MIT-CSAIL-TR-2009-050
CBCL-282

October 14, 2009

Iterative Projection Methods for Structured Sparsity Regularization

Lorenzo Rosasco, Sofia Mosci, Matteo Santoro, Alessandro Verri, and Silvia Villa

Iterative Projection Methods for Structured Sparsity Regularization

Lorenzo Rosasco[⋆], Sofia Mosci[†], Matteo Santoro[†], Alessandro Verri[†], Silvia Villa[‡]

⋆ *CBCL, McGovern Institute, Artificial Intelligence Lab, BCS, MIT*

† *DISI, Università di Genova*

‡ *DISI - DIMA, Università di Genova*

lrosasco@mit.edu, {mosci,santoro,verri}@disi.unige.it, villa@dima.unige.it

October 14, 2009

Abstract

In this paper we propose a general framework to characterize and solve the optimization problems underlying a large class of sparsity based regularization algorithms. More precisely, we study the minimization of learning functionals that are sums of a differentiable data term and a convex non differentiable penalty. These latter penalties have recently become popular in machine learning since they allow to enforce various kinds of sparsity properties in the solution. Leveraging on the theory of Fenchel duality and subdifferential calculus, we derive explicit optimality conditions for the regularized solution and propose a general iterative projection algorithm whose convergence to the optimal solution can be proved. The generality of the framework is illustrated, considering several examples of regularization schemes, including ℓ_1 regularization (and several variants), multiple kernel learning and multi-task learning. Finally, some features of the proposed framework are empirically studied.

1 Introduction

In this paper we use convex analysis tools to propose a general framework for solving convex non differentiable minimization problems underlying many regularized learning algorithms.

In learning from examples one tries to infer some quantity of interest from a training set which is randomly sampled and corrupted by noise. Learning schemes which are simply tailored to minimize a data fit objective term, often lead to unstable solutions that fail to generalize to new data. An effective way to restore stability and find meaningful solutions is to resort to regularization techniques [39, 21, 37, 11]. This class of methods typically involves the minimization of an objective function which is the sum of two terms: the first one is a data fit term, whereas the second one is a penalty that favors "simple" models. Approaches based on Tikhonov regularization, including Support Vector Machines or Regularized Least Squares, are probably the most popular examples in this class of methods and are based on convex differentiable penalties. Recently, methods such as the *lasso* [38] – based on ℓ_1 regularization – received considerable attention for their property of providing *sparse* solutions. Sparsity has become a popular way to deal with small samples of high dimensional data and, in a broad sense, refers to the possibility of writing the solution in terms of a *few* building blocks. The success of ℓ_1 regularization motivated exploring different kinds of sparsity properties for linear models as well as kernel methods [45, 43, 44, 3, 35, 1, 28, 36].

A common feature of the latter class of algorithms is that they are based on convex non differentiable penalties, which are often suitable sums of euclidean (or Hilbertian) norms. In this paper we refer to this general class of methods as *structured sparsity regularization* algorithms and we study the problem of computing the regularized solution. The presence of a non differentiable penalty makes the solution of the minimization problem non trivial and recently there has been a considerable amount of work devoted to this problem, largely focused on, and motivated by, ℓ_1 regularization. In this context, an extensive list of references and an overview of many approaches can be found in [42]. Among the proposed optimization schemes it is worth mentioning, for example, interior point methods [27], coordinate descent [41], iterative soft-thresholding [12] and homotopy methods [17]. In particular the LARS algorithm [19] which is popular in machine learning belongs to the latter class of methods. As we mentioned above, besides ℓ_1 regularization, in machine learning several techniques have been proposed based on non differentiable penalties. Interestingly, for given algorithms, *ad hoc* optimization procedures have been proposed, including, in some cases, greedy procedures with no convergence guarantees.

In this work we recognize a common structure among many different regularization algorithms and discuss the application of a general optimization strategy to solve the corresponding variational problem. Indeed, a large class of algorithms corresponds to minimize a functional which is the sum of a differential data term and a convex penalty which is one homogeneous (for example a sum of suitable norms). This observation allows to propose a unifying framework and derive a powerful iterative procedure to compute the regularized solution. Using the theory of Fenchel duality we decouple the contributions due to the data fit term, and the penalty term: at each iteration the gradient of the data term is projected on a set which is defined by the considered penalty. The explicit form of the projection can often be written in closed form and iteratively computed, otherwise. The obtained procedure is typically easy to implement and its convergence to the optimal solution is proved when the functional is strictly convex. Proving convergence (and convergence rates) when the functional is simply convex is a more challenging problem. On the other hand the assumptions we need to ensure convergence can always be enforced by considering a suitable perturbation of the original functional obtained adding a strictly convex term. As we discuss in details, interestingly, such a term induces a preconditioning of the problem and can be shown to often reduce substantially the number of required computations without affecting the sparsity and prediction properties of the obtained solution. This is a crucial point that we discuss both theoretically and experimentally.

Our work can be seen as the application to a large class of learning algorithms of an approach that has recently received a lot of attention in the context of signal processing and inverse problems [6, 10, 12, 23, 13, 24, 42, 22, 32, 40]. In particular, the procedure we consider to compute the projection can be seen as a generalization of the algorithm proposed in [8] to solve total variation regularization. From a mathematical point of view there exist very general and abstract results on these kind of algorithms and among them we mention forward-backward splitting methods [31, 18, 9], iterative projection algorithms [4, 5]. To the best of our knowledge this is the first attempt to apply this class of iterative projection methods to a large class of learning schemes including multiple task and multiple kernel learning. The mathematical context we consider trade-offs simplicity and generality and allows to give simplified proofs in a unifying framework.

The paper is organized as follows. In Section 2, we begin by setting the notation and recalling some basic mathematical properties necessary to introduce the iterative algorithm and state its main properties. We state all the mathematical and algorithmic results first, and postpone the proofs to the Appendix A. In Section 4, in order to show the wide applicability of our work, we apply the results to several learning schemes. In Section 5 we describe some experimental results. Finally, Section 6 concludes the paper and contains a brief discussion of future work.

2 Iterative Projection Algorithm

Here, after describing the general class of regularized learning algorithms under study, we proceed discussing the iterative procedure to compute the regularized solution and provide a detailed analysis. The latter consists in three main steps. First we show that the regularized solution satisfies a suitable fixed point equation involving a projection on a convex set, so that we can consider the iteration corresponding to the associated successive approximation scheme. Second, we use the fixed point-theorem to prove convergence of the proposed procedure. Finally, we show how to compute the projection by generalizing previous results for total variation regularization.

2.1 Setting

Given a a Hilbert space $\mathcal{H}$ and a fixed positive number τ, we consider the problem of computing:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathcal{E}_\tau(f) = \operatorname*{argmin}_{f \in \mathcal{H}} \{F(f) + 2\tau J(f)\}, \tag{1}$$

where $F : \mathcal{H} \to \mathbb{R}$, $J : \to \mathbb{R} \cup \{+\infty\}$ can be interpreted as the data and penalty terms, respectively. In the following, F is assumed to be differentiable and strictly convex, while J is required to be lower semicontinuous, convex, coercive (see Ch. 1 and Ch.2 of [20]) and one-homogeneous,

$$J(\lambda f) = \lambda J(f),$$

for all $f \in \mathcal{H}$ and $\lambda \in \mathbb{R}^+$. Note that our analysis still holds if one assume the coerciveness of $F + 2\tau J$, and not specifically of J. Before presenting our results we discuss several examples for F and J.

Loss term. In the supervised learning, given a training set $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R} \times Y$, with $Y = [-M, M]$, $M > 0$, the most common choice for the data term F is the empirical risk associated to some cost function $\ell : \mathbb{R} \times Y \to \mathbb{R}^+$, i.e.

$$F(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i). \tag{2}$$

Examples of loss functions generating convex and differentiable functionals F via (2) are the exponential loss $\ell(f(x), y) = e^{-yf(x)}$, the logistic loss $\log(1 + e^{-yf(x)})$, and the square loss $(y - f(x))^2$. In general, the corresponding empirical risk will be only convex, and strict convexity can be ensured under further assumptions on the data. An alternative way to enforce strict convexity is to add the strictly convex term $\mu \|f\|_{\mathcal{H}}^2$ for some small positive parameter μ. As we discuss in the following, this can be seen as a preconditioning of the problem, and, if μ is small enough, one can see that the solution does not change (see Section 2.3 for a more detailed discussion of this point). Another possible expression for the data term is

$$F(f) = \|Af - y\|_{\mathcal{Y}}^2, \tag{3}$$

where $A : \mathcal{H} \to \mathcal{Y}$ is a bounded linear operator between Hilbert spaces $\mathcal{H}$, $\mathcal{Y}$, that might depend on the data, and $y \in \mathcal{Y}$ is a measurement function from which we aim at reconstructing f. In practical situations $\mathcal{H}$ and $\mathcal{Y}$ are typically finite dimensional euclidean spaces and A is a matrix. This latter choice is general enough to deal more general setting such as multi-task learning.

Penalty term. The assumptions on the penalty – lower semicontinuity, coerciveness, convexity and one-homogeneity – are satisfied by a general class of penalties that are sum of norms in distinct Hilbert spaces $(\mathcal{G}_k, \|\cdot\|_k)$:

$$J(f) = \sum_{k=1}^{p} ||\mathcal{J}_k(f)||_k, \tag{4}$$

where, for all k, $\mathcal{J}_k : \mathcal{H} \to \mathcal{G}_k$ is a bounded linear operator [1]. This is the class of penalties we consider. For example, if the estimator is assumed to be described by a generalized linear model $f(x) = \sum_{j=1}^{p} \psi_j(x)\beta_j$, the ℓ_1 norm of the coefficients $J(\beta) = \sum_{j=1}^{p} |\beta_j|$ is a special case of the above penalty. If the coefficients are divided into "blocks", a penalty of the form (4), has been proposed in the so called *group lasso* and *composite absolute penalties* algorithms. Similar penalties have been used for multiple task learning and sparse principal component analysis. In particular another example is multiple kernel learning where the estimator is assumed to be $f = f_1 + \cdots + f_p$ and every f_j belongs to a specific RKHS $\mathcal{H}_j$ with kernel K_j and norm $\|\cdot\|_j$. In this case, the penalty term takes the form $\sum_{j=1}^{p} \|f_j\|_j$.

We remark that the examples above are just a few examples of learning methods to which the proposed approach can be applied. In the next section we show how the corresponding optimization problems can be solved using the same simple procedure.

2.2 Algorithm

In this section we describe the iterative procedure for computing the solution f^* of the convex minimization problem (1).

Towards this end we recall some basic facts in convex analysis and introduce some definitions (see [20]). If $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ is a Hilbert space, the subdifferential at $f \in \mathcal{H}$ of a convex functional $Q : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ is denoted with $\partial Q(f)$ and is defined as the set

$$\partial Q(f) := \{h \in \mathcal{H} \,:\, Q(g) - Q(f) \geq \langle h, g - f \rangle_{\mathcal{H}}, \quad \forall g \in \mathcal{H}\}.$$

If Q is not only convex but also differentiable, then the subdifferential reduces to a unique element which is precisely the gradient $\nabla Q(f)$ of Q at f. Given the above definition we let

$$K := \partial J(0),$$

and denote with $\pi_{\lambda K} : \mathcal{H} \to \mathcal{H}$ the projection on $\lambda K \subset \mathcal{H}$, $\lambda \in \mathbb{R}^+$ (which is well defined since the subdifferential is always a convex, closed set, and it is nonempty because $J(0) = 0$).

Given the above definitions, the optimization scheme we derive is given by Algorithm 1. The parameter σ can be seen as a step-size, which choice is crucial to ensure convergence and is discussed in the following. As we mentioned before, our approach decouples the contributions of the two functionals J and F. At each iteration, the projection $\pi_{\lambda K}$– which is entirely characterized by J – is applied to a term that depends only on F. Fenchel duality [20] is the key tool that, combined with one-homogeneity, allows us to characterize the contribution of J. In the following we state and prove the key results toward deriving Algorithm 1.

Algorithm 1 General Algorithm

```
Require: $\sigma, \tau > 0$
Initialize: $f^0 = 0$
while convergence not reached do
  $p := p + 1$
      $f^p = \left(I - \pi_{\frac{\tau}{\sigma}K}\right)\left(f^{p-1} - \frac{1}{2\sigma}\nabla F(f^{p-1})\right)$      (5)
end while
return $f^p$
```

[1] We also need the technical assumption $\cap_k \mathrm{kernel}(\mathcal{J}_k) = \{0\}$ to ensure coerciveness of J

2.3 Fixed Point Equation

We start showing that the optimal solution of problem (1) is the unique fixed point of a family of functionals parameterized by the step-size σ.

Theorem 1 *Given $\tau > 0$, $F : \mathcal{H} \to \mathbb{R}$ strictly convex and differentiable and $J : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ lower semicontinuous, coercive, convex and one-homogeneous, the minimizer f^* of $\mathcal{E}_\tau$ is the unique fixed point of the map $\mathcal{T}_\sigma : \mathcal{H} \to \mathcal{H}$ defined by*

$$\mathcal{T}_\sigma(f) = \left(I - \pi_{\frac{\tau}{\sigma}K}\right)\left(f - \frac{1}{2\sigma}\nabla F(f)\right).$$

We postpone the proof to Appendix A, but it is worth remarking that strict convexity of F is assumed only to ensure uniqueness of the minimizer of $\mathcal{E}_\tau$, and that the fixed point equation is indeed satisfied by each minimizer of $\mathcal{E}_\tau$ in the case F is only convex.

We note that Algorithm 1 is simply the successive approximation scheme associated to the above fixed point equation. If the map $\mathcal{T}_\sigma$ is a contraction convergence of the iteration is ensured by Banach fixed point theorem and convergence rates can be easily obtained. Recall that we say that a map $\mathcal{T}_\sigma$ is a contraction if

$$|\mathcal{T}_\sigma(f) - \mathcal{T}_\sigma(g)| \leq L_\sigma \|f - g\|, \qquad \forall f, g \in \mathcal{H}$$

and $L_\sigma < 1$. In fact, in our setting $\mathcal{T}_\sigma$ depends on σ, and we can choose the latter so that $L_\sigma < 1$. In this case the following inequality relates the solution f^p at $p - th$ iteration step and the solution f^* of the minimization problem,

$$\|f^* - f^p\| \leq \frac{L_\sigma^p}{1 - L_\sigma}\|f^1 - f^0\|.$$

The constant L_σ depends only on the data fit term as can be seen by the following result.

Proposition 1 *Assume the penalty term to satisfy the assumptions in Theorem 1 and F to be twice differentiable with continuous second derivative $\nabla^2 F : \mathcal{H} \to \mathcal{L}(\mathcal{H}, \mathcal{H})$. Moreover let $a(f) \geq b(f)$ denote the largest and smallest eigenvalues of $\nabla^2 F(f)$ and assume that there exist $a \geq b > 0$ such that $a \geq a(f) \geq b(f) \geq b$ for all $f \in \mathcal{H}$. Then the map $\mathcal{T}_\sigma$ is a contraction if we choose σ such that*

$$\max\left\{\left|1 - \frac{a}{2\sigma}\right|, \left|1 - \frac{b}{2\sigma}\right|\right\} < 1. \tag{6}$$

The optimal a priori choice for the step-size is given by

$$\sigma = \frac{a + b}{4}$$

and in this case we can choose $L_\sigma = \frac{a-b}{a+b}$.

Again, we postpone the proof to Appendix A and explicitly compute L_σ in several cases in Section 3.1. In the above theorem $\nabla^2 F : \mathcal{H} \to \mathcal{L}(\mathcal{H}, \mathcal{H})$ denotes the second derivative of F. To write it in this form, with an abuse of notation, we implicitly identified the linear operator $\nabla F(f) \in \mathcal{L}(\mathcal{H}, \mathbb{R})$ with an element $\nabla F(f) \in \mathcal{H}$, and then we computed the second derivative (see [30] for more details).

The above result shows that in general, for a strictly convex F, if the smallest eigenvalue of the second derivative is not uniformly bounded from below by a strictly positive constant, it might not be possible to choose σ so that $L_\sigma < 1$. The next corollary shows that this can always be done if F is perturbed by adding the term $\mu \|\cdot\|_{\mathcal{H}}^2$, with $\mu > 0$.

Corollary 1 *Assume the penalty term to satisfy the assumptions in Theorem 1 and F to be convex and twice differentiable with continuous second derivative $\nabla^2 F$. Moreover let $a(f) \geq b(f) \geq 0$ denote the largest and smallest eigenvalues of $\nabla^2 F(f)$ and suppose that $a(f) \leq a$. Consider the perturbed function $F_\mu = F + \mu \|\cdot\|^2_{\mathcal{H}}$, with $\mu > 0$ and set $b = \inf_{f\in\mathcal{H}} b(f)$. Then the map $\mathcal{T}_\sigma$ induced by F_μ is a contraction if we choose σ such that*

$$\max\left\{\left|1 - \frac{\mu}{\sigma} - \frac{a}{2\sigma}\right|, \left|1 - \frac{\mu}{\sigma} - \frac{b}{2\sigma}\right|\right\} < 1.$$

The optimal a priori choice for the step-size is given by

$$\sigma = \frac{a+b}{4} + \mu,$$

and in this case we can choose $L_\sigma = \frac{a-b}{a+b+4\mu}$.

The above corollary highlights the role of the μ-term, $\mu \|\cdot\|^2_{\mathcal{H}}$, as a natural preconditioning of the algorithm. One can also argue that, if μ is chosen small enough, the solution is expected not to change and in fact converges to a precise minimizer of $F + \tau J$. Indeed, the quadratic term performs a further regularization that allows to select, as μ approaches 0, the minimizer of $F + \tau J$ having minimal norm (see for instance [16]). Another possibility to drop the strong convexity assumption and select a specific minimizer of $F + \tau J$ is to consider a sequence $\lambda_p \to 1$, and slightly change Algorithm 1 multiplying the $p - th$ iteration by λ_p. Using the results in [4] it is possible to get strong convergence of the modified iterative sequence to a chosen minimizer. Moreover we expect that the refined results about convergence rate obtained for ℓ^1 regularization in [25] could be extended to Algorithm 1, without requiring strict convexity, if J is assumed to be as in (4). Other improvements of the proposed procedure are worth to be investigated: among them we mention the possibility of allowing errors in the evaluation of the projection operator and of ∇F, and obviously the study of iteration-dependent parameters choice (see [10, 24] for results in this direction in the case of ℓ^1 regularization).

In the next section we discuss how to compute the projection π_K.

2.4 Computing the Projection

We discuss how to compute the projection π_K when J is of the form

$$J(f) = \sum_{k=1}^{p} ||\mathcal{J}_k(f)||_k, \tag{7}$$

where, for all $k = 1, \ldots, p$, $\mathcal{G}_k$ is a Hilbert space with norm $\|\cdot\|_k$ and $\mathcal{J}_k : \mathcal{H} \to \mathcal{G}_k$ is a bounded linear operator.

In the following proposition we characterize the set $\partial J(0)$ and give a useful representation of the projection on this set.

Proposition 2 *Let $J(f)$ as in (7) and*

- *$\mathcal{G} = \prod_{k=1}^{p} \mathcal{G}_k$, so that $v = (v_1, \ldots, v_p) \in \mathcal{G}$ with $v_k \in \mathcal{G}_k$ and $\|v\| = \sum \|v_k\|_k$;*
- *$\mathcal{J} : \mathcal{H} \to \mathcal{G}$ such that $\mathcal{J}(f) = (\mathcal{J}_1(f), \ldots, \mathcal{J}_p(f))$ and $\mathrm{Ker}\mathcal{J} = \{0\}$.*

Then

$$\partial J(0) = \{\mathcal{J}^T v : v \in \mathcal{G}, \|v_k\|_k \leq 1 \ \forall k\},$$

where $\mathcal{J}^T : \mathcal{G} \to \mathcal{H}$ *is the adjoint of* $\mathcal{J}$*, and can be written as* $\mathcal{J}^T v = \sum_{k=1}^p \mathcal{J}_k^T v_k$*.*
Moreover the projection of an element $g \in \mathcal{H}$ *on the set* $\lambda K := \lambda \partial J(0)$ *is given by* $\lambda \mathcal{J}^T \bar{v}$*, where*

$$\bar{v} \in \operatorname*{argmin}_{v \in \mathcal{G}, \ \|v_k\|_k \leq 1} \left\|\lambda \mathcal{J}^T v - g\right\|_{\mathcal{H}}^2. \tag{8}$$

We refer to Appendix A for the proof of the above result. Note that even though from the definition $\bar{v}$ may not be unique, if $\mathcal{J}$ has non trivial null space, the definition of the projection $\pi_{\lambda K}(g)$ is always unique.
As we will discuss in the following, in several specific cases the nonlinear projection π_K can be written in a closed form. Nonetheless, in general its computation is not straightforward. An efficient solution to an analogue problem has been recently proposed in the context of total variation image denoising [8]. We generalize this latter approach to derive an iterative scheme for computing the solution of problem (8) induced by penalties J of the form (7). Towards this end, we note that the Karush-Kuhn-Tucker conditions associated to (8) ensure the existence of a set of Lagrange multipliers α_k, such that for all k

$$\mathcal{J}_k(\lambda \mathcal{J}^T v - g) + \alpha_k v_k = 0,$$

with either $\|v_k\|_k = 1$ and $\alpha_k > 0$, or $\|v_k\|_k < 1$ and $\alpha_k = 0$. In both cases v_k satisfies

$$\mathcal{J}_k(\lambda \mathcal{J}^T v - g) + \left\|\mathcal{J}_k(\lambda \mathcal{J}^T v - g)\right\|_k v_k = 0 \quad \forall k. \tag{9}$$

The above equation leads to a fixed point equation which solution can be computed by means of the iteration given in the theorem below.

Theorem 2 *Given* J *as in Proposition 2, let* $\eta \leq (\left\|\mathcal{J}\mathcal{J}^T\right\|)^{-1}$*,* $v^0 = 0$ *and for any* $q \geq 0$*, set*

$$v_k^{q+1} = \frac{v_k^q - \eta \mathcal{J}_k\left(\mathcal{J}^T v^q - g/\lambda\right)}{1 + \eta \left\|\mathcal{J}_k\left(\mathcal{J}^T v^q - g/\lambda\right)\right\|_k}. \tag{10}$$

Then $\left\|\lambda \mathcal{J}^T v^q - \pi_{\lambda K}(g)\right\|_{\mathcal{H}}$ *converges to 0 as* $q \to \infty$*.*

Again, the proof is given in Appendix A and the explicit form of the projection for several different examples is discussed in Section 4. We remark that the convergence in the above result refers to the projection rather than to the possibly not unique function $\bar{v}$. Before dealing with examples, we discuss convergence and step-size choice for Algorithm 1 in some more specific, but still general, situations.

3 Some Relevant Algorithmic Issues

In this section we further discuss some issues related to the application of the general framework described above. First, we instantiate the discussion in the previous section describing how to choose the step size in several cases of interest. Second, we recall some data-driven step-size choices which are often shown empirically to lead to convergence speed up. Finally, we discuss a useful continuation strategy that can be used when solutions corresponding to various regularization parameters have to be computed.

3.1 Computing the a-priori step-size

We discuss the a-priori step-size choice given in Proposition 1, in two specific setting of interests. First, we consider supervised learning problems where, given a training set $\{(x_i, y_i)\}_{i=1}^n$, with $x \in X \subset \mathbb{R}^d$ and $y \in Y = [-M, M]$, we have to find an unknown functional relation $f : X \to Y$. We consider loss functions $\ell : \mathbb{R} \times Y \to \mathbb{R}^+$ that are convex and twice differentiable in the first argument. Moreover we consider functions belonging to a RKHS [2]. In particular we make use of the following well known facts. A function f in a RKHS $\mathcal{H}$ with kernel K, can be seen as a hyperplane $f(x) = \langle \Phi(x), \beta \rangle_{\mathcal{F}}$, where $(\mathcal{F}, \langle \cdot, \cdot \rangle_{\mathcal{F}})$ is a Hilbert space - *the feature space* - $\beta \in \mathcal{F}$ and $\Phi : X \to \mathcal{F}$ is called *feature map* [37] if

$$\langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}} = K(x, x').$$

In particular we make use of the following properties, $\forall f \in \mathcal{H}$, $\|f\|_{\mathcal{H}} \le \|\beta\|_{\mathcal{F}}$ and

$$\sup_{x \in X} |f(x)| \le \kappa \|\beta\|_{\mathcal{F}},$$

where, for the latter inequality to hold true, we need to assume that $\sup_{x \in X} \|\Phi(x)\|_{\mathcal{F}} \le \kappa$, (the kernel is bounded). In the following we consider in particular two examples of feature maps. The first is given by the reproducing kernel K by setting $\Phi(x) = K(x, \cdot)$ so that $(\mathcal{F}, \langle \cdot, \cdot \rangle_{\mathcal{F}})$ is simply $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ and $f = \beta$, implying $\|f\|_{\mathcal{H}} = \|\beta\|_{\mathcal{F}}$. The second example corresponds to considering a finite set of functions (a dictionary) $(\psi_j)_{j=1}^p$ and setting $\Phi(x) = (\psi_1(x), \ldots, \psi_p(x))$ so that $\mathcal{F}$ can be identified with $\mathbb{R}^p$ with the corresponding inner product. In this case $\|f\|_{\mathcal{H}} \le \|\beta\|_{\mathcal{F}}$ where the equality holds if the dictionary is an orthonormal basis.

Given the above premises, the specific data terms F we consider can be written as

$$F(\beta) = \sum_{i=1}^{n} \ell(\langle \Phi(x_i), \beta \rangle_{\mathcal{F}}, y_i) + \mu \|\beta\|_{\mathcal{F}}^2. \tag{11}$$

where $\mu \ge 0$. [2]

The following result studies the property of the map $\mathcal{T}_\sigma$ induced by the above functional, and in particular provides the optimal choice for the step-size σ using Proposition 1. We show that the optimal σ is determined by the loss function and the covariance operator defined by

$$\begin{array}{rccl} \mathrm{Cov}: & \mathcal{F} & \to & \mathcal{F} \\ & \beta & \mapsto & \sum_{i=1}^n \langle \Phi(x_i), \beta \rangle_{\mathcal{F}} \Phi(x_i). \end{array}$$

It is well-known that Cov is selfadjoint, so that if a and b are respectively the largest and the smallest eigenvalues of Cov, then it follows $a \ge b \ge 0$ [26].

Proposition 3 *Assume the penalty term to satisfy the assumptions in Theorem 1 and F to be given by (11). Moreover let a and b denote the largest and smallest eigenvalues of* Cov *and $0 \le L_{min} \le \ell''(w, y) \le L_{max}$, $\forall w \in \mathbb{R}, y \in Y$, where ℓ'' denotes the second derivative of ℓ with respect to w. Then the map $\mathcal{T}_\sigma$ is a contraction with constant L_σ, if we choose σ such that*

$$\max\left\{|1 - \frac{\mu}{\sigma} - \frac{L_{max} a}{2\sigma}|, |1 - \frac{\mu}{\sigma} - \frac{L_{min} b}{2\sigma}|\right\} < 1. \tag{12}$$

[2] Clearly if we choose $\Phi(x) = K(x, \cdot)$ we have

$$F(\beta) = F(f) = \sum_{i=1}^{n} \ell(f(x_i), y_i) + \mu \|f\|_{\mathcal{H}}^2.$$

The optimal a priori choice for the step-size is given by

$$\sigma = \frac{aL_{max} + bL_{min}}{4} + \mu,$$

and in this case we can choose $L_\sigma = \frac{aL_{max} - bL_{min}}{aL_{max} + bL_{min} + 4\mu}$.

We give the proof of the above result in Appendix A. Note again that if we let μ be equal to zero, then equation (12) may be never satisfied when either $L_{\min}$ or b are zero. We add examples for specific loss functions.

Example 1 (Square Loss) *Consider the square loss* $\ell(w, y) = (w - y)^2$. *Then* $\ell''(w, y) = L_{min} = L_{max} = 2\ \forall w \in \mathbb{R}, y \in Y$ *and the optimal a priori choice for the step-size is given by* $\sigma = \frac{a+b+2\mu}{2}$.

Example 2 (Exponential Loss) *If we consider the exponential loss* $\ell(w, y) = e^{-wy}$, *then* $\ell''(w, y) = y^2 e^{-wy}$. *Since* $Y = [-M, M]$ *we can assume without loss of generality that* $f(x) \in [-M, M] \quad \forall x$, *so that* $0 \leq \ell''(w, y) \leq M^2 e^{M^2}$. *The optimal a priori choice for the step-size is then given by* $\sigma = \frac{aM^2 e^{M^2}}{4} + \mu$.

Next we consider a data term of the form (3). More precisely, given two Hilbert spaces $\mathcal{H}, \mathcal{Y}$, and a bounded operator $A : \mathcal{H} \to \mathcal{Y}$, we consider

$$F = \|Af - y\|_{\mathcal{Y}}^2 + \mu \|f\|_{\mathcal{H}}^2 \tag{13}$$

which is strictly convex if $\mu > 0$ *or* A is injective. In particular, when $A = I$ the equation $f = \mathcal{T}_\sigma(f)$ admits an explicit solution f^*, which is unique even when $\mu = 0$. In fact, since $\frac{1}{2}\nabla F(f) = (1 + \mu)f + y$, by setting $\sigma = 1 + \mu$, we obtain

$$f^* = \frac{y}{1 + \mu} - \pi_{\frac{\tau}{1+\mu}K}\left(\frac{y}{1 + \mu}\right) = \frac{1}{1 + \mu}\left(y - \pi_{\tau K}(y)\right).$$

For a general operator A, the solution of $f = \mathcal{T}_\sigma(f)$ does not admit a closed form, but we can compute it using Algorithm 1, provided that the map $\mathcal{T}_\sigma$ is a contraction.

Proposition 4 *Assume the penalty term to satisfy the assumptions in Theorem 1 and F to be given by (13). Let a and b be the smallest and largest eigenvalues of* $A^T A$, *where* A^T *denotes the adjoint of A. Then the map* $\mathcal{T}_\sigma$ *is a contraction if we choose* σ *such that*

$$max\left\{\left|1 - \frac{a + \mu}{\sigma}\right|, \left|1 - \frac{b + \mu}{\sigma}\right|\right\} < 1.$$

The optimal a priori choice for the step-size is given by $\sigma = \frac{a+b+2\mu}{2}$, *and in this case we can choose* $L_\sigma = \frac{a-b}{a+b+2\mu}$.

3.2 Adaptive Step-Size Choice

In the previous sections we proposed a general scheme as well as a parameter set-up ensuring convergence of the proposed procedure. Here, we discuss some heuristics that were observed to consistently speed up the convergence of the iterative procedure. In particular, we mention the Barzilai-Borwein methods – see for example [40, 32, 33] for references. The rationale behind these methods is that one can compute a step-size that mimics the behavior of the Hessian of the data-fit term F at the most recent iteration. More precisely in the following we will consider

$$\sigma_t = \frac{\langle s^t, r^t \rangle}{\|s^t\|^2}$$

where $s^t = f^t - f^{t-1}$ and $r^t = \nabla F(f^t) - \nabla F(f^{t-1})$. Alternatively, one can consider

$$\sigma_t = \frac{\|r^t\|^2}{\langle s^t, r^t \rangle}.$$

More sophisticated step-size strategy can be designed alternating these two choices (see [32, 33]). Here we just recall that, though the above choices lack a theoretical justification, they were empirically shown to yield improved convergence in several studies (see [32, 33] and references therein).

3.3 Continuation Strategies and Regularization Path

Finally, we recall the *continuation* strategy proposed in [25] to compute efficiently the solutions corresponding to different regularization parameter values, often called *regularization path*. In this case, one should run the iterative procedure (up-to convergence) for each regularization parameter value. The general idea of the approach proposed in [25] is that one can try to reduce the number of iterations needed to compute the solutions corresponding to each parameter value by choosing suitable initializations. More precisely, one can fix an ordered sequence of regularization parameter values $\tau_1 > \tau_2 > \cdots > \tau_p$ and start considering the larger value. The corresponding solution can be usually computed in a fast way since it is very sparse, though possibly under-fitting the data. Then, one proceeds considering the next parameter value τ_2 and use the previously computed solution as the starting point of the corresponding procedure. It can be observed that with this initialization much fewer iterations are typically required to achieve convergence. The same *warm starting* strategy is then repeated to compute the solutions corresponding to the following parameter values.

4 Examples

In this section we discuss several examples of the general Algorithm 1, specializing our analysis to a number of well known regularization schemes.

4.1 Lasso and elastic net regularization

We start considering the following functional

$$\mathcal{E}_\tau^{(\ell_1 \ell_2)}(\beta) = \|\Psi\beta - y\|^2 + \mu \sum_{j=1}^{M} \beta_j^2 + 2\tau \sum_{j=1}^{M} w_j |\beta_j|, \tag{14}$$

where Ψ is a $n \times M$ matrix, β, y are the vectors of coefficients and measurements respectively, and $(w_j)_{j=1}^M$ are positive weights. The matrix Ψ is given by the features ψ_j in the dictionary evaluated at some points $x_1, \ldots, x_n$.

Minimization of the above functional corresponds to the so called elastic net regularization, or ℓ_1-ℓ_2 regularization, proposed in [45], and reduces to the lasso algorithm [38] if we set $\mu = 0$. Using the notation introduced in the previous sections, we set $F(\beta) = \|\Psi\beta - y\|^2 + \mu \sum_{j=1}^M \beta_j^2$

and $J(\beta) = \sum_{j=1}^{M} w_j|\beta_j|$. Moreover we denote by $\mathbf{S}_{\tau/\sigma}$ the soft-thresholding operator defined component-wise by

$$[\mathbf{S}_{\tau/\sigma}(\beta)]_j = \mathrm{sign}(\beta_j)(|\beta_j| - \lambda w_j)_+.$$

The minimizer of (14) can be computed by Algorithm 2. It is easy to check that the argument of

Algorithm 2 Iterative Soft Thresholding

```
Require: $\sigma, \tau, \mu > 0$
Initialize: $\beta^0 = 0$
while convergence not reached do
  $p := p + 1$
    $$\beta^p = \mathbf{S}_{\frac{\tau}{\sigma}}\left((1 - \frac{\mu}{\sigma})\beta^{p-1} + \frac{1}{\sigma}\Psi^T(y - \Psi\beta^{p-1})\right) \qquad (15)$$
end while

return $\beta^p$
```

$\mathbf{S}_{\tau/\sigma}$ is obtained simply computing the derivative of $F(\beta)$ and the main point while passing from equation (5) to equation (15) is the computation of the projection $\pi_{\lambda K}$. Applying Proposition 2 to $J(\beta) = \sum_{j=1}^{M} w_j|\beta_j|$, with $\mathcal{G}_j = \mathbb{R}$ and $\mathcal{J}_j(\beta) = w_j\beta_j \ \forall j = 1, \dots, M$, allows to solve (8) component-wise as

$$\bar{v}_j = \underset{|v_j|\leq 1}{\mathrm{argmin}}(\lambda w_j v_j - \beta_j)^2 = \min\left\{1, \frac{|\beta_j|}{\lambda w_j}\right\}\mathrm{sign}(\beta_j),$$

where we used the fact that $\mathcal{J}_j^T v = w_j v_j$.
The operator $\mathbf{S}_{\tau/\sigma}$ introduced above corresponds to the non linear operation $(I - \pi_{\lambda K})$, which acts on each component as:

$$[(I - \pi_{(\lambda K)})(\beta)]_j = \beta_j - \min\{|\beta_j|, \lambda w_j\}\mathrm{sign}(\beta_j) = \mathrm{sign}(\beta_j)(|\beta_j| - \lambda w_j)_+.$$

From the above equation it follows that the iteration (15) with $\mu = 0$ leads to the iterated soft-thresholding studied in [12] (see also [42] and references therein). When $\mu > 0$, the iteration (15) becomes the damped iterated soft-thresholding proposed in [14]. In the former case, the operator $\mathcal{T}_\sigma$ in (15) is not contractive but only non-expansive, convergence in this case is proved in [12].

4.2 Group lasso

We consider a variation of the above algorithms where the features are assumed to be disposed in *blocks*. This latter assumption is used in [43] to define the so called group lasso, which amounts to minimizing

$$\mathcal{E}_\tau^{(grLasso)}(\beta) = \|\Psi\beta - y\|^2 + \mu\|\beta\|^2 + 2\tau\sum_{k=1}^{M} w_k\sqrt{\sum_{j\in\mathcal{I}_k}\beta_j^2} \qquad (16)$$

for $\mu = 0$, where $(\psi_j)_{j\in\mathcal{I}_k}$ for $k = 1, \dots, M$ is a block partition of the feature set $(\psi_j)_{j\in\mathcal{I}}$. As in the previous case the main step towards specializing Algorithm 1 to this particular example is the computation of $\pi_{\lambda K}$. Note that when applying Proposition 2 to $\mathcal{J}_k(\beta) = \sqrt{w_k}\beta_{\mathcal{I}_k}$ with $\mathcal{G}_k = \mathbb{R}^{|\mathcal{I}_k|}$, equation (8) can be decomposed component-wise as

$$\begin{aligned} \bar{v}_k &= \operatorname*{argmin}_{v\in\mathbb{R}^{|\mathcal{I}_k|},\ \|v\|_k\leq 1} \|\lambda w_k v - \beta_k\|_k^2 \\ &= \min\left\{1, \frac{||\beta^{(k)}||}{\lambda w_k}\right\} \frac{\beta^{(k)}}{||\beta^{(k)}||} \end{aligned}$$

where $\bar{v} = (\bar{v}_1, \dots, \bar{v}_M)$ with $\bar{v}_k \in \mathbb{R}^{|\mathcal{I}_k|}$, and $\beta^{(k)} \in \mathbb{R}^{|\mathcal{I}_k|}$ is the vector built with the components of $\beta \in \mathbb{R}^{|\mathcal{I}|}$ corresponding to the elements $(\psi_j)_{j\in\mathcal{I}_k}$.
The nonlinear operation $(I - \pi_{\lambda K})$ – denoted by $\tilde{\mathbf{S}}_{\tau/\sigma}$ – acts on each block as

$$\begin{aligned} [(I - \pi_{(\lambda K)})(\beta)]^{(k)} &= \beta^{(k)} - \min\left\{\lambda w_k, ||\beta^{(k)}||\right\} \frac{\beta^{(k)}}{||\beta^{(k)}||} \\ &= \frac{\beta^{(k)}}{||\beta^{(k)}||}(||\beta^{(k)}|| - \lambda w_k)_+ \end{aligned}$$

The minimizer of (16) can hence be computed through Algorithm 3.

Algorithm 3 Group lasso Algorithm

```
Require: $\tau, \sigma > 0$
Initialize: $\beta^0 = 0$
while convergence not reached do
    $p := p + 1$
        $\beta^p = \tilde{\mathbf{S}}_{\frac{\tau}{\sigma}}\left((1 - \frac{\mu}{\sigma})\beta^{p-1} + \frac{1}{\sigma}\Psi^T(y - \Psi\beta^{p-1})\right)$
end while

return $\beta^p$
```

4.3 Composite Absolute Penalties

In [44], the authors propose a novel penalty, named Composite Absolute Penalty (CAP), based on assuming possibly overlapping groups of features. Given $\gamma_k \in \mathbb{R}^+$, for $k = 0, 1, \dots, M$, the penalty is defined as:

$$J(\beta) = \sum_{k=1}^{M}(\sum_{j\in\mathcal{I}_k} \beta_j^{\gamma_k})^{\frac{\gamma_0}{\gamma_k}},$$

where $(\psi_j)_{j\in\mathcal{I}_k}$ for $k = 1, \dots, M$ is not necessarily a block partition of the feature set $(\psi_j)_{j\in\mathcal{I}}$. This formulation allows to incorporate in the model not only groupings, but also hierarchical structures present within the features, for instance by setting $\mathcal{I}_k \subset \mathcal{I}_{k-1}$. The choice of γ_k for $k \geq 1$ corresponds to a priori information about the sparsity within a group, while the choice of γ_0 is on the other hand related to the sparsity among groups. For $\gamma_0 = 1$, the CAP penalty is one-homogeneous and the solution can be computed through Algorithm 1. Furthermore, when $\gamma_k = 2$ for all $k = 1, \dots, M$, it can be regarded as a particular case of (7), with $\|\mathcal{J}_k(\beta)\|^2 = \sum_{j=1}^d \beta_j^2 \mathbf{1}_{\mathcal{I}_k}(j)$, with $\mathcal{J}_k : \mathbb{R}^{|\mathcal{I}_k|} \to \mathbb{R}^{m_k}$ and $m_k = |\mathcal{I}_k|$. Considering the least square error, we study the minimization of the functional

$$\mathcal{E}_\tau^{(CAP)}(\beta) = \|\Psi\beta - y\|^2 + \mu\,\|\beta\|^2 + 2\tau \sum_{k=1}^M w_k \sqrt{\sum_{j \in \mathcal{I}_k} \beta_j^2}, \tag{17}$$

which is exactly a CAP functional when $\mu = 0$. Note that, due to the overlapping structure of the features groups, the minimizer of (17) cannot be computed blockwise as in Algorithm 3, because the solution of the minimization problem (8) does not decouple on the blocks. However we can approximate the projection using Theorem 2, through the iterative scheme (10), by identifying $\mathcal{J}_k$ with the projection on the components corresponding to $\mathcal{I}_k$, and $\mathcal{J}$ as $(\mathcal{J}_1^T \dots \mathcal{J}_k^T)^T$. We can then compute the minimizer of (17) through Algorithm 4.

Algorithm 4 CAP Algorithm

```
Require: $\tau, \sigma > 0$
Initialize: $\beta^0 = 0$
for $p = 1, 2, \dots,$ MAX_ITER_EXT do
  set $v^0 = 0$, $\tilde{\beta} = (1 - \frac{\mu}{\sigma})\beta^{p-1} + \frac{1}{\sigma}\Psi^T(y - \Psi\beta^{p-1})$
  for $q = 1, 2, \dots,$ MAX_ITER_INT do
    $v_k^{q+1} = \dfrac{v_k^q - \eta\mathcal{J}_k(\mathcal{J}^T v^q - \sigma\tilde{\beta}/\tau)}{1 + \eta||\mathcal{J}_k(\mathcal{J}^T v^q - \sigma\tilde{\beta}/\tau)||}$
  end for
  $\beta^p = \tilde{\beta} - \frac{\tau}{\sigma}\mathcal{J}^T v^{\texttt{MAX\_ITER\_INT}}$
end for
return $\beta^{\texttt{MAX\_ITER\_EXT}}$
```

4.4 Multiple kernel learning

Multiple kernel learning (MKL) [3, 35] is the process of finding an optimal kernel from a prescribed (convex) set $\mathcal{K}$ of basis kernels, for learning a real-valued function by regularization. This approach has applications in kernel selection, and data fusion from heterogeneous data sources, and nonlinear feature selection [29]. In the case where the set $\mathcal{K}$ is the convex hull of a finite number of kernels $k_1, \dots, k_M$, and the loss function is the square loss, it is possible to show [34] that the problem of multiple kernel learning corresponds to find f^* belonging to

$$\operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n (\sum_{j=1}^M f_j(x_i) - y_i)^2 + \tau g\left((\sum_{j=1}^M \|f_j\|_{\mathcal{H}_j})^2 \right) \right\}, \tag{18}$$

where $\mathcal{H} = \mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_M$ so that $f = \sum_{j=1}^M f_j$, $f_j \in \mathcal{H}_j$ and $g : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing function. Two popular choices for the function g are the identity and the square root, both leading to a representation of f^* w.r.t. the basis kernels. In the following we consider the optimization problem

$$\operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n (\sum_{j=1}^M f_j(x_i) - y_i)^2 + \mu \sum_{j=1}^M \|f_j\|_{\mathcal{H}_j}^2 + 2\tau \sum_{j=1}^M \|f_j\|_{\mathcal{H}_j} \right\},$$

which recovers (18) for $\mu = 0$, when g is twice the square root. Note that by choosing the square root, our general hypotheses on the penalty term J are satisfied.

Though the space of functions is infinite dimensional the minimizer of the above functional (18) can be shown to have a finite representation. In fact, one can generalize the representer theorem to show that the vector of optimal components for the solution of the optimization problem (18) can be expressed as $f_j^*(\cdot) = \sum_{i=1}^n \alpha_{j,i} k(x_i, \cdot)$ for all $j = 1, \ldots, M$. Introducing the following notation:

$$\begin{aligned}
&\alpha = (\alpha_1, \ldots, \alpha_M)^T \text{ with } \alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,n})^T, \\
&\mathbf{k}_j(x) = (k_j(x_1, x), \ldots, k_j(x_n, x)), \\
&\mathbf{k}(x) = (\mathbf{k}_1(x), \ldots, \mathbf{k}_M(x))^T \\
&\mathrm{K} = \operatorname{diag}(\mathrm{K}_1, \ldots, \mathrm{K}_M) \text{ with } [\mathrm{K}_j]_{ii'} = k_j(x_i, x_i'), \\
&\mathbf{y} = (\underbrace{y^T, \ldots, y^T}_{M \text{ times}})^T
\end{aligned}$$

we can write the solution of (18) as $f^*(x) = \alpha_1^T \mathbf{k}_1(x) + \cdots + \alpha_M^T \mathbf{k}_M(x))$.
The search for the solution can then be restricted to a finite dimensional space spanned by $\mathbf{k}_1, \ldots, \mathbf{k}_M$. Hence, the iteration on the vector of components can be written as

$$\mathcal{T}_\sigma(f) = \left(I - \pi_{\tau/\sigma K}\right) \left(\left((1 - \frac{\mu}{\sigma})\alpha - \frac{1}{\sigma n} (\mathrm{K}\alpha - \mathbf{y}) \right)^T \mathbf{k} \right).$$

Defining $\mathcal{J} : \mathcal{H} \to \mathcal{H}$ to be the identity operator and $\mathcal{J}_j(f) = f_j$, we apply Proposition 2, obtaining that the projection is defined as $\pi_{\lambda K}(g) = \lambda \bar{v}$ with

$$\bar{v} = \operatorname*{argmin}_{v \in \mathcal{H}, \|\mathcal{J}_j v\|_{\mathcal{H}_j} \leq 1} \|\lambda v - g\|_{\mathcal{H}}^2,$$

which can be computed block-wise as

$$\bar{v}_j = \min\left\{1, \frac{\|\mathcal{J}_j g\|_{\mathcal{H}_j}}{\lambda}\right\} \frac{\mathcal{J}_j g}{\|\mathcal{J}_j g\|_{\mathcal{H}_j}} = \min\left\{1, \frac{\sqrt{\alpha_j^T \mathrm{K}_j \alpha_j}}{\lambda}\right\} \frac{\alpha_j^T \mathbf{k}_j}{\sqrt{\alpha_j^T \mathrm{K}_j \alpha_j}},$$

with $g = (\alpha_1 \cdot \mathbf{k}_1, \ldots, \alpha_M \cdot \mathbf{k}_M)$. The operation $(I - \pi_{\lambda K})(g)$, therefore acts on g componentwise by changing the coefficients of the expansion i.e. we can write it as $\hat{\mathbf{S}}_\lambda(\mathrm{K}, \alpha)^T \mathbf{k}$ for $j = 1, \ldots, M$ where

$$\hat{\mathbf{S}}_\lambda(\mathrm{K}, \alpha)_j = \frac{\alpha_j^T}{\sqrt{\alpha_j^T \mathrm{K}_j \alpha_j}} (\sqrt{\alpha_j^T \mathrm{K}_j \alpha_j} - \lambda)_+.$$

This peculiarity allows for computing the regularized solution using Algorithm 5, which involves only the coefficients.

4.5 Multitask Learning

Learning multiple tasks simultaneously has been shown to improve performance relative to learning each task independently, when the tasks are related in the sense that they all share a small set of features (see for example [1, 28, 36] and references therein).

Algorithm 5 MKL Algorithm

```
set $\alpha^0 = 0$
for $p = 1, 2, \ldots, \texttt{MAX\_ITER}$ do
    $\alpha^p = \hat{\mathbf{S}}_{\tau/\sigma}\left(K, \left((1 - \frac{\mu}{\sigma})\alpha^{p-1} - \frac{1}{\sigma n}(\mathrm{K}\alpha^{p-1} - \mathbf{y})\right)\right)$
end for
return $\left(\alpha^{\texttt{MAX\_ITER}}\right)^T \mathbf{k}$.
```

In particular, given T tasks modeled as

$$f_t(x) = \sum_{j=1}^{d} \beta_{j,t}\psi_j(x)$$

for $t = 1, \ldots, T$, according to [36], regularized multi-task learning amounts to the minimization of the functional

$$\mathcal{E}_\tau^{(MT)}(\beta) = \sum_{t=1}^{T} \frac{1}{n_t} \sum_{i=1}^{n_t} (\psi(x_{t,i})\beta_t - y_{t,i})^2 + \mu \sum_{t=1}^{T}\sum_{j=1}^{d} \beta_{t,j}^2 + 2\tau \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} \beta_{t,j}^2}. \qquad (19)$$

The last term combines the tasks and ensures that common features will be selected across them. Again functional (19) is a particular case of (1), and, defining

$$\beta = (\beta_1^T, \ldots, \beta_T^T)^T,$$

$$\Psi = \mathrm{diag}(\Psi_1, \ldots, \Psi_T), \qquad [\Psi_t]_{ij} = \psi_j(x_{t,i}),$$

$$y = (y_1^T, \ldots, y_T^T)^T,$$

$$\mathbf{N} = \mathrm{diag}(\underbrace{1/n_1, \ldots, 1/n_1}_{n_1 \text{times}}, \underbrace{1/n_2, \ldots, 1/n_2}_{n_2 \text{times}}, \ldots, \underbrace{1/n_T, \ldots, 1/n_T}_{n_T \text{times}}).$$

its minimizer can be computed through Algorithm 6. Using Proposition 2 the projection corresponds to a task-wise soft-thresholding $\tilde{\mathbf{S}}_\lambda$ acting simultaneously on the regression coefficients relative to the same variable in all the tasks.

Algorithm 6 Multi-Task Learning Algorithm

```
set $\beta^0 = 0$
for $p = 1, 2, \ldots, \texttt{MAX\_ITER}$ do
    $\beta^p = \tilde{\mathbf{S}}_{\frac{\tau}{\sigma}}\left((1 - \frac{\mu}{\sigma})\beta^{p-1} + \frac{1}{\sigma}\Psi^T\mathbf{N}(y - \Psi\beta^{p-1})\right)$
end for
return $\beta^{\texttt{MAX\_ITER}}$
```

Note that, when $n_1 = n_2 = \cdots = n_T = n$, dividing by the diagonal matrix $\mathbf{N}$ amounts to multiplying by the factor $1/n$.

4.6 Total Variation–based Image Denoising

As a last example we consider total variation regularization for image denoising, which amounts to the minimization of the following functional

$$\mathcal{E}_\tau^{(TV)}(f) = \|f-y\|^2 + 2\tau \sum_{i,j=1}^{n} \|[\nabla(f)]_{ij}\| \tag{20}$$

where y is a noisy $n \times n$ image, from which we aim at extracting the true image f, and ∇ is a linear discretization of the gradient operator. The minimization of $\mathcal{E}_\tau^{TV}$ can be easily recast in terms of (1). In fact, $f = \{f_{ij}\}_{i,j=1}^{n}$ so that $\mathcal{H} = \mathbb{R}^{n\times n}$, the operator A is the identity, $\mu = 0$ and $\mathcal{J}_{ij}(f) = (\nabla f)_{ij} \in \mathbb{R}^2$.

Since $A = I$, as pointed out in Section 2, the solution is simply $f^* = y - \pi_{\tau K}(y)$, and the projection can be efficiently implemented through the iterative algorithm (10). If one approximates the operator ∇ by means of finite differences of neighbors pixels,

$$[(\nabla f)_{i,j}]_1 = \begin{cases} f_{i+1,j} - f_{i,j} & \text{if } i < n \\ 0 & \text{if } i = n \end{cases} \qquad [(\nabla f)_{i,j}]_2 = \begin{cases} f_{i,j+1} - f_{i,j} & \text{if } j < n \\ 0 & \text{if } j = n. \end{cases}$$

With this choice the adjoint of $\mathcal{J}$ is given by

$$\begin{aligned}(\nabla^T v)_{i,j} &= \begin{cases} [v_{i-1,j}]_1 - [v_{i,j}]_1 & \text{if } \ 1 < i < n \\ -[v_{i,j}]_1 & \text{if } \ i = 1 \\ [v_{i-1,j}]_1 & \text{if } \ i = n \end{cases} + \begin{cases} [v_{i,j-1}]_2 - [v_{i,j}]_2 & \text{if } \ 1 < j < n \\ -[v_{i,j}]_2 & \text{if } \ j = 1 \\ [v_{i,j-1}]_2 & \text{if } \ j = n \end{cases} \\ &= -(\text{div } v)_{i,j}\end{aligned}$$

and the minimizer of (20) can be computed through the iterative Algorithm 7. Through our approach we recover the algorithm proposed in [8].

Algorithm 7 Total Variation Algorithm

```
set $v^0 = 0$
for $p = 0, 1, \dots,$ MAX_ITER do
    $$v_{i,j}^{q+1} = \frac{v_{i,j}^q + \eta(\nabla(\text{div } v^q + \sigma y/\tau))_{i,j}}{1 + \eta\,\|(\nabla(\text{div } v^q + \sigma y/\tau))_{i,j}\|_{\mathbb{R}^2}}.$$
end for
return $-\frac{\tau}{\sigma}\text{div } v^{\texttt{MAX\_ITER}}$
```

5 Experiments and Discussions

In this section we describe several experiments aimed at testing some features of the proposed method. In particular, we investigate the effect of adding a (small) strictly convex perturbation to the original functional, weighted by a positive parameter μ. One can see that such a perturbation term simplifies the mathematical analysis of the proposed procedure, and here we argue that it might actually bear benefits from the numerical point of view without affecting statistical properties. In the following we are interested into understanding the role of the perturbation term with respect to:

- *prediction*: do different values of μ modify the prediction error of the obtained estimator?

- *selection*: does μ increase/decrease the sparsity level of the obtained estimator?

- *running time*: is there an actual computational improvement due to the use of $\mu > 0$?

We discuss the above questions for the multi-task scheme proposed in [36]. and show results which are consistent with those reported in [15] for the elastic-net estimator. These two methods are only two special cases of our framework, but indeed we believe that all the other learning algorithms considered in this paper share the same properties.

We note that a computational comparison of different optimization approaches is cumbersome since we consider many different learning schemes and is beyond the scope of this paper. Extensive analyses of different approaches to solve ℓ_1 regularization can be found in [25] and [32], where the authors show that projected gradient methods compare favorably to state of the art methods. We expect that similar results will hold for learning schemes other than ℓ_1 regularization.

5.1 Validation protocol and simulated data

In this section, we briefly present the set-up used in the experiments, by first describing the data sets. Simulated data were considered to test the properties of the proposed method in a controlled scenario. More precisely, we considered T regression tasks

$$y = x \cdot \beta^{(t)} + \varepsilon \qquad t = 1, \ldots, T$$

where x is uniformly drawn from $[0,1]^d$, ε is drawn from the zero-mean Gaussian distribution with $\sigma = 0.1$ and the regression vectors are

$$\beta_t^\dagger = (\underbrace{\beta_{t,1}^\dagger, ..., \beta_{t,r}^\dagger}_{r}, \underbrace{0, 0, ..., 0}_{d-r}).$$

with $\beta_{t,j}^\dagger$ uniformly drawn from $[-1, 1]$. In other words the only relevant variables are the first r.

In order to obtain a fully data driven procedure we use cross validation to choose the regularization parameters τ, λ for the sparse regularization and RLS respectively. Indeed one can see that the cross validation protocol yields relatively large values of τ and very small values of λ. After re-training with the optimal regularization parameters, a test error is computed on an independent set of data. Each validation protocol is replicated 20 times by resampling both the input data and the regression coefficients, $\beta_t^\dagger$, in order to assess the stability of the results.

Finally, following [7, 40], we consider a debiasing step after running the sparsity based procedure. This last step is a post-processing and corresponds to training a regularized least square (RLS) estimator[3] on the selected components to avoid an undesired shrinkage of the corresponding coefficients.

5.2 A preliminary result

The potential benefits of multi task learning compared to learning each task independently, has already been demonstrated both on simulated and real data – see [1, 28, 36] and references therein. Here we just confirm such results in our framework. Towards this end we compare

[3] A simple ordinary least square is often sufficient and here a little regularization is used to avoid possible unstable behaviors especially in the presence of small samples.

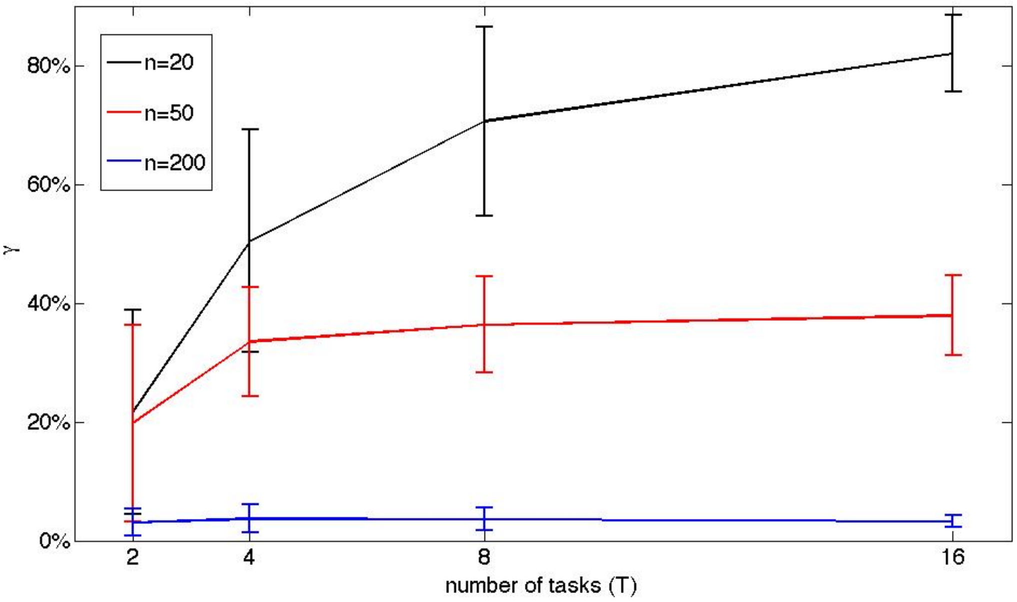


Figure 1: Comparison measure (γ) between a multiple task and independent task approach, for different values of n. For these experiments we worked with $d = 100$, $r = 10$, $T = 2, 4, 8, 16$, and $n = 20, 50$ and 100. The number of points for both validation and test is 1000.

single task learning using Algorithm 2 and multi-task learning using Algorithm 6. We apply the experimental protocol described in the previous section (for $\mu = 0$) and use the following comparison measure

$$\gamma = \frac{err_S - err_M}{err_S},$$

where err_M is the multi-task error and err_S is the single task error. The results obtained varying the number of training points and tasks are reported in Figure 1. This experiment confirms the effectiveness of multitask learning: the relative difference between test error increases with the number of tasks, i.e. the advantage of combining multiple tasks is larger when the number of task is large. Furthermore, such effect is more evident with small training sets.

5.3 Role of the Strictly convex Penalty

Next, we investigate the impact of adding the perturbation $\mu > 0$. We consider $T = 2$, $r = 3$, $d = 10, 100, 1000$, and $n = 8, 16, 32, 64, 128$. For each data set, that is for fixed d and n, we apply the validation protocol described in Subsection 5.1 for increasing values of μ. The number of samples in the validation and test sets is again 1000. As in the previous set of experiments we replicate each trial 20 times and report the mean results. Error bars are omitted in order to increase the readability of the Figures.

We preliminary discuss an observation suggesting a useful way to vary μ. As a consequence of Corollary 1, when $\mu = 0$ and $b = 0$, the Lipschitz constant, L_σ, of the map $\mathcal{T}_\sigma$ in Theorem 1 is 1 so that $\mathcal{T}_\sigma$ is not a contraction. By choosing $\mu = \frac{\|\nabla^2 F\|}{4}\alpha$ with $\alpha > 0$, the Lipschitz constant becomes

$$L_\sigma = (1 + \alpha)^{-1} < 1,$$

and the map $\mathcal{T}_\sigma$ induced by F_μ is a contraction. In particular in multiple task learning with linear features (see Section 4.5) $X = \Psi$, so that $\nabla^2 F = 2X^T X/n$ and $\left\|\nabla^2 F\right\| = 2a/n$, where a is the largest eigenvalue of the simmetric matrix $X^T X$. We therefore let $\mu = \frac{a}{2n}\alpha$ and vary the absolute parameter α as $\alpha = 0,\ 0.001,\ 0.01,\ 0.1$. We then compare the results obtained for different values of α.

We now analyze in the details the outcome of our results in terms of the three aspects raised at the beginning of this section.

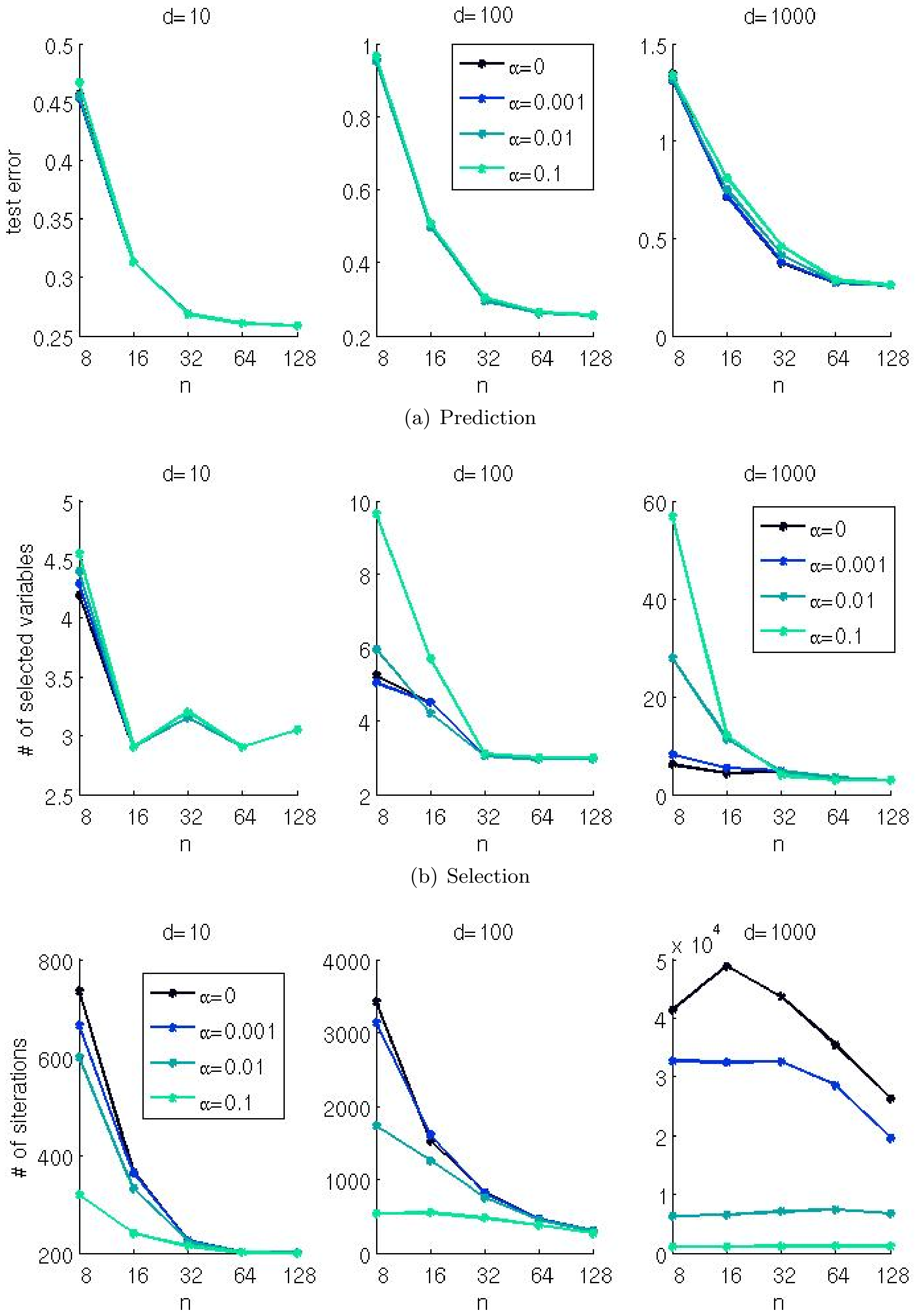
- *prediction* The test errors associated to different values of μ are essentially overlapping, meaning that the perturbation term does not impact the prediction performance of the algorithm when the τ parameter is accurately tuned. This result is consistent with the theoretical results for the elastic net estimator – see [14].
- *selection*

  In principle the presence of the perturbation term tends to reduce the sparsity of the solution in the presence of very small samples. In practice one can see that such an effect decreases when the number of input points n increases and is essentially negligible even when $n << d$.
- *running time* From the computational point of view we expect larger values of μ (that is α) to correspond to fewer iterations. This effect is clear in our experiments. Interestingly when $n << d$ small values of μ allow to substantially reduce the computational burden while preserving sparsity and prediction properties of the algorithm (compare $\alpha = 0$ and $\alpha = 0.001$ when $d = 1000$). Moreover, one can observe that the number of iterations decreases as the number of points increases. This result might seems surprising, but can be explained recalling that the condition number of the underlying problem is likely to improve as n gets bigger.

Finally, we can see that adding the small strictly convex perturbation with $\mu > 0$, has a preconditioning effect on the iterative procedure and can substantially reduce the number of required computations without affecting the sparsity and prediction properties of the obtained solution.

5.4 Impact of choosing the step-size adaptively

In this section we assess the effectiveness of the adaptive approach proposed in section 3 to speed up the convergence of the algorithm. Specifically, we show some results obtained by running the iterative optimization with two different choices of the step-size, namely the one fixed a-priori – as described in section 3.1 – and the adaptive alternative of section 3.2.
The experiments have been conducted by first drawing randomly the dateset and finding the optimal solution using the complete validation scheme, and then running two further experiments using, in both cases, the optimal regularization parameters but the two different strategies for the step-size.
We compared the number of iterations necessary to compute the solution and looked at the ratio between those required by the fixed and the adaptive strategies respectively. In Figure 3, it is easy to note that such ratio is always greated than one, and actually it ranges from the order of tens to the order of hundreds. Moreover, the effectiveness of using an adaptive strategy becomes more and more evident as the number of input variables increases.
Consistently with the results showed in the previous section, for a fixed input dimension, the



(a) Prediction

(b) Selection

(c) Running time

Figure 2: Results obtained in the experiments varying the size of the training set and the number of input variables. The properties of the algorithms are evaluated in terms of the prediction error, the ability of selecting the *true* relevant variables, and finally the number of iteration required for the convergence of the algorithm.

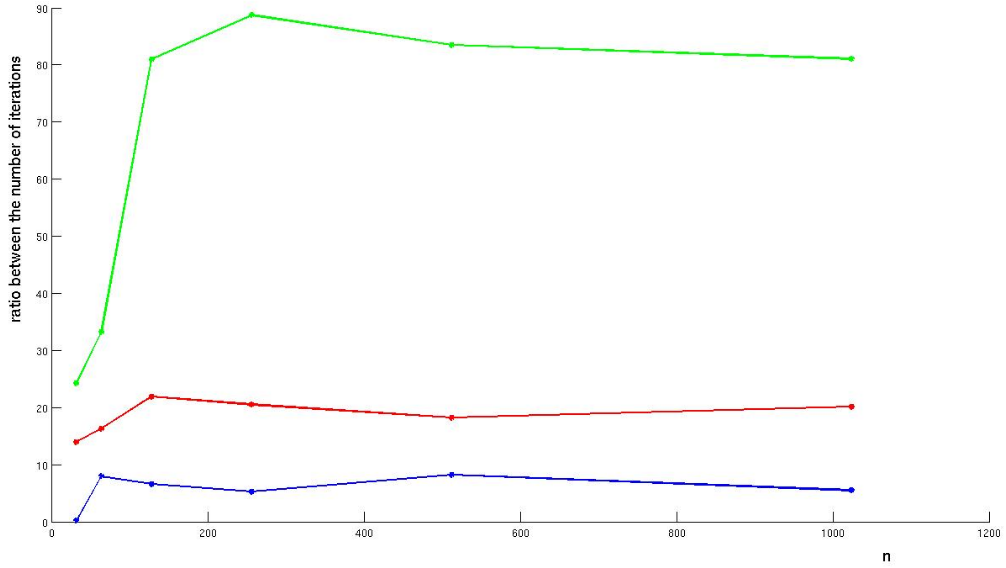


Figure 3: Comparison of the number of iterations required to compute the regression function using the fixed and the adaptive step-size. The blue plot refers to the experiments using $d = 10$, the red plot to $d = 100$, while the green plot to $d = 500$.

iterations required by the fixed step-size approach decreases when the number of training samples increases. Indeed also in the case of the adaptive choice appoach the number of iterations decreases but at a slightly faster rate. Therefore, the ratio tends to either remain approximately constant or decrease slightly.

6 Conclusions

This paper shows that many algorithms based on regularization with convex non differentiable penalties can be described within a common framework. This allows to derive a general optimization procedure whose convergence is guaranteed. The proposed procedure highlights and separates the roles played by the loss terms and the penalty terms, in fact, it corresponds to the iterative projection of the gradient of the loss on a set defined by the penalty. The projection has a simple characterization in the setting we consider: in many case it can be written in closed form and corresponds to a soft-thresholding operator, in all the other cases it can be iteratively calculated. The obtained procedure is simple and its convergence proof is relatively straightforward in the strictly convex case. One can always force such a condition considering a suitable perturbation of the original functional. Interestingly if such a perturbation is small it will act as a preconditioning of the problem and lead to the better computational performances without changing the properties of the solution. A more general and abstract setting can be considered. Steps in this direction are taken for example in [10]. The setting we consider here is general enough to be of interest for many learning algorithms and allows to have simplified proofs.

In future work we will consider several natural developments of our study. In particular one can study more carefully the properties the adaptive step-size choice and consider domain decomposition techniques aimed at dealing with large scale problems which are common in machine learning.

Acknowledgments

This work has been partially supported by the FIRB project LEAP RBIN04PARL and by the EU Integrated Project Health-e-Child IST-2004-027749.

A Proofs

In this section we collect the proofs of the results in the paper. We start by proving Theorem 1. The proof requires a few basic concepts from convex analysis [20]. In particular we recall that the Fenchel conjugate of a convex functional is defined as

$$\begin{array}{rccl} J^* & : & \mathcal{H} & \rightarrow & \mathbb{R} \cup \{+\infty\} \\ & & g & \mapsto & \sup_{f\in\mathcal{H}} \langle f, g\rangle_{\mathcal{H}} - J(f), \end{array}$$

and satisfies the well known Young-Fenchel equality:

$$g \in \partial J(f) \iff f \in \partial J^*(g). \tag{21}$$

The above equality is the key for the proof of Theorem 1 and leads to a dual formulation of the minimization problem (1). Another important fact is that the conjugate of a one-homogeneous functional J is the indicator function of the convex set $K = \partial J(0)$ and this implies that the solution of the dual problem reduces to the projection onto K. In the proof of Proposition 2, we are also going to use some standard properties of the subdifferential, that can be found in [20], Chapter 1. For the convenience of the reader we recall them here, without stating all the needed assumptions that are systematically satisfied in our setting.

P1) *Sum rule*: if F and J are convex, then $\partial(F + J)(f) = \partial F(f) + \partial J(f)$;

P2) *Chain rule*: let L be a linear operator and F a convex function, then

$$\partial(F \circ L)(f) = L^T(\partial F(L(f)))$$

P3) *Subdifferential of the norm in a Hilbert space H*:

$$(\partial \|\cdot\|)(0) = \{v \in H : \|v\| \leq 1\} := B(H, 1).$$

We can now give the proof of Theorem 1.

Proof 1 (Theorem 1) *Since $\mathcal{E}_\tau$ is lower semicontinuous, strictly convex and coercive, it admits a unique minimizer, which is characterized by the Euler equation*

$$0 \in 2\tau\partial J(f) + \nabla F(f).$$

Using (21) this is equivalent to

$$f \in \partial J^* \left(-\frac{1}{2\tau} \nabla F(f) \right).$$

If we let $g = (f - \frac{1}{2}\nabla F(f))$, and add g/τ to both sides of the above relation, then we obtain

$$0 \in \frac{1}{\tau}(g - f) - \frac{g}{\tau} + \frac{1}{\tau}\partial J^* \left(\frac{1}{\tau}(g - f) \right).$$

It follows that $w = \frac{1}{\tau}(g - f)$ is the minimizer of

$$\frac{1}{2}\left\|w' - \frac{g}{\tau}\right\|_{\mathcal{H}}^2 + \frac{1}{\tau}J^*(w').$$

Since the penalty is one-homogeneus its Fenchel conjugate J^* is the indicator function of K, and we obtain

$$w = \underset{w' \in K}{\operatorname{argmin}} \left\|w' - \frac{g}{\tau}\right\|_{\mathcal{H}}^2 = \pi_K\left(\frac{g}{\tau}\right),$$

which immediately gives $f = g - \tau\pi_K\left(\frac{g}{\tau}\right) = g - \pi_{\tau K}(g)$. We conclude noting that we can multiply both F and τ by $\sigma > 0$, without modifying the minimizer of (1), which is therefore the unique fixed point of the mapping $\mathcal{T}_\sigma : \mathcal{H} \to \mathcal{H}$

$$\mathcal{T}_\sigma(f) = f - \frac{1}{2\sigma}\nabla F(f) - \pi_{\frac{\tau}{\sigma}K}\left(f - \frac{1}{2\sigma}\nabla F(f)\right),$$

and this ends the proof.

Next, we prove convergence and step-size choice in the general case.

Proof 2 (Proposition 1) *We first observe that the contraction $\mathcal{T}_\sigma$ can be decomposed as $\mathcal{T}_\sigma = (I - \pi_{\frac{\tau}{\sigma}K}) \circ B_\sigma$, with $B_\sigma(f) := f - \frac{1}{2\sigma}\nabla F(f)$. Since $(I - \pi_{\frac{\tau}{\sigma}K})$ has unitary Lipschitz constant as an immediate consequence of the projection theorem, it is enough to prove that the inner mapping B_σ is a contraction. According to a corollary of the Mean Value Theorem (see Corollary 4.3 of [30] for the infinite dimensional version), every Fréchet differentiable mapping B such that $\sup_{f\in\mathcal{F}} \|B'(f)\| < 1$ is a contraction, therefore it is enough to prove that the norm of B'_σ is bounded by the unit. We have:*

$$B'_\sigma(f) = I - \frac{1}{2\sigma}\nabla^2 F(f),$$

therefore

$$\|B'_\sigma\| \leq \max\left\{\left|1 - \frac{1}{2\sigma}a\right|, \left|1 - \frac{1}{2\sigma}b\right|\right\}.$$

Since $a \geq b > 0$ the r.h.s is strictly less than 1 and the first part of the thesis follows. The minimization of the function $\sigma \mapsto \max\{\left|1 - \frac{1}{2\sigma}a\right|, \left|1 - \frac{1}{2\sigma}b\right|\}$ gives the best a priori choice of σ, that is $\sigma = \frac{a+b}{4}$.

Proof 3 (Corollary 1) *It is enough to note that $\nabla^2 F_\mu = \nabla^2 F + 2\mu I$, implying that the smallest eigenvalue of $F + \mu\|\cdot\|_{\mathcal{H}}^2$ is uniformly bounded from below by 2μ. The rest of the thesis easily follows applying Proposition 1 to F_μ.*

Next we consider the results allowing to compute the projection. First we prove Proposition 2.

Proof 4 (Proposition 2) *Using properties (P1) and (P2) stated at the beginning of the Section, and setting $K = \partial J(0)$, we have*

$$K = \sum_{k=1}^{p} \left(\partial\left(f \mapsto \|\mathcal{J}_k f\|_k\right)\right)(0) = \sum_{k=1}^{p} \mathcal{J}_k^T \left(\partial\|\cdot\|_k\right)(0)$$

where thanks to property (P3), $\left(\partial\|\cdot\|_k\right)(0) = \{v_k \in \mathcal{G}_k : \|v_k\|_k \leq 1\}$. Then we can identify the set K with

$$K = \{\mathcal{J}^T v : v \in \mathcal{G}, \|v_k\|_k \leq 1\ \forall k\}.$$

The projection on λK is then defined as $\pi_{\lambda K}(g) = \lambda\mathcal{J}^T\bar{v}$, where $\bar{v}$ is given by (8).

Then we prove Theorem 2.

Proof 5 (Theorem 2) *Equation (9) holds also if we multiply by $-\eta$ with $\eta > 0$ and add v_k to both sides, hence obtaining*

$$-\eta \left(\mathcal{J}_k(\lambda \mathcal{J}^T v - g) + \left\| \mathcal{J}_k(\lambda \mathcal{J}^T v - g) \right\|_k v_k \right) + v_k = v_k,$$

so that v_k satisfies the fixed point equation

$$v_k = \frac{v_k - \eta \mathcal{J}_k \left(\mathcal{J}^T v - g/\lambda \right)}{1 + \eta \left\| \mathcal{J}_k \left(\mathcal{J}^T v - g/\lambda \right) \right\|_k}.$$

By induction it is easy to see that $\left\| v_k^q \right\|_k \leq 1$, for all k, q. We then define $\kappa = \left\| \mathcal{J} \mathcal{J}^T \right\|$, and introduce $h^q = (h_1^q, \ldots, h_p^q)$ and $\rho^q = (\rho_1^q, \ldots, \rho_p^q)$ with $h^q, \rho^q \in \mathcal{G}$ such that $h_k^q = \mathcal{J}_k(\mathcal{J}^T v^q - g/\lambda) \in \mathcal{G}_k$ and $\rho_k^q = \left\| h_k^q \right\| v_k^{q+1} \in \mathcal{G}_k$, so that $v_k^{q+1} = v_k^q - \eta(h_k^q + \rho_k^q)$.

$$\begin{aligned}
&\left\| \mathcal{J}^T v^{q+1} - \tfrac{g}{\lambda} \right\|_{\mathcal{H}}^2 - \left\| \mathcal{J}^T v^q - \tfrac{g}{\lambda} \right\|_{\mathcal{H}}^2 = \\
&\left\| \mathcal{J}^T (v^q - \eta(h^q + \rho^q)) - \tfrac{g}{\lambda} \right\|_{\mathcal{H}}^2 - \left\| \mathcal{J}^T v^q - \tfrac{g}{\lambda} \right\|_{\mathcal{H}}^2 = \\
&-2\eta \langle \mathcal{J}^T (h^q + \rho^q), \mathcal{J}^T v^q - \tfrac{g}{\lambda} \rangle_{\mathcal{H}} + \eta^2 \left\| \mathcal{J}^T (h^q + \rho^q) \right\|_{\mathcal{H}}^2 = \\
&-2\eta \langle h^q + \rho^q, h^q \rangle + \eta^2 \left\| \mathcal{J}^T (h^q + \rho^q) \right\|_{\mathcal{H}}^2 = \\
&-\eta \left\| h^q + \rho^q \right\|^2 - \eta \langle h^q + \rho^q, h^q - \rho^q \rangle + \eta^2 \left\| \mathcal{J}^T (h^q + \rho^q) \right\|_{\mathcal{H}}^2 \leq \\
&-\eta \left[(1 - \eta\kappa) \left\| h^q + \rho^q \right\|^2 + (\left\| h^q \right\|^2 - \left\| \rho^q \right\|^2) \right] = \\
&-\eta \textstyle\sum_{k=1}^p \left[(1 - \eta\kappa) \left\| h_k^q + \rho_k^q \right\|_k^2 + (\left\| h_k^q \right\|_k^2 - \left\| \rho_k^q \right\|_k^2) \right].
\end{aligned}$$

The r.h.s in the above equation is a sum of p nonegative terms:

$$\underbrace{(1 - \eta\kappa) \left\| h_k^q + \rho_k^q \right\|_k^2}_{(1)} + \underbrace{(\left\| h_k^q \right\|_k^2 - \left\| \rho_k^q \right\|_k^2)}_{(2)}$$

In fact, (1) *is clearly nonnegative for $\eta \leq 1/\kappa$, whereas* (2) ≥ 0 *since $\left\| v_k^{q+1} \right\|_k \leq 1$ which implies $\|\rho_k\|_k \leq \|h_k\|_k$ We now examine the case where the $\left\| \mathcal{J}^T v^{q+1} - \frac{g}{\lambda} \right\|_{\mathcal{H}}^2 - \left\| \mathcal{J}^T v^q - \frac{g}{\lambda} \right\|_{\mathcal{H}}^2 = 0$. This requires both* (1) *and* (2) *to be null for all k. When $\eta < 1/\kappa$,* (1) $= 0$ *only if $\left\| h_k^q + \rho_k^q \right\|_k = 0$ which implies both* (2) $= 0$ *and $v_k^{q+1} = v_k^q$. When $\eta = 1/\kappa$,* (1) *is clearly null whereas* (2) $= 0$ *only if $\left\| h_k^q \right\|_k = \left\| \rho_k^q \right\|_k$ for all k which again implies $v_k^{q+1} = v_k^q$. Hence if $\eta \leq \left\| \mathcal{J} \mathcal{J}^T \right\|^{-1}$, either $\left\| \mathcal{J}^T v^q - g/\lambda \right\|_{\mathcal{H}}$ is decreasing or $v^{q+1} = v^q$.*
Let $m = \lim_{n \to \infty} \left\| \mathcal{J}^T v^q - g/\lambda \right\|$, and $\bar{v}$ be the limit of a converging subsequence (v^{q_l}) of (v^q). Clearly we have $m = \left\| \mathcal{J}^T \bar{v} - g/\lambda \right\| = \left\| \mathcal{J}^T \bar{v}' - g/\lambda \right\|$, where $\bar{v}'$ is the limit of (v^{q_l+1}). From the above calculations we see that since $\left\| \mathcal{J}^T \bar{v}' - g/\lambda \right\| - \left\| \mathcal{J}^T \bar{v} - g/\lambda \right\| = 0$, it must be $\bar{v}_k = \bar{v}_k'$ $\forall k$. Hence $\bar{v}$ satisfies the Euler equation (9) and therefore solves (8). Since the projection is unique, we deduce that all the sequence $\lambda \mathcal{J}^T v^q$ converges to $\pi_{\lambda K}(g)$.

We next consider the step-size choice studied in Proposition 3.

Proof 6 (Proposition 3) *In order to apply Proposition 1, it is enough to show that the conditions on the eigenvalues of the second derivative of F are satisfied. Using the same notations as in Proposition 1, and relying on the chain rule (see [30]) we are able to explicitly compute ∇F, that is*

$$\nabla F(\beta) = \sum_{i=1}^n l'(\langle \Phi(x_i), \beta \rangle_{\mathcal{F}}, y_i) \Phi(x_i) + 2\mu\beta.$$

Reasoning as in the previous step, and again relying on the chain rule, we get

$$\nabla^2 F(\beta)(\beta') = \sum_{i=1}^{n} l''(\langle\Phi(x_i),\beta\rangle, y_i)\langle\Phi(x_i),\beta'\rangle\Phi(x_i) + 2\mu I$$

Defining $A_\beta(\beta') := \sum_{i=1}^{n} l''(\langle\Phi(x_i),\beta\rangle, y_i)\langle\Phi(x_i),\beta'\rangle\Phi(x_i)$*, we note that* A_β *is a self-adjoint linear operator. In particular, using the fact that thanks to the convexity of* ℓ $L_{max}a$*,* $L_{min}b$ *are respectively an upper and a lower bound of the eigenvalues of* A_β*, thanks to Corollary 1 we get the desired inequality and the optimal step choice.*

Finally we study the general least squares case. Although it can be viewed as a consequence of Proposition 1, we prefer to derive the desired inequality directly from the definition of $\mathcal{T}_\sigma$.

Proof 7 (Proposition 4)

$$\begin{array}{cl} & \|\mathcal{T}_\sigma(f) - \mathcal{T}_\sigma(f')\| = \\ = & ||(I - \pi_{\frac{\tau}{\sigma}K})(f - \frac{1}{2\sigma}\nabla F(f)) - (I - \pi_{\frac{\tau}{\sigma}K})(f' - \frac{1}{2\sigma}\nabla F(f'))|| \\ \leq & ||f - \frac{1}{2\sigma}\nabla F(f) - f' + \frac{1}{2\sigma}\nabla F(f')|| \\ = & ||I - \frac{1}{\sigma}(A^T A + \mu)||^2 \, \|f - f'\| \\ = & max\left\{\left|1 - \frac{a+\mu}{\sigma}\right|, \left|1 - \frac{b+\mu}{\sigma}\right|\right\} \|f - f'\| \\ =: & L_\sigma \|f - f'\| \, . \end{array}$$

The optimal a priori choice for the step-size is given by the value of σ *minimizing* L_σ*, that is*

$$\sigma = \frac{a + b + 2\mu}{2},$$

and one can simply verify that $L_\sigma = \frac{a-b}{a+b+2\mu}$.

References

[1] A. Argyriou, R. Hauser, C. A. Micchelli, and M. Pontil. A dc-programming algorithm for kernel selection. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006.

[2] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[3] F. R. Bach, G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, volume 69 of *ACM International Conference Proceeding Series*, 2004.

[4] H. H. Bauschke. The approximation of fixed points of compositions of nonexpansive mappings in Hilbert space. *J. Math. Anal. Appl.*, 202(1):150–159, 1996.

[5] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Rev.*, 38(3):367–426, 1996.

[6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[7] Emmanuel Candès and Terence Tao. Rejoinder: “The Dantzig selector: statistical estimation when p is much larger than n” [Ann. Statist. **35** (2007), no. 6, 2313–2351; mr2382644]. *Ann. Statist.*, 35(6):2392–2404, 2007.

[8] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

[9] G. H.-G. Chen and R. T. Rockafellar. Convergence rates in forward-backward splitting. *SIAM J. Optim.*, 7(2):421–444, 1997.

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

[11] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc. (N.S.)*, 39(1):1–49 (electronic), 2002.

[12] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1457, 2004.

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

[14] C. De Mol, E. De Vito, and L. Rosasco. Elastic-net regularization in learning theory, 2009.

[15] C. De Mol, S. Mosci, M. Traskine, and A. Verri. A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16, 2009.

[16] A. L. Dontchev and T. Zolezzi. *Well-posed optimization problems*, volume 1543 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.

[17] I. Drori and D. L. Donoho. Solution of ℓ^1 minimization problems by lars/homotopy methods. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[18] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization.* Ph.D. diss., Cambridge, US, 1989. CICS-TH-140.

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

[20] I. Ekeland and R. Temam. *Convex analysis and variational problems.* North-Holland Publishing Co., Amsterdam, 1976.

[21] T. Evgeniou, M. Pontil, and T. Poggio. Statistical learning theory: A primer. *Int. J. Comput. Vision*, 38(1):9–13, 2000.

[22] M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12(8):906–916, 2003.

[23] 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.

[24] M. Fornasier, I. Daubechies, and I. Loris. Accelerated projected gradient methods for linear inverse problems with sparsity constraints. *J. Fourier Anal. Appl.*, 2008.

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

[26] M. Hein and O. Bousquet. Kernels, associated structures and generalizations. Technical Report 127, Max Planck Institute for Biological Cybernetics, July 2004.

[27] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and Gorinevsky D. A method for large-scale ℓ^1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 4(1):606–617, 2007.

[28] R. A. Kubota and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005.

[29] G. R. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, November 2004.

[30] S. Lang. *Real analysis.* Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, second edition, 1983.

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

[32] I. Loris. On the performance of algorithms for the minimization of l_1-penalized functionals. *Inverse Problems*, 25(3):035008, 16, 2009.

[33] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanni. Accelerating gradient projection methods for $
ell_1$-constrained signal recovery by steplength selection rules, 2009.

[34] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. Mach. Learn. Res.*, 6:1099–1125, 2005.

[35] C. A. Micchelli and M. Pontil. Feature space perspectives for learning the kernel. *Mach. Learn.*, 66(2-3):297–319, 2007.

[36] G. Obozinski, B. Taskar, and M.I. Jordan. Multi-task feature selection. Technical report, Dept. of Statistics, UC Berkeley, June 2006.

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

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

[39] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

[40] S.J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Image Process.*, 2009. forthcoming.

[41] T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

[42] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ^1-minimization with applications to compressed sensing. *SIAM J. Imaging Sciences*, 1(1):143–168, 2008.

[43] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

[44] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, 2008. to appear.

[45] Z. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.