

基于量子行走的量子信息处理

参加人：段宇丞，刘诗洋，薛天扬

指导教师：许国富

山东大学 济南市历城区 250100

摘要：游走过程是处理信息问题的一大方法，具有广泛应用。本文中，我们首先对一维链上的Hadamard量子行走、格点环上的Hadamard量子行走进行模拟，并将其结果与相应的经典行走比较。对于格点环，我们研究了其回归周期特性。在此基础上，一方面对经典量子算法进行模拟，再对量子行走下的算法实现进行数值模拟，证明其相对经典算法依然具有显著加速（如平方级、指数级加速）；一方面使用量子行走过程加速机器学习，对若干个不同机器学习模型如kd树模型、支持向量机、玻尔兹曼机提出相应的量子处理方案。最后简要介绍了下如何提取量子行走过程中的winding number。

关键词：量子行走，Grover搜索算法，机器学习

Quantum Information Processing Based on Quantum Walk

Y.-C. Duan¹, SH.-Y. Liu², T.-Y. Xue²

Academic Advisor: G. F. Xu*

Department of Physics, Shandong University, Jinan 250100, China

Abstract: Quantum Walk is one of the methods to deal with information processing problems, which has a wide application. In this paper, we first simulate the Hadamard quantum walk on a one-dimensional chain and the Hadamard quantum walk on a lattice ring, and then compare the results with the corresponding classical walks. For lattice - point rings, the regression periodic properties are studied. On this basis, on the one hand, the classical quantum algorithm is simulated, and then the implementation of the algorithm under the quantum walk is simulated. It is proved that the algorithm still has significant acceleration compared with the classical algorithm (e.g., square-order acceleration, exponential acceleration). On the one hand, the quantum walking process is used to accelerate machine learning, and the corresponding quantum processing scheme is proposed for several different machine learning models, such as KD tree model, support vector machine and Boltzmann machine. Finally, we briefly introduce how to extract the winding number during quantum walking.

Key Words: Quantum Walk; Grover algorithm; machine learning;

1 引言

1) 开展该研究的背景及目的

2011 年来，计算机界的摩尔定律已逐渐失效。量子计算利用广大的直积空间和纠缠操作高效处理大量输入信息，与 5G、大数据、云计算、机器学习一道成为新一代工业革命的重要候选人。

硬件方面，逻辑门是计算机的关键与核心。量子逻辑门不再是用电子的宏观统计特性来表征，而是用微观粒子的个体行为状态来描述，例如电子等微观粒子的自旋、电子在不同能级上的跃迁等等。由于量子逻辑门作用在真实世界量子比特上，从而对直积空间的量子态做出指数级操作，故量子逻辑门所组成的通用逻辑门以及逻辑关系是经典逻辑门不可比拟的。

软件方面，量子算法则是融入了量子力学很多的特征，如**量子的相干性，叠加性，并行性，纠缠性，波函数塌缩**等。这些性质大大提升了计算的效率，构建出一种新型的计算模式—量子算法。对一些特殊的问题，在量子算法中能够找到有效算法，在处理“足够多”的数据量时，量子算法相对经典算法有指数级加速。

量子随机行走是实现量子算法尤其是搜索算法的一个重要工具。原则上，在一切计算问题可以被约化

为“编码”、“搜索”的今天，一个强大的搜索算法不仅能解决大多数问题，甚至能将一般的 NP 型问题转化为 P 型问题。故此，如何了解、发展和应用量子行走和量子搜索算法尤为重要。近年来，在光学系统、NMR 系统、超导平台上的量子行走实验均得到验证。它能够比经典算法更快遍历整个空间的特性深深吸引着科学工作者，在应用中正显现出越来越大的潜力。

我们的研究目的即是了解与模拟量子行走在不同情况下的特性，并对特定算法进行模拟，与经典的行走方法进行对比，从而验证其量子优越性。进一步，我们尝试将量子行走应用到机器学习过程的加速中，

2) 展示的内容

我们将会展示量子行走原理、一维量子行走和经典行走模拟、多边形量子行走和经典行走模拟、Grover 算法原理、量子行走 Grover 算法编码、Grover 算法模拟、超立方体量子行走搜索模拟、二维平面量子行走搜索模拟、kb 树搜索机器学习算法、支持向量机和 QGM 方法中的量子行走应用。

3) 研究的意义

此选题研究基于量子行走的量子逻辑门与量子算法，属于前沿研究，具有较高的科学意义。机器学习是一门多领域交叉学科，涉及概率论、统计学、凸分析、算法复杂度理论等多门学科。“人类如何学习”是一个有关自我本识的重大问题，从这个意义上说，机器学习具有一定的自然科学探索色彩。当下量子计算提升处理计算机特定问题的计算能力，一个好的量子算法无疑可以为机器学习过程研究给以巨大推动。

2 研究部分

2.1 量子行走

2.1.1 量子行走原理

量子行走是一个研究量子过程的方法，直观地讲，是将态空间的相邻位移以随机行走的方式描述。在实际操作中，许多实验方法可以使用量子行走的方法，如偏振玻片对光子的选择、晶格上的电子运动等。其基本描述方法有两种：硬币型递推描述法和傅里叶变换法；其基本模式可分为两种：连续型行走和离散型行走。在本项目中，我们着眼于格点离散行走和光量子行走，使用硬币空间、离散量子行走的研究方法。下面我们取一维行走为例，说明其基本原理如下[3]：

首先我们表示出量子态。设行走空间为 H_p ，则对一维链上的一个节点，下一时刻有向左和向右两个行走方向。则我们给 H_p 空间直积上一个硬币空间 H_c ，其中 H_c 为二维空间，基向量为 $|L\rangle, |R\rangle$ 。故完整的量子态可以写为位置态和硬币态的直积： $|\psi\rangle = |i\rangle(a|L\rangle + b|R\rangle)$ 。其中 $|i\rangle$ 表示第 i 个格点上的量子态。

其次我们表示出行走算符。每次的行走可以表示为硬币空间的一次对硬币的操作，和一个在位置空间由硬币态控制运动算符的直积。由量子门的酉性，故有三个自由变量，写出对硬币的一般化操作：

$$U_{\zeta,\xi,\theta} = \begin{pmatrix} e^{i\xi} \cos \theta & e^{i\xi} \sin \theta \\ e^{i\xi} \sin \theta & -e^{-i\xi} \cos \theta \end{pmatrix}$$

对行走操作，为硬币态作为控制位的门操作。对离散行走，令 p 为动量算符，则向左与向右位移 l 长度的算符可表示为：

$$S_L = e^{-\frac{ipl}{\hbar}} \quad S_R = e^{\frac{ipl}{\hbar}} = S_L^\dagger$$

设格点间距为 l ，则这两个算符对位置空间的作用如下：

$$S_L|i\rangle = |i-1\rangle \quad S_R|i\rangle = |i+1\rangle$$

故根据硬币态 $|L\rangle, |R\rangle$ 而移动的算符可以写为：

$$S = S_L \otimes |L\rangle\langle L| + S_R \otimes |R\rangle\langle R|$$

我们假设每次行走是在改变硬币态之后发生的。故总的量子行走过程为对 $H_p \otimes H_c$ 空间进行两次连续操作：

$$|\psi_{t+1}\rangle = [S(I_p \otimes U_{\zeta,\xi,\theta})]|\psi_t\rangle$$

这就是离散硬币量子行走的方法。

2.1.2 一维量子行走模拟

例如对于 $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle(|L\rangle + i|R\rangle))$, 使用 python 模拟在硬币算符为 Hadamard 算符 $U_{\zeta,\xi,\theta} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ 情

况下行走 100 步后的位置分布（图 1）：

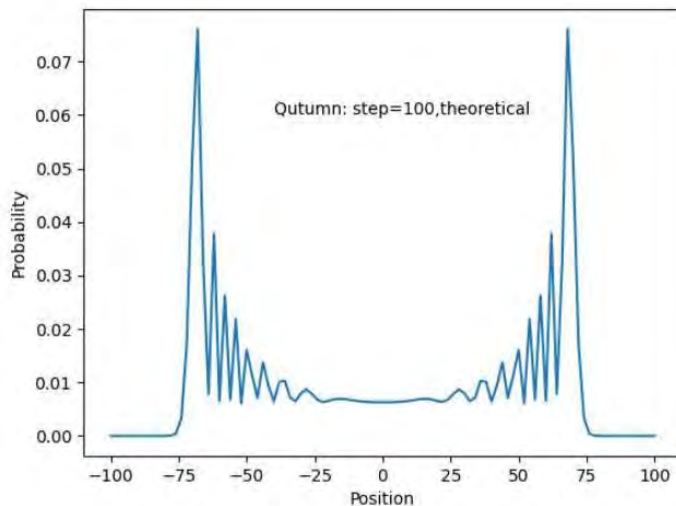


图 1. 100 步一维量子行走的概率分布图

而 100 步经典随机行走的模拟结果为（图 2）：

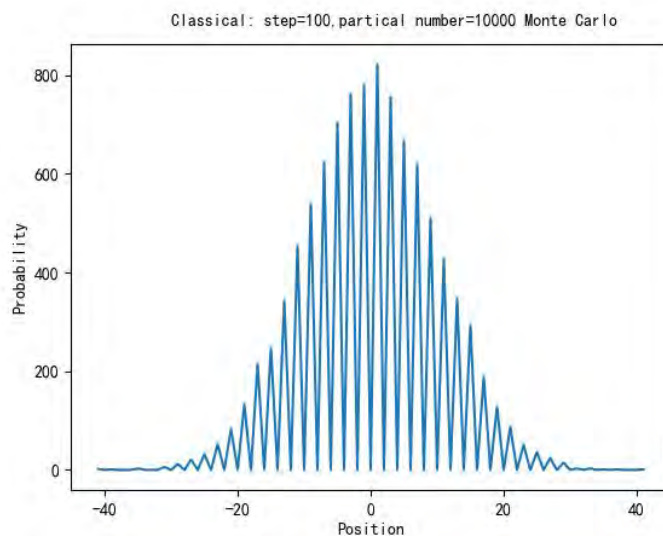


图 2. 100 步一维经典行走的概率分布图

与经典随机行走的高斯分布不同，可以看出在 Hadamard 量子行走下的末态横向延展更长。使用同样的方法，我们可以研究在一般的无向图中的量子行走。已经证明，在二叉树上的量子行走能够比经典游走更快遍历整棵树（指数加速）。

2.1.3 多边形量子行走模拟

多边形量子随机行走有与经典不同的现象。对奇数多边形，我们将量子态从 0 点释放，再时刻探测各个节点的波函数值而作图（图 3）。在这些图表中，对 0 点波函数进行傅里叶变换，发现频域存在明显峰，说明多边形量子随机行走具有一定回归性。

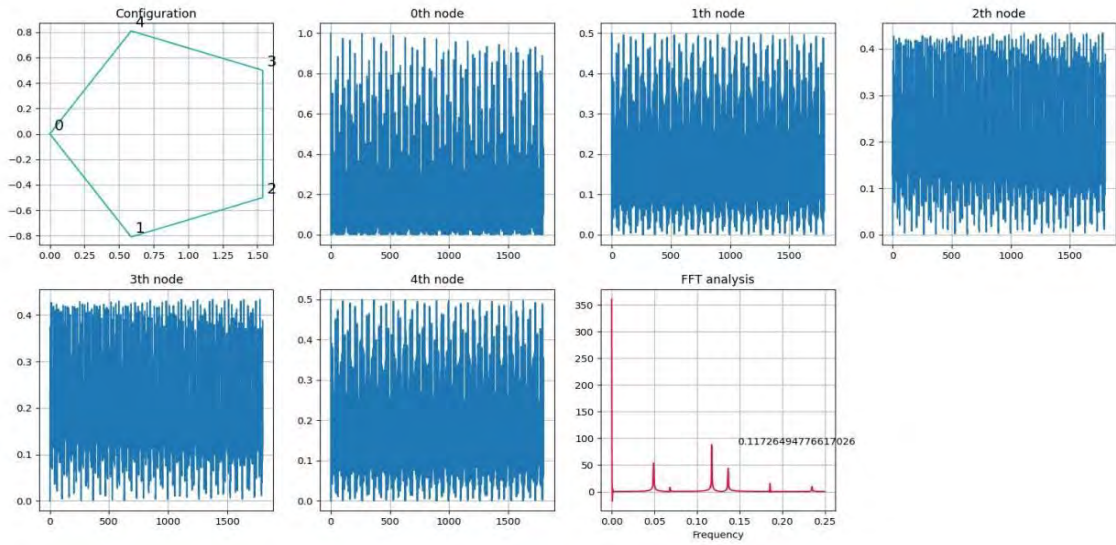


图 3. 奇数多边形量子行走的概率分布图及 FFT

对偶数多边形（图 4）：

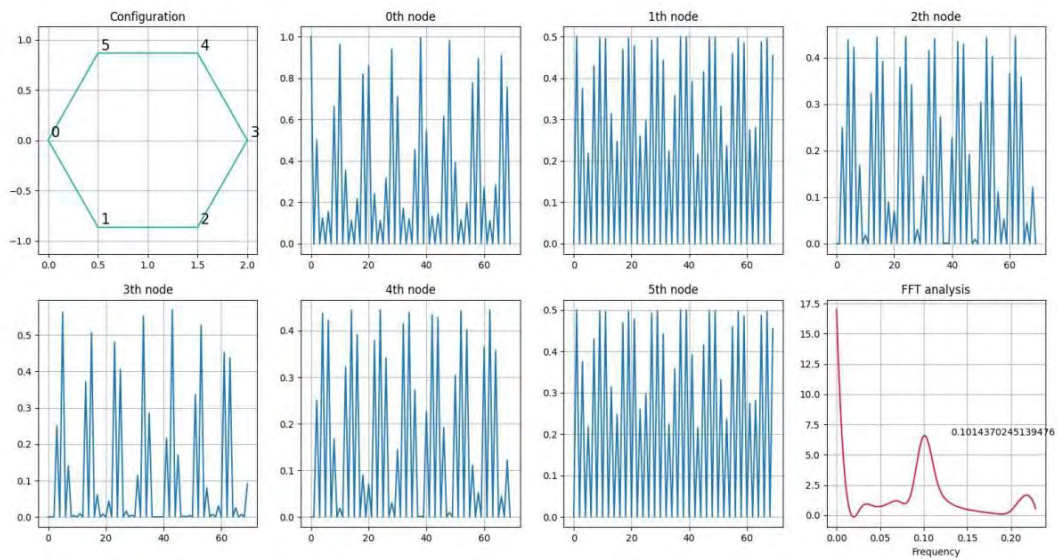


图 4. 偶数多边形量子行走的概率分布图及 FFT

也可以看出明显的震荡。

从多边形的一角出发，在多边形的角上进行一维经典随机行走，随步数增加，也可以得出多边形所有角的概率变化（图 5，图 6）。我们发现，**奇数多边形概率随步数增加会收敛，偶数多边形会逐渐均匀震荡**（其实很好理解）：

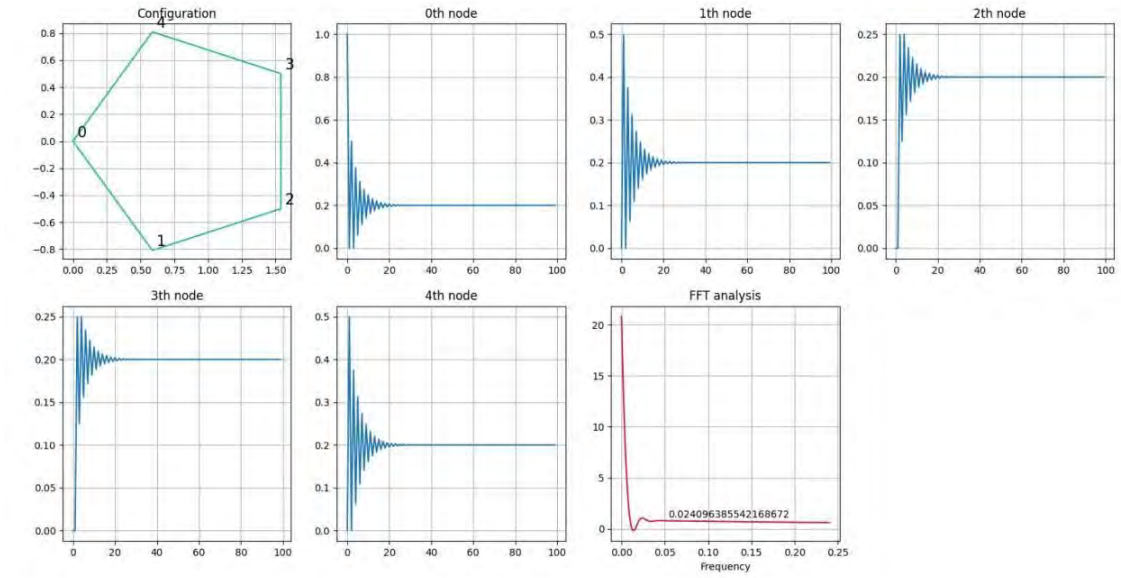


图 5. 奇数多边形经典的概率分布图及 FFT

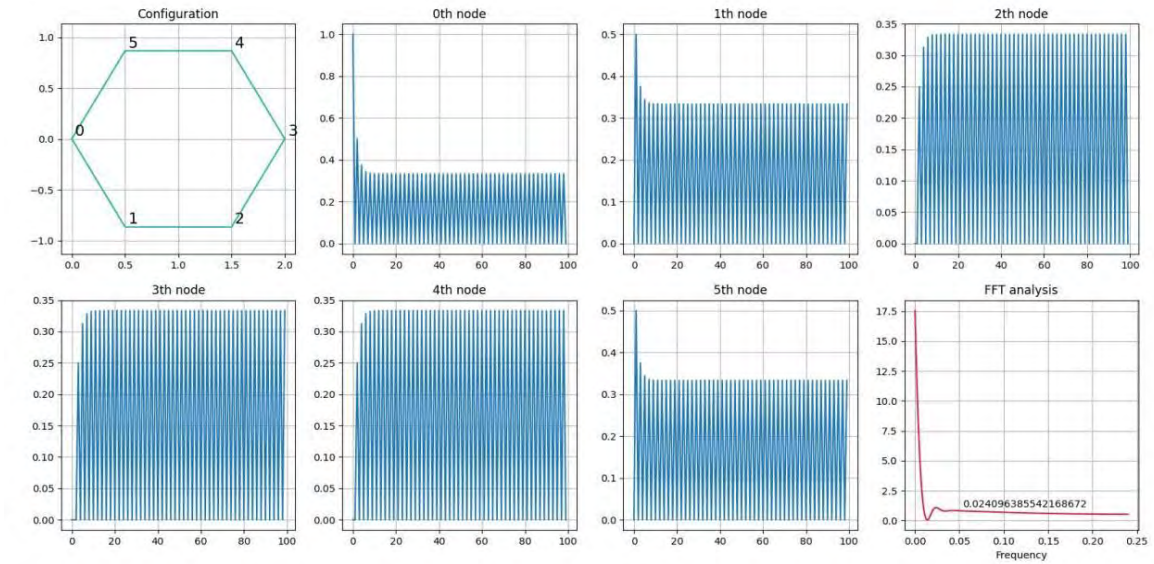


图 6. 偶数多边形经典行走的概率分布图及 FFT

2.2 量子搜索

2.2.1 Grover 搜索算法原理

已经存在的可以使用量子行走实现的算法是 Grover 算法[1]。实际上，Grover 算法可以算作一种量子行走算法。其基本原理是将搜索地址一一对应到 N 个量子比特的直积空间中，然后使用现实世界的量子门对其作出统一操作从而做到并行处理，提升运算速度（可达到 $O(\sqrt{N})$ ）。其需要的量子门如下：

Hadamard 门：

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

N 阶 Hadamard 门：

$$H^{(\otimes N)} = \underbrace{H \otimes H \otimes \dots \otimes H}_{N \uparrow}$$

N 阶搜索 Oracle:

$$O = \begin{pmatrix} I_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & I_n \end{pmatrix}$$

其中 I_i 为第 i 个搜索节点的特征数, 当第 i 个是搜索点时, 该值为 -1, 否则为 1。如上述的矩阵所示, Oracle 是一个酉操作, 在量子算法中的作用是标记搜索问题的解, 作用效果如下:

$$|x\rangle|q\rangle \xrightarrow{Oracle} |x\rangle|q \oplus f(x)\rangle$$

其中 $|x\rangle$ 是搜索的索引寄存器, 可以用函数 $f(x)$ 来表示一个判定问题。假设 $|x\rangle$ 是问题的解, 则 $f(x)=1$, 否则 $f(x)=0$ 。

Grover 算法是一种迭代的算法 (图 7), 每迭代一次, 量子态即向搜索解步进一次。其分为制备初态、迭代、测量三个部分:

1. 在直积空间 $|x\rangle$ 上, 对量子态作用 Oracle。
2. 对量子态作用 n 阶 Hadamard 门。
3. 翻转直积空间 $|0\rangle$ 态的相位。即作用算符—— $2|0\rangle\langle 0| - I$
4. 对量子态作用 n 阶 Hadamard 门。

其中 1、2 步为初态制备, 3、4 步为迭代。

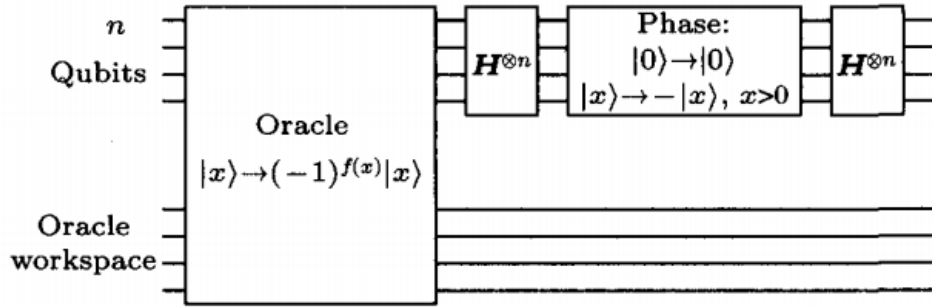


图 7. Grover 算子结构

注意到 Oracle 只翻转搜索目标的量子态, 而又有:

$$H^{\{\otimes N\}}(2|0\rangle\langle 0| - I)H^{\{\otimes N\}} = 2|\psi_0\rangle\langle \psi_0| - I$$

故一个 Grover 操作相当于在解空间表象下进行两次态的翻转, 倘若我们先把量子态写成解空间形式:

$$|\Psi\rangle = a\left(\frac{1}{\sqrt{k}}\sum_x^k |x_m\rangle\right) + b\left(\frac{1}{\sqrt{N-k}}\sum_x^{N-k} |x_n\rangle\right) = a|\beta\rangle + b|\alpha\rangle$$

其中, $\frac{1}{\sqrt{k}}\sum_x^k |x_m\rangle$ 代表所有搜索问题解的线性加和, $\frac{1}{\sqrt{N-k}}\sum_x^{N-k} |x_n\rangle$ 代表所有非搜索解的线性加和。可以

证明这两个向量可以描述 Grover 算法中涉及到的态。以这两个态为基矢, 表示上述的所有操作 (图 8):

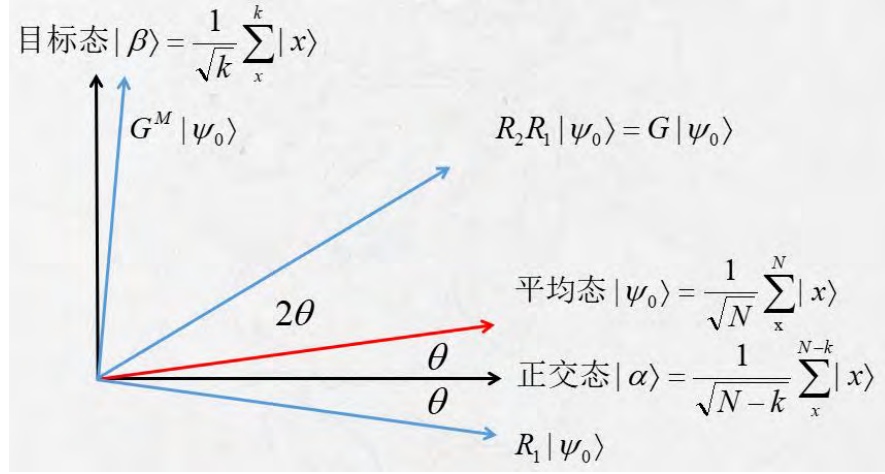


图 8. Grover 迭代过程

初态 $|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ 在这个空间内与 $|\alpha\rangle$ 基矢的夹角为 θ 。

对很大的 N 与很小的 k , 有 $\theta = \frac{1}{\sqrt{N}}$, 对每一次 Grover 运算, Oracle 将态绕 $|\alpha\rangle$ 翻转, $H^{(\otimes N)}(2|0\rangle\langle 0| - I)H^{(\otimes N)}$

将态绕 $|\psi_0\rangle$ 翻转, 则相当于向 $|\beta\rangle$ 步进一个 2θ , 经过 $M = \frac{\pi}{4}\sqrt{N}$ 步操作后, 态矢就步进到离目标态 $|\beta\rangle$ 很近的位置, 此时进行测量, 则得到目标量子态, 即完成搜索。这就是整个 Grover 算法复杂度为 $O(\sqrt{N})$ 的原因。根据 Oracle 调用次数来定义的复杂, 此亦称为“查询复杂度”。

2.2.2 量子行走 Grover 搜索算法原理

对于使用量子行走的实现方案, 只需要将 Grover 过程作用于硬币空间:

1. 制备初态为 $|\psi_0\rangle = |s^c\rangle \otimes |s^s\rangle$, 其中 $|s^c\rangle = \frac{1}{\sqrt{N}} \sum_{d=1}^N |d\rangle$, 表示 n 个方向的平均叠加态, 与 $|s^c\rangle$ 相同, $|s^s\rangle$ 代表 2^n 个位置的平均叠加态。

2. 构造硬币空间算符 C' , 对于非标注位置使用算符 $C_0 = G = (-I + 2|s^c\rangle\langle s^c|)$, 而对于非标注位置使用算符 $C_1 = -I$, 作用 $t_f = \frac{\pi}{2}\sqrt{2^n}$ 次。

3. 以基矢 $|d, \bar{x}\rangle$ 测量。

易证, 经过 $t_f = \frac{\pi}{2}\sqrt{2^n} = O(\sqrt{N})$ 次 SC' 作用后, 可以以 $\frac{1}{2} - O(\frac{1}{n})$ 的概率得到搜索结果, 具有和 Grover 算法一样的搜索效率[2]。

2.2.3 Grover 量子算法模拟

我们使用了10个量子比特, 输入方法采用字符串形式, '0000000000' 即代表我们的十个量子比特均位于基态。可知, 做直积后, 态空间变成 $2^{10} = 1024$ 维的, 撰写程序Zhihu.py (见SUPPLEMENTARY MATERIALS) 内的 "Hierbert_space" 来生成输入态的直积空间, 输出一个1024维的列向量 A_0 作为系数矩阵和1024个字符串态作为基矢。

设置要搜索目标: ['0100010000', '0100000000', '0000100000', '0000000001']。在 Quantum_Gate.py 中撰写生成 n 维量子门程序, 在本例中为 2^{10} 维量子门, 依据定义使用 scipy 包里的 kron 函数分别生成 1024

维 Oracle, Hadamard, 翻转门, 则基本的运算直接通过矩阵相乘即可得到。

在系数矩阵 A_0 前作用 Hadamard 门得到初态, 在选定循环次数 N 中循环作用 Oracle、Hadamard 门、翻转门、Hadamard 门, 每一次迭代均提取系数矩阵 A_0 的模方。最后画出每一次迭代后 A_0 的模方 (图 9):

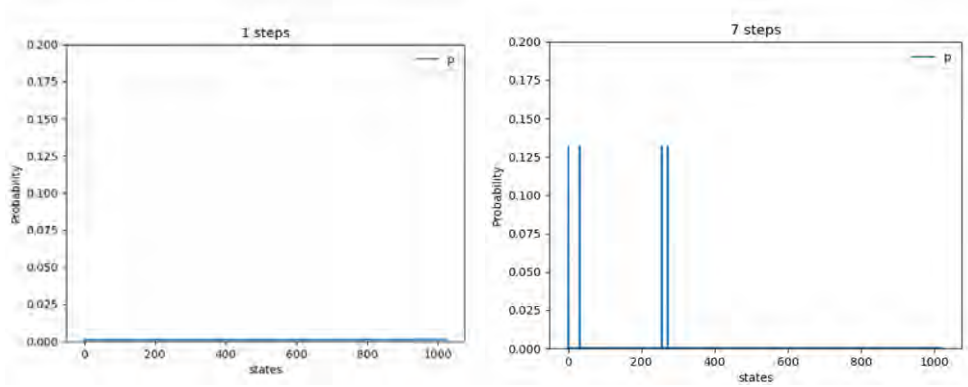


图 9. Grover 概率迭代

从图中可以清晰的看出, 一开始 1024 个态均处在等概率的直线上, 每经过一次 Grover 后, 目标 4 态的概率直线上涨, 而其他态的概率同时下降。与此同时, 对于 1024 个态的搜索来说, 其末态已经达到了 4 个态各占 23% 的概率, 符合其复杂度为 $O(\sqrt{N})=10$ 的理论值。

对这 10 次迭代, 我们提取出了每一次迭代后 10 个量子比特各自的系数 α , β 的值 (图 10):

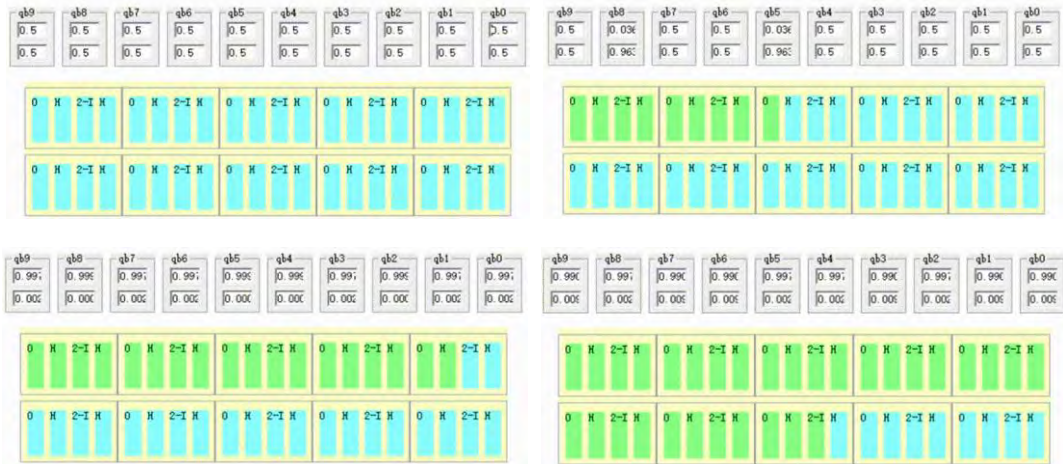


图 10. 单个量子比特在 Grover 迭代中的变化

可见, 每一步迭代均对量子比特的态产生了影响, 其中, $H^{(\otimes N)}$ 的演化是线性的, 因为每个量子比特的变化都可以被写成单个 Hadamard 门的作用, 也就是说 $H^{(\otimes N)}$ 是可以做成单个量子比特 Hadamard 门的直积的。而 $(2|0\rangle\langle 0| - I)$ 和 Oracle 的演化没有这个性质, 此被称作是纠缠的, 这样的作用是不能拆分成单个量子比特子空间直积的。自此我们用模拟的方法证明了 Grover 算法的复杂度和其 Oracle 量子门的纠缠性。

2.2.4 超立方体量子行走 Grover 搜索算法模拟

n 个量子比特形成的空间可以用一个 n 维超立方体描述, 每一个节点象征着直积空间中的一个态, 称为量子位。其中, 每条边相连的量子位之间仅翻转一个比特。这种类似格雷码的编排方法使得量子行走可以作用到直积空间中。举例如图, 下面是三个量子比特的立方体模型 (图 11):

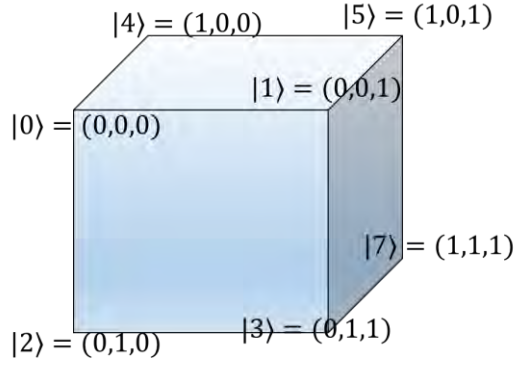


图 11. 三个量子比特的立方体模型

此时， $2^3 = 8$ 个节点上都占据一个态，每条边链接的态均仅差一个量子比特。

对于总量子态，为硬币态和位置态的直积。用 d 代表单个硬币态索引， \bar{x} 代表单个位置索引，则总态可以写为：

$$|\Psi\rangle = \sum_{d=0}^{n-1} \sum_{\bar{x}}^{2^n} |d, \bar{x}\rangle$$

对 n 维超立方体，每个节点上有 n 个节点与之相连，对应了 n 维空间中的单位向量。于是，此时的量子行走不再是以左右划分的，而是以沿空间向量方向划分的，对于一个节点上有 n 种量子行走方法，故硬币空间为 n 维。此时的位移算符为：

$$S = \sum_{d=0}^{n-1} \sum_{\bar{x}}^{2^n} |d, \bar{x} + \bar{e}_d\rangle \langle d, \bar{x}|$$

其中 \bar{e}_d 表示超立方体的第 d 个基矢。

与 Grover 算法的模拟类似，我们选取了 6 个量子比特，使用空数组生成了有 $2^6 = 128$ 个态的直积空间，每个量子态都与 6 个量子态直接相连。

设置需要搜索的量子态，这里搜索以下 5 个态：

['10010100', '01000100', '01011110', '10100000', '11000000']

随后，对数组上的 128 个位置都加入一个 6 维数组，代表其硬币空间的 6 个基矢分量。将其记做 $Coin_0$ ，即为我们需要处理的系数矩阵。

对数组撰写量子行走函数 Walk，即输入 Coin 后，能够返回一个按照 Coin 标识行走过后的系数矩阵。

然后对 Coin 进行上述算法的操作，将每一次行走后对应的位置态概率输出到 results，结果如下：

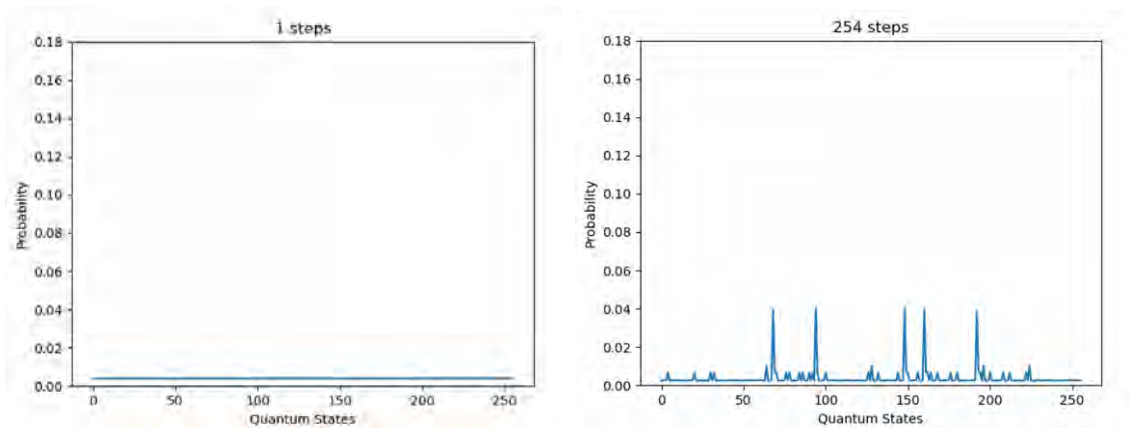


图 12. 超立方行走的概率变化

经过多步量子行走后位置空间的概率已基本集中到了我们想要搜索的解上。其中，除了五个要搜索的目标态外，还有一些概率在 0.016 附近的量子态也出现了较高的概率，这是一些和目标态直接相连的量子态，在量子行走过程中由于波函数的弥散而具有一定的搜索概率。

2.2.4 二维量子行走搜索算法模拟

随后，我们更改了硬币态空间为 4 维，将位置态空间用二维网格描述，重写行走算子、Grover 算子，设置搜索点[“(5, 5)”]，对二维平面上的搜索进行研究，分别使用 Matlab 与 Python 进行模拟，得到结果如下图：

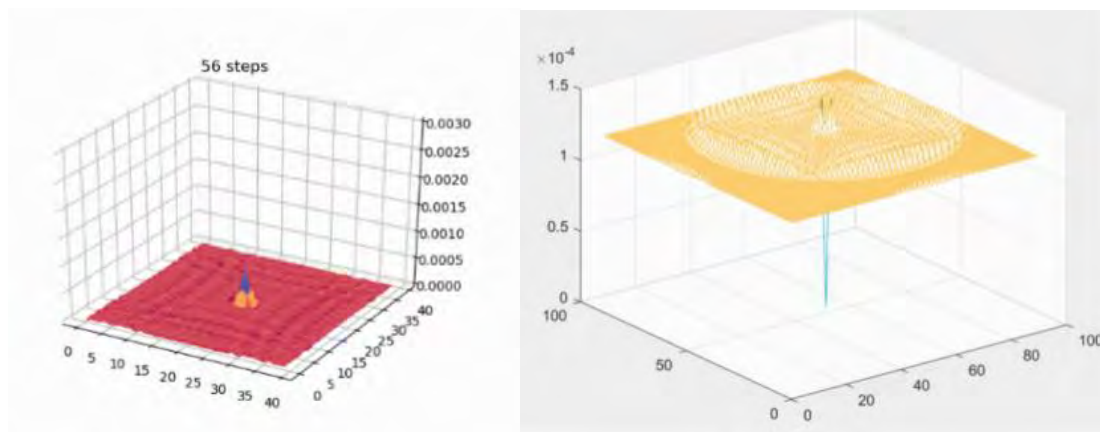


图 12. 二维平面行走的概率变化

可以看出，二维量子搜索过程中，目标位的量子态与其他方向不同，且受扰动的量子态会像波一样从目标位向其他方向扩展。

2.3 量子机器学习

传统机器学习的研究方向主要包括决策树、随机森林、人工神经网络、贝叶斯学习等方面的研究。机器学习能够获利于量子计算，从而突破一些传统机器学习的极限。我们接下来介绍几种我们认为可以在机器学习中应用的量子行走算法。

2.3.1 量子行走加速 kd 树搜索机器学习算法

对一个树状图[8]（图 13）上的量子行走，倘若研究从一棵树的根部出发，经时间 t 后粒子位置在另一棵树的根部的概率，[7]中证明在这样的相向二叉树中从最左边进入到最右边出来的时间，使用量子随机行走相比于经典随机行走有指数加速。进一步，对任意二叉树，量子随机行走的遍历时间相比于经典随机行走有指数加速。又如图 13b 所示由随机连接两棵树的叶子组成的图，同样得到使用连续量子随机行走可以以多项式效率穿过，而这是经典随机行走无法达到的。

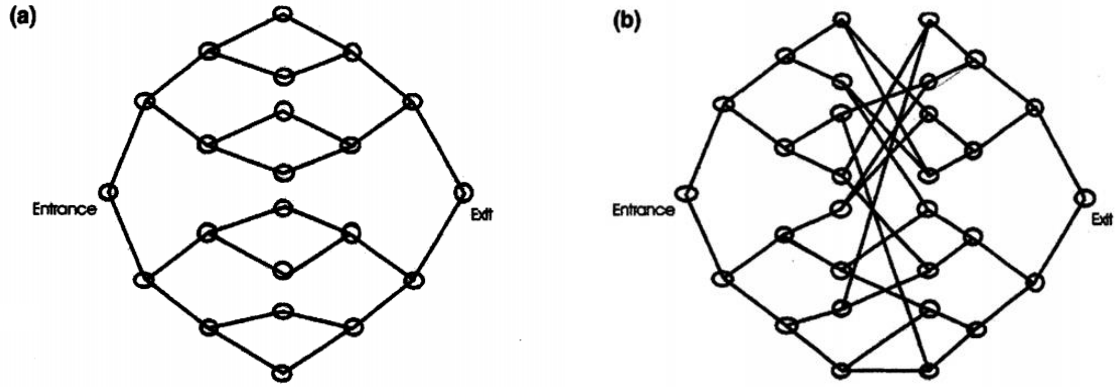


图 13. 相向二叉树的结构

Kd 树是一种对 k 维空间中的实例点进行存储以便对其进行快速检索而形成的结构。对于 kd 树搜索的机器学习算法[5]，我们可以很轻易地将量子行走方法应用到最后的搜索过程中，在参数空间中实现对机器学习的指数级加速，如一维空间 kd 树的构型（图 14）。

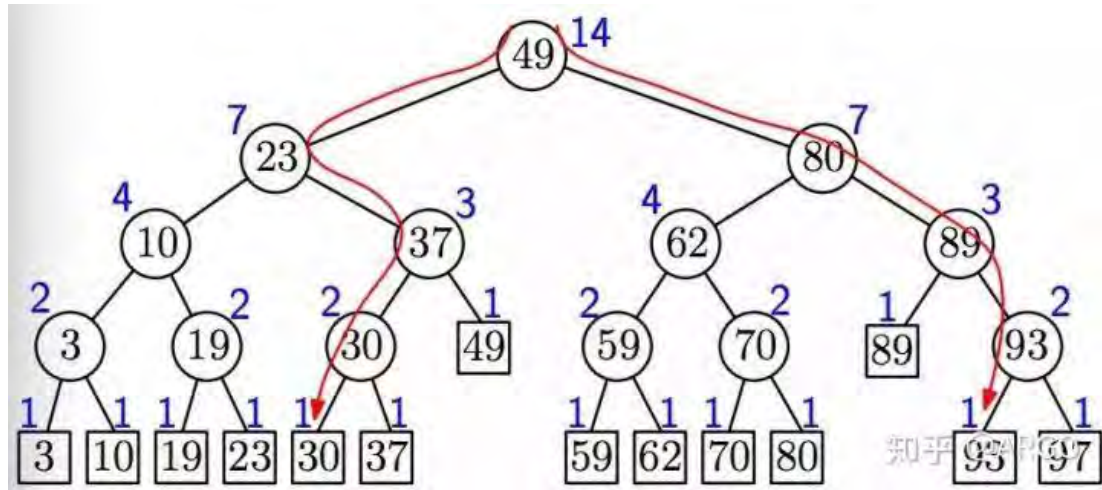


图 14. 一种一维空间的 kd 树

2.3.2 基于 Grover 搜索的支持向量机

最简单的量子支持向量机形式是：如果目标函数的参数是离散化的，则在目标空间中进行穷举搜索。对最小值的检索是基于 Grover 检索算法的一个变体实行的，所以量子行走算法对此类算法的加速是显而易见的。

这一方法对于目标函数的形式没有限制。由于其并不依赖梯度下降等算法，故非凸的目标函数也可行。而合理设计非凸损失函数则有可能避免外部实例和分类噪音对损失函数产生的影响。

2.3.3 Ising 模型和 QGM

鉴于篇幅，这里简要介绍一下使用量子行走怎样构建普适的量子机器学习。玻尔兹曼机是一个可以进行普适量子机器学习的模型[10]，其分布函数定义为：

$$p(x) = \frac{1}{Z} \exp(-E(v, h)) = \frac{1}{Z} \prod_{i=1}^m \prod_{j=1}^n \exp(h_i w_{ij} v_j) \prod_{j=1}^n \exp(\alpha_j v_j) \prod_{i=1}^m \exp(h_i \beta_i)$$

与伊辛模型比较：

$$\mathcal{H} = - \sum_{\langle i, j \rangle} J Z_i Z_j + \sum_i B_i Z_i$$

可见对可操作变量 v ，有可能从伊辛模型去构建相应的玻尔兹曼机[9,11]。在[12]中，我们可以通过

构建 QGM 图的方式来实现普适的量子玻尔兹曼机。事实上，玻尔兹曼机的概念源于物理，其物理实现也是可以选择很多系统的，如用量子位实现的 Ising 模型（图 15）[12]。

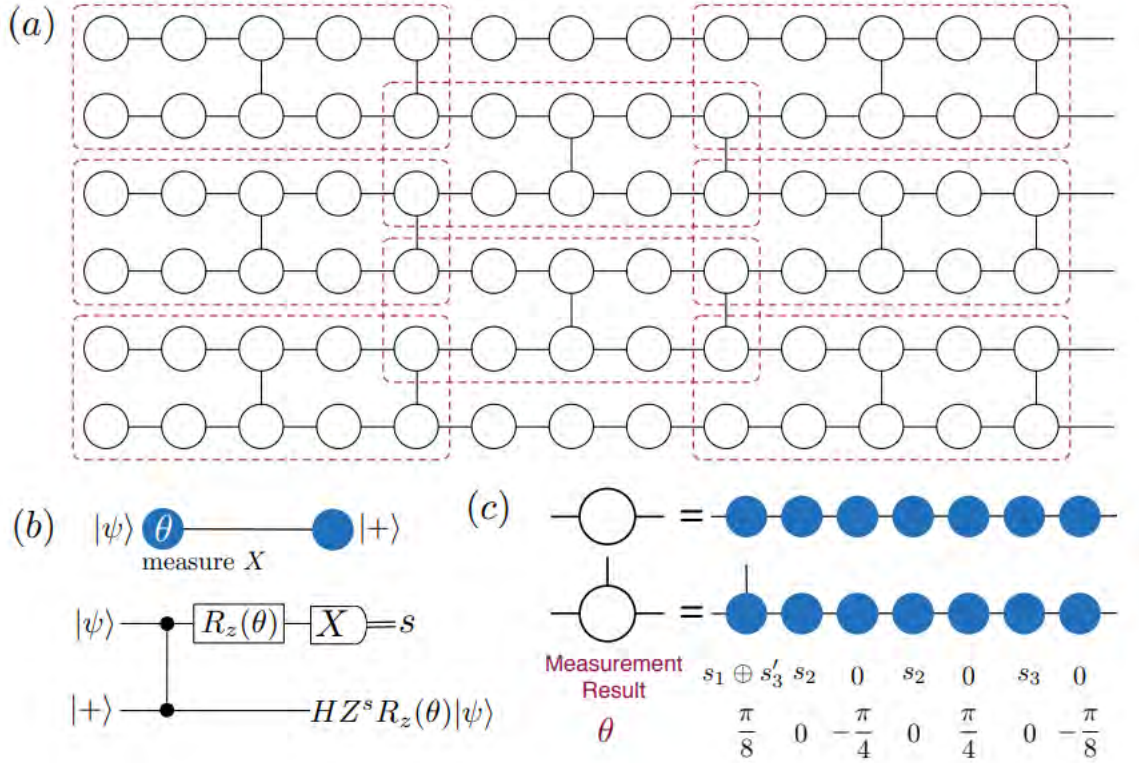


图 15. (a)brickwork state. 每个圆代表一个 $|+\rangle$ 状态，每一行表示一个 CZ 操作。(b)传播通过测量一个量子位。(c)每个白色圆圈不同的旋转角度被 7 个物理量子位代替旋转角度固定。角度变化被编码到不同的测量结果中[12]。

事实上，倘若将 hidden variable 采纳做硬币空间 H_c ，visible variable 作为 H_p 的话，在 QGM 上的每一次训练都可以表示为在这个 Ising 图上的一次行走。这既体现了量子行走的描述普适性[13]。

3.其他可能的量子行走信息处理方法

在[14]中，R. Yin 给出了在环上量子行走的首次探测时间与其在相空间哈密顿量演化的 winding number 的关系，是一个新奇的拓扑性质。这里我们提出，如果将首次探测作为一个算符作用在晶格上，则可以通过量子傅里叶变换的相位估计来得到系统的 winding number，这个方法可能具有一定应用前景。

4 结语

该理论与数值模拟向我们展示了量子随机行走相较经典随机行走在遍历方面的指数加速、在搜索方面的平方加速，并给出了普适搜索问题的一般编码方式。一维行走、多边形行走和在最后介绍的树状图行走都为我们展示了量子行走的新奇性质。在普适机器学习和普适搜索问题上的表现更是揭示了量子随机行走在量子计算以及量子算法领域具有的强大的应用潜力。而新研究又将量子行走的拓扑性质展现了出来，有望进行Berry相的进一步研究。

5 特点/特色/创新性声明

创新性：我们首次研究了多边形行走及其FFT频谱，又对一般的量子行走下的搜索问题进行了编码讨论。我们创新性地考虑将量子行走拥有的优良性质融入机器学习算法中，并给出三个方向（kd树、支持向量机和玻尔兹曼机），我们也创新性地考虑了将量子行走Winding number提取成可测量物理量的过程。

特色：我们进行了大量独立的模拟，对理论结果作了验证。

参考文献：

- [1] Michael A. Nielsen, Isaac L. Chuang. Quantum Computation and Quantum Information. 2nd ed; Cambridge University Press 2000.

- [2] N. Shenvi, J. Kempe and K. B. Whaley, Quantum random—walk search algorithm, *Phys. Rev. A* 67, 052307(2003).
- [3] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, *Phys. Rev. A* 48, 1687(1993).
- [4] F. Magniez, A. Nayak, P. Pudguchter and M. Santha, On the hitting times of quantum versus random walks, In: *Proceedings of the 20th annual ACM—SIAM SODA*, ACM, New York, PP. 86—95(2009).
- [5] 李航, 统计学习方法. 清华大学出版社, 2012.
- [6] Peter Wittek, 张毅 译, 量子机器学习中数据挖掘的量子计算方法. 哈尔滨工业大学出版社, 2019.
- [7] E. Farhi, Quantum computation and decision trees, *Phys. Rev. A* 58, 915 (1998).
- [8] A. M. Childs, E. Farhi and S. Gutmann, An Example of the Difference Between Quantum and Classical Random Walks, *Quantum Information Processing* 1, PP. 35—43(2002).
- [9] X. Gao, S.-T. Wang, L.-M. Duan, Quantum supremacy for simulating a translation-invariant Ising spin model. *Phys. Rev. Lett.* 118, 040502 (2017).
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
- [11] Mohammad H. Amin,^{1, 2} Evgeny Andriyash,¹ Jason Rolfe,¹ Bohdan Kulchitskyi,³ and Roger Melko^{3, 4}. Quantum Boltzmann Machine. *arXiv:1601.02036v1 [quant-ph]* 8 Jan 2016
- [12] X. Gao^{1 *}, Z.-Y. Zhang^{1,2}, L.-M. Duan^{1,2†}. A quantum machine learning algorithm based on generative models. *SCIENCE ADVANCES* 2018.
- [13] 李敏, 量子随机行走的基本性质及应用研究, 合肥 中国科学技术大学. 2013.4 博士学位论文.
- [14] R. Yin,¹ K. Ziegler,² F. Thiel,¹ and E. Barkai¹. Large fluctuations of the first detected quantum return time. *arXiv:1903.03394v2* 11 Mar 2019

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at <https://github.com/exquisite-scarlet/quantum-.git>