Asset allocation without pain: learning dynamic strategies directly from market data

Yuying Li

University of Waterloo

yuying@uwaterloo.ca

Collaborators: Chendi Ni and Peter Forsyth at UW and R. Carroll at Neuberger

Workshop: *Machine Learning in Quantitative Finance and Risk Management*

July 2, 2020

Outline

1. (Long Term) Optimal Dynamic Asset Allocation Problem
2. Direct: Learning from Nonparametric Augmented Market Data
3. Efficient: A NN Single Training Optimization for Dynamic Allocation
4. Objectives, Outperforming Stochastic Target, Distribution Shaping
5. Empirical Assessment: Accuracy, Robustness, and Characteristics
6. Concluding Remarks

Optimal Stochastic Dynamic Dynamic Allocation

- Allocate to a set of assets at $0 = t_0 < t_1 < \cdots < t_N = T$
- Future asset returns are random $\vec{R}(t_n)$, $n = 0, \cdots, N-1$
- Assume shorting and leverage are not allowed
- Determine optimal allocate strategy (weights) $\vec{\rho}_0$, ..., $\vec{\rho}_{N-1}$

$$\min_{\{\vec{\rho}_0,\ldots,\vec{\rho}_{N-1}\}} g(W(T)) \qquad (\text{Opt}^*)$$
$$\text{subject to} \quad 0 \le \vec{\rho}_n \le 1,\ 1^T \vec{\rho}_n = 1, n = 0, 1, \ldots, N-1,$$

Example: Pension Investment to Achieve Retirement Goals

- Pension funding shortfall increasingly shifts pensions from **Defined Benefit** to **Defined Contribution**

- Individual investors and wealth managers need high performance allocation strategies to meet retirement goals

Example:

- Yearly contributions $\{q(t_n)\}$ to the retirement account
- Optimally rebalance stock and bond yearly for 30 years to achieve retirement goals

Nested nonlinear dependence on controls

Given cash injection $\{q(t_n)\}$ and allocation $\{\vec{\rho}_0, ..., \vec{\rho}_{N-1}\}$, wealth $W(t_n)$:

$$
\begin{aligned}
\textit{for} \quad & n = 1, 2, ..., N-1 \\
& W(t_n^+) = W(t_n^-) + q(t_n) \\
& W(t_{n+1}^-) = \vec{\rho}_n^T \vec{R}(t_n) W(t_n^+) \\
& \qquad\qquad = \left(\vec{\rho}_n^T \vec{R}(t_n)\right)(W(t_n^-) + q(t_n)) \\
\textit{end} &
\end{aligned}
$$

Note. $W(T)$ becomes more nonlinear as N increases

Modelling and Computational Challenges

:

- **Data scarcity**:
  Only a single market return path realization is available, e.g., US stock and bond monthly returns in the last since since 1926
  - a typical characteristics in financial ML

- **High dimensionality**:
  Potentially many assets, stochastic benchmark

- **Additional complexity**:
  Shaping distribution, tax

:

Traditional Approach

- Parametric model for asset returns
  - e.g., double exponential jump diffusion model, regime switching model, etc
- Dynamic programming, e.g., HJB, converts multi-step optimization to single-step optimization
  - Compute strategy backwards in time
  - Computing strategy at rebalancing time for every possible state

Pain:

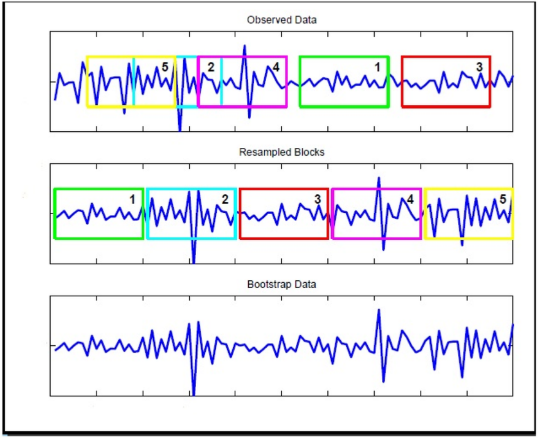
- Erroneous model assumption
- Difficult to estimate model
- Curse of dimensionality

How can we determine optimal strategies with less pain?

- Is it possible to learn an optimal long term allocation strategy directly from market?
- Is it possible to avoid curse of dimensionality?
- Can it be done robustly?
- How does solution compare to benchmark strategies in industry?
- How do we optimally outperforming a stochastic benchmark?

Nonparametric Return Data Augmentation

We use Stationary Block Resampling to augment the single market path.



Steps

- Randomly sample blocks of random size from observation sequence and concatenate blocks
- Blocksize follows a geometric distribution
- Observation sequence is assumed to be circular

Training and Testing

Augments supporting training using block resampled data:

- strategy is trained from random permutations of market running sessions of random lengths

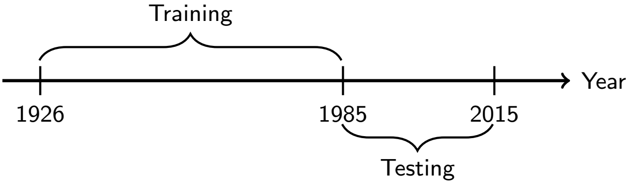
What about testing data?

In typical ML,

- Training data instances and testing data instances are sampled from the same distribution
- Testing instances are not seen in the training data set

Conduct out–of-sample and out-of-distribution testing

- **Out-of-sample test**: randomly block resample using the same expected blocksize
- **Out-of-distribution test**: randomly block resample using different expected blocksizes
- Training data and testing data from non-overlap-observation segments



- When assessing accuracy, we also use simulations from a parametric model for training/testing

Properties of block resampling out-of-sample test

Does block resampling leads to sound out-of-sample testing?

THEOREM. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two paths of N data points generated from a sequence of N_{tot} distinct observations using the stationary block bootstrap resampling with the expected blocksizes of $\hat{b}_1$ and $\hat{b}_2$ respectively. The probability of $\mathcal{P}_1$ and $\mathcal{P}_2$ being identical is

$$\frac{1}{N_{tot}}\left(\left(1-\frac{1}{\hat{b}_1}\right)\left(1-\frac{1}{\hat{b}_2}\right)+\frac{\frac{1}{\hat{b}_1}+\frac{1}{\hat{b}_1}-\frac{1}{\hat{b}_1\hat{b}_2}}{N_{tot}}\right)^{N-1}.$$

Remark. Let $N_{tot}=90\times 12$, $N=30\times 12$, $\hat{b}_1=\hat{b}_2=2\times 12$. For training set with 100,000 paths and testing set with 10,000 paths, the probability of existing a pair of identical training and testing paths is bounded by

$$100,000\times 10,000\times 8.737\times 10^{-39}<10^{-29}.$$

Scenario Optimal Control Formulation

Recall that we want to solve the original problem (Opt*) directly.

Given L sample paths $\{\vec{R}^{(j)}(t_n), n = 1, ..., N, j = 1, ..., L\}$, (Opt*) becomes

$$\min_{\{\vec{p}^{(j)}(t_0),...,\vec{p}^{(j)}(t_{N-1}),\forall j\}} \frac{1}{2} g(W^{(1)}(T), ..., W^{(L)}(T)) \qquad \text{(SPOpt)}$$
$$\text{subject to} \quad 0 \le \vec{p}^{(j)}(t_n) \le 1, n = 0, 1, ..., N-1, j = 1, ..., L$$
$$1^T \vec{p}^{(j)}(t_n) = 1, n = 0, 1, ..., N-1, j = 1, ..., L,$$

Challenges:

- Excessively large $O(MNL)$ variables/constraints: $\vec{p}^{(j)}(t_n),\ n = 0, 1, \ldots, N-1,\ j = 1, \ldots, L$
- Need a control model $\vec{p}(t_n)$ for out-of-sample

Optimal Control Model: a Machine Learning Approach

Ideas:

- Determine a single control function $\vec{p}(F(t_n))$ using state variables and time-to-go as features (parameterized by a set of weights)
- Solve the single optimization directly

$$
\begin{aligned}
\min_{\{\text{parameters of } \vec{p}(\cdot)\}} \quad & \frac{1}{2} g(W^{(1)}(T), ..., W^{(L)}(T)) \\
\text{subject to} \quad & 0 \leq \vec{p}(F^{(j)}(t_n)) \leq 1, n = 0, 1, ..., N-1, j = 1, ..., L \\
& 1^T \vec{p}(F^{(j)}(t_n)) = 1, n = 0, 1, ..., N-1, j = 1, ..., L,
\end{aligned}
$$

Neural Network Model $\vec{p}(\cdot)$

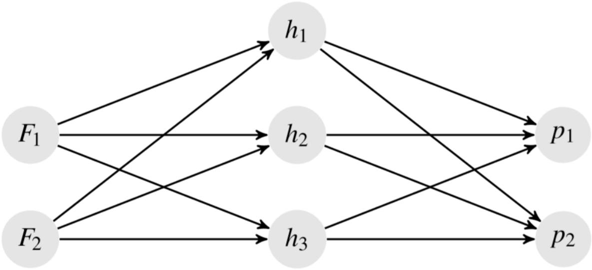


FIGURE 1: *A 2-layer NN representing control functions*

What about $O(MNL)$ constraints?

Neural Network Model

Logistic sigmoid outputs controls: weights for output layer $x \in R^{lM}$:

$$\vec{p}_m(F(t_n)) = \frac{e^{x_{km} h_k(F(t_n))}}{\sum_i e^{x_{ki} h_k(F(t_n))}}, \; 1 \le m \le M.$$

Input Features: $F(t_n) \in R^d$, weights for **hidden layer** $z \in R^{dl}$, and sigmoid activation yields:

$$h_j(F(t_n)) = \sigma(F_i(t_n) z_{ij}), \quad \sigma(u) = \frac{1}{1 + e^u},$$

where double summation convention denotes

$$F_i(t_n) z_{ij} \equiv \sum_{i=1}^{d} F_i z_{ij}, \; j = 1, ..., l.$$

Constraints are automatically satisfied:

$$0 \le \vec{p}_m(F(t_n)) \le 1, \quad 1^T \vec{p}(F(t_n)) = 1.$$

Training NN Optimization Problem

$$\min_{z \in R^{dl}, x \in R^{lM}} \quad \frac{1}{2} g(W^{(1)}(T), ..., W^{(L)}(T)) \qquad \text{(NNOpt)}$$

$$\text{where} \quad \vec{p}_m(F^{(j)}(t_n)) = \frac{e^{x_{km} h_k(F^{(j)}(t_n))}}{\sum_i e^{x_{ki} h_k(F^{(j)}(t_n))}}, m = 1, ..., M, n = 0, ..., N-1, j = 1, ..., L$$

$$h_k(F^{(j)}(t_n)) = \sigma(F_i^{(j)}(t_n) z_{ik}), k = 1, ..., l, n = 0, ..., N-1, j = 1, ..., L$$

(SPOpt): constrained, $O(MNL)$ variables, $O(MNL)$ constraints

(NNOpt): unconstrained, $l(d+M)$ variables, far smaller than $O(MNL)$
e.g., $d = 2$ (features), $l = 3$ (hidden nodes), $M = 2$ (assets)

Universal Approximation Theorem (Hornik 91): any smooth function can be represented by NN

Computational Cost:
gradient: $O(l(d+M)NL)$
Hessian : $O(l^2(d+M)^2 LN)$.

Objective Function: reaching a wealth target level W^*

Goal #1: minimize shortfall from the target W^*, we set

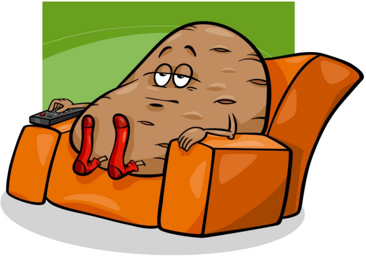
$$g(W(T)) \equiv \mathbf{E}\left[(\min(W(T) - W^*, 0))^2\right]$$

- Minimize the expected quadratic shortfall with respect to the target wealth W^*.

Constant Proportion Portfolio

How good is this optimal strategy?

- Benchmark: constant proportion 50/50 (Couch Potato) portfolio, 50% stocks and 50% bonds, annual rebalance.



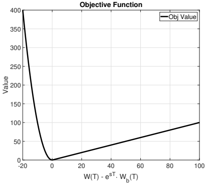
- Why not beat the benchmark $W_b(t)$ directly?

Shaping Distribution

- **Goal #2**: formulate an optimization problem to optimally beat the benchmark portfolio. Use the asymmetric objective function:

$$g(W(T)) \equiv \mathbb{E}\Big[\min\big(W(T) - e^{sT}\cdot W_b(T), 0\big)^2 + \max\big(W(T) - e^{sT}\cdot W_b(T), 0\big)\Big]$$

- Outperforming by a spread s in rate of return over the benchmark
- Quadratic-underperformance and linear outperformance objective



- wealth of the stochastic target, $W_b(t_n)$, becomes a state variable

Empirical Assessment Using Market Data

The US historical market data from 1926 - 2015 from the Center for Research in Security Prices (CRSP).

We consider allocations:

- Cap-weighted CRSP index and 3-month T-bill (2 assets)
- Equal-weighted CRSP index and 10-year treasury (2 assets)
- Index, 3-month and 10-year treasury (3 assets)

Data augmentation:

- Bootstrap market data: Bootstrap resampled paths from historical market path.
- Parametric (synthetic) market data: parameters a double exponential jump diffusion model is estimated from historical path

Accuracy comparison with ground truth (HJB), two assets

Wealth target $W^* = 705$, expected terminal wealth of 50/50

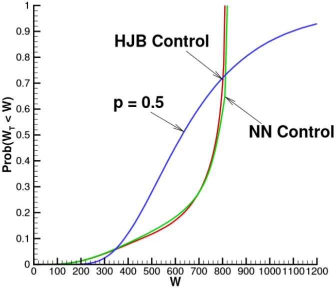
Training Performance on Parametric Model: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < 500)$	$Pr(W_T < 600)$
constant proportion ($p = .5$)	705	350	630	0.28	0.45
NN adaptive	705	159	782	0.13	0.18
HJB Optimal	705	153	782	0.12	0.17

Parametric model results from 160,000 Monte Carlo simulation runs

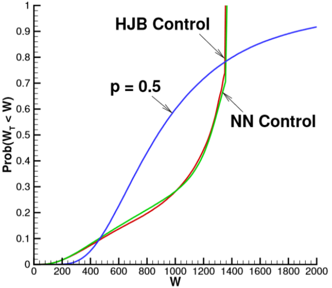
- **Accuracy**: NN training optimization achieves accuracy comparable to HJB (ground truth)
- **Performance**: optimal strategy achieves higher median, significantly lower shortfall probability for wealth level slightly below the target

Comparison with Ground Truth: CDF (2 assets)

Note. Lower curve ⇒ better performance (smaller shortfall probability)



(a) *Cap-weighted CRSP, 3 month T-bill.*



(b) *Equal-weighted CRSP, 10 year Treasury.*

Training and Out-of-Distribution Testing (3 assets)

Training with Expected Blocksize $\hat{b} = 0.5$ years: Market Cap Weighted					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < 500)$	$Pr(W_T < 600)$
Expected Blocksize $\hat{b} = 0.5$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	860	450	758	0.18	0.31
NN adaptive	860	264	986	0.15	0.20
Expected Blocksize $\hat{b} = 1$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	857	429	761	0.18	0.30
NN adaptive	865	264	994	0.15	0.20
Expected Blocksize $\hat{b} = 2$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	849	414	758	0.18	0.30
NN adaptive	867	254	986	0.13	0.19
Expected Blocksize $\hat{b} = 5$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	841	383	769	0.17	0.29
NN adaptive	878	246	994	0.12	0.18
Expected Blocksize $\hat{b} = 8$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	827	350	769	0.16	0.28
NN adaptive	886	236	996	0.11	0.16
Expected Blocksize $\hat{b} = 10$ years					
constant proportion ($p = (0.6, 0.1, 0.3)$)	826	337	772	0.16	0.27
NN adaptive	893	230	1002	0.10	0.15

Cap-weighted index, 3-month T-bill and 10-year treasury. Training data: expected blocksize $\hat{b} = 0.5$ years. Test data:

$\hat{b} = 1, 2, 5, 10$ (years)

Stochastic Target: Terminal Wealth Distribution

Elevated target is $e^{sT} W_{50/50}(\cdot)$ with the spread $s = 1\%$

Expected blocksize $= 0.5$ years for both training and testing.

Out-of-distribution testing with other blocksize is similar.

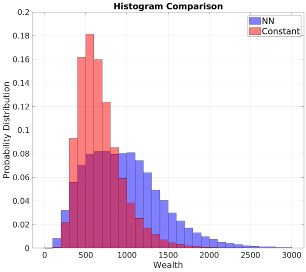


Figure: Training: $\hat{b} = 0.5$ years

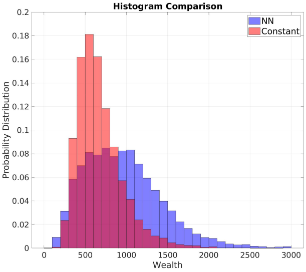
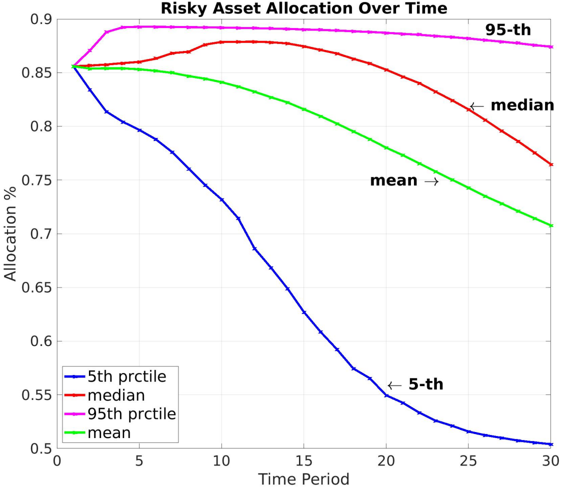


Figure: Testing: $\hat{b} = 0.5$ years

Strategy Characteristics: Risky Asset Allocation Over Time



Risky asset allocation

- decreases over time
- mostly stays above 50%

(Out-of-sample) Historical Path from 1985 - 2015

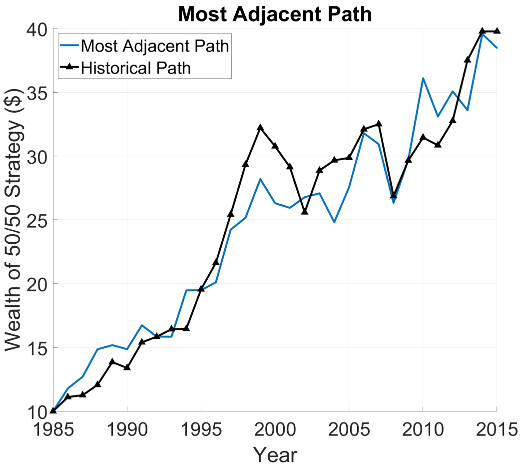


Figure: Historical and closest from training wealth paths: constant proportion

- The historical path is not in the training set

Backtest : Historical Path from 1985 - 2015

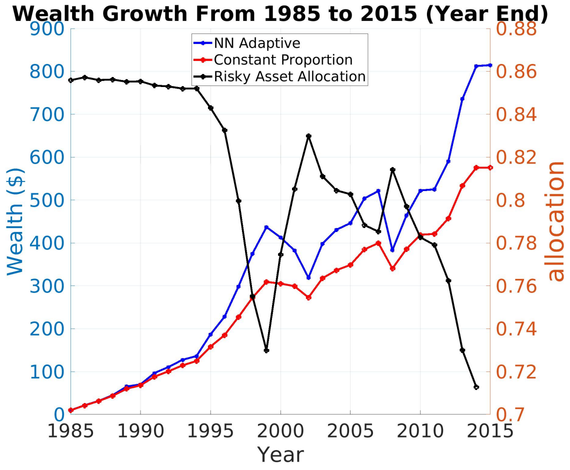


Figure: Cap-weighted CRSP index and 3-month T-bill (training on bootstrap data)

- The cumulative wealth of the NN adaptive strategy is higher than the benchmark strategy during the entire investment period.
- A contrarian strategy

Concluding Remarks

We propose a framework to learn dynamic optimal allocation strategy directly from market.

In the proposed framework, we

- generate raining and testing data directly from block resampling of market return path
- show that block resampling generates sound out-of-sample and out-of-distribution testing
- solve a single scenario training optimization for dynamic strategy
- demonstrate NN strategy achieves high accuracy and efficiency

Concluding Remarks

By designing suitable objectives, we determine optimal strategies to

- achieve a target wealth level
- outperform a benchmark by shaping terminal wealth distribution

Based on historical market data, we show that optimal strategies

- consistently outperform constant proportion benchmark strategy
- perform robustly out-of-sample and out-of-distribution
- outperform on the (out-of-sample) historical path

Optimal strategy is a contrarian strategy and, on average,

- risky asset allocation decreases over time

The proposed method can be applied to many financial decision problems.

References

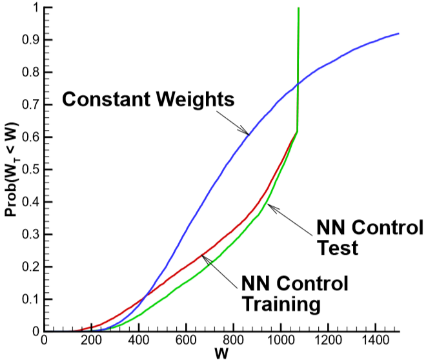
Politis, D. and H. White (2004).
Automatic block-length selection for the dependent bootstrap.
Econometric Reviews 23, 53–70.

K. Hornik.
Approximation capabilities of multilayer feedforward networks.
Neural Networks, 4(2):251–257, 1991.

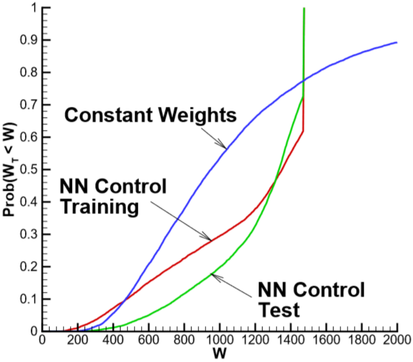
Li, Y. and P. A. Forsyth (2019).
A data-driven neural network approach to optimal asset allocation for target based defined contribution pension plans.
Insurance: Mathematics and Economics 86, 189–204.

Ni, C., Li, Y. , and P. A. Forsyth (2020).
Optimal Asset Allocation For Outperforming A Stochastic Benchmark Target
Journal of Quantitative Finance , submitted.

Appendix: Shortfall Probability at Wealth W (3 assets)



(a) *Cap-weighted CRSP, 3 month T-bill, 10 year Treasury*



(b) *Equal-weighted CRSP, 3 month T-bill, 10 year Treasury.*

Stochastic Target: 2 assets (cap weighted index)

Median Internal Rate of Return (IRR)

Strategy	Training	Testing
constant proportion (p =.5)	4.38%	4.37%
neural network (NN) adaptive	6.46%	6.45%

IRR: average annual return rate to reach the terminal wealth

$$W(T) = \sum_{t=0}^{T-1} q(t)(1 + IRR)^{T-t}.$$

Bootstrap Resampling Test with Different $\hat{b}$

Training Results on Bootstrap Data: Expected Blocksize $\hat{b} = 0.5$ years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	678	276	624	0.50	0.84
adaptive	963	474	913	0.27	0.50
Testing Results on Bootstrap Data: Expected Blocksize $\hat{b} = 2$ years					
Strategy	$E(W_T)$	$std(W_T)$	$median(W_T)$	$Pr(W_T < median(W_T^{CP}))$	$Pr(W_T < median(W_T^{NN}))$
constant proportion($p = 0.5$)	679	267	629	0.50	0.84
adaptive	962	449	921	0.26	0.50

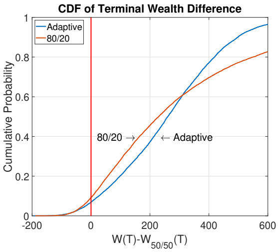
Table: Test results on bootstrap market data with a different blocksize.

Note:

- optimal strategy achieves higher mean, median, lower probability of falling short of median wealths
- results are similar for all blocksizes.

Comparison to 80/20 strategy

CDF of wealth difference from 50/50 strategy



Observations

- significant probability of outperforming with large magnitude
- small probability of underperforming