STAT 4830
Slides Draft 1

Kelly Wang, Aiden Lee, Dhruv Gupta, Frank Ma, 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

Basic Test: 7 stocks: Tesla, Google, Microsoft, Amazon, Apple, Meta, NVIDIA. 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.

Next Steps

Planned Improvements:

- Expand asset universe (10–20 stocks).
- 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.