HODLRlib: A Library for Hierarchical Matrices

Sivaram Ambikasaran[1]**, Karan Raj Singh**[2]**, and Shyam Sundar Sankaran**[1]

1 Department of Mathematics, Indian Institute of Technology Madras **2** Department of Computational & Data Sciences, Indian Institute of Science

Summary

`HODLRlib` is a flexible library for working with matrices that have a Hierarchical Off-Diagonal Low-Rank (HODLR) (Ambikasaran & Darve, 2013) structure. The current version performs matrix operations like matrix-vector products, solving linear systems, Cholesky-like symmetric factorization and determinant computation in almost linear complexity.

Significant speedups are obtained with `HODLRlib` when dealing with matrices that have underlying data sparsity (rank-structured). Most rank-structured matrices fall in the class of Hierarchical matrices: some examples are $\mathcal{H}$ (Hackbusch, 1999; Hackbusch & Khoromsky, 2000), $\mathcal{H}^2$ (Hackbusch & Borm, 2002), HSS (Chandrasekaran, Gu, & Pals, 2006) and HODLR. These formats differ from each other in their matrix partitioning, low-rank sub-blocks and nested basis behaviour. A thorough comparison of these different formats has been provided in Ambikasaran (2013).

Among these hierarchical matrices, HODLR matrices are applicable for a wide range of problems of practical interest (Ambikasaran, Foreman-Mackey, Greengard, Hogg, & O'Neil, 2016; Hartmann, Ingold, & Neto, 2018; Kressner & Susnjara, 2017). In addition, they possess the simplest data-sparse representation, which allows for easier implementation and obtaining large speed-ups.

`HODLRlib` strives to provide functionality for `HODLR` matrices similar to that which libraries such as STRUMPACK (Ghysels, Li, Rouet, Williams, & Napov, 2016) and H2lib (Hackbusch & Borm, 2002) provide for the HSS and $\mathcal{H}^2$ matrix formats respectively. Some unique features of our library are Cholesky-like symmetric factorization and determinant computation. While it is difficult to determine which matrix format works best, studies (F. H. Rouet, 2015) in the past have revealed that it isn't a case of "one size fits all". Rather, it reveals that the optimum hierarchical matrix format is dependent on the problem and the size considered.

`HODLRlib` is an optimized implementation of the ideas illustrated in Ambikasaran & Darve (2013), Ambikasaran et al. (2016) and Ambikasaran, O'Neil, & Singh (2014). The goal of `HODLRlib` is to serve as a high-performance, easy to use reference implementation for working with any HODLR matrix. We believe that apart from the features provided by the library, the simplicity and the extendability of the implementation distinguishes ours from other libraries. Our interfaces are kept simple with a minimum number of dependencies, with well-documented code to ensure that a potential user/developer can hit the ground running as soon as possible.

Our code makes use of shared-memory parallelism through OpenMP. The solver is fairly generic and can handle matrices not necessarily arising out of kernel functions. Further,

the solver has been optimized and the running time of the solver is now massively (a few orders of magnitude) faster than the running times reported in the original articles.

`HODLRlib` is designed such that the matrix corresponding to the linear system to be solved is abstracted through the `HODLR_Matrix` object, which needs to have the function `getMatrixEntry`. This function takes in the arguments as the index in the matrix and returns the particular entry, which facilitates the reduction in storage costs, since only a few entries of the low-rank sub-blocks are needed to reconstruct these sub-blocks. This instance of `HODLR_Matrix` is then passed to the `HODLR_Tree` class, whose methods are used for the various matrix operations.

The current release has the following capabilities:

- MatVecs: Obtains Ax at a cost of $\mathcal{O}\left(N \log N\right)$
- Factorization: Factors the matrix A into the desired form at a cost of $\mathcal{O}\left(N \log^2\left(N\right)\right)$
- Cholesky-like Symmetric Factorization: Obtains $A = WW^T$ at a cost of $\mathcal{O}\left(N \log^2\left(N\right)\right)$
- Solve: Solves linear systems $Ax = b$ at an additional cost of $\mathcal{O}\left(N \log\left(N\right)\right)$
- Determinant Computation: Additional Cost of $\mathcal{O}\left(N \log N\right)$

`HODLRlib` is released under the MPL2 license.

Acknowledgements

References

Ambikasaran, S. (2013). *Fast algorithms for dense numerical linear algebra and applications* (PhD thesis). Stanford University.

Ambikasaran, S., & Darve, E. (2013). An $\mathcal{O}(N \log N)$ fast direct solver for partial hierarchically semi-separable matrices. *Journal of Scientific Computing*, *57*(3), 477–501. doi:10.1007/s10915-013-9714-z

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., & O'Neil, M. (2016). Fast direct methods for gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, *38*(2), 252–265. doi:10.1109/TPAMI.2015.2448083

Ambikasaran, S., O'Neil, M., & Singh, K. R. (2014). Fast symmetric factorization of hierarchical matrices with applications. *arXiv e-prints*, arXiv:1405.0223.

Chandrasekaran, S., Gu, M., & Pals, T. (2006). A fast *ULV* decomposition solver for hierarchically semiseparable representations. *SIAM Journal on Matrix Analysis and Applications*, *28*(3), 603–622. doi:10.1137/S0895479803436652

Ghysels, P., Li, X. S., Rouet, F.-H., Williams, S., & Napov, A. (2016). An efficient multicore implementation of a novel hss-structured multifrontal solver using randomized sampling. *SIAM Journal on Scientific Computing*, *38*(5), S358–S384. doi:10.1137/15M1010117

Hackbusch, W. (1999). A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$ matrices. *computing*, *62*(2), 89–108. doi:10.1007/s006070050015

Hackbusch, W., & Borm, S. (2002). Data sparse approximation by adaptive $\mathcal{H}^2$ - matrices. *computing*, *69*(1), 1–35. doi:10.1007/s00607-002-1450-4

Hackbusch, W., & Khoromsky, B. N. (2000). sparse $\mathcal{H}$ - matrix arithmetic. Part II. Application to multi-dimensional problems. *computing*, *64*(1), 21–47. doi:10.1007/s006070050002

Hartmann, M., Ingold, G.-L., & Neto, P. A. M. (2018). Advancing numerics for the casimir effect to experimentally relevant aspect ratios. *Physica Scripta*, *93*(11), 114003. doi:10.1088/1402-4896/aae34e

Kressner, D., & Susnjara, A. (2017). Fast computation of spectral projectors of banded matrices. *SIAM Journal on Matrix Analysis and Applications*, *38*(3), 984–1009. doi:10.1137/16M1087278

Rouet, F. H. (2015). A comparison of different low-rank approximation techniques. Retrieved from https://personalpages.manchester.ac.uk/staff/theo.mary/doc/LA15.pdf