STAT 4830 Week 3 Deliverable (Draft)
Portfolio Maximization

Problem statement (concise). This project optimizes portfolio weights $w_t \in \mathbb{R}^N$ at each rebalancing time t to construct a portfolio that outperforms standard benchmarks out of sample. Given information available at time t and next-period asset returns r_{t+1}, the core decision problem selects weights that trade off expected return and risk, for example by solving

$$\max_{w_t} \ \mathbb{E}_t[w_t^\top r_{t+1}] - \gamma \,\mathrm{Risk}(w_t), \qquad \text{s.t. } \sum_i w_{t,i} = 1 \text{ and feasible weight bounds,}$$

where expected returns are initially estimated using simple predictive signals (e.g., historical averages or lag-based models) so that the early emphasis remains on numerical optimization and constraint handling rather than complex signal engineering. The problem matters because portfolio construction is the mechanism through which noisy predictive information is translated into economic outcomes, and many forecasting approaches fail to beat benchmarks once estimation error, risk control, and constraints are imposed; even modest improvements in robust portfolio optimization can yield meaningful gains in risk-adjusted performance. Success will be measured by strictly out-of-sample evaluation against clear baselines (e.g., equal-weighted and market benchmarks), using primary metrics such as Sharpe ratio and cumulative return and secondary diagnostics such as volatility and maximum drawdown to ensure that outperformance is not driven by excessive risk-taking. The project faces practical constraints: experiments must be computationally feasible on a single CPU machine, validation must respect temporal ordering to avoid look-ahead bias, and feasible portfolios must satisfy realistic investment constraints (full investment and position limits), all while controlling overfitting in a low signal-to-noise setting through regularization and conservative model selection. Required data consist of time-series returns for a specified universe of tradable assets at a fixed horizon (e.g., daily or monthly), together with any information used to estimate expectations and risk using only data available up to time t; all features and labels must be causally aligned. Key risks include overfitting and leakage, unstable or extreme optimized weights, excessive turnover in the absence of appropriate penalties, and the possibility that apparent statistical improvements fail to survive realistic frictions such as transaction costs, drawdown constraints, or performance instability across time.

Technical Approach

Technical approach (concise). The project is formulated as a joint *asset selection and portfolio allocation* problem solved at each rebalancing time t. Given a universe of N candidate assets, we estimate conditional expected returns $\mu_t \approx \mathbb{E}_t[r_{t+1}] \in \mathbb{R}^N$ and a conditional covariance matrix $\Sigma_t \in \mathbb{R}^{N\times N}$ using only information available up to time t. The ideal formulation selects both a subset of assets and their portfolio weights by maximizing a risk-adjusted objective subject to realistic constraints, including full investment, long-only positions, turnover control, and a limit on the number of holdings. While this can be expressed as a mixed-integer optimization problem with explicit selection variables, directly solving such problems at each rebalancing step is computationally expensive. Accordingly, the initial implementation adopts a principled convex relaxation in which asset selection is induced via sparsity-promoting penalties on portfolio weights, yielding an objective of the form

$$\max_{w_t} \ \mu_t^\top w_t - \gamma \, w_t^\top \Sigma_t w_t - \lambda \|w_t\|_1 - \kappa \|w_t - w_{t-1}\|_1, \quad \text{s.t.} \ \sum_i w_{t,i} = 1, \ w_{t,i} \geq 0,$$

which remains tractable when Σ_t is positive semidefinite. Portfolio weights are optimized numerically in PyTorch by parameterizing feasible weights through differentiable transformations and applying gradient-based methods (initially simple gradient descent, with momentum and adaptive variants explored later). Validation is conducted using strictly time-ordered and walk-forward backtests, with portfolio performance evaluated out of sample against standard benchmarks using metrics such as Sharpe ratio, cumulative return, volatility, drawdown, and turnover. Sanity checks verify constraint satisfaction, reproducibility, and correct causal data usage, while synthetic test cases are used to confirm numerical correctness. All Week 3 experiments are constrained to be CPU-feasible on a single machine, with modest asset universes and targeted hyperparameter sweeps, prioritizing a robust, auditable optimization pipeline before scaling to larger universes or more complex selection mechanisms.

Initial Results and Next Steps

Initial results (concise). A complete end-to-end implementation of the portfolio optimization pipeline has been developed and validated on a small, liquid asset universe. The code successfully constructs time-aligned estimates of expected returns and covariances, solves the relaxed portfolio optimization problem at each rebalancing date, and produces a sequence of out-of-sample portfolio returns. As evidence of correctness, the optimization objective decreases consistently during training phases, constraints (full investment and nonnegativity) are satisfied numerically to tolerance, and results are reproducible under fixed random seeds. Basic performance metrics include cumulative return and Sharpe ratio computed over a held-out backtest period, alongside volatility, maximum drawdown, and turnover. In simple test cases with synthetic data (e.g., diagonal covariance matrices and known expected returns), the optimizer recovers the theoretically optimal allocations, providing further validation of the implementation. Current limitations are substantial but intentional: the asset universe is small, predictive signals are simple, transaction costs are approximated via turnover penalties, and overall performance improvements over benchmarks are modest. Resource usage is light, with all experiments running on CPU in seconds per backtest and minimal memory overhead. An unexpected challenge has been the sensitivity of optimized weights to hyperparameter choices and feature scaling, underscoring the importance of regularization and numerical stability even in low-dimensional settings.

Next steps (concise). Immediate improvements will focus on strengthening robustness and expanding the scope of the optimization framework. On the technical side, the next iteration will refine covariance estimation (e.g., shrinkage methods) and systematically tune risk-aversion, sparsity, and turnover parameters to reduce sensitivity and improve stability. A key challenge to address is balancing sparsity and turnover penalties so that asset selection is meaningful without inducing excessive trading. Questions for course staff include best practices for time-series cross-validation in portfolio settings and guidance on diagnosing optimizer instability. Alternative approaches under consideration include explicit cardinality constraints via approximate mixed-integer methods, alternative risk measures such as downside risk or CVaR, and objectives tied more directly to economic performance (e.g., maximizing out-of-sample Sharpe ratio). The main lesson so far is that portfolio performance depends as much on optimization design, constraints, and validation discipline as on predictive signal quality; even simple signals can yield materially different outcomes depending on how the optimization problem is formulated and solved.