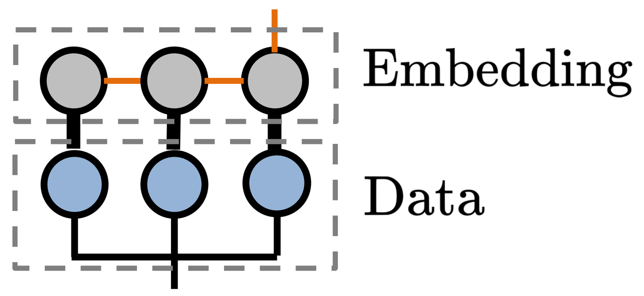
Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs

Linjian Ma and Edgar Solomonik

Department of Computer Science, University of Illinois at Urbana Champaign

Problem: sketching tensor network input data



Problem: Given a tensor network input, $\boldsymbol{x}$, find a linear **Gaussian** tensor network embedding, $\boldsymbol{S}$, such that the embedding is (ϵ, δ)-accurate and

- The number of rows of $\boldsymbol{S}$ (sketch size m) is low
- Asymptotic cost to compute $\boldsymbol{S}\boldsymbol{x}$ is minimized

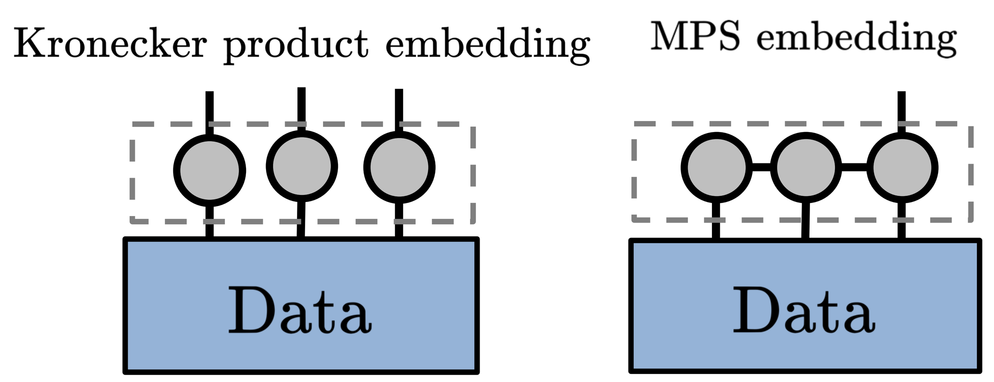
Gaussian tensor network embedding: each element in each tensor is an i.i.d. Gaussian random variable

An (oblivious) embedding $\boldsymbol{S} \in \mathbf{R}^{m \times s}$ is (ϵ, δ)-accurate if[1]

$$\Pr\left[\left|\frac{\|\boldsymbol{S}\boldsymbol{x}\|_2 - \|\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\right| > \epsilon\right] \leq \delta \quad \text{for any } \boldsymbol{x}$$

Previous work:

- Gaussian matrix/Kronecker product embedding[2]: inefficient in computational cost
- Tree embedding (e.g. MPS)[2,3]: efficient for specific data (Kronecker product, MPS), but efficiency unclear for general tensor network data



Assumptions throughout our analysis:

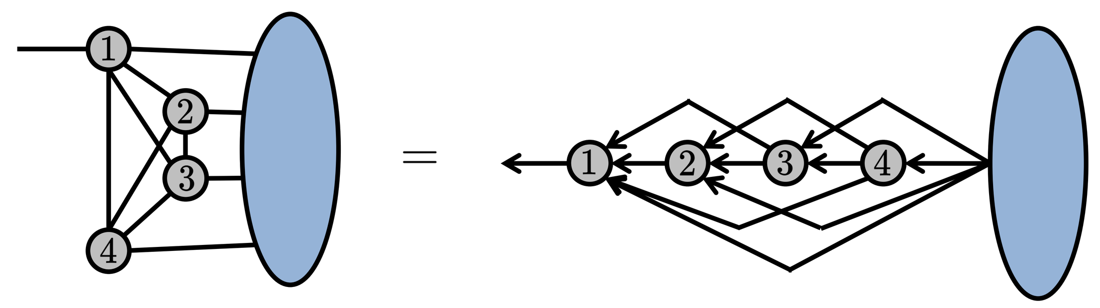
- Classical $O(n^3)$ matmul cost
- Consider embeddings defined on **graphs with no hyperedges**
- Each dimension to be sketched
  - has a **size lower bounded by the sketch size**
  - is only adjacent to one data tensor

References

[1][Woodruff, 2014] [2][Ahle et al, SODA 2020] [3][Rakhshan and Rabussea, AISTATS 2020] [4][Malik, ICML 2022] [5][Daas et al, 2021] [6][Pham and Pagh, KDD 2013]

Sufficient condition for (ϵ, δ)-accurate embedding

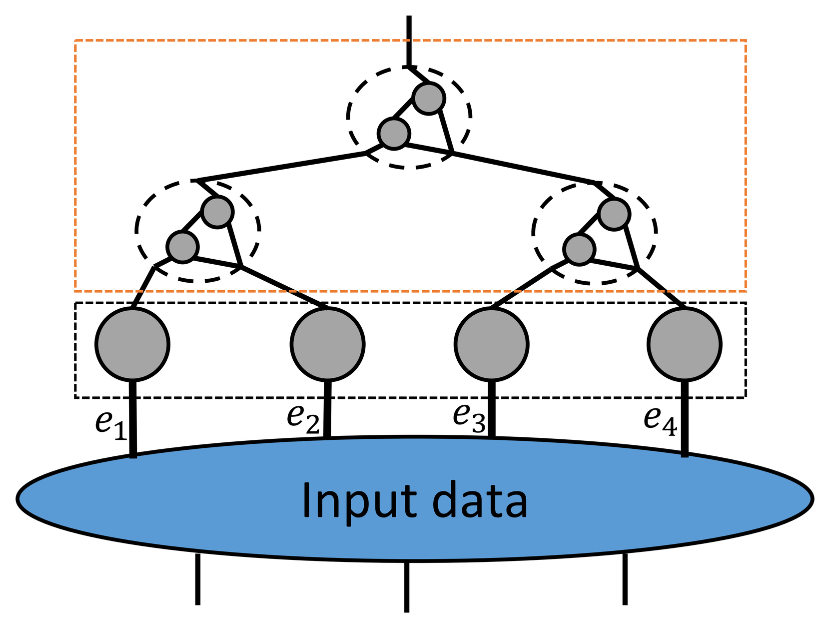
The embedding with a graph structure $G = (V, E, w)$ is accurate if there exists a linear ordering of V such that in its induced DAG, the weighted sum of out-going edges adjacent to each $v \in V$ is $\Omega(N \log(1/\delta)/\epsilon^2)$



This sufficient condition yields a new sketch size bound for MPS embedding

	sketch size upper bound for (ϵ, δ)-accurate embedding
This work	$O(N \log(1/\delta)/\epsilon^2)$
Previous results[3]	$O((1+2/R)^N \log^{2N}(1/\delta)/\epsilon^2)$

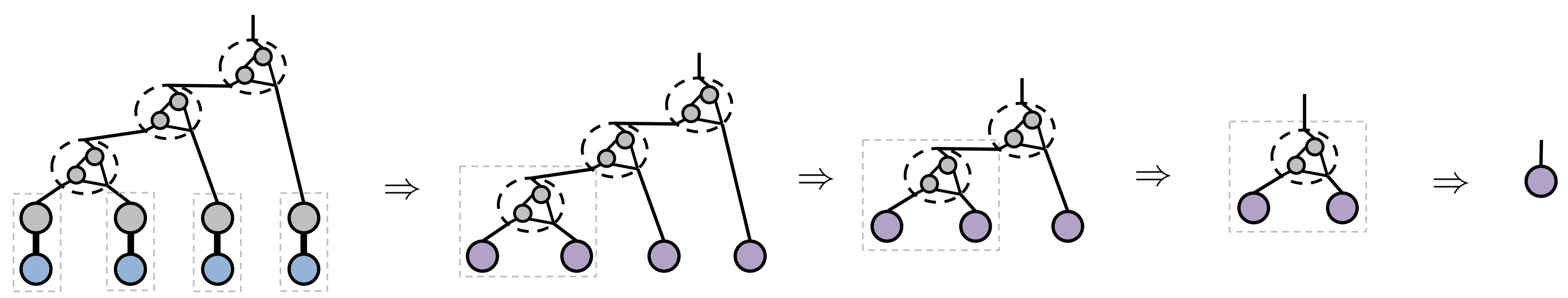
The sketching algorithm with efficient computational cost and sketch size



- The embedding has a **general graph** rather than a tree structure
- Each matrix in the Kronecker product embedding is applied before output data dimensions are merged
- Each vertex in the binary tree part is applied when a pair of output data dimensions are merged

Example: sketching Kronecker product data

Consider contracting an input Kronecker product data from left to the right, the sketching contraction path is as follows



The algorithm reduces cost by up to $O(\sqrt{m})$ compared to using tree embeddings

Analysis of the algorithm

c: asymptotic sketching cost for our algorithm

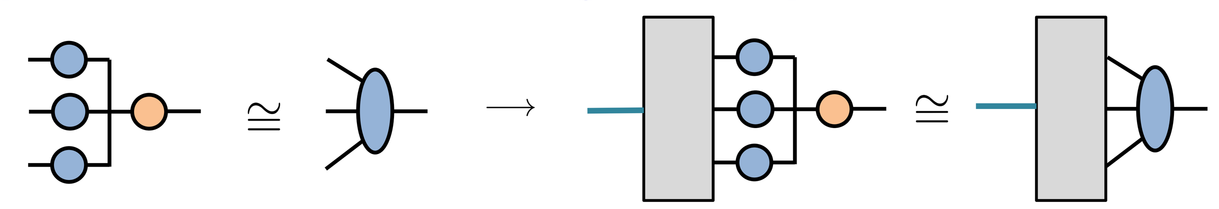
c_{opt}: optimal asymptotic sketching cost under the embedding sufficient condition

m: sketch size

Input data tensor network structure	Optimality of the algorithm
General hypergraph	$c = O(\sqrt{m} \cdot c_{\text{opt}})$
General graph	$c = O(m^{0.375} \cdot c_{\text{opt}})$
Each data tensor has a dimension to be sketched (e.g. Kronecker product, MPS)	$c = c_{\text{opt}}$

Applications

CP decomposition with alternating least squares



- Consider the rank-R CP decomposition of a tensor with order N and dimension size s
- When performing a low-rank decomposition with $s \gg R^{1.5}$, our algorithm is $\Omega(NR)$ times better than recursive leverage score sampling[4]
- Larger preparation cost is needed
- Preparation cost can be reduced via using other sparse embeddings (e.g. Countsketch[1])

Tensor train rounding

- Recently proposed randomized rounding algorithm[5]: using MPS embedding to perform randomized range finder
- Our analysis shows the optimality of this algorithm

Following work

- Generalize the analysis to other embeddings, such as Countsketch[1] and Tensorsketch[6]
- Generalize the analysis to consider other data tensor networks (those without the assumption that each dimension to be sketched has large size)