

量子助力金融优化： 量子变分-凸优化混合算法在组合优化问题上的应用

2024 年 11 月 1 日

1 实验背景

金融市场中通常面对许多复杂的数学问题，本实验聚焦于股票的组合优化问题。组合优化领域常用均值方差模型，通过获取足够多时间序列的股票收益率数据，以收益率的均值预测股票未来收益，以收益率的方差体现股票收益率的波动性。本实验采用均值方差模型，将量子算法与金融背景相结合，旨在为组合优化问题提出一种解决方案。

均值方差模型如下，这是一个凸优化问题：

$$\begin{aligned} \max_w & (w^T r - \gamma w^T V w) \\ \text{s.t. } & e^T w = 1, \\ & 0 \leq w \leq 1, \end{aligned} \quad (1)$$

其中， w 是表示权重的 k 维向量， r 是表示预期收益的 k 维向量， γ 是风险厌恶系数， V 是表示方差的 $k \times k$ 矩阵。

2 实验原理

量子算法的优势在于它强大的并行计算能力，然而它难以直接处理凸优化这样的二次数学问题。因此，首先使用经典算法 ADMM 将凸优化问题转变成量子算法方便处理的线性问题，然后使用变分量子算法解决线性问题获得凸优化的解。

2.1 凸优化算法：交替方向乘子法 ADMM

2.1.1 原始公式

$$\begin{aligned} \min_{w,z} & (f(x) + g(z)) \\ \text{s.t. } & AX + Bz = c, \end{aligned} \quad (2)$$

其中 f 和 g 是凸函数， A 和 B 是矩阵， c 是向量

问题 2 的增广拉格朗日函数为

$$L_\rho(x, y, z) = f(x) + g(z) + \lambda^T (AX + Bz - c) + \frac{\rho}{2} \cdot \|AX + Bz - c\|_2^2 \quad (3)$$

其中， λ 是表示拉格朗日对偶变量的尺度的向量， ρ 是用户定义的正的惩罚系数，用于控制惩罚项的强度，实际上只有这一个惩罚项。

定义 $\mu = \rho^{-1}\lambda$ ，ADMM 使用的迭代方法如下：

$$\begin{aligned}
x^{k+1} &= \arg \min_x f(x) + \frac{\rho}{2} \cdot \|Ax + Bz^k - c + \mu^k\|_2^2 \\
z^{k+1} &= \arg \min_z f(z) + \frac{\rho}{2} \cdot \|Ax^{k+1} + Bz - c + \mu^k\|_2^2 \\
\mu^{k+1} &= \mu^k + Ax^{k+1} + Bz^{k+1} - c
\end{aligned} \tag{4}$$

其中, x^k 和 z^k 是第 k 次迭代的决定性变量。这三个迭代步骤的意义是: 更新 x , 要求每次迭代得到的 x 使包含 $f(x)$ 和惩罚项的总和最小; 更新 z , 要求包含 $g(z)$ 和惩罚项的总和最小; 更新 μ , 这是拉格朗日乘子的更新, 通过将当前残差 $Ax^{k+1} + Bz^{k+1} - c$ 加到当前的拉格朗日乘子上, 这样 μ 会根据当前迭代的残差进行调整。

定义 $r^k = Ax^k + Bz^k - c$ 为第 k 次迭代的原始残差, $s^k = \rho A^T B(z^k - z^{k-1})$ 为第 k 次迭代的对偶残差, 迭代停止的标准是原始残差和对偶残差分别小于用户定义的停止容差 ϵ^p 和 ϵ^d 。

2.1.2 ADMM 求解凸优化问题

本项目关注的凸优化问题的目标函数包含一个二次项 (方差项) 和一个一次项 (预期收益项), 不妨令 $f(x)$ 和 $g(z)$ 的表达式如公式 5 所示, 这样的形式恰好符合凸优化问题对目标函数和约束条件的要求

$$\begin{aligned}
f(x) &= \frac{1}{2} x^T Q x + x^T p \\
s.t. & Ax = b
\end{aligned} \tag{5}$$

$$g(z) = I_{l \leq z \leq u}(z) \tag{6}$$

其中, $g(z)$ 是表示线性不等式约束的指示函数。按照 ADMM 算法的思路表示原问题, 可写作

$$\begin{aligned}
\min & \left(\frac{1}{2} x^T Q x + x^T p + I_{l \leq z \leq u}(z) \right) \\
s.t. & x - z = 0
\end{aligned} \tag{7}$$

应用公式 4 提出的迭代方法, 可以得到以下公式:

$$x^{k+1} = \arg \min_{x|Ax=b} \frac{1}{2} x^T Q x + x^T p + \frac{\rho}{2} \cdot \|x - z^k + \mu^k\|_2^2 \tag{8a}$$

$$z^{k+1} = \arg \min_{l \leq z \leq u} \frac{\rho}{2} \cdot \|x^{k+1} - z + \mu^k\|_2^2 \tag{8b}$$

$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \tag{8c}$$

对于公式 8, 首先, 注意到 (8b) 是一个带有箱约束的最小二乘问题, 因此可以解析求解:

$$\Pi(x) = \begin{cases} l, & \text{if } x < l \\ x, & \text{if } l \leq x \leq u \\ u, & \text{if } x > u \end{cases} \tag{9}$$

所以, 公式 8b 得到的解析解如下式:

$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \tag{10}$$

公式 8a 是一个等式约束的二次规划, 也可以解析求解。具体来说, 当 x 和 λ 满足 KKT 最优性条件时, x 是公式 8a 的解。KKT 条件应该有两个: (1) 公式 8a 关于变量 x 的梯度为 0, (2) 公式 8a 满足关于 x 的约束条件。这样, KKT 条件可表示如下:

$$\begin{aligned}
L(x^{k+1}, \lambda) &= \frac{1}{2}x^T Qx + x^T p + \frac{\rho}{2} \cdot \|x - z^k + \mu^k\|_2^2 - \lambda(Ax - b) \\
L'_x(x^{k+1}, \lambda) &= Qx + p + \rho(x - z^k + \mu^k) - \lambda A = 0 \\
Ax &= b
\end{aligned}$$

令 $\eta^{k+1} = -\lambda$ 化简后以矩阵的形式表示如下：

$$\begin{pmatrix} Q + \rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = - \begin{pmatrix} p - \rho(z^k - \mu^k) \\ -b \end{pmatrix} \quad (11)$$

也即，形如公式 5、6 的函数可通过以下迭代式求解：

$$\begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = - \begin{pmatrix} Q + \rho I & A^T \\ A & 0 \end{pmatrix}^{-1} \begin{pmatrix} p - \rho(z^k - \mu^k) \\ -b \end{pmatrix} \quad (12a)$$

$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \quad (12b)$$

$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \quad (12c)$$

如果 ρ 是静态的，那么公式 12a 中的系数矩阵在每次迭代中保持不变，仅在 ADMM 算法开始时被分解一次。这样的特性对于提高计算效率非常有帮助。

现在将这个凸优化问题与均值方差模型对比：

$$\begin{aligned}
\min f(x) &= \frac{1}{2}x^T Qx + x^T p \\
g(z) &= I_{l \leq z \leq u}(z) \\
s.t. & Ax = b \\
\min_w & \gamma w^T V w - w^T r \\
s.t. & e^T w = 1, \\
& 0 \leq w \leq 1,
\end{aligned} \quad (13)$$

若要应用公式 12 的结论，应做如下变换：

$$Q = 2\gamma V \quad (14a)$$

$$p = -r \quad (14b)$$

$$A = e^T \quad (14c)$$

$$b = 1 \quad (14d)$$

$$l = 0 \quad (14e)$$

$$u = 1 \quad (14f)$$

带入到结果如下：

$$\begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = - \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} -r - \rho(z^k - \mu^k) \\ -1 \end{pmatrix} = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} r + \rho(z^k - \mu^k) \\ 1 \end{pmatrix} \quad (15a)$$

$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \quad (15b)$$

$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \quad (15c)$$

$$\text{令 } M = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}, \quad b = \begin{pmatrix} r + \rho(z^k - \mu^k) \\ 1 \end{pmatrix}$$

M 是一个常数矩阵，而且是实对称矩阵，所以只需在迭代开始前计算一次 M^{-1} ，就可以专注于变量之间的迭代。根据公式 15 的要求，需要解出 x 的值才能计算另外两个变量，所以现在的问题是求解公式 15a 这个线性方程组。

接下来，我们尝试用变分量子算法求解线性方程组。

2.2 变分量子算法

目前已经有很多关于量子算法求解线性问题的研究，其中最著名的当属 HHL 算法，它就是为了解线性方程组而提出的。以线性问题 $Mx = b$ 为例，HHL 算法针对的情况和本项目类似，已知矩阵 M 和向量 b ，求解未知向量 x 。但进一步了解，我认为 HHL 算法无法应用到本项目，原因如下：

1. HHL 算法用到的哈密顿模拟要求系数矩阵 M 是稀疏矩阵，且稀疏度与算法性能直接相关。但本项目中的系数矩阵异常稠密，甚至只有一个零值，大大削弱了量子算法原本的性能优势。
2. HHL 算法要求已知 M 的特征向量或奇异矩阵，这在制备量子态 $|b\rangle$ 在特征向量上的投影时会用到。然而对于本项目的体量，如果已知 M 的特征向量，将 M 进行谱分解后直接求解向量 x 会更合理更高效，使用 HHL 算法是舍近求远。

综上，本项目采用变分量子算法，设计哈密顿算符，使待解的量子态作为哈密顿算符的本征态，将求解线性方程组的问题转化成求解哈密顿算符的本征态的问题。

2.2.1 设计哈密顿算符 H

回到最初的线性方程组，若将 $Mx = b$ 中的 x 和 b 向量归一化并构建为量子态，则原问题的解可表示为 $|x\rangle = \frac{M^{-1}|b\rangle}{\|M^{-1}|b\rangle\|}$ 。设计哈密顿算符为 $H = M^\dagger(I - |b\rangle\langle b|)M$ ，则 $|x\rangle$ 是 H 的一个本征值为零的本征态，因为有：

$$\begin{aligned} \langle x|H|x\rangle &= \frac{\langle b|M^{-\dagger}}{\|\langle b|M^{-\dagger}\|} M^\dagger(I - |b\rangle\langle b|)M \frac{M^{-1}|b\rangle}{\|M^{-1}|b\rangle\|} \\ &= \frac{\langle b|(I - |b\rangle\langle b|)|b\rangle}{\|M^{-1}|b\rangle\|^2} \\ &= \frac{\langle b|b\rangle - \langle b|b\rangle\langle b|b\rangle}{\|M^{-1}|b\rangle\|^2} \\ &= \frac{\langle b|b\rangle - \langle b|b\rangle}{\|M^{-1}|b\rangle\|^2} \\ &= 0 \end{aligned} \tag{16}$$

又因为 M 是厄密矩阵，所以 $\langle x|H|x\rangle = \langle x|M(I - |b\rangle\langle b|)M|x\rangle = \sum_i |\langle x|M|x_i\rangle|^2 \geq 0$ ，其中， $I - |b\rangle\langle b| = \sum_i |x_i\rangle\langle x_i|$ ， $\langle x_i|x_j\rangle = \delta_{i,j}$ 。所以哈密顿算符 H 是一个半正定矩阵， $|x\rangle$ 也就是 H 的基态。求解线性方程组的任务就转化为寻找哈密顿算符基态的任务。

2.2.2 设计量子线路图

为了实现变分量子算法寻找哈密顿量基态，应首先使用含参量子电路模拟一个量子态 $|x\rangle = \frac{M^{-1}|b\rangle}{\|M^{-1}|b\rangle\|}$ ，然后将哈密顿算符表示出来，并计算其在模拟量子态上的期望。由于哈密顿算符的期望的理论值为 0，所以将求解梯度的算符的损失函数定义为 $L(\theta) = \langle \phi(\theta)|H|\phi(\theta)\rangle$ ，将其合并到量子神经网络中，使用梯度下降法持续优化参数 θ 。根据最终得到的最优参数 θ 计算出使 H 取到基态本征值的量子态。

在构建电路时，除了含参量子门，还需放置控制非门在不同量子线路之间模拟量子纠缠。

2.3 小结

总的来说，本项目的思路是，先通过经典凸优化算法 ADMM 将目标函数和约束条件转化为一个迭代问题，其中主要的计算任务是求解一组线性方程组。线性方程组的系数矩阵和向量均已知，所以构造变分量子线路求解。得到线性方程组的解后，更新迭代方程，又会产生新的向量 b ，然后重新计算一组解。直到迭代方程的原始残差和对偶残差分别小于停止容差，迭代停止，此时量子线路中的量子态即为最终结果。

3 实验结果

本项目从雅虎财经网站获得纳斯达克交易所 (NASDAQ) 的 15 只股票从 2023-06-01 到 2024-06-30 的收益率数据，15 只股票分别是 AKR、AMD、AMKR、AMZN、ANET、AROC、BAC、BAM、BK、BKKT、BKNG、BKR、BLK、CBOE、CHKP，都是资产份额十亿美元级别的股票。

股票收益率均值和协方差矩阵的计算参考了 Barra 模型，引入半衰期的概念，确保距离现在越近的收益率数据在均值计算中所占权重越大、对参数计算的影响越大。由于求解预期收益率并非本项目的重点，所以使用 2024-06-30 的收益率数据作为股票的预期收益，使用 2024-06-29 及之前的数据用作协方差矩阵的计算。

量子部分中，15 只股票得到的协方差矩阵 V 是 15×15 ，经 ADMM 算法整理后，参与迭代的系数矩阵 $M = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}$ 是 16×16 的实对称矩阵。所以本项目所需的量子线路数 $n = \log_2 16 = 4$ 。量子部分使用 mindquantum 库，所用到的代码在华为云上运行。

本项目中用到的参数及其说明如下表：

表 1: 参数及其说明

参数	参数说明	取值
k	股票个数	15
h	时间窗，用于协方差矩阵的计算	252
τ	半衰期	252
n	量子线路的量子比特数	4
γ	风险厌恶系数	15
ρ	增广拉格朗日函数中的惩罚系数	1
λ	拉格朗日对偶变量	
μ	$\rho^{-1}\lambda$	
ϵ^p	原始残差的停止容差	0.001
ϵ^d	对偶残差的停止容差	0.001

3.1 关于 γ 取值的确定

根据公式 13， γ 是衡量目标函数中预期收益和风险两项的相对比例的重要参数，所以在确定它的初值时，应保证：(1) 预期收益和风险两项在同一数量级，目标函数值不至于被某一项影响过大；(2) 目标函数值不能过小，否则可能因为计算精度不够而使变化不够明显。

当所有股票的权重数值都相等时，令预期收益和风险两项相等的风险厌恶系数是 10，综合考虑最终选择

$$\gamma = 15$$

3.2 量子线路图

由于构造的系数矩阵 M 是 16×16 的大小，所以在 $n = 4$ 个量子比特的量子态构建浅层量子电路实现对 $|x\rangle$ 量子态的近似编码。共计 16 个 $R_Y(\theta)$ 门和 7 个 CNOT 门。需要注意的是，电路中的 4 个 $R_X(\theta)$ 门为 encoder 部分，它们的参数从始至终都是 0，且不参与变分量子算法的计算，相当于 4 个 $R_X(0)$ 门，对量子态不产生任何变化。

初始的含参量子门电路如图 1 所示。

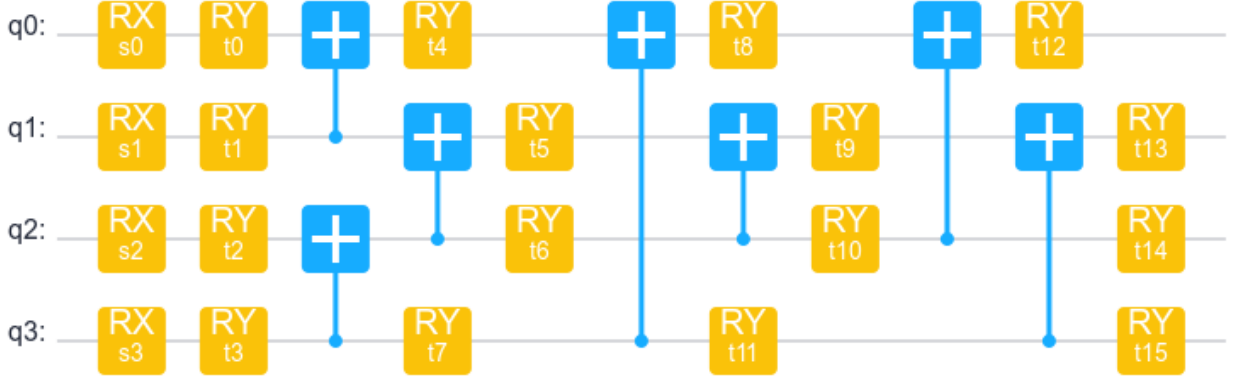


图 1: 含参量子线路图

3.3 迭代次数

定义原始残差的停止容差为 $\epsilon^p = 0.001$ ，对偶残差的停止容差为 $\epsilon^d = 0.001$ ，确保在原始残差和对偶残差都小于停止容差之前不会停止迭代。共计迭代 121 次，所以，图 2 和图 3 上均有 121 个点。

3.3.1 哈密顿算符的期望值

哈密顿算符以稀疏矩阵的形式作用于图 1 所示的线路图，在每次迭代中得到一个期望值。根据章节 2.2.1 的推导，这个期望的理论值为 0，图 2 为历次迭代中哈密顿算符期望值，可以观察到只有 2 次迭代的结果与理论值的误差超过 0.07%，证明本项目所用理论正确，所得结果有效。

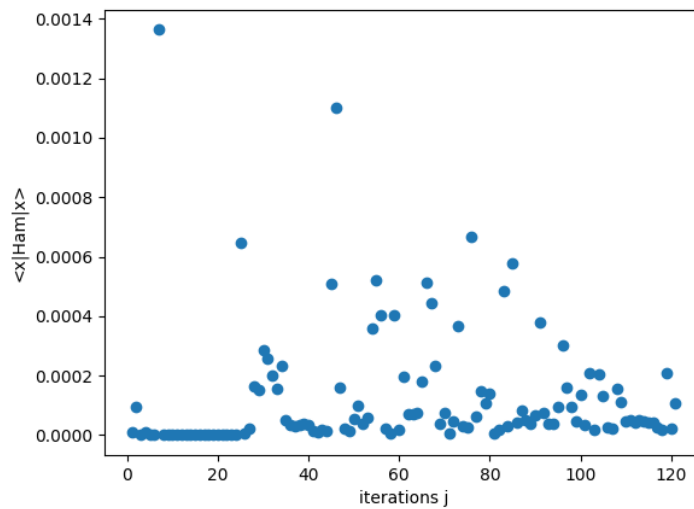


图 2: 哈密顿算符期望值

3.3.2 目标函数值 $\gamma w^T V w - w^T r$

将每次迭代中计算到的线性方程组的解带入目标函数中，观察到目标函数的变化如图 3所示，呈现整体下降趋势，直到最后趋于稳定，与章节 2.1.2中 ADMM 算法的目标函数寻找最小值的趋势相吻合。

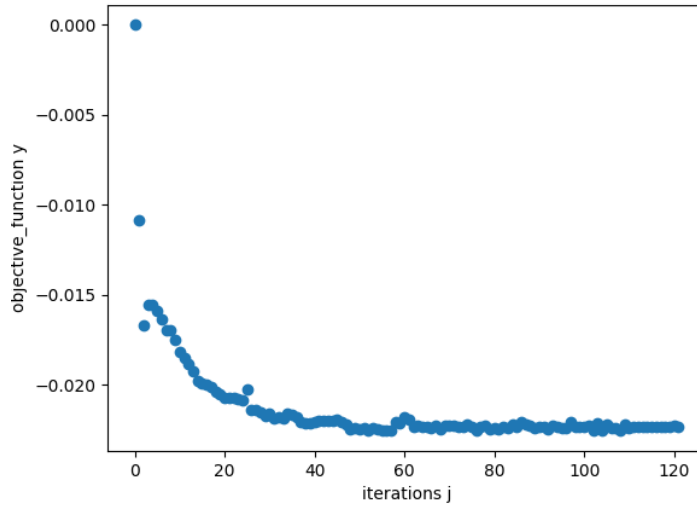


图 3: 目标函数值

3.4 与经典算法的结果对照

全部计算完成后，所得量子线路图中各量子门的最优参数如图 4所示。

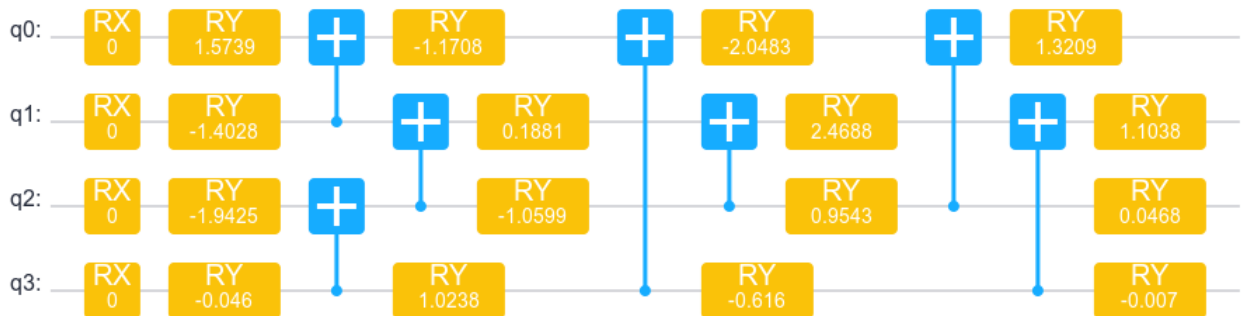


图 4: 含最优参数的量子线路图

根据这些参数得到股票的最优权重如表 2的第二列所示。其中部分股票的权重为负值，考虑到构建迭代时已经考虑到权重的取值在 $[0, 1]$ ，所以推测应为计算过程中积累的误差最终表现为权重为负。将权重中的负值赋为零后重新归一化得到的权重如表 2的第三列。第四列为使用经典凸优化算法的 *cvxpy* 库获得的权重结果。由于第二列是量子-经典算法实际得到的结果，所以按照第二列的数值分析结果。

对比两种方法的目标函数值，量子-经典算法的数值稍高，与经典算法的偏差在 -0.16% 。对比两种方法的权重数据，发现两种算法对股票的选择都集中在第 0、2、11 号股票上，分别是 AKR、AMKR、BKR。说明本项目最开始选择的 15 只股票和参数设置（如 γ ）存在较大的局限性，这三只股票的预期收益和稳定性明显好于其余股票，其他股票对目标函数的最终结果的贡献较低，导致组合优化几乎只是在这三只股票之间进行的，没有体现出分散投资的优越性。

表 2: 量子-经典算法与经典方法计算结果对比

序号	量子-经典算法权重	剔除负值的量子-经典算法权重	经典算法权重
0	2.43929843e-01	2.42882733e-01	1.92118776e-01
1	-1.48296628e-03	0	2.10135440e-22
2	4.80616983e-01	4.78553853e-01	5.25081379e-01
3	-6.96886244e-04	0	7.49192735e-23
4	5.56502021e-05	5.54113142e-05	9.69256530e-23
5	2.69331050e-03	2.68174901e-03	1.48358147e-22
6	-4.21995417e-04	0	9.86542067e-23
7	3.87218790e-03	3.85556588e-03	6.50809847e-23
8	-2.43143670e-04	0	1.67722455e-22
9	-3.99785463e-04	0	-6.30689551e-23
10	-9.07155027e-04	0	7.13622351e-23
11	2.72523170e-01	2.71353318e-01	2.82799845e-01
12	6.20030959e-04	6.17369371e-04	1.64311697e-22
13	-1.51856999e-04	0	6.53047173e-23
14	-7.38576272e-06	0	6.70058343e-23
$\gamma w^T V w - w^T r$	-0.02232539049260529	-0.022278119863211213	-0.022361270760996213

4 反思与讨论

本项目采用 ADMM 和变分量子算法相结合的思路求解股票市场的组合优化问题。首先用 ADMM 算法将组合优化问题转化为迭代求解线性方程组的问题，然后再使用变分量子算法构造神经网络求解方程组。迭代结束时量子线路中的量子态即为组合优化问题的解。将最终得到的权重与经典方法计算得到的权重对比，注意到两者的目标函数值和权重基本一致，但量子经典混合算法的计算精度远不及经典方法，导致有较多负值产生。实际应用中，应将这些负值赋为零，并重新归一化权重。

经总结，我认为本项目仍有以下问题值得进一步探索：

1. 稠密的系数矩阵 M 的表示。目前较成熟的求解线性方程组的量子方法 HHL 要求系数矩阵是稀疏的，而且量子加速的程度也与稀疏度密切相关。主要是因为 HHL 用到的哈密顿模拟要求系数矩阵是稀疏的。本项目中，系数矩阵 M 是异常稠密的矩阵，但本项目没有想到合理的解决办法，而是选择将它转化为稀疏矩阵的表示方法并直接参与运算。稀疏矩阵的表示方法在 mindquantum 库中是被允许直接用于构造哈密顿算符的，再加上本项目量子规模较小，相当于钻了 mindquantum 库的空子。
2. ADMM（或其他凸优化方法）与量子算法更合适的结合方式。使用 ADMM 迭代方法求解组合优化问题的一大优点是参与迭代的线性方程组的系数矩阵是常数矩阵，每次迭代需要重新计算的只有线性方程组中右侧的向量，但本项目中使用的哈密顿算符中同时出现了 M 和 b ，所以实际上每次迭代仍然要重新计算哈密顿算符，没有利用好 ADMM 算法的优势。