STAT 4830
Final Presentation

Portfolio Refinement Through Iterative Sequential Modeling (PRISM)

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

The Stock Market Problem

Uncertainty in the Market

[Picture from WSJ]

Motivation

Can we develop an algorithm that is robust to market shocks?

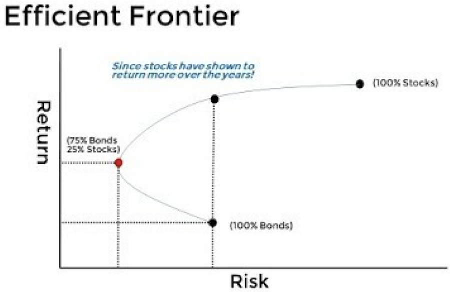
Our Goal: Optimize a daily portfolio of assets to maximize risk-adjusted returns while incorporating penalties that limit:

- drawdown (protecting against significant losses)
- turnover (maintaining portfolio stability)
- concentration risk (ensuring proper diversification)

Measurable Outcome:
Generating returns superior to a given baseline ("safe") portfolio.

Mean-Variance Optimization (Markowitz, 1952)

- Balance expected return against portfolio variance.
- Core insight - combining assets with imperfect correlations reduces risk and can rise expected return.
- Led to the discovery of the efficient frontier.



Sharpe Ratio (Sharpe, 1966)

$$SR = \left(\frac{R_t - r_{f,t}}{\sigma_t} \right)$$

- Shows that maximizing the Sharpe ratio selects the "tangency portfolio" – the point on the efficient frontier with the highest risk-adjusted return.
- Employed as one of <u>many</u> inputs in real-world portfolio construction.

Conditional Value-at-Risk (CVaR): "Expected Shortfall"

- Measures the average loss in the worst $\alpha\%$ of outcomes (i.e. "in the worst 5% of days, how much do I lose on average?")
- **Rockafellar and Uryasev (2000):** developed an approach to maximize Sharpe subject to $CVaR \leq X$.
- Widely adopted after the 2008 crisis to build more shock-resilient portfolios.

Database	Query
Web of Science	("stock" OR "equity") AND ("portfolio optimization" OR "asset allocation") AND ("risk measures" OR "risk metrics") AND ("drawdown" OR "value-at-risk" OR "conditional value-at-risk" OR liquidity OR skewness OR kurtosis) AND (optimization OR "gradient descent" OR "adam" OR "rmsprop")
Scopus	TITLE-ABS-KEY("portfolio optimization" OR "asset allocation") AND ("risk measures" OR "risk metrics") AND ("drawdown" OR "value-at-risk" OR "conditional value-at-risk" OR liquidity OR skewness OR kurtosis) AND ("stock" OR "equity")
EconLit	("portfolio optimization" OR "portfolio selection") AND ("risk measures" OR "risk metrics" OR "alternative risk premia") AND ("drawdown" OR "value-at-risk" OR "conditional value-at-risk" OR "tail risk" OR "drawdown range" OR "short selling") AND ("liquidity" OR "factor-based" OR "multi-factor" OR "factor selection" OR "factor exposure")

- 177 articles across Web of Science, Scopus and EconLit.
- **Four Key Metrics:** Sortino Ratio, Maximum Drawdown, Turnover, and Concentration Penalty.

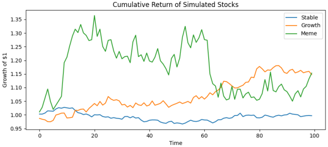
Objective Function:

$$\max_{\mathbf{w}_t} \quad \alpha_1 \cdot \mathsf{Sortino}_t(\mathbf{R}_p, \mathbf{r}_f) - \alpha_2 \cdot \mathsf{MaxDD}_t(\mathbf{R}_p) - \alpha_3 \cdot \mathsf{Turnover}(\mathbf{w}_t, \mathbf{w}_{t-1}) - \alpha_4 \cdot \mathsf{CP}(\mathbf{w}_t)$$

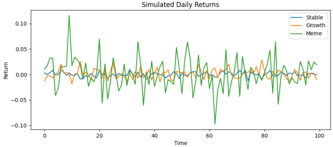
Constraints:

$$\sum_{i=1}^{N} w_{t,i} = 1 \quad \text{where } w_{t,i} \geq 0 \, \forall t, i$$

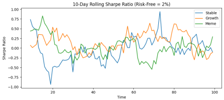
Visualizing Risk and Return Metrics



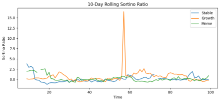
Cumulative Returns



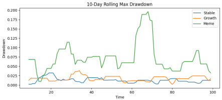
Daily Returns



Sharpe Ratio



Sortino Ratio



Max Drawdown

Variable Definitions

Weighted Portfolio's Returns:

$$R_p = \sum_{i=1}^{n} w_{i,t} \cdot R_{i,t}$$

$n \in \mathbb{N}$	Number of assets held in the portfolio
$w_{i,t} \in \mathbb{R}$	Weight for asset i at time t
$R_{i,t} \in \mathbb{R}$	Asset i's return at time t
$r_{f,t} \in \mathbb{R}$	Risk-free rate at time t
$\alpha_j \in \mathbb{R}_+$	Objective weights, $j \in \{1, 2, 3, 4\}$
$m \in \mathbb{R}_+$	Window size for historical calculation
$\varepsilon \in \mathbb{R}_+$	Small constant for numerical stability

$$\text{Sortino}_t(\mathbf{R}_p, \mathbf{r}_f) = \frac{\mathbb{E}[\mathbf{R}_p - \mathbf{r}_f]}{\sigma_{\text{downside}} + \varepsilon} \tag{1}$$

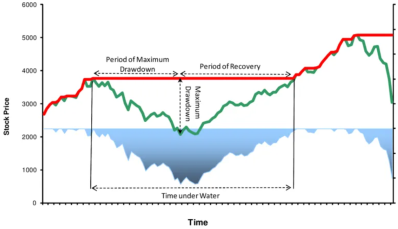
$$\text{where} \quad \sigma_{\text{downside}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \min(R_p - r_{f,t}, \text{NA})^2} \tag{2}$$

(Calculated based on data from the past m days, i.e. $\mathbf{R}_p$, $\mathbf{r}_f \in \mathbb{R}^m$.)

Maximum Drawdown
Obj. Func. (2nd Term)

$$\mathrm{MaxDD}_t(\mathbf{R}_p) = \frac{\max_{T\in[t-m,t]} \mathrm{CR}_T - \mathrm{CR}_t}{\max_{T\in[t-m,t]} \mathrm{CR}_T + \varepsilon} \tag{3}$$

$$\text{where} \quad \mathrm{CR}_t = \prod_{i=t-m}^{t} (1 + R_i) \tag{4}$$



(Calculated based on data from the past m days, i.e. $\mathbf{R}_p \in \mathbb{R}^m$.)

Turnover Term

Obj. Func. (3rd Term)

Proxy to represent transaction costs (for $\mathbf{w}_t \in \mathbb{R}^n$):

$$\text{Turnover}(\mathbf{w}_t, \mathbf{w}_{t-1}) = \frac{1}{2} \sum_{i=1}^{N} |w_{i,t} - w_{i,t-1}| \tag{5}$$

Concentration Penalty

Obj. Func. (4th Term)

$$\mathsf{CP}(\mathbf{w}_t) = \max(\mathsf{ENP}_{\min} - \mathsf{ENP}(\mathbf{w}_t), 0) + \max(\mathsf{ENP}(\mathbf{w}_t) - \mathsf{ENP}_{\max}, 0) \tag{6}$$

Effective Number of Positions (for $\mathbf{w}_t \in \mathbb{R}^n$):

$$\mathsf{ENP}(\mathbf{w}_t) = \frac{1}{\mathsf{HHI}(\mathbf{w}_t) + \varepsilon} \tag{7}$$

$$\mathsf{HHI}(\mathbf{w}_t) = \sum_{i=1}^{N} (w_{i,t})^2 \tag{8}$$

- **Online Gradient Descent (OGD)**
  - Sequential optimization adapting to changing market conditions
  - No need to retrain on historical data - continuous learning
  - Differentiable objective function with automatic gradient computation
- **Connection to Course Concepts**
  - Gradient-based optimization with constraints
  - Multi-objective function balancing competing financial goals
  - Online learning for sequential decision making

- **Critical Hyperparameters**
  - Objective weights (α_i) - balance risk/return tradeoff
  - Window size - determines optimization lookback period
  - Learning rate - controls adaptation speed
  - ENP constraints - enforce diversification bounds
- **Hyperparameter Exploration**
  - Window sizes: 5, 21, 63, 252 days (1wk, 1mo, 3mo, 1yr)
  - Objective weights: grid search prioritizing Sortino and drawdown control
  - ENP constraints: calibrated to allow sector concentration while preventing single-stock dominance
  - Learning rates: tested 0.1-1.0 to balance stability and responsiveness

Implementation

Key Choices

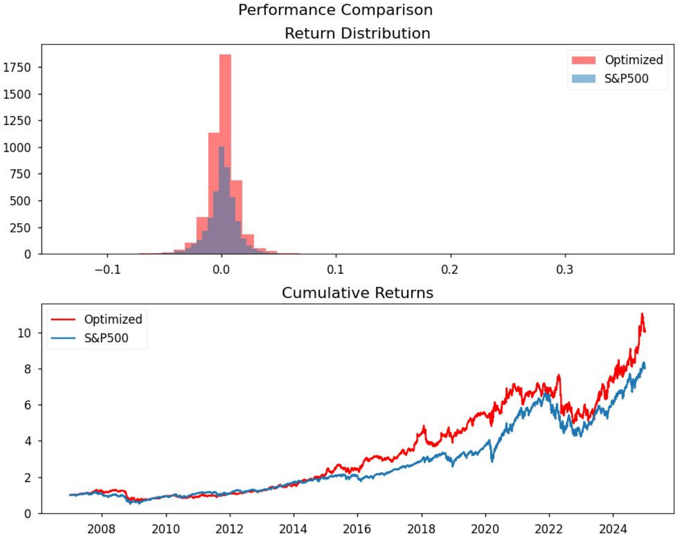
- **PyTorch Framework**
  - Automatic differentiation for gradient computation
  - Built-in optimizers with configurable learning rates
- **Implementation Details**
  - Softmax normalization to enforce $\sum_i w_i = 1$ constraint
  - Rolling window implementation for adaptive optimization
  - Tried out an SQL query system - Frank

- **Computational Challenges**
  - Scalability with large asset universes (> 500 stocks)
  - Memory constraints when tracking long optimization histories
- **Adaptations & Solutions**
  - Adjustable objective weights to control balance between goals
  - Portfolio concentration constraints to manage diversification
  - Epsilon factors to prevent division by zero in metric calculations

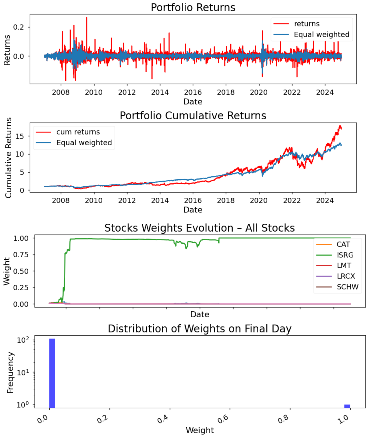
Presenting our key quantitative results:

https://droov-opt.hf.space/

Final Output
Graphs



Intermediate Results

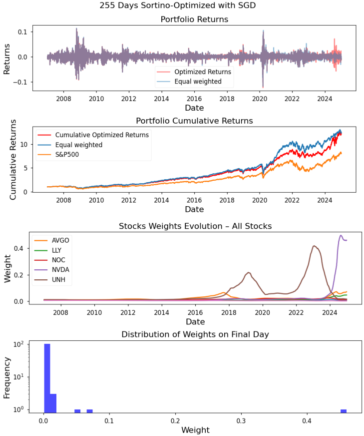


Observations

- Portfolio outperforms equal-weighted benchmark, especially post-2020.
- Optimization leads to highly concentrated positions (e.g., single stock weight near 100%).
- Weight distribution on final day reveals near-binary allocation: almost all in one stock.
- We were actually more volatile than market –¿ did we really optimize sharpe?

Intermediate Results - Optimizing Sortino Only

Graphs

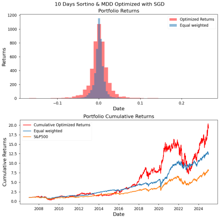


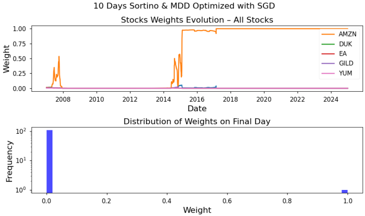
Observations

- We are overperforming equal-weighting, but outperforming S&P.
- Optimization leads to exposure across several stocks, but still relatively concentrated (e.g., NOC, UNH).
- Unclear whether we are more or less volatile than market.

Intermediate Results - Optimizing Sortino and MDD

Graphs



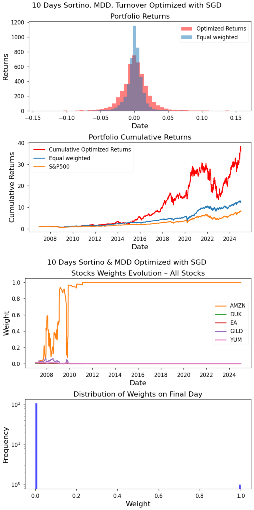


Observations

- 10 days was way too volatile, even with max drawdown penalty
- Portfolio cumulative returns outperform S&P500 and equal-weighted benchmarks, but at what cost...
- Allocation is again highly concentrated — majority in AMZN for extended periods.
- Weight distribution still shows heavy tail: one dominant stock, many with near-zero allocation.

Intermediate Results - Optimizing Sortino, MDD, Turnover

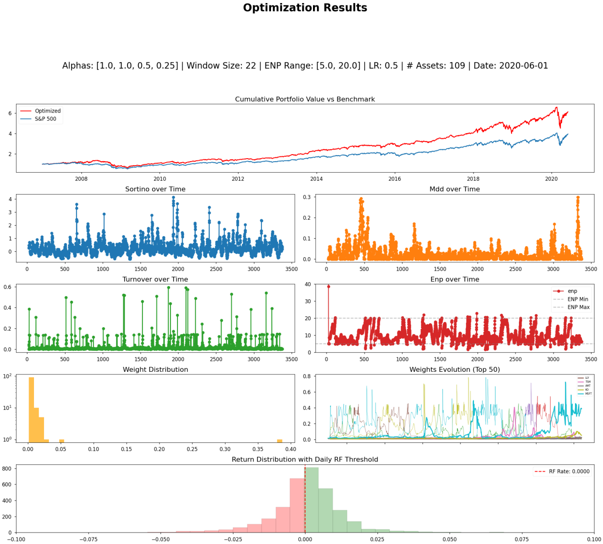
Graphs



Observations

- Portfolio maintains strong outperformance vs. benchmarks.
- Turnover constraint reduces trading frequency but maintains exposure to dominant stocks.
- Concentration issue persists

Intermediate Results
Full Objective Function



Intermediate Results - Hyperparameters

Graphs

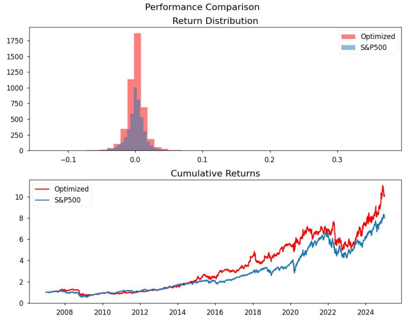
```
 Optimal Hyperparameters:
Window Size      : 181
Learning Rate    : 0.8655
Alpha Weights    : [1.0, 6.080, 0.935, 5.154]
Best Sortino     : 0.0958
```

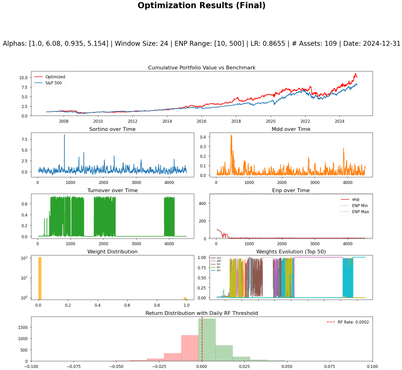
Observations

- 5 tunable hyperparamters
- used 2007-2010 for tuning, and subsequent years for evaluation
- used Gaussian Process Minimize from scikit-optimization
- supposedly optimal hyperparameters looked a little suspicious

Final Results

Graphs





- **Comparing our results to baseline methods:**
  - Improvements made moving from Sharpe $\rightarrow$ Sortino $\rightarrow$ our final objective function.
  - We also compare ourselves to the S&P500 and an equally weighted portfolio of our universe of stocks.
  - We iteratively made improvements to baseline based on qualitative metrics, but we did not actually have a systematic way to compare different optimizations
- **Interpretation:**
  The supposedly optimized strategy had an all-history Sharpe of 0.0363, max drawdown of 0.606, and annualized excess return of 7.44% annually
  Although the results didn't look mindblowing, we are earning a decent return not attributable to market!

Reflections

- **Primary Challenge**
  - Deciding and optimizing objective weights
  - Deciding on benchmarks and criteria for evaluating our strategy
- **Expectations**
  - Getting access to financial data was surprisingly easy to integrate into our code.
  - Choosing our universe of stocks was difficult for implementation and justification reasons.
  - Migration to PyTorch took more time than expected.
- **Evolution**
  - Lit review - simple 7-stock model with sortino, MDD using numpy - stock universe expansion - pytorch migration, additional objectives like turnover and ENP
- **AI Assistance**
  - Claude helped write the first versions of our implementations.
  - ChatGPT helped refine our project outline and trajectory for improvements