Streaming and Distributed Algorithms for Robust Column Subset Selection

Shuli Jiang [1] Dongyu Li [1] Irene Mengze Li [1] Arvind V. Mahankali [1] David P. Woodruff [1]

[1]Carnegie Mellon University

ℓ_p Column Subset Selection in the Streaming and Distributed Settings

In streaming ℓ_p column subset selection (CSS), we are given:

- A matrix $A \in \mathbb{R}^{d\times n}$, where typically $d \ll n$. The columns of A arrive one by one in a stream.
- $p \in [1, 2)$ and $k \in \mathbb{N}$

The goal is to find k columns of A which minimize the ℓ_p reconstruction error. In other words, we wish to find $S \subset [n]$ such that $|S| = k$ and $\min_{V\in\mathbb{R}^{k\times n}} \|A_S V - A\|_p$ is minimized, where A_S is the subset of columns of A whose indices are in S. Our algorithm should meet the following constraints: (1) it should require at most **one pass** over the columns of A, (2) it should use **very low space** --- we want space complexity that is close to the optimal $O(dk)$ space complexity, and (3) it should be efficient --- we want at most **polynomial running time**.

We also allow **bi-criteria guarantees**, meaning S is allowed to contain $\widetilde{O}(k)$ columns instead of only k, and we seek **approximation algorithms**, meaning that for some reasonably small $\alpha \geq 1$, if $S^*, V^* = \text{argmin}_{S,V}\|A_S V - A\|_p$, then we desire $S \subset [n]$ and $V \in \mathbb{R}^{k\times n}$ such that $\|A_S V - A\|_p \leq \alpha\|A_{S^*}V^* - A\|_p$.

Notation: For a matrix $M \in \mathbb{R}^{d\times n}$, we define its ℓ_p norm to be $\|M\|_p := (\sum_{i=1}^{d}\sum_{j=1}^{n}|M_{i,j}|^p)^{1/p}$, and its $\ell_{p,2}$ norm to be $\|M\|_{p,2} := (\sum_{j=1}^{n}\|M_{*j}\|_2^p)^{1/p}$, where M_{*j} is the j^{th} column of M.

Our streaming algorithm can also be extended to a distributed protocol in the column-partition model. Here, there are s servers, with the i^{th} server holding $A^i \in \mathbb{R}^{d\times n_i}$ (which is a subset of the columns of A), and one coordinator that can communicate with each of the servers. In this setting, we would like an ℓ_p CSS algorithm that uses **very low communication ($O(sdk)$ bits of communication is optimal)** and **very few rounds of communication** between the servers and coordinator.

Related Work

- [CGK+17, DWZ+19, SWZ19, MW20] analyze an algorithm which samples columns uniformly at random over the course of $O(\log n)$ rounds, and [MW20] showed this achieves a $\widetilde{O}(k^{1/p-1/2})$-approximation for ℓ_p low rank approximation. **A naive streaming implementation of this algorithm would require $O(\log n)$ passes, and a distributed implementation would require $O(\log n)$ rounds.**
- [SWZ17] gives a streaming algorithm (resp. $O(1)$-round distributed algorithm) for ℓ_p low rank approximation in the column-update model (resp. column-partition model), with $\widetilde{O}(dk)$ space (resp. $\widetilde{O}(sdk)$ communication), but **it is not clear how to turn these into CSS algorithms.**

Our Result

We give a streaming algorithm for ℓ_p CSS which, given a matrix $A \in \mathbb{R}^{d\times n}$ in the column-update streaming model, returns a subset $S \subset [n]$ of size $\widetilde{O}(k)$ such that S is an $\widetilde{O}(k^{1/p-1/2})$-approximate solution compared to the column subset with the best reconstruction error:

$$\min_{V\in\mathbb{R}^{|S|\times d}} \|A_S V - A\|_p \leq \widetilde{O}(k^{1/p-1/2}) \min_{S^*\subset[n],|S^*|=k,V^*\in\mathbb{R}^{k\times d}} \|A_{S^*}V^* - A\|_p$$

The space complexity is $\widetilde{O}(dk)$, and the running time is $\widetilde{O}(\text{nnz}(A)k + kd + k^3)$, where $\text{nnz}(A)$ is the number of nonzero entries of A.

We also give a distributed protocol for ℓ_p CSS in the column-partition model, which returns a subset $S \subset [n]$ of size $\widetilde{O}(k)$ achieving the same approximation guarantee as above. Our protocol requires only 1 round, uses $\widetilde{O}(sdk)$ space, and has running time $\widetilde{O}(\text{nnz}(A)k + kd + k^3)$.

Our Algorithm and Key Technical Tools

Below, we describe our algorithms/their analyses, and highlight the main technical novelties.

- In both the streaming and distributed settings, we reduce to $\ell_{p,2}$ CSS. We make use of *strong coresets* for $\ell_{p,2}$ low rank approximation. Given a matrix $B \in \mathbb{R}^{n\times d}$, strong coresets allow us to select $\widetilde{O}(k)$ columns of B which are representative of B in the following sense:

  Strong Coresets for $\ell_{p,2}$-norm Low Rank Approximation [SW18]:
  Let $B \in \mathbb{R}^{d\times n}$ be a matrix. Then for any desired $\varepsilon, \delta \in (0, 1)$, we can efficiently construct a sampling and rescaling matrix T, with $O(d \cdot \text{poly}(\log(d/\varepsilon), \log(1/\delta)))$ columns such that, with probability $1 - \delta$,

  $$\min_{\text{rank } k\ V} \|UV - BT\|_{p,2} = (1 \pm \varepsilon) \min_{\text{rank } k\ V} \|UV - B\|_{p,2}$$

  We refer to BT as a *strong coreset* of B. To our knowledge, strong coresets for $\ell_{p,2}$ low rank approximation have not been used before for ℓ_p CSS, or even ℓ_p low rank approximation.

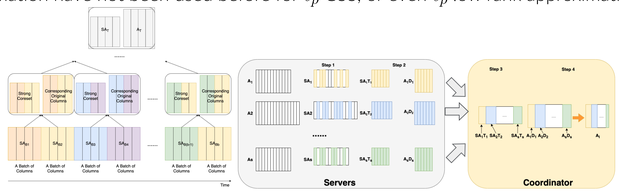


Figure 1:Left: overview of streaming algorithm. Right: overview of distributed protocol.

- For a matrix B with d rows, $\|B\|_{p,2}$ can be less than $\|B\|_p$ by a factor of up to $d^{1/p-1/2}$, so naively reducing to $\ell_{p,2}$ CSS gives a $d^{1/p-1/2}$-approximation factor. Thus, in both our streaming and distributed algorithms, we reduce the number of rows to $\widetilde{O}(k)$ (and thus obtain a $\widetilde{O}(k^{1/p-1/2})$ approximation factor) using the following tool:

  Sketching while Preserving Costs of All Small Column Subsets of A:
  Let $S \in \mathbb{R}^{t\times d}$ be a matrix with p-stable random variables as entries, where $t = k \cdot \text{polylog}(nd)$. Then, with high probability, for *all subsets* $T \subset [n]$ with size $k \cdot \text{polylog}(k)$, and all $V \in \mathbb{R}^{|T|\times n}$,

  $$\|A_T V - A\|_p \leq \|SA_T V - SA\|_p$$

  If T^*, V^* are the optimal column subset and corresponding optimal right factor, then

  $$\|SA_T V - SA\|_p \leq O(\log^{1/p}(nd))\|A_{T^*}V^* - A\|_p$$

  Since the lower bound holds **for all $T \subset [n]$ with $|T| = k \cdot \text{polylog}(k)$**, we can run a CSS algorithm on SA and simply choose the corresponding columns of A. To show the lower bound, we use a net argument **together with a union bound over all subsets T of size $\widetilde{O}(k)$**, different from previously known sketching arguments for affine embeddings/low rank approximation which only give no contraction for a single subspace.
- In our streaming algorithm, we divide the columns in the stream into contiguous batches of size $\widetilde{O}(k)$, compute coresets of those batches, and merge those coresets (by taking coresets of their concatenations) in a binary tree fashion using the merge and reduce framework (described in [McG14]). At the end of the stream, we concatenate all coresets and obtain one matrix SA_T, and run an $O(1)$ approximation algorithm for $\ell_{p,2}$ CSS based on [CW15]) to select an $\widetilde{O}(k)$-sized subset of SA_T. To identify the corresponding columns of A, for each coreset C of a batch of SA, we maintain the corresponding columns of A, using $\widetilde{O}(dk)$ space.
- In our distributed algorithm, each server i computes a coreset SA_iT_i of SA_i, and sends SA_iT_i to the coordinator. The coordinator concatenates SA_iT_i to form SA_T, and applies the $O(1)$ approximate $\ell_{p,2}$ CSS algorithm based on [CW15] to SA_T to select an $\widetilde{O}(k)$-sized subset $SA_{\widetilde{T}}$ of SA_T. Server i also sends A_iT_i to the coordinator (leading to $\widetilde{O}(sdk)$ communication overall), so that the coordinator can compute $A_{\widetilde{T}}$.

Greedy $\ell_{p,2}$ Column Subset Selection

We can also use the following greedy heuristic in our protocol, in the place of the $\ell_{p,2}$-CSS algorithm based on [CW15]:

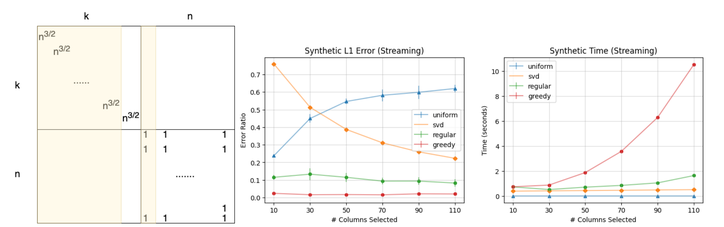
Input: $A \in \mathbb{R}^{d\times n}$, $k \in \mathbb{N}$, $p \in [1, 2)$, $r \leq n$, $\delta \in (0, 1)$.
Output: $T \subset [n]$ with $|T| = r$
$T \leftarrow \varnothing$
for $i = 1$ to r **do**
  $C \leftarrow$ Sample $\frac{n}{k}\log(\frac{1}{\delta})$ indices from $[n] \setminus T$ uniformly at random.
  Column index $j^* \leftarrow \text{argmin}_{j\in C}(\min_V \|A_{T\cup j}V - A\|_{p,2})$
  $T \leftarrow T \cup j^*$.
end for

We show that $r = \Theta(k/\varepsilon^2)$ (omitting problem-dependent parameters) suffices in order to have:

$$\mathbb{E}[\min_V \|A_T V - A\|_{p,2}] \leq \min_V \|A_L V - A\|_{p,2} + \epsilon\|A\|_{p,2}$$

To our knowledge no guarantees for greedy CSS in the $\ell_{p,2}$ norm were known before. ([ABF+16] shows a similar result for greedy CSS in the Frobenius norm, and our analysis is based on [ABF+16].)

Experiments



In our experiments, we study the case $p = 1$. We compare: (1) the rank-k SVD, (2) a uniformly random sampling baseline, (3) our algorithm using the $\ell_{1,2}$ CSS algorithm of [CW15], and (4) our algorithm using the greedy heuristic above for $\ell_{1,2}$ CSS. The last three algorithms output a subset of k columns of the data matrix A, and the rank-k SVD outputs a rank-k approximation to A. The sizes of the coresets T_i, and the number of rows in the p-stable matrix S, are chosen to be proportional to the target rank k. Here we show results on a synthetic dataset (shown in the figure above to the left) for target ranks $k \in \{10, 30, 50, 70, 90, 110\}$. Greedy k-CSS$_{1,2}$ denotes our protocol using the greedy heuristic, and Regular k-CSS$_{1,2}$ denotes our protocol using the algorithm of [CW15].

References

[ABF+16] Jason Altschuler, Aditya Bhaskara, Gang Fu, Vahab S. Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed algorithms. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2539–2548. JMLR.org, 2016.

[CGK+17] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p low-rank approximation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 806–814. PMLR, 2017.

[CW15] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 310–329. IEEE Computer Society, 2015.

[DWZ+19] Chen Dan, Hong Wang, Hongyang Zhang, Yuchen Zhou, and Pradeep Ravikumar. Optimal analysis of subset-selection based ℓ_p low-rank approximation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2537–2548, 2019.

[McG14] Andrew McGregor. Graph stream algorithms: A survey. *SIGMOD Rec.*, 43(1):9–20, May 2014.

[MW20] Arvind V. Mahankali and David P. Woodruff. Optimal ℓ_1 column subset selection and a fast PTAS for low rank approximation. *CoRR*, abs/2007.10307, 2020.

[SW18] Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 802–813. IEEE Computer Society, 2018.

[SWZ17] Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 688–701. ACM, 2017.

[SWZ19] Zhao Song, David P. Woodruff, and Peilin Zhong. Towards a zero-one law for column subset selection. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6120–6131, 2019.