A Two-term Penalty Function for Inverse Problems with Sparsity Constrains

Paul Rodriguez
Department of Electrical Engineering
Pontificia Universidad Católica del Perú
Lima, Peru
Email: prodrig@pucp.edu.pe

***Abstract*—Inverse problems with sparsity constrains, such Basis Pursuit denoising (BPDN) and Convolutional BPDN (CBPDN), usually use the ℓ_1-norm as the penalty function; however such choice leads to a solution that is biased towards zero. Recently, several works have proposed and assessed the properties of other non-standard penalty functions (most of them non-convex), which avoid the above mentioned drawback and at the same time are intended to induce sparsity more strongly than the ℓ_1-norm.**

In this paper we propose a two-term penalty function consisting of a synthesis between the ℓ_1-norm and the penalty function associated with the Non-Negative Garrote (NNG) thresholding rule. Although the proposed two-term penalty function is non-convex, the total cost function for the BPDN / CBPDN problems is still convex. The performance of the proposed two-term penalty function is compared with other reported choices for practical denoising, deconvolution and convolutional sparse coding (CSC) problems within the BPDN / CBPDN frameworks. Our experimental results show that the proposed two-term penalty function is particularly effective (better reconstruction with sparser solutions) for the CSC problem while attaining competitive performance for the denoising and deconvolution problems.

I. Introduction

A sparse representation is an adaptive signal decomposition consisting of a linear combination of atoms from an overcomplete dictionary, where the coefficients of the linear combination are optimized according to some sparsity criterion. One of the most well-known methods for computing such a sparse representation is Basis Pursuit Denoising (BPDN) [1] which consists of the minimization

$$\arg\min_{\{\mathbf{u}\}} \;\; \frac{1}{2}\left\|\Phi\mathbf{u}-\mathbf{b}\right\|_2^2 + p(\mathbf{u},\lambda) \qquad (1)$$

where $\mathbf{b}$ is the input data, an $N_1 \times N_2$ image in the present context, Φ represents the overcomplete dictionary, $\mathbf{u}$ is the sparse representation, and $p(.)$ is the penalty function, which usually is the ℓ_1-norm, i.e. $p(\mathbf{u},\lambda) = \lambda \cdot \|\mathbf{u}\|_1$.

Moreover, a convolutional sparse representation (CSR) [2], [3] models an entire image as a sum over a set of convolutions of coefficient maps, of the same size as the image, with their corresponding dictionary filters. Given a set of dictionary filters, the most widely used formulation of the convolutional sparse coding (CSC) problem is Convolutional BPDN (CBPDN) [4], defined as

$$\arg\min_{\{\mathbf{u}_k\}} \;\; \frac{1}{2}\left\|\sum_{k=1}^{K} H_k * \mathbf{u}_k - \mathbf{b}\right\|_2^2 + \sum_{k=1}^{K} p(\mathbf{u}_k,\lambda) \qquad (2)$$

where $\{H_k\}$ is a set of K (usually but not necessarily) non-separable $L_1 \times L_2$ filters and $\{\mathbf{u}_k\}$ is the corresponding set coefficient maps (each with $N = N_1 \times N_2$ samples); $\mathbf{b}$ and $p(.)$ are the same as for (1).

Recently, several works [5], [6], [7], [8] have proposed or assessed the use of different penalty functions for (1); the list includes the Logarithmic and Arctangent penalty functions [6], as well as those associated with the Non-negative Garrote (NNG) [9], SCAD [10] and Firm [11] thresholding rules. The key sought-after property for these alternatives is to induce sparsity more strongly than the ℓ_1-norm penalty function. Due to the similarities (see discussion in Section II-B) between (1) and (2), at a high level, known theoretical results for (1) do apply to (2), however computational results (reconstruction quality, sparsity) do not necessarily follow the same trend for (1) and (2) (see results in Section IV).

In this paper we propose a parametric two-term penalty function (originally reported in [12]), defined as

$$p(\mathbf{x},\lambda,\alpha,\beta) = \lambda\cdot\alpha\cdot\|\mathbf{x}\|_1 + p_{\mathrm{nng}}(\mathbf{x},\lambda\cdot\beta), \qquad (3)$$

with $\alpha > 0$, $\beta > 0$, and $p_{\mathrm{nng}}(\mathbf{x},\gamma)$ is the penalty function associated with the Non-Negative Garrote (NNG) thresholding rule (see (12)-(13)).

II. Previous related work

We start this section with a summary of the properties of several non ℓ_1-norm penalty functions, to then continue with a brief review of the well-known FISTA and ADMM algorithms, which we use to solve (1)-(2) and carry out the computational experiments reported in Section IV; finally we outline known results and conditions for which the FISTA and ADMM algorithms, considering a non-convex penalty function, converge to the global minimum.

A. Non ℓ_1-norm penalty functions

In [5] the Firm thresholding rule was used to solve problem (1) via a modified version of the ISTA and FISTA algorithms; computational results showed that a larger regularization parameter (λ in (1)) can be used, when compared to the case of

ISBN 978-0-9928626-7-1

the soft-thresholding rule (associated to the ℓ_1-norm penalty function, see (11)), thus leading to a significant reduction of the total number of iterations.

[6] assessed the convergence properties of the ISTA algorithm for any thresholding rule[1] and presented numerical examples comparing the ℓ_0, ℓ_1 norms with the NNG, SCAD and Firm thresholding rules when solving problem (1) applied to denoising and inpaiting of images. One of the main conclusions of [6] is that the best thresholding rule (reconstruction quality point of view) is problem dependent. It is also mentioned that NNG, SCAD and the Firm thresholding rules have very similar performance.

The Logarithmic and Arctangent (parametric) penalty functions were introduced in [7]; it also focused on the conditions to be met by non-convex penalty functions so as to ensure the convexity of the total cost function of (1). While these penalty functions exhibit less bias (than the ℓ_1-norm) they are particularly convenient in algorithms for solving (1) that do not used their corresponding thresholding rules directly but the derivative of their penalty functions, such IRLS [14], FOCUSS [15] and (majorization-minimization) MM-based [16].

[8] provided a general framework to show the convergence of the ISTA algorithm when solving (1) along with a non-convex penalty function. Experimentally, it was shown that the use of large step sizes in ISTA can accelerate its convergence when used along the Firm thresholding rule; furthermore, a novel weakly convex penalty function, designed to promote integer values (see [8, Section V]), was also proposed.

B. Numerical algorithms for (1) and (2)

In a generic fashion, either (1) or (2) can be written as

$$\underset{\{\mathbf{x}\}}{\arg\min} \quad f(\mathbf{x}) + p(\mathbf{x}, \lambda). \tag{4}$$

Furthermore, for both cases we can also consider that $f(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$; in the case of (1) the relationship is direct, while for (2) we have that $\mathbf{x} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K]$ and $A\mathbf{x} = \sum_{k=1}^{K} H_k * \mathbf{u}_k$. Using this notation, for either (1) or (2) $\nabla f(\mathbf{u}) = A^T(A\mathbf{x} - \mathbf{b})$.

1) FISTA: The Fast Iterative Shrinkage Thresholding algorithm (FISTA) [17] is closely related to ISTA, but has faster convergence. For convenience the computational steps of FISTA are reproduced in Algorithm 1. Note that L in this algorithm can be updated for each iteration by following a backtracking step size rule [17, Section 4].

It is worth mentioning that ISTA/FISTA algorithms have been previously used to solve (2): [18] and [19] proposed an ISTA and FISTA based algorithms respectively, computing the necessary gradient in the spatial domain, assuming a non-separable filter bank (FB) $\{H_k\}$, while [12] proposed a FISTA based algorithm, assuming a separable FB.

2) ADMM: The ADMM algorithm [20] is a well-known and versatile method used to solve an optimization problem of the form $\min_{\mathbf{u},\mathbf{v}} f(\mathbf{x}) + p(\mathbf{y}, \lambda)$ s.t. $F\mathbf{x} + G\mathbf{y} - \mathbf{c} = 0$.

Inputs: λ (parameter), L (Lipschitz constant of $\nabla f(\mathbf{x})$)
Step 0: Set $\mathbf{y}_1 = \mathbf{x}_0$ (initial guess), $\beta_1 = 1$
Step n: $(n \geq 1)$ Compute
1 $\mathbf{x}^n = \mathrm{thresh}(\mathbf{x}^{n-1} + \frac{1}{L}A^T(\mathbf{b} - A\mathbf{y}^n), \frac{\lambda}{L})$
2 $\beta_{n+1} = \frac{1+\sqrt{1+4\beta_n}}{2}$
3 $\mathbf{y}^{n+1} = \mathbf{x}^n + \frac{\beta_n - 1}{\beta_{n+1}}(\mathbf{x}^n - \mathbf{x}^{n-1})$

Algorithm 1: FISTA applied to (4) for which $f(\mathbf{u}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$. $\mathrm{thresh}(\cdot)$, in line 1, is the corresponding thresholding rule[3] for the particular penalty function in (4).

The ADMM iterations with scaled dual variable are given by (5)-(7) and can be readily applied to problem (1) as well as to problem (2). For the latter case, see [4, Section 3]; moreover, recent ADMM-based approaches compute the convolutions in the frequency domain [21], [22], [23], [4], [24].

$$\mathbf{x}^{(n+1)} = \min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\|F\mathbf{x} + G\mathbf{y}^{(n)} - \mathbf{c} + \mathbf{z}^{(n)}\|_2^2 \tag{5}$$

$$\mathbf{y}^{(n+1)} = \min_{\mathbf{y}} p(\mathbf{y}, \lambda) + \frac{\rho}{2}\|F\mathbf{x}^{(n+1)} + G\mathbf{y} - \mathbf{c} + \mathbf{z}^{(n)}\|_2^2 \tag{6}$$

$$\mathbf{z}^{(n+1)} = \mathbf{z}^{(n)} + F\mathbf{x}^{(n+1)} + G\mathbf{y}^{(n+1)} - \mathbf{c}\,. \tag{7}$$

It is worth noting that, when G is the identity[2], the solution to (6) is given by (8), where $\mathrm{thresh}(\cdot)$ is the corresponding thresholding rule[3] for the selected penalty function.

$$\mathbf{y}^{n+1} = \mathrm{thresh}(F\mathbf{x}^{(n+1)} + \mathbf{z}^{(n)}, \frac{\lambda}{\rho}), \tag{8}$$

C. Convergence under non-convex penalty functions

In this section we provide a summary of the results presented in [6], [8] and [25], [26], which focused on the convergence of the ISTA and ADMM algorithms respectively, when solving (1) for non-convex penalty functions.

Definition II.1. *A function $p : \Re^N \longrightarrow \Re$ is said to be c-semi convex (see [6, Definition 1]) or c-weakly convex (see [8, Definition 1]) if $g(\mathbf{x}) = \frac{\tau}{2}\|\mathbf{x}\|_2^2 + p(\mathbf{x}, \lambda)$ is convex when $\tau \geq c \geq 0$.*

On what follows we use the term "c-semi convex" to indistinctly refer to either "c-semi convex" or "c-weakly convex".

If the penalty function $p(.)$, for problem (1), is c-semi convex, then

- It can be shown that the cost function defined by problem (1) is convex.
- The ISTA algorithm converges to the global minimum of (1), see [6, Theorem 2]. A similar result was presented in [8, Proposition 4].

Furthermore, in [25] an empirical study was carried out to assess the practical performance of ADMM algorithm on several nonconvex applications, including the case when the penalty function $p(.)$ in (1), applied to image denoising, is the ℓ_0-norm. For the above mentioned case, the ADMM algorithm usually converged; moreover, it was noted that the choice of the ρ parameter (see (5)-(6)) greatly affects the convergence

[1] Given a thresholding rule, a corresponding penalty function $p(\cdot)$ can always be computed (see [13], [6], [7]).

[2] This is the case for the problems consider in this work.

[3] If $p(\mathbf{x}, \gamma) = \gamma \cdot \|\mathbf{x}\|_1$, then $\mathrm{thresh}(.)$ is soft thresholding, defined in (11).

rate and quality of the denoised image and thus even simple adaptive schemes, such the ones described in [20, Section 3.4], should be considered.

In a more general context, [26] proved the convergence of the ADMM algorithm for several nonconvex penalty functions. For such purpose, the restricted prox-regularity (see [26, Definition 2]) was defined; this definition is related to c-semi convex (see Definition II.1), although instead of enforcing a global condition it only requires to be hold over a subset, resulting in a weaker condition and thus implicitly implying that any c-semi convex penalty function, when solving (1) via the ADMM algorithm, do converge to the global minimum.

III. Proposed two-term penalty function

In this Section we describe the proposed two-term penalty function defined in (3), reproduced below for convenience:

$$p_{\mathrm{mix}}(\mathbf{x},\lambda,\alpha,\beta) = \lambda\cdot\alpha\cdot\|\mathbf{x}\|_1 + p_{\mathrm{nng}}(\mathbf{x},\lambda\cdot\beta), \tag{9}$$

where $\alpha > 0$, $\beta > 0$, and $p_{\mathrm{nng}}(\mathbf{x},\gamma)$ is defined in (13). We also analyze the conditions for which problems (1)-(2) with (9) as the penalty function converge to the global minimum via either the FISTA or ADMM algorithms.

Since (9) can be understood as synthesis of the ℓ-1 norm and the penalty function associated with the NNG thresholding rule, we start with a brief summary of both.

The solution to optimization problem

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{y}\| + p(\mathbf{x},\lambda), \tag{10}$$

usually referred as the proximity operator[4] for $p(.)$, is given by[5]

$$\mathbf{x}^* = \mathrm{shrink}(\mathbf{y},\lambda) = \mathrm{sign}(\mathbf{y})\cdot\max\{0,|\mathbf{y}|-\lambda\} \tag{11}$$

when $p(\mathbf{x},\lambda) = \lambda\cdot\|\mathbf{x}\|_1$, and

$$\mathbf{x}^* = \mathrm{thresh}_{\mathrm{nng}}(\mathbf{y},\lambda) = \mathbf{y}\cdot\max\{0, 1-\frac{\lambda^2}{\mathbf{y}^2}\} \tag{12}$$

when

$$p(\mathbf{x},\lambda) = \lambda^2\cdot\left(\mathrm{asinh}\left(\frac{|\mathbf{x}|}{2\lambda}\right) + \frac{|\mathbf{x}|}{\sqrt{\mathbf{x}^2+\lambda^2}+|\mathbf{x}|}\right) \tag{13}$$

which is the penalty function associated with the NNG thresholding rule (12). It is worth mentioning that (12) is c-semi convex (see Definition II.1) with constant $c = \frac{1}{2}$.

Proposition III.1. *The penalty function (9) is c-semi convex with constant* $c = \frac{1}{2}$.

Proof. Using Definition II.1, we have that $g(\mathbf{x}) = \frac{\tau}{2}\|\mathbf{x}\|_2^2 + p_{\mathrm{mix}}(\mathbf{x},\lambda,\alpha,\beta)$, where $p_{\mathrm{mix}}(.)$ is defined in (9). Observing that each component of $g(\cdot)$ is independent[6] then we only need to consider the scalar version $g(x) = \frac{\tau}{2}x^2 + p_{\mathrm{mix}}(x,\lambda,\alpha,\beta)$ in order to check for convexity.

Taking $\lambda_\alpha = \lambda\cdot\alpha$ and $\lambda_\beta = \lambda\cdot\beta$, the first and second derivative for $g(.)$ are given by

$$g'(x) = \tau x + \lambda_\alpha\,\mathrm{sign}(x) + \frac{2\lambda_\beta}{\sqrt{x^2+4\lambda_\beta^2}+|x|}\,\mathrm{sign}(x) \tag{14}$$

$$g''(x) = \tau + \lambda_\alpha\delta(x) + \frac{2\lambda_\beta^2\delta(x)}{\sqrt{x^2+4\lambda_\beta^2}+|x|} - \frac{2\lambda_\beta^2\,\mathrm{sign}(x)\left(\mathrm{sign}(x)+\frac{x}{4\lambda_\beta^2+x^2}\right)}{\left(\sqrt{x^2+4\lambda_\beta^2}+|x|\right)^2} \tag{15}$$

Noting that $g''(0) = \tau+\lambda_\alpha+\lambda_\beta-\frac{1}{2}$, $g''(0^+) = \tau-\frac{1}{2}$ and that $g''(.)$ is strictly decreasing and symmetric, then $g''(x) > 0\,\forall x$ if $\tau \geq \frac{1}{2}$, and thus $g(x) = \frac{\tau}{2}x^2 + p_{\mathrm{mix}}(x,\lambda,\alpha,\beta)$ is c-semi convex with constant $c = \frac{1}{2}$. □

Given the result of Proposition III.1 then when (9) is used as the penalty function for either (1)-(2), such problems are convex and have a unique minimizer. Furthermore, the thresholding rule[7] associated with penalty (9) is given by

$$\mathrm{thresh}(x,\lambda,\alpha,\beta) = \begin{cases}\gamma(x,\lambda,\alpha,\beta) & \text{if } |x| > \lambda\cdot(\alpha+\beta)\\ 0 & \text{otherwise}\end{cases} \tag{16}$$

where $\gamma(x,\lambda,\alpha,\beta) = \frac{x\left(x-\mathrm{sign}(x)\frac{2}{\lambda\cdot\alpha}\right)+\lambda^2\cdot\alpha^2-\lambda^2\cdot\beta^2}{x+\lambda\cdot\alpha}$.

When the constraint $\alpha+\beta = 1$ is included, then (16) is in fact a synthesis between soft-thresholding and NNG, parametrized by the constants α,β [8]. When (1)-(2) are solved via the FISTA or ADMM algorithms, experimentally we have observed that it is convenient to vary, at each main iteration, the values of α and β, starting with α close to 1, and then slowly decreasing it. The computational results in Section IV indicate that this simple strategy is very effective.

IV. Results

All related experiments were carried on an Intel i7-4710HQ (2.5 GHz, 6MB Cache, 32GB RAM) based laptop with a nvidia GTX980M GPU card. Images from the USC-SIPI database [27], rescaled to the range $[0,1]$, were used as test images for all cases.

We solve either (1) or (2) via the FISTA and the ADMM algorithms; for the former, we developed a Matlab library (GPU-enabled for the specific case of (2)), which has been made publicly available [28]; for the latter, we make use of the SPORCO[9] library [29]. It is worth mentioning that our FISTA implementation does include a backtracking step subroutine (see Section II-B1), that greatly improves its convergence rate. whereas the ADMM implementation includes the adaptive schemes described in [20, Section 3.4], which as mentioned

[4] $\mathrm{prox}_{p,\lambda} = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\| + p(\mathbf{x},\lambda)$

[5] Scalar operations applied to a vector are considered to apply element-wise.

[6] $g(\mathbf{x}) = \frac{\tau}{2}\sum_k x_k^2 + \lambda\cdot\alpha\sum_k |x_k| + (\lambda\beta)^2\cdot\sum_k\left(\mathrm{asinh}\left(\frac{|x_k|}{2\lambda\beta}\right) + \frac{|x_k|}{\sqrt{x_k^2+(\lambda\beta)^2}+|x_k|}\right)$

[7] Can be directly computed by setting (14) equal to zero.

[8] For $\alpha = 1, \beta = 0$, (16) is equivalent to soft-thresholding, i.e. (11), whereas for $\alpha = 0, \beta = 1$ (16) is equivalent to NNG, i.e. (12).

[9] We have made some minor variations to this library; patches are available at [28].

ISBN 978-0-9928626-7-1 © EURASIP 2017

in Section II-C, greatly improve the ADMM convergence rate when solving (1) along with a nonconvex penalty functions.

For all related experiments we consider[10] the ℓ_1-norm, NNG, proposed (see (9)) and Logarithmic penalty functions, labeled "L1", "NNG", "Mix" and "Log" respectively, and present reconstruction metrics (SNR, PSNR, MSE and SSIM [30]) and and a sparsity measure, defined as $100 \cdot \frac{\|\mathbf{u}\|_0}{N}$ where $\mathbf{u}$ represents the solution to (1) or (2) and N is the number of pixels in the input image. Furthermore, for the denoising and deconvolution cases we also report reconstruction metrics for the Total Variation (TV) [31] and Expected Patch Log Likelihood (EPLL) [32] methods, which are used as baselines.

A. Denoising and deconvolution

For the denoising and deconvolution cases we consider that the observed image $\mathbf{b}$ is given by

$$\mathbf{b} = H * \mathbf{v} + \boldsymbol{\eta}, \tag{17}$$

where $\mathbf{v}$ represents the original image, H is a blurring kernel and $\boldsymbol{\eta}$ is Gaussian noise with standard deviation σ^2. For the denoising case $H = I$ whereas for the deconvolution case H is a 13×13 sample Gaussian filter with unit standard deviation.

Method		SNR	PSNR	SSIM	MSE	Sp (%)	Cost
ADMM	L1	17.29	30.54	0.82	7e-4	15.8	3.86e2
	NNG	16.28	29.53	0.80	8e-4	17.4	4.22e2
	Mix	17.15	30.40	0.80	7e-4	17.2	3.75e2
	Log	15.74	28.99	0.81	9e-4	15.7	4.17e2
FISTA	L1	17.44	30.69	0.83	6e-4	13.9	3.82e2
	NNG	17.47	30.72	0.83	6e-4	10.7	4.02e2
	Mix	17.44	30.69	0.83	6e-4	14.0	3.80e2
	Log	15.80	29.05	0.81	9e-4	13.9	4.13e2
TV		18.33	31.59	0.87	5e-4	–	–
EPLL		19.99	33.24	0.89	4e-4	–	–

TABLE I

DENOISING RESULTS FOR BPDN WITH DIFFERENT PENALTY COST FUNCTIONS WITH $\lambda = 0.075$, WHEN APPLIED TO THE LENA TEST IMAGE, CORRUPTED WITH $\sigma = 0.05$; THE OBSERVED IMAGE (SEE (17)) HAS SNR = 11.49, PSNR = 24.74, SSIM = 0.36 AND MSE = $2.5e$-3. THE TV AND EPLL RESULTS ARE INCLUDED AS BASELINES.

We use the biorthogonal 6.8 wavelet as dictionary Φ for the BPDN problem[11] defined in (1), considering 2 and 3 decomposition levels for the denoising and deconvolution cases respectively. The ADMM and FISTA algorithm are run until an exit condition is reached or the maximum number of global iterations are reached (60 and 30 for the denoising and deconvolution cases respectively).

In Tables I and II we list the final values of the collected statistics for the denoising and deconvolution cases respectively, with noise level of $\sigma = \{0.05\}$[12]. The performance of the "L1", "Mix" and "NNG" penalty functions are very similar ("L1" and "Mix" slightly outperform the "NNG") and significantly better than the "Log" penalty function. These results are representative for the majority of images in the USC-SIPI database [27].

Method		SNR	PSNR	SSIM	MSE	Sp (%)	Cost
ADMM	L1	13.76	27.01	0.76	1.5e-3	4.33	3.36e+2
	NNG	13.05	26.30	0.67	1.7e-3	6.38	3.17e+2
	Mix	13.67	26.92	0.73	1.5e-9	5.33	3.28e+2
	Log	12.41	25.66	0.73	2.0e-3	4.66	3.48e+2
FISTA	L1	13.78	27.03	0.76	1.5e-3	4.09	3.39e+2
	NNG	13.67	26.92	0.74	1.5e-3	4.31	3.40e+2
	Mix	13.78	27.03	0.76	1.5e-3	4.12	3.39e+2
	Log	12.38	25.63	0.72	2.0e-3	4.35	3.47e+2
TV		14.14	27.39	0.78	1.4e-3	–	–
EPLL		14.39	27.64	0.79	1.3e-3	–	–

TABLE II

DECONVOLUTION RESULTS FOR BPDN WITH DIFFERENT PENALTY COST FUNCTIONS WITH $\lambda = 0.015$, WHEN APPLIED TO THE LENA TEST IMAGE, CORRUPTED WITH $\sigma = 0.05$; THE OBSERVED IMAGE (SEE (17)) HAS SNR = 9.49, PSNR = 22.74, SSIM = 0.19 AND MSE = $4.0e - 3$. THE TV AND EPLL RESULTS ARE INCLUDED AS BASELINES.

B. Convolutional sparse coding

For the experiments in this Section, we use the dictionary filter bank (FB) $\{H_k\}$ consisting of 144 non-separable filters of size 12×12 distributed with the SPORCO library [29].

It worth noting that there are some differences between the ADMM and FISTA implementations use to solve (2). The former, which is distributed with the SPORCO library, computes the convolutions in the frequency domain in order to improve its computational performance (see [4] for details). The latter implementation computes the convolutions in the spatial domain, by first pre-computing a separable approximation of the original non-separable FB (see [12] for details). Both algorithms are run until a given maximum number of global iterations is reached, independently chosen for the ADMM (130 iterations) and FISTA (200 iterations) as to give the best results (reconstruction quality and sparsity).

In Table III we list the final values of the collected statistics for the CSC problem. We first highlight the different performance of the "L1" and "Mix" penalty functions, for either the ADMM or FISTA implementations: the latter gives superior results from reconstruction quality as well as from sparsity point of view than the former. These results are representative for the majority of images in the USC-SIPI database [27].

Moreover, it also worth noting that the trend observed for the denoising and deconvolution problems (see Tables I and II in Section IV-A), i.e. "L1", "Mix" and "NNG" having a similar performance while at the same time having better performance than "Log", is not same for the CSC problem. For the CSC, the use of the "Log" penalty function gives slightly inferior results than either the "L1" or "Mix", whereas results for the "NNG" case, the results are mixed: (i) via the ADMM algorithm, "NNG" gives rather poor reconstruction results, albeit very

[10]The SCAD and Firm thresholding rules are not considered since in [6] it is reported that they have similar performance than NNG.

[11]Due to the characteristics of Wavelets, the problem to be solved is slightly different than (1), and it is given by $\arg\min_{\{\mathbf{u}\}} \frac{1}{2}\|H * (\Phi\mathbf{u}) - \mathbf{b}\|_2^2 + p(W\mathbf{u}, \lambda)$, where W is a diagonal matrix of zeros and ones, which masks out the approximation coefficients of the Wavelet transform.

[12]Results for other σ values are not included due to space limitations, but can be obtained via our publicly available [28] code.

ISBN 978-0-9928626-7-1 © EURASIP 2017

sparse; (ii) via the FISTA algorithm, "NNG" gives very high reconstruction metrics, however the resulting sparsity is very high (about 1.5 times as much when compared to "L1" or "Mix"). Overall, these results also hint a confirmation of the conclusions presented in [6]: the best thresholding rule is problem dependent.

Method		SNR	PSNR	SSIM	MSE	Sp (%)	Cost
ADMM	L1	32.68	45.92	0.99	1.9e-5	31.58	17.52
	NNG	22.02	35.25	0.94	2.2e-4	14.48	41.59
	Mix	33.28	46.51	0.99	1.6e-5	21.99	19.41
	Log	32.10	45.33	0.98	2.1e-5	55.96	18.81
FISTA	L1	29.75	42.98	0.98	3.7e-5	35.76	23.01
	NNG	40.75	54.00	0.99	0.3e-5	55.14	42.65
	Mix	34.84	48.09	0.99	1.2e-5	39.33	25.79
	Log	29.13	42.37	0.97	4.3e-5	61.76	24.54

TABLE III
CSC RESULTS FOR SOLVING CBPDN VIA THE ADMM AND FISTA ALGORITHMS, WITH DIFFERENT PENALTY COST FUNCTIONS FOR $\lambda = 0.01$, WHEN APPLIED TO THE ORIGINAL LENA TEST IMAGE.

V. CONCLUSION

We have proposed a two-term penalty function and evaluated its performance for denoising and deconvolution problems, under the BPDN framework, and convolutional sparse coding (CSC) problem, under the Convolutional BPDN framework.

Our experimental results, carried out via the ADMM and FISTA algorithms, show that the proposed two-term penalty function, when compared to alternatives such the ℓ_1-norm, Logarithmic and the penalty function associated with the Non-Negative Garrote thresholding rule, is particularly effective (better reconstruction with sparser solutions) for the CSC problem while attaining competitive performance for the denoising and deconvolution problems.

ACKNOWLEDGMENT

The authors thank Brendt Wohlberg for valuable discussion on the use of the SPORCO library [29].

REFERENCES

[1] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.

[2] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3517–3524.

[3] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus, "Deconvolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2010, pp. 2528–2535.

[4] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE TIP*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[5] S. Voronin and H. Woerdeman, "A new iterative firm-thresholding algorithm for inverse problems with sparsity constraints," *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 151 – 164, 2013.

[6] M. Kowalski, "Thresholding rules and iterative shrinkage/thresholding algorithm: A convergence study," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 4151–4155.

[7] I. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1078–1092, March 2014.

[8] I. Bayram, "On the convergence of the iterative shrinkage/thresholding algorithm with a weakly convex penalty," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1597–1608, March 2016.

[9] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, no. 4, pp. 373–384, Nov. 1995.

[10] J. Fan, "Comments on "wavelets in statistics: A review" by a. antoniadis," *J. of the Italian Statistical Society*, vol. 6, no. 2, p. 131, 1997.

[11] H. Gao and A. Bruce, "Waveshrink with firm shrinkage," *Statistica Sinica*, vol. 7, pp. 855–874, 1997.

[12] G. Silva, J. Quesada, P. Rodriguez, and B. Wohlberg, "Fast convolutional sparse coding with separable filters," March 2017, accepted to *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[13] A. Antoniadis, "Wavelet methods in statistics: some recent developments and their applications," *Statist. Surv.*, vol. 1, pp. 16–55, 2007. [Online]. Available: http://dx.doi.org/10.1214/07-SS014

[14] A. Beaton and J. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, no. 2, pp. 147–185, 1974.

[15] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE TIP*, vol. 45, no. 3, pp. 600–616, Mar 1997.

[16] J. Bioucas-Dias, M. Figueiredo, and J. Oliveira, "Total variation-based image deconvolution: a majorization-minimization approach," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, May 2006, pp. II.861–II.864.

[17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[18] M. Zeiler, G. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 2018–2025.

[19] R. Chalasani, J. C. Principe, and N. Ramakrishnan, "A fast proximal method for convolutional sparse coding," in *International Joint Conference onNeural Networks (IJCNN)*, Aug. 2013, pp. 1–5.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[21] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 391–398.

[22] B. Wohlberg, "Efficient convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 7173–7177.

[23] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 5135–5143.

[24] B. Wohlberg, "Boundary handling for convolutional sparse representations," in *IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 1833–1837.

[25] Z. Xu, S. De, M. Figueiredo, C. Studer, and T. Goldstein, "An empirical study of ADMM for nonconvex problems," *CoRR*, vol. abs/1612.03349, 2016. [Online]. Available: http://arxiv.org/abs/1612.03349

[26] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *CoRR*, vol. abs/1511.06324, 2015.

[27] "USC-SIPI image database," Available from http://sipi.usc.edu/database/.

[28] P. Rodriguez, "Simulations for two-term penalty function for inverse problems with sparsity constrains," Matlab library available from http://sites.google.com/a/istec.net/prodrig/Home/en/pubs/mixthresh, 2017.

[29] B. Wohlberg, "SParse Optimization Research COde (SPORCO)," Software library available from http://purl.org/brendt/software/sporco, 2016.

[30] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[31] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992.

[32] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 479–486.

ISBN 978-0-9928626-7-1 © EURASIP 2017