EFFICIENT ALTERNATING MINIMIZATION WITH APPLICATIONS TO WEIGHTED LOW RANK APPROXIMATION

Zhao Song*, Mingquan Ye†, Junze Yin‡, Lichen Zhang§

∗ : Simons Institute for the Theory of Computing, † : University of Illinois at Chicago, ‡ : Rice University, § : MIT CSAIL

Introduction

The Problem That We Study. Let $M \in \mathbb{R}^{n\times n}$ be a noisy, higher-rank matrix with $M = M^* + N$ where M^* is the rank-k ground truth and N is a higher-rank noise matrix. Let $W \in \mathbb{R}^{n\times n}_{\geq 0}$ be a non-negative weight matrix. The goal is to find a rank-k matrix $\widetilde{M} \in \mathbb{R}^{n\times n}$ such that

$$\|\widetilde{M} - M^*\| \leq \delta \cdot \|W \circ N\| + \epsilon$$

by observing the matrix $W \circ M$.

Assumptions on the Matrix M. Given a noisy, possibly higher-rank observation $M \in \mathbb{R}^{n\times n}$ such that $M = M^* + N$, where M^* is the rank-k ground truth we want to recover and N is the noise matrix. We assume:

1. M^* is μ-incoherent: Let $M^* = U\Sigma V^\top$ be its SVD, we assume

$$\max\{\|U_{i,:}\|_2^2, \|V_{i,:}\|_2^2\}_{i=1}^n \leq \frac{\mu k}{n}.$$

   We use τ to denote the condition number of M^*: $\tau = \sigma_{\max}(M^*)/\sigma_{\min}(M^*)$.

2. Weight W has a γ-spectral gap to all-1's matrix:

$$\|W - \mathbf{1}_n\mathbf{1}_n^\top\| \leq \gamma n.$$

3. Weight W is (α, β)-bounded: Let $M^* = U\Sigma V^\top$ be its SVD, we assume for any $i \in [n]$ and $0 < \alpha \leq 1 \leq \beta$,

$$\alpha I \preceq U^\top D_{W_i} U \preceq \beta I,$$
$$\alpha I \preceq V^\top D_{W_i} V \preceq \beta I.$$

Main Result. Our main theorem is stated as follows:

Theorem 1. *Given a noisy, possibly higher-rank observation $M \in \mathbb{R}^{n\times n}$ where M^* is the rank-k ground truth and N is the noise matrix that satisfies the above assumptions, we can get an algorithm that uses random initialization, runs in $O(\log(1/\epsilon))$ iterations and generates an $n \times n$ matrix $\widetilde{M}$ such that*

$$\|\widetilde{M} - M^*\| \leq O(\alpha^{-1}k\tau)\|W \circ N\| + \epsilon,$$

The total running time is

$$\widetilde{O}((\|W\|_0 \cdot k + nk^3)\log(1/\epsilon)).$$

Algorithm From [Li et al., 2016]

Alternating minimization.

1. $U_0, V_0 \leftarrow \mathrm{SVD}(W \circ (M + N), k)$
2. For all $t \in \{1, \ldots, T\}$,
   (a) $U_t \leftarrow \arg\min_{U\in\mathbb{R}^{n\times k}} \|W \circ (M + N) - W \circ (UV_{t-1}^\top)\|_F^2$
   (b) $V_t \leftarrow \arg\min_{V\in\mathbb{R}^{n\times k}} \|W \circ (M + N) - W \circ (U_tV^\top)\|_F^2$
3. Return $U_TV_T^\top$

Strength and Weakness

- It is commonly used in practice, but
- It requires $O((\|W\|_0 \cdot k^2 + nk^3)\log(1/\epsilon))$ where $\|W\|_0$ denote the number of nonzero entries in W.

References

[1] Yuanzhi Li, Yingyu Liang, and Andrej Risteski. "Recovery guarantee of weighted low-rank approximation via alternating minimization". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2358–2367.

[2] Jelani Nelson and Huy L Nguyên. "OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings". In: *2013 ieee 54th annual symposium on foundations of computer science*. IEEE. 2013, pp. 117–126.

Our Results That Improve [Li et al., 2016]

Sketching as a dimension reduction tool. We can cast the weighted low rank approximation

$$\min_{Y\in\mathbb{R}^{n\times k}} \|W \circ (M - XY^\top)\|_F^2.$$

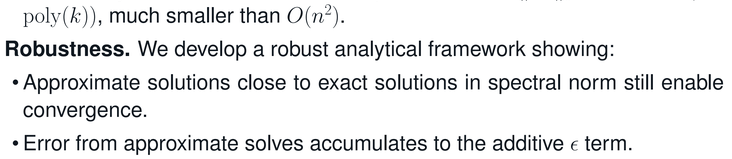
into n linear regressions

$$\min_{y\in\mathbb{R}^k} \|D_{\sqrt{W_i}}M_{:,i} - D_{\sqrt{W_i}}Xy\|_2^2,$$

where $D_{\sqrt{W_i}}$ denote the $n \times n$ diagonal matrix that puts $\sqrt{W_i}$ on the diagonal, where W_i is the i-th column of W.

To solve these regressions faster, one can pick a random sketching matrix $S \in \mathbb{R}^{s\times n}$ where $s = O(\epsilon_0^{-2}k)$ and instead solve

$$\min_{y\in\mathbb{R}^k} \|SD_{\sqrt{W_i}}M_{:,i} - SD_{\sqrt{W_i}}Xy\|_2^2.$$



By picking a sparse sketching matrix S [2], the above regression can be solved in

$$\widetilde{O}(\epsilon_0^{-1}\mathrm{nnz}(X) + \epsilon_0^{-2}k^3)$$

time with high probability, and the output solution y has cost at most $(1 + \epsilon_0) \cdot \mathrm{OPT}$ where OPT is the optimal regression cost.

However, this approach provides only forward error guarantees, which are insufficient for analyzing convergence. Converting forward error to backward error requires extremely small error parameters, making polynomial dependencies on ϵ_0^{-1} impractical.

Our approach of using the sketching matrix. We design a high-precision regression solver with only logarithmic dependency on the error parameter $\log(1/\epsilon_0)$.

- Uses a dense sketching matrix S with $O(k)$ rows that preserves norms.
- Applies S to create a smaller matrix and computes its QR decomposition.
- Uses the right QR factor as a preconditioner.
- Employs gradient descent to iteratively optimize toward the solution.

This approach achieves

$$O((nk + k^3)\log(1/\epsilon_0))$$

runtime for computing approximate solutions.

The sparsity of the weight matrix. Further optimization leverages the sparsity of weight matrix W:

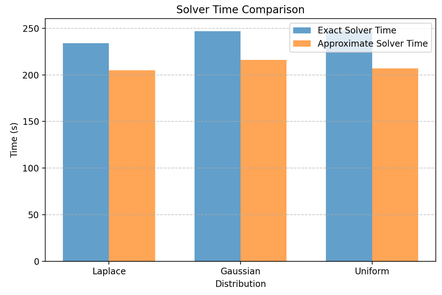
- For sparse W, many regression problems effectively become much smaller.
- Total runtime improves to $\widetilde{O}((\|W\|_0k + nk^3)\log(1/\epsilon_0))$.
- Particularly valuable for matrix completion where $\|W\|_0$ is often $\widetilde{O}(n \cdot \mathrm{poly}(k))$, much smaller than $O(n^2)$.

Robustness. We develop a robust analytical framework showing:

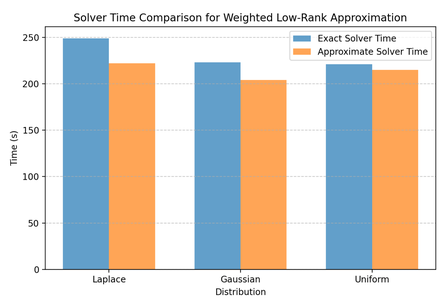
- Approximate solutions close to exact solutions in spectral norm still enable convergence.
- Error from approximate solves accumulates to the additive ϵ term.
- Only polylogarithmically more iterations are needed for good guarantees.

Experiments

Experiment 1 – low rank matrix completion For each row of the weight matrix $W \in \mathbb{R}^{n\times n}_{\geq 0}$, we randomly select 400 entries to be equal to 1 and the remaining 400 entries to be 0.

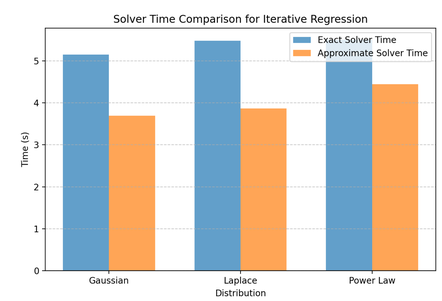


Experiment 2 – weighted low rank approximation The weight matrix W is constructed via $\mathbf{1}_n\mathbf{1}_n^\top + G$ for G being a random matrix with standard Gaussian entries.



We note that in these two settings, our algorithm achieves a speedup compared to the algorithm of [1].

Experiment 3 – iterative regression solvers We test the performance of regression solvers for $n = 10^6$, $k = 500$ with a sketch size $m = 5500$, and run our solver for 5 iterations.



Follow Us on YouTube and Bilibili

- Youtube Channel: `https://www.youtube.com/@zhaosong2031`
- Bilibili Channel: `https://space.bilibili.com/3546587376650961/video`