (1) is

$$\mathcal{R}^{GL}(f) = \sum_{g=1}^G p_g \sqrt{\sum_{a \in g} ||\partial_a f||_{L_2}^2} \ . \qquad (4)$$

Second, we look at the elastic net penalty proposed initially in Zou and Hastie (2005). This uses a convex combination of the ℓ_1 and square of the ℓ_2 norm and has been shown to have better selection properties over the vanilla ℓ_1 norm regularization in the presence of highly correlated features. Unlike the ℓ_1 penalty, the combined elastic net is also strictly convex. The corresponding elastic-net-like regularizer to be used in (1) is

$$\mathcal{R}^{EN}(f) = \mu \sum_{a=1}^{d} ||\partial_a f||_{L_2} + (1-\mu) \sum_{a=1}^{d} ||\partial_a f||_{L_2}^2, \quad \mu \in [0,1] \ . \tag{5}$$

2.2 EMPIRICAL VERSIONS OF REGULARIZERS

A common problem of the regularizers introduced above is that in practice they cannot be evaluated due to the unknown probability measure ρ_x on the input space $\mathcal{X}$. Therefore instead of the partial-derivative norms defined in expectation in (2)

$$||\partial_a f||_{L_2} = \sqrt{\mathbb{E}|\partial_a f(\mathbf{x})|^2} \tag{6}$$

we use their sample estimates replacing the expectation by the training sample average

$$||\partial_a f||_{2_n} = \sqrt{\frac{1}{n}\sum_{i}^{n} |\partial_a f(\mathbf{x}^i)|^2} \ . \tag{7}$$

This corresponds to the move from expected loss to the empirical loss introduced in section 1 and is enabled by the i.i.d. sample assumptions.

In result, the regression function is learned from the empirical version of (1)

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \ \widehat{\mathcal{L}}(f) + \tau \widehat{\mathcal{R}}(f) \ , \tag{8}$$

where $\widehat{\mathcal{R}}(f)$ are the empirical analogues of the regularizers (3), (4) and (5) replacing the population partial-derivative norms $||\partial_a f||_{L_2}$ by their sample estimates $||\partial_a f||_{2_n}$. The function space $\mathcal{F}$ is discussed next.

3 LEARNING IN RKHS

In this paper, the hypothesis space $\mathcal{F}$ within which we learn the function f is a reproducing kernel Hilbert space (RKHS). We recall (e.g. Saitoh and Sawano (2016)) that a RKHS is a function space $\mathcal{F}$ of real-valued functions over $\mathcal{X}$ endowed with an inner product $\langle .,. \rangle_{\mathcal{F}}$ and the induced norm $||.||_{\mathcal{F}}$ that is uniquely associated with a positive semidefinite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The kernel k has the reproducing property $\langle k_{\mathbf{x}}, f \rangle_{\mathcal{F}} = f(\mathbf{x})$ and, in particular, $\langle k_{\mathbf{x}}, k_{\mathbf{x}'} \rangle_{\mathcal{F}} = k(\mathbf{x}, \mathbf{x}')$, where $k_{\mathbf{x}} \in \mathcal{F}$ is the kernel section centred at $\mathbf{x}$ such that $k_{\mathbf{x}}(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$ for any two $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Furthermore, the space $\mathcal{F}$ is the completion of the linear span of the functions $\{k_{\mathbf{x}} : \mathbf{x} \in \mathcal{X}\}$.

In addition to these fairly well known properties of the RKHS and its kernel, the author in Zhou (2008) has shown that if k is continuous and sufficiently smooth the kernel partial-derivative functions belong to the RKHS and have a partial-derivative reproducing property. More specifically, we define the kernel partial-derivative function $[\partial_a k_{\mathbf{x}}] : \mathcal{X} \to \mathbb{R}$ as

$$[\partial_a k_{\mathbf{x}}](\mathbf{x}') = \frac{\partial}{\partial x_a} k(\mathbf{x}, \mathbf{x}') \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} \ . \tag{9}$$

The function $[\partial_a k_{\mathbf{x}}] \in \mathcal{F}$ has the reproducing property $\langle [\partial_a k_{\mathbf{x}}], f \rangle_{\mathcal{F}} = \partial_a f(\mathbf{x})$. In particular $\langle [\partial_a k_{\mathbf{x}}], k_{\mathbf{x}'} \rangle_{\mathcal{F}} = \partial_a k_{\mathbf{x}'}(\mathbf{x})$ and $\langle [\partial_a k_{\mathbf{x}}], [\partial_b k_{\mathbf{x}'}] \rangle_{\mathcal{F}} = \frac{\partial^2}{\partial x_a \partial x'_b} k(\mathbf{x}, \mathbf{x}')$.

Remark 2. *Since the notation above may seem somewhat knotty at first, we invite the reader to appreciate the difference between the function* $[\partial_a k_{\mathbf{x}}]$ *and the partial derivative of the kernel section with respect to the ath dimension* $\partial_a k_{\mathbf{x}}$. *Clearly,* $[\partial_a k_{\mathbf{x}}](\mathbf{x}') \neq \partial_a k_{\mathbf{x}}(\mathbf{x}')$ *for any* $\mathbf{x} \neq \mathbf{x}' \in \mathcal{X}$. *However, due to the symmetry of the kernel we do have* $[\partial_a k_{\mathbf{x}}](\mathbf{x}') = \partial_a k_{\mathbf{x}'}(\mathbf{x}) = \frac{\partial}{\partial x_a} k(\mathbf{x}, \mathbf{x}')$.

3.1 SOLUTION REPRESENTATION

The variational (infinite-dimensional) problem (8) is difficult to handle as is. However, it has been previously shown for a multitude of RKHS learning problems that their solutions $\widehat{f}$ can be expressed as finite linear combinations of the kernel evaluations over the training data Argyriou and Dinuzzo (2014). This property, known as *representer theorem*, renders the problems amenable to practical computations.

Proposition 1. *The minimising solution* $\widehat{f}$ *of the variational problem*

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \ \widehat{\mathcal{L}}(f) + \tau \widehat{\mathcal{R}}(f) + \nu ||f||_{\mathcal{F}}^2 \ , \tag{10}$$

where $\tau, \nu \geq 0$ *and* $\widehat{\mathcal{R}}(f)$ *is any of the empirical versions of the three formulations* (3), (4), (5) *can be represented as*

$$\widehat{f} = \sum_{i}^{n} \alpha_i \, k_{\mathbf{x}^i} + \sum_{i}^{n} \sum_{a}^{d} \beta_{ai} \, [\partial_a k_{\mathbf{x}^i}] \ . \tag{11}$$

The proof (available in the appendix) follows the classical approach (e.g. Schölkopf et al. (2001)) of decomposition of $\mathcal{F}$ into the space spanned by the representation and its orthogonal complement.

The proposition extends the representer theorem of Rosasco et al. (2013) to the new regularizers (4) and (5).

Note that we included the induced Hilbert norm $||f||_{\mathcal{F}}$ into (10) as a useful generalization that reduces to our original problem (8) if $\nu = 0$. On the other hand, when $\tau = 0$ we recover a classical kernel regression problem which is known to have another simpler representation consisting just of the first term in (11).

4 ALGORITHM

In this section we describe the new algorithm we developed to solve problem (10) with the three sparse regularizers introduced in section 2. The algorithm is versatile so that it requires only small alterations in specific steps to move from one regularizer to the other. Importantly, unlike the algorithm proposed in Rosasco et al. (2013) for solving only the lasso-like problem, our algorithm does not need to rely on proximal approximations since all the proximal steps can be evaluated in closed forms. Our algorithm also directly provides values of the partial derivatives of the learned function indicating the learned sparsity.

4.1 FINITE DIMENSIONAL FORMULATION

To be able to develop a practical algorithm we first need to reformulate the variational optimisation problem (10) into its finite dimensional equivalent. For this we introduce the following objects: the n-long vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$, the dn-long vector $\boldsymbol{\beta} = [\beta_{11}, \ldots, \beta_{1n}, \beta_{21} \ldots \beta_{dn}]^T$, the $n \times n$ symmetric PSD kernel matrix $\mathbf{K}$ such that $K_{ij} = k(\mathbf{x}^i, \mathbf{x}^j)$, the $n \times n$ (non-symmetric) kernel derivative matrices $\mathbf{D}^a$ and $\tilde{\mathbf{D}}^a, a \in \mathbb{N}_d$ such that $D^a_{ij} = [\partial_a k_{\mathbf{x}^i}](\mathbf{x}^j) = \partial_a k_{\mathbf{x}^j}(\mathbf{x}^i) = \tilde{D}^a_{ji}$, the $n \times n$ (non-symmetric) kernel 2nd derivative matrices $\mathbf{L}^{ab}, a, b \in \mathbb{N}_d$ such that $\mathbf{L}^{ab}_{ij} = \frac{\partial^2}{\partial x^i_a \partial x^j_b} k(\mathbf{x}^i, \mathbf{x}^j) = \frac{\partial}{\partial x^j_b}[\partial_a k_{\mathbf{x}^i}](\mathbf{x}^j) = \mathbf{L}^{ba}_{ji}$. Further, we need the following concatenations:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^1 \\ \ldots \\ \mathbf{D}^d \end{bmatrix} \quad \mathbf{L}^a = [\mathbf{L}^{a1} \ldots \mathbf{L}^{ad}] \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}^1 \\ \ldots \\ \mathbf{L}^d \end{bmatrix}$$

and specifically for the groups g in $\mathcal{R}^{GL}$ the partitions

$$\ddot{\mathbf{D}}^g = \begin{bmatrix} \mathbf{D}^{g_1} \\ \ldots \\ \mathbf{D}^{g_{p_g}} \end{bmatrix} \quad \ddot{\mathbf{L}}^g = \begin{bmatrix} \mathbf{L}^{g_1} \\ \ldots \\ \mathbf{L}^{g_{p_g}} \end{bmatrix} \quad ,$$

where the subscripts g_i are the corresponding indexes of the input dimensions.

Proposition 2. *The variational problem* (10) *is equivalent to the finite dimensional problem*

$$\operatorname*{argmin}_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \mathcal{J}1(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \tau \mathcal{J}2(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \nu \mathcal{J}3(\boldsymbol{\alpha}, \boldsymbol{\beta}), \qquad (12)$$

where

$$\mathcal{J}1(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{n}||\mathbf{y} - \mathbf{K}\boldsymbol{\alpha} - \mathbf{D}^T\boldsymbol{\beta}||_2^2$$

$$\mathcal{R}^L : \mathcal{J}2(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\sqrt{n}} \sum_a^d ||\mathbf{D}^a\boldsymbol{\alpha} + \mathbf{L}^a\boldsymbol{\beta}||_2$$

$$\mathcal{R}^{GL} : \mathcal{J}2(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\sqrt{n}} \sum_g^G p_g \, ||\ddot{\mathbf{D}}^g \boldsymbol{\alpha} + \ddot{\mathbf{L}}^g \boldsymbol{\beta}||_2$$

$$\mathcal{R}^{EN} : \mathcal{J}2(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\mu}{\sqrt{n}} \sum_a^d ||\mathbf{D}^a\boldsymbol{\alpha} + \mathbf{L}^a\boldsymbol{\beta}||_2 + \frac{1-\mu}{n} \sum_a^d ||\mathbf{D}^a\boldsymbol{\alpha} + \mathbf{L}^a\boldsymbol{\beta}||_2^2$$

$$\mathcal{J}3(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\mathbf{D}^T\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}$$

The proof (available in the appendix) is based on the finite dimensional representation (11) of the minimising function, and the kernel and derivative reproducing properties stated in section 3.

The problem reformulation (12) is instructive in terms of observing the roles of the kernel and the derivative matrices and is reminiscent of the classical finite dimensional reformulation of Hilbert-norm regularised least squares. However, for the development of our algorithm we derive a more convenient equivalent form.

Proposition 3. *The variational problem* (10) *is equivalent to the finite dimensional problem*

$$\operatorname*{argmin}_{\boldsymbol{\omega}} \frac{1}{n}||\mathbf{y} - \mathbf{F}\boldsymbol{\omega}||_2^2 + \tau \mathcal{J}(\boldsymbol{\omega}) + \nu \boldsymbol{\omega}^T \mathbf{Q} \boldsymbol{\omega}, \qquad (13)$$

where

$$\mathcal{R}^L : \mathcal{J}(\boldsymbol{\omega}) = \frac{1}{\sqrt{n}} \sum_a^d ||\mathbf{Z}^a\boldsymbol{\omega}||_2$$

$$\mathcal{R}^{GL} : \mathcal{J}(\boldsymbol{\omega}) = \frac{1}{\sqrt{n}} \sum_g^G p_g \, ||\ddot{\mathbf{Z}}^g \boldsymbol{\omega}||_2$$

$$\mathcal{R}^{EN} : \mathcal{J}(\boldsymbol{\omega}) = \frac{\mu}{\sqrt{n}} \sum_a^d ||\mathbf{Z}^a\boldsymbol{\omega}||_2 + \frac{1-\mu}{n} \sum_a^d ||\mathbf{Z}^a\boldsymbol{\omega}||_2^2 \ ,$$

with

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \quad \begin{matrix} \mathbf{F} = [\mathbf{K}\mathbf{D}^T] \\ \mathbf{Z}^a = [\mathbf{D}^a\mathbf{L}^a] \\ \ddot{\mathbf{Z}}^g = [\ddot{\mathbf{D}}^g\ddot{\mathbf{L}}^g] \end{matrix} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ 2\mathbf{D} & \mathbf{L} \end{bmatrix}$$

The proof is trivial using (12) as an intermediate step.

4.2 DEVELOPMENT OF GENERIC ALGORITHM

Problem (13) is convex though its middle part $\mathcal{J}(\boldsymbol{\omega})$ is non-differentiable for all three discussed regularizers. Indeed, it is the singularities of the norms at zero points that yield the sparse solutions. A popular approach for solving convex non-differentiable problems is the proximal gradient descent Parikh and Boyd (2013). At every step it requires evaluating the proximal operator defined for any function $f : \mathbb{R}^m \to \mathbb{R}^m$ and any vector $\mathbf{v} \in \mathbb{R}^m$ as

$$prox_f(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) + \frac{1}{2}||\mathbf{x} - \mathbf{v}||_2^2 \ . \qquad (14)$$

However, proximal operators for the functions $\mathcal{J}$ in (13) do not have closed forms or fast methods for solving which makes the proximal gradient descent algorithm difficult to use.

We therefore propose to introduce a linearizing change of variables $\mathbf{Z}^a\boldsymbol{\omega} = \boldsymbol{\varphi}_a$ and cast the problem in a form amenable for the ADMM method Boyd (2010)

$$\min \ \mathcal{E}(\boldsymbol{\omega}) + \tau\,\mathcal{I}(\boldsymbol{\varphi}), \quad \text{s.t. } \mathbf{Z}\boldsymbol{\omega} - \boldsymbol{\varphi} = 0 \ . \qquad (15)$$

In the above

$$\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \dots \\ \boldsymbol{\varphi}_d \end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix} \mathbf{Z}^1 \\ \dots \\ \mathbf{Z}^d \end{bmatrix} \ ,$$

(or concatenation of the double-dot version for the group structure), $\mathcal{E} : \mathbb{R}^{n+nd} \to \mathbb{R}$ is the convex differentiable function

$$\mathcal{E}(\boldsymbol{\omega}) = \frac{1}{n}||\mathbf{y} - \mathbf{F}\boldsymbol{\omega}||_2^2 + \nu\,\boldsymbol{\omega}^T\mathbf{Q}\,\boldsymbol{\omega} \ ,$$

and $\mathcal{I} : \mathbb{R}^{nd} \to \mathbb{R}$ is the convex non-differentiable function corresponding to each regularizer such that $\mathcal{I}(\boldsymbol{\varphi}) = \mathcal{J}(\boldsymbol{\omega})$ for every $\mathbf{Z}\,\boldsymbol{\omega} = \boldsymbol{\varphi}$.

At each iteration the ADMM algorithm consists of the following three update steps (the standard approach of augmented Lagrangian with $\boldsymbol{\lambda}$ as the scaled dual variable and κ as the step size):

$$S1 : \ \boldsymbol{\omega}^{(k+1)} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \ \mathcal{E}(\boldsymbol{\omega}) + \frac{\kappa}{2}\,||\mathbf{Z}\boldsymbol{\omega} - \boldsymbol{\varphi}^{(k)} + \boldsymbol{\lambda}^{(k)}||_2^2$$

$$S2 : \ \boldsymbol{\varphi}^{(k+1)} = \underset{\boldsymbol{\varphi}}{\operatorname{argmin}}\, \tau\mathcal{I}(\boldsymbol{\varphi}) + \frac{\kappa}{2}||\mathbf{Z}\,\boldsymbol{\omega}^{(k+1)} - \boldsymbol{\varphi} + \boldsymbol{\lambda}^{(k)}||_2^2$$

$$S3 : \ \boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \mathbf{Z}\,\boldsymbol{\omega}^{(k+1)} - \boldsymbol{\varphi}^{(k+1)}$$

The first step $S1$ is a convex quadratic problem with a closed form solution

$$S1 : \ (\nu\mathbf{Q} + \nu\mathbf{Q}^T + 2n^{-1}\mathbf{F}^T\mathbf{F} + \kappa\mathbf{Z}^T\mathbf{Z})\,\boldsymbol{\omega}^{(k+1)} = 2n^{-1}\mathbf{F}^T\mathbf{y} + \kappa\mathbf{Z}^T(\boldsymbol{\varphi}^{(k)} - \boldsymbol{\lambda}^{(k)})$$

By comparing with (14) we observe that the second step $S2$ is a proximal update. The advantage of our problem reformulation and our algorithm is that this has a closed form for all the three discussed regularizers.

Proposition 4. *The proximal problem in step $S2$ is decomposable by the d partitions of vector $\boldsymbol{\varphi}$ (or G partition in case of the group structure) and the minimising solution is*

$$\mathcal{R}^L : \ \boldsymbol{\varphi}_a^{(k+1)} = (\mathbf{Z}^a\,\boldsymbol{\omega}^{(k+1)} + \boldsymbol{\lambda}_a^{(k)})\left(1 - \frac{\tau}{\kappa\sqrt{n}||\mathbf{Z}^a\,\boldsymbol{\omega}^{(k+1)} + \boldsymbol{\lambda}_a^{(k)}||_2}\right)_+$$

$$\mathcal{R}^{GL} : \ \boldsymbol{\varphi}_g^{(k+1)} = (\ddot{\mathbf{Z}}^g\,\boldsymbol{\omega}^{(k+1)} + \ddot{\boldsymbol{\lambda}}_g^{(k)})\left(1 - \frac{\tau\,p_g}{\kappa\sqrt{n}||\mathbf{Z}^g\,\boldsymbol{\omega}^{(k+1)} + \boldsymbol{\lambda}_g^{(k)}||_2}\right)_+$$

$$\mathcal{R}^{EN} : \ \boldsymbol{\varphi}_a^{(k+1)} = \frac{\mathbf{Z}^a\,\boldsymbol{\omega}^{(k+1)} + \boldsymbol{\lambda}_a^{(k)}}{2\tau(1-\mu)/(\kappa n) + 1}\left(1 - \frac{\tau\mu}{\kappa\sqrt{n}||\mathbf{Z}^a\,\boldsymbol{\omega}^{(k+1)} + \boldsymbol{\lambda}_a^{(k)}||_2}\right)_+$$

Here $(v)_+ = \min(0, v)$ is the thresholding operator.

The decomposability comes from the additive structure of $\mathcal{I}$. The derivation follows similar techniques as used for classical ℓ_1 and ℓ_2 proximals.[1]

4.3 PRACTICAL IMPLEMENTATION

In practice, the $\mathbf{Q}$, $\mathbf{F}$ and $\mathbf{Z}$ matrices are precomputed in a preprocessing step and passed onto the algorithm as inputs. The matrices are directly computable using the kernel function k and its first and second order derivatives evaluated at the training points (following the matrix definitions introduced in section 4.1).

The algorithm converges to a global minimum by the standard properties of ADMM. In our implementation (available at `https://bitbucket.org/dmmlgeneva/nvsd_uai2018/`) we follow a simple updating rule Boyd (2010, sec. 3.4.1) for the step size κ. We use inexact minimization for the most expensive step $S1$, gradually increasing the number of steepest descent steps, each with complexity $\mathcal{O}\left((nd)^2\right)$.

Furthermore, we use $S2$ to get the values of the training sample partial-derivative norms defined in equation (7) as $||\partial_a f^{(k)}||_{2_n} = ||\boldsymbol{\varphi}_a^{(k)}||_2/\sqrt{n}$. The sparsity pattern is obtained by examining for which of the dimensions $a \in \mathbb{N}_d$ the norm is zero $||\partial_a f^{(k)}||_{2_n} = 0$.

[1] For $\mathcal{R}^{EN}$ it is more practical to add the quadratic term into $\mathcal{E}(\boldsymbol{\omega})$ in $S1$ and use the corresponding scaled version of the $\mathcal{R}^L$ proximal in $S2$.

5 EMPIRICAL EVALUATION

We conducted a set of synthetic and real-data experiments to document the efficacy of our structured-sparsity methods and the new algorithm under controlled and more realistic conditions. We compare our methods NVSD(L), NVSD(GL) and NVSD(EN) in terms of their predictive accuracy and their selection ability to the simple (non-sparse) kernel regularised least squares (Krls), to the sparse additive model (SpAM) of Ravikumar et al. (2007), to the non-linear cross-covariance-based method using the Hilbert Schmidt Independence Criterion in a lasso-like manner (HSIC) of Yamada et al. (2014), and to the derivative-based lasso-like method (Denovas) of Rosasco et al. (2013).[2] We compared also to simple mean and linear sparse and non-sparse models. All of these performed considerably worse than the non-linear models and therefore are not listed in the summary results. For all the sparse kernel methods we consider a two-step debiasing procedure based on variable selection via the base algorithm followed by a simple kernel regularised least squares on the selected variables.[3]

5.1 SYNTHETIC EXPERIMENTS

We motivate each synthetic experiment by a realistic story-line and explain the data generating process here below. In all the synthetic experiments we fix the input dimension to $d = 18$ with only 6 input variables $\{1, 2, 3, 7, 8, 9\}$ relevant for the model and the other 12 irrelevant.

E1 In the first experiment we focus on the NVSD(GL) which assumes the input variables can be grouped a priori by some domain knowledge (e.g. each group describes a *type* of input data such as a different biological process) and the groups are expected to be completely in or out of the model. The input variables are generated independently from a standard normal distribution and they are grouped by three into 6 groups. The output is generated from the 1st and the 3rd group as

$$y = \sum_{i=1}^{3}\sum_{j=i}^{3}\sum_{k=j}^{3} x_i x_j x_k + \sum_{q=7}^{9}\sum_{r=q}^{9}\sum_{s=r}^{9} x_q x_r x_s + \epsilon \ ,$$

with $\epsilon \sim N(0, 0.01)$. For learning we fix the kernel to 3rd order polynomial.

[2]For HISC and Denovas we used the author's code, for SpAM the R implementation of Zhao et al. (2014). For all algorithms we kept the default settings.

[3]This is native to Denovas and necessary for HSIC which otherwise does not produce a predictive model.

E2 In the second experiment we do not assume any a priori grouping of the variables. Instead some of the variables are strongly correlated (perhaps relating to a single phenomenon), a case for NVSD(EN). The input variables are generated similarly as in E1 but with the pairs $\{1, 7\}$, $\{2, 8\}$ and $\{3, 9\}$ strongly correlated (Pearson's population correlation coefficient 0.95). The remaining (irrelevant) input variables are also pair-wise correlated and the output is generated as

$$y = \sum_{i,j,k=1}^{3} x_i x_j x_k + \sum_{q,r,s=7}^{9} x_q x_r x_s + \epsilon \ ,$$

with $\epsilon \sim N(0, 0.01)$. For learning we fix the kernel to 3rd order polynomial.

E3 In the third experiment we assume the inputs are noisy measurements of some true phenomenon (e.g. repeated measurements, measurements from multiple laboratories) for which there is no reason to prefer one over the other in the model. We first generate the true data $z_i \sim N(0, 1), i = 1, \ldots, 6$ and use these to generate the outputs as

$$y = 10(z_1^2 + z_3^2)e^{-2(z_1^2+z_3^2)} + \epsilon \ ,$$

with $\epsilon \sim N(0, 0.01)$. We then generate the noisy measurements that will be used as inputs for the learning: for each z_i we create three noisy measurements $x_{ij} = z_i + N(0, 0.1), j = 1, 2, 3$ (a group for the NVSD(GL) method); the input vector is the concatenation of all x_{ij} so that from the 18 long concatenated input vector $\mathbf{x}$ again only the set $\{1, 2, 3, 7, 8, 9\}$ of the dimensions is relevant for predicting the output y. For learning we fix the kernel to Gaussian with width $\sigma = 4$.

Remark 3. *In all the synthetic experiments we use the same experimental protocol. We split the data into train sets varying the size in* $n = \{30, 50, 70, 90, 110\}$, *a validation set of length 1000, and a test set of length 1000. We train the models over the train sets and use the validation set to select the regularization hyper-parameters (and therefore the models) based on the minimal validation MSE. We use dense grids of 50 points for the* τ *search (automatically established by the algorithm) and 5 points grid for* $\mu \in \{0.1, \ldots, 0.9\}$. *Complete settings (also for the baseline methods) are detailed in the replication files publicly available at* `https://bitbucket.org/dmmlgeneva/nvsd_uai2018/`.

We report the average results across 50 independent replications of the experiments in table 1. We measure

Table 1: Results of Synthetic Experiments

		Train size	30	50	70	90	110
E1	RMSE	Krls	12.79	11.66	10.99	10.43	9.80
		SpAM	11.41	9.47	8.66	8.22	7.75
		HSIC	11.37	10.00	8.58	7.28	5.68
		Denovas	11.66	10.87	12.37	13.28	11.78
		NVSD(L)	11.55	10.22	9.36	7.90	7.13
		NVSD(GL)	**9.92**	**7.89**	**6.34**	**1.94**	**2.41**
	Selection error	Krls	0.67	0.67	0.67	0.67	0.67
		SpAM	0.54	0.56	0.59	0.57	0.58
		HSIC	0.50	0.48	0.42	0.35	0.32
		Denovas	0.49	0.50	0.53	0.67	0.73
		NVSD(L)	0.49	0.47	0.48	0.39	0.32
		NVSD(GL)	**0.28**	**0.24**	**0.22**	**0.05**	**0.11**
E2	RMSE	Krls	27.69	24.83	22.53	19.14	18.04
		SpAM	31.24	29.21	29.25	27.11	26.03
		HSIC	21.74	15.50	12.02	9.42	7.67
		Denovas	24.23	34.33	17.51	8.89	11.20
		NVSD(L)	21.24	16.59	11.79	8.61	7.35
		NVSD(EN)	**17.53**	**10.05**	**5.67**	**4.29**	**3.29**
	Selection error	Krls	0.67	0.67	0.67	0.67	0.67
		SpAM	0.57	0.55	0.49	0.52	0.46
		HSIC	0.52	0.42	0.42	0.35	0.32
		Denovas	0.46	0.54	0.40	0.30	0.26
		NVSD(L)	0.46	0.43	0.36	0.31	0.29
		NVSD(EN)	**0.35**	**0.20**	**0.14**	**0.09**	**0.08**
E3	RMSE	Krls	0.65	0.55	0.54	0.53	0.50
		SpAM	0.51	0.49	0.47	0.47	0.46
		HSIC	0.52	0.47	0.45	0.44	0.43
		Denovas	0.55	0.51	0.50	0.51	0.50
		NVSD(L)	0.51	0.44	0.44	0.41	0.34
		NVSD(GL)	0.51	**0.41**	**0.39**	**0.33**	0.31
		NVSD(EN)	**0.50**	0.43	0.42	0.36	**0.30**
	Selection error	Krls	0.67	0.67	0.67	0.67	0.67
		SpAM	0.65	0.61	0.60	0.58	0.59
		HSIC	0.59	0.51	0.53	0.47	0.44
		Denovas	0.49	0.45	0.47	0.45	0.41
		NVSD(L)	0.33	0.30	0.40	0.34	0.23
		NVSD(GL)	**0.26**	**0.20**	**0.24**	**0.15**	**0.14**
		NVSD(EN)	0.30	0.33	0.35	0.25	0.16

Best results in bold; underlined when structured-sparsity methods significantly better than all other methods using Wilcoxon signed-rank test at 5% significance level.

the prediction accuracy by the root mean squared error (RMSE) over the test sets and the selection accuracy by the Tanimoto distance between the true sparsity and the learned sparsity patterns (section 4.3).

Our structured-sparsity methods clearly outperform all the non-structured sparse learning methods achieving better prediction accuracy based on more precise variable selection, typically with statistically significant differences. Also, the prediction and selection accuracy generally increases (errors decrease) for larger training sample sizes suggesting our methods are well-behaved in terms of the standard statistical learning paradigms. In the E3 experiment, NVSD(GL) performs the best having the benefit of the prior knowledge of the variable groupings. Remarkably, NVSD(EN) follows closely after even without such prior information, learning about the groups of correlated variables from the data when building the model.

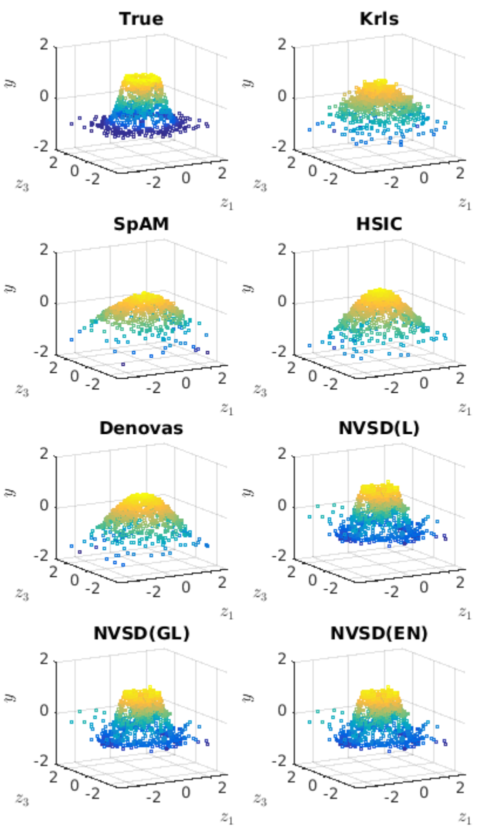

Figure 1: Predictions for the E3 experiment over the test data. We picked an example for the model trained with 110 instances (the 17th replication) which illustrates well the advantage our NVSD methods have over the baselines in capturing the True complex non-linear structure.

Krls can only learn full models and therefore performs rather poorly on these by-construction sparse problems. From the other three baselines, HSIC typically achieves the second best results (after our NVSD methods). SpAM is not particularly suitable for the non-additive structures of our experiments. Finally, in all the experiments our NVSD(L) outperforms Denovas though they share the same lasso-like problem formulation. We attribute this to our new algorithm developed in section 4 which, unlike Denovas, does not rely on approximations of the proximal operators.

5.2 REAL-DATA EXPERIMENTS

For the real-data experiments we used a collection of regression datasets from UCI Lichman (2013) and LIACC[4] repositories listed in table 2.

Table 2: Real Datasets Desription

Code	Name	Inputs	Test Size	Source
AI	Airfoil Self Noise	5	700	UCI
BH	Boston Housing	10	200	UCI
CC	Concrete Compressive	8	450	UCI
EN	Energy Efficiency	8	300	UCI
CP	Computer Activity	21	1000	LIACC
EL	F16 Elevators	17	1000	LIACC
KN	Kynematics	8	1000	LIACC

We report the average results across 50 replications of the experiments in tables 3 and 4. We use RMSE over the test data for measuring the prediction accuracy. For the real datasets we do not know the ground-truth sparsity patterns. Instead of measuring the selection error we therefore count the number of input variables selected by each method. Krls has no selection ability, its support size is hence equal to the total number of input variables in each problem.

Remark 4. *We followed similar experimental protocol as for the synthetic experiments. We fixed the training sample size for all experiments to 100 instances and used 200-1000 instances for the validation and test sets (depending on the total number of available observations). We pre-processed the data by normalizing the inputs and centering the outputs. For all the experiments we used a Gaussian kernel with the width set to the median distance calculated over the nearest 20 neighbours, and the 3rd order polynomial kernel. With the exception of the EN dataset, the Gaussian kernel yielded better results and was therefore kept for the final evaluation. Full details of the settings can be found in the replication files publicly available at* `https://bitbucket.org/dmmlgeneva/nvsd_uai2018/`.

Results in table 3 are for the original data for which we have no prior knowledge about possible variable groupings. Therefore we only use the non-structured methods and our NVSD(EN) that do not rely on any such prior information.

Our NVSD methods learned sparse non-linear models achieving better or comparable results than the baselines in 4 out of the 5 experiments (BH, CP, EN, EL). For CC reducing the number of input dimensions does not seem to bring any advantages and the methods tend to learn full models. For several experiments SpAM finished with errors and therefore the results in the table are missing.

[4]http://www.dcc.fc.up.pt/∼ltorgo/Regression/DataSets.html

Table 3: Results of Real-data Experiments

	Experiment	BH	CP	CC	EN	EL
RMSE	Krls	4.00	12.27	8.70	1.83	5.10
	SpAM	4.33	∼	12.70	∼	∼
	HSIC	4.02	9.39	8.73	**1.19**	9.07
	Denovas	4.02	9.21	12.07	3.02	6.01
	NVSD(L)	3.96	8.43	**8.67**	1.50	<u>4.81</u>
	NVSD(EN)	**<u>3.93</u>**	**7.88**	8.70	1.20	**<u>4.67</u>**
Support size	Krls	10.00	21.00	8.00	8.00	17.00
	SpAM	9.00	∼	2.82	∼	∼
	HSIC	6.12	8.26	5.88	5.08	0.00
	Denovas	8.80	4.76	4.38	4.96	10.52
	NVSD(L)	8.20	3.78	7.36	7.26	14.06
	NVSD(EN)	8.06	4.58	7.98	6.66	13.00

Best results in bold; underlined when NVSD methods significantly better than all the baselines using Wilcoxon signed-rank test at 5% significance level. For several experiments SpAM finished with errors.

To explore the performance and benefits of NVSD(GL) method we had to construct variable groups that could potentially help the model learning. We adopted two strategies:

1. For CP and EL datasets we constructed the groups based on the NVSD(EN) results. For CP we grouped together the 5 most often selected variables across the 50 replications of the experiment and created 3 other groups from the remaining variables. For EL we created five groups by 3-4 elements putting together variables with similar frequencies of occurrence in the support of the learned NVSD(EN) models over the 50 replications.
2. For AI, CC, and KN datasets we doubled the original input data dimensions by complementing the input data by a copy of each input variable with permuted instance order. We then constructed two groups, the first over the original data, the second over the permuted copy.

Table 4 confirms that our NVSD(GL) is able to use the grouping information based on prior knowledge to select better, more relevant subset of variables than the non-structured baselines. Thanks to this it achieves significantly better prediction accuracy in all the experiments.

6 CONCLUSIONS AND FUTURE WORK

In this work we addressed the problem of variable selection in non-linear regression problems. We followed

Table 4: Results of Real-data Experiments with Groups

	Experiment	AI	CP	CC	KN	EL
RMSE	Krls	5.08	12.27	10.34	2.07	5.10
	SpAM	$\sim$	$\sim$	13.31	2.20	$\sim$
	HSIC	4.64	9.39	9.29	2.05	9.07
	Denovas	5.12	9.21	11.49	2.10	6.01
	NVSD(L)	4.45	8.43	9.58	2.03	4.81
	NVSD(GL)	**4.16**	**7.43**	**8.79**	**1.96**	**4.76**
Support size	Krls	10.00	21.00	16.00	16.00	17.00
	SpAM	$\sim$	$\sim$	2.60	11.32	$\sim$
	HSIC	5.08	8.26	6.16	11.82	0.00
	Denovas	5.94	4.76	6.96	9.72	10.52
	NVSD(L)	4.76	3.78	8.16	13.58	14.06
	NVSD(GL)	5.00	5.84	8.00	11.84	13.82

Best results in bold; underlined when NVSD methods significantly better than all the baselines using Wilcoxon signed-rank test at 5% significance level. For several experiments SpAM finished with errors.

up from the work of Rosasco et al. (2013) arguing for the use of partial derivatives as an indication of the pertinence of an input variable for the model. Extending the existing work, we proposed two new derivative-based regularizers for learning with structured sparsity in non-linear regression similar in spirit to the linear elastic net and group lasso.

After posing the problems into the framework of RKHS learning, we designed a new NVSD algorithm for solving these. Unlike the previously proposed Denovas our new algorithm does not rely on proximal approximations. This is most likely the main reason why our NVSD(L) method achieved systematically better predictive performance than Denovas on a broad set of experiments. We also empirically demonstrated the advantages our structured sparsity methods NVSD(GL) and NVSD(EN) bring for learning tasks with a priori known group structures or correlation in the inputs.

These promising results point to questions requiring further attention:

Our NVSD algorithm achieves better results in terms of prediction accuracy than Denovas, however, at the cost of longer training times. Its $\mathcal{O}\left((nd)^2\right)$ complexity is not favourable for scaling in neither instances nor dimensions. Exploring avenues for speeding up, possibly along the lines of random features construction, is certainly an important next step in making the algorithm operational for more practical real-life problems.

The method is based on the partial-derivative arguments and therefore assumes the functions (and therefore the kernels) are at least 2nd order differentiable (and square-integrable). We use here the polynomial and Gaussian kernel as the most commonly used examples. What other properties of the kernels are necessary to ensure good performance and how the methods could be extended to other, more complex kernels are relevant questions.

The full problem formulation (e.g. equation (10) in proposition 1) combines the sparse regularizers with the function Hilbert-norm. This combination has been proposed in Rosasco et al. (2013) to ensure that the regularization part of the problem is strongly convex and the problem is well-posed in terms of the generalization properties.

However, interactions of the Hilbert norm with the sparsity inducing regularizers of section 2 and the effects on the learning and selection properties are not yet fully clear. Empirically (from Rosasco et al. (2013) and our own experiments) the models are often little sensitive to variations in ν^5.

In addition, the $\mathcal{R}^{EN}$ regularizer is already strongly convex even without the Hilbert norm. To what degree combining it with the Hilbert norm is necessary to guarantee good generalization for outside the training needs to be further investigated. So does its behaviour and the possible improvements it can bring when learning from inputs with non-linear dependencies. In view of the above considerations, our paper is posing the motivations, foundations and principles for further studies on partial derivative-based regularizations.

Acknowledgements

This work was partially supported by the research projects HSTS (ISNET) and RAWFIE #645220 (H2020). The computations were performed at University of Geneva on the Baobab and Whales clusters.

[5] We fix it based on a small subset of replications instead of including it into the full hyper-parameter search.

References

Argyriou, Andreas and Francesco Dinuzzo (2014). "A Unifying View of Representer Theorems". In: *International Conference on Machine Learning (ICML)*.

Bach, Francis (2009). "High-Dimensional Non-Linear Variable Selection through Hierarchical Kernel Learning". In: *ArXiv 0909.0844*.

Boyd, Stephen (2010). "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends in Machine Learning* 3.1.

Chan, Antoni B, Nuno Vasconcelos, and Gert R G Lanckriet (2007). "Direct convex relaxations of sparse SVM". In: *International Conference on Machine Learning (2007)*.

Chen, Jianbo et al. (2017). "Kernel Feature Selection via Conditional Covariance Minimization". In: *Advances in Neural Information Processing Systems (NIPS)*.

Gurram, Prudhvi and Heesung Kwon (2014). "Optimal sparse kernel learning in the empirical kernel feature space for hyperspectral classification". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.4, pp. 1217–1226.

Hastie, Trevor, Robert Tibshirani, and Martin Wainwright (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Koltchinskii, Vladimir and Ming Yuan (2010). "Sparsity in multiple kernel learning". In: *Annals of Statistics* 38.6, pp. 3660–3695.

Lichman, M. (2013). "UCI Machine Learning Repository". In: *http://archive.ics.uci.edu/ml*.

Lin, Yi and Hao Helen Zhang (2006). "Component selection and smoothing in multivariate nonparametric regression". In: *Annals of Statistics* 34.5, pp. 2272–2297.

Parikh, Neal and Stephen Boyd (2013). "Proximal algorithms". In: *Foundations and Trends in Optimization* 1.3.

Ravikumar, Pradeep et al. (2007). "Spam: Sparse additive models". In: *Advances in Neural Information Processing Systems (NIPS)*.

Rosasco, Lorenzo, S Villa, and S Mosci (2013). "Nonparametric sparsity and regularization". In: *Journal of Machine Learning Research* 14, pp. 1665–1714.

Saitoh, Saburou and Yoshihiro Sawano (2016). *Theory of Reproducing Kernels and Applications*. Springer.

Schölkopf, Bernhard, Ralf Herbrich, and Alex J Smola (2001). "A Generalized Representer Theorem". In: *COLT/EuroCOLT*.

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 58.1, pp. 267–288.

Tyagi, Hemant, Andreas Krause, and Z Eth (2016). "Efficient Sampling for Learning Sparse Additive Models in High Dimensions". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Weston, Jason et al. (2003). "Use of the Zero-Norm with Linear Models and Kernel Methods". In: *Journal of Machine Learning Research* 3, pp. 1439–1461.

Yamada, Makoto et al. (2014). "High-dimensional feature selection by feature-wise kernelized Lasso." In: *Neural Computation* 26.1, pp. 185–207.

Yin, Junming, Xi Chen, and Eric P Xing (2012). "Group Sparse Additive Models". In: *International Conference on Machine Learning (ICML)*.

Yuan, Ming and Yi Lin (2006). "Model selection and estimation in regression with grouped varibles". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67.

Zhao, Tuo et al. (2014). *CRAN - Package SAM*.

Zhou, Ding Xuan (2008). "Derivative reproducing properties for kernel methods in learning theory". In: *Journal of Computational and Applied Mathematics* 220.1-2, pp. 456–463.

Zou, Hui and Trevor Hastie (Apr. 2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.