

Contents

1	DRMG applied to two-dimensional classical lattice models	1
1.1	Abstract	1
1.2	Partition functions of classical lattices	1
1.3	Transfer matrices of lattice models	2
1.3.1	1D Ising model	2
1.3.1.1	Fixed boundary conditions	3
1.3.2	2D Ising model	4
1.4	Partition function of the 2D Ising model as a tensor network	5
1.4.1	Tensor network of the partition function of a system of four spins	6
1.4.2	Thermodynamic limit	6
1.4.3	The transfer matrix as a tensor network	8
1.5	Transfer matrix renormalization group	8
1.5.1	The infinite system algorithm for the transfer matrix	10
1.5.2	Physical interpretation of the reduced density matrix	11
1.6	Corner transfer matrix renormalization group	13
1.6.1	Corner transfer matrices	13
1.6.2	Corner transfer matrix as a tensor network	14
1.6.3	Corner transfer matrix renormalization group method	14
1.7	Calculation of observable quantities	15
1.8	Validity of approximation	15
1.9	Equivalence to variational approximation in the space of ma- trix product states.	17
A	Introduction to tensor networks	18
A.1	Tensors, or multidimensional arrays	18
A.2	Tensor contraction	19
A.3	Tensor networks	19
A.3.1	Graphical notation	20
A.3.2	Reshaping tensors	21
A.3.3	Computational complexity of contraction	21
	Bibliography	22

1

DRMG applied to two-dimensional classical lattice models

1.1 Abstract

This chapter explains how to apply to ideas of the density matrix renormalization group to two-dimensional classical lattices. The Ising model is used throughout.

First, we explain the transfer-matrix formulation for classical partition functions.

Then, we show how to renormalize the transfer matrix using DMRG. This was first done by Nishino [1]. To make notation easier, we redefine the transfer matrix in terms of a tensor network.

Then, we explain the corner transfer matrix renormalization group. This method, first introduced by Nishino and Okunishi [2], combines ideas from Baxter [baxter1982exactly, 3, 4] and White [5] to significantly speed up the renormalization of the transfer matrix.

1.2 Partition functions of classical lattices

The central quantity in equilibrium statistical mechanics is the partition function Z , which, for a discrete system such as a lattice, is defined as

$$Z = \sum_s \exp(-\beta H(s)), \quad (1.1)$$

where the sum is over all microstates s , H is the energy function, and $\beta = T^{-1}$ the inverse temperature.

1.3 Transfer matrices of lattice models

Transfer matrices are used to re-express the partition function of classical lattice systems, allowing them to be solved exactly or approximated.

We will introduce the transfer matrix in the context of the 1D classical Ising model, first introduced and solved using the transfer matrix method by Ising [6] in his PhD thesis.

1.3.1 1D Ising model

talk a bit about model (magnetism etc).

Somewhere, I should introduce – very briefly – how the magnetization and free energy per site are defined. Maybe just calculate them for 1D Ising model.

Consider the 1D zero-field ferromagnetic Ising model [6], defined by the energy function

$$H(\sigma) = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j. \quad (1.2)$$

Here, we sum over nearest neighbors $\langle ij \rangle$ and the spins σ_i take the values ± 1 . $J > 0$.

Assume, for the moment, that the chain consists of N spins, and apply periodic boundary conditions. The partition function of this system is given by

$$Z_N = \sum_{\sigma_1, \dots, \sigma_N \in \{-1, 1\}} \exp(-\beta H(\sigma)) \quad (1.3)$$

Exploiting the local nature of the interaction between spins, we can write

$$Z_N = \sum_{\sigma_1, \dots, \sigma_N \in \{-1, 1\}} \prod_{\langle i, j \rangle} e^{K \sigma_i \sigma_j} \quad (1.4)$$

where we defined $K \equiv \beta J$.

Now, we define the 2×2 matrix

$$T_{\sigma\sigma'} = \exp(K\sigma\sigma'). \quad (1.5)$$

A possible choice of basis is

$$(|\uparrow\rangle = 1, |\downarrow\rangle = -1) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right). \quad (1.6)$$

In terms of this matrix, Z_N is written as

$$Z_N = \sum_{\sigma_1, \dots, \sigma_N} T_{\sigma_1 \sigma_2} \cdots T_{\sigma_N \sigma_1} = \text{Tr } T^N. \quad (1.7)$$

T is called the transfer matrix. In the basis of Equation 1.6, T is written as

$$T = \begin{bmatrix} e^K & e^{-K} \\ e^{-K} & e^{-K} \end{bmatrix}. \quad (1.8)$$

T is, in fact, diagonalizable. So, we can write $T^N = PD^N P^{-1}$, where P consists of the eigenvectors of T , and D has the corresponding eigenvalues on the diagonal. By the cyclic property of the trace, we have

$$Z_N = \lambda_1^N + \lambda_2^N, \quad (1.9)$$

where

$$\begin{aligned} \lambda_1 &= e^K + e^{-K}, \\ \lambda_2 &= e^K - e^{-K}. \end{aligned}$$

Thus, we have reduced the problem of finding the partition function to an eigenvalue problem.

In the thermodynamic limit $N \rightarrow \infty$

$$Z = \lim_{N \rightarrow \infty} \lambda_1^N \quad (1.10)$$

where λ_1 is the non-degenerate largest eigenvalue (in absolute value) of T .

1.3.1.1 Fixed boundary conditions

We may also apply fixed boundary conditions. The partition function is then written as

$$Z_N = \langle \sigma' | T^N | \sigma \rangle, \quad (1.11)$$

where $|\sigma\rangle$ and $|\sigma'\rangle$ are the right and left boundary spins.

In the large- N limit, T^N tends towards the projector onto the eigenspace spanned by the eigenvector belonging to the largest eigenvalue

$$|\lambda_1\rangle = \lim_{N \rightarrow \infty} \frac{T^N |\sigma\rangle}{\|T^N |\sigma\rangle\|}. \quad (1.12)$$

Equation 1.12 is true for any $|\sigma\rangle$ that is not orthogonal to $|\lambda_1\rangle$.

The physical significance of the normalized lowest-lying eigenvector $|\lambda_1\rangle$ is that $\langle\lambda_1|\uparrow\rangle$ and $\langle\lambda_1|\downarrow\rangle$ represent the Boltzmann weight of $|\uparrow\rangle$ and $|\downarrow\rangle$ at the boundary of a half-infinite chain.

Maybe picture of above claim?

1.3.2 2D Ising model

Talk about exact solution (Onsager). Why is it important? Maybe star-triangle relation (Baxter). Not all IRF models solvable.

Next, we treat the two-dimensional, square-lattice Ising model. In two dimensions, the energy function is still written as in Equation 1.2, but now every lattice site has four neighbors.

Let N be the number of columns and l be the number of rows of the lattice, and assume $l \gg N$. In the vertical direction, we apply periodic boundary conditions, as in the one-dimensional case. In the horizontal direction, we keep an open boundary. We refer to N as the system size.

Picture.

Similarly as in the 1D case, the partition function can be written as

$$Z_N = \sum_{\sigma} \prod_{\langle i,j,k,l \rangle} W(\sigma_i, \sigma_j, \sigma_k, \sigma_l) \quad (1.13)$$

where the product runs over all groups of four spins sharing the same face. The Boltzmann weight of such a face is given by

$$W(\sigma_i, \sigma_j, \sigma_k, \sigma_l) = \exp \left\{ \frac{K}{2} (\sigma_i \sigma_j + \sigma_j \sigma_k + \sigma_k \sigma_l + \sigma_l \sigma_i) \right\} \quad (1.14)$$

We can express the Boltzmann weight of a configuration of the whole lattice as a product of the Boltzmann weights of the rows

$$Z_N = \sum_{\sigma} \prod_{r=1}^l W(\sigma_1^r, \sigma_2^r, \sigma_1^{r+1}, \sigma_2^{r+1}) \dots W(\sigma_{N-1}^r, \sigma_N^r, \sigma_{N-1}^{r+1}, \sigma_N^{r+1}) \quad (1.15)$$

where σ_i^r denotes the value of the i th spin of row r .

Now, we can generalize the definition of the transfer matrix to two dimensions, by defining it as the Boltzmann weight of an entire row

$$T_N(\sigma, \sigma') = W(\sigma_1, \sigma_2, \sigma'_1, \sigma'_2) \dots W(\sigma_{N-1}, \sigma_N, \sigma'_{N-1}, \sigma'_N) \quad (1.16)$$

If we take the spin configurations of an entire row as basis vectors, T_N can be written as a matrix of dimensions $2^N \times 2^N$.

Similarly as in the one-dimensional case, the partition function now becomes

$$Z_N = \sum_{\sigma} \prod_{r=1}^l T_N(\sigma^r, \sigma^{r+1}) = \text{Tr} T_N^l \quad (1.17)$$

In the limit of an $N \times \infty$ cylinder, the partition function is once again determined by the largest eigenvalue¹.

$$Z_N = \lim_{l \rightarrow \infty} T_N^l = \lim_{l \rightarrow \infty} (\lambda_0)_N^l \quad (1.18)$$

The partition function in the thermodynamic limit is given by

$$Z = \lim_{N \rightarrow \infty} Z_N \quad (1.19)$$

1.4 Partition function of the 2D Ising model as a tensor network

In calculating the partition function of 1D and 2D lattices, matrices of Boltzmann weights like W and T play a crucial role. We have formulated them in a way that is valid for any interaction-round-a-face (IRF) model, defined by

$$H \sim \sum_{\langle i,j,k,l \rangle} W(\sigma_i, \sigma_j, \sigma_k, \sigma_l), \quad (1.20)$$

where the summation is over all spins sharing a face. W can contain 4-spin, 3-spin, 2-spin and 1-spin interaction terms. The Ising model is a special case of the IRF model, with W given by Equation 1.14.

We will now express the partition function of the 2D Ising model as a tensor network. The transfer matrix T is redefined in the process. This allows us to visualize the equations in a way that is consistent with the many other tensor network algorithms under research today. For an introduction to tensor network notation, see Appendix A.

¹As in the 1D case, T is symmetric, so it is orthogonally diagonalizable.

1.4.1 Tensor network of the partition function of a system of four spins

We define

$$Q(\sigma_i, \sigma_j) = \exp(K\sigma_i\sigma_j) \quad (1.21)$$

as the Boltzmann weight of the bond between σ_i and σ_j . It is the same as the 1D transfer matrix in Equation 1.5.

The Boltzmann weight of a face W decomposes into a product of Boltzmann weights of bonds

$$W(\sigma_i, \sigma_j, \sigma_k, \sigma_l) = Q(\sigma_i, \sigma_j)Q(\sigma_j, \sigma_l)Q(\sigma_l, \sigma_k)Q(\sigma_k, \sigma_i). \quad (1.22)$$

It is now easy to see that the partition function is equal to the contracted tensor network in Figure 1.1:

$$\begin{aligned} Z_{2 \times 2} &= \sum_{\sigma_1, \sigma_2, \sigma_3, \sigma_4} \sum_{a, b, c, d} \delta_{\sigma_1, a} Q(a, b) \delta_{\sigma_2, b} Q(b, c) \delta_{\sigma_3, c} Q(c, d) \delta_{\sigma_4, d} Q(d, a) \\ &= \sum_{\sigma_1, \sigma_2, \sigma_3, \sigma_4} W(\sigma_1, \sigma_2, \sigma_3, \sigma_4). \end{aligned} \quad (1.23)$$

where the Kronecker delta is defined as usual:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \quad (1.24)$$

1.4.2 Thermodynamic limit

We define the matrix P by

$$P^2 = Q. \quad (1.25)$$

as in Figure 1.2. This allows us to write the partition function of an arbitrary $N \times l$ square lattice as a tensor network of a single recurrent tensor a_{ijkl} , given by

$$a_{ijkl} = \sum_{a, b, c, d} \delta_{abcd} P_{ia} P_{jb} P_{kc} P_{ld}, \quad (1.26)$$

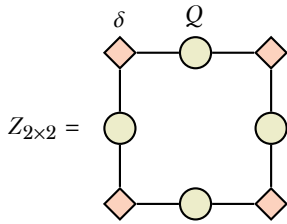


Figure 1.1: A tensor network representation of the partition function of the Ising model on a 2×2 lattice. See Equation 1.23.

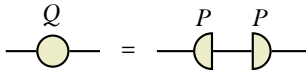


Figure 1.2: Graphical form of Equation 1.25.

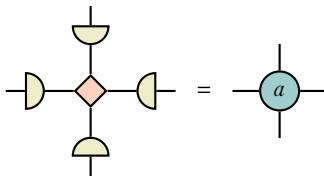


Figure 1.3: Graphical form of Equation 1.26.

where the generalization of the Kronecker delta is defined as

$$\delta_{i_1 \dots i_n} = \begin{cases} 1 & \text{if } i_1 = \dots = i_n \\ 0 & \text{otherwise.} \end{cases} \quad (1.27)$$

See Figure 1.3 and Figure 1.4. At the edges and corners, we define suitable tensors of rank 3 and 2, which we will also denote by a :

$$\begin{aligned} a_{ijk} &= \sum_{abc} \delta_{abc} P_{ia} P_{jb} P_{kc}, \\ a_{ij} &= \sum_{ab} \delta_{ab} P_{ia} P_{jb}. \end{aligned}$$

The challenge is to approximate this tensor network in the thermodynamic limit.

1.4.3 The transfer matrix as a tensor network

Say something about reshaping legs. It is implicit now.

With our newfound representation of the partition function as a tensor network, we can redefine the row-to-row transfer matrix from Equation 1.16 as the tensor network expressed in Figure 1.5. For all l , it is still true that

$$Z_{N \times l} = \text{Tr} T_N^l = \sum_{i=1}^{2^N} \lambda_i^l, \quad (1.28)$$

so the eigenvalues must be the same. That means that the new definition of the transfer matrix is related to the old one by a basis transformation

$$T_{\text{new}} = P T_{\text{old}} P^T. \quad (1.29)$$

1.5 Transfer matrix renormalization group

There is a deep connection between quantum mechanical lattice systems in d dimensions and classical lattice systems in $d+1$ dimensions. Via the imaginary time path integral formulation, the partition function of a one-dimensional

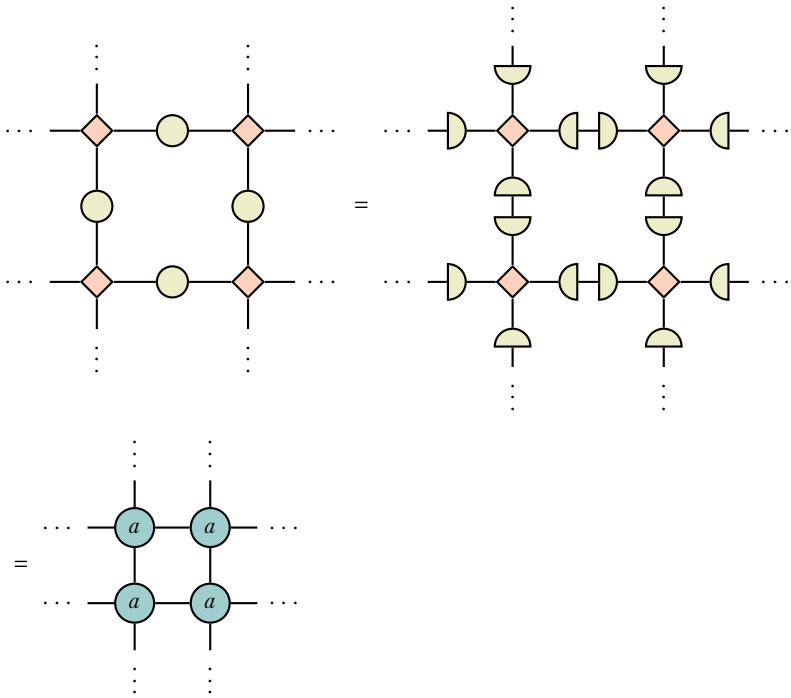


Figure 1.4: $Z_{N \times l}$ can be written as a contracted tensor network of $N \times l$ copies of the tensor a .

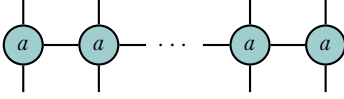


Figure 1.5: The definition of T_N as a network of N copies of the tensor a .

quantum system can be written as the partition function of an effective two-dimensional classical system. The ground state of the quantum system corresponds to the largest eigenvector of the transfer matrix of the classical system.

For more on the quantum-classical correspondence, see ??.

1.5.1 The infinite system algorithm for the transfer matrix

Nishino [1, 2] was the first to apply density matrix renormalization group methods in the context of two-dimensional classical lattices.

Analogous to the infinite system DMRG algorithm for approximating the Hamiltonian of quantum spin chains, our goal is to approximate the transfer matrix in the thermodynamic limit as well as possible within a restricted number of basis states m . We will do this by adding a single site at a time, and truncating the dimension from $2m$ to m at each iteration.

For simplicity, we assume that, at the start of the algorithm, the transfer matrix already has dimension m . We call this transfer matrix P_N . A good choice of initial transfer matrix is obtained by contracting a couple of a -tensors, until dimension m is reached. See Figure 1.6.

To specify fixed instead of open boundary conditions, we may use as boundary tensor a slightly modified version of the three-legged version of a , namely

$$a_{ijk}^\sigma = \sum_{abc} \delta_{\sigma abc} P_{ia} P_{jb} P_{kc}, \quad (1.30)$$

that represents an edge site with spin fixed at σ .

We enlarge the system with one site by contracting with an additional a -tensor, obtaining P_{N+1} . See the first network in Figure 1.7.

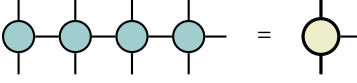


Figure 1.6: A good starting point for the half-row transfer P_N is obtained by contracting a couple of a -tensors, until P_N reaches dimension m .

In order to find the best projection from $2m$ basis states back to m , we embed the system in an environment that is the mirror image of the system we presently have. We call this matrix T_{2N+2} . It represents the transfer matrix of $2N+2$ sites. We find the largest eigenvalue and corresponding eigenvector, as shown in Figure 1.8.

The equivalent of the *reduced density matrix of a block* in the classical case is:

$$\rho_{N+1} = \sum_{\sigma_B} \langle \sigma_B | \lambda_0 \rangle \langle \lambda_0 | \sigma_B \rangle, \quad (1.31)$$

where we have summed over all the degrees of freedom of one of the half-row transfer matrices P_{N+1} . See the first step of Figure 1.9.

The optimal renormalization

$$\tilde{P}_{N+1} = O P_{N+1} O^\dagger \quad (1.32)$$

is obtained by diagonalizing ρ_{N+1} and keeping the eigenvectors corresponding to the m largest eigenvalues. See the second step of Figure 1.9.

With this blocking procedure, we can successively find

$$P_{N+1} \rightarrow P_{N+2} \rightarrow \dots, \quad (1.33)$$

until we have reached some termination condition.

Say something about termination condition.

1.5.2 Physical interpretation of the reduced density matrix

Generalizing the remarks from subsection 1.3.1.1 to the two-dimensional case, we see that the normalized lowest-lying eigenvector of the transfer matrix T_N contains the Boltzmann weights of spin configurations on the boundary of a half-infinite $N \times \infty$ lattice.

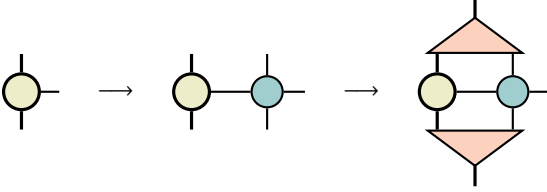


Figure 1.7: In the first step, P_{N+1} is obtained by contracting the current half-row transfer matrix P_N with an additional a -tensor. In the second step, P_{N+1} is truncated back to an m -dimensional matrix, with the optimal low-rank approximation given by keeping the basis states corresponding to the m largest eigenvalues of the density matrix. See Figure 1.9.

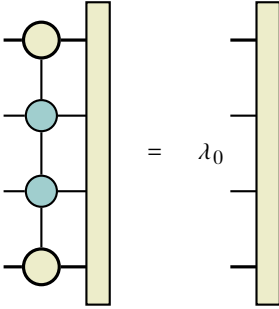


Figure 1.8: Equation for the lowest-lying eigenvector of the row-to-row transfer matrix T_{2N+2} .

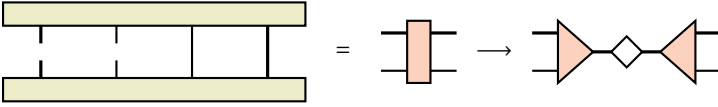


Figure 1.9: Graphical form of Equation 1.31. In the second step, ρ_{N+1} is diagonalized and only the eigenvectors corresponding to the m largest eigenvalues are kept.

Therefore, the classical equivalent of the quantum mechanical reduced density matrix, given by Equation 1.31, and by the first network in Figure 1.9, represents the Boltzmann weights of configurations along a cut in an $N \times \infty$ lattice.

Nishino and Okunishi [2], drawing on ideas from Baxter, realized the Boltzmann weights of configurations along this cut could be obtained by employing *corner transfer matrices*, making it unnecessary to solve the eigenvalue problem in Figure 1.8. Their method, called the Corner Transfer Matrix Renormalization Group method, consumes far less resources while maintaining precision. For this reason, it is the method of choice for most of the simulations in this thesis.

1.6 Corner transfer matrix renormalization group

1.6.1 Corner transfer matrices

The concept of corner transfer matrices for 2D lattices was first introduced by Baxter [3, 4, 7]. Whereas the row-to-row transfer matrix Equation 1.16 corresponds to adding a row to the lattice, the corner transfer matrix adds a quadrant of spins. It was originally defined by Baxter as

$$A_{\sigma, \sigma'} = \begin{cases} \sum \prod_{\langle i, j, k, l \rangle} W(\sigma_i, \sigma_j, \sigma_k, \sigma_l) & \text{if } \sigma_1 = \sigma'_1 \\ 0 & \text{if } \sigma_1 \neq \sigma'_1. \end{cases} \quad (1.34)$$

Here, the product runs over groups of four spins that share the same face, and the sum is over all spins in the interior of the quadrant.

In a symmetric and isotropic model such as the Ising model, we have

$$W(a, b, c, d) = W(b, a, d, c) = W(c, a, d, b) = W(d, c, b, a) \quad (1.35)$$

and the partition of an $N \times N$ lattice is expressed as

$$Z_{N \times N} = \text{Tr } A^4. \quad (1.36)$$

In the thermodynamic limit, this partition function is equal to the partition function of an $N \times \infty$ lattice, given by Equation 1.17.

The matrix in Equation 1.31, containing the Boltzmann weights of spins along a cut down the middle of an $N \times \infty$ system, is *approximated* by

$$\rho = A^4. \quad (1.37)$$

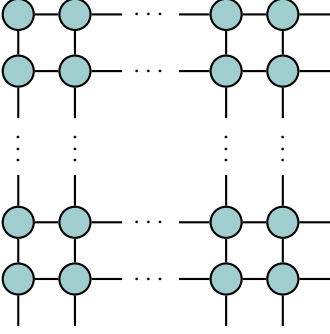


Figure 1.10: Corner transfer matrix expressed as a contraction of a -tensors.

The difference is that A^4 represents a square system of size $N \times N$ with a cut, instead of an $N \times \infty$ strip with a cut. In the thermodynamic limit, both systems become the same. In the corner transfer matrix renormalization group method, A^4 is used to find the optimal projector onto m basis states.

1.6.2 Corner transfer matrix as a tensor network

Similarly to how we redefined the row-to-row transfer matrix (Equation 1.16) as the tensor network in Figure 1.5, we can redefine the corner transfer matrix (Equation 1.34) as the tensor network in Figure 1.10. Again, the new and old definitions of A are related by a basis transformation

$$A_{\text{new}} = P A_{\text{old}} P^T. \quad (1.38)$$

Something about A being symmetric. What about non-symmetric cases?

1.6.3 Corner transfer matrix renormalization group method

Say something about boundary conditions, also with TMRG algorithm.

The algorithm proceeds very much like the transfer matrix renormalization group method. In addition to renormalizing the half-row transfer matrix P , we also renormalize the corner transfer matrix A at each step, using the projector obtained from diagonalizing A^4 .

Compare complexities of both algorithms.

We first initialize P_N and A_N , imposing boundary conditions as we see fit.

We then obtain the unrenormalized A_{N+1} by adding a layer of spins to the quadrant represented by A_N . This is done by contracting with two half-row transfer matrices P_N and a single a -tensor, as shown in the first step of Figure 1.11. We obtain the unnormalized P_{N+1} as before, as shown in the first step of Figure 1.7.

To find the optimal projector from $2m$ to m basis states, we can directly diagonalize A_{N+1}^4 , or, equivalently, A_{N+1} . As always, we keep the basis states corresponding to the m largest eigenvectors. This is shown in Figure 1.12. We use the projector to obtain the renormalized versions of A_{N+1} and T_{N+1}

$$\tilde{A}_{N+1} = O A_{N+1} O^\dagger, \quad (1.39)$$

$$\tilde{T}_{N+1} = O T_{N+1} O^\dagger. \quad (1.40)$$

shown in the second steps of Figure 1.11 and Figure 1.7.

We repeat the above procedure to successively obtain

$$A_{N+1} \rightarrow A_{N+2} \rightarrow \dots, \quad (1.41)$$

$$T_{N+1} \rightarrow T_{N+2} \rightarrow \dots \quad (1.42)$$

until a convergence criterion is reached.

Talk about convergence criterion.

1.7 Calculation of observable quantities

To do

1.8 Validity of approximation

To do. Talk about spectrum of reduced density matrix/ctm and how it relates to energy gap/correlation length and critical behaviour.

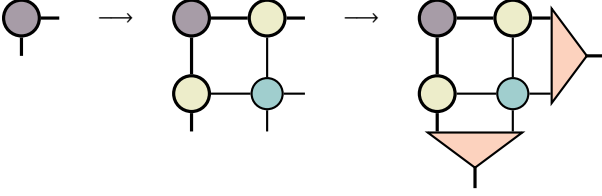


Figure 1.11: In the first step, the unrenormalized A_{N+1} is obtained by contracting with two copies of P_N and a single a -tensor. This corresponds to adding a layer of spins to the quadrant, thus enlarging it from $N \times N$ to $N + 1 \times N + 1$. In the second step, A_{N+1} is renormalized with the projector obtained from diagonalizing A_{N+1}^4 and keeping the basis states corresponding to the m largest eigenvalues.

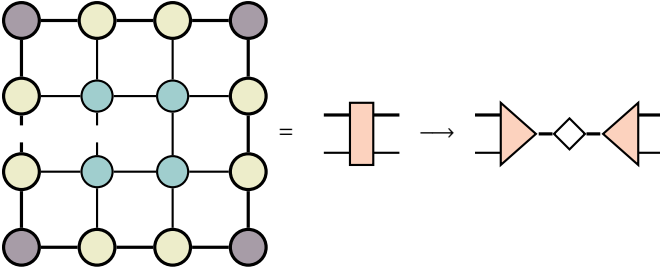


Figure 1.12: The matrix A_{N+1}^4 is approximately equal to ρ_{N+1} in Equation 1.31. Compare the graphical forms of this network and the one shown in Figure 1.9. We obtain the optimal projector by diagonalizing A_{N+1}^4 , or equivalently A_{N+1} .

1.9 Equivalence to variational approximation in the space of matrix product states.

Talk about Baxter, Rommer and Ostlund, and more recent MPS algorithms. Maybe in appendix?

A

Introduction to tensor networks

A.1 Tensors, or multidimensional arrays

Make clear that there is a difference between tensors that are defined with a metric.

In the field of tensor networks, a tensor is a multidimensional table with numbers – a convenient way to organize information. It is the generalization of a vector

$$v_i = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}, \quad (\text{A.1})$$

which has one index, and a matrix

$$M_{ij} = \begin{bmatrix} M_{11} & \dots & M_{1n} \\ \vdots & & \vdots \\ M_{m1} & \dots & M_{mn} \end{bmatrix}, \quad (\text{A.2})$$

which has two. A tensor of rank N has N indices:¹

$$T_{i_1 \dots i_N}. \quad (\text{A.3})$$

A tensor of rank zero is just a scalar.

¹The definition of rank in this context is not to be confused with the rank of a matrix, which is the number of linearly independent columns. Synonyms of tensor rank are tensor degree and tensor order.

A.2 Tensor contraction

Tensor contraction is the higher-dimensional generalization of the dot product

$$\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i, \quad (\text{A.4})$$

where a lower-dimensional tensor (in this case, a scalar, which is a zero-dimensional tensor) is obtained by summing over all values of a repeated index.

Examples are matrix-vector multiplication

$$(\mathbf{M}\mathbf{a})_i = \sum_j M_{ij} a_j, \quad (\text{A.5})$$

and matrix-matrix multiplication

$$(\mathbf{A}\mathbf{B})_{ij} = \sum_k A_{ik} B_{kj}, \quad (\text{A.6})$$

but a more elaborate tensor multiplication could look like

$$w_{abc} = \sum_{d,e,f} T_{abcdef} v_{def}. \quad (\text{A.7})$$

As with the dot product between vectors, matrix-vector multiplication and matrix-matrix multiplication, a contraction between tensors is only defined if the dimensions of the indices match.

A.3 Tensor networks

A tensor network is specified by a set of tensors, together with a set of contractions to be performed. For example:

$$M_{ab} = \sum_{i,j,k} A_{ai} B_{ij} C_{jk} D_{kb}, \quad (\text{A.8})$$

which corresponds to the matrix product $ABCD$.

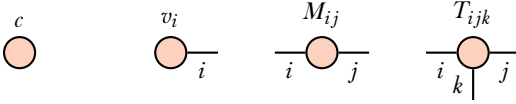


Figure A.1: Open-ended lines, called legs, represent unsummed indices. A tensor with no open legs is a scalar.

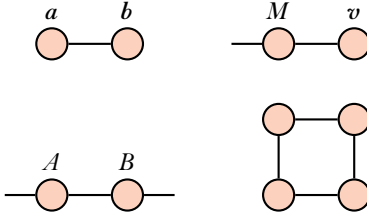


Figure A.2: Connected legs represent contracted indices. The networks in the figure represent $\sum_i a_i b_i$ (dot product), $\sum_j M_{ij} a_j$ (matrix-vector product), $\sum_k A_{ik} B_{kj}$ (matrix-matrix product) and $\text{Tr } ABCD$, respectively.

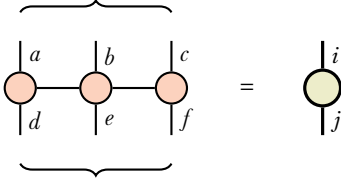
A.3.1 Graphical notation

It is highly convenient to introduce a graphical notation that is common in the tensor network community. It greatly simplifies expressions and makes certain properties manifest.

Each tensor is represented by a shape. Open-ended lines, called legs, represent unsummed indices. See Figure A.1. If it is clear from the context, index labels may be omitted from the open legs.

Each contracted index is represented by a connected line. See Figure A.2.

Many tensor equations, while burdensome when written out, are readily understood in this graphical way. As an example, consider the matrix trace in Figure A.2, where its cyclic property is manifest.

Figure A.3: Reshaping a tensor T_{abcdef} to T_{ij}

A.3.2 Reshaping tensors

Several indices can be taken together to form a single, joint index, that runs over all combinations of the indices that fused into it. For example, an $m \times n$ matrix can be reshaped into an mn vector.

$$M_{ij} = v_a \quad a \in \{1, \dots, mn\}. \quad (\text{A.9})$$

Some convention has to be chosen to map the joint index i, j onto the single index a , for example

$$a = (n - 1)i + j, \quad (\text{A.10})$$

which orders the indices of the matrix M row by row

$$11, 12, \dots, 1n, 21, 22, \dots, mn - 1, mn. \quad (\text{A.11})$$

Graphically, an index contraction is represented as a bundling of the open legs of a tensor network. See Figure A.3 for an example.

A.3.3 Computational complexity of contraction

Computational complexity.

Bibliography

- [1] Tomotoshi Nishino. “Density matrix renormalization group method for 2D classical models”. In: *Journal of the Physical Society of Japan* 64.10 (1995), pp. 3598–3601.
- [2] Tomotoshi Nishino and Kouichi Okunishi. “Corner transfer matrix renormalization group method”. In: *Journal of the Physical Society of Japan* 65.4 (1996), pp. 891–894.
- [3] RJ Baxter. “Dimers on a rectangular lattice”. In: *Journal of Mathematical Physics* 9.4 (1968), pp. 650–654.
- [4] RJ Baxter. “Variational approximations for square lattice models in statistical mechanics”. In: *Journal of Statistical Physics* 19.5 (1978), pp. 461–478.
- [5] Steven R White. “Density matrix formulation for quantum renormalization groups”. In: *Physical Review Letters* 69.19 (1992), p. 2863.
- [6] Ernst Ising. “Beitrag zur theorie des ferromagnetismus”. In: *Zeitschrift für Physik* 31.1 (1925), pp. 253–258.
- [7] Rodney J Baxter. *Exactly solved models in statistical mechanics*. Elsevier, 1982. Chap. 13.