A comprehensive machine learning framework for dynamic portfolio choice with transaction costs

August 2025; Deep Learning Summer School, University of Turin

Luca Gaegauf

Institute for Banking and Finance, The University of Zurich

Simon Scheidegger

Department of Economics, University of Lausanne, Enterprise for Society (E4S)

Fabio Trojani

University of Geneva, University of Turin and SFI

Background, Motivation and Objective

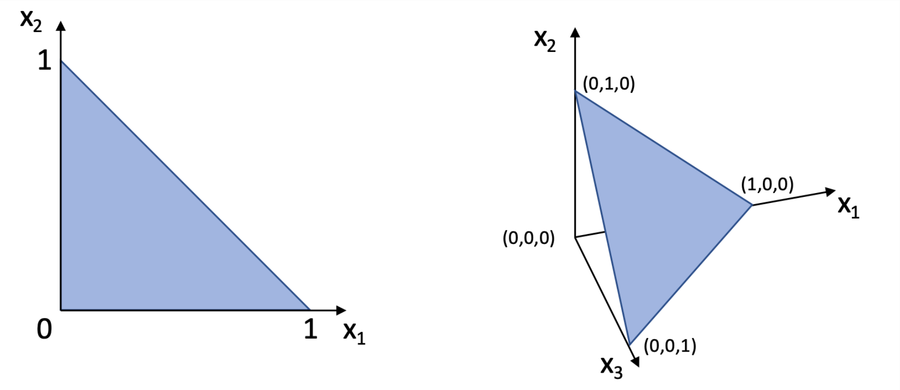
Dynamic portfolio choice

- Optimization of intertemporal utility subject to dynamic budget constraint
- Solution characterized by means of associated Bellman equation
- Solvable in closed-form under stringent assumptions about, e.g., utility, return dynamics, market completeness and cost structure
- Existing numerical solution methods are limited by the curse of dimensionality

Transaction costs raise the challenge

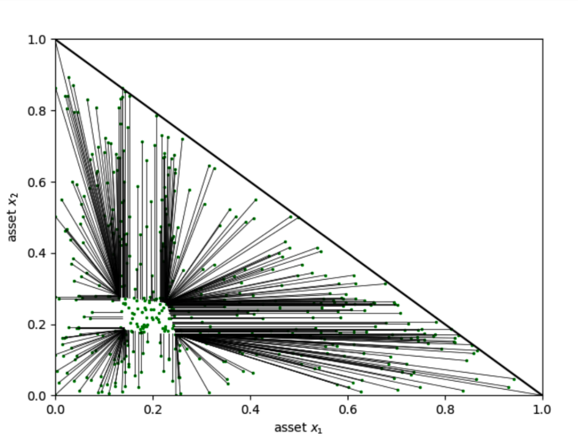
- Budget constraint depends on state vector of dimension not smaller than number of risky assets available for investment:
  - Curse of dimensionality is an even harder challenge
  - No closed-form solution available even in benchmark IID model with consumption and 1 risky asset (Constantinides 79, 86)
  - Existing numerical solution methods hardly applicable with more than 3-4 risky assets
- Intrinsically ”distinct” optimal portfolio behavior over an irregularly-shaped domain
- Not everywhere differentiable optimization problem with varying smoothness and nonlinearity properties over admissible domain

Feature 1: Curse of dimensionality and irregularly-shaped domain



- Budget constraint depends on *D*-dimensional vector of risky assets' wealth shares
- Irregularly-shaped domain with, e.g., no-short selling/borrowing constraints
- A challenge for grid-based function approximation methods

Feature 2: Intrinsically "distinct" portfolio behavior inside admissible domain



- "Distinct" portfolio behavior inside/outside endogenous no-trade region (NTR)
- Naturally varying smoothness properties in neighborhoods of boundary of NTR

Objective

A comprehensive machine learning solution framework for dynamic portfolio choice with proportional transaction costs:

- Scalable
- Accomodating nonlinearities, nondifferentiabilities and irregularly-shaped domains
- ”Tunable-smooth” over different subregions of the domain
- Allowing a quantification of the uncertainty of computed solutions
- Extendable, e.g., to general equilibrium and state-dependent opportunity sets

Benchmark Setting

Dynamic optimal consumption/investment problem

- Trading dates $t \in \{0, 1, \ldots, T\}$ and IID asset returns $(R_f, \boldsymbol{R}_{t+1})$
- State variables $(W_t, \boldsymbol{x}_t) \in \mathcal{D}_t$ for wealth and risky assets' wealth fractions
- Consumption/investment feedback controls $(C_t, \boldsymbol{\delta}_t) \in \mathcal{D}(W_t, \boldsymbol{x}_t)$
- Time-additive optimal consumption investment problem:

$$V(W_0, x_0) := \max_{(C, \boldsymbol{\delta})} \mathbb{E}\left[\sum_{t=1}^{T} u(t, C_t) \middle| W_0, \boldsymbol{x}_0\right] \tag{1}$$

- Not everywhere differentiable dynamic budget constraint:

$$(W_{t+1}, \boldsymbol{x}_{t+1}) = BC(W_t, \boldsymbol{x}_t, C_t, \boldsymbol{\delta}_t; R_f, \boldsymbol{R}_{t+1}) \tag{2}$$

Case I: Not differentiable dynamic budget constraint with PTC

- Risky assets' wealth fraction dynamics:

$$\boldsymbol{x}_{t+1} = \frac{W_t}{W_{t+1}}(\boldsymbol{x}_t + \boldsymbol{\delta}_t) \odot \boldsymbol{R}_{t+1}$$

- Wealth dynamics with homogeneous proportional transaction cost (PTC) $\tau \geq 0$:

$$W_{t+1} = (W_t - C_t - \mathbf{1}^\top(\boldsymbol{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|)W_t)R_f + W_t(\boldsymbol{x}_t + \boldsymbol{\delta}_t)^\top \boldsymbol{R}_{t+1}$$

  - $W_t - C_t - \mathbf{1}^\top(\boldsymbol{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|)W_t$ equals saving net of risky investment and TC
  - PTC induces not differentiable wealth dynamics

- For $\tau = 0$, wealth dynamics only depends on "aggregate control" $\boldsymbol{w}_t := \boldsymbol{x}_t + \boldsymbol{\delta}_t$, i.e., its state space does not scale with the number of risky assets

Case II: Irregularly-shaped domain with proportional transaction costs

- No-short selling and no-borrowing constraints:

$$(C_t, \boldsymbol{\delta}_t) \in \mathcal{D}(W_t, \mathbf{x}_t) = \left\{(C, \boldsymbol{\delta}) : \mathbf{x}_t + \boldsymbol{\delta} \geq \mathbf{0} \text{ and } C/W_t + \mathbf{1}^\top(\mathbf{x}_t + \boldsymbol{\delta} + \tau|\boldsymbol{\delta}|) \leq 1\right\}$$

- No-aggregate leverage constraint:

$$(C_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\mathbf{x}_t) = \left\{(C, \boldsymbol{\delta}) : \mathbf{1}^\top(\mathbf{x}_t + \boldsymbol{\delta}) \leq 1\right\}$$

Recursive form and Bellman equation

1. For some economically motivated terminal utility function $U(\cdot, \cdot)$, let:

$$V_T(W_T, \boldsymbol{x}_T) := U(W_T, \boldsymbol{x}_T) \tag{3}$$

2. Define recursively, for any $t = T-1, \dots, 0$:

$$V_t(W_t, \boldsymbol{x}_t) = \max_{(C_t, \boldsymbol{\delta}_t) \in \mathcal{D}(W_t, \boldsymbol{x}_t)} \left\{ u(t, C_t) + \mathbb{E}[V(W_{t+1}, \boldsymbol{x}_{t+1}) | W_t, \boldsymbol{x}_t] \right\} , \tag{4}$$

   where

$$(W_{t+1}, \boldsymbol{x}_{t+1}) = BC(W_t, \boldsymbol{x}_t, C_t, \boldsymbol{\delta}_t; R_f, \boldsymbol{R}_{t+1})$$

3. Obtain desired value function and optimal controls from last iteration at $t = 0$

Key issues for computational solution framework

1. "Exact" computation of $V_t(\cdot, \cdot)$ over entire domain $\mathcal{D}_t$ is infeasible/inappropriate
2. Need approximation of $V_t(\cdot, \cdot)$ from suitable set of "exact" computations $\{V_t(W_i, \boldsymbol{x}_i)\}_{i=1}^N$
   - Error in approximation of $V_t(\cdot, \cdot)$ needs to be quantifiable
   - Accuracy to bound error propagation in global solution as, e.g., D or T grows
   - Capture varying degrees of nonlinearity/nondifferentiability over domain $\mathcal{D}_t$
   - Efficiency with "exact" computations performed only within irregularly-shaped domain $\mathcal{D}_t$
3. Grid-free probabilistic approximation methods are preferable

Comprehensive Machine Learning Framework

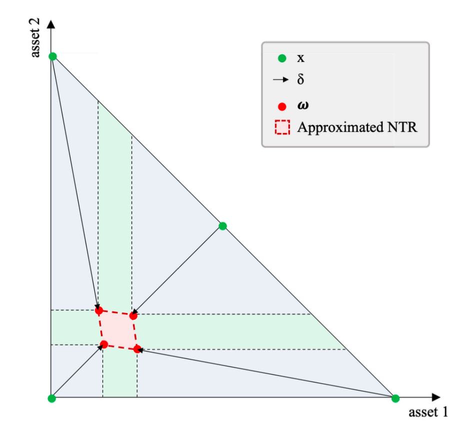
Comprehensive Machine Learning Framework

- Based on recursive grid-free probabilistic approximation of value function $V_t(\cdot, \cdot)$
- Two building blocks:
  1. Accurate description of unknown NTR:
     $$\boldsymbol{\Omega}_t := \{(W_t, \boldsymbol{x}_t) \in \mathcal{D}_t : \boldsymbol{\delta}_t^{opt} = \boldsymbol{0}\}$$
     by means of suitable approximate NTR $\hat{\boldsymbol{\Omega}}_t$
  2. Accurate approximation of $V_t(\cdot, \cdot)|_{\hat{\boldsymbol{\Omega}}_t}$ and $V_t(\cdot, \cdot)|_{\mathcal{D}_t \setminus \hat{\boldsymbol{\Omega}}_t}$ using two different associated Gaussian Process Regressions (GPRs)

Approximate NTR I

- Homothetic portfolio problem (e.g., $u(t, C) = \beta^t u(C)$ for a power utility with RRA $\gamma > 0$)
- Approximate NTR $\hat{\mathbf{\Omega}}_t$ is a D-dimensional convex polytope with 2^D vertices
  - Covers most known NTR in the literature
  - NTR computations as convex hull of its vertices
- Computation of approximate NTR from 2^D computations of optimal portfolio policy

Approximate NTR II



- Sampling points $\{x_1, \ldots, x_{2^D}\}$ in blue region identifies approximate NTR vertices as $\omega_i = x_i + \delta_i^{opt}$

Computation of t^{th}-period approximate NTR

Algorithm:

Input: $(t+1)-$period's (surrogate) value function $\mathcal{V}_{t+1}$

Data: Matrix $\tilde{X}_t := \{\tilde{\mathbf{x}}_{ti}\}_{i=1}^{N} \in \mathcal{D}_t^N$ of $N = 2^D$ points

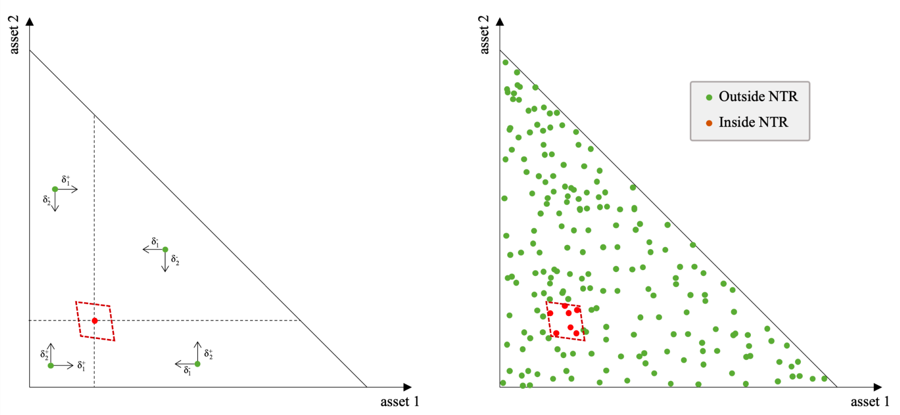
Output: Approximate NTR $\hat{\boldsymbol{\Omega}}_t$

1. **For** $i = 1$ **to** N **do**:
   Compute vertex $\hat{\boldsymbol{\omega}}_{ti} := \tilde{\mathbf{x}}_{\mathbf{ti}} + \hat{\hat{\boldsymbol{\delta}}}_{ti}^{opt}$ *by solving Bellman equation (4) with* $\mathcal{V}_{t+1}$ *as next period's value function*
   end

2. Compute approximate NTR:

$$\hat{\boldsymbol{\Omega}}_t := \left\{ \sum_{i=1}^{n} \lambda_i \hat{\boldsymbol{\omega}}_{ti} : \sum_{i=1}^{N} \lambda_i = 1 \text{ and } \lambda_1, \ldots, \lambda_N \geq 0 \right\}$$

Usefulness of approximate NTR for solving the portfolio problem



- Recovers signs of optimal portfolio policies
- Enables heterogenous sampling of value function training points inside/outside NTR
- Naturally identifies regions of value function non-differentiability/high-nonlinearity

Value function approximation with two GPRs I

- Homothetic portfolio choice problem (e.g., power utility): For any $t = T-1, \ldots, 0$,

$$V_t(W_t, \mathbf{x}_t) = W_t^{1-\gamma} v_t(\mathbf{x}_t) \; ; \; \gamma > 0$$

- Bellman equation: For any $t = T-1, \ldots, 0$:

$$v_t(\mathbf{x}_t) = \max_{(c_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\mathbf{x}_t)} \left\{ u(c_t) + \beta \mathbb{E} \left[\pi_{t+1}^{1-\gamma} v_{t+1}(\mathbf{x}_{t+1}) | \mathbf{x}_t \right] \right\} \tag{5}$$

where $c_t := C_t / W_t$ and

$$\begin{aligned} \pi_{t+1} &:= (1 - c_t - \mathbf{1}^\top (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau |\boldsymbol{\delta}_t|)) R_f + (\mathbf{x}_t + \boldsymbol{\delta}_t)^\top \mathbf{R}_{t+1} \\ \mathbf{x}_{t+1} &= \frac{1}{\pi_{t+1}} (\mathbf{x}_t + \boldsymbol{\delta}_t) \odot \mathbf{R}_{t+1} \end{aligned}$$

Value function approximation with two GPRs II

- Account for intrinsically distinct value function properties inside/outside NTR:

$$v_{1t} := v_t \mathbf{1}_{\hat{\boldsymbol{\Omega}}_t} \; ; \; v_{2t} := v_t \mathbf{1}_{\mathcal{D}_t \setminus \hat{\boldsymbol{\Omega}}_t}$$

- "Disaggregated" Bellman equation: For any $t = T-1, \ldots, 0$:

$$v_{1t}(\mathbf{x}_t) + v_{2t}(\mathbf{x}_t) = \max_{(c_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\mathbf{x}_t)} \left\{ u(c_t) + \beta \mathbb{E} \left[\pi_{t+1}^{1-\gamma} \left(v_{1t+1}(\mathbf{x}_{t+1}) + v_{2t+1}(\mathbf{x}_{t+1}) \right) | \mathbf{x}_t \right] \right\} \tag{6}$$

- Two distinct GPRs for approximating the value function inside/outside NTR:

$$\mathcal{GP}_{1t}(\mathbf{x}_t) + \mathcal{GP}_{2t}(\mathbf{x}_t) = \max_{(c_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\mathbf{x}_t)} \left\{ u(c_t) + \beta \mathbb{E} \left[\pi_{t+1}^{1-\gamma} \left(\mathcal{GP}_{1t+1}(\mathbf{x}_{t+1}) + \mathcal{GP}_{2t+1}(\mathbf{x}_{t+1}) \right) | \mathbf{x}_t \right] \right\}$$

Gaussian process regression: Main Idea I

- Supervised machine learning method with nonparametric Bayesian interpretation

- Treats computed value function as Gaussian process with mean μ and kernel k:

$$v_{it}(x) \sim \mathcal{GP}(\mu_{it}(x), k_{it}(x, x'))$$

  - For any $N \in \mathbb{N}$, Gaussian finite sample (prior) distributions of:

$$(v_{it}(x_1), \ldots, v_{it}(x_N))$$

  - For any $N, M \in \mathbb{N}$, closed-form Gaussian conditional (posterior) distributions of:

$$(v_{it}(x'_1), \ldots, v_{it}(x'_M)) | (v_{it}(x_1), \ldots, v_{it}(x_N))$$

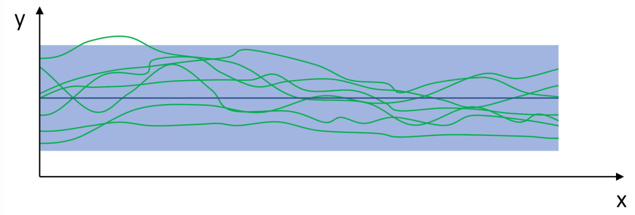
Gaussian process regression: Main idea II

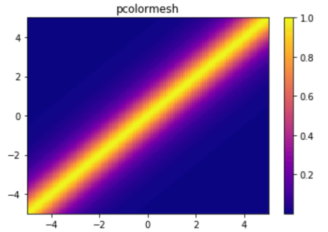
- Approximations of unobserved function values given set of observed function values:
  - Point estimate via $\mathbb{E}[(v_{it}(x'_1), \dots, v_{it}(x'_M))|(v_{it}(x_1), \dots, v_{it}(x_N))]$
  - Uncertainty measurement via $\mathbb{C}ov[(v_{it}(x'_1), \dots, v_{it}(x'_M))|(v_{it}(x_1), \dots, v_{it}(x_N))]$
- $k(x, x')$ captures function values similarity in different regions of the domain:
  - Matern kernel for modelling pronounced function nonlinearities:

$$k_M(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} d(x, x')\right)^{\nu} K_{\nu} \left(\sqrt{2\nu} d(x, x')\right)$$

    with $d(x, x') := (x - x')[\mathrm{diag}(l)]^{-2}(x - x')$
- Hyper-parameters obtained by maximising log likelihood over training sample

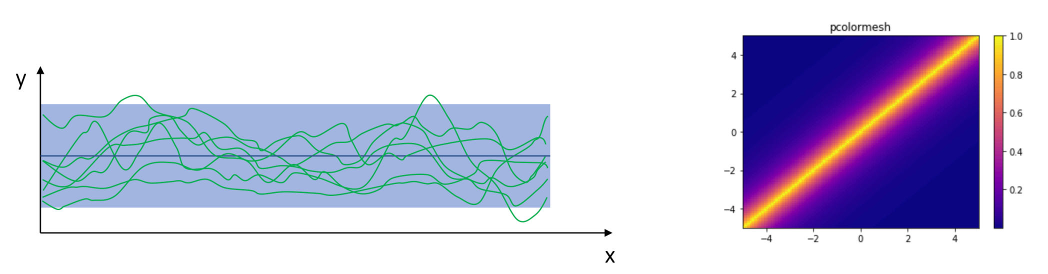
Gaussian process regression: Tunable smoothness with Matern kernel I





- Broader regions of high similarity model a priori smoother value function realizations
- Training of kernel hyperparameters allows learning of relevant degree of smoothness

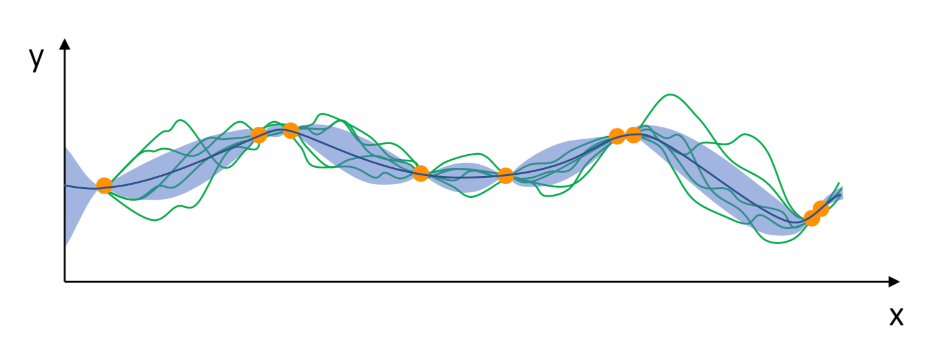
Gaussian process regression: Tunable smoothness with Matern kernel II



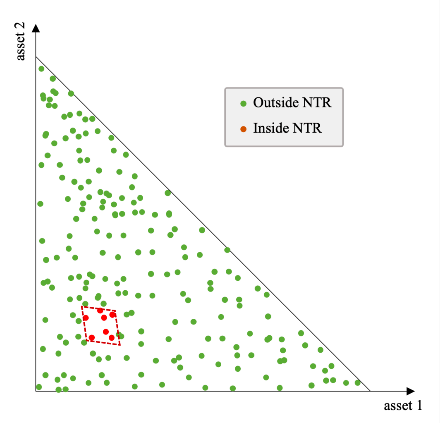
- Smaller regions of high similarity model a priori less smooth value function realizations

- Training of kernel hyperparameters allows learning of relevant degree of smoothness

Gaussian process regression: Posterior distributions



Putting things together



- Randomly generate training samples of points inside/outside NTR
- Train a distinct GPR on each subsample and predict remaining value function and policy values inside/outside NTR

Algorithm

Input: Terminal value function v_T, time horizon T, sampling size N

Output: Surrogate value functions $\{\mathcal{V}_{t-1}\}_{t=1}^{T}$ and approximate NTRs $\{\hat{\boldsymbol{\Omega}}_{t-1}\}_{t=1}^{T}$

1. Set $\mathcal{V}_T = v_T$
2. **For** $t = T$ **to** 1 **do**:
   Compute approximate NTR $\hat{\boldsymbol{\Omega}}_{t-1}$ *using* $\mathcal{V}_t$ *as next period's value function* and sample N points $\boldsymbol{X}_{t-1} = \{\boldsymbol{x}_{t-1i}\}_{i=1}^{N} \in \mathcal{D}_{t-1}^{N}$
      For $i = 1$ **to** N **do**:
      Obtain value function and policy values $(\hat{v}_{t-1i}, \hat{c}_{t-1i}, \hat{\boldsymbol{\delta}}_{(t-1)i})$ *for* $\boldsymbol{x}_{t-1i}$ *by solving Bellman equation (8) using* $\mathcal{V}_t$ *as next period's value function.*
      end
   Fill training sets $\hat{\mathcal{D}}_{1t}$, $\hat{\mathcal{D}}_{2t}$ based on whether $\boldsymbol{x}_{t-1i} \in \hat{\boldsymbol{\Omega}}_{t-1}$ or $\boldsymbol{x}_{t-1i} \notin \hat{\boldsymbol{\Omega}}_{t-1}$ for some i
   Learn from $\hat{\mathcal{D}}_{1t}$, $\hat{\mathcal{D}}_{2t}$ two GPR surrogates $\mathcal{V}_{1(t-1)}, \mathcal{V}_{2(t-1)}$ of $v_{1(t-1)}$, $v_{2(t-1)}$, to finally obtain a surrogate $\mathcal{V}_{t-1} = \mathcal{V}_{1(t-1)} + \mathcal{V}_{2(t-1)}$ of v_{t-1}.
   end

Selected Findings

Findings I: Value function fit

N	Mean (%)	99.9^{th} percentile (%)	Max (%)
100	0.0070	0.101	0.110
300	0.0022	0.0372	0.0456
500	0.0007	0.0137	0.0159
700	0.0004	0.0061	0.0070

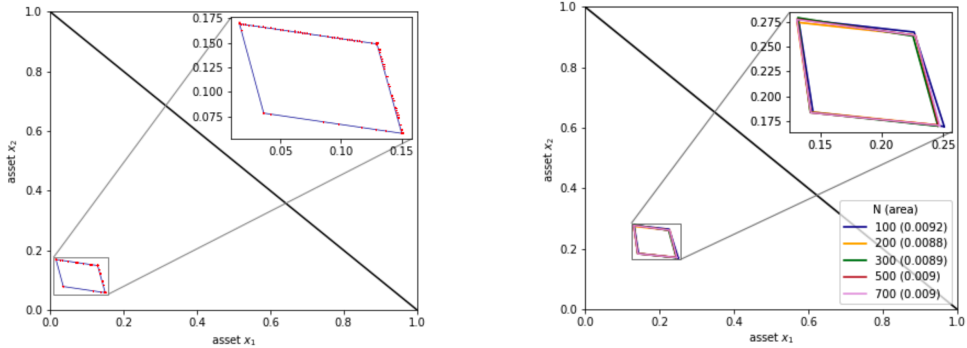
Table 3: Out-of-sample RAE value function fit summary statistics (in percent) of the final iteration (iteration 7) for various sample sizes. The summary statistics were computed using solutions from a model where the NTR was oversampled.

Findings II: Optimal policy fit

	2 risky assets		3 risky assets		5 risky assets	
N	500		700		2000	
REE (%)	mean	max	mean	max	mean	max
δ_1	0.0222	0.334	0.0341	0.2832	0.0377	0.2847
δ_2	0.0179	0.332	0.0308	0.2817	0.0306	0.3239
δ_3			0.0270	0.2833	0.0299	0.3831
δ_4					0.0294	0.2858
δ_5					0.0334	0.4485
c	1.0×10^{-8}	5.3×10^{-7}	1.0×10^{-8}	3.4×10^{-8}	4.4×10^{-6}	8.6×10^{-5}

Table 5: REE statistics for a high-dimensional portfolio optimization problem for final iteration.

Findings III: NTR fit



(a) NTR approximation error between computed policy and policy implied by approximated NTR.

(b) Comparing the final iteration's NTR computed using various sample sizes.

Directions of improvement and extensions

Natural dimensions of direct improvement

- Oversampling of approximate NTR/NTR boundary
- Bayesian reinforcement learning exploiting GPR uncertainty quantification inside/ourside NTR
- Gaussian-Hermite quadrature or Monte Carlo simulation for integral approximations, depending on sampling size and state space dimension

Extensions I: Stochastic opportunity set

- A stochastic opportunity set implies further state variables $\boldsymbol{z}_t$ in the Bellman equation
- $\boldsymbol{z}_t$ impacts the conditional distribution of returns given time $t-$information
- Bellman equation: For any $t = T-1, \ldots, 0$,

$$v_t(\boldsymbol{x}_t, \boldsymbol{z}_t) = \max_{(c_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\boldsymbol{x}_t, \boldsymbol{z}_t)} \left\{ u(c_t) + \beta \mathbb{E}\left[\pi_{t+1}^{1-\gamma} v_{t+1}(\boldsymbol{x}_{t+1}, \boldsymbol{z}_{t+1}) | \boldsymbol{x}_t, \boldsymbol{z}_t \right] \right\} \tag{7}$$

where $c_t := C_t / W_t$ and

$$\begin{aligned} \pi_{t+1} &:= (1 - c_t - \mathbf{1}^\top (\boldsymbol{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|)) R_f + (\boldsymbol{x}_t + \boldsymbol{\delta}_t)^\top \boldsymbol{R}_{t+1} \\ \boldsymbol{x}_{t+1} &= \frac{1}{\pi_{t+1}} (\boldsymbol{x}_t + \boldsymbol{\delta}_t) \odot \boldsymbol{R}_{t+1} \end{aligned}$$

- $\boldsymbol{z}_t-$state dynamics may be defined over an irregularly-shaped state space in, e.g., regime-switching/mixture models

Extensions II: Stochastic frictions

- Stochastic frictions may imply further state variables $(\boldsymbol{z}_t, W_t)$ in the Bellman equation
- They impact the budget constraint of the portfolio problem
- Bellman equation: For any $t = T-1, \ldots, 0$,

$$v_t(\boldsymbol{x}_t, \boldsymbol{z}_t, W_t) = \max_{(c_t, \boldsymbol{\delta}_t) \in \mathcal{D}(\boldsymbol{x}_t)} \left\{ u(c_t) + \beta \mathbb{E}\left[\pi_{t+1}^{1-\gamma} v_{t+1}(\boldsymbol{x}_{t+1}, \boldsymbol{z}_{t+1}, W_{t+1}) | \boldsymbol{x}_t, \boldsymbol{z}_t, W_t \right] \right\}$$

where $c_t := C_t / W_t$ and

$$\begin{aligned} \pi_{t+1} &:= (1 - c_t - \mathbf{1}^\top (\boldsymbol{x}_t + \boldsymbol{\delta}_t + \tau(\boldsymbol{x}_t, \boldsymbol{z}_t, W_t) |\boldsymbol{\delta}_t|)) R_f + (\boldsymbol{x}_t + \boldsymbol{\delta}_t)^\top \boldsymbol{R}_{t+1} \\ \boldsymbol{x}_{t+1} &= \frac{1}{\pi_{t+1}} (\boldsymbol{x}_t + \boldsymbol{\delta}_t) \odot \boldsymbol{R}_{t+1} \end{aligned}$$

- Transaction cost function may be defined on irregularly-shaped domain and feature nonlinearities/nondifferentiability (e.g., with threshold-type structures)

Extensions III: Very high-dimensional asset space

- Gaussian process approximations with active subspaces (Bilionis Scheidegger (2019)):
  - Reduce model dimension via a projection on lower-dimensional linear manifold of the input space having maximal response variation
- Geometric deep learning (Bronstein Bruna Cohen Veličković (2021)):
  - Exploit geometric model properties such as symmetries, to build improved/scalable learning architectures
- Scalable deep kernels (Wilson Hu Salakhutdinov Xing (2016)):
  - Combine structural properties of deep learning with nonparametric flexibility of kernels
  - Closed-form kernels as drop-in replacements for standard kernels, with benefits in expressive power and scalability

Conclusions

Conclusions

Scalable machine learning solution framework for dynamic portfolio choice with TC:

- Accomodating nonlinearities, nondifferentiabilities and irregularly-shaped domains
- ”Tunable-smooth” and allows uncertainty quantification

Relies on accurate probabilistic approximations of value and policy functions:

- A novel accurate approximation of the irregularly-shaped NTR
- Two accurate GPR approximations of the value function inside/outside the NTR
- Bayesian reinforcement learning

Thank you!