Additive Error Guarantees for Weighted Low Rank Approximation

Aditya Bhaskara Aravinda Kanchana Ruwanpathirana Maheshakya Wijewardena

School of Computing, University of Utah

Low Rank Approximation

- **Classic formulation:** Given matrix A, find a rank-k matrix L that minimizes $\|A - L\|_F^2 = \sum_{i,j}(A_{ij} - L_{ij})^2$. Can be solved efficiently (e.g., SVD).
- Natural variants are NP hard: e.g., different *importance* for different entries.
- **Weighted LRA:** Given matrix A and weight matrix W of the same size, find a rank-k matrix L that minimizes $Cost(L) = \sum_{i,j} W_{ij} \cdot (A_{ij} - L_{ij})^2$.
- Can also consider ℓ_p error, $\sum_{i,j} W_{ij}(A_{ij} - L_{ij})^p$.

Motivation

varying importances

3.8	4.3	9	3	5
2	8.1	7	0.3	9.1
1.1	8	7.1	4	1
7	6.2	6	2.2	8

varying noises

3.8	4.3	9+ε	3+ε	5+ε
2+ε	8.1	7+ε	0.3	9.1
1.1+ε	8	7.1	4+ε	1+ε
7+ε	6.2	6+ε	2.2	8

Prior Work

- Alternating minimization heuristic [1] – no guarantees
- Multiplicative error bounds – assumes low rank W and time complexity exponential in rank of W. [2, 3]
- Additive error bounds – simple algorithm, but requires extra 'low communication complexity' assumption on W.

Our goal: design and analyze efficient, practical algorithms for weighted and ℓ_p error low rank approximation

Results

Informal. There exist greedy iterative algorithms that achieve *additive error* guarantees, under mild assumption on *target* matrix L (informally, the opt solution).
Formally, suppose the target L satisfies (for some parameter Λ)

$$\frac{\|L\|_F^2}{\|A\|_F^2} \le \Lambda. \quad \text{(target not too different from } A \text{ in Frobenius norm)}$$

Theorem. (Weighted LRA) For any $\epsilon > 0$, there is a greedy algorithm that outputs L' of rank $O(k\Lambda/\epsilon^2)$, satisfying $Cost(L') \le Cost(L) + \epsilon\|A\|_F^2$.

Extensions to ℓ_p norm error.

- Analogous result holds under an ℓ_p error objective, with a different greedy step.
- Implies an *unconditional* algorithm for unweighted ℓ_p LRA, for $p \ge 2$.

Algorithm outline

Idea: Build approximation of columns a_j using a set of 'basis vectors' Z
Initialize $Z = \emptyset$, $x_j^{(0)} = 0$ (approximation for column a_j) for all j;
for $t = 1, 2, \dots, k'$ **do**
 (I) Solve an optimization problem to find $\mathbf{z}$ that captures sufficient mass from the "residual" $(a_j - x_j^{(t)})$;
 (II) Add $\mathbf{z}$ to Z and update approximations of columns $x_j^{(t)}$;
end
Return Z and $L' = [x_1^{(k')}, \dots, x_n^{(k')}]$;

Analysis

Let $X^{(t)}$ be the matrix approximating A at step t (columns $x_j^{(t)}$), and L be the target low rank approximation.

Basic idea. As long as $Cost(X^{(t)}) < Cost(L)$, there exists a vector $\mathbf{z}$ that reduces the cost "significantly". (Reminiscent of Set Cover.)

- "Per column" analysis. For column j, define $f_j : \mathbb{R}^n \mapsto \mathbb{R}$ as

$$f_j(v) = \sum_{r \in [d]} w_{j,r}(a_{j,r} - v_r)^2.$$

- Key Lemma: Suppose y is the current approximation for column a_j and suppose the "ideal" approximation is $z = \sum_i \alpha_i u_i$. If $f_j(z) < f_j(y)$, there exists index i such that adding u_i to y reduces f_j by $\Omega_\alpha((f_j(y) - f_j(z))^2)$.
- (One column → matrix) If $Cost(L) := \Gamma$ and $Cost(X^{(t)})$ is Δ_t, there exists $\mathbf{z}$ in the algorithm such that

$$\Delta_{t+1} \le \Delta_t - \frac{(\Delta_t - \Gamma)^2}{4\Lambda}. \quad \text{(Implies desired convergence rate)}$$

- Optimization problem: required $\mathbf{z}$ can be obtained by solving:

$$\max \sum_j \langle \nabla f_j(x_j^{(t)}), u \rangle^2 \text{ subject to } \|u\| \le 1.$$

- Reduces to finding top singular vector of appropriate matrix!

Extension to ℓ_p error

S When $p > 2$, same high-level framework applies, but:

- Requires more involved analysis to prove "progress" (uses recent works on ℓ_p regression to show smoothness of f_j).
- Optimization problem is now instance of matrix $2 \mapsto p$ norm computation – can be solved via convex relaxations when $p \ge 2$.

Experiments

The following schemes were used to derive the weight matrices for the plots.

- W_1: Each element is sampled from the interval $[0, 1]$ uniformly at random.
- W_3: Generate a random binary matrix with each entry 1 with probability 0.1 and then set the first 100 columns of first 150 rows to 1.

Error Against the Rank

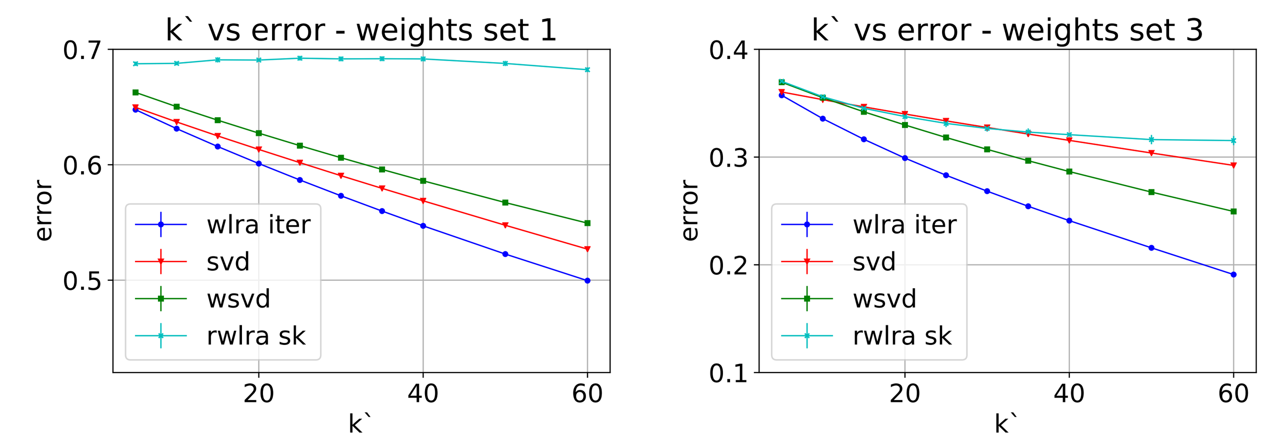

Figure: Comparison of cost of the approximation against the rank of the matrix.

Error Against the Signal-to-noise Ratio

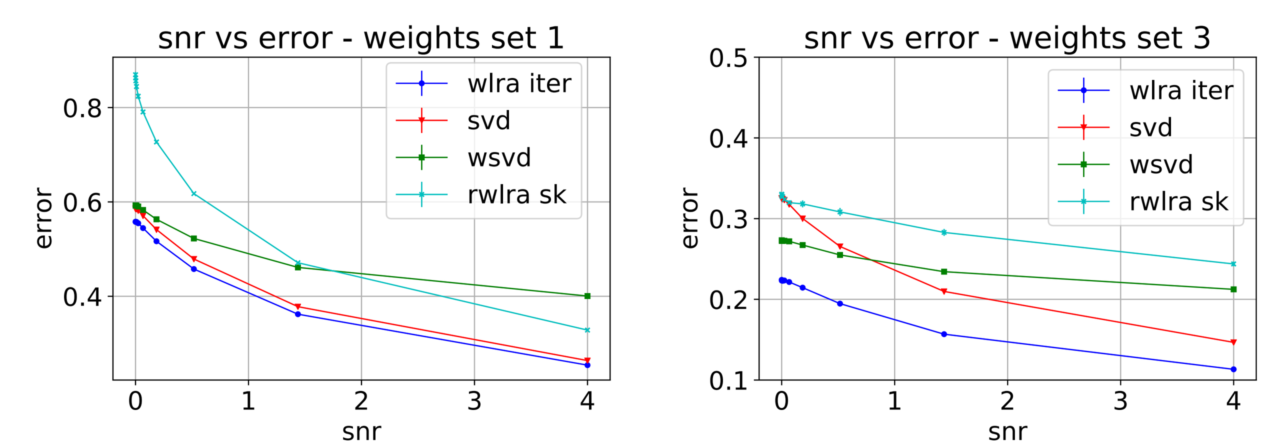

Figure: Comparison of cost of the approximation against the signal to noise ratio of the matrix.

Conclusions

- We study greedy pursuit algorithms for weighted low rank approximation, and show that they yield good bi-criteria approximations with a small *additive error*. Holds for ℓ_p error, for $p \ge 2$, under a realistic assumption on target low rank matrix.
- Proposed algorithm is easy to implement and works well in practice.

References

[1] Nathan Srebro and Tommi Jaakkola. "Weighted Low-Rank Approximations". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA: AAAI Press, 2003, pp. 720–727. ISBN: 1577351894.

[2] Frank Ban, David Woodruff, and Richard Zhang. "Regularized Weighted Low Rank Approximation". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 4059–4069.

[3] Ilya Razenshteyn, Zhao Song, and David P. Woodruff. "Weighted Low Rank Approximations with Provable Guarantees". In: *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 250–263. ISBN: 9781450341325.