Markowitz Portfolio on DeFi Loan Platforms

July 8, 2025

Contents

1. Introduction

FL to team:

1. This notes extends the last idea (presented by TG) of forming a loan portfolio by further utilising the deposit as collateral to gain borrowing power.

2. By using the deposit as collateral, users can short/borrow other crypto on the lending platform.

3. An intricate balance must be maintained among the mean return, volatility, and the potential loss due to liquidation

4. One critical difference between the last idea and the idea in this notes is that we do not rebalance the portfolio weights continuously in time. Instead, we fix the weights at the beginning and let the portfolio value changes according the the assets returns and the enforcement of liquidation.

5. There are some advantages if we pursue in this direction

   (a) The result is comparable to any other portfolio optimisation scheme (it is a Markowitz portfolio anyway)

   (b) There is a practical relevancy to investors. Our work is beneficial to the DeFi lending platforms as we detail the scheme to form a portfolio on the lending platform. We should consider apply the AAVE research grant https://aavegrants.org/.

(c) We will use machine learning algorithms, e.g. NN, XGBoost, along the way, so we can use the buzz word "machine learning" when we market our work.

AAVE and Compound are decentralized finance (DeFi) platforms originally designed to facilitate the borrowing and lending of crypto assets. These platforms have become dominant players in the DeFi ecosystem, with total value locked (TVL) reaching over \$10 billion across both platforms, showcasing their significant influence and adoption in the market.

Although initially developed for decentralized lending, the platforms' functionalities are surprisingly well-suited for implementing sophisticated portfolio management strategies. In particular, there are two core features make these platforms appealing to users looking to build and manage diversified crypto portfolios.

Interest-bearing collateral. Assets deposited into these platforms can be used as collateral for loans while also accruing interest. This dual functionality enables an efficient capital utilization. For example, users can maintain long positions in assets they expect to appreciate, while using those same assets as collateral to borrow other assets. This borrowed capital can be redeployed for short positions or to diversify into other investments, providing an opportunity to form a wide range of portfolio strategies. In addition to the potential profit generated from the price appreciation of long-held assets, users also benefit from passive income in the form of interest, as collateralized assets accrue interest via interest-bearing tokens like `aTokens` in AAVE or `cTokens` in Compound. This combination of borrowing power and interest generation allows users to simultaneously enhance their portfolio's earning potential while maintaining exposure to the crypto assets they favor.

Unified margin assessment. Users can borrow and lend multiple assets simultaneously, whereas debt and collateral across all crypto holdings within an account are collectively assessed. This enables the diversification of portfolio in two important ways. First, it allows users to reduce the overall portfolio volatility by holding a mix of negatively correlated assets. Second, the aggregation of collateral and debt mitigates the risk of liquidation, as the platform evaluates the entire portfolio's performance rather than individual assets. This means that a decline in the value of one asset or an increase in the price of a borrowed asset is less likely to trigger a liquidation event, as other assets in the portfolio may offset the change in leverage. This feature is particularly crucial in the crypto market, where prices are notoriously volatile and fluctuations can be sudden and severe.

Nonetheless, there are notable constraints and risks associated with overcollateralization, liquidation, interest rate risk, and bad debt risk.

Overcollateralization is a fundamental requirement on DeFi lending platforms to financially bind the borrowers to repay the loan in an anonymous and non-recourse environment. Users must provide collateral that exceeds the value of the assets they wish to borrow in prior, often ranging from 120% to 150% of the loan value, depending on the collateral asset (Need some references here). Overcollateralization reduces capital efficiency for users by locking up more assets than they can borrow. Platforms generally impose a larger haircut on volatile assets, meaning that holding long positions in such assets provides less borrowing power. In the crypto market, research (Need some references here) has shown that volatility is positively correlated with price drift, often driven by herding effects. This means that more volatile assets, while potentially more profitable, offer less borrowing power, creating a trade-off between profits and borrowing power for users.

Liquidation risk is another significant concern. If a position's health factor falls below a critical threshold, the position becomes undercollateralized, and the platform initiates liquidation. During liquidation, a portion of the collateral is sold off to cover the debt, and borrowers incur liquidation penalties. In highly volatile crypto markets, where prices can fluctuate rapidly, liquidation risk becomes particularly acute. One solution to mitigate this risk is to closely monitor the position and either supply additional collateral or repay part of their debt as needed to maintain sufficient collateralization. Alternatively, users can form a diversified portfolio of collateral in advance to reduce the likelihood of liquidation. The former approach requires additional funds or the unwinding of part of the portfolio, while the latter involves constructing a portfolio that balances potential returns with reduced liquidation risk.

Interest rate risk is a critical aspect that borrowers must account for when using AAVE or Compound. Interest rates on borrowed assets are variable and fluctuate based on supply and demand within the platform. When demand for borrowing is high, interest rates can rise sharply, increasing the cost of maintaining a borrowed position. If the interest rate on borrowed assets increases beyond the returns generated by those assets, the debt can become increasingly expensive to service, eroding potential profits and putting additional pressure on users to repay or refinance their loans.

Bad debt risk refers to the possibility (to be finished)

In this paper, we propose a method for constructing a Markowitz-type crypto portfolio that leverages the functionalities of DeFi lending platforms, with a focus on carefully balancing risk and return under the constraints imposed by the platforms. Achieving this balance requires accounting for several factors intrinsic to the crypto market. First, the joint dynamics of crypto asset prices, characterized by high volatility and intricate dependency structures, must be modeled accurately. We approach this by modeling individual price dynamics using ARIMA-GARCH models and capturing the dependency structure through an appropriate copula To Team: This is to be discussed. Additionally, potential losses from liquidation events must be carefully calculated, as these can significantly erode portfolio value if not properly managed. To estimate potential liquidation losses, we employ machine learning methods—such as neural networks or XGBoost To Team: This is to be discussed—to learn from simulated portfolio returns. Finally, we propose a projected gradient ascent method to search for an optimal portfolio allocation under the constraints imposed by the platforms.

2.   Important Features

Specific features of DeFi (and potentially specific platforms):
Features of CC we want to take into account: heavy tails, volatility clustering, long-range dependence, leverage effect (some literature says it is inverse to stocks), mean-reversion.

2.1.   Determination and Accrual of Borrowing and Lending Interest

(to be filled)

2.2.   Borrowing power

(to be filled)

2.3.   Liquidation Procedure

(to be filled)

3.   The portfolio problem

3.1.   Crypto portfolio on a lending platform

Suppose we form a crypto portfolio on a DeFi lending platform that offers n kinds of asset to be borrowed and lent. We initiate the portfolio with an *initial fund* V_0 at time 0 and unwind the portfolio at time T. At every time step $t = 1, 2, ..., T$, the liquidators liquidate the portfolio if the portfolio's health factor is below one.

In the first step of forming a crypto portfolio, we allocate V_0 to the assets available to be lent out on the platform. Consider a *capital weight vector* to denote the way we allocate the initial capital

$$\mathbf{c} = [c_1 \ c_2 \ \ldots \ c_n]^\top,$$

where $c_i \geq 0$ for $i = 1, 2, \ldots, n$ is the capital we allocate to asset i and deposit to the lending platform as collateral. We restrict $\|\mathbf{c}\| = V_0$ so to allocate all the initial fund to the portfolio.

In the second step, we decide the amount of asset we borrow from the lending platform. We define a *borrowing weight vector* to capture the amount of asset we borrow from the pools

$$\mathbf{w}_0^b = [w_{1,0}^b \ w_{2,0}^b \ \ldots \ w_{n,0}^b]^\top,$$

where $w_{i,0}^b \geq 0$ for $i = 1, 2, \ldots, n$ is the amount we borrow asset i.

We restrict $c_i w_{i,0}^b = 0$ for $i = 1, 2, \ldots, n$ such that there is no long and short of the same asset at the same time. In addition, due to the over-collateralisation constraint, the maximum amount of borrowing is capped by the total borrowing power of the capital, i.e.

$$\|\mathbf{w}_0^b\| \leq \mathbf{c}^\top \mathbf{m},$$

where $\mathbf{m} = [m_1 \ m_2 \ \ldots \ m_n]^\top$ is a vector denoting the maximum LTV which m_i is maximum LTV of asset i as collateral.

Finally, we allocate all the borrowed fund back to the portfolio and describe the allocation by a *borrowed fund weight vector*

$$\mathbf{y} = [y_1 \ y_2 \ \ldots \ y_n]^\top \text{ such that } \|\mathbf{y}\| = \|\mathbf{w}_0^b\|,$$

where $y_i \geq 0$ for $i = 1, 2, \ldots, n$ is the amount of borrowed fund invested into asset i. Again, we restrict $y_i w_{i,0}^b = 0$ for $i = 1, 2, \ldots, n$, such that there is no long and short of the same asset at the same time.

The initial total amount of long position we allocate to the assets denoted by a vector

$$\mathbf{w}_0^l = [w_{1,0}^l \ w_{2,0}^l \ \ldots \ w_{n,0}^l] = \mathbf{c} + \mathbf{y}$$

where $w_{i,0}^l = c_i + y_i$ is the aggregated amount of fund invested into asset i.
At time 0, the portfolio value is

$$V_0 = \|\mathbf{w}_0^l\| - \|\mathbf{w}_0^b\|;$$

The portfolio health factor is

$$H_0 = \frac{(\mathbf{w}_0^l)^\top \mathbf{h}}{\|\mathbf{w}_0^b\|},$$

where $\mathbf{h} = [h_1 \ h_2 \ \ldots \ h_n]^\top$ is the *liquidation threshold vector* with h_i denoting the liquidation threshold of asset i when it serves as collateral.

Consider two *returns vectors*

$$\mathbf{r}_t^l = [r_{1,t}^l \ r_{2,t}^l \ \ldots \ r_{n,t}^l]^\top \text{ and } \mathbf{r}^b = [r_{1,t}^b \ r_{2,t}^b \ \ldots \ r_{n,t}^b]^\top$$

where $r_{i,t}^{l/b}$s are the log returns if the user lends out/borrows one unit of asset i at time t. To ease the complexity of the notation and computation, the returns already take into account of the borrowing and lending interest accrual. Specifically, denote the annualised borrowing rate, lending rate, and the log return over time $t-1$ and t of asset i are $\iota_{i,t}^b$, $\iota_{i,t}^l$ and $r_{i,t}$, respectively, the return of borrowing/shorting the asset is $r_{i,t}^b = -\iota_{i,t}^b - r_{i,t}$, and the return of lending/longing the asset is $r_{i,t_1}^l = \iota_{i,t}^l + r_{i,t}$. In addition, $r_{i,t}^b$s are assumed to be backed by sufficient collateral such that the returns are generated from borrowing positions that does not experience any liquidation. We tackle the loss generated from liquidation in a later section. The historical values of $r_{i,t}^{l/b}$s can be computed directly from the historical price and interest rate data provided by the platforms.

At time t, the portfolio weights change according to the return of the lending and borrowing position and liquidation procedure. For $i = 1, 2, ..., n$,

$$w_{i,t}^{l/b} = \begin{cases} w_{i,t-1}^{l/b} \exp(r_{i,t}^l) & \text{if there is no liquidation,} \\ w_{i,t}^{l'/b'} & \text{if there is at least one liquidation triggered,} \end{cases}$$

where $w_{i,t}^{l'/b'}$s are the portfolio weights remaining after the liquidation procedure and their calculations are described in Algorithm 1.

The portfolio value at time t is

$$V_t = \sum_{i=1}^n w_{i,t}^l - \sum_{i=1}^n w_{i,t}^b.$$

The portfolio health factor is at time t is

$$H_t = \frac{(\mathbf{w}_t^l)^\top \mathbf{h}}{\|\mathbf{w}_t^b\|},$$

where $\mathbf{w}_t^l = [w_{1,t}^l \ w_{2,t}^l \ \ldots \ w_{n,t}^l]^\top$ and $\mathbf{w}_t^b = [w_{1,t}^b \ w_{2,t}^b \ \ldots \ w_{n,t}^b]^\top$.

Algorithm 1: Calculation of Portfolio Weights After Liquidation. During the liquidation process, liquidators can repay debt and seize a specific borrowing-collateral pair at each liquidation call, based on their discretion. They can also choose the amount of liquidation within a defined range. On AAVE, liquidators can liquidate up to half of the debt position of an asset. In this algorithm, we assume the liquidators repay the asset with the highest value of debt and seize the most abundant collateral to maximise the profit a one liquidation call. Multiple liquidation calls can occur within a short period of time (in a single block), so the liquidation process continues until either the collateral is exhausted or the borrower's health factor is restored to one. In reality, the success of each call depends on gas fees and competition from other liquidators. However in our algorithm, we assume that liquidation calls are always successful.

3.2. Objective function

We form a portfolio that the mean return of the portfolio is maximised while keeping the variance and loss due to liquidation at balanced levels. The objective function of interest is

$$F(\mathbf{w}) = \mathbf{w}^\top \mu - \alpha_1 \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w} - \alpha_2 L(\mathbf{w}), \tag{1}$$

where

- $\mathbf{w} = \begin{bmatrix} \mathbf{c} + \mathbf{y} \\ \mathbf{w}_0^b \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0^l \\ \mathbf{w}_0^b \end{bmatrix}$

- μ and $\mathbf{\Sigma}$ are the mean vector and covariance matrix of the returns cumulated over the investment horizon $\sum_{t=0}^T \begin{bmatrix} \mathbf{r}_t^l \\ \mathbf{r}_t^b \end{bmatrix}$

- $\alpha_1, \alpha_2 \geq 0$ are penalty weights

- $L(\mathbf{w})$ is the loss due to liquidation over the investment horizon given the initial allocation of fund.

In practice, we can only estimate the objective function. The objective function estimate is

$$\hat{F}(\mathbf{w}) = \mathbf{w}^\top \hat{\mu} - \alpha_1 \mathbf{w}^\top \hat{\mathbf{\Sigma}} \mathbf{w} - \alpha_2 \hat{L}(\mathbf{w}), \tag{2}$$

where $\hat{\mu}$, $\hat{\mathbf{\Sigma}}$, and $\hat{L}(\mathbf{w})$ are the estimate counterparts of μ, $\mathbf{\Sigma}$, and $L(\mathbf{w})$ conditioned to all information available up to time 0, respectively.

The estimation of $\hat{L}(\mathbf{w})$ is particularly tricky because of the complex liquidation loss calculation (as discussed in Algorithm 1). We propose a machine learning method which is detailed in Section 5, but in the discussion of the optimisation problem and optimisation scheme, we assume $\hat{L}(\mathbf{w})$ is readily assessable.

The constrained maximisation problem is

$$\hat{\mathbf{w}}^* = \arg\max_{\mathbf{c}, \mathbf{y}, \mathbf{w}_0^b} \hat{F}(\mathbf{w}), \tag{3}$$

subjected to

- $\|\mathbf{c}\| = V_0$

- $c_i \geq 0$, $w_i^b \geq 0$, and $c_i w_i^b = 0$ for $i = 1, 2, ..., n$

- $\|\mathbf{w}_0^b\| \leq \mathbf{c}^\top \mathbf{m}$

- $y_i \geq 0$ for $i = 1, 2, ..., n$

- $\|\mathbf{y}\| = \|\mathbf{w}_0^b\|$

- $y_i w_i^b = 0$ for $i = 1, 2, ..., n$.

3.3.   Optimisation scheme

In light of the plentiful constraints, we propose a projected gradient ascent scheme to optimise the initial portfolio weights. An overview of the projected gradient ascent algorithm is detailed in Algorithm 2. The detail of the projection step is documented in Algorithm 3.

1: **Input:** Objective function $\hat{F}(\mathbf{w})$, learning rate η, initial portfolio weights $\mathbf{w}^{(0)}$, maximum iterations K, convergence threshold ϵ
2: **Initialize:** Set $k = 0$, $\mathbf{w}^{(0)}$ such that it satisfies all constraints
3: **while** $k < K$ and $\|\nabla \hat{F}(w^{(k)})\| > \epsilon$ **do**
4:     Compute gradient: $\nabla \hat{F}(\mathbf{w}^{(k)})$
5:     Update weights: $\tilde{\mathbf{w}}^{(k+1)} = \mathbf{w}^{(k)} + \eta \nabla \hat{F}(\mathbf{w}^{(k)})$
6:     Project onto feasible region:
$$\mathbf{w}^{(k+1)} = \mathrm{Proj}(\tilde{\mathbf{w}}^{(k+1)})$$

7:     Check for convergence: if $\|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| < \epsilon$, stop
8:     Set $k = k + 1$
9: **end while**
10: **Output:** Optimized portfolio weights $\mathbf{w}^{(k+1)}$

Algorithm 2: Projected Gradient Ascent for Portfolio Weight Optimization

3.4.   Poor investors' objective function

The learning of $\hat{L}$ can be very expansive. One can replace $\hat{L}$ by health factor and obtain a *poor investors' objective function*

$$\tilde{F}(\mathbf{w}) = \mathbf{w}^\top \hat{\mu} - \alpha_1 \mathbf{w}^\top \hat{\mathbf{\Sigma}} \mathbf{w} + \alpha_3 H(\mathbf{w}), \tag{4}$$

where $\alpha_3 \geq 0$ is the weight term to encourage a higher health factor, and $H(\mathbf{w}) = (\mathbf{w}^l)^\top \mathbf{h} / \|\mathbf{w}^b\|$ is the health factor.

3.5.   Wealth process dynamics

In the formulation of Aït-Sahalia, Cacho-Diaz and Hurd (2009), the asset prices follow the dynamics:

$$\frac{dS_{i,t}}{S_{i,t^-}} = (r + R_i)\, dt + \sum_{j=1}^n \sigma_{i,j}\, dW_{j,t} + J_i\, dY_t, \quad i = 1, \ldots, n$$

where $S_{i,t}$ is the price of asset i at time t, r is the risk-free rate, R_i is the risk premium, $W_{j,t}$ is a Wiener process, J_i is the jump size of asset i and Y_t is a Lévy process. Given that we use fixed weights and we do not have a riskless asset, the wealth process has from the above paper is slightly modified and follows the dynamics:

$$\frac{dV_t}{V_t} = \sum_{i=1}^n w_{i,t}^{l/b} V_t \frac{dS_{i,t}}{S_{i,t^-}}$$

$$= (w_{i,t}^{l/b} R V_t) dt + V_t w_{i,t}^{l/b} \sigma dW_t + V_t w_{i,t}^{l/b} J dY_t$$

1: **Input:** Initial portfolio value V_0, maximum LTV vector $\mathbf{m}$, capital vector $\mathbf{c}$, borrowing weights $\mathbf{w}^b$, allocation of borrowed fund $\mathbf{y}$, and the gradients $\frac{\partial \hat{F}}{\partial c_i}(c_i)$, $\frac{\partial \hat{F}}{\partial w_i^b}(w_i^b)$, and $\frac{\partial \hat{F}}{\partial y_i}(y_i)$ for $i = 1, 2, ..., n$.

2: Clip the vectors element-wise to enforce non-negativity:

$$\mathbf{c} = \mathrm{clip}(\mathbf{c}, 0, \infty), \ \ \mathbf{w}^b = \mathrm{clip}(\mathbf{w}^b, 0, \infty), \ \ \mathbf{y} = \mathrm{clip}(\mathbf{y}, 0, \infty)$$

3: Enforce no long-short positions on the same asset by taking the steeper gradient:
    a. Define two index sets

$$\mathcal{I} = \left\{ i = 1, 2, ..., n : c_i w_i^b \neq 0 \ \cap \ \frac{\partial \hat{F}}{\partial c_i}(c_i) + \frac{\partial \hat{F}}{\partial y_i}(y_i) < \frac{\partial \hat{F}}{\partial w_i^b}(w_i^b) \right\}$$

$$\mathcal{J} = \left\{ j = 1, 2, ..., n : c_j w_j^b \neq 0 \ \cap \ \frac{\partial \hat{F}}{\partial c_j}(c_j) + \frac{\partial \hat{F}}{\partial y_j}(y_j) \geq \frac{\partial \hat{F}}{\partial w_j^b}(w_j^b) \right\}$$

    b. Set $c_i = 0$ for $i \in \mathcal{I}$
    c. Set $w_j^b = 0$ for $j \in \mathcal{J}$

4: Normalize $\mathbf{c}$ to ensure all initial portfolio value are invested into lending pools

$$\mathbf{c} = \mathbf{c} \odot (V_0 / \|\mathbf{c}\|)$$

5: Enforce aggregated short position being less than the total borrowing power
6: **if** $\|\mathbf{w}^b\| > \mathbf{c}^\top \mathbf{m}$ **then**
7:     Normalize $\mathbf{w}^b$ by scaling with borrowing power

$$\mathbf{w}^b = \mathbf{w}^b \odot (\mathbf{c}^\top \mathbf{m} / \|\mathbf{w}^b\|)$$

8: **end if**
9: Enforce no long-short on the same asset for $\mathbf{y}$:
    a. Define an index set (with the updated $\mathbf{w}^b$)

$$\mathcal{K} = \left\{ k = 1, 2, ..., n : y_k w_k^b \neq 0 \ \cap \ \frac{\partial \hat{F}}{\partial c_k}(c_k) + \frac{\partial \hat{F}}{\partial y_k}(y_k) < \frac{\partial \hat{F}}{\partial w_k^b}(w_k^b) \right\}$$

    b. Set $y_k = 0$ for $k \in \mathcal{K}$

10: Normalize $\mathbf{y}$ to ensure all borrowed fund are invested into lending pools

$$\mathbf{y} = \mathbf{y} \odot (\|\mathbf{w}^b\| / \|\mathbf{y}\|)$$

11: **Output:** Updated values for $\mathbf{c}$, $\mathbf{w}^b$, and $\mathbf{y}$

Algorithm 3: Projection with No Long-Short Constraint and Normalization

4. Joint dynamics of crypto prices

We could consider the following: ARIMA-GARCH with some copula, BEKK, HMM

5. Learning scheme of liquidation loss

Neural network / XGBoost

6. Empirical results

6.1. Data

6.2. Backtesting procedure

7. Discussion

References