STAT 4830 Lightning Talk

Portfolio Refinement Through Iterative Sequential Modeling (PRISM)

Team PRISM:
Dhruv Gupta, Aiden Lee, Frank Ma,
Kelly Wang, Didrik Wiig-Andersen

Wharton
University of Pennsylvania

Our Problem

Goal:
Optimize a daily portfolio of assets to achieve high risk-adjusted returns while respecting certain constraints on . . .

leverage, *drawdown*, and *volatility*.

Specifically, our project focuses on maximizing a combination of financial risk metrics and ratios while limiting maximum drawdown over a given historical period.

Why This Matters

- Traditional mean-variance optimization oversimplifies risk by only using variance.
- Real-world portfolios must manage multiple risk dimensions (drawdown, volatility) and practical constraints (leverage, short selling) at the same time.
- Variability in the real-world and extreme events (like COVID) don't follow a nice and familiar distribution, so due to their unpredictability, traditional trading strategies have different tolerance levels to these unexpected events.
- By addressing these complexities, we aim to create a more realistic and robust decision-making strategy for creating a dynamic portfolio.

Success Metric

What the fitness score consist of...

- Sortino Ratio:

$$\frac{\mathbb{E}[R_P - R_f]}{\sigma_d}$$

  - R_P = Return of our portfolio
  - R_f = Risk-free rate
  - σ_d = Standard deviation of negative returns (i.e., downside deviation for returns below a minimum acceptable return).

- Risk Metric - Controlling maximum drawdown, where

$$MDD(P) = \frac{\text{90-day min. portfolio value}}{\text{90-day max. portfolio value}} - 1 \leq 0$$

- Portfolio Constraints - we ask the model to minimize transaction costs, minimize the change in our stock positions between days, and try to maintain stable returns over time

Technical Approach

Mathematical Formulation:

Let $\mathbf{w} = (w_i)$ denote the vector of weights for asset i in the portfolio. We define the portfolio return as

$$R_P = \sum_i w_i \cdot R_i$$

Our model's goal is to maximize our defined "fitness score" by learning our weights in vector $\mathbf{w}$. Given this goal, our objective function is

$$\max_{\mathbf{w}} \quad \alpha \cdot \frac{\mathbb{E}[R_P - R_f]}{\sigma_d} + \beta \cdot (MDD(P)) - \lambda \sum_i |w_i - w_i^{\text{prev}}|$$

Objective Function

$$\max_{\mathbf{w}} \quad \alpha \cdot \frac{\mathbb{E}[R_P - R_f]}{\sigma_d} + \beta \cdot (MDD(P)) - \lambda \sum_i |w_i - w_i^{\text{prev}}|$$

Definitions:

- R_f is the risk-free rate s.t. $\frac{\mathbb{E}[R_p - R_f]}{\sigma_d}$ is the Sortino ratio.
- α scales the impact of the Sortino ratio.
- β scales the impact of the drawdown term.
- λ is the transaction cost penalty, which penalizes large changes in portfolio weights between periods.
- w_i^{prev} denotes the weight of asset i in the previous period.

Note: Originally used σ_p, the portfolio's standard deviation, because we were using the Sharpe Ratio

Problem Formulation

Training & Inferencing

- We take a universe of equities, get data on them between 2022 and current day, and train an online gradient descent model
- We take 2022-2024 to be "training" (i.e. we allow it adjust odds and do not evaluate over this period)
- We compute portfolio returns, volatility, and drawdown within the OGD, taking a step for every market day

Validation

- We also calculate cumulative portfolio returns for each equity over the time period of 2024-present day
- We compare to our most effective models (found using several different hyperparameter tests)

Results

Evidence of Working Implementation

- Original Test: A small 7-asset dataset was loaded into our PyTorch pipeline.
  - Companies:
    Tesla, Google, Microsoft, Amazon, Apple, Meta, NVIDIA
- Hyperparameters: We experimented with random values and chose the set that resulted in the best model specifications.
- Updated Test: We increased universe of stocks to 109 different tickers.
- We also want to penalize entropy (i.e. encourage sparsity).

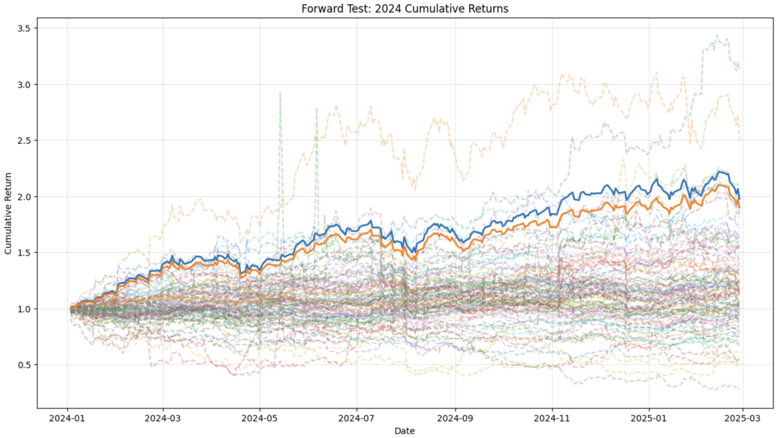

Figure 1: Intermediate Results - 2024 Cumulative Returns

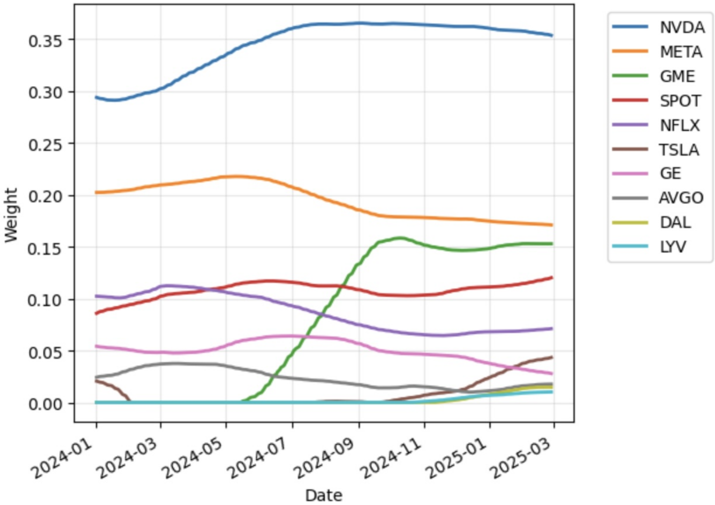

Figure 2: Intermediate Results - Cumulative Returns of our algorithm.

Next Steps

1. **Expand Data Universe**
   - In addition to our current 109 stocks; add in crypto and ETFs to expand our exposure to different asset classes.
2. **Additional Risk Measures**
   - Perform a literature review to find additional risk measures to incorporate into our model.
     - **Progress so far:** We have collected a corpus of 173 articles by applying three queries to EconLit, Web of Science, and Scopus.
3. **Validation**
   - Acquire a longer historical window and consider testing our model against time periods that experienced sudden market shocks due to extreme events.
   - Test our model with real-time data implementation.
4. **Refine Constraints**
   - Enforce a leverage limit of 1.5 and allow short selling up to 30% of the portfolio.