Chair of Scientific Computing in Computer Science
School of Computation, Information and Technology
Technical University of Munich

Scalable Hierarchical Approximation of Dense Kernel Matrices

Keerthi Gaddameedi, Severin Reiz, Tobias Neckel, and Hans-Joachim Bungartz

Motivation

Diffusion Maps → Eigendecomposition →

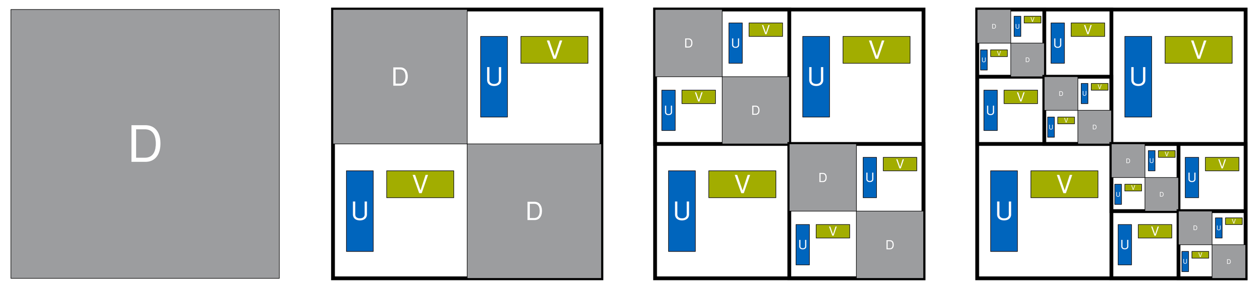
BLAS Level	Memory	Computations
(1) Vector-Vector	$\mathcal{O}(N)$	$\mathcal{O}(N)$
(2) Matrix-Vector	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
(3) Matrix-Matrix	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$

→ GOFMM

- Investigate scalability and efficiency of hierarchical matrix approximations from GOFMM for dense kernel matrices obtained in manifold learning algorithms such as Diffusion Maps

Geometry-Oblivious Fast Multipole Method

Given a dense SPD matrix $K \in \mathbb{R}^{N\times N}$, we construct a hierarchically low-rank approximation $\tilde{K} = D + UV$ with small relative error $||K - \tilde{K}||/||K||$, where D is a block-diagonal matrix and U, V are low-rank matrices [4].



- **Compression(K)**:
  1. HierarchicalPartitioning()
  2. NeighborBasedPruning()
  3. Skeletonization()

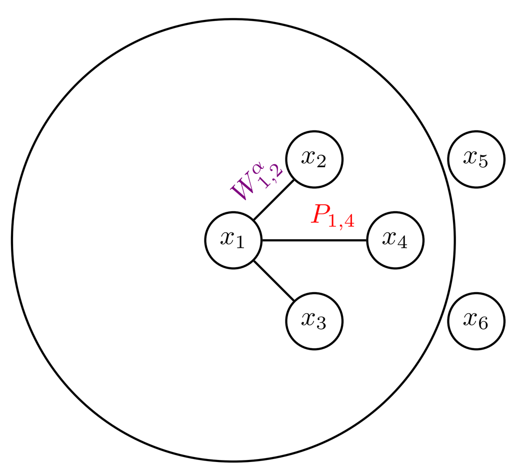
- **HierarchicalPartitioning()**:
  1. Split K into a ball tree with m leaves
  2. Uses a distance metric
  3. Geometric, kernel or angle distance

- **NeighborBasedPruning()**:
  1. Compute neighbor, near and far interaction lists
  2. Blocks in far list are approximated

- **Skeletionization()**:
  1. Approximate off-diagonal blocks
  2. Interpolative decomposition

Diffusion Maps

Manifold learning algorithm for dimensionality reduction of non-linear datasets.
It computes the dimension of the underlying manifold using an affinity matrix of the data points [1].



Given a dataset
$X = x_1, x_2, x_3 \ldots, x_n$

1. Gaussian kernel $W_{ij} = e^{\frac{-||x_i - x_j||_2^2}{\sigma^2}}$,
2. σ = radius of neighborhood
3. Q = Degree of a node
4. α = Influence of density on underlying geometry
   $\alpha = 0 \implies$ Maximal influence

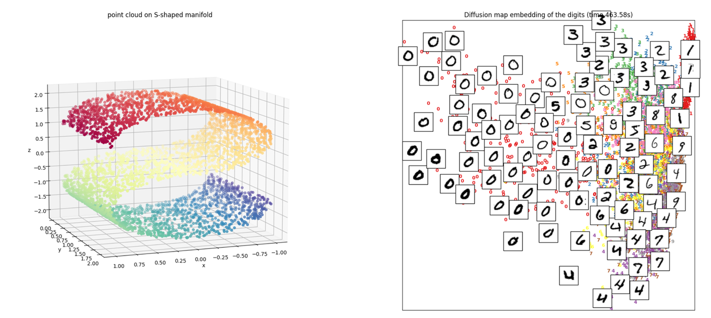
- **DiffusionMaps()**:
  1. Compute the affinity matrix W_{ij}
  2. Normalize the matrix $W^{\alpha}_{ij} = \frac{W_{ij}}{Q_i^{\alpha} Q_j^{\alpha}}$
  3. Define a Markov chain $P_{ij} = \frac{W^{\alpha}_{ij}}{Q_i^{\alpha}}$
  4. Perform t random walks to obtain P^t
  5. Eigendecomposition(P^t) $\rightarrow$ Bottleneck
  6. Lower dimension $d(t)$ = max{ $l : \lambda_l^t > \delta\lambda_1^t$ }

Hierarchical Decomposition of Kernel Matrices

1. Point Clouds:

- Scurve and MNIST datasets with 16k samples
- Point cloud manifold generated using layer 3 in the datafold [3] library

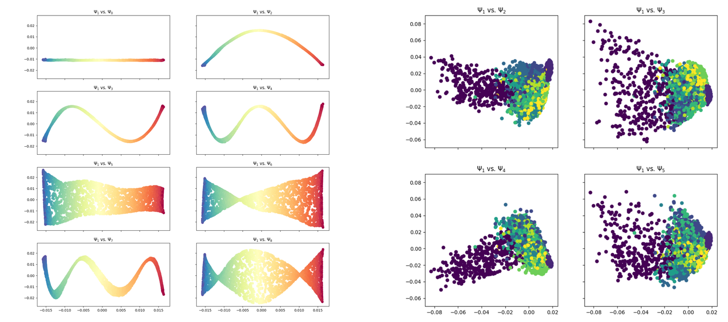
Workflow hierarchy of datafold library



Scurve MNIST

2. Diffusion Maps:

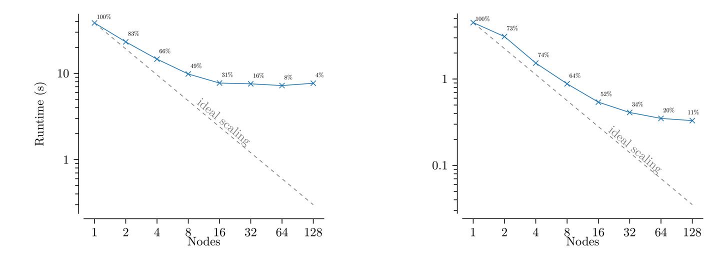
- Computation of intrinsic dimension of the PCManifold
- Involves computing given number of eigenpairs of the kernel matrix
- A fraction of the largest eigenpairs are then used to map the data in a lower-dimensional space
- Implementation of the algorithm in datafold.dynfold



Eigenvector comparison

3. Eigendecomposition:

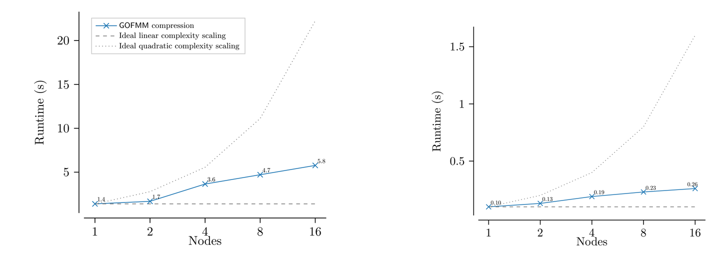
- Diffusion maps algorithm requires eigendecomposition of kernel matrices
- Implicitly restarted Arnoldi iteration is used from scipy.sparse.linalg.eigsh
- Matrix-Vector products are the computational bottleneck of such methods



GOFMM Compression GOFMM Evaluation
a.) Strong scaling

4. Scalability of GOFMM on SuperMUC-NG:

- 6D point clouds of sizes up to $200k$ with an accuracy of order 1e-4
- Scaling done for problem sizes up to $100k \times 100k$ [2]
- Compression(K) - Compute hierarchical approximations of kernel matrices
- Computational complexity of matvec($\tilde{K}$) reduced to $\mathcal{O}(NlogN)$



GOFMM Compression GOFMM Evaluation
b.) Weak scaling

References

[1] Ronald Coifman and Stéphane Lafon. *Diffusion Maps.* Applied and computational harmonic analysis, 21(1):5–30, 2006.

[2] Keerthi Gaddameedi et al. "Efficient and Scalable Kernel Matrix Approximations Using Hierarchical Decomposition". In: *Intelligent Computers, Algorithms, and Applications.* Ed. by Christophe Cruz, Yanchun Zhang, and Wanling Gao. Singapore: Springer Nature Singapore, 2024, pp. 3–16. ISBN: 978-981-97-0065-3.

[3] Daniel Lehmberg et al. "datafold: data-driven models for point clouds and time series on manifolds". In: *JOSS* 5.51 (2020), p. 2283. DOI: 10.21105/joss.02283. URL: https://doi.org/10.21105/joss.02283.

[4] Chenhan D. Yu et al. "Geometry-oblivious FMM for compressing dense SPD matrices". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* SC '17. Denver, Colorado: Association for Computing Machinery, 2017. ISBN: 9781450351140. DOI: 10.1145/3126908.3126921. URL: https://doi.org/10.1145/3126908.3126921.