Learning Bilevel Proximal Programs for Joint Feasibility and Optimality Pursuit

Paper ID: 920

Abstract

Classical optimization techniques often formulate the feasibility of the problems as set, equality or inequality constraints. However, explicitly designing these constraints is indeed challenging for complex real-world applications and too strict constraints may even lead to intractable optimization problems. On the other hand, it is still hard to incorporate data-dependent information into conventional numerical iterations. To partially address the above limits and inspired by the leader-follower gaming perspective, this work first introduces a bilevel-type formulation to jointly investigate the feasibility and optimality of nonconvex and nonsmooth optimization problems. Then we develop an algorithmic framework to couple forward-backward proximal computations to optimize our established bilevel leader-follower model. We prove its convergence and estimate the convergence rate. Furthermore, a learning-based extension is developed, in which we establish an unrolling strategy to incorporate *data-dependent network architectures* into our iterations. Fortunately, it can be proved that by introducing some mild checking conditions, all our original convergence results can still be preserved for this learnable extension. As a nontrivial byproduct, we demonstrate how to apply this *ensemble-like methodology* to address different low-level vision tasks. Extensive experiments verify the theoretical results and show the advantages of our method against existing state-of-the-art approaches.

Introduction

In standard optimization programs, there is only one single model who tries to figure out an optimal solution subject to a certain set of constraints such that their objectives reach the best. Following this perspective, many vision and learning tasks are formulated as the following optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{x}) := g(\mathbf{x}) + \psi(\mathbf{x}), \tag{1}$$

where functions g and ψ typically capture the loss of data fitting and the regularization, respectively. In this work, we assume that the loss g is smooth, while the regularization ψ can be nonsmooth and the problem (1) is a nonconvex problem. Naturally, many optimization problem have constraints, but it is hard to find the physical constraints. Since the feasibility of $\mathbf{x}$ often plays a very important role in real problems, traditional constraint modeling strategies have to exactly formulate constraint set to meet their particular focuses. Unfortunately, explicitly formulating the feasibility of the task is challenging. Even worse, enforcing complex constraints will significantly increase the difficulty of optimization or lead to intractable problems.

Related Works

Introducing constraints to energy optimization models is a common and intuitive idea to narrow down the solution space when solving particular learning and vision problems. The most widely used constraints include sets, equalities or inequalities (Dechter and Cohen 2003). However, designing correct and exact constraints for real-world applications are often challenging. Moreover, a lot of additional efforts must be made to handle these constrained structures, thus leading to complex iteration schemes. Even worse, the convergence of these iteration behaviors for constrained optimization algorithms are also very hard to be guaranteed. For example, it is well known that the convergence of standard alternating direction methods of multipliers (ADMM) can only be guaranteed for problems with simple linear equality constraints and no more than two variables (Boyd 2011). All the above limitations inspired us to develop new paradigms to formulate implicit constraints for nonconvex optimization models in real-world learning and vision applications.

Bilevel Optimization (BO) is known as a mathematical program, in which there are typically two individual models, i.e., the upper-level and lower-level subproblems (Bennett et al. 2008). BOs were commonly used in a number of real-world problems, including economics, decision science, and engineering, etc (Bard 2013). Recently, there are many practical applications of BO in learning and vision areas. For example, the works in (Chen and Pock 2017) utilized bilevel models to learn parameters of higher-order Markov random field regularization for various image-related tasks. Gradient based methods for solving the BO problem are considered for the task of multi-label segmentation (Ochs et al. 2016). A bilevel variational model is developed in (De los Reyes and Schönlieb 2013) for learning a good image denoising model. The work in (Liu et al. 2013) reformulated BO as PDE-constrained optimal control to addressed different vi-

sion tasks within a single diffusion model. Indeed, the training phase of convolutional neural network (CNN) can also be reformulated within the BO framework. In (Beck and Teboulle 2009), authors proposed a cutting-plane algorithm for bilevel models with strongly convex and differential upper objective function and convex lower subproblem. This algorithm is actually not feasible in the sense that the generated sequence does not necessarily belong to the constraint set of the model. Recent studies in (Sabach and Shtern 2017) improved the above cutting-plane scheme by sequential averaging method. In the work, we will introduce a BO perspective to design energy-based feasibility constraints and more efficient optimization algorithms to Eq. (1).

Our Contributions

Drawing on the idea of bilevel optimization (Ye and Zhu 2010) and the well-known Stackelberg's leader-follower competition theory in economics (Von Stackelberg 2010)[1], we first introduce a flexible energy function model to characterize feasibility constraint that incorporate implicit constraints into optimization models to jointly investigate the feasibility and optimality for challenging real-world applications. Specifically, we first assume that the objective and constraint in optimization models are two players in a game, we then consider our objective as a leader, that tries to optimize the next move under consideration of the move of the follower (i.e., constraints). By formulating these two players (i.e., leader and follower) as two composite minimization subproblems, we establish a Joint Feasibility and Optimality Pursuit (JFOP for short) formulation as follows

$$\begin{aligned} \texttt{Leader}\ (\texttt{L}): \quad & \min_{\mathbf{x}} \Psi(\mathbf{x}) := g(\mathbf{x}) + \psi(\mathbf{x}), \\ \texttt{Follower}\ (\texttt{F}): \quad & s.t.\ \mathbf{x} \in \arg\min_{\mathbf{x}} F(\mathbf{x}) := f(\mathbf{x}) + \phi(\mathbf{x}). \end{aligned} \tag{2}$$

It can be seen that (L) in Eq. (2) is just the original objective in Eq. (1), while we actually formulate the constraint as a follower optimization model (denoted as (F)) to implicitly formulate our feasibility of the task in the game. Similar to that in (L) subproblem, f and ϕ in (F) that can be nonconvex also respectively represent the fidelity and prior for the pursuit of feasibility.

Then, we develop Bilevel Proximal Programs (BPP) scheme to couple both convex and nonconvex forward backward proximal computations to address the general nonsmooth and nonconvex JFOP problem. We strictly prove the global convergence and estimate the convergence rate of BPP. Furthermore, we develop a practical learning feasibility constraint methodology, named Learning BPP (LBPP), to incorporate deeply trained network architectures (as implicit data-dependent constraints) to address challenging computer vision problems, such as image deblurring, super-resolution and inpainting. In summary, the contributions of this paper mainly include:

- We propose a novel leader-follower perspective (i.e., JFOP) to introduce implicit feasibility constraints for nonconvex and nonsmooth optimization. By unrolling the additional follower iterations using data-dependent deep architectures, we also develop an algorithmic framework to learn constraints from data to speed up the iterations and explore task-related local optimal solutions.
- We prove in theory that the proposed BPP framework can generate globally converged sequence to the original optimization model in Eq. (1) and estimate the convergence rate, thus verify the correctness and effectiveness of the proposed JFOP paradigm. By introducing some mild checking conditions, the same theoretical results can also be proved for the data-dependent LBPP approach.
- As a nontrivial byproduct, we finally show how to apply JFOP with BPP to solve optimization model Eq. (1) in particular applications. We also consider JFOP as a flexible ensemble framework to incorporate task priors and learnable architectures to address different real-world computer vision tasks. Extensive experiments show that our method achieves state-of-the-art results on all the tested problems.

The Proposed Algorithm

In this section, we first develops a linearly coupled proximal scheme, i.e., Bilevel Proximal Programs (BPP), to address the nonconvex JFOP problem and analysis the convergence of this scheme. Then, we extend our BPP to a learning-based scheme, named LBPP.

Bilevel Proximal Programs

In general, the leader subproblem in Eq. (2) is a composite optimization model, thus we just adopt standard Proximal Gradient (PG) scheme to update it as follows

$$\mathcal{L}_s\left(\mathbf{x}\right) := \texttt{prox}_{s\psi}(\mathbf{x} - s\nabla g(\mathbf{x}))\,,$$

where $\texttt{prox}_{s\psi}(\mathbf{x}) = \arg\min_{\mathbf{y}} \left\{\psi(\mathbf{y}) + \frac{1}{2s}\|\mathbf{y}-\mathbf{x}\|^2\right\}$. Within our JFOP framework, it is also necessary to address the optimization issues for the follower subproblem. Here we just denote the updating of this subproblem at k-th iteration as $\mathcal{F}(\mathbf{x}^k)$ and the following theoretical results will guarantee that by investigating some very mild checking conditions, we actually can adopt any first-order methods (e.g., PG or ADMM) to solve the follower optimization model. With $\mathcal{L}_s$ and $\mathcal{F}$, the formal updating rule for the JFOP optimization model in Eq. (2) can be written as the following adaptive proximal linear coupling:

$$\mathbf{x}^{k+1} = \alpha^k \mathcal{F}(\mathbf{x}^k) + (1-\alpha^k)\mathcal{L}_s(\mathbf{x}^k),$$

where $\{\alpha^k\}$ is a sequence of real numbers in the range $[0, 1)$ and will be analyzed and determined later. We summarize our complete iteration in Alg. 1.

Then we would like to discuss the convergence behaviors for the proposed BPP algorithm. Let us recall some standard definitions in variational analysis (Rockafellar and Wets 2009). The limiting subdifferential of a given function Ψ is denoted as $\partial\Psi$. We also suggest readers to refer

[1]The Stackelberg competition theory is referring to a strategic game in economics in which the leader firm moves first and then the follower firms move sequentially. One may refer to (Von Stackelberg 2010) for more details.

Algorithm 1 Bilevel Proximal Programs

Require: The input $\mathbf{x}^0$, parameters $s \in (0, 1/L^g)$, $t \in (0, 1/L^f)$ and $\{\alpha^k | \alpha^k \in [0, 1)\}$.

1: **while** not converged **do**
2: $\quad \mathbf{f}^{k+1} \in \mathcal{F}(\mathbf{x}^k)$ and $\mathbf{l}^{k+1} \in \mathcal{L}_s(\mathbf{x}^k)$.
3: $\quad \mathbf{z}^{k+1} = \alpha^k \mathbf{f}^{k+1} + (1 - \alpha^k)\mathbf{l}^{k+1}$.
4: $\quad$ **if** $\Psi(\mathbf{z}^{k+1}) \leq \Psi(\mathbf{l}^{k+1})$ **then**
5: $\quad\quad \mathbf{x}^{k+1} = \mathbf{z}^{k+1}$.
6: $\quad$ **else**
7: $\quad\quad \mathbf{x}^{k+1} = \mathbf{l}^{k+1}$.
8: $\quad$ **end if**
9: **end while**

to (Rockafellar and Wets 2009) for some other properties, such as proper, lower-semicontinuous, coercive and metrically subregular, which will be useful in the following analysis. Our convergence analyses are also based on the following fairly loose assumptions.

Assumption 1. *The object function in Eq.* (1) *should satisfy:*

1. *The function Ψ is proper, lower-semicontinuous and coercive function.*
2. *$g(\mathbf{x})$ is Lipschitz smooth, i.e., for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, we have*

$$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\| \leq L^g \|\mathbf{x} - \mathbf{y}\|,$$

where L^g is the Lipschitz constants for ∇g.

Then we prove the following theorem to summarize our main convergence results for BPP[2].

Theorem 1. *Suppose that $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ be an iteration sequence generated by Alg. 1 and the Assumption 1 holds. Then, the iterations sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ is bounded and there exists subsequences $\{\mathbf{x}^{k_q}\}_{q\in\mathbb{N}}$, such that it converges to the critical point (denoted as $\mathbf{x}^*$) of the minimization problem* (1).

Remark 1. *The boundness of sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ can be directly obtained from Assumption 1. So $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ has convergent subsequences. To guarantee the convergence, our BPP ensures that the objective function is sufficiently reduced in each iteration:*

$$\Psi(\mathbf{x}^{k+1}) \leq \Psi(\mathbf{x}^k) - \left(\frac{1}{2s} - \frac{L^g}{2}\right)\|\mathbf{l}^{k+1} - \mathbf{x}^k\|^2. \quad (3)$$

Remark 2. *If we further assume that $\sum_{k=1}^{\infty} \alpha^k < \infty$, and F is proper, lower-semicontinuous and coercive function and f is continuously differentiable with the Lipschitz constant L^g, then with the semi-algebraic property[3], we can further have that the sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ has finite length, i.e.,*

$$\sum_{k=1}^{\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| < \infty. \quad (4)$$

[2]Due to space limits, we move the proofs of all our theoretical results to the Supplemental Materials.

[3]Please refer to (Bolte, Sabach, and Teboulle 2014) for the formal definition of semi-algebraic function. Actually, many functions arising in learning and vision areas, including ℓ_0 norm, rational ℓ_p norms (i.e., $p = p_1/p_2$ with positive integers p_1 and p_2) and their finite sums or products, are all semi-algebraic.

Thus we can prove that the sequence $\{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|\}_{k\in\mathbb{N}}$ is summable, i.e., there exists $m > n > l$ such that

$$\|\mathbf{x}^m - \mathbf{x}^n\| \leq \sum_{k=n}^{m-1} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| \to 0,$$

as $l \to \infty$. Then it follows that $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ is a Cauchy sequence and hence is a globally convergent sequence.

By introducing a desingularizing function as that in (Bolte, Sabach, and Teboulle 2014), we can further estimate the convergence rate of BPP in the following corollary based on the results proved in Theorem 1.

Corollary 1. *Define a desingularising function $\varphi(t) = ct^{1-\theta}$, where $c > 0$ is a given constant. Then there exists k large enough satisfying: (i) If $\theta = 0$, then the sequence $\{\mathbf{x}^k\}$ convergences in a finite number of steps. (ii) If $\theta \in (0, \frac{1}{2}]$, then there exist $w_1 > 0$ and $\tau \in [0, 1)$ such that $\|\mathbf{x}^k - \mathbf{x}^*\| \leq w_1\tau^k$. (iii) If $\theta \in (\frac{1}{2}, 1)$, then there exist $w_2 > 0$ such that $\|\mathbf{x}^k - \mathbf{x}^*\| \leq w_2 k^{-\frac{1-\theta}{2\theta-1}}$.*

Learning-based BPP

In this section, we aim to develop Learning-based BPP (LBPP) to generate data-dependent and task-specific deep models. That is, we would like to incorporate specially designed network architectures into each charaterized constraint iteration to further improve the practical performance of BPP for particular applications. LBPP is more suitable for specific problems and provide an reliable method for data-dependent tasks. We will prove that the convergence results can still be guaranteed only under some mild conditions. As a nontrivial byproduct, we also establish a new JFOP formulation to address a variety of low-level vision applications.

Specifically, the network-based building block at the k-th iteration is denoted as $\mathcal{D}(\cdot; \boldsymbol{\theta}^k)$, where $\boldsymbol{\theta}^k$ are network parameters[4]. Then we calculate a temporary variable $\tilde{\mathbf{x}}^k = \mathcal{D}(\mathbf{x}^k; \boldsymbol{\theta}^k)$. By further considering data-dependent form $F_\mu^k(\mathbf{x}) := F(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{x} - \tilde{\mathbf{x}}^k\|^2$ as the proximal approximation of $F(\mathbf{x})$ (with parameter $\mu > 0$) at the k-th iteration, we actually update the follower subproblem in Eq. (2) by standard proximal gradient (PG) scheme in the follower subproblem which can be described as

$$\mathcal{F}_t^{\mathcal{D}}(\tilde{\mathbf{x}}^k) = \mathtt{prox}_{s\phi}\left(\tilde{\mathbf{x}}^k - s(\nabla f(\tilde{\mathbf{x}}^k) + \mu(\tilde{\mathbf{x}}^k - \mathbf{x}^k))\right). \quad (5)$$

Then we summarize the complete LBPP scheme in Alg 2. It can be observed that we actually introduce a condition (in Steps 4 of Alg 2) to guide the updating of LBPP. This is just to prevent any improperly designed/trained architectures, which may deflect our iterative trajectory towards unwanted solutions.

Obviously, the objective function $\Psi(\mathbf{x})$ is sufficiently descent in Alg. 2 and we can obtain the same result as stated in Theorem 1. The following remark is given to further understand Alg. 2.

[4]Please refer to the next section for the detailed structures of $\mathcal{D}$ in real-world applications.

Algorithm 2 Learning-based BPP

Require: The input $\mathbf{x}^0$, parameters $s \in (0, 1/L^g)$, $t \in (0, 1/L^f)$, $\{\alpha^k | \alpha^k \in [0, 1)\}$ and $C > 0$.

while not converged **do**
  $\mathbf{l}^{k+1} \in \mathcal{L}_s(\mathbf{x}^k)$.
  $\mathbf{f}^{k+1} \in \mathcal{F}_t^{\mathcal{D}}(\tilde{\mathbf{x}}^k)$ where $\tilde{\mathbf{x}}^k = \mathcal{D}(\mathbf{x}^k; \boldsymbol{\theta}^k)$.
  if $\|\tilde{\mathbf{x}}^k - \mathbf{x}^k\| \leq C\|\mathbf{f}^{k+1} - \mathbf{x}^k\|$ **then**
    $\mathbf{z}^{k+1} = \alpha^k \mathbf{f}^{k+1} + (1 - \alpha^k)\mathbf{l}^{k+1}$.
  else
    $\mathbf{f}^{k+1} = \mathcal{F}_t(\mathbf{x}^k)$.
    $\mathbf{z}^{k+1} = \alpha^k \mathbf{f}^{k+1} + (1 - \alpha^k)\mathbf{l}^{k+1}$
  end if
  if $\Psi(\mathbf{z}^{k+1}) \leq \Psi(\mathbf{l}^{k+1})$ **then**
    $\mathbf{x}^{k+1} = \mathbf{z}^{k+1}$.
  else
    $\mathbf{x}^{k+1} = \mathbf{l}^{k+1}$.
  end if
end while

Corollary 2. *Suppose that the Assumption 1 holds. Then the iterations sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ generated by Alg. 2 is bounded and there exists subsequences $\{\mathbf{x}^{k_p}\}_{p\in\mathbb{N}}$ convergence to the critical point of the problem* (1).

Remark 3. *The deep iteration of $\{\mathbf{f}^{k+1}\}$ is a data-dependent scheme which learning the implicit feasibility constraints in an explicit form. It is easy to check that the convergence rate of LBPP can also be estimated in the same manner as that stated in BPP. Thus we do not repeatedly state these results in this part.*

Applications in Computer Vision

In this section, we first demonstrate how to introduce follower constraint energy within JFOP framework to efficiently optimize nonconvex nonsmooth Eq. (1) in specific applications. Furthermore, by following the perspective of LBPP, we also construct an optimization driven deep model to address various real-world low-level vision tasks.

Nonconvex Optimization via BPP

Here we consider a particular non-blind deconvolution problem, which aims to recover latent image $\mathbf{x}$ from blurred observation $\mathbf{b}$. By formulating this problem using sparse coding model $\mathbf{b} = \mathbf{D}\boldsymbol{\beta} + \mathbf{n}$, where $\boldsymbol{\beta}$ denotes the sparse code, $\mathbf{D}$ is a given dictionary[5] and $\mathbf{n}$ is the unknown noises, we have a specific case of Eq. (1) as follows

$$\min_{\boldsymbol{\beta}} \|\mathbf{b} - \mathbf{D}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_p, \tag{6}$$

where $p \in (0, 1)$, $\lambda > 0$.

We emphasize that different from these existing image modeling techniques, which only consider single-level fidelity and prior, JFOP allows us to introduce bilevel principles (i.e., leader and follower) to respectively formulate the optimality and feasibility of the tasks. As for the follower subproblem in Eq. (2), since it is related to the feasibility of tasks, we aim to just introduce a relatively simple but well-defined model to enforce our distribution assumptions on the latent image. Then, we introduced the widely used Total Variation (TV) model (Osher et al. 2005) as feasible constraint in the follower subproblem. The JFOP model can be described as

$$\begin{aligned} &\min_{\mathbf{x}} \|\mathbf{BWx} - \mathbf{b}\|^2 + \lambda_1\|\mathbf{W}^\top\mathbf{x}\|_p, \\ &s.t.,\ \mathbf{x} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{k} \otimes \mathbf{x} - \mathbf{b}\|^2 + \lambda_2 \textstyle\sum_{j\in\{h,v\}} \|\nabla_j\mathbf{x}\|_q, \end{aligned} \tag{7}$$

where λ_2 are thresholding parameters, $q \in (0, 1)$ and ∇_h and ∇_v respectively denote the gradient on the horizontal and vertical directions.

LBPP-based Ensemble Framework

Thanks to the flexible bilevel structure, we can also directly utilize our JFOP formulation together with LBPP algorithm to design ensemble frameworks for computer vision tasks (e.g., image restoration and enhancement).

It is well-known that both the analysis and synthesis models can be used to address image enhancement and restoration tasks. However, these two categories of methods always have their respective pros and cons. So it is natural to integrate the analysis and synthesis mechanisms to develop new algorithms for the above mentioned low-level vision tasks. Specifically, by introducing a nonconvex logarithmic sparse regularization $\ell(\nabla_j\mathbf{x}) = \sum_i \log(1 + \theta|\nabla_j x_i|^2)$ to the general sparse coding model in the leader subproblem and TV model in the follower subproblem in Eq. (2), we directly formulate our JFOP model as

$$\begin{aligned} &\min_{\hat{\mathbf{x}}} \tfrac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\beta}\|^2 + \gamma\|\boldsymbol{\beta}\|_1 + \tfrac{\rho}{2}\textstyle\sum_{j\in\{h,v\}} \ell(\nabla_j\mathbf{x}) + \chi_\Omega(\mathbf{x}), \\ &s.t.,\ \arg\min_{\mathbf{x}} F(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|^2 + \lambda\textstyle\sum_{j\in\{h,v\}} \|\nabla_j\mathbf{x}\|_1, \end{aligned} \tag{8}$$

where $\chi_\Omega(\mathbf{x})$ is the indicator function on $\Omega = [0, 1]$, i.e., if $\mathbf{x} \in \Omega$ then $\chi_\Omega(\mathbf{x}) = 0$, otherwise $\chi_\Omega(\mathbf{x}) = \infty$. The matrix $\mathbf{A}$ denotes the task-related operation (e.g., blur, mask and interpolation), $\mathbf{x}$ and $\mathbf{b}$ denote the latent image and corrupted observation, respectively. $\hat{\mathbf{x}}$ represents $\hat{\mathbf{x}} = [\mathbf{x}^T, \boldsymbol{\beta}^T]^T$. Notice that the aforementioned parameters γ, ρ, θ and λ are all positive constants.

By considering the latent image $\mathbf{x}$ as the uniform augment of our gaming (i.e., the auxiliary variables $\boldsymbol{\beta}$ are calculated only for their own subproblems) in Eq. (8), we actually obtain a JFOP formulation to integrate both leader-level synthesis and follower-level analysis mechanisms to address different low-level vision applications, including deblurring, super-resolution and inpainting, etc. We incorporate experimentally designed and trained network architectures into the follower-level iterations, thus obtain the LBPP solution for these problems. In summary, the proposed LBPP indeed integrates advantages from different methodologies (i.e., analysis, synthesis and data-dependent modules) to address various problems.

As for the network architecture $\mathcal{D}$, we just follow (Zhang et al. 2017b) to build a series of denoising CNNs which consist of 7 dilated convolution layers with 64 kernels of the

[5]We follow standard settings in image processing to define $\mathbf{D}$ as the inverse wavelet transform in our problem. Indeed, the form $\mathbf{D}$ is denote as $\mathbf{D} = \mathbf{BW}^\top$, where $\mathbf{B}$ is the matrix of the blur kernel k, $\mathbf{W}^\top$ is the inverse of the wavelet transform of $\mathbf{W}$.

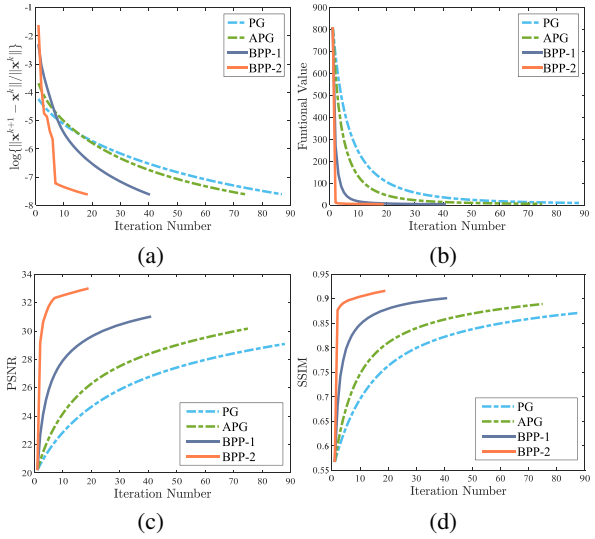


Figure 1: Illustrating convergence behaviors of BPP comparing with the PG and APG when sloving (6). BPP-1 and BPP-2 are two different constraint strategies with PG and ADMM respectively. The iteration errors after log transform and functional value are respectively plotted in the subfigures (a) and (b). The PSNR and SSIM are respectively plotted in the subfigures (c) and (d). The iteration stop threshold is set as $5e^{-4}$.

-	29.0886	30.1701	31.0194	**33.0061**
Input	PG	APG	BPP-1	BPP-2

Figure 2: The deblur results of PG, APG, and various of BPPs (BPP-1, BPP-2 are BPP with PG and ADMM as $\mathcal{F}$ for the follower subproblem, respectively).

size 3×3. The ReLU and batch normalization layers are also incorporated into $\mathcal{D}$ accordingly. We randomly select 800 images from the ImageNet dataset (Krizhevsky, Sutskever, and Hinton 2012) as our training data.

Experiments Result

In this section, we firstly verified our theoretical results by investigating the iteration behaviors of the proposed BPP on the stranded deconvolution formulation. We then evaluated the performance of our LBPP both with traditional and learning based methods on different vision applications. We conducted these experiments on a computer with Intel Core i7-7700 CPU (3.6 GHz), 32GB RAM and an NVIDIA GeForce GTX 1060 6GB GPU.

Theoretical Verifications

To verify our theoretical investigations, we first performed experiments on non-blind deconvolution. Notice that this problem can be directly addressed by our BPP and LBPP.

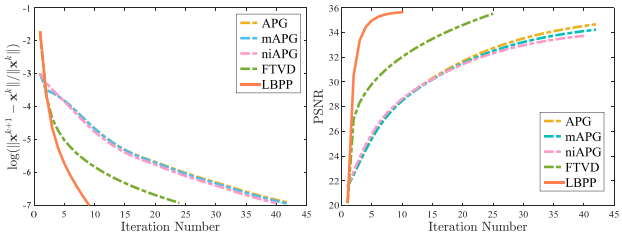


Figure 3: Comparing iteration behaviors of LBPP to classical APGs, including exact ones (APG and mAPG), inexact APG (niAPG) and FTVD with addtional Gaussian noise level 1‰. The iteration errors after log transform and PSNR are respectively plotted in the subfigures (a) and (b). The iteration stop threshold is set as 10^{-3}.

Table 1: Averaged PSNR and SSIM on the benchmark image set (Schmidt and Roth 2014). The first column σ represents the Gaussian noise level. The first row is the comparsion methods.

σ	Metric	APG	mAPG	niAPG	FTVD	LBPP
1%	PSNR	27.32	26.68	27.24	27.56	**28.48**
	SSIM	0.71	0.67	0.73	0.77	**0.81**
2%	PSNR	25.61	25.20	25.63	26.63	**27.06**
	SSIM	0.63	0.60	0.64	0.73	**0.75**
3%	PSNR	24.63	24.39	24.76	24.88	**26.13**
	SSIM	0.57	0.55	0.61	0.62	**0.71**

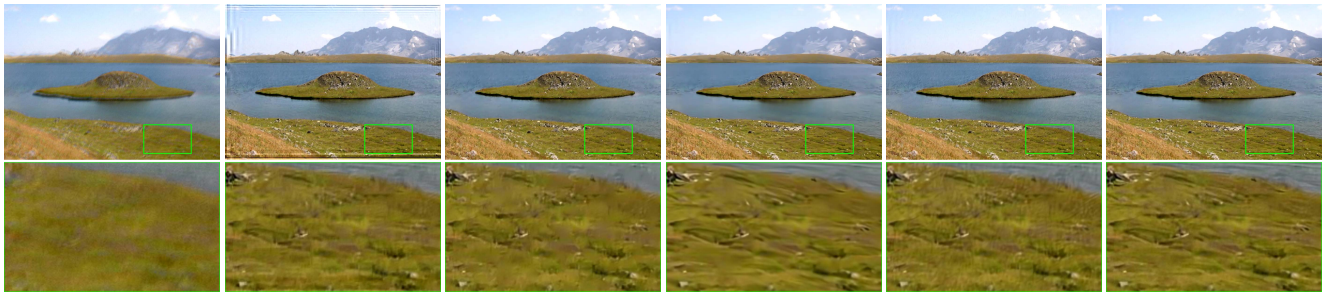
Convergence of BPP: Fig. 1 shows the experiments that compared the performance of the algorithm BPP to PG and APG when solving the minimization problem Eq. (6). Further, we introduce two different strategies that BPP-1 and BPP-2 for short which respresent PG and ADMM when solving the follower subproblem respectively to illustrate the effect of the follower part. And the corresponding visual results of PG, APG, BPP-1 and BPP-2 are showed in Fig. 2. In the subfigures (a) and (b) of Fig. 1, we plot the curves of iteration error after log transform ($\log(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|/\|\mathbf{x}^k\|)$) and objective functional value for BPPs with PG and APG respectively. And subfigures (c) and (d) plot the PSNR and SSIM scores with iterations. We concluded that our proposed scheme performance better and faster than the single subproblem in Eq. (7). Further, the corresponding visual comparison is shown in Fig. 2 which stated that the proposed method can recover image sharpness and naturalness.

Convergence of LBPP: As a further baseline, we first compared our LBPP with traditional methods, such as, APG (Li and Lin 2015), monotone APG, i.e. mAPG (Li and Lin 2015), niAPG (Yao et al. 2017) and FTVD (Wang et al. 2008) (ADMM) with additional 1‰ noise level and 27×27 kernel size. The comparison results are shown in Fig. 3. Obviously, it can be seen that the iterative error after log transform and the PSNR comparison stated that the LBPP is much faster and showed higher PSNR score.

We then further compared our LBPP with the tradition-

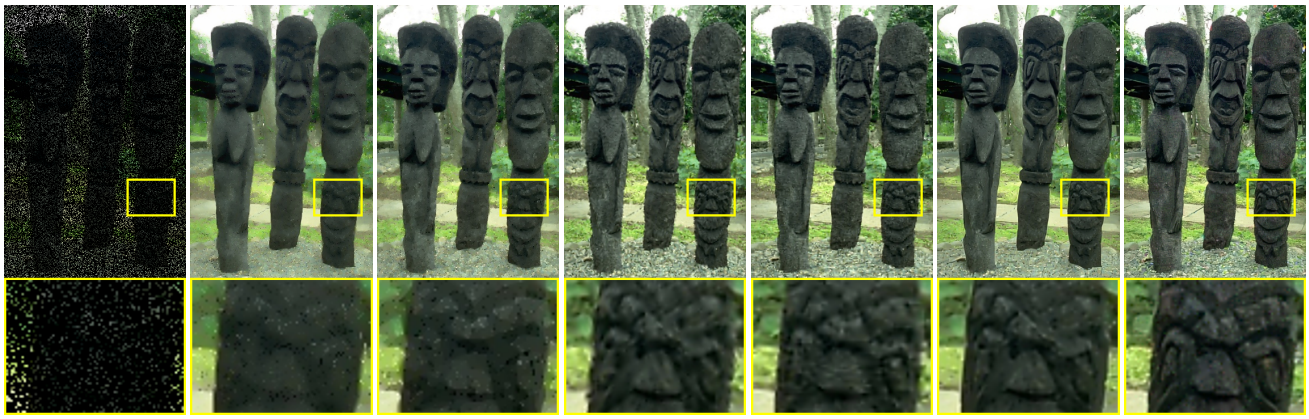
Table 2: Averaged quantitative results of image deblurring on Sun *et al*.' benchmark.

	TV	HL	EPLL	CSF	IDDBM3D	RTF	MLP	IRCNN	PADNet	Ours
PSNR	30.67	31.03	32.44	31.55	30.79	32.45	31.47	32.61	32.71	**32.90**
SSIM	0.85	0.85	0.88	0.87	0.86	0.89	0.88	0.89	0.89	**0.90**
Time (s)	6.38	**0.49**	721.98	0.50	48.66	240.98	4.59	16.67	7.13	2.41



– Blurry Image | 29.10 / 0.79 IDDBM3D | 31.63 / 0.82 RTF | 31.60 / 0.82 IRCNN | 31.79 / 0.83 PADNet | **31.92 / 0.84** Ours

Figure 4: Image deblurring results (with PSNR / SSIM scores) on a challenging example image in Sun *et al*.' benchmark. The zoomed in comparisons are presented below each image.



– Input | 22.7823 / 0.5492 ISDSB | 23.1478 / 0.6561 TV | 23.4491 / 0.6430 WNNM | 23.8422 / 0.6430 FoE | 24.1544 / 0.6609 IRCNN | **27.2059 / 0.8327** Ours

Figure 5: Image inpainting results (with PSNR / SSIM scores) on a challenging example image with 80% missing pixels.

al methods under three different Gaussian noise levels (i.e., 1%, 2% and 3%) on the image set (collected by (Schmidt and Roth 2014)) and the corresponding results are shown in Tab. 2 with quantitative performance (i.e., PSNR and SSIM metrics). It can be seen that our LBPP outperforms classical numerical solvers by a large margin in terms of performance and accuracy.

State-of-the-art Comparisons

We then evaluated our LBPP (learning with BPP iterations) on a variety of low-level vision applications including image deblurring, single image super-resolution and image inpainting.

Image Deblurring: We firstly considered image deblurring task, which is a direct application of the deconvolution model. In deblurring problem, matrix $\mathbf{A}$ stated in the application part is the blur kernel and $\mathbf{b}$ is blurry image. As usual, the blurry images are synthesized by first applying a blur kernel and further adding additive Gaussian noise. We consider the circular boundary conditions when performing the convolution. We reported the results of our LBPP on Sun *et al*' challenging benchmark (Sun et al. 2013) which contains 80 images together with other state-of-the-art methods, including the traditional method (e.g. IDDBM3D (Danielyan, Katkovnik, and Egiazarian 2012), TV (Wang et al. 2008), HL (Krishnan and Fergus 2009)), parameters learning based methods (e.g. EPLL (Zoran and Weiss 2011), CSF (Schmidt and Roth 2014), RTF (Schmidt et al. 2016)) and network

Table 3: Averaged quantitative comparison of image completion on CBSD68 dataset. The first row is the proportion of masks. The first column is the Comparison methods on inpainting.

Mask	20%	40%	60%	80%	Text
TV	36.30 0.97	32.22 0.93	29.20 0.86	26.07 0.74	35.29 0.97
FoE	38.23 0.95	34.01 0.90	30.81 0.81	27.64 0.65	37.05 0.95
VNL	28.87 0.95	27.55 0.91	26.13 0.85	24.23 0.75	28.58 0.95
ISDSB	35.20 0.96	31.32 0.91	28.23 0.83	24.92 0.70	34.91 0.96
WNNM	36.42 0.98	31.75 0.94	28.71 0.89	25.63 0.78	34.89 0.97
IRCNN	39.04 0.98	34.92 0.95	28.71 0.89	25.63 0.78	37.26 0.97
Ours	**39.38** **0.98**	**34.94** **0.96**	**31.52** **0.91**	**27.88** **0.81**	**37.38** **0.98**

based methods (e.g. MLP (Schuler et al. 2013), IRCNN (Zhang et al. 2017b), PADNet (Liu et al. 2018)) .

It can be seen in Tab. 2 that our method obtained the best quantitative performance (i.e., PSNR and SSIM metrics). Moreover, it is also faster than most of the compared methods. We also illustrated the visual comparisons on image deblurring in Fig. 4. It can be seen that our method generated clearer image with more details and thus reasonably obtained the higher PSNR score.

Image Inpainting: In image inpainting task, the matrix $\mathbf{A}$ is mask, and $\mathbf{b}$ is the missing pixels image. Then, we conducted experiments on the problem of image inpainting, which aims to recover the missing pixels of the observation. By introducing a mask matrix to perform point multiplication on the latent image, this application can also be formulated and addressed by our JFOP framework. Here we compared our LBPP with TV (Osher et al. 2005), FOE (Roth and Black 2009), VNL (Arias et al. 2011), ISDSB (He and Wang 2014), WNNM (Gu et al. 2017) and IRCNN (Zhang et al. 2017b) on this problem. We normalized the pixel values to $[0, 1]$. We generated random masks of different levels including 20%, 40%, 60%, 80% missing pixels on CBSD68 dataset (Zhang et al. 2017a). Moreover, we collected 12 different text masks to further evaluate the proposed algorithm.

Tab. 3 presented the PSNR and SSIM comparsion result with different masks. It can be seen that our method can performance better than the state-of-the-art approaches regardless the proportion of masks. Further, in comparison with the visual performance of LBPP with all these methods, we presented the 80% missing pixels comparisons in Fig. 5 which showed inpainting results on an example image from CSD-B68 dataset. It can be observed that our result successfully recovered the image with better visual quality, especially in the zoomed-in regions with rich details. Moreover, our PSNR and SSIM scores are also higher than others.

Figure 6: Image super-resolution results on a challenging example image with $\times 4$ scale. We only plot results on the zoomed in region. The PSNR / SSIM scores are reported below each image.

Single Image Super-resolution: Single image super-resolution is another important low-level vision problem and many researches have been investigated in recent years. Actually, this task can also be formulated as a linear inverse problem, but with more complex transformation process. In Fig. 6, we first compared LBPP with state-of-the-art methods (e.g. naive bicubic interpolation, TNRD (Chen and Pock 2017), SRCNN (Dong et al. 2016), VDSR (Kim, Kwon Lee, and Mu Lee 2016)) on an example image from (Huang, Singh, and Ahuja 2015). It can be seen that LBPP significantly performs superiority against these empirically designed super-resolution networks (e.g. SRCNN and VDSR) from both qualitative and quantitative perspectives.

Conclusion

In this paper, we proposed a characterized feasibility constraint framework for the nonconvex nonsmooth Joint Feasibility and Optimality Pursuit (JFOP) model and proved the global convergence with some relatively loose assumptions and estimated the asymptotic convergence rate for the proposed scheme. Then the extended learning fasibility constraint scheme, i.e., learning BPP (LBPP) was established, in which we brought up a theoretically guaranteed strategy to incorporate data-dependent deep networks at each iteration resulting in much faster and more efficient solver for Eq. (1) in practical real-world tasks. Extensive experiments on some challenging tasks showed that our method has better visual performance and quantitative scores against other state-of-the-art methods.

References

Arias, P.; Facciolo, G.; Caselles, V.; and Sapiro, G. 2011. A variational framework for exemplar-based image inpainting. *IJCV* 93(3):319–347.

Bard, J. F. 2013. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media.

Beck, A., and Teboulle, M. 2009. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE TIP* 18(11):2419–2434.

Bennett, K.; Kunapuli, G.; Hu, J.; and Pang, J.-S. 2008. Bilevel optimization and machine learning. *Computational Intelligence: Research Frontiers* 25–47.

Bolte, J.; Sabach, S.; and Teboulle, M. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* 146(1-2):459–494.

Boyd, S. 2011. Alternating direction method of multipliers. In *Talk at NIPS workshop on optimization and machine learning*.

Chen, Y., and Pock, T. 2017. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE TPAMI* 39(6):1256–1272.

Danielyan, A.; Katkovnik, V.; and Egiazarian, K. 2012. Bm3d frames and variational image deblurring. *IEEE TIP* 21(4):1715–1728.

De los Reyes, J. C., and Schönlieb, C.-B. 2013. Image denoising: Learning the noise model via nonsmooth pde-constrained optimization. *Inverse Problems & Imaging* 7(4).

Dechter, R., and Cohen, D. 2003. *Constraint processing*. Morgan Kaufmann.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2016. Image super-resolution using deep convolutional networks. *IEEE TPAMI* 38(2):295–307.

Gu, S.; Xie, Q.; Meng, D.; Zuo, W.; Feng, X.; and Zhang, L. 2017. Weighted nuclear norm minimization and its applications to low level vision. *IJCV* 121(2):183–208.

He, L., and Wang, Y. 2014. Iterative support detection-based split bregman method for wavelet frame-based image inpainting. *IEEE TIP* 23(12):5470–5485.

Huang, J.-B.; Singh, A.; and Ahuja, N. 2015. Single image super-resolution from transformed self-exemplars. In *IEEE CVPR*, 5197–5206.

Kim, J.; Kwon Lee, J.; and Mu Lee, K. 2016. Accurate image super-resolution using very deep convolutional networks. In *IEEE CVPR*, 1646–1654.

Krishnan, D., and Fergus, R. 2009. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 1033–1041.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Li, H., and Lin, Z. 2015. Accelerated proximal gradient methods for nonconvex programming. In *NIPS*.

Liu, R.; Lin, Z.; Zhang, W.; Tang, K.; and Su, Z. 2013. Toward designing intelligent pdes for computer vision: An optimal control approach. *Image and vision computing* 31(1):43–56.

Liu, R.; Fan, X.; Cheng, S.; Wang, X.; and Luo, Z. 2018. Proximal alternating direction network: A globally converged deep unrolling framework. In *AAAI*.

Ochs, P.; Ranftl, R.; Brox, T.; and Pock, T. 2016. Techniques for gradient-based bilevel optimization with nonsmooth lower level problems. *Journal of Mathematical Imaging and Vision* 56(2).

Osher, S.; Burger, M.; Goldfarb, D.; Xu, J.; and Yin, W. 2005. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation* 4(2):460–489.

Rockafellar, R. T., and Wets, R. J.-B. 2009. *Variational analysis*, volume 317. Springer Science & Business Media.

Roth, S., and Black, M. J. 2009. Fields of experts. *IJCV* 82(2).

Sabach, S., and Shtern, S. 2017. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization* 27(2):640–660.

Schmidt, U., and Roth, S. 2014. Shrinkage fields for effective image restoration. In *IEEE CVPR*, 2774–2781.

Schmidt, U.; Jancsary, J.; Nowozin, S.; Roth, S.; and Rother, C. 2016. Cascades of regression tree fields for image restoration. *IEEE TPAMI* 38(4):677–689.

Schuler, C. J.; Christopher Burger, H.; Harmeling, S.; and Scholkopf, B. 2013. A machine learning approach for non-blind image deconvolution. In *IEEE CVPR*, 1067–1074.

Sun, L.; Cho, S.; Wang, J.; and Hays, J. 2013. Edge-based blur kernel estimation using patch priors. In *ICCP*, 1–8.

Von Stackelberg, H. 2010. *Market structure and equilibrium*. Springer Science & Business Media.

Wang, Y.; Yang, J.; Yin, W.; and Zhang, Y. 2008. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* 1(3):248–272.

Yao, Q.; Kwok, J. T.; Gao, F.; Chen, W.; and Liu, T.-Y. 2017. Efficient inexact proximal gradient algorithm for nonconvex problems. *IJCAI*.

Ye, J. J., and Zhu, D. 2010. New necessary optimality conditions for bilevel programs by combining the mpec and value function approaches. *SIAM Journal on Optimization* 20(4):1885–1905.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017a. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP* 26(7):3142–3155.

Zhang, K.; Zuo, W.; Gu, S.; and Zhang, L. 2017b. Learning deep cnn denoiser prior for image restoration. In *IEEE CVPR*, volume 2.

Zoran, D., and Weiss, Y. 2011. From learning models of natural image patches to whole image restoration. In *IEEE ICCV*, 479–486.