Machine Learning Algorithms for Financial Markets

Edoardo Vittori – Intesa Sanpaolo
Matteo Rampazzo – Epsilon SGR

20th February, AAAI 24

Algorithms in the Financial Markets

Algorithms are becoming increasingly prevalent in the financial markets

Market

- The global algorithmic trading market reached a value of more than $15B in 2023
- Market to grow at a CAGR of around 10%

Uses of algorithms

- Optimal execution
- Market making
- Hedging
- Trading
- Portfolio optimization

Advantages

- Reduce response time
- Reduce operational errors
- Analyze data flow in real-time

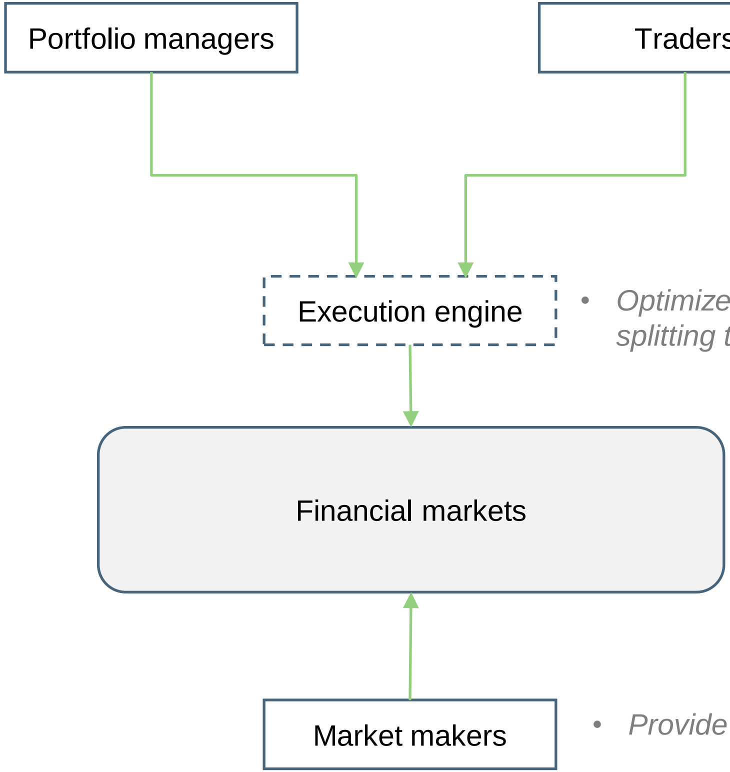
Challenges

- Overfitting
- Non-stationarity
- Simulating realistic markets

Schematic Overview of Financial Markets

Focus on the most influential actors

- *Decide the investment strategy*
- *Low frequency, large sizes*
- *Invest client liquidity*



Traders:

- *Decide trading strategy*
- *Higher frequencies, smaller sizes*
- *Invest own liquidity*

Execution engine:

- *Optimizes execution by splitting the order in time*

Financial markets can be:

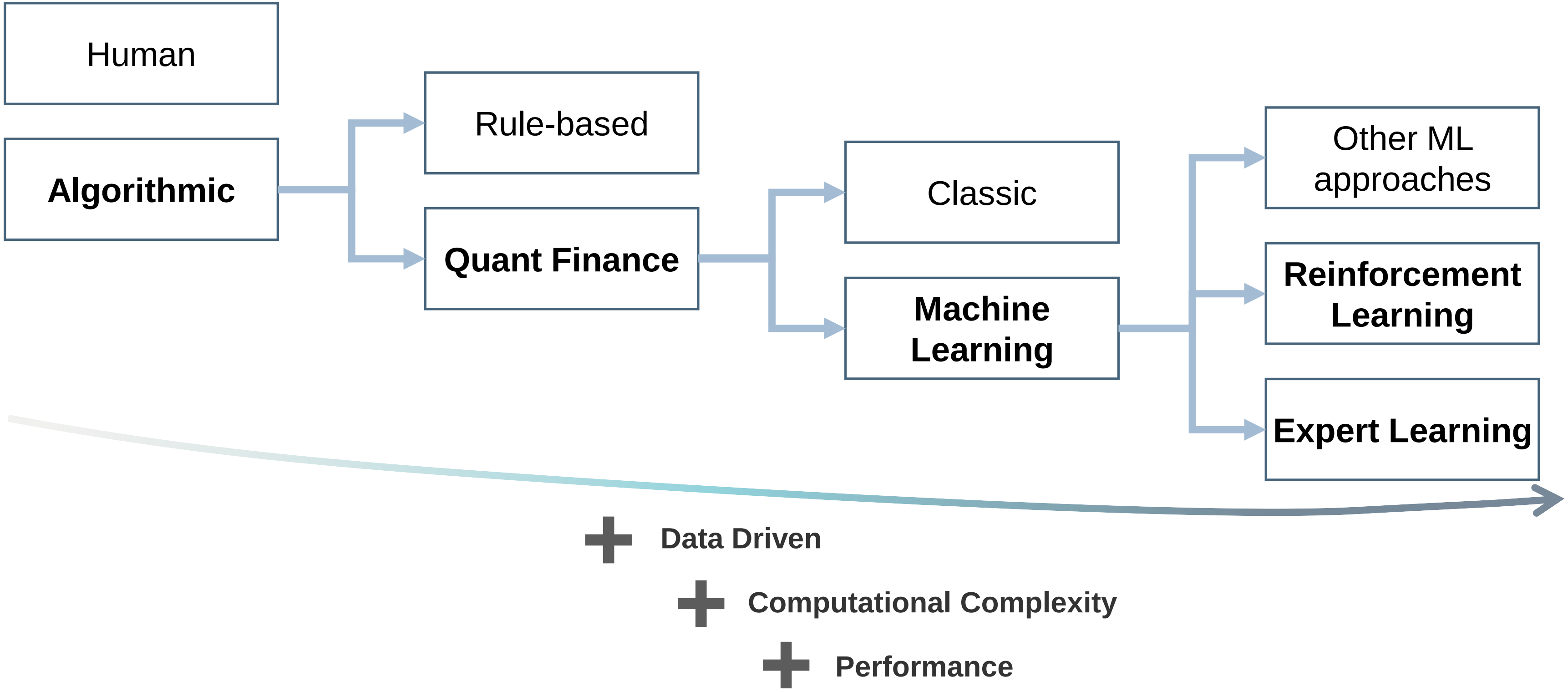
- *Regulated exchanges such as: NYSE, Nasdaq, LSE, Euronext*
- *Multi-lateral trade facilities*
- *Dark pools*

Market makers:

- *Provide liquidity to the financial markets*

Algorithmic Trading Technologies

Classification by technology type, with a focus on today's topics



Today’s Tutorial

We will focus on three financial problems

RL and EL Intro

- Reinforcement Learning
- Expert Learning

Quantitative Trading

- Introduction to quantitative trading
- Trading strategy with reinforcement learning

Quantitative Investing

- From classical portfolio theory to online learning
- Best practices to build a robust portfolio optimization framework
- A real application

Optimal Execution

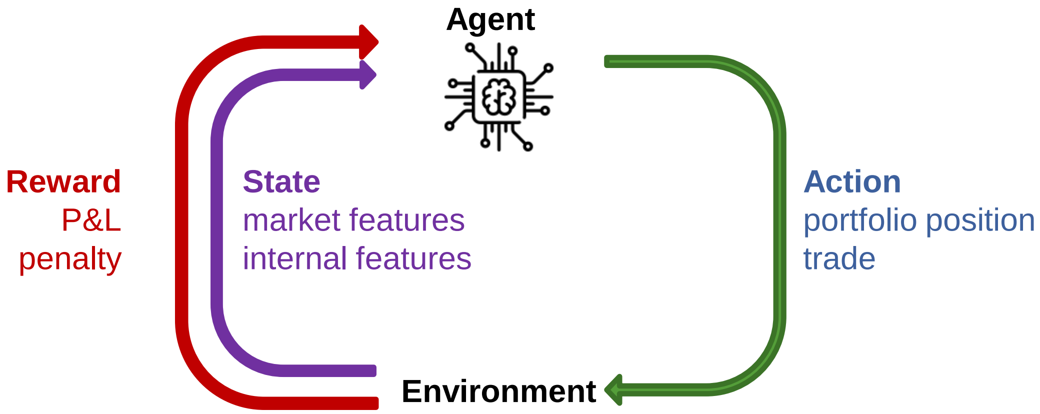
- The origins of price impact and the optimal execution setup
- Learning optimal execution

AGENDA

- **RL and EL Intro**
- Quantitative Trading
- Quantitative Investing
- Optimal Execution

Reinforcement Learning Basics

Markov Decision Process: process which describes interaction between agent and environment



- The objective is finding the policy π which maximizes the discounted sum of the rewards
- $J_\pi = \mathbb{E}_\pi[\sum \gamma^t R_t]$

Q-function and Policy

RL algorithms enable the learning of the policy π

The objective is to find the π that maximises $J_\pi = \mathbb{E}_\pi[\sum \gamma^t R_t]$

Q-learning

- Q-function

$$Q_\pi = \mathbb{E}_\pi[\sum \gamma^t R_t | s_0, a_0]$$

- Bellman Equation

$$Q_\pi = r(s,a) + \gamma \mathbb{E}_{s',a'}[Q_\pi(s',a')]$$

- Q-learning algorithm

$$Q_t(s,a) = r(s,a) + \gamma \max_{a'} Q_t(s',a')$$

- Q-learning is a tabular algorithm which can be generalized using function approximators such as Xgboost.

Policy Search

- Policy gradient theorem

$$\nabla_\theta J_{\pi_\theta} = \mathbb{E}[\nabla \log \pi_\theta(a|s) Q_{\pi_\theta}(s,a)]$$

- Policy update

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J_{\pi_\theta}$$

- The policy is a parametric and differentiable function, usually a neural network

Creating a Trading Strategy with RL

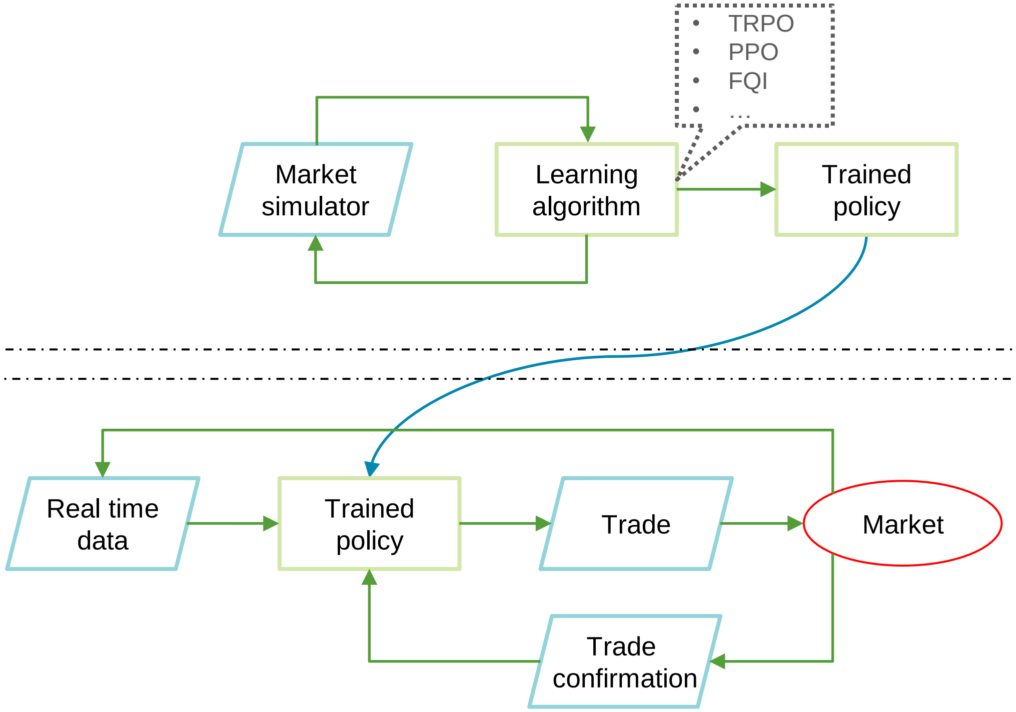
Training, testing and use in production

Phase 1

- Training
- Hyperparameter tuning
- Testing and evaluation

Phase 2

- Production



Expert Learning / Online Learning

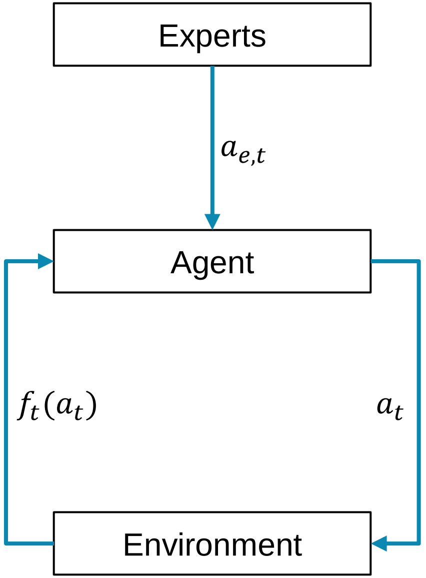
Algorithms which converge to the best expert

Characteristics

- Field of research close to RL
- Objective is to learn sequential decision processes
- Online algorithms
- Expert learning algorithms choose at each timestep which experts to follow
- Regret guarantees: finding the best expert in sub-linear time

- Regret $R_T = \sum_{t=1}^{T} f_t(a_t, y_t) - \inf_{e \in E} \sum_{t=1}^{T} f_t(a_{e,t}, y_t)$

Expert interaction scheme



An Example of Expert Learning Algorithms

Exponential Weighted Average (EWA)

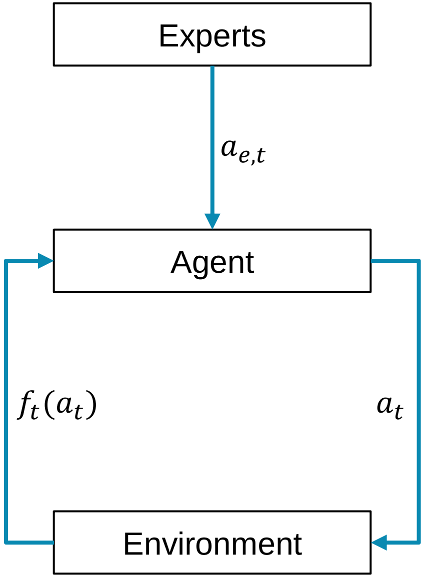
Pseudocode of EWA

- Initialize $\boldsymbol{w}_1 = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$ uniformly over the experts (strategies) and pick η
- For $t \in \{1, \dots, T\}$ do:
  - Collect experts' predictions $a_{e,t}$
  - Play $a = \frac{\sum_i w_i a_i}{\sum_i w_i}$
  - Observe loss $x_i = f_i(a_i)$ of each expert
  - Update weights with new information $w_{i,t} = w_{i,t-1} e^{-\eta * x_i}$

Characteristics

- The loss is a function of the current portfolio
- Regret $O(\sqrt{T \log(m)})$

Expert interaction scheme



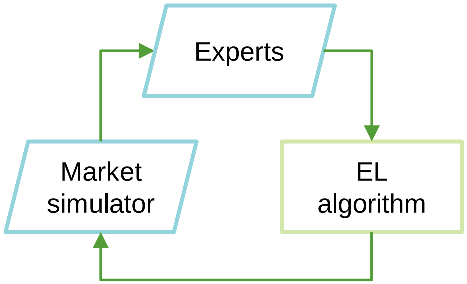
Cesa-Bianchi, Nicolo, and Gábor Lugosi. Prediction, learning, and games. Cambridge university press, 2006.

Creating a Trading Strategy with Expert Learning

Tuning and use in production

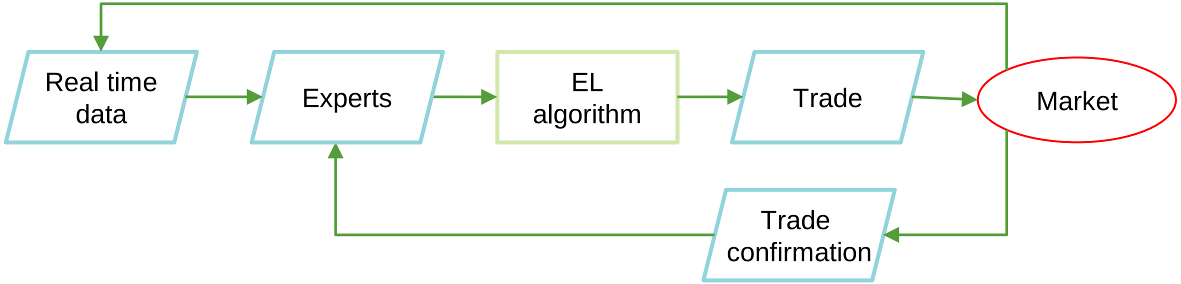
Phase 1

- Hyperparameter tuning
- Testing and evaluation



Phase 2

- Production



AGENDA

– RL and EL Intro
– Quantitative Trading
– Quantitative Investing
– Optimal Execution

Introduction to Quantitative Trading

Defining and building a quantitative trading strategy

Quant Trading Definition

Quantitative trading uses mathematical and statistical models to identify trading opportunities

Common quantitative trading strategies

- Momentum
- Mean-reversion
- Statistical arbitrage
- Seasonality
- Market making

Building a quant trading strategy

Objective	Financial assets, frequency, style
Data	Price, LOB, fundamental, economic, news
Build strategy	Define trading rules
Testing	Performance evaluation on historical data
Live tests	Performance evaluation on live data
Production	Connect to market via APIs, setup strategy on server

Rule-based Quantitative Trading Example

Mean reverting strategy - performance

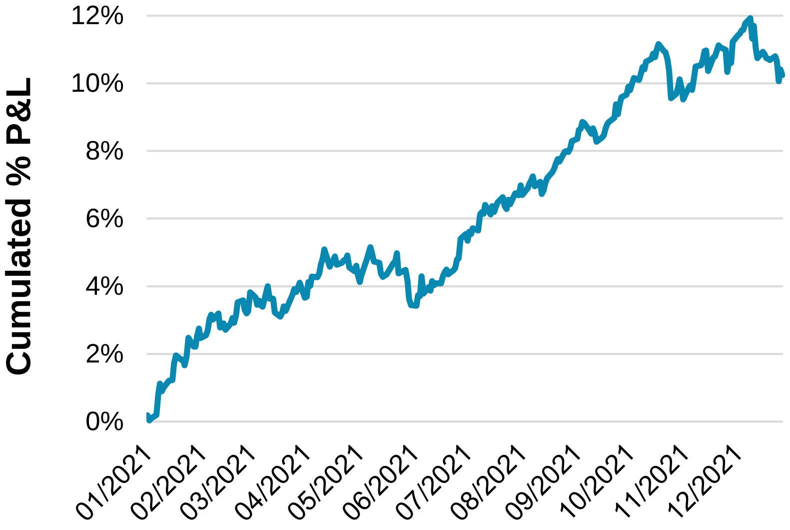
Strategy description

- Positioning $= -\sum_{i=0}^{T-2}(T-i-1)R_{t-i}$
- $\mathbb{E}(P\&L) = \frac{1}{2}(T\mathrm{Var}_1 - \mathrm{Var}_\mathrm{T} - \mu^2 T(T-1))$
- Asset: EURUSD FX spot
- Frequency: 10 minutes
- T = 120 minutes

Legend

T = time horizon in minutes
R = returns
Var_1 = 1-period variance
Var_T = T-period variance

P&L of backtest on 2021



Transaction costs

Each trade generates a cost proportional to the trade size

Example of LOB

Bid		Offer	
Volume	Price	Price	Volume
136	4044.50	4045.00	62
327	4044.00	4045.50	293
348	4043.50	4046.00	427
620	4043.00	4046.50	426
358	4042.50	4047.00	463
330	4042.00	4047.50	348
325	4041.50	4048.00	327
318	4041.00	4048.50	294
305	4040.50	4049.00	281
512	4040.00	4049.50	288

Defining Transaction Costs

- $\text{mid price} = \frac{1}{2}(\text{best offer} + \text{best bid})$
  - 4044.75
- $\text{spread} = (\text{best offer} - \text{best bid})$
  - 0.50
- $\text{transaction costs} = \text{trade size} * \frac{1}{2}\text{spread}$
- $\text{step p\&l} = \text{position} * \text{market movement} - \text{transaction costs}$

Rule-based Quantitative Trading Example

Mean reverting strategy – performance with costs

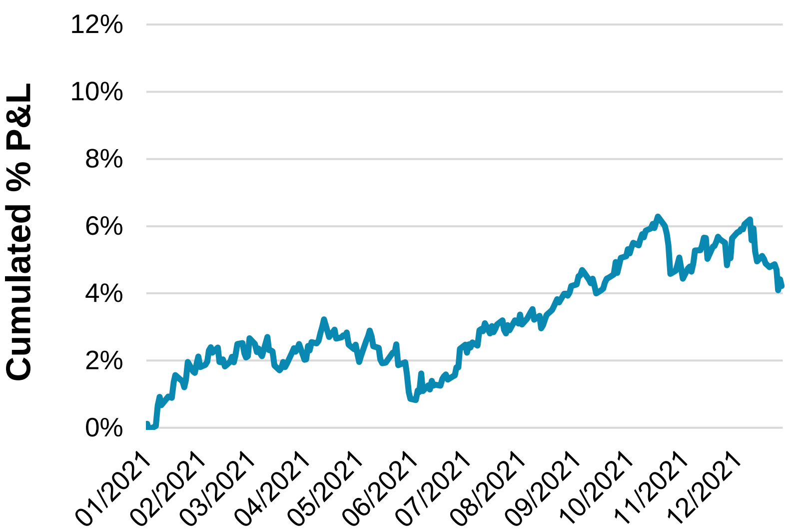
Strategy description

- Positioning $= -\sum_{i=0}^{T-2}(T-i-1)R_{t-i}$
- Asset: EURUSD FX spot
- Frequency: 10 minutes
- T = 60 minutes
- Transaction costs: $\frac{1}{2}\,\mathrm{spread}$

Can we improve?

- Consider costs when generating the strategy?
- Move on from a strictly defined trading rule?

P&L of backtest trading on 2021



Reinforcement Learning for Quantitative Trading

Problem description and MDP definition

Quantitative Trading

Definition

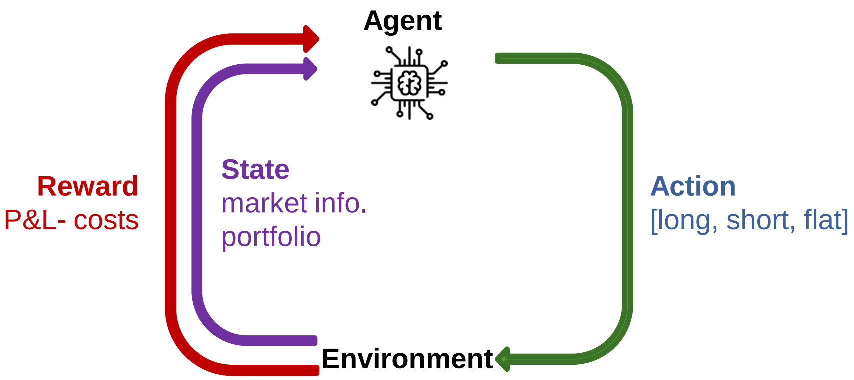
- At each timestep, decide whether to go long, short or flat to maximize gains

MDP

- **State:** price window, bid-ask spread, current portfolio, date/time
- **Action:** long, short, flat
- **Reward:** P&L – transaction costs

Characteristics

- Alpha seeking
- Low market correlation



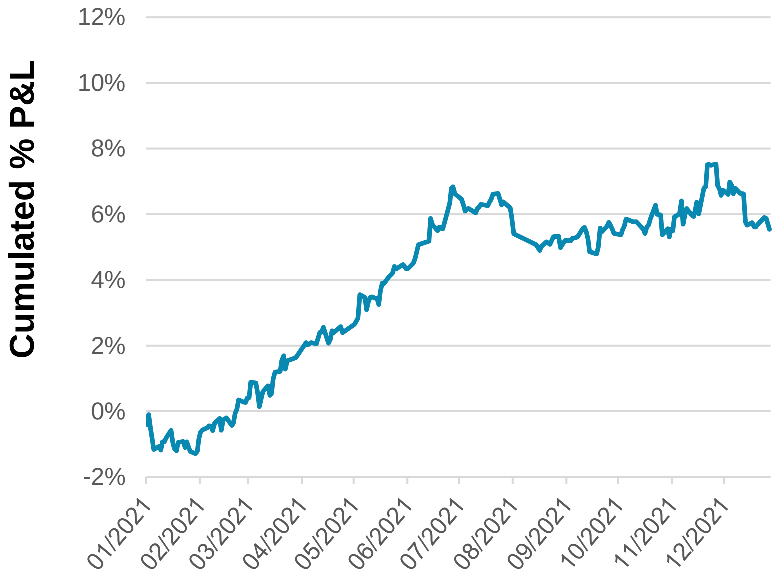
Reinforcement Learning for FX Trading (1/2)

Experimental results - performance

Experiment

- Intraday trading on EURUSD FX
- Training with reinforcement learning on historical data 2018-2019
- Validation on historical data 2020
- Backtesting on historical data out-of-sample 2021

P&L of backtest EURUSD FX trading on 2021



Learning FX Trading Strategies with FQI and Persistent Actions, ICAIF 2021

Reinforcement Learning for FX Trading (2/2)

Experimental results - policy

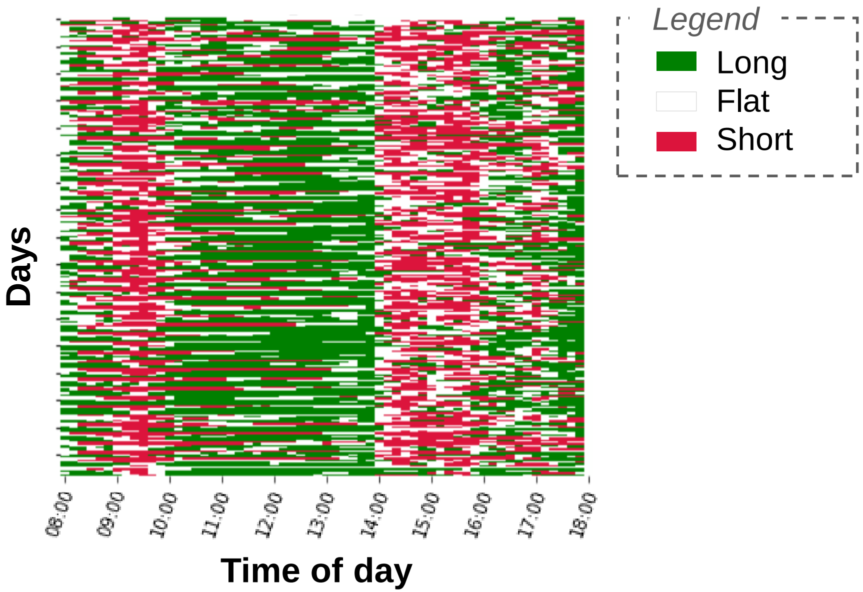
Experiment

- Intraday trading on EURUSD FX
- Training with reinforcement learning on historical data 2018-2019
- Validation on historical data 2020
- Backtesting on historical data out-of-sample 2021

Can we improve?

- Market non-stationarity

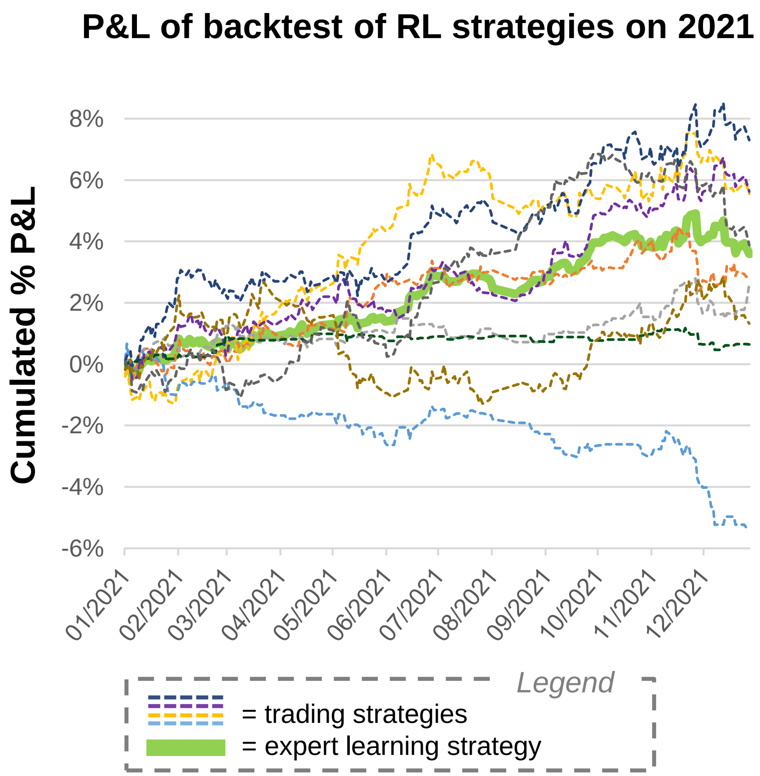
Actions chosen by agent

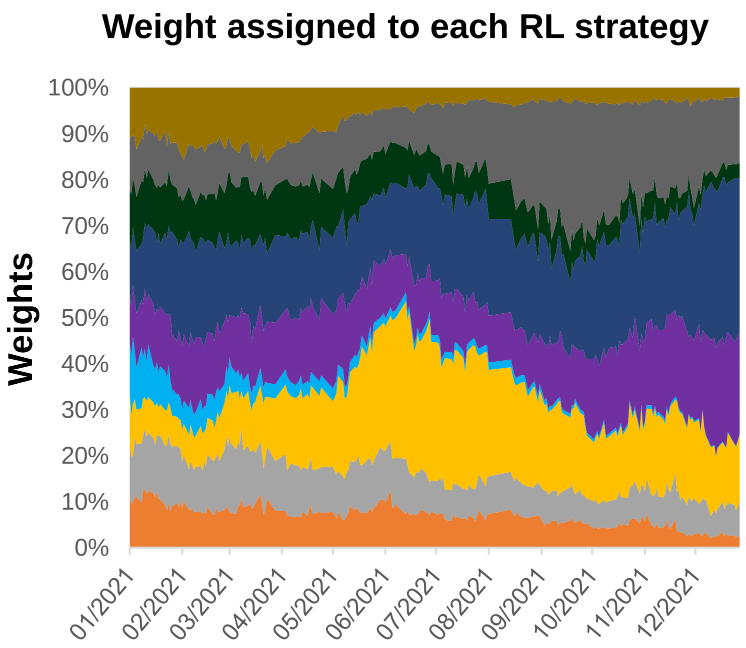


Learning FX Trading Strategies with FQI and Persistent Actions, ICAIF 2021

Reinforcement and Expert Learning for FX Trading

Experimental results - performance





Addressing Non-Stationarity in FX Trading with Online Model Selection of Offline RL Experts, ICAIF 2022

AGENDA

– RL and EL Intro

– Quantitative Trading

– Quantitative Investing

– Optimal Execution

What we'll cover today

INTRO

ASSET ALLOCATION
- Introduction
- Classical portfolio theory
- Pitfalls

ALGOS

ONLINE LEARNING
- A new paradigm
- Algorithms
- How machines can help

BEST PRACTICES

HOW TO AVOID OVERFITTING
- Model calibration
- Ensemble models
- Model evaluation

USE CASE

A REAL APPLICATION
- An algorithmic way to trade markets

AGENDA

The Asset Allocation Problem

Introduction

Objective

- Allocate funds among a set of assets to target a specific goal such as
  - having a balanced exposure on markets
  - minimizing risk and diversify investments
  - maximizing the portfolio return given a specific risk constraint

Naive Approaches

- Equally Weighted
- 60% Equity, 40% Bond
- 120 minus your age

Risk Models

- Minimum Variance
- Inverse volatility
- Equal Risk Contribution

Expected Returns Models

- Mean-Variance: Markowitz
- Risk Budget with Expected Returns

The Asset Allocation Problem

Introduction

Objective

- Allocate funds among a set of assets to target a specific goal such as
  - having a balanced exposure on markets
  - minimizing risk and diversify investments
  - maximizing the portfolio return given a specific risk constraint

Naive Approaches	Risk Models	Expected Returns Models
• Equally Weighted • 60% Equity, 40% Bond • 120 minus your age	• Minimum Variance • Inverse volatility • Equal Risk Contribution	• Mean-Variance: Markowitz • Risk Budget with Expected Returns

The Asset Allocation Problem

Introduction

Objective

- Allocate funds among a set of assets to target a specific goal such as
  - having a balanced exposure on markets
  - minimizing risk and diversify investments
  - maximizing the portfolio return given a specific risk constraint

Naive Approaches

- Equally Weighted
- 60% Equity, 40% Bond
- 120 minus your age

Risk Models

- Minimum Variance
- Inverse volatility
- Equal Risk Contribution

Expected Returns Models

- Mean-Variance: Markowitz
- Risk Budget with Expected Returns

The Asset Allocation Problem

Introduction

Objective

- Allocate funds among a set of assets to target a specific goal such as
  - having a balanced exposure on markets
  - minimizing risk and diversify investments
  - maximizing the portfolio return given a specific risk constraint

Naive Approaches	Risk Models	Expected Returns Models
• Equally Weighted • 60% Equity, 40% Bond • 120 minus your age	• Minimum Variance • Inverse volatility • Equal Risk Contribution	• **Mean-Variance: Markowitz** • Risk Budget with Expected Returns

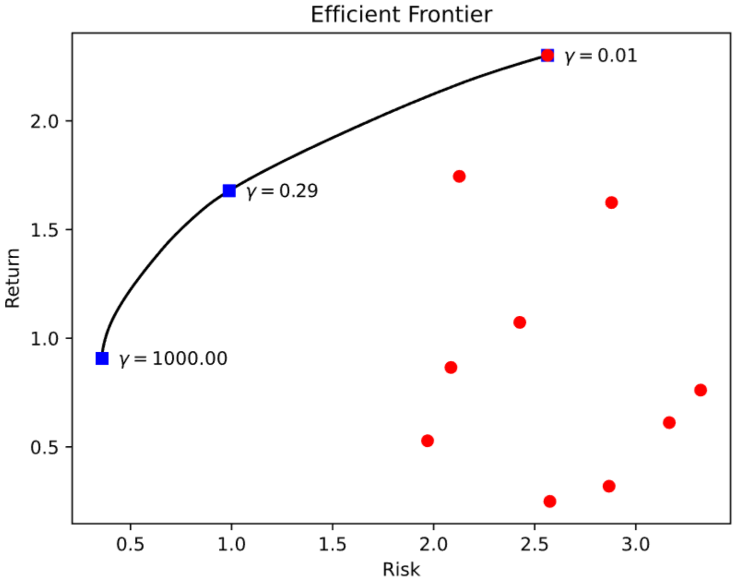
The Standard Approach

Markowitz

- Classical portfolio optimization maximize the risk-adjusted return

$$\max \ \mu^T w - \gamma w^T \Sigma w \ \ subject\ to\ 1^T w = 1, w \geq 0$$

- Pitfalls:
  - Outputs are highly sensitive to expected **returns estimates**
  - Variance-covariance matrix requires **lots of good data to be estimated**
  - Implicit assumption of **stable correlations**
  - **Single period** framework



Models in Finance

Old problems…

ISSUES

- Low signal-to-noise
- Reflexive and irrational markets
- Small data

SOLUTION

Models in Finance

…new tools

ISSUES

- Low signal-to-noise
- Reflexive and irrational markets
- Small data

SOLUTION

- Leverage models that
  - avoid the forecasting step
  - are adaptive to markets
  - don't need lots of data

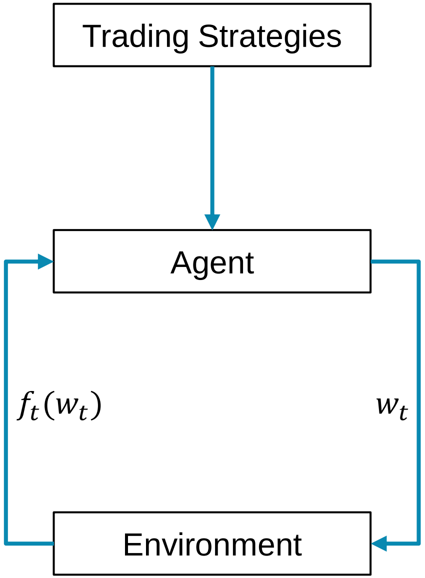
Online Learning for Portfolio Optimization

A new paradigm

- **Online Learning algos** applied to portfolio optimization aim at maximizing the portfolio's expected growth rate in a multi-period scenario

$$\max \sum_{t=1}^{n} \log w_t\, x_t \quad subject\ to\ \ 1^T w = 1, w \geq 0$$

- The general framework follows these steps:
  - Initialize weights: $w_1 = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$
  - For each time-period $t = 1, 2, \dots, n$:
    - Start from the current portfolio positioning: w_t
    - Observe strategy returns x_t and the portfolio loss $f_t(w_t)$ = - $\log w_t\, x_t$
    - Update the online portfolio weights



Online Learning for Portfolio Optimization

Algorithms: an example

Exponential Gradient

General

$$w_{t+1} = max\ \eta \log w\ x_t\ - R(w, w_t)$$

Helmbold

$$w_{t+1,i} = w_{t,i}\ exp\left(\eta \frac{x_{t,i}}{w_t\ x_t}\right) /Z,\ i = 1, \ldots, m$$

Idea

Track the asset with the best performance in last period but keep the new portfolio close to the previous portfolio

If the return of the asset is greater than current period wealth change then the proportion of the asset should be increased

Online Learning for Portfolio Optimization

Algorithms classification

Underlying Assumption

Follow the leader

Increase the relative weights of more successful assets

→ **Momentum**

Follow the loser

Increase the relative weights of less successful assets or transfer the weights from winners to losers

→ **Mean Reversion**

Pattern matching

Build a portfolio based on similar historical patterns with no explicit weights transfer directions

→ **Statistical Correlation**

Li and Hoi (2013), Online Portfolio Selection: A Survey

Models in Finance

Old problems…

ISSUES

- Low signal-to-noise
- Reflexive and irrational markets
- Small data

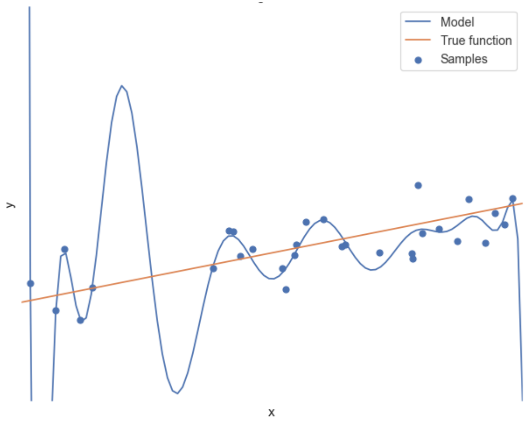
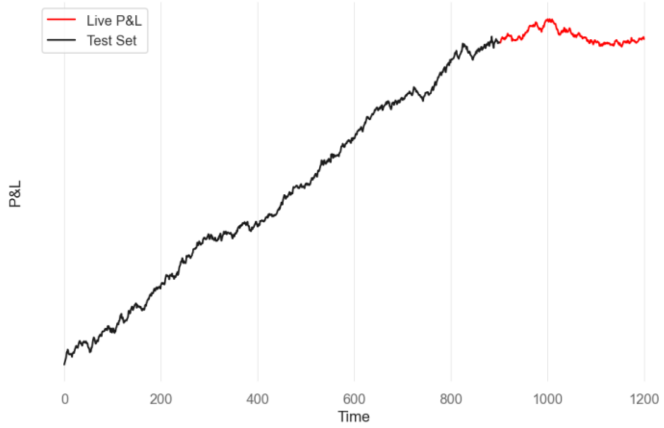
SOLUTION

- Leverage models that
  - avoid the forecasting step
  - are adaptive to markets
  - don't need lots of data

- Overfitting

Overfitting

Old problems…

	Train Set	Test Set
Problem	Fit too complex models to minimize train set errors with the risk of fitting noise instead of the signal	Fit models that have good performance on the test set but don't have good generalization properties on unseen data
Example		

Lopez De Prado (2020), *Machine Learning for Asset Managers*, *Cambridge Elements*

Models in Finance

…new tools

ISSUES	SOLUTION
• Low signal-to-noise • Reflexive and irrational markets • Small data	• Leverage models that ○ avoid the forecasting step ○ are adaptive to markets ○ don’t need lots of data
• Overfitting	• Leverage robust approaches ○ Simulation and synthetic data ○ Ensemble and stacking models ○ Bootstrapping and synthetic data

AGENDA

01 From classical portfolio theory to online learning

02 Best practices to build a robust portfolio optimization framework

03 Use case

Model Calibration

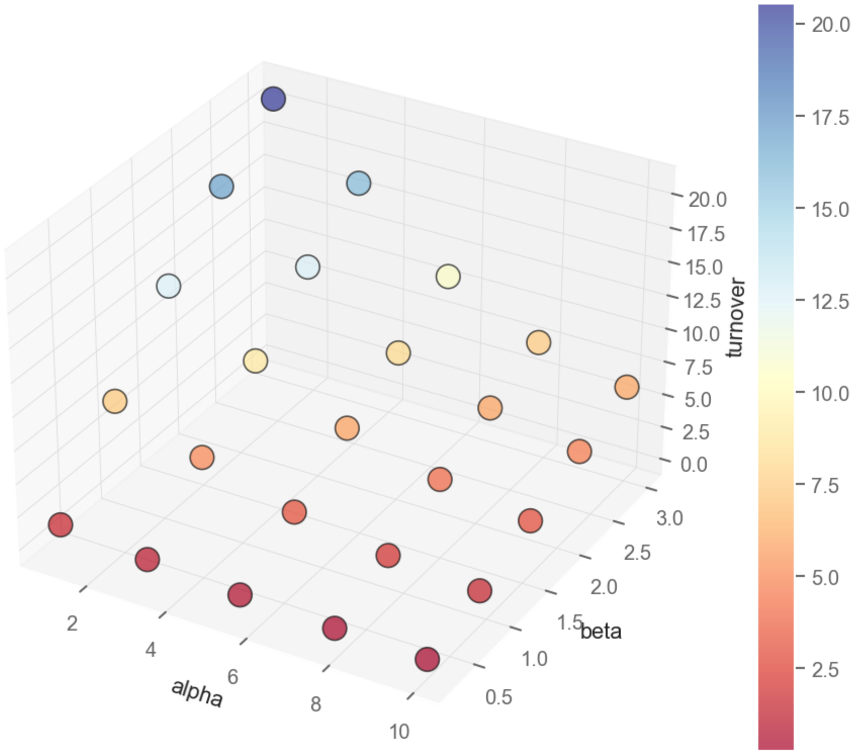
A robust approach to parameter tuning

Objective

- Find parameters domain optimizing the trade-off between **alpha generation, costs and models behavior**

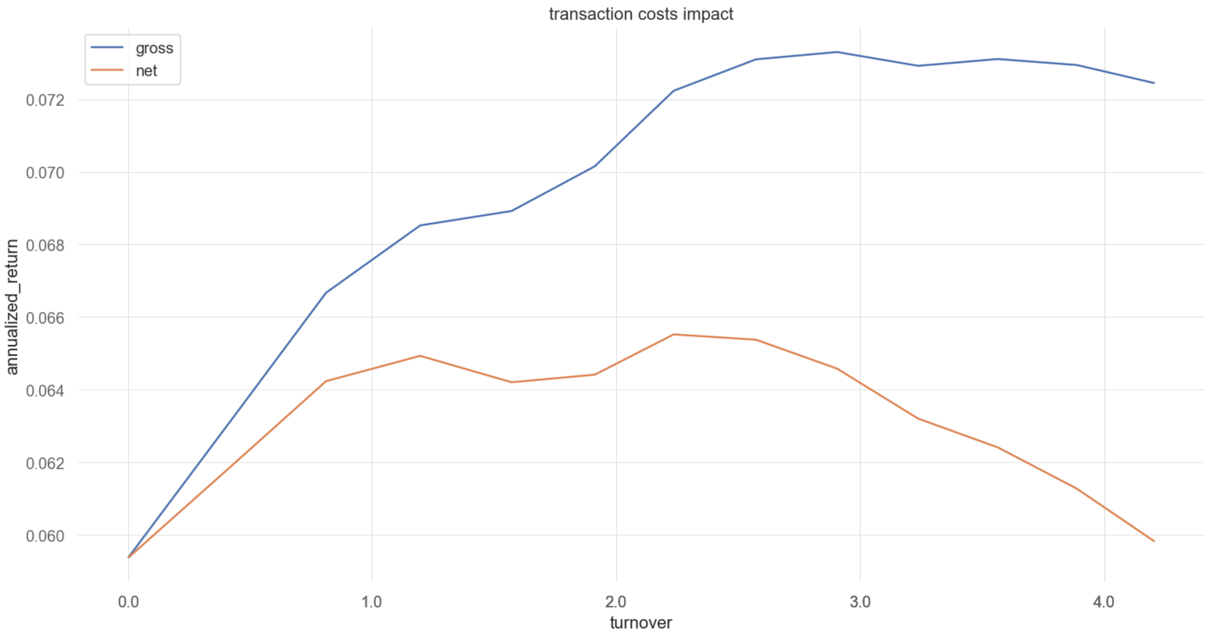
Methodology

- Generate **synthetic data** of the investable universe via Monte Carlo **simulation**
- Do a grid search over the parameters monitoring **turnover dynamic**
- Select a **suitable parameter domain**



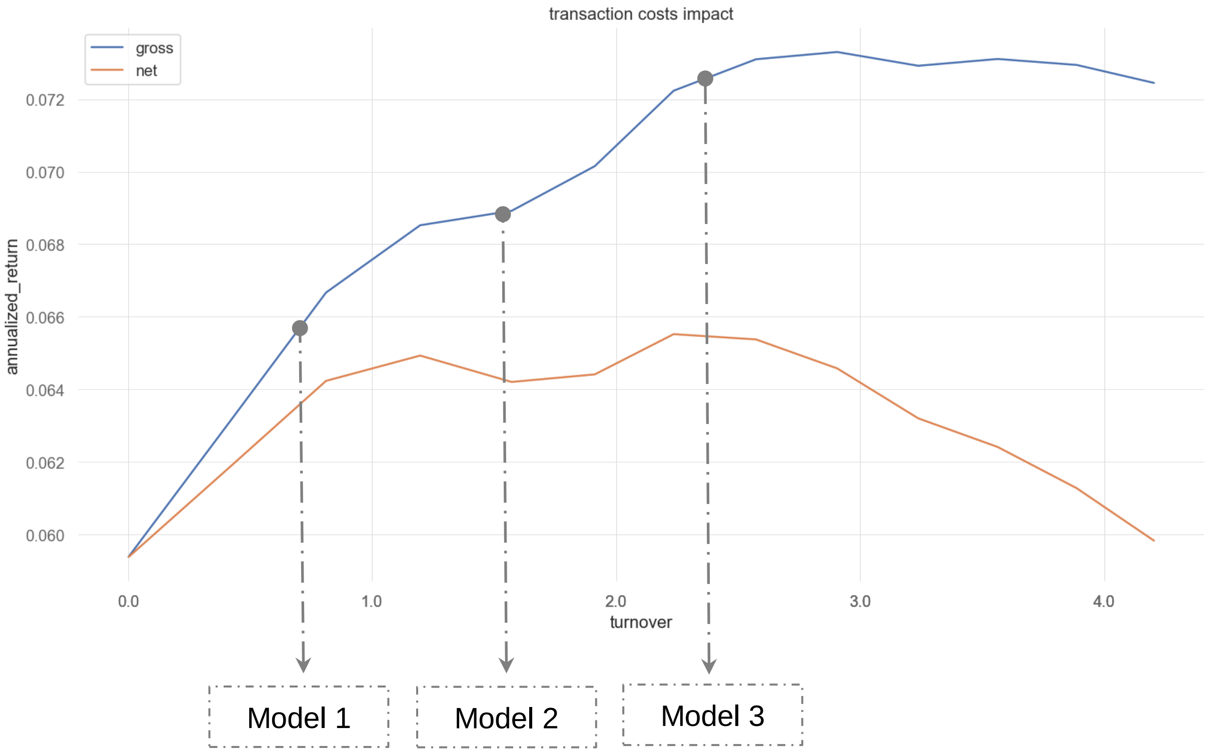
Why is Turnover Important?

Transactions costs



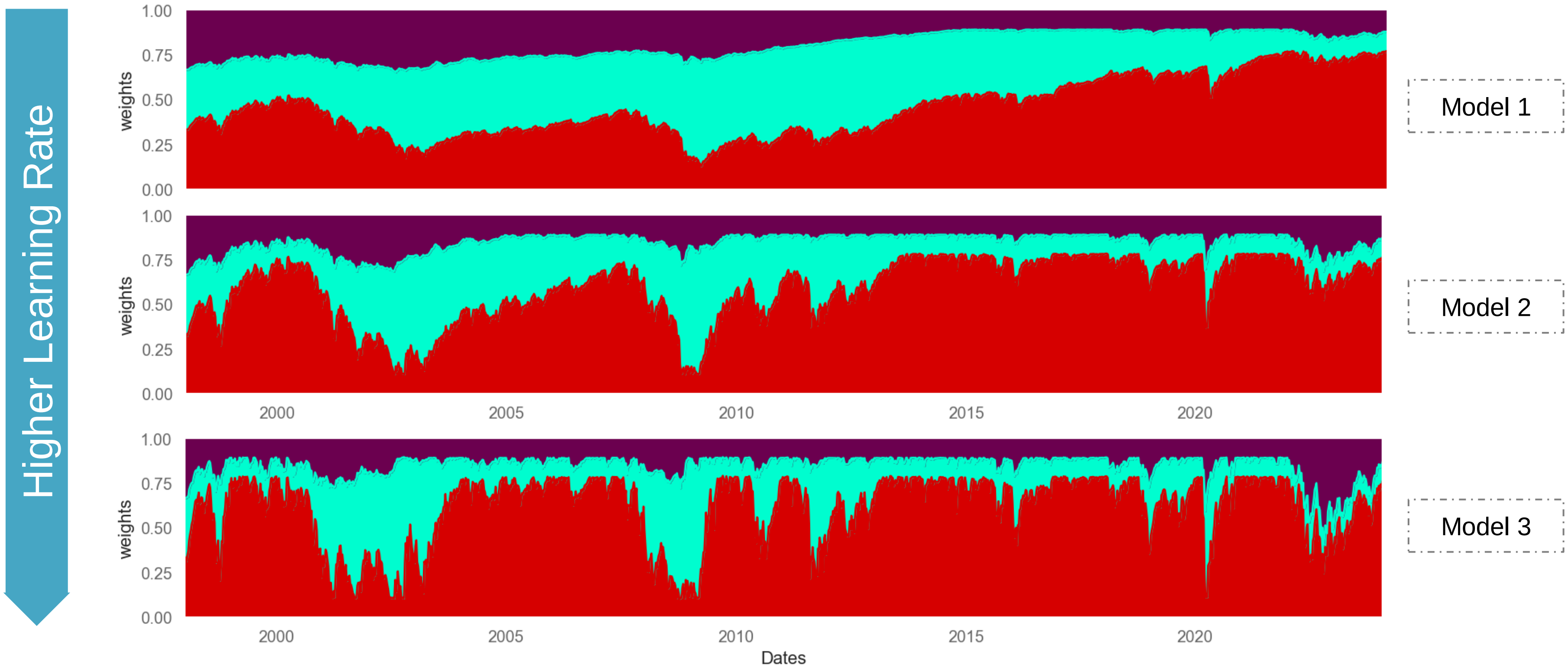
Why is Turnover Important?

Transactions costs



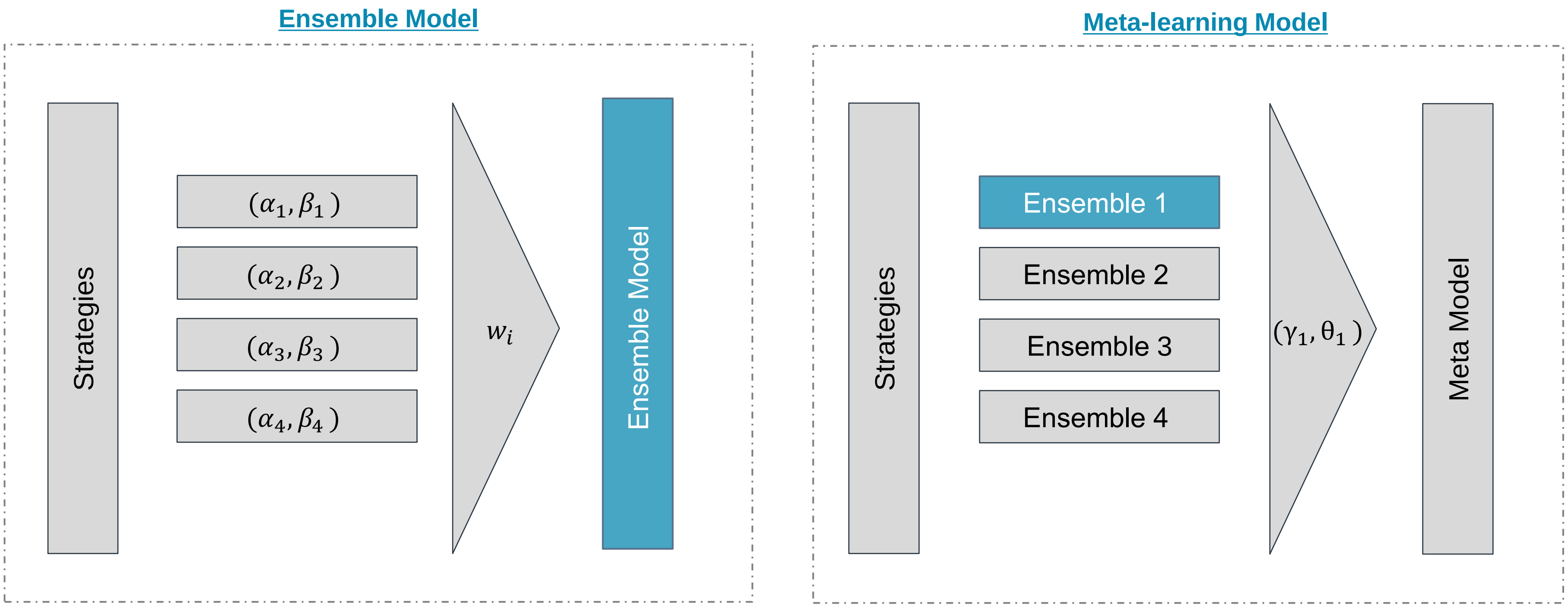
Why is Turnover Important?

Learning rates and market regimes



Ensemble Models

Combining weak learners



(x, y) = model with parameters x and y

Model Evaluation

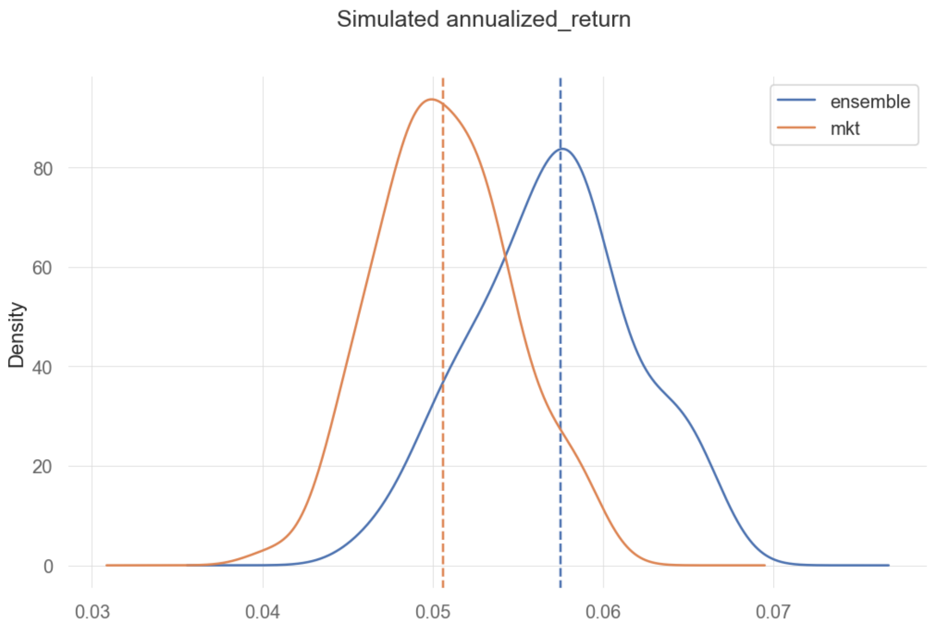
Bootstrapping

Objective

- Check **model robustness** by looking at results distributions instead of a single point estimate

Methodology

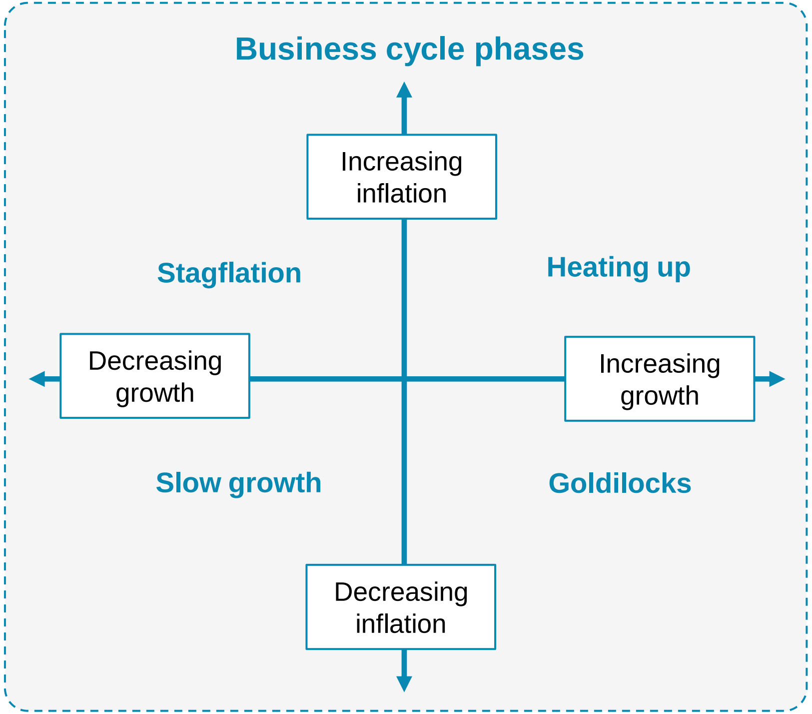
- Run multiple simulations from observed data each time removing a **random** x% of the time series and thus creating a **synthetic dataset**
- Plot and evaluate **results distributions**



AGENDA

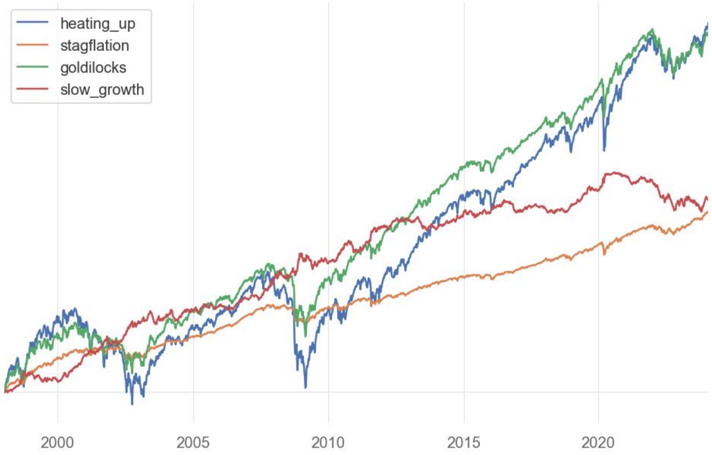
An Algorithmic Way to Trade Markets

Methodology



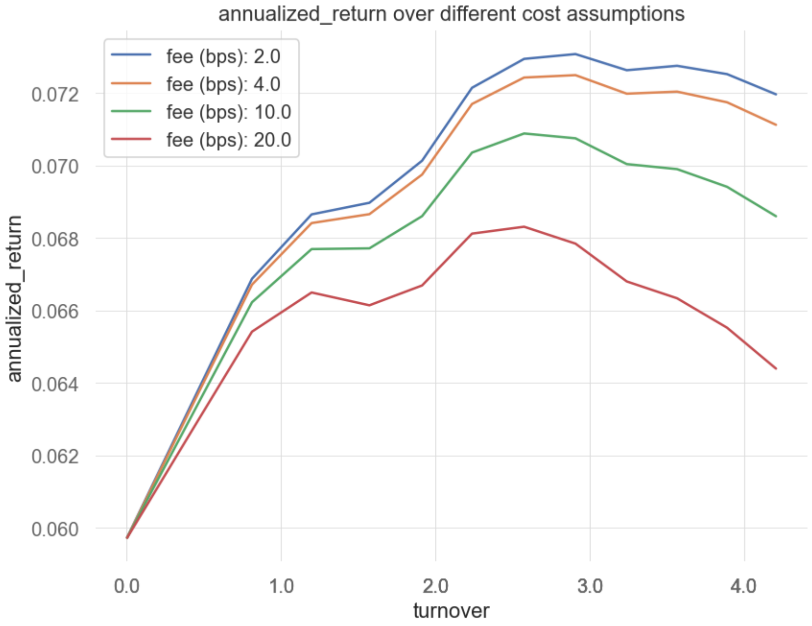
Methodology

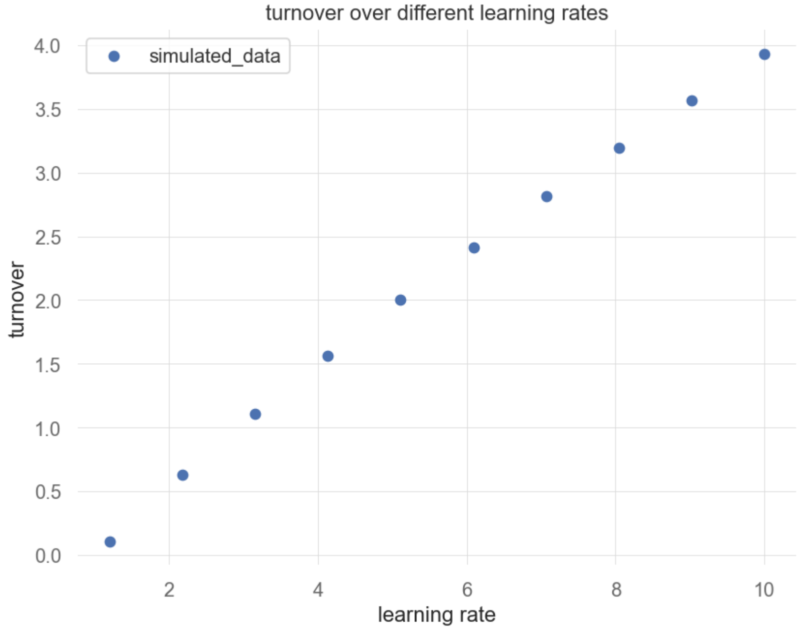
- Split markets into 4 business cycle phases
- Define one strategy for each regime
- Apply online algos to dynamically weight strategies



An Algorithmic Way to Trade Markets

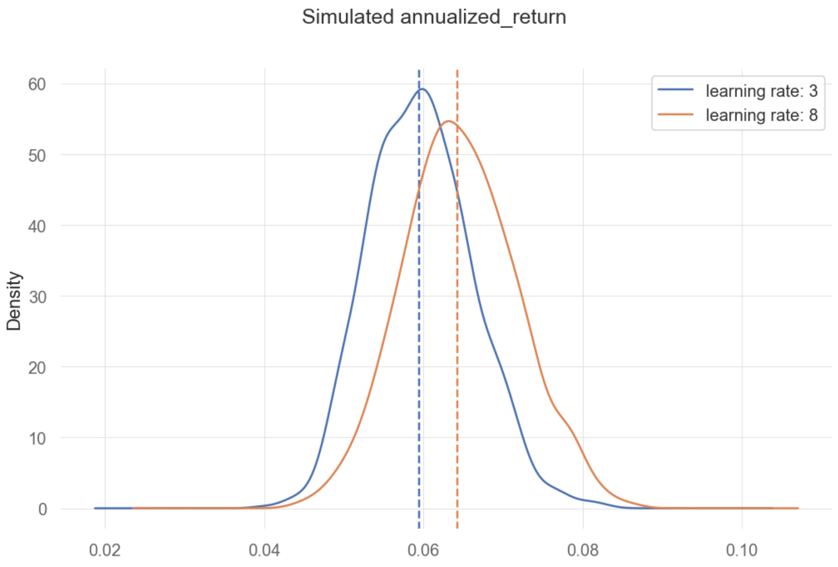
Model calibration

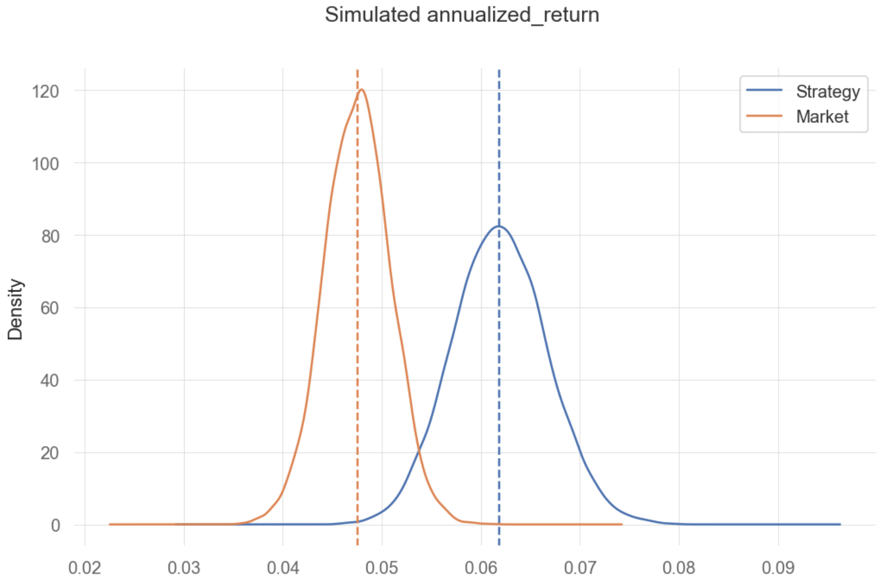




An Algorithmic Way to Trade Markets

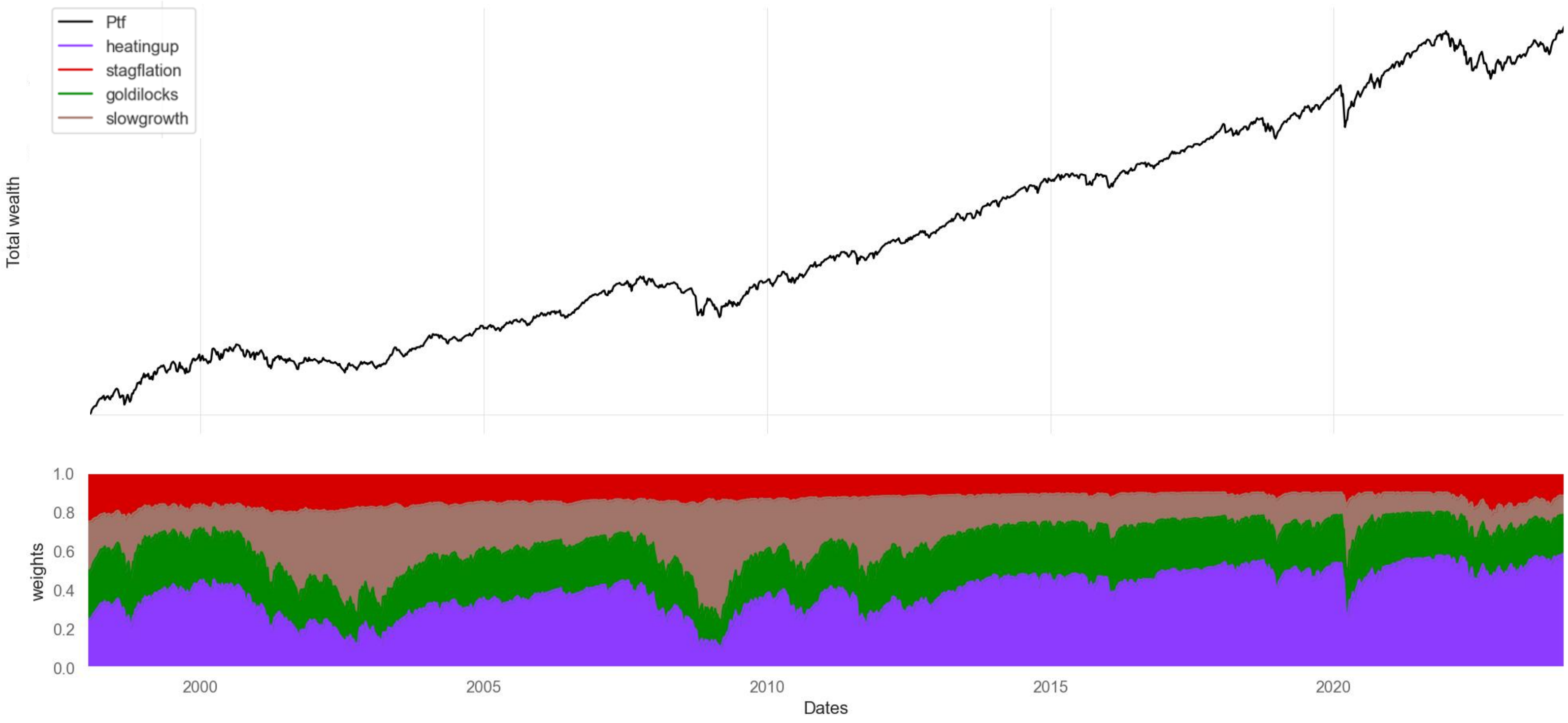
Model evaluation





An Algorithmic Way to Trade Markets

Strategy P&L



Summary

A modern approach to disentangle markets complexity

ISSUES

- Low signal-to-noise
- Reflexive and irrational markets
- Small data

SOLUTION

- Leverage models that
  - avoid the forecasting step
  - are adaptive to markets
  - don't need lots of data

ISSUES

- Overfitting

SOLUTION

- Leverage robust approaches
  - Simulation and synthetic data
  - Ensemble and stacking models
  - Bootstrapping and synthetic data

AGENDA

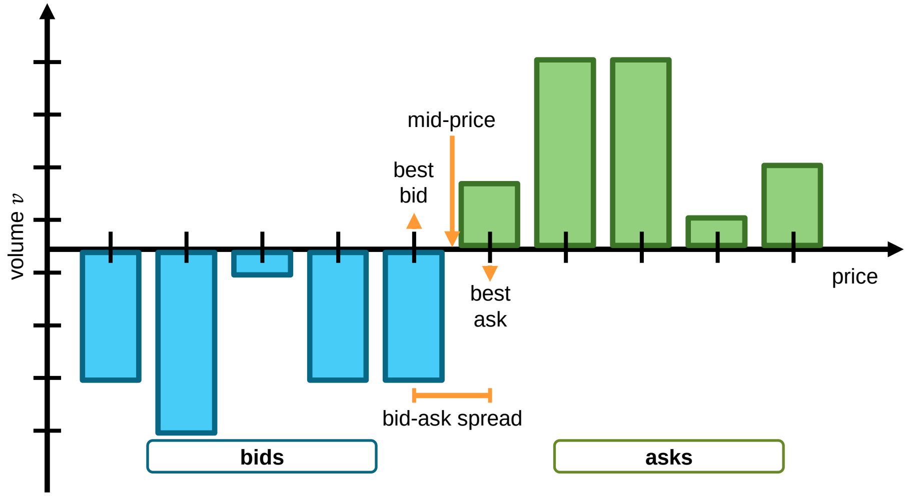
- – RL and EL Intro
- – Quantitative Trading
- – Quantitative Investing
- – Optimal Execution

Introduction to the Limit Order Book

Market and limit orders

Order types

- **Market order** is an order to execute immediately at the best price possible
- **Limit order** is an order that specifies both the price and volume of a trade
- A limit order sits in the order book until it is either executed against a matching market order or **canceled**



Example of Limit Order Book

Origins of Price Impact

Immediate market impact

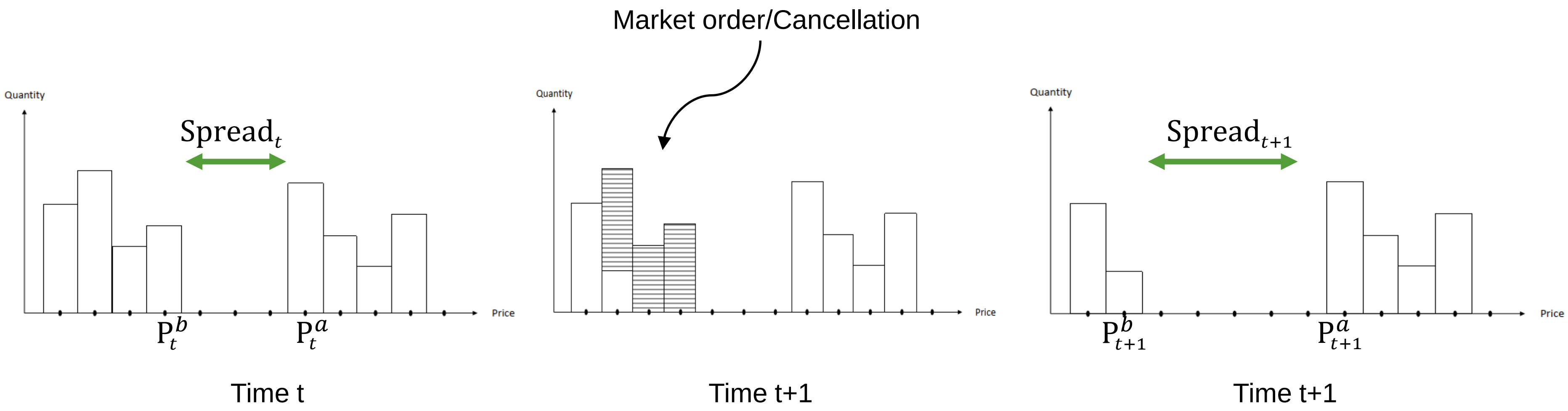
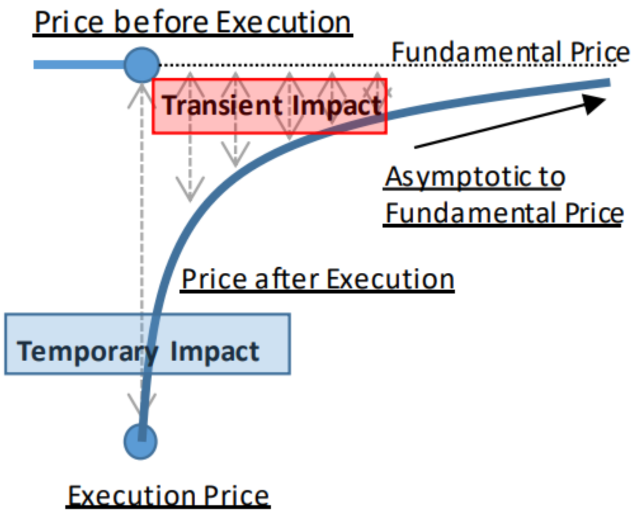


Illustration of an **immediate** market impact

Origins of Price Impact

Temporary and permanent market impact



Obizhaeva Wang model of an exponential **transient** market impact

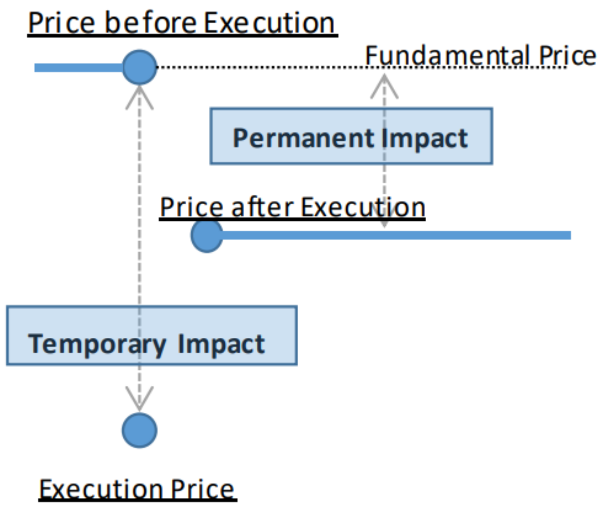


Illustration of a **permanent** market impact

Dynamic Optimal Execution Models with Transient Market Impact and Downside risk, Ono et al. (2017)

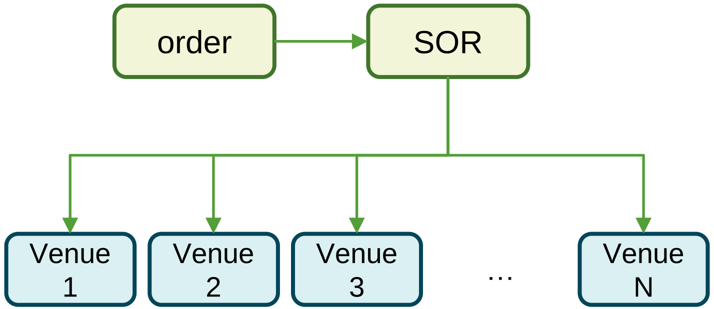
Optimal Execution Setup

The three layers of the optimal execution problem

Goal: Sell (buy) shares $x_0 > 0$ by time T>0.

- An execution algorithm has three layers:
  - **At the highest level**: one decides how to slice the order, when to trade, in what size, and for how long.
  - **At the mid-level**: given a slice, one decides whether to place market or limit orders and at what price level(s).
  - **At the lowest level**: given a limit or market order, one decides to which venue(s) should this order be routed.

Smart Order Routing



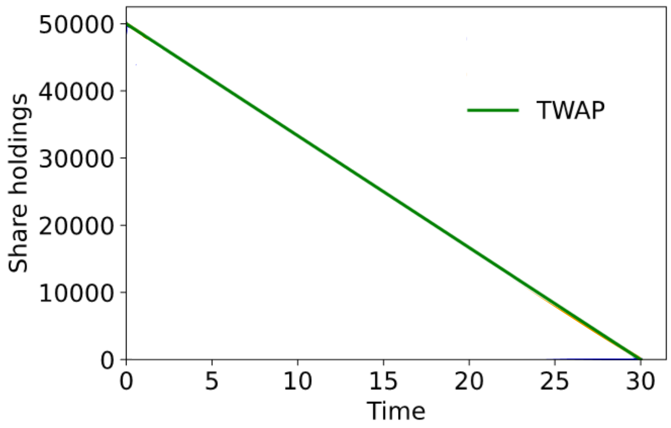
Optimal execution setup

Optimal Execution Set-Up

Goal: Sell (buy) shares $x_0 > 0$ by time T>0 with N>0 timesteps.

- $X = (X_t)_{0 \le t \le T}$ the execution strategy.
- X_t is the inventory at time t where $X_0 = x_0$ and $X_T = 0$.
- $\widetilde{P}_t = \left(\widetilde{P}_t\right)_{0 \le t \le T}$ the transaction price.
- $\mathcal{R}(X) = -\int_0^T \widetilde{P}_t dX_t$ the generated revenue.
- TWAP: $(X_t) = \frac{x_0}{\mathrm{N}} \; \forall t$

Execution trajectories



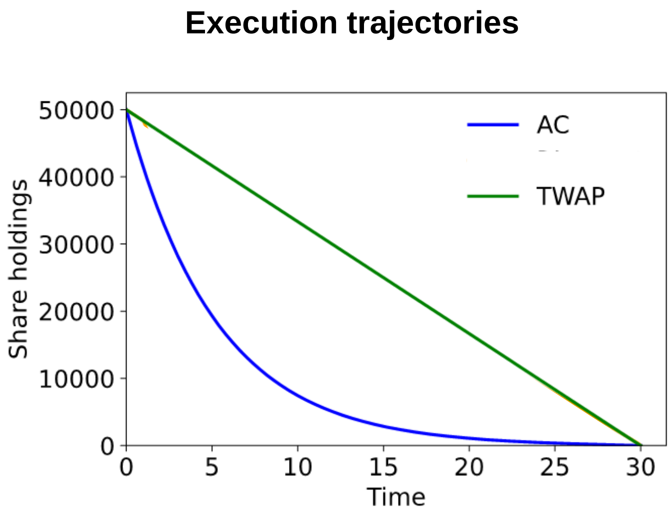
The Almgren-Chriss Model

A classical approach to the optimal execution problem

Almgren Chriss

- $P_t = (P_t)_{0 \le t \le T}$ the **observed**.
- $S_t = (S_t)_{0 \le t \le T}$ the **unaffected** mid-price (Becherer et al. 2018).
- $I_t = (I_t)_{0 \le t \le T}$ the price impact: $I_t = P_t - S_t$
- Linear Impact Almgren-Chriss Model

$$I_t = \gamma[X_t - X_0] + \lambda \dot{X}_t$$



Beyond the Almgren-Chriss Model with RL

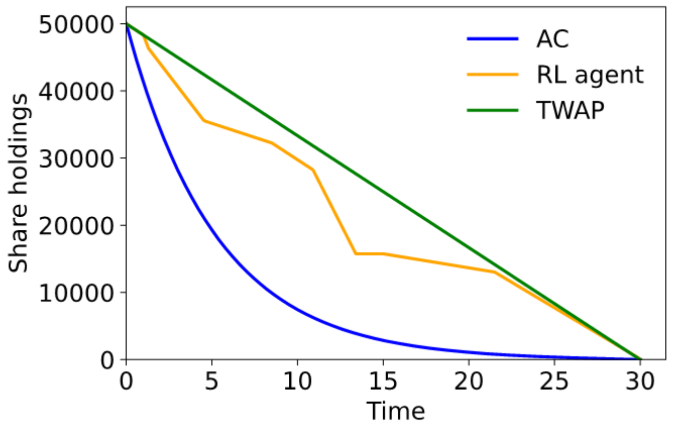
Can we learn a strategy which is not a deterministic function of time remaining?

Almgren Chriss

- $P_t = (P_t)_{0 \le t \le T}$ the **observed**.
- $S_t = (S_t)_{0 \le t \le T}$ the **unaffected** mid-price (Becherer et al. 2018).
- $I_t = (I_t)_{0 \le t \le T}$ the price impact: $I_t = P_t - S_t$
- Linear Impact Almgren-Chriss Model

$$I_t = \gamma[X_t - X_0] + \lambda \dot{X}_t$$

Execution trajectories



Augmenting Traders with Learning Machines, PhD Thesis, Edoardo Vittori, 2022

Reinforcement Learning for Optimal Execution

Problem definition and MDP description

Optimal Execution

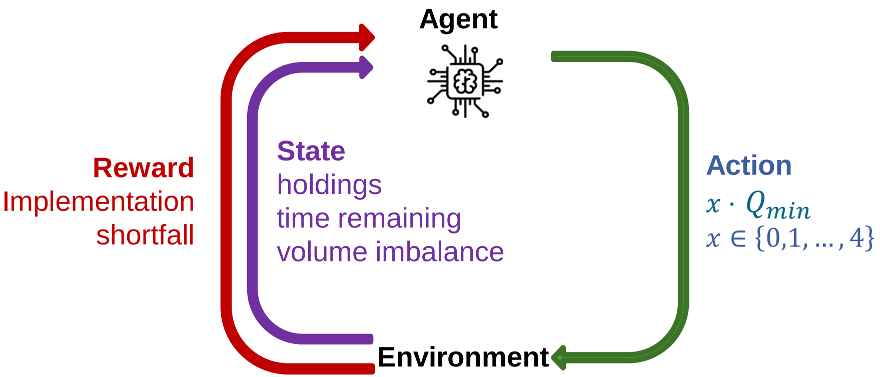
Definition

- Execute X shares in N timesteps
- Decide at each timestep the trade to execute to minimize the difference between arrival and execution price

MDP

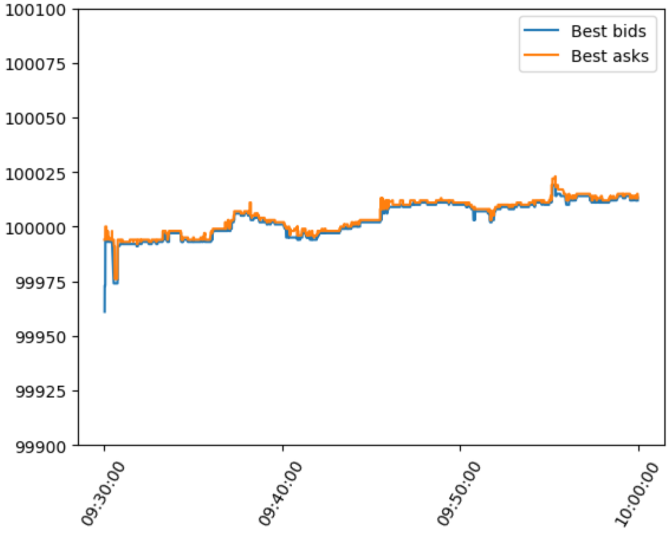
- **State:** Percentage holdings remaining, percentage time remaining, volume imbalance up to 5 levels of the limit order book, best bid price, best ask price.
- **Action:** do nothing, market order $Q_k = Q_{min}$ x k, k = {1, …, 4}.
- **Reward:**

$$r_t = \underbrace{Q_t^k \times (P_0 - P_t)}_{\text{implementation shortfall}} - \underbrace{\alpha d_t}_{\text{penalty}}$$

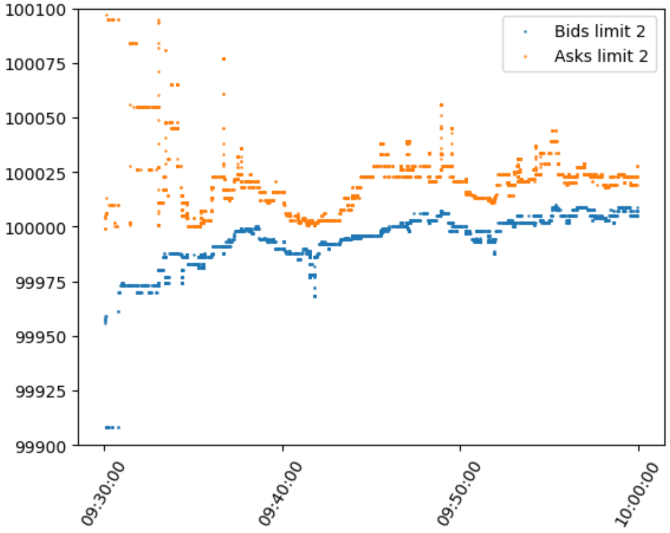


Limit Order Book Simulator

ABIDES – a multi-agent market simulator



Best Bid and Ask Prices from ABIDES Simulation.



Second Limit Bid and Ask Prices from ABIDES Simulation

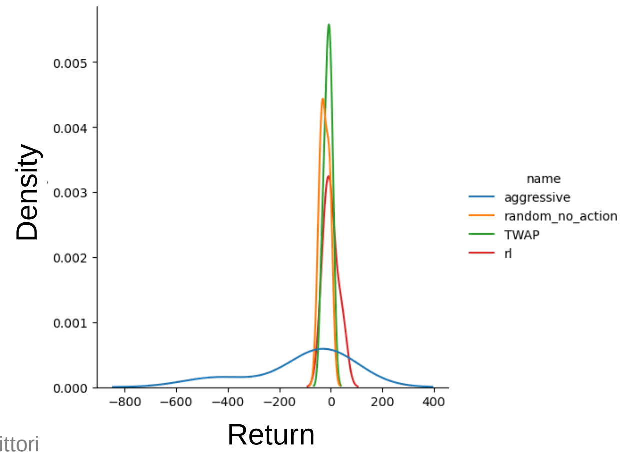
Experimental Results

Return comparison between RL agent and benchmark on a market simulated with ABIDES

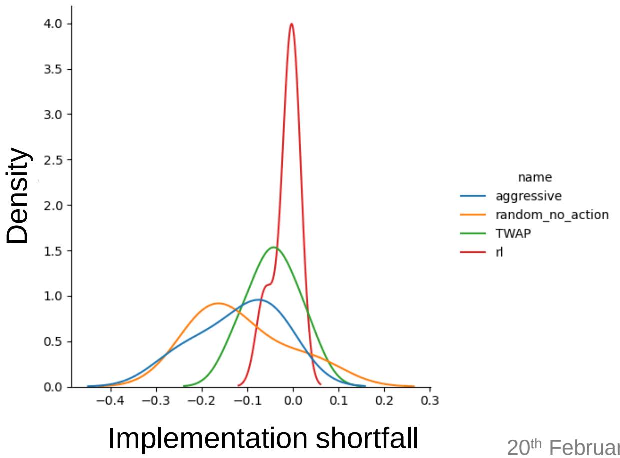
Characteristics

- Simulating with ABIDES the optimal execution exercise for 4 hours to execute 20k shares.
- $r_t = \underbrace{Q_t^k \times (P_0 - P_t)}_{\text{implementation shortfall}} - \underbrace{\alpha d_t}_{\text{penalty}}$

Return distribution of a DQN agent compared to other benchmark strategies



Implementation shortfall reward distribution of a DQN agent compared to other benchmark strategies



Q&A

Edoardo Vittori
edoardo.vittori@intesasanpaolo.com

Matteo Rampazzo
matteo.rampazzo@epsilonsgr.it