Bi-criteria Algorithms for ℓ_p-Low Rank Approximation

Arvind Mahankali, Prof. David Woodruff (Advisor)

Carnegie Mellon University

Problem

We study the problem of ℓ_p-low rank approximation. That is, given $k \in \mathbb{N}, \alpha \geq 1$ and a matrix $A \in \mathbb{R}^{n \times d}$, the goal is to find a matrix B of small rank such that

$$\|A - B\|_p \leq \alpha \cdot \min_{A_k \text{ rank } k} \|A - A_k\|_p$$

We allow our solution B to have rank greater than k, and a solution to the above problem is called an *α-approximation.*

Motivation

- Reduces space needed to store a matrix
- Reduces time needed to multiply a matrix by a vector
- Applications in machine learning, computer vision, information retrieval
- ℓ_p-norm loss is more robust than the more standard ℓ_2-norm loss

Previous Results

- Column Subset Selection [5]: Gives an $O(k \log k)$-approximation, with rank $O(k \log n)$, in polynomial time.
- Guessing a Sketch [1]: Gives a $(1 + \varepsilon)$-approximation, with rank k, in $n^{\text{poly}(k/\varepsilon)}$ time.

Column Subset Selection Lower Bound

We first show that the column subset selection approach of [5] cannot do too well.

Column Subset Selection Lower Bound

Let $A \in \mathbb{R}^{n \times d}$, $k \in \mathbb{N}$, and $c > 0$. Then, an algorithm for ℓ_p-low rank approximation that returns $\hat{A} = UV$, where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{r \times d}$, and the columns of U are columns of A, cannot achieve better than an $O(k^{\frac{1}{p} - \frac{1}{2} - c})$ approximation factor, unless $r = \Omega(n)$.

This was shown in [4] for $p = 1$, and we extend it to $p \in [1, 2)$ using similar techniques.

Column Subset Selection Algorithm

We now give an algorithm for ℓ_p-low rank approximation that makes use of column subset selection.

Column Subset Selection Upper Bound

Let $A \in \mathbb{R}^{n \times d}$ and $k \in \mathbb{N}$. Then, there is an algorithm which runs in polynomial time and returns $U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{r \times d}$ such that $r = O(k\text{poly}(\log n))$ and with high probability,

$$\|UV - A\|_p \leq O(k^{\frac{1}{p} - \frac{1}{2}} \text{poly}(\log k)) \min_{A_k \text{ rank } k} \|A - A_k\|_p$$

In light of the lower bound, this is essentially optimal (up to log factors).

Column Subset Selection Algorithm - Continued

The algorithm is as follows:

- Sample $t = O(k\text{poly}(\log k))$ columns uniformly at random, which will be included in U, and discard the $\Omega(d)$ remaining columns of A which have the lowest regression cost against these t columns. Repeat this step $O(\log d)$ times and take the sample for which the $\Omega(d)$ discarded columns have the smallest regression cost.
- Then, recurse — consider A' without the $\Omega(d)$ discarded columns or the sampled columns, and repeat the above process.

Analysis of Column Subset Selection Algorithm

We give a sketch of the analysis — using the reasoning given here gives a $k^{\frac{1}{p} - \frac{1}{2}}\text{poly}(\log d)$-approximation algorithm, and a refined version of this analysis leads to the desired $k^{\frac{1}{p} - \frac{1}{2}}\text{poly}(\log k)$ approximation factor. Let B_0 be a column sample of size t, and A_i be another uniformly random column of A. Consider the new matrix $B = [B_0, A_i]$. Then, there exists a column subset of size $\frac{t}{2}$ which spans a good rank-$\frac{t}{2}$ approximation to B, with approximation factor $O(k^{\frac{1}{p} - \frac{1}{2}})$ [4]. Such a subset S can be constructed by sampling the columns of B according to the column Lewis weights [2] of V^*, where U^*V^* is the best rank-k approximation to B.

Note that V^* is difficult to directly compute efficiently, but we can still reason about this sampling procedure as follows. With constant probability, the probability on the column A_i (under the distribution given by the Lewis weights of V^*) is small, meaning it is not chosen with constant probability. In addition, with constant probability, the column subset of B obtained by sampling according to the Lewis weights of V^* achieves a small regression cost on A_i, even conditioned on A_i not being a member of the subset. Hence, by Markov's inequality, a small regression cost is achieves on $\Omega(d)$ columns of A. By repeating this sample procedure with replacement, $O(\log d)$ times, we can boost the success probability and finally union bound over the different samples.

Reduction to k-Median

$O(1)$-Approximation Algorithm via k-Median

Let $A \in \mathbb{R}^{n \times d}$ and $k \in \mathbb{N}$. Then, there is an algorithm that runs in polynomial time and returns $\hat{A}$ such that $r = (\log d)^{\text{poly}(k)}$ and

$$\|\hat{A} - A\|_1 \leq O(1) \min_{A_k \text{ rank } k} \|A_k - A\|_1$$

with constant probability.

Our algorithm is as follows. First, we obtain a $\text{poly}(k)\log(d)$-approximation B with rank k in polynomial time using an algorithm from [4]. Then, we apply an $O(1)$-approximation algorithm for k-median in the ℓ_1 distance, which was given in [3], to $A - B$, with the desired number of centers being $(\log d)^{\text{poly}(k)}$. Let M_i be the center that is closest to the i^{th} column of $A - B$ — then, we return $B + M$, where M is the matrix whose columns are the M_i.

Analysis of $O(1)$-Approximation Algorithm

A key point in the analysis is that there is an approximation for $A - B$ with at most $(\log d)^{\text{poly}(k)}$ distinct columns, and hence it suffices to solve the k-median problem with this number of centers. We can show this by multiplying the objective by a carefully chosen Lewis weight sampling matrix S with $\text{poly}(k)$ rows. The new objective becomes

$$\min_{U,V} \|SUV - S(A - B)\|_1$$

Rounding the columns of $S(A - B)$ to a $\frac{1}{\text{poly}(k)\log(d)}$-net leads to our conclusion — the net has $(\log d)^{\text{poly}(k)}$ distinct elements, up to scaling, and rounding leads to an additive error of $\frac{\|S(A-B)\|_1}{\text{poly}(k)\log(d)} = O(1) \min_{A_k \text{ rank } k} \|A - A_k\|_1$.

$O(\log d)$-Approximation Algorithm with rank $O(k)$

$O(\log d)$-Approximation Algorithm

Let $A \in \mathbb{R}^{n \times d}$ and $k \in \mathbb{N}$. Then, there is an algorithm that runs in time $2^{2^{\text{poly}(k)}}\text{poly}(nd)$ and returns $\hat{A}$ with rank $O(k)$ such that

$$\|\hat{A} - A\|_1 \leq O(\log d) \min_{A_k \text{ rank } k} \|A_k - A\|_1$$

This algorithm makes use of a rounding procedure similar to the one used in the analysis of our $O(1)$-approximation algorithm. In order to force the rank to be $O(k)$, we use polynomial optimization.

Acknowledgements

Arvind Mahankali is very grateful to his advisor, Prof. David Woodruff, for patiently introducing him to theoretical computer science research.

References

[1] Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P. Woodruff.
A PTAS for lp-low rank approximation.
In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 747–766, 2019.

[2] Michael B. Cohen and Richard Peng.
l_p row sampling by lewis weights.
CoRR, abs/1412.0588, 2014.

[3] Piotr Indyk.
Sublinear time algorithms for metric space problems.
In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 428–434, 1999.

[4] Zhao Song, David P. Woodruff, and Peilin Zhong.
Low rank approximation with entrywise l_1-norm error.
In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 688–701, 2017.

[5] Zhao Song, David P. Woodruff, and Peilin Zhong.
Towards a zero-one law for column subset selection.
In *Advances in Neural Information Processing Systems*, pages 6120–6131, 2019.