Universität Zürich[UZH]

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

Luca Gaegauf
University of Zurich

Simon Scheidegger
University of Lausanne

Fabio Trojani
University of Geneva

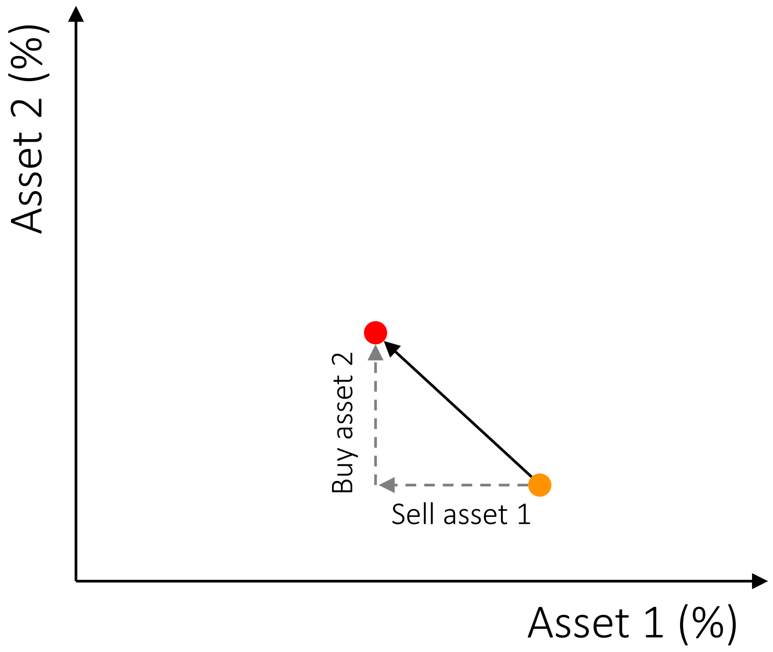
Agenda

1) Motivation
2) Portfolio choice model
3) Algorithm
4) Evaluation
5) Economic results

1) Motivation

Proportional transaction costs induce NTR

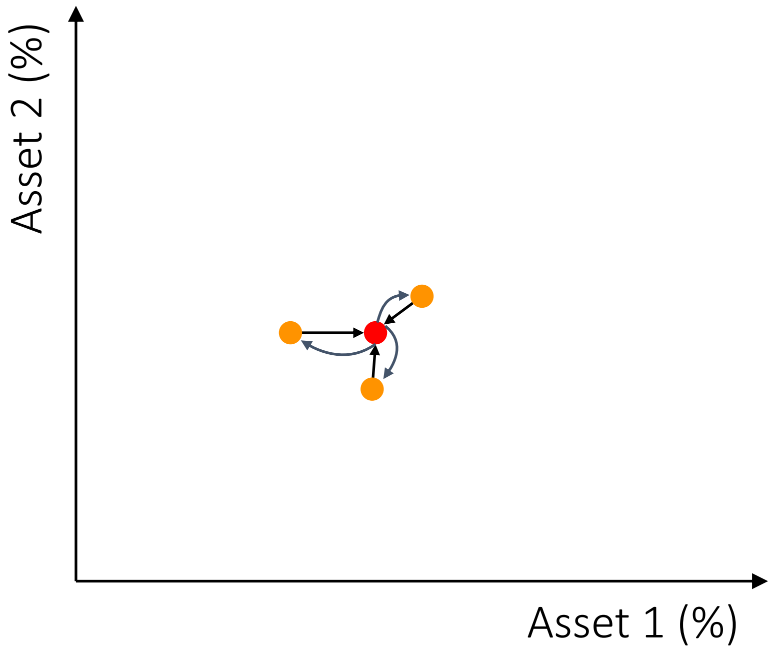
- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.



1) Motivation

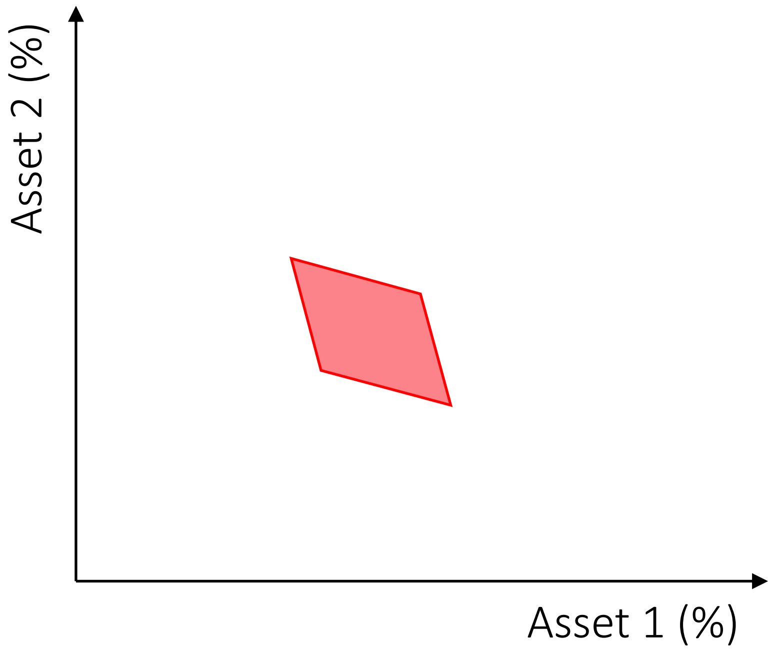
Proportional transaction costs induce NTR

- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.



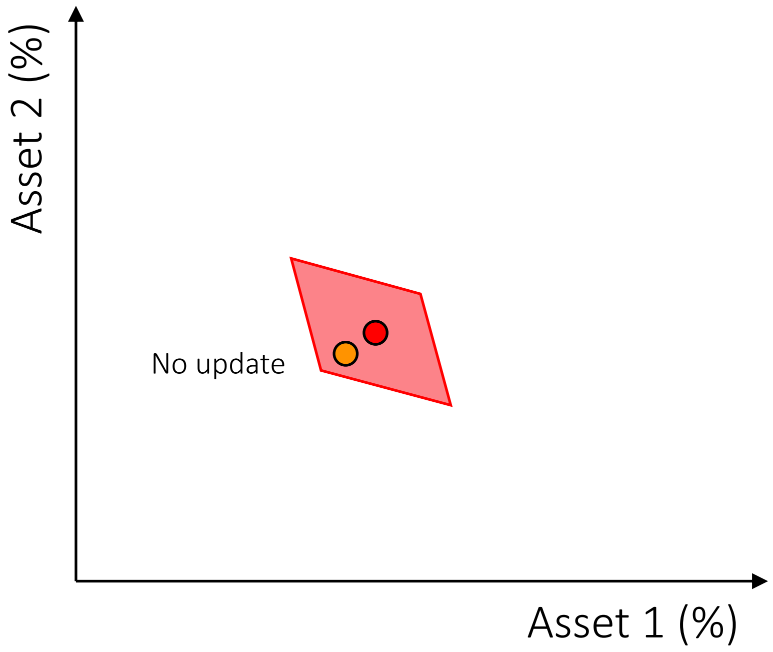
1) Motivation
Proportional transaction costs induce NTR

- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.
  → Proportional transaction costs.
- **Key finding**: no-trade-region (NTR) (Magill and Constantinides, 1976; Constantinides, 1979, 1986).



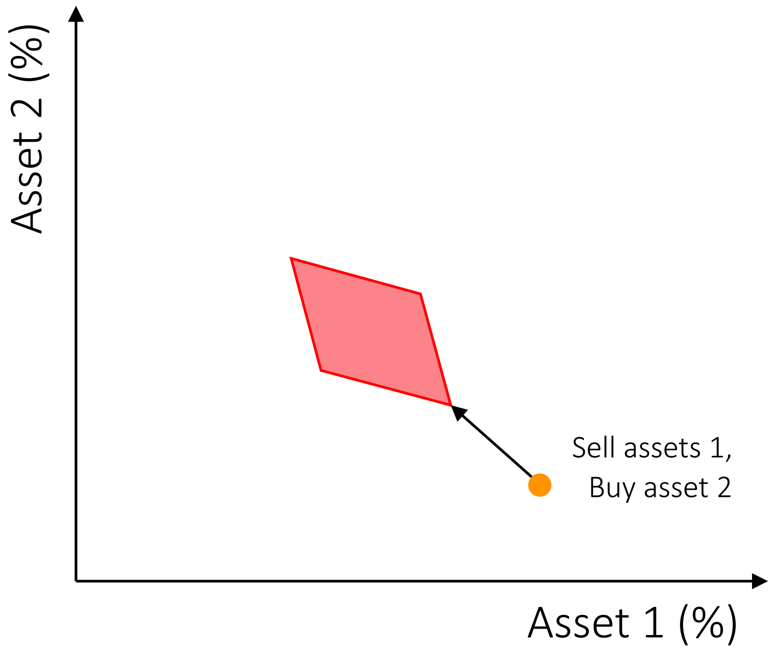
1) Motivation
Proportional transaction costs induce NTR

- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.
  → Proportional transaction costs.
- **Key finding**: no-trade-region (NTR) (Magill and Constantinides, 1976; Constantinides, 1979, 1986).



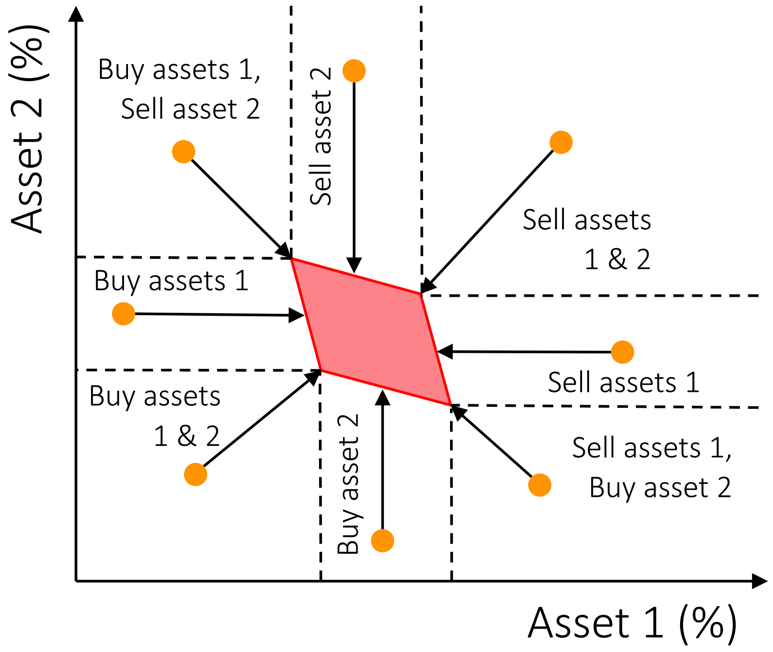
1) Motivation
Proportional transaction costs induce NTR

- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.
  → Proportional transaction costs.
- **Key finding**: no-trade-region (NTR) (Magill and Constantinides, 1976; Constantinides, 1979, 1986).



1) Motivation
Proportional transaction costs induce NTR

- Continuous-time portfolio optimization (*e.g.*, Merton, 1971) suggests agents should update portfolio holdings infinitely often to maintain optimal allocation.
  → Proportional transaction costs.
- **Key finding**: no-trade-region (NTR) (Magill and Constantinides, 1976; Constantinides, 1979, 1986).



Dybvig & Pezzo (2020)

2) Dynamic portfolio optimization model
based on Schober, Valentin, & Pflüger (2022)

Stochasticity	Irregular geometry of state space	Strong nonlinearities and/or kinks in equilibrium functions	High-dimensional state space
Continuous risky asset returns	Borrowing and shorting constraints	Proportional transaction costs	State space increases linearly with the number of assets

2) Dynamic portfolio optimization model
The investor's dynamic problem

- Agent maximizes:

$$V_t(W_t, \mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t W_t) + \beta \mathbb{E}_t [V_{t+1}(W_{t+1}, \mathbf{x}_{t+1})]\}$$

for $t < T$,

State: Current wealth W_t $(W_0 > 0)$
and risky asset holdings
$\mathbf{x}_t = (x_{1,t}, \ldots, x_{M,t})^\top \in [0, 1]^M$.

Policy: Consumption c_t
and risky asset reallocation
$\boldsymbol{\delta}_t = (\delta_{1,t}, \ldots, (\delta_{M,t})^\top \in [-1, 1]^M$,
with: $\delta_{i,t} > 0$: Buy i^{th} asset.
$\delta_{i,t} < 0$: Sell i^{th} asset.

2) Dynamic portfolio optimization model

The investor's dynamic problem

- Agent maximizes:

$$V_t(W_t, \mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t W_t) + \beta \mathbb{E}_t [V_{t+1}(W_{t+1}, \mathbf{x}_{t+1})]\}$$

for $t < T$,

subject to:

Risky assets pay $R_t \sim \mathcal{LN}$.

$$W_{t+1} = (\mathbf{R}_t^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t)\, W_t,$$

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

$$b_t = 1 - \mathbf{1}^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t.$$

Bond pays risk-free rate R_f.

Proportional transaction cost $\tau \in [0, 1]$ paid on each risky asset: $\tau|\delta_{i,t}|$ for $i \in [1, ..., M]$.

2) Dynamic portfolio optimization model
The investor's dynamic problem

- Agent maximizes:

$$V_t(W_t, \mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t W_t) + \beta \mathbb{E}_t [V_{t+1}(W_{t+1}, \mathbf{x}_{t+1})]\}$$

for $t < T$,

subject to:

$$W_{t+1} = (\mathbf{R}_t^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t)\, W_t,$$

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

$$b_t = 1 - \mathbf{1}^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t.$$

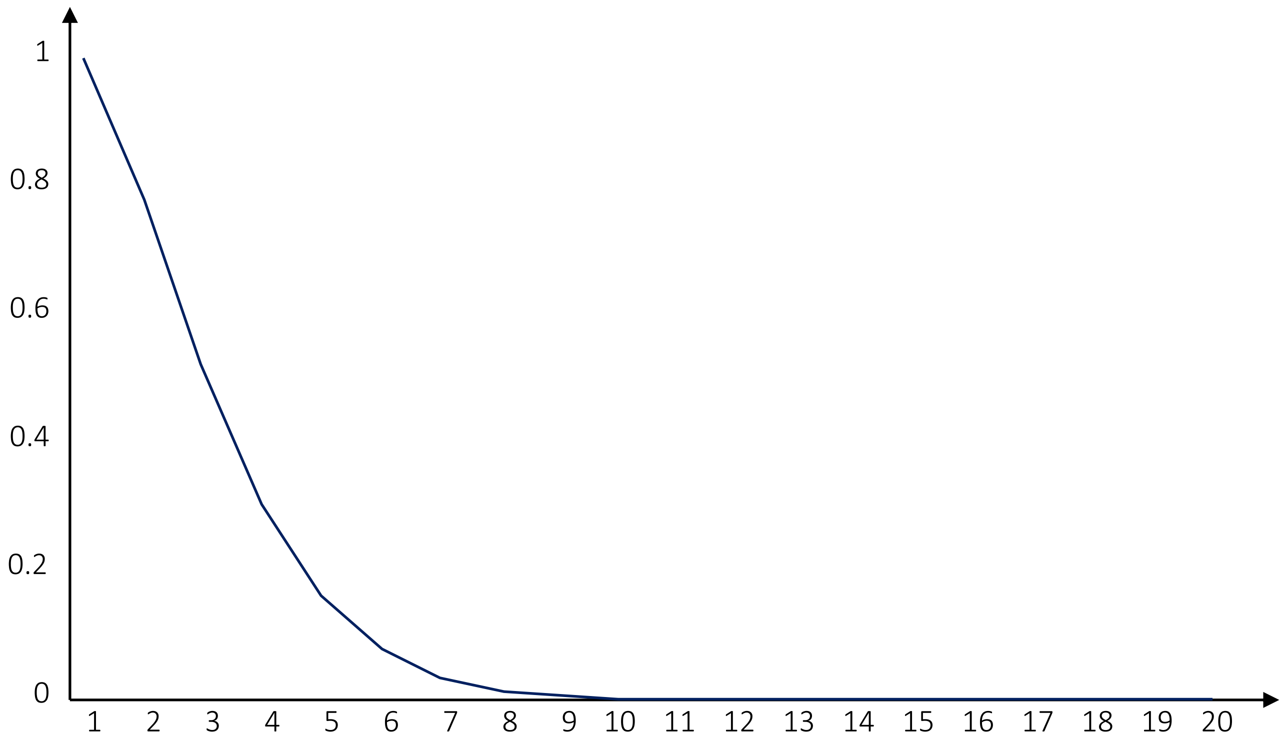
$$\left.\begin{array}{l} \boldsymbol{\delta}_t \geq -\mathbf{x}_t \\ b_t \geq 0 \end{array}\right\} \text{No shorting}$$

No borrowing

$$\mathbf{1}^\top \cdot x_t \leq 1$$

for $t \in [0, T]$

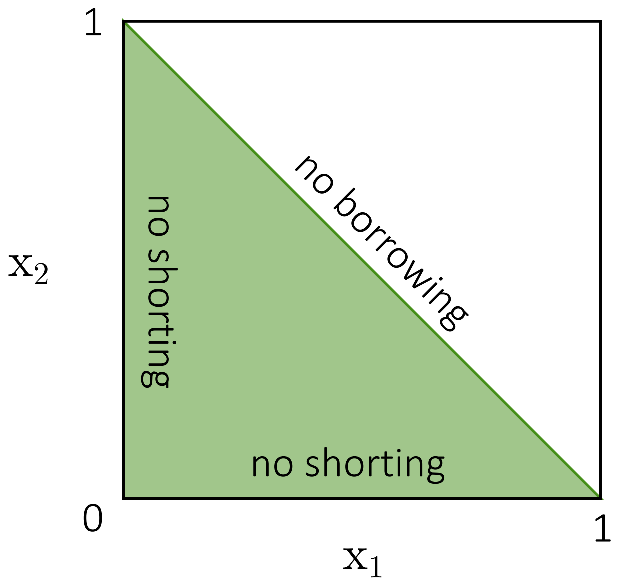
1) Motivation
High-dimensional irregular state space geometries

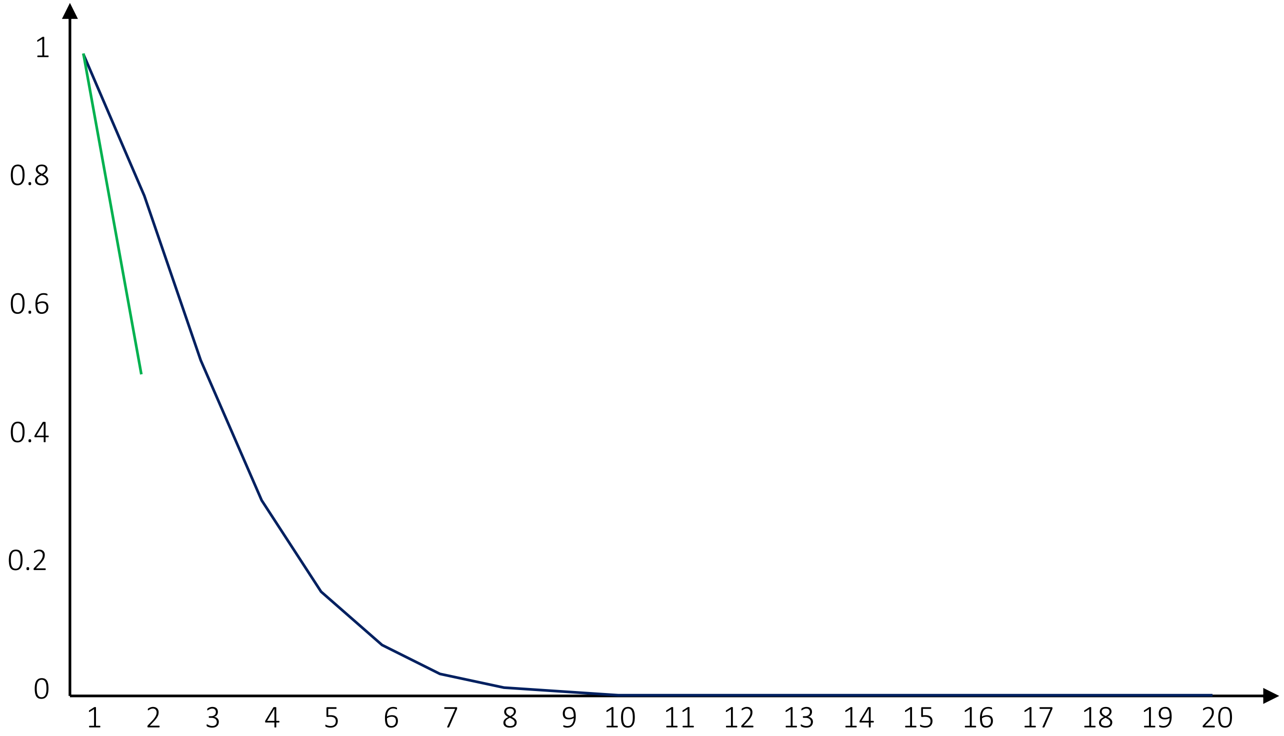


Stochasticity | Irregular geometry of state space | Strong nonlinearities and/or kinks in equilibrium functions | High-dimensional state space

1) Motivation
High-dimensional irregular state space geometries

D = 2

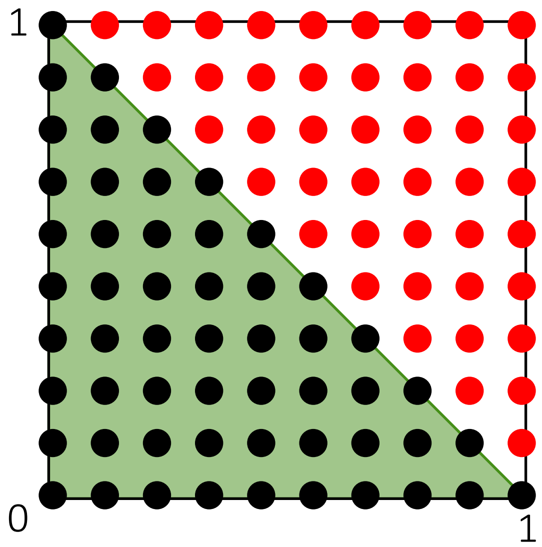


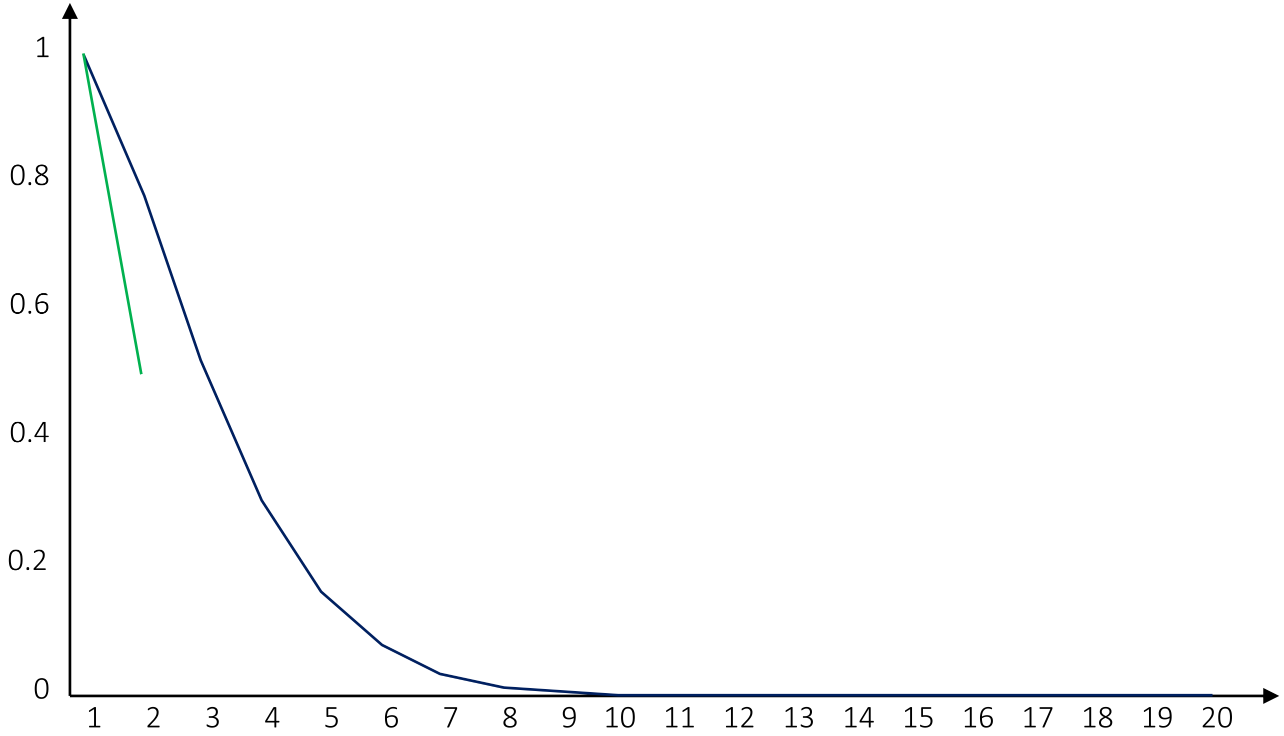


Stochasticity | Irregular geometry of state space | Strong nonlinearities and/or kinks in equilibrium functions | High-dimensional state space

1) Motivation
High-dimensional irregular state space geometries

D = 2



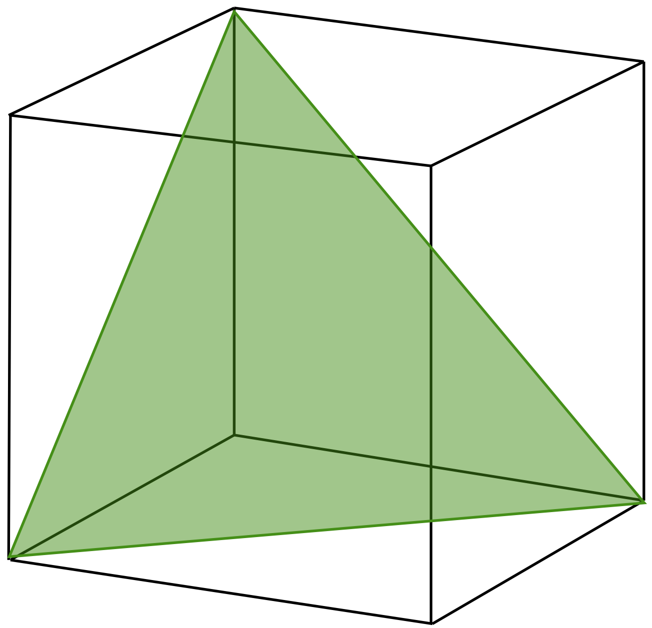


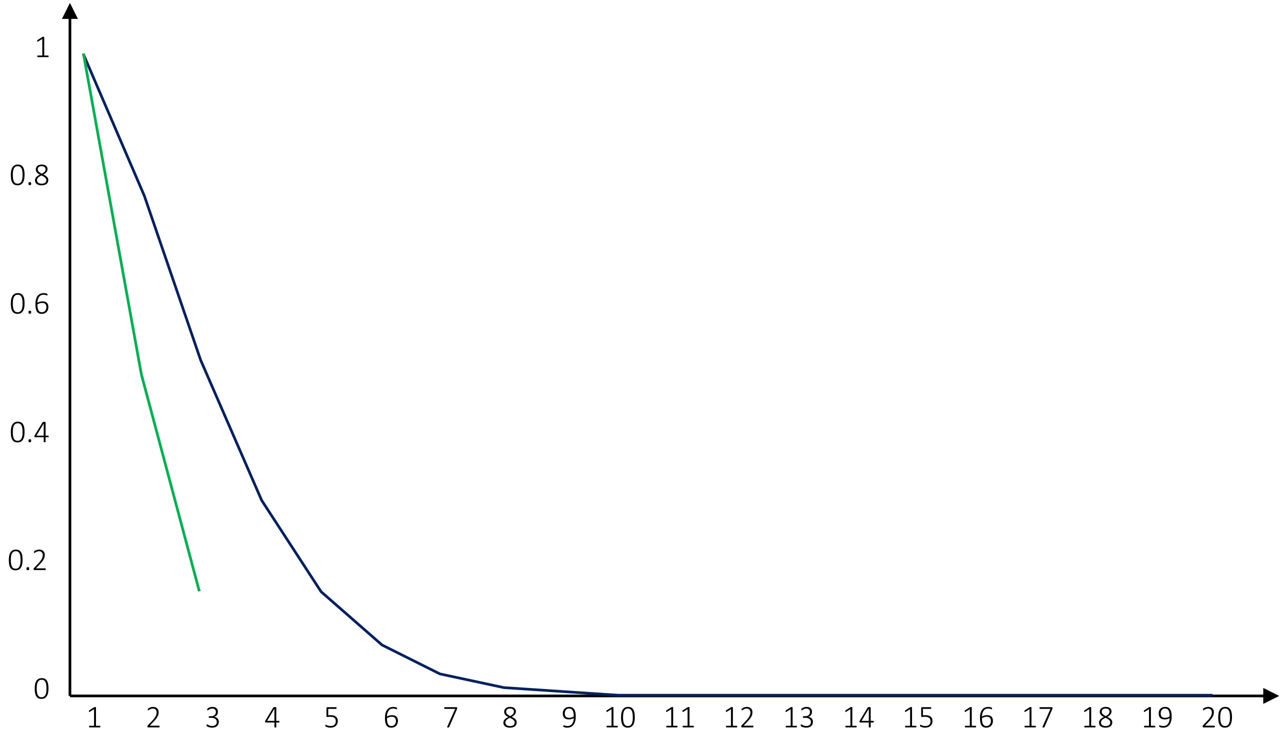
Stochasticity | Irregular geometry of state space | Strong nonlinearities and/or kinks in equilibrium functions | High-dimensional state space

1) Motivation

High-dimensional irregular state space geometries

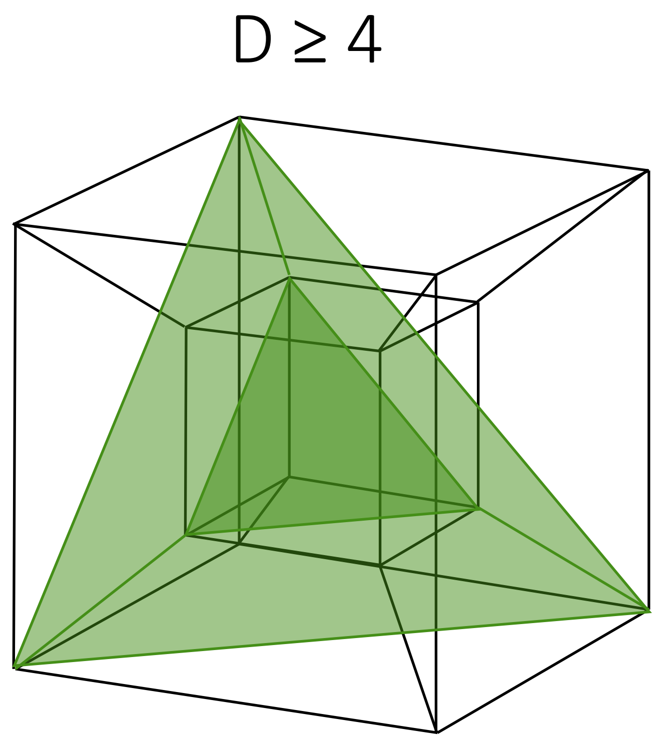
D = 3

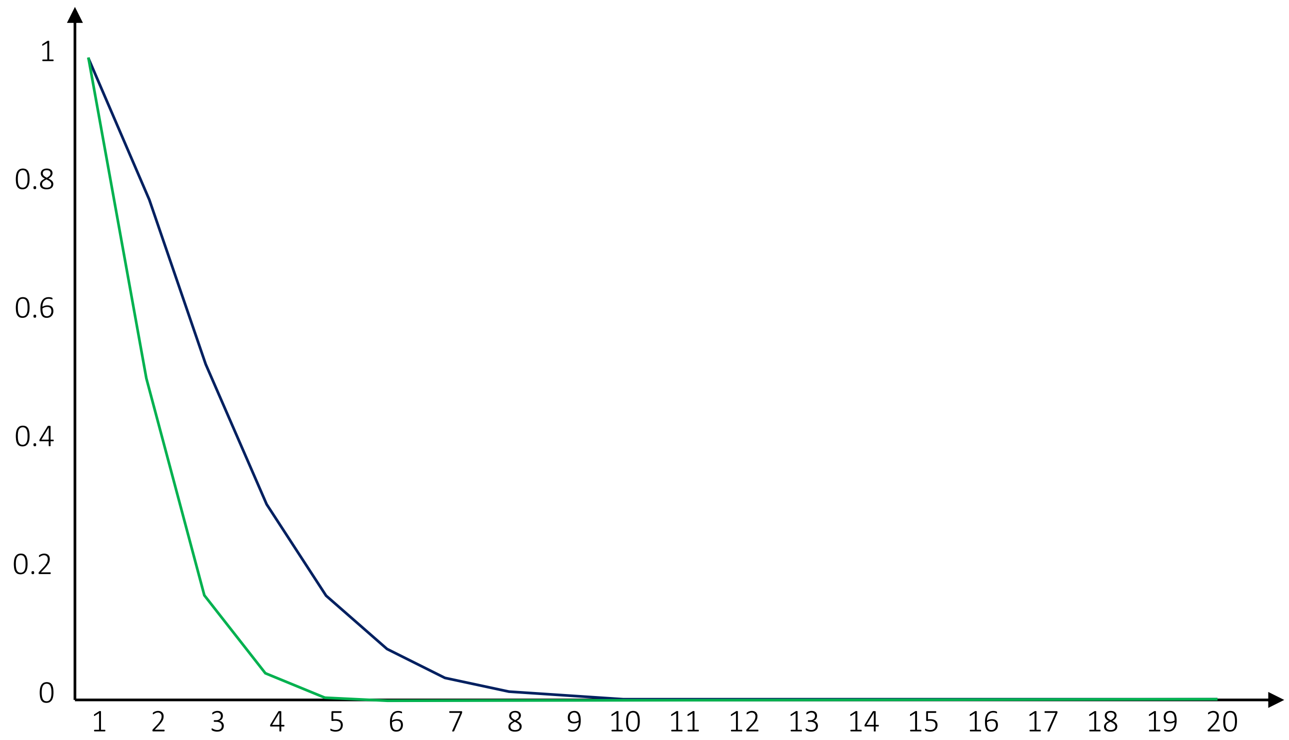


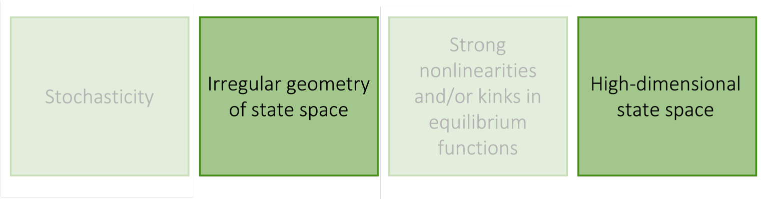


Stochasticity | Irregular geometry of state space | Strong nonlinearities and/or kinks in equilibrium functions | High-dimensional state space

1) Motivation
High-dimensional irregular state space geometries







3) Algorithm
Main components

Approximate	Optimize	Data
Gaussian process regression	Bellman with numerical solver	
NTR approximation		

3) Algorithm
Features of Gaussian process regressions

- Grid-free,
- Non-linear,
- “Tune-ably” smooth,
- Provides predictions,
- Provides information about uncertainty of prediction,
- Scalable,
- Other technical features.

3) Algorithm
Gaussian process regressions

A Gaussian Process defines a distribution over functions:

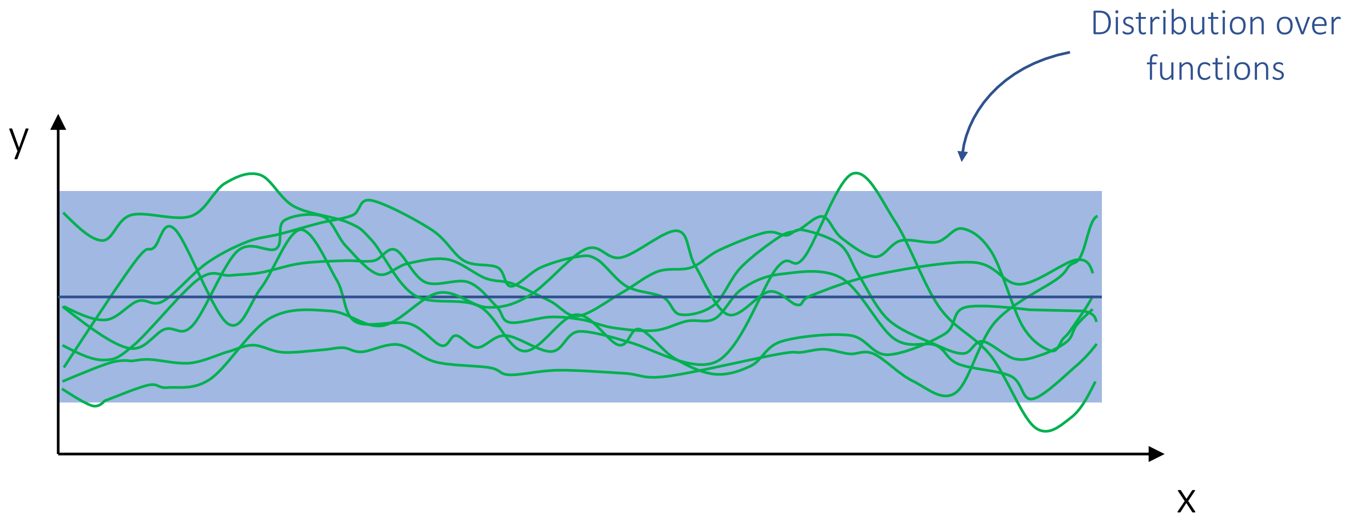
- Mean function: $\mathrm{m}(\mathrm{x})$,
- Covariance function (kernel): $\mathrm{k}(\mathrm{x},\mathrm{x}')$.
  - The choice of kernel reflects prior beliefs about the data.
  - Measures the similarity between two points.

Matern kernel

$$\mathcal{K}(\mathrm{x},\mathrm{x}'|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)$$

3) Algorithm

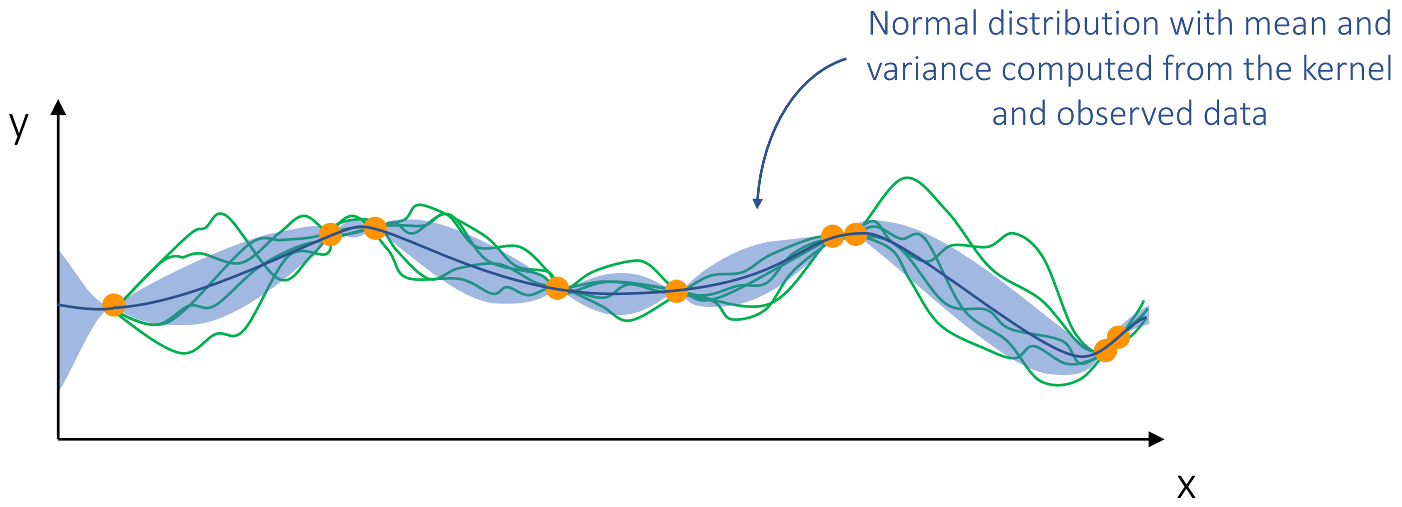
GPR: Sampling functions from the prior



$$\mathcal{K}(\text{x},\text{x'}|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\text{x},\text{x'})\right)^{\nu}K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\text{x},\text{x'})\right)$$

3) Algorithm
GPR: Sampling functions from the posterior

- Use training data to update beliefs about the unknown function.
- Predictions at new points are given by the posterior distribution.



$$\mathcal{K}(\mathrm{x},\mathrm{x}'|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)$$

3) Algorithm
Gaussian process regression

- Agent maximizes:

$$v_t(\mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t) + \beta \mathbb{E}_t \left[\pi_{t+1}^{1-\gamma}\, \mathcal{GP}_{t+1}(\mathbf{x}_{t+1})\right]\} \text{ for } t < T,$$

$$\boxed{\mathcal{GP} : \mathbb{R}^M \rightarrow \mathbb{R} : \mathbf{x} \rightarrow \mathcal{GP}(\mathbf{x})}$$

subject to:

$$\pi_{t+1} := (\mathbf{R}_t^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t),$$

$$b_t = 1 - \mathbf{1}^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t.$$

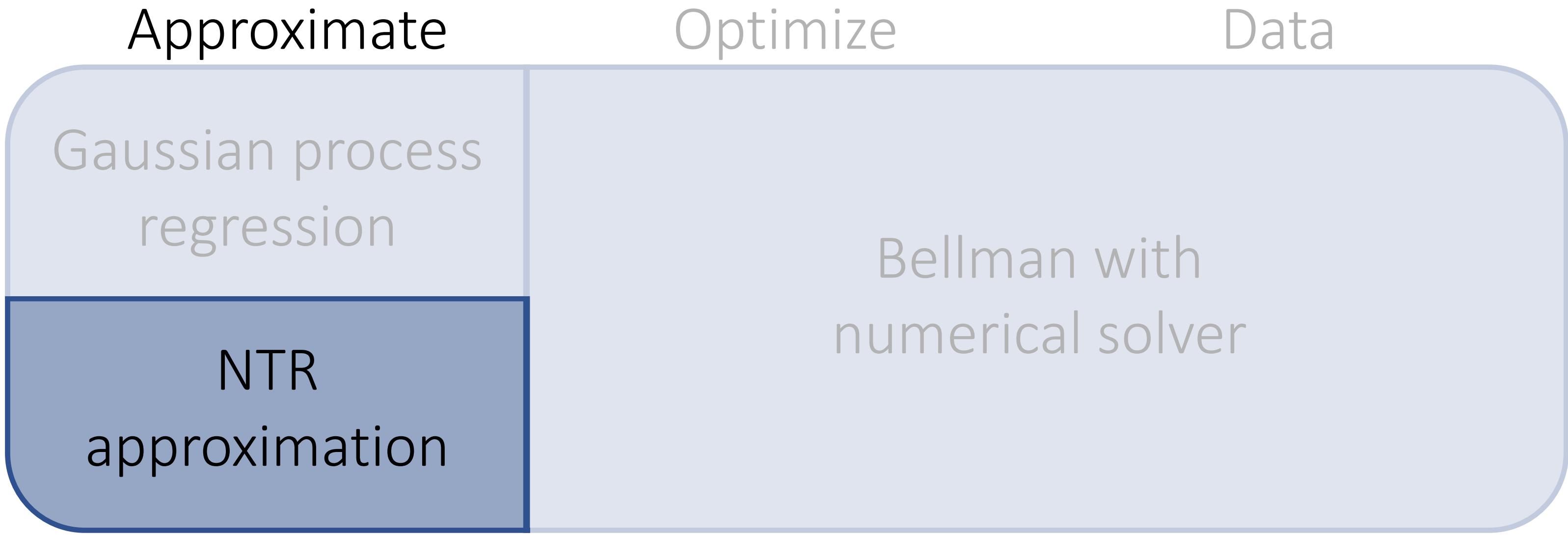
$$\mathbf{x}_{t+1} = \frac{((\mathbf{x}_t + \boldsymbol{\delta}_t) \odot \mathbf{R}_t)}{\pi_{t+1}}$$

$$\boldsymbol{\delta}_t \geq -\mathbf{x}_t$$

$$b_t \geq 0$$

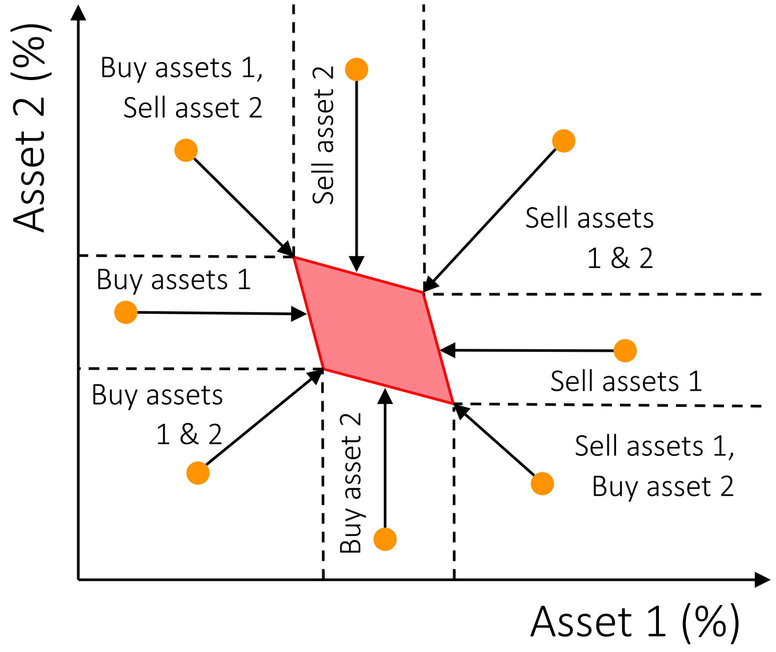
$$\mathbf{1}^\top \cdot x_t \leq 1 \qquad \text{for } t \in [0, T]$$

3) Algorithm
Components: NTR approximation



3) Algorithm
Approximating the NTR

- Pre-determine sign of optimal δ-policy,
- Compute initial guesses,
- Use multiple GPRs to segment state space,
- Data sampling (heuristic or algorithmic).



Dybvig & Pezzo (2020)

4) Evaluation
Euler errors and value function fit

- Relative Euler errors:

$$\tilde{\varepsilon}_{c_t}(\mathbf{x}_t) = \frac{\beta \mathbb{E}_t \left[R_f \pi_{t+1}^{-\gamma} \left((1-\gamma) v_{t+1}(\mathbf{x}_{t+1}) - \nabla_{\mathbf{x}_{t+1}} v_{t+1}(\mathbf{x}_{t+1}) \mathbf{x}_{t+1} \right) \right]^{-\frac{1}{\gamma}}}{c_t} - 1 = 0$$

$$\tilde{\varepsilon}_{\delta_{i,t}}(\mathbf{x}_t) = \frac{\left[\frac{\beta \mathbb{E}_t \left[R_{i,t} \pi_{t+1}^{-\gamma} \left((1-\gamma) v_{t+1}(\mathbf{x}_{t+1}) - \nabla_{\mathbf{x}_{t+1}} v_{t+1}(\mathbf{x}_{t+1}) (\mathbf{e}_i - \mathbf{x}_{t+1}) \right) \right]}{-\frac{\partial c_t}{\partial \delta_{i,t}}} \right]^{-\frac{1}{\gamma}}}{c_t} - 1 = 0$$

- Value function fit:

$$RAE(v, \hat{v}) = \frac{|v - \hat{v}|}{|v|},$$

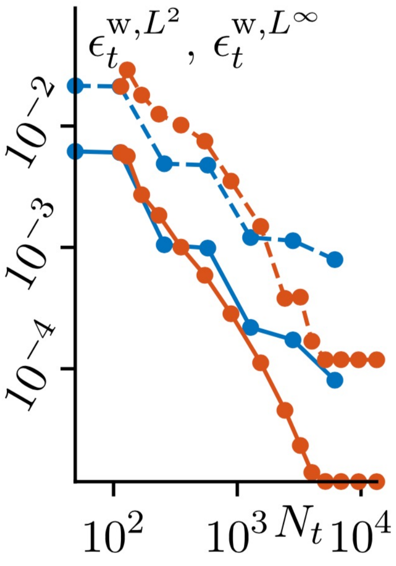
4) Evaluation
Relative Euler errors of consumption: many assets

	2 risky assets		3 risky assets		5 risky assets	
N	500		700		2000	
REE ($\times 10^{-2}$)	mean	max	mean	max	mean	max
δ_1	0.0222	0.334	0.0341	0.283	0.0377	0.284
δ_2	0.0179	0.332	0.0308	0.282	0.0306	0.324
δ_3			0.0270	0.283	0.0299	0.383
δ_4					0.0294	0.286
δ_5					0.0332	0.449
c	0.00000001	0.00000053	0.00000001	0.0000003	0.000004	0.000086

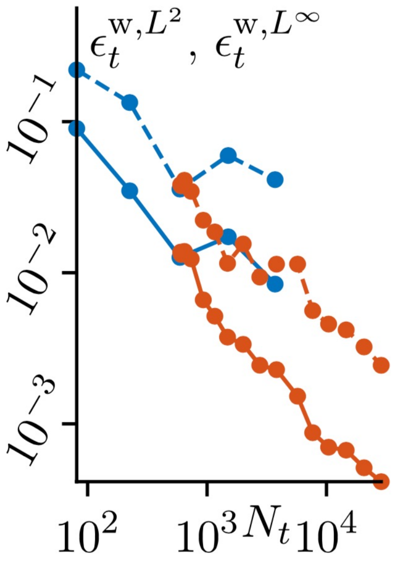
4) Evaluation

Relative Euler errors of consumption: many assets

(b) $d = 2$

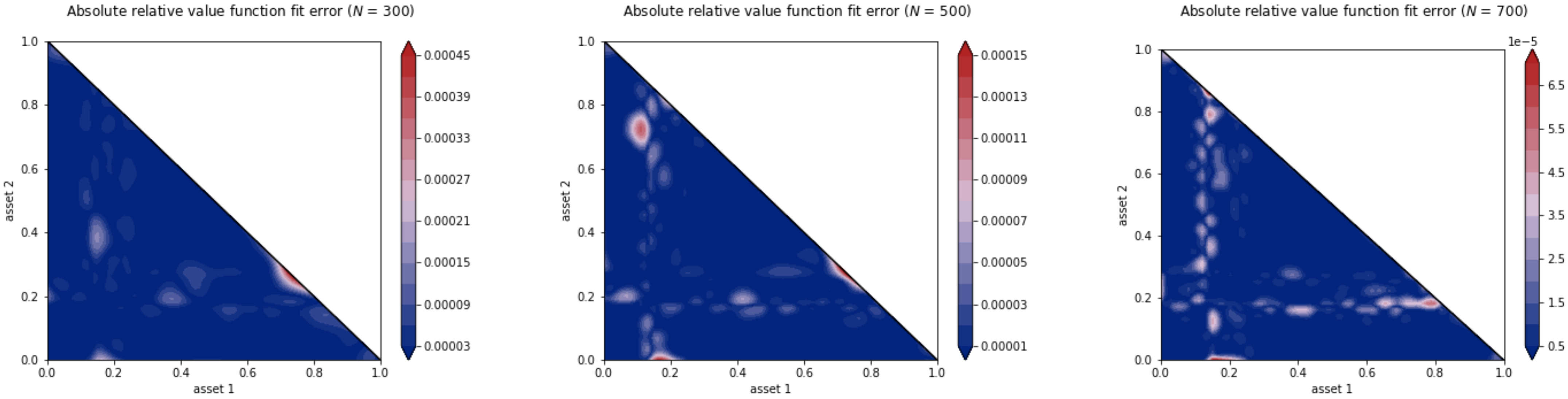


(c) $d = 3$



	2 risky assets		3 risky assets		5 risky assets	
N	500		700		2000	
REE ($\times 10^{-2}$)	mean	max	mean	max	mean	max
δ_1	0.0222	0.334	0.0341	0.283	0.0377	0.284
δ_2	0.0179	0.332	0.0308	0.282	0.0306	0.324
δ_3			0.0270	0.283	0.0299	0.383
δ_4					0.0294	0.286
δ_5					0.0332	0.449
c	0.00000001	0.00000053	0.00000001	0.0000003	0.000004	0.000086

4) Evaluation
Value function fit across sample size



	N	Mean	99.9^{th} percentile	Max
RAE	300	0.000022	0.000372	0.000456
	500	0.000007	0.000137	0.000159
	700	0.000004	0.000061	0.000070

5) Results: Size of the NTR
NTR shrinks as agents can trade more assets

Relative size of NTR in N-simplex

	Risky assets		
t	2	3	5
0	1.80	0.37	0.008
1	1.93	0.38	0.009
2	2.08	0.38	0.006
3	2.24	0.37	0.005
4	2.40	0.37	0.003
5	2.53	0.33	0.008
6	1.98	0.20	0.003
Simplex volume	0.5	0.167	0.00833

Liquidity premia

Risky assets	Liquidity premium (%)
2	3.08
3	1.15
5	1.60

5) Results: Lifetime utility loss

Incorrect handling of transaction costs detriments agents

Risky assets	% of lifetime utility lost	
	No TC	Quadratic TC
2	0.20	0.15
3	12.54	9.49
5	29.70	23.08

References

- Cai, Y., Judd, K. L., and Xu, R. (2013). Numerical solution of dynamic portfolio optimization with transaction costs. Technical report, National Bureau of Economic Research.
- Constantinides, G. M. (1979). Multiperiod consumption and investment behavior with convex transactions costs. Management Science, 25(11):1127–1137.
- Constantinides, G. M. (1986). Capital market equilibrium with transaction costs. In Theory Of Valuation, pages 207–227. World Scientific.
- Dybvig, P. H. and Pezzo, L. (2020). Mean-variance portfolio rebalancing with transaction costs. Available at SSRN 3373329.
- Magill, M. J. and Constantinides, G. M. (1976). Portfolio selection with trans- actions costs. Journal of economic theory, 13(2):245–263.
- Mei, X., DeMiguel, V., and Nogales, F. J. (2016). Multiperiod portfolio optimization with multiple risky assets and general transaction costs. Journal of Banking & Finance, 69:108–120.
- Muthuraman, K. and Zha, H. (2008). Simulation-based portfolio optimization for large portfolios with transaction costs. Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, 18(1):115–134.

Appendix

Executive summary

Economic model:

- Dynamic portfolio optimization,
- Finite-horizon,
- Discrete-time,
- 1 agent with CRRA utility,
- 1 risk-free asset,
- 2–5 risky assets,
- Proportional transaction costs,
- Consumption,
- No shorting and no borrowing,
- No production, labor, or additional endowments.

Solution method:

- Gaussian process regression,
- No-trade-region (NTR) approximation.

Evaluation:

- Evaluated on value-function fit statistics, Euler errors, and NTR approximation errors,
- Similar or better performance with fewer points than competing methods.

Results:

- Systematic analysis of NTR,
- Monte Carlo experiments.

Investor's problem

- Agent maximizes:

Agent's time discount factor ($\beta < 1$)

Agent lives for T periods

$$(\delta_t,\ c_t) \in \arg\max_{\delta_t, c_t} \mathbb{E}_0 \left[\sum_{t=0}^{T} \beta^t u(c_t W_t) \right]$$

Agent's utility function (CRRA)

Investor’s problem

- Agent maximizes:

$$(\boldsymbol{\delta}_t,\ c_t) \in \arg\max_{\boldsymbol{\delta}_t, c_t} \mathbb{E}_0 \left[\sum_{t=0}^{T} \beta^t u(c_t W_t) \right]$$

subject to:

$$W_{t+1} = (\mathbf{R}_t^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t)\ W_t,$$

$$b_t = 1 - \mathbf{1}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \underbrace{\tau|\boldsymbol{\delta}_t|}_{\text{Proportional transaction cost}}) - c_t.$$

State: $\{W_t,\ \mathbf{x}_t\}$

Current wealth: W_t (initial endowment $W_0 > 0$).

Current asset holdings: $\mathbf{x}_t = (x_{1,t},\ \ldots,\ x_{M,t})^{\top} \in [0,\ 1]^M$

Policy: $\{\boldsymbol{\delta}_t,\ c_t\}$

Consumption: c_t.

Bond holding: $b_t \in [0,\ 1]$.

Pays $R_f = \exp(r)$.

Asset reallocation: $\boldsymbol{\delta}_t = (\delta_{1,t},\ \ldots,\ \delta_{M,t})^{\top} \in [-1,\ 1]^M$

Pays $\mathbf{R} = (R_1,\ \ldots,\ R_M)^{\top} \sim LN.$

$\delta_{i,t} > 0$: Buy i^{th} asset.

$\delta_{i,t} < 0$: Sell i^{th} asset.

Proportional transaction cost $\tau \in [0,\ 1]$ paid on each risky asset: $\tau|\delta_{i,t}|$ for $i \in [1,\ \ldots,\ M]$..

Investor’s problem

- Agent maximizes:

$$(\boldsymbol{\delta}_t,\ c_t) \in \arg\max_{\boldsymbol{\delta}_t, c_t} \mathbb{E}_0 \left[\sum_{t=0}^{T} \beta^t u(c_t W_t) \right]$$

subject to:

$$W_{t+1} = (\mathbf{R}_t^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\ b_t)\ W_t,$$

$$b_t = 1 - \mathbf{1}^\top \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t.$$

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

$$\boldsymbol{\delta}_t W_t \geq -\mathbf{x}_t W_t$$

$$b_t W_t \geq 0$$

$$\mathbf{1}^\top \cdot x_t \leq 1 \qquad \text{for } t \in [0,\ T]$$

State: $\{W_t,\ \mathbf{x}_t\}$

Current wealth: W_t (initial endowment $W_0 > 0$).

Current asset holdings: $\mathbf{x}_t = (x_{1,t},\ \ldots,\ x_{M,t})^\top \in [0,\ 1]^M$

Policy: $\{\boldsymbol{\delta}_t,\ c_t\}$

Consumption: c_t.

Bond holding: $b_t \in [0,\ 1]$.

Pays $R_f = \exp(r)$.

Asset reallocation: $\boldsymbol{\delta}_t = (\delta_{1,t},\ \ldots,\ \delta_{M,t})^\top \in [-1,\ 1]^M$

Pays $\mathbf{R} = (R_1,\ \ldots,\ R_M)^\top \sim LN.$

$\delta_{i,t} > 0$: Buy i^{th} asset.

$\delta_{i,t} < 0$: Sell i^{th} asset.

Proportional transaction cost $\tau \in [0,\ 1]$ paid on each risky asset: $\tau|\delta_{i,t}|$ for $i \in [1,\ \ldots,\ M]$..

The investor's dynamic problem

- Agent maximizes:

$$V_t(W_t, \mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t W_t) + \beta \mathbb{E}_t \left[V_{t+1}(W_{t+1}, \mathbf{x}_{t+1})\right]\} \text{ for } t < T,$$

subject to:

$$W_{t+1} = (\mathbf{R}_t{}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t)\; W_t,$$

$$b_t = 1 - \mathbf{1}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t.$$

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

$$\boldsymbol{\delta}_t W_t \geq - \mathbf{x}_t W_t$$

$$b_t W_t \geq 0$$

$$\mathbf{1}^{\top} \cdot x_t \leq 1 \qquad \text{for } t \in [0, T]$$

The investor's dynamic problem

- Agent maximizes:

$$\boxed{v(\mathbf{x}) = V(W, \mathbf{x}) \,/\, W^{1-\gamma}}$$

$$v_t(\mathbf{x}_t) = \max_{\boldsymbol{\delta}_t,\, c_t} \{u(c_t) + \beta\mathbb{E}_t\left[\ \pi_{t+1}^{1-\gamma}\, v_{t+1}(\mathbf{x}_{t+1})\ \right]\} \quad \text{for } t < T,$$

subject to:

$$\pi_{t+1} := (\mathbf{R}_t^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t), \qquad \boxed{W_{t+1} = W_t\, \pi_{t+1}}$$

$$b_t = 1 - \mathbf{1}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t,$$

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

$$\boldsymbol{\delta}_t \geq -\,\mathbf{x}_t$$

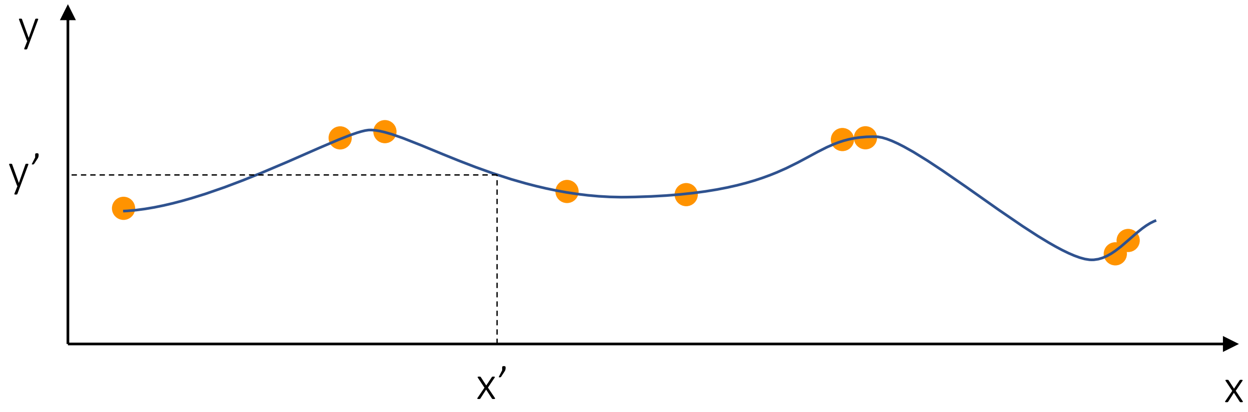
$$b_t \geq 0$$

$$\mathbf{1}^{\top} \cdot x_t \leq 1 \qquad \text{for } t \in [0, T]$$

Gaussian process regressions
The difference to standard machine learning methods

- Machine learning methods:

Input → Output



Gaussian process regressions

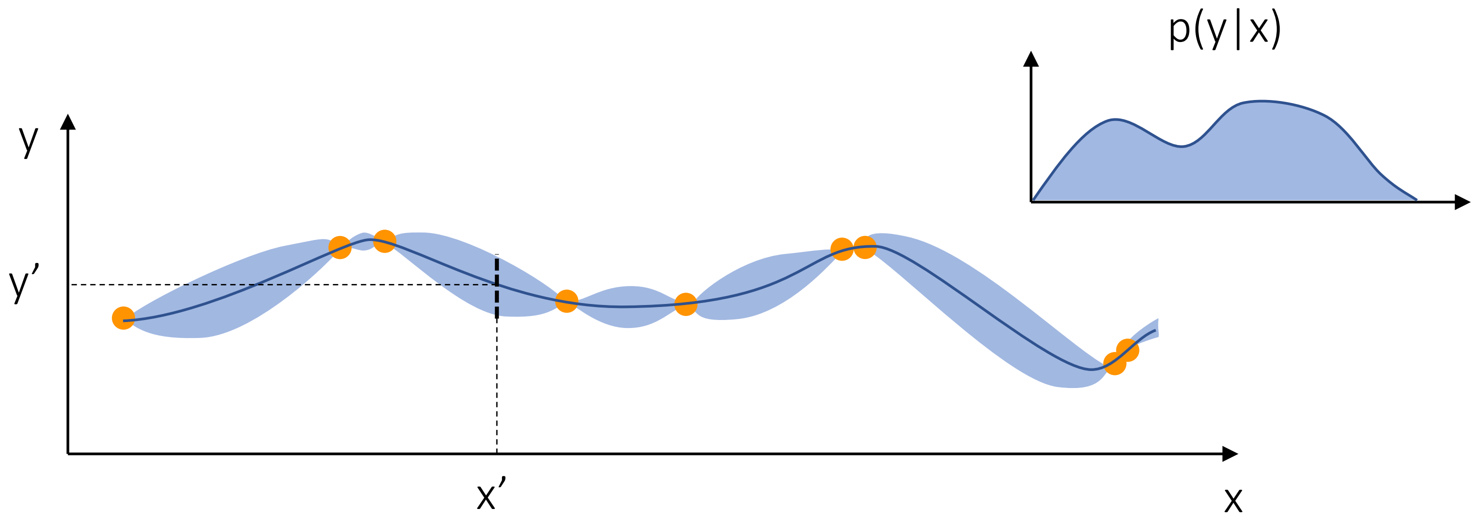
The difference to standard machine learning methods

- Machine learning methods:

Input → Output

- Gaussian process regression:

Input → Distribution



Gaussian process regressions

GPR: the kernel

Kernel is a function which measures the similarity between two points, x and x', written as:

$$\mathcal{K}(\mathrm{x},\ \mathrm{x}' \mid \theta)$$

where θ are the kernel specific hyperparameters.

Gaussian process regressions

GPR: the kernel

Kernel is a function which measures the similarity between two points, x and x', written as:

$$\mathcal{K}(\mathrm{x},\ \mathrm{x}' \mid \theta)$$

where θ are the kernel specific hyperparameters.

Matern:

$$\mathcal{K}(\mathrm{x},\mathrm{x}'|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x},\mathrm{x}')\right)$$

Gaussian process regressions

GPR: the kernel

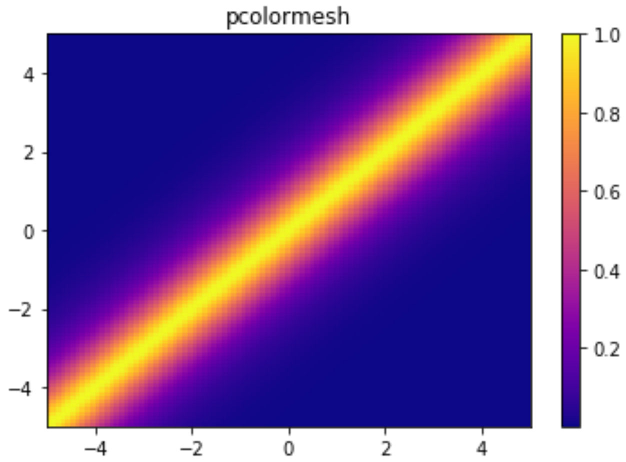
Kernel is a function which measures the similarity between two points, x and x', written as:

$$\mathcal{K}(\mathrm{x},\ \mathrm{x}' \mid \theta)$$

where θ are the kernel specific hyperparameters.

Matern:

$$\mathcal{K}(\mathrm{x,x'}|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x,x'})\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x,x'})\right)$$

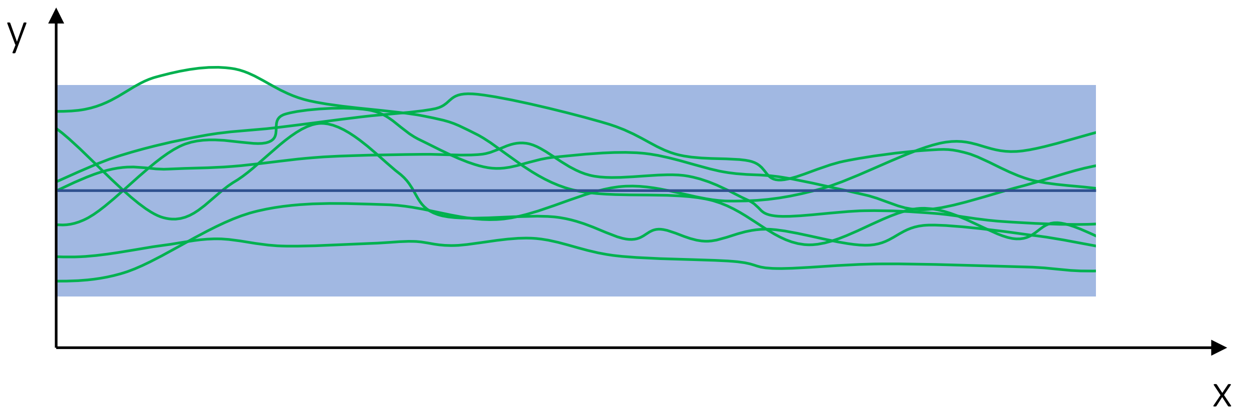


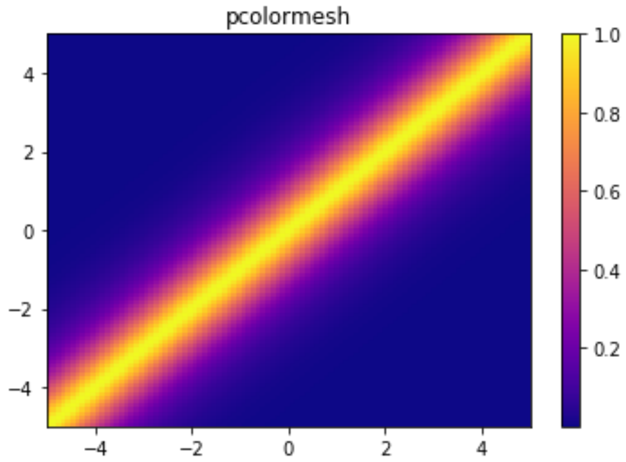
Gaussian process regressions

GPR: Sampling functions from prior

Matern:

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



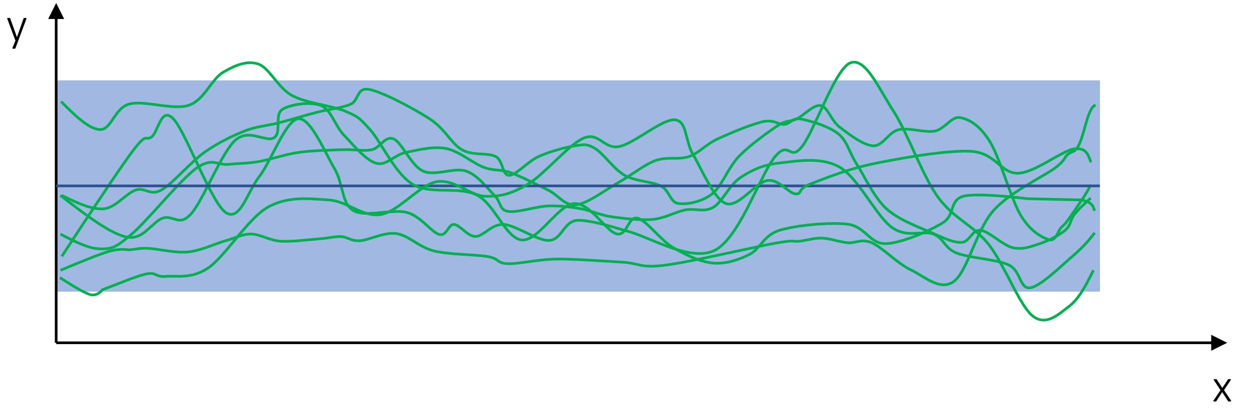


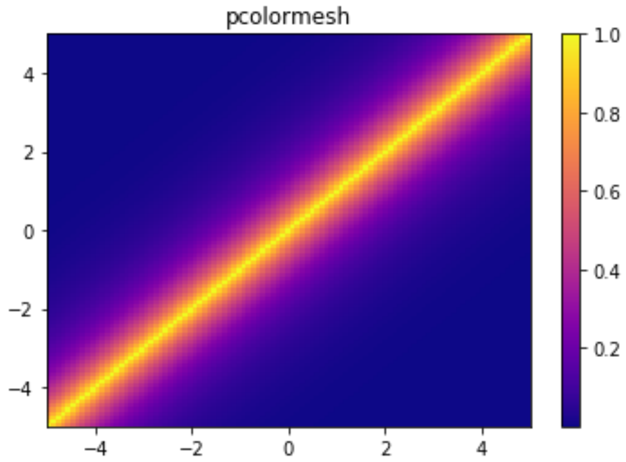
Gaussian process regressions

GPR: Sampling functions from prior

Matern:

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



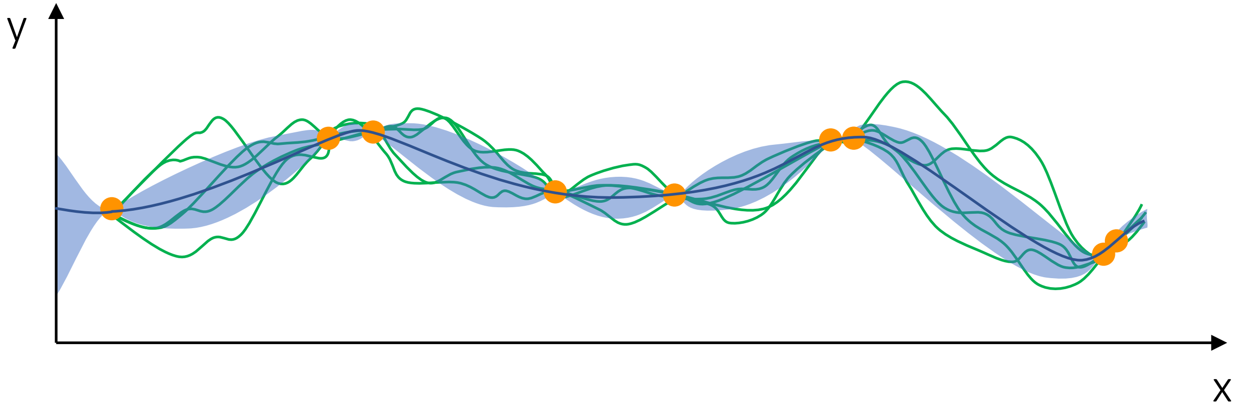


Gaussian process regressions

GPR: Sampling functions from posterior

Matern:

$$\mathcal{K}(\mathrm{x,x'}|\nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x,x'})\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathrm{x,x'})\right)$$



Gaussian process regressions

- Assume $\mathrm{y_i} = \mathrm{f}(\mathrm{x_i}) + \epsilon_\mathrm{i}$, with $\epsilon_\mathrm{i} \sim \boldsymbol{\mathcal{N}}(0,\, \sigma_\epsilon^2)$.
- Observed data: $\mathbf{X},\, \mathbf{X}^*,\, \mathbf{y}$,
- Unobserved true function outputs for $\mathbf{X},\, \mathbf{X}^*: \mathbf{f},\, \mathbf{f}^*$.
- Let $\mathrm{K_{X,X}}$ be an $\mathrm{N} \times \mathrm{N}$ matrix of similarities $\mathcal{K}(\mathrm{x_i}, \mathrm{x_j} | \theta)$.

Gaussian process regressions

- Assume $\mathrm{y_i} = \mathrm{f(x_i)} + \epsilon_\mathrm{i}$, with $\epsilon_\mathrm{i} \sim \boldsymbol{\mathcal{N}}(0,\, \sigma_\epsilon^2)$.
- Observed data: $\mathbf{X},\ \mathbf{X}^*,\ \mathbf{y}$,
- Unobserved true function outputs for $\mathbf{X},\ \mathbf{X}^* :\ \mathbf{f},\ \mathbf{f}^*$.
- Let $\mathrm{K_{X,X}}$ be an $\mathrm{N}\times\mathrm{N}$ matrix of similarities $\mathcal{K}(\mathrm{x_i},\mathrm{x_j}|\theta)$.

- **GP assumption**: $\mathbf{y}$ and $\mathbf{f}^*$ are distributed as an $(\mathrm{N}+\mathrm{M})$-dimensional multivariate normal:

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \hat{K}_{x,x} & K_{x,x^*} \\ K_{x^*,x} & K_{x^*,x^*} \end{bmatrix} \right)$$

where $\widehat{\mathrm{K}}_{\mathrm{X,X}} = \mathrm{K_{X,X}} + \sigma_\epsilon^2\mathbf{I}$

Gaussian process regressions

- **GP assumption**: $\mathbf{y}$ and $\mathbf{f}^*$ are distributed as an $(\mathrm{N}+\mathrm{M})$-dimensional multivariate normal:

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \hat{K}_{x,x} & K_{x,x^*} \\ K_{x^*,x} & K_{x^*,x^*} \end{bmatrix}\right)$$

where $\widehat{\mathrm{K}}_{\mathrm{X,X}} = \mathrm{K}_{\mathrm{X,X}} + \sigma_\epsilon^2 \mathbf{I}$

- If vector is normally distributed and partially observed, we have:

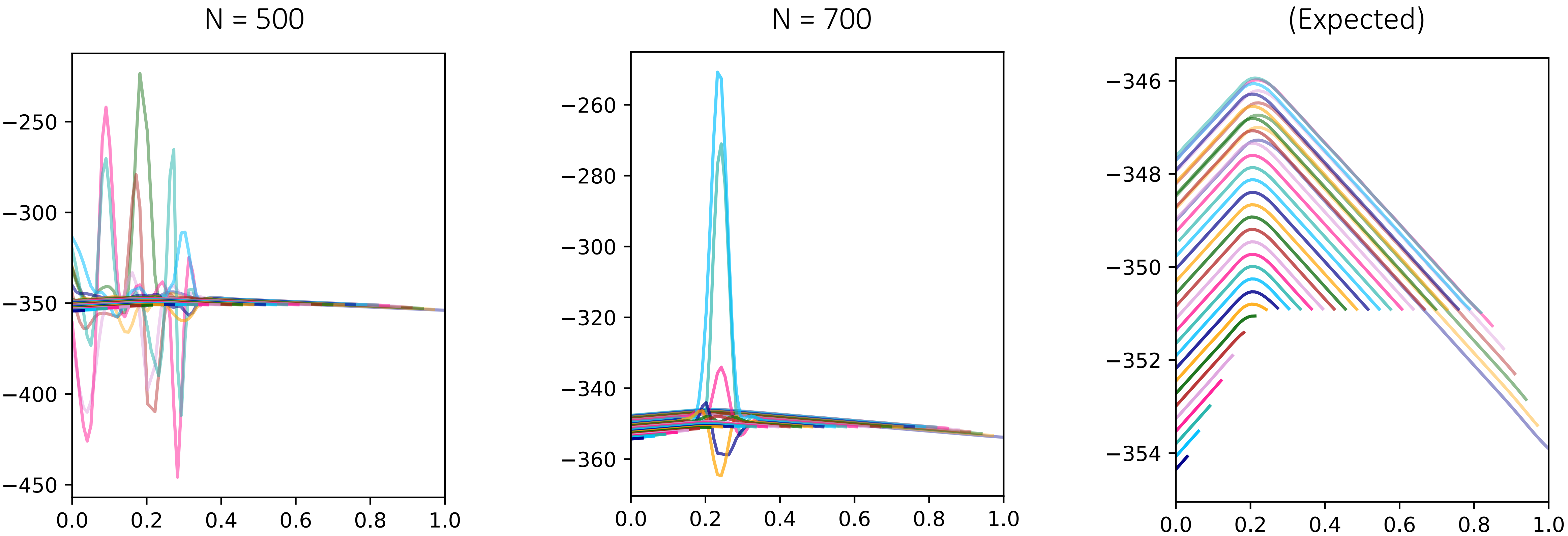
$$f^*|X^*, \mathcal{D} \sim \mathcal{N}\left(K_{x^*,x} K_{x,x}^{-1} y,\; K_{x^*,x^*} - K_{x^*,x} K_{x,x}^{-1} K_{x,x^*}\right)$$

Algorithm: Vanilla dynamic programming with GPRs

- For $\mathrm{t} \in [\mathrm{T}, 0]$:
  - Sample $\{\mathbf{x}_{\text{t-1}}\}_{\mathrm{i}=1}^{\mathrm{N}} \in \boldsymbol{\mathcal{X}}$.
  - For each $\mathbf{x}_{\text{t-1,i}} \in \{\mathbf{x}_{\text{t-1}}\}_{\mathrm{i}=1}^{\mathrm{N}}$:
    - If t = T:
      - Solve for $\mathrm{v}_{\text{T-1,i}}$ and $\{\boldsymbol{\delta}_{\text{T-1,i}},\, \mathrm{c}_{\text{T-1,i}}\}$ using terminal value function $\mathrm{v}_{\mathrm{T}}(\mathbf{x}_{\mathrm{T}})$.
    - Else:
      - Solve for $\mathrm{v}_{\text{t-1,i}}$ and $\{\boldsymbol{\delta}_{\text{t,i}},\, \mathrm{c}_{\text{t,i}}\}$ using GPR approximated value function $\mathcal{GP}_{\mathrm{v}}(\mathbf{x}_{\mathrm{t}})_{\mathrm{t}}$.
  - Fit GPR: $\mathcal{GP}_{\mathrm{v}_{\text{t-1}}}(\mathbf{x}_{\text{t-1}})$: $\mathrm{v}_{\text{t-1,i}} \sim \mathbf{x}_{\text{t-1}}$.

Output: $\{\mathcal{GP}_{\mathrm{v}_{\mathrm{t}}}\}_{\mathrm{t}=0}^{\text{T-1}}$, $\{\mathrm{v}_{\text{t,i}},\, \boldsymbol{\delta}_{\text{t,i}},\, \mathrm{c}_{\text{t,i}}\}_{\mathrm{t}=0}^{\text{T-1}}$.

Vanilla dynamic programing is unstable

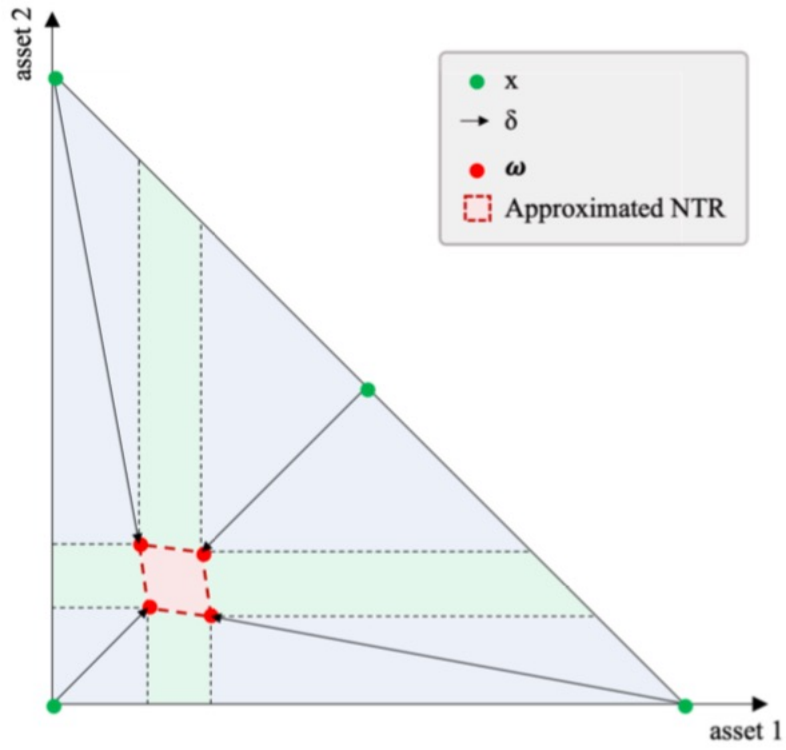


Algorithm: Approximating the NTR

Algorithm 2: Approximate the t^{th}-period NTR in the discrete-time finite-horizon portfolio choice model with proportional transaction costs.

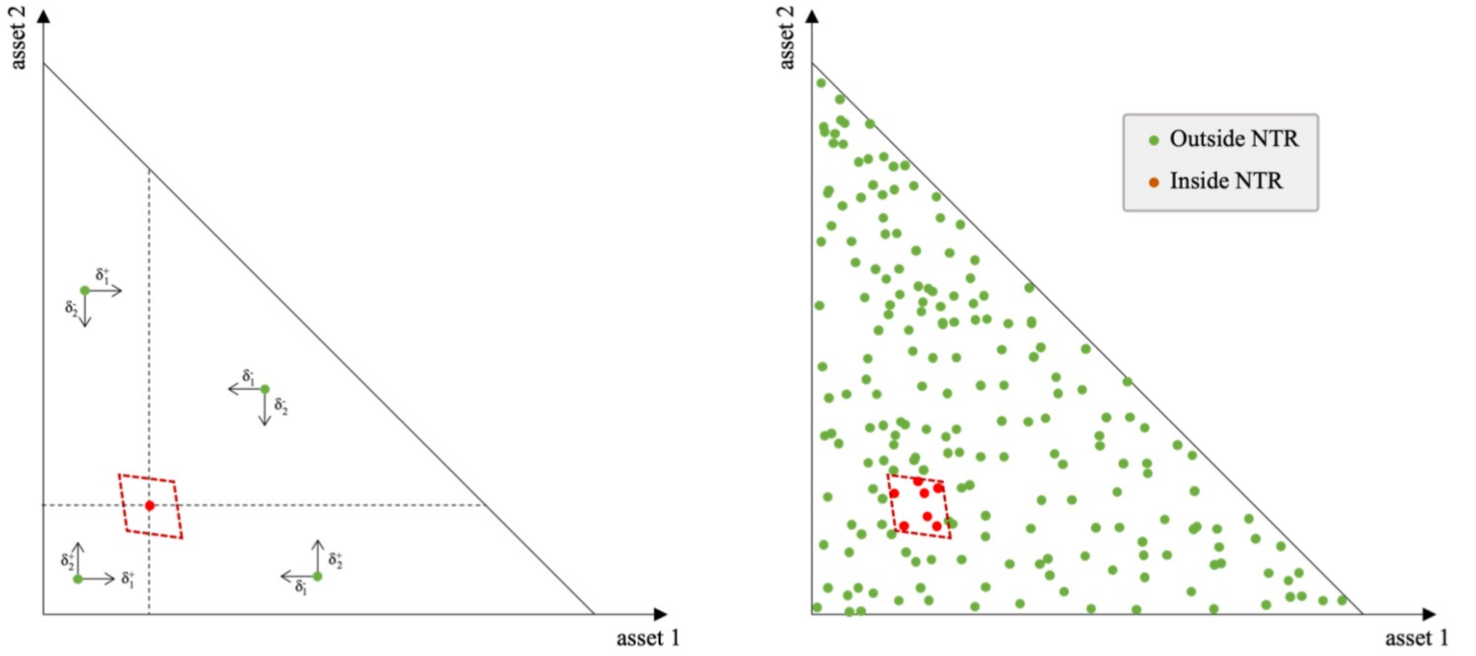
Input: $t+1^{\text{st}}$ period's value function (surrogate) $\mathcal{V}_{t+1}$.
Data: Set of $N = 2^D$ points $\tilde{\mathbf{X}}_t = \{\tilde{\mathbf{x}}_{i,t}\}_{i=1}^{N} \subset \mathcal{B}$.
Result: Set of approximated NTR vertices: $\{\hat{\boldsymbol{\omega}}_{i,t}\}_{i=1}^{N}$; Approximated NTR: $\hat{\boldsymbol{\Omega}}_t$.

for $\tilde{\mathbf{x}}_{i,t} \in \tilde{\mathbf{X}}_t$ **do**
  Obtain policy $\hat{\tilde{\boldsymbol{\delta}}}_{i,t}$ for $\tilde{\mathbf{x}}_{i,t}$ by solving the optimization problem given by the Bellman equation (Eq. (2.10)) using $\mathcal{V}_{t+1}$ as next period's value function.
  Compute the approximate NTR vertices $\hat{\boldsymbol{\omega}}_{i,t} = \tilde{\mathbf{x}}_{i,t} + \hat{\tilde{\boldsymbol{\delta}}}_{i,t}$.
end
Compute the NTR approximation: $\hat{\boldsymbol{\Omega}}_t = \{\boldsymbol{\lambda}\hat{\boldsymbol{\omega}}_t \mid \boldsymbol{\lambda} \in (0,1)^N, \sum_{i=1}^{N} \boldsymbol{\lambda}_i = 1\}$.



Using the approximated NTR to facilitate solving the model

- Pre-determine sign of optimal δ-policy,
- Compute initial guesses,
- Use multiple GPRs to segment state space,
- Data sampling (heuristic or algorithmic),
- NTR approximation.



(a) Use the NTR to determine the sign of the $\boldsymbol{\delta}$-policy. Each point in the same state-space segment (as defined by the mid point; dotted lines) are given the same policy bounds in the solver.

(b) GPRs fit the value function on different regions of the state space as defined by the approximated NTR. Two dataset are composed of red points inside (green points outside) of the NTR, respectively.

Figure 2.3: Using the approximated NTR to facilitate solving the model

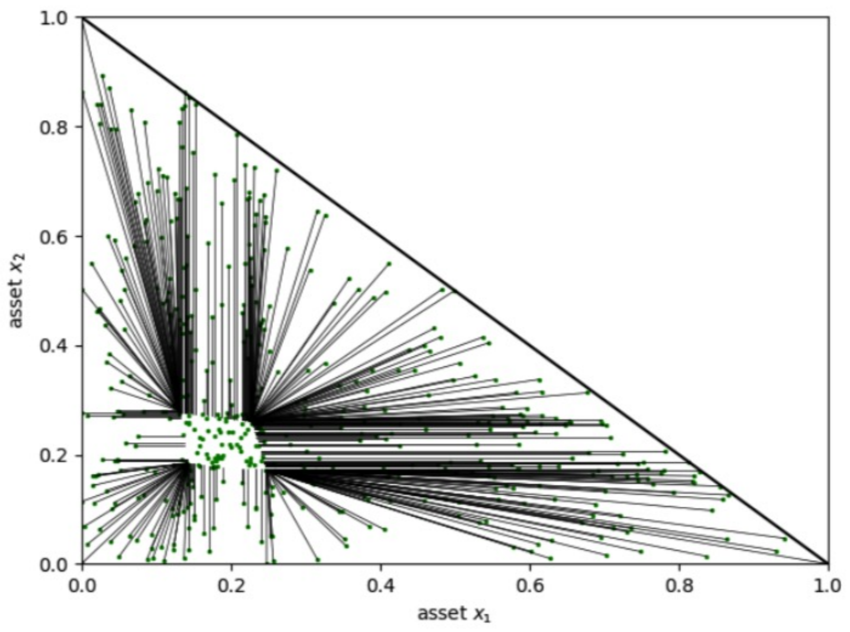
Algorithm: Dynamic programming with GPRs and NTR approximation

Algorithm 3: Dynamic programming with Gaussian process regressions and the NTR approximation for discrete-time finite-horizon portfolio optimization problems with proportional transaction costs.

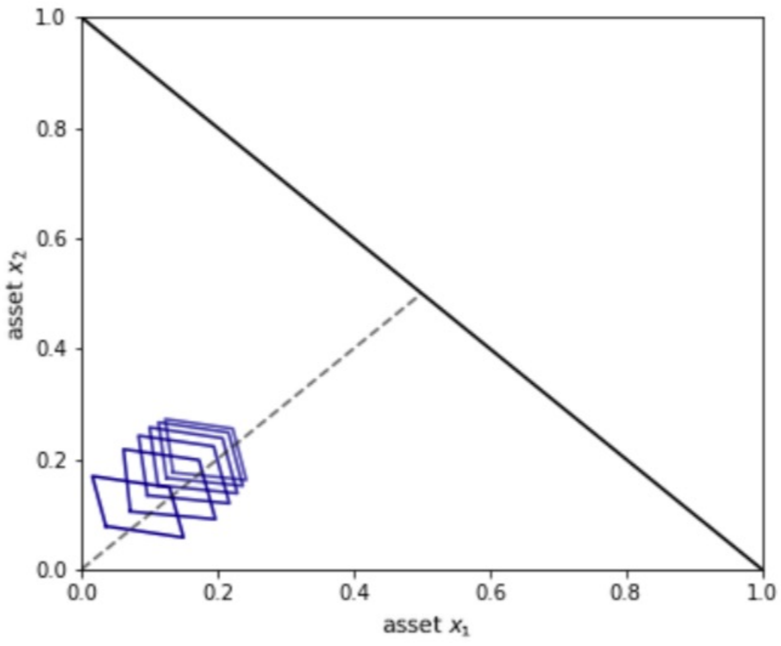
Input: Terminal value function v_T; time horizon T; sample size N.
Result: Set of GP surrogates of the value functions $\{v_{t-1}\}_{t=0}^{T-1}$; set of approximated NTRs $\{\hat{\mathbf{\Omega}}_t\}_{t=0}^{T-1}$.

Set $\mathcal{V}_T = v_T$ (Eq. (2.11)).
for $t \in [T, ..., 1]$ **do**
    Approximate NTR $\hat{\mathbf{\Omega}}_{t-1}$ (Alg. 2) using $\mathcal{V}_t$ as the next period's value function.
    Sample N points $\mathbf{X}_{t-1} = \{\mathbf{x}_{i,t-1}\}_{i=1}^{N} \subset \mathcal{B}$ from D-dimensional simplex.
    for $\mathbf{x}_{i,t-1} \in \mathbf{X}_{t-1}$ **do**
        Obtain value $\hat{v}_{i,t-1}$ and policy $\{\hat{\boldsymbol{\delta}}_{i,t-1}, \hat{c}_{i,t-1}\}$ for $\mathbf{x}_{i,t-1}$ by solving the optimization problem given by the Bellman equation (Eq. (2.10)) using $\mathcal{V}_t$ as the next period's value function.
    end
    Define the training sets:
        $\mathcal{D}_{\text{in},t-1} = \{(\mathbf{x}_{i,t-1}, \hat{v}_{i,t-1}) \ : \ \text{if } \mathbf{x}_{i,t-1} \in \hat{\mathbf{\Omega}}_{t-1}\}$,
        $\mathcal{D}_{\text{out},t-1} = \{(\mathbf{x}_{i,t-1}, \hat{v}_{i,t-1}) \ : \ \text{if } \mathbf{x}_{i,t-1} \notin \hat{\mathbf{\Omega}}_{t-1}\}$.
    Given $\mathcal{D}_{\text{in},t-1}$ and $\mathcal{D}_{\text{out},t-1}$, learn a surrogate of v_{t-1} for inside and outside of the NTR $\{\mathcal{G}_{\text{in},t-1}, \mathcal{G}_{\text{out},t-1}\}$ (using the respective datasets) with GPs:
    Set $\mathcal{V}_{t-1} = \{\mathcal{G}_{\text{in},t-1}, \mathcal{G}_{\text{out},t-1}\}$.
end

Solution results from benchmark parameterization



(a) NTR forms from $\boldsymbol{\delta}$-policy (black line) as computed by constraint optimizer for each (green) point in the state space (iteration 7).



(b) Trajectory of approximated NTRs over time. NTRs move towards the origin as time approaches the terminal time.

Evaluation of all errors
Euler errors, value function fit, and NTR approximation

- Relative Euler errors:

$$\tilde{\varepsilon}_{c_t}(\mathbf{x}_t) = \frac{\beta \mathbb{E}_t \left[R_f \pi_{t+1}^{-\gamma} \left((1-\gamma) v_{t+1}(\mathbf{x}_{t+1}) - \nabla_{\mathbf{x}_{t+1}} v_{t+1}(\mathbf{x}_{t+1}) \mathbf{x}_{t+1} \right) \right]^{-\frac{1}{\gamma}}}{c_t} - 1 = 0$$

$$\tilde{\varepsilon}_{\delta_{i,t}}(\mathbf{x}_t) = \frac{\left[\frac{\beta \mathbb{E}_t \left[R_{i,t} \pi_{t+1}^{-\gamma} \left((1-\gamma) v_{t+1}(\mathbf{x}_{t+1}) - \nabla_{\mathbf{x}_{t+1}} v_{t+1}(\mathbf{x}_{t+1}) (\mathbf{e}_i - \mathbf{x}_{t+1}) \right) \right]}{-\frac{\partial c_t}{\partial \delta_{i,t}}} \right]^{-\frac{1}{\gamma}}}{c_t} - 1 = 0$$

- Value function fit:

$$RAE(v, \hat{v}) = \frac{|v - \hat{v}|}{|v|},$$

- NTR approximation:

$$\mathrm{SE}(\widehat{\delta}, \widetilde{\delta}) = (\widehat{\delta} - \widetilde{\delta})^2$$

Baseline model calibration

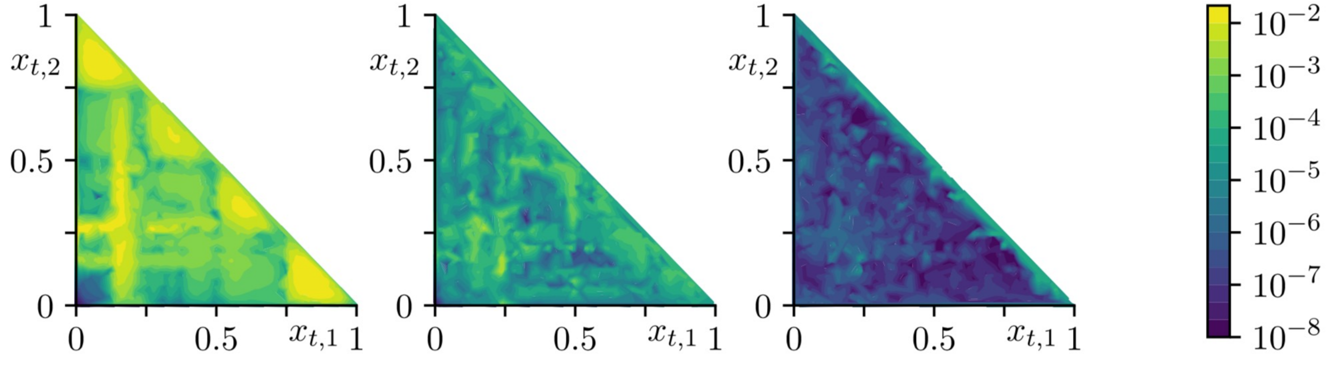
Parameter	Value
T	6
γ	3.5
β	0.97
R_f	4%
τ	1%

$$\boldsymbol{\mu} = \begin{pmatrix} 0.0572 \\ 0.0638 \\ 0.07 \\ 0.0764 \\ 0.0828 \end{pmatrix}, \quad \boldsymbol{\Sigma} = 10^{-2} \begin{pmatrix} 2.56 & 0.576 & 0.288 & 0.176 & 0.096 \\ 0.576 & 3.24 & 0.90432 & 1.0692 & 1.296 \\ 0.288 & 0.90432 & 4 & 1.32 & 1.68 \\ 0.176 & 1.0692 & 1.32 & 4.84 & 2.112 \\ 0.096 & 1.296 & 1.68 & 2.112 & 5.76 \end{pmatrix}.$$

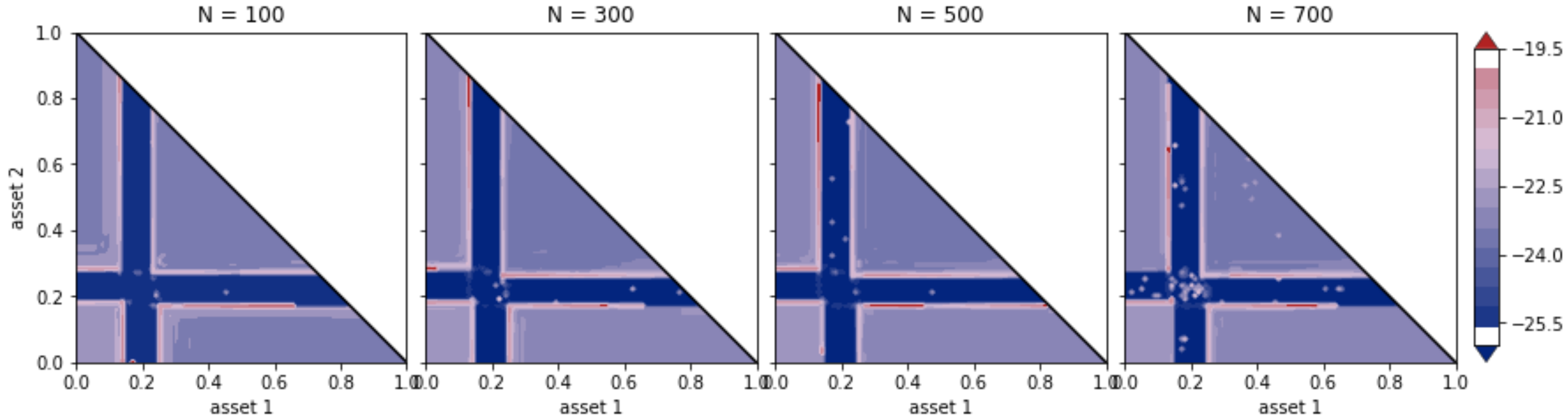
Schober, Valentin, and Pflüger (2020)

Evaluation

Relative Euler errors of consumption across sample size



(a) $N_0 = 129$ $(\varepsilon_0^{\mathrm{w},L^2} = 5.3 \cdot 10^{-3})$ **(b)** $N_0 = 889$ $(\varepsilon_0^{\mathrm{w},L^2} = 2.1 \cdot 10^{-4})$ **(c)** $N_0 = 5159$ $(\varepsilon_0^{\mathrm{w},L^2} = 1.2 \cdot 10^{-5})$



N	Mean	Max
100	$1.4 \cdot 10^{-10}$	$7.2 \cdot 10^{-8}$
300	$1.3 \cdot 10^{-10}$	$4.4 \cdot 10^{-9}$
500	$1.4 \cdot 10^{-10}$	$5.3 \cdot 10^{-10}$
700	$1.1 \cdot 10^{-10}$	$4.4 \cdot 10^{-10}$

Schober, Valentin, and Pflüger (2020)

Relative Euler errors by sample size

N		Mean	99.9^{th} percentile	Max
		"Oversample"		
300	δ_1	0.000222 (2.899e-08)	0.00273 (5.345e-06)	0.00283 (6.826e-06)
	δ_2	0.000189 (3.942e-08)	0.00284 (4.751e-06)	0.00392 (5.499e-06)
	c	1.251e-10	3.968e-09	4.354e-09
500	δ_1	0.000222 (4.598e-08)	0.00278 (7.911e-06)	0.00334 (9.935e-06)
	δ_2	0.000179 (4.747e-08)	0.00274 (5.426e-06)	0.00332 (1.105e-05)
	c	1.394e-10	4.709e-09	5.339e-09
700	δ_1	0.000223 (2.753e-08)	0.00304 (3.945e-06)	0.00393 (4.846e-06)
	δ_2	0.000185 (4.087e-08)	0.00284 (5.996e-06)	0.00395 (8.217e-06)
	c	1.141e-10	3.874e-09	4.405e-09

Bisection results for liquidity premium

Bisection results for liquidity premium

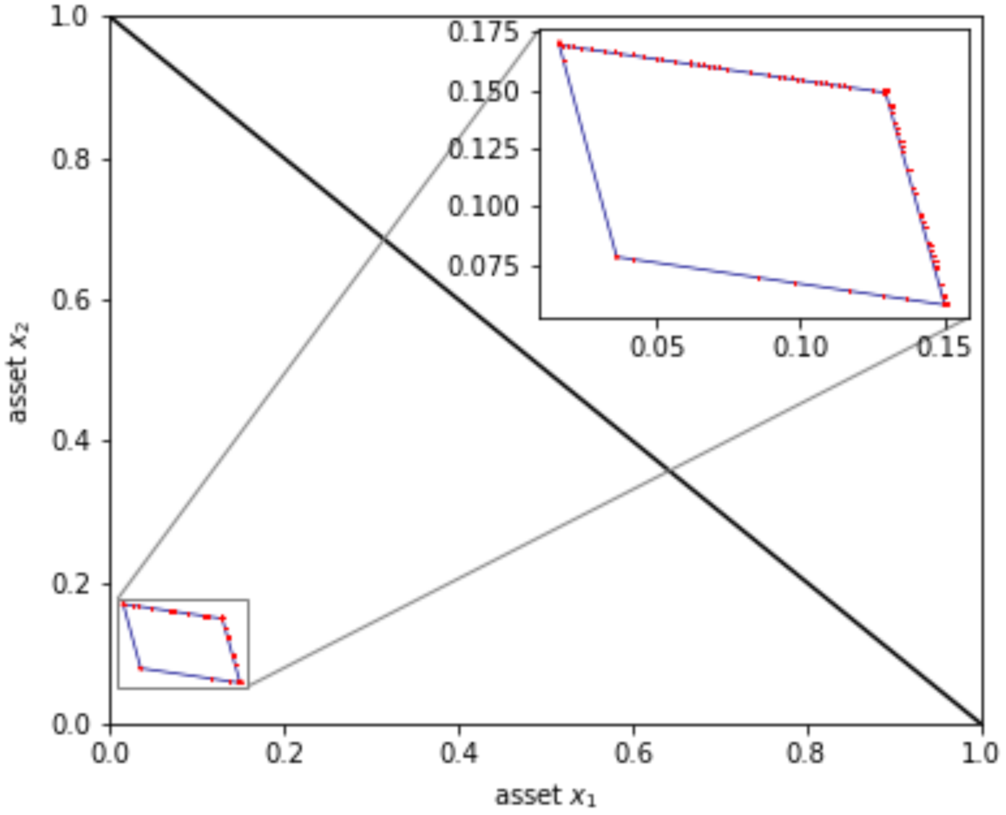
2 risky assets		3 risky assets		5 risky assets	
λ	$U_{\tau=0}/U_{\tau\neq0}$	λ	$U_{\tau=0}/U_{\tau\neq0}$	λ	$U_{\tau=0}/U_{\tau\neq0}$
0.5	115.65	0.4444	118.37	0.3584	125.82
0.75	111.36	0.7222	114.38	0.6792	120.98
0.875	105.63	0.8611	107.89	0.8396	111.08
0.9375	102.00	0.93055	103.76	0.9198	105.09
0.96875	100.02	0.965275	101.54	0.9599	101.93
0.96918562*	100.00	0.98854459*	100.00	0.98402378*	99.98
1.00	97.95	1.00	99.23	1.00	98.71

Note: "*" indicates the "best guess"; Lifetime utilities for with and without transaction cost (best guess): 2 risky assets: [-2242.7005, -2242.6088]; 3 risky assets: [-251,789,580.012, -251,789,112.587]; 5 risky assets: [-17,380,459,596,785.88, -17,377,577,421,688.197].

Liquidity premium

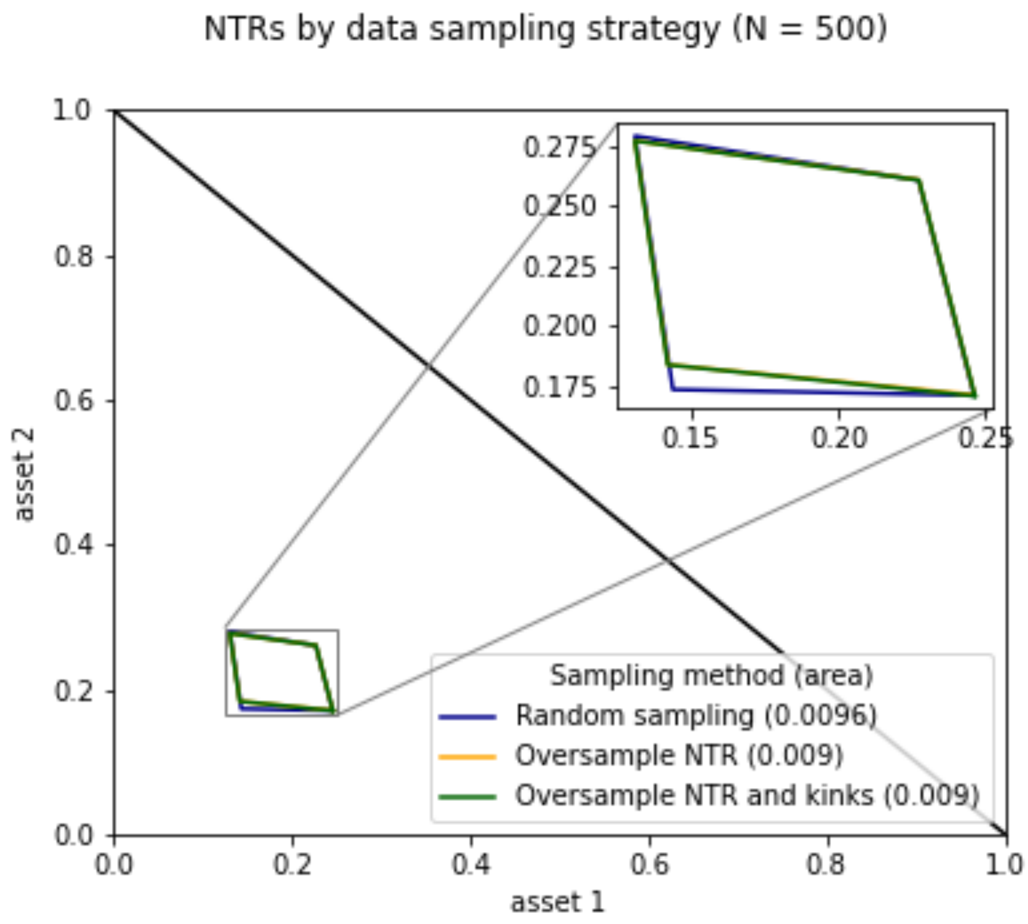
Risky assets	Liquidity premium (%)
2	3.08
3	1.15
5	1.60

No-trade region approximation evaluation

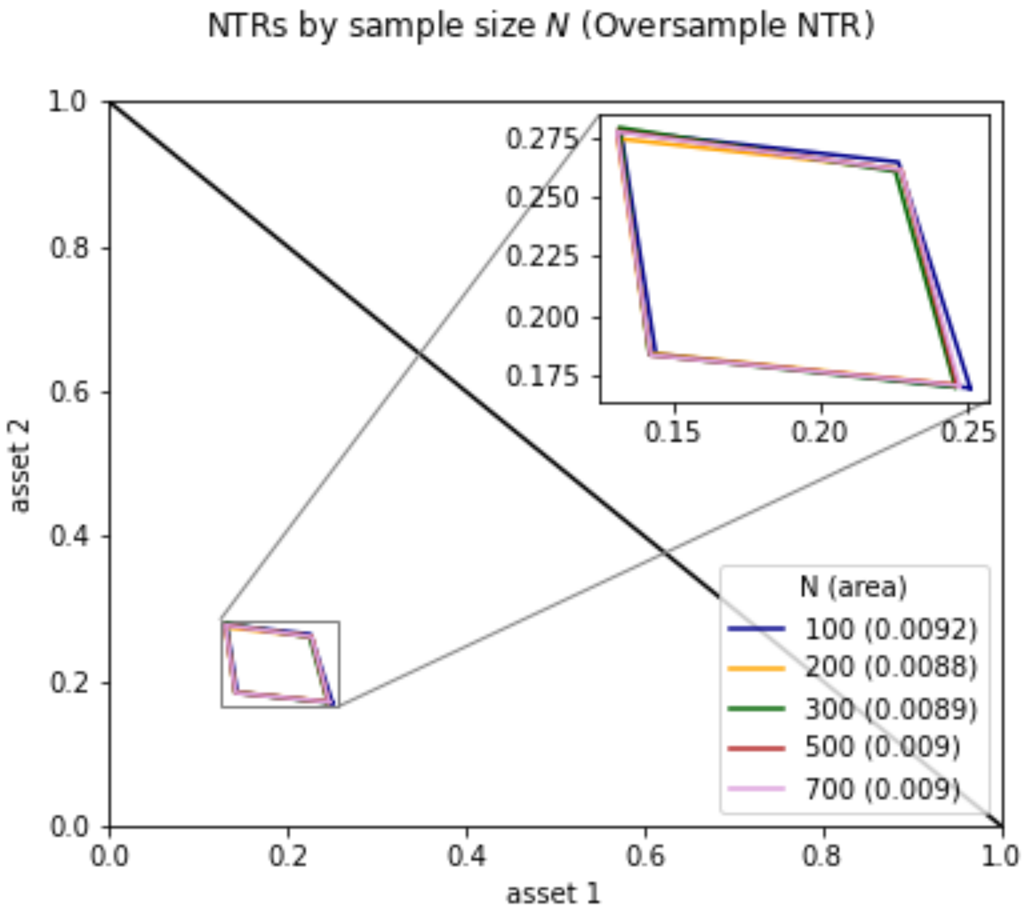


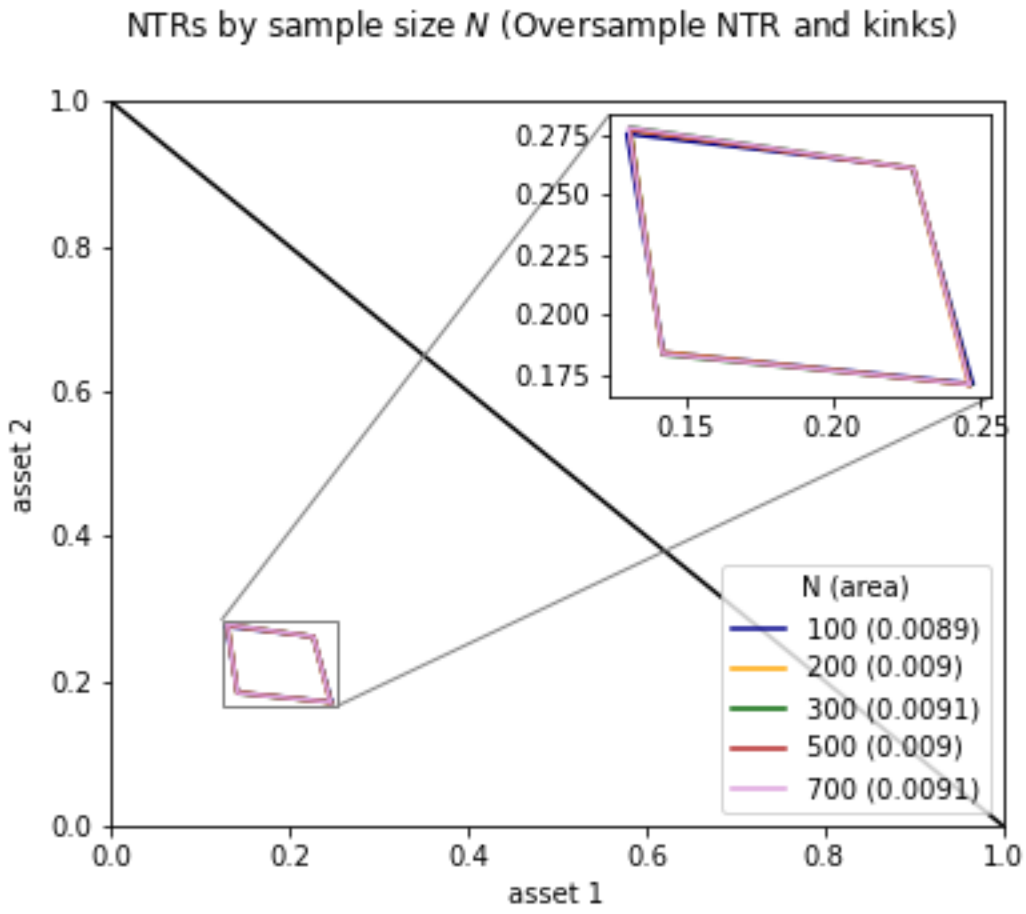
t	Mean squared error	Max squared error
0	0.0000015	0.0000075
1	0.0000015	0.0000068
2	0.0000012	0.0000060
3	0.0000017	0.0000079
4	0.0000011	0.0000055
5	0.0000008	0.0000046
6	0.0000005	0.0000022

NTR approximation precision stabilizes by oversampling kinks



NTR approximation precision stabilizes by oversampling kinks





NTR approximation precision stabilizes by oversampling kinks: Hausdorff

$$d_H(\hat{\mathbf{\Omega}}_N, \hat{\mathbf{\Omega}}_M) = \max\left\{ \sup_{x \in \hat{\mathbf{\Omega}}_N} d(x, \hat{\mathbf{\Omega}}_M), \sup_{y \in \hat{\mathbf{\Omega}}_M} d(\hat{\mathbf{\Omega}}_N, y) \right\}$$

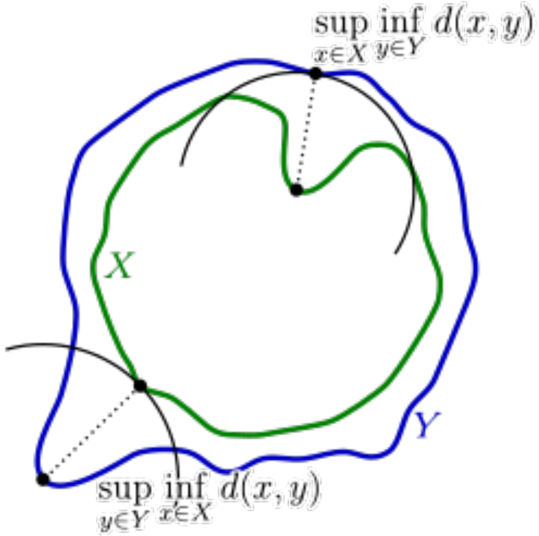


Table 2.7: Sample size induced NTR approximation error

N	100	200	300	500	700
100	0.0				
200	0.0052	0.0			
300	0.0051	0.0027	0.0		
500	0.0047	0.0020	0.0019	0.0	
700	0.0038	0.0022	0.0027	0.0018	0.0

NOTE: Hausdorff distance (see Eq. (2.24)) between NTRs across sample size computed using solutions from a model where the NTR was oversampled.

Stylized model calibration

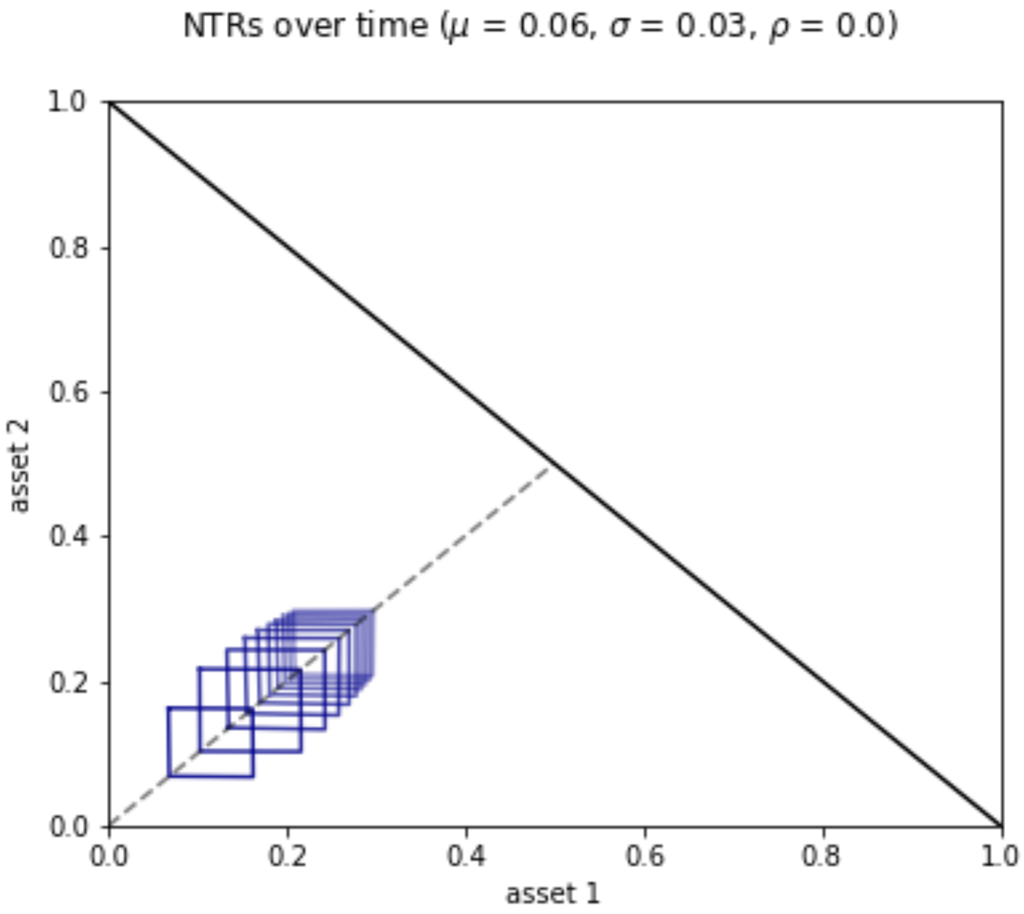
Parameter	Value
T	10
γ	3.5
β	0.97
R_f	4%
τ	1%

$$\boldsymbol{\mu} = \begin{pmatrix} 0.06 \\ 0.06 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 0.03 & 0.0 \\ 0.0 & 0.03 \end{pmatrix}.$$

Symmetrical asset experiments			
1)	μ	0.055	0.065
2)	σ	0.025	0.035
3)	ρ	0.002	0.005
4)	τ	0.005	0.015
5)	β	0.95	0.99
6)	γ	3.0	4.0
7)	R_f	0.03	0.05
Asymmetrical asset experiments			
8)	$\boldsymbol{\mu}$	$\begin{pmatrix} 0.065 & 0.06 \end{pmatrix}^\top$	
9)	$\boldsymbol{\sigma}$	$\begin{pmatrix} 0.03 & 0.035 \end{pmatrix}^\top$	
10)	$\boldsymbol{\tau}$	$\begin{pmatrix} 0.005 & 0.01 \end{pmatrix}^\top$	

NTR experiments

NTR example over time



NTR experiments
Experiments parameterization

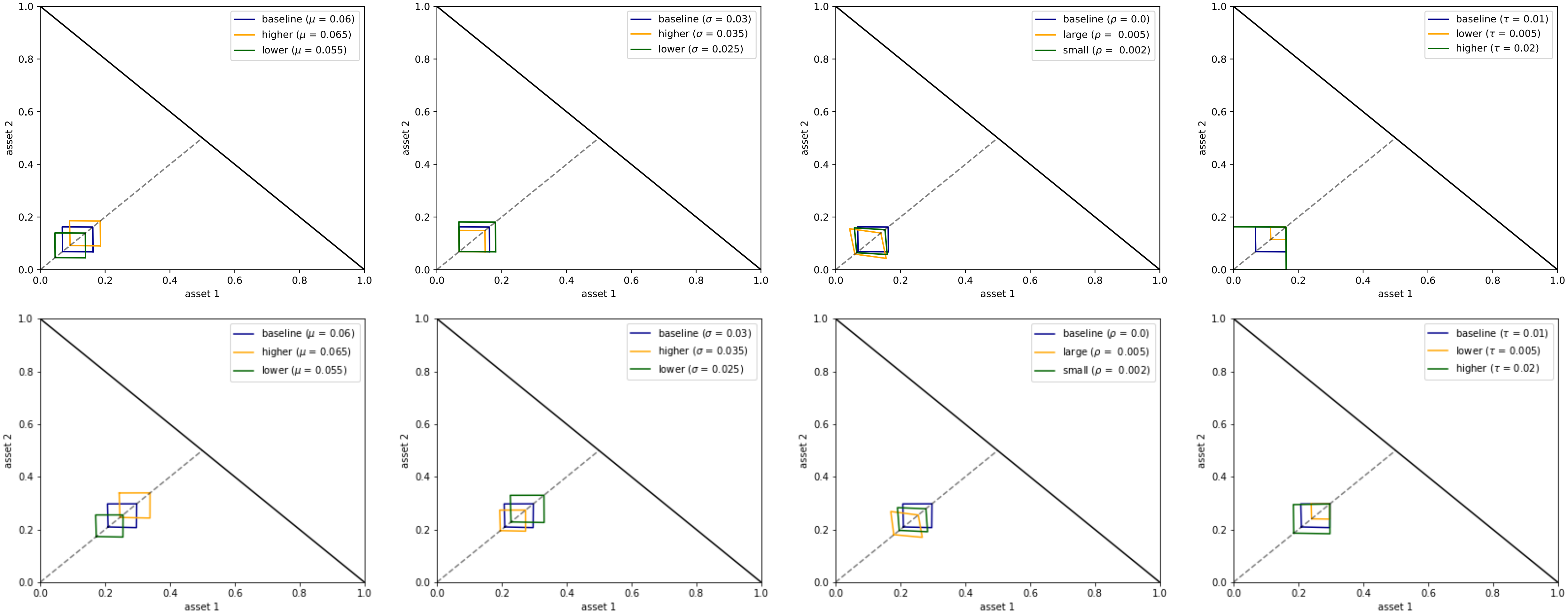
$$\boldsymbol{\mu} = \begin{pmatrix} 0.06 \\ 0.06 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 0.03 & 0.0 \\ 0.0 & 0.03 \end{pmatrix}$$

Parameterization with identical assets			
1)	μ	0.055	0.065
2)	σ	0.025	0.035
3)	ρ	0.002	0.005
4)	τ	0.005	0.015
5)	β	0.95	0.99
6)	γ	3.0	4.0
7)	R_f	0.03	0.05
Parameterization with non-identical assets			
8)	$\boldsymbol{\mu}$		$\begin{pmatrix}0.065 & 0.06\end{pmatrix}^\top$
9)	$\boldsymbol{\sigma}$		$\begin{pmatrix}0.03 & 0.035\end{pmatrix}^\top$
10)	$\boldsymbol{\tau}$		$\begin{pmatrix}0.005 & 0.01\end{pmatrix}^\top$

NTR experiments

Stylized symmetric experiments

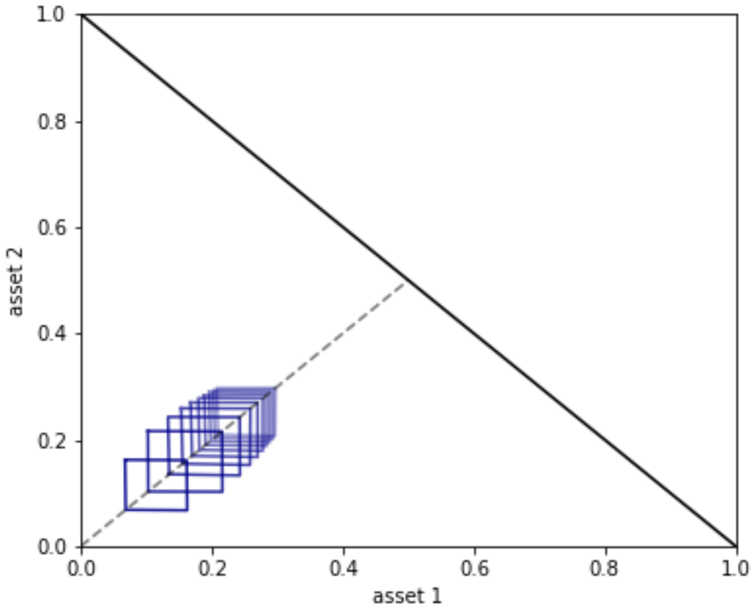
$\boldsymbol{\mu}$: return | $\boldsymbol{\sigma}$: variance | ρ: correlation | τ: transaction cost



NTR experiments

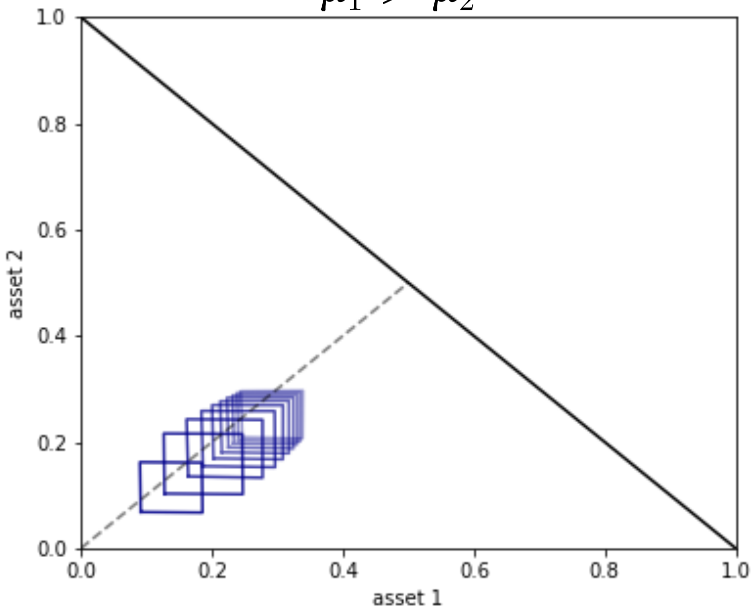
Stylized asymmetric convergence experiments

Symmetric



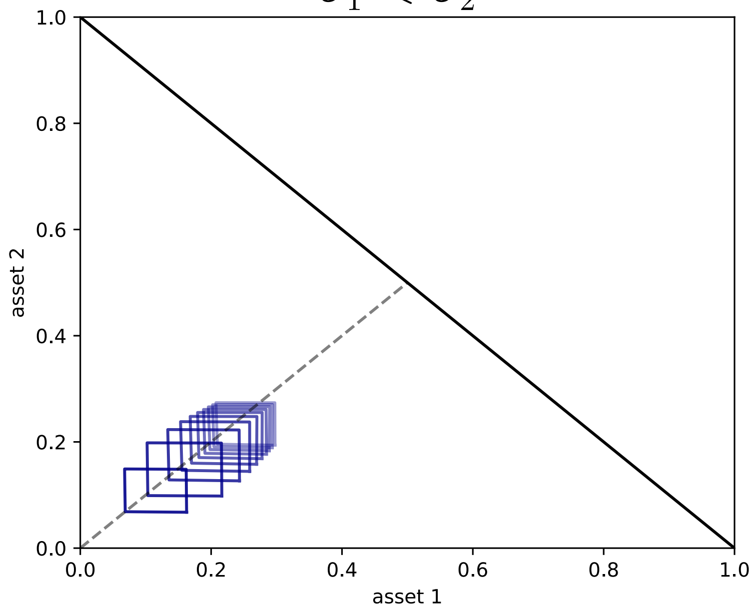
Asymmetric return ($\boldsymbol{\mu}$)

$\mu_1 > \mu_2$

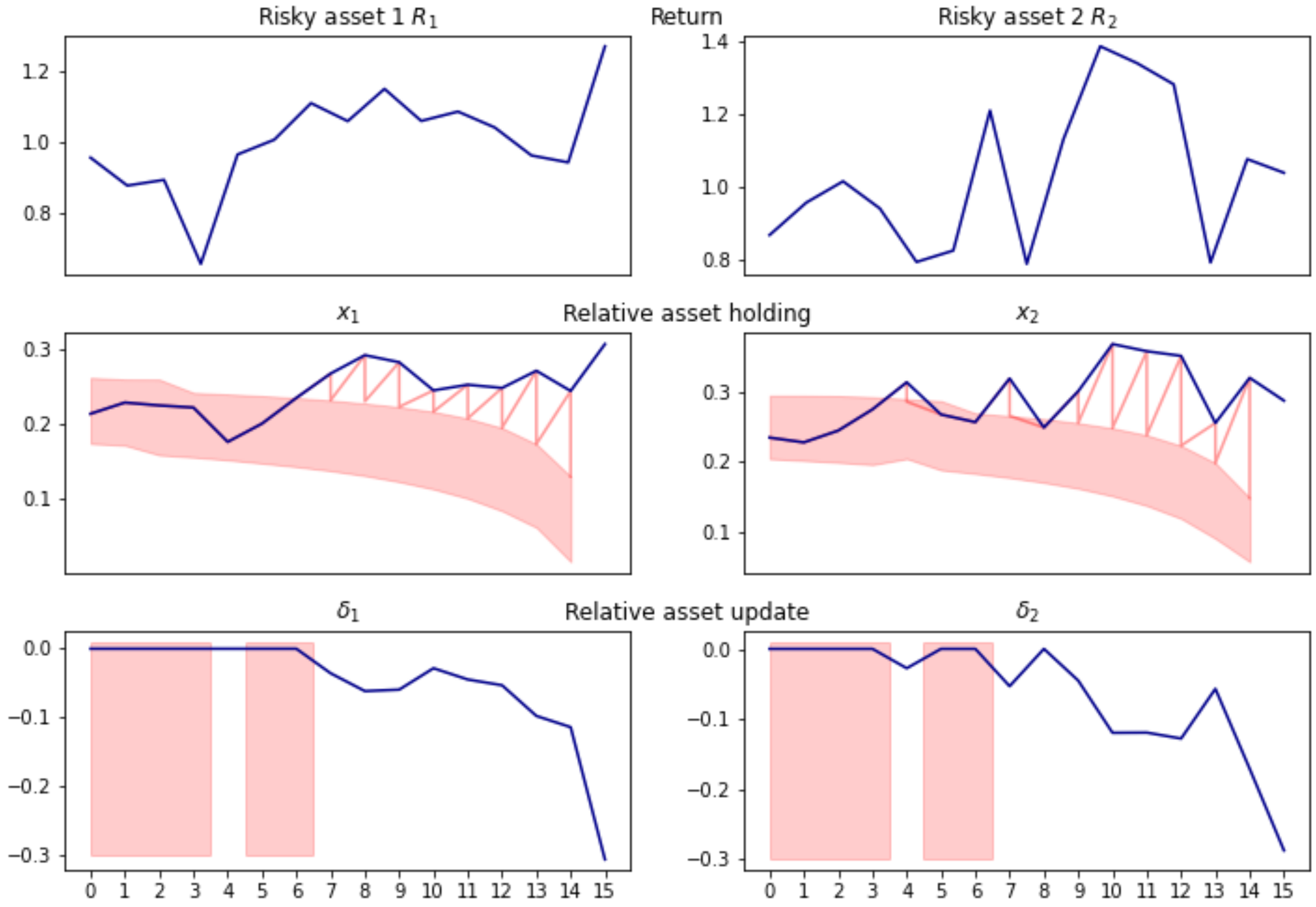


Asymmetric variance ($\boldsymbol{\sigma}$)

$\sigma_1 < \sigma_2$

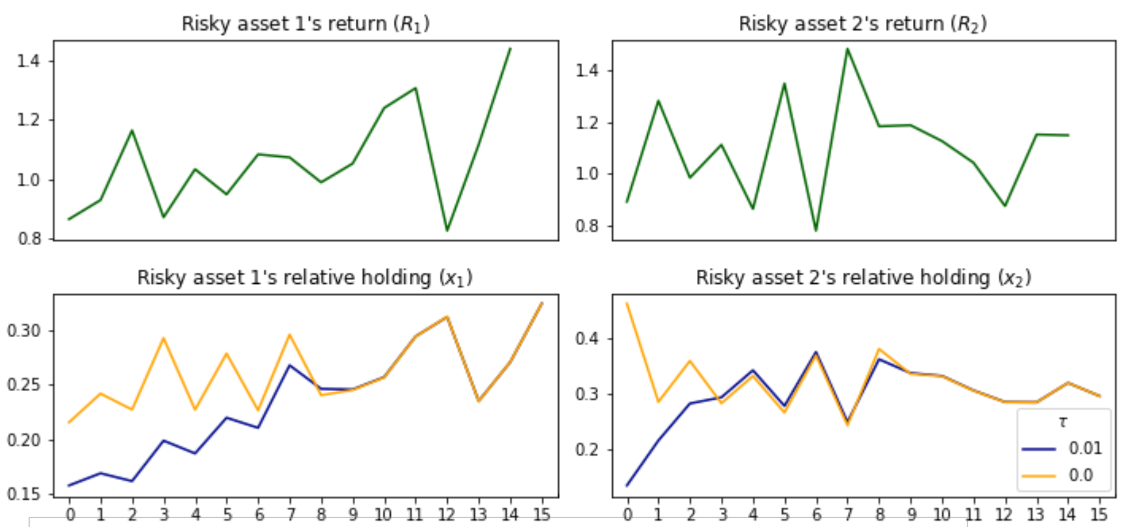


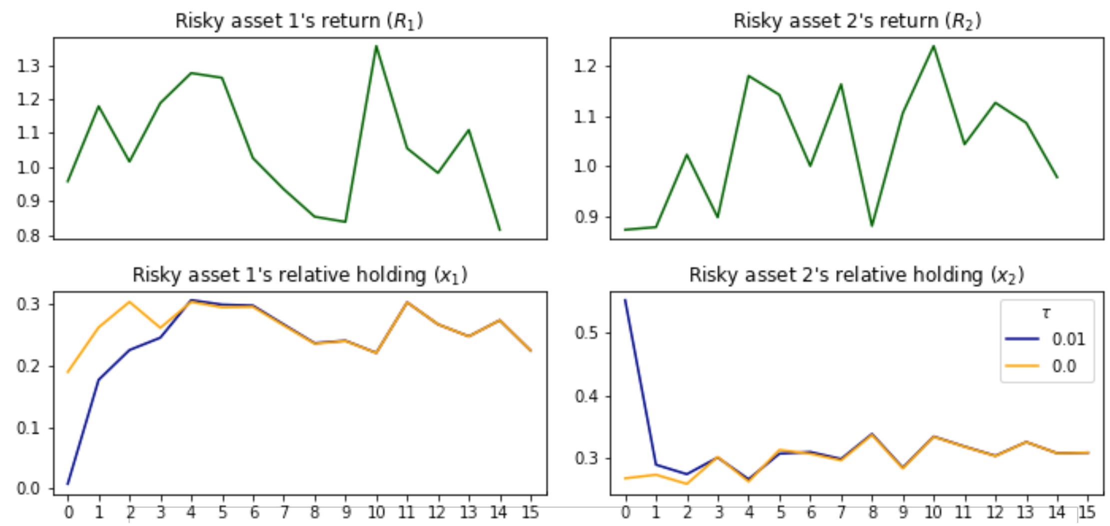
Monte Carlo simulations
A selected example



Monte Carlo simulations

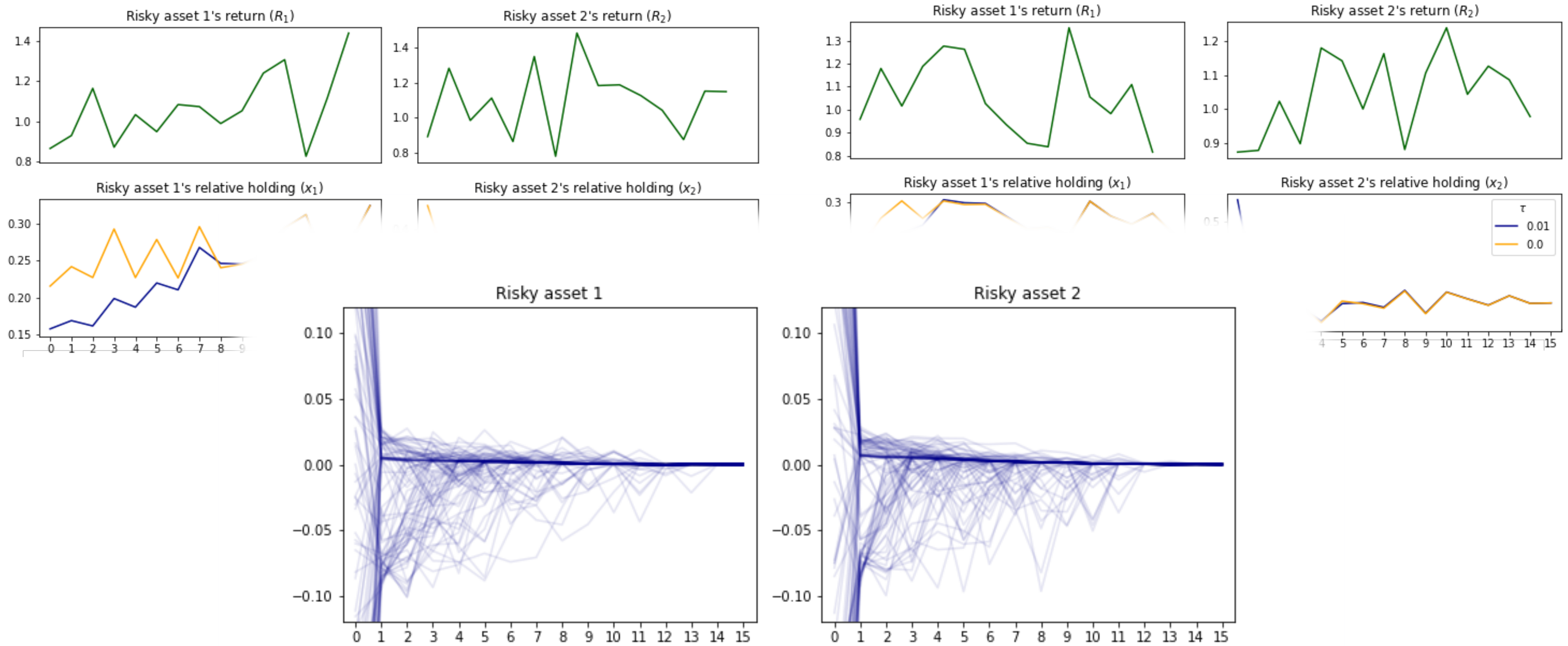
Simulation path compared to no transaction cost setting





Monte Carlo simulations

Simulation path compared to no transaction cost setting



Monte Carlo simulations
Lifetime utility loss

Percentage of expected lifetime utility lost due to transaction costs

Risky assets	% of U_0 lost
2	2.05%
3	0.77%
5	1.29%

Note: Lifetime utilities for with and without transaction cost: 2 risky assets: [-2242.7005, -2196.8066]; 3 risky assets: [-251,789,580.012, -249,857,885.799]; 5 risky assets: [-17,380,459,596,785.88, -17,156,148,727,970.414].

Percentage of expected lifetime utility lost due to trading as if there were no transaction costs

Risky assets	% of U_0 lost
2	0.20%
3	12.54%
5	29.70%

Note: These results were computed using the original baseline calibration horizon of $T = 7$. Lifetime utilities for with and without transaction cost: 2 risky assets: [-348.85, -349.56]; 3 risky assets: [-19,437.05, -21,874.74]; 5 risky assets: [-2,080,796.05, -2,698,782.50].

Risky asset returns

$$\log(\mathrm{R}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \Longleftrightarrow \quad \mathrm{R} \sim \exp(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$$

$$\boldsymbol{\mu} = \boldsymbol{\varphi} - \frac{\boldsymbol{\sigma}^2}{2}$$

Drift

$$\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_{\mathrm{M}})^{\top}$$

Volatility

$$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{\mathrm{M}})^{\top}$$

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Upsilon}\boldsymbol{\Lambda}$$

Correlation matrix of the log-returns

$$\boldsymbol{\Upsilon} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\mathrm{M}})$$

Liquidity premium

$$\mathbb{E}_0\left[\sum_{t=0}^{T}\beta^t u(c_t W_t)\;\middle|\;\boldsymbol{\mu},\ \boldsymbol{\Sigma}\right]$$

$$\mu = \varphi - \sigma^2/2$$

subject to:

$$\pi_{t+1} := (\mathbf{R}_t{}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t),$$

$$b_t = 1 - \mathbf{1}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t + \tau|\boldsymbol{\delta}_t|) - c_t,$$

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

$$\boldsymbol{\delta}_t \geq -\mathbf{x}_t$$

$$b_t \geq 0$$

$$\mathbf{1}^{\top} \cdot x_t \leq 1 \qquad \text{for } t \in [0, T]$$

$$\mathbb{E}_0\left[\sum_{t=0}^{T}\beta^t u(c_t W_t)\;\middle|\;\tilde{\mu},\ \boldsymbol{\Sigma}\right]$$

$$\tilde{\mu} = \lambda\, \varphi - \sigma^2/2$$

subject to:

$$\pi_{t+1} := (\mathbf{R}_t{}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) + R_f\, b_t),$$

$$b_t = 1 - \mathbf{1}^{\top} \cdot (\mathbf{x}_t + \boldsymbol{\delta}_t) - c_t,$$

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

$$\boldsymbol{\delta}_t \geq -\mathbf{x}_t$$

$$b_t \geq 0$$

$$\mathbf{1}^{\top} \cdot x_t \leq 1 \qquad \text{for } t \in [0, T]$$