

中国科学技术大学

学士学位论文



张量网络优化算法

作者姓名： 孙玉鑫

学科专业： 凝聚态物理

导师姓名： 何力新 教授

完成时间： 二〇二〇年七月一日

University of Science and Technology of China
A dissertation for bachelor's degree



Optimization Algorithms of Tensor Network

Author: Sun Yuxin

Speciality: Condensed Matter Physics

Supervisor: Prof. He Lixin

Finished time: July 1, 2020

致 谢

我非常感谢在大学生活中帮助我，指导我的老师与同学们。

首先我要感谢实验室的何力新教授对我科研方面的指导。作为本科生初次接触科研工作，有很多方面都不了解，而对量子体系的模拟计算由于需要大量课本外的知识，对本科生来说更是难上加难。我十分感谢何老师给我在他的实验室学习的机会，并且在研究方向上给予指导，让我从简单问题入手循序渐进，逐渐了解实验室研究所需的各种知识；以及在我论文写作遇到困难时，何老师也认真指导我如何克服困难解决问题，如何正确的安排论文结构等。

感谢实验室中已经毕业出国的李东昊师兄对我的帮助。师兄作为只大我一届的学长，与我一个实验室，也曾是我所在的学生会部门的部长，因此我在很多事情上都与他有共同语言。当我在学习生活中遇到问题时，我也更倾向于找李师兄一起讨论。另外我能够了解到并最终加入何老师的实验室，也是李东昊师兄在听到我希望做计算物理相关工作时向我推荐的。

感谢这两年以来，实验室中的其他老师与师兄们的帮助。感谢董少钧师兄在我初到实验室时对我的辅导，每次我遇到各种问题时，师兄总是认真的给我讲解，直到我彻底明白之前不懂的知识点，让我的学习效率远超过自己去阅读晦涩难懂的论文。师兄就像是我在实验室的辅导员一样，辅导着我科研的方方面面。感谢张浩师兄对我的指导，师兄用 Linux 系统工作已达到炉火纯青的地步，并且在系统与编程方面向我分享了许多有用的知识，激发我进一步学习的兴趣。感谢实验室中的韩永建老师，王超师兄与张盟师兄，他们在与我讨论问题时能认真的向我讲解我的知识盲区，在我组会报告出现错误时敏锐的找出错误并纠正，对我的科研与学习给予方方面面的帮助。

特别感谢我的班主任封常青老师对我学习生活中的帮助。封老师是一位对待学生学习生活问题非常负责的老师，每次同学们遇到问题时封老师都会抽出自己的时间帮助同学们解决问题；平时发布学校通知，开班会，找同学谈话了解情况等辅导班级的工作，封老师也非常上心。本论文在写作的过程中，我的家人生病住院，我也因此心态受到影响，耽误了论文的写作进度。封老师察觉到我的异常后，便迅速找我了解情况，并在后续不间断的关心、询问我的情况，督促我完成论文，在我浑浑噩噩整日无所事事时，帮助我排忧解难，点醒颓废的我。没有封老师的帮助，我可能就会因为自己的颓废而使得各种问题接踵，最终影响自

己的毕业。现在我想，毕业多年之后，我仍然会对自己能幸运地被分到封老师的班级而心存感激。

目 录

中文内容摘要	3
英文内容摘要	4
第一章 绪论	6
一、背景介绍	6
二、为何要使用张量网络	6
第二章 张量网络态	8
第一节 张量	8
一、张量的定义	8
二、张量的运算	9
第二节 张量网络	10
一、张量网络的定义	10
二、张量网络的图表示法	11
三、张量网络的运算	11
第三节 张量网络与量子态	12
一、量子态的常规表示	12
二、矩阵乘积态	14
三、投影纠缠态	18
第三章 张量网络算法	21
第一节 张量网络的求基态算法	21
一、虚时演化算法	21
二、变分法	22
三、梯度下降法	23
第二节 梯度下降算法的优化	25
一、随机梯度下降	25
二、方差缩减算法	27
三、张量网络与 SVRG	28

第四章 数值实验	30
第一节 算法测试结果	30
一、复现 SVRG	30
二、算法改造	31
第二节 结论与推广	33
一、结论	33
二、算法推广	33
参考文献	36
附录 A 补充材料	39
一、代码仓库	39
二、相关文献	39

中文内容摘要

张量网络正成为量子多体系统研究中备受欢迎的表示方法，它的出现给强关联系统的研究提供了全新的视角。

本文从基本的张量网络定义开始，简要地介绍了张量网络的常用形式 **MPS** 与 **PEPS**，并通过 **MPS** 介绍张量网络中的时间演化算法，以及进一步衍生的虚时演化求基态算法。之后本文详细介绍了张量网络的梯度优化算法以及与随机梯度下降有关的工作，并尝试将机器学习中优化梯度算法的 **SVRG** 方法使用在张量网络上以提高性能。

关键词：张量网络; 矩阵乘积态; 投影纠缠对; 梯度优化算法; 方差缩减的随机梯度下降

Abstract

The Tensor Network method, which gives people a new approach to Strongly Correlated Systems, is becoming more and more popular in quantum many body research.

This paper is a brief introduction from the definition of tensor network to its commonly used form: MPS and PEPS. We briefly summarize the Time Evolution methods in MPS and the imaginary time evolution technique. Besides, we illustrate the procedure of gradient optimization and try to improve its performance with some optimization algorithms in machine learning.

Key Words: Tensor Network; Matrix Product State; Projected Entangled Pair State; Gradient Optimization; Stochastic Variance Reduced Gradient

符号说明

SVD 奇异值分解 (Singular Value Decomposition)

MPS 矩阵乘积态 (Matrix Product State)

PEPS 投影纠缠 (对) 态 (Projected Entangled Pair State)

VBS 价键态 (Valance Bond State)

tMPS 矩阵乘积态的时间演化 (Time Evolution with MPS)

TEBD 逐块时间演化 (Time Evolution Block Decimation)

SU 简单更新 (Simple Update)

$P(w)$ 损失函数, 这里向量 w 与引用文献保持一致不再加粗

reweighting 利用给定概率分布计算另一概率分布下物理量期望值的技术

第一章 绪论

一、背景介绍

理解量子多体系统也许是凝聚态物理中最具挑战性的问题。科学家们一直在致力于研究高温超导，拓扑序，量子自旋液体等现象背后的物理本质，但仅仅通过理论分析已经很难得出这种复杂系统的所有性质，因此人们提出简化后的理论模型，并通过计算机模拟来尝试复现真实世界中存在的现象。

近些年来，张量网络作为一种数值模拟手段正在受到越来越多人的重视。不同于传统的给出量子态各个基矢上的系数的表示方法，张量网络将量子态表示为一系列张量的缩并，其相比于经典的表示方法，天然的反映出量子系统中的纠缠与关联的特性。

张量网络表示法具有非常高的灵活性，我们可以研究一个系统在不同维度下的表现，可以研究有限系统与无穷大系统之间的区别，还可以研究系统在不同边界条件、不同对称性下的性质，也可以研究波色、费米或分数系统的特点。并且在张量网络上发现的技术在其他的领域也有很好的应用空间，如量子化学等。

二、为何要使用张量网络

1. 扩展技术的边界

目前所有的模拟方法都有其自身的局限性。最常见的方法如平均场理论^[1]，它无法很好的描述量子关联；量子蒙特卡洛方法^[2]对于费米子系统等不适用，会出现符号问题；精确对角化方法^[3]计算量过大，这使得可计算的系统只能被限制在很小的区域；密度泛函理论^[4]依赖于我们对电子交换关联泛函的选择，且通常针对不同的问题，不同的泛函形式各有优劣。同样的，张量网络算法并不完美，它也有一定的适用范围。它对量子态的纠缠形式做出了限制，但是也和其他方法一样，给我们提供全新的视角来模拟量子态，使得更多的模型可被计算机模拟。

2. 新的物理描述

通过彼此相连的张量构成的网络来表示量子态，这种方法抓住了系统的主要特征——纠缠。通过这种方式来描述量子态与经典方式截然不同，我们不再是仅仅处理一堆基矢上的系数，而是通过图表的形式描述与理解量子体系。它被认为是描述量子系统的自然语言，对凝聚态物理来说是一种全新的语言。

3. 真实态空间过大

使得张量网络备受欢迎的一个主要原因可能便是真实系统的态空间往往过于巨大了。多体系统被表示为多个系统的直积，表示系统需要的基矢将随系统规模指数及上升。即使是最简单的 N 个自旋- $\frac{1}{2}$ 自旋，其态空间维度也是 2^N 。

幸运的是我们没有必要将所有基矢都包括进来，因为绝大多数自然界存在的多体系统，其相互作用都是局域化的。这使得一些系统的基态满足纠缠熵面积定律，我们将在下文介绍这一性质。我们通常不必考虑整个希尔伯特空间，而只需考虑满足纠缠熵面积定律的状态，这种量子态通常只占希尔伯特空间之一隅，却恰恰是张量网络擅长表示的。

第二章 张量网络态

第一节 张量

一、张量的定义

1. 抽象定义

若 V_i 与 W_i 都是域 F 上的有限线性空间, \otimes 是张量积, 那么一个张量即是原空间 W_i 与对偶空间 V_i^* 的张量积空间中的一个元素, 若张量积空间由 p 个原空间与 q 个对偶空间构成, 则其上张量被称为 (p, q) 阶的:

$$\text{order-}(p, q) \text{ tensor } T \in W_1 \otimes W_2 \otimes \cdots \otimes W_p \otimes V_1^* \otimes V_2^* \otimes \cdots \otimes V_q^*$$

若 $\{e^{(i)}_k\}$ 与 $\{\eta^{(i)}_k\}$ 分别为空间 W_i 与对偶空间 V_i^* 的基, 则利用爱因斯坦求和约定, 张量可以被表示为分量和的形式:

$$T = T^{i_1 \cdots i_p}_{j_1 \cdots j_q} e^{(1)}_{i_1} \otimes \cdots \otimes e^{(p)}_{i_p} \otimes \eta^{(1)j_1} \otimes \cdots \otimes \eta^{(q)j_q}$$

用于描述量子多体系统的张量网络是由具体的张量表示的, 因此为了方便与准确的表示本文中的张量, 我们将张量的定义特殊化以方便描述本文中使用的张量。

2. 具体定义

一个实数域或复数域上的有限多维数组 $\{T_{i_1 i_2 \cdots i_p}\}$ 被称为一个张量, 若表示一个张量的数组的维度为 p , 则称此张量为 p 阶的。若第 k 个指标 i_k 的取值范围为集合 S_k , 则 $|S_k|$ 称为指标 i_k 的维数。若两个同阶张量的指标维数依次相等, 则称两个张量是相同形状的。

相较于上文中的定义, 这里应用了以下改动:

(1) 指定数域

一个量子系统至多在复数域即可被表示, 因此这里把数域指定为实数域或复数域, 此时张量可表示为实数或复数域上的多维数组。

(2) 约定形式

本文中的张量有两种指标, 一种是由实空间取向标记的缩并指标, 一种是由对应的自旋分量 (通常为 \hat{S}_z 的本征值) 标记的物理指标。这里我们至多只需研究张量在不同自旋右矢下的变换, 因此不严格区分协变与逆变指标, 默认张量为 $(0, p)$ 或 $(p, 0)$ 阶的。

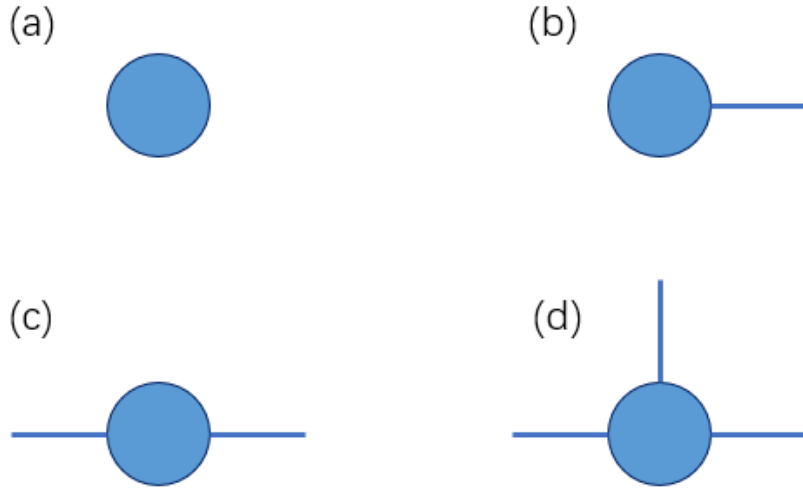


图 2.1 张量网络的图表示

注：(a) 零阶张量，标量；(b) 一阶张量，矢量；(c) 二阶张量，矩阵；(d) 三阶张量。

二、张量的运算

1. 加法运算

两个相同形状的张量 A 与 B 之间可以定义加法： $C = A + B$ ，即：

$$C_{i_1 i_2 \dots i_p} = A_{i_1 i_2 \dots i_p} + B_{i_1 i_2 \dots i_p} \quad (2.1)$$

2. 数乘运算

对任意 $\alpha \in \mathbb{C}$ ，可定义张量与其的数乘运算： $M' = \alpha M = M\alpha$ ，即：

$$M'_{i_1 i_2 \dots i_p} = \alpha M_{i_1 i_2 \dots i_p} \quad (2.2)$$

3. 缩并运算

若两个张量 A 与 B 间的一个或者一些指标维数分别相同，则可以定义缩并运算对重复指标遍历求和：

$$C_{\alpha\beta} = \sum_{\gamma=1}^{D_\gamma} A_{\alpha\gamma} B_{\gamma\beta} \quad (2.3)$$

或

$$C_{\alpha\beta} = \sum_{\delta, \gamma, \nu=1}^{D_\delta, D_\gamma, D_\nu} A_{\alpha\delta\gamma\nu} B_{\gamma\beta\nu\delta} \quad (2.4)$$

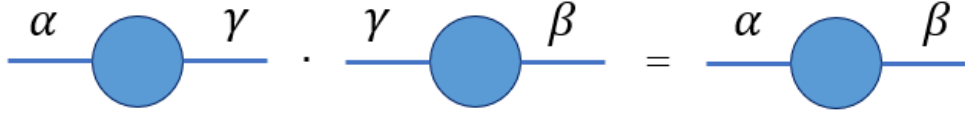


图 2.2 张量缩并

注：相同指标 γ 缩并后消失，图示与矩阵乘法一致。

所得张量的未缩并指标可继续缩并：

$$M_{\alpha mn} = \sum_{\beta=1}^{D_\beta} C_{\alpha\beta} F_{\beta mn} = \sum_{\delta, \gamma, \nu, \beta=1}^{D_\delta, D_\gamma, D_\nu, D_\beta} A_{\alpha\delta\gamma\nu} B_{\gamma\beta\nu\delta} F_{\beta mn} \quad (2.5)$$

4. 分解运算

分解可看作缩并的逆运算。若给定张量 $\{M_{\alpha mn}\}$ ，则如公式 (2.5) 中将张量表示为几个张量的缩并的形式，称为张量的分解。

由以上定义易知，相同形状的张量构成线性空间，其加法与数乘的性质与矢量一致，且张量之间缩并的先后顺序不影响结果。下文中为了书写方便，声明求和约定后，张量缩并均采用爱因斯坦求和约定，即默认重复出现的指标需要求和。例如公式 (2.5) 重新写为：

$$M_{\alpha mn} = C_{\alpha\beta} F_{\beta mn} = A_{\alpha\delta\gamma\nu} B_{\gamma\beta\nu\delta} F_{\beta mn}$$

第二节 张量网络

一、张量网络的定义

张量网络是一组张量的集合与其上的缩并规则构成的数学结构。整个张量网络等效于一个巨大的张量^[5]。

张量网络的全部结构由无向图 $G = (V, E)$ 表示，其中顶点集 V 为空顶点集 V_L 与张量顶点 V_T 的不交并，边集 E 为 E_L 与 E_T 的不交并 (指交集为空的两个集合的并集)。表示规则为：

- (1) 每个张量与 V_T 中元素一一对应；
- (2) 每个未缩并指标与 V_L , E_L 中元素都一一对应；
- (3) 每对缩并指标与 E_T 中元素一一对应。

边集 E 中元素被称为脚, E_L 中元素只与一个张量顶点相连, 我们称其为“悬空脚”, 又因为其对应真实的自旋等有物理意义的量, 所以也被称为物理脚; E_T 中元素连接两个张量顶点, 我们称其为“缩并脚”或者“键”。

二、张量网络的图表示法

每个张量网络都对应一个无向图, 因此我们可以将图画出, 通过图形来了解张量网络的结构。在通常的无向图上, 我们将张量顶点扩大为空心图形着重显示, 用以代表此顶点对应点的张量。这种无向图的表示被称为张量网络图。

在张量网络图中, 每个 n 阶张量一定有 n 个脚, 需要缩并的张量之间公共脚的个数对应需要缩并的指标个数, 每个脚对应的指标的维数不表示在图中。相比于文字叙述或代数表达, 张量网络图能够更直观的表示张量网络的结构, 也使得我们能更高效的描述对网络施加的缩并、分解等操作。

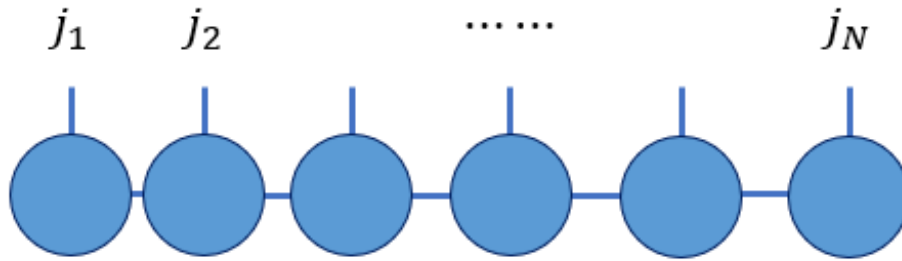


图 2.3 张量网络之矩阵乘积态

三、张量网络的运算

1. 张量网络的缩并

一个张量网络中的张量之间存在缩并关系, 当我们应用这些关系将某些张量缩并为一个张量时, 其张量网络图中对应的边表现为收缩至消失, 且其两端的张量顶点融合为一个, 如图2.2。这种操作被称之为张量网络对其边(键)的缩并。

2. 张量网络的分解

如同张量的分解可看做张量缩并的逆运算，张量网络的分解也可借助其缩并来定义。当我们将网络中的某些张量分别拆分为其他张量的缩并时，这种操作被称之为张量的分解，如图2.4所示。

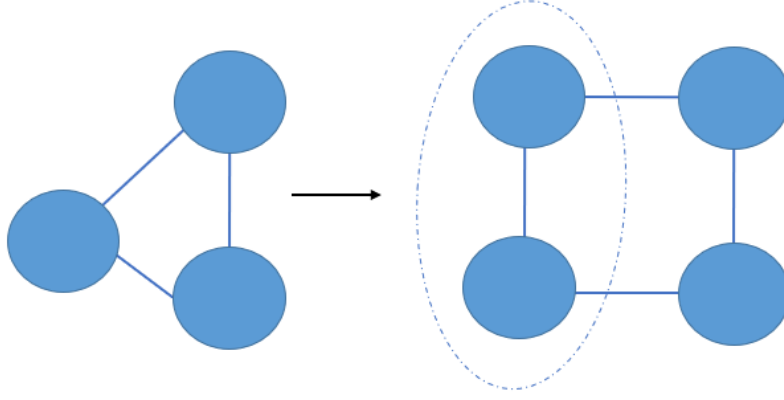


图 2.4 张量网络的分解

第三节 张量网络与量子态

一、量子态的常规表示

上文主要介绍了张量网络的相关定义以及性质，本节我们将说明张量网络具体应怎样应用至量子态的表示。

在从多体量子态过渡到张量网络之前，我们先回顾一下多体系统的一些概念。

1. 直积态

两个量子系统的纯态分别记为 $|\phi_1\rangle \in \mathcal{H}_1$ 与 $|\phi_2\rangle \in \mathcal{H}_2$ ，纯态 $|\varphi\rangle \in \mathcal{H}$ 表示两个量子系统视为一个整体时的量子态，则 $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ 。若 $|\varphi\rangle$ 具有形式

$$|\varphi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \quad (2.6)$$

则 $|\varphi\rangle$ 被称为直积态。否则，其被称为纠缠态。对于两体纠缠态，其不能直接写为两个纯态的直积，被表示为一系列直积之和

$$|\varphi\rangle = \sum_{i,j}^{d_i, d_j} c_{i,j} |i\rangle |j\rangle \quad (2.7)$$

为了研究这个两体系统的性质，我们可能会用到一些数学处理。

2. 奇异值分解

对任意矩阵 $\mathbf{A} \in \mathbb{C}^{M \times N}$ 做奇异值分解^[6](Singular Value Decomposition, SVD) 得

$$\mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\dagger \quad (2.8)$$

$$(1) \mathbf{U} \in \mathbb{C}^{M \times L}, \mathbf{U}^\dagger \mathbf{U} = \mathbf{I};$$

$$(2) \mathbf{V}^\dagger \in \mathbb{C}^{L \times N}, \mathbf{V}^\dagger \mathbf{V} = \mathbf{I};$$

(3) $\mathbf{\Lambda}$ 为 $L \times L$ 非负实对角阵, 对角元素 λ_i 称为奇异值, 且非 0 对角元素个数称之为 \mathbf{A} 的秩, 对任意酉不变范数有 $\|\mathbf{A}\|^2 = \sum_i \lambda_i^2$. 其中 $L = \min(M, N)$.

3. 施密特分解

对于有限维的两体纠缠系统, 存在一种称为施密特分解的方法, 它将 (2.7) 中的量子态分解为以下形式

$$|\varphi\rangle = \sum_{i=1}^{\chi} \lambda_i |i_A\rangle |i_B\rangle \quad (2.9)$$

其中 λ_i 为正数且有 $\sum_i \lambda_i^2 = 1$. 施密特分解可由奇异值分解得来, 我们只需将 SVD 分解应用在 $\{c_{i,j}\}$ 上,

$$|\varphi\rangle = \sum_{i,j}^{d_i, d_j} c_{i,j} |i\rangle |j\rangle \quad (2.10)$$

$$= \sum_{i,j}^{d_i, d_j} \sum_k^{\chi} u_{i,k} \lambda_k v_{k,j} |i\rangle |j\rangle \quad (2.11)$$

$$= \sum_k^{\chi} \lambda_k \left(\sum_i^{d_i} u_{i,k} |i\rangle \right) \left(\sum_j^{d_j} v_{k,j} |j\rangle \right) \quad (2.12)$$

$$= \sum_{k=1}^{\chi} \lambda_k |k_A\rangle |k_B\rangle \quad (2.13)$$

$$= \sum_{i=1}^{\chi} \lambda_i |i_A\rangle |i_B\rangle \quad (2.14)$$

矩阵 $\{c_{i,j}\}$ 的秩 χ 在这里被称为施密特秩, 被某一表象下的 $|\varphi\rangle$ 唯一确定。施密特秩可以用来衡量两体系统之间纠缠状态数的多少, 也经常会出现对多体系统的分析中。

4. 纠缠熵

在零温时, 纯态量子系统的冯诺依曼熵为零, 但选取一部分子系统单独查看, 其熵可能不为零。这部分仅仅由量子系统的纠缠产生的熵被定义为冯诺依曼

纠缠熵^[7]

$$S(\rho_A) = -\text{Tr}(\rho_A \log \rho_A) = -\text{Tr}(\rho_B \log \rho_B) \quad (2.15)$$

其中 A, B 是系统一分为二的两部分, ρ_A 是密度矩阵对 B 系统求部分迹得到的约化密度矩阵, 反之同理。若利用施密特分解将系统 $|\varphi\rangle$ 分为 (2.9) 中的两部分, 则冯诺依曼纠缠熵可以表达为

$$S = -\sum_i^{\chi} \lambda_i^2 \log(\lambda_i^2) \quad (2.16)$$

纠缠熵是一个用来衡量量子系统纠缠大小的物理量。一个具有纠缠的基态的强关联系统有时能表现出特殊的集体量子行为^[8], 因此纠缠熵在研究强关联系统时有重要意义。通常情况下, 基态附近的量子态均满足纠缠熵面积定律, d 维系统的纠缠熵正比于边界面积 L^{d-1} 。

5. 多体系统

通常情况下, 我们会遇到由 N 粒子构成的量子多体系统, 其每一个粒子具有维度为 p 的态空间。最常见的例子即是一个由自旋 $\frac{1}{2}$ 的粒子组成的 N 粒子自旋系统, 其 $p = 2$ 。要表示这样的量子态, 我们可以选择单体 \hat{S}_z 算符的本征态 $|j_k\rangle$ 为每个粒子态空间的基, 这样整体态空间的基可由每个粒子态空间基的直积构成, 整体的态矢 $|\varphi\rangle \in (\mathbb{C}^d)^{\otimes N}$ 表示为

$$|\varphi\rangle = \sum_{j_1 \cdots j_N} c_{j_1 \cdots j_N} |j_1 \cdots j_N\rangle = \sum_{j_1 \cdots j_N} c_{j_1 \cdots j_N} |j_1\rangle \otimes \cdots \otimes |j_N\rangle \quad (2.17)$$

其中 $j_k = \uparrow$ 或 \downarrow 。系数 $c_{j_1 \cdots j_N} \in \mathbb{C}$ 对于所有的指标构成一个张量 $\{c_{j_1 \cdots j_N}\}$, 将其用张量网络图表示如图2.5。在这种表示中, 表示一个 N 体系统的量子态需要的参数的个数为 2^N , 它随系统规模的增大而指数级增大。若想利用计算机来计算这个多体系统的有关性质, 所需程序的空间复杂度至少是 $\mathcal{O}(2^N)$ 的。在这种情况下, 程序可计算的系统大小就会因为内存有限等原因被限制在很小的规模。

为了解决这个问题, 我们希望寻找上述系统的其它的表示方式。这种表示方式应该避免指数级的空间复杂度, 并且能正确的表示系统的状态。

二、矩阵乘积态

1. 描述

为了避免过高的空间复杂度, 一种可行的方法是将整个张量拆分为不同的小张量的缩并。其中被构造出的一种一维结构叫做矩阵乘积态 (Matrix Product

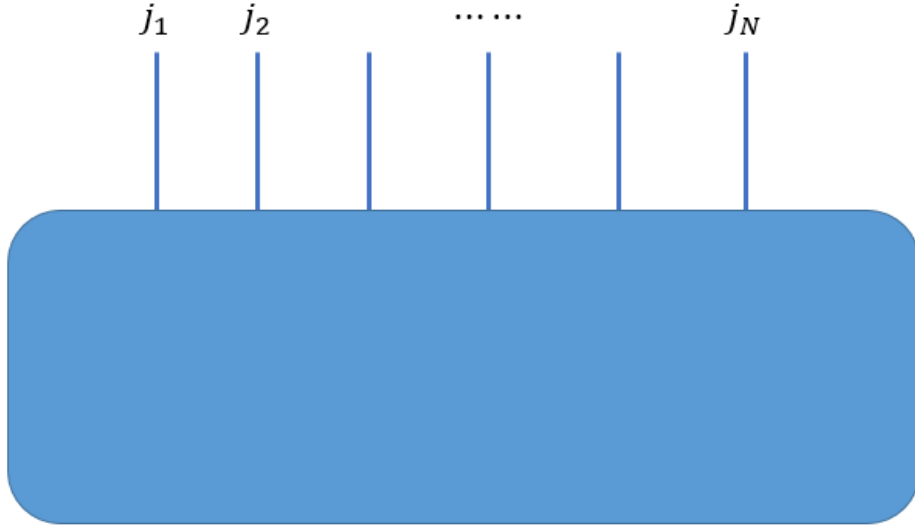


图 2.5 系数矩阵看做张量

State, MPS), 它在周期边界条件下将 $|\varphi\rangle$ 表示为

$$|\varphi\rangle = \sum_{j_1 \cdots j_N} \text{Tr} \left(\mathbf{A}_1^{j_1} \mathbf{A}_2^{j_2} \cdots \mathbf{A}_N^{j_N} \right) |j_1 j_2 \cdots j_N\rangle \quad (2.18)$$

在开边界下表示为

$$|\varphi\rangle = \sum_{j_1 \cdots j_N} \langle l | \mathbf{A}_1^{j_1} \mathbf{A}_2^{j_2} \cdots \mathbf{A}_N^{j_N} | r \rangle |j_1 j_2 \cdots j_N\rangle \quad (2.19)$$

上式中 $\mathbf{A}_n^{j_n}$ 表示第 n 个张量上自旋 j_n 对应的矩阵 (2 阶张量)。在周期边界条件下, 考虑上文描述的 N 粒子 p 自旋系统, 表示其量子态需要 $\mathcal{O}(2^N)$ 的空间复杂度。公式 (2.18) 中需要的参数个数为小矩阵的和, 系统中每个粒子的每个状态对应一个参数矩阵, 因此整体的空间复杂度为 $\mathcal{O}(NpD^2)$, 其中 D 来自于假设每个矩阵 $\mathbf{A}_n^{j_n} \in \mathbb{C}^{D \times D}$ 。在新的表示方式下, 系统的空间复杂度降低为多项式级, 这样在系统规模不断扩大时, 我们计算所需的内存开销增速显著减慢, 这意味着我们可以承受对更大规模系统的计算。

2. VBS 态

另一种定义 MPS 的方式是通过 Valance Bond States^{[9]8} 来构造系统的量子态。我们考虑上文中一维周期边界条件上的自旋链, 每个格点上由两个相同的自由度为 D 的粒子 a_i 与 b_i , 每个粒子与相邻格点形成最大纠缠态

$$|\varphi(b_i a_{i+1})\rangle = |I\rangle = \sum_{\alpha=1}^D |\alpha, \alpha\rangle, \quad (2.20)$$

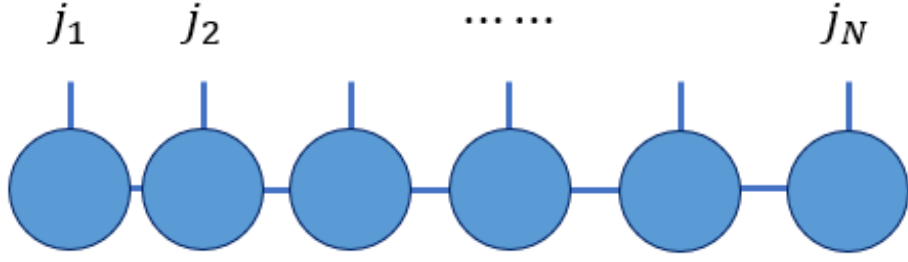


图 2.6 开边界条件的 MPS

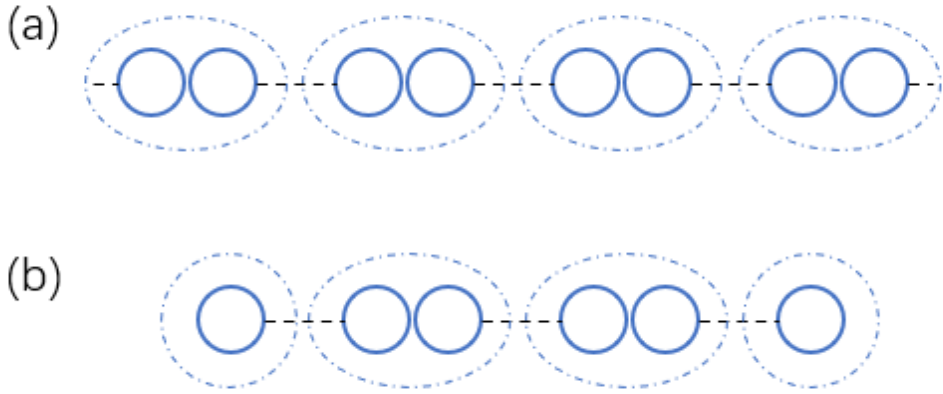


图 2.7 VBS 态

(a) 周期边界条件下的 VBS 态；(b) 开边界条件下的 VBS 态。同一虚线圈中的对象对应真实空间同一个格点

整个系统由

$$|\Phi\rangle = |I\rangle^{\otimes N} \quad (2.21)$$

构成，如图2.7(a)。每一对粒子 a_i, b_i 构成比原先单粒子空间更大的希尔伯特空间。在构造完成后，我们需要将非物理的自由度排除，使其回到原先的空间，因此在第 n 个格点考虑我们需要用到的投影算符 $\hat{A}(n)$

$$\hat{A}(n) = \sum_{i=1}^p \sum_{\alpha, \beta=1}^D A(n)_{\alpha, \beta}^i |i\rangle \langle \alpha, \beta| \quad (2.22)$$

将其逐点作用在 $|\Phi\rangle$ 上^{[10]16}，我们便得到了与 (2.18) 相同的量子态 $|\varphi\rangle$ 。公式中出现的 $A(n)_{\alpha, \beta}^i$ 正是张量网络中出现的张量 \mathbf{A}_n 的各个分量。这里使用的边界条

件为周期边界条件，对于开边界条件也可类似地构造，如图2.7(b)。

3. 性质

(1) 边界条件

开边界条件下，MPS 不具有平移对称性，因为原则上它的每个张量都可以是不同的。在周期边界条件下，我们将系统由指定的元胞无限平移而成，那么对应的张量网络也看做由指定张量序列无限平移形成的极限，它可以具有我们所表示系统的平移对称性。

(2) 纠缠熵面积定律

纠缠熵面积定律指出，张量网络子系统的纠缠熵仅仅与子系统边界有关。MPS 遵守纠缠熵面积定律，由于是一维系统，其边界是零维的端点，因此其边界与系统规模无关，纠缠熵 $S(L) = -\text{Tr}(\rho_L \log \rho_L) = \mathcal{O}(\log D)$ 为常数^[11]，其中 ρ_L 为子系统 L 的约化密度矩阵。

纠缠熵面积定律是张量网络能够有效表示量子态的原理所在。对于有能隙的系统，其基态总是遵守纠缠熵面积定律，而任何遵守纠缠熵面积定律的量子态都能被 MPS 所表示^[12]。这就解释了为何 MPS 可以用更少的参数表示量子态：当我们关注系统在基态附近的性质时，系统所处的量子态只能是那些遵循纠缠熵面积定律的量子态，这些态都可以被 MPS 表示，系统的参数在面积定律的约束下大大减少。

(3) 关联长度有限

两体关联反应一个算符在不同位置处时其期望值的相关程度

$$C(r) = \langle O_i O'_{i+r} \rangle - \langle O_i \rangle \langle O'_{i+r} \rangle \quad (2.23)$$

在文献 [11] 的 130 页中给出了一个简单的例子，证明了一个长度为一的在 MPS 周期边界条件下关联函数是指数下降的。这种指数下降在一维有能隙系统中普遍存在，这侧面反映了为何 MPS 能很好的表示这种系统。反之，对处于相变点附近的，具有趋于无穷的关联长度的系统，MPS 无法表示。

(4) 精确缩并复杂度

计算两个 MPS 态的内积，即缩并两个 MPS，其时间复杂度总可以达到 $\mathcal{O}(NpD^3)$ 。这个性质不难理解，因为我们的计算难度与计算 N 个矩阵的乘积并无本质区别。正是如此，我们的 MPS 算法可以高效运行，或者说不同于时间复杂度指数增加的算法，MPS 上的算法复杂度计算大系统是可接受的。

三、投影纠缠态

1. 描述

矩阵直积态利用多体系统基态的特点对多体系统的整个希尔伯特空间做了近似, 并因此能够高效的描述一维多体系统。将这种方法推广到二维系统的一个尝试便是投影纠缠态 (Projected Entangled Pair States, PEPS), 它是许多二维格点算法的基础。同上文中矩阵乘积态的思想一致, 我们希望将 (2.17) 中系数构成的张量分解为小张量的缩并来减少参数。若我们的系统是一个 $N \times M$ 的方形格点, 我们可以按与系统格点相同的方式分解张量

$$|\varphi\rangle = \sum_{\{j\}} \text{Tr} [A(1,1)^{j_{1,1}} A(1,2)^{j_{1,2}} \dots A(N,M)^{j_{N,M}}] |j_{1,1} j_{1,2} \dots j_{N,M}\rangle \quad (2.24)$$

这里的每个 $A(n,m)^{j_{n,m}}$ 都是一个 4 阶张量, 含有四个与周围张量缩并的键, 整体上每个格点对应一个 5 阶张量 $A(n,m) = A_{n,m}(l,r,u,d,j_{n,m})$, 只有一个悬空脚为 j_{nm} .

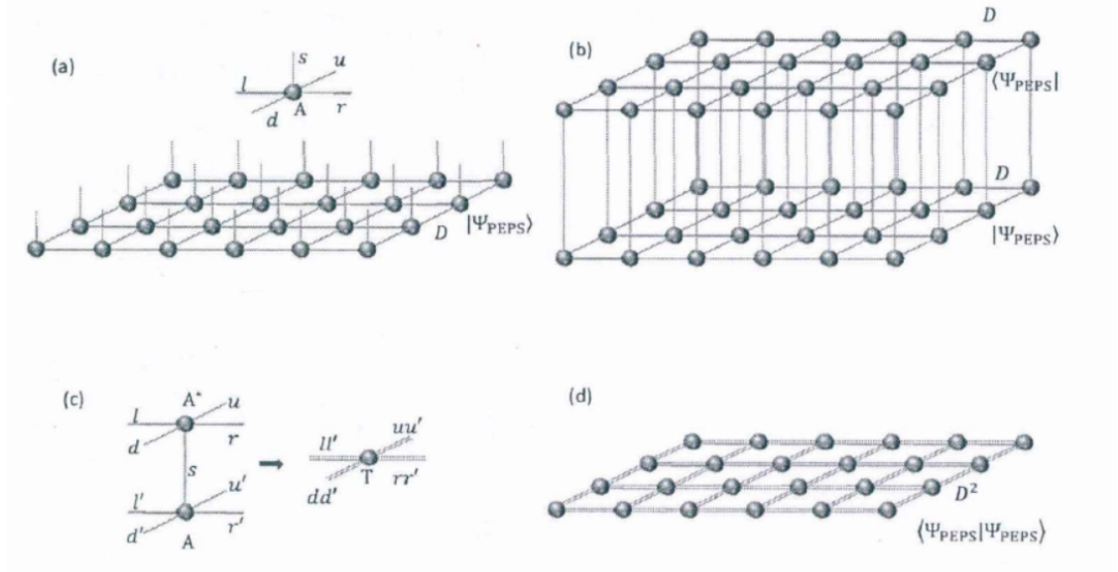


图 2.8 开边界条件的 MPS

注: 图片引用自实验室师兄的文章^{[13][31]}. (a) 键的维度为 D 的 PEPS 的图表示; (b) 左矢和对应右矢构成的双层张量网络; (c) 先对每个格点物理指标进行求和; (d) 由 c 中 $D^2 \times D^2 \times D^2 \times D^2$ 的张量构成的张量网络。

2. VBS 态

另外还有一种看待投影纠缠态的方式, 是像上文中 MPS 那样将 PEPS 看做由高维希尔伯特空间中最大纠缠态投影得到的 VBS 态。我们考虑在系统的每个

格点上共有 4 个虚拟的自由度为 D 的粒子分别与上下左右的格点中的相邻粒子成键，如图2.9. 类似的构造

$$A(n, m) = \sum_{i=1}^p \sum_{l,r,u,d=1}^D A(n, m)_{l,r,u,d}^i |i\rangle \langle l, r, u, d| \quad (2.25)$$

便可得到 (2.24) 中的结果

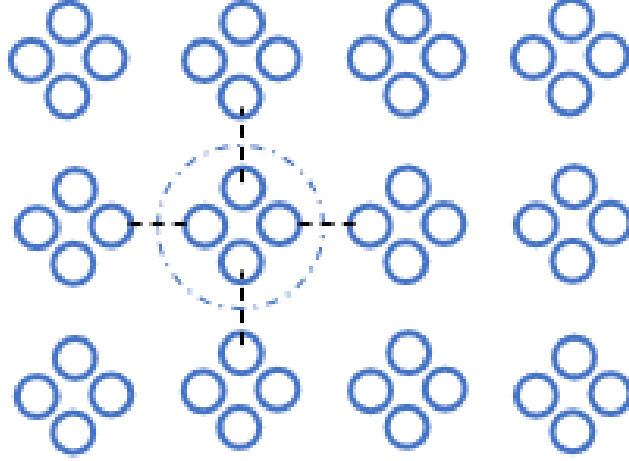


图 2.9 二维 VBS 态单格点示意

3. 性质

(1) 边界条件

对于开边界条件的 PEPS，每个张量都可以是不同的，无平移对称性。对于周期边界条件的 PEPS，我们的系统被看做由我们适当选择的元胞无限平移而成的 2 维晶格，因此张量网络对于任意格矢的平移都是对称的，它实际上表示了系统趋于无穷大时的极限状态。另外由于 PEPS 是 2 维的，我们可以选择某个方向为周期边界条件而另一个方向为开边界条件^{[14]3}，其图表示类似于圆柱体，这里不再展开说明。

(2) 纠缠熵面积定律

纠缠熵面积定律适用于张量网络，但 PEPS 中的纠缠熵面积定律为 2 维形式，即任意子系统的纠缠熵正比于分割子系统的周长。若一个子系统的边界长为 L ，那么纠缠熵 $S(L) = \mathcal{O}(L \log D)$ ，其中 D 为上文中设置的键的维度，也可理解为 VBS 中设置的相邻系统之间的纠缠程度。

同 MPS 一样，PEPS 的纠缠熵面积定律保证了其有能力表示有能隙系统的

基态以及低能激发态。

(3) 关联函数多项式衰减

PEPS 与 MPS 之间一个很大的不同便是 PEPS 中两点之间的关联是随距离多项式衰减的，而上文的 MPS 是指数衰减的。仅仅这一项不同就使得 PEPS 具有了 MPS 不同的性质：从原理上来说 PEPS 不仅可以描述有能隙态，还可以描述临界态。呈多项式衰减的关联是临界系统的一大特征，因为临界系统具有标度不变性，其关联长度可以趋于无穷大^{[11]138}。上文所说的 MPS 无法描述临界系统便是因为其关联长度永远是有限的，无法具有临界系统的特征。

(4) 精确缩并复杂度

不论我们用什么顺序计算两个 PEPS 之间的缩并，其时间复杂度均是 $\mathcal{O}(\exp(N))$ 的^{[11]139}。这一论断可被数学精确证明，它告诉我们尝试计算两个 PEPS 之间的缩并是低效的，对大系统来说更是不现实的。目前计算 PEPS 多采用近似算法，例如边界 MPO 近似^[15]等。

第三章 张量网络算法

第一节 张量网络的求基态算法

上一章的内容主要集中在张量网络的定义以及性质上，没有涉及到具体的算法；本章将基于 MPS 介绍张量网络上的常用算法，并为下一章的算法改进做铺垫。

一、虚时演化算法

一个量子系统 $|\psi\rangle$ 在能量表象下可写为

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + \dots \quad (3.1)$$

其中假设每个能量本征态有 $0 \leq E_0 \leq E_1 \leq \dots$ ， $|0\rangle$ 为其非简并基态。一般情况下我们研究的系统哈密顿量不含时，因此时间演化算符可表示为 $\hat{U} = \exp(-i\hat{H}t)$ 。利用时间演化算符可以得到任意时刻的量子态为

$$\hat{U}(t)|\psi\rangle = e^{-i\hat{H}t} (c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + \dots) \quad (3.2)$$

$$= c_0 e^{-iE_0 t} |0\rangle + c_1 e^{-iE_1 t} |1\rangle + c_2 e^{-iE_2 t} |2\rangle + \dots \quad (3.3)$$

$$= e^{-iE_0 t} \left(c_0 |0\rangle + \sum_{n=1} c_n e^{-i(E_n - E_0)t} |n\rangle \right) \quad (3.4)$$

在上述公式中，我们对其进行一个数学处理，即令 $t = -i\tau$ ， $\tau \in \mathbb{R}$ ，我们得到一个新的公式

$$\hat{U}(-i\tau)|\psi\rangle = e^{-\tau E_0} \left(c_0 |0\rangle + \sum_{n=1} c_n e^{-\tau(E_n - E_0)} |n\rangle \right) \quad (3.5)$$

在正常的时间演化中，每个能量本征态的系数在复平面内转动，呈周期性变化；而在虚时演化中，每一项均是指数衰减的。进一步的定义虚时间演化算符

$$\hat{U}_I(\tau) = \hat{U}(-i\tau) e^{\tau E_0} \quad (3.6)$$

可得量子态随时间变化

$$\hat{U}_I(\tau)|\psi\rangle = c_0 |0\rangle + \sum_{n=1} c_n e^{-\tau(E_n - E_0)} |n\rangle \longrightarrow c_0 |0\rangle \quad (t \rightarrow \infty) \quad (3.7)$$

随着长时间的演化，系统从初态逐渐演化到基态。需要注意的是 \hat{U}_I 不是酉算符，结果需要进行归一化；且初态有一个要求为 $\langle\psi|0\rangle \neq 0$ 。

在张量网络中, \hat{U} 是一个整体的算符, 也被看做一个张量, 计算其作用在张量网络上的结果时计算复杂度较高, 因此我们采用与上文同样的思想, 将其分解为小张量表示。首先我们将演化时间分割为小段

$$e^{-it\hat{H}} = \left(e^{-i\hat{H}\Delta t} \right)^N \quad (3.8)$$

另外在我们研究的体系如伊辛模型、海森堡模型中, 体系的哈密顿量均是由相邻粒子间的两体哈密顿量组成的 $\hat{H} = \hat{h}_1 + \hat{h}_2 + \dots + \hat{h}_m$. 我们利用一阶 Trotter 展开将上式展开

$$e^{-i\Delta t\hat{H}} = e^{-i\Delta t \sum_{i=1}^m \hat{h}_i} \quad (3.9)$$

$$= e^{-i\hat{h}_1\Delta t} e^{-i\hat{h}_2\Delta t} \dots e^{-i\hat{h}_m\Delta t} + \mathcal{O}(\Delta t^2) \quad (3.10)$$

之后便可以将两体算符 $\exp(-i\hat{h}_k\Delta t)$ 按一定的规则作用在 MPS 上, 而不必直接计算整个多体算符如何直接与张量网络作用。基于上面的展开式, 有两种常用的时间演化算法, 其在处理两体算符的作用方式上有不同之处, 这两种方法分别被称作 tMPS^{[16]75} 与 TEBD^[17]。在此基础上做上文提到的虚时间处理, 就可以得到对应的虚时演化求基态算法。以上算法既适用于有限系统, 又适用于有周期边界条件的一维无穷大系统, 也可以扩展至 PEPS 中。由于详细的解释算法原理篇幅过长, 且与本文相关性不大, 这里就不再深入解释。

二、变分法

求体系基态的一个经典算法便是变分法^[13]。在量子系统中, 我们已经知道系统的哈密顿量 $\hat{H} = \hat{h}_1 + \hat{h}_2 + \dots + \hat{h}_{N-1}$, 且知道系统的变分波函数的表示形式为 MPS:

$$|\varphi\rangle = \sum_{s_1 \dots s_N} \text{Tr}(\mathbf{A}^{s_1} \mathbf{A}^{s_2} \dots \mathbf{A}^{s_N}) |s_1 s_2 \dots s_N\rangle \quad (3.11)$$

那么我们便可以利用变原理求系统 E 对 $|\varphi\rangle$ 的极小值:

$$E_{\text{gs}} = \min_{|\varphi\rangle} E(|\varphi\rangle) = \min_{|\varphi\rangle} \frac{\langle \varphi | \mathbf{H} | \varphi \rangle}{\langle \varphi | \varphi \rangle} \quad (3.12)$$

其中量子态需要满足归一化条件, 因此引入拉格朗日乘子 λ , 上述问题转化为

$$\min_{|\varphi\rangle} (\langle \varphi | \mathbf{H} | \varphi \rangle - \lambda \langle \varphi | \varphi \rangle) \quad (3.13)$$

其中表示量子态的全部参数都在构成 MPS 的张量中, 因此 $|\varphi\rangle$ 是 $\{\mathbf{A}^{s_i}\}$ 的函数。我们令

$$\frac{\partial}{\partial A_{l,r}^{s_i}} (\langle \varphi | \mathbf{H} | \varphi \rangle - \lambda \langle \varphi | \varphi \rangle) = 0 \quad (3.14)$$

这是一个关于所有矩阵元的方程，联立便可解得系统的基态。

通常情况下，关于矩阵元 $A_{l,r}^{s_i}$ 的方程是高度非线性的，且自变量均耦合在一起，因此即使借助计算机的力量我们也很难直接解得基态。对于这种难以直接求解的方程组，一个解决办法便是使用迭代法。结合我们的 MPS 形式，我们可以使用这样一种迭代法：我们先把第一个格点对应的矩阵 A^{s_1} 当作自变量，其他格点的矩阵看做已知量，这样一来我们的方程中做多出现未知数的平方项，问题便成为了一个容易求解的二次型问题。我们求解出 A^{s_1} 后，再用同样的方法求解 A^{s_2} ，并进一步求解其他格点对应的矩阵。我们每求出一个新的解，新的系统能量便会下降一点，通过不断的迭代使得系统的能量收敛，我们便得到了系统的基态能量与基态波函数。此方法同样可以应用到 PEPS 上。

三、梯度下降法

对于 PEPS 的严格缩并的时间复杂度是随键的维度 D 指数上升的，因此优化 PEPS 波函数一直存在相当的难度。人们提出了很多近似的缩并方法，使得时间复杂度降低为多项式级，但仍至少有 $\mathcal{O}(D^{10})$ 。如此大的指数使得在实际的计算中时间复杂度随设定的 D 的大小仍快速增大，在实际的计算中我们仍然只能取较小的 D 值。为了减少计算复杂度，人们提出了 SU 方法，其使得计算复杂度减小至 $\mathcal{O}(D^5)^{[18]}$ ，但由于计算时对目标张量周围的环境过度的简化，其计算精度大大下降。

计算 PEPS 的缩并来得到物理量的方法，都需要在计算开销与计算精度之间做取舍，为此人们又想出了蒙特卡洛抽样算法，这是一种利用抽样来计算物理量的方法。随机梯度下降法结合蒙特卡洛抽样算法——一种以放弃确定解为代价大大提高缩并计算速度的方法，成为可以提高计算速度，且可以大规模并行的方法。这里我们仍先利用 MPS 简化问题，介绍梯度优化方法。

1. 蒙特卡洛采样

在梯度优化法之前，人们就已经发明了蒙特卡洛采样技术，采样法通过对随机的自旋构型进行抽样，代替了对整个系统的遍历操作。通过在张量网络中引入蒙特卡洛抽样^[18-20]，我们可以将与严格缩并有关的计算复杂度大大简化。

在具有时间反演对称性的系统中(通常为无磁性系统)，张量网络可以只用实数表示。若我们有实数域中 MPS 表示的量子态

$$|\varphi\rangle = \sum_{s_1 \cdots s_N} \text{Tr}(A^{s_1} A^{s_2} \cdots A^{s_N}) |s_1 s_2 \cdots s_N\rangle \quad (3.15)$$

其能量表达式为

$$E = \frac{\langle \varphi | \mathbf{H} | \varphi \rangle}{\langle \varphi | \varphi \rangle} \quad (3.16)$$

我们可以在其中插入单位算符 $\mathbf{I} = \sum_S |S\rangle \langle S|$, 得到

$$\begin{aligned} E &= \frac{\sum_{S,S'} \langle \varphi | S' \rangle \langle S' | \mathbf{H} | S \rangle \langle S | \varphi \rangle}{\sum_S \langle \varphi | S \rangle \langle S | \varphi \rangle} \\ &= \frac{\sum_{S,S'} W(S') \langle S' | \mathbf{H} | S \rangle W(S)}{W(S)^2} \\ &= \frac{1}{Z} \sum_S W(S)^2 E(S) \end{aligned}$$

得到配分函数的形式。其中

$$Z = \sum_S W(S)^2 \quad (3.17)$$

$$W(S) = \langle S | \varphi \rangle = \text{Tr}(\mathbf{A}^{s_1} \mathbf{A}^{s_2} \dots \mathbf{A}^{s_N}) \quad (3.18)$$

$$E(S) = \sum_{S'} \frac{W(S')}{W(S)} \langle S' | \mathbf{H} | S \rangle \quad (3.19)$$

这里 $|S\rangle$ 指的是自旋构型 $|s_1 s_2 \dots s_N\rangle$ 中的一个构型, 对 S, S' 等指标求和时要遍历所有自旋构型。下文中我们将 $\mathbf{A}_m^{s_m}$ 的矩阵元 $(\mathbf{A}_m^{s_m})_{l,r}$ 简记为 $A_{lr}^{s_m}$

2. 计算能量梯度

计算能量 E 对矩阵元的导数^{[19]3}, 暂时略去 A 的指标, 可得

$$\begin{aligned} \frac{\partial E}{\partial A_{lr}^{s_m}} &= \frac{\partial}{\partial A} \left(\frac{1}{Z} \sum_S W(S)^2 E(S) \right) \\ &= \frac{1}{Z} \frac{\partial}{\partial A} \left(\sum_S W(S)^2 E(S) \right) + \frac{\partial}{\partial A} \left(\frac{1}{Z} \right) \sum_S W(S)^2 E(S) \\ &= \frac{1}{Z} \sum_S 2W(S) \frac{\partial W(S)}{\partial A} E(S) - \frac{1}{Z^2} \frac{\partial Z}{\partial A} \sum_S W(S)^2 E(S) \\ &= 2 \frac{1}{Z} \sum_S W(S)^2 \left(\frac{1}{W(S)} \frac{\partial W(S)}{\partial A} \right) E(S) \\ &\quad - \frac{1}{Z^2} \frac{\partial}{\partial A} \left(\sum_S W(S)^2 \right) \sum_S W(S)^2 E(S) \\ &= 2 \frac{1}{Z} \sum_S W(S)^2 \Delta(S) E(S) \\ &\quad - 2 \frac{1}{Z^2} \sum_S W(S)^2 \Delta(S) \sum_S W(S)^2 E(S) \\ &= 2 \langle \Delta_{lr}^{s_m}(S) E(S) \rangle - 2 \langle \Delta_{lr}^{s_m}(S) \rangle \langle E(S) \rangle \end{aligned}$$

即

$$\frac{\partial E}{\partial A_{lr}^{s_m}} = 2\langle \Delta_{lr}^{s_m}(S)E(S) \rangle - 2\langle \Delta_{lr}^{s_m}(S) \rangle \langle E(S) \rangle \quad (3.20)$$

上式中的 $\langle \dots \rangle$ 表示以 $W(S)^2$ 为权重的蒙特卡洛采样得到的期望值，其中矩阵

$$\Delta^{s_m}(S) = \frac{1}{W(S)} \frac{\partial W(S)}{\partial A^{s_m}} = \frac{1}{W(S)} \text{Tr} (A_1^{s_1} \cdots A_{s_{m-1}}^{s_{m-1}} A_{m+1}^{s_{m+1}} \cdots A_N^{s_N}) \quad (3.21)$$

得到梯度后，便可以根据一般的梯度下降算法，对波函数进行优化

$$A_{lrud}^{s_m}(n+1) = A_{lrud}^{s_m}(n) - p \cdot \delta t(n) \cdot \text{sign} \left(\frac{\partial E}{\partial A_{lrud}^{s_m}} \right) \quad (3.22)$$

其中 $\delta t(n)$ 为第 n 步的步长。这里 $p \in [0, 1]$ 为随机变量，只利用梯度的符号，是考虑到在局域极值较多时，一定程度的随机性可以使波函数跳出局域极值，是经过人们测试得到的经验做法。

对于对 $W(S)^2$ 的抽样，我们可以将其当作一个体系的配分函数，利用马尔科夫链上的细致平衡条件做重要抽样，得到由自旋构型组成的马尔科夫链，其自旋构型的分布自然地满足 $W(S)^2$ 的广义概率分布。

借助上面的表达式，我们对能量梯度的计算就变为对抽样得到的随机构型下矩阵的缩并，即计算一系列矩阵 $A_m^{s_m}$ 的乘积。以上的推导在 PEPS 中仍然成立，形式上只需将矩阵 $A_{lr}^{s_m}$ 变为张量 $A_{lrud}^{s_m}$ 即可。在计算的时间复杂度上，计算单层张量网络的缩并时 MPS 与矩阵乘法相同，为 $\mathcal{O}(D^3)$ ；而 PEPS 则为 $\mathcal{O}(D^6)$ 。通常情况下，梯度优化法作为一个全局优化算法，用来优化由虚时演化等方法得到的粗略的基态波函数，得到更精确的解。

第二节 梯度下降算法的优化

一、随机梯度下降

在张量网络中，我们为了更高效的计算出基态，采用了与蒙特卡洛抽样算法结合的梯度下降算法，其本质上为随机梯度下降算法^{[21][22]}

现假设我们的待下降函数 (损失函数) 具有以下形式

$$R(w) = E[f(w); \xi] \quad (3.23)$$

其可能的最小值为 R^* . 定义样本均值为

$$R_n(w) = \frac{1}{n} \sum_{i=1}^n f_i(w) \quad (3.24)$$

算法 3.1 随机梯度下降 (Stochastic Gradient Descent, SG/SGD) 算法**input :** 待下降函数 $f(w_k)$, 梯度 $g(w_k, \xi_k)$, 初态 w_1 **output:** 末态 w^* 以及函数值 $f(w^*)$

```

1 初始化随机数, 输入初态  $w_1$ ;
2 for  $k = 1, 2, \dots$  do
3   生成随机数  $\xi_k$ ;
4   计算随机梯度  $g(w_k, \xi_k)$ ;
5   选择步长  $\alpha_k \geq 0$ ;
6   将新的状态设置为  $w_{k+1} \leftarrow w_k - \alpha_k g(w_k, \xi_k)$ .
7 end

```

其中我们在一次抽样中获得样本随机数 $\{\xi_{[i]}\}_{i=1}^n$, 第 i 个样本为 $f_i(w) = f(w; \xi_{[i]})$ 。如果我们不使用随机梯度下降, 而是使用通常的全梯度下降, 每次计算出函数精确的梯度, 即 $g(w) = \nabla R(w)$, 那么存在一个常数 $\rho \in (0, 1)$ 使得算法的收敛速度为

$$R_n(w_k) - R_n^* \leq \mathcal{O}(\rho^k) \quad (3.25)$$

在最差的情况下, 将误差减小至给定的 $\epsilon > 0$ 的工作量为 $\mathcal{O}(n \log(1/\epsilon))$ 。

如果使用随机梯度下降, 则收敛速度相比之下没有使用精确梯度时那样快

$$\mathbb{E}[R_n(w_k) - R_n^*] = \mathcal{O}\left(\frac{1}{k}\right) \quad (3.26)$$

$$\mathbb{E}[R(w_k) - R^*] = \mathcal{O}\left(\frac{1}{k}\right) \quad (3.27)$$

并且需要步长 $\alpha_k = \mathcal{O}(1/k)$ 。但是由于每次计算梯度时的开销是随机抽样得到, 当系统规模增大时计算梯度的开销往往不发生变化, 因此将误差减小至指定 ϵ 时的工作量是 $\mathcal{O}(1/\epsilon)$ 的, 随机梯度下降算法在系统规模较大时往往具有更大的优势。

对强凸函数条件下随机梯度下降法的分析^{[21][27]}指出, 该算法在最小值附近会有一定的不确定性, 损失函数不能完全收敛到最小值

$$\mathbb{E}[R(w_k) - R^*] \leq KM + \mathcal{O}(\rho^k) \quad (3.28)$$

其中 M 为算法中梯度函数的方差的最小上界, 在最小值 w^* 邻域内, $M = \text{Var}_\xi(g(w^*; \xi))$ 仍不为零。当我们需要比较精确的最小值时, 随机梯度下降牺牲准确度来提高每一步计算速度的方法就会适得其反。

二、方差缩减算法

是否存在这样的算法，使得其在节省计算开销的同时保证结果收敛至最小值？实际上答案是肯定的，这需要对算法进行一定的改进。

在公式 (3.28) 中，随机梯度下降算法在算法刚开始进行时使得损失函数能够充分下降，但在接近收敛时，由于方差的上界 M 始终不为零，算法的结果与真实最小值之间便存在一个无法缩小的区间，我们无法证明函数值与真实最小值的差落入区间后还能继续缩小。我们希望构造这样一种算法，其使得梯度 $g(w_k; \xi_k)$ 的方差能够随着 $|w_k - w^*|^2$ 的缩小也不断缩小至零，这样我们便可以将 M 缩小至零，具有此性质的算法也可以保证最终结果收敛至最小值。

1. SVRG 算法

SVRG(Stochastic Variance Reduced Gradient)^[22] 算法对原始的随机梯度下降进行了改进，在保持了每一步计算开销较小的同时保证了损失函数收敛于其最小值。其主要特征与其名字相同，便是在算法进行的过程中，损失函数的梯度的方差会逐渐收敛至零。当算法的方差能够收敛至零时，其函数值则因为上文中公式的限制也被控制到趋向于零。

算法 3.2 SVRG 算法

input : 待求问题 P , 更新频率 m 与学习速率 η

output: 末态 w^* 以及函数值 $P(w^*)$

```

1 初始化随机数, 输入初态  $\tilde{w}_0$ ;
2 for  $s = 1, 2, \dots$  do
3      $\tilde{w} = \tilde{w}_{s-1}$ ;
4      $\tilde{\mu} = \frac{1}{n} \sum_{i=1}^n \nabla \psi_i(\tilde{w})$ ;
5      $w_0 = \tilde{w}$ ;
6     for  $t = 1, 2, \dots, m$  do
7         随机选取  $i_t \in \{1, \dots, n\}$ ;
8          $g_{t-1} = \nabla \psi_{i_t}(w_{t-1}) - \nabla \psi_{i_t}(\tilde{w}) + \tilde{\mu}$ ;
9         更新  $w_t = w_{t-1} - \eta g_{t-1}$ ;
10    end
11     $\tilde{w}_s = w_m$ 
12 end
```

对于问题

$$\min P(w), \quad P(w) = \frac{1}{n} \sum_{i=1}^n \psi_i(w) \quad (3.29)$$

SVRG 中构造出了新的梯度函数, 新的算法由双重循环构成:

为了说明在新的梯度更新规则下, 梯度函数的方差趋向于零, 我们首先指出 \tilde{w} 与 w_t 均趋向于一个相同的数 w^* , 此时有 $\tilde{\mu} \rightarrow 0$. 当函数接近收敛时, $\tilde{w} \rightarrow w_t$, 由函数的连续性得 $\psi_i(w_t) \rightarrow \psi_i(\tilde{w})$. 因此上文中 g_t 将收敛至零。显然当 w_t 收敛至 w^* 时, g_t 的涨落也将随自身数值趋于零而趋于零, 因此 $M \rightarrow 0$. 在公式 (3.28) 的控制下, 函数值将严格收敛至最小值。

在一般情况下, 当我们的参数选择合适时, 达到指定精度的步数为 $\log(1/\epsilon)$, 且在每个内循环中, 梯度的计算与 SGD 相当, 因此 SVRG 能够更高效的下降, 即与全梯度下降具有相同收敛速率, 且能达到更高的精度。但是 SVRG 也不是全方面优于 SGD, 在某些情况下 SVRG 计算一次全梯度的开销很大, 这时进行一次完整的 SVRG 迭代的工作量已经可以完成多次 SGD 从而使得损失函数下降的更多。

2. 其他算法

除了 SVRG 算法, 还有 SAG^[23], SAGA^[24-25] 等算法也采用了相同的思想, 即在算法迭代的过程中控制梯度的方差趋于零。这些算法也具有相同的收敛速率与精度, 但相比之下 SVRG 更适合应用到张量网络上。上文只着重介绍了 SVRG 算法, 是为了下文中尝试将其应用至张量网络的梯度优化算法, 也许其他梯度下降法中仍有许多深刻的原理可以应用, 这需要大家进一步的研究。

三、张量网络与 SVRG

在上节介绍的张量网络梯度优化算法中, 给定张量的梯度可以抽象为 $g(w_k; \xi_k)$, 这里的 ξ_k 代表第 k 次抽样时得到的马尔科夫链对应的随机自旋样本。我们写出 SVRG 的形式

$$g(w_k; \tilde{w}, \xi) = \nabla \psi_\xi(w_k) - \nabla \psi_\xi(\tilde{w}) + \nabla R(\tilde{w}) \quad (3.30)$$

其中的 $R(w)$ 函数要保证在 $w \rightarrow w^*$ 时方差为零, $\psi(w_k)$ 是连续的以保证 $w_k \rightarrow w$ 时 $\psi(w_k) \rightarrow \psi(w)$.

1. 算法改造

本文构造 $\nabla R(w)$ 为张量网络的能量函数, $\nabla \psi_\xi(w_k)$ 为系统在 w_k 时 ξ 对应的马尔科夫链上的能量梯度。这里的 ξ 不再表示哪个 ψ_i , 而是表示由 ξ 对应的整

个马尔科夫链上的数据集，类似于 SCSG^[26] 算法中每次迭代中使用的 Batch。为了保证公式中两个 ψ 函数相同，它们应使用相同的马尔科夫链 ξ 来计算能量。当系统处于 \tilde{w} 态时，若我们抽样得到了马尔科夫链 ξ ，则在 w_k 时直接使用这个马尔科夫链得到的样本没有物理意义，这是因为张量网络产生变化，系统的量子态在不同自旋分量上的投影发生改变，原先的马尔科夫链不满足新的细致平衡条件。

为了使原先概率分布下得到的马尔科夫链可以应用至新概率分布下得到有意义的物理量，我们介绍一种被称为 reweighting 的技术。假设一个物理量 A 的每个状态 A_i 出现的广义概率为 ρ_i ，则将其期望值在一个广义概率为 ρ' 的分布中表示的结果为

$$\begin{aligned}\langle A \rangle_\rho &= \frac{\sum_i \rho_i A_i}{\sum_i \rho_i} \\ &= \frac{\sum_i \frac{\rho_i}{\rho'_i} A_i \rho'_i}{\sum_i \rho'_i} \bigg/ \frac{\sum_i \frac{\rho_i}{\rho'_i} \rho'_i}{\sum_i \rho'_i} \\ &= \left\langle \frac{\rho}{\rho'} A \right\rangle_{\rho'} \bigg/ \left\langle \frac{\rho}{\rho'} \right\rangle_{\rho'}\end{aligned}$$

上式中的 ρ_i 在张量网络中即为 $W(S_i)^2$ 。想求得在 w_k 对应的物理量，只需将 \tilde{w} 处算出的 $W(S)^2$ 作为 ρ' ， w_k 处的作为 ρ ，重新计算公式 (3.20) 即可。

计算 $g(w_k; \tilde{w}, \xi)$ 的期望值，

$$\begin{aligned}E_\xi(g(w_k; \tilde{w}, \xi)) &= E(\nabla \psi_\xi(w_k)) - E(\nabla \psi_\xi(\tilde{w}) - \nabla R_n(\tilde{w})) \\ &= E(\nabla \psi_\xi(w_k)) - 0 \\ &= \nabla R(w_k)\end{aligned}$$

是能量梯度的无偏估计，符合一般随机梯度下降算法的要求。

形式上来看，SVRG 算法很容易应用至张量网络的梯度优化算法中。但实际上，因为加入了马尔科夫抽样，为了保证 $\psi_\xi(\tilde{w})$ 与 $\psi_\xi(w_k)$ 中两个函数 $\psi_\xi(\cdot)$ 一致，必须两次都选取相同的样本，而随着状态 \tilde{w} 因为梯度下降法的迭代变动至 w_k 时，原先的马尔科夫链已经不适用，所以这里必须利用 reweighting 或类似方法对原先马尔科夫链的样本进行转换。

第四章 数值实验

第一节 算法测试结果

一、复现 SVRG

为保证对论文 [22] 的理解正确以及不遗漏技术细节，这里本文使用 Julia 编程语言复现了论文中对 SVRG 与 SGD 的性能测试。

所有测试中损失函数表达式均为

$$P(w) = \frac{1}{n} \sum_{i=1}^n f_i(w) \quad (4.1)$$

$f_i(w)$ 为随机生成的 w 的正定二次型，保证论文中强凸条件以及莱布尼兹连续条件被满足。

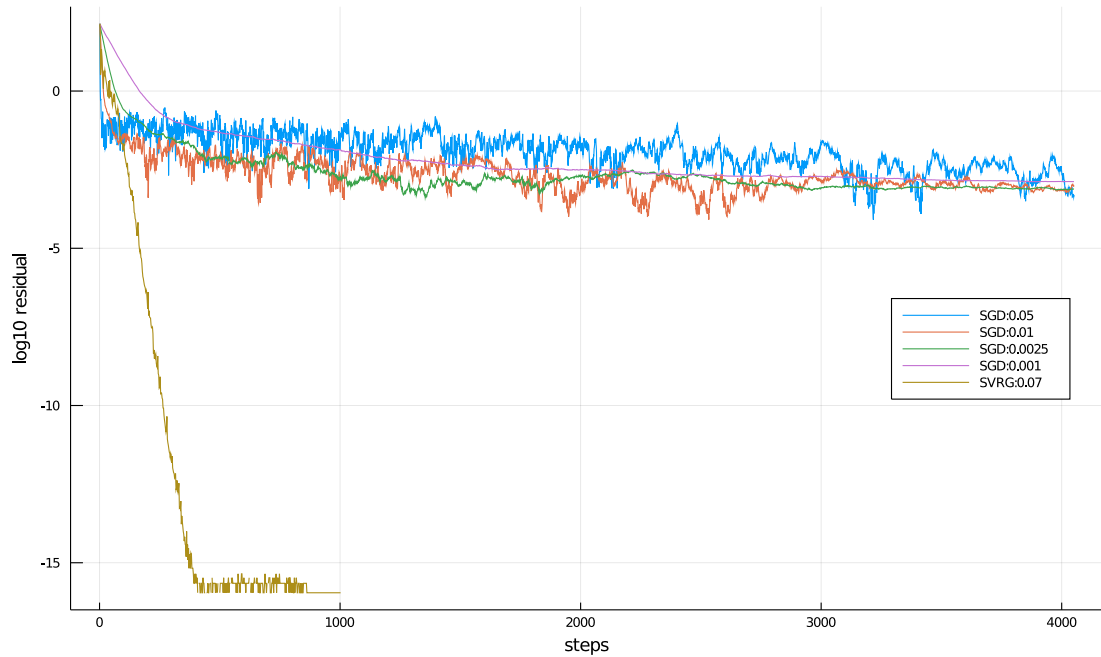


图 4.1 SGD 与 SVRG 性能测试

注：图中复现了 SVRG 原始论文中的算法性能比较^{[22]7}，图例中前 4 条为 SGD 在最优学习速率附近不同值处的表现，最后一条为 SVRG 接近最优参数下的一个结果，数字表示步长 (学习速率)。图中对 SVRG 数据的横坐标进行了 $\frac{L_0 + L_1 m}{m L_{\text{SGD}}} \approx 1.25$ 的拉伸，保证相同横坐标下每一步平均使用的马尔科夫链长相等。

在图4.1中 SVRG 的表现远优于基础的 SGD 算法，且与原论文中测试结果一致，即论文中的 SVRG 在面对极值问题时的性能远优于随机梯度下降，且线性

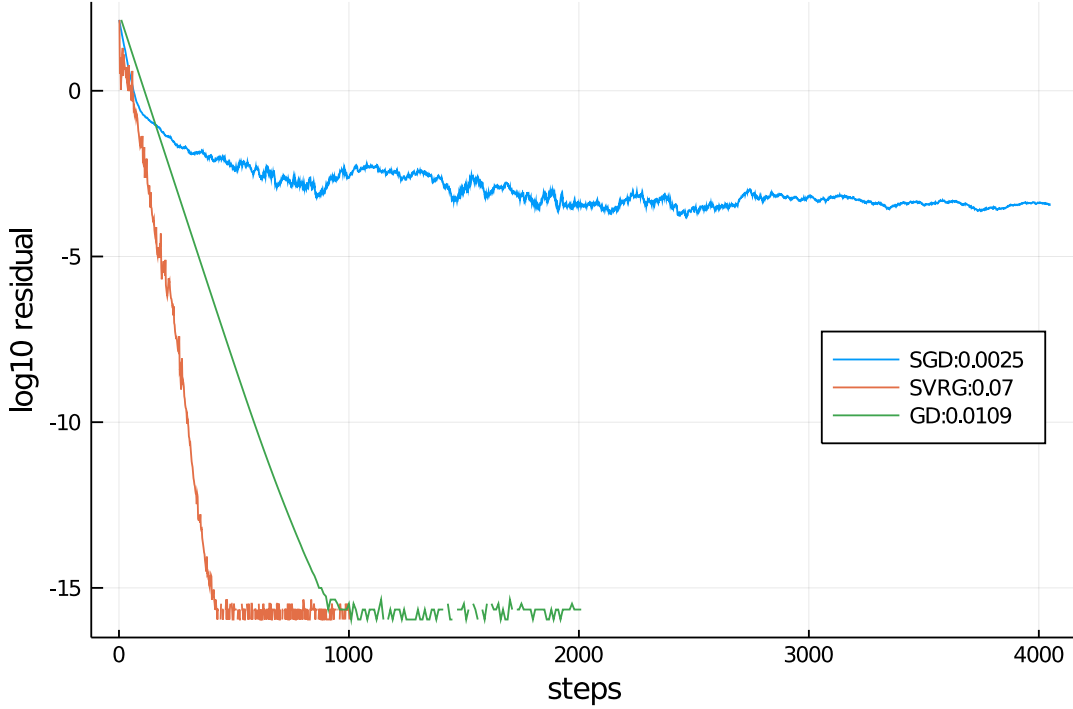


图 4.2 SGD、SVRG 与 GD 性能测试

注：接上图，加入了与常用无随机性梯度下降 (GD) 的比较，其横坐标被扩大了约 1.25 倍以保证相同横坐标下每一步平均使用的马尔科夫链长相等。等工作量下最佳 SVRG 优于最佳全梯度下降。

收敛。实验中，随机梯度下降为了保证结果的收敛，必须使步长随迭代次数 k 做 $\mathcal{O}(1/k)$ 的缩小，并且结果的收敛速度在对数图中不断变慢。SVRG 算法中步长始终是不变的，保证了对数图中常数下降速度，符合收敛速度 $\mathcal{O}(\rho^k)$ 的理论预测。

二、算法改造

张量网络的梯度优化法中的随机梯度下降与原始版本不同，加入了马尔科夫链上的重要抽样方法，因此我们将其改造为 SVRG 时也要做适当的改进，利用上一章中的 reweighting 方法，我们将 SVRG 与马尔科夫抽样方法结合，将算法 3.2 改造为算法 4.1 并测试其有效性。

与 SGD 相比，SVRG 每一步的平均开销为 $(L_{\text{outer}} + 2mL_{\text{inner}})/m$ ，若 SGD 中使用的马尔科夫链长为 L ，则两种算法每走一步，其理想情况下计算开销之比为

$$\gamma = \frac{L_{\text{outer}} + 2mL_{\text{inner}}}{mL} \quad (4.2)$$

为了反映等工作量前提下两种算法的性能，本章所有数据涉及到 SGD 与 SVRG 的能量随步长变化曲线的，SVRG 数据的横坐标均乘以 γ 以保证横坐标相同时其

算法 4.1 马尔科夫链上的 SVRG 算法

input : 待求问题 $R = E$, 更新频率 m 与学习速率 η

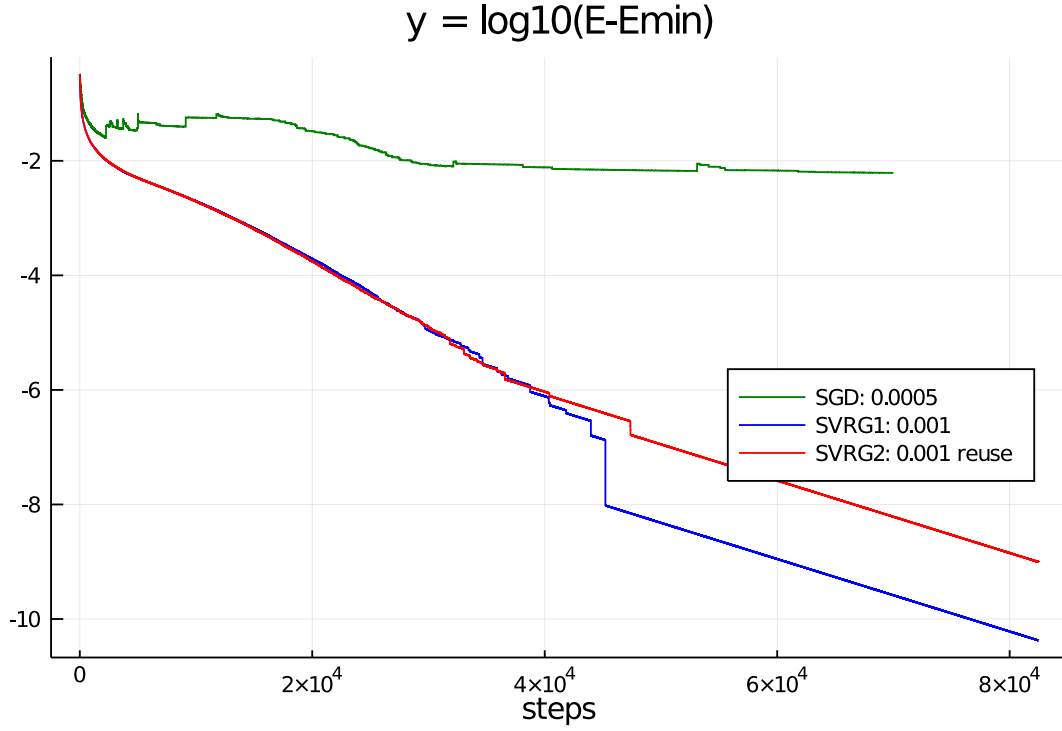
output: 末态 w^*

- 1 初始化随机数, 输入初态张量网络 \tilde{w}_0 ;
- 2 **for** $s = 1, 2, \dots$ **do**
- 3 $\tilde{w} = \tilde{w}_{s-1}$;
- 4 $\tilde{\mu} = \nabla R(\tilde{w}) = \nabla \left(\frac{1}{Z} \sum_S W_{\tilde{w}}(S)^2 E(S) \right)$;
- 5 $w_0 = \tilde{w}$;
- 6 由 \tilde{w} 生成马尔科夫链, 样本集合为 \mathcal{L} , 样本数 $|\mathcal{L}| = L_{\text{outer}}$;
- 7 **for** $t = 1, 2, \dots, m$ **do**
- 8 选择一 (SVRG1): 无视 \mathcal{L} , 生成新马尔科夫链 ξ 使得 $|\xi| = L_{\text{inner}}$;
- 9 选择二 (SVRG2): 随机选取 $\xi \in \mathcal{L}$ 且 $|\xi| = L_{\text{inner}}$;
- 10 利用公式 (3.20) 计算样本 ξ 下能量梯度 $\nabla \psi_\xi$ 的值;
- 11 $g_{t-1} = \nabla \psi_\xi(w_{t-1}; \tilde{w}) - \nabla \psi_\xi(\tilde{w}) + \tilde{\mu}$;
- 12 更新 $w_t = w_{t-1} - \eta g_{t-1}$;
- 13 **end**
- 14 $\tilde{w}_s = w_m$
- 15 **end**

工作量与 SGD 一致。

为了避免在大规模服务器上做测试, 本文采用了最简单的张量网络, 即长度为 1 的 MPS, 只有一个一阶张量, 回到了原始的 $|\varphi\rangle = \sum c_i |s_i\rangle$ 的形式, 利用最简单的计算验证新的 SVRG 算法是否可行。

结果如图 4.3 所示, SVRG 在开始时速度不稳定, 迭代一定步数后变为平稳的 $\mathcal{O}(\rho^k)$. 相比之下, 普通的随机梯度下降速度越来越落后于 SVRG. 增大态空间维度, 则使算法收敛的步长需要缩小, 如图 4.4 所示。即使加入了马尔科夫抽样与 reweighting 方法, SVRG 缩减方差的原理依然成立, 因此改造后的 SVRG 算法对于优化物理系统的能量极小值仍然可行。

图 4.3 SGD, SVRG 测试 ($d = 20$)

注：横坐标为步长，SVRG 的两条线分别代表算法4.1中两种选择，图例中数字为学习速率 η

第二节 结论与推广

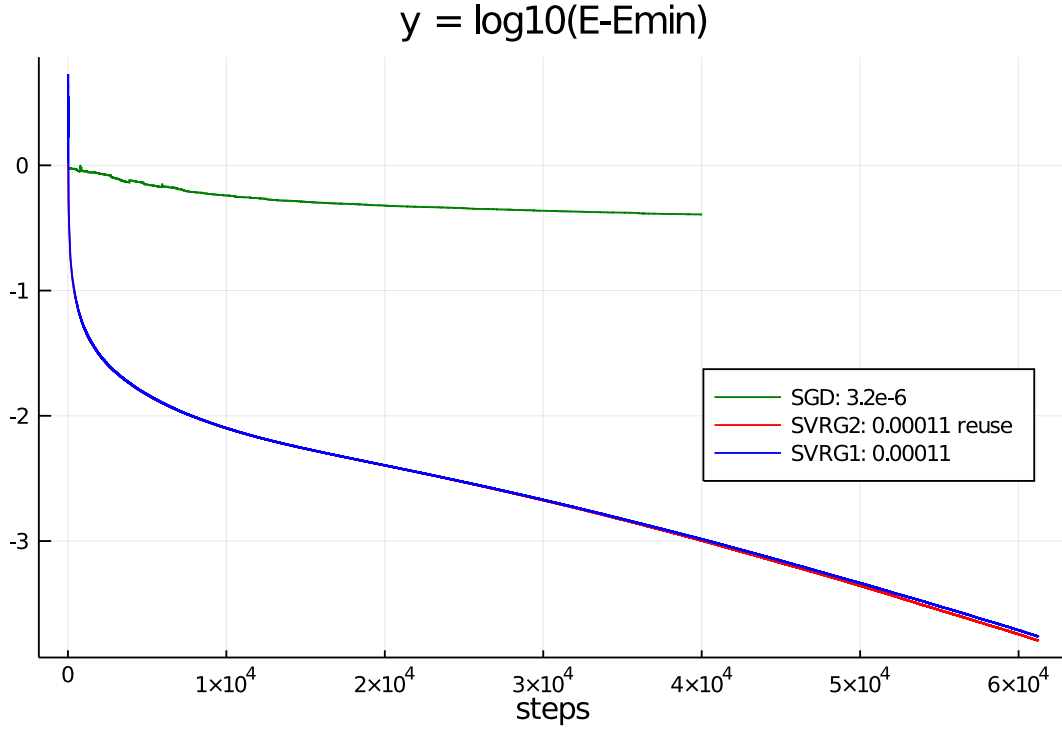
一、结论

前人通过改造 SGD 算法，使其能够使用至张量网络上，并取得了相当的成果。本文受方差缩减的随机梯度下降启发，尝试将普通 SGD 改造为 SVRG 的方法应用至张量网络的 SGD 上。在张量网络中采用 SVRG 之前，需要首先验证改造后算法的有效性，本文的数值实验中，SVRG 在相当的迭代次数后表现为理论预测的 $\mathcal{O}(\rho^k)$ 收敛，说明新的算法可以正确收敛至能量极小值，且是线性收敛的。这验证了被改造后算法的有效性，使人们能初步确定新的算法是一个正确的算法。

二、算法推广

1. 一般 MPS

正如前人将随机梯度下降结合马尔科夫链上的重要抽样方法应用于张量网络，我们将 SVRG 算法同样应用于张量网络。在 SVRG 中，如果不计算 $\psi_{\xi}(w)$ ，即

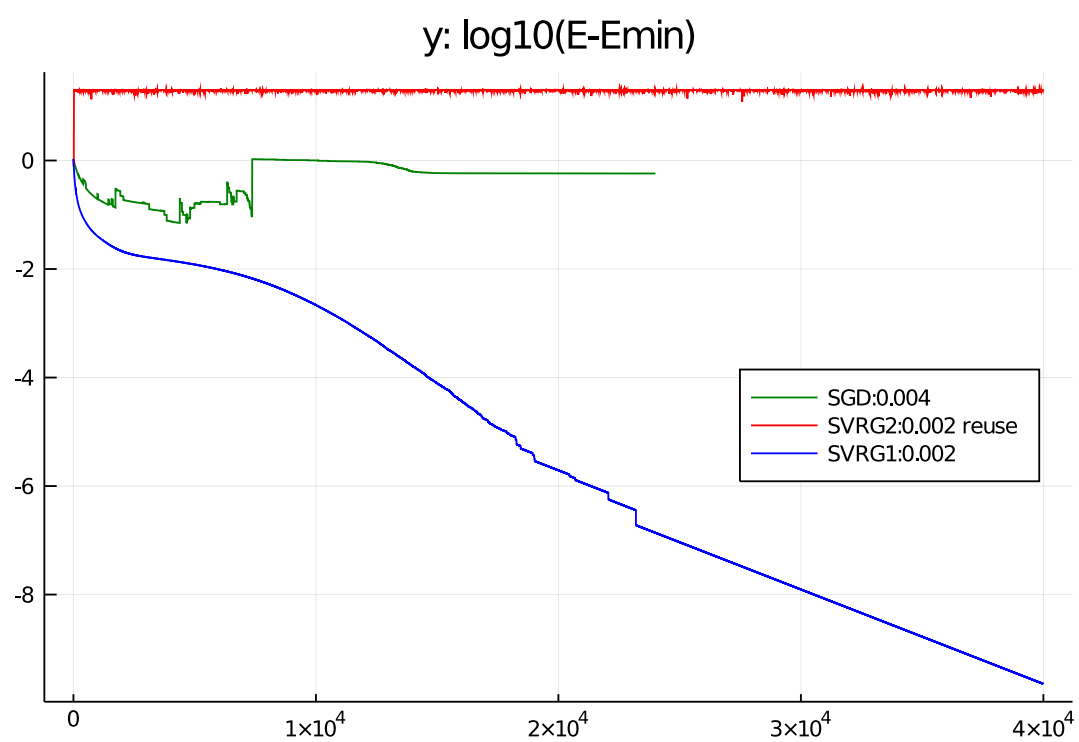
图 4.4 SGD, SVRG 测试 ($d = 200$)

注：增大态空间维度 d . SVRG 的两条线分别代表算法4.1中两种选择

马尔科夫链长为零，则我们的算法退化为全梯度下降；如果不计历史梯度 $R(\tilde{w})$ 与 $\psi_\xi(\tilde{w})$ 则退化为随机梯度下降。我们做出推测，SVRG 在长度不为一的 MPS 上，应该与 SGD 或全梯度下降在其上一样有效。同样的，SVRG 也继承了原先算法的缺点，即应使用一个接近基态的初态进行优化，比如由虚时演化得到的近似基态^[19]，否则容易如图4.5陷入局域极小值中。

2. 推广至 PEPS

由于 PEPS 与 MPS 中一个显著的不同便是 PEPS 的精确缩并时间复杂度是指数级的，而近似缩并仍然有很高的时间复杂度，因此我们无法断定 SVRG 在 PEPS 中一样有效，因为随着斯通的增大，计算一步全梯度 $R(w)$ 的开销可能会远大于方差缩减节省的开销。我们需要更多的数值实验来研究其在 PEPS 上的有效性。

图 4.5 SGD, SVRG 测试 ($d = 20$)

注：此图中的 SGD 与 SVRG2 陷入局域极小值

参 考 文 献

- [1] Vedral V. Mean-field approximations and multipartite thermal correlations. *New Journal of Physics*, 2004, 6:22-22. DOI: 10.1088/1367-2630/6/1/022.
- [2] Nightingale M, Umrigar C. Quantum monte carlo methods in physics and chemistry: volume 154. Springer, 1999.
- [3] Progress in scientific computing: Lanczos algorithms for large symmetric eigenvalue computations vol. II programs. Birkhäuser Boston, 2012.
- [4] DENG M S T K E. Density functional theory(Chinese Edition). 北京: Defense Industry Press, 2014.
- [5] Hsiao W H. Journal Club: Brief Introduction to Tensor Network. :7.
- [6] Stewart G W. On the Early History of the Singular Value Decomposition. *SIAM Review*, 1993, 35(4):551-566. DOI: 10.1137/1035134.
- [7] Eisert J, Cramer M, Plenio M B. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 2010, 82(1):277-306. DOI: 10.1103/RevModPhys.82.277.
- [8] Vidal G, Latorre J I, Rico E, et al. Entanglement in Quantum Critical Phenomena. *Physical Review Letters*, 2003, 90(22):227902. DOI: 10.1103/PhysRevLett.90.227902.
- [9] 董少钧. 张量网络算法从玻色子系统到费米子系统. 中国科学技术大学, 2017.
- [10] Bridgeman J C, Chubb C T. Hand-waving and interpretive dance: An introductory course on tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 2017, 50(22):223001. DOI: 10.1088/1751-8121/aa6dc3.
- [11] A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 2014, 349:117-158. DOI: 10.1016/j.aop.2014.06.013.
- [12] Eisert J. Entanglement and tensor network states. *arXiv:1308.3318 [cond-mat, physics:quant-ph]*, 2013.
- [13] Liu W. 二维量子多体系统的张量网络态算法 - 中国博士学位论文全文数据库. 2017.

- [14] Osorio Iregui J, Troyer M, Corboz P. Infinite matrix product states versus infinite projected entangled-pair states on the cylinder: A comparative study. *Physical Review B*, 2017, 96(11):115113. DOI: 10.1103/PhysRevB.96.115113.
- [15] Lubasch M, Cirac J I, Bañuls M C. Unifying projected entangled pair state contractions. *New Journal of Physics*, 2014, 16(3):033014. DOI: 10.1088/1367-2630/16/3/033014.
- [16] Schollwoeck U. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 2011, 326(1):96-192. DOI: 10.1016/j.aop.2010.09.012.
- [17] Vidal G. Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension. *Physical Review Letters*, 2007, 98(7):070201. DOI: 10.1103/PhysRevLett.98.070201.
- [18] Wang L, Pižorn I, Verstraete F. Monte Carlo simulation with tensor network states. *Physical Review B*, 2011, 83(13):134421. DOI: 10.1103/PhysRevB.83.134421.
- [19] Liu W Y, Dong S J, Han Y J, et al. Gradient optimization of finite projected entangled pair states. *Physical Review B*, 2017, 95(19):195154. DOI: 10.1103/PhysRevB.95.195154.
- [20] Sandvik A W. Variational Quantum Monte Carlo Simulations with Tensor-Network States. *Physical Review Letters*, 2007, 99(22). DOI: 10.1103/PhysRevLett.99.220602.
- [21] Bottou L, Curtis F E, Nocedal J. Optimization Methods for Large-Scale Machine Learning. *arXiv:1606.04838 [cs, math, stat]*, 2018.
- [22] Johnson R, Zhang T. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. :9.
- [23] Roux N L, Schmidt M, Bach F R. A Stochastic Gradient Method with an Exponential Convergence _Rate for Finite Training Sets. :9.
- [24] Hofmann T, Lucchi A, Lacoste-Julien S, et al. Variance Reduced Stochastic Gradient Descent with Neighbors. *arXiv:1506.03662 [cs, math, stat]*, 2016.
- [25] Defazio A, Bach F, Lacoste-Julien S. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. *arXiv:1407.0202 [cs, math, stat]*, 2014.
- [26] Lei L, Jordan M. Less than a Single Pass: Stochastically Controlled Stochastic

Gradient//Singh A, Zhu J. Proceedings of Machine Learning Research: volume 54
Proceedings of the 20th International Conference on Artificial Intelligence and
Statistics. Fort Lauderdale, FL, USA: PMLR, 2017: 148-156. <http://proceedings.mlr.press/v54/lei17a.html>.

附录 A 补充材料

一、代码仓库

本文中使用的论文模版、数据与 Julia 代码均上传至 Github 仓库内，大家可通过命令“`git clone git@github.com:mrsun-97/ustcthesis.git`”下载至本地 (http 协议同样支持)，后续将更新针对一般 MPS 的测试代码。

二、相关文献

本论文研究的前置工作来自于同组的董少钧师兄与刘文渊师兄的论文 [19], 本论文主要是为提高文中 SGD 性能而在 MPS 上做出一些尝试。师兄论文已随本论文一起提交至印刷厂，应已经附于本文末尾。