Robust Hierarchical Matrix Approximation from Sketches

Diana Halikias, Tyler Chen, Feyza D. Keles, Cameron Musco, Christopher Musco, David Persson

Abstract

Sketching is a tool for dimensionality reduction that lies at the heart of many fast and highly accurate "matrix-free" algorithms for fundamental tasks such as solving linear systems and eigenvalue problems, low-rank approximation, and trace estimation. Broadly, to solve a problem in the sketching model, one only queries a matrix of interest $A \in \mathbb{R}^{n\times n}$ with relatively few matrix-vector products $x \mapsto Ax$ and $y \mapsto A^\top y$, as opposed to accessing and working with A's individual entries. The sketching model is increasingly prevalent in numerical linear algebra for three reasons. First, A may be unknown and accessible only via sketching. Second, even if A is known, it may be too large to operate on or fit in memory. Finally, many matrices that arise in applications exhibit structure that enables fast matrix-vector products.

Hierarchical matrices are one such matrix class that frequently arises in practice. These matrices exhibit low-rank structure away from the diagonal, which represents the smoothness of long-range interactions between points in a discretized domain. Shorter-range interactions are treated recursively, as they are subdivided into finer domains over which the matrix is approximately low-rank again. This structure has been exploited in a variety of applications, including fast direct solvers for differential and integral equations, discretizations of boundary integral operators, preconditioners, and even infinite-dimensional operator learning. Below, we define $H \in \mathbb{R}^{n\times n}$, a hierarchical matrix with 3 levels of partitioning. Each off-diagonal block is given by a rank-k factorization.

$$H = \begin{array}{|cc|cc|cc|}
\hline
H_{11} & W_3Z_3^\top & \multicolumn{2}{c|}{W_1Z_1^\top} & \multicolumn{2}{c|}{\multirow{2}{*}{$W_0Z_0^\top$}} \\
U_3V_3^\top & H_{22} & \multicolumn{2}{c|}{} & \multicolumn{2}{c|}{} \\
\hline
\multicolumn{2}{|c|}{\multirow{2}{*}{$U_1V_1^\top$}} & H_{33} & W_4Z_4^\top & \multicolumn{2}{c|}{\multirow{2}{*}{}} \\
\multicolumn{2}{|c|}{} & U_4V_4^\top & H_{44} & \multicolumn{2}{c|}{} \\
\hline
\multicolumn{2}{|c|}{\multirow{4}{*}{$U_0V_0^\top$}} & H_{55} & W_5Z_5^\top & \multicolumn{2}{c|}{\multirow{2}{*}{$W_2Z_2^\top$}} \\
\multicolumn{2}{|c|}{} & U_5V_5^\top & H_{66} & \multicolumn{2}{c|}{} \\
\multicolumn{2}{|c|}{} & \multicolumn{2}{c|}{\multirow{2}{*}{$U_2V_2^\top$}} & H_{77} & W_6Z_6^\top \\
\multicolumn{2}{|c|}{} & \multicolumn{2}{c|}{} & U_6V_6^\top & H_{88} \\
\hline
\end{array}$$

Peeling algorithms recover a hierarchical matrix like H using $\mathcal{O}(k\log_2(n))$ sketches with H and $H^\top$. In general, they selectively apply the randomized SVD to recover all of the low-rank blocks at a given level, starting with the top level. That is, using $\mathcal{O}(k)$ cleverly constructed input vectors which consist of alternating blocks of zeros and random Gaussian entries, one can restrict the outputs to sketch each of the individual low-rank blocks. Then, one can recover $W_0Z_0^\top$ and $U_0V_0^\top$ to high accuracy using a low-rank approximation algorithm. The learned blocks are stored in an approximation matrix $\tilde{H}^{(1)} \in \mathbb{R}^{n\times n}$:

$$\tilde{H}^{(1)} = \begin{bmatrix} 0 & \tilde{W}_0\tilde{Z}_0^\top \\ \tilde{U}_0\tilde{V}_0^\top & 0 \end{bmatrix}$$

These learned blocks are then "peeled" away, as the same process is applied to the matrix $H - \tilde{H}^{(1)}$ to recover $W_1 Z_1^\top, U_1 V_1^\top, W_2 Z_2^\top$, and $U_2 V_2^\top$ simultaneously with $\mathcal{O}(k)$ sketches. This is because $H - \tilde{H}^{(1)}$ zeros out the first level's off-diagonal blocks, and subsequent matrix-vector products can sketch the action of the low-rank blocks at the next level. Once learned, these blocks are stored along with the first level's blocks in $\tilde{H}^{(2)}$. The algorithm continues recursively, peeling away the learned blocks repeatedly and moving to finer blocks toward the diagonal. There are $\log_2(n)$ levels, and each is recovered using $\mathcal{O}(k)$ sketches, yielding an overall complexity of $\mathcal{O}(k \log_2(n))$ queries.

Peeling algorithms are extremely useful and observed to be stable in practice. However, the predetermined order of the algorithm, as well as its recursive subtraction, raise questions about its theoretical stability, particularly when the underlying matrix does not have hierarchical structure. For example, if the largest off-diagonal blocks are not exactly rank-k, but rather numerically rank-k as is often the case in applications, error may be propagated from the first level to all subsequent levels and deteriorate the overall approximation quality. In this talk, we describe the first provably stable and near-optimal variant of the peeling algorithm. That is, for a general matrix B, we use $\mathcal{O}(k \log_2^4(n)/\varepsilon^3)$ sketches to obtain an approximation $\tilde{B}$ satisfying $\|B - \tilde{B}\|_F \leq (1+\varepsilon)\|B - \hat{B}\|_F$, where $\hat{B}$ is the best hierarchical approximation to B. We complement this upper bound by proving that any matrix-vector query algorithm must use at least $\Omega(k \log_2(n) + k/\varepsilon)$ queries to obtain a $(1+\varepsilon)$-approximation.

We discuss the variety of techniques used to derive these results. To control the propagation of error between levels of hierarchical approximation, we introduce a new perturbation bound for low-rank approximation, which is of independent interest in numerical linear algebra. We show that the widely used Generalized Nyström method enjoys inherent stability when implemented with noisy matrix-vector products. This brings to light a surprising fact; the same result cannot be obtained if the more standard randomized SVD method is used for low-rank approximation within peeling.

For even stronger control of error buildup across recursive levels, we also introduce a new "randomly perforated" Gaussian sketching distribution. The key idea is to increase the sparsity of the query vectors, so that a higher fraction of nonzero blocks are set to zero. Thus, when recovering each block at a given level, we incur error due to a smaller number of inexactly recovered blocks from the previous levels. We note that this may not decrease the magnitude of error if the error is all concentrated on a few blocks. Thus, we choose the nonzero blocks of our sketches randomly, ensuring that the expected error when recovering each block at each level is small.

We also describe lower bounds on the query complexity of hierarchical matrix recovery and approximation. These results build on a growing body of work on lower bounds for adaptive matrix-vector product algorithms. We reduce the problem to fixed-pattern sparse matrix approximation, which arises when we restrict to recovering the diagonal block matrices of a hierarchical matrix, a strictly easier problem than hierarchical matrix recovery. Formally, we prove that, if we had an algorithm for finding a near-optimal hierarchical approximation with $\mathcal{O}(k/\varepsilon)$ sketches, then the result could be post-processed to obtain a near-optimal block-diagonal approximation, which we know to be impossible. This is then combined with a query complexity lower bound for exact recovery to obtain the lower bound of $\Omega(k \log_2(n) + k/\varepsilon)$.

Finally, I will emphasize how our work in hierarchical matrix approximation fits into the new paradigm of stability analysis for randomized sketching algorithms, which are increasingly common in modern linear algebra techniques. Moving forward, we may also consider analyzing the stability of recovery algorithms for the subfamily of hierarchical semi-separable matrices, which also frequently arise in practice. Studying peeling also provides insight into an analysis of other similar recursive algorithms, such as butterfly and skeletonization factorizations.