Binno: A 1st-order method for Bi-level Nonconvex Nonsmooth Optimization for Matrix Factorizations

Laura Selicato^{a,b,*}, Flavia Esposito^b, Andersen Ang^c

^a National Research Council (CNR), Water Research Institute (IRSA) Bari, Italy
 ^b University of Bari Aldo Moro, Department of Mathematics Bari, Italy
 ^c School of Electronics and Computer Science, University of Southampton, UK

Abstract

In this work, we develop a method for nonconvex, nonsmooth bi-level optimization and we introduce Binno, a first order method that leverages proximal constructions together with carefully designed descent conditions and variational analysis. Within this framework, Binno provably enforces a descent property for the overall objective surrogate associated with the bi-level problem. Each iteration performs blockwise proximal-gradient updates for the upper and the lower problems separately and then forms a calibrated, block-diagonal convex combination of the two tentative iterates. A linesearch selects the combination weights to enforce simultaneous descent of both level-wise objectives, and we establish conditions guaranteeing the existence of such weights together with descent directions induced by the associated proximal-gradient maps. We also apply Binno in the context of sparse low-rank factorization, where the upper level uses elementwise ℓ_1 penalties and the lower level uses nuclear norms, coupled via a Frobenius data term. We test Binno on synthetic matrix and a real traffic-video dataset, attaining lower relative reconstruction error and higher peak signal-to-noise ratio than some standard methods.

Keywords: Nonconvex Optimization, Bi-level Optimization, Optimization

Algorithm, Matrix Factorization 2008 MSC: 65K10, 90C26, 90C30,

Email address: lauraselicato@cnr.it (Laura Selicato)

^{*}Corresponding author

1. Introduction

Bi-level optimization problems consist of two nested optimization tasks organized in a hierarchical structure. The upper-level problem determines variables that influence the lower-level problem. A solution of a bi-level problem is optimal with respect to (wrt) both levels, but under a hierarchy: the upper-level decision anticipates and guides the optimal response of the lower level. Such optimization settings naturally arise in decision science and learning problems, where dependencies exist between two coupled optimization processes [1–3].

In this paper, we are interested in the following bi-level problem

$$\underset{\boldsymbol{x} \in \mathbb{R}^{n}, \boldsymbol{y} \in \mathbb{R}^{m}}{\operatorname{argmin}} \left\{ \psi_{1}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq f_{1}(\boldsymbol{x}) + g_{1}(\boldsymbol{y}) + H(\boldsymbol{x}, \boldsymbol{y}) \right\}$$
s.t. $(\boldsymbol{x}, \boldsymbol{y}) \in \underset{\boldsymbol{x} \in \mathbb{R}^{n}, \boldsymbol{y} \in \mathbb{R}^{m}}{\operatorname{argmin}} \left\{ \psi_{2}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq f_{2}(\boldsymbol{x}) + g_{2}(\boldsymbol{y}) + H(\boldsymbol{x}, \boldsymbol{y}) \right\},$ (1)

where the upper level problem concerns the minimization of a non-convex nonsmooth function $\psi_1: \mathbb{R}^n \times \mathbb{R}^m \to (-\infty, +\infty]$ with

- $f_1: \mathbb{R}^n \to (-\infty, +\infty]$ and $g_1: \mathbb{R}^m \to (-\infty, +\infty]$ are convex, proper and lower semicontinuous functions;
- $H: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ a C^1 function block-wise (with respect to one variable at a time).

Referring to (1), we note that the lower level problem has the same structure as the upper level one, with $f_2 \neq f_1$ and $g_2 \neq g_1$. Problem (1) with $\psi_2 = 0$ frequently arises in machine learning scenario, where two variables \boldsymbol{x} and \boldsymbol{y} are governed by regularizers f_1 and g_1 that encode constraints or feasible regions. A classic example arises when these functions represent indicator functions of half-space constraints [4, 5].

For convex inner problems and strongly convex (often smooth) outer objectives, first-order bi-level methods with provable rates are available. A prominent line is the Sequential Averaging Method (SAM) and its bi-level specialization BiG-SAM, which view bi-level optimization as a fixed-point selection problem and average a proximal-gradient step for the inner composite with a (contractive) gradient step for the outer objective; sublinear O(1/k) rates are known under standard Lipschitz/strong-convexity assumptions [6]. Recent variants relax projections, incorporate inertial or conditional-gradient

updates, or target "simple" convex bi-level problems [7, 8]. However, two important gaps remain: (i) most analyses assume convexity (often strong convexity) and smoothness at the outer level; (ii) existing SAM-type schemes average one upper-level step with one inner fixed-point map, while block-structured, composite, possibly nonconvex models with explicit proximal treatment at both levels have received less attention, as far as we known.

Motivated by Proximal Alternating Linearized Minimization (PALM) for nonconvex, nonsmooth single-level composites [9], we propose a bi-level generalization that executes blockwise proximal-gradient updates for both levels and then forms a calibrated convex combination of the upper- and lower-driven iterates to steer the sequence toward an upper-level preferred solution within the lower-level solution set. This design preserves the modularity and per-block simplicity of proximal methods while embedding bi-level guidance directly into the iteration.

1.1. Contribution and Paper Organization

In this paper, we propose a bi-level generalization of PALM to solve Problem (1) that we call Binno (Bi-level nonconvex nonsmooth optimization). In particular, in section 2 we give notations and auxiliary results that are needed for the forthcoming sections. We detail the proposed method Binno and some theoretical considerations in section 3. We apply Binno in a matrix factorization problem in section 4 where a sparse low-rank representation is required from a data matrix. section 5, explains numerical experiments comparing Binno to standard methods in sparse low-rank applications on synthetic and real datasets.

2. Background and mathematical tools

This work relies on the mathematical tools detailed below.

Proximal gradient update. In optimization problems, in the form of (1) with $\psi_2 = 0$, the proximal gradient method [4, 10, 11] is a widely used solution approach. It aims to solve optimization problems in this form:

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{argmin}} \ p(\boldsymbol{x}) + q(\boldsymbol{x}) \tag{2}$$

where $p: \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable, L-smooth (gradient is L-Lipschitz) and $q: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex, proper and lower semicontinuous. The

proximal gradient update run iterations rules as:

$$\boldsymbol{x}_{k+1} = \operatorname{prox}_q^{\nu} (\boldsymbol{x}_k - \nu \nabla p(\boldsymbol{x}_k)),$$

where $k \in \mathbb{N}$ is iteration, $\nu > 0$ is stepsize, and $\operatorname{prox}_q^{\nu} : \mathbb{R}^n \to \mathbb{R}^n$ is the proximal operator defined as

$$\mathrm{prox}_q^{\nu}(\boldsymbol{x}) \coloneqq \operatorname*{argmin}_{\boldsymbol{\xi} \in \mathbb{R}^n} \bigg\{ q(\boldsymbol{\xi}) + \frac{1}{2\nu} \|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \bigg\}, M_{\nu f} \coloneqq \min_{\boldsymbol{\xi} \in \mathbb{R}^n} \bigg\{ q(\boldsymbol{\xi}) + \frac{1}{2\nu} \|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \bigg\},$$

with $M_{\nu f}$ is the Moreau envelope associated to f. Under certain assumptions on p, q, ν (see [11]), the sequence $\{x_k\}_{k\in\mathbb{N}}$ produced by proximal gradient update converges to a global minimizer of the Problem (2).

The following lemma is useful later.

Lemma 1 (Theorem 12.30, [12]). If f is convex, proper and lower semi-continuous, then $M_{\nu f} \in C^1$ and for all $\mathbf{x} \in \mathbb{R}^n$, we have $\nabla M_{\nu f} = \frac{1}{\nu}(\mathbf{x} - prox_{\nu f}(\mathbf{x}))$. Consequently, its gradient is $1/\nu$ Lipschitz continuous.

We use G to denote the proximal gradient map of the proximal gradient operator. For problem (2), let ∂q denotes a subgradient of function q, then

$$m{x}_{+} = \operatorname{prox}_{q}^{
u}(m{x} -
u
abla p(m{x})) = m{x} -
u m{G}(m{x})$$

 $m{G}(m{x}) = \frac{1}{
u}(m{x} - \operatorname{prox}_{q}^{
u}(m{x} -
u
abla p(m{x}))) \in \nabla p(m{x}) + \partial q(m{x} -
u m{G}(m{x})).$

Proximal Alternating Linearized Minimization (PALM). Since the structure of each level of our Problem 1, we recall PALM algorithm [9]. It considers

$$\underset{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^m}{\operatorname{argmin}} f(\boldsymbol{x}) + g(\boldsymbol{y}) + H(\boldsymbol{x}, \boldsymbol{y}),$$

that is Problem (1) without the bi-level structure and with functions f, g, H as described previously. The PALM algorithm performs the proximal gradient update alternatively on each subproblems of (1) as follow:

$$oldsymbol{x}_{k+1} = \operatorname{prox}_f^{
u} ig(oldsymbol{x}_k -
u
abla_{oldsymbol{x}} H(oldsymbol{x}_k, oldsymbol{y}_k) ig), \ oldsymbol{y}_{k+1} = \operatorname{prox}_q^{
u} ig(oldsymbol{y}_k -
u
abla_{oldsymbol{y}} H(oldsymbol{x}_{k+1}, oldsymbol{y}_k) ig).$$

Sequential Averaging Method (SAM). Consider the problem

$$\min_{\boldsymbol{x}} \left\{ \varphi(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x}) \right\},\tag{3}$$

with f smooth function (Lipschitz gradient L_f) and g proper, lower semicontinuous, convex function. Consider a nonexpansive map $T: \mathbb{R}^n \to \mathbb{R}^n$ and a contraction $S: \mathbb{R}^n \to \mathbb{R}^n$, SAM generates the sequence

$$\boldsymbol{x}_k = \alpha_k S(\boldsymbol{x}_{k-1}) + (1 - \alpha_k) T(\boldsymbol{x}_{k-1}), \text{ with } \alpha_k \in (0, 1], \alpha_k \downarrow 0, \sum_k \alpha_k = \infty,$$

and converges to a point $\boldsymbol{x}^* \in \operatorname{Fix}(T) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : T(\boldsymbol{x}) = \boldsymbol{x}\}$. Also, it satisfies the variational inequality $\langle \boldsymbol{x}^* - S(\boldsymbol{x}^*), \, \boldsymbol{x} - \boldsymbol{x}^* \rangle \geq 0$ for all $\boldsymbol{x} \in \operatorname{Fix}(T)$; thus \boldsymbol{x}^* is the fixed point according to S. In a bi-level context, for the inner problem as (3) and outer problem $\min_{\boldsymbol{x} \in X^*} \omega(\boldsymbol{x})$, with ω strongly convex, smooth, and $\nabla \omega$ is L_{ω} -Lipschitz, define the ProxGrad map

$$T(\boldsymbol{x}) = \operatorname{prox}_{g}^{t}(\boldsymbol{x} - t\nabla f(\boldsymbol{x})), \qquad t \in (0, 1/L_{f}],$$

which is nonexpansive and satisfies $Fix(T) = X^*$, the solution set of the inner problem. For the outer problem, set the contraction

$$S(\mathbf{x}) = \mathbf{x} - s\nabla\omega(\mathbf{x}), \qquad s \in \left(0, \frac{2}{L_{\omega} + \sigma}\right].$$

With these choices, SAM is to:

$$egin{aligned} oldsymbol{y}_k &= \operatorname{prox}_g^t ig(oldsymbol{x}_{k-1} - t
abla f(oldsymbol{x}_{k-1}) ig), \ oldsymbol{z}_k &= oldsymbol{x}_{k-1} - s
abla \omega(oldsymbol{x}_{k-1}), \ oldsymbol{x}_k &= lpha_k oldsymbol{z}_k + (1 - lpha_k) oldsymbol{y}_k, \end{aligned}$$

and the iterates converge to $x^* \in X^*$ solving the bi-level task via the first-order optimality condition $\langle \nabla \omega(x^*), x - x^* \rangle \geq 0$, for all $x \in X^*$.

3. The proposed method Binno

The idea of Binno is to iteratively and alternatively update each block of the variables by approximately solve each single-level subproblem using proximal gradient update, and then using a convex combination of the updated sequences from each single-level. In particular, starting from initial guess (x_0, y_0) , at each iteration $k \in \mathbb{N}$, we perform a PALM step (see Section 2) on the upper level problem, with a subscript u, using proximal gradient update obtaining (x_u, y_u) . Similarly, we perform the PALM step on the lower level problem (with a subscript l), obtaining (x_l, y_l) . Then we get (x_{k+1}, y_{k+1}) by a convex combination of $(\boldsymbol{x}_u, \boldsymbol{y}_u)$ and $(\boldsymbol{x}_l, \boldsymbol{y}_l)$. Fig.1 depicts a flow chart of the evolution of the sequence highlighting some issues that emerge (with question marks). We let $\tilde{\boldsymbol{x}}$ be the gradient-only update of \boldsymbol{x} (i.e., before applying the prox operator). Performing a simply convex combination, like in SAM, is not appropriate in this setting. Some problems are:

- 1. As $f_1 \neq f_2$, the upper- and lower-level of \boldsymbol{x} target different proximal maps, so \boldsymbol{x}_u and \boldsymbol{x}_l generally point toward distinct fixed points. This discrepancy complicates both the analysis and the effect of their averaging.
- 2. As $g_1 \neq g_2$, the updates of \boldsymbol{y} are conditionally computed from different \boldsymbol{x} iterates $(\boldsymbol{y}_u \text{ uses } \boldsymbol{x}_u \text{ while } \boldsymbol{y}_l \text{ uses } \boldsymbol{x}_l)$. This cross-level coupling induces potentially conflicting descent directions and a more intricate dynamic for \boldsymbol{y} .

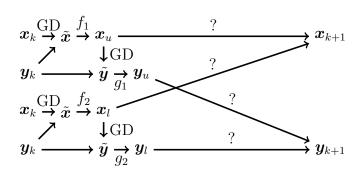


Figure 1: The structure of Binno. GD stands for gradient descent. In this work, our goal is to answer how to deal with question mark.

3.1. Theory of Binno

At each iteration k, we perform the following.

1. At the upper level subproblem, we perform a proximal gradient descent (ProxGrad) step on x_k while y_k is held fix

$$\boldsymbol{x}_{u} = \operatorname{prox}_{f_{1}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k})), \tag{4}$$

where $\nu > 0$ is a stepsize and $\operatorname{prox}_{f_1}^{\nu}$ is the prox operator of function f_1 under parameter ν . This step is performed at the upper level, so we name it \boldsymbol{x}_u with a subscript u.

2. At the upper level subproblem, we perform a ProxGrad step on y_k while x is held fixed at the most recent value x_u

$$\boldsymbol{y}_{u} = \operatorname{prox}_{q_{1}}^{\nu} (\boldsymbol{y}_{k} - \nu \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{u}, \boldsymbol{y}_{k})). \tag{5}$$

3. At the lower level subproblem, we perform a ProxGrad step on \boldsymbol{x}_k while \boldsymbol{y}_k is held fix

$$\boldsymbol{x}_{l} = \operatorname{prox}_{f_{2}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k})). \tag{6}$$

4. At the lower level subproblem, we perform a ProxGrad step on y_k while x is held fixed at the most recent value x_l

$$\mathbf{y}_{l} = \operatorname{prox}_{q_{2}}^{\nu} (\mathbf{y}_{k} - \nu \nabla_{\mathbf{y}} H(\mathbf{x}_{l}, \mathbf{y}_{k})). \tag{7}$$

5. We obtain the solution $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ by performing convex combination of $(\boldsymbol{x}_u, \boldsymbol{y}_u)$ and $(\boldsymbol{x}_l, \boldsymbol{y}_l)$, mathematically as

$$\begin{pmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_{k+1} \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_u \\ \boldsymbol{y}_u \end{pmatrix} + \begin{pmatrix} 1 - \alpha & 0 \\ 0 & 1 - \beta \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_l \\ \boldsymbol{y}_l \end{pmatrix}, \tag{8}$$

with α, β are some real constants to be determined.

Algorithm 1 summarizes each steps in a pseudo-code.

Algorithm 1: Binno for Problem (1)

- 1 Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^n$ and $\boldsymbol{y}_0 \in \mathbb{R}^m$
- **2** for k = 1, 2, ... do
- 3 | Upper-level update: Get (x_u, y_u) from (x_k, y_k) by (4), (5).
- 4 Lower-level update: Get $(\boldsymbol{x}_{\ell}, \boldsymbol{y}_{\ell})$ from $(\boldsymbol{x}_{k}, \boldsymbol{y}_{k})$ by (6), (7).
- Convex combination: Get $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ by combining $(\boldsymbol{x}_u, \boldsymbol{y}_u)$ and $(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)$ by (8).

We use a line search scheme for obtaining α, β such that we have simultaneous descent conditions on both ψ_1 and ψ_2 :

$$\psi_1(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}) \le \psi_1(\boldsymbol{x}_k, \boldsymbol{y}_k)$$
 and $\psi_2(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}) \le \psi_2(\boldsymbol{x}_k, \boldsymbol{y}_k)$.

Remark. Using the proximal gradient map (see section 2), we have

$$\mathbf{x}_{k+1} = \alpha \mathbf{x}_{u} + (1 - \alpha) \mathbf{x}_{l}
= \alpha \operatorname{prox}_{f_{1}}^{\nu} (\mathbf{x}_{k} - \nu \nabla_{\mathbf{x}} H(\mathbf{x}_{k}, \mathbf{y}_{k}))
+ (1 - \alpha) \operatorname{prox}_{f_{2}}^{\nu} (\mathbf{x}_{k} - \nu \nabla_{\mathbf{x}} H(\mathbf{x}_{k}, \mathbf{y}_{k}))
= \alpha (\mathbf{x}_{k} - \nu \mathbf{G}_{u}(\mathbf{x}_{k})) + (1 - \alpha) (\mathbf{x}_{k} - \nu \mathbf{G}_{l}(\mathbf{x}_{k}))
= \mathbf{x}_{k} - \nu (\alpha \mathbf{G}_{u}(\mathbf{x}_{k}) + (1 - \alpha) \mathbf{G}_{l}(\mathbf{x}_{k}))
= \mathbf{x}_{k} - \nu \mathbf{d}_{\mathbf{x}} \quad \text{with} \quad \mathbf{d}_{\mathbf{x}} = \alpha \mathbf{G}_{u}(\mathbf{x}_{k}) + (1 - \alpha) \mathbf{G}_{l}(\mathbf{x}_{k}).
\mathbf{y}_{k+1} = \beta \mathbf{y}_{u} + (1 - \beta) \mathbf{y}_{l}
= \beta \operatorname{prox}_{g_{1}}^{\nu} (\mathbf{y}_{k} - \nu \nabla_{\mathbf{y}} H(\mathbf{x}_{u}, \mathbf{y}_{k}))
+ (1 - \beta) \operatorname{prox}_{g_{2}}^{\nu} (\mathbf{y}_{k} - \nu \nabla_{\mathbf{y}} H(\mathbf{x}_{l}, \mathbf{y}_{k}))
= \beta (\mathbf{y}_{k} - \nu \mathbf{G}_{u}(\mathbf{y}_{k})) + (1 - \beta) (\mathbf{y}_{k} - \nu \mathbf{G}_{l}(\mathbf{y}_{k}))
= \mathbf{y}_{k} - \nu (\beta \mathbf{G}_{u}(\mathbf{y}_{k}) + (1 - \beta) \mathbf{G}_{l}(\mathbf{y}_{k}))
= \mathbf{y}_{k} - \nu \mathbf{d}_{\mathbf{y}} \quad \text{with} \quad \mathbf{d}_{\mathbf{y}} = \beta \mathbf{G}_{u}(\mathbf{y}_{k}) + (1 - \beta) \mathbf{G}_{l}(\mathbf{y}_{k}).$$

where

$$G_{u}(\boldsymbol{x}_{k}) = \frac{1}{\nu} (\boldsymbol{x}_{k} - prox_{f_{1}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}))),$$

$$G_{l}(\boldsymbol{x}_{k}) = \frac{1}{\nu} (\boldsymbol{x}_{k} - prox_{f_{2}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}))),$$

$$G_{u}(\boldsymbol{y}_{k}) = \frac{1}{\nu} (\boldsymbol{y}_{k} - prox_{g_{1}}^{\nu} (\boldsymbol{y}_{k} - \nu \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{u}, \boldsymbol{y}_{k}))),$$

$$G_{l}(\boldsymbol{y}_{k}) = \frac{1}{\nu} (\boldsymbol{y}_{k} - prox_{g_{2}}^{\nu} (\boldsymbol{y}_{k} - \nu \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{l}, \boldsymbol{y}_{k}))).$$

Moreover, d_x and d_y are descent directions:

for upper level wrt
$$\mathbf{x}$$
 if $\langle \partial \psi_1(\mathbf{x}_k, \mathbf{y}_k), \mathbf{d}_{\mathbf{x}} \rangle < 0$
 $\iff \langle \partial f_1(\mathbf{x}_k) + \nabla_{\mathbf{x}} H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{d}_{\mathbf{x}} \rangle < 0,$
for upper level wrt \mathbf{y} if $\langle \partial \psi_1(\mathbf{x}_u, \mathbf{y}_k), \mathbf{d}_{\mathbf{y}} \rangle < 0$
 $\iff \langle \partial g_1(\mathbf{y}_k) + \nabla_{\mathbf{y}} H(\mathbf{x}_u, \mathbf{y}_k), \mathbf{d}_{\mathbf{y}} \rangle < 0,$
for lower level wrt \mathbf{x} if $\langle \partial \psi_2(\mathbf{x}_k, \mathbf{y}_k), \mathbf{d}_{\mathbf{x}} \rangle < 0$
 $\iff \langle \partial f_2(\mathbf{x}_k) + \nabla_{\mathbf{x}} H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{d}_{\mathbf{x}} \rangle < 0,$
for lower level wrt \mathbf{y} if $\langle \partial \psi_2(\mathbf{x}_l, \mathbf{y}_k), \mathbf{d}_{\mathbf{y}} \rangle < 0$
 $\iff \langle \partial g_2(\mathbf{y}_k) + \nabla_{\mathbf{y}} H(\mathbf{x}_l, \mathbf{y}_k), \mathbf{d}_{\mathbf{y}} \rangle < 0.$

Thus, to solve the question mark in Figure 1, we have to prove the following theorem which provides the existence of (α, β) satisfying the descent conditions on both ψ_1, ψ_2 . **Theorem 1.** In the setting of Problem (1), there exists $\alpha \in [0, 1], \beta \in [0, 1]$ that the following conditions hold

$$\langle \partial f_1(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \quad \alpha \boldsymbol{G}_u(\boldsymbol{x}_k) + (1 - \alpha) \boldsymbol{G}_l(\boldsymbol{x}_k) \rangle < 0$$

$$\langle \partial f_2(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \quad \alpha \boldsymbol{G}_u(\boldsymbol{x}_k) + (1 - \alpha) \boldsymbol{G}_l(\boldsymbol{x}_k) \rangle < 0,$$

$$\langle \partial g_1(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_u, \boldsymbol{y}_k), \quad \beta \boldsymbol{G}_u(\boldsymbol{y}_k) + (1 - \beta) \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle < 0,$$

$$\langle \partial g_2(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_l, \boldsymbol{y}_k), \quad \beta \boldsymbol{G}_u(\boldsymbol{y}_k) + (1 - \beta) \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle < 0,$$

with

$$G_{u}(\boldsymbol{x}_{k}) = \frac{1}{\nu} (\boldsymbol{x}_{k} - prox_{f_{1}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}))),$$

$$G_{l}(\boldsymbol{x}_{k}) = \frac{1}{\nu} (\boldsymbol{x}_{k} - prox_{f_{2}}^{\nu} (\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}))),$$

$$G_{u}(\boldsymbol{y}_{k}) = \frac{1}{\nu} (\boldsymbol{y}_{k} - prox_{g_{1}}^{\nu} (\boldsymbol{y}_{k} - \nu \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{u}, \boldsymbol{y}_{k}))),$$

$$G_{l}(\boldsymbol{y}_{k}) = \frac{1}{\nu} (\boldsymbol{y}_{k} - prox_{g_{2}}^{\nu} (\boldsymbol{y}_{k} - \nu \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{l}, \boldsymbol{y}_{k}))).$$

To prove Theorem 1, we need some preliminary results.

Lemma 2. Let z be convex, proper, lower semicontinuous function, and $\mathbf{x}_0 \in int\ (dom\ z)$. Then $\partial z(\mathbf{x}_0)$ is a nonempty bounded set, i.e. there exists a constant c such that $\|\partial z(\mathbf{x}_0)\| \leq c$.

Proof. If z is closed and convex function, then $\partial z(x_0)$ is a nonempty bounded set [13, Theorem 3.1.15]. As a proper convex function is closed if and only if it is lower semi-continuous, we get the result.

Lemma 3. Under the assumptions and settings of Theorem 1, we have

$$\begin{aligned} \left| \left\langle \partial f_1(\boldsymbol{x}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\rangle \right| &< c_1 \| \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \|, \quad \left| \left\langle \partial f_2(\boldsymbol{x}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\rangle \right| &< c_2 \| \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \|, \\ \left| \left\langle \partial g_1(\boldsymbol{y}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{y}_k) \right\rangle \right| &< c_3 \| \boldsymbol{G}_{\Delta}(\boldsymbol{y}_k) \|, \quad \left| \left\langle \partial g_2(\boldsymbol{y}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{y}_k) \right\rangle \right| &< c_4 \| \boldsymbol{G}_{\Delta}(\boldsymbol{y}_k) \|, \end{aligned}$$

where $G_{\Delta} \in \{G_u, G_l\}$ and c_1, c_2, c_3, c_4 are constants for functions f_1, f_2, g_1, g_2 respectively, as in Lemma 2.

Proof. We prove the lemma for f_1 . By Cauchy-Schwarz inequality:

$$\left|\left\langle \partial f_1(\boldsymbol{x}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\rangle \right| \leq \left\| \partial f_1(\boldsymbol{x}_k) \right\| \left\| \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\| \stackrel{lemma \ 2}{\leq} c_1 \| \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \|.$$

The rest is similar for f_2, g_1, g_2 with their respective constants c_2, c_3, c_4 .

Lemma 4. Under the assumptions and settings of Theorem 1, given the bi-smooth constants L_1, L_2 for the gradient of H referred to \boldsymbol{x} and \boldsymbol{y} , respectively, then the following holds:

$$\begin{aligned}
&\left|\left\langle \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\rangle\right| &< L_1 \|\boldsymbol{G}_{\Delta}(\boldsymbol{x}_k)\|, \\
&\left|\left\langle \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{\Delta}, \boldsymbol{y}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{y}_k) \right\rangle\right| &< L_2 \|\boldsymbol{G}_{\Delta}(\boldsymbol{y}_k)\|, \quad \boldsymbol{x}_{\Delta} \in \{\boldsymbol{x}_u, \boldsymbol{x}_l\}.
\end{aligned}$$

Proof. We prove the lemma for \boldsymbol{x} . As H is bi-differentiable and bi-smooth, it implies $\|\nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k)\| \leq L_1$. By the Cauchy-Schwarz inequality

$$\left|\left\langle \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \boldsymbol{G}_{\Delta}(\boldsymbol{x}_k) \right\rangle \right| \leq \|\nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k)\| \|\boldsymbol{G}_{\Delta}(\boldsymbol{x}_k)\| \leq L_1 \|\boldsymbol{G}_{\Delta}(\boldsymbol{x}_k)\|.$$

The proof for \boldsymbol{y} is similar with $\nabla_{\boldsymbol{y}} H(\boldsymbol{x}_{\Delta}, \boldsymbol{y}_{k})$ and its respective bi-smooth constants L_{2} .

Lemma 5. Under the assumptions and settings of Theorem 1, then $\|G_{\Delta}(x_k)\|$ and $\|G_{\Delta}(y_k)\|$ are bounded, where $G_{\Delta} \in \{G_u, G_l\}$.

Proof. We prove the lemma for $G_u(x_k)$; the rest is similar. The prox operator is a contraction,

$$\|\operatorname{prox}_f(s) - \operatorname{prox}_f(z)\| \le \|s - z\|.$$

Take $f = f_1$ and choosing $s = x_k$ and $z = x_k - \nu \nabla_x H(x_k, y_k)$ we have

$$\left\|\operatorname{prox}_{f_1}^{\nu}(\boldsymbol{x}_k) - \operatorname{prox}_{f_1}^{\nu}(\boldsymbol{x}_k - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k))\right\| \leq \left\|\nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k)\right\| \stackrel{H \text{ smooth}}{\leq} \nu L_1.$$
(9)

Then

$$\begin{aligned} & \|\boldsymbol{G}_{u}(\boldsymbol{x}_{k})\| \\ &= \left\| \frac{1}{\nu} \left(\boldsymbol{x}_{k} - \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k}) + \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k}) - \operatorname{prox}_{f_{1}}^{\nu} \left(\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}) \right) \right) \right\| \\ &\leq \frac{1}{\nu} \left[\|\boldsymbol{x}_{k} - \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k})\| + \left\| \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k}) - \operatorname{prox}_{f_{1}}^{\nu} \left(\boldsymbol{x}_{k} - \nu \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}) \right) \right\| \right] \\ &\leq \frac{1}{\nu} \left[\|\boldsymbol{x}_{k} - \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k})\| + \nu L_{1} \right] \\ &= \underbrace{\frac{1}{\nu} \|\boldsymbol{x}_{k} - \operatorname{prox}_{f_{1}}^{\nu}(\boldsymbol{x}_{k})\|}_{=\|\nabla M_{\nu f_{1}}(\boldsymbol{x}_{k})\|} + L_{1} \stackrel{lemma \ 1}{\leq} \underbrace{\frac{1}{\nu} + L_{1}}_{-1}. \end{aligned}$$

We are now ready to prove Theorem 1.

Proof of Theorem 1. We focus on \boldsymbol{x}_k , for \boldsymbol{y}_k the proof is similar. First we have

$$\frac{\left\langle \partial f_{1}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \alpha \boldsymbol{G}_{u}(\boldsymbol{x}_{k}) + (1 - \alpha) \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \right\rangle}{= \alpha \underbrace{\left\langle \partial f_{1}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \boldsymbol{G}_{u}(\boldsymbol{x}_{k}) \right\rangle}_{:=q_{1}} + (1 - \alpha) \left\langle \partial f_{1}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \right\rangle, \\
\left\langle \partial f_{2}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \alpha \boldsymbol{G}_{u}(\boldsymbol{x}_{k}) + (1 - \alpha) \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \right\rangle, \\
= \alpha \left\langle \partial f_{2}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \boldsymbol{G}_{u}(\boldsymbol{x}_{k}) \right\rangle \\
+ (1 - \alpha) \underbrace{\left\langle \partial f_{2}(\boldsymbol{x}_{k}) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \ \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \right\rangle}_{:=q_{2}}.$$

The terms q_1 is negative since $G_u(x_k)$ is a descent directions for the upper problem ψ_1 , disregarding the lower problem. It is similar for $q_2 < 0$.

For the other parts, we have:

$$\langle \partial f_{1}(\boldsymbol{x}_{k}), \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \rangle + \langle \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \rangle$$

$$\leq |\langle \partial f_{1}(\boldsymbol{x}_{k}), \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \rangle| + |\langle \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_{k}, \boldsymbol{y}_{k}), \boldsymbol{G}_{l}(\boldsymbol{x}_{k}) \rangle|$$

$$\leq c_{1} \|\boldsymbol{G}_{l}(\boldsymbol{x}_{k})\| + L_{1} \|\boldsymbol{G}_{l}(\boldsymbol{x}_{k})\|$$

$$= (c_{1} + L_{1}) \|\boldsymbol{G}_{l}(\boldsymbol{x}_{k})\| \stackrel{lemma \ 5}{\leq} (c_{1} + L_{1}) \left(\frac{1}{\nu} + L_{1}\right).$$

Then exists $k_1 := (c_1 + L_1)(\frac{1}{\nu} + L_1)$ such that

$$\langle \partial f_1(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \ \boldsymbol{G}_l(\boldsymbol{x}_k) \rangle \leq k_1.$$

Similarly, exists $k_2 := (c_2 + L_1)(\frac{1}{\nu} + L_1)$ such that

$$\langle \partial f_2(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k), \; \boldsymbol{G}_u(\boldsymbol{x}_k) \rangle \leq k_2.$$

Finally, let $\nabla_{\boldsymbol{x}} H_k = \nabla_{\boldsymbol{x}} H(\boldsymbol{x}_k, \boldsymbol{y}_k)$ we have

$$\alpha \langle \partial f_1(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H_k, \; \boldsymbol{G}_u(\boldsymbol{x}_k) \rangle + (1 - \alpha) \langle \partial f_1(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H_k, \; \boldsymbol{G}_l(\boldsymbol{x}_k) \rangle$$

 $\leq \alpha q_1 + (1 - \alpha) k_1,$

$$\alpha \langle \partial f_2(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H_k, \boldsymbol{G}_u(\boldsymbol{x}_k) \rangle + (1 - \alpha) \langle \partial f_2(\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}} H_k, \boldsymbol{G}_l(\boldsymbol{x}_k) \rangle$$

 $\leq \alpha k_2 + (1 - \alpha) q_2.$

Similarly for \boldsymbol{y} and $\boldsymbol{\beta}$ we have

• for the upper problem: let $q_3 := \langle \partial g_1(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_u, \boldsymbol{y}_k), \boldsymbol{G}_u(\boldsymbol{y}_k) \rangle$ and $k_3 := (c_3 + L_2)(\frac{1}{\nu} + L_2)$, then

$$\langle \partial g_1(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_u, \boldsymbol{y}_k), \quad \beta \boldsymbol{G}_u(\boldsymbol{y}_k) + (1 - \beta) \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle$$

$$= \beta \langle \partial g_1(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_u, \boldsymbol{y}_k), \boldsymbol{G}_u(\boldsymbol{y}_k) \rangle$$

$$+ (1 - \beta) \langle \partial g_1(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_u, \boldsymbol{y}_k), \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle$$

$$\leq \beta q_3 + (1 - \beta) k_3.$$

• for the lower problem: let $q_4 := \langle \partial g_2(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_l, \boldsymbol{y}_k), \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle$ and $k_4 := (c_4 + L_2)(\frac{1}{\nu} + L_2)$, then

$$\langle \partial g_2(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_l, \boldsymbol{y}_k), \quad \beta \boldsymbol{G}_u(\boldsymbol{y}_k) + (1 - \beta) \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle$$

$$= \beta \langle \partial g_2(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_l, \boldsymbol{y}_k), \boldsymbol{G}_u(\boldsymbol{y}_k) \rangle$$

$$+ (1 - \beta) \langle \partial g_2(\boldsymbol{y}_k) + \nabla_{\boldsymbol{y}} H(\boldsymbol{x}_l, \boldsymbol{y}_k), \boldsymbol{G}_l(\boldsymbol{y}_k) \rangle$$

$$\leq \beta k_4 + (1 - \beta) q_4.$$

Lastly, we join everything considering $\ell_i = |q_i|$ for $i = 1, \ldots, 4$

$$\alpha(-\ell_1 - k_1) \le -k_1 \implies 0 \le \frac{k_1}{\ell_1 + k_1} \le \alpha \le 1; \tag{10a}$$

$$\alpha(\ell_2 + k_2) \le \ell_2 \implies 0 \le \alpha \le \frac{\ell_2}{\ell_2 + k_2} \le 1;$$
 (10b)

$$\beta(-\ell_3 - k_3) \le -k_3 \implies 0 \le \frac{k_3}{\ell_3 + k_3} \le \beta \le 1.$$
 (10c)

$$\beta(\ell_4 + k_4) \le \ell_4 \quad \Longrightarrow \quad 0 \le \beta \le \frac{\ell_4}{\ell_4 + k_4} \le 1. \tag{10d}$$

The range for α, β gives the descent conditions in the theorem hold.

4. Application to Sparse Low Rank Factorization

In this section, we consider a sparse low rank Factorization (SLRF) problem [14] constructed as bi-level problem in (1).

Given a matrix $M \in \mathbb{R}_+^{m \times n}$, we aim to solve:

$$\underset{\mathbf{X} \in \mathbb{R}^{m \times r}}{\operatorname{argmin}} \quad \lambda_{1} \| \mathbf{X} \|_{1} + \lambda_{2} \| \mathbf{Y} \|_{1} + \frac{1}{2} \| \mathbf{X} \mathbf{Y} - \mathbf{M} \|_{F}^{2}$$

$$\text{s.t.} \quad (\mathbf{X}, \mathbf{Y}) \in \underset{\mathbf{X} \in \mathbb{R}^{m \times r}}{\operatorname{argmin}} \gamma_{1} \| \mathbf{X} \|_{*} + \gamma_{2} \| \mathbf{Y} \|_{*} + \frac{1}{2} \| \mathbf{X} \mathbf{Y} - \mathbf{M} \|_{F}^{2},$$

$$(\text{SLRF})$$

where $\|\cdot\|_1$ is the elementwise ℓ_1 -norm, $\|\cdot\|_F$ is the F-norm and $\|\cdot\|_*$ is the nuclear norm. The functions in (SLRF) according to (1) are $f_1: \mathbb{R}^{m \times r} \to \mathbb{R}$, $f_1(\boldsymbol{X}) = \lambda_1 \|\boldsymbol{X}\|_1$, $f_2: \mathbb{R}^{m \times r} \to \mathbb{R}$, $f_2(\boldsymbol{X}) = \lambda_2 \|\boldsymbol{X}\|_*$, $g_1: \mathbb{R}^{r \times n} \to \mathbb{R}$, $g_1(\boldsymbol{Y}) = \gamma_1 \|\boldsymbol{Y}\|_1$, $g_2: \mathbb{R}^{r \times n} \to \mathbb{R}$, $g_2(\boldsymbol{Y}) = \gamma_2 \|\boldsymbol{Y}\|_*$, $H: \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n} \to \mathbb{R}$, $H(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{M}\|_F^2$. We solve the problem with Binno method using the following steps.

X-update. We solve the upper level subproblem in (SLRF), performing a ProxGrad step on X_k while Y is held fix at the recent value

$$\boldsymbol{X}_{u} = \operatorname{prox}_{f_{1}}^{\nu} (\boldsymbol{X}_{k} - \nu \nabla_{\boldsymbol{X}} H(\boldsymbol{X}_{k}, \boldsymbol{Y})).$$

At the lower level subproblem (SLRF), we perform a ProxGrad step on X_k while Y is held fix at the recent value

$$X_l = \operatorname{prox}_{f_2}^{\nu} (X_k - \nu \nabla_X H(X_k, Y)).$$

We obtain X_{k+1} by performing convex combination of X_u and X_l , mathematically as $X_{k+1} = \alpha X_u + (1 - \alpha) X_l$.

Y-update. We solve the upper level subproblem in (SLRF), performing a ProxGrad step on Y_k while X is held fix at the recent value

$$Y_u = \operatorname{prox}_{g_1}^{\nu} (Y_k - \nu \nabla_Y H(X_u, Y_k)).$$

At the lower level subproblem (SLRF), we perform a ProxGrad step on Y_k while X is held fix at the recent value

$$Y_l = \text{prox}_{q_2}^{\nu} (Y_k - \nu \nabla_{Y} H(X_l, Y_k)).$$

We obtain Y_{k+1} by performing convex combination of Y_u and Y_l , mathematically as $Y_{k+1} = \beta Y_u + (1 - \beta) Y_l$.

Algorithm 2 summarizes the previous steps for solving problem (SLRF) with Binno.

4.1. Useful Theoretical Results

We first list some useful theoretical results for this section for generic matrices.

Proposition 1. For $f_1: \mathbb{R}^{m \times r} \to \mathbb{R}$ being the element-wise ℓ_1 norm, then $\|S\|_2 \leq \lambda_1 \sqrt{mr}$ for any $S \in \partial f_1$.

Algorithm 2: Binno for (SLRF) Problem

```
Input: M \in \mathbb{R}^{m \times n}, r
  1 Inizialization: \mathbf{X}_0 \in \mathbb{R}^{m \times r}; \mathbf{Y}_0 \in \mathbb{R}^{r \times n} for k = 1, 2, ... do
              Update for X:
  2
                  Upper-level update: \boldsymbol{X}_{k}^{u} = \operatorname{prox}_{f_{1}}^{\nu} \left( \boldsymbol{X}_{k-1} - \nu \nabla_{\boldsymbol{X}} H(\boldsymbol{X}_{k-1}, \boldsymbol{Y}_{k-1}) \right)
  3
                  Lower-level update: \boldsymbol{X}_{k}^{l} = \operatorname{prox}_{f_{2}}^{\nu} (\boldsymbol{X}_{k-1} - \nu \nabla_{\boldsymbol{X}} H(\boldsymbol{X}_{k-1}, \boldsymbol{Y}_{k-1}))
  4
                  Finding range for \alpha according to section 4.2
  \mathbf{5}
                   Convex combination: \mathbf{X}_k = \alpha \mathbf{X}_k^u + (1 - \alpha) \mathbf{X}_k^l
  6
              Update for Y:
  7
                  Upper-level update: \boldsymbol{Y}_{k}^{u} = \operatorname{prox}_{g_{1}}^{\nu} \left( \boldsymbol{Y}_{k-1} - \nu \nabla_{\boldsymbol{Y}} H(\boldsymbol{X}_{k}^{u}, \boldsymbol{Y}_{k-1}) \right)
  8
                  Lower-level update: \mathbf{Y}_k^l = \text{prox}_{q_2}^{\nu} (\mathbf{Y}_{k-1} - \nu \nabla_{\mathbf{Y}} H(\mathbf{X}_k^l, \mathbf{Y}_{k-1}))
  9
                  Finding range for \beta according to section 4.3
10
                  Convex combination: \mathbf{Y}_k = \beta \mathbf{Y}_k^u + (1 - \beta) \mathbf{Y}_k^u
11
      Output: X \in \mathbb{R}^{m \times r}, Y \in \mathbb{R}^{r \times n}
```

Proof. The function $f_1(\mathbf{X}) = \lambda_1 \|\mathbf{X}\|_1 = \lambda_1 \sum_{ij} |x_{ij}|$ is not differentiable at $x_{ij} = 0$ but subdifferentiable:

$$\partial \|\boldsymbol{X}\|_{1} = \left\{ \boldsymbol{P} \in \mathbb{R}^{m \times r} : p_{ij} \in \begin{cases} \text{sign } x_{ij} & \text{if } x_{ij} \neq 0; \\ p \in [-1, 1] & \text{if } x_{ij} = 0. \end{cases} \right\}.$$

Let
$$\mathbf{S} \in \lambda_1 \partial \|\mathbf{X}\|_1$$
 be any element of the subdifferential. Then $\|\mathbf{S}\|_2 \leq \|\mathbf{S}\|_F = \sqrt{\sum_{ij} s_{ij}^2} \leq \sqrt{\sum_{ij} |s_{ij}|^2} = \lambda_1 \sqrt{mr}$.

Proposition 2. For $f_2 : \mathbb{R}^{m \times r} \to \mathbb{R}$ as the nuclear norm, then $\|\mathbf{S}\|_2 \leq 2\gamma_1$ for any $\mathbf{S} \in \partial f_2$.

Proof. Let $X \in \mathbb{R}^{m \times r}$, the nuclear norm $\|X\|_* = \sum_{i=1}^r \sigma_i(X)$ is not differentiable but subdifferentiable. Consider the SVD $X = U\Sigma V^{\top}$ with k = rank(X), $U \in \mathbb{R}^{m \times k}$, $\Sigma = \text{Diag}(\sigma_i(X)) \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{r \times k}$, the subdifferential of $\|X\|_*$ is [15]

$$\partial \| \boldsymbol{X} \|_* = \{ \boldsymbol{U} \boldsymbol{V}^\top + \boldsymbol{W} \mid \boldsymbol{W} \in \mathbb{R}^{m \times r}, \ \boldsymbol{U}^\top \boldsymbol{W} = \boldsymbol{0}, \ \boldsymbol{W} \boldsymbol{V} = \boldsymbol{0}, \ \| \boldsymbol{W} \|_2 \le 1 \}.$$

We show
$$\|\partial f_2(\boldsymbol{X})\|_2 \le c_2$$
 in lemma 3. Let $\boldsymbol{S} \in \gamma_1 \partial \|\boldsymbol{X}\|_*$, so $\|\boldsymbol{S}\|_2 = \gamma_1 \|\boldsymbol{U}\boldsymbol{V}^\top + \boldsymbol{W}\|_2 \le \gamma_1 (\|\boldsymbol{U}\boldsymbol{V}^\top\|_2 + \|\boldsymbol{W}\|_2) \le \gamma_1 (1+1) = 2\gamma_1$.

Lemma 6. For the elementwise matrix ℓ_1 -norm and a matrix $\mathbf{X} \in \mathbb{R}^{m \times r}$, then $\|\mathbf{X} - prox_{\lambda_1\|\cdot\|_1}^{\nu}(\mathbf{X})\|_2 \leq \nu \lambda_1 \sqrt{mr}$.

Proof. The operator $\operatorname{prox}_{\lambda_1\|\cdot\|_1}^{\nu}$ is the soft-thresholding operator [11], thus

$$\left[\boldsymbol{X} - \operatorname{prox}_{\lambda_1 \| \cdot \|_1}^{\nu}(\boldsymbol{X})\right]_{ij} = \begin{cases} \nu \lambda_1 & \text{if } \boldsymbol{X}_{ij} < \nu \lambda_1, \\ \boldsymbol{X}_{ij} & \text{if } |\boldsymbol{X}_{ij}| \leq \nu \lambda_1, \\ -\nu \lambda_1 & \text{if } \boldsymbol{X}_{ij} < -\nu \lambda_1. \end{cases}$$

So
$$\|\boldsymbol{X} - \operatorname{prox}_{\lambda_1 \| \cdot \|_1}^{\nu}(\boldsymbol{X})\|_2 \le \|\boldsymbol{X} - \operatorname{prox}_{\lambda_1 \| \cdot \|_1}^{\nu}(\boldsymbol{X})\|_F \le \sqrt{\sum_{ij} (\nu \lambda_1)^2} = \nu \lambda_1 \sqrt{mr}.$$

Lemma 7. For the nuclear norm, then $\|\mathbf{X} - prox_{\gamma_1\|\cdot\|_*}^{\nu}(\mathbf{X})\|_2 \leq \gamma_1 \nu$.

Proof. Consider the nuclear norm, then $\operatorname{prox}_{\gamma_1\|.\|_*}^{\nu}$ as

$$\operatorname{prox}_{\gamma_1\|\cdot\|_*}^{\nu}(\boldsymbol{X}) = \underset{\boldsymbol{A}}{\operatorname{argmin}} \ \gamma_1 \nu \|\boldsymbol{A}\|_* + \frac{1}{2} \|\boldsymbol{A} - \boldsymbol{X}\|^2 = \operatorname{SVT}_{\gamma_1 \nu}(\boldsymbol{X})$$

is the Singular Value Thresholding (SVT) by the Von Neumann inequality[16]. The SVT of $\boldsymbol{X} \stackrel{SVD}{=} \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$ is SVT $_{\gamma_1\nu}(\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{D}_{\gamma_1\nu}(\boldsymbol{\Sigma})\boldsymbol{V}^{\top}$, with $\boldsymbol{D}_{\gamma_1\nu}(\boldsymbol{\Sigma}) = \mathrm{Diag}([\sigma_i - \gamma_1\nu]_+)$ is the soft-thresholded $\boldsymbol{\Sigma}$ with σ_i as the singular values of \boldsymbol{X} and $[a]_+ = \max\{a, 0\}$. Then,

$$\begin{aligned} \left\| \boldsymbol{X} - \text{SVT}_{\gamma_1 \nu}(\boldsymbol{X}) \right\|_2 &= \left\| \boldsymbol{U} \left(\boldsymbol{\Sigma} - \boldsymbol{D}_{\gamma_1 \nu}(\boldsymbol{\Sigma}) \right) \boldsymbol{V}^\top \right\|_2 &= \left\| \boldsymbol{\Sigma} - \boldsymbol{D}_{\gamma_1 \nu}(\boldsymbol{\Sigma}) \right\|_2 \\ &= \max_i \left| \sigma_i - [\sigma_i - \gamma_1 \nu]_+ \right|. \end{aligned}$$

We have two cases

- Case 1 $\sigma_i > \gamma_1 \nu$: then $|\sigma_i [\sigma_i \gamma_1 \nu]_+| = |\sigma_i (\sigma_i \gamma_1 \nu)| = |\gamma_1 \nu| = \gamma_1 \nu$.
- Case 2 $\sigma_i \leq \gamma_1 \nu$: then $|\sigma_i [\sigma_i \gamma_1 \nu]_+| = |\sigma_i 0| = |\sigma_i| = \sigma_i \leq \gamma_1 \nu$.

Thus
$$\|\boldsymbol{X} - \text{SVT}_{\gamma_1 \nu}(\boldsymbol{X})\|_2 \le \gamma_1 \nu$$
.

Lemma 8. For $H: \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n} \to \mathbb{R}$ defined as $\frac{1}{2} \| \boldsymbol{M} - \boldsymbol{X} \boldsymbol{Y} \|_F^2$, the bismooth constant wrt \boldsymbol{X} is $L_1 = \| \boldsymbol{Y} \boldsymbol{Y}^\top \|_2$ and wrt \boldsymbol{Y} is $L_2 = \| \boldsymbol{X}^\top \boldsymbol{X} \|_2$.

Proof. The constants are $L_1 = \|\nabla_{\boldsymbol{X}}^2 H(\boldsymbol{X}, \boldsymbol{Y})\| = \|\boldsymbol{Y}\boldsymbol{Y}^\top\|_2$, the spectral norm of $\boldsymbol{Y}\boldsymbol{Y}^\top$, and $L_2 = \|\nabla_{\boldsymbol{Y}}^2 H(\boldsymbol{X}, \boldsymbol{Y})\| = \|\boldsymbol{X}^\top \boldsymbol{X}\|_2$.

4.2. Finding the constant α

Here, the results are for any X and Y according to the iteration structure in Algorithm 2. To find constants in (10a) and (10b) we need to find

$$k_1 = (c_1 + L_1)(1/\nu + L_1); l_1 = \left| \langle \partial f_1(\mathbf{X}) + \nabla_{\mathbf{X}} H(\mathbf{X}, \mathbf{Y}), \mathbf{G}_u(\mathbf{X}) \rangle \right|;$$

$$k_2 = (c_2 + L_1)(1/\nu + L_1); l_2 = \left| \langle \partial f_2(\mathbf{X}) + \nabla_{\mathbf{X}} H(\mathbf{X}, \mathbf{Y}), \mathbf{G}_l(\mathbf{X}) \rangle \right|;$$

where c_1, c_2 are as in Lemma 3 for f_1 and f_2 respectively; L_1 is the bi-smooth constant for H wrt \boldsymbol{X} . In particular, we have $c_1 = \lambda_1 \sqrt{mr}$ for proposition 1, $c_2 = 2\gamma_1$ for proposition 2, and $L_1 = \|\boldsymbol{Y}\boldsymbol{Y}^\top\|_2$ for lemma 8. To find the values l_1 and l_2 , we have the following propositions.

Proposition 3. For
$$l_1 = q_1 = \langle \partial f_1(\boldsymbol{X}) + \nabla_{\boldsymbol{X}} H(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{G}_u(\boldsymbol{X}) \rangle$$
, we have
$$l_1 = |q_1| \leq (\lambda_1 \sqrt{mr} + \|\boldsymbol{Y}\boldsymbol{Y}^\top\|_2)^2.$$

Proof.
$$l_{1} = |q_{1}| = \left| \langle \partial f_{1}(\boldsymbol{X}) + \nabla_{\boldsymbol{X}} H(\boldsymbol{X}, \boldsymbol{Y}), \ \boldsymbol{G}_{u}(\boldsymbol{X}) \rangle \right|$$

$$l_{1} \leq \left| \langle \partial f_{1}(\boldsymbol{X}), \boldsymbol{G}_{u}(\boldsymbol{X}) \rangle \right| + \left| \langle \nabla_{\boldsymbol{X}} H(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{G}_{u}(\boldsymbol{X}) \rangle \right|$$

$$\leq \left| (c_{1} + L_{1}) \| \boldsymbol{G}_{u}(\boldsymbol{X}) \|_{2}$$

$$\leq \left| (c_{1} + L_{1}) \left(\frac{1}{\nu} \| \boldsymbol{X} - \operatorname{prox}_{\lambda_{1} \| \cdot \|_{1}}^{\nu}(\boldsymbol{X}) \right) \|_{2} + L_{1} \right)$$

$$\stackrel{lemma \ 6}{\leq} \left| (c_{1} + L_{1}) (\lambda_{1} \sqrt{mr} + L_{1}) \right|$$

$$\stackrel{proposition \ 1}{=} \left| (c_{1} + L_{1})^{2} \stackrel{lemma \ 8}{=} \left(\lambda_{1} \sqrt{mr} + \| \boldsymbol{Y} \boldsymbol{Y}^{\top} \|_{2} \right)^{2}.$$

Proposition 4. For $q_2 = \langle \partial f_2(\mathbf{X}) + \nabla_{\mathbf{X}} H(\mathbf{X}, \mathbf{Y}), \mathbf{G}_l(\mathbf{X}) \rangle$, we have

$$l_2 = |q_2| \le (\gamma_1 + ||\mathbf{Y}\mathbf{Y}^\top||_2)(2\gamma_1 + ||\mathbf{Y}\mathbf{Y}^\top||_2).$$

Proof.
$$l_2 = |q_2| = |\langle \partial f_2(\mathbf{X}), \mathbf{G}_l(\mathbf{X}) \rangle + \langle \nabla_{\mathbf{X}} H(\mathbf{X}, \mathbf{Y}), \mathbf{G}_l(\mathbf{X}) \rangle|,$$

$$l_{2} \leq |\langle \partial f_{2}(\boldsymbol{X}), \boldsymbol{G}_{l}(\boldsymbol{X}) \rangle| + |\langle \nabla_{\boldsymbol{X}} H(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{G}_{l}(\boldsymbol{X}) \rangle|$$

$$\leq |c_{lemma} | |c_{lemma} |$$

16

Theorem 2. According to 10a and 10b, α should respect the condition

$$0 \le \frac{1/\nu + \|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2}}{\lambda_{1}\sqrt{mr} + 1/\nu + 2\|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2}} \le \alpha \le \frac{\gamma_{1} + \|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2}}{\gamma_{1} + 1/\nu + 2\|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2}} \le 1;$$

with
$$\nu \geq \frac{1}{\sqrt{(\|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2} + \gamma_{1})(\|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2} + \lambda_{1}\sqrt{mr})} - \|\boldsymbol{Y}\boldsymbol{Y}^{\top}\|_{2}}$$

Proof.

$$(10a), (10b) \implies \begin{cases} \frac{k_1}{l_1 + k_1} \le \alpha \le 1; \\ \alpha \le \frac{l_2}{l_2 + k_2} \le 1. \end{cases}$$

$$\iff \begin{cases} \frac{(c_1 + L_1)(1/\nu + L_1)}{(c_1 + L_1)^2 + (c_1 + L_1)(1/\nu + L_1)} \le \alpha \le 1; \\ \alpha \le \frac{(c_2 + L_1)(\gamma_1 + L_1)}{(c_2 + L_1)(\gamma_1 + L_1)} \le 1; \end{cases}$$

$$\iff \begin{cases} \frac{1/\nu + L_1}{c_1 + 2L_1 + 1/\nu} \le \alpha \le 1; \\ \alpha \le \frac{\gamma_1 + L_1}{\gamma_1 + 2L_1 + 1/\nu} \le 1; \end{cases}$$

$$\iff \frac{1/\nu + L_1}{\lambda_1 \sqrt{mr} + 1/\nu + 2L_1} \le \frac{\gamma_1 + L_1}{\gamma_1 + 1/\nu + 2L_1}$$

Let $x = 1/\nu$, the above expression can be converted to, after some algebra, as $x^2 + 2L_1x - (\gamma_1L_1 + \gamma_1c_1 + L_1c_1)$. We have $ax^2 + bx - c \le 0$ for $a > 0, b \ge 0, c > 0$, hence

$$x = \frac{-b \pm \sqrt{b^2 + 4c}}{2} = \pm \sqrt{(L_1 + \gamma_1)(L_1 + \lambda_1 \sqrt{mr})} - L_1.$$

As $\nu \geq 0$ we take the positive root

$$0 \le \frac{1}{\nu} \le \sqrt{(L_1 + \gamma_1)(L_1 + \lambda_1 \sqrt{mr})} - L_1.$$

Hence $\nu \geq \frac{1}{\sqrt{(L_1 + \gamma_1)(L_1 + \lambda_1 \sqrt{mr})} - L_1}$, which gives the expression in the theorem.

4.3. Finding the constant β

Similar to α , we have the following for β . To find constants in (10c) and (10d) we need to find

$$k_3 = (c_3 + L_2(\boldsymbol{X}_u)) (1/\nu + L_2(\boldsymbol{X}_u)),$$

$$l_3 = |\langle \partial g_1(\boldsymbol{Y}_k) + \nabla_{\boldsymbol{Y}} H(\boldsymbol{X}_u, \boldsymbol{Y}_k), \boldsymbol{G}_u(\boldsymbol{Y}_k) \rangle|;$$

$$k_4 = (c_4 + L_2(\boldsymbol{X}_l)) (1/\nu + L_2(\boldsymbol{X}_l)),$$

$$l_4 = |\langle \partial g_2(\boldsymbol{Y}_k) + \nabla_{\boldsymbol{Y}} H(\boldsymbol{X}_l, \boldsymbol{Y}_k), \boldsymbol{G}_l(\boldsymbol{Y}_k) \rangle|;$$

where c_3, c_4 the constants for g_1 and g_2 respectively as in lemma 3; $L_2(\mathbf{X}_{\Delta})$ is the bi-smooth constant for H wrt \mathbf{Y} computed wrt $\mathbf{X}_{\Delta} \in \{\mathbf{X}_u, \mathbf{X}_l\}$. In particular, we have $c_3 = \lambda_2 \sqrt{rn}$ by proposition 1, $c_4 = 2\gamma_2$ by proposition 2, and $L_2 = \|\mathbf{X}_{\Delta}\mathbf{X}_{\Delta}^{\mathsf{T}}\|_2$ by lemma 8. In this subsection, we underlying the subscript only where its necessary, to avoid confusion.

Now we can find the values l_3 and l_4 .

Proposition 5. For $q_3 = \langle \partial g_1(\mathbf{Y}_k) + \nabla_{\mathbf{Y}} H(\mathbf{X}_u, \mathbf{Y}_k), \mathbf{G}_u(\mathbf{Y}_k) \rangle$, we have

$$l_3 = |q_3| \le (\lambda_2 \sqrt{rn} + ||\boldsymbol{X}_u^{\top} \boldsymbol{X}_u||_2)^2.$$

Proof.
$$l_3 = |q_3| = |\langle \partial g_1(\mathbf{Y}_k) + \nabla_{\mathbf{Y}} H(\mathbf{X}_u, \mathbf{Y}_k), \mathbf{G}_u(\mathbf{Y}_k) \rangle|$$

$$\begin{array}{ll} l_{3} & \leq & \left|\left\langle \partial g_{1}(\boldsymbol{Y}_{k}), \boldsymbol{G}_{u}(\boldsymbol{Y}_{k})\right\rangle\right| + \left|\left\langle \nabla_{Y}H(\boldsymbol{X}_{u}, \boldsymbol{Y}_{k}), \boldsymbol{G}_{u}(\boldsymbol{Y}_{k})\right\rangle\right| \\ \leq & \left(c_{3} + L_{2}(\boldsymbol{X}_{u})\right) \|\boldsymbol{G}_{u}(\boldsymbol{Y}_{k})\|_{2} \\ \leq & \left(c_{3} + L_{2}(\boldsymbol{X}_{u})\right) \left(\frac{1}{\nu} \left\|\boldsymbol{Y}_{k} - \operatorname{prox}_{\|\cdot\|_{1}}^{\nu}(\boldsymbol{Y}_{k})\right\|_{2} + L_{2}(\boldsymbol{X}_{u})\right) \\ \stackrel{lemma \ 6}{\leq} & \left(c_{3} + L_{2}(\boldsymbol{X}_{u})\right) (\lambda_{2}\sqrt{rn} + L_{2}(\boldsymbol{X}_{u})) \\ \stackrel{proposition \ 1}{=} & \left(c_{3} + L_{2}(\boldsymbol{X}_{u})\right)^{2} \stackrel{lemma \ 8}{\leq} (\lambda_{2}\sqrt{rn} + \|\boldsymbol{X}_{u}^{\top}\boldsymbol{X}_{u}\|_{2})^{2}. \end{array}$$

Proposition 6. For $q_4 = \langle \partial g_2(\mathbf{Y}_k) + \nabla_{\mathbf{Y}} H(\mathbf{X}_l, \mathbf{Y}_k), \mathbf{G}_l(\mathbf{Y}_k) \rangle$, we have

$$l_4 = |q_4| \le (2\gamma_2 + ||\mathbf{X}_l^{\top} \mathbf{X}_l||_2)(\gamma_2 + ||\mathbf{X}_l^{\top} \mathbf{X}_l||_2).$$

Proof.
$$l_{4} = |q_{4}| = \left| \left\langle \partial g_{2}(\mathbf{Y}_{k}) + \nabla_{\mathbf{Y}} H(\mathbf{X}_{l}, \mathbf{Y}_{k}), \mathbf{G}_{l}(\mathbf{Y}_{k}) \right\rangle \right|.$$

$$l_{4} \quad \leq \left| \left\langle \partial g_{2}(\mathbf{Y}_{k}), \mathbf{G}_{l}(\mathbf{Y}_{k}) \right\rangle \right| + \left| \left\langle \nabla_{\mathbf{Y}} H(\mathbf{X}_{l}, \mathbf{Y}_{k}), \mathbf{G}_{l}(\mathbf{Y}_{k}) \right\rangle \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) \| \mathbf{G}_{l}(\mathbf{Y}_{k}) \|_{2}$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) \left(\frac{1}{\nu} \| \mathbf{Y}_{k} - \operatorname{prox}^{\nu}_{\| \cdot \|_{*}}(\mathbf{Y}_{k}) \|_{2} + L_{2}(\mathbf{X}_{l}) \right)$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l})) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l})) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l}) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l}) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l}) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left| (c_{4} + L_{2}(\mathbf{X}_{l}) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) (\gamma_{2} + L_{2}(\mathbf{X}_{l}) \right|$$

$$\leq \left$$

Theorem 3. According to 10c and 10d, β should respect the condition

$$0 \le \frac{1/\nu + \|\boldsymbol{X}_{u}^{\top}\boldsymbol{X}_{u}\|_{2}}{\lambda_{2}\sqrt{rn} + 1/\nu + 2\|\boldsymbol{X}_{u}^{\top}\boldsymbol{X}_{u}\|_{2}} \le \beta \le \frac{\gamma_{2} + \|\boldsymbol{X}_{l}^{\top}\boldsymbol{X}_{l}\|_{2}}{\gamma_{2} + 1/\nu + 2\|\boldsymbol{X}_{l}^{\top}\boldsymbol{X}_{l}\|_{2}} \le 1;$$

with
$$\nu \ge \frac{2}{\sqrt{N^2 + 4(\lambda_2 \gamma_2 \sqrt{rn} + \gamma_2 \|\boldsymbol{X}_u^{\top} \boldsymbol{X}_u\|_2 + \|\boldsymbol{X}_l^{\top} \boldsymbol{X}_l\|_2 \lambda_2 \sqrt{rn})} - N}$$
 where $N = \|\boldsymbol{X}_l^{\top} \boldsymbol{X}_l\|_2 + \|\boldsymbol{X}_u^{\top} \boldsymbol{X}_u\|_2$.

Proof.

$$(10c), (10d) \Rightarrow \begin{cases} \frac{k_3}{l_3 + k_3} \le \beta \le 1; \\ \beta \le \frac{l_4}{l_4 + k_4} \le 1. \end{cases}$$

$$\iff \begin{cases} \frac{(c_3 + L_2(\mathbf{X}_u)) \left(\frac{1}{\nu} + L_2(\mathbf{X}_u)\right)}{(c_3 + L_2(\mathbf{X}_u))^2 + (c_3 + L_2(\mathbf{X}_u)) \left(\frac{1}{\nu} + L_2(\mathbf{X}_u)\right)} \le \beta \le 1; \\ \beta \le \frac{(c_4 + L_2(\mathbf{X}_l)) (\gamma_2 + L_2(\mathbf{X}_l))}{(c_4 + L_2(\mathbf{X}_l)) (\gamma_2 + L_2(\mathbf{X}_l)) + (c_4 + L_2(\mathbf{X}_l)) \left(\frac{1}{\nu} + L_2(\mathbf{X}_l)\right)} \le 1; \\ \iff \begin{cases} \frac{1/\nu + L_2(\mathbf{X}_u)}{c_3 + 2L_2(\mathbf{X}_u) + 1/\nu} \le \beta \le 1; \\ \beta \le \frac{\gamma_2 + L_2(\mathbf{X}_l)}{1 + 2L_2(\mathbf{X}_l)} \le 1. \end{cases}$$

$$\iff \frac{1/\nu + L_2(\boldsymbol{X}_u)}{\lambda_2 \sqrt{rn} + 2L_2(\boldsymbol{X}_u) + 1/\nu} \le \frac{\gamma_2 + L_2(\boldsymbol{X}_l)}{\gamma_2 + 1/\nu + 2L_2(\boldsymbol{X}_l)}$$

Let $x = 1/\nu$, the above expression can be converted to, after some algebra, as

$$x^{2} + (L_{2}(\boldsymbol{X}_{l}) + L_{2}(\boldsymbol{X}_{u}))x - (\lambda_{2}\gamma_{2}\sqrt{rn} + \gamma_{2}L_{2}(\boldsymbol{X}_{u}) + L_{2}(\boldsymbol{X}_{l})\lambda_{2}\sqrt{rn}) \leq 0.$$

We have $ax^2+bx-c \le 0$ for $a > 0, b \ge 0, c > 0$, so $x = (-b\pm\sqrt{b^2+4c})/2$. Let $D = b^2+4c = (L_2(\boldsymbol{X}_l)+L_2(\boldsymbol{X}_u))^2+4(\lambda_2\gamma_2\sqrt{rn}+\gamma_2L_2(\boldsymbol{X}_u)+L_2(\boldsymbol{X}_l)\lambda_2\sqrt{rn})$, we have:

$$x = \frac{-(L_2(\boldsymbol{X}_l) + L_2(\boldsymbol{X}_u)) \pm \sqrt{D}}{2} \stackrel{\nu \ge 0}{\Longrightarrow} 0 \le \frac{1}{\nu} \le \frac{\sqrt{D} - (L_2(\boldsymbol{X}_l) + L_2(\boldsymbol{X}_u))}{2}$$

So
$$\nu \ge \frac{2}{\sqrt{N^2 + 4(\lambda_2 \gamma_2 \sqrt{rn} + \gamma_2 \|\boldsymbol{X}_u^\top \boldsymbol{X}_u\|_2 + \|\boldsymbol{X}_l^\top \boldsymbol{X}_l\|_2 \lambda_2 \sqrt{rn})} - N}$$
.

Remark. (Numerical stability) The value ν has the form

$$\nu = \frac{1}{\sqrt{(a+c)(a+b)} - a},$$

which may lead to catastrophic cancellation in numerical analysis. To remove catastrophic cancellation, we implement ν as

$$\nu = \frac{\sqrt{(a+c)(a+b)} + a}{ab + ac + bc}.$$

5. Numerical Experiments

We evaluate the performance of our algorithm on synthetic and real datasets, comparing the accuracy, fidelity, and efficiency of Binno with respect to other SLRF algorithms. All the experiments were conducted in MATLAB 2024b and executed on a machine with an i7 octa-core processor and 16GB of RAM¹. Below, we detail the algorithms chosen for comparison, the datasets, the metrics, and the results.

We compare Binno against three different methods:

¹The code is available at https://github.com/flaespo/Binno.git.

- Nonnegative Matrix Factorization (NMF) with sparse matrix (NMFLS)[17]
- Non-smooth/Adaptive Augmented Lagrangian Algorithm (NSA) [18] in two versions.

NMFLS employs a standard NMF routine with the classical Lee-Seung multiplicative updates for the Frobenius norm [19]. While, NSA implements the Alternating Direction Method of Multipliers (ADMM) [12, 20] scheme for the NP-hard robust principal component analysis (RPCA) problem obtained by solving a convex optimization problem, namely the robust principal component pursuit (RPCP). The scheme is based on partial SVD for the low-rank update and entrywise soft-thresholding for the sparse update, with identical stopping rules. Compared to the first version, the second implementation is a lighter refactor that preserves the core NSA/ADMM iteration. It streamlines the parameterization (consolidating the threshold expression), initializes the partial SVD with a smaller warm start to reduce early computational overhead, and replaces the optional post hoc denoising with more fine-grained per-iteration diagnostics [21, 22]. For the comparison algorithms, we use the codes from the LRSLibrary [23]².

5.1. Dataset

The synthetic dataset is generated as a rectangular data matrix $M \in \mathbb{R}^{100 \times 80}$ designed to exhibit a low-rank, sparse structure perturbed by mild noise. Specifically, two latent factors $X \in \mathbb{R}^{100 \times 5}$ and $Y \in \mathbb{R}^{5 \times 80}$ are sampled so that approximately 30% of their entries are nonzero, with nonzero values drawn from a standard normal distribution. Their product $M_{\star} = XY$ serves as the clean signal and has nominal rank r = 5. To model measurement imperfections, we add small, entrywise independent Gaussian perturbations with standard deviation 0.01, yielding the observed matrix $M = M_{\star} + N$. This construction provides a controlled testbed in which the ground-truth low-rank structure and sparsity pattern are known while observations remain realistically noisy.

The real dataset is a traffic video database, consisting of 254 video sequences of highway traffic in Seattle, collected from a single stationary traffic camera over two days [24, 25]. The database contains a variety of traffic

²a MATLAB suite of low-rank and sparse decomposition methods https://github.com/andrewssobral/lrslibrary/tree/master/algorithms

patterns and weather conditions. Each video was recorded in color with a resolution of 320×240 pixels with between 42 to 52 frames at 10 fps. Each sequence (clip) was converted to grayscale, resized to 80×60 pixels, and then clipped to a 48×48 window over the area with the most total motion. The fixed viewpoint makes the sequences well suited to low-rank/sparse modeling of background-foreground dynamics and related video decomposition tasks.³

5.2. Metrics

We compare Binno with respect to NMFLS, NSA-v1, and NSA-v2 in terms of Peak signal-to-noise ratio (PSNR), reconstruction error, and computational time. These metrics jointly assess fidelity, structure-preserving accuracy, and efficiency. Reconstruction error as Frobenius-norm relative error, computed with

$$Err = \|\boldsymbol{M} - \boldsymbol{L}\|_F / \|\boldsymbol{M}\|_F,$$

where M is the observed matrix and L its estimate low-rank. This quantity, standard in low-rank modeling and RPCA, serves as a concise proxy for overall reconstruction quality [26]. PSNR quantifies the fidelity of a reconstruction by comparing the maximum representable signal level to the average power of the reconstruction error, and is reported on a logarithmic (decibel) scale to accommodate wide dynamic ranges. It is widely used for quantitative comparisons in image/video reconstruction and compression [27]. Operationally, PSNR is computed from the mean squared error (MSE) between the reference and the reconstructed data, smaller MSE yields larger PSNR, according to

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right).$$

where MAX denotes the peak representable value.

5.3. Results

In the following, we report results on synthetic and real datasets and observe consistent qualitative trends across the two settings. Fig.2 summarizes the synthetic setup and its ground truth: sparse factors \boldsymbol{X} and \boldsymbol{Y} with

³This dataset can be found in https://github.com/andrewssobral/lrslibrary/tree/master/dataset/trafficdb.

prescribed sparsity levels and their product M = XY, perturbed by small additive noise and the convergence of the objective bi-level functions ψ_1 and ψ_2 with respect to iterates for this case. This construction provides a controlled testbed in which the target low-rank sparse structure is known and can be visually inspected alongside quantitative criteria in Table 1.

Table 1: Evalueted metrics for synthetic dataset.

Method	Time	Reconstruction error	
Binno	0.093	0.0135	
NMFLS	0.102	1.0412	
NSA-v1	0.169	0.3259	
NSA-v2	0.065	0.3259	

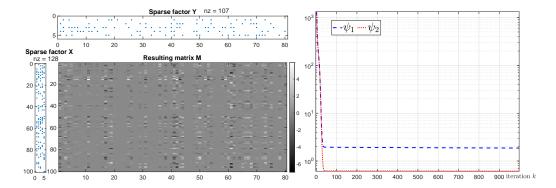


Figure 2: Synthetic experiment: (left) sparse factors and resulting matrix; (right) Objective bi-level functions convergence with respect to iterate.

Fig.3 illustrates the convergence behavior of the two objective functions ψ_1 and ψ_2 of our bi-level problem (1) across iterations for the real dataset. Also in this case, the plot shows a stable decrease consistent with the descent safeguards built into the algorithmic design (proximal-gradient blocks and calibrated averaging).

Fig.4 reports a representative clip from the dataset: we show the original frames and their noisy observations, together with the recovered low-rank component \boldsymbol{L} and sparse component \boldsymbol{S} for each method under comparison. Qualitatively, all methods produce visually comparable decompositions. To assess robustness across diverse dynamics, examples for six distinct clips are provided in Fig.5, which displays the corresponding low-rank reconstructions and sparse supports for all baselines and for Binno.

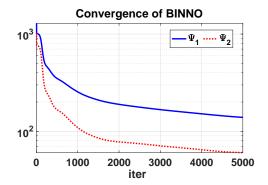


Figure 3: Objective bi-level function convergence wrt iterates for the real dataset.

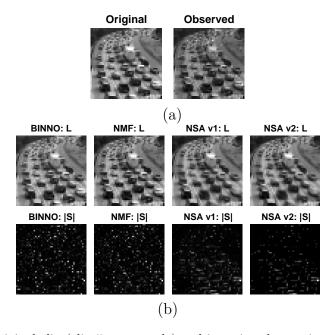


Figure 4: (a) Original clip (clip 5 as example) and its noisy observation; (b) Low-rank \boldsymbol{L} and sparse \boldsymbol{S} matrices into which the observed matrix is decomposed for all the tested algorithms.

Finally, table 2 summarizes the quantitative results across clips. While Binno incurs slightly higher time on average, it achieves the lowest reconstruction error and the highest PSNR. We report PSNR and relative error as mean±standard deviation over clips; Binno attains the best scores on all evaluated sequences.

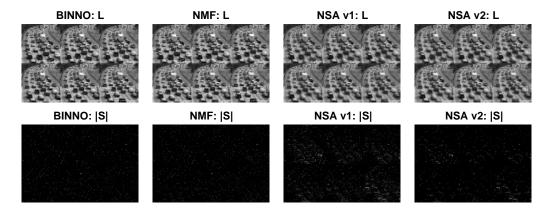


Figure 5: Low-rank L and sparse S matrices into which the observed matrix is decomposed for all the tested algorithms, for 6 different clips.

Table 2: Mean±std of the evalueted metrics computed over all clips.

Method	Time	Reconstruction error	PSNR
Binno	4.092 ± 0.3615	0.0663 ± 0.0012	35.16 ± 1.36
NMFLS	3.0769 ± 1.4457	0.0680 ± 0.0025	34.27 ± 1.34
NSA-v1	0.1754 ± 0.05	0.0982 ± 0.0094	28.43 ± 1.53
NSA-v2	0.1903 ± 0.05	0.0982 ± 0.0094	28.43 ± 1.53

6. Conclusion

In this paper, we propose a new approach to solving non-convex and non-smooth bi-level optimization problems. We introduce a novel algorithm, called Binno, which is grounded in solid theoretical considerations based on the use of proximal point methods, descent conditions, and variational properties of the involved functions. This framework allows Binno to preserve the descent property of the overall solution of the problem.

We also present a practical application of our theoretical method to the sparse low-rank approximation problem, which frequently arises in real-world scenarios where one seeks to extract meaningful information from large data matrices while maintaining a sparse representation.

Experiments on both synthetic and real datasets demonstrate the effectiveness of Binno compared to several state-of-the-art algorithms in this field, showing the power of Binno outperforms traditional methods.

Acknowledgment

F.E., L.S. are members of the Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica (GNCS-INdAM).

Funding

F.E., and L.S. are partially supported by "INdAM - GNCS Project", CUP: E53C24001950001.

F.E. are supported by Piano Nazionale di Ripresa e Resilienza (PNRR), Missione 4 "Istruzione e Ricerca"-Componente C2 Investimento 1.1, "Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale", Progetto PRIN-2022 PNRR, P2022BLN38, Computational approaches for the integration of multi-omics data. CUP: H53D23008870001.

Authors contribution

All authors contributed equally to this work.

Conflict of interest

The authors have no relevant financial interest to disclose.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process.

During the preparation of this work, the authors used GPT-4.5 in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content.

References

- [1] S. Dempe, A. Zemkoho (Eds.), Bilevel optimization, 2020th Edition, Springer optimization and its applications, Springer Nature, Cham, Switzerland, 2020.
- [2] N. Del Buono, F. Esposito, L. Selicato, R. Zdunek, Bi-level algorithm for optimizing hyperparameters in penalized nonnegative matrix factorization, Applied Mathematics and Computation 457 (2023) 128184. doi:https://doi.org/10.1016/j.amc.2023.128184.

- [3] N. Del Buono, F. Esposito, L. Selicato, R. Zdunek, Penalty hyperparameter optimization with diversity measure for nonnegative low-rank approximation, Applied Numerical Mathematics 208 (2025) 189–204, special Volume on Numerical Analysis and Scientific Computation with Applications. doi:https://doi.org/10.1016/j.apnum.2024.10.002.
- [4] P. L. Combettes, J.-C. Pesquet, Proximal Splitting Methods in Signal Processing, Springer New York, New York, NY, 2011, pp. 185–212. doi: 10.1007/978-1-4419-9569-8_10.
- [5] N. Parikh, S. Boyd, et al., Proximal algorithms, Foundations and Trends in Optimization 1 (3) (2014) 127–239. doi:10.1561/2400000003.
- [6] S. Sabach, S. Shtern, A first order method for solving convex bilevel optimization problems, SIAM Journal on Optimization 27 (2) (2017) 640–660. doi:10.1137/16M105592X.
- [7] J. Cao, R. Jiang, E. Y. Hamedani, A. Mokhtari, An accelerated gradient method for convex smooth simple bilevel optimization, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.
- [8] K.-H. Giang-Tran, N. Ho-Nguyen, D. Lee, A projection-free method for solving convex bilevel optimization problems, Mathematical Programming 213 (1-2) (2024) 907–940. doi:10.1007/s10107-024-02157-1.
- [9] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1-2) (2013) 459–494. doi:10.1007/s10107-013-0701-9.
- [10] B. Martinet, Brève communication. régularisation d'inéquations variationnelles par approximations successives, Revue française d'informatique et de recherche opérationnelle. Série rouge 4 (R3) (1970) 154–158.
- [11] A. Beck, First-Order Methods in Optimization, Society for Industrial and Applied Mathematics, 2017. doi:10.1137/1.9781611974997.

- [12] H. H. Bauschke, P. L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer International Publishing, 2017. doi:10.1007/978-3-319-48311-5.
- [13] Y. Nesterov, Lectures on Convex Optimization, Springer International Publishing, 2018. doi:10.1007/978-3-319-91578-4.
- [14] P. Sprechmann, A. M. Bronstein, G. Sapiro, Learning Efficient Sparse and Low Rank Models, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (9) (2015) 1821–1833. doi:10.1109/TPAMI. 2015.2392779.
- [15] G. Watson, Characterization of the subdifferential of some matrix norms, Linear Algebra and its Applications 170 (1992) 33–45. doi: https://doi.org/10.1016/0024-3795(92)90407-2.
- [16] J.-F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982. doi:10.1137/080738970.
- [17] Y. Ji, J. Eisenstein, Discriminative improvements to distributional sentence similarity, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.
- [18] N. S. Aybat, D. Goldfarb, G. Iyengar, Fast first-order methods for stable principal component pursuit (2011). doi:10.48550/ARXIV.1105.2126.
- [19] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788-791. doi:10.1038/ 44565.
- [20] Z. Lin, H. Li, C. Fang, Alternating Direction Method of Multipliers for Machine Learning, Springer Nature Singapore, 2022. doi:10.1007/ 978-981-16-9840-8.
- [21] N. S. Aybat, D. Goldfarb, S. Ma, Efficient algorithms for robust and stable principal component pursuit problems, Computational Optimization and Applications 58 (1) (2014) 1–29. doi:https://doi.org/10.1007/s10589-013-9613-0.

- [22] N. S. Aybat, G. Iyengar, An alternating direction method with increasing penalty for stable principal component pursuit, Computational Optimization and Applications 61 (3) (2015) 635–668. doi: 10.1007/s10589-015-9736-6.
- [23] A. Sobral, T. Bouwmans, E.-h. Zahzah, LRSLibrary: Low-Rank and Sparse tools for Background Modeling and Subtraction in Videos, in: Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing, CRC Press, Taylor and Francis Group., 2015.
- [24] A. Chan, N. Vasconcelos, Probabilistic kernels for the classification of auto-regressive visual processes, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, IEEE, pp. 846–851. doi:10.1109/cvpr.2005.279.
- [25] A. B. Chan, N. Vasconcelos, Classification and retrieval of traffic video using auto-regressive stochastic processes, in: IEEE Proceedings. Intelligent Vehicles Symposium, 2005., IEEE, 2005, pp. 771–776. doi: 10.1109/IVS.2005.1505198.
- [26] N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Review 53 (2) (2011) 217–288. doi:10.1137/090771806.
- [27] Z. Wang, A. C. Bovik, Modern Image Quality Assessment, Morgan & Claypool, 2006. doi:https://doi.org/10.1007/978-3-031-02238-8.