ADMM PENALTY PARAMETER SELECTION WITH KRYLOV SUBSPACE RECYCLING TECHNIQUE FOR SPARSE CODING

Youzuo Lin, Brendt Wohlberg, and Velimir Vesselinov

Los Alamos National Laboratory
Los Alamos, NM 87545, USA

ABSTRACT

Sparse representations are widely used in a broad variety of fields. A number of different methods have been proposed to solve the sparse coding problem, of which the alternating direction method of multipliers (ADMM) is one of the most popular. One of the disadvantages of this method, however, is the need to select an algorithm parameter, the penalty parameter, that has a significant effect on the rate of convergence of the algorithm. Although a number of heuristic methods have been proposed, as yet there is no general theory providing a good choice of this parameter for all problems. One obvious approach would be to try a number of different parameters at each iteration, proceeding further with the one that delivers the best reduction in functional value, but this would involve a substantial increase in computational cost. We show that, when solving the sparse coding problem for a dictionary corresponding to an operator with a fast transform, requiring iterative methods to solve the main linear system arising in the ADMM solution, it is possible to explore a large range of parameters at marginal additional cost, thus greatly improving the robustness of the method to the choice of penalty parameter.

Index Terms— Sparse Coding, Alternating Direction Method of Multipliers, Krylov Subspace, Subspace Recycling

1. INTRODUCTION

Sparse representation has become a well established method in a wide variety of disciplines, including statistics, signal processing, computer vision, and machine learning [1, 2, 3, 4]. There are a number of different formulations of *sparse coding*, the inverse problem of computing the sparse representation of a given signal with respect to a given dictionary. In this work we will focus on one of the most common forms, Basis Pursuit DeNoising (BPDN) [5]

$$\arg\min_{\mathbf{x}} \left\{ \|D\,\mathbf{x} - \mathbf{y}\|_2 + \lambda \|\mathbf{x}\|_1 \right\}, \tag{1}$$

This research was supported by Los Alamos National Laboratory Environmental Programs Projects.

where $D \in \mathbb{R}^{m \times n}$ is the dictionary, $\mathbf{y} \in \mathbb{R}^m$ is the input signal, $\mathbf{x} \in \mathbb{R}^n$ is the corresponding sparse representation of the input signal, and λ is the regularization parameter.

Numerous computational methods have been developed to solve problem Eq. (1) [6, 7, 8]. The Alternating Direction Method of Multipliers (ADMM) [8] is one of the leading methods for this problem [9, 10]. When applying ADMM to the BPDN problem in Eq. (1), we have the iterations [8, Ch. 6]

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \Big\{ \|D\,\mathbf{x} - \mathbf{y}\|_2 + \rho \|\mathbf{x} - \mathbf{u}^{(k+1)} + \mathbf{d}^{(k+1)}\|_2^2 \Big\}, \tag{2}$$

$$\mathbf{s}^{(k+1)} = \mathcal{S}_{\lambda/\rho}\left(\mathbf{x}^{(k+1)} + \mathbf{d}^{(k)}\right), \tag{3}$$

$$\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{s}^{(k+1)}, \tag{4}$$

where $\mathcal{S}_{\mu}(\mathbf{v}) = \mathrm{sign}(\mathbf{v}) \odot \max(0, |\mathbf{v}| - \mu)$ is the soft thresholding operation and ρ is the penalty parameter. The choice of ρ is critical to obtaining good convergence [11].

The major computational cost involved in this algorithm is in solving Eq. (2). Various computational methods have been proposed for solving this problem. When the dictionary D is an explicit matrix, an LU or Cholesky pre-factorization of $D^T D + \rho I$ can be used for an efficient solution at each iteration [8], in which case the computational cost in solving Eq. (2) becomes $\mathcal{O}(nm^2)$ flops for the factorization [12], and $\mathcal{O}(nm)$ flops for the iterations. When D is represented as a transform operator, iterative solvers such as the Conjugate Gradient method are necessary for solving Eq. (2) [13]. The cost is $\mathcal{O}(nm)$ for all iterations.

A poor choice of ρ value can lead to very slow convergence. In order to obtain the optimal solution to the BPDN problem using ADMM method, complicated heuristics have been implemented to select an appropriate ADMM penalty parameter [14, 11, 15, 16]. These existing ADMM penalty parameter selection methods, however, can be complicated to implement, computationaly demanding, unreliable, or inapplicable to the BPDN problem. Therefore, an efficient and robust computational technique is needed to solve the minimization in Eq. (2).

The technique proposed in this work not only reduces the

computational cost at each solve of Eq. (2), but also provides an automatic ADMM penalty parameter selection scheme to yield good performance to the minimization problem in Eq. (1).

2. KRYLOV SUBSPACE GENERATION - GOLUB-KAHAN-LANCZOS BIDIAGONALIZATION

Krylov subspace based iterative solvers are more suitable than others for large-scale applications [17]. Since the proposed approach relies heavily on these solvers, here we provide necessary detail regarding the generation of the Krylov subspace utilizing Golub-Kahan-Lanczos (GKL) technique [17, 18].

For a given linear least-squares problem, $\min_{\mathbf{x}} \|A\,\mathbf{x} - \mathbf{b}\|_2$, we start the recursion with the right-hand side vector $\mathbf{b}$

$$\beta^{(1)}\mathbf{u}^{(1)} = \mathbf{b}, \quad \alpha^{(1)}\mathbf{v}^{(1)} = A'\,\mathbf{u}^{(1)}, \tag{5}$$

where $\|\mathbf{u}^{(1)}\|_2 = \|\mathbf{v}^{(1)}\|_2 = 1$. For $j = 1, 2, \ldots$, we take

$$\begin{cases} \beta^{(j+1)}\mathbf{u}^{(j+1)} = A\,\mathbf{v}^{(j)} - \alpha^{(j)}\mathbf{u}^{(j)}, \\ \alpha^{(j+1)}\mathbf{v}^{(j+1)} = A'\,\mathbf{u}^{(j+1)} - \beta^{(j+1)}\mathbf{v}^{(j)}, \end{cases} \tag{6}$$

where $\alpha^{(j+1)} \geq 0$ and $\beta^{(j+1)} \geq 0$. After k steps of the recursion in Eqs. (5) and (6), we can decompose the matrix A into three matrices: $U^{(k+1)}$, $B^{(k)}$, and $V^{(k)}$

$$\begin{aligned} V^{(k)} &= \left[\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}\right], \\ U^{(k+1)} &= \left[\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(k+1)}\right], \end{aligned}$$

and

$$B^{(k)} = \begin{bmatrix} \alpha^{(1)} & & & & \\ \beta^{(2)} & \alpha^{(2)} & & & \\ & \beta^{(3)} & \ddots & & \\ & & \ddots & \alpha^{(k)} & \\ & & & \beta^{(k+1)} & \end{bmatrix}.$$

The GKL bidiagonalization procedure also generates a subspace, named Krylov subspace, which is spanned by the column vectors in V_k, i.e.

$$\mathcal{K}_k = \mathrm{span}(V^{(k)}) = \mathcal{K}_k(A'\,A, A'\,\mathbf{b}). \tag{7}$$

3. ROBUST AND EFFICIENT SPARSE REPRESENTATION

3.1. Linear System Transformation and Matrix Structure

Using $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{s}^{(k+1)} + \mathbf{d}^{(k+1)}$, we can transform the minimization problem in Eq. (2) into a standard form

$$\tilde{\mathbf{x}}^{(k+1)} = \arg\min_{\tilde{\mathbf{x}}} \left\{ \|D\,\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|_2 + \rho\|\tilde{\mathbf{x}}\|_2^2 \right\}, \tag{8}$$

where $\tilde{\mathbf{y}} = D\,\mathbf{s}^{(k+1)} - D\,\mathbf{d}^{(k+1)} - \mathbf{y}$. The standardized form in Eq. (8) can be therefore posed equivalently as

$$\tilde{\mathbf{x}}^{(k+1)} = \arg\min_{\tilde{\mathbf{x}}} \left\{ \left\| \begin{bmatrix} D \\ \sqrt{\rho}\,I \end{bmatrix} \tilde{\mathbf{x}} - \begin{bmatrix} \tilde{\mathbf{y}} \\ 0 \end{bmatrix} \right\|_2 \right\}. \tag{9}$$

With $\tilde{\mathbf{x}}$ calculated, the original solution can be obtained correspondingly

$$\mathbf{x}^{(k+1)} = \tilde{\mathbf{x}}^{(k+1)} + \mathbf{s}^{(k+1)} - \mathbf{d}^{(k+1)}. \tag{10}$$

We observe that in Eq. (9), the system matrices consist of two parts: the dictionary matrix D and the identity matrix I. To select the optimal ADMM penalty parameter ρ, the dictionary matrix D remains the same while ρ varies. A significant amount of computational cost can be wasted without considering this special matrix structure.

3.2. Efficient Sparse Representation with Krylov Subspace Recycling

Employing a similar bidiagonalization procedure as in Eqs. (5) and (6), we have

$$\min_{\tilde{\mathbf{x}}} \left\{ \left\| \begin{bmatrix} D \\ \sqrt{\rho}\,I \end{bmatrix} \tilde{\mathbf{x}} - \begin{bmatrix} \tilde{\mathbf{y}} \\ 0 \end{bmatrix} \right\|_2 \right\} \rightarrow \min_{\bar{\mathbf{x}}} \left\{ \left\| \begin{bmatrix} B^{(k)} \\ \sqrt{\rho}\,I \end{bmatrix} \bar{\mathbf{x}} - \beta^{(1)}\mathbf{e}^{(1)} \right\|_2 \right\}. \tag{11}$$

The corresponding Krylov subspace generated in Eq. (11) is

$$\mathcal{K}_k = \mathrm{span}\left\{D'\,D + \rho I, D'\,\tilde{\mathbf{y}}\right\}. \tag{12}$$

From the definition of a Krylov subspace in Eq. (7), it is clear that the subspace is invariant under translations of the system matrix. Therefore, we will have

$$\begin{aligned} \mathcal{K}_k &= \mathrm{span}\left\{D'\,D + \rho I, D'\,\tilde{\mathbf{y}}\right\}, \\ &= \mathrm{span}\left\{D'\,D, D'\,\tilde{\mathbf{y}}\right\}. \end{aligned} \tag{13}$$

We observe that in Eq. (13), the Krylov subspace generated through the bidiagonalization procedure is independent of the ADMM penalty parameter ρ. This invariance property of the Krylov subspace allows us to develop a novel ADMM penalty selection strategy.

3.3. Robust ADMM Penalty Parameter Selection

We generate n logarithmically spaced ADMM penalty parameters ρ between 10^a and 10^b. We randomly select one out of the n values as the seed system and the remaining $n-1$ as the non-seed systems, and further solve them, respectively. The objective function values in Eq. (1) are evaluated and compared for each update of $\mathbf{x}$ at every iteration when Eq. (2) is solved. We select the ρ that yields the smallest objective function value in Eq. (1) and use that to further obtain $\mathbf{s}$ and update $\mathbf{d}$ in Eqs. (3) and (4). It is worth mentioning that the selection of the optimal penalty parameter is independent of the linear solvers, although it can be computational extremely intensive and impractical when direct methods are employed. The proposed computational method, on the other hand, can obtain multiple solutions with different ADMM penalty parameters at marginal additional cost.

3.4. Seed and Non-Seed Systems

Solving the seed system achieves two goals: obtaining the Krylov subspace in Eq. (13) and solving for the solution $\tilde{\mathbf{x}}$. To generate the the Krylov subspace, we employ the GKL bidiagonalization as in Eqs. (5) and (6) discussed previously.

The solution $\tilde{\mathbf{x}}$ can be obtained by solving the approximated linear system in Eq. (11). We observe that Eq. (11) is an augmented least-squares problem with a system-matrix consisting of a lower-bidiagonal matrix $B^{(k)}$ and a diagonal matrix $\sqrt{\rho}\, I$. Careful exploration of the matrix structure of Eq. (11) can further reduce the computational cost in solving for the solution $\tilde{\mathbf{x}}$. We employ Givens rotations to eliminate the diagonal matrix $\sqrt{\rho}\, I$ and meanwhile transform the lower-bidiagonal matrix $B^{(k)}$ into a upper-bidiagonal matrix. Specifically, provided with a Givens matrix $G_{i,j}$ for the i^{th} row and j^{th} column, at each iteration, we have

$$G_{1,2}\, G_{1,k+2} \begin{bmatrix} B^{(k)} \\ \sqrt{\rho}\, I \end{bmatrix}. \tag{14}$$

Because of the orthogonality of the Givens matrix, we can transform the least-squares problem in Eq. (11) into an equivalent problem

$$\min_{\bar{\mathbf{x}}} \left\{ \left\| G_{1,2}\, G_{1,k+2} \left(\begin{bmatrix} B^{(k)} \\ \sqrt{\rho}\, I \end{bmatrix} \bar{\mathbf{x}} - \beta^{(1)} \mathbf{e}^{(1)} \right) \right\|_2 \right\}. \tag{15}$$

Performing k steps of Givens rotation, we will have an equivalent least-squares problem

$$\min_{\bar{\mathbf{x}}} \left\{ \left\| \left(\begin{bmatrix} (R^{(k)})' \\ 0 \end{bmatrix} \bar{\mathbf{x}} - \begin{bmatrix} (\mathbf{g}^{(k)})' \\ (\boldsymbol{\phi}^{(k)})' \end{bmatrix} \right) \right\|_2 \right\}. \tag{16}$$

It is worth mentioning that the upper bidiagonal matrix $(R^{(k)})'$ in Eq. (16) now contains the ADMM penalty parameter, ρ. Many efficient methods can be used to solve the least squares problem in Eq. (16) such as Gaussian elimination and back-substitution [19]. A three-term-recursion can be derived to update the solution

$$\begin{aligned} \tilde{\mathbf{x}}^{(k)} &= \tilde{\mathbf{x}}^{(k-1)} + (\phi^{(k)})'(\mathbf{z}^{(k)})', \\ (\mathbf{z}^{(k)})' &= \frac{1}{(\gamma^{(k)})'} (\mathbf{v}^{(\mathbf{k})} - (\theta^{(k-1)})'(\mathbf{z}^{(k-1)})'). \end{aligned} \tag{17}$$

Solving the non-seed systems involves one purpose only, to obtain the solution $\tilde{\mathbf{x}}$. Since the Krylov subspace has been generated, we will recycle the subspace for all the non-seed systems. With the subspace generated, all that is needed to solve for $\tilde{\mathbf{x}}$ of the non-seed systems is to solve the system in Eq. (16), which leads to the three-term-recursion in Eq. (17) with different values of the ADMM penalty parameter ρ. Therefore, solving the non-seed systems is computationally efficient.

3.5. Computational Cost

Solving the seed system can be computationally intensive. It involves matrix-vector multiplications to generate the Krylov subspace as shown in Eqs. (5) and (6). Hence, the cost of solving the seed system is $\mathcal{O}(nm)$. While solving the non-seed system involves only the projection operations given in Eq. (17), so its cost is $\mathcal{O}(n)$. Therefore, a significant amount of cost has been reduced using subspace recycling technique.

4. NUMERICAL RESULTS

We now provide two numerical tests to demonstrate the performance of our new computational technique for solving the BPDN problem. We apply our method to standard image denoising problem [20]: a clean image $\mathbf{x}$ is corrupted with an additive zero-mean white Gaussian noise with standard deviation $\sigma = 0.05$. The image denoising is an important problem since it not only serves a wide application, but also includes all the ingredients of inverse problem, over which computational methods and techniques can be assessed.

We select the discrete cosine transform (DCT) basis as the dictionary in Eq. (1). The testing image is a gray scale "Lena" image with size of 512×512. In order to test the robustness of our approach to the choice of initial penalty parameter ρ_{init}, we repeat our experiment for all values of ρ_{init} in the set $\{0.05,\ 0.08,\ 0.1,\ 0.2,\ 0.3,\ 1.0,\ 3.0,\ 10.0,\ 20.0$ and $30.0\}$. For each choice of ρ_{init} we generate $n = 10$ ADMM penalty parameters at which the functional will be evaluated using the proposed method. These values are logarithmically spaced between 10^a and 10^b, where $a = -ln(\rho_{\text{init}})$ and $b = ln(\rho_{\text{init}})$.

The stopping criterion is an important issue for any iterative method. We employ two stopping criteria to justify the convergence of the iteration

$$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \geq \tau \|\mathbf{x}^{(k)}\|_2, \tag{18}$$

$$\text{Iteration Number} < 100, \tag{19}$$

where $\tau = 1.0 \times 10^{-6}$. We select BPDN with CG method as a reference method ("BPDN-CG") and denote our method as "BPDN-RS" short for BPDN with Recycled Subspace.

4.1. Test on Linear Solver Efficiency

In this test, we provide the computation time using both methods of "BPDN-CG" and "BPDN-RS". In Fig. 1, we give the computation time in solving the systems in Eq. (2) using $n = 10$ penalty parameters using our method. We observe that at each iteration, almost all the computation time is utilized in the first ADMM penalty parameter, which corresponds to the seed system. As discussed previously, the computation cost in solving the seed system is dominated by the generation of the Krylov subspace. Solving the non-seed systems is much more efficient than solving the seed system. As shown in the

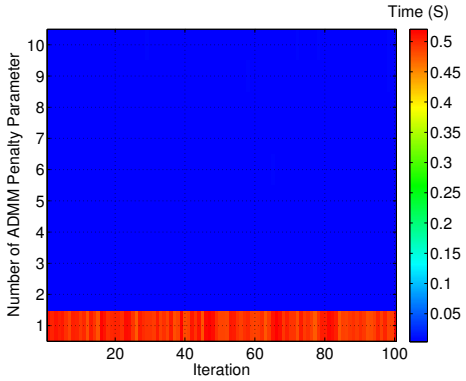

Fig. 1. The computation time costs in solving the systems in Eq. (2) using $n = 10$ penalty parameters for our method. The time cost in solving the non-seed system is trivial. It is about an order less time consuming than solving the seed system.

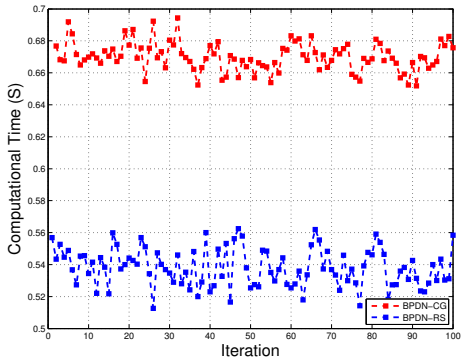

Fig. 2. The overall computation time at each iteration using BPDN with CG solver (in red) and BPDN with our subspace recycling technique (in blue). Our method is consistently more efficient than BPDN-CG method.

Fig. 1, the time cost in solving the non-seed system is trivial. It is about an order of magnitude less time consuming than solving the seed system, which confirms with our computational cost analysis in previous sections. We also provide in Fig. 2 the results of the overall computation time at each iteration using the reference method BPDN-CG and our method, BPDN-SR. The overall computation time of BPDN-CG includes the time solving Eq. (2) using one ADMM penalty parameter. On the other hand, the overall computation time of our method at each iteration includes the time solving Eq. (2) with $n = 10$ ADMM penalty parameters. We notice that our method is consistently more efficient than BPDN-CG method.

4.2. Test on Reconstruction Accuracy

The selection of ADMM penalty parameter is critical to the resulting reconstruction in solving the BPDN problem in Eq. (1). In this test, we provide results illustrating the reconstruction accuracy of our computational method. We reported

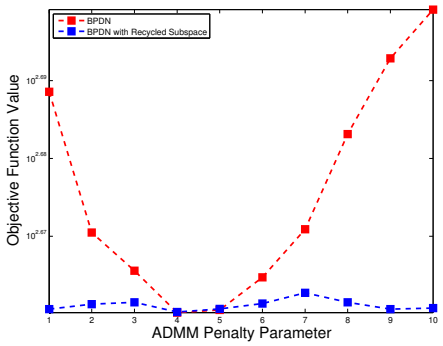

Fig. 3. Plots of the objective function values versus ρ_{init} by solving BPDN in Eq. (1) using the same ρ_{init} values (in red) and by using our new technique (in blue). The results obtained using our method yields objective function value consistently close to the optimal value regardless of the ρ_{init}.

the resulting objective function values versus different ρ_{init} values in Fig. 3. The results of solving BPDN in Eq. (1) using the ρ_{init} values is plotted in red. We observe that the objective function values varies significantly with different ρ_{init} values. An increased objective function value in the image denoising application means either over smoothed or under regularized resulting image. By having the plot of objective function values versus ρ_{init} values, we can locate the optimal ADMM penalty parameter in-between $\rho = 0.2$ and $\rho = 0.3$ in this testing problem, and the corresponding objective function value is roughly 458. On the other hand, our computational technique yields a much more stable results. As indicated by the blue curve in Fig. 3, the results obtained using our method yields objective function value consistently close to the optimal value regardless of ρ_{init}, demonstrating the robustness of our method.

5. CONCLUSIONS

We have developed a robust and computationally efficient ADMM penalty parameter selection technique using Krylov subspace recycling technique. We generate multiple penalty parameters at each iteration and proceeding further with the one that yields the largest reduction in objective function value. To reduce the excessive computational cost in exploring large range of parameters, we employ Krylov subspace recycling technique, which consisting of solving for the seed and the non-seed systems. In the seed system, a solution is computed while Krylov subspace is generated. In the non-seed systems, the generated subspace can be recycled to obtain the new solution with high accuracy. We further demonstrate the performance of our method by applying it to sparse representation. Through our numerical tests, we show that our method yields comparable costs to the traditional linear solvers while significantly improve the robustness and computational efficiency of the resulting reconstruction.

6. REFERENCES

[1] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[3] L. Li and B. Jafarpour, "Effective solution of nonlinear subsurface flow inverse problems in sparse bases," *Inverse Problems*, vol. 26, no. 10, pp. 105016, 2010.

[4] A. Khalil, M. N. Almasri, M. McKee, and J. J. Kaluarachchi, "Applicability of statistical learning algorithms in groundwater quality modeling," *Water Resources Research*, vol. 41, no. 5, 2005.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[6] I. F. Gorodnistsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[7] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Optimization*, vol. 3, no. 1, pp. 1–122, 2010.

[9] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[10] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2016.

[11] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[12] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2000.

[13] P. Rodríguez and B. Wohlberg, "Performance comparison of iterative reweighting methods for total variation regularization," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 1758–1762.

[14] Z. Xu, M. Figueiredo, and T. Goldstein, "Adaptive ADMM with spectral penalty parameter selection," *arXiv:1605.07246v3*, 2016.

[15] A. Raghunathan and S. D. Cairano, "Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection," in *American Control Conference*, 2014, pp. 4324–4329.

[16] Brendt Wohlberg, "ADMM penalty parameter selection by residual balancing," arxiv:1704.06209, 2017.

[17] Y. Saad, *Iterative Methods for Sparse Linear Systems (Second Edition)*, SIAM, 2003.

[18] A. Bjorck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, third edition, 1996.

[20] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, USA, June 2006, pp. 1924–1931.