A Regularization Approach to Nonlinear Variable Selection

L. Rosasco
CBCL, McGovern Institute, MIT
43 Vassar Street,
Cambridge, MA 02139, United States

M. Santoro, S. Mosci, A. Verri
DISI, Università di Genova
via Dodecaneso 35,
16146, Genova, Italy

S. Villa
DIMA, Università di Genova
via Dodecaneso 35,
16146, Genova, Italy

Abstract

In this paper we consider a regularization approach to variable selection when the regression function depends nonlinearly on a few input variables. The proposed method is based on a regularized least square estimator penalizing large values of the partial derivatives. An efficient iterative procedure is proposed to solve the underlying variational problem, and its convergence is proved. The empirical properties of the obtained estimator are tested both for prediction and variable selection. The algorithm compares favorably to more standard ridge regression and ℓ_1 regularization schemes.

1 Introduction

In this work, we are interested into the variable selection problem within the supervised learning framework, when the input-output dependence is described by a nonlinear model.

Given a set of input-output pairs $\mathbf{z} = (x_i, y_i)_{i=1}^n$ sampled i.i.d. with respect to an unknown probability measure ρ, the task of supervised learning is to estimate the regression function $f_\rho(x) = \mathbb{E}[y|x]$. The output space $\mathcal{Y}$ can be either a subset of $\mathbb{R}$ or simply $\{+1, -1\}$ in binary classification. In general,the input space is $\mathcal{X} \subset \mathbb{R}^p$. The problem we have in mind is motivated by the assumption that f_ρ depends on $d \ll p$ variables only. The key question is how to turn this assumption into a learning algorithm. We study this problem in the context of regularization where – given the empirical risk $\mathcal{E}$ – one aims at designing a regularizer (penalty) $\mathcal{R}$, so that

$$\operatorname*{argmin}_{f \in \mathcal{H}} \{\mathcal{E}(f) + \tau \mathcal{R}(f)\}$$

provides us with an estimator that is a (nonlinear) function of few variables. Towards this end we will propose a new penalty which enforces the partial derivatives of the learned estimator to be *small* in an appropriate sense. The *size* of the partial derivatives of the obtained estimator indicates which variables can be discarded.

The variational problem corresponding to this new functional cannot be solved using simple gradient descent methods. Towards this end we develop an iterative projection procedure based on a forward-backward splitting approach (Daubechies et al., 2007; Lions and Mercier, 1979; Combettes and Wajs, 2005; Hale et al., 2008; Rosasco et al., 2009). Convergence of the method can be proved and the entire regularization path for the algorithm can be efficiently computed using a simple continuation method (Hale et al., 2008). The performance of the proposed algorithm is assessed on toy data as well as on a benchmark data set.

The problem of variable selection has recently received considerable attention, motivated by high dimensional learning problems such as microarray data analysis (Golub et al., 1999). The last few years witnessed substantial progresses in the case when the regression function is described by a linear model – see Lal et al. (2006) for a survey – and regularization with a sparsity penalty (ℓ_1 and its variations) proved to be a suitable strategy (Tibshirani, 1996; Efron et al., 2004). In the case of nonlinear models the situation is much less understood. A number of authors consider a dictionary of nonlinear features and use a sparsity-based algorithm to select the relevant elements of the dictionary. The simplest of such approaches is that of considering a generalized linear model where the regression function is assumed to be a linear combination of nonlinear functions, each one depending on a single variable only, see for example Ravikumar et al. (2008). A more advanced approach would be to consider dictionaries

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

encoding more complex interactions among the variables (Lin and Zhang, 2006). For example, one could consider the features defined by a second-degree polynomial kernel. The shortcoming of this approach is that the size of the dictionary grows more than exponentially as one considers high order interactions. Recently, Bach (2008) showed that it is still possible to learn with such dictionaries if the subsets of variables that can be selected (the sparsity patterns) are suitably restricted (e.g. the atoms of the dictionary can be embedded into a directed acyclic graph). Our approach differs from these latter works since we do not try to design dictionaries encoding variables interactions but we use partial derivatives to derive a new regularizer that induces a different form of sparsity.

We recall that a number of heuristics for nonlinear feature selection are surveyed by Guyon et al. (2006). We also mention a series of related works considering the problem of nonlinear dimensionality reduction, either in the supervised setting (see Wu et al. (2008) and references therein) or in the unsupervised setting – see KPCA (Schölkopf and Smola, 2002) and manifold learning (Belkin and Niyogi, 2008). More recently, a method based on estimating the gradient of the regression function is proposed by Mukherjee and Zhou (2006). We note that the problem of subset selection that we consider in this paper is somewhat different, because rather than *extracting new* features we just aim at *selecting* the most relevant variables. Indeed, in all the dimensionality reduction methods, variable selection is not embedded in the training phase and can be achieved only as a postprocessing.

The paper is organized as follows. In Section 2 we present a new regularized functional, where the penalty is based on partial derivatives, and propose an iterative algorithm for finding its minimizer. In Section 3 we describe the derivation of the proposed algorithm. Finally in Section 4 the proposed scheme is empirically evaluated and compared with benchmark subset selection techniques as standard ridge regression on synthetic and real data.

Notation: In the following, we will use lower indices to indicate different elements of a space – e.g. $x_i \in \mathcal{X}$ – and upper indices to indicate different components of a vector – e.g. $x = (x^j)_{j=1}^p \in \mathcal{X}$. We use indices $i, s \in \{1, \ldots, n\}$ for the examples and $j, l \in \{1, \ldots p\}$ for coordinates.

2 Regularization for Nonlinear Subset Selection

In the following we study the minimization problem

$$\begin{aligned} f_\tau \quad = \quad & \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \\ & 2\tau \sum_{j=1}^p \sqrt{\frac{1}{n} \sum_{i=1}^n |\partial_j f(x_i)|^2} \end{aligned} \tag{1}$$

where $\partial_j f(x)$ denotes the j^{th} partial derivative of f at some point x and the hypotheses space, $\mathcal{H}$, is assumed to be a reproducing kernel Hilbert space of smooth functions. Recall that our primary goals are to *estimate* the true underlying f_ρ as well as to *detect* the subset of relevant variables, *given* the training set $\mathbf{z}$.

The above penalty searches for an estimator for which each partial derivative is *small* when evaluated on most points of the training set. This choice is suggested by the fact that, if a function does not depend on a variable, the corresponding partial derivative will be zero (or small) for all (or most) points, and such a variable can be considered *irrelevant*. In theory, a natural choice in order to enforce the j^{th} derivative to be small "on average" would be to penalize the $L^2(X, \rho_X)$ norm

$$\|\partial_j f\|_\rho = \sqrt{\int_{\mathcal{X}} |\partial_j f(x)|^2 d\rho_X} \tag{2}$$

where ρ_X is the probability distribution of the input points. Since in practice ρ_X is unknown, and we only have an i.i.d. sample $\mathbf{x} = (x_1, \ldots, x_n)$, a feasible alternative to (2) is its empirical counterpart, which is the one used in (1). To build a regularizer we then have to consider the different partial derivatives at once, and different regularizers can be obtained considering different norms for the vector $\vec{R}(f) \in \mathbb{R}^p$, whose j^{th} element is:

$$\vec{R}_j(f) = \sqrt{\frac{1}{n} \sum_{i=1}^n |\partial_j f(x_i)|^2}.$$

The most natural choice is probably

$$\mathcal{R}_0(f) = \|\vec{R}(f)\|_0 = \#\{j \mid \vec{R}_j(f) \neq 0\}.$$

When considering linear functions, $\mathcal{H} = \{f \ : \ f(x) = \beta \cdot x, \beta \in \mathbb{R}^p\}$, the above penalty is the so called ℓ_0 norm, $\|\beta\|_0$, which counts the number of non-zero coefficients. It is well known that there are no efficient algorithms to minimize $\mathcal{E} + \tau \mathcal{R}_0$, and ℓ_1 regularization – i.e. using the ℓ_1 norm $\|\beta\|_1$ – is an efficient convex relaxation to this problem. Reasoning along the same line we replace $\mathcal{R}_0$ with

$$\mathcal{R}_1(f) = \|\vec{R}(f)\|_1 = \sum_{j=1}^p |\vec{R}_j(f)|,$$

Algorithm 1 Iterative Projection Algorithm

```
Require: $\sigma, \eta, \tau > 0$
Initialize: $\alpha_0 = 0, \beta_0 = 0,\ t = 0$
while convergence not reached do
    $\alpha_t = \alpha_{t-1} - (2\sigma)^{-1}\left(K\alpha_{t-1} + Z\beta_{t-1} - \mathbf{y}\right)$
    set $v_0 = 0,\ q = 0$ and
    for $j = 1, \dots p$ do
        while convergence not reached do
            $v^j_{q+1} = \frac{v^j_q - \eta\left(L^j\left(v_q - \frac{\sigma}{\tau}\beta_{t-1}\right) - \frac{\sigma}{\tau}Z_j\alpha_t\right)}{1 + \eta\|L^j\left(v_q - \frac{\sigma}{\tau}\beta_{t-1}\right) - \frac{\sigma}{\tau}Z_j\alpha_t\|_n},$
            $q := q + 1$
        end while
        set $\bar{v}^j_t = v^j_q$
    end for
    $t = t + 1, \qquad \beta_t = \beta_{t-1} - \frac{\tau}{\sigma}\bar{v}_t.$
end while
return $(\alpha_t, \beta_t)$
```

so that the functional in (1) is given by $\mathcal{E} + 2\tau\mathcal{R}_1$. Note that, indeed, the proposed penalty reduces to ℓ_1 regularization in the case of linear models. Though convexity makes the corresponding algorithm more tractable, the solution of the underlying variational problem is not straightforward due to nonsmoothness and we propose an efficient optimization procedure in the next section.

2.1 An Iterative Projection Algorithm

We will show (see the beginning of Subsec. 3.3) that if $\mathcal{H}$ is a RKHS of sufficiently smooth functions, a straightforward generalization of the representer theorem ensures that a generic solution of problem (1) can be conveniently written as a linear combination:

$$f_\tau(x) = \sum_{i=1}^{n} \frac{1}{n}\alpha_i k(x_i, x) + \sum_{i=1}^{n}\sum_{j=1}^{p} \frac{1}{n}\beta^j_i z_{j,x_i}(x), \quad (3)$$

where $\alpha, \beta^j \in \mathbb{R}^n$ for all $j = 1, \dots, p$ and

$$z_{j,x_i}(x') = \frac{\partial k(x', x)}{\partial x^j}\Big|_{x=x_i}. \quad (4)$$

In order to compute the coefficients, let us introduce the two vectors $\beta = (\beta^1, \dots, \beta^p)$ and $v = (v^1, \dots, v^p)$, where $v^j \in \mathbb{R}^n$ for all $j = 1, \dots, p$. Also, let's define the matrices K, Z_j, L_j as

$$(K)_{i,s} = \frac{1}{n}k(x_i, x_s),$$

$$(Z_j)_{i,s} = \frac{1}{n}z_{j,x_i}(x_s)$$

and

$$(L_j\beta)_i = \sum_{l=1}^{p}\sum_{s=1}^{n} L_{j,l}(x_i, x_s)\beta^l_s,$$

where

$$L_{j,l}(x_i, x_s) = \frac{1}{n^2}\frac{\partial^2 k(x, x')}{\partial x^j \partial x'^l}\Big|_{x=x_i, x'=x_s}.$$

As shown in the next section, the coefficients can be computed using the iterative Algorithm 1.

Before describing the derivation of this algorithm and discussing its convergence properties, we add several remarks. First, the above algorithm follows recent works studying optimization procedures for ℓ_1-based regularization and related algorithms – see Hale et al. (2008), Daubechies et al. (2007), Figueiredo et al. (2007) – and more general learning schemes – see for example Rosasco et al. (2009).

Second, the proposed procedure requires the choice of an appropriate stopping rule, which will be discussed later, and of the step sizes σ, η. Simple a priori choices ensuring convergence are discussed in the following section and are the one we used in our experiments. We expect more sophisticated step size choices to considerably speed up the iteration, at the price of a more complicated convergence analysis, which is outside the scope of the paper.

Third, while computing solutions corresponding to different regularization parameters we use the continuation method suggested by Hale et al. (2008). Starting from a large regularization parameter value, the obtained solution is used to initialize the algorithm for the next smaller regularization parameter value and the same strategy is iterated for all the other parameter values. We found this warm restart procedure to greatly improve the computational requirement needed to calculate the whole regularization path.

Finally, a critical issue is related to the choice of a suitable selection criterion that enables one to identify the variables that are more relevant to the specific regression problem. In fact, while in standard ℓ_1-based regularization the relevant variables correspond to the non zero entries of the regression coefficients, using our approach this is no longer valid. Nevertheless a natural way to select the relevant variables is to set a threshold on the *size* of the partial derivatives evaluated on the training set points, that is:

$$\sum_{i=1}^{n} |\partial_j f_\tau(x_i)|^2 = ||Z_j\alpha + L_j\beta||^2 \qquad \text{for } \ j = 1, \dots, p. \quad (5)$$

3 Derivation of the Iterative Projection Procedure

In this section we derive the procedure proposed in the previous section. We start rewriting problem (1) in a more convenient way, allowing for a useful characterization of the regularized solution f_τ.

3.1 RKHS and derivatives

A key observation is that partial derivatives have a particularly useful representation if we choose the hypotheses space to be a RKHS. In fact if the kernel is *sufficiently smooth* it is possible to prove (Zhou, 2008) that $z_{j,x} \in \mathcal{H}$ for all $x \in X$ where $z_{j,x}$ is defined in (4), and the following reproducing property for derivatives, that will be the main tool towards deriving Algorithm 1, holds

$$\partial_j f(x_0) = \langle f, z_{j,x_0} \rangle_{\mathcal{H}}.$$

In the following it will be useful to view partial derivatives and gradients as linear operators. Consider $\mathbb{R}^n$ with the standard inner product normalized by a factor $1/n$ denoted by $\langle \cdot, \cdot \rangle_n$ and the corresponding norm $\| \cdot \|_n$. We define $\hat{D}_j : \mathcal{H} \to \mathbb{R}^n$ as

$$\hat{D}_j f = \left(\left(\partial_j f\right)(x_1), \ldots, \left(\partial_j f\right)(x_n) \right),$$

and its adjoint $\hat{D}_j^* : \mathbb{R}^n \to \mathcal{H}$ as

$$\hat{D}_j^* v = \frac{1}{n} \sum_{i=1}^{n} v_i z_{j,x_i}$$

It is also useful to view the gradient of f as an operator. Let $\mathbb{R}^{np}$ be p times the Cartesian product of $\mathbb{R}^n$ so that if $v = (v^1, \ldots, v^p), w = (w^1, \ldots, w^p)$ belong to $\mathbb{R}^{np}$ then $\langle v, w \rangle_{\mathbb{R}^{np}} = \sum_{j=1}^{p} \left\langle v^j, w^j \right\rangle_n$. The *empirical gradient* $\hat{\nabla} : \mathcal{H} \to \mathbb{R}^{np}$ and its adjoint $\hat{\nabla}^* : \mathbb{R}^{np} \to \mathcal{H}$ are defined by

$$\hat{\nabla} f = \left(\hat{D}_j f \right)_{j=1}^{p},$$

and

$$\hat{\nabla}^* v = \sum_{j=1}^{p} \hat{D}_j^* v^j,$$

respectively.

Note that $\hat{\nabla} f$ can be viewed as a $p \times n$ matrix so that $\mathcal{R}_1(f) = \sum_{j=1}^{p} \|\hat{D}_j f\|_n$, can be seen as the norm $2, 1$ of the gradient, obtained summing up the Euclidean norms of each row.

3.2 Iterative Projected Gradient

The functional under study is not differentiable, and standard minimization techniques cannot be used. Actually, minimizing non differentiable objective functions is quite common in sparse approximation, therefore a number of authors have already proposed some suitable solutions. A popular technique employs proximity operators, a powerful generalization of the notion of projection operators. Following the mainstream proposed by Combettes and Wajs (2005), in a previous work – in which we focus on structured sparsity for linear variable selection – we proposed an optimization framework consisting of an iterative projection and based on subgradient calculation to solve an appropriate fixed point equation satisfied by f_τ, see Rosasco et al. (2009) for details. Indeed, both the theorem 1 below, which represents the theoretical basis for iterative algorithm 1, and the equation (11) generalize analogous results derived in that paper.

For convenience, let's introduce the sampling operator $S_n : \mathcal{H} \to \mathbb{R}^n$ and its adjoint $S_n{}^*$, defined by $S_n(f) = (f(x_1), \ldots, f(x_n))$ and $S_n{}^* v(x) = \sum_{i=1}^{n} v_i k(x_i, x)$ respectively. The kernel matrix is simply $K = S_n S_n{}^*$ (De Vito et al., 2005). Using the above notations, solving problem (1) amounts to finding

$$f_\tau = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \{ \underbrace{\|S_n f - \mathbf{y}\|_n^2}_{\mathcal{E}(f)} + 2\tau \underbrace{\sum_{j=1}^{p} \|\hat{D}_j f\|_n}_{\mathcal{R}_1(f)} \}. \quad (6)$$

Theorem 1. *Given $\sigma > 0$, and assuming $\mathcal{E} + 2\tau\mathcal{R}_1$ to be strictly convex, f_τ is the unique solution of the fixed point equation*

$$f = \left(I - P_{\frac{\tau}{\sigma}\mathcal{C}} \right) \left(f - (2\sigma)^{-1} S_n{}^* \left(S_n f - \mathbf{y} \right) \right) \quad (7)$$

where $P_{\mathcal{C}}$ is a suitable projection defined below.

The solution of the above problem can be calculated by the following iteration

$$f_{t+1} = \left(I - P_{\frac{\tau}{\sigma}\mathcal{C}} \right) \left(f_t - (2\sigma)^{-1} S_n{}^* \left(S_n f_t - \mathbf{y} \right) \right) \quad (8)$$

with $f^0 = 0$ (Combettes and Wajs, 2005). Under suitable assumptions, it is possible to prove that

$$\lim_{t \to \infty} \|f_t - f_\tau\|_{\mathcal{H}} = 0, \quad (9)$$

if σ is appropriately chosen. Following Proposition 1 in Rosasco et al. (2009) it turns out that the best a priori chosen step-size σ depends only on the smallest and biggest eigenvalues of the kernel matrix K.

Remark. Note that in order to have a unique solution, strict convexity of the functional is required.

Since in general this assumption is too restrictive, it is possible to avoid it by adding a further strictly convex regularization term to $\mathcal{R}_1$. We also remark that strong convergence holds also without requiring strict convexity – see Theorem 3.4 in Combettes and Wajs (2005).

The projection $P_{\mathcal{C}}$ is the core of the above algorithm, and is related to the subgradient of the regularizer $\mathcal{R}_1$ at 0. If we let

$$B^p_n = \{v = (v^1, \dots, v^p) \mid \|v^j\|_n \leq 1, \;\; j = 1, \dots, p\},$$

then we can use Proposition 2 in Rosasco et al. (2009) to show that

$$\mathcal{C} = \partial\mathcal{R}_1(0) = \big\{f \in \mathcal{H} \mid f = \hat\nabla^* v \text{ with } v \in B^p_n\big\}.$$

Moreover, it is possible to show that $\partial\mathcal{R}_1(0)$ is a closed convex subset of $\mathcal{H}$, and the projection $P_{\frac{\tau}{\sigma}\mathcal{C}}$ of a $f \in \mathcal{H}$ on $\frac{\tau}{\sigma}\mathcal{C}$, is given by

$$P_{\frac{\tau}{\sigma}\mathcal{C}} f = \frac{\tau}{\sigma}\hat\nabla^*\bar v, \qquad (10)$$

where

$$\bar v = \operatorname*{argmin}_{v \in B^p_n}\{\|f - \frac{\tau}{\sigma}\hat\nabla^* v\|^2_{\mathcal{H}}\}.$$

Finally, it can be shown that $\bar v$ can be calculated component-wise using the iteration:

$$v^j_{q+1} = \frac{v^j_q - \eta\hat D_j\left(\hat\nabla^* v_q - \frac{\sigma}{\tau}f\right)}{1 + \eta\|\hat D_j(\hat\nabla^* v_q - \frac{\sigma}{\tau}f)\|_n} \qquad (11)$$

with $v^j_0 = 0$. As long as $\eta = 1/\|\hat\nabla\hat\nabla^*\|$ it is possible to prove that

$$\lim_{q\to\infty}\|\frac{\tau}{\sigma}\hat\nabla^* v_q - P_{\frac{\tau}{\sigma}\mathcal{C}} f\|_{\mathcal{H}} = 0. \qquad (12)$$

3.3 Towards an Implementation

In order to implement the above algorithm we need to find a finite dimensional representation of f. Note that from (8) and (10) the minimizer of (6) satisfies $f_\tau \in \text{Range}({S_n}^*) + \text{Range}(\hat\nabla^*)$. Henceforth it satisfies the following form of the representer theorem

$$f_\tau = {S_n}^*\alpha + \hat\nabla^*\beta = \sum_{i=1}^n \frac{1}{n}\alpha_i k(x_i, \cdot) + \sum_{i=1}^n\sum_{j=1}^p \frac{1}{n}\beta^j_i z_{j,x_i} \qquad (13)$$

with $\alpha \in \mathbb{R}$ and $\beta = (\beta^1, \dots, \beta^p)$ with $\beta^j \in \mathbb{R}^n$, for $j = 1, \dots, p$. Note that Z_j, L_j defined in Section 2.1 are the matrices associated to the operators $S_n\hat D^*_j : \mathbb{R}^n \to \mathbb{R}^n$ and $\hat D_j\hat\nabla^* : \mathbb{R}^{np} \to \mathbb{R}^n$, respectively. Moreover we define Z and L as the matrices associated to the operators $S_n\hat\nabla^* : \mathbb{R}^{np} \to \mathbb{R}^n$ and $\hat\nabla\hat\nabla^* : \mathbb{R}^{np} \to \mathbb{R}^{np}$, respectively. Then we have the following result.

Proposition 1. *For $f_0 = 0$, the solution at step $t+1$ is given by*

$$f_{t+1} = {S_n}^*\alpha_{t+1} + \hat\nabla^*\beta_{t+1}$$

with

$$\alpha_{t+1} = \alpha_t - \frac{1}{2\sigma}\left(K\alpha_t + Z\beta_t - \mathbf{y}\right) \qquad (14)$$

and

$$\beta_{t+1} = \beta_t - \frac{\tau}{\sigma}\bar v_{t+1}, \qquad (15)$$

where $\bar v_{t+1}$ is such that

$$\frac{\tau}{\sigma}\hat\nabla^*\bar v_{t+1} = P_{\frac{\tau}{\sigma}\mathcal{C}}\left({S_n}^*\alpha_{t+1} + \hat\nabla^*\beta_t\right). \qquad (16)$$

Proof. We proceed by induction. The base case is clear. Then, by the inductive hypotheses we have that

$$f_t = {S_n}^*\alpha_t + \hat\nabla^*\beta_t.$$

Therefore, using (8) it follows that f_{t+1} can be expressed as:

$$\left(I - P_{\frac{\tau}{\sigma}\mathcal{C}}\right)\left({S_n}^*\left(\alpha_t - (2\sigma)^{-1}\left(K\alpha_t + Z\beta_t - \mathbf{y}\right)\right) + \hat\nabla^*\beta_t\right)$$

and the proposition is proved, letting α_{t+1}, β_{t+1} and $\bar v_{t+1}$ as in Equations (14), (15) and (16). □

The following results shows how to explicitly calculate the projection.

Proposition 2. *Let $\eta = 1/\|L\|$. If $f \in \mathcal{H}$ can be written as ${S_n}^*\alpha + \nabla^*\beta$, then the projection $P_{\frac{\tau}{\sigma}\mathcal{C}} f$ is given by $\frac{\tau}{\sigma}\hat\nabla^*\bar v$ where $\bar v$ can be calculated starting from $v_0 = 0$ and using the following iteration*

$$v^j_{q+1} = \frac{v^j_t - \eta\left(L_j\left(v_q - \frac{\sigma}{\tau}\beta\right) - \frac{\sigma}{\tau}Z_j\alpha\right)}{1 + \eta\|L_j\left(v_q - \frac{\sigma}{\tau}\beta\right) - \frac{\sigma}{\tau}Z_j\alpha\|_n}. \qquad (17)$$

Proof. In order to get (17) we can plug the representation (13) in (11) and use the fact that $\hat D_j {S_n}^* = S_n\hat D^*_j = Z_j$. □

In practical implementation, Algorithm 1 requires the definition of a suitable stopping rule. Equations (9) and (12) suggest the following choice

$$\|f_t - f_{t-1}\|_{\mathcal{H}} \leq \text{tol} \quad \text{and} \quad \|\hat\nabla^*(v_q - v_{q-1})\|_{\mathcal{H}} \leq \text{tol}.$$

Note that these quantities are computable since

$$\begin{aligned}\|f_t - f_{t-1}\|^2_{\mathcal{H}} &= \|{S_n}^*\delta\alpha + \hat\nabla^*\delta\beta\|^2_{\mathcal{H}}\\ &= \left\langle\delta\alpha, K\delta\alpha\right\rangle_n\\ &\quad +2\left\langle\delta\alpha, Z\delta\beta\right\rangle_n\\ &\quad +\left\langle\delta\beta, L\delta\beta\right\rangle_n\end{aligned}$$

where $\delta\alpha = \alpha_t - \alpha_{t-1}$ and $\delta\beta = \beta_t - \beta_{t-1}$, and similarly

$$\|\hat{\nabla}^*(v_q - v_{q-1})\|_{\mathcal{H}}^2 = \langle (v_q - v_{q-1}), L(v_q - v_{q-1}) \rangle_n .$$

4 Experimental Analysis

In this section we present some preliminary experiments conducted in order to assess empirically the effectiveness of the proposed regularization approach on both synthetic data sets and a standard benchmark comprising real data. The aims of the experiments are: (i) to verify that the proposed algorithm selects the correct subset of relevant variables when the underlying regression function is nonlinear; (ii) to assess the prediction performance of the estimator learned with our algorithm, and to compare it with a number of more standard alternatives for both sparse and non sparse regression.

We compare Algorithm 1 with: non-sparse kernel-based ridge regression with a Gaussian kernel (which will be denoted *G-ridge* hereinafter), and the sparse ℓ_1 regularization on a linear model both with respect to the input variables (denoted simply ℓ_1), and with respect to a nonlinear combination of the input variables (denoted ℓ_1^{feat}). Specifically, for ℓ_1^{feat} we consider functions which are the linear combination of the expansion of a 4^{th} degree polynomial defined over the input variables. Note that we do not compare experimentally our approach with the ones based on sparse nonlinear additive models proposed by Ravikumar et al. (2008). This is mainly due to the fact that the *additiveness* assumption required by such models is too restrictive and is not satisfied by the regression functions we are interested in (see Subsection 4.1).

As for the actual deployment of Algorithm 1, we considered two different settings. In the first we use the algorithm as it has been described so far, (this setting is denoted $Alg1$). Furthermore, we decided to rely on a fairly standard approach based on a double optimization – see Candès and Tao (2005), De Mol et al. (2009) for ℓ_1 and ℓ_1-ℓ_2 regularization – where Algorithm 1 is used for selecting the variables only, and subsequently a second optimization is performed using the *G-ridge* algorithm on the selected variables in order to provide accurate variable selection and good prediction performance at the same time. This second setting is referred to as $Alg1^{GR}$. As pointed out in Section 2, an important point of the algorithm is the choice of a threshold value for the quantity in (5) to discriminate among relevant and non relevant variables. In the following experiments we used as threshold the regularization parameter τ, which empirically proved to be a very effective solution. In order to guarantee a fair comparison among the different methods, we employ a common validation protocol for all considered algorithms, based on different data sets for training, validation and test, where the validation set is used for choosing the value of τ minimizing the error on such a set, and the test set for estimating the prediction accuracy. All the experiments presented below have been performed using a prototype Matlab implementation.

4.1 Synthetic Data

In this set of experiments, we generated a number of synthetic data sets as follows. Given the interval $[-3,3]^6 \subseteq \mathbb{R}^6$ we sampled $n = 80$ training points randomly drawn from the uniform distribution and two separate validation and test sets of size $m = 200$. According to our initial assumption, the output labels are computed using a noise-corrupted regression function f_ρ that depends nonlinearly from $\{x_1, x_2\}$ only, i.e $y = f_\rho(x_1, x_2) + w$, where x_i denotes the i^{th} component of the vector x. For each x, the w is a white noise, sampled from a normal distribution with zero mean and variance that is a small fraction of the average output of f_ρ over the input domain. Specifically, we focused on the following examples $\boldsymbol{f}_1 = (x_1^4 - x_1^2) \cdot (3 + x_2)$; $\boldsymbol{f}_2 = 2(x_1^3 - x_1) \cdot (2x_2 - 1) \cdot (x_2 + 1) + (x_2^3 - x_2 + 3)$; and $\boldsymbol{f}_3 = -2(2x_1^2 - 1) \cdot (x_2) \cdot e^{-x_1^2 - x_2^2}$.

In order to evaluate the stability of the results, for each example we run the five algorithms several times by keeping fixed the test set and letting the training examples vary.

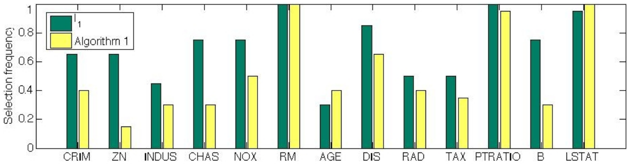
The ability to select correctly the relevant variables has been assessed by computing *true* (TP) and *false* (FP) positive rates. For lack of space we do no report a table summarizing the results of all the tests. However, the experimental evidence is that the $Alg1^{GR}$ algorithm has constantly a TP rate equal to 1, and the FP rate is less than 0.05. With $Alg1$ the FP rate increases since the optimization procedure leads to selecting a higher number of variables in order to keep a low prediction error. The selection performances of ℓ_1 are extremely poor, since the algorithms almost always selects all the variables. The ℓ_1^{feat} approach shows a high variance in the selection of the variables, and the TP rate is less than 0.8.

The prediction errors of the learned algorithms are reported in Table 1. From the results therein it emerges that: the ℓ_1 algorithm is not appropriate to predict the output values of the above functions. Furthermore, the performance of a linear selection of nonlinear features computed from the original input variables is viable as long as the true regression function is actually within the set spanned by the basis features (as for $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$), otherwise the performances degrade quite dramatically (as for $\boldsymbol{f}_3$). Overall, the best algorithmic approach to nonlinear feature selection is to

Table 1: *Average generalization errors associated to the considered algorithms.*

Learning Algorithm	*Regression Function* f_1	f_2	f_3
$Alg1$	0.030 ± 0.006	0.23 ± 0.06	0.071 ± 0.012
$Alg1^{GR}$	$\mathbf{0.008 \pm 0.003}$	$\mathbf{0.10 \pm 0.04}$	$\mathbf{0.003 \pm 0.002}$
ℓ_1	0.096 ± 0.003	3.32 ± 0.07	0.103 ± 0.003
ℓ_1^{feat}	0.026 ± 0.007	0.17 ± 0.04	0.079 ± 0.003
G-ridge	0.030 ± 0.003	0.21 ± 0.03	0.015 ± 0.003

Figure 1: *Results obtained with the Boston Housing data set. We report the selection frequency of each variable.*



run the Algorithm 1 for selecting the relevant features and to perform a second optimization for estimating the regression function. This leads clearly to better results then those obtained by *G-ridge* on all the input variables. Finally, in most cases, the prediction performance of the Algorithm 1 alone is competitive with the other methods.

4.2 Real Data

In this experiment, we use the standard Boston Housing data, i.e. the housing data for 506 census tracts of Boston, available from the UCI Machine Learning Database Repository: `http://archive.ics.uci.edu/ml/`. Each census tract represents a data-point, described by 13 features, whereas the output is the housing price. In our experiment, we randomly partition the data into 50 training, 228 validation and 228 test points. For comparison we use the four algorithms: ℓ_1, *G-ridge*, $Alg1$ and $Alg1^{GR}$. We perform the experiments 20 times. For each repetition the data points are centered and normalized with respect to the training set points. In Figure 1 we report a table with the average squared test errors of all the tested algorithms and the selection frequencies for the $Alg1^{GR}$ and ℓ_1.

Table 2: *Results obtained with the Boston Housing data set. We report a comparison of the average squared test errors relative to the different algorithms.*

Learning Algorithm	ℓ_1	*G-ridge*	$Alg1$	$Alg1^{GR}$
Test Error	30 ± 4	22 ± 5	23 ± 5	$\mathbf{21 \pm 3}$

From Figure 1 we can observe that the most frequently selected features are the same in both algorithms, that is average number of rooms per dwelling (RM), pupil-teacher ratio by town (PTRATIO), and % lower status of the population (LSTAT). Nevertheless we can notice that $Alg1^{GR}$ presents a larger gap between the features that are most frequently selected and the other features, indicating that selection performed via $Alg1^{GR}$ is more stable than via ℓ_1-regularization. A number of previous studies have analyzed this data set. In particular Zhang (2009) applies different subset selection techniques and compares their prediction accuracy. Indeed, the prediction errors provided by these techniques are comparable with our results for ℓ_1 regularization and are higher than those achieved by kernel methods, with and without subset selection. This behavior may suggest that, on this data set, nonlinearity is stronger than sparsity. In fact, *G-ridge*, which does not perform subset selection, provides better estimate than ℓ_1 regularization which trades off nonlinearity for sparsity. Only $Alg1^{GR}$, which combines both approaches, is able to perform a more stable subset selection though maintaining, and even improving, the prediction accuracy of the solution.

5 Conclusions

We have proposed a new regularization scheme for nonlinear variable selection, promoting estimators which depend on a small number of the original variables.

To the best of our knowledge this is the first *direct* approach to nonlinear variable selection, which on one hand does not require the use of nonlinear features, and on the other hand directly promotes sparsity in the original variables. We described an iterative procedure to solve the underlying variational problem, which convergence to the optimal solution is proved.

The initial experimental results on synthetic and real data are promising. Indeed, they show that our method outperforms linear subset selection techniques as well as nonlinear regression methods, both in terms of prediction and variable selection, since it performs selection as the former, though maintaining nonlinearity as the latter.

Acknowledgements

This paper describes a joint research work done at the Center for Biological and Computational Learning (within the McGovern Institute for Brain Research at MIT), at the Department of Brain and Cognitive Sciences (affiliated with the Computer Sciences and Artificial Intelligence Laboratory), and at the Departments of Computer Science and Mathematics of the University of Genoa. The authors have been partially supported by the Integrated Project Health-e-Child IST-2004-027749 and by grants from DARPA (IPTO and DSO), National Science Foundation (NSF-0640097, NSF-0827427), and Compagnia di San Paolo, Torino. Additional support was provided by: Adobe, Honda Research Institute USA, King Abdullah University Science and Technology grant to B. DeVore, NEC, Sony and especially by the Eugene McDermott Foundation.

References

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, Sep 2008.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74 (8):1289–1308, 2008.

E. J. Candès and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35:2313–2351, 2005.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200 (electronic), 2005.

I. Daubechies, G. Teschke, and L. Vese. Iteratively solving linear inverse problems under general convex constraints. *Inverse Problems and Imaging*, 1(1):29–46, 2007.

C. De Mol, Mosci, M. S. Traskine, and A. Verri. A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16(5):677–690, May 2009.

E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, and F. Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6:883–904, May 2005.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. Technical report, IEEE Journal of Selected Topics in Signal Processing, 2007.

T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

I. Guyon, H. Bitter, Z. Ahmed, M. Brown, and J. Heller. *Multivariate Non-Linear Feature Selection with Kernel Methods*, pages 313–326. Springer Berlin / Heidelberg, 2006.

E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIOPT*, 19(3):1107–1130, 2008.

T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff. Embedded methods. *Foundations and Applications, Studies in Fuzziness and Soft Computing*, 207:137–165, 2006.

B. Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 34(5):2272–2297, 2006.

P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

S. Mukherjee and D. Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7:319–349, 2006.

P. Ravikumar, H. Liu, J. Lafferty, and L. Wasserman. Spam: Sparse additive models. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*. 2008.

L. Rosasco, S. Mosci, M. Santoro, A. Verri, and S. Villa. Iterative projection methods for structured sparsity regularization. Technical Report MIT-CSAIL-TR-2009-50 / CBCL-282, Massachusetts Institute of Technology, Cambridge, MA, 2009.

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

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 56:267–288, 1996.

Q. Wu, F. Liang, and S. Mukherjee. Consistency of regularized sliced inverse regression for kernel models. *Technical report. Duke University and University of Illinois Urbana-Champaign*, 2008.

Tong Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928. 2009.

Ding-Xuan Zhou. Derivative reproducing properties for kernel methods in learning theory. *J. Comput. Appl. Math.*, 220:456–463, 2008.