SPLATT: Enabling Large-Scale Sparse Tensor Analysis

Shaden Smith and George Karypis
Department of Computer Science & Engineering, University of Minnesota

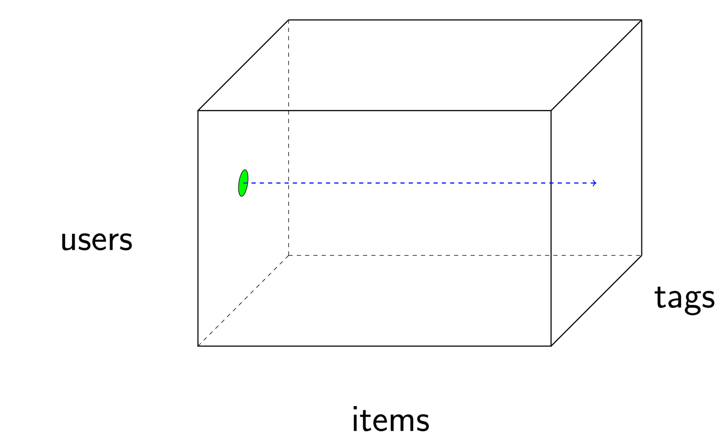
SPLATT: The Surprisingly ParalleL spArse Tensor Toolkit

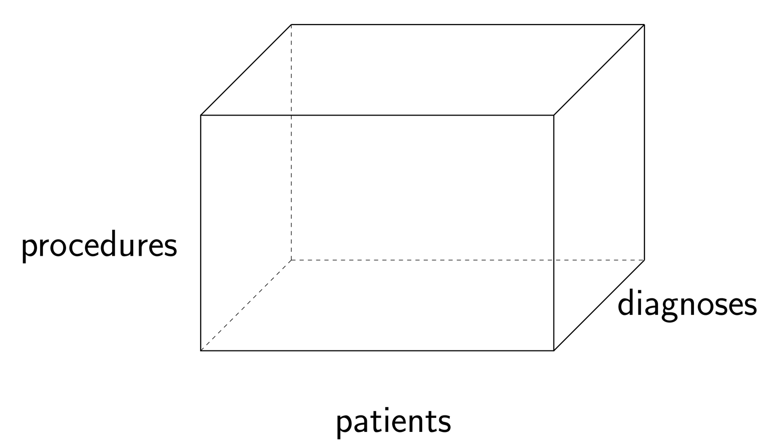
http://cs.umn.edu/~splatt/

Tensor Introduction

Tensors

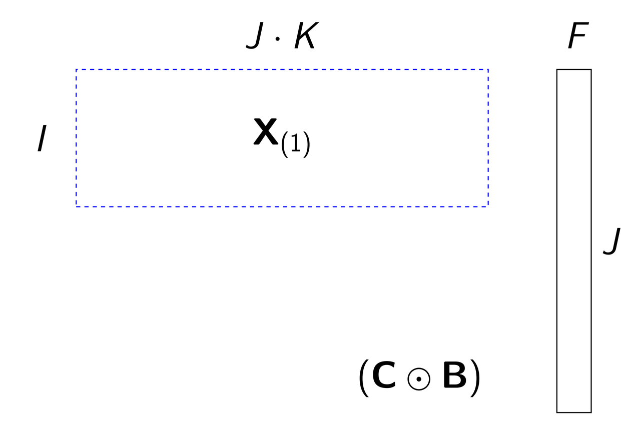
- *Tensors* are the generalization of matrices





Important Tensor & Matrix Operations

- Tensor $\mathcal{X}$ can be *matricized* along one mode
  - $\mathcal{X}$ is $I \times J \times K$ and $\mathbf{X}_{(1)}$ is $I \times JK$
- The *Khatri-Rao* product is the column-wise Kronecker product
  - $\mathbf{B}$ is $J \times F$ and $\mathbf{C}$ is $K \times F$, $(\mathbf{C} \odot \mathbf{B})$ is $JK \times F$
  - Effectively "stretches" matrices to match matricized tensor

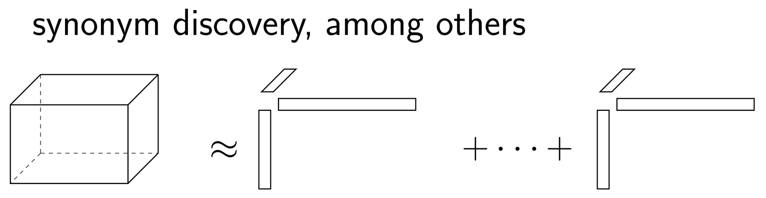


The Canonical Polyadic Decomposition

- The CPD is an extension of the SVD to tensors
- We are interested in *low-rank* factorizations, with $F \ll \{I, J, K\}$

$$\underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\text{minimize}} \frac{1}{2} \|\mathbf{X}_{(1)} - (\mathbf{C} \odot \mathbf{B})^{\intercal} \mathbf{A}\|_F^2$$

- Uses in item recommendation, personalized healthcare, synonym discovery, among others



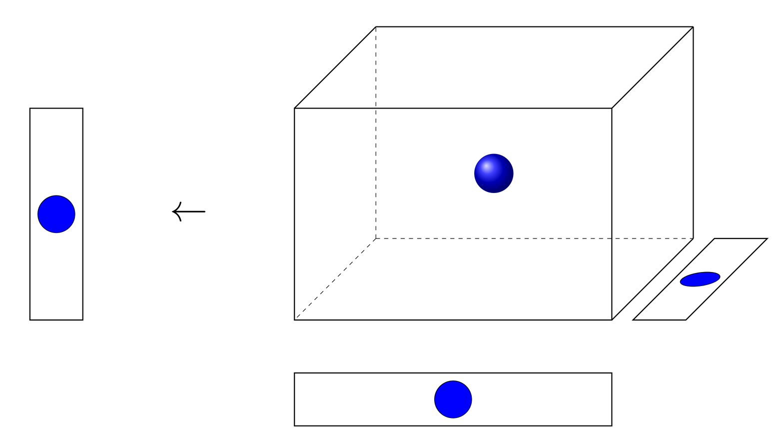
Matricized Tensor Times Khatri-Rao Product

- The CPD is typically computed using alternating least squares

$$\mathbf{A} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \left[(\mathbf{C} \odot \mathbf{B})^{\intercal} (\mathbf{C} \odot \mathbf{B})\right]^{\dagger}$$

- MTTKRP is the bottleneck of CPD-ALS

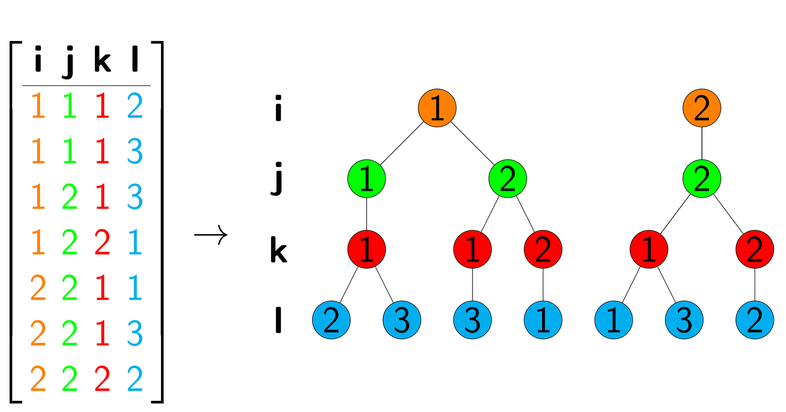
$$\mathbf{A}(i,:) \leftarrow \mathbf{A}(i,:) + \mathcal{X}(i,j,k) \left[\mathbf{B}(j,:) * \mathbf{C}(k,:)\right]$$



Shared-Memory Computation

Compressed Sparse Fiber (CSF)

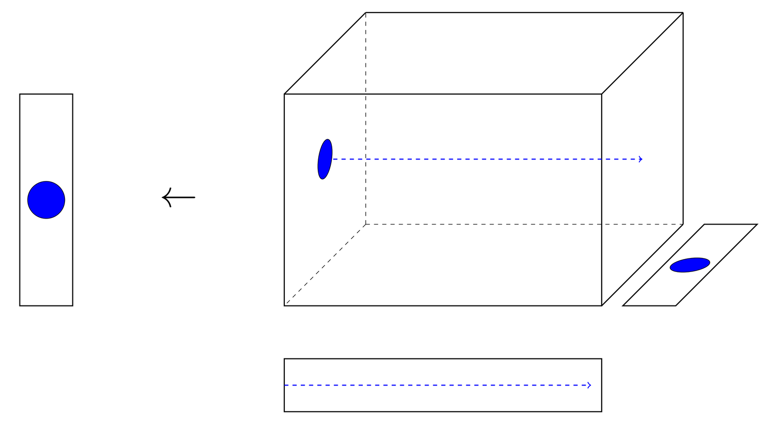
- CSF naturally exposes opportunities for saving operations



MTTKRP with CSF

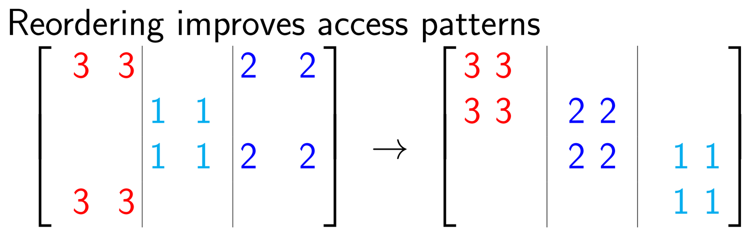
- We process whole *fibers* at a time instead of non-zeros
- Fewer operations and memory accesses!

$$\mathbf{A}(i,:) \leftarrow \mathbf{A}(i,:) + \mathbf{C}(k,:) * \left[\sum_{\mathcal{X}(i,:,k)} \mathcal{X}(i,j,k)\mathbf{B}(j,:)\right]$$

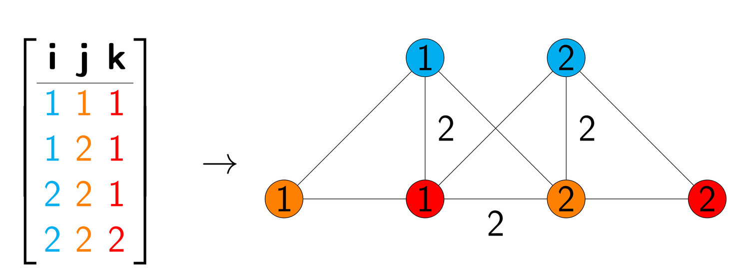


Reordering

- Reordering improves access patterns

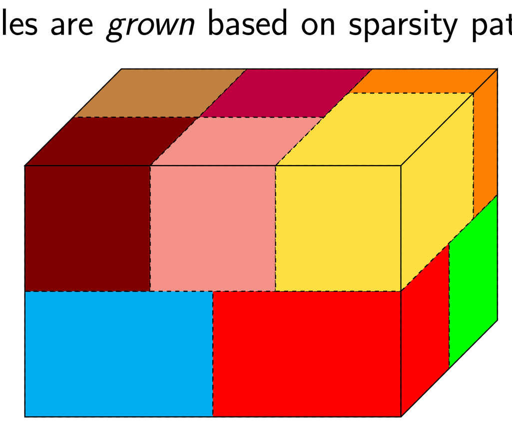


- A tri-partite graph models the sparsity pattern
- Partitioning induces a reordering of each mode



Cache Tiling

- Rescheduling nonzeros improves temporal locality
- Cache tiles are *grown* based on sparsity pattern



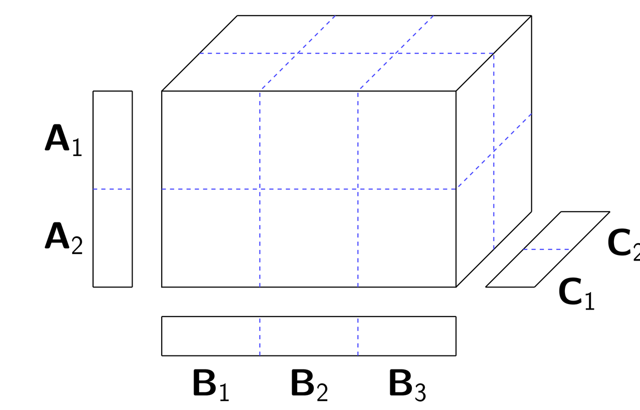
Distributed-Memory Computation

Two Communication Overheads

- We must decide how non-zeros are assigned to processes
- If non-zeros $\mathcal{X}(i,j,k_1)$ and $\mathcal{X}(i,j,k_2)$ are owned by different processes:
  1. Local MTTKRP results on processes must be aggregated
  2. Rows $\mathbf{C}(k_1,:)$ and $\mathbf{C}(k_2,:)$ must be exchanged each iteration

Medium-Grained Decomposition

- Our *medium*-grained decomposition is a great middle ground between coarse- and fine-grained
- Processes collaborate in *layers* to keep communication small while maintaining balance
- Within an MPI process, all shared-memory optimizations still apply!

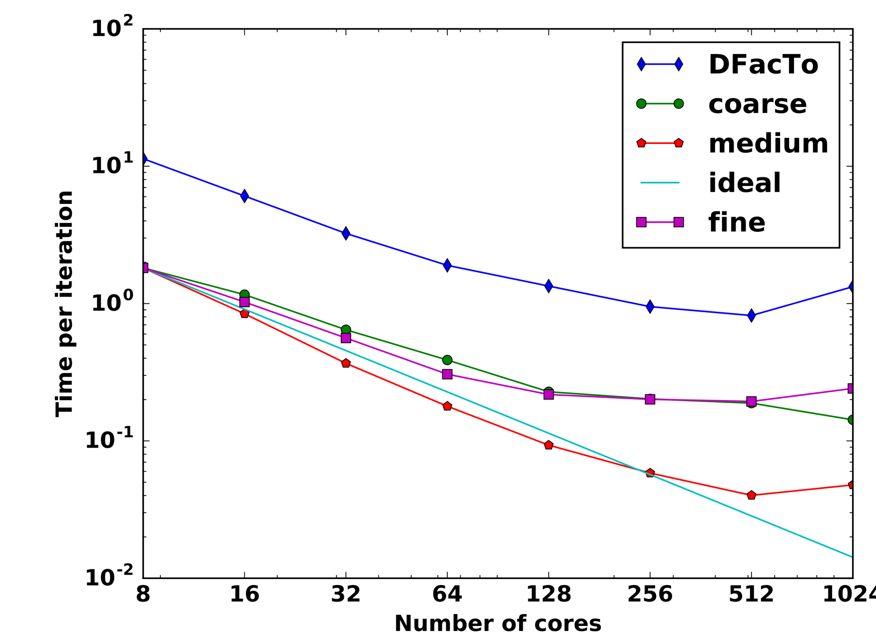


Tensor Decompositions

- *Coarse*-grained decompositions assign whole tensor slices to each process
  - No MTTKRP aggregation!
- *Fine*-grained decompositions assign non-zeros at an individual level
  - Lowest communication volume on average, but communication can be *imbalanced*

Scaling: Netflix

- 100 million (user, movie, day) ratings
- Over 25× faster than competing open source software!



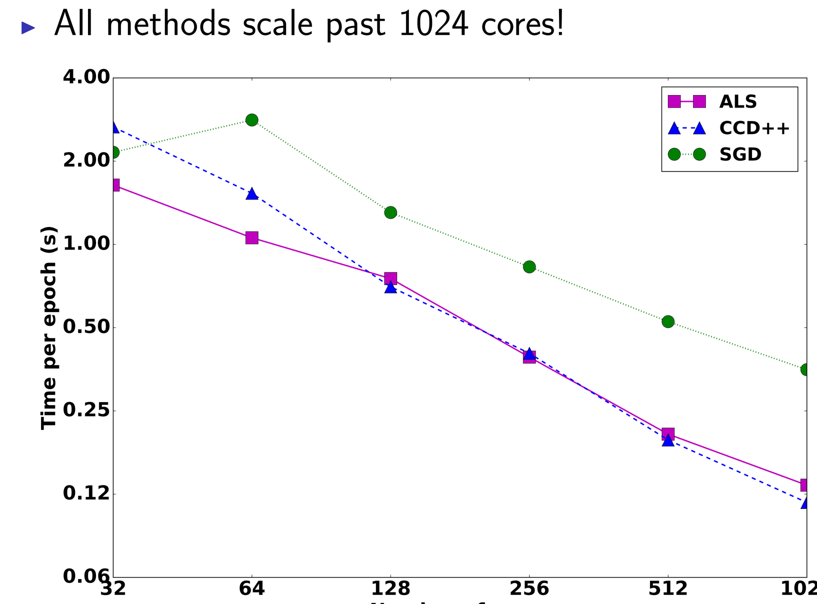
Tensor Completion – ongoing work with Jongsoo Park @ Intel PCL

Missing Values

- The traditional CPD does not apply when values are *missing* instead of numerically zero
- Tensor *completion* is concerned with predicting missing values
- The optimization problem is now defined over only the observed entries

Scaling: Yahoo! Music

- 250 million (user, song, month) ratings
- All methods scale past 1024 cores!

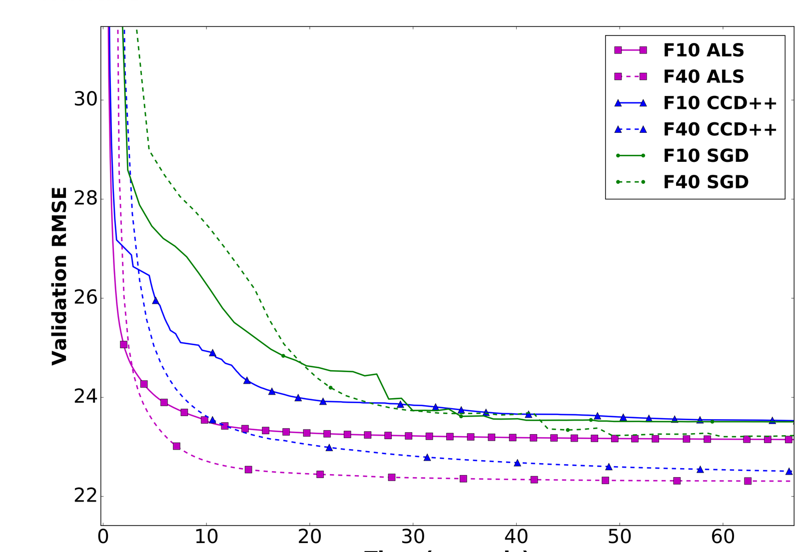


Optimization Algorithms

- Three algorithms are popular for matrices
  1. ALS
  2. Coordinate descent (CCD++)
  3. Stochastic Gradient Descent (SGD)
- No optimization algorithm is best in all cases
- Best method depends on factorization rank and parallel architecture

Convergence: Yahoo! Music

- All algorithms reach similar solutions in less than a minute



Publications

1. Shaden Smith, Jongsoo Park, and George Karypis. An exploration of optimization algorithms for high performance tensor completion. In *Proceedings of the 2016 ACM/IEEE conference on Supercomputing*, 2016, to appear. *Best Student Paper Finalist.*
2. Shaden Smith and George Karypis. A medium-grained algorithm for distributed sparse tensor factorization. In *30th IEEE International Parallel & Distributed Processing Symposium (IPDPS'16)*, 2016.
3. Shaden Smith and George Karypis. Tensor-matrix products with a compressed sparse tensor. In *5th Workshop on Irregular Applications: Architectures and Algorithms*, 2015.
4. Shaden Smith, Niranjay Ravindran, Nicholas D. Sidiropoulos, and George Karypis. SPLATT: Efficient and parallel sparse tensor-matrix multiplication. In *International Parallel & Distributed Processing Symposium (IPDPS'15)*, 2015.

Acknowledgments

Access to research and computing facilities was provided in part by the Minnesota Supercomputing Institute and the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work was supported in part by NSF (IIS-0905220, OCI-1048018, CNS-1162405, IIS-1247632, IIP-1414153, IIS-1447788), Army Research Office (W911NF-14-1-0316), Intel Software and Services Group, and the Digital Technology Center at the University of Minnesota.