Tensor Factorizations and Random Embeddings

Roland Wirth
Facility for Rare Isotope Beams, Michigan State University
Tensors

Tensors: multi-indexed arrays

\[ T_{i_1i_2\ldots i_n} \]

Number of indices \( n \): mode.

Tensors are at the heart of many-body methods:

- Hamiltonian matrix elements \( \langle pq|H|rs \rangle \equiv H_{pqrs} \)
- Cluster amplitudes \( t_{ia}, t_{ijab}, \ldots \)
- Magnus operator \( \Omega_{pq}, \Omega_{pqrs}, \ldots \)

Methods like Coupled Cluster and In-Medium SRG can be seen as Tensor networks, e.g.

\[
\frac{dE}{ds} = \frac{1}{4} \sum_{abcd, klm} \eta_{ab}^{kl} \Gamma_{am}^{kl} \lambda_{cdm}^{bk} + \cdots = \text{Diagram}
\]
Tensors define computational requirements:

- **Storage**: need to store $N^n$ numbers for mode-$n$.
- **Evaluation**: number and length of sums given by mode and index length, e.g. $N^7$ for energy flow equation.

High-precision calculations have to use three-body objects

⇒ Mode-6 tensors
⇒ $N^6$ storage and (naively) $N^9$ evaluation cost

Can we improve the scaling through a controlled approximation?
Tensor Factorization

Exploit structures in the tensors

- NN interactions have much less than $N^4$ coupling constants
- Many cluster amplitudes are small or correlated

How? Factorize into given **Tensor format**

- Simple: Canonical polyadic decomposition (CPD)

$$\tilde{T}_{pqrs} = \sum_{\alpha=1}^{r_{CPD}} V_p^{(1,\alpha)} V_q^{(2,\alpha)} V_r^{(3,\alpha)} V_s^{(4,\alpha)}$$

- Efficient: Tensor hyper-contraction (THC)

$$\tilde{T}_{pqrs} = \sum_{\alpha,\beta=1}^{r_{THC}} V_p^{(1,\alpha)} V_q^{(2,\alpha)} C_{\alpha\beta} V_r^{(3,\beta)} V_s^{(4,\beta)}$$
First step: try to factorize Hamiltonian

\[ \Delta H^J \]

\[ \log_N(r_{THC}) \]

A. Tichai et al., PRC 99, 034320 (2019)
To compute a CPD: Minimize $\|T - \tilde{T}\|_F$ via alternating least squares

1. Choose random factors
2. Fix factors 2, ..., $n$, solve least-squares problem for first factor
3. Repeat for remaining factors
4. Check factorization error, repeat if necessary

To compute THC:

1. Perform eigenvalue decomposition
2. Discard small eigenvalues $|\lambda_i| < \epsilon$
3. Perform CPD on eigenvectors

ALS is computationally intensive & slowly convergent.
Random Embeddings

- Assume tensor has low-rank CPD
  \[ \tilde{T}_{pqrs} = \sum_{\alpha=1}^{r_{CPD}} v_p^{(1,\alpha)} v_q^{(2,\alpha)} v_r^{(3,\alpha)} v_s^{(4,\alpha)} \]

- Evaluation of tensor network = Calculating inner products
- Preserving matrix elements not essential, preserving inner products is

**Johnson-Lindenstrauss lemma**

There exists a random mapping to a $k$-dimensional subspace that preserves distances between a set of $m$ vectors to a relative error of $\varepsilon$ with high probability. The dimension is $k = \Theta(\log(m)/\varepsilon^2)$.

Mapping can be found by drawing random Gaussian matrices.
Second-order perturbation theory

\[ \frac{\tilde{E}(2)}{E(2)} \]

\[ \tilde{N}/N \]

\[ ^{16}\text{O} \]

\[ e_{\text{max}} = 4 \]

\[ J = 0 \]

A. Zare, priv. comm. (mod.)
Random Projection: Advantages & Challenges

- Extremely simple
- Computationally cheap
- Low storage requirements

- Random: How to decide to keep subspace or reroll?
- Does not simplify evaluation
- Scaling not as favorable compared to THC?

Hybrid approach? Project, then factorize?
Acknowledgments

■ Thanks to my group and collaborators
  ■ T. Duguet, J. Ripoche, A. Tichai
    CEA Saclay
  ■ H. Hergert, J. M. Yao
    FRIB/NSCL, Michigan State University
  ■ M. Iwen, A. Zare
    Dept. of Mathematics / CMSE, Michigan State University

■ Thank you for your attention!