STAT 4830
Slides Draft (Week 4)

Portfolio Refinement Through Iterative Stochastic Modeling (PRISM)

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

Problem Statement

Goal: Optimize a daily portfolio of assets to achieve high risk-adjusted returns while respecting constraints on leverage, drawdown, and volatility.

Why This Matters:

- Traditional mean-variance optimization oversimplifies risk.
- Real-world portfolios require multi-dimensional risk management.
- Market variability necessitates robust strategies.

Success Metric: Fitness Score

Components:

- Sharpe Ratio: $\frac{E[R_p - R_f]}{\sigma_p}$
- Risk Metrics: Maximum Drawdown reduction, controlled volatility.
- Factor Exposure: Sensitivity to risk factors.
- Portfolio Constraints: Minimize transaction costs and maintain stable returns.

Constraints & Data Requirements

Constraints:

- Leverage: Within a maximum range (e.g., 1.5–2.0).
- Drawdown: Must remain below 20%.
- Short-selling: Limited exposure.

Data Requirements:

- Historical daily/weekly returns (3–5 years).
- Data sourced from yfinance API.

Mathematical Formulation

Objective Function:

$$\max \quad \alpha \cdot \frac{E[R_P] - R_f}{\sigma_p} + \beta \cdot (-MDD(p)) - \lambda \sum_i |w_i - w_i^{\text{prev}}|$$

Definitions:

- R_f is the risk-free rate, so that $\frac{E[R_p - R_f]}{\sigma_p}$ represents the Sharpe ratio.
- α scales the impact of the Sharpe 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.

Constraints

- If no leverage: $\sum_i w_i = 1$ and $w_i \geq 0 \quad \forall i$.
- If allowing leverage: $\sum_i |w_i| \leq L_{\max}$, where $L_{\max}$ is the maximum allowable leverage.
- Short-selling limit: $w_i \geq -\delta \quad \forall i$ where $\delta \geq 0$ specifies the maximum allowed short position per asset.

Optimization Strategy

PyTorch-Based Approach:

- Represent weights as tensors.
- Compute portfolio returns, volatility, and drawdown within PyTorch.
- Use gradient-based methods (Adam, LBFGS) for optimization.

Initial Results

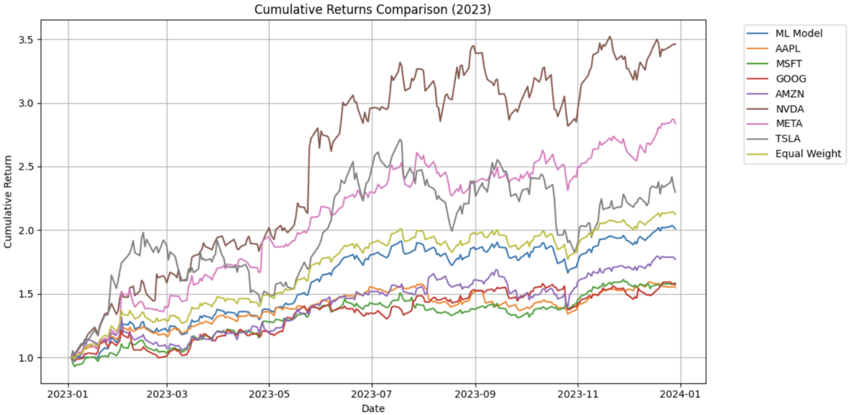
Initial Test:
Portfolio consisting of Tesla, Google, Microsoft, Amazon, Apple, Meta, NVIDIA. Our output portfolio beats an equally weighted portfolio. Trained on daily data from the year of 2023.

Preliminary Performance:

- Initial Sharpe Ratio: 1.05 (test dataset)
- Drawdown: $\approx 25\%$ peak-to-trough
- Validated portfolio return calculations

Challenges:

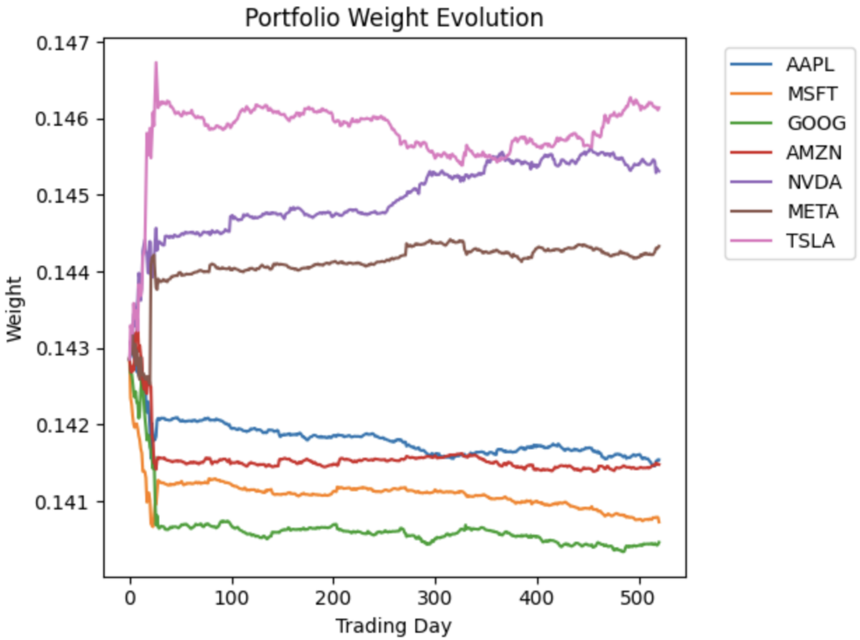
- Handling short-selling constraints in PyTorch.
- Computational complexity in large datasets.



Detailed Performance Metrics

	ML Model	AAPL	MSFT	GOOG	AMZN	NVDA	META	TSLA	Equal Weight
Total Return	101.03	54.80	58.35	57.11	77.04	246.10	183.76	129.86	112.39
Annual Vol	26.54	19.92	25.07	30.55	32.96	48.36	39.75	52.54	26.11
Sharpe Ratio	2.64	2.11	1.81	1.51	1.79	2.74	2.74	1.78	2.89
Max Drawdown	-13.05	-14.93	-12.99	-17.88	-19.64	-18.29	-12.97	-32.72	-11.93

Figure: Our model is denoted *ML Model*


Portfolio Weight Evolution
0.147
0.146
0.145
0.144
0.143
0.142
0.141
Weight
0
100
200
300
400
500
Trading Day
AAPL
MSFT
GOOG
AMZN
NVDA
META
TSLA

Next Steps

Planned Improvements:

- Expand asset universe (10–20 stocks).
- Expand training data's time horizon.
- Integrate advanced risk measures (CVaR).
- Implement rolling rebalancing strategies.
- Introduce transaction cost modeling.

Self-Critique

Strengths:

- Effective implementation of multi-objective optimization.
- Demonstrated initial backtesting results.

Areas for Improvement:

- Optimize objective function weights.
- Diversify asset selection.
- Test impact of different window sizes.

Conclusion

Key Takeaways:

- Multi-objective portfolio optimization is complex but achievable.
- PyTorch provides flexibility but requires careful constraint handling.
- Future work will refine constraints, expand datasets, and validate models.