Simple algorithms to solve sparsity based regularization via Fenchel duality

S. Mosci, M. Santoro, A. Verri, and S. Villa
DISI, Università di Genova, Italy

L. Rosasco
Center for Biological and Computational Learning, Massachusetts Institute of Technology
DISI, Università di Genova, Italy

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 sum of a differentiable data term and a convex non differentiable penalty. Non convex penalties has recently become popular since they allow to enforce some kind of sparsity in the solution. Leveraging on the theory of Fenchel duality and subdifferential calculus, we derive optimality conditions for the regularized solution and propose a simple yet general iterative projection algorithm whose convergence to the optimal solution can be proven. The power of the general framework is illustrated, considering several examples of regularization schemes including multi-task and multi-kernel learning.

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. The supervised learning problem amounts to find an unknown functional relation, given a training set of input-output pairs that are randomly sampled and corrupted by noise. Learning schemes which are simply tailored to minimize a data fit objective term, typically lead to unstable solutions that do not generalize to new examples. An effective way to restore stability and find meaningful solutions is to resort to regularization techniques. 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 and are based on convex differentiable penalties.

Recently methods such as the *Lasso* [Tibshirani, 1996] – based on ℓ_1 regularization– and variants like elastic net or *group lasso*, received considerable attention because of their property to provide *sparse* solutions. The key towards sparsity properties is considering convex non differentiable penalties. More generally this kind of penalties have been used to deal with complex model for multi-task and multi-kernel learning.

In this paper we refer to the general class of methods using convex non differentiable penalties as *sparsity based 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 quadratic programming techniques are often used. The contribution of our work is to show that, under fairly mild assumption on the penalty (see below), sparsity based algorithms can be studied within a unifying framework, that allows to develop a simple iterative procedure which is

very easy to implement and converges to the optimal solution. Using Fenchel duality we decouple the contributions due to the data, and the penalty terms: at each iteration the gradient of the data term is projected on a set which is defined by the considered penalty. The iterative soft thresholding method recently proposed to solve the Lasso minimization is a special case of our framework and several other examples can be given.

2 Iterative Projected Algorithm

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

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

where $F, J : \mathcal{H} \to \mathbb{R}$ 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 convex and one-homogeneous,

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

for all $f \in \mathcal{H}$ and $\lambda \in \mathbb{R}^+$. Before presenting our results we give several examples for F and J.
Loss term. In the context of supervised learning, 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.

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

Examples of convex and differentiable loss functions are the exponential loss $e^{-yf(x)}$, the logistic loss $\log(1 + e^{-yf(x)})$, and especially 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 assumption on the data. An alternative way to enforce strict convexity is to add the strictly convex term $\mu \|f\|^2_{\mathcal{H}}$ for some small positive parameter μ. This can be seen as a preconditioning of the problem and – if μ is small – one can see empirically that the solution does not change. Another important example of loss function is $F(f) = \|Af - y\|^2_{\mathcal{Y}}$, where $A : \mathcal{H} \to \mathcal{Y}$ is a bounded linear operator between the Hilbert spaces $\mathcal{H}$ and $\mathcal{Y}$, and $y \in \mathcal{Y}$ is a *measurement* function from which we aim at reconstructing f. This latter choice is general enough to deal with eigen-problems underlying many unsupervised methods such as principal component analysis or spectral clustering.
Penalty term. The assumptions on the penalty– convexity and one-homogeneity– are satisfied by a general class of penalties that are sum of norms in distinct Euclidean spaces:

$$J(f) = \sum_{k=1}^{p} ||\mathcal{J}_k(f)||, \tag{2}$$

where, for all k, $\mathcal{J}_k : \mathcal{H} \to \mathbb{R}^{m_k}$ is a bounded linear operator and $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^{m_k}$. 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 is a special case of the above penalty $J(\beta) = \sum_{j=1}^{p} |\beta_j|$. If the coefficients are divided into "blocks", a penalty of the form (2), corresponding to the sum of the euclidean norm of each block, has been proposed in the so called group lasso and composite absolute penalties algorithms. Similar penalties have been used for multiple task learning (see the following) and sparse principal component analysis. 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 $\sum_{j=1}^{p} \|f_j\|_j$.

The above examples are only a few of the specific instances of problem (1) satisfying the required assumptions but one can see how loosely related learning schemes can be cast within a common general framework. In next section we show how the corresponding optimization problem can be solved using the same simple procedure.

2.1 Fixed Point Equation via Fenchel Duality

In this section we describe an iterative scheme to compute the optimal solution of problem (1). The proposed procedure is summarized in Algorithm 2.1, where K denotes the subdifferential

[Ekeland and Temam, 1976] of J evaluated at zero i.e. $K := \partial J(0)$, and $\pi_{\lambda K} : \mathcal{H} \to \mathcal{H}$ is the projection on $\lambda K \subset \mathcal{H}$, $\lambda \in \mathbb{R}^+$. The parameter σ can be seen as a step-size, whose choice is crucial to ensure convergence.

Algorithm 2.1 General Algorithm

initialize $\sigma, \tau > 0$
set $f^0 = 0$
while `stopping criterion is not met` **do**
$p = p + 1$

$$f^p = \left(I - \pi_{\frac{\tau}{\sigma}K}\right)\left(f^{p-1} - \frac{1}{2\sigma}\nabla F(f^{p-1})\right) \tag{3}$$

end while

As we mentioned before, our method decouples the contributions of the two functionals J and F. At each iteration of the algorithm the projection $\pi_{\lambda K}$ – which is entirely characterized by J – is applied to a term that depends only on F. It is worth noting that this line of reasoning is developed in a systematic way in the so called *forward-backward splitting* methods. In our approach, Fenchel duality ([Ekeland and Temam, 1976]) is the key tool that, combined with one-homogeneity, allows us to characterize the contribution of J.

In the following we briefly describe the key steps toward deriving Algorithm 2.1. The first step is contained in Theorem 1 below 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 $J : \mathcal{H} \to \mathbb{R}$ convex and one-homogeneous, the minimizer f^* of $\mathcal{E}$ is the unique fixed point of the map $\mathcal{T}_\sigma : \mathcal{H} \to \mathcal{H}$*

$$\mathcal{T}_\sigma(f) = \left(I - \pi_{\frac{\tau}{\sigma}K}\right)\left(f - \frac{1}{2\sigma}\nabla F(f)\right). \tag{4}$$

The next step is to show convergence of the successive approximation scheme derived from the fixed point equation. This latter result is a consequence of the Banach Fixed Point Theorem. In fact, if for some $\sigma > 0$ the map $\mathcal{T}_\sigma$ is a contractive map – i.e. there exists $L_\sigma < 1$ such that $\|\mathcal{T}_\sigma(f) - \mathcal{T}_\sigma(f')\| \leq L_\sigma \|f - f'\|$ for all $f, f' \in \mathcal{H}$ – then Banach Fixed point Theorem immediately implies the Algorithm 2.1 converges to f^* and the following inequality describes the speed of convergence:

$$\|f^* - f_p\| \leq \frac{L_\sigma^p}{1 - L_\sigma} \|f_1 - f_0\|.$$

For all the examples of loss functions we previously described, the value of L_σ can be explicitly computed and step size can be chosen as the value minimizing L_σ. Clearly in this case it is possible to get explicit convergence rates and hence a stopping rule for the iterative procedure.
Finally the last step is to show that the projection $\pi_{\lambda K}$ can be effectively computed. When the penalty is of the form 2, it is possible to prove that $\pi_{\lambda K}(g) = \lambda \bar{v}$ with

$$\bar{v} = \operatorname*{argmin}_{v \in \mathcal{H}, \|\mathcal{J}_k v\|_{\mathbb{R}^{m_k}} \leq 1} \|\lambda v - g\|_{\mathcal{H}}^2. \tag{5}$$

Detailed proofs of the above results will appear in a forthcoming longer version of this paper.

3 Examples

In this section we illustrate our results in a few special cases of interest in machine learning: elastic net regularization and multi-task and multiple kernel learning.

3.1 Elastic Net Regularization

Elastic net regularization [Zou and Hastie, 2005] is given by the minimization of the functional:

$$\mathcal{E}^{(\ell_1\ell_2)}(\beta) = \|\Psi\beta - y\|^2 + \mu\sum_{j=1}^{M}\beta_j^2 + 2\tau\sum_{j=1}^{M} w_j|\beta_j|, \tag{6}$$

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 set of positive weights. The matrix Ψ can be thought as given by the features ψ_j in the dictionary evaluated at some point $x_1, \ldots, x_n$. The minimization of the above functional reduces to the Lasso algorithm [Tibshirani, 1996] if $\mu = 0$.
If we set $m_k = 1$ and $\mathcal{J}_k\beta = w_k\beta_k \ \forall k = 1, \ldots, M$ (see equation 2) then the projection 5 can be directly computed from the regression coefficients β_j, and the iteration 3 becomes:

$$\beta^p = \mathbf{S}_{\frac{\tau}{\sigma}}((I - \frac{\mu}{\sigma})\beta^{p-1} + \frac{1}{\sigma}\Psi^T(y - \Psi\beta^{p-1})), \tag{7}$$

where $\mathbf{S}_{\frac{\tau}{\sigma}}$ is the iterated soft-thresholding operator defined component-wise as $(\mathbf{S}_{\frac{\tau}{\sigma}}(\beta))_j = \text{sign}(\beta_j)(\beta_j - \frac{\tau}{\sigma})_+$ [Daubechies et al., 2004]. The above results has been derived with a different approach in [De Mol et al., 2008].

3.2 Sparse multitask regularization

Learning multiple tasks simultaneously has been shown to improve performance relative to learning each task independently, when the tasks are related. Given T tasks modelled as $f_t(x) = \sum_{j=1}^{d}\beta_{j,t}\psi_j(x)$ for $t = 1, \ldots, T$, we consider the following minimization problem [Obozinski et al., 2006]

$$\mathcal{E}^{(MT)}(\beta) = \sum_{t=1}^{T}\sum_{i=1}^{n_t}(\psi(x_{t,i})\beta_t - y_{t,i})^2 + 2\tau\sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T}\beta_{t,j}^2}. \tag{8}$$

If we let,

$$\beta = (\beta_1^T, \ldots, \beta_T^T)^T$$
$$\Psi = \text{diag}(\Psi_1, \ldots, \Psi_T), \quad [\Psi_t]_{ij} = \psi_j(x_i).$$

equation 3 in algorithm 2.1 is replaced by task-wise soft thresholding

$$\beta^p = \mathbf{S}_{\frac{\tau}{\sigma}}(\beta^{p-1} + \frac{1}{\sigma}\Psi^T(y - \Psi\beta^{p-1})).$$

where the $\mathbf{S}_{\frac{\tau}{\sigma}}$ is the iterative soft-thresholding operator defined in the previous subsection.

3.3 Multiple kernel learning

Multiple kernel learning is the process of finding an optimal kernel from a prescribed convex set, $\mathcal{K}$, of basis kernels, k_j, for learning a real-valued function by regularization. When the set $\mathcal{K}$ is the convex hull of a finite number of kernels $k_1, \ldots, k_M$, multiple kernel learning [Micchelli and Pontil, 2005] amounts to considering

$$\mathcal{E}^{(MK)}(f) = \sum_{i=1}^{n}\left(\sum_{j=1}^{p} f_j(x_i) - y_i\right)^2 + 2\tau\sum_{j=1}^{p}\|f_j\|_{\mathcal{H}_j} \tag{9}$$

where $f = f_1 + \cdots + f_p$ and every f_j belongs to a specific RKHS $\mathcal{H}_j$ with kernel K_j and norm $\|\cdot\|_j$. A form of the representer theorem shows that the solution to the above problem can be expressed as

$$f^*(\cdot) = \left(\sum_{i=1}^{n}\alpha_{1,i}k(x_i, \cdot), \ldots, \sum_{i=1}^{n}\alpha_{p,i}k(x_i, \cdot)\right),$$

so that the optimization problem is finite dimensional. Introducing the notation

$\alpha = (\alpha_1, \ldots, \alpha_p)$ with $\alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,n})$,
$\mathbf{k}(x) = (\mathbf{k}_1(x), \ldots, \mathbf{k}_M(x))^T$ with $\mathbf{k}_j(x) = (k_j(x_1, x), \ldots, k_j(x_n, x))$,
$\mathrm{K} = (\mathrm{K}_1, \ldots, \mathrm{K}_p)$ with $[\mathrm{K}_j]_{ii'} = k_j(x_i, x_i')$.

one can write the optimal solution as $f^*(x) = (\alpha_1^T \mathbf{k}_1(x), \ldots, \alpha_p^T \mathbf{k}_p(x))$. With little work one can see that, for functions of this kind, equation 3 becomes:

$$\mathcal{T}_\sigma(f) = \left(I - \pi_{\tau/\sigma K}\right) \left(\left(\alpha - \frac{1}{\sigma n} \mathrm{K}^T \left(\mathrm{K}\alpha - y \right) \right)^T \mathbf{k} \right). \tag{10}$$

In this case, the projection $\pi_{\tau/\sigma K}$ can be computed block-wise, across all the regression coefficients relative to the same f_j as:

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

where $g = (\alpha_1^T \mathbf{k}_1, \ldots, \alpha_p^T \mathbf{k}_p)$ as in (10).

4 Conclusions

In this paper we show that a large class of regularization schemes using non differentiable penalties can be solved using an iterative projection algorithm. The proposed procedure is extremely simple and converges to the optimal solution. The main operations involved in each iteration are matrix vector multiplications that can be performed extremely fast in many situation. The general results are illustrated for multi-task and multi-kernel learning.

Future work is aimed at further refining the iterative procedure to obtain faster algorithms. Indeed, one can consider data dependent choice of the step size or continuation methods to reduce the number of needed iteration, at the price of a considerably more involved analysis.

References

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

[De Mol et al., 2008] De Mol, C., De Vito, E., and Rosasco, L. (2008). Elastic net regularization in learning theory. *To be published in the Journal of Complexity*.

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

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

[Obozinski et al., 2006] Obozinski, G., Taskar, B., and Jordan, M. (2006). Multi-task feature selection. Technical report, Dept. of Statistics, UC Berkeley.

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

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