Improved Non-Parametric Sparse Recovery with Data Matched Penalties

Marco Signoretto*, Kristiaan Pelckmans†, Lieven De Lathauwer‡ and Johan A.K. Suykens*
* Katholieke Universiteit Leuven, ESAT-SCD/SISTA
Kasteelpark Arenberg 10, B-3001 Leuven (BELGIUM)
Email: marco.signoretto@esat.kuleuven.be and johan.suykens@esat.kuleuven.be
†Division of Systems and Control, Department of Information Technology, Uppsala University
Box 337 SE-751, 05 Uppsala (SWEDEN)
Email: kp@it.uu.se
‡Group Science, Engineering and Technology, Katholieke Universiteit Leuven, Campus Kortrijk
E. Sabbelaan 53, 8500 Kortrijk (BELGIUM)
Email: lieven.delathauwer@kuleuven-kortrijk.be

***Abstract*—This contribution studies the problem of learning sparse, nonparametric models from observations drawn from an arbitrary, unknown distribution. This specific problem leads us to an algorithm extending techniques for Multiple Kernel Learning (MKL), functional ANOVA models and the Component Selection and Smoothing Operator (COSSO). The key element is to use a data-dependent regularization scheme adapting to the specific distribution underlying the data. We then present empirical evidence supporting the proposed learning algorithm.**

I. Introduction

Recent research in machine learning and statistics witnessed a growing interest in learning models that are both achieving good generalization performance, as well as resulting in insight into the model structure. Among other, this task has been tackled via non-parametric extensions of the LASSO [1]. In the context of machine learning this type of problems has been studied in the Multiple Kernel Learning (MKL) framework, both from a functional view-point ([2]; [3]) and from the point of view of optimization ([4]; [5]). The subject was also addressed in the context of functional ANOVA models under the name of COmponent Selection and Smoothing Operator (COSSO) [6].

When the input data are dependent, theoretical insights as well as empirical evidence show that the estimated models are more likely to include irrelevant components. On a more general track, namely for general penalized empirical risk minimization problems, results have been shown to depend on how well the underlying structure is captured by the penalty function [7]. This suggests to convey structural assumptions on the problem by suitably crafting the penalty, which is the approach underlying recent works on parametric modeling such as [8] and [9]. In this contribution we illustrate a novel type of data-dependent penalty that improves sparse recovery by adapting to the observed dependence structure. The idea is translated into an algorithm based on COSSO, which is termed ADACOSSO. Full details and analysis on the proposed approach are provided in the manuscript [10]. Here we report the main ideas and present new empirical evidence supporting ADACOSSO.

The structure of the paper is as follows. We begin by presenting our notation and some preliminaries in Section II. In Section III we introduce the formulation for nonparametric sparse recovery and discuss recent theoretical insight in the problem of interest, namely structure detection in the presence of associated input data. We then propose to replace the original block-wise penalties with data-dependent norms adapted to the in-sample dependence (Section IV). Successively, we elaborate on an algorithm that exploits the adaptation to learn a sparse nonparametric model (Section V). In Section VI we present experimental results on synthetic examples and on sparse reconstruction of images. We end the paper in Section VII with some concluding remarks.

II. Notation and Preliminaries

For any $p \in \mathbb{N}$ we use the convention of denoting the set $\{1, \ldots, p\}$ by $\mathbb{N}_p$. We indicate an arbitrary vector $(x_j : j \in \mathbb{N}_d) \in \mathbb{R}^d$ by x and we write e.g. x_j^i to denote the $j-$th component of $x^i \in \mathbb{R}^d$. It is assumed that the input space $\mathcal{X}$ is a compact subset in $\mathbb{R}^d$ and that the output space $\mathcal{Y}$ is a subset of $\mathbb{R}$. We further denote by ρ the underlying probability measure on $\mathcal{X} \times \mathcal{Y}$ and by ξ the corresponding marginal probability measure on $\mathcal{X}$. These measures are both unknown but a finite set of input-output pairs

$$\mathcal{D}_n = \left\{ \left(x^i, y_i\right) \ : \ i \in \mathbb{N}_n \right\} \subset \mathcal{X} \times \mathcal{Y}$$

drawn i.i.d. according to ρ is available. Associated to the latter we have the empirical measure $\rho_n = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}_n} \delta_{(x,y)}$ where $\delta_{(x,y)}$ denotes the Dirac measure at (x, y). For a loss function $l : \mathbb{R} \times \mathcal{Y} \to [0, +\infty]$ the empirical risk of a function $h : \mathcal{X} \to \mathbb{R}$ is defined upon $\mathcal{D}_n$ as

$$C_n(h) = \int_{\mathcal{X} \times \mathcal{Y}} l\left(h\left(x\right), y\right) d\rho_n(x, y) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}_n} l\left(h\left(x\right), y\right). \quad (1)$$

It is the discrete version of its theoretical counterpart, the expected risk measured upon the underlying probability: $C(h) = \int_{\mathcal{X}\times\mathcal{Y}} l\left(h(x), y\right) d\rho(x, y)$. In the following we deal with spaces of real-valued functions defined on $\mathcal{X}$. We write $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ to indicate that the space of functions $\mathcal{H}$ is a Hilbert space defined according to the inner product $\langle\cdot,\cdot\rangle$. In particular, we shall consider the space of square integrable functions $L_2(\mathcal{X}, \xi)$ $\langle f, g\rangle_{L_2(\mathcal{X},\xi)} = \int_{\mathcal{X}} f(x)g(x)d\xi(x)$ and having associated norm $\|f\|_{L_2(\mathcal{X},\xi)} = \sqrt{\langle f, f\rangle_{L_2(\mathcal{X},\xi)}}$. Let now $\left\{\left(\mathcal{H}^j, \langle\cdot,\cdot\rangle_j\right) \ : \ j \in \mathbb{N}_m\right\}$ be a set of reproducing kernel Hilbert spaces of functions defined on $\mathcal{X}$ [11], [12]. For any $j \in \mathbb{N}_m$ denote the reproducing kernel (r.k. in the following) of $\mathcal{H}^j$ by $k^j : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$. It is assumed that for any $j \in \mathbb{N}_m$, the *component space* $\mathcal{H}^j$ is a subspace of $L_2(\mathcal{X}, \xi)$. This is the case under mild conditions on the kernel function and/or $\mathcal{X}$, see [10] and reference therein.

For an index set $\mathcal{A} \subseteq \mathbb{N}_m$ we define the space of (possibly non uniquely defined) additive functions

$$\mathcal{H}_{\mathcal{A}} := \left\{ h : \mathcal{X} \to \mathbb{R} \text{ s.t. } h = \sum_{j\in\mathcal{A}} h^j,\ h^j \in \mathcal{H}^j,\ j \in \mathcal{A} \right\}$$

and use $\mathcal{H}$ to indicate the *model space* $\mathcal{H}_{\mathbb{N}_m}$. Denote by $\mathbb{R}^m_+$ the positive orthant in $\mathbb{R}^m$. The space $\mathcal{H}$ can be endowed with the inner product parametrized in the weight vector $\theta \in \mathbb{R}^m_+$

$$\langle h, g\rangle_\theta = \inf \left\{ \sum_{j\in\mathbb{N}_m} \frac{1}{\theta_j} \left\langle h^j, g^j \right\rangle_j \ : \ h,\ g \in \mathcal{H} \right\} \quad (2)$$

where the infimum is intended over all the possible decompositions giving rise to $h, g \in \mathcal{H}$. The associated norm is $\|f\|_\theta = \sqrt{\langle f, f\rangle_\theta}$. It follows from the arguments in [11] that, when endowed with such an inner product, $\mathcal{H}$ admits a r.k.

$$k^\theta(x, y) = \sum_{j\in\mathbb{N}_m} \theta_j k^j(x, y) \ , \quad (3)$$

see also [12]. In the following we simply write $\langle\cdot,\cdot\rangle$, $\|\cdot\|$ and k when θ has all the entries equal to 1. We say that an arbitrary function $h \in \mathcal{H}$ is *known via its additive representation*, if a specific decomposition $h = \sum_{j\in\mathbb{N}_m} h^j$ is considered. We denote by $\mathcal{A}_h$ its *sparsity pattern*, i.e., we define

$$\mathcal{A}_h := \left\{ j \in \mathbb{N}_m \ : \ h^j \neq 0 \right\} \quad (4)$$

and we further denote by $\mathcal{A}^c_h$ its complementary set. To conclude with the notation used in the following, we introduce the penalty functional

$$J(h) = \inf \left\{ \sum_{j\in\mathbb{N}_m} \|h^j\|_j \ : \ h \in \mathcal{H} \right\} \ . \quad (5)$$

III. Sparse Recovery and Concurvity in Function Spaces

Recent works have been devoted to extend the standard LASSO to a non-parametric setting. With the notation introduced above, this can be formulated as the empirical risk minimization problem defined, for $\tau > 0$, as

$$\hat{h}^\tau = \arg\inf \left\{ C_n(h) + \tau J(h) \ : \ h \in \mathcal{H} \right\} \quad (6)$$

where again the infimum is taken with respect to the additive decompositions and can be shown to be attained. The objective functional trades off the empirical error C_n, with the complexity penalty J encouraging sparsity of the solution with respect to the component functions. Specific instances of the former problem have been considered to accomplish different tasks. One of these, for example, corresponds to selecting the optimal convex combination of kernel functions arising from different parameter values. Here we are especially interested in revealing the structure underlying the input-output data. We shall assume that every kernel in the ensemble encodes a specific feature from the input domain. A particular case that fits into this general idea is found in functional ANOVA models [13]. In this case component functions refer either to main effects or to higher order interactions among subsets of variables.

Traditionally, the study of the properties of estimators like $\hat{h}^\tau$ has been carried out by comparing them with some ideal representation of the process under study: the *oracle function* f, assumed known via its additive representation. This is the approach followed in [14], where, as a main result, a probabilistic bound relating $\hat{h}^\tau$ with f is derived. An interesting oracle to target is the best $d-$sparse additive representation, when it exists, i.e., given an integer $0 < d < m$,

$$f^d = \arg\inf \left\{ C(h) \ : \ h \in \mathcal{H}_{\mathcal{A}},\ \mathcal{A} \subset \mathbb{N}_m,\ \mathrm{card}(A) = d \right\} \ .$$

One interesting question is under what conditions $\hat{h}^\tau$ is close to the oracle, where different specifications of "close" might be of interest. In particular it is relevant to know when we can correctly identify via $\hat{h}^\tau$ the relevant components indexed by the sparsity pattern $\mathcal{A}_f$, i.e., when it does hold that $\mathcal{A}_{\hat{h}^\tau} = \mathcal{A}_f$. This practically amounts to correctly detect the structure behind the data, thus achieving a better insight in the process under study. Existing oracle inequalities [14],[10] support the empirical evidence that the estimator is more likely to include irrelevant terms — where being irrelevant here is dictated by the oracle — as soon as the components in the model space are dependent. As in the analogous conditions for parametric sparse recovery (see e.g. [15]), here the statistical associations play a role via the *intra*-dependence (within the relevant components) and the *inter*-dependence (between functions in $\mathcal{H}_{\mathcal{A}_f}$ and $\mathcal{H}_{\mathcal{A}^c_f}$). For the latter, a crucial measure of association introduced in [14], is the supremum of the cosine of the angle between functions in $\mathcal{H}_{\mathcal{A}_f}$ and $\mathcal{H}_{\mathcal{A}^c_f}$ respectively:

$$\delta(\mathcal{A}_f) = \sup \left\{ \frac{\langle h, g\rangle_{L_2(\mathcal{X},\xi)}}{\|h\|_{L_2(\mathcal{X},\xi)}\|g\|_{L_2(\mathcal{X},\xi)}} \ : \ h \in \mathcal{H}_{\mathcal{A}_f},\ g \in \mathcal{H}_{\mathcal{A}^c_f} \right\} \ . \quad (7)$$

We call this the *mutual coherence in the model space*, as it is similar to the ordinary notion of mutual coherence of a predefined finite dictionary of functions. This index can

be seen as a partial characterization of *concurvity*, namely the (approximate) collinearity of the transforms of the input random variables contained in the model space [16].

IV. Data Matched Penalties

To prevent the inclusion of spurious terms, one might constrain the set of feasible functions in the model space, so that the value of concurvity indices, such as (7), are kept under control. However this is unpractical as we do not have access neither to $\mathcal{A}_f$ nor to ξ defining the inner product in the formula of the mutual coherence. Instead, in this Section we propose a data-dependent approach to improve the sparse recovery according to the in-sample dependence. The framework that we are about to discuss is in a similar spirit as the data-dependent geometric regularization, introduced in the context of semi-supervised learning in [17]. The aim is to incorporate in the learning task the geometry of the underlying probability distribution of the input data. We begin by introducing a class of norms for the component spaces, which is instrumental for our approach.

A. Data-Dependent Adaptive Norms

Denote by ξ_n the empirical marginal measure on $\mathcal{X}$. Consider the space $L^2(\mathcal{X}, \xi_n)$, isomorphic to $\mathbb{R}^n$, and endow it with the inner product

$$\langle f, g \rangle_{L^2(\mathcal{X},\xi_n)} = \frac{1}{n} \sum_{x \ : \ (x,y) \in \mathcal{D}_n} f(x) g(x)$$

and associated norm $\|f\|_{L^2(\mathcal{X},\xi_n)} = \sqrt{\langle f, f \rangle_{L^2(\mathcal{X},\xi_n)}}$.

Our starting point is the set of component spaces $\left\{ \left(\mathcal{H}^j, \langle \cdot, \cdot \rangle_j \right) \ : \ j \in \mathbb{N}_m \right\}$ introduced above. It is assumed that $m \leq n$. That is, we have at most as many components as points in the sample. For $j \in \mathbb{N}_m$ denote by A^j both a linear operator $A^j : L^2(\mathcal{X}, \xi_n) \to L^2(\mathcal{X}, \xi_n)$, the nature of which will be clear later, and the associated $n \times n$ matrix representation. Further let $S^j : \mathcal{H}^j \to L^2(\mathcal{X}, \xi_n)$ be $S^j := A^j \bullet I_{\mathcal{D}_n}$ where $I_{\mathcal{D}_n}\left(h^j\right) = \left(h^j(x) \ : \ (x,y) \in \mathcal{D}_n \right)$ is the vector of evaluations of h^j according to the input points in $\mathcal{D}_n$ and $\bullet$ is used to indicate function composition. We now modify the norm of the component spaces by incorporating a new term depending upon the sample. Namely for $h^j \in \mathcal{H}^j$ and $\mu > 0$ we define the new norm

$$\left\| h^j \right\|^2_{\mathcal{D}_n, j} := \left\| h^j \right\|^2_j + \mu n \left\| S^j(h^j) \right\|^2_{L^2(\mathcal{X},\xi_n)} \ . \tag{8}$$

When endowed with this norm, $\mathcal{H}^j$ can be shown to be a RKHS with the r.k. $k^j_{\mathcal{D}_n} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined by:

$$k^j_{\mathcal{D}_n}(x,y) = k^j(x,y) - \left(\bar{k}^j(x) \right)^\top \left(\frac{1}{\mu} I + M^j K^j \right)^{-1} M^j \bar{k}^j(y) \tag{9}$$

where $\bar{k}^j(x)$ is the vector $\left(k^j(z,x) \ : \ (z,y) \in \mathcal{D}_n \right)$, $M^j = {A^j}^\top A^j$ and K^j is the kernel matrix arising from the kernel function k^j and the points in the given sample. A proof is essentially based on simple orthogonality arguments and can be found in [18]. Notice that we have not yet specified the nature of A^j; a specific choice will play a central role in our derivation, as it will be clear in a moment.

Based on the set of norms corresponding to (8) we define the new norm for the space of additive models $\mathcal{H}$ by setting for a weight vector $\theta \in \mathbb{R}^m_+$

$$\|h\|^2_{\theta,\mathcal{D}_n} = \inf \left\{ \sum_{j \in \mathbb{N}_m} \frac{1}{\theta_j} \left\| h^j \right\|^2_{\mathcal{D}_n, j} \ : \ h \in \mathcal{H} \right\} \ . \tag{10}$$

Correspondingly, the associated reproducing kernel is now

$$k_{\theta,\mathcal{D}_n}(x,y) = \sum_{j \in \mathbb{N}_m} \theta_j k^j_{\mathcal{D}_n}(x,y) \ . \tag{11}$$

We again drop the subscript θ in the case of unitary weights. Let now $\{\mathcal{R}^j \ : \ j \in \mathbb{N}_m\}$ be a set of one-dimensional mutually orthogonal subspaces of $L^2(\mathcal{X}, \xi_n)$. Denote by $P^j_\perp$ the projector operator mapping an element of $L^2(\mathcal{X}, \xi_n)$ onto the orthogonal complement of $\mathcal{R}^j$. We address the adaptation in a supervised way by formulating the following *data-dependent norm adaptation problem*. For $\lambda_0 > 0$, consider finding $\left\{ P^j_\perp \ : \ j \in \mathbb{N}_m \right\}$ and $h = \sum_{j \in \mathbb{N}_m} h^j \in \mathcal{H}$ that minimize:

$$Q\left(h, \left\{ P^j_\perp \ : \ j \in \mathbb{N}_m \right\} \right) =$$
$$C_n(h) + \lambda_0 \sum_{j \in \mathbb{N}_m} \left(\|h^j\|^2_j + \mu n \left\| P^j_\perp \bullet I_{\mathcal{D}_n}\left(h^j\right) \right\|^2_{L^2(\mathcal{X},\xi_n)} \right) . \tag{12}$$

By minimizing (12) we fit an additive model to the data meanwhile penalizing the coherence with respect to the empirical measure. This is achieved by driving the vectors in $\left\{ I_{\mathcal{D}_n}\left(h^j\right) \ : \ j \in \mathbb{N}_m \right\}$ to live in orthogonal subspaces. The empirical measure is used as a proxy for the underlying measure and the approach is readily motivated in the spirit of the oracle inequalities. Notice however that while the analysis in the latter is based on a worst-case scenario, the "decorrelation" of the components we aim at is data-driven. When μ is set to zero the problem corresponds to a standard penalized empirical risk minimization. As μ is increased, the smoothness of the solution is traded for the geometric property of having less inter-dependence with respect to the empirical measure.

B. An Algorithm for the Data-Dependent Norm Adaptation Problem

For a fixed set of projectors $\left\{ P^j_\perp \ : \ j \in \mathbb{N}_m \right\}$, (12) can be restated as

$$Q(h) = C_n(h) + \lambda_0 \|h\|^2_{\mathcal{D}_n} \ , \tag{13}$$

which is a regularized empirical risk functional with data-dependent squared norm penalty. It is well known that a minimizer of the latter admits a representation as $\sum_i \alpha_i k_{\mathcal{D}_n}\left(x^i, \cdot\right)$ where $\alpha \in \mathbb{R}^n$ and $k_{\mathcal{D}_n}$ is given by (11) in terms of the modified kernel functions (9) with $M^j = P^j_\perp$ and μ fixed at a predefined value. Such a minimizer can be computed

by solving an optimization problem which depends on the choice of the loss function. In the case of quadratic loss a solution is available in closed form and can be computed solving a system of linear equations ([12]; [19]). On the other hand, if we fix $h \in \mathcal{H}$, the minimization of (12) amounts to find projection matrices $\{P^j \ : \ j \in \mathbb{N}_d\}$ that minimize $\sum_{j\in\mathbb{N}_m} \left\| \left(I_{\mathcal{D}_n} - P^j \bullet I_{\mathcal{D}_n}\right)\left(h^j\right)\right\|^2_{L^2(\mathcal{X},\xi_n)}$ where for any $j \in \mathbb{N}_m$, P^j is the projection onto $\mathcal{R}^j$. Denote now by H the $n \times m$ matrix with $j-$th column $I_{\mathcal{D}_n}\left(h^j\right)$ and let $\|\cdot\|_F$ be the Frobenius norm. Further, for a positive integer s, denote by I_s the $s \times s$ identity matrix and by $\mathrm{diag}(x)$ a diagonal matrix with diagonal $x \in \mathbb{R}^s$. The latter problem can be equivalently formulated as

$$\min\Big\{\|H - Q\,\mathrm{diag}(\lambda)\|_F \ : \ Q^\top Q = I_m,\ Q \in \mathbb{R}^{n\times m},\ \lambda \in \mathbb{R}^m\Big\}\ . \quad (14)$$

Once an approximation $Q\,\mathrm{diag}(\lambda)$ is found, we simply set $P^j_\perp = I_n - q^j(q^j)^\top$ where q^j is the $j-$th column of Q. In turn (14) can be solved efficiently, see [10, Section 5.3] for details. The pseudocode for a solution strategy exploiting the facts above is reported on Table I. By applying a limited number

Input: Data $\mathcal{D}_n$; *reproducing kernels* $k^j \ : \mathcal{X}\times\mathcal{X}\to\mathbb{R},\ j\in\mathbb{N}_m$
Output: *data-dependent kernels* $k^j_{\mathcal{D}_n} \ : \mathcal{X}\times\mathcal{X}\to\mathbb{R},\ j\in\mathbb{N}_m$.
$(k^j_{\mathcal{D}_n}, \|\cdot\|_{\mathcal{D}_n,j}) \ \leftarrow (k^j, \|\cdot\|_j)\ \forall\ j\in\mathbb{N}_m$
repeat
$\hat{h} \leftarrow \arg\min\left\{C_n(h) + \lambda_0\|f\|^2_{\mathcal{D}_n} \ : \ h\in\mathcal{H}\right\}$
$\hat{H} \leftarrow \left(I_{\mathcal{D}_n}\left(\hat{h}^1\right),\ldots,I_{\mathcal{D}_n}\left(\hat{h}^m\right)\right)$
$(\hat{Q},\hat{\lambda}) \leftarrow \underset{Q^\top Q = I_m,\ \lambda\in\mathbb{R}^m}{\arg\min}\left\{\|\hat{H} - Q\,\mathrm{diag}(\lambda)\|_F\right\}$
for $\hat{Q} = (\hat{q}^1,\ldots,\hat{q}^m)$, $P^j_\perp \leftarrow I_n - \hat{q}^j\hat{q}^{j\top}\ \forall\ j\in\mathbb{N}_m$
set data-dependent kernels via (9) with $M^j = P^j_\perp$
until stopping criterion met

TABLE I
PSEUDOCODE FOR THE DATA-DEPENDENT NORM ADAPTATION

of iterations of our algorithm, we obtain what can be seen as an iterative data-dependent improvement on the original set of norms and corresponding kernels. Thus our stopping criterion is simply determined by a predefined number of iterations (10 in our experiments).

V. SPARSE FUNCTIONAL ANOVA MODELS WITH THE ADAPTIVE COSSO

In this section we elaborate on the use of data matched penalties for learning sparse models. We thus go back to the estimator (6) where J is substituted by its data-dependent version $J_{\mathcal{D}_n}$, obtained by replacing the norms with the data dependent ones. Correspondingly, this amounts to replacing the original set of kernels with the one introduced in Subsection IV-B. We consider the case of quadratic loss so that $C_n(h) = \sum_{(x,y)\in\mathcal{D}_n}(h(x)-y)^2$. This problem has been studied in [6] for the standard penalty J. The authors also consider an unpenalized bias term. Including the bias in the present setting involves minor changes and it has not been reported above for simplicity of presentation. The analysis in [6] is done explicitly in terms of smoothing spline ANOVA models. For this case, the mathematical background as well as the closed form for the ensemble of kernel functions can be found e.g. in [12]. Although there is no need to stick to this setting, the former is a solid formalism to deal with multivariate data. Moreover, the original kernels in the ensemble have the advantage of being parameter-free.

Recall that for $\theta \in \mathbb{R}^m_+$, $\|\cdot\|_{\theta,\mathcal{D}_n}$ is the weighted data-dependent norm (10). For fixed τ, finding a solution for the data-dependent version of (6) with an unpenalized bias amounts to finding in a predefined model space $\mathcal{H}\oplus\{1\}$ the solution of :

$$\min\Big\{C_n(h) + \lambda_0\|h\|^2_{\theta,\mathcal{D}_n} + \lambda\sum_{j\in\mathbb{N}_m}\theta_j \ : \ h\in\mathcal{H}\oplus\{1\}, \theta\in\mathbb{R}^m_+\Big\}. \quad (15)$$

Here $\{1\}$ is the space of unpenalized constant functions, the pair $(\lambda_0,\lambda)\in\mathbb{R}^2_+$ has a known relation with τ, and λ_0 can be fixed at any value. Proofs and details can be found in [6]. In the following we compare the performance of the estimator corresponding to J (COSSO) with the one corresponding to $J_{\mathcal{D}_n}$ (ADACOSSO). In [6] a solution for the standard COSSO is found by first setting as an initial value $\theta_j = 1$ for all $j \in \mathbb{N}_m$. At this point we perform the data-dependent adaptation as detailed in Subsection IV-B. Once the set of modified kernels has been obtained, we estimate the model

$$\hat{f}(x) = \sum_{x^i \ : \ (x^i,y_i)\in\mathcal{D}_n} \hat{\alpha}_i k_{\hat{\theta},\mathcal{D}_n}(x^i,x) + \hat{b} \quad (16)$$

via the same alternating approach as in [6]. In the optimization some θ_j are shrunk to zero, thus ensuring sparsity in the component functions.

VI. EXPERIMENTAL RESULTS

A. Synthetic Datasets

We tested the performance of the two procedures on a number of synthetic examples. As building blocks for our generating models we employed the same collection of univariate functions used in [6]:

$$g_1(z) = z\ ,\ g_2(z) = (2z-1)^2\ ,\ g_3(z) = \frac{\sin(2\pi z)}{2-\sin(2\pi z)}\ ,$$
$$g_4(z) = 0.1\sin(2\pi z) + 0.2\cos(2\pi z) + 0.3\sin^2(2\pi z) + 0.4\cos^3(2\pi z) + 0.5\sin^3(2\pi z)\ .$$

We considered 4 blocks of input covariates of equal size v. Within each block, we simulated input data according to the *Compound Symmetry* scheme of [6]. That is, denoting by $\mathcal{B}_l = \{b_{l1},\ldots,b_{lv}\}$ the index set of the $l-$th block, we generated $x_j = (w_j + tu_l)/(t+1)$ for $j\in\mathcal{B}_l,\ l\in\mathbb{N}_4$, where w_j and u_l were both i.i.d. following a uniform distribution in $[0,1]$. In this way the parameter t can be used to induce a certain degree

of concurvity between the functions of variable belonging to the same block. The generating function was taken to be

$$f(x) = 6g_1(x_{b_{11}}) + 6g_2(x_{b_{21}}) + 3g_3(x_{b_{31}}) + 3g_4(x_{b_{41}}) \quad (17)$$

and therefore $4(v-1)$ covariates were not used in the generating process. For $i \in \mathbb{N}_n$, we drew x^i from the distribution detailed above. The corresponding output observation was set to $y_i = f(x^i) + \epsilon_i$ where the variance of the zero-mean gaussian variable ϵ was fixed so that the signal-to-noise ratio $\mathrm{SNR} = \mathrm{var}[f(x)]/\mathrm{var}[\epsilon]$ was at a pre-specified value. The comparison was done in terms of both selection and prediction accuracy. The selection accuracy was assessed by means of two popular measures in information retrieval: *Precision* $\mathrm{Pr} = \mathrm{card}\left(\mathcal{A}_{\hat{h}} \cap \mathcal{A}_f\right)/\mathrm{card}\left(\mathcal{A}_{\hat{h}}\right)$, and *Recall* $\mathrm{Re} = \mathrm{card}\left(\mathcal{A}_{\hat{h}} \cap \mathcal{A}_f\right)/\mathrm{card}\left(\mathcal{A}_f\right)$. The prediction accuracy was estimated via $\mathrm{RSS} = \frac{1}{n_t}\left\|y_t - \hat{y}_t\right\|^2_{\mathbb{R}^{n_t}}$, where y_t is the target vector associated to an i.i.d. test sample, $\hat{y}_t$ is the corresponding prediction and $n_t = 1000$. For both the COSSO and the initial kernel functions in ADACOSSO we considered smoothing spline ANOVA kernels and took the whole set of main effects associated to the input covariates. Following [6] we set λ at a convenient value in association with the original kernel functions. We then tuned both the shrinkage parameter of the COSSO and the pair (λ, μ) in the ADACOSSO via $5-$fold CV. In Table II we report the results averaged over 100 simulations for different values of v, n, t and SNR. In [10] a different dependence structure was used in the simulations and the BIC was considered instead of CV as a model selection criterion. As shown by Pr and Re, ADACOSSO improved the detection of the structure of the oracle, here identified with the generating function (17). Also, better prediction performances (RSS) were obtained. This is because in the non-adaptive procedure relevant and irrelevant components are equally penalized and thus in the final model relevant terms tend to be overly shrunk. Thanks to the adaptive approach in the ADACOSSO, relevant components (according to C_n in (12)) tend to be less penalized in the final estimate.

TABLE II
SELECTION AND PREDICTION ACCURACY AT DIFFERENT REGIMES.

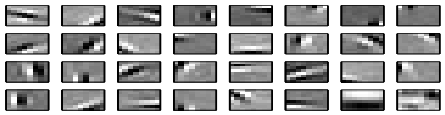
$n = 40$, $v = 5$, SNR $= 6$.				
		$t = 1.5$	$t = 3$	$t = 4.5$
Pr	COSSO	0.38(0.26)	0.21(0.19)	0.18(0.12)
	ADACOSSO	**0.63**(0.26)	**0.54**(0.21)	**0.49**(0.20)
Re	COSSO	0.51(0.24)	0.51(0.30)	0.51(0.30)
	ADACOSSO	**0.64**(0.24)	**0.66**(0.25)	**0.67**(0.24)
RSS	COSSO	6.23(0.98)	5.43(0.81)	4.75(0.74)
	ADACOSSO	**5.24**(0.82)	**4.53**(0.86)	**3.93**(0.71)
$n = 100$, $v = 10$, SNR $= 3$.				
		$t = 1.5$	$t = 3$	$t = 4.5$
Pr	COSSO	0.54(0.24)	0.30(0.14)	0.25(0.12)
	ADACOSSO	**0.77**(0.17)	**0.62**(0.19)	**0.56**(0.16)
Re	COSSO	0.85(0.15)	0.75(0.22)	0.71(0.20)
	ADACOSSO	**0.89**(0.15)	**0.81**(0.20)	**0.78**(0.18)
RSS	COSSO	5.75(0.73)	5.35(0.53)	5.03(0.40)
	ADACOSSO	**5.21**(0.47)	**5.03**(0.48)	**4.72**(0.44)
$n = 100$, $v = 20$, SNR $= 10$.				
		$t = 1.5$	$t = 3$	$t = 4.5$
Pr	COSSO	0.54(0.21)	0.29(0.12)	0.23(0.10)
	ADACOSSO	**0.81**(0.15)	**0.65**(0.16)	**0.6**(0.15)
Re	COSSO	0.86(0.18)	0.81(0.17)	0.68(0.21)
	ADACOSSO	**0.97**(0.09)	**0.91**(0.13)	**0.87**(0.17)
RSS	COSSO	3.86(0.62)	3.03(0.41)	2.68(0.30)
	ADACOSSO	**3.20**(0.44)	**2.70**(0.35)	**2.40**(0.31)
$n = 200$, $v = 25$, SNR $= 6$.				
		$t = 1.5$	$t = 3$	$t = 4.5$
Pr	COSSO	0.77(0.13)	0.43(0.16)	0.32(0.14)
	ADACOSSO	**0.88**(0.19)	**0.73**(0.14)	**0.64**(0.14)
Re	COSSO	1(0)	0.92(0.09)	0.80(0.13)
	ADACOSSO	1(0)	**0.96**(0.12)	**0.91**(0.17)
RSS	COSSO	3.09(0.39)	2.98(0.27)	2.80(0.23)
	ADACOSSO	**3.03**(0.32)	**2.87**(0.24)	**2.62**(0.23)

B. Non-parametric Sparse Reconstruction of Natural Images

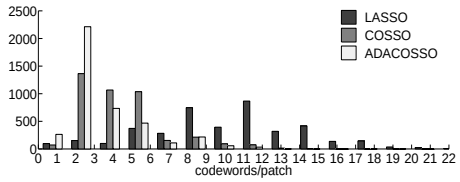
In [20] sparse coding of natural images was investigated with the purpose of understanding the response properties of visual neurons. The authors proposed a learning algorithm to derive a dictionary of codewords. The coding strategy was designed to ensure that image patches are likely to be represented by a sparse linear superposition of basis elements. We applied their approach and as training data we used patches extracted from the set of monochrome images available at the authors' website. In this way, we found 64 codewords of 8×8 pixels each (Figure 1(a)). We then aimed at reconstructing one of the images in the set (Figure 1(c)) based upon the dictionary elements. We divided the picture in 4096 patches of the same dimensions as the codewords. Each patch is represented as a vector $y^l \in \mathbb{R}^{64}$ for $l \in \mathbb{N}_{4096}$. The reconstruction amounts at approximating each $i-$th entry of the $l-$th patch y^l_i by $\hat{y}^l = \sum_{j \in \mathbb{N}_{64}} h^j(x^i_j)$ where x^i_j is the $i-$th pixel in the vector representation for the $j-$th codeword. Each function h^j is either a linear function of its argument (linear sparse coding) or a non-parametric function with some vanishing components. The idea of relaxing the linearity assumption was proposed under the name of *functional sparse coding* in [21]. Here we applied the standard LASSO to find a sparse linear fit and determined sparse non-parametric fits via both COSSO and ADACOSSO. For each patch we trained models for different values of the parameters with the whole set of 64 input-output pairs. To assess the quality of an approximation we computed the peak signal-to-noise ratio $\mathrm{PSNR} = 20\log_{10}\left(255/\sqrt{1/64\|y^l - \hat{y}^l\|^2}\right)$. The whole set of solutions corresponding to different parameters was considered. For each patch we looked for the model that minimized the total numbers of codewords used, subject to an average value of PSNR over the population of patches greater than 60 dB. This was approximatively the value of PSNR obtained on average by the LASSO in a previous experiment. This type of search over the set of models is an integer problem that can be relaxed into a standard linear programming problem. Figure 1(b) reports the distribution of codewords/patch obtained with this model selection approach. A considerable gain in sparseness was achieved with the non-parametric procedures, especially when the norm adaptation step was performed.

VII. CONCLUSION

In this paper we have proposed a novel data-dependent approach that improves the sparse recovery in the presence of associated input covariates. The idea has been employed in combination with the algorithmical strategy used in COSSO. However, we conclude by stressing that the norm adaptation



(a) A subset of the codewords extracted using the learning algorithm in [20].



(b) The distribution of codewords/patch for the second experiment. The mean of the distributions are 10.08, 4.51 and 3.64 respectively for LASSO, COSSO and ADACOSSO.



(c) The image reconstructed via ADACOSSO indistinguishable from original.

Fig. 1. Sparse reconstruction of a natural image.

problem can be specialized to other loss functions and used in combination with other MKL algorithms. Indeed, it copes with a problem that is not attached to the specific algorithmical implementation. Rather, it deals with the intrinsic nature of sparse recovery in function spaces.

ACKNOWLEDGMENT

Research supported by Research Council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, CIF1, STRT1/08/023, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, G.0320.08, G.0558.08, G.0557.08, G.0588.09, G.0377.09; Research Communities ICCoS, ANMMM and MLDM; the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization", 2007–2011), EU: ERNSI. IWT: PhD Grants, McKnow-E, Eureka-Flite+, SBO LeCoPro, SBO Climaqs.

REFERENCES

[1] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[2] C. Micchelli and M. Pontil, "Learning the Kernel Function via Regularization," *Journal of Machine Learning Research*, vol. 6, pp. 1099–1125, 2005.

[3] ——, "Feature space perspectives for learning the kernel," *Machine Learning*, vol. 66, no. 2, pp. 297–319, 2007.

[4] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[5] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," *ACM International Conference Proceeding Series*, 2004.

[6] Y. Lin and H. Zhang, "Component selection and smoothing in multivariate nonparametric regression," *Annals of Statistics*, vol. 34, no. 5, pp. 2272–2297, 2006.

[7] S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," *Advances in Neural Information Processing Systems*, 2009.

[8] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, pp. 3468–3497, 2009.

[9] R. Jenatton, J. Audibert, and F. Bach, "Structured Variable Selection with Sparsity-Inducing Norms," *Technical Report - http://arxiv.org/abs/0904.3523v1*, 2009.

[10] M. Signoretto, K. Pelckmans, L. De Lathauwer, and J. Suykens, "Data-dependent Norm Adaptation for Sparse Recovery in Kernel Ensembles Learning," *Technical Report 09-97 ESAT SCD/SISTA*, 2009.

[11] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, pp. 337 – 404, 1950.

[12] G. Wahba, *Spline Models for Observational Data*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia: SIAM, 1990, vol. 59.

[13] C. Gu, *Smoothing spline ANOVA models*, Springer, Ed. Springer, 2002.

[14] V. Koltchinskii and M. Yuan, "Sparse Recovery in Large Ensembles of Kernel Machines." COLT, 2008.

[15] P. Zhao and B. Yu, "On Model Selection Consistency of Lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[16] A. Buja, T. Hastie, and R. Tibshirani, "Linear smoothers and additive models," *Annals of Statistics*, vol. 17, no. 2, pp. 453–510, 1989.

[17] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[18] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *ICML*, vol. 22, 2005, pp. 824–831.

[19] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least squares support vector machines*. World Scientific, 2002.

[20] B. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[21] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, "Sparse additive models," *to appear in Journal of the Royal Statistical Society. Series B (Methodonlogical)*.