Learning an Alternating Bergman Network for Non-convex and Non-smooth Optimization Problems

Yiyang Wang[1], Risheng Liu[2,3,4(✉)], and Zhixun Su[1,5]

[1] School of Mathematical Science, Dalian University of Technology, Dalian, China
yywerica@gmail.com, zxsu@dlut.edu.cn
[2] DUT-RU International School of Information Science and Engineering, Dalian University of Technology, Dalian, China
rsliu@dlut.edu.cn
[3] Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen, China
[4] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, China
[5] National Engineering Research Center of Digital Life, Guangzhou, China

Abstract. Recently, non-convex and non-smooth problems have received considerable interests in the fields of image processing and machine learning. The proposed conventional algorithms rely on carefully designed initializations, and the parameters can not be tuned adaptively during iterations with corresponding to various real-world data. To settle these problems, we propose an alternating Bregman network (ABN), which discriminatively learns all the parameters from training pairs and then is directly applied to test data without additional operations. Specifically, parameters of ABN are adaptively learnt from training data to force the objective value drop rapidly toward the optimal and then obtain a desired solution in practice. Furthermore, the basis algorithm of ABN is an alternating method with Bregman modification (AMBM), which solves each subproblem with a designated Bregman distance. This AMBM is more general and flexible than previous approaches; at the same time it is proved to receive the best convergence result for general non-convex and non-smooth optimization problems. Thus, our proposed ABN is an efficient and converged algorithm which rapidly converges to desired solutions in practice. We applied ABN to sparse coding problem with ℓ_0 penalty and the experimental results verify the efficiency of our proposed algorithm.

Keywords: Non-convex optimization · Alternating direction method · Sparse approximation · Learning-based algorithm · Optimization network

Y. Sun et al. (Eds.): IScIDE 2017, LNCS 10559, pp. 11–27, 2017.
DOI: 10.1007/978-3-319-67777-4_2

1 Introduction

Recently, a variety of applications in the fields of data mining [1,2], signal and image processing [3–7], machine learning and statistical inference [8–12] have been modeled as non-convex and non-smooth optimization problems [13,14]. Among them, a class of non-convex and non-smooth problems has gained considerable attentions and many algorithms are designed for solving them, which can be formulated as a general problem with n variables as follows:

$$\min_{\mathbf{X}:=(\mathbf{x}_1,\ldots,\mathbf{x}_n)} \Psi(\mathbf{X}) = H(\mathbf{X}) + \sum\nolimits_{i=1}^{n} f_i(\mathbf{x}_i), \tag{1}$$

where variables $\{\mathbf{x}_i\}_{i=1}^n$ can be either vectors or matrices throughout this paper. The objective function in problem (1) satisfies:

1. f_is are proper, lower semi-continuous functions.
2. H is a C^1 function; ∇H is Lipschitz continuous on a bounded set with constant M.
3. $\Psi(\mathbf{X})$ is a coercive, Kurdyka-Łojasiewicz (KŁ) function[1].

This problem (1) covers a variety of problems in various areas. For example, the non-negative matrix factorization problem framework [15,16] is always formed as a special two-block case of problem (1). The well-known sparse coding problems with various non-convex and non-smooth regularizers like ℓ_0 penalty [5,14] and MCP [17,18] belong to one-block case of problem (1).

Due to the non-convexity and non-smoothness of problem (1), there is not much work focusing on proposing converged and efficient algorithms for solving these complex problems. In 2009, the authors in [19] first propose a proximal algorithm (PA) for solving a one-block case of problem (1). This proposed algorithm is established with global convergence: *the algorithm generates a Cauchy sequence that converges to a critical point of the problem*, which as far as we know, is the best result in general non-convex and non-smooth optimization problems. Afterwards, they in [20,21] propose proximal alternating method (PAM) and proximal alternating linearized method (PALM) for solving problem (1). Following their pioneer work, plenty of algorithms for solving problem (1) have been proposed from different perspectives. The authors in [22] present a randomized/deterministic block prox-linear algorithm (BPA) for solving non-convex problems that may have block separable non-smooth terms. In [23], the authors introduce accelerated proximal gradient (APG) to general non-convex and non-smooth problems. They introduce a monitor in APG to ensure the objective function to have sufficient descent property. Moreover, the convergence rates of related algorithms are studied as well [19,24] by adding additional assumptions on KŁ functions.

All these proposed algorithms are demonstrated to converge to a critical point of the problem, however, in most cases, the converged critical points are not the desired solutions for specific problems. Since there always exists many

[1] The definition of KŁ function will be introduced afterwards.

critical points for non-convex and non-smooth problems, at the same time, the conventional algorithms converge to specific critical points by carefully selecting the algorithm parameters, thus it is extremely hard and time-consuming for conventional algorithms to converge to desired solutions in applications. One way to address this problem is to initialize the algorithm very close to the desired solution. For example, when solving the dictionary learning problem [21,25], the initialization of the dictionary is done via filling in local DCT transformation, which looks quite similar to the desired dictionary solution. With the carefully designed initializations, the process of tuning the algorithm parameters will be greatly simplified in practice.

On the other hand, in theory, the convergence rates of the previous algorithms are only affected by the objective function $\Psi(\mathbf{X})$ [23,24]. However, the truth is, carefully choosing appropriate parameters in the algorithms does affect the iteration time in practice. For example, the proximal parameters in all the algorithms mentioned above should not be set too large, otherwise the algorithm slowly converges with tiny step sizes. In the previous work, some strategies like line search with Barzilai-Borwein (BB) initialization [23,26] are proposed together with the algorithms to search appropriate parameters. However, the parameters in previously conventional algorithm framework can not be adaptively adjusted and tuned during iterations with corresponding to different data in various application problems. Adaptively selecting the parameters in every iteration does certainly shorten the whole number of iterations.

To address the above mentioned questions, we in this paper propose a data-driven general algorithm framework, which adaptively decides the algorithm parameters during iterations with the help of training data. Learning from training data give a guidance of the convergence direction so that it converges very close to the desired solution. Moreover, the algorithm parameters of the learning-based algorithm are adaptively decided during iterations to force the objective value drop rapidly to the optimal, thus it converges to the desired solution with much less iterations than conventional algorithms. It can be seen that learning an algorithm from data brings a new idea on designing an efficient and robust method for solving non-convex and non-smooth optimization problems, which is quite different from designing algorithms in conventional ways [19–23].

Further on, our learning-based algorithm framework named alternating Bregman network (ABN) can be regarded as a special network [27,28], which basis is an alternating method with Bregman modification (AMBM) designed in this paper for solving problem (1). Different from the previous work, our proposed AMBM solves each subproblem by adding a designated Bregman distance. Specifically, for different subproblems, AMBM adds different Bregman distances for the subproblems to simplify the solving processes. The previous algorithms, PA, PAM, PALM and BPA can all be seen as special cases of our AMBM. Thus our AMBM is more general, flexible and applicable than the previous proposed algorithms. At the same time, AMBM is established to retain the best convergence property and its convergence rate can also be theoretically analyzed for

special objective function $\Psi(\mathbf{X})$, which shares the same conclusions with the previous algorithms. At last, our main contributions are summarized as follows:

1. We propose an efficient algorithm framework named AMBM for solving the non-convex and non-smooth optimization problems. In consideration of the complex subproblems in applications, AMBM appropriately solves each subproblem by adding a designated Bregman distance and is more general and flexible than existing algorithms. At the same time, AMBM has the same convergence property with the best result in general non-convex and non-smooth optimization.
2. On the basis of AMBM, we introduce an algorithm network named ABN, which is discriminatively learnt from training data. Different from conventional methods, ABN adaptively tune the algorithm parameters under the guidance of training data for rapidly converging to the desired solutions in practice.
3. We conduct experiments on solving sparse coding problem with ℓ_0 penalty. The experimental results in Sect. 5 demonstrate the efficiency of ABN and assist analyzing the convergence of our proposed algorithm.

The following part of this paper is organized as: we first give some necessary preliminaries in Sect. 2; then, in Sect. 3 we present the basis algorithm of ABN, i.e., AMBM on both detailed implementations and convergence analyses; follow that, we give the learning-based framework ABN in Sect. 4 and show experimental results in Sect. 5.

2 Preliminaries

To simplify the deduction, we analyze the convergence properties of AMBM and ABN on a special problem of (1):

$$\min_{\mathbf{z}:=(\mathbf{x},\mathbf{y})} \Psi(\mathbf{z}) = f(\mathbf{x}) + H(\mathbf{x},\mathbf{y}) + g(\mathbf{y}). \tag{2}$$

For simplicity, we replace the notations in problem (1) with the new ones: correspondingly, f and g are proper, lower semi-continuous functions; H is a C^1 function and Ψ is a KŁ function. Though the convergence properties in this paper are analyzed on problem (2), it is straightforward to extend it to n-block problem (1) [21].

Definition 1 *(KŁ inequality and KŁ function).* *A proper, lower semi-continuous function σ has KŁ property at $\tilde{\mathbf{u}}$ if there exists $\eta, \varepsilon > 0$ and a function $\psi \in \Lambda_\eta$ such that for all $\mathbf{u} \in \mathcal{U}(\tilde{\mathbf{u}}, \varepsilon) \cap [\sigma(\tilde{\mathbf{u}}) < \sigma(\mathbf{u}) < \sigma(\tilde{\mathbf{u}}) + \eta]$,*

$$\psi'(\sigma(\mathbf{u}) - \sigma(\tilde{\mathbf{u}}))\, dist(0, \partial\sigma(\mathbf{u})) \geq 1,$$

where $\mathcal{U}(\tilde{\mathbf{u}}, \varepsilon)$ denotes a neighborhood of $\tilde{\mathbf{u}}$; $dist(\mathbf{u}, S) := \inf\{\|\mathbf{v} - \mathbf{u}\|, \mathbf{v} \in S\}$ with $\|\cdot\|$ representing the ℓ_2 norm for vector and Frobenius norm for matrix

throughout the paper. Λ_η stands for a class of functions $\psi : [0, \eta) \to \mathbb{R}^+$ such that (1) $\psi(0) = 0$, $\psi'(s) > 0$ for all $s \in (0, \eta)$; (2) ψ is C^1 on $(0, \eta)$ and continuous at 0. Specially, $\psi(s)$ can be chosen as $cs^{1-\theta}$ with $c > 0$ and $\theta \in [0, 1)$ if Ψ is a semi-algebraic function [21]. Furthermore, if σ satisfies the KŁ property at each point in the domain of σ, then σ is called a KŁ function.

KŁ functions include strongly convex functions, real analytic functions and semi-algebraic functions, e.g., capped-l_1 penalty [26], ℓ_p $(0 \leq p < 1)$ norm [29], minimax concave penalty (MCP) [18] and smoothly clipped absolute deviation (SCAD) [30].

Definition 2 *(Bregman distance).* *φ is a convex differential function and its Bregman distance*

$$d_\varphi(\mathbf{u}, \mathbf{v}) = \varphi(\mathbf{u}) - \varphi(\mathbf{v}) - \nabla\varphi(\mathbf{v})^\top(\mathbf{u} - \mathbf{v}),$$

satisfies (1) $d_\varphi(\mathbf{u}, \mathbf{v}) \geq 0$, $\forall \mathbf{u}, \mathbf{v}$; (2) $d_\varphi(\mathbf{u}, \mathbf{v})$ is convex in $\mathbf{u}$, but not necessarily in $\mathbf{v}$.

We list the following φs according to the most commonly used Bregman distances [31,32], including

1. Euclidean distance: $\varphi(\mathbf{u}) = \|\mathbf{u}\|^2$;
2. Mahalanobis distance: $\varphi(\mathbf{u}) = \mathbf{u}^\top\mathbf{Q}\mathbf{u}$ with a symmetric positive definite $\mathbf{Q}$;
3. Kullback-Leibler divergence: $\varphi(\mathbf{u}) = \sum_j \mathbf{u}_j \log_2 \mathbf{u}_j$.

Assumption 3. *Suppose that $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ is the sequence generated by AMBM, then Bregman distances related to functions φ_1^k and φ_2^k are assumed to satisfy $d_{\varphi_1^k}(\mathbf{x}^{k+1}, \mathbf{x}^k) \geq \frac{\gamma_1^k}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$, $d_{\varphi_2^k}(\mathbf{y}^{k+1}, \mathbf{y}^k) \geq \frac{\gamma_2^k}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2$ with bounded and positive parameters $\{\gamma_1^k\}_{k\in\mathbb{N}}$ and $\{\gamma_2^k\}_{k\in\mathbb{N}}$. Moreover, $\nabla\varphi_1^k$ and $\nabla\varphi_2^k$ are assumed to be Lipschitz continuous on bounded sets with bounded constants ζ_1^k and ζ_2^k.*

3 The Algorithmic Framework and Convergence Analyses

Before providing the learning-based algorithm network ABN, we in this section first introduce its basis algorithm AMBM, an alternating method with Bregman modification for solving problem (2). Firstly, we present the detailed implementation of AMBM and discuss its relationship with the previous algorithms. Then, we give convergence analyses, including the global convergence property and the convergence rate for our proposed AMBM. At last, we give examples to show the superiority of AMBM.

By adding Bregman distance, our proposed AMBM for solving problem (2) considers the following updates:

$$\begin{aligned} \mathbf{x}^{k+1} &\in \arg\min_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}^k) + d_{\varphi_1^k}(\mathbf{x}, \mathbf{x}^k), \\ \mathbf{y}^{k+1} &\in \arg\min_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}^{k+1}, \mathbf{y}) + d_{\varphi_2^k}(\mathbf{y}, \mathbf{y}^k), \end{aligned} \tag{3}$$

where $d_{\varphi_1^k}$ and $d_{\varphi_2^k}$ are the Bregman distances with respect to φ_1^k and φ_2^k that satisfy Assumption 3.

The proposed AMBM solves each subproblem of $\mathbf{x}^{k+1}$ and $\mathbf{y}^{k+1}$ in a flexible formulation with specific Bregman distances. Many existing methods [19,21,23] can be regarded as a special case of AMBM. For example, it is chosen as Euclidean distance in PA [19]. However, in PALM [21], the Bregman distances are chosen as Mahalanobis distances under special conditions to meet the requirement of symmetric, positive and definite Q for both subproblems of $\mathbf{x}$ and $\mathbf{y}$. All the subproblems in these previous work share the same Bregman distance. However, an appropriate Bremgan distance will certainly benefits to solving subproblems efficiently. For example, for solving the L_0 gradient minimization problem in [33]: $\min_{\mathbf{x},\mathbf{y}} \|\mathbf{y}-\mathbf{c}\|^2 + \lambda\|\mathbf{x}\|_0 + \|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2$, the Bregman distance should be chosen as the Euclidean distance for $\mathbf{y}$-subproblem but the Mahalanobis distance for $\mathbf{x}$-subproblem so that each subproblem can be solved explicitly and efficiently. Thus our AMBM provides the chance for efficiently solving each subproblem with different Bregman distances. Further discussions on our proposed AMBM algorithm will be shown in Sect. 3.2. We would like to give the convergence analyses of our AMBM at first.

3.1 Convergence Analyses

Under the Assumption 3, we can obtain the key lemma as follows. The two assertions proposed in the following key lemma is the cornerstone for proving the main convergence theorem, i.e., Theorem 4.

Lemma 1. *Supposing $\{(\mathbf{x}^k,\mathbf{y}^k)\}_{k\in\mathbb{N}}$ be a bounded sequence generated by AMBM for solving the problem (2), then there exist positive integers α and β such that the following assertions hold:*

$$\Psi(\mathbf{z}^k) - \Psi(\mathbf{z}^{k+1}) \geq \alpha(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2 + \|\mathbf{y}^{k+1}-\mathbf{y}^k\|^2), \tag{4}$$

$$dist(0, \partial\Psi(\mathbf{z}^k)) \leq \beta(\|\mathbf{x}^k-\mathbf{x}^{k-1}\| + \|\mathbf{y}^k-\mathbf{y}^{k-1}\|), \tag{5}$$

where constants are denoted: $\alpha = \min_{k\in\mathbb{N}}\{\frac{\gamma_1^k}{2}, \frac{\gamma_2^k}{2}\}$ and $\beta = \max_{k\in\mathbb{N}}\{\zeta_1^k, M+\zeta_2^k\}$.

Proof. Firstly, we prove the sufficiently descent property in Eq. (4). From the update of AMBM, i.e. Eq. (3), we have the following inequalities:

$$f(\mathbf{x}^{k+1}) + H(\mathbf{x}^{k+1},\mathbf{y}^k) + d_{\varphi_1^k}(\mathbf{x}^{k+1},\mathbf{x}^k) \leq f(\mathbf{x}^k) + H(\mathbf{x}^k,\mathbf{y}^k) + d_{\varphi_1^k}(\mathbf{x}^k,\mathbf{x}^k),$$
$$g(\mathbf{y}^{k+1}) + H(\mathbf{x}^{k+1},\mathbf{y}^{k+1}) + d_{\varphi_2^k}(\mathbf{y}^{k+1},\mathbf{y}^k) \leq g(\mathbf{y}^k) + H(\mathbf{x}^{k+1},\mathbf{y}^k) + d_{\varphi_2^k}(\mathbf{y}^k,\mathbf{y}^k).$$

By adding the above two inequalities together, we have that

$$\begin{aligned}
&\Psi(\mathbf{x}^k,\mathbf{y}^k) - \Psi(\mathbf{x}^{k+1},\mathbf{y}^{k+1}) \\
\geq& f(\mathbf{x}^k) + H(\mathbf{x}^k,\mathbf{y}^k) + g(\mathbf{y}^k) - f(\mathbf{x}^{k+1}) - H(\mathbf{x}^{k+1},\mathbf{y}^{k+1}) - g(\mathbf{y}^{k+1}) \\
\geq& d_{\varphi_1^k}(\mathbf{x}^{k+1},\mathbf{x}^k) - d_{\varphi_1^k}(\mathbf{x}^k,\mathbf{x}^k) + d_{\varphi_2^k}(\mathbf{y}^{k+1},\mathbf{y}^k) - d_{\varphi_2^k}(\mathbf{y}^k,\mathbf{y}^k).
\end{aligned}$$

From the definition of the Bregman distance, we can see that $d_{\varphi_1^k}(\mathbf{x}^k, \mathbf{x}^k) = 0$ and $d_{\varphi_2^k}(\mathbf{y}^k, \mathbf{y}^k) = 0$. Together with the inequalities in Assumption 3, we conclude the proof of the sufficiently descent property Eq. (4):

$$\begin{aligned}&\Psi(\mathbf{x}^k, \mathbf{y}^k) - \Psi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})\\ \geq&\frac{\gamma_1^k}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{\gamma_2^k}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \geq \alpha(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2),\end{aligned}$$

where $\alpha = \min_{k\in\mathbb{N}}\{\frac{\gamma_1^k}{2}, \frac{\gamma_2^k}{2}\}$.

Secondly, we prove the subgradient lower bound property, i.e. Eq. (5). We first write the first-order optimality condition of the update in Eq. (3):

$$\begin{aligned}0 &= \mathbf{u}_1^k + \nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^{k-1}) + \nabla d_{\varphi_1^k}(\mathbf{x}^k, \mathbf{x}^{k-1}),\\ 0 &= \mathbf{u}_2^k + \nabla_{\mathbf{y}} H(\mathbf{x}^k, \mathbf{y}^k) + \nabla d_{\varphi_2^k}(\mathbf{y}^k, \mathbf{y}^{k-1}),\end{aligned}$$

where $\mathbf{u}_1^k \in \partial f(\mathbf{x}^k)$ and $\mathbf{u}_2^k \in \partial g(\mathbf{y}^k)$. Then we conclude that

$$\begin{aligned}&(\nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^k) - \nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^{k-1}) - \nabla d_{\varphi_1^k}(\mathbf{x}^k, \mathbf{x}^{k-1}), -\nabla d_{\varphi_2^k}(\mathbf{y}^k, \mathbf{y}^{k-1}))\\ \in&(\partial_{\mathbf{x}}\Psi(\mathbf{x}^k, \mathbf{y}^k), \partial_{\mathbf{y}}\Psi(\mathbf{x}^k, \mathbf{y}^k)).\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\|\partial\Psi(\mathbf{x}^k, \mathbf{y}^k)\| \leq \|\partial_{\mathbf{x}}\Psi(\mathbf{x}^k, \mathbf{y}^k)\| + \|\partial_{\mathbf{y}}\Psi(\mathbf{x}^k, \mathbf{y}^k)\|\\ \leq&\|\nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^k) - \nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^{k-1})\| + \|\nabla d_{\varphi_1^k}(\mathbf{x}^k, \mathbf{x}^{k-1})\| + \|\nabla d_{\varphi_2^k}(\mathbf{y}^k, \mathbf{y}^{k-1})\|\\ =&\|\nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^k) - \nabla_{\mathbf{x}} H(\mathbf{x}^k, \mathbf{y}^{k-1})\| + \|\nabla\varphi_1^k(\mathbf{x}^k) - \nabla\varphi_1^k(\mathbf{x}^{k-1})\| + \|\nabla\varphi_2^k(\mathbf{y}^k) - \nabla\varphi_2^k(\mathbf{y}^{k-1})\|\\ \leq&M\|\mathbf{y}^k - \mathbf{y}^{k-1}\| + \zeta_1^k\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \zeta_2^k\|\mathbf{y}^k - \mathbf{y}^{k-1}\|.\end{aligned}$$

The equality comes from the definition of Bregman distance, and the second inequality is derived from the assumptions on the Lipschitz continuity of ∇H, $\nabla\varphi_1^k$ and $\nabla\varphi_2^k$. Thus, we have finished the proofs of the two inequalities in Lemma 1. ■

An additional hypothesis is added on the boundedness of $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$. This boundedness holds in several scenarios such as the function Ψ has bounded level sets [21]. Obviously, the objective function Ψ is sufficiently descent (Eq. (4) in Lemma 1) during iterations. This non-increasing property is the key for proving the main theorem. Moreover, the main theorem can be proved in exactly the same way as [21]. Thus we only present the main theorem as follows and refer readers to [21] for detailed proof.

Theorem 4 *(Convergence result). Under the Assumption 3, $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k\in\mathbb{N}}$ is supposed to be a bounded sequence generated by AMBM for problem (2), we can conclude from Lemma 1 that $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ is a Cauchy sequence that converges to a critical point $\widehat{\mathbf{z}}$ of Ψ.*

The convergence rate for non-convex and non-smooth optimization is quite complicated. The authors in [19] provide estimations when the function Ψ is semi-algebraic. Since the convergence rate of AMBM can be similarly estimated as [19], we only provide the conclusion in Theorem 5. Moreover, from the Theorem 5 we can tell that the convergence rate is not affected by the algorithm but the objective function Ψ.

Theorem 5 *(Convergence rate). If the function Ψ is semi-algebraic and the desingularizing function ψ is chosen as $\psi(s) = cs^{1-\theta}$ with $c > 0$ and $\theta \in [0, 1)$. Then by assuming the iterative sequence $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k\in\mathbb{N}}$ is bounded, we have the following estimations.*

(1) If $\theta = 0$, then $\exists K_1$ s.t. for $\forall k > K_1$, $\Psi(\mathbf{z}^k) = \Psi(\widehat{\mathbf{z}})$ and AMBM terminates in finite steps.
(2) If $\theta \in (0, 1/2]$, then there exists $\omega > 0$ and $\varrho \in [0, 1)$ such that $\|\mathbf{z}^k - \widehat{\mathbf{z}}\| \leq \omega\varrho^k$.
(3) If $\theta \in (1/2, 1)$, then there exists $\omega > 0$ such that $\|\mathbf{z}^k - \widehat{\mathbf{z}}\| \leq \omega k^{-\frac{1-\theta}{2\theta-1}}$.

The convergence property of AMBM can be summarized in a sentence: *if $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ is a bounded sequence and Assumption 3 holds, $\mathbf{z}^k$ always converges to a critical point $\widehat{\mathbf{z}}$ of Ψ no matter where the iteration starts.* The Bregman modification in AMBM provides a flexible framework but at the same time keeps the global convergence property (i.e., Theorem 4) and remains the convergence rate (i.e., Theorem 5) unchanged with the previous work [20,21,23,34], which is so far the best result identified by researchers as far as we know. It should be emphasized that a critical point of non-convex problem could be a local minimizer under some conditions, e.g. the second-order sufficient conditions [35].

3.2 Further Discussions on AMBM

It is apparent that not a few Bregman distances satisfy Assumption 3. However, different parameters γ lead to various Bregman distances. For clarity, we still take the L_0 gradient minimization problem [33]: $\min_{\mathbf{x},\mathbf{y}} \|\mathbf{y}-\mathbf{c}\|^2 + \lambda\|\mathbf{x}\|_0 + \|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2$ as an example.

For efficiently updating the $\mathbf{x}^{k+1}$, we should add a special Mahalanobis distance with $\varphi(\mathbf{x}) = \mu\|\mathbf{x}\|^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}^k\|^2$ and $\mathbf{Q} = \mu\mathbf{I} - \mathbf{A}^\top\mathbf{A}$, where $\mathbf{I}$ denotes the identity matrix. Moreover, parameter μ should satisfy $\mu > \lambda_m$ (λ_m denotes the maximum eigenvalue of $\mathbf{A}^\top\mathbf{A}$) to make the Bregman distance $d_\varphi(\mathbf{x}, \mathbf{x}^k)$ satisfy $d_\varphi(\mathbf{x}^{k+1}, \mathbf{x}^k) \geq \frac{\gamma}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$ with $\gamma = \mu - \lambda_m$. Hence, different μ in $\{\mu|\mu = \gamma + \lambda_m, \gamma > 0\}$ lead to different Bregman distances $d_\varphi(\mathbf{x}, \mathbf{x}^k)$ and then affect the step sizes of each iteration. In the following sections, we add the decisive parameter μ on $d_{\varphi(\mu)}(\mathbf{x}, \mathbf{x}^k)$ to specify particular Bregman distance.

It is well-known that an appropriate step size is beneficial to the convergence performance in first-order methods. Many strategies like line search [23] and BB rule [26] are proposed to search a good step size for each iteration. Inspired by data-driven methods, we in this paper learn the step size (i.e. parameter

μ) of each iteration adaptively and then produce an efficiently solving system, i.e., ABN for specific problems. The details of this learning-based algorithm are presented in the following section.

4 The Alternating Bregman Network and Theoretical Guarantee

To force the algorithm rapidly converging to a desired solution, we propose the learning-based algorithm network ABN on the basis of AMBM. The algorithm parameters of ABN is learnt from training data first and then is directly applied to all the test data with same distribution. This novel learning-based algorithm approaches to the desired solution rapidly and at the same time remains the same convergence property with its basis algorithm AMBM. In this section, at first, we present the motivation of learning an algorithm from data and give the general form of ABN. Then, detailed algorithm implementations are given for both learning and test part. At last, analyses on the theoretical guarantee of ABN are given with discussions.

4.1 Motivation and Detailed Implementation

As discussed in the previous section, the step size in each iteration plays an important role in first-order algorithms. Though many strategies like line search have been proposed to search a better step size, selecting appropriate parameters to force the algorithm converge to a desired solution is still extremely hard and time-consuming. Inspired by data-driven methods [27,28], we bring the assistance of the training data in our proposed AMBM and learn the step sizes adaptively.

Many existing data-driven algorithm network in machine learning [27,28] always contain two main stages. In the first stage, i.e., the training stage, the training data $\{\mathbf{z}_1^*, \ldots, \mathbf{z}_P^*\}$ are used to help train an efficient and universally applicable system for specific applications. Then in the second stage, i.e., the test stage, the pre-learnt algorithm net can be directly used for test data without manual operations. Inspired by the leaning-based process, we propose to learn an algorithm net to avoid manually setting step sizes in iterations. After learning the convergence algorithm with fixed parameters, prescribed expected error and certain iteration number, each test datum gets the result by applying the system directly without extra manually operations. But one must keep in mind that the learnt system is not available for all possible test samples, but only for *input variables drawn from the same distribution as the training samples* [28].

First, we would like to present the strategy for learning the algorithm parameters of ABN with the help of training data. By regarding the ground truth $\mathbf{z}_p^* \in \{\mathbf{z}_1^*, \ldots, \mathbf{z}_P^*\}$ as a critical point of the optimization problem (2), the step size parameters μ_1^k and μ_2^k (corresponding to the Bregman distance $d_{\varphi_1^k(\mu_1^k)}(\mathbf{x}, \mathbf{x}^k)$ and $d_{\varphi_2^k(\mu_2^k)}(\mathbf{y}, \mathbf{y}^k)$) should be chosen to help ABN converges rapidly to the

Algorithm 1. ABN: the learning part.

Parameters setting: ε, $l_{\max}$, ϵ and $T = 0$.
Variables initialization: $\{\mathbf{z}_p^0\}_{p=1}^P$, μ_1^0 and μ_2^0
while $\sum_p \|\mathbf{z}_p^k - \mathbf{z}_p^*\| < \varepsilon$ **do**
  $T = T + 1$.
  Parameters setting: $l = 1$, $\{\nu_1^l\}_{l\in\mathbb{N}}$, $\{\nu_2^l\}_{l\in\mathbb{N}}$, $\mu_1^{k,0} = \mu_1^{k-1}$ and $\mu_2^{k,0} = \mu_2^{k-1}$.
  while $l < l_{\max}$ **do**
    (For example, use Projected Gradient Descent.)
    $\delta\mathbf{x}_p^{k,l} = \frac{\partial \mathbf{x}_p^{k+1}}{\partial \mu_1}(\mu_1^{k,l})$, $\delta\mathbf{y}_p^{k,l} = \frac{\partial \mathbf{y}_p^{k+1}}{\partial \mu_2}(\mu_2^{k,l})$.
    $\mu_1^{k,l+1} = \Pi_{\chi_1^k}(\mu_1^{k,l} - \nu_1^l \Sigma_p \frac{\partial \mathcal{L}_P}{\partial \mathbf{x}_p^k}\delta\mathbf{x}_p^{k,l})$.
    $\mu_2^{k,l+1} = \Pi_{\chi_2^k}(\mu_2^{k,l} - \nu_2^l \Sigma_p \frac{\partial \mathcal{L}_P}{\partial \mathbf{y}_p^k}\delta\mathbf{y}_p^{k,l})$.
    If $\|\mu_1^{k,l+1} - \mu_1^{k,l}\| + \|\mu_2^{k,l+1} - \mu_2^{k,l}\| \le \epsilon$ break;
    Set $l = l + 1$;
  end while
  Output $\mu_1^k = \mu_1^{k,l+1}$ and $\mu_2^k = \mu_2^{k,l+1}$.
  $\mathbf{x}_p^{k+1} \in \arg\min_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}_p^k) + d_{\varphi_1^k(\mu_1^k)}(\mathbf{x}, \mathbf{x}_p^k)$.
  $\mathbf{y}_p^{k+1} \in \arg\min_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}_p^{k+1}, \mathbf{y}) + d_{\varphi_2^k(\mu_2^k)}(\mathbf{y}, \mathbf{y}_p^k)$
end while
Save $\{\mu_1^k\}_{k=1}^T$ and $\{\mu_2^k\}_{k=1}^T$.

Algorithm 2. ABN: the test part.

With fixed $\{\mu_1^k\}_{k=1}^T$ and $\{\mu_2^k\}_{k=1}^T$ learnt by Alg.1
Variable initialization: $\mathbf{z}^0$.
for $k = 1, 2, ..., T$ **do**
  $\mathbf{x}^{k+1} \in \arg\min_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}^k) + d_{\varphi_1^k(\mu_1^k)}(\mathbf{x}, \mathbf{x}^k)$.
  $\mathbf{y}^{k+1} \in \arg\min_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}^{k+1}, \mathbf{y}) + d_{\varphi_2^k(\mu_2^k)}(\mathbf{y}, \mathbf{y}^k)$
end for
Output $\mathbf{z}^T$.

ground truth $\mathbf{z}_p^*$. Specifically, this intuitive command can be formulated as an optimization problem, that is,

$$\mu_1^k \in \min_{\mu_1} \mathcal{L}(\mathbf{x}_p^{k+1}(\mu_1), \mathbf{x}_p^*), \quad \mu_2^k \in \min_{\mu_2} \mathcal{L}(\mathbf{y}_p^{k+1}(\mu_2), \mathbf{y}_p^*), \tag{6}$$

where we add subscript p to specify $\mathbf{x}_p^{k+1}$ and $\mathbf{y}_p^{k+1}$ as the iterative sequence related to $\mathbf{z}_p^* = (\mathbf{x}_p^*, \mathbf{y}_p^*)$. On the other hand, the step size μ_1^k and μ_2^k should ensure the Assumption 3 holds for $d_{\varphi_1^k(\mu_1^k)}(\mathbf{x}, \mathbf{x}^k)$ and $d_{\varphi_2^k(\mu_2^k)}(\mathbf{y}, \mathbf{y}^k)$, that is, $\mu_1^k \in \mathcal{X}_1^k$ and $\mu_2^k \in \mathcal{X}_2^k$ with

$$\begin{aligned} \mathcal{X}_1^k &:= \{\mu_1 | d_{\varphi_1^k(\mu_1)}(\mathbf{x}^{k+1}, \mathbf{x}^k) \ge \frac{\gamma_1^k}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2, \gamma_1^k > 0\}, \\ \mathcal{X}_2^k &:= \{\mu_2 | d_{\varphi_2^k(\mu_2)}(\mathbf{y}^{k+1}, \mathbf{y}^k) \ge \frac{\gamma_2^k}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2, \gamma_2^k > 0\}. \end{aligned} \tag{7}$$

Then by taking the average error of all the training data into consideration, we propose the learning process of ABN and present it as a bi-level optimization problem as follows to obtain μ_1^k and μ_2^k

$$\begin{aligned}
&\min_{\mu_1,\mu_2} \sum\nolimits_{p=1}^{P} \mathcal{L}(\mathbf{x}_p^{k+1}(\mu_1), \mathbf{x}_p^*) + \mathcal{L}(\mathbf{y}_p^{k+1}(\mu_2), \mathbf{y}_p^*), \\
&\quad \text{s.t. } \mathbf{x}_p^{k+1} \in \arg\min_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}_p^k) + d_{\varphi_1^k(\mu_1^k)}(\mathbf{x}, \mathbf{x}_p^k), \\
&\quad \mathbf{y}_p^{k+1} \in \arg\min_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}_p^{k+1}, \mathbf{y}) + d_{\varphi_2^k(\mu_2^k)}(\mathbf{y}, \mathbf{y}_p^k), \\
&\quad \mu_1 \in \mathcal{X}_1^k, \mu_2 \in \mathcal{X}_2^k.
\end{aligned} \tag{8}$$

Then μ_1^k and μ_2^k can be obtained by applying projected gradient descent, stochastic projected gradient descent [36] or other efficient algorithms to solve the bi-level problem (8). We take the projected gradient descent as an example, then μ_1^k and μ_2^k are obtained by iteratively updates the following equation with certain parameter ν_1^l and ν_2^l

$$\begin{aligned}
\mu_1^{k,l+1} &= \Pi_{\mathcal{X}_1^k}(\mu_1^{k,l} - \nu_1^l \frac{\partial \mathcal{L}_P}{\partial \mu_1}(\mu_1^{k,l})),\ \mu_1^{k,0} = \mu_1^{k-1} \in \mathcal{X}_1^k, \\
\mu_2^{k,l+1} &= \Pi_{\mathcal{X}_2^k}(\mu_2^{k,l} - \nu_2^l \frac{\partial \mathcal{L}_P}{\partial \mu_2}(\mu_2^{k,l})),\ \mu_2^{k,0} = \mu_2^{k-1} \in \mathcal{X}_2^k,
\end{aligned} \tag{9}$$

with $\mathcal{L}_P = \sum_{p=1}^{P} \mathcal{L}(\mathbf{x}_p^{k+1}(\mu_1), \mathbf{x}_p^*) + \mathcal{L}(\mathbf{y}_p^{k+1}(\mu_2), \mathbf{y}_p^*)$. After learning the step size parameters adaptively from training data, these parameters together with the number of iterations T can be fixed due to a prescribed expected error.

Therefore, for the test data from the same distribution, we can automatically obtain the convergence solution after T-step iteration. Moreover, we summarize the procedures of "learning" the algorithm ABN in Algorithm 1 for solving problem (2). The "test" part of ABN with fixed step size parameters, number of iteration T is summarized in Algorithm 2.

Remark 1. The proposed ABN can be seen as a bi-level optimization problem [37]. The upper level optimization solving $\{\mu_1^k\}_{k=1}^T$ and $\{\mu_2^k\}_{k=1}^T$ is the leader which tries to anticipate the next move of the lower level problem for solving $\{\mathbf{z}^{k+1}\}_{k=1}^T$. The basic idea of learning an algorithm by using bi-level optimization has been proposed by others [38], but the lower level optimization in their work is a convex problem. Moreover, learning with bi-level optimization has also been used for applications in image processing [27,39]. However, no theoretical analysis has been made to guarantee the convergence of these bi-level problems. The convergence analysis is quite challenging even for convex objective functions [38].

4.2 Theoretical Guarantee

It is observed that the best step size parameters $\{\mu_1^k\}_{k=1}^T$ and $\{\mu_2^k\}_{k=1}^T$ are chosen from sets χ_1^k and χ_2^k respectively. Thus parameters $\{\mu_1^k\}_{k=1}^T$ and $\{\mu_2^k\}_{k=1}^T$ ensure

that the corresponding Bregman distances satisfy the inequality in Assumption 3. In addition, ABN also has the sufficient descent property, i.e. Eq. (4) and subgradient lower bound property, i.e. Eq. (5). Hence ABN has the same convergence property of AMBM, that is, *suppose* $\{\mathbf{z}^k\}_{k\in\mathbb{N}}$ *is a bounded sequence and Assumption* 3 *holds,* $\mathbf{z}^k$ *always converges to a critical point* $\widehat{\mathbf{z}}$ *of* Ψ *no matter where the iteration starts.* However, ABN adaptively decides the step sizes during iterations, which ensures ABN perform better than AMBM with manually operating parameters. Though a faster convergence of ABN can not be guaranteed from theoretical analysis, it is verified through practical applications (see experiments in Sect. 5).

Remark 2. The iterative number T for the algorithm system is affected by the convergence rate of the basic algorithm. For example, we assume that the objective function Ψ satisfies the KŁ property with desingularizing function $\psi(s) = cs^{1-\theta}$ with $\theta \in (0, \frac{1}{2}]$. So from the Theorem 5, there exists $\omega > 0$ and $\varrho \in [0, 1)$ such that $\|\mathbf{z}^T - \mathbf{z}^*\| \leq \omega\varrho^T < \varepsilon$. Then it can be deducted that $T \ln \varrho \leq \ln \frac{\varepsilon}{\omega}$. With the fact that $\ln \varrho < 0$, we conclude that for achieving the error precision of ε, a theoretical T should satisfy $T \geq \frac{\ln \varepsilon - \ln \omega}{\ln \varrho}$.

5 Experiments

We conduct experiments to help analyze the convergence property of our proposed algorithm ABN. We first introduce the experimental setup and then give the analyses based on experimental results.

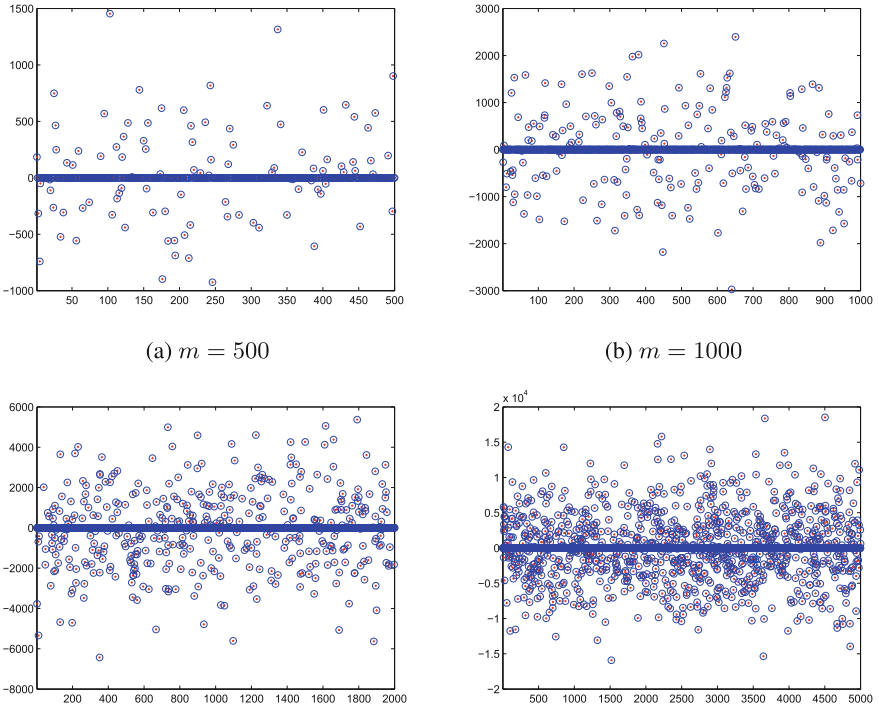
5.1 Experimental Setup

We evaluate our ABN algorithm by considering the Sparse Coding algorithm with ℓ_0 regularizer, that is, $f_i(\mathbf{x}_i) = \lambda|\mathbf{x}_i|_0$ and $g(\mathbf{x}) = \sum_i \|\mathbf{A}\mathbf{x}_i - \mathbf{y}_i\|^2$. We test the ABN on synthetic data $\mathbf{x}_i$s ($\mathbf{x}_i \in \mathbb{R}^m$) which are sparse and somehow high dimensional. In addition, the observe vector $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + noise$. The dictionary $\mathbf{A}$ in our experiments contains orthogonal bases. The regularizer parameter λ is set as $5e3$ in all the experiments and we carefully select λ to ensure the ground truth be the local minimizer of the problem. All algorithms are implemented in Matlab and executed on an Intel(R) Core(TM) $i5$-4200M CPU (2.50 GHz) with 8 GB memory.

5.2 Experimental Evaluations and Analyses

For solving sparse coding with ℓ_0 penalty, the Bregman distance $d_{\varphi(\mu^k)}(\mathbf{x}, \mathbf{x}^k)$ is chosen as a special Mahalanobis distance with $\mathbf{Q} = \mu\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ and the $\mu^k > \lambda_m$ with λ_m denotes the maximum eigenvalue of $\mathbf{A}^\top\mathbf{A}$. Some fixed parameters are set as $\nu^l = 1$ for all l, $l_{\max} = 20$ and $\epsilon = 1e^{-3}$.

Firstly, the experimental results on applying ABN to recover the signal with different dimensions are shown in Fig. 1. The blue circles are the ground truth of



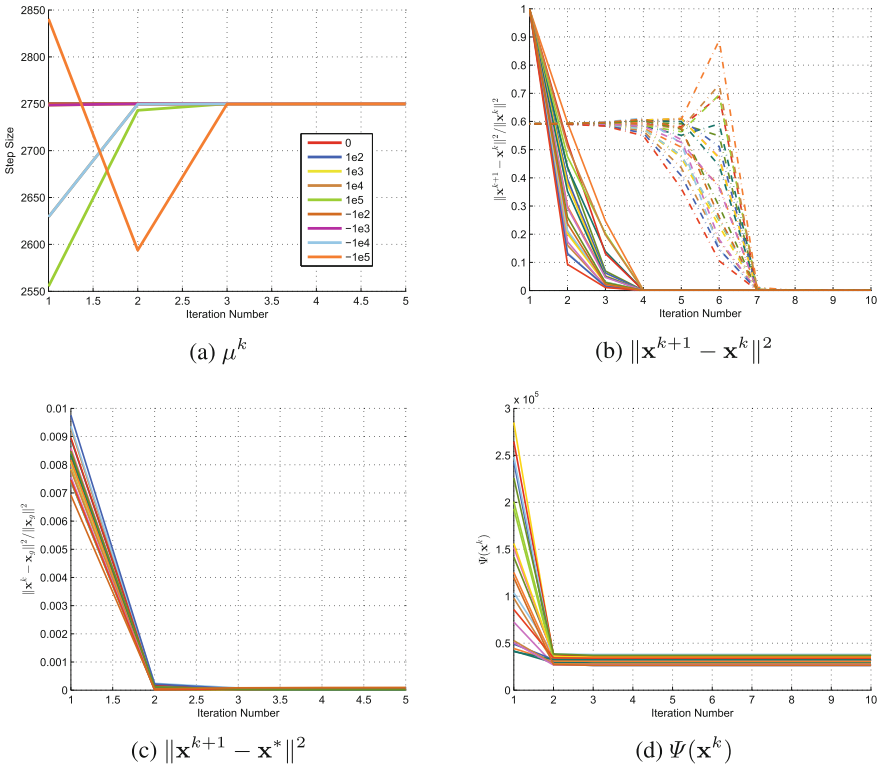
(a) $m = 500$ (b) $m = 1000$

(c) $m = 2000$ (d) $m = 5000$

Fig. 1. The recovered signals with different dimensions, which are recovered by the proposed ABN with learnt step sizes and iteration number. (Color figure online)

signals, and the red dots are the recovered signals by our proposed ABN method with learnt step sizes and iteration number. We give this result to demonstrate the accuracy on recovering signals solved by our algorithm.

Secondly, we give experiments to help verify the convergence property of ABN with learnt parameters calculated by Algorithm 1. It should be mentioned that some parameters in Fig. 2 are fixed with $m = 50$ and $\mu^{k,0} = 1.3\lambda_m$. By giving different initializations, the changes of the step sizes are given in Fig. 2(a). It can be seen from the figure that the step sizes changes a lot when the initialization (e.g., $\mathbf{x}_{ij}^0 = 1e5$) is far from the global minimizer. Then, we give the convergence performances on the test data in Fig. 2(b)–(d). Specifically, we give the relative error $(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2)/\|\mathbf{x}^k\|^2$ in Fig. 2(b). The solid lines in Fig. 2(b) are corresponding to the relative error curves with learnt μ^ks while the dashed lines are with the curves of fixed $\mu^k = 1.3\lambda_m$. Obviously, the convergence property of ABN with learnt step sizes are quite similar on various test data. Moreover, it is faster and more robust of the learnt algorithm to converge to a small neighborhood of the local minimizer. The last two figures in Fig. 2 are the relative error with the ground truth $\mathbf{x}^*$, i.e. $(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2)/\|\mathbf{x}^*\|^2$ and the changes of the



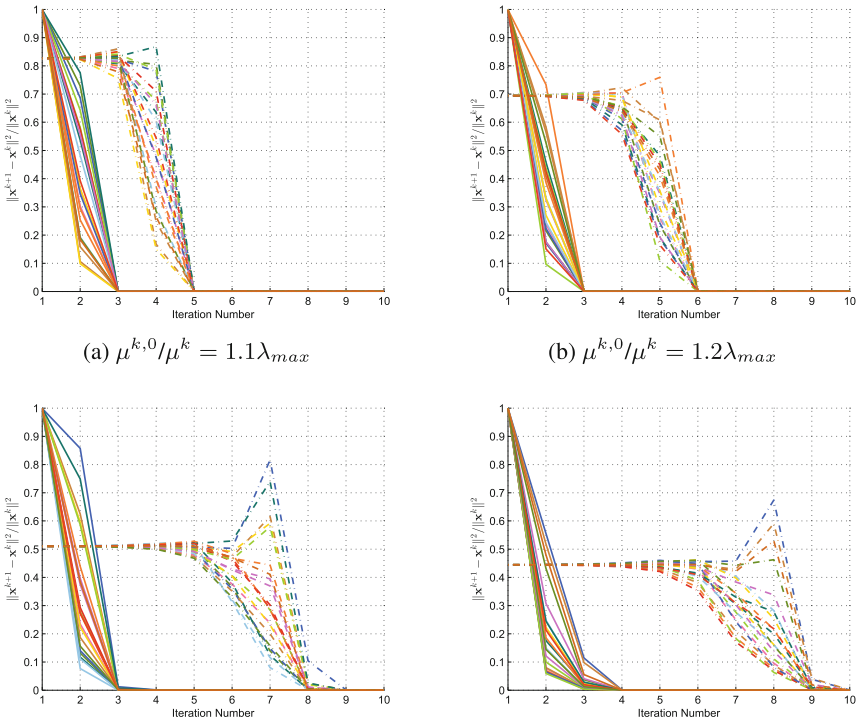
(a) μ^k (b) $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$

(c) $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2$ (d) $\Psi(\mathbf{x}^k)$

Fig. 2. Convergence property on (a) step size changes with various initial $\mathbf{x}^0$ and (b)–(d) the convergence properties of ABN on test data with adaptively step sizes learnt by Algorithm 1. The dotted curves in (b) can be regarded as one-block PALM [21] with fixed step sizes.

objective value $\Psi(\mathbf{x}^k)$. Both of them show the efficiency of our learnt algorithm ABN.

In the last, we show some evidences on $\mu^{k,0}$ in Fig. 3 We can obtain from the figures in Fig. 3 that the step size μ^k does affect the convergence rate. It seems like that a larger μ^k we have, a slower converge it is. For example, it takes 5 steps (dashed lines in Fig. 3(a)) for the case of $\mu^k = 1.1\lambda_m$ to converge to a relatively small error but it takes 9 steps for $\mu^k = 1.5\lambda_m$ in Fig. 3(d). However, the performances of the learnt algorithm ABN does not have such big differences. The solid lines are the convergence curves of the learnt algorithm ABN. The initial $\mu^{k,0}$s are set to the corresponding values of the fixed step sizes. It can be seen from the figures that it has 1 step difference between $\mu^{k,0} = 1.1\lambda_m$ and $\mu^{k,0} = 1.5\lambda_m$. This also demonstrates the robust and efficiency of ABN on deciding appropriate step sizes of iterations.



(a) $\mu^{k,0}/\mu^k = 1.1\lambda_{max}$ (b) $\mu^{k,0}/\mu^k = 1.2\lambda_{max}$

(c) $\mu^{k,0}/\mu^k = 1.4\lambda_{max}$ (d) $\mu^{k,0}/\mu^k = 1.5\lambda_{max}$

Fig. 3. The impact of $\mu^{k,0}$ and μ^k on ABN with learnt parameters (solid curves) and one-block PALM [21] with fixed step sizes (dotted curves).

6 Conclusion

We in this paper propose a learning-based algorithm network ABN for solving non-convex and non-smooth optimization problems. The basis algorithm of ABN is AMBM, which solves each subproblem with a specialized Bregman distance. Our proposed AMBM is more general and flexible than existing algorithm, and is proved to receive so far the best convergence result for general non-convex and non-smooth optimization problems. Different from the conventional algorithms, we propose the algorithm network ABN on the basis of AMBM to adaptively learn the algorithm parameters from training data to rapidly converge to desired solutions of the problems. Thus, our proposed ABN is an efficient and converged algorithm that adaptively tunes the algorithm parameters during iterations for fast converging performance in practice.

Acknowledgements. Risheng Liu is supported by the National Natural Science Foundation of China (Nos. 61672125, 61300086, 61572096, 61432003 and 61632019), the Fundamental Research Funds for the Central Universities (DUT2017TB02) and the Hong Kong Scholar Program (No. XJ2015008). Zhixun Su is supported by National

Natural Science Foundation of China (No. 61572099) and National Science and Technology Major Project (No. 2014ZX04001011).

References

1. Berry, M.W., Brown, M., Langvill, A.N., Pauca, V.P., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Comput. Stat. Data Anal. **52**(1), 155–173 (2007)
2. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: ACM SIGKDD (2006)
3. Zuo, W., Meng, D., Zhang, L., Feng, X., Zhang, D.: A generalized iterated shrinkage algorithm for non-convex sparse coding. In: ICCV, pp. 217–224 (2013)
4. Sandler, R., Lindenbaum, M.: Nonnegative matrix factorization with earth movers distance metric for image analysis. IEEE TPAMI **33**(8), 1590–1602 (2011)
5. Wang, Z., Ling, Q., Huang, T.S.: Learning deep ℓ_0 encoders. In: AAAI
6. Gong, P., Zhang, C., Lu, Z., Huang, J.Z., Ye, J.: A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In: ICML (2013)
7. Lu, C., Tang, J., Yan, S., Lin, Z.: Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. IEEE TIP **25**(2), 829–839 (2016)
8. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE TPAMI **33**(8), 1548–1560 (2010)
9. Benetos, E., Kotropoulos, C.: Non-negative tensor factorization applied to music genre classification. IEEE TASLP **18**(8), 1955–1967 (2010)
10. Jia, S., Qian, Y.: Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE TGRS **47**(1), 161–173 (2009)
11. Peng, X., Lu, C., Yi, Z., Tang, H.: Connections between nuclear-norm and frobenius-norm-based representations. IEEE TNNLS
12. Deng, Y., Bao, F., Dai, Q.: A unified view of nonconvex heuristic approach for low-rank and sparse structure learning. In: Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing
13. Yuan, G., Ghanem, B.: A proximal alternating direction method for semi-definite rank minimization. In: AAAI (2016)
14. Wang, Y., Liu, R., Song, X., Su, Z.: Linearized alternating direction method with penalization for nonconvex and nonsmooth optimization. In: AAAI (2016)
15. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2001)
16. Wang, Y.X., Zhang, Y.J.: Nonnegative matrix factorization: a comprehensive review. IEEE TKDE **25**(6), 1336–1353 (2013)
17. Shi, J., Ren, X., Dai, G., Wang, J.: A non-convex relaxation approach to sparse dictionary learning. In: CVPR (2011)
18. Zhang, C.H.: Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. **38**(2), 894–942 (2010)
19. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. MP **116**(1–2), 5–16 (2009)
20. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-lojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
21. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. MP **146**(1–2), 459–494 (2014)

22. Xu, Y., Yin, W.: A globally convergent algorithm for nonconvex optimization based on block coordinate update, arXiv preprint arXiv:1410.1386
23. Li, H., Lin, Z.: Accelerated proximal gradient methods for nonconvex programming. In: NIPS (2015)
24. Frankel, P., Garrigos, G., Peypouquet, J.: Splitting methods with variable metric for kurdyka-łojasiewicz functions and general convergence rates. J. Optim. Theory Appl. **165**(3), 874–900 (2015)
25. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE TIP **15**(12), 3736–3745 (2007)
26. Pinghua, G., Zhang, C., Lu, Z., Huang, J., Jieping, Y.: A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization porblems. In: ICML (2013)
27. Zuo, W., Ren, D., Gu, S., Lin, L.: Discriminative learning of iteration-wise priors for blind deconvolution. In: CVPR (2015)
28. Gregor, K., Lecun, Y.: Learning fast approximations of sparse coding. In: ICML (2010)
29. Foucart, S., Lai, M.-J.: Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0<q \leq 1$. ACHA **26**(3), 395–407 (2009)
30. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
31. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with bregman divergences. JMLR **6**(4), 1705–1749 (2005)
32. Fischer, A.: Quantization and clustering with bregman divergences. J. Multivar. Anal. **101**(9), 2207–2221 (2010)
33. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l_0 gradient minimization. ACM TOG **30**(6), 174 (2011)
34. Kang, Y., Zhang, Z., Li, W.: On the global convergence of majorization minimization algorithms for nonconvex optimization problems, arXiv preprint arXiv:1504.07791
35. Nocedal, J., Wright, S.: Numerical Optimization. Springer Science & Business Media, New York (2006). doi:10.1007/978-0-387-40065-5
36. Sra, S., Nowozin, S., Wright, S.J.: Optimization for Machine Learning. MIT Press, Cambridge (2011)
37. Dempe, S.: Foundations of Bilevel Programming. Nonconvex Optimization & Its Applications, vol. 61
38. Ochs, P., Ranftl, R., Brox, T., Pock, T.: Bilevel optimization with nonsmooth lower level problems. In: Aujol, J.-F., Nikolova, M., Papadakis, N. (eds.) SSVM 2015. LNCS, vol. 9087, pp. 654–665. Springer, Cham (2015). doi:10.1007/978-3-319-18461-6_52
39. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: CVPR (2014)

http://www.springer.com/978-3-319-67776-7

Intelligence Science and Big Data Engineering
7th International Conference, IScIDE 2017, Dalian, China, September 22-23, 2017, Proceedings
Sun, Y.; Lu, H.; Zhang, L.; Yang, J.; Huang, H. (Eds.)
2017, XVI, 689 p. 234 illus., Softcover
ISBN: 978-3-319-67776-7