[credit for course materials: Prof. Jan von Delft]

1. Overlaps and normalization

Consider overlap of 2-site MPS:

introduce

reorder

Ket:

Use diagrammatic rules to keep track of contraction patterns:

Bra:

We accommodated complex conjugation via Hermitian conjugation and index transposition:

This scheme switches upper and lower indices -> inverts all arrows in diagram.

Note that in diagram the vertex is left, right, whereas on ,

sits left, right.

It will simplify the structure of diagrams representing overlaps.

Generalization to many-site MPS:

Square brackets indicate that each site has a different A matrix.

We can use shorthand notation and schematic:

Recipe for ket formula: as chain grows, attach new matrices on the right (in the same order as vertices in diagram) resulting in MPS.

Bra:

We rewrite using Hermitian conjugates, change schematic by transposing indices and inverting arrows. To recover MPS structure, order Hermitian conjugate matrices to appear in order opposite to vertex order in diagram. Recipe for bra formula: as chain grows, attach new matrices on the left, *opposite to vertex order in diagram.

Now consider overlap between two MPS:

Exercise: derive result algebraically.

Contraction order matters! If we perform matrix multiplication first, for fixed , and then sum over , we get terms, each of which is a product of matrices. Exponentially costly!

Calculation is tractable if we rearrange summations:

Diagrammatic depiction: ‘closing zipper’ from left to right

The set of two-leg tensors can be computed iteratively:

Initialization:

Iteration step:

Final answer:

Cost estimate (assume all A’s are):

One iteration:

Total cost:

Remark: a similar iteration scheme can be used to ‘close zipper’ from right to left’:

Initialization and iteration step:

Normalization: Try above scheme with

Left-normalization

A 3-leg tensor is called 'left-normalized' if it satisfies

Graphical notation:

When all A's are left-normalized, closing the zipper left-to-right is easy, since all reduce to identity matrices:

Hence:

Left-normalized states are automatically normalized to unity.

<u>Right-normalization</u>

So far we have viewed an MPS as being built up from left to right, hence used right-pointing arrows on ket diagram. Sometimes it is useful to build it up from right to left, running left-pointing arrows.

Building blocks:

Ket:

Bra:

Iterating, we obtain kets and bras of the form

A 3-leg tensor is called right-normalized if it satisfies

Graphical notation for right-normalization:

When all A’s are right-normalized, closing zipper right-to-left is easy

Conclusion: MPS built purely from left-normalized or purely from right-normalized are automatically normalized to 1.

2. Various canonical MPS forms

Left-canonical (lc-) MPS

Right-canonical (rc-) MPS:

Site-canonical (sc-) MPS:

Bond-canonical (bc-) (or mixed) MPS:

How to bring an arbitrary MPS into one of these forms?

Transforming to left-normalized form

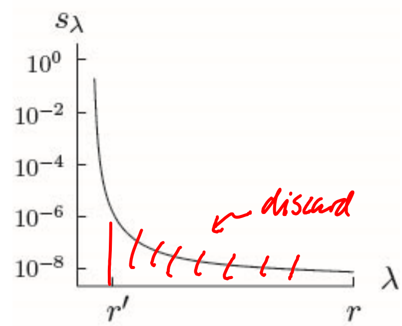
Given:

Goal: left-normalize

Strategy: take a pair of adjacent tensors, , and use SVD:

Left-Normalization assured by:

Truncation can be performed by discarding some of the smallest singular values(remains left-normalized!)



Note: if we don't need to truncate we can use QR (cheaper).

By iterating, starting from , we left-normalize

To left-normalize the <u>entire</u> MPS, choose

As last step, left-normalize last site using SVD on final :

lc-form:

The final singular value, , determines normalization:

<u>Transforming to right-normalized form</u>

Given:

Goal: right normalize

Strategy: take a pair of adjacent tensors, , and use SVD:

Here, right-normalization is assured by:

Starting from , move left to

To right-normalize entire chain, choose

and as before determines normalization.

Transforming to site-canonical form

Left-normalize states starting from site .

Then right-normalize states starting from site .

Result:

We get an orthonormal set from the states (**Exercise in HW**):

This basis is a 'local site basis' for site . Its dimension

is usually the dimension of the full Hilbert space.

<u>Transforming to bond-canonical form</u>**:**

Start from e.g. sc-form, use SVD for , combine either

(1)

(2)

Let's try option 1 first.

We get an orthonormal set of states again:

This basis is the 'local bond basis' for bond . It has dimension

where = dimension of singular matrix .

Now try option 2:

This basis is the 'local bond basis' for bond . It has dimension

where = dimension of singular matrix

3. Matrix elements and expectation values

One-site operator

E.g. for spin 1/2:

Consider two states in site-canonical form for site :

Matrix element:

Close zipper from left and right:

Consider expectation value:

Left-normalization means

Right-normalization means

Hence,

<u>Two-site operator</u> (e.g. for spin chain Hamiltonian term)

Matrix elements:

4. Schmidt decomposition

Consider a quantum system composed of two subsystems

with dimensions and orthonormal bases

To be specific, think of physical basis:

General form of pure state on joint set

Density matrix:

Reduced density matrix of subsystem :

with

Analogously, RDM of subsystem :

Singular value decomposition:

Use SVD to find basis for that diagonalizes

SVD of

With indices:

Hence

where

are orthonormal sets of states for .

Orthonormality is guaranteed by

Restrict to the non-zero singular values to get a

Schmidt decomposition:

Classical state:

Entangled state:

r is known as **Schmidt number**

In this representation, RDMs are diagonal:

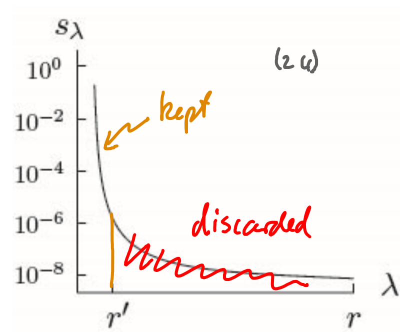
Entanglement entropy:

How can one approximate

Define truncated state using singular values:

(If a normalized state is needed, rescale)

Truncation error: sum of squares of discarded singular values



Useful to obtain 'cheap' representation of if singular values decay rapidly.