DRSTP SPCTM school 2018
Introduction to tensor networks for QMBS

Exercise 2

Problem 2.1 Repetition

Take some time to think about the following questions, to check if you have understood some of the key ideas.

1. What does the area law of the entanglement entropy mean?
2. What is the Schmidt decomposition?
3. How can you truncate a bond in an MPS?
4. Can an MPS ansatz be used for any state? When is it particularly useful?
5. What's the maximal entanglement entropy an MPS with bond dimension D can reproduce?
6. What's the relation between the Schmidt coefficients and the eigenvalues of the reduced density matrix?
7. When do you obtain a pure state for the reduced density matrix of some region A? What's the entanglement entropy in this case?
8. What's the difference between matrix product states and matrix product operators?
9. What is a left-, right-, and mixed-canonical MPS?
10. When truncating a bond in an MPS why is it important to be in mixed-canonical form?
11. What is the computational cost when multiplying two tensors?
12. What is the leading computational cost in D when computing the norm $\langle\Psi|\Psi\rangle$ of an arbitrary MPS with bond dimension D? What is the leading computational cost when computing the expectation value of a local observable with MPS?

Problem 2.2 Computational cost to contract a tensor network

1. Consider the tensor networks (a)-(c) in the following figure which we contract by a pairwise multiplication. We will assume that each index (leg) has a dimension D. What is the leading computational cost as a function of D to contract these networks, i.e. of which order $O(D^k)$ is the contraction?

2. Define these tensors in Matlab and use ncon to contract these networks. Use bond dimensions $D = 2$ for the open legs and use a few different bond dimensions for the connecting legs (e.g. 3, 4, 5, 6, not too large otherwise it will become too expensive). If you do a mistake in the contraction (i.e. if the index dimensions do not match) you will then obtain an error message from ncon.

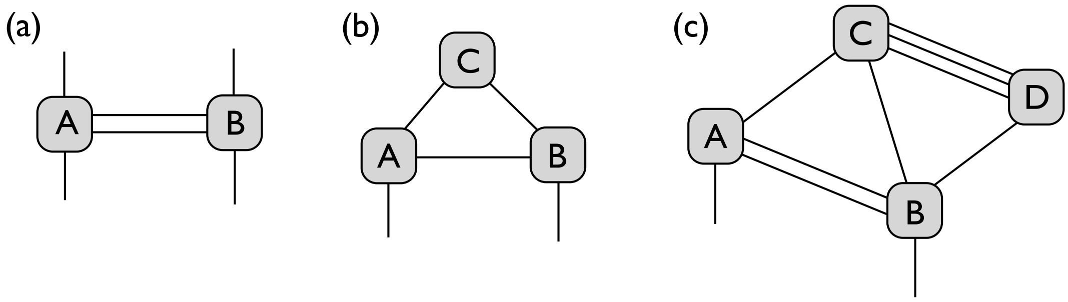

Figure 1: Example tensor networks to be contracted.

Problem 2.3 Drawing tensor network diagrams

Consider an MPS representing a state $|\Psi\rangle$ of a 6-site system. Represent the following formulas as a tensor network (draw it on paper):

1. $\langle\Psi|\hat{O}_{45}|\Psi\rangle$, where $\hat{O}_{45}$ is a 2-site operator acting on sites 4 and 5.

2. $\hat{\rho}_{456} = \mathrm{Tr}_{123}\hat{\rho}$, where Tr_{123} is the trace over sites 1, 2, 3, and $\hat{\rho}_{456}$ is the reduced density matrix on sites 4, 5, 6, and $\hat{\rho} = |\Psi\rangle\langle\Psi|$ the density operator (pure state).

3. $\langle\Psi|\hat{O}_2\hat{O}_4|\Psi\rangle$, assuming that the MPS is in a mixed canonical form with respect to site 2. Draw how the network simplifies in this case. The operators $\hat{O}_2$ and $\hat{O}_4$ are one-site operators acting on sites 2 and 4, respectively.

Problem 2.4 Imaginary time evolution algorithm with MPS

The goal of this exercise is to implement an imaginary time evolution algorithm starting from a random initial MPS in order to obtain the ground state of a local Hamiltonian. As an example we will consider the $S = 1/2$ Heisenberg model of a $N_s = 16$ site system with open boundary conditions and study the energy as a function of the bond dimension D of the MPS.

1. Write a piece of code which initializes a random (unnormalized) MPS. Let's stick to the convention that the order of indices of each MPS tensor is: left-down-right. Hints: You can store the individual MPS tensors in a Matlab cell array. The code simplifies later if you use rank-3 tensors also at the left and right boundary, i.e. instead of using a $2 \times D$ tensor at the boundary you can use a $1 \times 2 \times D$ tensor on the left boundary (and a $D \times 2 \times 1$ tensor at the right boundary).

2. Write a function which contracts two tensors with a two-body operator (using ncon) and which performs an SVD (using tensorsvd), keeping D singular values, with the option to absorb the singular values either on the left or on the right. Be careful that you permute your resulting tensors after the SVD accordingly. Normalize the MPS tensor containing the singular values by the norm. Note that if the wave function is in its Schmidt decomposed form then the norm is simply computed by $norm = \sqrt{Tr(s * s)}$.

3. Construct the imaginary time evolution 2-body operator $\exp(-\tau \hat{H}_{i,i+1})$. You can do this by taking the matrix exponential of the 2-site Hamiltonian matrix using the *expm* function in Matlab, and reshaping it into a $2 \times 2 \times 2 \times 2$ tensor. Be careful with the labelling of the legs (the correct order of the legs is 1: upper left, 2:upper right, 3: lower left, 4: lower right).

4. Next we use this function to apply the imaginary time evolution operator $\exp(-\tau \hat{H}_{i,i+1})$ successively between all nearest-neighbor pairs. Write a function to perform a sweep from left-to-right, and then back from right-to-left. This will correspond to two time steps τ (second order Trotter-Suzuki decomposition). Note that you can keep the MPS in the correct mixed-canonical form on-the-fly.

5. Call this function $M/2$ times, where M is the maximal number of time steps (i.e. $\beta = M * \tau$).

6. Now write a function to evaluate the energy by summing up all the 2-body Hamiltonian terms (i.e. without constructing the MPO). You can do this in the following way: compute first the energy of a left-canonical MPS by evaluating the Hamiltonian term first on sites 1 and 2. Then bring the MPS into mixed canonical form with respect to site 2. Then evaluate the Hamiltonian term on sites 2 and 3. Then bring the MPS into mixed-canonical form with respect to site 3, etc., until you have reached the right boundary and evaluated all Hamiltonian terms. The energy then simply corresponds to the sum of all terms.

7. Run simulations for different D's (e.g. between 2...20) using a time step $\tau = 0.1$ and observe how the energy convergences to the exact ground state energy $-6.91173714558...$ with increasing D. Check also the effect of changing the time step τ. In each simulation you can recompute the energy after, e.g. 50 steps, and stop the simulation if the energy change is below a certain tolerance (e.g. 10^{-6}). To

speed up things, you can use an MPS with a smaller bond dimension as an initial state for a simulations at larger bond dimension.