Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

A General Efficient Hyperparameter-Free Algorithm for Convolutional Sparse Learning

Zheng Xu, Junzhou Huang
Department of Computer Science and Engineering
The University of Texas at Arlington
701 S. Nedderman Drive, Arlington, Texas 76019

Abstract

Structured sparse learning has become a popular and mature research field. Among all structured sparse models, we found an interesting fact that most structured sparse properties could be captured by convolution operators, most famous ones being total variation and wavelet sparsity. This finding has naturally brought us to a generalization termed as **Convolutional Sparsity**. While this generalization bridges the convolution and sparse learning theory, we are able to propose a general, efficient, hyperparameter-free optimization algorithm framework for convolutional sparse models, thanks to the analysis theory of convolution operators. The convergence of the general, hyperparameter-free algorithm has been comprehensively analyzed, with a non-ergodic rate of $\mathcal{O}(1/\varepsilon^2)$ and ergodic rate of $\mathcal{O}(1/\varepsilon)$, where ε is the desired accuracy. Extensive experiments confirm the superior performance of our general algorithm in various convolutional sparse models, even better than some application-specialistic algorithms.

Introduction

We investigate the following convolutional sparse learning (Huang, Zhang, and Metaxas 2011; Huang, Zhang, and others 2010) framework where the original signal is sparse in terms of the representation filtered by some convolution operators. It can be formulated as:

$$\min_{\beta} f(\beta) + g(\beta) + \sum_{i=1}^{m} \lambda_i \|k_i \star \beta\|_1, \tag{1}$$

where $f(\beta)$ is a proper, lower semi-continuous, sufficiently smooth convex function. Famous examples of f include the least squares ($\|X\beta - \mathbf{y}\|^2/2$) and logistic ($\sum_j \log(1 + \exp(\beta^T \mathbf{x}_j + b_j))$) regressions. $g(\beta)$ is a proper, lower semi-continuous, convex function, which usually appears as a regularizer, e.g., nuclear norm ($\|\cdot\|_*$), Tikhonov regularizer $\|\cdot\|_2^2$, etc. $\lambda_i > 0$ is the tuning parameter, balancing among different convolution operations, and $k_i \in L^1$ is the absolute integrable convolution kernel function. $\cdot \star \cdot$ is the convolution operator defined rigidly on locally compact abelian group.

This generic convolutional sparse framework generalizes a sufficient large class of sparse learning models, which includes but not limited to, fused LASSO (Tibshirani et al. 2005), directional total variation (Bayram and Kamasak 2012), wavelet sparsity (Ma et al. 2008; Chen and Huang 2012) and even higher-order total variation (Toma et al. 2014; Chan et al. 2015), etc. These most recent models have been demonstrated effective in various applications, e.g., feature selection, computer vision, etc. Our convolutional sparse framework bridges a gap between the standard sparsity (Candes, Romberg, and Tao 2006), without structural priori that could suffer from the requirements of more measurements, and the dictionary sparsity (Candes et al. 2011; Liu, Yuan, and Ye 2013) that lacks the decent properties from the convolution operators. The standard sparsity can be viewed as a specialized case in our framework while only one convolution filter is used and the kernel is fixed to $k = [1]$. The dictionary sparse recovery (Candes et al. 2011; Liu, Yuan, and Ye 2013), in a general case, employs $\lambda\|D\beta\|_1$ as the regularizer instead of $\sum_{i=1}^{m} \lambda_i \|k_i \star \beta\|_1$ in our framework, where D is a linear bounded operator and also usually referred as a redundant dictionary. While the dictionary sparsity seems more generic than our framework, however, in practice, the model usually suffers from the complicated analysis of D. For instance, the design of algorithm in (He and Yuan 2012) requires the estimation of the spectrum norm of linear operator D, but it generally requires $\mathcal{O}(N^3)$ time complexity for an accurate numerical computation, which is absolutely prohibited in large-scale applications. Furthermore, it is worth mentioning another seemingly similar framework *convolutional sparse coding* (CSC) (Heide, Heidrich, and Wetzstein 2015). The CSC model basically focus on learning a group of convolution operators acting as the measurement matrix X, i.e., in $f(\beta) = \|X\beta - y\|_2^2$, it replaces X with a set of convolution operators and uses a standard sparse constraint. Overall, the CSC model is a special case of standard sparse learning and, in succession, a special case of our convolutional sparse framework.

At present, the choice of optimization algorithm is still customarily left as an application-specific detail for the end-user to determine. While the convolutional sparse framework generalizes a wide class of sparse learning models, we can now imagine a general, easy-to-use algorithm that solves all convolutional sparse models. However, it is hard to efficiently solve the convolutional sparse formulation, since the formulation couples an non-smooth non-separable convolutional sparse regularizer and a non-smooth term $g(\beta)$ with

the smooth function $f(\beta)$. Alternating Direction Method of Multipliers (ADMM) is a standard method to solve such general model, but it usually requires lots of time to tune the penalty parameters and it can be arbitrarily slow to converge to a high accuracy (Boyd et al. 2011). FCSA (Huang et al. 2011) might be another choice for solving such model, which utilizes the idea of the composition of non-expansive monotone operators (Combettes* 2004). However, every subproblem in FCSA requires the iterative solution, which is very computationally expensive. In (Condat 2013; Chambolle and Pock 2014), the Fenchel-type dual problem is suggested to instead solve such type of model to avoid the iterative solution of the subproblem. But these methods still rely on the accurate estimation of the largest singular value of the composite convolution operator $\sum_{i=1}^{m} \lambda_i k_i \star \cdot$ to find a converged step-size setting, which, as mentioned before, is not feasible in practice. Overall, an efficient, accurate and easy-to-use algorithm is highly demanded for convolutional sparse framework.

To resolve this issue, we propose a **general, efficient, and hyperparameter-free** algorithm for convolutional sparse framework, extending the algorithm described in (Condat 2013; Chambolle and Pock 2014). Thanks to the introduction of the convolution operator, we are able to break the step-size assumptions in (Condat 2013; Chambolle and Pock 2014) and replace it with a fixed, easy-to-compute, accelerated step-size settings. While there is no convergence analysis for the resulting algorithm, we offer the convergence analysis, together with the non-ergodic and ergodic convergence rate analysis. The non-ergodic and ergodic convergence rates for reaching a ε-optimal solution are $\mathcal{O}(1/\varepsilon^2)$ and $\mathcal{O}(1/\varepsilon)$, respectively. To our best knowledge, it is the first work to show the non-ergodic rate for such primal-dual algorithm. The ergodic rate result seems same as the previous methods (Condat 2013; Chambolle and Pock 2014), but extensive experiments demonstrate that our algorithm performs much faster in practice due to its more aggressive step-size setting and computational inexpensiveness of the algorithmic constants. In summary, our contributions include:

1. We propose the convolutional sparsity framework, modeling a wide class of sparse learning problems involving sparse representation under convolution filtering. To our best knowledge, it is the first attempt to model a class of sparsity problem with convolution filtering in the literature.
2. We propose a general, efficient, and hyperparameter-free algorithm for convolutional sparse framework, extending the efficient primal-dual algorithm in (Condat 2013; Chambolle and Pock 2014). The proposed algorithm comes with a more aggressive and computationally cheap step-size setting than (Condat 2013; Chambolle and Pock 2014), benefiting from the decent properties of the convolution operators. While there is no existing convergence analysis for the proposed algorithm, we extend the weak convergence theorem (Condat 2013) and ergodic rate result (Chambolle and Pock 2014) to the proposed algorithm. Moreover, we provide the first non-ergodic rate analysis in this class of primal-dual sparse learning algorithm, better characterizing the convergence behavior of the proposed algorithm.

Convolutional Sparse Models

In this section, we list few existing examples that fits in the convolutional sparsity framework by playing around the properties of convolution operators with elegant explanations. The paragraphs below are sorted in the order of our estimated simplicity and interest, with simplest and most interesting first.

Total Variation via Convolution The fused LASSO (Tibshirani et al. 2005) solves:

$$\min_{\beta} \frac{1}{2}\|X\beta - y\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\nabla\beta\|_1, \qquad (2)$$

where $\beta \in \mathbb{R}^N$ and $\nabla\beta$ is the forward difference (**1D total variation**) operator, i.e., $(\nabla\beta)_i = \beta_{i+1} - \beta_i$, which can therefore be represented as $\nabla\beta = [1, -1] \star \beta$. Also we note the standard sparsity can be represented as $\beta = [1] \star \beta$. Thus when we set $k_1 = [1]$, $k_2 = [1, -1]$ and $f(\beta) = (1/2)\|X\beta - y\|^2$, the fused LASSO model (2) is represented in (1).

Distributivity Law The convolutional sparsity framework also enjoys the **distributivity law** of convolution. A typical example is the directional total variation (Bayram and Kamasak 2012), which is a direction-sensitive variant of regular total variation. It applies a rotation and scaling operation onto the horizontal and vertical gradient of a vector field with two spatial dimensions. The crux of directional total variation denoising[1] is basically

$$\min_{\beta} \frac{1}{2}\|\beta - \mathbf{y}\|^2 + \lambda \left\| \begin{bmatrix} \alpha\cos\theta\nabla_h + \alpha\sin\theta\nabla_v \\ -\sin\theta\nabla_h + \cos\theta\nabla_v \end{bmatrix} \beta \right\|_1. \qquad (3)$$

$\alpha \in \mathbb{R}$ is the scaling factor, θ is the rotation angle, and ∇_h, ∇_v are the horizontal and vertical discrete gradient operators, respectively. Note ∇_h, ∇_v act as the convolution operators with kernel $[1, -1]$ and $[1, -1]^T$. By the distributivity law of convolution, the equivalent formulation of (3) in convolutional sparsity framework is thus

$$\begin{aligned} \min_{\beta} \frac{1}{2}\|\beta - \mathbf{y}\|^2 &+ \lambda \left\| \begin{bmatrix} \alpha\cos\theta + \alpha\sin\theta & -\alpha\cos\theta \\ -\alpha\sin\theta & 0 \end{bmatrix} \star \beta \right\|_1 \\ &+ \lambda \left\| \begin{bmatrix} \cos\theta - \sin\theta & \sin\theta \\ -\cos\theta & 0 \end{bmatrix} \star \beta \right\|_1. \end{aligned} \qquad (4)$$

Associative Law In our framework, the **associative law** of convolution is another interesting property, which is often seen when a compound convolution operator is used. One might be familiar with the higher-order total variation, which is well adopted in various applications, e.g., (Lenzen, Becker, and Lellmann 2013; Toma et al. 2014; Chan et al. 2015). To

[1] To simplify the representation, we use anisotropic directional total variation, which is slightly different from the regular (isotropic) version. However, in practice, they achieve similar performance.

simply demonstrate how it works in convolutional sparsity, we take the formulation in (Chan et al. 2015), which is

$$\min_{\beta} \frac{1}{2}\|X\beta - \mathbf{y}\|_2^2 + \lambda_1\|\nabla\beta\|_1 + \lambda_2\|\Delta\beta\|_1. \quad (5)$$

Here, ∇ is the discrete first-order total variation (TV) while $\Delta = \nabla^2$ is the second-order TV (also known as *discrete Laplacian operator*). To state the problem in convolutional sparsity framework, we can simply set $k_1 = [1, -1]$ and $k_2 = [1, -2, 1]$. We note the implementation of discrete Laplacian operator actually takes advantage of associative law of convolution. Since $\Delta = \nabla^2$, we thus have $\Delta\beta = \nabla\nabla\beta = k_1 \star k_1 \star \beta = (k_1 \star k_1) \star \beta = k_2 \star \beta$. This property implies that we can simply represent a higher-order TV as a single convolution operator.

Primal-Dual Algorithm Framework for Convolutional Sparse Models

In this section, we will describe our algorithm to solve problem (1) and its convergence analysis. We start with the description of some basic concepts and the primal-dual problem in this paper. We then detail the proposed algorithm with elegant interpretations. Finally, we complete our main results with the convergence theorem and the complexity bounds in terms of both non-ergodic and ergodic convergence rates.

Preliminary

Convolution Here, we first define the convolution operator in this paper. Let $\mathcal{X}$ be a locally compact abelian (LCA) group with identity e. Let us fix a left Haar measure λ on $\mathcal{X}$, using the notation $dx := d\lambda(x)$. The Lebesgue space $L^p(\mathcal{X}) \equiv L^p(\mathcal{X}, d\lambda), 1 < p < +\infty$, of $\mathcal{X}$ with respect to λ is endowed with the usual p**-norm**: $\|\beta\|_p = \left(\int_{\mathcal{X}} |\beta(x)|^p dx\right)^{1/p}$. We note $\mathcal{B}_p(r)$ denotes the p-norm ball of radius r, which will be shortened to $\mathcal{B}_p$ if $r = 1$.

In the sequel, we set $\mathcal{X} := (\mathbb{Z}, +)$ as an additive group on integer set $\mathbb{Z}$, $e := 0$, with $\lambda(y)$ being the counting measure with every point assigned mass 1. Thus the we define the **convolution operator** (discrete convolution) in this paper: $(k \star \beta)[x] = \sum_{y=-\infty}^{+\infty} k[x - y] \cdot \beta[y]$.

If k or β has a finite support, the convolution could be instead summed over the finite support set, e.g., $[1, -1] \star [1, -1] = [1, -2, 1]$.

In the sequel, we mainly consider the following **composite (multi-kernel) convolution operator**:

$$\begin{aligned}\mathcal{C}_{[m]}\beta = \mathcal{C}_{k_1,k_2,\ldots,k_m}\beta &= [\mathcal{C}_{k_1}\beta, \mathcal{C}_{k_2}\beta, \ldots, \mathcal{C}_{k_m}\beta] \\ &= [k_1 \star \beta, k_2 \star \beta, \ldots, k_m \star \beta]. \quad (6)\end{aligned}$$

Here, the square bracket denotes the concatenation of convolution operators. Throughout this paper, we use superscript H to denote the Hermitian adjoint of the linear operator, e.g., $\mathcal{C}_{[m]}^H$.

Fenchel's Duality Now we assume the LCA group $\mathcal{X}$ is Hilbert and equipped with inner product $\langle\cdot,\cdot\rangle$. $\mathcal{U}_i$ is the dual space of $\mathcal{X}$ with respect to linear operator $\mathcal{C}_{k_i}$. We define U-norm as $\|\beta\|_U = \langle\beta, U\beta\rangle^{1/2}$ for a positive-definite self-adjoint linear bounded operator U. A function $f \in L^p(\mathcal{X})$ is **L-strongly smooth** if it is Fréchet differentiable and its gradient is L-Lipschitz, i.e., $\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$, where L is the **Lipschitz smooth constant** of f.

Given a proper, lower semi-continuous convex function $h \in L^1(\mathcal{X})$, its **Legendre-Fenchel conjugate function** is denoted, through **Legendre transformation**, by $h^*(\mu) = \sup_\beta\langle\mu, \beta\rangle - h(\beta)$. It is well-known that the Legendre transformation, of a proper, lower semi-continuous convex function, is an involution, i.e., $h^{**} = h$. A **linear proximal operator** of function f at β_0 is $\mathbf{proj}_f(\beta_0) = \arg\min_\beta \left\{\frac{1}{2}\|\beta - \beta_0\|_2^2 + f(\beta)\right\}$. We denote $\mathcal{I}_S(x)$ as the indicator function of set S, which is 0 if $x \in S$, otherwise $+\infty$. A **Euclidean projection** onto convex set S is the linear proximal operator of the indicator function of convex set S at β_0, i.e., $\Pi_S(\beta_0) = \mathbf{proj}_{\mathcal{I}_S}(\beta_0) = \arg\min_\beta \left\{\frac{1}{2}\|\beta - \beta_0\|_2^2 + \mathcal{I}_S(\beta)\right\}$.

The **primal-dual** problem of (1) is,

$$\min_{\beta\in\mathcal{X}} \max_{\mu_i\in\mathcal{B}_\infty} f(\beta) + g(\beta) + \sum_{i=1}^m \lambda_i\langle k_i \star \beta, \mu_i\rangle, \quad (7)$$

which is based on the fact that the Legendre transformation of $h(\beta) = \lambda_i\|k_i \star \beta\|_1$ is $h^*(\mu_i) = \sup_{\mu_i} \lambda_i\langle k_i \star \beta, \mu_i\rangle - \mathcal{I}_{\mathcal{B}_\infty}(\mu_i)$, where $\mu_i \in \mathcal{U}_i$ is the dual variable. The dual objective, related to μ_i, is reduced to an affine mapping with a constraint on the convex set formed by ℓ_∞ unit norm ball. To simplify our description, without any loss of generality, we use $k_i := \lambda_i k_i$ and $\mathcal{C}_{[m]}\beta := \sum_{i=1}^m k_i \star \beta$ hereafter, and thus $\sum_{i=1}^m \lambda_i\langle k_i \star \beta, \mu_i\rangle$ becomes $\langle\mathcal{C}_{[m]}\beta, \mu\rangle$ where $\mathcal{U} = \prod_{i=1}^m \mathcal{U}_i$ is the direct product of all dual spaces. Thus the primal-dual form (7) is simplified to:

$$\min_{\beta\in\mathcal{X}} \max_{\mu\in\mathcal{B}_\infty} f(\beta) + g(\beta) + \langle\mathcal{C}_{[m]}\beta, \mu\rangle. \quad (8)$$

Additionally, we assume the linear proximal operator of g is easy to evaluate (i.e., the $\mathbf{prox}_g$ has an explicit form or is computationally efficient to solve to a high precision).

We then define the **primal-dual gap** used throughout this paper, Let $P(\beta) = f(\beta) + g(\beta) + \|\mathcal{C}_{[m]}\beta\|_1$, and $D(\mu) = -f^*(\mathcal{C}_{[m]}^H\mu) - g^*(\mathcal{C}_{[m]}^H\mu) - \mathcal{I}_{\mathcal{B}_\infty}(-\mu)$ be the primal and dual objective, respectively, Hence the **primal-dual gap** of problem (8) is defined as $P(\beta) - D(\mu)$.

Primal-Dual Algorithm for Convolutional Sparsity

Primal Update To solve the primal-dual problem (7), our method starts with an initial primal-dual vector pair $(\beta^{(0)}, \mu^{(0)})$ on $\mathcal{X} \times \mathcal{U}$. At step t, we will first update the primal variable through gradient descent, with a selected step size $1/(L + \zeta)$ where $\zeta = \sqrt{\sum_{i=1}^m \|k_i\|_1^2}$, which is simply

$$\beta^{(t+1)} = \mathbf{prox}_{\frac{g}{L+\zeta}}\left(\beta^{(t)} - \frac{\nabla f(\beta^{(t)}) + \mathcal{C}_{[m]}^H\mu^{(t)}}{L + \zeta}\right). \quad (9)$$

Here, L is the Lipschitz smooth constant of f, which bounds the function f from sharp mutation. The constant ζ provides a stability estimation of the composite multi-kernel convolution

operator $\mathcal{C}_{[m]}$, bounding the sensitivity of the convolution operator $\mathcal{C}_{[m]}$. ∇f, as the gradient of f, is a singleton along the domain of f, thanks to the smoothness of f. $\mathcal{C}_{[m]}^H \mu^{(t)}$ is the gradient of the primal-dual bridge $\langle \mathcal{C}_{[m]}\beta, \mu\rangle$ with respect to β. The step size of the primal update step is fixed to $1/(L+\zeta)$, which implies that the step size is controlled by both the smoothness of f and also the sensitivity of convolution operators. If L is large, meaning that the function f could have sudden change, the step size is set smaller to avoid oscillation when approaching the optima. The step size is also smaller when ζ becomes larger, which indicates a small noise could lead to large error in target variable.

Dual Update The proposed algorithm performs an extrapolation step by $\tilde{\beta}^{(t)} = 2\beta^{(t+1)} - \beta^{(t)}$, which is also seen in extragradient methods, e.g., (Cai, Gu, and He 2014), to accelerate convergence. The dual objective is then updated by

$$\mu^{(t+1)} = \Pi_{\mathcal{B}_\infty}(\mu^{(t)} + \frac{1}{\zeta}\mathcal{C}_{[m]}\tilde{\beta}^{(t)}). \tag{10}$$

The dual update is generally composed of two steps. First, we accent the dual objective along the gradient direction of $\langle \mathcal{C}_{[m]}\tilde{\beta}^{(t)}, \mu\rangle$ at point $\mu^{(t)}$. Secondly, the resulting dual variable is projected, through the Euclidean projection operator $\Pi_{\mathcal{B}_\infty}$, onto the feasible domain; that is, the ℓ_∞ unit norm ball. The step size of dual update is fixed to $1/\zeta$, controlling the effect of small perturbation on primal variable. The proposed algorithm is summarized in Algorithm 1.

The Step Size Constant ζ One might be curious about the magical constant number $\zeta = \sqrt{\sum_{i=1}^m \|k_i\|_1^2}$ in our step size setting, which seems not standard in the optimization field. In fact, ζ could be roughly viewed as an *upper estimation* for the spectrum norm of the composite convolution operator $\mathcal{C}_{[m]}$. Let us just consider $\mathcal{C}_{[1]}$ with only one associated convolution kernel $k_1 = [1, -1]$, and according to Young's inequality, we instantly have $\|\mathcal{C}_{[1]}\| \leq \|k_1\|_1 = \zeta = \sqrt{\|k_1\|_1^2} = 2$. For step-size computation, the only numerical computation required in our method is $|1| + |-1| = 2$. However, if we directly compute $\|\mathcal{C}_{[1]}\|$ by the power method, as required by (Condat 2013; Chambolle and Pock 2014), the computation cost will be at $\mathcal{O}(N^3)$ where N is the feature dimension of β. Our algorithm thus significantly reduces the computational cost for the step-size estimation, since the computational cost of our method is feature dimension independent. Due to the page limit, we offer the detailed analysis on this magical constant ζ and direct theoretical application of our result in the supplementary material.

Since our step-size setting is an upper estimation of $\|\mathcal{C}_{[m]}\|$, it hence breaks the step size assumption as appears in (Condat 2013; Chambolle and Pock 2014). As a result, the proposed method does not have any theoretical convergence guarantee and convergence rate analysis yet. We will complete these studies in the next section.

Algorithm 1 Primal-Dual Algorithm for Convolutional Sparsity

Let $\zeta = \sqrt{\sum_{i=1}^m \|k_i\|_1^2}$, L is a Lipschitz smooth constant of f. Choose $\beta^{(0)} \in \mathcal{X}$, $\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_m^{(0)}) \in \mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \cdots \times \mathcal{U}_m$.
Iterate: for $t = 0, 1, 2, \ldots$
Update primal variable: $\beta^{(t+1)} = \mathbf{prox}_{\frac{g}{L+\zeta}}\left(\beta^{(t)} - 1/(L+\zeta)[\nabla f(\beta^{(t)}) + \mathcal{C}_{[m]}^H \mu^{(t)}]\right)$.
Update dual variable: $\mu^{(t+1)} = \Pi_{\mathcal{B}_\infty}(\mu^{(t)} + (1/\zeta)\mathcal{C}_{[m]}(2\beta^{(t+1)} - \beta^{(t)}))$.

Convergence Analysis and Complexity Bounds

In this section, we present our convergence analysis of the proposed algorithm. As mentioned before, the proposed algorithm does not follow the step-size assumption in (Condat 2013; Chambolle and Pock 2014), so the analysis therein is not suitable for the proposed algorithm. Furthermore, combining (Condat 2013) and (Chambolle and Pock 2014), we only have the results in the weak convergence and ergodic rate analysis. We hereby extend the convergence and ergodic results to our proposed algorithm and provide the non-ergodic rate analysis, which is the first result in such primal-dual sparse learning algorithm.

First, we show in Theorem 1, following the step size settings in Algorithm 1, the weak convergence to the primal-dual optima pair is guaranteed.

Theorem 1 (Convergence)**.** *Consider procedure primal-dual update for convolutional sparsity given in Algorithm 1, following the step size pair* $(1/(L+\zeta), 1/\zeta)$ *where* L *is the Lipschitz smooth constant of* f*, and* $\zeta = \sqrt{\sum_{i=1}^m \|k_i\|_1^2}$*. Then, there exists a pair* $(\beta^*, \mu^*) \in \mathcal{X} \times \mathcal{U}$ *solution to* (8)*, such that the sequences* $\{\beta^{(t)}\}$ *and* $\{\mu^{(t)}\}$ *generated by Algorithm 1 weakly converge to* β^* *and* μ^**, respectively.*

We next present the complexity bounds for the Algorithm 1 in terms of both non-ergodic rate and ergodic rate. The Algorithm 1 is shown to maintain a non-ergodic rate of $\mathcal{O}(1/\varepsilon^2)$, which is optimal for the non-smooth convex optimization (Nesterov 2013).

Theorem 2 (Non-ergodic Rate)**.** *Let* $\{\beta^{(t)}\}$ *and* $\{\mu^{(t)}\}$ *be the sequences generated by Algorithm 1,* $[\beta^*, \mu^*]$ *being the optima of problem* (8)*. Let* $\zeta = \sqrt{\sum_{i=1}^m \|k_1\|_1^2}$ *and* $\varepsilon > 0$*. Then, for every* $T > 0$ *such that*

$$T \geq \left\lceil \frac{3}{\sqrt{2}\varepsilon^2}\left[(L+\zeta)\|\beta^* - \beta^{(0)}\|_2^2 + \zeta\|\mu^* - \mu^{(0)}\|_2^2 - 2\langle \mathcal{C}_{[m]}(\beta^* - \beta^{(0)}), \mu^* - \mu^{(0)}\rangle\right]\right\rceil, \tag{11}$$

we are guaranteed that $P(\beta^{(T)}) - D(\mu^{(T)}) \leq \varepsilon$.

In the sequel, the ergodic convergence rate is revealed of the proposed algorithm, which is similar with that in (Chambolle and Pock 2014). However, due to the complexity reduction in step-size computation and the more aggressive step size setting, our algorithm usually performs much faster in practice, as shown in Experiment Section.

Theorem 3 (Ergodic Rate)**.** *Let* $\{\beta^{(t)}\}$ *and* $\{\mu^{(t)}\}$ *be the sequences generated by Algorithm 1,* $[\beta^*, \mu^*]$ *being the optima of problem* (8)*. Let* $\bar{\beta}^{(t)} = \sum_{i=1}^{t} \beta^{(i)}/t$*,* $\bar{\mu}^{(t)} = \sum_{i=1}^{t} \mu^{(i)}/t$ *and* $\zeta = \sqrt{\sum_{i=1}^{m} \|k_1\|_1^2}$*. Let* $\varepsilon > 0$*, thus for every* $T > 0$ *such that*

$$T \geq \left\lceil \frac{1}{2\varepsilon} \left[(L+\zeta)\|\beta^* - \beta^{(0)}\|_2^2 + \zeta\|\mu^* - \mu^{(0)}\|_2^2 - 2\langle \mathcal{C}_{[m]}(\beta^* - \beta^{(0)}), \mu^* - \mu^{(0)} \rangle \right] \right\rceil, \quad (12)$$

we are guaranteed that $P(\bar{\beta}^{(T)}) - D(\bar{\mu}^{(T)}) \leq \varepsilon$*.*

According to the submission guideline of AAAI conference, we provide the long proof of these theorems in the supplementary material.

Numerical Results

In this section, we demonstrate the effectiveness and efficiency of our method by conducting the numerical experiments on two most recent sparse learning models, fused LASSO and joint Total Variation and Nuclear Norm regularization (TVNN), that fits in convolutional sparse framework. All the experiments in this section are conducted on a desktop computer with Intel Core i7-4770 CPU and 16 gigabyte RAM. All methods are evaluated in MATLAB 2013b, Windows 7 Enterprise.

Fused LASSO

Here, we consider the following fused LASSO problem, which has been extensively used for sparse learning and feature selection; see also (Tibshirani et al. 2005; Xin et al. 2014),

$$\min_{\beta} \frac{1}{2}\|X\beta - y\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\nabla\beta\|_1. \quad (13)$$

In this experiment, we hereby set $X \in \mathbb{R}^{n\times p}$ as a normal Gaussian matrix, and the smallest Lipschitz smooth constant of $f(\beta)$ has been estimated as $\mathcal{O}((\sqrt{n}+\sqrt{p})^2)$ according to (Rudelson and Vershynin 2010). $\beta \in \mathbb{R}^p$ is randomly generated as ground truth. We use the exact same parameter setting as in (Liu, Yuan, and Ye 2010), i.e., $\lambda_1 = 0.001, \lambda_2 = 0.01$. Due to the randomness of the data simulation, the average results of 100 repeating experiments are shown in Table 1 and Table 2.

Table 1: Time Comparison in Fused LASSO Experiment

n/p	SLEP	C& P	CVX	fGFL	Proposed
100/1000	0.1332	0.3437	1.6412	0.7669	**0.0850**
200/1000	0.3265	0.8178	8.2135	14.2822	**0.1042**
500/1000	0.2428	0.4389	19.6028	16.3207	**0.1601**
100/2000	0.2281	1.0858	3.4152	2.6651	**0.1102**
200/2000	0.2653	1.1479	10.0197	47.3517	**0.1545**
500/2000	0.5729	1.4796	54.9067	54.0380	**0.4316**
100/5000	0.6746	6.6807	8.7366	223.9324	**0.3094**
200/5000	0.9702	7.1142	32.1315	235.0583	**0.6959**
500/5000	1.8248	7.9889	146.1687	15.0901[2]	**1.5480**

[2]The fGFL method performs surprisingly fast and effective in 500/5000 setting. We don't know the reason of this behavior.

We compare our method with SLEP (Liu, Ji, and Ye 2009)[3], Chambolle & Pock (hereafter C & P) (Chambolle and Pock 2014), CVX (Grant and Boyd 2014) and fGFL (Xin et al. 2014). We show, in Table 1, the running time comparison of different methods. The proposed method is able to outperform all other compared state-of-the-art methods in solving fused LASSO formulation in terms of efficiency. The per-iteration complexity of SLEP is in the same order as our methods, but it employs an iterative line search procedure to search suitable step size, which makes it slightly slower than our methods. CVX is a complex general framework which is less optimized for this specific formulation. fGFL takes too much time generating the graph structure in the fused LASSO problem. Chambolle and Pock's method shares similar primal-dual technique with ours but differs in step size setting. Our method uses only linear time complexity in estimating the step sizes while Chambolle and Pock's method runs at the cubic time complexity. This experiment clearly shows that, as the problem scale (i.e., p) grows, the time cost of Chambolle and Pock increases more rapidly than that in the proposed method.

Table 2: Relative Error in Fused LASSO Experiment

n/p	SLEP	C & P	CVX	fGFL	Proposed
100/1000	1.2089	**0.9395**	1.2189	0.9397	**0.9395**
200/1000	1.1354	**0.8862**	1.1393	0.8898	**0.8862**
500/1000	0.9230	**0.6867**	0.9462	0.6891	**0.6867**
100/2000	1.1609	**0.9721**	1.1707	0.9722	**0.9721**
200/2000	1.1448	**0.9523**	1.1488	0.9533	**0.9523**
500/2000	1.1333	**0.8636**	1.1495	0.8662	**0.8636**
100/5000	1.1254	**0.9903**	1.1410	0.9911	**0.9903**
200/5000	1.1426	**0.9837**	1.1478	0.9845	**0.9837**
500/5000	1.2111	**0.9434**	1.2229	**0.9434**	**0.9434**

Table 2 exhibits the accuracy of different methods in terms of relative error which is $\|\beta^* - \beta^{(T)}\|_2/\|\beta^*\|_2$, where β^* is the ground truth vector and $\beta^{(T)}$ is for stopping result of specific optimization method. The SLEP method uses a heuristic line search technique, which assumes that the Lipschitz smooth constant L of function f is always larger than one, resulting in its inability to exactly solve the original problem (13). CVX requires much more iterations to solve the problem accurately. fGFL can solve the fused LASSO objective into a comparable accuracy of the proposed method's solution but is much more costly in overall runtime. Chambolle and Pock's algorithm is always able to reach almost same accuracy as proposed method since they have the same primal-dual problem structure. However, due to the higher complexity in step size estimation, the proposed algorithm is faster than the Chambolle and Pock's, and becomes even faster and faster when problem scale grows larger. Hence, the accuracy of proposed method is superior to those state-of-the-art methods designed for the fused LASSO model.

[3]SLEP (Liu, Ji, and Ye 2009) is the toolbox containing the code for algorithm described in (Liu, Yuan, and Ye 2010). We follow the instruction from the author of (Liu, Yuan, and Ye 2010) to cite (Liu, Ji, and Ye 2009) instead.

Joint Total Variation and Nuclear Norm Regularization

In this subsection, we show another application of Algorithm 1. Consider the recovery problems regularized by both total variation norm and nuclear norm (TVNN), which has been recently studied in many computer vision applications (Shi et al. 2013; Yao et al. 2015) . It is written as the following optimization problems,

$$\min_{\beta} \frac{1}{2}\|X\beta - y\|_F^2 + \lambda_1\|\beta\|_* + \lambda_2\|\nabla\beta\|_1. \qquad (14)$$

Here, β has two spatial dimensions (i.e., a matrix), with X serving as a linear bounded subsampling operator. $\|\beta\|_* = \sum_i \sigma_i(\beta)$ is the sum of *all singular values* of β, which is also known as the **nuclear norm**. ∇, in this problem, refers to a discrete gradient operator in both horizontal and vertical directions. This formulation fits in our framework, applying Algorithm 1, by letting $f(\beta) = (1/2)\|X\beta - y\|_F^2$, $g(\beta) = \lambda_1\|\beta\|_*$ and setting $k_1 = [1, -1]$, $k_2 = [1, -1]^T$ and $\nabla := \mathcal{C}_{[2]} = \mathcal{C}_{k_1,k_2}$. The Lipschitz smooth constant L of f is set to 1 since $\|X\beta\|_F^2 \leq \|\beta\|_F^2, \forall\beta \in \mathcal{X}$. The computation of $\mathbf{proj}_g$ is through the *singular value thresholding* due to (Goldfarb and Ma 2011; Ma, Goldfarb, and Chen 2011). The parameter setting is $\lambda_1 = 100, \lambda_2 = 1$, following the same setting in (Huang et al. 2011).

In this experiment, we compare our method with ADMM_TVNN (Shi et al. 2013) and FCSA_TVNN (Huang et al. 2011), solving the same formulation. Figure 1 shows the convergence behavior of these three methods in terms of iteration-loss, time-loss, iteration-PSNR (Peak Signal-to-Noise Ratio) and time-PNSR curves. ADMM_TVNN converges much slower than FCSA_TVNN and the proposed method, due to its possibly arbitrarily slow convergence behavior (Boyd et al. 2011). FCSA_TVNN is not an exact algorithm and hence its stopping PSNR is higher. It also has higher per-iteration complexity, since its subproblem requires iterative solution, which leads to slower convergence in terms of runtime. This experiment demonstrates the superior performance of the proposed algorithm compared with state-of-the-art algorithms in solving such complicated formulation as (14) in computer vision field.

Discussion

As an algorithm designed for the generalized framework, Algorithm 1 is still able to improve the state-of-the-art results for fused LASSO and TVNN regularized recovery, even outperforming methods that are highly optimized for specific formulations, e.g., SLEP (Liu, Ji, and Ye 2009) for fused LASSO problem. Our method employs several analytic, computationally efficient results of composite convolution operators to accelerate the optimization process, particularly in step-size settings. Besides, our algorithm has lower per-iteration complexity since every step in each iteration can be explicitly evaluated with low complexity. Since the primal and dual objectives of our method are updated exactly, our method is able to reach a higher overall accuracy.

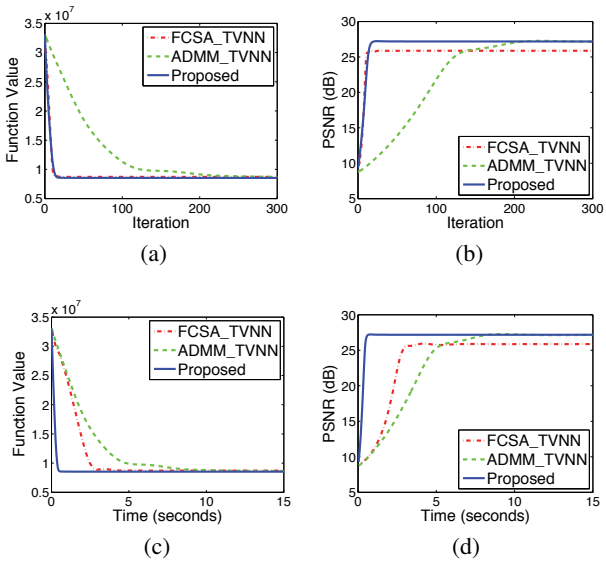

Figure 1: The convergence speeds of ADMM_TVNN (Shi et al. 2013), FCSA_TVNN (Huang et al. 2011) and the proposed method.

Conclusion

The first contribution of this paper is the convolutional sparse modeling. We find a middle ground between standard sparsity and dictionary sparsity, sharing the benefits from both sides, i.e., the computational efficiency and the less requirement of the measurement number. Also, convolutional sparse modeling has the potential to include more existing sparse learning works, e.g. (Chen and Huang 2012; Xu et al. 2015). Furthermore, one might use it to discover some unveiled convolutional sparse learning models. To our best knowledge, it is the first work to model a class of sparse constraints with convolution operators.

Our second contribution is theoretical. We analyze the spectral properties of the composite convolution operator and apply the decent property onto the design of the optimization algorithm to reduce the complexity significantly. While there is no specific convergence analysis of the resulting algorithm, we complete our study by providing these studies. Under the new design of the proposed generic algorithm, we prove that the convergence to the optima is guaranteed and the complexity bounds maintain $\mathcal{O}(1/\varepsilon^2)$ and $\mathcal{O}(1/\varepsilon)$ in terms of non-ergodic and ergodic convergence rates. While the convergence theorem and the ergodic result are gentle extensions to (Condat 2013; Chambolle and Pock 2014), our non-ergodic rate result is the first to appear in the sparse learning literature.

Acknowledgment

This work was partially supported by U.S. NSF IIS-1423056, CMMI-1434401, CNS-1405985 and the NSF CAREER grant IIS-1553687.

References

Bayram, I., and Kamasak, M. E. 2012. Directional total variation. *Signal Processing Letters, IEEE* 19(12):781–784.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Cai, X.; Gu, G.; and He, B. 2014. On the o (1/t) convergence rate of the projection and contraction methods for variational inequalities with lipschitz continuous monotone operators. *Computational Optimization and Applications* 57(2):339–363.

Candes, E. J.; Eldar, Y. C.; Needell, D.; and Randall, P. 2011. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis* 31(1):59–73.

Candes, E. J.; Romberg, J. K.; and Tao, T. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics* 59(8):1207–1223.

Chambolle, A., and Pock, T. 2014. On the ergodic convergence rates of a first-order primal-dual algorithm. *preprint*.

Chan, R. H.; Liang, H.; Wei, S.; Nikolova, M.; and Tai, X.-C. 2015. High-order total variation regularization approach for axially symmetric object tomography from a single radiograph. *Inverse Problems and Imaging* 9(1):55–77.

Chen, C., and Huang, J. 2012. Compressive sensing mri with wavelet tree sparsity. In *Advances in neural information processing systems*, 1115–1123.

Combettes*, P. L. 2004. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization* 53(5-6):475–504.

Condat, L. 2013. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications* 158(2):460–479.

Goldfarb, D., and Ma, S. 2011. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics* 11(2):183–210.

Grant, M., and Boyd, S. 2014. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

He, B., and Yuan, X. 2012. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences* 5(1):119–149.

Heide, F.; Heidrich, W.; and Wetzstein, G. 2015. Fast and flexible convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 5135–5143. IEEE.

Huang, J.; Zhang, S.; Li, H.; and Metaxas, D. 2011. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding* 115(12):1610–1622.

Huang, J.; Zhang, T.; and Metaxas, D. 2011. Learning with structured sparsity. *The Journal of Machine Learning Research* 12:3371–3412.

Huang, J.; Zhang, T.; et al. 2010. The benefit of group sparsity. *The Annals of Statistics* 38(4):1978–2004.

Lenzen, F.; Becker, F.; and Lellmann, J. 2013. *Adaptive second-order total variation: An approach aware of slope discontinuities*. Springer.

Liu, J.; Ji, S.; and Ye, J. 2009. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University.

Liu, J.; Yuan, L.; and Ye, J. 2010. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 323–332. ACM.

Liu, J.; Yuan, L.; and Ye, J. 2013. Dictionary lasso: Guaranteed sparse recovery under linear transformation. *arXiv preprint arXiv:1305.0047*.

Ma, S.; Yin, W.; Zhang, Y.; and Chakraborty, A. 2008. An efficient algorithm for compressed mr imaging using total variation and wavelets. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8. IEEE.

Ma, S.; Goldfarb, D.; and Chen, L. 2011. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming* 128(1-2):321–353.

Nesterov, Y. 2013. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Rudelson, M., and Vershynin, R. 2010. Non-asymptotic theory of random matrices: extreme singular values. *arXiv preprint arXiv:1003.2990*.

Shi, F.; Cheng, J.; Wang, L.; Yap, P.-T.; and Shen, D. 2013. Low-rank total variation for image super-resolution. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 155–162.

Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(1):91–108.

Toma, A.; Sixou, B.; Denis, L.; Pialat, J.-B.; and Peyrin, F. 2014. Higher order total variation super-resolution from a single trabecular bone image. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, 1152–1155. IEEE.

Xin, B.; Kawahara, Y.; Wang, Y.; and Gao, W. 2014. Efficient generalized fused lasso and its application to the diagnosis of alzheimer's disease. In *AAAI*, 2163–2169. Citeseer.

Xu, Z.; Li, Y.; Axel, L.; and Huang, J. 2015. Efficient preconditioning in joint total variation regularized parallel mri reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 563–570. Springer.

Yao, J.; Xu, Z.; Huang, X.; and Huang, J. 2015. Accelerated dynamic mri reconstruction with total variation and nuclear norm regularization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 635–642. Springer.