BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

Sampling-based decomposition algorithms for arbitrary tensor networks

Osman Malik
Applied Mathematics & Computational Research Division
Berkeley Lab

Tensor Network Reading Group · 30 January 2024

Our focus: Decomposition of *large* tensors

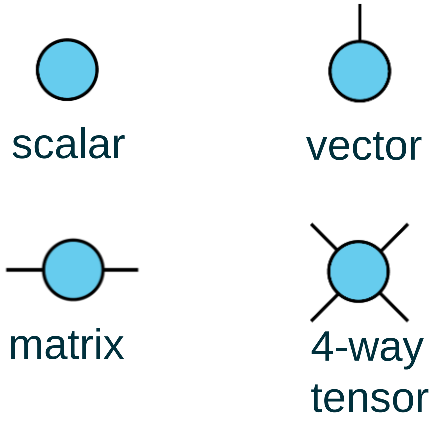
Example: Analyzing internet traffic

- Collected by Center for Applied Internet Data Analysis (CAIDA) at UCSD [Kepner et al., IEEE HPEC, 2021]
- During 2019 and 2020 over 40,000,000,000,000 (40 *trillion*!) unique packets were collected
- Can be represented at 3-way tensor with entry (s, d, t) indicate the number of packets sent from source s to destination d at time t
- We look at a small subtensor with 6.9 billion nonzeros and size $3.6\ \mathrm{m} \times 11\ \mathrm{m} \times 54\ \mathrm{k}$ in [Bharadwaj et al., preprint, 2022]
- Full dataset stored on magnetic tape
  $\Rightarrow$ Expensive to look at the dataset

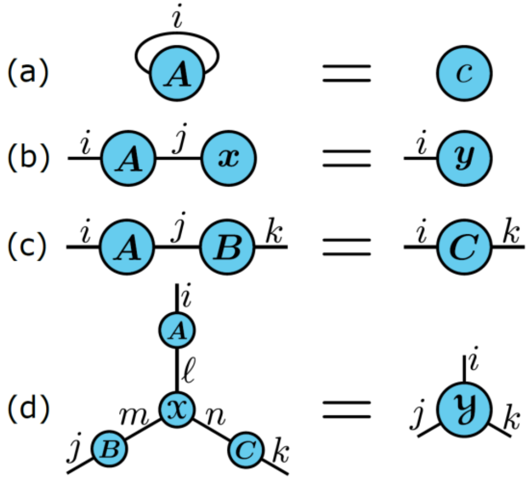
Graphical tensor network notation

- Graphical notation:

<u>Basic building blocks</u>



<u>Graphical representations of decompositions…</u>



<u>…and their mathematical formulations</u>

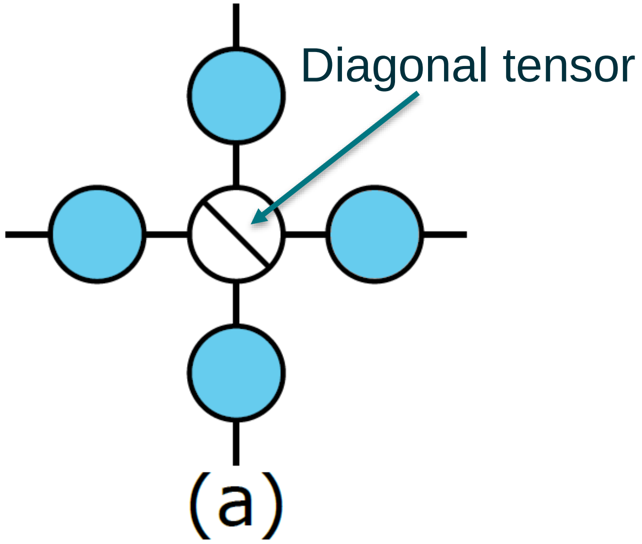
(a) $\sum_i \boldsymbol{A}_{ii} = \mathrm{trace}(\boldsymbol{A}) = c$,

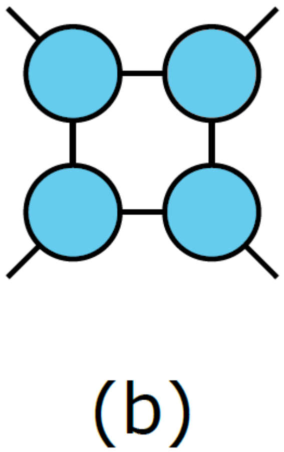
(b) $\sum_j \boldsymbol{A}_{ij} \, \boldsymbol{x}_j = \boldsymbol{y}_i$,

(c) $\sum_j \boldsymbol{A}_{ij} \, \boldsymbol{B}_{jk} = \boldsymbol{C}_{ik}$,

(d) $\sum_{\ell mn} \boldsymbol{\mathcal{X}}_{\ell mn} \boldsymbol{A}_{i\ell} \boldsymbol{B}_{jm} \boldsymbol{C}_{kn} = \boldsymbol{\mathcal{Y}}_{ijk}$.

Some other tensor decompositions

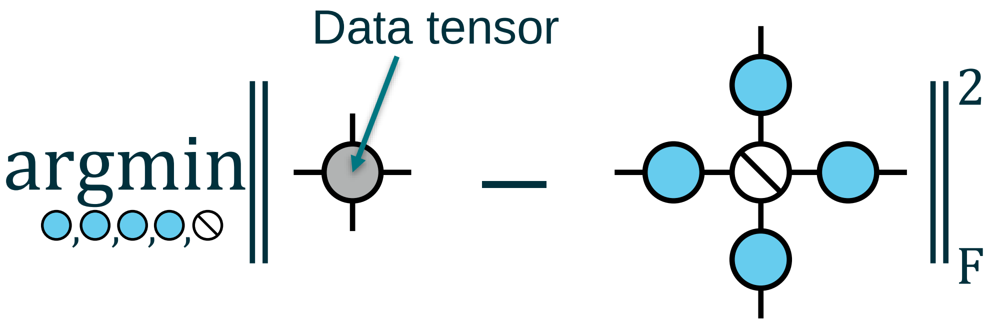




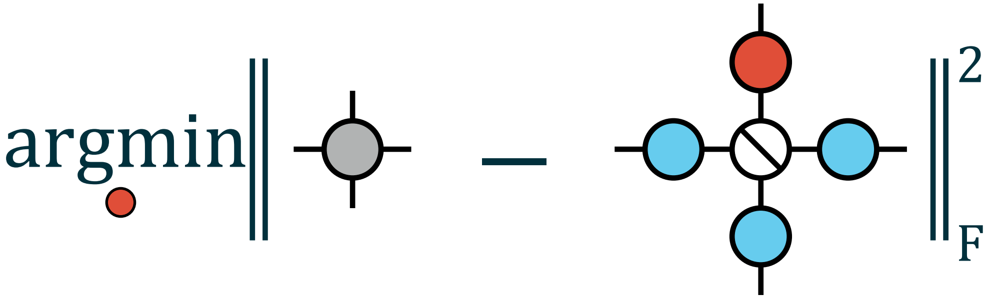
(a) CP Decomposition: $X_{ijkl} = \sum_{r=1}^{R} \lambda_r \; A_{ir} \; B_{jr} \; C_{kr} \; D_{lr}$

(b) Tensor ring decomposition: $X_{ijkl} = \sum_{rstu} A_{uir} B_{rjs} C_{skt} D_{tlu}$

Alternating least squares (ALS) for tensor fitting



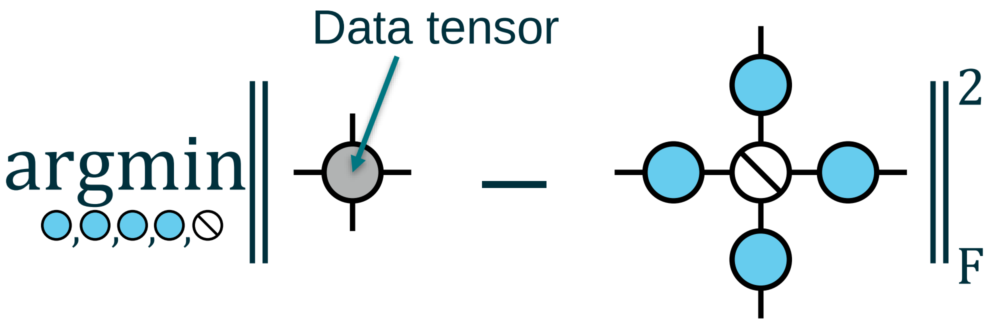
Difficult, non-convex problem!



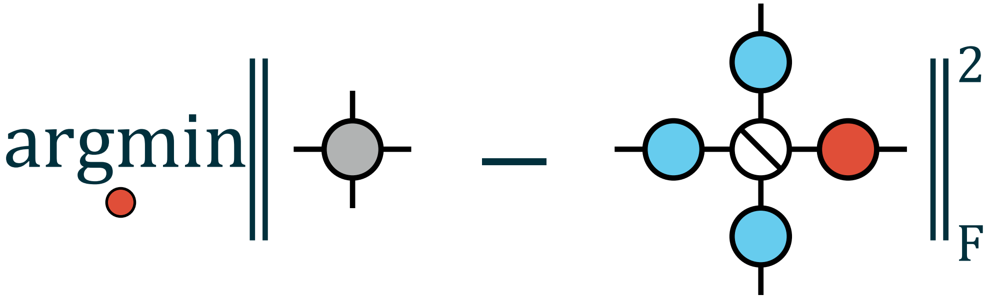
ALS instead updates one thing at a time

→ A linear least squares problem

Alternating least squares (ALS) for tensor fitting



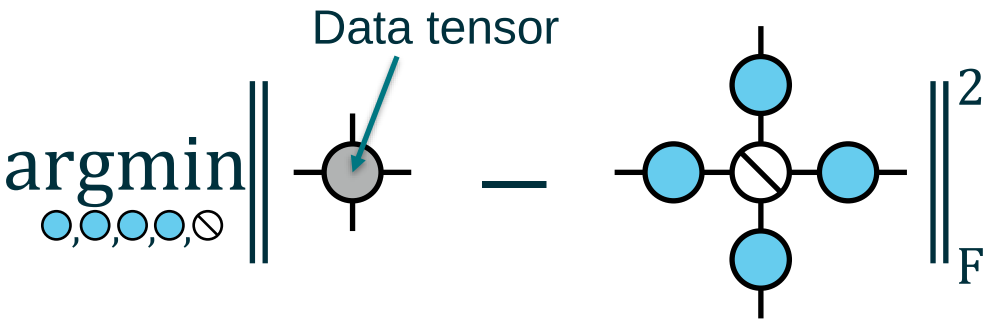
Difficult, non-convex problem!



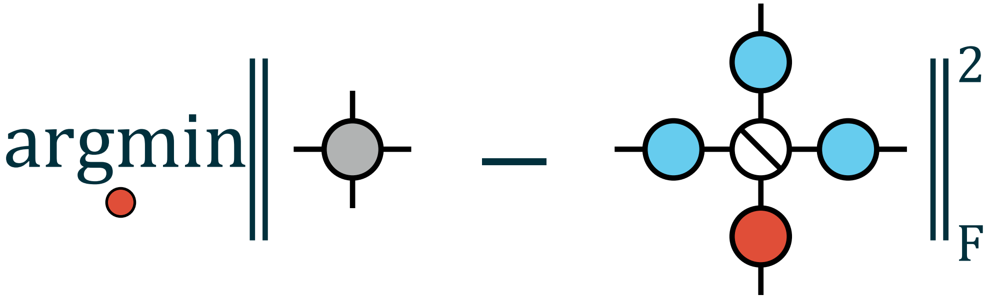
ALS instead updates one thing at a time

→ A linear least squares problem

Alternating least squares (ALS) for tensor fitting



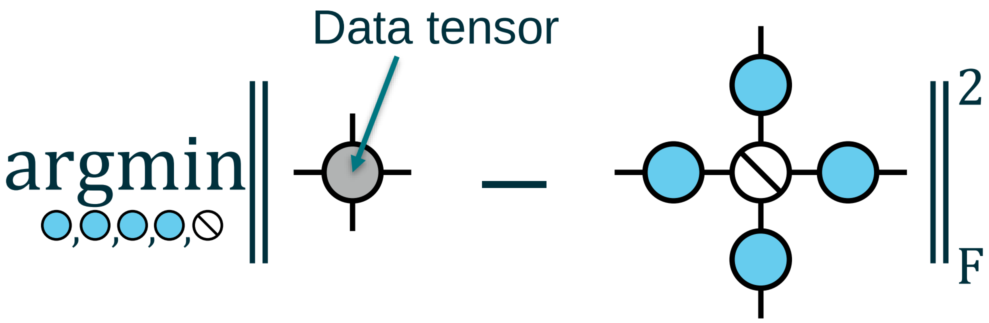
Difficult, non-convex problem!



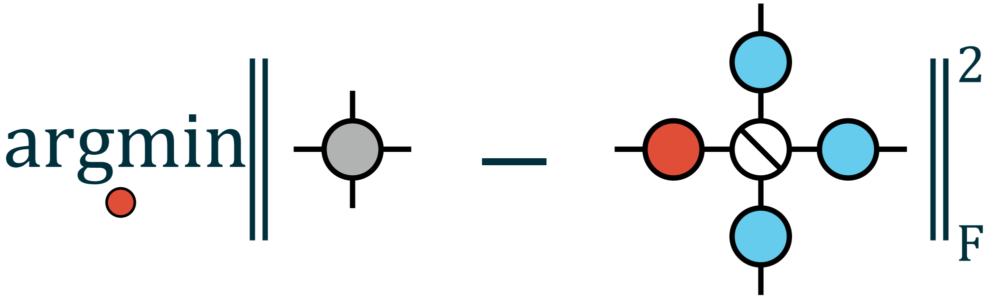
ALS instead updates one thing at a time

→ A linear least squares problem

Alternating least squares (ALS) for tensor fitting



Difficult, non-convex problem!

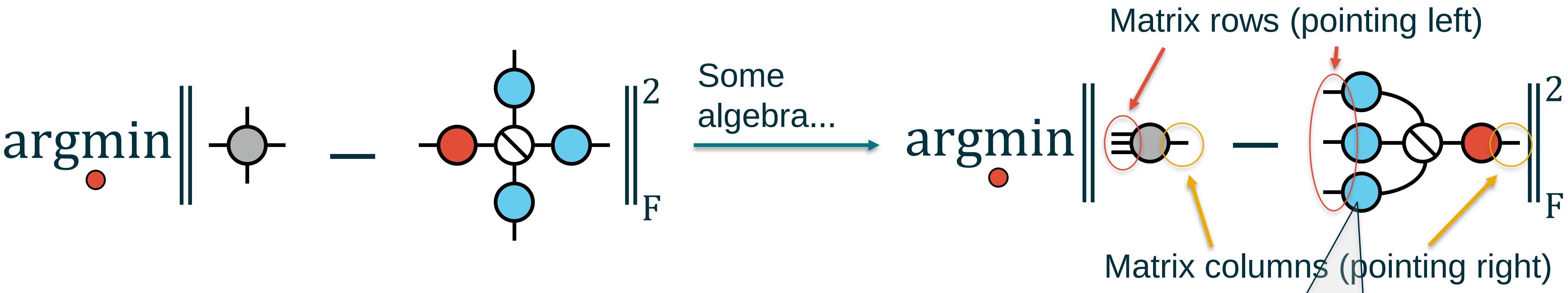


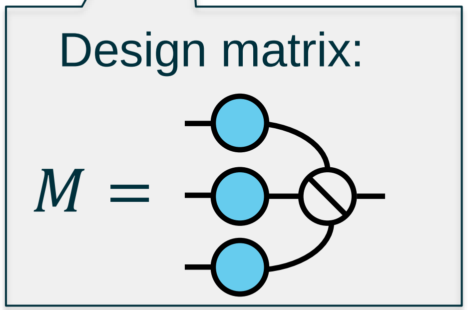
ALS instead updates one thing at a time

→ A linear least squares problem

ALS works for any tensor network

- Always results in linear least squares problem with structured design matrix

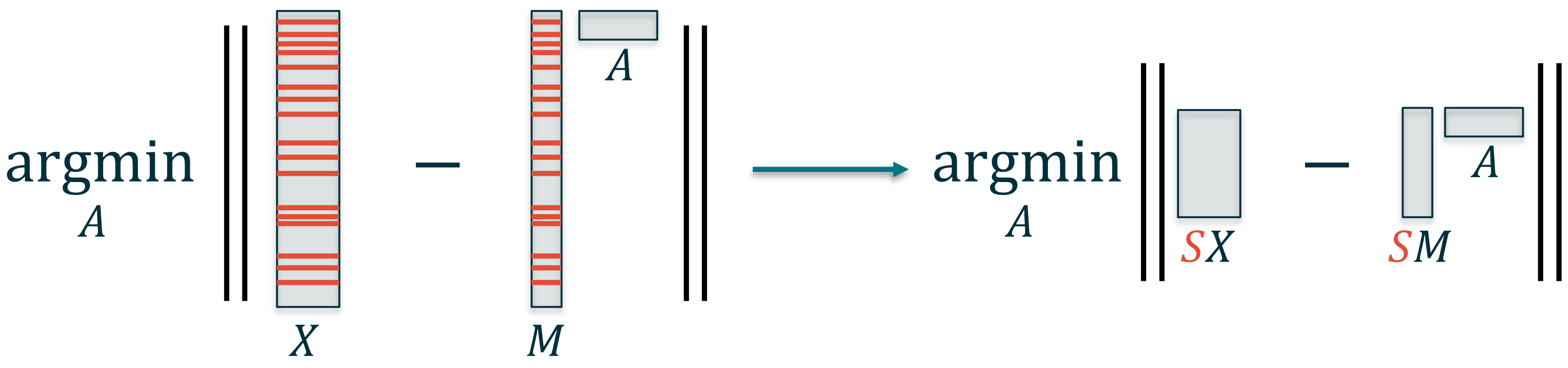




- Other options available (e.g., gradient descent), but ALS is widespread

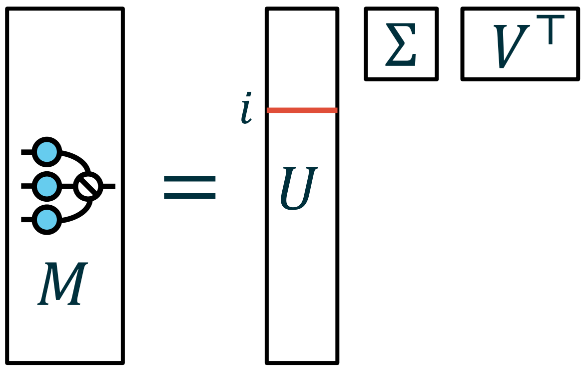
Sampling for overdetermined least squares

- Trouble for large tensors:
  - If data tensor is N-way with each dimension I (i.e., $I \times \cdots \times I$), then M has up to I^{N-1} rows.
  - Cost of solving least squares problem scales at least like I^{N-1}
- Popular idea from Randomized Numerical Linear Algebra (RandNLA): Randomly sample some of the equations!



Leverage score sampling distribution

- Let $U\Sigma V^\top = M$ be thin SVD for M.



- Draw row i with probability $p_i = \|U_{i:}\|_2^2/\mathrm{rank}(M)$.
- Sampling according to $(p_i)_i$ with replacement results in strong guarantees [Drineas et al., 2006, 2008, 2011; Larsen & Kolda, 2022].

Guarantees for leverage score sampling

- Let $U\Sigma V^\top = M$ be thin SVD for M.
- Define a distribution on the rows of M via $p_i = \frac{\|U_{i:}\|_2^2}{\mathrm{rank}(M)}$.
- If rows are sampled iid according to $(p_i)_i$, then with probability at least $1-\delta$
  - $\tilde{A} = \mathrm{argmin}_A \|SX - SMA\|_\mathrm{F}$ satisfies

$$\|X - M\tilde{A}\|_\mathrm{F} \leq (1+\varepsilon)\min_A \|X - MA\|,$$

  - provided enough samples (which depends on δ and ε) are drawn.
- Treating δ and ε as fixed, $O(R\log(R))$ samples are enough where $R = \mathrm{rank}(M)$.
- Upshot: Sampling can yield input sublinear per iteration cost in ALS
  - (i.e., cost is $o(\text{number of entires in } X)$).

Several recent works leverage sampling in ALS for tensor decomposition

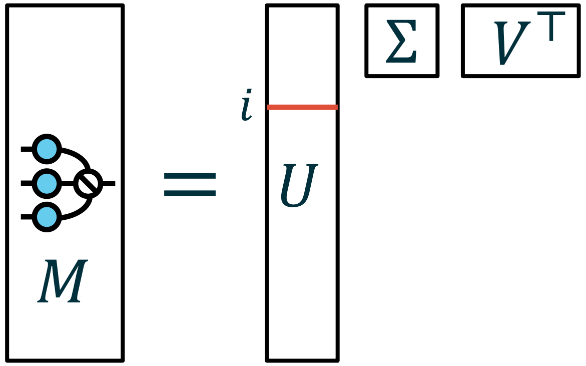
Paper	Tensor decomposition(s)	Exact leverage score distribution?
Cheng et al. [NeurIPS, 2016]	CP	Approximate
Larsen & Kolda [SIMAX 43(3), 2022]	CP	Approximate
M. & Becker [ICML, 2021]	Tensor ring	Approximate
M. [ICML, 2022]	CP, Tensor ring	Approximate
Fahrbach et al. [arXiv:2209.04876, 2022]	Tucker (regularized)	Exact
M. et al. [arXiv:2210.03828, 2022]	Any tensor network decomposition	Exact
Bharadwaj et al. [NeurIPS, 2023] Next week!	CP	Exact

This talk!

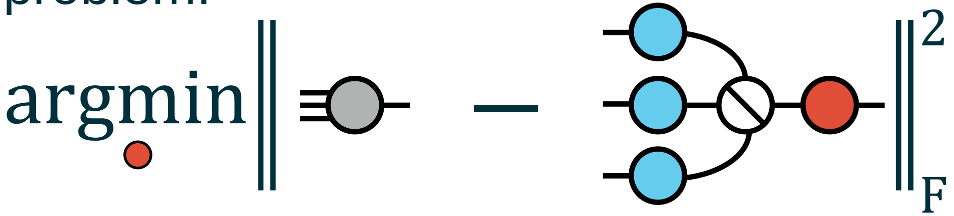
The challenge with leverage score sampling

- Recall $p_i = \frac{\|U_{i:}\|_2^2}{\text{rank}(M)}$, where $M = U\Sigma V^\top$ is a thin SVD
  - Computing U is as expensive as solving linear system (e.g., $I^{N-1}R^2$ for rank-R CPD)
  - Storing (p_i) requires as many numbers as there are rows in M (e.g., I^{N-1})
- Want: Sample from (p_i) while avoiding both

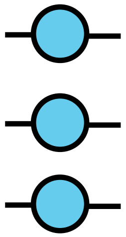
Previous works use a product of simpler distributions



- Recall problem:



- Cheng et al. [NeurIPS, 2016], Larsen & Kolda [SIMAX 43(3), 2022]: Sample according to leverage scores of each factor matrix



3 smaller leverage scores distributions
$\{p^{(1)}_{i_1}\}, \{p^{(2)}_{i_2}\}, \{p^{(3)}_{i_3}\}$

Draw an index from each:
$i_1 \sim \{p^{(1)}_{i_1}\}$,
$i_2 \sim \{p^{(2)}_{i_2}\}$,
$i_3 \sim \{p^{(3)}_{i_3}\}$

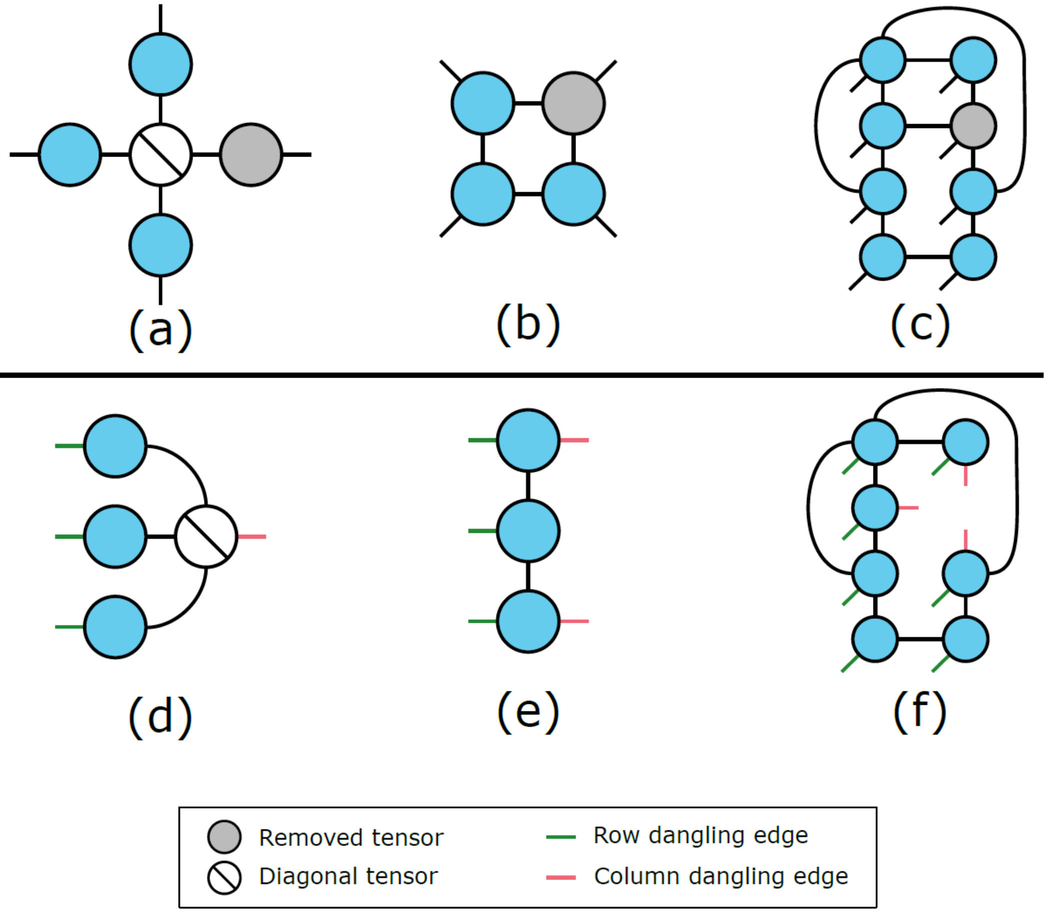
Compute corresponding "big" index
$i = (i_1, i_2, i_3)$

Pros:
- Cheap to compute smaller distributions.
- Very fast to sample.

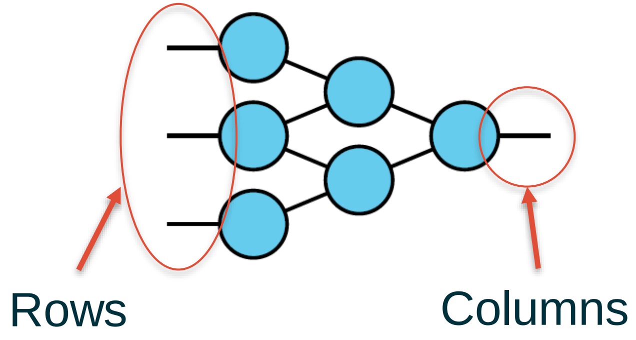
Cons:
- Not sampling from exact leverage score distribution.
- R^N dependence in sampling complexity.

We leverage fact that M has tensor network structure for efficient computation

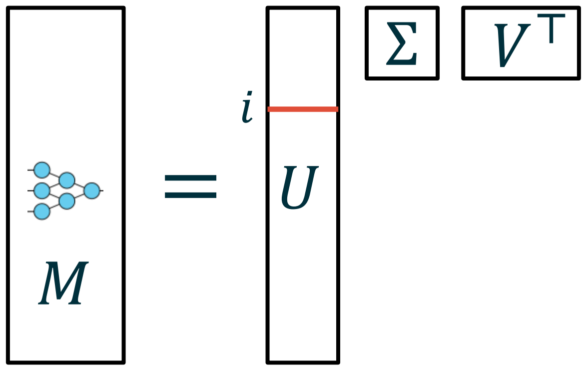
	CP	Tensor ring	"Tiger decomposition" [Li & Sun, ICML 2022]
Tensor network being fitted	(a)	(b)	(c)
Structure of design matrix M	(d)	(e)	(f)



For sake of illustration, let's assume M is



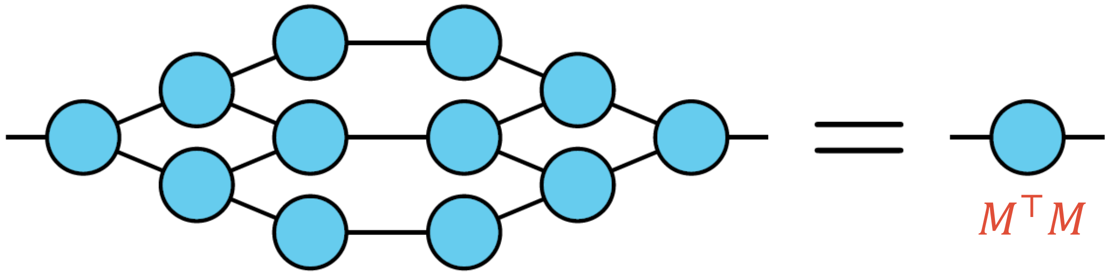
Compute Gram matrix $M^\top M$ and its pseudoinverse



- Recall formula:

$$p_i \propto \|U_{i:}\|_2^2 = e_i^\top M (M^\top M)^+ M^\top e_i$$

- Step 1: Compute $M^\top M$:

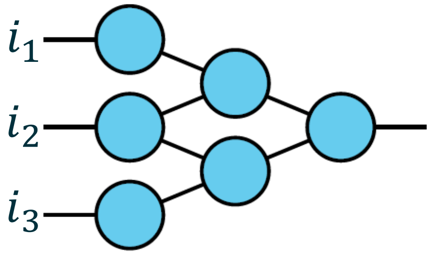


- Can be done efficiently* via tensor contraction
- Compute pseudoinverse of Gram matrix: $\Phi \coloneqq (M^\top M)^+$. This is affordable*.

*For "reasonable" tensor networks this is typically the case. Not hard to cook up a counter-examples though.

Sample rows by sequentially sampling subindices

- Sampling row i ⇔ sampling subindices (i_1, i_2, i_3)



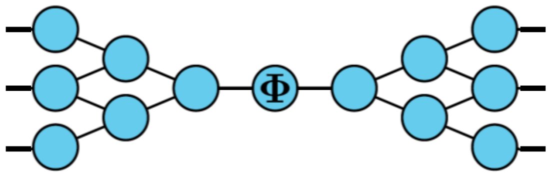
- Strategy:
  – Sample i_1
  – Sample i_2 *conditionally* on realization of i_1
  – Sample i_3 *conditionally* on realization of i_1, i_2

Sampling first index i_1

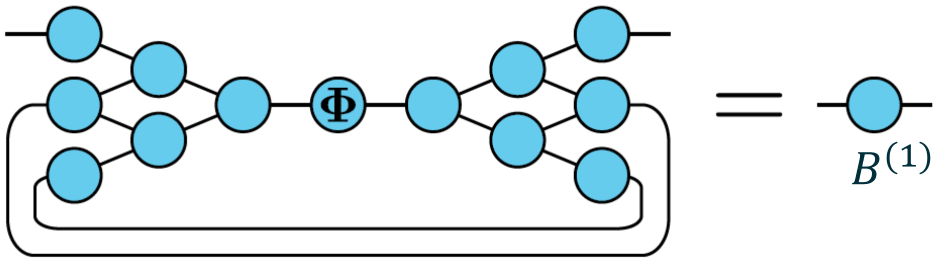
- Recall formula:

$$p_{(i_1,i_2,i_3)} = p_i \propto \|U_{i:}\|_2^2 = e_i^\top \, M\Phi M^\top \, e_i$$

$M\Phi M^\top =$



- Distribution for i_1 is: $\Pr(i_1) = \sum_{i_2} \sum_{i_3} p_{(i_1,i_2,i_3)}$
- Can be computed efficiently via contraction:

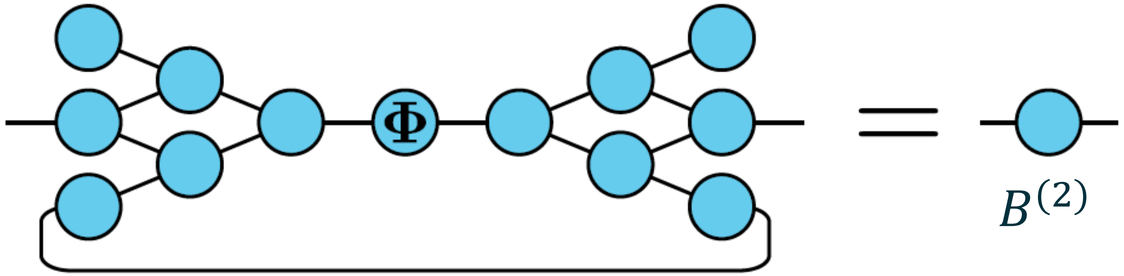


Now $B^{(1)}$ has probabilities along diagonal:
$\Pr(i_1) = B^{(1)}_{i_1 i_1}$

- Sample i_1 according to $(\Pr(i_1))_{i_1}$

Sampling subsequent indices $i_2, \dots, i_N$

- Distribution for i_2 conditionally on i_1 is: $\Pr(i_2 \mid i_1) = \sum_{i_3} p_{(i_1,i_2,i_3)}$
- Can be computed efficiently via contraction:



Now $B^{(2)}$ has probabilities along diagonal:
$\Pr(i_2 \mid i_1) = B^{(2)}_{i_2 i_2}$

- Finally, the distribution for i_3 conditionally on i_1, i_2 is $\Pr(i_3 \mid i_1, i_2) = p_{(i_1,i_2,i_3)}$
- This directly generalizes to more indices and other tensor formats

Improvements to computational complexity for CP decomposition

Computing rank R CP decomposition of an N-way tensor X of size $I \times \cdots \times I$
#it is number of ALS iterations

Paper	Method	Complexity*
E.g., Kolda & Bader [SIREV 51(3), 2009]	CP-ALS	$\#\text{it} \cdot N(N+I)I^{N-1}R$
Cheng et al. [NeurIPS, 2016]	SPALS	$I^N + \#\text{it} \cdot N(N+1)R^{N+1}$
Larsen & Kolda [SIMAX 43(3), 2022]	CP-ARLS-LEV	$\#\text{it} \cdot N(R+I)R^N$
Malik [ICML, 2022]	CP-ALS-ES	$\#\text{it} \cdot N^2R^3(R+NI)$
Malik et al. [arXiv:2210.03828, 2022]	TNS-CP	$\#\text{it} \cdot N^3IR^3$
Bharadwaj et al. [NeurIPS, 2023] Next week!	STS-CP	$\#\text{it} \cdot (N^2R^3 \log I + NIR^2)$

*Leading order complexity. Ignores log factors and treats accuracy (ε) and failure probability (δ) as constants. Number of iterations #it may differ between methods.

Improvements to computational complexity for tensor ring decomposition

Computing rank $(R, \ldots, R)$ tensor ring decomposition of an N-way tensor X of size $I \times \cdots \times I$
#it is number of ALS iterations

Paper	Method	Complexity*
Zhao et al. [arXiv:1606.05535]	TR-ALS	$\#\text{it} \cdot N I^N R^2$
Yuan et al. [ICASSP, 2019]	rTR-ALS	$N I^N K + \#\text{it} \cdot N K^N R^2$
Zhao et al. [arXiv:1606.05535]	TR-SVD	$I^{N+1} + I^N R^3$
Ahmadi-Asl et al. [Mach learn: sci technol, 2020]	TR-SVD-Rand	$I^N R^2$
Malik & Becker [ICML, 2021]	TR-ALS-Sampled	$\#\text{it} \cdot N I R^{2N+2}$
Malik [ICML, 2022]	TR-ALS-ES	$\#\text{it} \cdot N^3 R^8 (R + I)$
Malik et al. [arXiv:2210.03828, 2022]	TNS-TR	$\#\text{it} \cdot N^3 I R^8$

*Leading order complexity. Ignores log factors and treats accuracy (ε) and failure probability (δ) as constants. Number of iterations #it may differ between methods.

Future directions

- Expand to other decompositions
- High-performance codes (shared or distributed memory)
- Recent tensor collaborators:

Vivek Bharadwaj
UC Berkeley

Riley Murray
Sandia National Labs

Beheshteh Rakhshan
Mila

Guillaume Rabusseau
Mila

Laura Grigori
INRIA

Aydın Buluç
LBNL and UC Berkeley

James Demmel
UC Berkeley

- Please contact me with questions:
  - Osman Malik
  - oamalik@lbl.gov
  - https://osmanmalik.github.io/

Thank you!

Preprint at:

References

- S. Ahmadi-Asl, A. Cichocki, A. H. Phan, M. G. Asante-Mensah, F. Mousavi, I. Oseledets, and T. Tanaka. *Randomized algorithms for fast computation of low-rank tensor ring model.* Machine Learning: Science and Technology, 2020.
- V. Bharadwaj, O. A. Malik, R. Murray, A. Buluç, and J. Demmel. *Distributed-memory randomized algorithms for sparse tensor CP decomposition*. Preprint arXiv:2210.05105, 2022.
- V. Bharadwaj, O. A. Malik, R. Murray, L. Grigori, A. Buluç, J. Demmel. Fast Exact Leverage Score Sampling from Khatri-Rao Products with Applications to Tensor Decomposition. NeurIPS, 2023.
- D. Cheng, R. Peng, Y. Liu, and I. Perros. *SPALS: Fast alternating least squares via implicit leverage scores sampling*. Advances In Neural Information Processing Systems, pages 721-729, 2016.
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. *Sampling algorithms for ℓ_2 regression and applications*. ACM-SIAM Symposium on Discrete Algorithms, pages 1127-1136, 2006.
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. *Relative-error CUR matrix decompositions*. SIAM Journal on Matrix Analysis and Applications, 30(2):844-881, 2008.
- P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlos. *Faster least squares approximation*. Numerische Mathematik, 117(2):219-249, 2011.
- M. Fahrbach, T. Fu, and M. Ghadiri. *Subquadratic Kronecker regression with applications to tensor decomposition*. Preprint arXiv:2209.04876, 2022.
- J. Kepner, M. Jones, D. Andersen, et al. *Spatial temporal analysis of 40,000,000,000,000 internet darkspace packets*. IEEE HPEC, 2021.
- T. G. Kolda and B. W. Bader. *Tensor decompositions and applications*. SIAM Review, 51(3): 455-500, Aug. 2009.
- B. W. Larsen and T. G. Kolda. *Practical leverage-based sampling for low-rank tensor decomposition*. SIAM Journal on Matrix Analysis and Applications, 43(3):1488{1517, 2022.
- C. Li and Z. Sun. *Evolutionary topology search for tensor network decomposition*. International Conference on Machine Learning, volume 119, pages 5947-5957. PMLR, 2020.
- O. A. Malik. *More efficient sampling for tensor decomposition with worst-case guarantees*. International Conference on Machine Learning, volume 162, pages 14887-14917. PMLR, 2022.
- O. A. Malik, V. Bharadwaj, R. Murray. *Sampling-Based Decomposition Algorithms for Arbitrary Tensor Networks.* Preprint arXiv:2210.03828, 2022.
- O. A. Malik and S. Becker. *A sampling-based method for tensor ring decomposition*. International Conference on Machine Learning, volume 139, pages 7400-7411. PMLR, 2021.
- L. Yuan, C. Li, J. Cao, and Q. Zhao. *Randomized tensor ring decomposition and its application to large-scale data reconstruction*. In IEEE International Conference on Acoustics, Speech and Signal Processing, pages 2127-2131. IEEE, 2019.
- Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki. *Tensor ring decomposition*. Preprint arXiv:1606.05535, 2016.`o