SUPPLEMENTARY MATERIALS: Supplementary Materials for "Structured Gradient Descent for Fast Robust Low-Rank Hankel Matrix Completion"*

HanQin Cai , Jian-Feng Cai , and Juntao You

I. The role of reweighting operator. In the paper, we introduce a reweighting operator, namely $\mathcal{D}$, for formulating the loss function in vector form:

$$\frac{1}{2p}\|\Pi_\Omega \mathcal{D}\boldsymbol{y} - \Pi_\Omega \mathcal{D}(\boldsymbol{x}+\boldsymbol{w})\|_2^2. \tag{S0}$$

The weighting step may seem to be unnecessary at the first glance, yet it plays an important role for the algorithm design for the following three reasons:

- **Motivation.** Fundamentally, we aim to recover the low-rank Hankel matrix $\mathcal{H}\boldsymbol{x}^\natural$ from

$$\mathcal{H}\Pi_\Omega \boldsymbol{y} = \mathcal{H}\Pi_\Omega \boldsymbol{x}^\natural + \mathcal{H}\Pi_\Omega \boldsymbol{w}^\natural.$$

It is natural that we minimize the loss function w.r.t. the robust Hankel matrix recovery problem:

$$\frac{1}{2p}\|\mathcal{H}\Pi_\Omega \boldsymbol{y} - \mathcal{H}\Pi_\Omega(\boldsymbol{x}+\boldsymbol{w})\|_\mathrm{F}^2.$$

In fact, we have

$$\begin{aligned}\frac{1}{2p}\|\mathcal{H}\Pi_\Omega \boldsymbol{y} - \mathcal{H}\Pi_\Omega(\boldsymbol{x}+\boldsymbol{w})\|_\mathrm{F}^2 &= \frac{1}{2p}\sum_{a\in\Omega}\sum_{i+j=a}\left([\mathcal{H}\boldsymbol{y}]_{i,j} - [\mathcal{H}(\boldsymbol{x}+\boldsymbol{w})]_{i,j}\right)^2 \\ &= \frac{1}{2p}\sum_{a\in\Omega}\varsigma_i\left(y_a - x_a - w_a\right)^2 \\ &= \frac{1}{2p}\|\Pi_\Omega \mathcal{D}\boldsymbol{y} - \Pi_\Omega \mathcal{D}(\boldsymbol{x}+\boldsymbol{w})\|_2^2\end{aligned}$$

where ς_i is the number of entries on the i-th antidiagonal. That said, given the relationship between the Hankel matrix loss function and the corresponding vector loss function, the reweighting $\mathcal{D}$ is necessary for the equivalence of the objective.

- **Technical necessity.** A fundamental tool for algorithm design and analysis in matrix completion is the restricted isometry property (RIP). When extending to the Hankel matrix setting, it must involve the reweighting operator $\mathcal{D}$ to keep the RIPs hold (see Lemmas 10 and 12). Using Lemma 12 as an example here, it states

$$\left\|\mathcal{H}\mathcal{D}^{-1}\left(p^{-1}\Pi_\Omega - \mathcal{I}\right)\mathcal{D}^{-1}\mathcal{H}^*\boldsymbol{A}\right\|_\mathrm{F}^2 =: \left\|\mathcal{G}\left(p^{-1}\Pi_\Omega - \mathcal{I}\right)\mathcal{G}^*\boldsymbol{A}\right\|_\mathrm{F}^2 \le \varepsilon_0\left\|\boldsymbol{A}\right\|_\mathrm{F}^2$$

---

*Supplementary material for SISC MS#M149100.
https://doi.org/10.1137/22M1491009

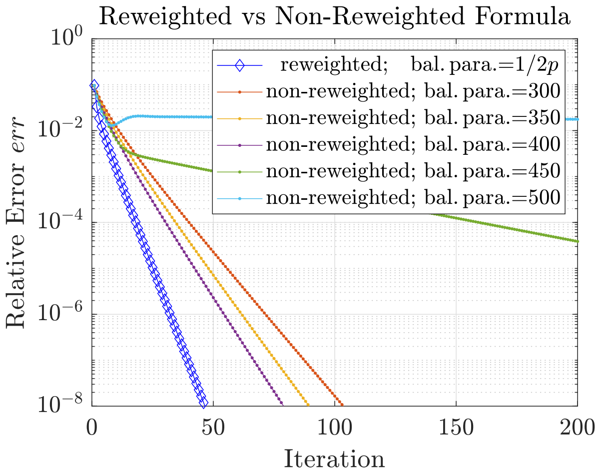

Figure S1. *Convergence behaviour: reweighted vs. non-weighted.*

under certain conditions. The RIPs provide not only analytic tools but also a guide for parameter choice—the balancing parameter for this loss term (S0) is set to be $1/2p$ for this very reason ($1/2$ is for the square). Without the reweighting operator, we have no theoretical guide on how to set this balancing parameter. Another helpful fact for our analysis is that $\mathcal{G}\mathcal{G}^* = \mathcal{H}\mathcal{D}^{-2}\mathcal{H}^*$ is a projection operator while $\mathcal{H}\mathcal{H}^*$ is not.

- **Empirical performance.** To further demonstrate the empirical advantages of the reweighted formula, in Figure S1, we compare the proposed HSGD (reweighted and balancing parameter $= 1/2p$) against the non-reweight version with varying balancing parameters for the loss term. One can see that the non-reweight version does not converge as fast as the proposed HSGD, even with the hand-tuned parameters.

II. The choice of parameters. In our numerical simulations for HSGD, parameters including the incoherence parameter μ, the rank r and the outlier sparsity α are required as inputs. The incoherence parameter μ and sparsity parameter α do not have to be exact according to their definitions, and a reasonably safe upper bound is sufficient. In fact, throughout our numerical simulations, the parameter μ is estimated by one step of Cadzow's algorithm [3]. This follows the same setup in our previous work [4]. In the following paragraphs, let's address α and r in detail.

- **Sparsity parameter** α: Recall that any α_1-sparse matrix is also α_2-sparse provided $\alpha_1 \leq \alpha_2$. That said, HSGD can tolerate a mild overestimation on α. In Figure S2, we compare HSGD's performance with overestimated α. The figure illustrates that the more overestimation in α, the slower HSGD converges. Nevertheless, HSGD is still linearly convergent to the ground truth with mildly overestimated α.
- **Rank parameter** r: Similar to many non-convex algorithms for low-rank related problems [4-11,19,28,39], we assume the exact rank r is given to HSGD in the analysis and numerical simulations. Nevertheless, HSGD can incorporate the stage-wise framework [18,29,41] then no prior knowledge of r is required. The strategy is to start the algorithm from a rank-1 approximation, then gradually increase the rank stage-by-stage until it

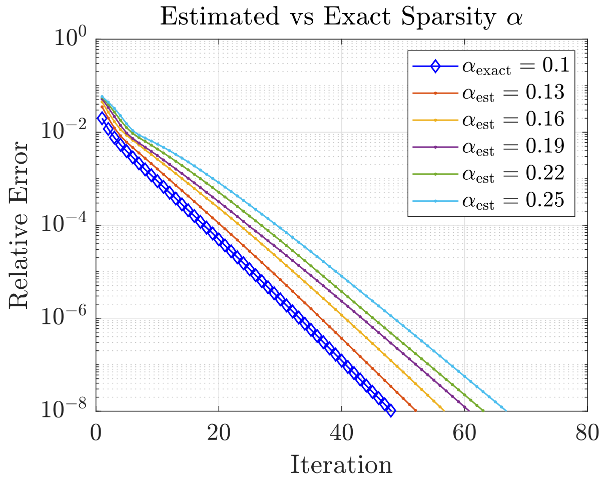

Figure S2. *Convergence behaviour: estimated α vs. exact α.*

finds a good rank estimation. Such a strategy is widely adaptive to the non-convex algorithms that require the knowledge of true rank. Thus, in this paper, we focus on developing the algorithm with a given rank.

For the reader's convenience, we briefly describe how HSGD is combined into the stage-wise framework:

Initialize $\boldsymbol{z}_1^0$ via spectral method with $r = 1$;
For stage $t = 1, 2, 3, \cdots$, do:
  set rank $r = t$;
  run K_t steps of HSGD;
  update $\boldsymbol{z}_{t+1}^0 = \boldsymbol{z}_t^{K_t}$;
end for if converged.

$\boldsymbol{z}_t^k$ is the output of HSGD at the k-th iteration of the t-th stage.

Note that the stage-wise approach naturally results in more iterations, i.e., higher computational costs. In Figure S3, we compare the convergence behaviors between vanilla HSGD and stage-wise HSGD. We observe that stage-wise HSGD struggles until it finds the true rank.

We emphasize that requiring the knowledge of true rank is a common problem for many non-convex approaches and the stage-wise framework is an easy solution to overcome this problem. Therefore, in this work, we focus on developing the algorithm with a given rank.

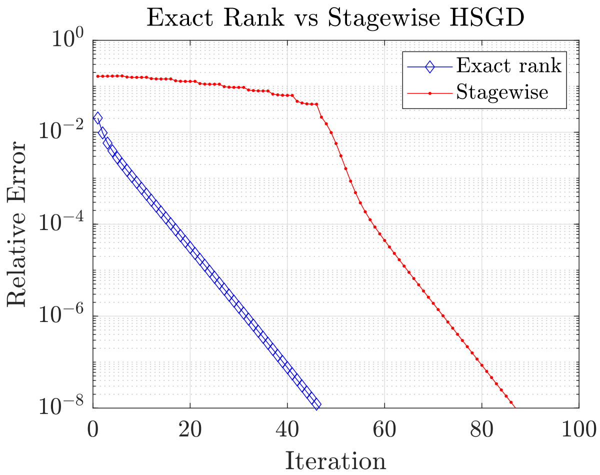

Figure S3. *Convergence behaviour: HSGD with exact r vs. stage-wise HSGD.*