IBM Qiskit Fall 2021 Challenge

Challenge 1. Portfolio Optimization

minimize risk, maximiz returns

Qiskit Finance Module solves this problem specifically

PortfolioOptimization object converts problem into a quadratic program

↳ solved with VQE or QAOA

risk is quantified by the historical variance of an equity value

* Q for IBM → you really want the $\frac{\Delta v}{\Delta t}$ for each stock over some time range Δt, where Δv is the change in ~~state~~ value $v_t - v_{t=0}$, $\Delta t = t - t_0$

But the notebook states to measure the mean value v of a stock. Nevermind, this is addressed

We are optimizing the risk (minimize) for a given ~~stock return~~ overall stock return by ~~weights~~ choosing the ~~amount~~ amount of each stock to purchase, given some budget. Or equivalently, the relative amount of each stock for any budget.

Q → how do we translate the simplified version of the problem described in the IBM notebook to a real-world version of the problem where

- not all assets have the same price
  ↳ is it as simple as rescaling?

Q → RandomDataProvider object returns ~~data~~ random stock data

.get_period_return_mean_vector

↳ $\vec{\mu}$ why is $\mu_1 > 0$?

why is $\mu_1 > \mu_0$?

$\mu_2 > \mu_0$?

In general, use case is to formulate a problem as a quadratic / linear optimization problem, then use VQE (variational quantum eigensolver) or QAOA to compute a solution.

1. Need to understand how to formulate problems into quadratic optimizations

2. Need to understand VQE / QAOA

"VQE / QAOA can only be applied to quadratic unconstrained binary optimization QUBO ~~prom~~ problems"

Becomes the equivalent of finding the ground state of a Hamiltonian

The MinimumEigenOptimizer translates the quadratic program to a Hamiltonian, then calls a VQE or QAOA to compute solution.

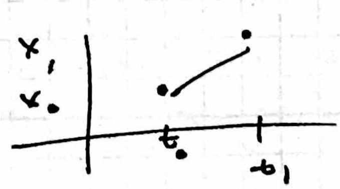
Wow. That's restrictive.

* trying to understand Qiskit_Finance. BaseDataProvider
.get_period_return_mean_vector

In code, for a given stock ~~ask~~ with values of

$$x_0, x_1, x_2, x_3 \dots x_n$$

returns are,
in code $\Rightarrow \frac{x_1}{x_0} - 1, \frac{x_2}{x_1} - 1, \frac{x_3}{x_2} - 1, \dots \frac{x_n}{x_{n-1}} - 1$



$$r_1 = \frac{x_1 - x_0}{t_1 - t_0} = x_1 - x_0$$

$$t_1 - t_0 = \Delta t = 1 \text{ day}$$

$$\% \text{ return}_1 = \frac{r_1}{x_0} = \frac{x_1}{x_0} - 1$$

μ = code returns $\langle \% r_i \rangle \rightarrow$ average of each days percentage return

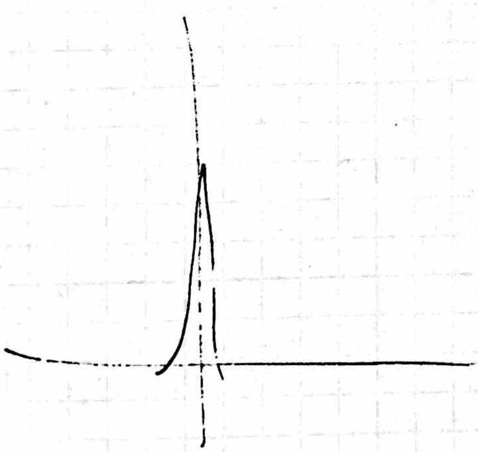
So, how is it possible that STOCK 1 ~~have~~ has a larger average ~~day~~ daily return than STOCK 0 even though it has an overall <u>negative</u> return after 30 years?

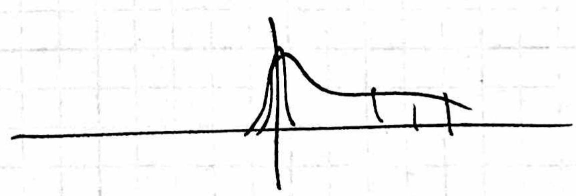
Here is the plot from the Notebook

BUT $\mu_0 = 1.59e-4$ $\mu_1 = 4.76e-4$

↓

has a large # of small negative returns & a relatively large number of positive returns





dist of μ

→ An Issue was created in Qiskit_Finance to address this concern

However, some configurations do NOT return the same answer. And when using the more verbose output from

- Challenge 1c

  Clearly trivial to set $B=3$ to increase the budget. But what needs to be changed to allow for more than one asset of the same type to be purchased??

QAOA – Quantum Approximate Optimization Algorithm

→ extends the VQE class

→ uses its own fine-tuned ansatz

arxiv: 1411.4028

QUBO → converted to Ising Model
Model

Solution using VQE

Variational quantum eigensolver

VQE object

↳ requires an ansatz (aka a trial state)

↳ and ~~an~~ a classical optimizer

→ the optimizer varies the parameters of the ansatz such that it works toward the minimum expectation value of the inputa Hamiltonian

The ansatz is a quantum circuit

TwoLocal is a factory for a particular class of quantum circuits

→ alternating rotation layers & entanglement layers

the rotation layers are single qubit gates applied on all qubits

the entanglement layer uses two-qubit gates

can be buit for an arbitrary number of qubits

For the solution here, seems that there are a large number of possible configurations