Machine Learning and the Implementable Efficient Frontier

Theis Ingerslev Jensen

Yale School of Management

Bryan Kelly

AQR, Yale School of Management, and NBER

Semyon Malamud

Swiss Finance Institute, EPFL, and CEPR

Lasse Heje Pedersen

AQR, Copenhagen Business School, CEPR

The views expressed are those of the authors and not necessarily those of AQR

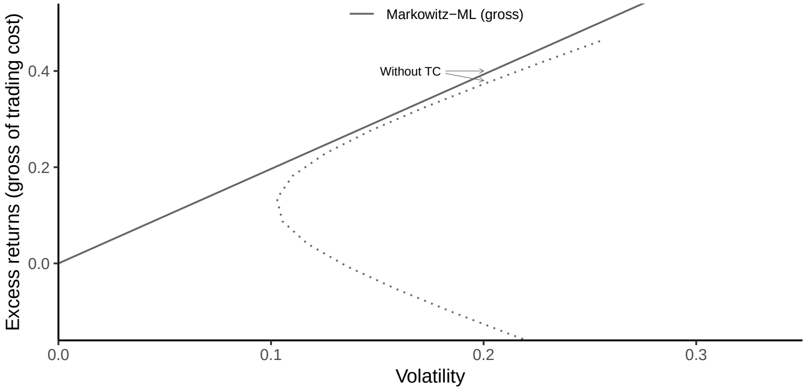
Motivation: ML and Implementable Portfolios

- ML models are great at predicting stock returns
  - For example, Gu et al. (2020)
- But most ML papers ignore trading costs, implying unrealistic
  - profits from illiquid stocks (Avramov et al., 2023)
  - key characteristics, e.g. short-term reversal (Chen et al., 2023)
- Questions:
  - Can investors benefit from ML after t-costs?
  - Which signals have greatest economic feature importance?
  - Lessons for asset pricing?

What We Do

- We introduce the "Implementable efficient frontier" (IEF)
  - After-cost, out-of-sample version of standard efficient frontier
- We show that
  - Standard ML implementations leads to a poor IEF
  - New theory-guided ML leads to a powerful IEF
  - Economic feature importance:
    - Quality and Value: large impact on the IEF
    - Short-Term Reversal: limited impact for a large investor

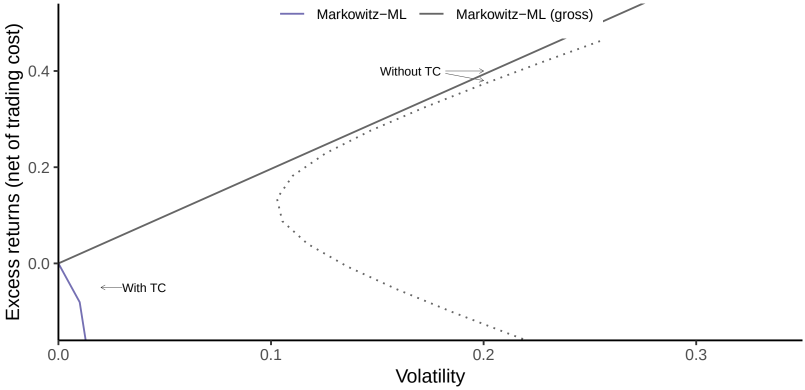
Almost the Standard Efficient Frontier – but OOS



Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

The Implementable Efficient Frontier

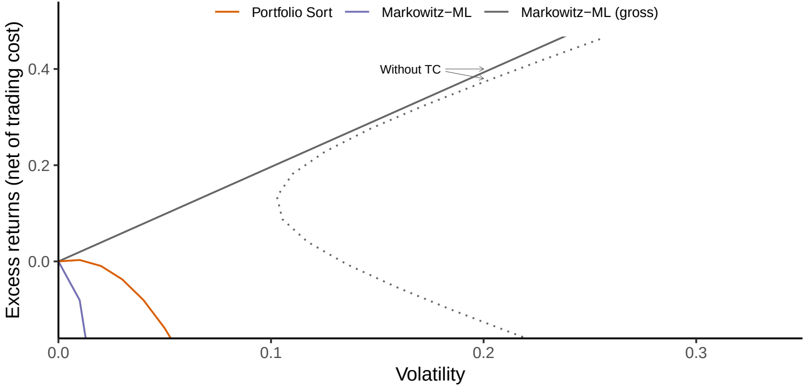


Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

The Implementable Efficient Frontier

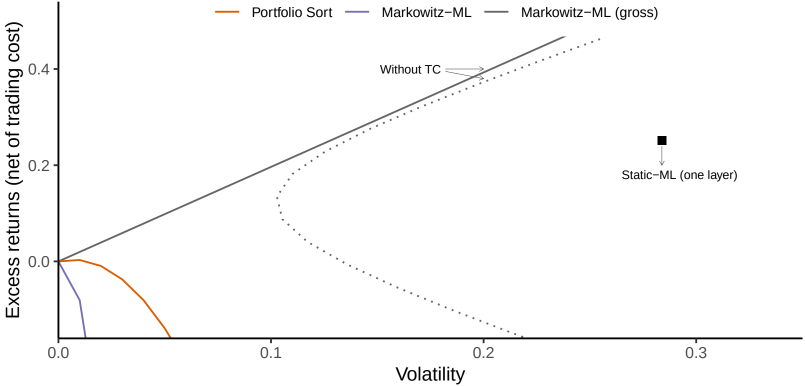


Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

The Implementable Efficient Frontier



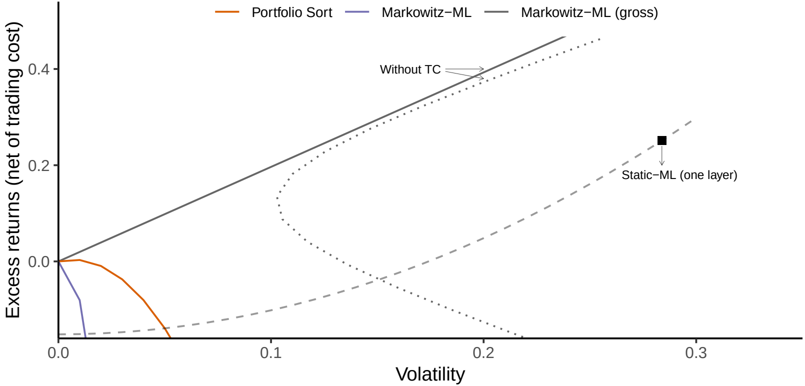
Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ○

The Implementable Efficient Frontier



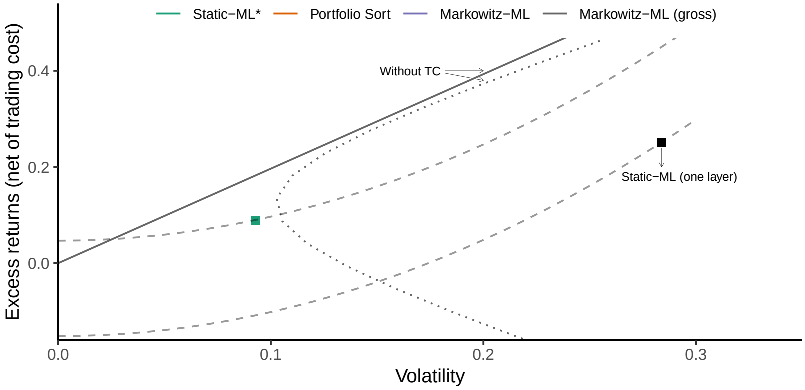
Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ∘

The Implementable Efficient Frontier



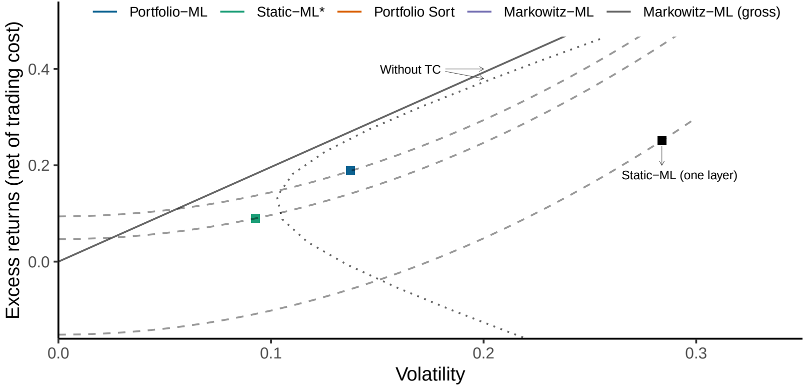
Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ○

The Implementable Efficient Frontier



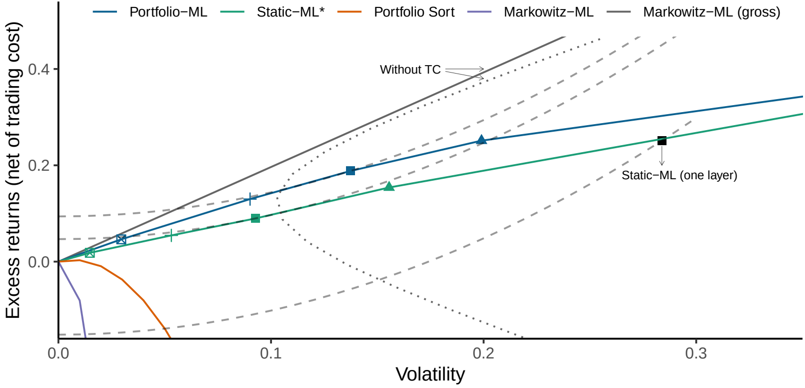
Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ○

The Implementable Efficient Frontier



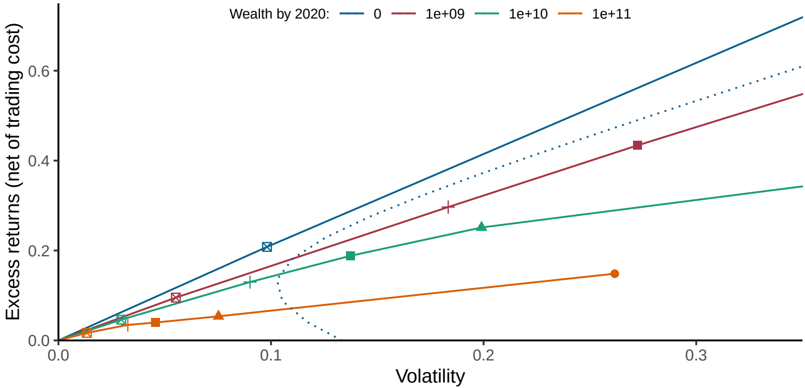
Risk and expected return net of t-costs with a wealth of $10B by 2020

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ○

The Implementable Efficient Frontier: By Assets



Risk and expected return net of t-costs using Portfolio-ML

Everything is out-of-sample: 1981-2020

Dotted line: Mean-variance frontier of risky assets, $\sum_i \pi_i = 1$, without t-costs

Markers: Relative risk aversion (left to right): 100 ⊠, 20 +, 10 □, 5 △, 1 ∘

Model

Model

Securities: N risky assets traded at times $t = \ldots, -1, 0, 1, 2, \ldots$

- Excess returns: $r_t = (r_{1,t}, \ldots, r_{N,t})'$
- Characteristics: $s_t = (s_{1,t}, \ldots, s_{N,t}) \in \mathbb{R}^{N \times K}$
- Expected returns

$$E_t[r_{t+1}] \;=\; \mu(s_t)$$

- T-costs
  - Market impact: $\frac{1}{2}\Lambda_t \tau_t$
  - Total trading cost: $\frac{1}{2}\tau_t' \Lambda_t \tau_t$
  - T-cost example

Model

Securities: N risky assets traded at times $t = ..., -1, 0, 1, 2, ...$

- Excess returns: $r_t = (r_{1,t}, ..., r_{N,t})'$
- Characteristics: $s_t = (s_{1,t}, ..., s_{N,t}) \in \mathbb{R}^{N \times K}$
- Expected returns

$$E_t[r_{t+1}] = \mu(s_t)$$

- T-costs
  - Market impact: $\frac{1}{2}\Lambda_t \tau_t$
  - Total trading cost: $\frac{1}{2}\tau_t' \Lambda_t \tau_t$
  - T-cost example

Investor

- Portfolio weight (key control variable): $\pi_{n,t} = \pi^{\$}_{n,t} / w_t$
- Trade:

$$\tau_t = \pi^{\$}_t - \text{diag}(1 + r^f_t + r_t)\pi^{\$}_{t-1} = w_t\left(\pi_t - g_t \pi_{t-1}\right)$$

where $g_t = \text{diag}\left(\frac{1+r^f_t+r_t}{1+g^w_t}\right)$ captures the growth in portfolio weights

Model: Objective

Mean-variance utility with risk aversion γ:

- Choose π_t for all t to maximize

$$\text{utility} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left[\text{return}_{t+1}(\pi_t) - \text{TC}_t(\pi_t, \pi_{t-1}) - \text{risk}_{t+1}(\pi_t)\right]$$

$$= \mathbb{E}\left[\mu(s_t)'\pi_t - \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)' \Lambda \left(\pi_t - g_t\pi_{t-1}\right) - \frac{\gamma}{2}\pi_t'\Sigma\pi_t\right],$$

Model: Objective

Mean-variance utility with risk aversion γ:

- Choose π_t for all t to maximize

$$\text{utility} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\left[\text{return}_{t+1}(\pi_t) - \text{TC}_t(\pi_t, \pi_{t-1}) - \text{risk}_{t+1}(\pi_t)\right]$$

$$= \mathbb{E}\left[\mu(s_t)'\pi_t - \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda\left(\pi_t - g_t\pi_{t-1}\right) - \frac{\gamma}{2}\pi_t'\Sigma\pi_t\right],$$

What is new? General μ and stationary state variables $\longrightarrow$ ML

- Percentage returns, r_t, their means, $\mu(s_t)$, variances, Σ, t-cost, Λ
- Fractional portfolio weights, π_t
- Portfolio growth, g_t, is a complication, cf. Constantinides (1986)
  - Gârleanu and Pedersen (2013) focus on $\pi^{\#}$ =number of shares and $r^{\#}$ =price changes so no growth, $g \equiv 1$
  - but $r^{\#}$ =price changes not stationary empirically

Our Dynamic Solution

Key Result: Optimal Strategy

Proposition (Optimal dynamic strategy)

The solution to the portfolio problem is

$$\pi_t \;=\; m\, g_t \pi_{t-1} + (I - m) A_t$$

with aim portfolio A_t

$$A_t \;=\; (I - m)^{-1} \sum_{\tau=0}^{\infty} (m\bar{g})^{\tau}\, c E_t \left[\underbrace{\frac{1}{\gamma} \Sigma^{-1} \mu(s_{t+\tau})}_{\text{Markowitz}_{t+\tau}} \right]$$

where $c = \frac{\gamma}{w} m \Lambda^{-1} \Sigma$ and m given in the paper.

Implementing the Dynamic Solution with ML

Machine Learning about Portfolio Weights: Portfolio-ML

- Standard ML objective minimizes squared return errors:

$$\min_f \frac{1}{NT} \sum_{n,t} (r_{n,t} - f(s_{n,t}))^2$$

- But we want economic objective that maximizes utility:

$$\max_\pi \text{utility}(\pi) = \frac{1}{T} \sum_{t=1}^{T} [\text{return}_{t+1}(\pi_t) - \text{TC}_t(\pi_t, \pi_{t-1}) - \text{risk}_{t+1}(\pi_t)]$$

Difficult to solve: π depends on current and past signals

- Our Proposition suggests two solutions:
    1. Use theoretical solution, plus standard ML over many horizons
    2. Use theoretical solution, plus trick to find A directly via ML

Implementing Key Result in Practice: Multiperiod-ML

- Use our Proposition to find A

$$A_t^{\text{Multiperiod-ML}} = (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\bar{g})^{\tau}c\frac{1}{\gamma}\Sigma^{-1}E_t[r_{t+1+\tau}]$$

by using ML to estimate $E_t[r^i_{t+1+\tau}]$ across horizons τ

- Compute portfolio $\pi_t^{\text{Multiperiod-ML}}$:
  - Initial portfolio: $\pi_0^{\text{Multiperiod-ML}} = 0$
  - Successive portfolios:
    $\pi_t^{\text{Multiperiod-ML}} = m\,g_t\pi_{t-1}^{\text{Multiperiod-ML}} + (I-m)A_t^{\text{Multiperiod-ML}}$

Implementing Key Result in Practice: Multiperiod-ML

- Use our Proposition to find A

$$A_t^{\text{Multiperiod-ML}} = (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\bar{g})^{\tau}c\frac{1}{\gamma}\Sigma^{-1}E_t[r_{t+1+\tau}]$$

  by using ML to estimate $E_t[r^i_{t+1+\tau}]$ across horizons τ

- Compute portfolio $\pi_t^{\text{Multiperiod-ML}}$:
  - Initial portfolio: $\pi_0^{\text{Multiperiod-ML}} = 0$
  - Successive portfolios:
    $\pi_t^{\text{Multiperiod-ML}} = m\,g_t\pi_{t-1}^{\text{Multiperiod-ML}} + (I-m)A_t^{\text{Multiperiod-ML}}$

- Insight: t-costs $\rightarrow$ relevance of short- and long-run returns
  - i.e., the whole term structure of returns
  - because investor will be "stuck" with positions over time

Machine Learning about Portfolio Weights: Portfolio-ML

- To find the optimal A_t, we propose a linear parametric portfolio:

$$A_t^{\text{Portfolio-ML}} = s_t \beta, \qquad \beta \in \mathbb{R}^p$$

- Maximizing utility with this parameterization leads to:

$$\hat{\beta} = (E_T[\tilde{\Sigma}_t])^{-1} E_T[\tilde{r}_{t+1}]$$

where $E_T[\tilde{\Sigma}_t]$ and $E_T[\tilde{r}_{t+1}]$ can be computed from observed data

Machine Learning about Portfolio Weights: Portfolio-ML

- To find the optimal A_t, we propose a linear parametric portfolio:

$$A_t^{\text{Portfolio-ML}} = s_t \beta, \qquad \beta \in \mathbb{R}^p$$

- Maximizing utility with this parameterization leads to:

$$\hat{\beta} = (E_T[\tilde{\Sigma}_t])^{-1} E_T[\tilde{r}_{t+1}]$$

where $E_T[\tilde{\Sigma}_t]$ and $E_T[\tilde{r}_{t+1}]$ can be computed from observed data

- **Closed-form solution to extremely hard problem!**

Portfolio-ML solution details

Machine Learning about Portfolio Weights: Portfolio-ML

- We add a couple of "ML modifications" to the linear solution
  1. ML modification 1: Add ridge penalty chosen via cross-validation
  2. ML modification 2: Use random features transform to create a non-linear parametric portfolio

$$A_t^{\text{Portfolio-ML}} = RF(s_{n,t})\beta$$

- $\hat{\beta}$ still has a closed-form solution, that is now regularized by the ridge penalty and captures non-linearities via the RF transform

Random features details | Benchmark portfolio choice methods with ML
Alternative implementation: Multiperiod-ML

Empirical Setup and Results

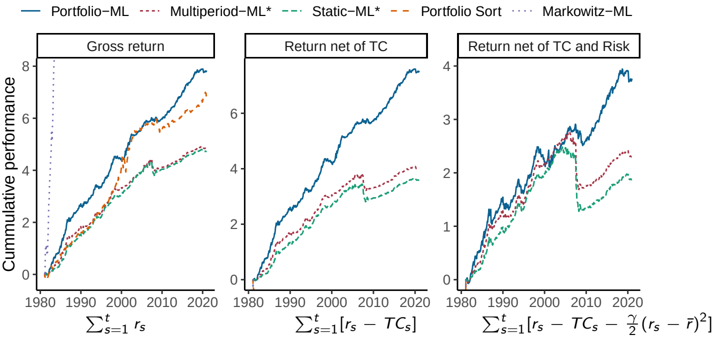
Data and Empirical Methodology

- Sample: Monthly data on US stocks, 1952–2020
  - CRSP stocks, market-cap $> 50^{\text{th}}$ percentile of NYSE stocks, i.e., roughly the 1000 largest stocks
- Risk: Estimate Σ_t via factor model based on characteristics
- T-cost: Assume diagonal Λ_t, with $\lambda_{n,n} \propto 1/\text{DTV}_{n,t}$

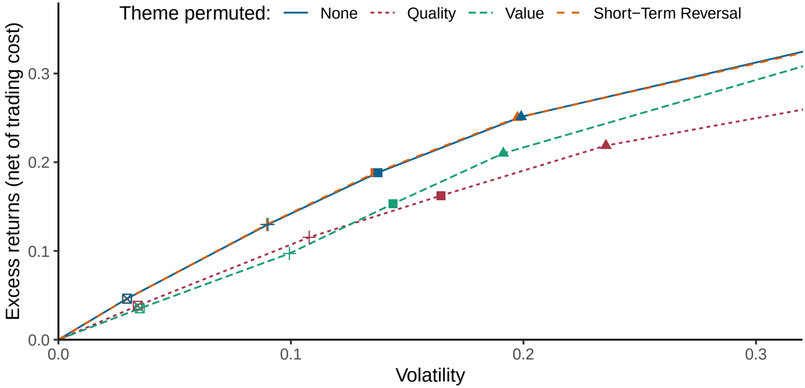
⏩ Candidate portfolios overview ⏩ ML and portfolio tuning

Out-of-Sample Performance 1981-2020

Method	R	Vol.	SR_{gross}	TC	R-TC	SR_{net}	Utility	Turnover	Lev.
One tuning layer									
Portfolio-ML	0.20	0.14	1.43	0.008	0.19	1.38	0.095	0.32	3.60
Multiperiod-ML	0.32	0.34	0.95	0.182	0.14	0.41	-0.437	1.47	12.70
Static-ML	0.28	0.27	1.06	0.033	0.25	0.94	-0.106	0.76	11.21
Portfolio Sort	0.17	0.15	1.10	1.972	-1.81	-11.87	-1.921	2.60	2.00
Markowitz-ML	3.12	1.56	2.00	+	-	-	-	56.33	53.15
Two tuning layers									
Multiperiod-ML*	0.11	0.08	1.33	0.014	0.09	1.16	0.060	0.40	2.50
Static-ML*	0.13	0.10	1.36	0.024	0.11	1.11	0.060	0.61	3.22

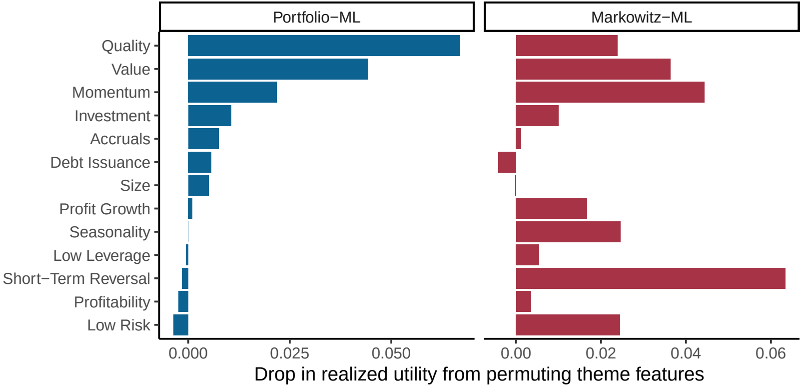


Economic Feature Importance



- Loosing information from Short-Term Reversal doesn't affect implementable efficient frontier (low economic feature importance)
- Loosing information from Quality or Value greatly affects the implementable efficient frontier (high economic feature importance)

Economic Feature Importance



Other Results

- Theoretical results
  - Proposition 1: The Sharpe ratio along the implementable efficient frontier is declining with σ. Hence, efficient frontier is no longer a straight line, and leverage is costly
  - Proposition 4: Trading cost increase when investor takes more risk, but aim becomes more tilted towards persistent signals and liquid stocks
  - Proposition 5: Tiny investors hold Markowitz, huge investors holds mostly risk-free asset plus "maximum dollar portfolio"

- Empirical results
  - Outperformance is statistically significant
  - Portfolio-ML trades smoothly, especially for illiquid stocks
  - Low correlation between Portfolio-ML and Markowitz-ML (important for SDF pricing!)

Conclusion

We develop a method for optimal portfolio selection when

- trading is costly
- returns predictable by a *general* function of characteristics
- driving state variables are stationary, thus handling portfolio growth

Findings

1. Intuitive solution, which can be implemented via ML
2. Empirical results show significant out-of-sample gains
3. Novel view of which security characteristics are important

Appendix

References Cited in Slides (see paper for further references)

Avramov, D., S. Cheng, and L. Metzker (2023). Machine learning vs. economic restrictions: Evidence from stock return predictability. *Management Science 69*(5), 2587–2619.

Chen, L., M. Pelger, and J. Zhu (2023). Deep learning in asset pricing. *Management Science*.

Constantinides, G. M. (1986). Capital market equilibrium with transaction costs. *Journal of Political Economy 94*, 842–862.

Gârleanu, N. and L. H. Pedersen (2013). Dynamic trading with predictable returns and transaction costs. *The Journal of Finance 68*(6), 2309–2340.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

T-costs - Example

T-costs: Market impact, $\frac{1}{2}\Lambda_t \tau_t$, so t-cost $\frac{1}{2}\tau_t' \Lambda_t \tau_t$

Example: 1 stock with $\frac{1}{2}\Lambda = 10^{-8}$ and price of \$100/share

- Trade: $\tau = \$1M$, i.e., buy 10,000 shares
  - Market impact: $\frac{1}{2}\Lambda\tau = 10^{-8} \times \$1M = 1\%$
    - Price moves from \$100 to \$102 with average fill of \$101
  - Total cost: $\$1M \times 1\% = \$10,000$
- Double the trade: $\tau = \$2M$, i.e., buy 20,000 shares
  - Market impact: $\frac{1}{2}\Lambda\tau = 10^{-8} \times \$2M = 2\%$
    - Price moves from \$100 to \$104 with average fill of \$102
  - Total cost: $\$2M \times 2\% = \$40,000$
    - Quadruples!

Back

Portfolio-ML Solution Details

- Portfolio depends on current and past aim portfolios, $A_t = s_t\beta$:

$$\pi_t = \sum_{\theta=0}^{\infty}\left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1}\right)(I-m)A_{t-\theta} = \underbrace{\left[\sum_{\theta=0}^{\infty}\left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1}\right)(I-m)s_{t-\theta}\right]}_{\equiv \Pi_t}\beta$$

- Inserting $\pi_t = \Pi_t\beta$ into the expression for utility, we get

$$\begin{aligned}
\text{utility} &= \frac{1}{T}\sum_t\left[r'_{t+1}\pi_t - \frac{\gamma}{2}\pi'_t\Sigma\pi_t - \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda\left(\pi_t - g_t\pi_{t-1}\right)\right] \\
&= \frac{1}{T}\sum_t\left[r'_{t+1}\Pi_t\beta - \frac{\gamma}{2}\beta'\Pi'_t\Sigma\Pi_t\beta - \frac{w}{2}\left(\Pi_t\beta - g_t\Pi_{t-1}\beta\right)'\Lambda\left(\Pi_t\beta - g_t\Pi_{t-1}\beta\right)\right] \\
&= \frac{1}{T}\sum_t\left[\underbrace{r'_{t+1}\Pi_t}_{\equiv\tilde{r}'_{t+1}}\beta - \frac{1}{2}\beta'\underbrace{\left[\gamma\Pi'_t\Sigma\Pi_t + w(\Pi_t - g_t\Pi_{t-1})'\Lambda(\Pi_t - g_t\Pi_{t-1})\right]}_{\equiv\tilde{\Sigma}_t}\beta\right] \\
&\equiv E_T[\tilde{r}'_{t+1}]\beta - \frac{1}{2}\beta' E_T[\tilde{\Sigma}_t]\beta
\end{aligned}$$

- Solution with ridge penalty $-\lambda\beta'\beta$ is closed-form: $\beta = (E_T[\tilde{\Sigma}_t] + \lambda I)^{-1}E_T[\tilde{r}_{t+1}]$

Random feature regression details

- Want to go from linear portfolio: $A_t^{\text{Portfolio-ML}} = s_t\beta$
- to a more general ML approach: $A_t^{\text{Portfolio-ML}} = f(s_t)$

Random feature regression details

- Want to go from linear portfolio: $A_t^{\text{Portfolio-ML}} = s_t\beta$
- to a more general ML approach: $A_t^{\text{Portfolio-ML}} = f(s_t)$
- Simple method: random feature (RF) regression

$$f(s_{n,t}) \approx RF(s_{n,t})\beta$$

where

$$RF(s_{n,t}) = \frac{1}{\sqrt{p}}\left[\sin(s_{n,t}'w^1), \cos(s_{n,t}'w^1), \ldots, \sin(s_{n,t}'w^{p/2}), \cos(s_{n,t}'w^{p/2})\right]',$$

$$w^j \in \mathbb{R}^{115} \sim iidN(0, \eta^2 I) \text{ for } j = 1, ..., p/2$$

Random feature regression details

- Want to go from linear portfolio: $A_t^{\text{Portfolio-ML}} = s_t \beta$
- to a more general ML approach: $A_t^{\text{Portfolio-ML}} = f(s_t)$
- Simple method: random feature (RF) regression

$$f(s_{n,t}) \approx RF(s_{n,t})\beta$$

where

$$RF(s_{n,t}) = \frac{1}{\sqrt{p}} \left[\sin(s'_{n,t} w^1), \cos(s'_{n,t} w^1), \ldots, \sin(s'_{n,t} w^{p/2}), \cos(s'_{n,t} w^{p/2}) \right]',$$

$$w^j \in \mathbb{R}^{115} \sim iidN(0, \eta^2 I) \text{ for } j = 1, ..., p/2$$

- Non-linear parametric portfolio with $\beta \in \mathbb{R}^p$

$$A_t^{\text{Portfolio-ML}} = \text{diag}\left(\frac{1}{\sigma_{n,t}}\right) RF(s_t)\beta$$

- Everything works just the same as linear
- Need to tune hyper-parameters p, η and ridge penalty λ

Back

Alternative Implementation: Multiperiod-ML

- Use ML to estimate $E_t[r^i_{t+1+\tau}]$ <u>across horizons τ</u>

- Compute the aim portfolio at each time

$$A_t^{\text{Multiperiod-ML}} = (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\bar{g})^{\tau} c\frac{1}{\gamma}\Sigma^{-1}E_t[r_{t+1+\tau}]$$

- Compute portfolio $\pi_t^{\text{Multiperiod-ML}}$:
  - Initial portfolio: $\pi_0^{\text{Multiperiod-ML}} = 0$
  - Successive portfolios:
    $\pi_t^{\text{Multiperiod-ML}} = m\, g_t \pi_{t-1}^{\text{Multiperiod-ML}} + (I-m)A_t^{\text{Multiperiod-ML}}$

Alternative Implementation: Multiperiod-ML

- Use ML to estimate $E_t[r^i_{t+1+\tau}]$ across horizons τ
- Compute the aim portfolio at each time

$$A_t^{\text{Multiperiod-ML}} = (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\bar{g})^{\tau} c\frac{1}{\gamma}\Sigma^{-1}E_t[r_{t+1+\tau}]$$

- Compute portfolio $\pi_t^{\text{Multiperiod-ML}}$:
  - Initial portfolio: $\pi_0^{\text{Multiperiod-ML}} = 0$
  - Successive portfolios:
    $\pi_t^{\text{Multiperiod-ML}} = m\, g_t \pi_{t-1}^{\text{Multiperiod-ML}} + (I-m)A_t^{\text{Multiperiod-ML}}$

- Insight: t-costs $\longrightarrow$ relevance of short- and long-run returns
  - i.e., the whole term structure of returns
  - because investor will be "stuck" with positions over time

Back

ML in Asset Pricing: Standard Approach

Two-step approach:

1. ML to predict returns

$$\min_{f:\mathbb{R}^K \to \mathbb{R}} \frac{1}{TN} \sum_{n,t} [r_{n,t+1} - f(s_{n,t})]^2 .$$

2. After ML is done
   - Use predictions to build long/short factor: $\pi_t^{\text{Factor-ML}}$
   - Possibly combine with risk estimates: $\pi_t^{\text{Markowitz-ML}}$
   - Possibly even perform t-cost optimization: $\pi_t^{\text{Static-ML}}$

ML in Asset Pricing: Standard Approach

Two-step approach:

1. ML to predict returns

$$\min_{f:\mathbb{R}^K \to \mathbb{R}} \frac{1}{TN} \sum_{n,t} [r_{n,t+1} - f(s_{n,t})]^2 .$$

2. After ML is done
   - Use predictions to build long/short factor: $\pi_t^{\text{Factor-ML}}$
   - Possibly combine with risk estimates: $\pi_t^{\text{Markowitz-ML}}$
   - Possibly even perform t-cost optimization: $\pi_t^{\text{Static-ML}}$

- **We compare our approach to each of these**
  - $\pi_t^{\text{Static-ML}}$ sophisticated: ML + risk + t-cost-optimization
    - should work reasonably well, but
    - only considers returns over $t+1$ while the optimal solution consider $t+1, t+2, \ldots$

back

Candidate Portfolio Methods

	$E_t^{ML}(r_{t+1})$	$\text{Var}_t(r_{t+1})$	T-cost	Fwd-looking dynamics
Factor-ML	Yes			
Markowitz-ML	Yes	Yes		
Static-ML	Yes	Yes	Yes	
Multiperiod-ML	Yes	Yes	Yes	$E_t^{ML}(r_{t+\tau})$
Portfolio-ML		Yes	Yes	utility(A_t)

Back

ML and Portfolio Tuning

- Machine Learning: Random feature regression (RFF)
  - either for predicting portfolio weights (portfolio-ML)
  - or for predicting returns (other methods)

- Portfolio tuning:
  - Out-of-sample (OOS) period starts in 1981
  - Two layers of "portfolio tuning"

    1st: Find RFF hyper-parameters for all methods

    2nd: Choose hyper-parameters that force Static-ML and Multiperiod-ML to take less risk
  - All hyper-parameters are updated yearly

Portfolio tuning details

Portfolio Tuning

Train-validation-test split to find hyperparameters

- 1st tuning layer: Find λ, p, η for Random Feature Regression
- 2nd tuning layer: Find optimal u, v, k for portfolios

$$
\begin{aligned}
E_t^*[r_{t+\tau}] &= uE_t[r_{t+\tau}] \\
\Sigma_t^* &= \Sigma_t + v\,\mathrm{diag}(\sigma_t) \\
\Lambda_t^* &= k\Lambda_t
\end{aligned}
$$

- OOS test period starts in 1981, hyper-parameters updated yearly

Hyper-parameter	Method		
1st tuning layer, h	Portfolio-ML	Multiperiod-ML	Static-ML
Ridge penalty, λ	$\{0, e^4, e^5, \ldots, e^8\}$	$\{0, e^{-10}, e^{-9.8}, \ldots, e^{10}\}$	$\{0, e^{-10}, e^{-9.8}, \ldots, e^{10}\}$
#random features, p	$\{2^6, 2^7, 2^8, 2^9\}$	$\{2^1, 2^2, \ldots, 2^{10}\}$	$\{2^1, 2^2, \ldots, 2^{10}\}$
Std of weights, η	$\{e^{-3}, e^{-2}\}$	$\{e^{-4}, e^{-3}, e^{-2}, e^{-1}\}$	$\{e^{-4}, e^{-3}, e^{-2}, e^{-1}\}$
2nd tuning layer, h^*		Multiperiod-ML*	Static-ML*
Adj. to mean, u		$\{0.25, 0.50, 1.00\}$	$\{0.25, 0.50, 1.00\}$
Adj. to variance, v		$\{1, 2, 3\}$	$\{1, 2, 3\}$
Adj. to t-cost, k		$\{1, 2, 3\}$	$\{\frac{1}{1}, \frac{1}{3}, \frac{1}{5}\}$

Back

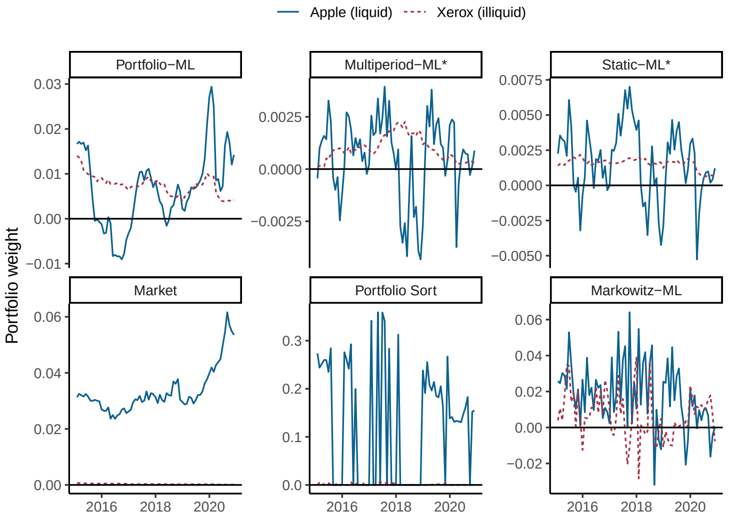
Outperformance is statistically significant

Table: Outperformance of One Method vs. Another

	Portfolio-ML	Multiperiod-ML*	Static-ML*	Portfolio Sort	Markowitz-ML
Portfolio-ML		95%	96%	100%	100%
Multiperiod-ML*	5%		51%	100%	100%
Static-ML*	4%	49%		100%	100%
Portfolio Sort	0%	0%	0%		100%
Markowitz-ML	0%	0%	0%	0%	

p-value of whether the average utility of the portfolio method in the row is greater than that in the column

Example Portfolio Weights: Apple vs. Xerox



Correlations

Table: Portfolio Correlations

	Portfolio-ML	Multiperiod-ML*	Static-ML*	Portfolio Sort
Multiperiod-ML*	0.51			
Static-ML*	0.55	0.80		
Portfolio Sort	0.24	0.46	0.53	
Markowitz-ML	0.17	0.50	0.59	0.56

Portfolio Statistics over Time