ExaNIML
An Exascale Library for Numerically Inspired Machine Learning

Severin Reiz[§], Hasan Ashraf[§], Tobias Neckel[§], George Biros[†], Hans-Joachim Bungartz[§]
[§] Technical University of Munich, [†] University of Texas at Austin, reiz@in.tum.de, ashraf@in.tum.de, neckel@in.tum.de, gbiros@ices.utexas.edu, bungartz@in.tum.de

Introduction

Motivation

- Significant gap in communities: Machine Learning (ML) $\leftrightarrow$ high-performance computing (HPC)
- ML needs considerable computing power $\rightarrow$ we need adequate software!
- ExaNIML: Library with algorithms
  - with modern applications from ML community
  - with enough concurrency for next generation distributed computing systems

Approach

- **Methods** from scientific computing domain
  - "Undervalued" near-linear complexity methods (fast multipole methods)[1]
  - Adaptive sparse grids to mitigate curse of dimensionality[2]
- **HPC:** Exploit potential of supercomputers
  - **Concurrency:** Choose suitable algorithms for parallel computing
  - Extract **computational bottlenecks** as low-level drivers in C++ or Kokkos
  - **Performance Portability** in-light of the upcoming new GPU and CPU architectures

Classification with Kernel Methods

Kernel Methods

Kernel Matrix

Occurs in many domains ...

- multi-class classification
- model-order reduction
- uncertainty quantification
- partial differential equations

Example: Binary classification

Ridge regression

- N data points $x_i \in \mathbb{R}^d$ and N binary labels y_i
- $f(x) = \text{sign}(\sum_{i=1}^{N} k(x, x_i)w_i) \rightarrow u = f(x_{test}) = K * w$
- Solving a linear system: kernel matrix K often not stable, nearly singular. Solve $\tilde{K} \rightarrow K + \lambda I$ instead

Our approach: Kernel Matrix Approximation

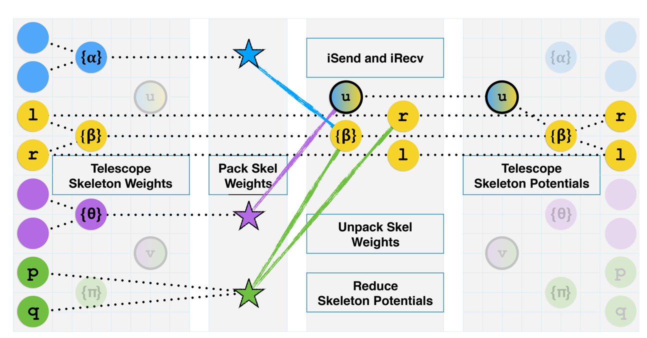
- Often K is a dense N-by-N matrix; this quadratic complexity often is the **computational bottleneck**
- To reach $\mathcal{O}(N)$ algorithms it requires approximation
- For the majority of applications off-diagonal blocks of K admit good low-rank approximations

Many ML libraries offer Kernel methods: to our knowledge none of them offer **kernel matrix approximation**

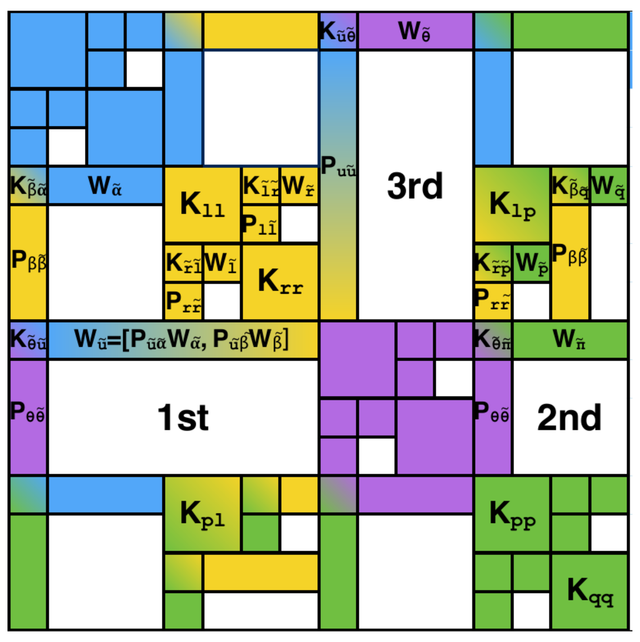
Key Computational Bottlenecks

Geometry-oblivious Fast Multipole Method[1]

- Hierarchically off-diagonal low-rank
- Speeds up algebraic operations



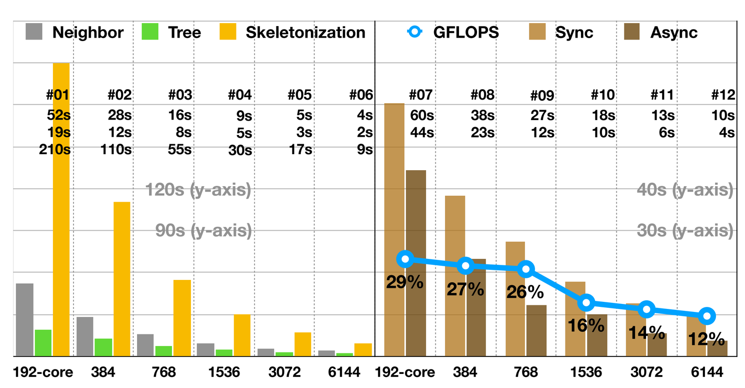
Dependency graph for asynchronous task analysis



4-process distributed $\mathcal{H}$-Matrix compression. Mixed colored sections and factors are shared for distributed nodes

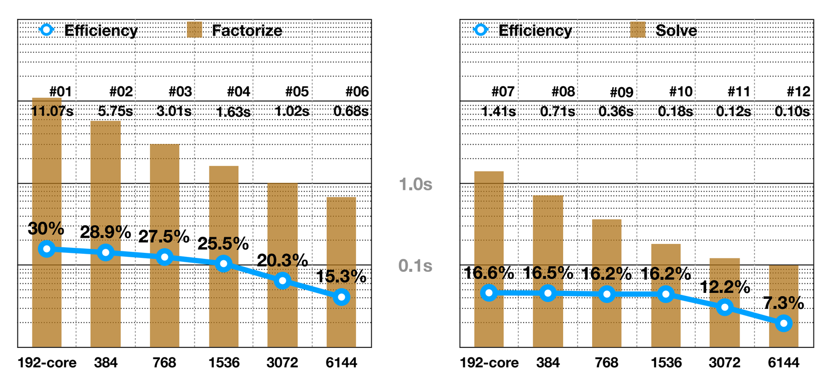
First scalability results

Multiplication[1]



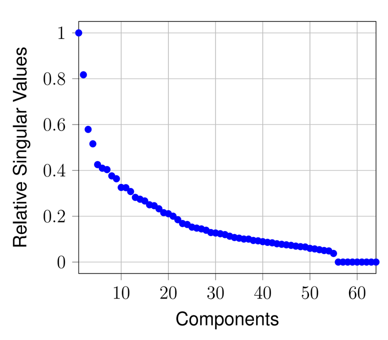
Time for compression (left) and multiplication (right) for a 6-d gaussian kernel matrix of 64M-by-64M

$\mathcal{O}(N)$ linear solver[3]

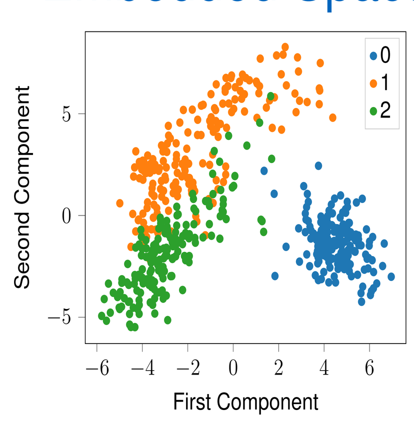


Time for ULV-Factorization (left) and forward-solve (right)

Dimensionality Reduction



Embedded Space



Reduce the dimensionality of dataset

Manifold Learning Algorithms

- (Kernel) Principal Component Analysis
- Isomap algorithm
- Hessian local eigenmaps, ...

Classification on Embedded Space

- Example forced to 2D manifold (plotting)
- Classification on lower dimensional manifold

$\rightarrow$ Sparse grid classification[4]

Approximation with Sparse Grids

Sparse Grids

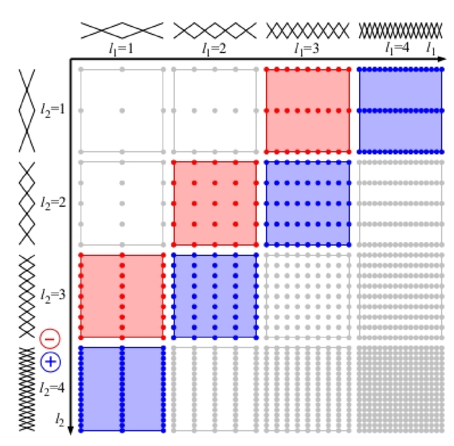
Sparse grids

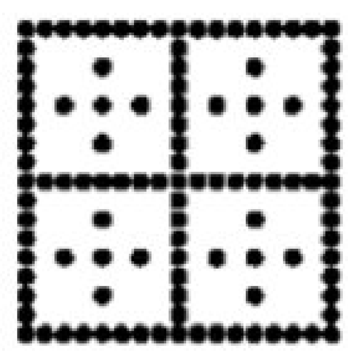
- Reduce number of grid points
- One approach: Combi-Technique Combine Full Grids (red and blue, c.f. figure on right)
- Suitable for 5-20 dimensions

Sparse Grids in Embedded Space

1. Manifold learning algorithm for **coarse** embedded space
2. **Fine** approximation in embedded space with **Sparse Grids**

Synergy between *Point-based* and *Grid-based* Methods





Interfaces

MPI-GOFMM
Kernel Matrix Approximation

scikit learn
Manifold Learning

TensorFlow
Deep Learning
ResNet, MobileNet,...

SG++
Sparse grid library
Data mining with
Sparse grid density estimation

Conclusion

- Method design
  - Run prominent models from current machine learning peers
  - Combine models with **hierarchical** kernel and **sparse grid** methods
- Library design
  - **Community/reproducibility:** ExaNIML library for others to play

References

[1] C. D. Yu, S. Reiz, and G. Biros, "Distributed-memory hierarchical compression of dense SPD matrices," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, SC '18, (Piscataway, NJ, USA), pp. 15:1–15:15, IEEE Press, 2018.

[2] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta numerica*, vol. 13, pp. 147–269, 2004.

[3] D. Y. Chenhan, S. Reiz, and G. Biros, "Distributed o (n) linear solver for dense symmetric hierarchical semi-separable matrices," in *2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, pp. 1–8, IEEE, 2019.

[4] B. Peherstorfer, D. Pflüger, and H.-J. Bungartz, "Density estimation with adaptive sparse grids for large data sets," in *Proceedings of the 2014 SIAM international conference on data mining*, pp. 443–451, SIAM, 2014.