Adaptive Order Non-Convex $L_p$-norm Regularization in Image Restoration

Manya Afonso and João Miguel Sanches
Institute of Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal
E-mail: mafonso,jms@isr.ist.utl.pt

Abstract. The optimization formulations of sparse signal recovery problems use a regularizer term which encourages the solution to be sparse or piece-wise smooth. Of special interest are the $\ell_p$ norms with $0 \leq p \leq 1$ which penalize large values and encourage the solution of the optimization problem to be sparse. In this work, we propose a new regularizer in which the exponent $p$, $0 \leq p \leq 1$ of the $\ell_p$-norm is a function of the signal element amplitude, leading to a function which is smooth and convex over non-negative values. This formulation adapts the value of the exponent $p$ according to a sigmoid function applied on the signal values, to restrict its values to between 0 and 1. Experiments show that this formulation is more accurate in denoising, than the $\ell_1$ and $\ell_0$ norm regularizers, and is more accurate than the $\ell_1$-norm when used iteratively for deblurring or inpainting.

1. Introduction
Non-convex regularizers such as the $\ell_p$ norms with $0 \leq p < 1$ are of interest in sparse estimation, especially in problems such as compressive sensing because of their ability to recover sparser solutions from fewer measurements than the convex regularizers such as the $\ell_1$ norm [3, 11]. If $x$ is the image to be estimated, and $y$ is a noisy observed image, both in vector representation, a widely used optimization formulation for estimating $x$ is,

$$\min_x \frac{1}{2} \|x - y\|^2 + \frac{\lambda}{2} \phi(x),$$

(1)

where $\phi(.)$ is a non-convex regularizer, and $\lambda > 0$ is the regularization parameter. For the $\ell_p$ norm regularizer, it is defined as $\phi(x) = (\sum_i |x_i|^p)^{1/p}$. Sometimes its $p$th power $\phi(x) = (\ell_p(x))^p = \sum_i |x_i|^p$ is used for mathematical tractability. If $\phi(.)$ is not convex, then eq (1) is not guaranteed to have a global minimum. For the $\ell_0$ norm, the problem is combinatorial, therefore often a relaxation with the convex $\ell_1$ norm is used, which has a closed form minimizer for (1), given by the soft threshold [7]. This is also called the LASSO problem [10].

The iteratively reweighted least squares (IRLS) method [3, 6] solves (1) by expressing a non-convex $\ell_p$ norm regularizer as a weighted quadratic norm, and therefore iteratively solving a least squares problem. A similar approach was used in [11] for a larger set of non-convex regularizers, by expressing (1) as a weighted LASSO problem.

For convex regularizers, a widely used solver is the split augmented Lagrangian and shrinkage algorithm (SALSA) [1], which belongs to the alternating direction method of multipliers (ADMM) family of algorithms, for which convergence conditions have been defined and proven.
More recently, it has been shown empirically that ADMM can still converge even if the regularizer \( \phi(.) \) is non-convex [15, 4]. It was proven in [14] that unlike the more general augmented Lagrangian method, ADMM can converge with a non-convex and possibly non-smooth objective function, as long as the objective can be separated into a smooth component and a possibly non-smooth and non-convex component, and the constraints are linear.

In [13], the authors solve the square root LASSO problem with a weakly convex regularizer which is convex over blocks of variables, and is defined in terms of the difference between the absolute value and a weighted square. In this paper, we follow a similar logic to define a non-convex regularizer which is convex over some intervals of interest. We propose a solver for problem (1) with this regularizer, based on IRLS. Experimental results show that this regularizer leads to a more accurate denoised image than using the \( \ell_1 \) and \( \ell_{0.5} \) norm regularizers, and that it is more accurate when used with SALSA for image deblurring and inpainting.

2. Proposed Approach

We define a new regularizer in which the exponent is not constant, but a function of the absolute amplitude,

\[
\phi(x) = \sum_i |x_i|^{q(|x_i|)},
\]

where the exponent \( q \) is a function of \( |x|, x \in \mathbb{R} \). Since we are interested in values of \( q \) in the interval \([0, 1]\), a natural choice is the sigmoid function \( \sigma(x) = 1/(1 + \exp(-x)) \). This function is used in binary classification [10] because of it can be used as a smooth approximation to a non-linear thresholding, since it approaches 0 if \( x < 0 \) and 1 if \( x > 0 \). In our case, we are working with the absolute value \( |x| \), therefore the threshold must be different from 0. We would also like to penalize different intervals of the dynamic range of the image, differently. We therefore adapt the sigmoid function to the following rule for the exponent \( q \),

\[
q(|x|) = 1/(1 + \exp(-\alpha(\log |x| - K))),
\]

where \( \alpha > 0 \) is a parameter controlling the level of smoothness and shape of the function \( q(.) \), for values of \( \alpha \) close to zero, the function varies slowly and for larger values, it switches between 0 and 1 with a steep slope, as shown in figure 1(a). The parameter \( K \in \mathbb{R} \) is an additional parameter to control the threshold at which the switching occurs, and depends on the minimum value of \( |x| \) below which it is considered negligible. In the limit that \( |x| \) is very small, the term \( \log |x| \) will be negative, leading to \( q(|x|) \) tending to zero. In this case, in our formulation, we use the limit that any real value raised to an exponent tending to 0 tends to 1. By doing this, we have a regularizer which is a smooth function of \( x \) over \( x \in (0, \infty) \). Note that this regularizer is not a norm.

For completeness of notation, we will define \( \alpha \) as one of the parameters of \( q(.) \), leading to

\[
q(|x|, \alpha, q_0) = \frac{q_0}{1 + \exp(-\alpha(\log |x| - K))},
\]

and \( \phi(x, \alpha, q_0) = \sum_i |x_i|^{q(|x_i|, \alpha, q_0)} \), where \( q_0 \) is the maximum value of the exponent. In this paper, we will use \( q_0 = 1 \) unless stated otherwise.

It was experimentally found that for \( \alpha > 0.5 \), the function (2) is always convex over \( x \geq 0 \), in that its second order gradient is always positive. It can be seen in figures 1(b) and 1(c) that the regularizer \( \phi(x) \) is convex for \( \alpha \) equal to 0.51 and 1, but not for 0.1. By comparison, \( \ell_{0.5} \) is clearly non-convex over the same range of values. The value of \( \alpha = 0.51 \) was also the one which was experimentally found to have the minimum value of \( \phi(x) \) closest to 0, in this case 0.1, as marked by the blue circles in the plots in figure 1(c). To simplify notation, we will use \( \phi(x) \) to indicate \( \phi(x, \alpha = 0.51, q_0 = 1) \) in the rest of this paper.
**Solver for Denoising**

We now solve the denoising problem by using our non-convex regularizer \( \phi(x) \) in (1). Using the differentiation rule for functions of the form \( f(x) = g(x)^h(x) \), the derivative of \( \phi(x) \) with respect to an element \( x_i \) can be shown to be,

\[
\frac{\partial \phi(x)}{\partial x_i} = \left( \frac{\partial q(|x_i|)}{\partial x_i} \log(|x_i|) + \frac{q(|x_i|)}{|x_i|} \right) \text{sign}(x_i) |x_i|^q(|x_i|). \tag{4}
\]

Since the derivative of the sigmoid function is \( d\sigma/dx = \sigma(x)(1 - \sigma(x)) \), it can be shown that

\[
\frac{\partial q(|x_i|)}{\partial x_i} = \frac{\alpha q(|x_i|)}{x_i} \left( 1 - \frac{q(|x_i|)}{q_0} \right). \tag{5}
\]

Using eq (4), the partial derivative of eq (1) wrt \( x_i \) can be expressed as

\[
x_i - y_i + \frac{\lambda}{2} \left( \frac{\partial q(|x_i|)}{\partial x_i} \log(|x_i|) + \frac{q(|x_i|)}{|x_i|} \right) \text{sign}(x_i) |x_i|^q(|x_i|) = 0
\]

\[
x_i - y_i + \frac{\lambda}{2} \left( \frac{\partial q(|x_i|)}{\partial x_i} \log(|x_i|) + \frac{q(|x_i|)}{|x_i|} \right) |x_i|^q(|x_i|)-1 x_i = 0
\]

\[
= (x_i) = a_i \tag{6}
\]

Using an approach similar to IRLS, we calculate the value of \( x_i \) at iteration \( t+1 \) as

\[
x^{(t+1)}_i = \frac{y_i}{1 + 0.5\lambda a^{(t)}_i}, \tag{7}
\]

where \( a_i \) is a function of \( x_i \) as formulated in eq (6), calculated using the value at the previous iteration \( x^{(t)}_i \).

If \( x \) is the wavelet representation of an image, when applying the regularizer \( \phi(.) \), we use the formulation,

\[
\min_x \frac{1}{2} ||x - W^Ty||^2 + \frac{\lambda}{2} \phi(x), \tag{8}
\]

where \( W^T \) is the forward wavelet transform.

The proposed algorithm can be used as a denoising operator [5] in solvers such as SALSA [1] for deconvolution and inpainting problems,

\[
\min_x \frac{1}{2} ||Ax - y||^2 + \frac{\lambda}{2} \phi(W^Tx), \tag{9}
\]

where \( x, y \) represent the pixels, \( A \) is the linear operator representing the convolution or the element-wise binary mask representing partial observation of pixels. The analysis formulation [9] is used to apply the proposed regularizer \( \phi(.) \) on the wavelet representation of \( x \). The variable splitting used in SALSA ensures that at each iteration, the denoising problem of the form (1) is separable.

**3. Results**

The proposed approach was compared for denoising against soft thresholding the wavelet coefficients (4 level, redundant Daubechies 4) [7] and IRLS for denoising with the \( (\ell_{0.5})^{0.5} \) regularizer, on the \( 256 \times 256 \) Lena and Cameraman images with added Gaussian noise (SNR 10 dB). All experiments were performed in Matlab on Windows 7 on an Intel i5 based system with 4 GB of RAM. The stopping criterion for the proposed method was the relative difference
between successive iterations falling below $10^{-3}$. The soft threshold and IRLS were both hand tuned to obtain the best set of parameters which produced the most accurate estimate. The figures of merit used are the Median Absolute Error (MAE), Improvement in Signal to Noise Ratio (ISNR), and the Structured Similarity Index Measure [2].

We can see from table 1 that the proposed approach leads to a more accurate estimate than the soft threshold or IRLS with $(\ell_{0.5})^{0.5}$ regularizer, i.e., it produces a larger ISNR and SSIM and lower MAE. The original and noisy images, along with the estimates obtained with the three methods are shown in figure 2.

For deblurring and inpainting, a few iterations of the proposed denoising solver is used as the denoising operator in SALSA, using the analysis wavelet prior formulation. The proposed approach is compared against SALSA with the $\ell_1$ norm regularizer on the wavelet coefficients. The blur used is $9 \times 9$ uniform blur, and for inpainting, 40% of the pixels are randomly discarded. In both cases, Gaussian noise with SNR 40 dB is added. From table 1, we see that the proposed approach leads to a higher ISNR and SSIM and lower MAE than the $\ell_1$ norm. The observed and estimated images are shown in figures 3 and 4. It can be seen from figures 4(b), 4(c), 4(e), and 4(f) that the proposed approach estimates better regions with edges such as the hair and hat edges in the Lena image, and the silhouette and tripod edges of the cameraman.

Table 1. Comparison of proposed approach with $\ell_1$ and $\ell_{0.5}$ norm regularization, for image denoising, deblurring, and inpainting.

| Problem       | Method          | MAE (dB) | ISNR (dB) | SSIM | Iters | Time (sec) | MAE (dB) | ISNR (dB) | SSIM | Iters | Time (sec) |
|---------------|-----------------|----------|-----------|------|-------|-----------|----------|-----------|------|-------|-----------|
| Denoising     | $\ell_1$ (Soft) | 7.139    | 2.654     | 0.675| -     | 0.01      | 8.928    | 2.960     | 0.542| -     | 0.01      |
| SNR 10 dB     | $\ell_{0.5}$ (IRLS) | 5.339    | 4.113     | 0.760| 12    | 2.00      | 6.741    | 4.407     | 0.634| 13    | 2.27      |
|               | Proposed        | 5.006    | 4.265     | 0.774| 12    | 4.85      | 6.286    | 4.510     | 0.644| 13    | 5.64      |
| Deblurring    | $\ell_1$ (SALSA) | 2.525    | 6.454     | 0.873| 45    | 8.76      | 2.283    | 6.489     | 0.860| 61    | 11.27     |
| 9 $\times$ 9 uniform | Proposed      | 2.510    | 6.978     | 0.877| 7     | 12.56     | 2.186    | 6.989     | 0.866| 7     | 12.67     |
| Inpainting    | $\ell_1$ (SALSA) | 1.399    | 14.155    | 0.825| 344   | 34.01     | 1.977    | 11.790    | 0.810| 444   | 43.56     |
| 40% missing   | Proposed        | 0.551    | 15.747    | 0.873| 61    | 30.67     | 0.705    | 12.652    | 0.849| 56    | 27.99     |

4. Conclusions and Future Work
In this paper, we have proposed a new regularizer for image restoration problems, which consists of an approximation to the $\ell_p$ norm, with the exponent $p$ varying as a function of $|x|$. The
resulting denoising problem is solved iteratively using IRLS. This formulation was experimentally found to lead to a more accurate estimate than the $\ell_1$ or $\ell_{0.5}$ norm regularizers, for image

**Figure 2.** Denoising results with the Lena and Cameraman images: (a,f) original; (b,g) noisy (SNR 10 dB); Estimates with: (c,h) $\ell_1$-norm (soft thresholding); (d,i) $\ell_{0.5}$-norm (IRLS); (e,j) proposed approach.

**Figure 3.** Deblurring results with the Lena and Cameraman images: (a,d) blurred ($9 \times 9$ uniform blur, SNR 40 dB); Estimates with: (b,e) $\ell_1$-norm; (c,f) proposed approach.

**Figure 4.** Inpainting results with the Lena and Cameraman images: (a,d) observed (40% pixels missing at random, SNR 40 dB); Estimates with: (b,e) $\ell_1$-norm; (c,f) proposed approach.
denoising, deblurring, and inpainting with wavelets. Adjusting the exponent to get a convex function over both positive and negative values of \( x \), and extending the formulation to total variation [12] will be addressed in a future paper.

Acknowledgements
This work was supported by the Fundação para a Ciência e Tecnologia (FCT), Portuguese Ministry of Science and Higher Education through a Post-doctoral fellowship (contract no. SFRH/BPD/79011/2011) and FCT project UID/EEA/50009/2013.

References
[1] M. Afonso, J. B. Dias, and M. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *IEEE Trans. on Im. Proc.*, vol. 19, no. 9, pp. 2345–2356, 2010.
[2] A. C. Bovik, *Handbook of image and video processing*. Academic Press, 2010.
[3] R. Chartrand and W. Yin, “Iteratively reweighted algorithms for compressive sensing,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3869–3872.
[4] ———, “Nonconvex sparse regularization and splitting algorithms,” in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 237–249.
[5] P. L. Combettes and V. R. Wajs, “Signal recovery by proximal forward-backward splitting,” *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
[6] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery,” *Comm. on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.
[7] D. Donoho, “De-noising by soft thresholding,” *IEEE Trans. Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.
[8] J. Eckstein and D. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, no. 3, pp. 293–318, 1992.
[9] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” *Inverse problems*, vol. 23, no. 3, p. 947, 2007.
[10] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.
[11] G. Gasso, A. Rakotomamonjy, and S. Canu, “Recovering sparse signals with a certain family of nonconvex penalties and dc programming,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, 2009.
[12] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D*, vol. 60, pp. 259–268, 1992.
[13] X. Shen, L. Chen, Y. Gu, and H. So, “Square-root lasso with nonconvex regularization: An admm approach,” *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 934–938, 2016.
[14] Y. Wang, W. Yin, and J. Zeng, “Global convergence of admm in nonconvex nonsmooth optimization,” *arXiv preprint arXiv:1511.06324*, 2015.
[15] Z. Xu, S. De, M. Figueiredo, C. Studer, and T. Goldstein, “An empirical study of admm for nonconvex problems,” *arXiv preprint arXiv:1612.03349*, 2016.