第3章 量子基本算法

从前两章已经了解了量子计算的基础理论和实现手段。本章介绍几种典型的量子算法，它们在量子机器学习算法中起着至关重要的作用。例如：量子态制备是量子机器学习的基础，利用它将样本数据存储在量子态中；量子搜索算法能够找到满足特定条件的目标；量子傅里叶变换能够将存储在基态和振幅中的信息相互转换；交换测试能够计算两个量子态的保真度，从而计算样本间的相似度；HHL 算法能够求解量子机器学习中的线性方程组。

3.1 量子态制备

3.1.1 4 维量子态制备

假设样本信息为

$$\boldsymbol{x}=(0.4 \quad 0.4 \quad 0.8 \quad 0.2)^{\mathrm{T}} \tag{3.1.1}$$

以这个样本为例说明如何将它制备到量子态中。因为该样本是 4 维的，即有 4 个特征，所以位置信息用 log4＝2 个量子比特表示，初始状态皆为$|0\rangle$。制备量子态的目的是得到量子态$|x\rangle=0.4|00\rangle+0.4|01\rangle+0.8|10\rangle+0.2|11\rangle$。该量子态将样本的 4 个特征存储在振幅中，$|00\rangle$、$|01\rangle$、$|10\rangle$、$|11\rangle$是位置信息，表示$|00\rangle$位置存储的是第一个特征 0.4，$|01\rangle$位置存储的是第二个特征 0.4，$|10\rangle$位置存储的是第三个特征 0.8，$|11\rangle$位置存储的是第四个特征 0.2。

4 个特征的结合过程如图 3.1 所示，首先将向量 $\boldsymbol{x}$ 中的元素两两结合得到第一层的数据，进一步将第一层的数据结合得到第零层的数据。由于 $0.4^2+0.4^2=(\sqrt{0.32})^2$，$0.8^2+0.2^2=(\sqrt{0.68})^2$，$(\sqrt{0.32})^2+(\sqrt{0.68})^2=1^2$，因此第一层的两个数据分别为$\sqrt{0.32}$和$\sqrt{0.68}$。也就是说，第 k 层的第 i 个数据 p_i^k 是由第 $k+1$ 层的第 $2i-1$ 个数据 p_{2i-1}^{k+1} 和第 $k+1$ 层的第 $2i$ 个数据 p_{2i}^{k+1} 结合而成，即$(p_i^k)^2=(p_{2i-1}^{k+1})^2+(p_{2i}^{k+1})^2$。

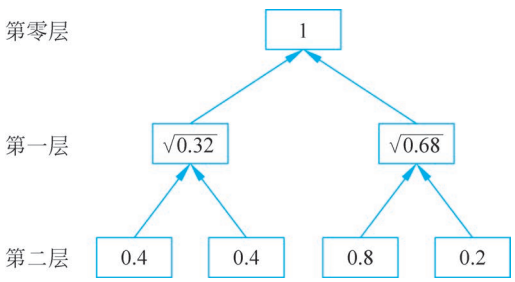


图 3.1 4 个特征的结合过程

量子线路的实现是由上到下的，先实现第一层的$\sqrt{0.32}$和$\sqrt{0.68}$，再分别实现第二层的 0.4 和 0.4、0.8 和 0.2。将 $\boldsymbol{x}$ 制备到量子态上的量子线路图如图 3.2 所示。

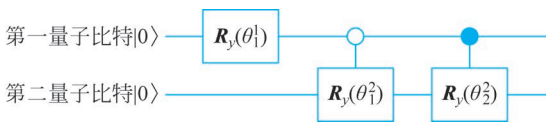


图 3.2 将 x 制备到量子态上的量子线路图

首先制备 2 个量子比特$|0\rangle|0\rangle$，然后使用旋转门$\boldsymbol{R}_y(\theta_1^1)$门作用于第一个量子比特$|0\rangle$，得到

$$(\sqrt{0.32}\,|0\rangle+\sqrt{0.68}\,|1\rangle)\,|0\rangle \tag{3.1.2}$$

由于

$$\boldsymbol{R}_y(\theta)\,|0\rangle=\begin{pmatrix}\cos\dfrac{\theta}{2} & -\sin\dfrac{\theta}{2}\\ \sin\dfrac{\theta}{2} & \cos\dfrac{\theta}{2}\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}\cos\dfrac{\theta}{2}\\ \sin\dfrac{\theta}{2}\end{pmatrix}=\cos\frac{\theta}{2}\,|0\rangle+\sin\frac{\theta}{2}\,|1\rangle \tag{3.1.3}$$

因此，$\cos\dfrac{\theta_1^1}{2}=\sqrt{0.32}$，即$\theta_1^1=2\arccos\sqrt{0.32}$。

再使用受控$\boldsymbol{R}_y(\theta_1^2)$门和$\boldsymbol{R}_y(\theta_2^2)$门作用于第二个量子比特。$\boldsymbol{R}_y(\theta_1^2)$受到 0 控制，当第一个量子比特处于状态$|0\rangle$时，将$\sqrt{0.32}$处理成 0.4 和 0.4，$\boldsymbol{R}_y(\theta_2^2)$受到 1 控制，当第一个量子比特处于状态$|1\rangle$时，将$\sqrt{0.68}$处理成 0.8 和 0.2，得到

$$0.4\,|00\rangle+0.4\,|01\rangle+0.8\,|10\rangle+0.2\,|11\rangle \tag{3.1.4}$$

由于

$$\begin{aligned}&0.4\,|00\rangle+0.4\,|01\rangle+0.8\,|10\rangle+0.2\,|11\rangle\\ =&\sqrt{0.32}\,|0\rangle\left[\frac{1}{\sqrt{0.32}}(0.4\,|0\rangle+0.4\,|1\rangle)\right]+\sqrt{0.68}\,|1\rangle\\ &\left[\frac{1}{\sqrt{0.68}}(0.8\,|0\rangle+0.2\,|1\rangle)\right]\\ =&\sqrt{0.32}\,|0\rangle\left(\sqrt{\frac{1}{2}}\,|0\rangle+\sqrt{\frac{1}{2}}\,|1\rangle\right)+\sqrt{0.68}\,|1\rangle\left(\sqrt{\frac{16}{17}}\,|0\rangle+\sqrt{\frac{1}{17}}\,|1\rangle\right)\end{aligned} \tag{3.1.5}$$

因此，在运用$\boldsymbol{R}_y(\theta_1^2)$和$\boldsymbol{R}_y(\theta_2^2)$这两个门时，$\theta_1^2=2\arccos\sqrt{\dfrac{1}{2}}$，$\theta_2^2=2\arccos\sqrt{\dfrac{16}{17}}$。

3.1.2 *M* 维量子态制备

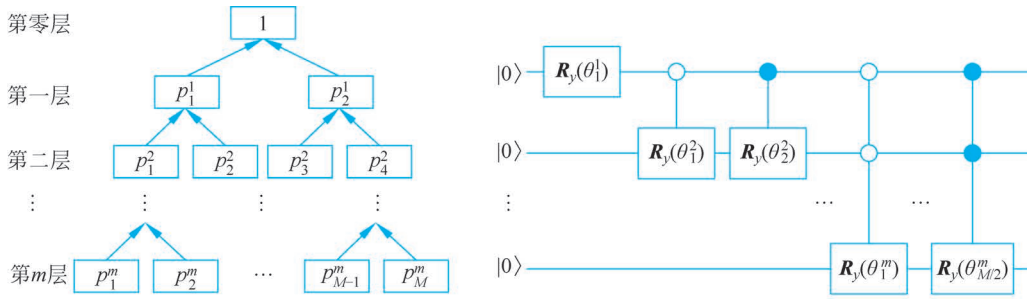
3.1.1 节给出了有 4 个特征的样本的制备过程，当样本有更多特征时，可以将上述过程扩展。

假设样本有 M 个特征，图 3.3(a)给出了这 M 个特征结合的过程，其中第 $m=\log M$ 层的 M 个数据对应于$\boldsymbol{x}=(x_0\quad x_1\quad\cdots\quad x_{M-1})^{\mathrm{T}}$，即 $p_{i+1}^m=x_i(i=0,1,\cdots,M-1)$。制备它们的量子线路如图 3.3(b)所示，其中第 j 层的第 k 个参数θ_j^k的计算公式为

$$\theta_j^k=2\arccos\frac{p_{2j-1}^k}{p_j^{k-1}} \tag{3.1.6}$$

对于 $i(i=0,1,\cdots,m-1)$受控的量子线路来说，其复杂度为 $O(i)$。因此，图 3.3(b)中线路合计要用到的单量子门和受控旋转门的总数量为$1+\sum_{i=1}^{m-1}i\times 2^i=(m-2)\times 2^m+$

3，所以其复杂度为 $O(m2^m)$。



(a) M个特征的结合过程　　(b) 制备线路

图 3.3　M 个特征的结合过程及制备它们的量子线路图

3.1.3　实现

本实验将 3.1.1 节中的 $\boldsymbol{x}=(0.4\quad 0.4\quad 0.8\quad 0.2)^{\mathrm{T}}$ 制备到量子态中，量子线路如图 3.4 所示。根据 3.1.1 节的分析，第一步实施 $\boldsymbol{R}_y(\theta_1^1)$ 制备第一层的数据，其中 $\theta_1^1=2\arccos\sqrt{0.32}=1.939$。第二步和第三步实施受控 $\boldsymbol{R}_y(\theta_1^2)$ 门和 $\boldsymbol{R}_y(\theta_2^2)$ 门制备第二层的数据，其中 $\theta_1^2=2\arccos\sqrt{\dfrac{1}{2}}=1.571$，$\theta_2^2=2\arccos\sqrt{\dfrac{16}{17}}=0.490$。

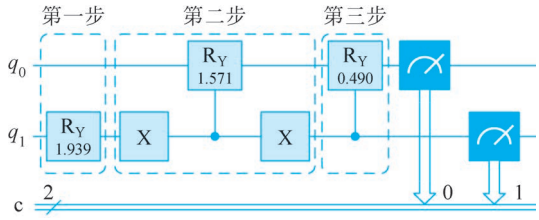


图 3.4　制备量子态的量子线路图

值得注意的是，在第二步中由于 qiskit 中只有 1 控制，无法直接实现 0 控制这一操作，因此在控制位的前后各增加了一个 $\boldsymbol{X}$ 门。第一个 $\boldsymbol{X}$ 门将 q_1 线路上的$|0\rangle$变成$|1\rangle$，这样 0 控制就变成了 1 控制。第二个 $\boldsymbol{X}$ 门将 q_1 线路上的$|1\rangle$恢复成$|0\rangle$，不影响程序的正确执行。在之后的量子线路中，如果出现 0 控制，那么都按照同样的操作进行处理。

实验结果如图 3.5 所示。图中横坐标的二进制序列从左到右按照 q_1q_0 的顺序排列，因此得到 00、01、10 和 11 的概率分别为 0.159、0.163、0.640 和 0.038。也就是说，00、01、10 和 11 的振幅分别为$\sqrt{0.159}\approx 0.4$、$\sqrt{0.163}\approx 0.4$、$\sqrt{0.640}\approx 0.8$ 和$\sqrt{0.038}\approx 0.2$，即最后得到的量子态为

$$0.4\,|00\rangle+0.4\,|01\rangle+0.8\,|10\rangle+0.2\,|11\rangle \tag{3.1.7}$$

正是需要制备的量子态。

量子态制备的代码如下：

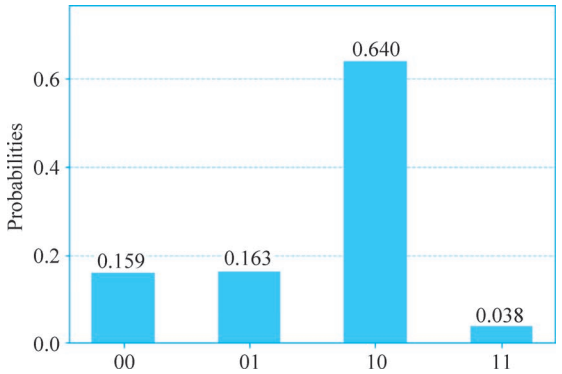


图 3.5　实验结果

```
%matplotlib inline
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute
from qiskit import Aer
from qiskit import IBMQ
from math import pi
import numpy as np
from qiskit.tools.visualization import plot_histogram
circuit = QuantumCircuit(2,2)

#第一步
circuit.ry(1.939,1)

#第二步
circuit.x(1)
circuit.cry(1.571,1,0)
circuit.x(1)

#第三步
circuit.cry(0.490,1,0)

#测量
circuit.measure(0,0)
circuit.measure(1,1)

#绘制线路图
circuit.draw(output = 'mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit,backend,shots = 20000)
sim_result = job_sim.result()

#绘制结果图
measurement_result = sim_result.get_counts(circuit)
plot_histogram(measurement_result)
```

3.2 量子搜索算法

假设数据集中有 N 个无序数据，搜索算法的任务是将符合条件的数据找出来。如果用经典计算机搜索符合条件的数据，那么需要将所有的数据检查一遍，即需要 N 次查询，因此其复杂度为 $O(N)$。

量子搜索算法是由 Grover 在 1996 年提出的一种算法，假设 N 个数据中符合条件的数据有 M 个，则量子搜索算法的复杂度为 $O\left(\sqrt{\frac{N}{M}}\right)$，远小于经典算法的复杂度。机器学习中经常需要从一堆数据中找到某些数据，如 K 近邻算法中要找到最相近的 K 个数据，因此量子搜索算法在量子机器学习中起着重要作用。

3.2.1 黑箱

2.6 节已经出现了黑箱的概念，量子搜索算法中也用到黑箱，此处黑箱的作用是将 N 个数据中符合条件的数据标记出来。

下面以 $N=2$ 为例介绍黑箱如何标记符合条件的数据。$N=2$ 意味着只有两个数据，可以用 0 和 1 来表示这两个数据，也就是只需要 1 个量子比特来表示。与 2.6 节一样，假设 $f(x):x\in\{0,1\}\to y\in\{0,1\}$ 是一个函数，若 x 是符合条件的数据，则 $f(x)=1$；若 x 不是符合条件的数据，则 $f(x)=0$。定义黑箱 $\boldsymbol{O}$ 具有如下功能：

$$\boldsymbol{O}(|x\rangle|y\rangle)=|x\rangle|y\oplus f(x)\rangle \tag{3.2.1}$$

式中：$|y\rangle$ 为辅助量子比特。

此处的辅助量子比特 $|y\rangle$ 的初态不像 2.6 节那样为 $|0\rangle$，而是 $|y\rangle=\frac{|0\rangle-|1\rangle}{\sqrt{2}}$。此时，有

$$\begin{aligned}\boldsymbol{O}\left[|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right]&=|x\rangle\frac{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\\&=\begin{cases}|x\rangle\left(\dfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right),f(x)=0\\|x\rangle\left(\dfrac{|1\rangle-|0\rangle}{\sqrt{2}}\right),f(x)=1\end{cases}\\&=(-1)^{f(x)}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\end{aligned} \tag{3.2.2}$$

可以看出，黑箱 $\boldsymbol{O}$ 作用前后 $|y\rangle$ 的状态都是 $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$，因此在书写 $\boldsymbol{O}$ 的功能时，通常省略 $|y\rangle$，则黑箱 $\boldsymbol{O}$ 的作用记为

$$\boldsymbol{O}|x\rangle=(-1)^{f(x)}|x\rangle \tag{3.2.3}$$

当 x 不是符合条件的数据，即 $f(x)$ 为 0 时，$\boldsymbol{O}|x\rangle=|x\rangle$，$|x\rangle$ 没变，也就是未被标记；当 x 是符合条件的数据，即 $f(x)$ 为 1 时，$\boldsymbol{O}|x\rangle=-|x\rangle$，$|x\rangle$ 变为 $-|x\rangle$，也就是用负号把

$|x\rangle$标记出来。因此黑箱将符合条件的数据标记了出来，有时也说黑箱标记了搜索问题的解。

图 3.6 是元素为 2 个时由黑箱标记符合条件的解的量子线路图。

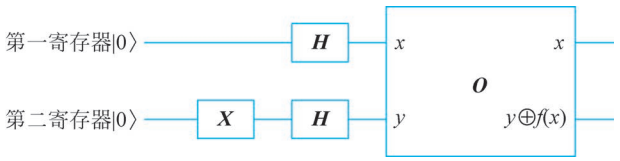


图 3.6　$N=2$ 时标记符合条件的解的量子线路图

当 $N=2$ 时，假设 $f(0)=1$，$f(1)=0$，即 0 是符合条件的解。标记符合条件的解的具体过程如下：

首先制备量子态

$$|\psi_0\rangle=|00\rangle \tag{3.2.4}$$

第一寄存器用于存储所有可能的解，即定义域集合；第二寄存器是辅助量子比特。

使用 $\boldsymbol{H}$ 门作用于第一寄存器得到所有可能的解，并使用 $\boldsymbol{X}$ 门和 $\boldsymbol{H}$ 门作用于第二寄存器得到量子态$\frac{|0\rangle-|1\rangle}{\sqrt{2}}$，则$|\psi_0\rangle$演化为

$$|\psi_1\rangle=\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)=\frac{1}{\sqrt{2}}|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)+\frac{1}{\sqrt{2}}|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \tag{3.2.5}$$

由式(3.2.1)可知，黑箱 $\boldsymbol{O}$ 作用于$|\psi_1\rangle$的过程如下：

$$\begin{aligned}\boldsymbol{O}(|\psi_1\rangle)&=\frac{1}{\sqrt{2}}\boldsymbol{O}\left[|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right]+\frac{1}{\sqrt{2}}\boldsymbol{O}\left[|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right]\\&=\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)+\frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\\&=-\frac{1}{\sqrt{2}}|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)+\frac{1}{\sqrt{2}}|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\end{aligned} \tag{3.2.6}$$

由上式可以看出，0 前边的系数变为负值，1 的系数没有变，这说明标记了符合条件的解。

当有 $N=2^n$ 个元素时，需要 n 个量子比特来表示所有需要搜索的数据。辅助量子比特仍然只用一个比特。若某个数据 x 是符合条件的解，则 $f(x)=1$；若 x 不是符合条件的解，则 $f(x)=0$。此时黑箱的作用仍然是 $\boldsymbol{O}|x\rangle=(-1)^{f(x)}|x\rangle$，由黑箱标记符合条件的解的量子线路图如图 3.7 所示。

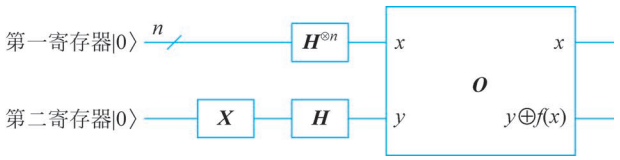


图 3.7　元素个数为 N 时标记符合条件的解的量子线路图

黑箱能够标记出符合条件的解，并不意味着找到了符合条件的解。这是因为用负号标记出符合条件的解，属于改变相对相位，但是目前常用的测量基为$|0\rangle$和$|1\rangle$，在这一组基下相对相位是不会引起可观测效应的，即观察不到。Grover 算法解决的正是如何将标记结果传递给外界的问题。

3.2.2 Grover 算法

Grover 量子搜索算法是一个迭代的过程，主要思路是从初始状态出发，重复进行多次变换，让符合条件的解的振幅越来越大，最后进行测量，就能以很高的概率得到正确的结果。假设要在 $N=2^n$ 个元素的搜索空间中进行搜索，元素编号为$\{0,1,\cdots,N-1\}$。这些编号存储在 n 个量子比特的量子态$|\psi\rangle$中：

$$|\psi\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \tag{3.2.7}$$

同时假设该搜索问题有 $M(1\leqslant M\leqslant N)$个符合条件的解。图 3.8 给出了 Grover 搜索算法的量子线路图，算法的目的是使用最少的 Grover 算子 $\boldsymbol{G}$ 搜索出问题的解。其原理是通过一次次使用 Grover 算子 $\boldsymbol{G}$，逐步提高符合条件的解的振幅，使得振幅最终能够接近 1 或者达到 1。这样在测量时就能够以接近 100%的概率将符合条件的解提取出来，即传递给外界。

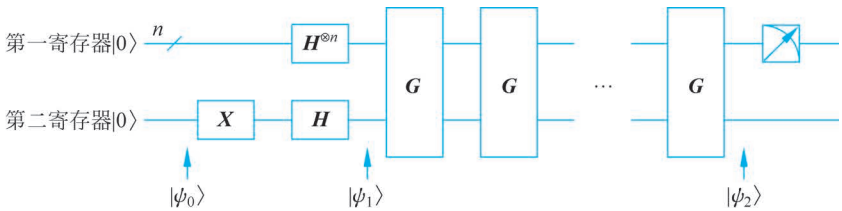


图 3.8 Grover 算法的量子线路

Grover 算法共需要 $n+1$ 个量子比特，初态为$|\psi_0\rangle=|0\rangle^{\otimes(n+1)}$。前 n 个量子比特组成第一寄存器，用于存储元素编号；另外 1 个构成第二寄存器，是辅助量子比特。Grover 算法步骤如下：

第一步：使用 n 个 $\boldsymbol{H}$ 门作用于第一寄存器演化出元素编号的叠加态，并使用 $\boldsymbol{X}$ 门和 $\boldsymbol{H}$ 门作用于第二寄存器演化出量子态$\frac{|0\rangle-|1\rangle}{\sqrt{2}}$。则$|\psi_0\rangle$演化为

$$|\psi_1\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \tag{3.2.8}$$

第二步：重复算子 $\boldsymbol{G}$ 以增大满足条件的解的概率，具体的重复次数将在 3.2.4 节算法分析中给出。每个算子 $\boldsymbol{G}$ 可分为以下 4 步，量子线路图如图 3.9 所示。

(1) 使用黑箱算子 $\boldsymbol{O}$ 将符合条件的解的符号取反；

(2) 使用 $\boldsymbol{H}^{\otimes n}$ 作用于第一寄存器；

(3) 使用条件相移算子 $2|0\rangle^{\otimes n}\langle 0|^{\otimes n}-\boldsymbol{I}$ 将$|0\rangle$以外的基态$|1\rangle,|2\rangle,\cdots,|N-1\rangle$

取反；

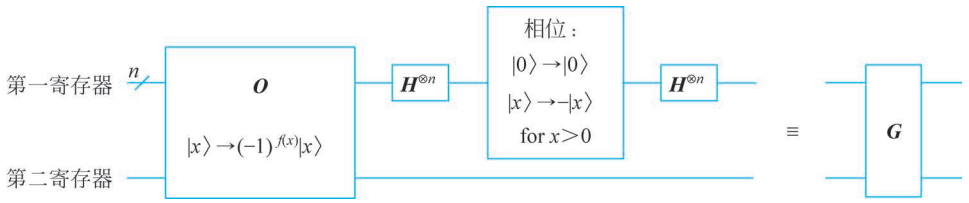


图 3.9　算子 G 的量子线路图

(4) 使用 $\boldsymbol{H}^{\otimes n}$ 作用于第一寄存器。

从图 3.9 能够看出，第二寄存器这个辅助量子比特仅用于黑箱算子 $\boldsymbol{O}$。根据 3.2.1 节的描述可知，辅助量子比特的状态在黑箱过程中保持不变，在讨论算法的过程中可以忽略。因此，在后续讨论算子 $\boldsymbol{G}$ 的过程中，乃至整个 Grover 算法中，都忽略辅助量子比特，仅讨论 $\boldsymbol{G}$ 算子和 Grover 算法对第一寄存器的影响。在此前提下，第(2)～(4)步可以总结为如下的演化算子：

$$\boldsymbol{H}^{\otimes n}(2\mid 0\rangle^{\otimes n}\langle 0\mid^{\otimes n}-\boldsymbol{I})\boldsymbol{H}^{\otimes n}=2\boldsymbol{H}^{\otimes n}\mid 0\rangle^{\otimes n}\langle 0\mid^{\otimes n}\boldsymbol{H}^{\otimes n}-\boldsymbol{H}^{\otimes n}\boldsymbol{I}\boldsymbol{H}^{\otimes n}=2\mid\psi\rangle\langle\psi\mid-\boldsymbol{I} \tag{3.2.9}$$

式中：$|\psi\rangle$见式(3.2.7)。

因此，Grover 算子可以表示为

$$\boldsymbol{G}=(2\mid\psi\rangle\langle\psi\mid-\boldsymbol{I})\boldsymbol{O} \tag{3.2.10}$$

式中只给出了算子 $\boldsymbol{G}$ 的定义，下一节详细解释 $\boldsymbol{G}$ 是如何增大满足条件的解的概率的。

3.2.3　G 算子的图形化解释

本节给出算子 $\boldsymbol{G}$ 的图形化解释，以说明算子 $\boldsymbol{G}$ 是如何增大满足条件的解的概率的。如式(3.2.7)所示，$|\psi\rangle$包含了所有数据的编号。假设 $f(x)=1$ 的所有 x 组成的量子态为

$$\mid\beta\rangle=\frac{1}{\sqrt{M}}\sum_{x\in\{x\mid f(x)=1\}}\mid x\rangle \tag{3.2.11}$$

$f(x)=0$ 的所有 x 组成的量子态为

$$\mid\alpha\rangle=\frac{1}{\sqrt{N-M}}\sum_{x\in\{x\mid f(x)=0\}}\mid x\rangle \tag{3.2.12}$$

则

$$\mid\psi\rangle=\sqrt{\frac{N-M}{N}}\mid\alpha\rangle+\sqrt{\frac{M}{N}}\mid\beta\rangle \tag{3.2.13}$$

由于$\langle\alpha\mid\beta\rangle=\langle\beta\mid\alpha\rangle=0$且$\langle\alpha\mid\alpha\rangle=\langle\beta\mid\beta\rangle=1$，因此$|\alpha\rangle$和$|\beta\rangle$可以看作一组基。令 $\sin\frac{\theta}{2}=\sqrt{\frac{M}{N}}$，则式(3.2.13)可以表示为

$$|\psi\rangle=\begin{pmatrix}\sqrt{\dfrac{N-M}{N}}\\ \sqrt{\dfrac{M}{N}}\end{pmatrix}=\begin{pmatrix}\cos\dfrac{\theta}{2}\\ \sin\dfrac{\theta}{2}\end{pmatrix} \tag{3.2.14}$$

黑箱算子 $\boldsymbol{O}$ 的作用是将符合条件的解的符号取反，$\boldsymbol{O}$ 算子作用于式(3.2.13)可得

$$\boldsymbol{O}|\psi\rangle=\sqrt{\frac{N-M}{N}}|\alpha\rangle-\sqrt{\frac{M}{N}}|\beta\rangle=\begin{pmatrix}\cos\dfrac{\theta}{2}\\ -\sin\dfrac{\theta}{2}\end{pmatrix} \tag{3.2.15}$$

由式(3.2.14)和式(3.2.15)可知，在基$|\alpha\rangle$和$|\beta\rangle$下，算子 $\boldsymbol{O}$ 的矩阵形式为$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$，且

$$2|\psi\rangle\langle\psi|-\boldsymbol{I}=2\begin{pmatrix}\cos\dfrac{\theta}{2}\\ \sin\dfrac{\theta}{2}\end{pmatrix}\begin{pmatrix}\cos\dfrac{\theta}{2} & \sin\dfrac{\theta}{2}\end{pmatrix}-\boldsymbol{I}=\begin{pmatrix}\cos\theta & \sin\theta\\ \sin\theta & -\cos\theta\end{pmatrix} \tag{3.2.16}$$

因此算子 $\boldsymbol{G}$ 的矩阵形式为

$$\boldsymbol{G}=(2|\psi\rangle\langle\psi|-\boldsymbol{I})\boldsymbol{O}=\begin{pmatrix}\cos\theta & \sin\theta\\ \sin\theta & -\cos\theta\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} \tag{3.2.17}$$

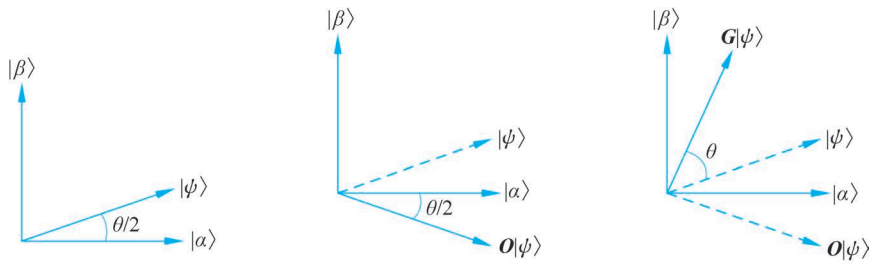
在 Grover 量子搜索算法中，对量子态$|\psi\rangle$实施一次算子 $\boldsymbol{G}$ 得到

$$\boldsymbol{G}|\psi\rangle=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\cos\dfrac{\theta}{2}\\ \sin\dfrac{\theta}{2}\end{pmatrix}=\begin{pmatrix}\cos\dfrac{3\theta}{2}\\ \sin\dfrac{3\theta}{2}\end{pmatrix}=\cos\frac{3\theta}{2}|\alpha\rangle+\sin\frac{3\theta}{2}|\beta\rangle \tag{3.2.18}$$

由式(3.2.18)可以看出，经过一次算子 $\boldsymbol{G}$ 之后，得到满足条件的解$|\beta\rangle$的振幅由 $\sin\dfrac{\theta}{2}$变成 $\sin\dfrac{3\theta}{2}$，不满足条件的解$|\alpha\rangle$的振幅由 $\cos\dfrac{\theta}{2}$变成 $\cos\dfrac{3\theta}{2}$。由于在 $\left[0,\dfrac{\pi}{2}\right]$ 内正弦函数是增函数，余弦函数是减函数，因此满足条件的解的概率上升，而不满足条件的解的概率下降。下面给出更直观的几何变换过程。

图 3.10(a)～(c)给出在$|\alpha\rangle$和$|\beta\rangle$定义的平面内单次使用算子 $\boldsymbol{G}$，态$|\psi\rangle$的变化情况。图 3.10(a)是初始态$|\psi\rangle=\left(\cos\dfrac{\theta}{2}\quad\sin\dfrac{\theta}{2}\right)^{\mathrm{T}}$，与$|\alpha\rangle$轴的夹角是$\dfrac{\theta}{2}$。图 3.10(b)是将黑箱算子 $\boldsymbol{O}$ 作用在$|\psi\rangle$上的结果，由于 $\boldsymbol{O}|\psi\rangle=\left(\cos\dfrac{\theta}{2}\quad-\sin\dfrac{\theta}{2}\right)^{\mathrm{T}}$，因此 $\boldsymbol{O}|\psi\rangle$与$|\alpha\rangle$轴的夹角是$-\dfrac{\theta}{2}$。$2|\psi\rangle\langle\psi|-\boldsymbol{I}$ 是一个镜像矩阵，即当 $2|\psi\rangle\langle\psi|-\boldsymbol{I}$ 作用在$\boldsymbol{O}|\psi\rangle$上时，相当于得到 $\boldsymbol{O}|\psi\rangle$关于$|\psi\rangle$的镜像。如图 3.10(c)所示，经过算子 $2|\psi\rangle\langle\psi|-\boldsymbol{I}$ 之后，$\boldsymbol{G}|\psi\rangle=\left(\cos\dfrac{3\theta}{2}\quad\sin\dfrac{3\theta}{2}\right)^{\mathrm{T}}$ 与$|\alpha\rangle$轴的夹角是$\dfrac{3\theta}{2}$。可见，使用一次算子 $\boldsymbol{G}$，随着角度由$\dfrac{\theta}{2}$增大为

$\frac{3\theta}{2}$，$|\beta\rangle$部分的概率增大为 $\sin^2\frac{3\theta}{2}$，而$|\alpha\rangle$部分的概率减小为 $\cos^2\frac{3\theta}{2}$。从图形上看，就是相比于$|\psi\rangle$，$\boldsymbol{G}|\psi\rangle$更接近满足条件的解$|\beta\rangle$。



(a) 初始态　(b) 黑箱算子$\boldsymbol{O}$的作用　(c) $2|\psi\rangle\langle\psi|-\boldsymbol{I}$的作用

图 3.10　单次使用算子 G 的几何解释

实施 R 次算子 $\boldsymbol{G}$ 可得

$$\boldsymbol{G}^R|\psi\rangle=\begin{pmatrix}\cos\dfrac{2R+1}{2}\\ \sin\dfrac{2R+1}{2}\end{pmatrix}=\cos\left(\frac{2R+1}{2}\theta\right)|\alpha\rangle+\sin\left(\frac{2R+1}{2}\theta\right)|\beta\rangle \quad (3.2.19)$$

当 $\sin\left(\frac{2R+1}{2}\theta\right)$ 接近 1 时，就可以以接近 1 的概率得到$|\beta\rangle$，即得到满足条件的解。基于此，算子 $\boldsymbol{G}$ 也称为振幅放大算子，能够逐渐增大满足条件的解的概率。

3.2.4　算法分析

用多少次算子 $\boldsymbol{G}$ 才能以较高的概率得到满足条件的解？即把$|\psi\rangle$移动到接近$|\beta\rangle$的地方需要重复多少次算子 $\boldsymbol{G}$？

因为系统的初态为

$$|\psi\rangle=\sqrt{\frac{N-M}{N}}|\alpha\rangle+\sqrt{\frac{M}{N}}|\beta\rangle \quad (3.2.20)$$

所以旋转 $\arccos\sqrt{\frac{M}{N}}$ 弧度，系统将进入$|\beta\rangle$状态。因为每重复一次 $\boldsymbol{G}$ 算子，弧度增加 θ，则需要重复算子 $\boldsymbol{G}$ 的次数为

$$R=\left\lfloor\frac{\arccos\sqrt{\frac{M}{N}}}{\theta}\right\rfloor \quad (3.2.21)$$

式中：$\lfloor\cdot\rfloor$表示下取整。

重复 R 次算子 $\boldsymbol{G}$ 得到的量子态为式(3.2.19)。此时得到$|\beta\rangle$的概率为 $\sin^2\left(\frac{2R+1}{2}\theta\right)$，也就是说算法成功的概率为

$$P=\sin^2\left(\frac{2R+1}{2}\theta\right) \tag{3.2.22}$$

下面分析当 $M\leqslant\frac{N}{2}$ 时 R 的下界，也就是找到 R 的最小值。由于 $\arccos\sqrt{\frac{M}{N}}\leqslant\frac{\pi}{2}$，结合式(3.2.21)可得

$$R\leqslant\left\lfloor\frac{\pi}{2\theta}\right\rfloor \tag{3.2.23}$$

由于 $M\leqslant\frac{N}{2}$，则

$$\frac{\theta}{2}\geqslant\sin\frac{\theta}{2}=\sqrt{\frac{M}{N}} \tag{3.2.24}$$

因此 $R\leqslant\left\lceil\frac{\pi}{4}\sqrt{\frac{N}{M}}\right\rceil$，即复杂度为 $O(\sqrt{N})$。

由式(3.2.23)可知 R 为属于 $\left(\frac{\pi}{2\theta}-1,\frac{\pi}{2\theta}\right]$ 的整数，且 $\left|R-\frac{\arccos\sqrt{\frac{M}{N}}}{\theta}\right|\leqslant\frac{1}{2}$，则

$$\arccos\sqrt{\frac{M}{N}}\leqslant\frac{2R+1}{2}\theta\leqslant\theta+\arccos\sqrt{\frac{M}{N}} \tag{3.2.25}$$

因为 $\cos\frac{\theta}{2}=\sqrt{\frac{N-M}{N}}$，所以 $\arccos\sqrt{\frac{M}{N}}=\frac{\pi}{2}-\frac{\theta}{2}$，且

$$\frac{\pi}{2}-\frac{\theta}{2}\leqslant\frac{2R+1}{2}\theta\leqslant\frac{\pi}{2}+\frac{\theta}{2} \tag{3.2.26}$$

又因为 $M\leqslant\frac{N}{2}$，所以

$$P=\sin^2\left(\frac{2R+1}{2}\theta\right)\geqslant\cos^2\frac{\theta}{2}=\frac{N-M}{N}\geqslant\frac{1}{2} \tag{3.2.27}$$

事实上，当 $\frac{N}{2}<M\leqslant N$ 时，由式(3.2.22)的曲线图(可参见文献[2])可知执行一次算子 $\boldsymbol{G}$ 就能达到大于 $\frac{1}{2}$ 的成功率。

综上，Grover 搜索算法执行 $O(\sqrt{N})$ 次算子 $\boldsymbol{G}$ 就能达到大于 $\frac{1}{2}$ 的成功率。因此该算法的复杂度为 $O(\sqrt{N})$。

3.2.5 实现

本实验要从 $\boldsymbol{X}=(x_0\quad x_1)$ 的四个状态{(0 0),(0 1),(1 0),(1 1)}中找到(1 1)。图 3.11 是 Grover 算法的线路图，量子比特 q_0 和 q_1 分别对应 x_0 和 x_1，q_2 是辅助量子比特，用于黑箱算子中对目标量子态进行翻转。

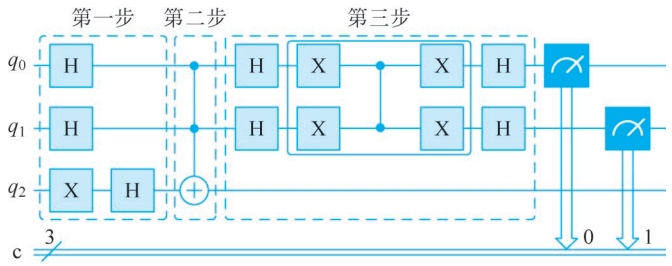


图 3.11　Grover 搜索算法量子线路图

第一步是制备量子态以及辅助量子比特初始状态；第二步实现黑箱算子 $\boldsymbol{O}$，将(1 1)的符号取反；第三步实现镜像矩阵 $2|\psi\rangle\langle\psi|-\boldsymbol{I}$。由式(3.2.9)可知，理论上镜像矩阵由 $\boldsymbol{H}^{\otimes 2}$、$2|0\rangle^{\otimes 2}\langle 0|^{\otimes 2}-\boldsymbol{I}$ 和 $\boldsymbol{H}^{\otimes 2}$ 组成：

$$2|\psi\rangle\langle\psi|-\boldsymbol{I}=\boldsymbol{H}^{\otimes 2}(2|0\rangle^{\otimes 2}\langle 0|^{\otimes 2}-\boldsymbol{I})\boldsymbol{H}^{\otimes 2} \tag{3.2.28}$$

式中

$$2|0\rangle^{\otimes 2}\langle 0|^{\otimes 2}-\boldsymbol{I}=2\begin{pmatrix}1\\0\\0\\0\end{pmatrix}(1\quad 0\quad 0\quad 0)-\boldsymbol{I}=\begin{pmatrix}2&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}-\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

$$=\begin{pmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{pmatrix}=\begin{pmatrix}\boldsymbol{Z}&\boldsymbol{0}\\\boldsymbol{0}&-\boldsymbol{I}\end{pmatrix} \tag{3.2.29}$$

在图 3.11 的第三步中，除去两边的 $\boldsymbol{H}^{\otimes 2}$，实线中的部分实现的就是 $2|0\rangle^{\otimes 2}\langle 0|^{\otimes 2}-\boldsymbol{I}$。其中两个点被一个竖线相连的门是受控 $\boldsymbol{Z}$ 门，又称为 $\boldsymbol{CZ}$ 门，即当 q_0 为 $|1\rangle$ 时，将 $\boldsymbol{Z}$ 门作用于 q_1。因此，第三步的实线框中实现的算子为

$$(\boldsymbol{X}\otimes\boldsymbol{X})(\boldsymbol{CZ})(\boldsymbol{X}\otimes\boldsymbol{X})=\left(\begin{pmatrix}0&1\\1&0\end{pmatrix}\otimes\begin{pmatrix}0&1\\1&0\end{pmatrix}\right)\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{pmatrix}\left(\begin{pmatrix}0&1\\1&0\end{pmatrix}\otimes\begin{pmatrix}0&1\\1&0\end{pmatrix}\right)$$

$$=\begin{pmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}=-\begin{pmatrix}\boldsymbol{Z}&\boldsymbol{0}\\\boldsymbol{0}&-\boldsymbol{I}\end{pmatrix} \tag{3.2.30}$$

可以看到，这与式(3.2.29)中要实现的 $2|0\rangle^{\otimes 2}\langle 0|^{\otimes 2}-\boldsymbol{I}=\begin{pmatrix}\boldsymbol{Z}&\boldsymbol{0}\\\boldsymbol{0}&-\boldsymbol{I}\end{pmatrix}$ 存在一个负号的差别。这个负号是一个全局相位，作用于量子态之后不会引起测量结果的任何变化，因此负号可以忽略。

线路的最后是测量。由图 3.12 可以看出测量结果 011 出现的概率为 1，即以 100% 的概率找到(1 1)。需要说明的是测量结果 011 分别代表量子比特 $q_2q_1q_0$ 的值，q_2 其实

没有被测量，显示默认值 0，q_1q_0 测量结果为 11，表示成功找到(1 1)。

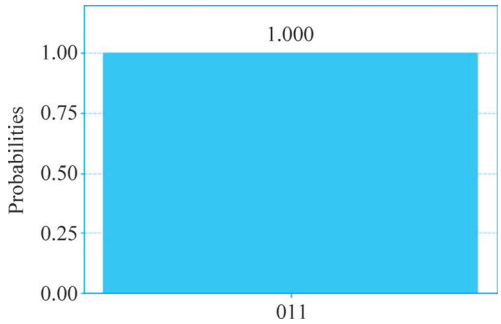


图 3.12　实验结果

Grover 搜索算法代码如下：

```
%matplotlib inline
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute
from qiskit import Aer
from qiskit import IBMQ
from math import pi
import numpy as np
from qiskit.tools.visualization import plot_histogram

circuit = QuantumCircuit(3,3)

#第一步
circuit.x(2)
for i in range(3):
    circuit.h(i)

#第二步
circuit.ccx(0,1,2)

#第三步
for i in range(2):
    circuit.h(i)
for i in range(2):
    circuit.x(i)
circuit.cz(0,1)
for i in range(2):
    circuit.x(i)
for i in range(2):
    circuit.h(i)

#测量
circuit.measure(0,0)
circuit.measure(1,1)

#绘制线路图
circuit.draw(output = 'mpl')
```

```
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit,backend,shots = 20000)
sim_result = job_sim.result()

#绘制结果图
measurement_result = sim_result.get_counts(circuit)
plot_histogram(measurement_result)
```

3.3 量子傅里叶变换

经典傅里叶变换在信号处理、密码学、统计学等很多领域中都起着非常重要的作用。量子傅里叶变换(Quantum Fourier Transform,QFT)是由经典傅里叶变换衍生而来的,除了具有像经典傅里叶变换一样的频域变换功能,还具有了一些新的功能。

3.3.1 离散傅里叶变换原理

在经典的数学或计算机科学中,人们通常将要解决的问题转换为更容易解决的问题。离散傅里叶变换就是这样一个转换工具,能够把信号从时域变换到频域。本书对傅里叶变换原理不做过多的描述,只给出定义。在离散傅里叶变换中,输入是一个长度为 N 的复向量$(x_0 \quad x_1 \quad \cdots \quad x_{N-1})$,输出是相同长度的复向量$(y_0 \quad y_1 \quad \cdots \quad y_{N-1})$,$y_k(k=0,1,\cdots,N-1)$定义如下:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, \mathrm{e}^{2\pi i j k / N} \tag{3.3.1}$$

3.3.2 量子傅里叶变换算法

【定义 3.3.1】 对于一组标准正交基$|0\rangle,|1\rangle,\cdots,|N-1\rangle$,量子傅里叶变换为作用于基态的线性组合。经过量子傅里叶变换之后,基态$|j\rangle(j = 0,1,\cdots,N-1)$演化为

$$\mathbf{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \mathrm{e}^{2\pi i j k / N} \,|k\rangle \tag{3.3.2}$$

对于任意量子态$\sum_{j=0}^{N-1} x_j |j\rangle$来说,其量子傅里叶变换为$\sum_{k=0}^{N-1} y_k |k\rangle$,其中 y_k 就是 x_j 的离散傅里叶变换。推导过程:

$$\begin{aligned}
\mathbf{QFT}\Big(\sum_{j=0}^{N-1} x_j \,|j\rangle\Big) &= \sum_{j=0}^{N-1} x_j \,\mathbf{QFT}\,|j\rangle \\
&= \sum_{j=0}^{N-1} x_j \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \mathrm{e}^{2\pi i j k / N} \,|k\rangle\right) \\
&= \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, \mathrm{e}^{2\pi i j k / N}\right) |k\rangle \\
&= \sum_{k=0}^{N-1} y_k \,|k\rangle
\end{aligned} \tag{3.3.3}$$

量子傅里叶变换是酉变换，可以用两种方法来证明：一种是通过矩阵的形式验证式(3.3.2)是酉变换；另一种是将其分解成简单的量子酉变换的张量积形式，从而验证量子傅里叶变换的酉性。下面介绍第二种方法。

令 $N=2^n$（n 是正整数），则基态可以写为 $|0\rangle,|1\rangle,\cdots,|2^n-1\rangle$。这样做便于将基态 $|j\rangle$ 写成二进制的形式 $j=j_1j_2\cdots j_n$，即 $j=j_1\times2^{n-1}+j_2\times2^{n-2}+\cdots+j_n\times2^0$。如果是小数，那么二进制小数 $0.j_1j_2\cdots j_n$ 可以表示为 $\frac{j_1}{2}+\frac{j_2}{4}+\cdots+\frac{j_n}{2^n}$。因此，式(3.3.2)所示的量子傅里叶变换等价于以下张量积形式：

$$
\begin{aligned}
&\mathbf{QFT}\,|j\rangle\\
&=\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}jk/N}\,|k\rangle\\
&=\frac{1}{\sqrt{2^n}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1}\mathrm{e}^{2\pi\mathrm{i}j\left(\sum_{l=1}^{n}k_l2^{-l}\right)}\,|k_1\cdots k_n\rangle\\
&=\frac{1}{\sqrt{2^n}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1}\bigotimes_{l=1}^{n}\mathrm{e}^{2\pi\mathrm{i}jk_l2^{-l}}\,|k_l\rangle\\
&=\frac{1}{\sqrt{2^n}}\bigotimes_{l=1}^{n}\left(\sum_{k_l=0}^{1}\mathrm{e}^{2\pi\mathrm{i}jk_l2^{-l}}\,|k_l\rangle\right)\\
&=\frac{1}{\sqrt{2^n}}\bigotimes_{l=1}^{n}\left(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}j2^{-l}}\,|1\rangle\right)\\
&=\frac{1}{\sqrt{2^n}}\left[\left(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.j_n}\,|1\rangle\right)\left(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.j_{n-1}j_n}\,|1\rangle\right)\cdots\left(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.j_1j_2\cdots j_{n-1}j_n}\,|1\rangle\right)\right]
\end{aligned}
\tag{3.3.4}
$$

式中：“$\bigotimes_{l=1}^{n}$”为 n 项的张量积。

式(3.3.4)表明，量子傅里叶变换的输出可以写成张量积的形式。因此，可以按照张量积形式构建量子傅里叶变换的量子线路图，如图 3.13 所示，其中相位变换算子 $\boldsymbol{R}_k$ 用于改变量子态的相位，定义为

$$
\boldsymbol{R}_k=\begin{pmatrix}1 & 0\\ 0 & \mathrm{e}^{2\pi\mathrm{i}/2^k}\end{pmatrix}
\tag{3.3.5}
$$

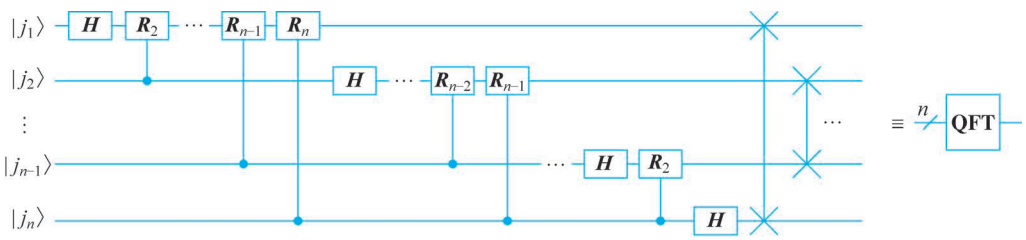


图 3.13　量子傅里叶变换的线路图

图 3.13 所示的量子线路图的输入为基态$|j_1 j_2 \cdots j_n\rangle$。

当$j_1=0$时，$\boldsymbol{H}$门作用于第一个量子比特产生量子态：

$$\boldsymbol{H}\,|j_1\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}}=\frac{|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1}\,|1\rangle}{\sqrt{2}} \tag{3.3.6}$$

当$j_1=1$时，$\boldsymbol{H}$门作用于第一个量子比特产生量子态：

$$\begin{aligned}\boldsymbol{H}\,|j_1\rangle&=\frac{|0\rangle-|1\rangle}{\sqrt{2}}=\frac{|0\rangle+\mathrm{e}^{\pi \mathrm{i}}\,|1\rangle}{\sqrt{2}}\\&=\frac{|0\rangle+\mathrm{e}^{2\pi \mathrm{i}1/2}\,|1\rangle}{\sqrt{2}}=\frac{|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1}\,|1\rangle}{\sqrt{2}}\end{aligned} \tag{3.3.7}$$

因此，经过$\boldsymbol{H}$门之后，系统的状态变为

$$\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1}\,|1\rangle)\,|j_2\cdots j_n\rangle \tag{3.3.8}$$

经过受控$\boldsymbol{R}_2$门之后，系统的状态变为

$$\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1}\mathrm{e}^{2\pi \mathrm{i}j_2/2^2}\,|1\rangle)\,|j_2\cdots j_n\rangle=\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1 j_2}\,|1\rangle)\,|j_2\cdots j_n\rangle \tag{3.3.9}$$

继续使用受控$\boldsymbol{R}_3$、$\boldsymbol{R}_4$直至$\boldsymbol{R}_n$门可得

$$\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1 j_2\cdots j_n}\,|1\rangle)\,|j_2\cdots j_n\rangle \tag{3.3.10}$$

对第二个量子比特执行同样的操作。首先用$\boldsymbol{H}$门作用于第二个量子比特得到量子态

$$\frac{1}{\sqrt{2^2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1 j_2\cdots j_n}\,|1\rangle)(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_2}\,|1\rangle)\,|j_3\cdots j_n\rangle \tag{3.3.11}$$

再使用受控$\boldsymbol{R}_2$、$\boldsymbol{R}_3$直至$\boldsymbol{R}_{n-1}$门可得

$$\frac{1}{\sqrt{2^2}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1 j_2\cdots j_n}\,|1\rangle)(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_2\cdots j_n}\,|1\rangle)\,|j_3\cdots j_n\rangle \tag{3.3.12}$$

对后面$n-2$个量子比特执行同样的操作得到量子态

$$\frac{1}{\sqrt{2^n}}(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_1 j_2\cdots j_n}\,|1\rangle)(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_2\cdots j_n}\,|1\rangle)\cdots(|0\rangle+\mathrm{e}^{2\pi \mathrm{i}0.j_n}\,|1\rangle) \tag{3.3.13}$$

至此，除了量子比特的顺序颠倒之外，式(3.3.13)与式(3.3.4)完全相同。利用交换门将第一个量子比特和最后一个量子比特交换，第二个量子比特和倒数第二个量子比特交换，以此类推，使用$O(n)$个交换门就可以得到正确的顺序。

因为实现量子傅里叶变换的线路中所有量子门均为酉门，所以量子傅里叶变换也是酉变换。酉变换的逆等于其共轭转置，因此存在量子傅里叶逆变换，记为$\mathbf{QFT}^+$。具体到量子线路中，$\mathbf{QFT}^+$的线路就是将图 3.13 中所有的量子门的顺序反过来。图 3.14 给出量子傅里叶逆变换的线路图。

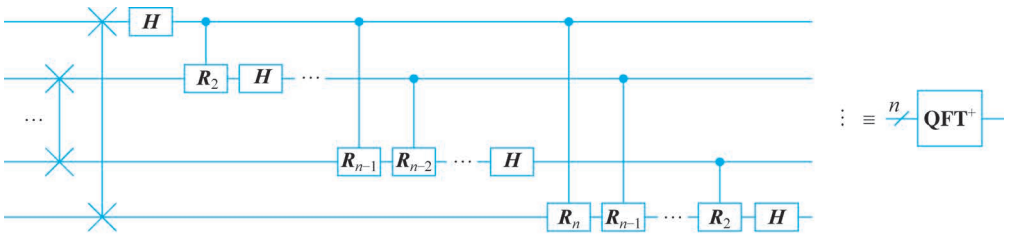


图 3.14　量子傅里叶逆变换的线路图

由式(3.3.4)可以看出，量子傅里叶变换将存储在基态中的信息$|j\rangle=|j_1j_2\cdots j_n\rangle$转移到振幅$e^{2\pi i0.j_1j_2\cdots j_{n-1}j_n}$中，而量子傅里叶逆变换能够将存储在振幅中的信息转移到基态中。也就是说，**QFT** 和 $\mathbf{QFT}^+$能够将数据在基态存储和振幅存储两种形式之间进行转换，这也是量子傅里叶变换相对于经典傅里叶变换的新功能。

对量子傅里叶变换的复杂度进行分析。在图 3.13 中共使用 n 个 $\boldsymbol{H}$ 门和$\frac{n(n-1)}{2}$个受控旋转门；此外，还有至多$\frac{n}{2}$个交换门，而每一个交换门由 3 个 CNOT 组成。因此，计算一次量子傅里叶变换需要 $O(n^2)$个基本门，而最有效的经典离散傅里叶变换需要$O(2^n n)$个基本逻辑门。所以在量子计算机上执行量子傅里叶变换的复杂度为在经典计算机上所需复杂度的对数级。

但是，量子计算机不能直接输出，无法确定量子态的振幅，数据读出比较困难，因此 **QFT** 并没有在真正意义上加速经典算法。不过，研究者已经发现 **QFT** 可用于量子相位估计等算法中，因此 **QFT** 在量子机器学习中发挥了重要的作用。

3.3.3　实现

本实验对量子态$|j_1j_2j_3j_4\rangle=|0101\rangle$做量子傅里叶变换。图 3.15 为量子傅里叶变换的线路图。

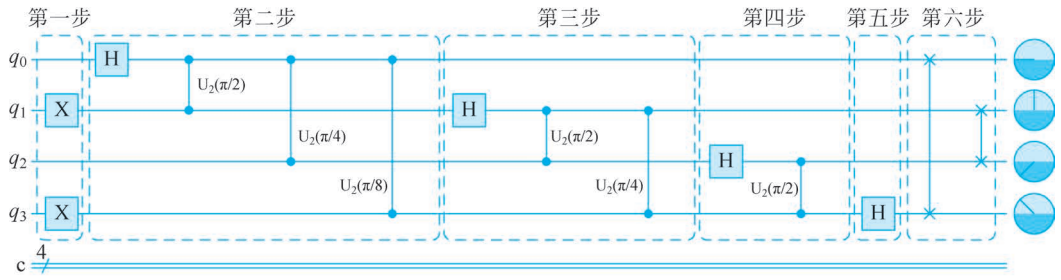


图 3.15　量子傅里叶变换的线路图

第一步为状态准备，即制备初始态$|0101\rangle$；第二步至第六步为量子傅里叶变换，在这些步骤中，要使用算子$\boldsymbol{R}_k=\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^k}\end{pmatrix}$，但是 qiskit 上没有此算子，因此使用自带的

$\boldsymbol{U}_1(\lambda)=\begin{pmatrix}1 & 0\\ 0 & \mathrm{e}^{\mathrm{i}\lambda}\end{pmatrix}$代替，对于不同的 $\boldsymbol{R}_k$ 门，令 $\boldsymbol{U}_1(\lambda)$中的 $\lambda=2\pi/2^k$ 即可。经过第二步之后得到的量子态为

$$\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.0101}|1\rangle)|101\rangle=\frac{1}{\sqrt{2}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{16}}|1\rangle)|101\rangle \tag{3.3.14}$$

经过第三步之后得到的量子态为

$$\begin{aligned}&\frac{1}{\sqrt{2^2}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.0101}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.101}|1\rangle)|01\rangle\\ =&\frac{1}{\sqrt{2^2}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{16}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{8}}|1\rangle)|01\rangle\end{aligned} \tag{3.3.15}$$

经过第四步之后得到的量子态为

$$\begin{aligned}&\frac{1}{\sqrt{2^3}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.0101}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.101}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.01}|1\rangle)|1\rangle\\ =&\frac{1}{\sqrt{2^3}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{16}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{8}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{1}{4}}|1\rangle)|1\rangle\end{aligned} \tag{3.3.16}$$

经过第五步之后得到的量子态为

$$\begin{aligned}&\frac{1}{\sqrt{2^4}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.0101}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.101}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.01}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}0.1}|1\rangle)\\ =&\frac{1}{\sqrt{2^4}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{16}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{8}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{1}{4}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{1}{2}}|1\rangle)\end{aligned} \tag{3.3.17}$$

经过第六步之后得到的量子态为

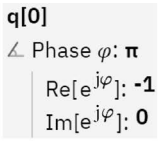
$$\begin{aligned}&\frac{1}{\sqrt{2^4}}(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{1}{2}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{1}{4}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{8}}|1\rangle)(|0\rangle+\mathrm{e}^{2\pi\mathrm{i}\frac{5}{16}}|1\rangle)\\ =&\frac{1}{\sqrt{2^4}}(|0\rangle+\mathrm{e}^{\pi\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{\pi}{2}\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{5\pi}{4}\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{5\pi}{8}\mathrm{i}}|1\rangle)\end{aligned} \tag{3.3.18}$$

由式(3.3.18)可以看出，q_0、q_1、q_2 和 q_3 都处于状态$|0\rangle$和$|1\rangle$的叠加，其中$|0\rangle$的相位全为0，$|1\rangle$的相位分别变为 π、$\frac{\pi}{2}$、$\frac{5\pi}{4}$和$\frac{5\pi}{8}$。

此外，在 qiskit 可以查看 q_0、q_1、q_2 和 q_3 处于状态$|1\rangle$的相位，如图 3.15 最右边的圆圈所示。4 个圆圈内部的短线，其倾斜的角度给出对应的量子比特的相位。这是 qiskit 给出的一个工具，当把光标放在圆圈上时，会显示相位的精确数值，如图 3.16 所示。

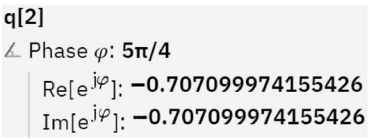
查看之后可以得到相位分别变为 π、$\frac{\pi}{2}$、$\frac{5\pi}{4}$和 $1.9635\approx\frac{5\pi}{8}$。也就是说，量子傅里叶变换之后，初始态$|0101\rangle$演化为

$$\frac{1}{\sqrt{2^4}}(|0\rangle+\mathrm{e}^{\pi\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{\pi}{2}\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{5\pi}{4}\mathrm{i}}|1\rangle)(|0\rangle+\mathrm{e}^{\frac{5\pi}{8}\mathrm{i}}|1\rangle) \tag{3.3.19}$$

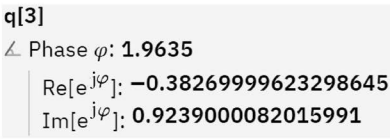


(a) q_0的相位

(b) q_1的相位



(c) q_2的相位



(d) q_3的相位

图 3.16　相位的精确数值

可以看出理论结果(式(3.3.18))与实验结果(式(3.3.19))是一样的。因此,可以通过查看相位得到量子傅里叶变换之后的结果。

对量子态|0101⟩做量子傅里叶变换的代码如下:

```
%matplotlib inline
from qiskit import QuantumCircuit
from qiskit import execute
from qiskit import IBMQ
from qiskit.tools.visualization import plot_histogram
import math

circuit = QuantumCircuit(4, 4)

#第一步
circuit.x(1)
circuit.x(3)

#第二步~第五步,定义函数 qft_rotations
circuit.barrier()
def qft_rotations(circuit, n, nu):
    if n == 0:
        return circuit
    n -= 1
    nu += 1
    circuit.h(3 - n)
    for qubit in range(n):
        circuit.cu1(math.pi/2 ** (qubit + 1), qubit + nu, 3 - n)
    circuit.barrier()
    qft_rotations(circuit, n, nu)
#调用函数 qft_rotations
qft_rotations(circuit,4,0)

#第六步
circuit.swap(0,3)
circuit.swap(1,2)

#绘制线路图
circuit.draw(output = 'mpl',plot_barriers = False)
```

3.4 量子相位估计

在量子计算中算子 $\boldsymbol{U}$ 都是酉矩阵，因此有特征值及对应的特征向量。假设 $\boldsymbol{U}$ 的特征值为 $e^{2\pi i\varphi}$，相应的特征向量为 $|u\rangle$，量子相位估计(Quantum Phase Estimation，QPE)利用 $\boldsymbol{U}|u\rangle=e^{2\pi i\varphi}|u\rangle$ 估计特征值 $e^{2\pi i\varphi}$ 的相位 $2\pi\varphi$。由于 2π 是常数，只要能够估计出 φ，就能完成相位估计，因此也将 φ 称为相位。量子相位估计算法在量子支持向量机、量子线性回归等算法中起着重要作用。

3.4.1 算法

量子相位估计算法以式 $\boldsymbol{U}|u\rangle=e^{2\pi i\varphi}|u\rangle$ 为基础，因此 QPE 有 $\boldsymbol{U}$ 和 $|u\rangle$ 两个输入，其中 $|u\rangle$ 存储到量子比特中，而 $\boldsymbol{U}$ 体现为相位估计算法中的一个算子。整个算法用到两个寄存器：第一寄存器用 t 个量子比特存储最终计算出来的特征值的相位 φ，也就是说第一寄存器是算法的输出；第二寄存器用 m 个量子比特存储特征向量 $|u\rangle$，这是算法的输入。算法的初始量子态为 $|0\rangle^{\otimes t}|0\rangle^{\otimes m}$。图 3.17 给出量子相位估计算法的线路图。

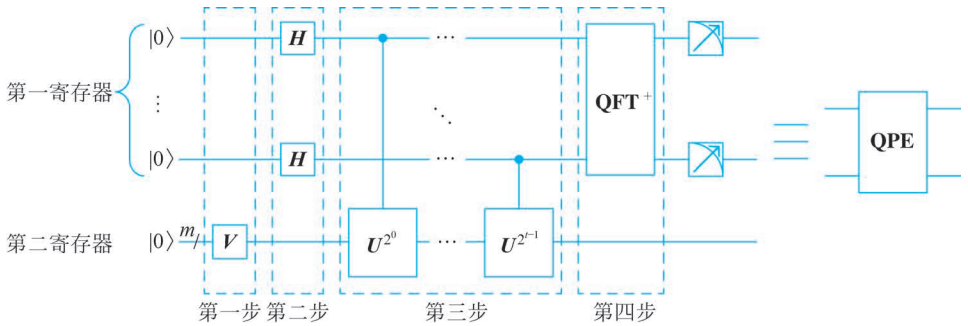


图 3.17 量子相位估计算法的线路图

第一步：使用酉变换 $\boldsymbol{V}$ 作用于第二寄存器，制备特征向量 $|u\rangle$，得到量子态 $|\psi_1\rangle$，即

$$|\psi_1\rangle=|0\rangle^{\otimes t}|u\rangle \tag{3.4.1}$$

第二步：利用 t 个 $\boldsymbol{H}$ 门作用于 $|\psi_1\rangle$ 的第一寄存器，构建叠加态 $|\psi_2\rangle$，即

$$|\psi_2\rangle=\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1}|k\rangle|u\rangle \tag{3.4.2}$$

第三步：受第一寄存器中第 $j(j=0,1,\cdots,t-1)$ 个量子比特的控制，将算子 $\boldsymbol{U}^{2^j}$ 作用于第二寄存器，将相位转移到第一寄存器的振幅中。这里 $\boldsymbol{U}^{2^j}$ 的含义是实施 2^j 次算子 $\boldsymbol{U}$。记 $k_1k_2\cdots k_t$ 为 k 的二进制形式，则 $k=k_12^{t-1}+k_22^{t-2}+\cdots+k_t2^0$，这里 $k_t\cdots k_2k_1$ 依次是第一寄存器中从上到下的第 $1,\cdots$，第 $t-1$，第 t 个量子比特。

受第一寄存器中第 j 个量子比特的控制，将矩阵 $\boldsymbol{U}^{2^j}$ 作用于第二寄存器这种操作记

为 $\boldsymbol{U}^{k_j 2^{t-j}}$。这种表示方法的含义：当控制位 $k_j=0$ 时，$\boldsymbol{U}^{k_j 2^{t-j}}=\boldsymbol{U}^0$，相当于不进行 $\boldsymbol{U}$ 操作；当控制位 $k_j=1$ 时，$\boldsymbol{U}^{k_j 2^{t-j}}=\boldsymbol{U}^{2^{t-j}}$，要进行 $\boldsymbol{U}^{2^{t-j}}$ 操作。因此，量子态 $|\psi_2\rangle$ 演化为

$$
\begin{aligned}
|\psi_3\rangle &= \frac{1}{2^{t/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_t=0}^{1}|k_1\cdots k_t\rangle\otimes\boldsymbol{U}^{k_1 2^{t-1}}\boldsymbol{U}^{k_2 2^{t-2}}\cdots\boldsymbol{U}^{k_t 2^0}|u\rangle \\
&= \frac{1}{2^{t/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_t=0}^{1}|k_1\cdots k_t\rangle\otimes\boldsymbol{U}^{k_1 2^{t-1}+k_2 2^{t-2}+\cdots+k_t 2^0}|u\rangle
\end{aligned}
\tag{3.4.3}
$$

又因为 $e^{2\pi i\varphi}$ 是 $\boldsymbol{U}$ 的特征值，$|u\rangle$ 是相应的特征向量，即 $\boldsymbol{U}|u\rangle=e^{2\pi i\varphi}|u\rangle$，则式(3.4.3)可以重写为

$$
\begin{aligned}
|\psi_4\rangle &= \frac{1}{2^{t/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_t=0}^{1}|k_1\cdots k_t\rangle\otimes e^{2\pi i\varphi\sum_{l=1}^{t}k_l 2^{t-l}}|u\rangle \\
&= \frac{1}{2^{t/2}}\sum_{k_1=0}^{1}\cdots\sum_{k_t=0}^{1}\bigotimes_{l=1}^{t}e^{2\pi i\varphi k_l 2^{t-l}}|k_l\rangle|u\rangle \\
&= \frac{1}{2^{t/2}}\bigotimes_{l=1}^{t}(|0\rangle+e^{2\pi i\varphi 2^{t-l}}|1\rangle)|u\rangle
\end{aligned}
\tag{3.4.4}
$$

此时，相位 φ 存储在 $|1\rangle$ 的振幅中。由于 φ 用 t 比特表示，即

$$
\varphi=0.\varphi_1\cdots\varphi_t=\frac{\varphi_1\times 2^{t-1}+\cdots+\varphi_t\times 2^0}{2^t}
$$

则式(3.4.4)中第一寄存器的状态可以表示为

$$
\frac{1}{\sqrt{2^t}}(|0\rangle+e^{2\pi i0.\varphi_t}|1\rangle)(|0\rangle+e^{2\pi i0.\varphi_{t-1}\varphi_t}|1\rangle)\cdots(|0\rangle+e^{2\pi i0.\varphi_1\cdots\varphi_{t-1}\varphi_t}|1\rangle)
\tag{3.4.5}
$$

第四步：利用量子傅里叶逆变换作用于第一寄存器将存储在振幅中的相位 φ 转移到基态中，即

$$
\mathbf{QFT}^{+}\left[\frac{1}{2^{t/2}}\bigotimes_{l=1}^{t}(|0\rangle+e^{2\pi i\varphi 2^{t-l}}|1\rangle)\right]=|\varphi_1\cdots\varphi_t\rangle
\tag{3.4.6}
$$

进而通过简单的计算可得

$$
\varphi=\frac{\varphi_1\times 2^{t-1}+\cdots+\varphi_t\times 2^0}{2^t}
$$

量子相位估计算法通常更简洁地表示为图 3.18 的形式。

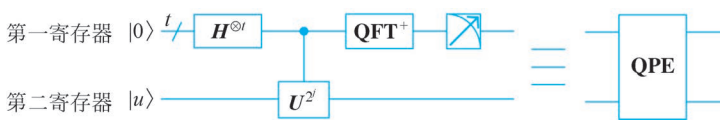


图 3.18　量子相位估计的简写形式

事实上，大多数情况下 t 个比特并不能精确的表示相位 φ，而且最后要通过测量才能得到 φ，也会存在误差。因此，最后得到的是 φ 的近似值 $\tilde{\varphi}$。下面分析第一寄存器中量子比特的数量与误差之间的关系。

假设 b 是 t 比特二进制数，且 $\frac{b}{2^t}=0.b_1\cdots b_t$ 是相位 φ 的最优下近似，也就是说 $\frac{b}{2^t}$ 和 φ 之间的差 $\delta=\varphi-\frac{b}{2^t}$ 满足 $0\leqslant\delta\leqslant\frac{1}{2^t}$。量子相位估计算法的输出记为 m，下面给出一个定理说明 m 与 b 之间的误差与第一寄存器量子比特数量之间的关系。

【定理 3.4.1】 对于任意的 $\varepsilon>0$，假设存在一个正整数 $c=2^{t-n}-1$，使得 $|m-b|<c$，则测量得到 m 的概率为

$$p(|m-b|<c)=\varepsilon\leqslant\frac{1}{2(c-1)}=\frac{1}{2(2^{t-n}-2)} \tag{3.4.7}$$

式中：t 为第一寄存器量子比特的数量；n 为能使 φ 达到 2^{-n} 精度的量子比特的数量。

定理证明参见文献[1]。

由定理 3.4.1 可以看出，为了至少以 $1-\varepsilon$ 的成功概率精确到 n 比特，在量子相位估计算法中，第一寄存器的比特数为

$$t=n+\left\lceil\log\left(2+\frac{1}{2\varepsilon}\right)\right\rceil \tag{3.4.8}$$

QPE 算法中第一步制备的是特征向量 $|u\rangle$，但是要想制备 $|u\rangle$，先得把特征向量 u 计算出来，再制备为量子态，整个过程并不容易，因此需要给出一种更简单的方法。因为酉矩阵的特征向量 $\{|u_j\rangle\}_{j=0}^{N-1}$ 可以做一组基，因此任意的 N 维量子态 $|b\rangle$ 都可以表示为 $|b\rangle=\sum_{i=0}^{N-1}\beta_i|u_i\rangle$，其中 β_i 是系数。又因为 $|u_i\rangle$ 可以表示为标准正交基 $\{|j\rangle\}_{j=0}^{N-1}$ 的线性组合，即 $|u_i\rangle=\sum_{j=0}^{N-1}c_{ij}|j\rangle$，其中 c_{ij} 是系数，则

$$\begin{aligned}|b\rangle&=\sum_{i=0}^{N-1}\beta_i|u_i\rangle=\sum_{i=0}^{N-1}\beta_i\left(\sum_{j=0}^{N-1}c_{ij}|j\rangle\right)\\&=\sum_{i=0}^{N-1}\left(\sum_{j=0}^{N-1}\beta_i c_{ij}|j\rangle\right)=\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}\beta_i c_{ij}\right)|j\rangle\end{aligned} \tag{3.4.9}$$

因此，$|b\rangle$ 可以表示为标准正交基的线性组合。令 $b_j=\sum_{i=0}^{N-1}\beta_i c_{ij}$，则 $|b\rangle=\sum_{j=0}^{N-1}b_j|j\rangle$，也就是说，$|b\rangle=\sum_{i=0}^{N-1}\beta_i|u_i\rangle$ 的制备可以转换为在标准正交基下展开后进行制备。

例如，当 $|b\rangle=|u_0\rangle+0|u_1\rangle+\cdots+0|u_{N-1}\rangle=|u_0\rangle$ 时，记 $|u_0\rangle=|u\rangle$ 是一个特征向量，则 $|b\rangle=|u\rangle=\sum_{j=0}^{N-1}b_j|j\rangle$，因此 $|u\rangle$ 的输入为 $\sum_{j=0}^{N-1}b_j|j\rangle$，也就是制备 $|b\rangle$ 在标准正

交基下的展开式即可。

为方便起见，在之后用到相位估计的算法中，理论中输入使用 $|b\rangle=\sum_{i=0}^{N-1}\beta_i|u_i\rangle$，实验中输入使用 $|b\rangle=\sum_{j=0}^{N-1}b_j|j\rangle$。

量子相位估计算法总结如下：

量子相位估计算法 QPE

输入：$|0\rangle^{\otimes(t+m)}$

过程：

(1) 使用 $\boldsymbol{V}$ 门作用于第二寄存器产生量子态 $|0\rangle^{\otimes t}|u\rangle$；

(2) 使用 t 个 $\boldsymbol{H}$ 门作用于第一寄存器 $\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle$；

(3) 使用受控 $\boldsymbol{U}$ 算子将相位 φ 转移到第一寄存器的概率幅中，$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\mathrm{e}^{2\pi ij\varphi}|j\rangle|u\rangle$；

(4) 执行量子傅里叶逆变换将相位转移到基态中，$|\widetilde{\varphi}\rangle|u\rangle$；

(5) 对第一寄存器的量子比特测量。

输出：$\widetilde{\varphi}$。

3.4.2 实现

本实验使用量子相位估计求出 $\boldsymbol{U}_1\left(\frac{\pi}{4}\right)=\begin{pmatrix}1 & 0\\ 0 & \mathrm{e}^{\mathrm{i}\pi/4}\end{pmatrix}$ 的特征值的相位。在式(3.4.8)中，令 $n=3$，$\varepsilon=0.1$，则第一寄存器中量子比特的数量为 3。由于

$$\begin{pmatrix}1 & 0\\ 0 & \mathrm{e}^{\mathrm{i}\pi/4}\end{pmatrix}|1\rangle=\mathrm{e}^{\mathrm{i}\pi/4}|1\rangle=\mathrm{e}^{2\pi\mathrm{i}\frac{1}{8}}|1\rangle$$

因此特征向量为 $|1\rangle$，相位为 $\frac{1}{8}$。本实验就是要将这个 $\frac{1}{8}$ 估计出来。相位估计的量子线路如图 3.19 所示，共用 4 个量子比特，前 3 个量子比特是第一寄存器，第 4 个量子比特是第二寄存器。

第一步将 $\boldsymbol{X}$ 门作用在第二寄存器上，产生特征向量 $|1\rangle$；第二步使用 3 个 $\boldsymbol{H}$ 门产生叠加态；第三步执行受控 $\boldsymbol{U}_1$ 操作；第四步执行量子傅里叶逆变换。在量子相位估计算法中，只执行一次量子线路便能得到实验结果。本实验中，为了验证算法准确率，执行了 1024 次。测量结果如图 3.20 所示，可以看出，算法以 100%的概率得到 001，进而可得相位为

$$\varphi=\frac{0\times2^2+0\times2^1+1\times2^0}{2^3}=\frac{1}{8}$$

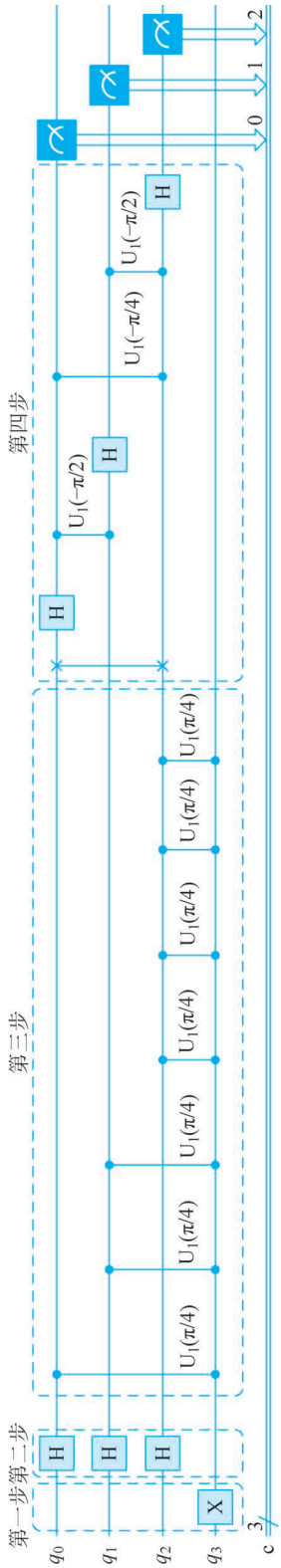


图 3.19　相位估计的量子线路图

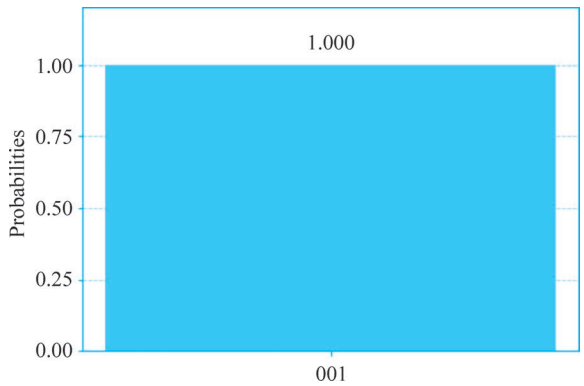


图 3.20　测量结果

量子相位估计的代码如下：

```
%matplotlib inline
from qiskit import QuantumCircuit
from qiskit import execute
from qiskit import IBMQ
from qiskit.tools.visualization import plot_histogram
import math

circuit = QuantumCircuit(4, 3)

#第一步：量子态制备
circuit.x(3)

#第二步
circuit.barrier()
for qubit in range(3):
    circuit.h(qubit)

#第三步：受控酉操作
repetitions = 1
for counting_qubit in range(3):
    for i in range(repetitions):
        circuit.cu1(math.pi/4, counting_qubit, 3);
    repetitions *= 2

#第四步：量子傅里叶逆变换
def qft_dagger(qc, n):
    for qubit in range(n//2):
        qc.swap(qubit, n-qubit-1)
    for j in range(n):
        for m in range(j):
            qc.cu1(-math.pi/float(2**(j-m)), m, j)
        qc.h(j)
circuit.barrier()
qft_dagger(circuit, 3)
```

```
35.
36.  #测量
37.  circuit.barrier()
38.  for n in range(3):
39.      circuit.measure(n,n)
40.
41.  #绘制线路图
42.  circuit.draw(output = 'mpl',plot_barriers = False,fold = -1)
43.  backend = Aer.get_backend('qasm_simulator')
44.  job_sim = execute(circuit, backend, shots = 8192)
45.  sim_result = job_sim.result()
46.
47.  #绘制结果图
48.  measurement_result = sim_result.get_counts(circuit)
49.  print(measurement_result)
50.  plot_histogram(measurement_result)
```

3.5 量子振幅估计

量子振幅估计(Quantum Amplitude Estimation,QAE)是结合振幅放大算子和量子相位估计的一种算法。如果能够将一个量子态按照某种规则分成两部分,则可以利用量子振幅估计算法估计出其中一部分的概率。例如,在 Grover 搜索算法中的式(3.2.13),量子态$|\psi\rangle$可以分为不包含搜索问题解的$|\alpha\rangle$和包含搜索问题解的$|\beta\rangle$。量子振幅估计算法能够将振幅相关信息存储到基态中,进而估计出得到$|\beta\rangle$的概率。

在量子机器学习算法中,给定一个酉变换$\boldsymbol{U}$,作用于初始量子态$|0\rangle|0\rangle^{\otimes n}$得到如下量子态:

$$\boldsymbol{U}|0\rangle|0\rangle^{\otimes n}=|\varphi\rangle=\sqrt{1-a}\,|0\rangle|\phi_0\rangle+\sqrt{a}\,|1\rangle|\phi_1\rangle \tag{3.5.1}$$

其中第 1 个量子比特是辅助量子比特,后面的 n 个量子比特用来存储量子态的两个部分。令$|\varphi_0\rangle=\sqrt{1-a}\,|0\rangle|\phi_0\rangle$,$|\varphi_1\rangle=\sqrt{a}\,|1\rangle|\phi_1\rangle$,则式(3.5.1)可以表示为

$$|\varphi\rangle=|\varphi_0\rangle+|\varphi_1\rangle \tag{3.5.2}$$

机器学习中,分类等信息通常存储在量子态$|\varphi_0\rangle$和$|\varphi_1\rangle$对应的振幅 a 中。存储在振幅中的信息需要经过多次执行算法并测量才能得到,这有时会给实施后续的操作造成一定的困难。而量子振幅估计算法先将振幅放大,再用量子相位估计算法将振幅 a 的相关信息转移到基态中,为后续操作提供了便利。因此,量子振幅估计算法在量子机器学习算法中起着非常重要的作用。

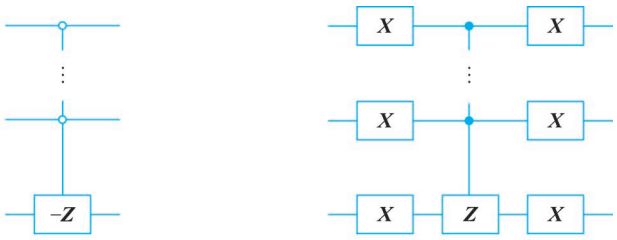
3.5.1 振幅放大

类似于 Grover 搜索算法的振幅放大算子,本节定义振幅放大算子如下:

$$\boldsymbol{Q}=\boldsymbol{U}\boldsymbol{S}_0\boldsymbol{U}^{-1}\boldsymbol{S}_f \tag{3.5.3}$$

式中:$\boldsymbol{S}_f=I-2|\varphi_1\rangle\langle\varphi_1|$能够改变式(3.5.2)中$|\varphi_1\rangle$的符号,而使$|\varphi_0\rangle$保持不变;$\boldsymbol{S}_0=\boldsymbol{I}-2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)}$只改变初始态$|0\rangle^{\otimes(n+1)}$的符号。

由于 $\boldsymbol{S}_f$ 只改变 $|\varphi\rangle=\sqrt{1-a}\,|0\rangle|\phi_0\rangle+\sqrt{a}\,|1\rangle|\phi_1\rangle$ 中 $|\varphi_1\rangle=\sqrt{a}\,|1\rangle|\phi_1\rangle$ 的符号，$|\varphi_0\rangle=\sqrt{1-a}\,|0\rangle|\phi_0\rangle$ 的符号保持不变，因此经过 $\boldsymbol{S}_f$ 作用之后 $|\varphi\rangle$ 变为 $|\varphi'\rangle=\sqrt{1-a}\,|0\rangle|\phi_0\rangle-\sqrt{a}\,|1\rangle|\phi_1\rangle$，而算子 $\boldsymbol{Z}$ 的作用为 $\boldsymbol{Z}|0\rangle=|0\rangle$，$\boldsymbol{Z}|1\rangle=-|1\rangle$，因此 $\boldsymbol{S}_f$ 的实现只需将 $\boldsymbol{Z}$ 门作用在量子态 $|\varphi\rangle$ 的第一个量子比特，即辅助量子比特上即可。$\boldsymbol{S}_0$ 只改变初始态 $|0\rangle^{\otimes(n+1)}$ 的符号，因此 $\boldsymbol{S}_0$ 的实现受到前 n 个量子比特都是 $|0\rangle$ 的控制，只对最后一个量子比特使用 $-\boldsymbol{Z}$。由于 $-\boldsymbol{Z}|0\rangle=-|0\rangle$ 且 $-\boldsymbol{Z}|1\rangle=|1\rangle$，也就是说只改变了初始态 $|0\rangle^{\otimes(n+1)}$ 的符号。$\boldsymbol{S}_0$ 的量子线路图如图 3.21(a)所示。在实现过程中，由于 $|0\rangle$ 控制以及 $-\boldsymbol{Z}$ 不能直接实现，因此将图 3.21(a)等价于图 3.21(b)的形式。



(a) $|0\rangle$ 控制的 $-\boldsymbol{Z}$　　(b) 等价线路

图 3.21　$\boldsymbol{S}_0$ 的量子线路图

定义归一化状态 $\dfrac{|\varphi_0\rangle}{\sqrt{1-a}}$ 和 $\dfrac{|\varphi_1\rangle}{\sqrt{a}}$，由于 $\langle\varphi_0|\varphi_1\rangle=\langle\varphi_1|\varphi_0\rangle=0$，且满足 $\langle\varphi_0|\varphi_0\rangle=1-a$ 和 $\langle\varphi_1|\varphi_1\rangle=a$，因此 $\dfrac{|\varphi_0\rangle}{\sqrt{1-a}}$ 和 $\dfrac{|\varphi_1\rangle}{\sqrt{a}}$ 为一组标准正交基。在这组基下，量子态 $|\varphi\rangle$ 可以表示为

$$|\varphi\rangle=\begin{pmatrix}\sqrt{1-a}\\ \sqrt{a}\end{pmatrix} \tag{3.5.4}$$

由于

$$\boldsymbol{Q}=\boldsymbol{U}\boldsymbol{S}_0\boldsymbol{U}^{-1}\boldsymbol{S}_f=(\boldsymbol{U}\boldsymbol{S}_0\boldsymbol{U}^{-1})(\boldsymbol{S}_f) \tag{3.5.5}$$

又因为

$$\begin{aligned}\boldsymbol{U}\boldsymbol{S}_0\boldsymbol{U}^{-1}&=\boldsymbol{U}(\boldsymbol{I}-2|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)})\boldsymbol{U}^{-1}\\&=\boldsymbol{U}\boldsymbol{I}\boldsymbol{U}^{-1}-2\boldsymbol{U}|0\rangle^{\otimes(n+1)}\langle 0|^{\otimes(n+1)}\boldsymbol{U}^{-1}\end{aligned}$$

由式(3.5.1)可知 $\boldsymbol{U}|0\rangle|0\rangle^{\otimes n}=|\varphi\rangle$，因此 $\boldsymbol{U}\boldsymbol{S}_0\boldsymbol{U}^{-1}=\boldsymbol{I}-2|\varphi\rangle\langle\varphi|$。根据式(3.5.4)，在基 $\dfrac{|\varphi_0\rangle}{\sqrt{1-a}}$ 和 $\dfrac{|\varphi_1\rangle}{\sqrt{a}}$ 下，令 $a=\sin^2\theta_a$，又因为此时 $\boldsymbol{S}_f=\boldsymbol{I}-\dfrac{2}{a}|\varphi_1\rangle\langle\varphi_1|$，则式(3.5.5)中的 $\boldsymbol{Q}$ 可以重写为

$$\begin{aligned}\boldsymbol{Q}&=(\boldsymbol{I}-2|\varphi\rangle\langle\varphi|)(\boldsymbol{I}-2|\varphi_1\rangle\langle\varphi_1|)\\&=\begin{pmatrix}2a-1 & -2\sqrt{a(1-a)}\\ -2\sqrt{a(1-a)} & 1-2a\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\end{aligned}$$

$$=\begin{pmatrix} 2a-1 & 2\sqrt{a(1-a)} \\ -2\sqrt{a(1-a)} & 2a-1 \end{pmatrix}$$

$$=\begin{pmatrix} -\cos 2\theta_a & \sin 2\theta_a \\ -\sin 2\theta_a & -\cos 2\theta_a \end{pmatrix} \tag{3.5.6}$$

$\boldsymbol{Q}$ 的特征值为 $\lambda_{\pm}=-\mathrm{e}^{\pm \mathrm{i}2\theta_a}$，对应的特征向量为

$$|\varphi_{\pm}\rangle=\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{1-a}}|\varphi_0\rangle \pm \frac{\mathrm{i}}{\sqrt{a}}|\varphi_1\rangle\right)$$

将量子态 $|\varphi\rangle$ 在 $\boldsymbol{Q}$ 的特征向量上展开，可得

$$\begin{aligned} |\varphi\rangle &= |\varphi_0\rangle + |\varphi_1\rangle \\ &= \frac{\sqrt{1-a}}{\sqrt{2}}(|\varphi_+\rangle + |\varphi_-\rangle) - \frac{\mathrm{i}\sqrt{a}}{\sqrt{2}}(|\varphi_+\rangle - |\varphi_-\rangle) \\ &= \frac{1}{\sqrt{2}}(\mathrm{e}^{-\mathrm{i}\theta_a}|\varphi_+\rangle + \mathrm{e}^{\mathrm{i}\theta_a}|\varphi_-\rangle) \end{aligned} \tag{3.5.7}$$

因此，对 $|\varphi\rangle$ 连续实施 j 次算子 $\boldsymbol{Q}$（记作 $\boldsymbol{Q}^j$）可得

$$\begin{aligned} &\boldsymbol{Q}^j|\varphi\rangle \\ &=\frac{1}{\sqrt{2}}(\mathrm{e}^{-(2j+1)\mathrm{i}\theta_a}|\varphi_+\rangle + \mathrm{e}^{(2j+1)\mathrm{i}\theta_a}|\varphi_-\rangle) \\ &=\frac{1}{2}\left(\mathrm{e}^{-(2j+1)\mathrm{i}\theta_a}\left(\frac{1}{\sqrt{1-a}}|\varphi_0\rangle + \frac{\mathrm{i}}{\sqrt{a}}|\varphi_1\rangle\right) + \mathrm{e}^{(2j+1)\mathrm{i}\theta_a}\left(\frac{1}{\sqrt{1-a}}|\varphi_0\rangle - \frac{\mathrm{i}}{\sqrt{a}}|\varphi_1\rangle\right)\right) \\ &=\frac{1}{2}\left(\frac{1}{\sqrt{1-a}}(\mathrm{e}^{-(2j+1)\mathrm{i}\theta_a} + \mathrm{e}^{(2j+1)\mathrm{i}\theta_a})|\varphi_0\rangle + \frac{\mathrm{i}}{\sqrt{a}}(\mathrm{e}^{-(2j+1)\mathrm{i}\theta_a} - \mathrm{e}^{(2j+1)\mathrm{i}\theta_a})|\varphi_1\rangle\right) \\ &=\frac{1}{\sqrt{1-a}}\cos((2j+1)\theta_a)|\varphi_0\rangle + \frac{1}{\sqrt{a}}\sin((2j+1)\theta_a)|\varphi_1\rangle \end{aligned} \tag{3.5.8}$$

由式(3.5.8)可以看出，经过 $\boldsymbol{Q}^j$ 作用之后，与振幅 a 相关的 θ_a 存储在振幅中，并且随着 j 的改变而改变，最终 $|\varphi_1\rangle$ 的振幅变为 $\frac{1}{\sqrt{a}}\sin((2j+1)\theta_a)$。

3.5.2 完整算法

振幅估计的目的是求得振幅 a，由于 $a=\sin^2\theta_a$，因此求振幅 a 就转换为求 θ_a 的值。而 θ_a 恰好出现在 $\boldsymbol{Q}$ 的特征值的指数上，即 $\lambda_{\pm}=\mathrm{e}^{\pm \mathrm{i}2\theta_a}$，这正是相位估计所能求的形式，因此在振幅放大之后可以使用相位估计求得 θ_a。

下面结合图 3.22 所示的量子振幅估计线路图，介绍量子振幅估计算法。

首先制备量子态 $|0\rangle^{\otimes m}|0\rangle^{\otimes n+1}$，其中 $|0\rangle^{\otimes m}$ 用于存储 θ_a，和相位估计一样，m 的大小和估计得到的 θ_a 的精度有关，$|0\rangle^{\otimes n+1}$ 用于存储 $|\varphi\rangle$。

第一步：使用算子 $\boldsymbol{U}$ 制备量子态 $|\varphi\rangle$。

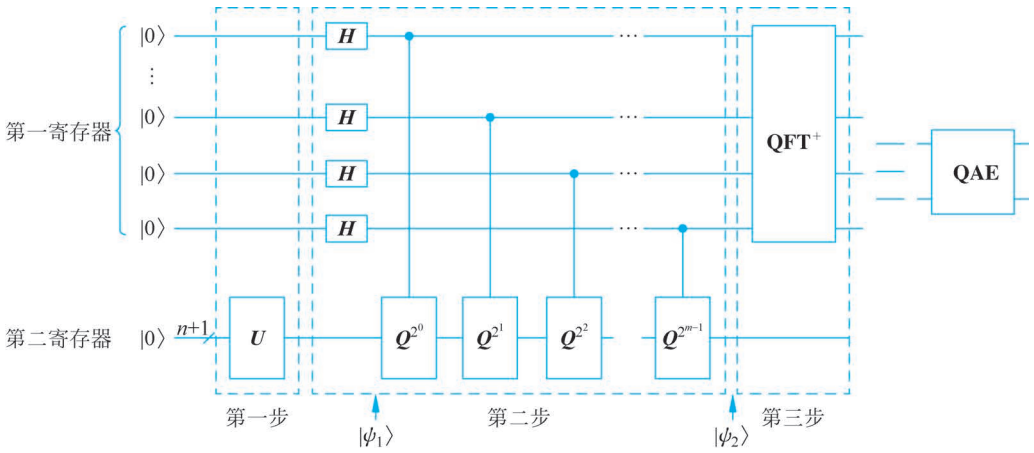


图 3.22　量子振幅估计的线路图

第二步：对第一寄存器执行 m 个 $\boldsymbol{H}$ 门制备叠加态，即

$$|\psi_1\rangle=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}|j\rangle|\varphi\rangle \tag{3.5.9}$$

以第一寄存器中的 m 个量子比特为控制比特，第二寄存器为目标比特，执行受控 $\boldsymbol{Q}^{2^t}$（$t=0,1,\cdots,m-1$）操作，得到量子态：

$$|\psi_2\rangle=\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{-\mathrm{i}2\theta_a j}|j\rangle\right)\mathrm{e}^{-\mathrm{i}\theta_a}|\varphi_+\rangle+\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{\mathrm{i}2\theta_a j}|j\rangle\right)\mathrm{e}^{\mathrm{i}\theta_a}|\varphi_-\rangle \tag{3.5.10}$$

令 $|S_{2^m}(\theta_a/\pi)\rangle=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{-\mathrm{i}2\theta_a j}|j\rangle$ 且 $\theta_a=\pi w$，则

$$\begin{aligned}&|S_{2^m}(1-\theta_a/\pi)\rangle\\&=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{-\mathrm{i}2\pi(1-w)j}|j\rangle=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{\mathrm{i}2\pi wj}\mathrm{e}^{-\mathrm{i}2\pi j}|j\rangle\\&=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{\mathrm{i}2\pi wj}(\cos 2\pi j-\mathrm{i}\sin 2\pi j)|j\rangle\\&=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{\mathrm{i}2\pi wj}|j\rangle=\frac{1}{\sqrt{2^m}}\sum_{j=0}^{2^m-1}\mathrm{e}^{\mathrm{i}2\theta_a j}|j\rangle\end{aligned} \tag{3.5.11}$$

则 $|\psi_2\rangle$ 可以重写为

$$|\psi_2\rangle=\frac{1}{\sqrt{2}}|S_{2^m}(\theta_a/\pi)\rangle\mathrm{e}^{-\mathrm{i}\theta_a}|\varphi_+\rangle+\frac{1}{\sqrt{2}}|S_{2^m}(1-\theta_a/\pi)\rangle\mathrm{e}^{\mathrm{i}\theta_a}|\varphi_-\rangle \tag{3.5.12}$$

第三步：对第一寄存器执行量子傅里叶逆变换得到 $|\psi_3\rangle$，将存储在振幅中的 θ_a 转移到基态中，即

$$|\psi_3\rangle=\frac{1}{\sqrt{2}}|2^m\theta_a/\pi\rangle e^{-i\theta_a}|\varphi_+\rangle+\frac{1}{\sqrt{2}}|2^m(1-\theta_a/\pi)\rangle e^{i\theta_a}|\varphi_-\rangle \quad (3.5.13)$$

对第一寄存器进行测量可能得到两种结果：$|y_1\rangle=|2^m\theta_a/\pi\rangle$和$|y_2\rangle=|2^m(1-\theta_a/\pi)\rangle$。即一种是 $\tilde{a}=\sin^2\theta_a=\sin^2(\pi y_1/2^m)$；另一种是 $\tilde{a}=\sin^2\theta_a=\sin^2(\pi-\pi y_2/2^m)$。由于 $\sin^2(\pi-\pi y_2/2^m)=\sin^2(\pi y_2/2^m)$，因此无论坍缩到哪一种情况都可以用 $\tilde{a}=\sin^2(\pi y/2^m)$ 计算 $\tilde{a}$。

事实上，量子振幅估计得到的 $\tilde{a}$ 与真实的 a 存在一定差距，定理 3.5.1 给出这个差的上限。这里不给出定理的具体证明过程，只给出结论，对证明过程感兴趣的读者可以参见文献[11]。

【定理 3.5.1】 对正整数 k，当 $k=1$ 时，振幅估计算法输出的 $\tilde{a}$ 与真实的 a 之间的差至少以$\frac{8}{\pi^2}$的概率满足

$$|\tilde{a}-a|\leqslant 2\pi k\frac{\sqrt{a(1-a)}}{2^m}+k^2\frac{\pi^2}{2^{2m}} \quad (3.5.14)$$

当 $k\geqslant 2$ 时，$\tilde{a}$ 与 a 之间的差至少以 $1-\frac{1}{2(k-1)}$的概率满足式(3.5.14)。如果 $a=0$，则有 $\tilde{a}=0$；如果 $a=1$，则有 $\tilde{a}=1$。

3.5.3 实现

本节使用量子振幅估计算法估计$|\varphi\rangle=\boldsymbol{U}|0\rangle|0\rangle=\frac{1}{\sqrt{2}}|0\rangle|0\rangle-\frac{1}{\sqrt{2}}|1\rangle|1\rangle$中$|1\rangle$的概率。量子线路图如图 3.23 所示，其中：$q_3$ 中存储的是$|\varphi\rangle$中的第一个量子比特，也就是辅助量子比特，用于区分两个部分；q_4 中存储的是$|\varphi\rangle$中的第二个量子比特，即式(3.5.1)中的两个部分$|\phi_0\rangle$和$|\phi_1\rangle$；$q_2q_1q_0$ 用于存储 θ_a，θ_a 用 3 个量子比特表示，意味着 $m=3$。

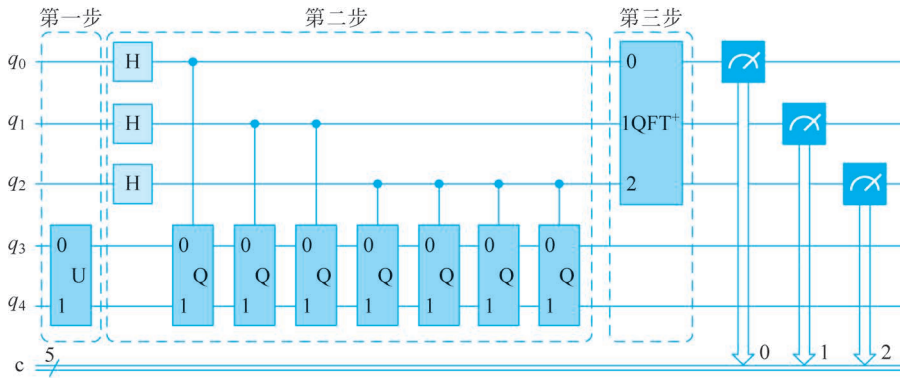


图 3.23 振幅估计的量子线路图

第一步使用 $\boldsymbol{U}$ 门制备量子态$|\varphi\rangle$，第二步和第三步是量子振幅估计算法。测量结果如图 3.24 所示，测量 $q_2q_1q_0$ 得到 010 和 110 的概率分别为 0.507 和 0.493。010 和 110 对应的十进制分别为 2 和 6，也就是说 y 的取值为 2 或 6。无论 y 是 2 还是 6，都有 $a=$

$\sin^2(\pi y/2^m)=\frac{1}{2}$。这与$|\varphi\rangle=\boldsymbol{U}|0\rangle|0\rangle=\frac{1}{\sqrt{2}}|0\rangle|0\rangle-\frac{1}{\sqrt{2}}|1\rangle|1\rangle$中$|1\rangle$的概率$\left(\frac{1}{\sqrt{2}}\right)^2=\frac{1}{2}$相吻合。

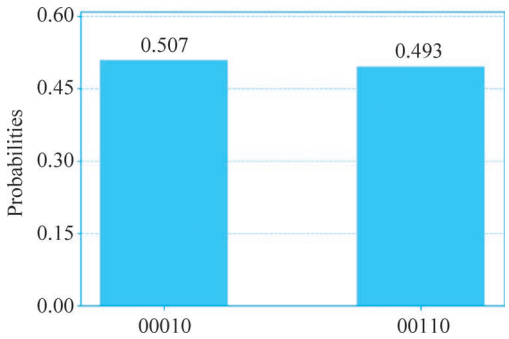


图 3.24　实验结果

量子振幅估计的代码如下：

```
%matplotlib inline
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute
from qiskit import BasicAer
from qiskit import IBMQ
from math import pi
from qiskit.tools.visualization import plot_histogram
from qiskit import Aer

circuit = QuantumCircuit(5,5)

#第一步：量子态制备,制备振幅估计的叠加态
def U():
    circuit = QuantumCircuit(2)
    circuit.h(0)
    circuit.cry(pi/2,0,1)
    circuit.x(0)
    circuit.cry(-pi/2,0,1)
    circuit.x(0)
    circuit.h(0)
    circuit = circuit.to_gate()
    circuit.name = "U"
    return circuit

circuit.append(U(),[i+3 for i in range(2)])

#第二步
circuit.barrier()
for i in range(3):
    circuit.h(i)

#对受控比特进行Q门操作
def Q():
```

```
        circuit = QuantumCircuit(2)
        circuit.z(0)
        circuit.append(U().inverse(),[i for i in range(2)])
        circuit.x(1)
        circuit.x(0)
        circuit.cz(0,1)
        circuit.x(0)
        circuit.x(1)
        circuit.append(U(),[i for i in range(2)])
        circuit = circuit.to_gate()
        circuit.name = "Q"
        c_U = circuit.control()
        return c_U

for i in range(3):
    for j in range(2 ** i):
        circuit.append(Q(),[i] + [m + 3 for m in range(2)])

#第三步：量子傅里叶逆变换
def qft_dagger(n):
    qc = QuantumCircuit(n)
    for qubit in range(n//2):
        qc.swap(qubit, n - qubit - 1)
    for j in range(n):
        for m in range(j):
            qc.cp( - pi/float(2 ** (j - m)), m, j)
        qc.h(j)
    qc.name = "QFT+"
    return qc

circuit.append(qft_dagger(3),[i for i in range(3)])

#测量
for i in range(3):
    circuit.measure(i,i)

#绘制线路图
circuit.draw(output = 'mpl',plot_barriers = False,fold = - 1)
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit, backend, shots = 8192)
sim_result = job_sim.result()
measurement_result = sim_result.get_counts(circuit)

#绘制结果图
print(measurement_result)
plot_histogram(measurement_result)
```

3.6 交换测试

在经典计算中，两个向量的内积在很多方面都有应用。在量子计算中，一个向量可以表示成一个量子态。交换测试(Swap Test，ST)可以计算两个量子态的内积，能够用于

衡量两个量子态所对应向量之间的相似程度。很多量子机器学习算法用到了交换测试。下面首先给出二维量子态的交换测试算法，然后扩展到 N 维。

3.6.1 算法

二维量子态的交换测试主要由三个酉算子组成，如图 3.24 中第二步所示。

令二维量子态$|a\rangle$和$|b\rangle$分别为

$$|a\rangle=a_0|0\rangle+a_1|1\rangle,\quad |b\rangle=b_0|0\rangle+b_1|1\rangle \tag{3.6.1}$$

式中

$$|a_0|^2+|a_1|^2=1,\quad |b_0|^2+|b_1|^2=1$$

下面结合图 3.25 给出具体的交换测试算法。

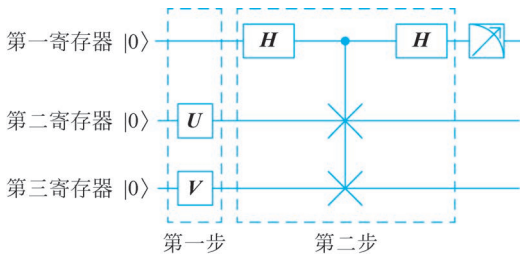


图 3.25 二维交换测试的量子线路图

制备初始量子态$|\psi_0\rangle=|0\rangle|0\rangle|0\rangle$，其中第一个$|0\rangle$是辅助量子比特，后两个$|0\rangle$分别用来存储量子态$|a\rangle$和$|b\rangle$。

第一步：使用 $\boldsymbol{U}$ 门和 $\boldsymbol{V}$ 门作用于第二和第三寄存器制备量子态$|a\rangle$和$|b\rangle$，则初始态$|\psi_0\rangle$演化为

$$|\psi_1\rangle=|0\rangle|a\rangle|b\rangle \tag{3.6.2}$$

第二步：交换测试。

首先使用 $\boldsymbol{H}$ 门作用于第一寄存器的辅助量子比特$|0\rangle$，则量子态$|\psi_1\rangle$演化为

$$|\psi_2\rangle=\frac{1}{\sqrt{2}}(|0\rangle|a\rangle|b\rangle+|1\rangle|a\rangle|b\rangle) \tag{3.6.3}$$

然后应用受控交换门将$|\psi_2\rangle$演化为

$$|\psi_3\rangle=\frac{1}{\sqrt{2}}(|0\rangle|a\rangle|b\rangle+|1\rangle|b\rangle|a\rangle) \tag{3.6.4}$$

再使用 $\boldsymbol{H}$ 门作用于辅助量子比特将$|\psi_3\rangle$演化为

$$\begin{aligned}|\psi_4\rangle&=\frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|a\rangle|b\rangle+\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)|b\rangle|a\rangle\right]\\&=\frac{1}{2}|0\rangle(|a\rangle|b\rangle+|b\rangle|a\rangle)+\frac{1}{2}|1\rangle(|a\rangle|b\rangle-|b\rangle|a\rangle)\end{aligned} \tag{3.6.5}$$

最后对辅助量子比特进行测量，得到$|0\rangle$的概率为

$$
\begin{aligned}
P(0) &= \left(\frac{1}{2}\langle a|\langle b|+\langle b|\langle a|\right)\left(\frac{1}{2}|a\rangle|b\rangle+|b\rangle|a\rangle\right) \\
&= \frac{1}{4}(\langle a|\langle b||a\rangle|b\rangle+\langle a|\langle b||b\rangle|a\rangle+\langle b|\langle a||a\rangle|b\rangle+\langle b|\langle a||b\rangle|a\rangle) \\
&= \frac{1}{4}(\langle b||a\rangle\langle a||b\rangle+\langle b||b\rangle\langle a||a\rangle+\langle a||a\rangle\langle b||b\rangle+\langle a||b\rangle\langle b||a\rangle) \\
&= \frac{1}{4}(\langle b|a\rangle\langle a|b\rangle+1+1+\langle a|b\rangle\langle b|a\rangle) \\
&= \frac{1}{2}+\frac{1}{2}|\langle a|b\rangle|^2 \qquad (3.6.6)
\end{aligned}
$$

同理，得到$|1\rangle$的概率为

$$P(1)=\frac{1}{2}-\frac{1}{2}|\langle a|b\rangle|^2$$

因此，量子态$|a\rangle$和$|b\rangle$的内积可以通过下式计算：

$$|\langle a|b\rangle|=\sqrt{2P(0)-1}=\sqrt{1-2P(1)} \qquad (3.6.7)$$

可以看到，交换测试实际得到的是$|a\rangle$和$|b\rangle$内积的模$|\langle a|b\rangle|$，并非内积$\langle a|b\rangle$。对于复数来讲，模可以提供有关该复数大小的信息，因此$|a\rangle$和$|b\rangle$内积的模足以表示$|a\rangle$和$|b\rangle$的相似度。

由于$P(0)$和$P(1)$都是概率，要想得到概率必须多次运行交换测试算法。为了较为准确地得到结果，需要运行$O\left(\frac{1}{\varepsilon}\right)$次交换测试，才能以误差$\varepsilon$得到概率$P(0)$和$P(1)$。由式(3.6.5)可以看出，每运行一次交换测试算法，有$|0\rangle$和$|1\rangle$两种情况：若测量结果为$|0\rangle$，则第二和第三寄存器的态坍缩为$|a\rangle|b\rangle+|b\rangle|a\rangle$；若测量结果为$|1\rangle$，则第二和第三寄存器的态坍缩为$|a\rangle|b\rangle-|b\rangle|a\rangle$。无论哪种情况，第二和第三寄存器的状态均不是初始的量子态$|a\rangle|b\rangle$。也就是说，输出态不能再一次用于新的交换测试的输入，因此在量子计算机上运行交换测试算法时，需要不断地重复制备量子态$|a\rangle$和$|b\rangle$，并重复运行算法，这是交换测试的一个不足。

若将$|a\rangle$和$|b\rangle$都扩展为N维量子态，即由$n=\log N$个量子比特表示，则计算$|a\rangle$和$|b\rangle$的内积模的量子线路如图3.26所示。其算法过程和二维是一样的，这里不再赘述。

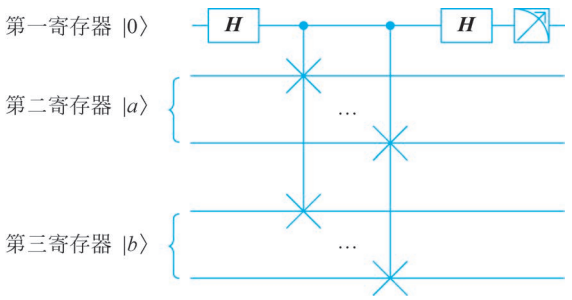


图 3.26　N 维交换测试的量子线路图

3.6.2 实现

本实验的目的在于计算两个量子态

$$|a\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}}=\left(\frac{1}{\sqrt{2}}\quad\frac{1}{\sqrt{2}}\right)^{\mathrm{T}},\qquad |b\rangle=|1\rangle=(0\quad 1)^{\mathrm{T}}$$

的内积模的平方

$$|\langle a\,|\,b\rangle|^2=\left|\frac{1}{\sqrt{2}}\times 0+\frac{1}{\sqrt{2}}\times 1\right|^2=0.5$$

以反映它们的相似度。

实现的量子线路图如图 3.27 所示。由于输入 q_1 和 q_2 初始态都是$|0\rangle$,因此第一步使用 ***H*** 门和 ***X*** 门分别制备量子态$|a\rangle$和$|b\rangle$。第二步为交换测试。最后对 q_0 进行测量,测量结果如图 3.28 所示,测量为$|0\rangle$的概率 $P(0)=0.746=\frac{1}{2}+\frac{1}{2}|\langle a\,|\,b\rangle|^2$,测量为$|1\rangle$的概率 $P(1)=0.254=\frac{1}{2}-\frac{1}{2}|\langle a\,|\,b\rangle|^2$,因此可得$|\langle a\,|\,b\rangle|^2=2P(0)-1=1-2P(1)=0.492$。此实验结果与真实内积 0.5 相吻合。

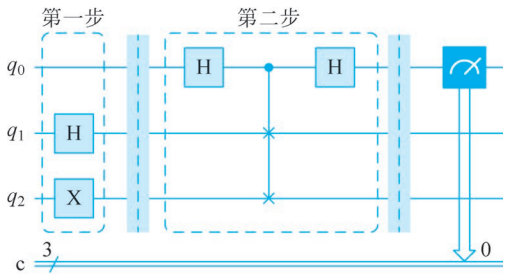


图 3.27 交换测试的量子线路图

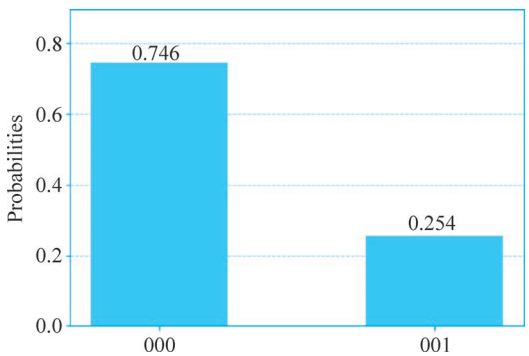


图 3.28 交换测试的实验结果

交换测试的代码如下:

```
1.    % matplotlib inline
2.    from qiskit import QuantumCircuit
```

```
from qiskit import execute
from qiskit import IBMQ
from qiskit.tools.visualization import plot_histogram

circuit = QuantumCircuit(3, 3)

#第一步：量子态制备
circuit.h(1)
circuit.x(2)

#第二步：swap-test
circuit.barrier(0,1)
circuit.h(0)
circuit.cswap(0,1,2)
circuit.h(0)
circuit.barrier(0,1)

#测量
circuit.measure([0],[0])

#绘制线路图
circuit.draw(output='mpl')
IBMQ.enable_account('token')
my_provider = IBMQ.get_provider()
backend   = my_provider.get_backend('ibmq_qasm_simulator')
job_sim = execute(circuit, backend, shots=1024)
sim_result = job_sim.result()

#绘制结果图
measurement_result = sim_result.get_counts(circuit)
plot_histogram(measurement_result)
```

3.7 哈达玛测试

交换测试给出了计算两个量子态内积模的方法，那么如何计算两个量子态内积呢？哈达玛测试（Hadamard Test，HT）能解决这个问题。由于量子态的振幅都是复数，因此内积计算分为两部分，一部分是计算内积的实部，另一部分是计算内积的虚部。

3.7.1 哈达玛测试计算内积的实部

假设量子态

$$|a\rangle = a_0|0\rangle + a_1|1\rangle, \quad |b\rangle = b_0|0\rangle + b_1|1\rangle \tag{3.7.1}$$

式中：a_0、a_1、b_0 和 b_1 都是复数，且 $|a_0|^2+|a_1|^2=1$，$|b_0|^2+|b_1|^2=1$。

则 $\langle a|b\rangle$ 的实部等于 $\langle b|a\rangle$ 的实部

$$\mathrm{Re}\langle a|b\rangle = \mathrm{Re}\langle b|a\rangle = \frac{\langle a|b\rangle + \langle b|a\rangle}{2} \tag{3.7.2}$$

式中：Re 表示实部。

图 3.29 给出内积实部的计算方法。下面结合该图给出具体步骤。令 $\boldsymbol{U}|0\rangle=|a\rangle$，$\boldsymbol{V}|0\rangle=|b\rangle$，两个量子寄存器初态均为$|0\rangle$。

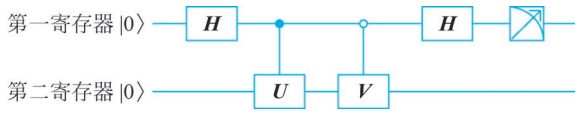


图 3.29 哈达玛测试中计算实部的量子线路图

第一步：使用 $\boldsymbol{H}$ 门作用于第一寄存器，则初始态$|0\rangle|0\rangle$演化为

$$|\psi_1\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle=\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle) \tag{3.7.3}$$

第二步：受第一寄存器的控制，制备$|a\rangle$和$|b\rangle$。当第一寄存器为$|0\rangle$时，使用 $\boldsymbol{V}$ 门作用于第二寄存器演化出$|b\rangle$，当第一寄存器的量子比特为$|1\rangle$时，使用 $\boldsymbol{U}$ 门作用于第二寄存器演化出$|a\rangle$，即

$$|\psi_2\rangle=\frac{1}{\sqrt{2}}(|0\rangle|b\rangle+|1\rangle|a\rangle) \tag{3.7.4}$$

第三步：使用 $\boldsymbol{H}$ 门作用于第一寄存器可得

$$\begin{aligned}|\psi_3\rangle&=\frac{1}{2}[(|0\rangle+|1\rangle)|b\rangle+(|0\rangle-|1\rangle)|a\rangle]\\&=\frac{1}{2}[|0\rangle(|b\rangle+|a\rangle)+|1\rangle(|b\rangle-|a\rangle)]\end{aligned} \tag{3.7.5}$$

对第一寄存器进行测量，得到$|0\rangle$的概率为

$$\begin{aligned}P(0)&=\frac{1}{2}(\langle b|+\langle a|)\frac{1}{2}(|b\rangle+|a\rangle)\\&=\frac{1}{4}(\langle b||b\rangle+\langle b||a\rangle+\langle a||b\rangle+\langle a||a\rangle)\\&=\frac{1}{4}(2+\langle a|b\rangle+\langle b|a\rangle)\\&=\frac{1}{4}(2+2\mathrm{Re}\langle a|b\rangle)\\&=\frac{1}{2}+\frac{\mathrm{Re}\langle a|b\rangle}{2}\end{aligned} \tag{3.7.6}$$

由此可以得到 $\mathrm{Re}\langle a|b\rangle=2P(0)-1$。

同理，得到$|1\rangle$的概率为

$$P(1)=\frac{1}{2}-\frac{\mathrm{Re}\langle a|b\rangle}{2}$$

因此也可以使用得到$|1\rangle$的概率求解实部，即 $\mathrm{Re}\langle a|b\rangle=1-2P(1)$。

3.7.2 哈达玛测试计算内积的虚部

在式(3.7.1)的基础上，令 $\mathrm{i}|b\rangle=\mathrm{i}b_0|0\rangle+\mathrm{i}b_1|1\rangle$，则

$$\mathrm{Im}\langle a \mid b\rangle = -\mathrm{Re}\langle a \mid \mathrm{i} \mid b\rangle \tag{3.7.7}$$

式中：Im 表示虚部。

由于算子 $\boldsymbol{S}=\begin{pmatrix}1 & 0\\ 0 & \mathrm{i}\end{pmatrix}$，由式(3.7.7)可以得出，只需在图 3.28 第一个 $\boldsymbol{H}$ 门后增加一个 $\boldsymbol{S}$ 门就可以得到 $\mathrm{Im}\langle a \mid b\rangle$。下面给出具体的计算方法。

第一步：使用 $\boldsymbol{H}$ 门作用于第一寄存器，则 $|0\rangle|0\rangle$ 演化为

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|0\rangle) \tag{3.7.8}$$

第二步：将 $\boldsymbol{S}$ 门作用于第一寄存器得到

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}\,|1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + \mathrm{i}\,|1\rangle\,|0\rangle) \tag{3.7.9}$$

第三步：受第一寄存器的控制，制备 $|a\rangle$ 和 $|b\rangle$。当第一寄存器为 $|0\rangle$ 时，使用 $\boldsymbol{V}$ 门作用于第二寄存器演化出 $|b\rangle$，当第一寄存器的量子比特为 $|1\rangle$ 时，使用 $\boldsymbol{U}$ 门作用于第二寄存器演化出 $|a\rangle$，即

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|b\rangle + \mathrm{i}\,|1\rangle\,|a\rangle) \tag{3.7.10}$$

第四步：使用 $\boldsymbol{H}$ 门作用于第一寄存器可得

$$\begin{aligned}|\psi_4\rangle &= \frac{1}{2}((|0\rangle + |1\rangle)\,|b\rangle + \mathrm{i}(|0\rangle - |1\rangle)\,|a\rangle)\\ &= \frac{1}{2}(|0\rangle(|b\rangle + \mathrm{i}\,|a\rangle)\,|b\rangle + |1\rangle(|b\rangle - \mathrm{i}\,|a\rangle))\end{aligned} \tag{3.7.11}$$

对第一寄存器量子比特测量，得到 $|0\rangle$ 的概率为

$$\begin{aligned}P(0) &= \frac{1}{2}(\langle b| - \langle a \mid \mathrm{i})\,\frac{1}{2}(|b\rangle + \mathrm{i}\,|a\rangle)\\ &= \frac{1}{4}(\langle b\,||\,b\rangle + \langle b \mid \mathrm{i} \mid a\rangle - \langle a \mid \mathrm{i} \mid b\rangle - \langle a \mid \mathrm{ii} \mid a\rangle)\\ &= \frac{1}{4}(2 + \langle b \mid \mathrm{i} \mid a\rangle - \langle a \mid \mathrm{i} \mid b\rangle)\\ &= \frac{1}{4}(2 + \mathrm{i}(\langle b \mid a\rangle - \langle a \mid b\rangle))\end{aligned} \tag{3.7.12}$$

由于

$$\mathrm{iIm}\langle a \mid b\rangle = -\mathrm{iIm}\langle b \mid a\rangle = \frac{\langle a \mid b\rangle - \langle b \mid a\rangle}{2} \tag{3.7.13}$$

则

$$P(0) = \frac{1}{4}(2 + 2\mathrm{Im}\langle a \mid b\rangle) = \frac{1}{2} + \frac{\mathrm{Im}\langle a \mid b\rangle}{2} \tag{3.7.14}$$

由此可以得到 $\mathrm{Im}\langle a \mid b\rangle = 2P(0) - 1$。

同理，得到$|1\rangle$的概率为

$$P(1)=\frac{1}{2}-\frac{\mathrm{Im}\langle a\mid b\rangle}{2}$$

因此也可以使用得到$|1\rangle$的概率求解虚部，即$\mathrm{Im}\langle a|b\rangle=1-2P(1)$。

当$|a\rangle$和$|b\rangle$的振幅都是实数时，内积的虚部为0，内积的实部也就是内积。在后续的量子机器学习算法中，要计算内积的$|a\rangle$和$|b\rangle$的振幅皆为实数，因此可以用哈达玛测试计算内积。

与交换测试一样，为了较为准确地得到内积，需要$O\left(\frac{1}{\varepsilon}\right)$次运行，才能以误差$\varepsilon$得到概率$P(0)$或$P(1)$。

3.7.3 实现

该实验求两个量子态$|a\rangle=0.999|0\rangle+0.045|1\rangle$和$|b\rangle=0.339|0\rangle+0.941|1\rangle$的内积$0.999\times0.339+0.045\times0.941=0.381$。

图3.30是在qiskit上实现的量子线路图。第一步对q_0进行$\boldsymbol{H}$门变换；第二步分别用1控制和0控制的旋转门制备$|a\rangle$和$|b\rangle$；第三步再次对q_0进行$\boldsymbol{H}$门变换；最后对q_0进行测量。由图3.31可以看出，得到$|1\rangle$的概率为0.312，因此通过实验得到$|a\rangle$和$|b\rangle$的内积的实部为$1-2\times0.312=0.376$，由于$|a\rangle$和$|b\rangle$的振幅都是实数，因此$|a\rangle$和$|b\rangle$内积为0.376，与真实内积相吻合。

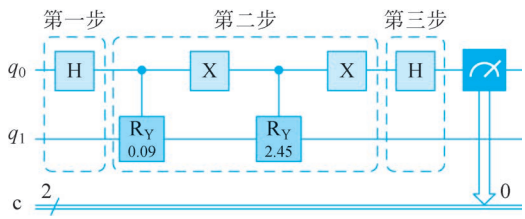


图3.30 哈达玛测试的量子线路图

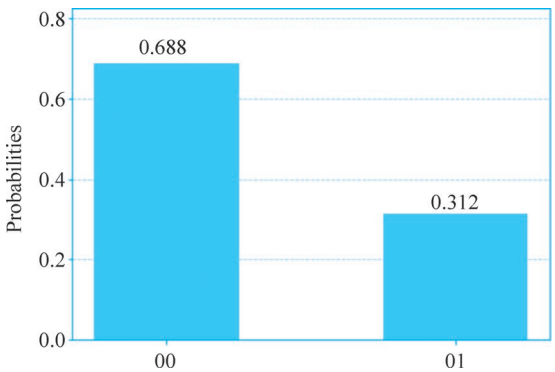


图3.31 哈达玛测试的实验结果

哈达玛测试的代码如下：

```
%matplotlib inline
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute
from qiskit import Aer
from qiskit import IBMQ
from math import pi
from qiskit.tools.visualization import plot_histogram

circuit = QuantumCircuit(2,2)

#第一步
circuit.h(0)

#第二步
circuit.cry(0.090,0,1)
circuit.x(0)
circuit.cry(2.452,0,1)
circuit.x(0)

#第三步
circuit.h(0)

#测量
circuit.measure(0,0)

#绘制线路图
circuit.draw(output = 'mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit,backend,shots = 20000)
sim_result = job_sim.result()

#绘制结果图
measurement_result = sim_result.get_counts(circuit)
plot_histogram(measurement_result)
```

3.8 HHL 算法

HHL 算法利用量子特性来求解线性方程组。线性方程组的求解问题可以归纳为给定一个 $N\times N$ 的可逆方阵 $\boldsymbol{A}$ 和一个 $N\times 1$ 的向量 $\boldsymbol{b}$，如何找到一个 $N\times 1$ 的向量 $\boldsymbol{x}$，使其满足如下方程：

$$\boldsymbol{Ax}=\boldsymbol{b} \tag{3.8.1}$$

该问题的一种解法是通过求解矩阵的逆 $\boldsymbol{A}^{-1}$ 获得 $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$。在经典计算中，求矩阵的逆需要付出 $O(N^3)$ 的代价。而 HHL 算法利用量子特性来求解线性方程组，能够降低复杂度。在 HHL 算法中非常重要的一环是哈密顿量模拟。下面首先给出哈密顿量模拟方法，然后介绍具体的 HHL 算法。

3.8.1 哈密顿量模拟

在量子力学中哈密顿量是描述系统总能量的算符，在数学上哈密顿量是厄米矩阵 $\boldsymbol{H}$。本节不深究哈密顿量的物理意义，仅将其理解为厄米矩阵。

哈密顿量模拟问题是指用量子门实现哈密顿量 $\boldsymbol{H}$ 的指数形式 $\mathrm{e}^{\mathrm{i}\boldsymbol{H}t}$。也就是说，对于给定的时间 t 和误差 $\varepsilon>0$，使用 $\log N$ 的多项式个量子门组成的 N 维酉算子 $\boldsymbol{U}$ 能够实现 $\mathrm{e}^{\mathrm{i}\boldsymbol{H}t}$，即 $\|\boldsymbol{U}-\mathrm{e}^{\mathrm{i}\boldsymbol{H}t}\|\leqslant\varepsilon$。使用 $\log N$ 的多项式个量子门，意味着算法复杂度为 $O(\mathrm{poly}(\log N))$。

事实上并不是所有的哈密顿量都可以被有效模拟，但是大多数哈密顿量可以写成许多更简单的、便于量子计算机实现的矩阵的和的形式，为哈密顿量模拟提供了方便。假设一个哈密顿量 $\boldsymbol{H}$ 分解为

$$\boldsymbol{H}=\sum_{k=1}^{L}\boldsymbol{H}_k \tag{3.8.2}$$

式中：$\boldsymbol{H}_k$ 是更简单的矩阵，通常由较为简单的门相乘来构成；L 为 n 的一个多项式；n 为模拟 $\boldsymbol{H}$ 时用到的量子比特数。

下面针对哈密顿量模拟给出两个定理。

【定理 3.8.1】 对于 $\boldsymbol{H}=\sum_{k=1}^{L}\boldsymbol{H}_k$，如果对所有的 j 和 k 都满足 $\boldsymbol{H}_j$ 和 $\boldsymbol{H}_k$ 对易，即 $[\boldsymbol{H}_j,\boldsymbol{H}_k]=\boldsymbol{H}_j\boldsymbol{H}_k-\boldsymbol{H}_k\boldsymbol{H}_j=0$，则对所有的正实数 t，有

$$\mathrm{e}^{\mathrm{i}\boldsymbol{H}t}=\mathrm{e}^{\mathrm{i}\boldsymbol{H}_1t}\mathrm{e}^{\mathrm{i}\boldsymbol{H}_2t}\cdots\mathrm{e}^{\mathrm{i}\boldsymbol{H}_Lt} \tag{3.8.3}$$

【例 3.8.1】 对于哈密顿量 $\boldsymbol{H}=\frac{1}{2}\begin{pmatrix}3&1\\1&3\end{pmatrix}$ 来说，令 $t=\frac{\pi}{2}$，则需要模拟的量为 $\mathrm{e}^{\mathrm{i}\boldsymbol{H}\frac{\pi}{2}}$。

由于

$$\boldsymbol{H}=\frac{1}{2}\begin{pmatrix}3&1\\1&3\end{pmatrix}=\frac{3\boldsymbol{I}}{2}+\frac{\boldsymbol{X}}{2}$$

因此，$\mathrm{e}^{\mathrm{i}\boldsymbol{H}\frac{\pi}{2}}$ 可以分解为

$$\mathrm{e}^{\mathrm{i}\boldsymbol{H}\frac{\pi}{2}}=\mathrm{e}^{\mathrm{i}\left(\frac{3\boldsymbol{I}}{2}+\frac{\boldsymbol{X}}{2}\right)\frac{\pi}{2}}=\mathrm{e}^{\mathrm{i}\frac{3\boldsymbol{I}}{2}\frac{\pi}{2}}\mathrm{e}^{\mathrm{i}\frac{\boldsymbol{X}}{2}\frac{\pi}{2}}=\mathrm{e}^{\mathrm{i}\boldsymbol{I}\frac{3\pi}{4}}\mathrm{e}^{\mathrm{i}\boldsymbol{X}\frac{\pi}{4}} \tag{3.8.4}$$

由 $\mathrm{e}^{\mathrm{i}\boldsymbol{A}x}=\cos(x)\boldsymbol{I}+\mathrm{i}\sin(x)\boldsymbol{A}$ 可知

$$\mathrm{e}^{\mathrm{i}\boldsymbol{I}\frac{3\pi}{4}}=\cos\left(\frac{3\pi}{4}\right)\boldsymbol{I}+\mathrm{i}\sin\left(\frac{3\pi}{4}\right)\boldsymbol{I}=-\frac{\sqrt{2}}{2}\boldsymbol{I}+\mathrm{i}\frac{\sqrt{2}}{2}\boldsymbol{I}=\frac{\sqrt{2}}{2}(-1+\mathrm{i})\boldsymbol{I}$$

是单位矩阵的常数倍，可以用 $\boldsymbol{I}$ 门实现。而

$$\mathrm{e}^{\mathrm{i}\boldsymbol{X}\frac{\pi}{4}}=\cos\left(\frac{\pi}{4}\right)\boldsymbol{I}+\mathrm{i}\sin\left(\frac{\pi}{4}\right)\boldsymbol{X}$$

$$=\begin{pmatrix}\cos\frac{\pi}{4}&\mathrm{i}\sin\frac{\pi}{4}\\\mathrm{i}\sin\frac{\pi}{4}&\cos\frac{\pi}{4}\end{pmatrix}=\begin{pmatrix}\cos\left(-\frac{\pi}{4}\right)&-\mathrm{i}\sin\left(-\frac{\pi}{4}\right)\\-\mathrm{i}\sin\left(-\frac{\pi}{4}\right)&\cos\left(-\frac{\pi}{4}\right)\end{pmatrix}$$

$$=\boldsymbol{R}_x\left(-\frac{\pi}{2}\right)$$

等价于 $\boldsymbol{R}_x\left(-\frac{\pi}{2}\right)$门。两部分都可以用简单的量子门来实现，把两部分合起来就可以有效模拟 $\mathrm{e}^{\mathrm{i}\boldsymbol{H}\frac{\pi}{2}}$。

【定理 3.8.2】 令 $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 是两个哈密顿量，则对任意的正实数 t，有

$$\lim_{n\to\infty}(\mathrm{e}^{\mathrm{i}\boldsymbol{B}\frac{t}{n}}\mathrm{e}^{\mathrm{i}\boldsymbol{C}\frac{t}{n}})^n=\mathrm{e}^{\mathrm{i}(\boldsymbol{B}+\boldsymbol{C})t} \tag{3.8.5}$$

该定理中并不要求 $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 是对易的。

这里不对上述两个定理进行证明，感兴趣的读者可以参见文献[1]。但是，在实验中不能做到 $n\to\infty$，通常使用高阶近似方法来近似。例如，下面的 $n=1$ 和 $n=2$ 两种情况：

$$\mathrm{e}^{\mathrm{i}(\boldsymbol{B}+\boldsymbol{C})t}=\mathrm{e}^{\mathrm{i}\boldsymbol{B}t}\mathrm{e}^{\mathrm{i}\boldsymbol{C}t}+O(t^2) \tag{3.8.6}$$

$$\mathrm{e}^{\mathrm{i}(\boldsymbol{B}+\boldsymbol{C})t}=(\mathrm{e}^{\mathrm{i}\boldsymbol{B}\frac{t}{2}}\mathrm{e}^{\mathrm{i}\boldsymbol{C}\frac{t}{2}})^2+O(t^3) \tag{3.8.7}$$

HHL 算法中将 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的 $\boldsymbol{A}$ 看作一个哈密顿量，通过对其模拟来求解线性方程组。但是，哈密顿量是厄米矩阵，如果 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的 $\boldsymbol{A}$ 不是厄米矩阵，那么可以将 $\boldsymbol{A}$ 构造成一个厄米矩阵。构造方法如下：

$$\widetilde{\boldsymbol{A}}=\begin{pmatrix}\boldsymbol{0} & \boldsymbol{A}\\ \boldsymbol{A}^+ & \boldsymbol{0}\end{pmatrix} \tag{3.8.8}$$

这样得到的 $\widetilde{\boldsymbol{A}}$ 是厄米矩阵。则 $\boldsymbol{Ax}=\boldsymbol{b}$ 转换为

$$\widetilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}=\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{pmatrix} \tag{3.8.9}$$

求解上式可得

$$\tilde{\boldsymbol{x}}=\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{x}\end{pmatrix} \tag{3.8.10}$$

式中：$\boldsymbol{0}$ 为 N 个 0 组成的列向量。因此，下面皆假设 $\boldsymbol{A}$ 是厄米矩阵。

3.8.2 算法基本思想

假设 $\boldsymbol{A}$ 是厄米矩阵，则它的谱分解可以写成

$$\boldsymbol{A}=\sum_{i=0}^{N-1}\lambda_i\,|u_i\rangle\langle u_i| \tag{3.8.11}$$

式中：$|u_i\rangle(i=0,1,\cdots,N-1)$为 $\boldsymbol{A}$ 的特征向量；λ_i 为对应的特征值。

在此基础上可以给出 HHL 算法的基本思想。

因为厄米矩阵 $\boldsymbol{A}$ 的全部特征向量组成了希尔伯特空间中的一组标准正交基，因此向量 $\boldsymbol{b}$ 可以表示为

$$|b\rangle=\sum_{i=0}^{N-1}\beta_i\,|u_i\rangle \tag{3.8.12}$$

则

$$\boldsymbol{A}^{-1}\boldsymbol{b}=\left(\sum_{i=0}^{N-1}\lambda_i^{-1}\mid u_i\rangle\langle u_i\mid\right)\left(\sum_{i=0}^{N-1}\beta_i\mid u_i\rangle\right)=\sum_{i=0}^{N-1}\frac{\beta_i}{\lambda_i}\mid u_i\rangle \tag{3.8.13}$$

由式(3.8.13)可以看出，求解 $\boldsymbol{A}^{-1}\boldsymbol{b}$ 转换成了求解 $\sum_{i=0}^{N-1}\frac{\beta_i}{\lambda_i}\mid u_i\rangle$，HHL 算法就是利用量子特性得到 $\sum_{i=0}^{N-1}\frac{\beta_i}{\lambda_i}\mid u_i\rangle$。

3.8.3 算法步骤

HHL 算法主要有相位估计、受控旋转以及逆相位估计三个步骤，如图 3.32 所示。HHL 算法共需要三个寄存器 $|0\rangle|0\rangle^{\otimes l}|0\rangle^{\otimes n}$（其中 $n=\log N$）：第一寄存器是一个辅助量子比特；第二寄存器的作用是暂存特征值，l 取决于相位估计的精度和成功率；第三寄存器用于存储向量 $\boldsymbol{b}$。下面介绍 HHL 算法的具体步骤。

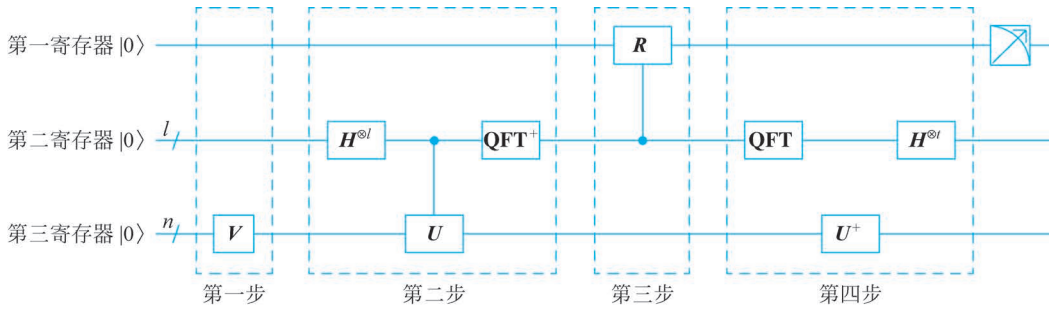


图 3.32 HHL 算法的量子线路

第一步：使用酉变换 $\boldsymbol{V}$ 作用于第三寄存器制备 $\mid b\rangle=\sum_{i=0}^{N-1}\beta_i\mid u_i\rangle$，则初始态 $|0\rangle|0\rangle^{\otimes l}|0\rangle^{\otimes n}$ 演化为量子态 $|\psi_1\rangle$，即

$$\mid\psi_1\rangle=\mid 0\rangle\mid 0\rangle^{\otimes l}\sum_{i=0}^{N-1}\beta_i\mid u_i\rangle \tag{3.8.14}$$

第二步：对第二寄存器和第三寄存器使用量子相位估计可得

$$\mid\psi_2\rangle=\mid 0\rangle\sum_{i=0}^{N-1}\beta_i\mid\tilde{\lambda}_i\rangle\mid u_i\rangle \tag{3.8.15}$$

由相位估计算法可知，大多数情况下基态中存储的并不是 λ_i，而是 λ_i 的近似值，记为 $\tilde{\lambda}_i$。

由 3.4.1 节可知，量子相位估计算法中 $\boldsymbol{U}$ 的设置至关重要，不同的 $\boldsymbol{U}$ 会得到不同的运行结果。因为 $\boldsymbol{A}$ 的特征值为 λ_i，所以 $\mathrm{e}^{\mathrm{i}\boldsymbol{A}t}$ 的特征值为 $\mathrm{e}^{\mathrm{i}\lambda_i t}$，与相位估计算法中的形式相似，能够通过相位估计将相位 λ_i 存储到基态中。因此 HHL 算法中令

$$\boldsymbol{U}=\mathrm{e}^{\mathrm{i}\boldsymbol{A}t}=\sum_{i=0}^{N-1}\mathrm{e}^{\mathrm{i}\lambda_i t}\mid u_i\rangle\langle u_i\mid \tag{3.8.16}$$

而且由于 $\boldsymbol{A}$ 是厄米矩阵，所以 $\mathrm{e}^{\mathrm{i}\boldsymbol{A}t}$ 是一个酉矩阵，满足量子门的要求。

其实 $\mathrm{e}^{\mathrm{i}\lambda_i t}$ 与相位估计算法中 $\boldsymbol{U}$ 的特征值的形式 $\mathrm{e}^{2\pi\mathrm{i}\varphi}$ 不完全相同，多了一个参数 t，

少了 2π，而且相位估计是将相位的 2^l 倍存储到基态中，因此令

$$t=\frac{2\pi}{2^l} \tag{3.8.17}$$

则

$$\boldsymbol{U}\mid u_i\rangle=\mathrm{e}^{\frac{2\pi\mathrm{i}\lambda_i}{2^l}}\mid u_i\rangle \tag{3.8.18}$$

就能够将相位 λ_i 存储到基态中。

第三步：由附录 B.3 可知，关于基态 $|\tilde{\lambda}_i\rangle$ 的函数 $f(\tilde{\lambda}_i)=\frac{C}{\tilde{\lambda}_i}$（$C$ 是归一化因子）能够在量子计算机上实现，因此以 $|\tilde{\lambda}_i\rangle$ 作为控制比特，使用 $f(\tilde{\lambda}_i)$ 对辅助量子比特进行旋转，将特征值存储到振幅中。

$$\mid\psi_3\rangle=\sum_{i=0}^{N-1}\left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_i^2}}\mid 0\rangle+\frac{C}{\tilde{\lambda}_i}\mid 1\rangle\right)\beta_i\mid\tilde{\lambda}_i\rangle\mid u_i\rangle \tag{3.8.19}$$

这里受控旋转可看作一个映射 $R(f)$，将辅助量子比特由基态 $|0\rangle$ 映射到 $|0\rangle$ 和 $|1\rangle$ 的叠加态上，同时将函数值 $f(\tilde{\lambda}_i)$ 提取到 $|1\rangle$ 的振幅上。

第四步：退计算。

一方面，对比想要的量子态（式(3.8.13)）和得到的量子态（式(3.8.19)）可以看出，式(3.8.19)中多了第二寄存器中存储的 $|\tilde{\lambda}_i\rangle$；另一方面，对比式(3.8.14)、式(3.8.15)和式(3.8.19)可以看出，只要对式(3.8.19)的第二寄存器和第三寄存器执行逆相位估计就可以将式(3.8.19)中第二寄存器演化为 $|0\rangle^{\otimes l}$，该步骤称为退计算。退计算的目的是解除第二寄存器与其余两个寄存器的纠缠。

执行退计算，量子态演化为

$$\mid\psi_4\rangle=\sum_{i=0}^{N-1}\left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_i^2}}\mid 0\rangle+\frac{C}{\tilde{\lambda}_i}\mid 1\rangle\right)\beta_i\mid 0\rangle\mid u_i\rangle \tag{3.8.20}$$

此时忽略第二寄存器，对第一寄存器进行测量，当测量结果为 1，则得到量子态

$$\mid\psi\rangle=\sum_{i=0}^{N-1}C\frac{\beta_i}{\tilde{\lambda}_i}\mid u_i\rangle \tag{3.8.21}$$

除去归一化因子 C，式(3.8.21)与式(3.8.13)等价。由此完成了用量子算法求解线性方程组。

需要注意的是，在 HHL 算法中退计算用的是逆相位估计。这里之所以用到逆相位估计，是因为第二步使用相位估计将特征值存储在第二寄存器中。逆相位估计最终目的是将特征值存储在第一寄存器的振幅中。在后续的其他算法中也有退计算的概念，均起到解纠缠的作用，也就是解除目标量子比特与辅助量子比特之间的纠缠，从而得到目标量子态。至于采用哪种方法退计算，要看具体算法中用哪种方法将目标量子比特与辅助量子比特纠缠在一起，一般采用逆运算来实现解纠缠，即退计算。

根据式(3.8.21)，HHL 算法的输出存放在第三寄存器的叠加态中。而 $|\psi\rangle$ 是一个向量，要想提取出向量的每一个元素并不容易。不过，在很多使用 HHL 的量子算法中并不

需要将$|\psi\rangle$的具体元素提取出来，而是直接使用$|\psi\rangle$进行后续的操作。在这种情况下HHL算法还是相当高效的，广泛应用于量子主成分分析、量子支持向量机、量子回归等量子机器学习算法中。

此外，该算法的复杂度主要体现在第二步和第四步中的哈密顿量模拟上，也就是$U=e^{iAt}$的实现，而哈密顿量模拟的复杂度为$O(\text{poly}(\log N))$。因此HHL算法的复杂度为$O(\text{poly}(\log N))$。

3.8.4 实现

对于方程组

$$\frac{1}{2}\begin{pmatrix}3 & 1\\ 1 & 3\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}=\begin{pmatrix}0\\ 1\end{pmatrix}$$

来说，如果使用经典矩阵求逆算法可得其解为

$$(x_1 \quad x_2)^{\mathrm{T}}=\left(-\frac{1}{4} \quad \frac{3}{4}\right)^{\mathrm{T}}$$

使用HHL算法求解该方程组时，令

$$\boldsymbol{A}=\frac{1}{2}\begin{pmatrix}3 & 1\\ 1 & 3\end{pmatrix},\quad \boldsymbol{b}=\begin{pmatrix}0\\ 1\end{pmatrix}$$

量子线路图如图3.33所示，其中q_0是受控旋转的辅助量子比特，q_1和q_2存储矩阵$\boldsymbol{A}$的特征值，q_3存储$\boldsymbol{b}$。第一步在q_3上制备$\boldsymbol{b}=\begin{pmatrix}0\\ 1\end{pmatrix}$，第二步为量子相位估计，第三步为受控旋转，第四步为逆量子相位估计，最后是测量。

由3.8.3节第二步的分析可知，$t=\dfrac{2\pi}{2^l}$。本实验中用两个量子比特存储$\boldsymbol{A}$的特征值，因此$l=2$，则模拟$\boldsymbol{U}=e^{i\boldsymbol{A}t}$被转换为模拟$\boldsymbol{U}=e^{i\boldsymbol{A}\frac{2\pi}{4}}$。量子相位估计要模拟的算子包括受控$\boldsymbol{U}^{2^0}=e^{i\boldsymbol{A}\frac{2\pi}{4}}=e^{i\frac{\boldsymbol{A}}{4}2\pi}$和受控$\boldsymbol{U}^{2^1}=e^{i\frac{\boldsymbol{A}}{2}2\pi}$。$e^{i\frac{\boldsymbol{A}}{4}2\pi}$可以做如下分解：

$$e^{i\frac{\boldsymbol{A}}{4}2\pi}=e^{i\frac{1}{2}(3\boldsymbol{I}+\boldsymbol{X})\frac{\pi}{2}}=e^{i\left(\frac{3\pi}{4}\right)\boldsymbol{I}}\times e^{-i\left(-\frac{\pi}{4}\right)\boldsymbol{X}} \tag{3.8.22}$$

又因为

$$e^{i\frac{3\pi}{4}\boldsymbol{I}}=\cos\frac{3\pi}{4}\boldsymbol{I}+i\sin\frac{3\pi}{4}\boldsymbol{I}=\begin{pmatrix}e^{i\frac{3\pi}{4}} & 0\\ 0 & e^{i\frac{3\pi}{4}}\end{pmatrix}$$

因此受q_1控制对q_2执行$e^{i\frac{3\pi}{4}\boldsymbol{I}}$等价于直接对$q_1$执行

$$\boldsymbol{U}_1\left(\frac{3\pi}{4}\right)=\begin{pmatrix}1 & 0\\ 0 & e^{i\frac{3\pi}{4}}\end{pmatrix}$$

此外，有

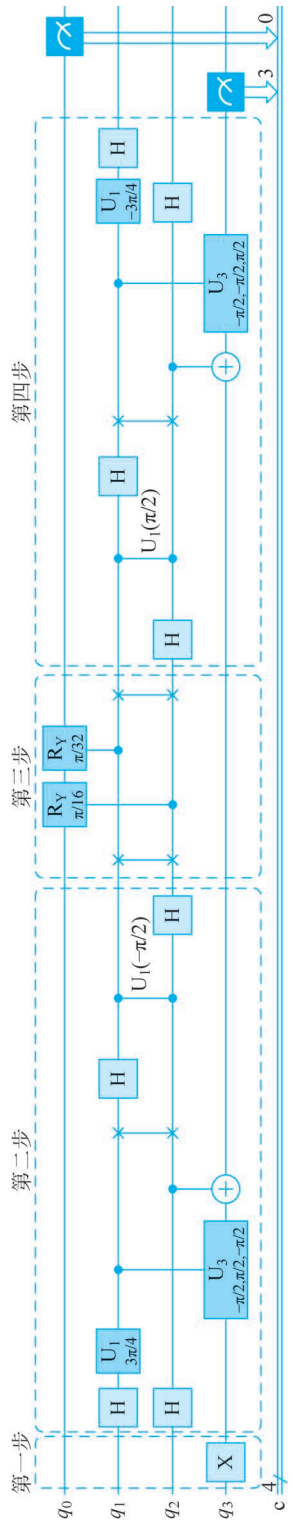


图 3.33　求解线性方程组的量子线路图

$$e^{-iX\left(-\frac{\pi}{4}\right)}=\cos\left(-\frac{\pi}{4}\right)\boldsymbol{I}-i\sin\left(-\frac{\pi}{4}\right)\boldsymbol{X}$$
$$=\begin{pmatrix}\cos\left(-\frac{\pi}{4}\right) & -i\sin\left(-\frac{\pi}{4}\right)\\ -i\sin\left(-\frac{\pi}{4}\right) & \cos\left(-\frac{\pi}{4}\right)\end{pmatrix}=\boldsymbol{U}_3\left(-\frac{\pi}{2},-\frac{\pi}{2},\frac{\pi}{2}\right)\quad(3.8.23)$$

因此受控 $\boldsymbol{U}^{2^0}=e^{i\frac{\boldsymbol{A}}{4}2\pi}$ 可以使用 $\boldsymbol{U}_1\left(\frac{3\pi}{4}\right)$ 和受控 $\boldsymbol{U}_3\left(-\frac{\pi}{2},-\frac{\pi}{2},\frac{\pi}{2}\right)$ 来实现。

由于 $\boldsymbol{U}^{2^1}=e^{i\frac{\boldsymbol{A}}{2}2\pi}$ 可以做如下分解：

$$e^{i\frac{\boldsymbol{A}}{2}2\pi}=e^{i(3\boldsymbol{I}+\boldsymbol{X})\frac{\pi}{2}}=e^{i\left(\frac{3\pi}{2}\right)\boldsymbol{I}}\times e^{-i\left(-\frac{\pi}{2}\right)\boldsymbol{X}}\quad(3.8.24)$$

又因为

$$e^{i\frac{3\pi}{2}\boldsymbol{I}}=\cos\frac{3\pi}{2}\boldsymbol{I}+i\sin\frac{3\pi}{2}\boldsymbol{I}=-i\boldsymbol{I}$$
$$e^{-i\left(-\frac{\pi}{2}\right)\boldsymbol{X}}=\cos\left(-\frac{\pi}{2}\right)\boldsymbol{I}-i\sin\left(-\frac{\pi}{2}\right)\boldsymbol{X}=i\boldsymbol{X}\quad(3.8.25)$$

因此

$$\boldsymbol{U}^{2^1}=e^{i\frac{\boldsymbol{A}}{2}2\pi}=e^{i\left(\frac{3\pi}{2}\right)\boldsymbol{I}}\times e^{-i\left(-\frac{\pi}{2}\right)\boldsymbol{X}}=-i\boldsymbol{I}\times i\boldsymbol{X}=\boldsymbol{X}\quad(3.8.26)$$

这说明 $\boldsymbol{U}^{2^1}$ 等价于量子非门，因此受控 $\boldsymbol{U}^{2^1}$ 等价于受控非门。

如图 3.34 所示，共有三种测量结果，其中 q_1 和 q_2 在逆量子相位估计之后变为 $|0\rangle$，仅当 q_0 的测量结果为 $|1\rangle$ 时，才能从 q_3 的叠加态中获取解的信息，即图中的 0001 与 1001 两种情况。为了更好地分析结果，输出 200000 次测量的结果统计：

{'0000':0, '1001':271, '0001':30, '1000':199699}

其中 0001 与 1001 分别为 30 次和 271 次。由于在 HHL 算法会产生归一化参数，因此这里并不能给出具体的解，但是可以给出解的平方的比例，即 $\left|\frac{x_1}{x_2}\right|^2=\frac{30}{271}$。而理论结果为 $\left|\frac{x_1}{x_2}\right|^2=\frac{(-1/4)^2}{(3/4)^2}=\frac{1}{9}$，实验结果与理论结果相吻合。

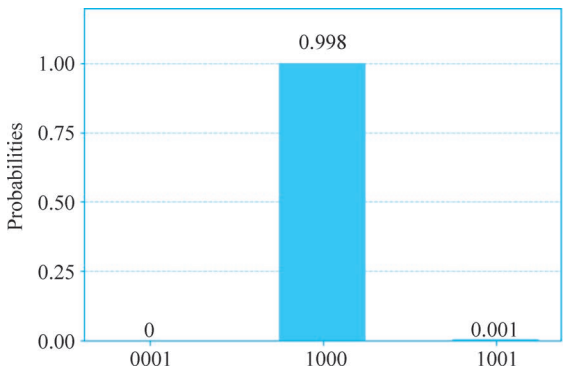


图 3.34　求解线性方程组实验结果

求解线性方程组的代码如下：

```
%matplotlib inline
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute
from qiskit import Aer
from qiskit import IBMQ
from math import pi
from qiskit.tools.visualization import plot_histogram

circuit = QuantumCircuit(4,4)

#第一步：制备|b>
circuit.x(3)
circuit.barrier(0,1,2,3)

#第二步：相位估计求特征值
circuit.h(1)
circuit.h(2)
circuit.u1(3*pi/4,1)
circuit.cu3(-pi/2,-pi/2,pi/2,1,3)
circuit.cx(2,3)
circuit.swap(1,2)
circuit.h(1)
circuit.cu1(-pi/2,2,1)
circuit.h(2)

#第三步：特征值取反
circuit.swap(1,2)
#受控旋转将特征值提取到辅助量子比特
circuit.cry(pi/16,2,0)
circuit.cry(pi/32,1,0)
#对2、3、4量子比特进行解纠缠
circuit.swap(1,2)

#第四步
circuit.h(2)
circuit.cu1(pi/2,2,1)
circuit.h(1)
circuit.swap(1,2)
circuit.cx(2,3)
circuit.cu3(-pi/2,pi/2,-pi/2,1,3)
circuit.u1(-3*pi/4,1)
circuit.h(1)
circuit.h(2)
circuit.barrier(0,1,2,3)

#测量
circuit.measure(3,3)
circuit.measure(0,0)

```

```
#绘制线路图
circuit.draw(output = 'mpl',plot_barriers = False)
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit,backend,shots = 200000)
sim_result = job_sim.result()

#绘制结果图
measurement_result = sim_result.get_counts(circuit)
plot_histogram(measurement_result)
```

3.9 本章小结

本章对量子机器学习算法中要用的一些基础算法进行介绍。3.1 节介绍的量子态制备是量子机器学习算法最基本的步骤，只有将数据等信息存储在量子计算机中才能实现后续的算法。3.2 节讲述量子搜索算法，该算法能够快速在非结构化无序数据库中找到符合条件的元素，在量子 K 近邻和量子聚类等需要寻找最相近的几个元素的算法中起着重要的作用。3.3 节是量子傅里叶变换，该算法及其逆能够将基态存储和振幅存储进行转换。3.4 节和 3.5 节是量子估计算法。3.4 节的量子相位估计算法能够估计特征值的相位，为后续需要提取特征值信息的算法打下基础，3.5 节的量子振幅估计能够估计量子态的振幅。3.6 节和 3.7 节是交换测试和哈达玛测试，能够分别以内积模和内积两种方式估计两个量子态之间的相似度。3.8 节的 HHL 算法利用量子特性加速解决线性方程组求解的问题，是量子支持向量机、量子线性回归等算法的核心。

表 3.1 总结了除量子态制备之外的量子基本算法的功能及其在量子机器学习中的应用。

表 3.1 量子基本算法的功能及其在量子机器学习中的应用

算　法	功　能	应　用
量子搜索	在非结构化无序数据中找到符合条件的元素	量子 K 近邻 量子分裂层次聚类 量子谱聚类 基于经典环境的例子强化学习
量子傅里叶变换	频域变换 将数据在基态存储和振幅存储之间进行转换	量子相位估计 量子振幅估计 HHL
量子相位估计	用于估计算子 $\boldsymbol{U}$ 的特征值 $e^{2\pi i\varphi}$ 的相位 φ	基于相位估计的量子主成分分析 量子奇异值阈值算法 量子线性判别分析 量子支持向量机 量子 K 近邻 量子线性回归 量子岭回归 量子谱聚类

续表

算　法	功　能	应　用
量子振幅估计	用于估计包含解的概率	量子 K 近邻 量子逻辑回归
交换测试	用于计算两个量子态的内积的模	基于交换测试的量子主成分分析 量子 K 近邻 量子 K 均值聚类 量子凝聚层次聚类 量子分裂层次聚类 量子生成对抗网络
哈达玛测试	用于计算两个量子态的内积	量子支持向量机 量子线性回归 量子岭回归 量子逻辑回归 量子神经网络
HHL	用于求解线性方程组	量子支持向量机 量子线性回归 量子岭回归

参考文献