Dynamic Portfolio Optimization Using Evolution Strategy

Kexin Yu (kexinyu@stanford.edu), Charles Xu (chx@google.com)

Motivation

- The classic **Markowitz** model that optimizes for the **Sharpe ratio** has proven to be suboptimal.
  - Summary is used directly as prediction.
  - **Variance is not a good risk measurement** since it penalizes positive shocks and says little about tail risks
- Other risk measurements such as Value-at-Risk and Expected Shortfall introduce **non-linear, non-convex risk constraints** and render the mean-variance approach less applicable
- Time series such as security prices often **violate** the assumption of **independent sampling** and **homoscedasticity**

We thus devise an **Evolution Strategy (ES)** for the multi-period portfolio optimization, which **dynamically** adjusts the asset weights in hope for significantly higher returns.

Problem Definition

M	Number of securities in the portfolio
$x = (x_1, \ldots, x_M)^T$	Weight vector whose *i*-th element is the weight allocated to asset *i*
$\xi = (\xi_1, \ldots, \xi_M)^T$	Price vector whose i-th element is the price of asset *i*
$x^{(t)}$ $\xi^{(t)}$	Observations at time *t*
$W^{(t)} = {x^{(t)}}^T \xi^{(t)}$	The portfolio value at time *t*

Objective: maximize $W^{(T)}$ by adjusting $\{x^{(t)} : T_0 \leq t < T\}$ given only $\{\xi^{(t)} : t \leq T_0\}$

Constraints:

- Equal Wealth across Rebalancing

  $${x^{(t-1)}}^T \xi^{(t)} - C(x^{(t-1)}, x^{(t)}, \xi) = {x^{(t)}}^T \xi^{(t)} \; \forall t$$
- No Short Position

  $$x_i^{(t)} \geq 0 \; \forall i, t$$
- Value at Risk

  $$VaR_\alpha \geq K \qquad \Pr(W^{(T)} < VaR_\alpha) = \alpha$$

  There is at most an α chance that the portfolio value at time *T* will end up less than *K*.
- Expected Shortfall

  $$ES_\alpha \geq S \qquad ES_\alpha = \mathbb{E}[W^{(T)} | W^{(T)} < VaR_\alpha]$$

  It shows the magnitude of tail risk. Given VaR_α has been violated, the expected value of portfolio at time *T* should be at least *S*.

Assumption: There are 252 trading days annually and 21 trading days monthly.

Data: We have built and open-sourced a tool that sweeps across Nasdaq and NYSE to fetch for each ticker its entire historical daily prices and record to durable storage. It coordinates a pool of workers with remote procedure calls (RPCs) using gRPC framework and protocol buffer as IDL (interface description language) and serialization protocol.

Models

Unlike Markowitz approach that makes decision only for a single upcoming period, we aim at more flexible investment plans, in which the portfolio is rebalanced at different time steps. A scenario tree is used to model the randomness of the time paths of asset prices based on historical data. Optimized with ES, every single path in the tree suggests a unique sequence of portfolios that meets various financial goals.

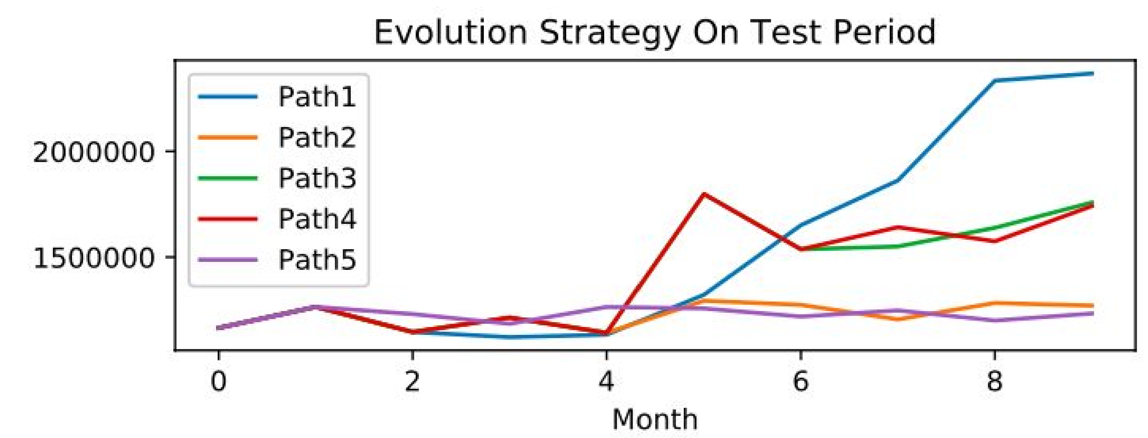
Tree Structure	An **ordered, directed** scenario tree that approximate the evolution of a portfolio over multiple time periods. • Each **level** of the tree associates with one time step of the holding period. • Each **node** defines a **strategy** at certain time with a specific portfolio weight vector • The **root** node starts with an initial capital. • **Leaf** nodes define final cumulative wealth. • The **binary** structure of the tree makes it easy to backtrack the path leading to each leaf node.
Mutation	Perturb weight matrix with **random Gaussian noise** subject to the **Equal Wealth** constraint and **No Short Position** constraint.
Selection	**(μ ; λ)** - selection scheme. In each generation, **μ** parents breed **λ** offspring individuals and the fittest are used as parents for the next generation. • The scheme is good at **leaving local optimal.** • **Fitness** is measured against the **Value-at-Risk** constraint and the **cumulative wealth.**
Recombination	**Global, intermediate** recombination scheme. The parental state for the next generation is updated by **mean value calculation** over the best **μ** of the **λ** offspring individuals.
Self-adaptation	**Covariance Matrix Self-Adaptation** • The covariance matrix is initialized with an identity matrix, updated at each generation and influences future mutation direction. • Mutation strength also evolves over time. • Each offspring individual has its own mutation strength within the same generation. • Able to follow the optimum and learn efficiently.
Transaction Costs	Accommodate **fixed, buy and sell** transaction costs in the mutation process to ensure the admissibility of proposed strategies.

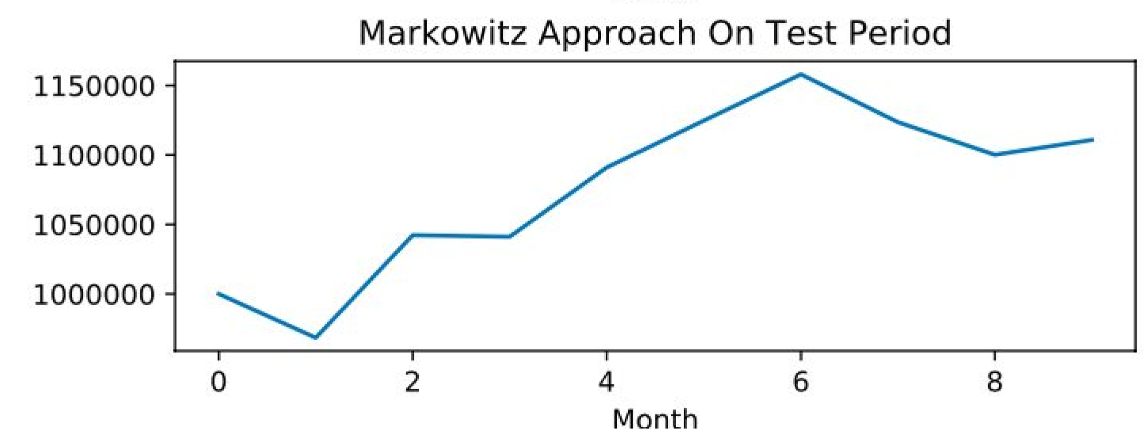
Results

① PROFITABILITY OF ES

We learned a 10-level scenario tree to model the asset price change over 10 consecutive months, randomly sampled paths (possible trading strategies) and computed the final wealth they obtain on the test period that immediately follows the training period. With optimized, evolving weights at each time step, the ES model proves to give more profitable plan than the stationary approach.

Dynamic ES achieves higher cumulative wealth than stationary Markowitz.

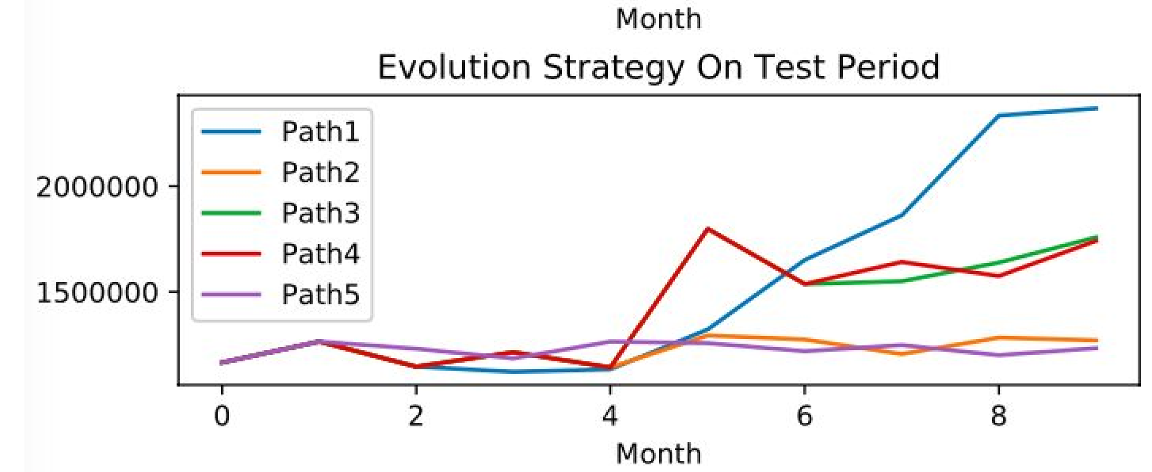




② PREDICTIVE CAPABILITY OF ES

To measure the applicability of trading signals learned from one period to another, we compared the cumulative wealth achieved by the same path at each time step over training/test period and observed similar trends. This also suggests a sliding window strategy.

A particular strategy exhibits similar performance over training/test period.



Discussion

① REBALANCING CRITERION

We backtracked the paths leading to randomly sampled leaf nodes to understand how ES rebalances the portfolio. It turned out the weight change heavily depends on the speed of price change. ES tends to assign lower weight to an asset with a decreasing rate of change in price.

② PATH DIVERSITY

With the tree structure, ES generates a variety of possible investment plans. Such a property can be exploited to fulfill different financial priorities. For instance, a short-term goal would favor a path with profitable inner nodes, while a long-term goal would pursue a path with high stability.

③ COMPLEXITY OF COSTS

With cost complexities, ES behaved more conservatively with a lower cumulative wealth. We realized the algorithm which repairs the mutation direction under the cost constraint converged slowly, since it simply returned the parent weight as a work-around and the portfolio changed minimally over the time. From a different perspective, frequent transactions sacrifice part of wealth and are less desirable in the cost model.

Future

① COMPUTATIONAL BOTTLENECK

With giant matrices, we found out the workloads were memory bound. We will explore alternative approaches to eigenvalue decomposition of the covariance matrix.

② FLEXIBLE TREE STRUCTURE

Once we are able to compute more efficiently, we will consider more levels and nodes in the scenario tree. Besides, we will explore the optimal rebalancing period by associating each level with quarterly/yearly prices.

③ BETTER EVOLUTION STRATEGY

We will evaluate different recombination schemes that may outperform the current mean value approach.

References

T. B. Hans-Georg Beyer, Steffen Finck, "Evolution on trees: On the design of an evolution strategy for scenario-based multi-period portfolio optimization under transaction costs," Swarm and Evolutionary Computation, vol. 17, pp. 74–87, 2014.