SPARCITY

An Optimization and Co-design Framework for Sparse Computation

SparCity for Sparse Tensors: Study on Feature Extraction and Smart Tensor Generation

MAELSTROM Dissemination workshop ✧ Nov. 7, 2023

Tugba Torun (ttorun@ku.edu.tr)

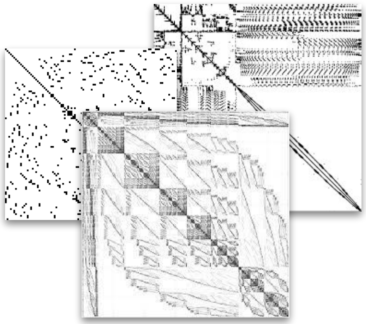
Koç University

EuroHPC Joint Undertaking

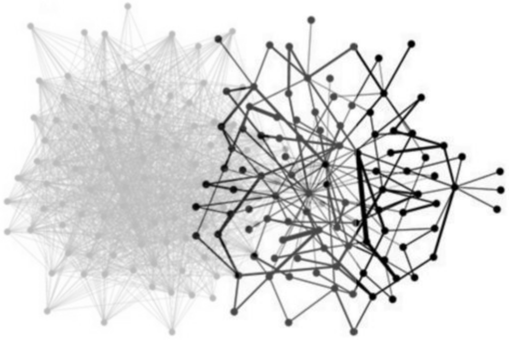
This project has received funding from the European High-Performance Computing Joint Undertaking under grant agreement No. 956213.

Feature extraction of sparse computation

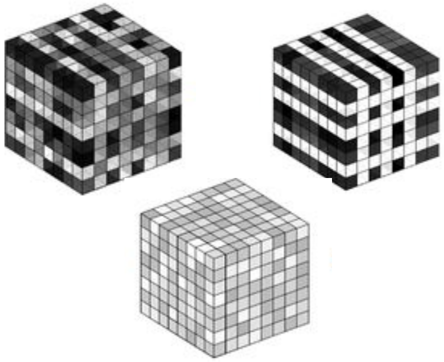
SPARSE MATRICES



GRAPHS



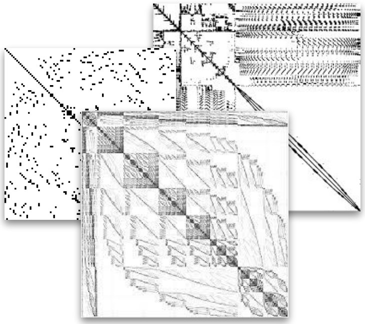
SPARSE TENSORS



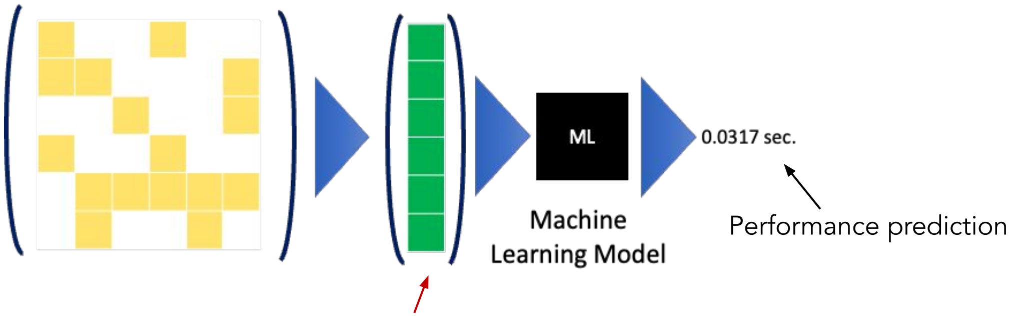
- Efficient sparse computing depends on statistical features that describe the sparsity pattern
- More than 100 extracted features integrated in one large and practical set of sparse features
- Used as inputs for automated sparse format and kernel selection ML-based approaches
- Low-overhead extraction methods for integration in the SparseBase framework

Feature extraction of sparse matrices

SPARSE MATRICES



SEMI-SUPERVISED ML FOR SPMV FORMAT SELECTION

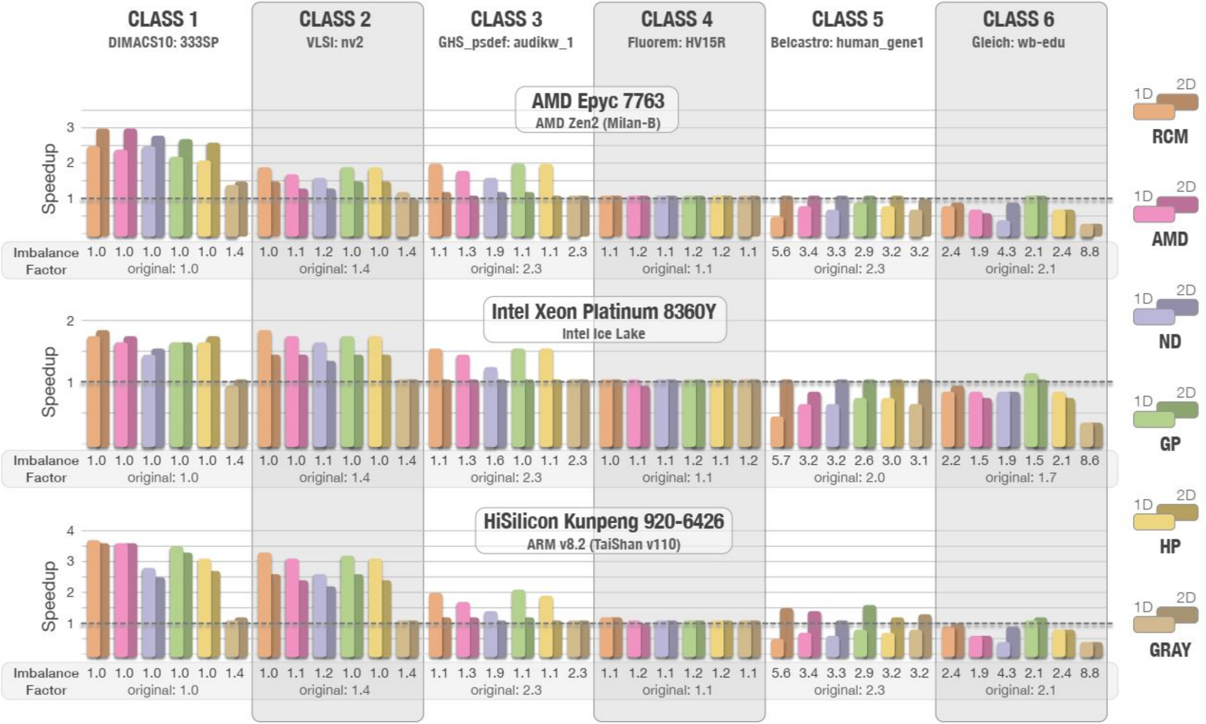


Which features can predict performance of sparse kernels ?

- Classification of features in three distinct categories: single-value, 1D and 2D features
  - Statistics to describe distribution of non-zero elements (per row, matrix, diagonal etc.)
  - Matrix pattern representation: row histograms (1D) and matrix bitmaps (2D)
- Features to express locality of non-zero elements: cache and group load/reuse rates
- First semi-supervised ML for sparse format selection: 21 features, 3 GPU architectures, 4 formats

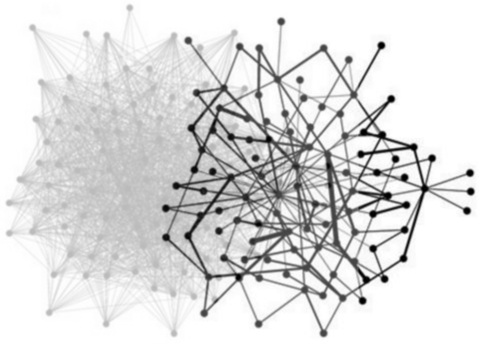
Feature extraction of sparse matrices

- Bringing Order to Sparsity: A Sparse Matrix Reordering Study on Multicore CPUs (will be at SC’23!)
  - Evaluated 6 reordering algorithms on 490 matrices across 8 multicore architectures for SpMV
  - Correlation with features: bandwidth, profile, offdiagonal nonzero count, load imbalance factor

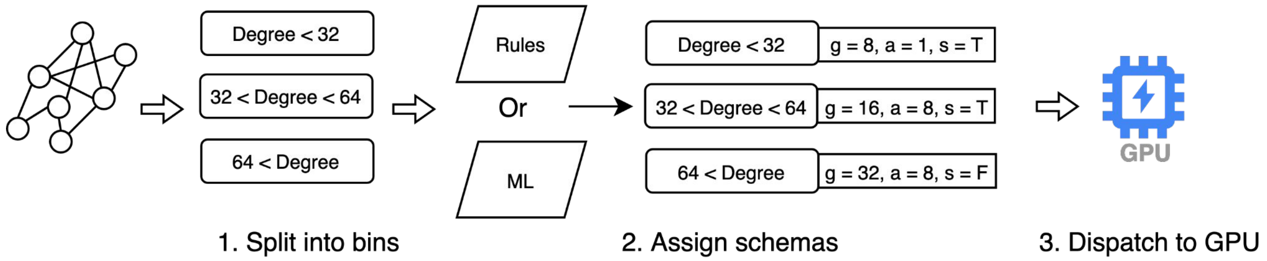


Feature extraction of graphs

GRAPHS



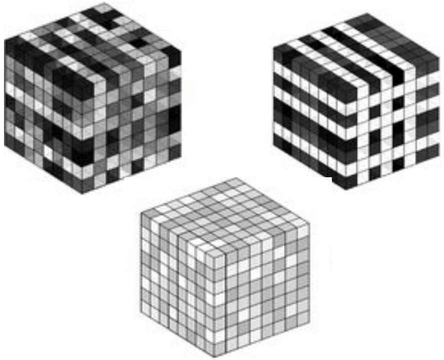
ML-DRIVEN JACCARD WEIGHT COMPUTATION ON GPUs

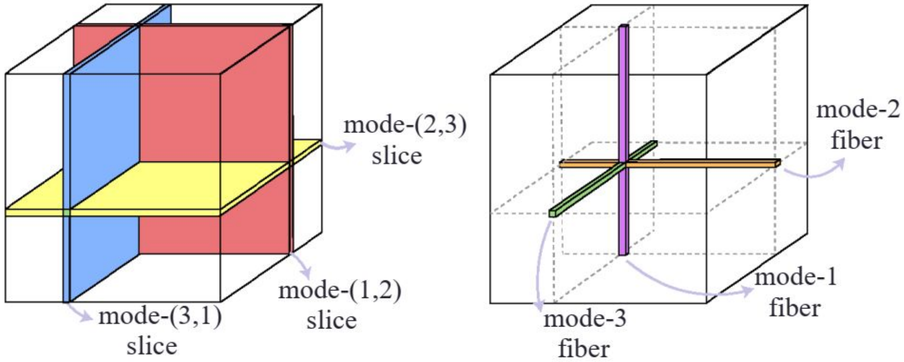


- Graphs are powerful data models, but highly irregular and hard to use in traditional ML
    - Graph embedding for dimensionality reduction: feature vectors for each vertex (costly, memory intensive)
- Focus on graph metrics usable as node and edge features in Machine and Deep Learning
    - Collection of features commonly used in literature: measures for centrality, betweeness, degrees, etc.
- Jaccard Weights (a special type of edge weight): coefficient between two vertices' neighborhoods
    - Proposed degree-aware computation distribution to improve load balancing of calculations
    - GPU kernel resources assigned via ML classifier

Feature extraction of sparse tensors

SPARSE TENSORS





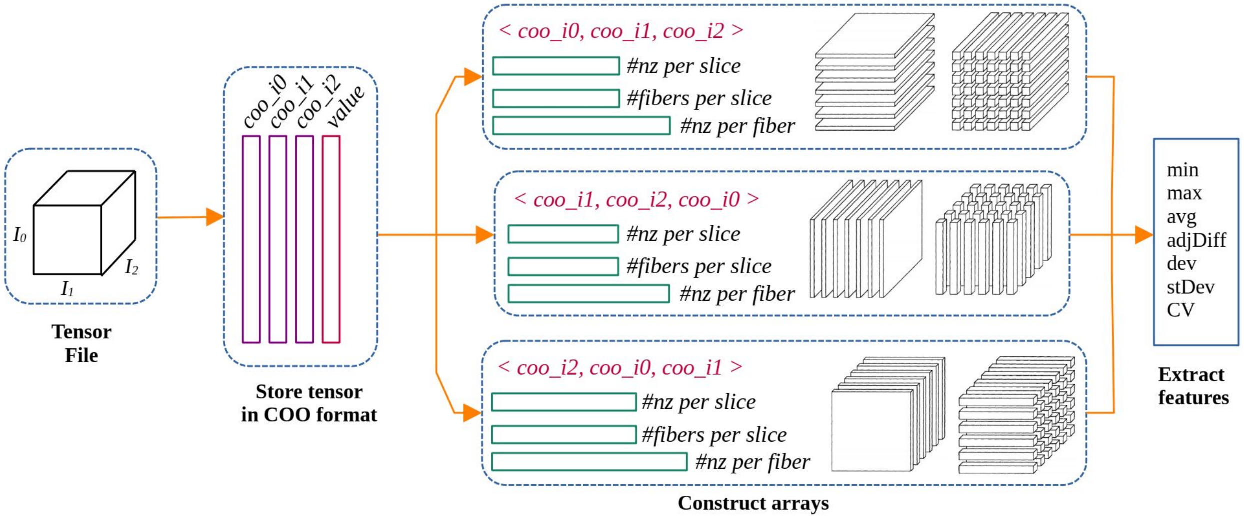
- Tensor : multi-dimensional array
  - *mode*: Dimension
  - *fiber :* Fixing every index but one
  - *slice:* Fixing every index but two
- Due to large sizes of real tensors, even feature extraction may take hours without caution
- Research focus on efficient extraction methods with low overheads

Feature Extraction Methods

- Methods for sparse tensor feature extraction (Total of 142 features for 3-mode tensors)
  - Naive: memory-bound, useful for small tensors
  - Hash-based : to eliminate memory issues
  - Sorting-based: conventional approach. first sort, then construct arrays
  - Grouping-based: semi-sorting, and constructing arrays simultaneously
  - Hybrid : combination of sorting & grouping

Global features

Feature	Description
size_m	Tensor mode size (in mode m)
nnz	Number of nonzeros
density	Density of nnz in the tensor
fiberCnt	Number of nz fibers
sliceCnt	Number of nz slices
fiberDensity	The density of nz fibers
sliceDensity	The density of nz slices



Mode & type-dependent features

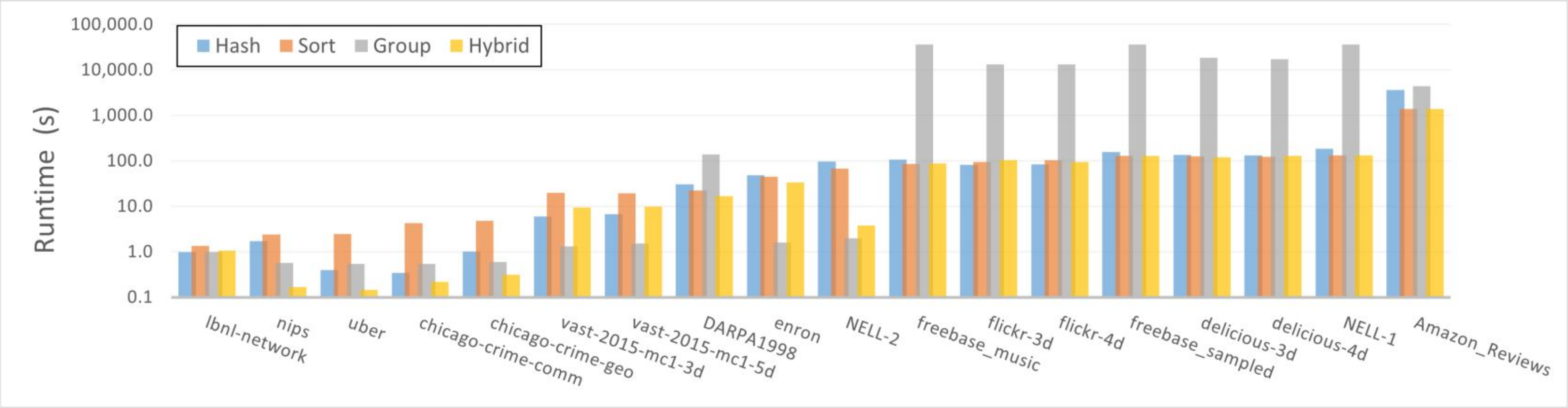
Feature	Description
all_cnt	all including zero
nz_cnt	nonzero count
nz_density	nonzero sparsity
max	max nonzero count
min	min nonzero count
dev	deviation (max-min)
avg	average nonzero count
imbal	imbalance: (max-avg)/max
stDev	standard deviation
cv	coefficient of variation
avg_onlynz	avg by excluding empty
imbal_onlynz	imbal by excluding empty
stDev_onlynz	stDev by excluding empty
cv_onlynz	cv by excluding empty

Dataset: 18 real-world tensors from FROSTT and HaTeN2 collections

- nine 3-mode, six 4-mode, and three 5-mode tensors (size < 75 GB)

							Sparsity		
Name	I_0	I_1	I_2	I_3	I_4	NNZ	Overall	Slice	Fiber
vast-2015-mc1-3d	165,427	11,374	2	-	-	2.6E+07	6.9E-03	1.0E+00	1.4E-02
1998darpa	22,476	22,476	23,776,223	-	-	2.8E+07	2.4E-09	1.0E+00	5.2E-05
freebase_music	23,343,790	23,344,784	166	-	-	1.0E+08	1.1E-09	9.8E-01	4.0E-07
freebase_sampled	38,954,435	38,955,429	532	-	-	1.4E+08	1.7E-10	9.5E-01	2.1E-07
nell1	2,902,330	2,143,368	25,495,389	-	-	1.4E+08	1.3E-11	1.0E+00	1.9E-06
nell2	12,092	9,184	28,818	-	-	7.7E+07	2.4E-05	1.0E+00	5.3E-02
amazon-reviews	4,821,207	1,774,269	1,805,187	-	-	1.7E+09	1.1E-10	1.0E+00	6.2E-05
delicious-3d	532,924	17,262,471	2,480,308	-	-	1.4E+08	3.2E-11	1.0E+00	2.9E-06
flickr-3d	319,686	28,153,045	1,607,191	-	-	1.1E+08	7.8E-12	1.0E+00	2.8E-06
uber	183	24	1,140	1,717	-	3.3E+06	3.8E-04	1.0E-01	6.7E-03
chicago-crime-comm	6,186	24	77	32	-	5.3E+06	1.5E-02	8.6E-01	2.5E-01
enron	6,066	5,699	244,268	1,176	-	5.4E+07	5.5E-09	6.9E-03	7.6E-06
nips	2,482	2,862	14,036	17	-	3.1E+06	1.8E-06	4.6E-02	7.3E-05
delicious-4d	532,924	17,262,471	2,480,308	1,443	-	1.4E+08	8.2E-12	4.2E-06	9.0E-11
flickr-4d	319,686	28,153,045	1,607,191	731	-	1.1E+08	2.9E-11	3.6E-06	1.9E-11
vast-2015-mc1-5d	165,427	11,374	2	100	89	2.6E+07	7.8E-07	3.2E-04	5.8E-06
chicago-crime-geo	6,185	24	380	395	32	6.3E+06	8.9E-06	2.5E-02	5.0E-04
lbnl-network	1,605	4,198	1,631	4,209	868,131	1.7E+06	5.2E-13	2.5E-07	9.9E-11

Comparing Feature Extraction Methods



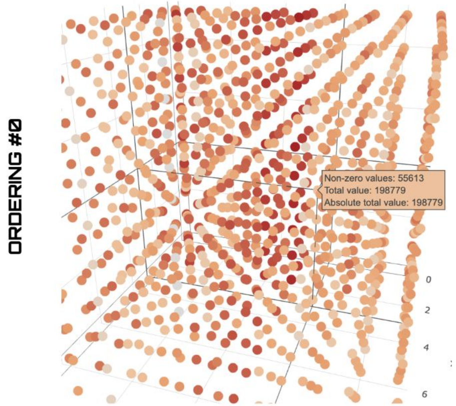
- Machine: AMD EPYC 7352 CPU (3.2GHz)
  - CPU: 2 x [24 cores, 48 threads]
  - Memory: 512 GB (8 x 64 GB DDR4)
- Compared the performance of feature extraction methods (using 96 threads)
  - Sort-based method is better for large tensors; Grouping-based method is better for smaller tensors
  - Hybrid method is the best for the average case (sweet spot)
    - 86% faster than grouping, 59% faster than sorting (on average)

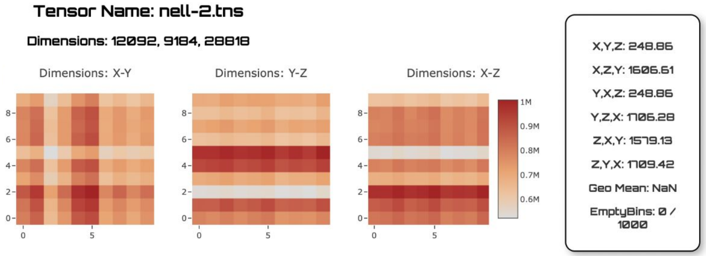
SparseBase: Pre-processing Base for Sparse Computation

https://github.com/sparcityeu/sparsebase

We develop SparseBase :

- Feature extraction for sparse matrices, tensors, and graphs
- Different storage formats and conversion
- Reordering algorithms
- Partitioning algorithms
- Visualization of reorderings

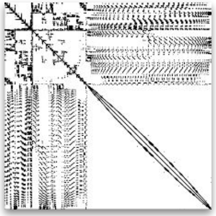




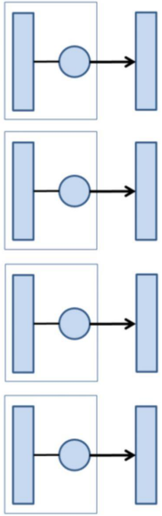
SparseBase: Pre-processing Base for Sparse Computation

https://github.com/sparcityeu/sparsebase

Sparse Data

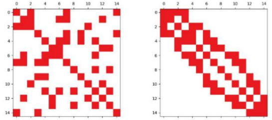


Sparsebase

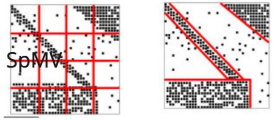


Parallel I/O

Reordering



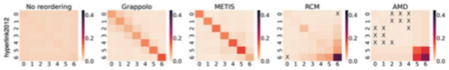
Partitioning



Multiple Formats

JAD ← CRS ← ELL PACK ← COO

Structural Features



Sparse Kernels

- Breadth First Search
- SpMM
- SpMV
- Connected Components
- Tensor Decomposition
- Sparse SVD
- Page Rank
- ...

Publications

- James D. Trotter, Sinan Ekmekçibaşı, Johannes Langguth, Tugba Torun, Emre Düzakın, Aleksandar Ilic, and Didem Unat (2023). Bringing Order to Sparsity: A Sparse Matrix Reordering Study on Multicore CPUs. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC ’23). Association for Computing Machinery, New York, NY, USA, Article 31, 1–13. DOI: 10.1145/3581784.3607046. - will be presented on Nov 14 in SC’23 (High Performance for Graph Operations)

- Amro Alabsi Aljundi, Taha Atahan Akyıldız, and Kamer Kaya (2022). Degree-Aware Kernels for Computing Jaccard Weights on GPUs. 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 897-907. DOI: http://doi.org/10.1109/IPDPS53621.2022.00092.

- Sunidhi Dhandhania, Akshay Deodhar, Konstantin Pogorelov, Swarnendu Biswas and Johannes Langguth (2021). Explaining the Performance of Supervised and Semi-Supervised Methods for Automated Sparse Matrix Format Selection, 50th International Conference on Parallel Processing Workshop, pp. 1-10. DOI: https://doi.org/10.1145/3458744.3474049.

Thank you!