Machine Learning Algorithms for Financial Markets

Edoardo Vittori

May 4th, 2024

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

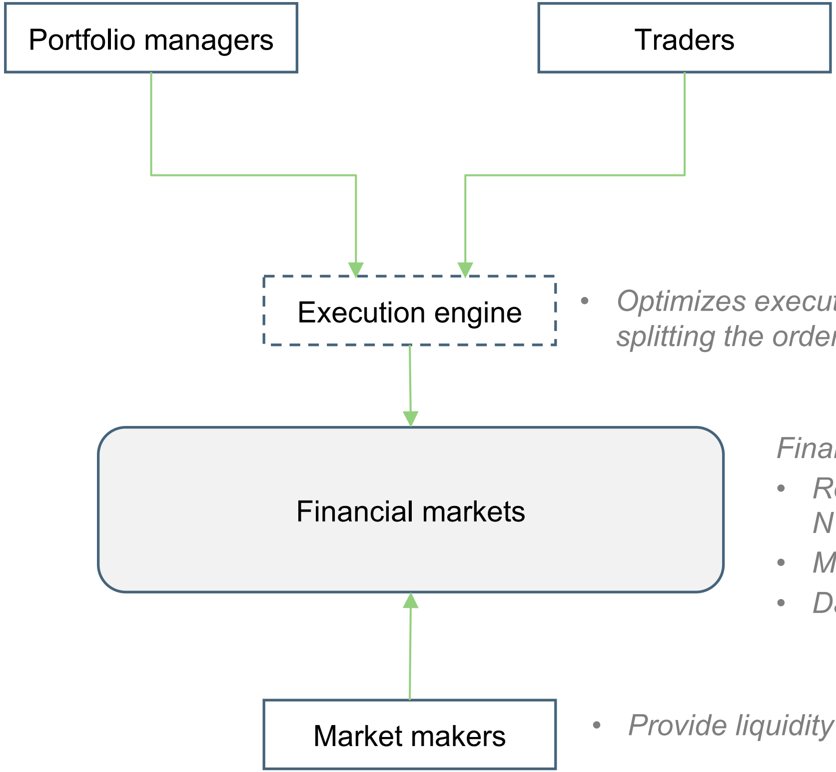
- Reduce response time
- Reduce operational errors
- Analyze data flow in real-time

Challenges

- Generalizing strategies
- Non-stationarity
- Simulating realistic markets

Schematic Overview of Financial Markets

Focus on the most influential actors



Portfolio managers:

- *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:

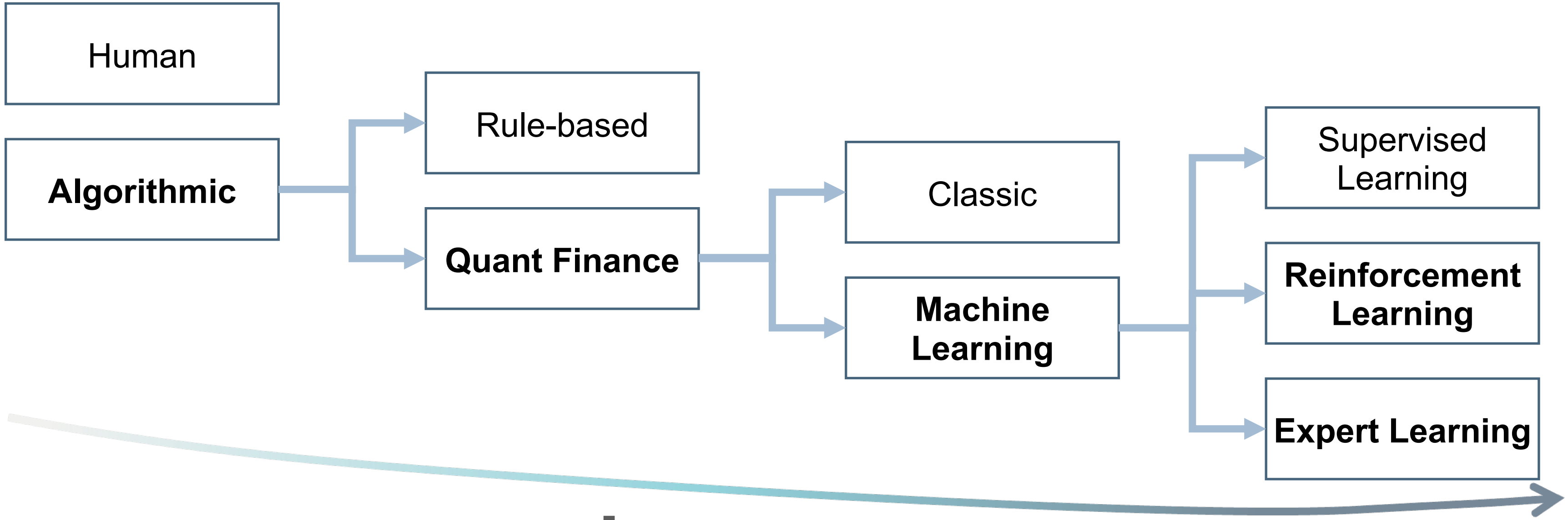
- *Regulated exchanges such as: NYSE, Nasdaq, LSE, Euronext*
- *Multi-lateral trade facilities*
- *Dark pools*

Market makers:

- *Provide liquidity to the financial markets*

Algorithmic Trading Technologies

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



- ➕ **Data Driven**
- ➕ **Computational Complexity**
- ➕ **Performance**

Today's Presentation

There are 3 main sections

ML toolkit

- Intro to ML
- Reinforcement learning
- Expert learning
- Hyperparameter tuning

Quantitative Trading

- Introduction
- An example
- Intraday trading with ML

Use case

- Intraday FX trading with ML

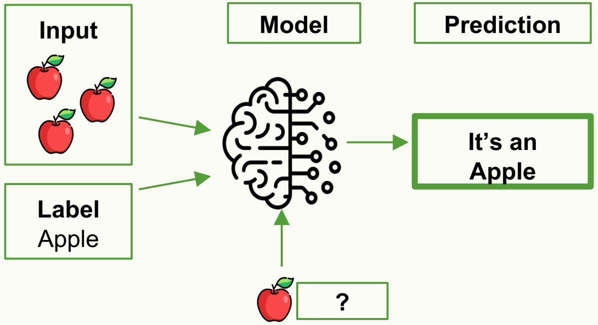
AGENDA

- **ML toolkit**
- Quantitative Trading
- Use case

Introduction to Machine Learning

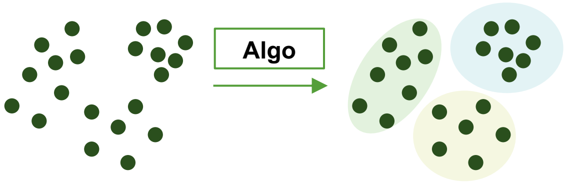
There are three main paradigms in ML

Supervised Learning



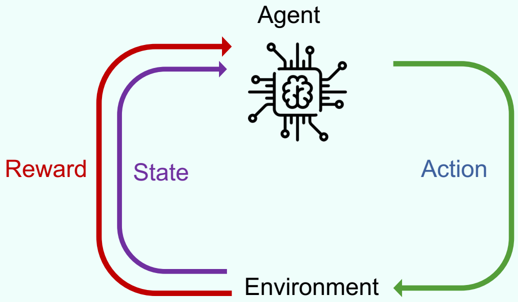
Usage
- Regression
- Classification

Unsupervised Learning



Usage
- Clustering
- Dimensionality reduction
- Feature extraction

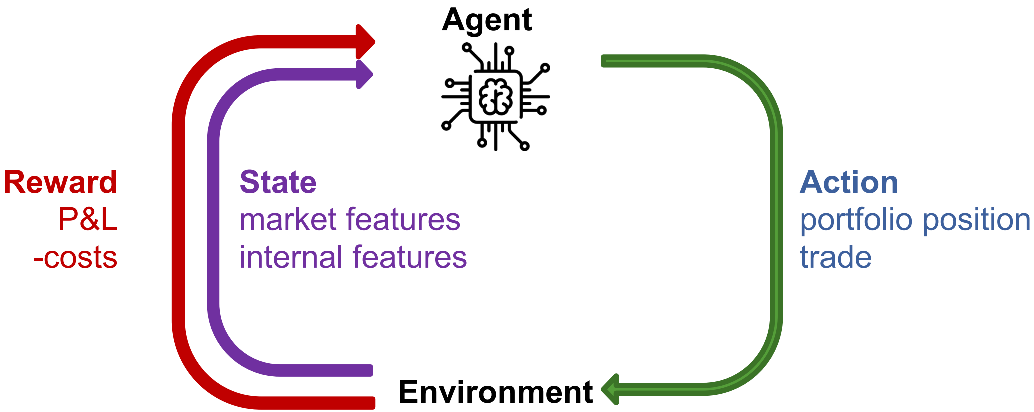
Reinforcement Learning



Usage
- Optimal Control
- Game playing

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 Search

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

Expert Learning

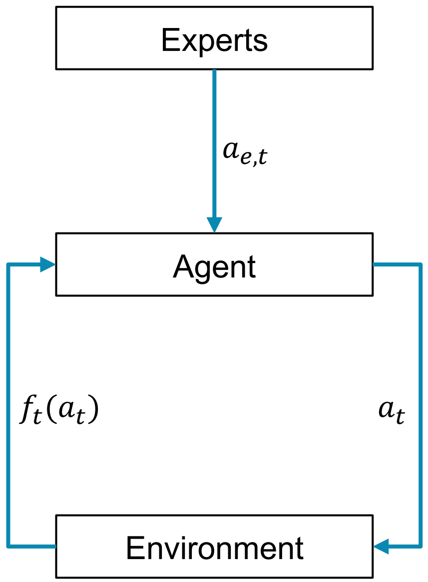
Algorithms which converge to the best expert

Characteristics

- Field of research close to RL
- Objective is to learn sequential decision processes
- Online algorithms with no training phase
- Expert learning algorithms choose at each timestep which experts to follow
- Regret guarantees: finding the best expert in sub-linear time

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

Expert interaction scheme



An Example of an Expert Learning Algorithm

Exponential Weighted Average (EWA)

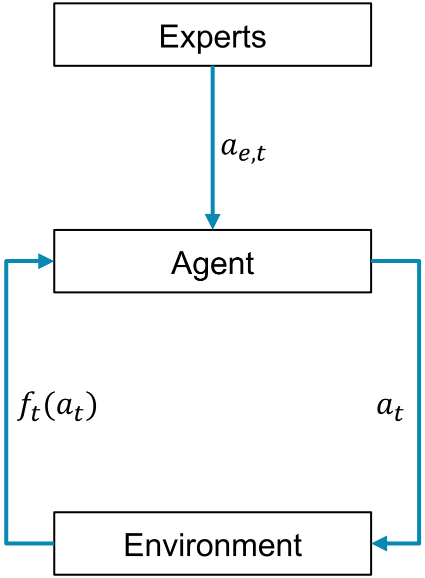
Pseudocode of EWA

- Initialize $\boldsymbol{w}_1 = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$ uniformly over the experts (strategies) and pick the learning rate 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

- Regret $O(\sqrt{T \log(m)})$

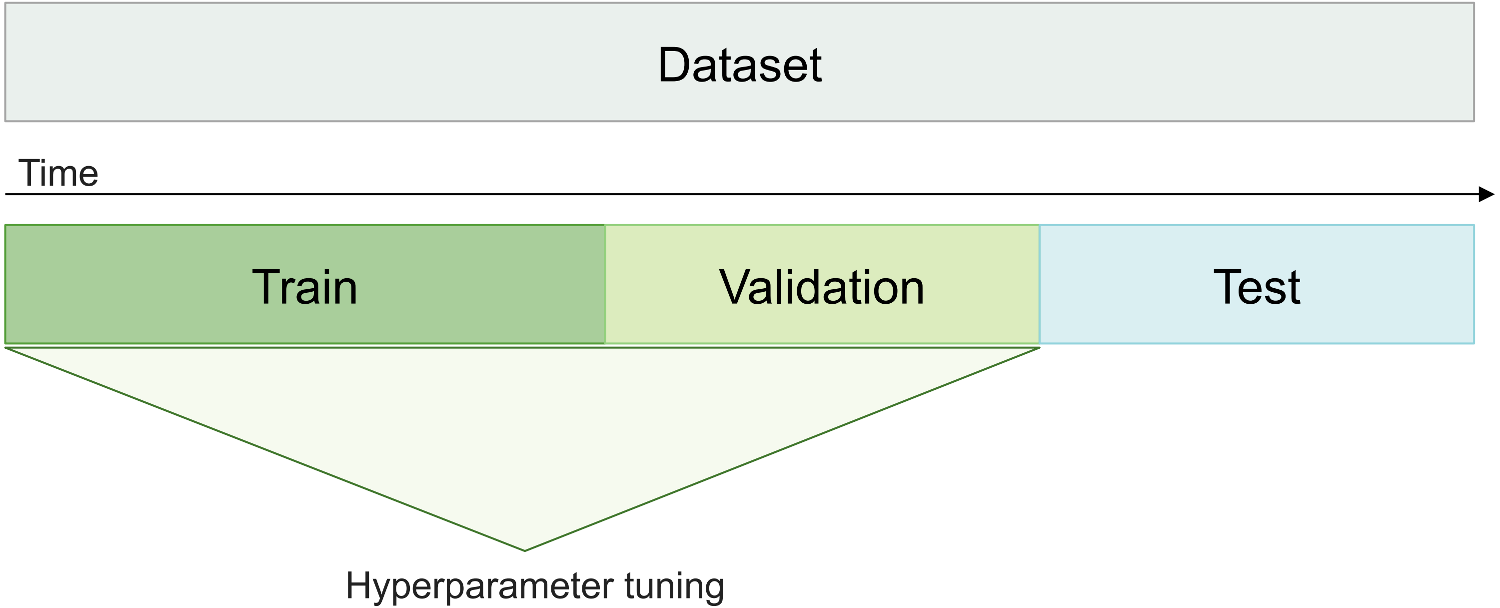
Expert interaction scheme



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

Managing overfitting and hyperparameter tuning

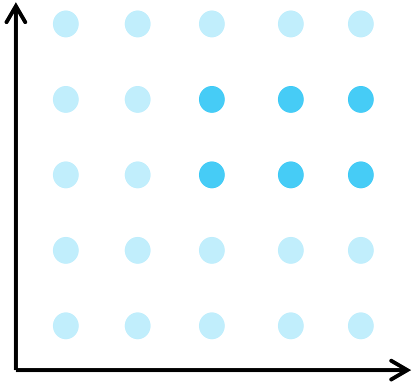
For all these algorithms, it is fundamental to tune the parameters



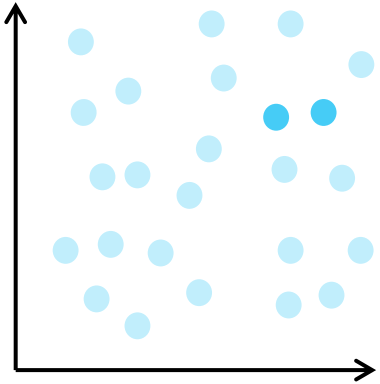
Hyperparameter Tuning

Training these models make take a large amount of time and compute power

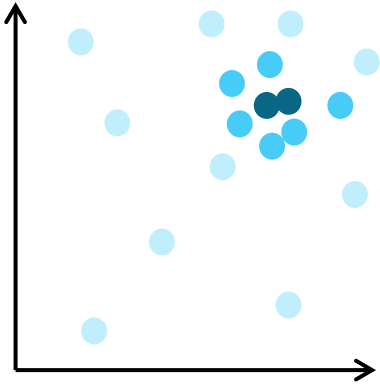
Grid search



Random search



Bayesian search



- Akiba, Takuya, et al. "Optuna: A next-generation hyperparameter optimization framework." *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019
- Bergstra, James, Dan Yamins, and David D. Cox. "Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms." *SciPy* 13 (2013): 20.
- Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." *Learning and Intelligent Optimization: 5th International Conference*, 2011.

AGENDA

– ML toolkit

– **Quantitative Trading**

– Use case

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

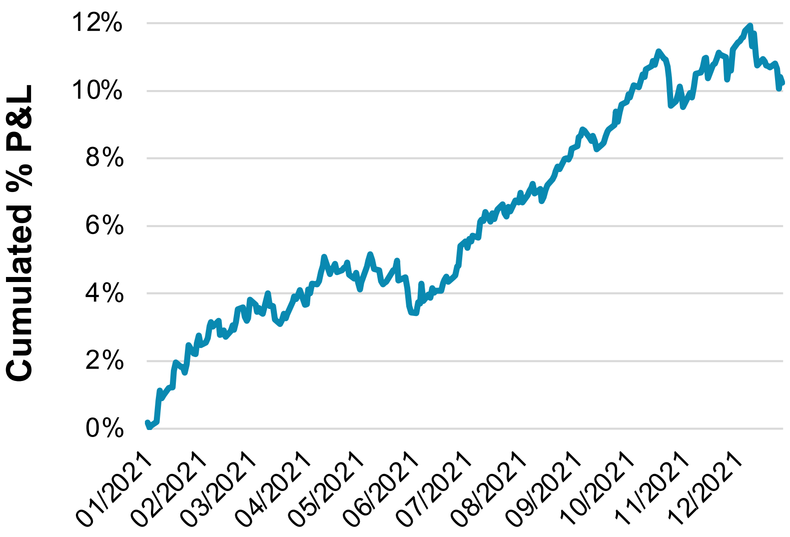
Strategy description

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

Legend

T = time horizon in minutes
R = returns
μ = average asset return
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

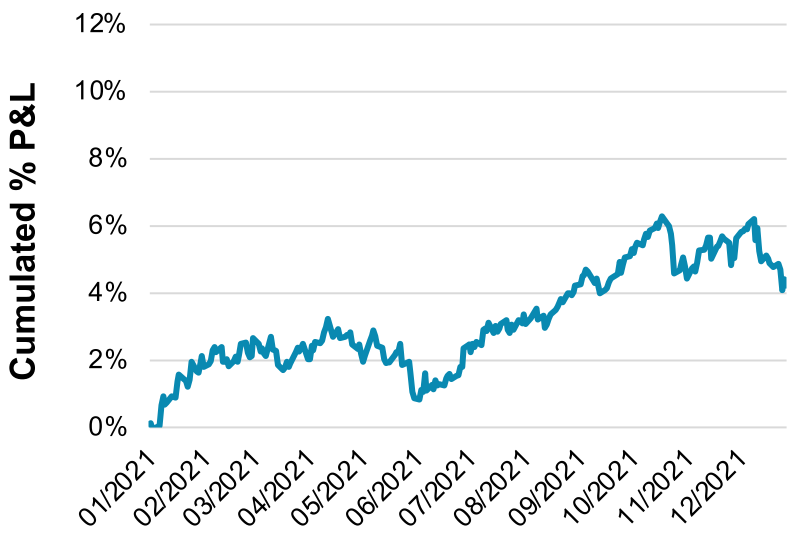
Strategy description

- **Trading rule**= $-\sum_{i=0}^{T-2}(T-i-1)R_{t-i}$
- T = 120 minutes
- Transaction costs = $\frac{1}{2}\text{spread}$

Can we improve?

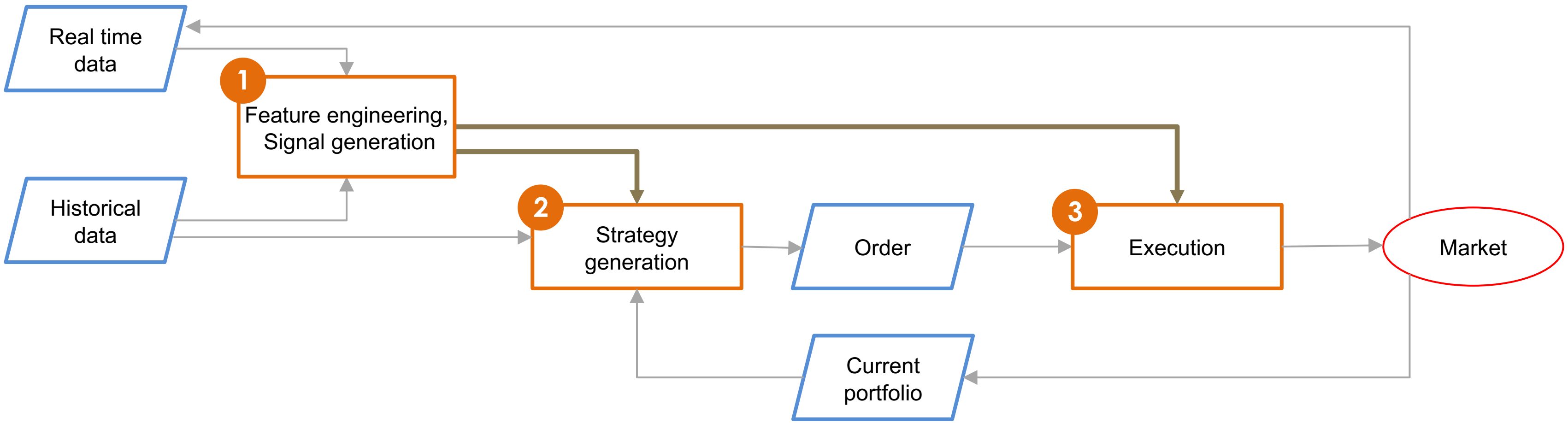
- Consider costs when generating the strategy?
- Find a strategy which does not work only on mean-reverting assets?
- Move on from a strictly defined trading rule?

P&L of backtest trading on 2021



Intraday Trading with Machine Learning

End-to-end workflow to build an intraday trading strategy

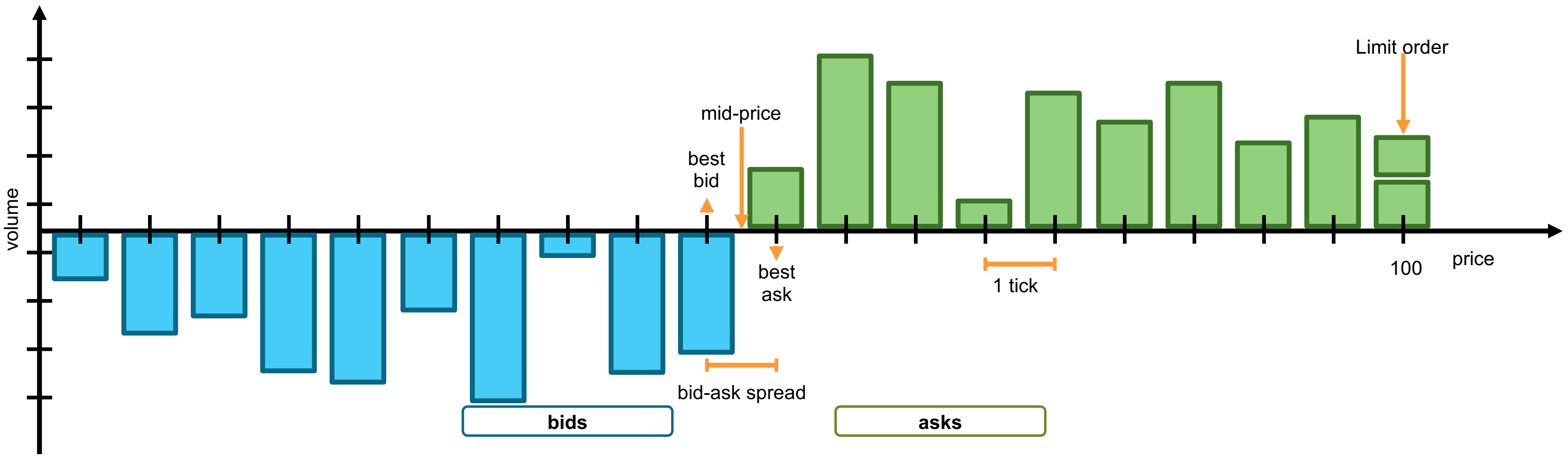


Automatic trading algorithms are comprised of 3 main components:

1. The **feature engineering** and **signal generation** component extracts **features** and **alpha signals** from the LOB
2. The **second component** takes the alpha signals and compressed LOB information to generate a **trading strategy**
3. The **execution algorithm** has the objective of minimizing transaction costs and controlling market impact

1 Limit Order Book (LOB) Data

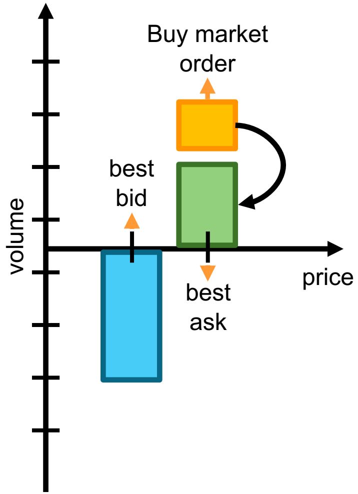
LOB data contains 10 price levels on the bid and on the ask



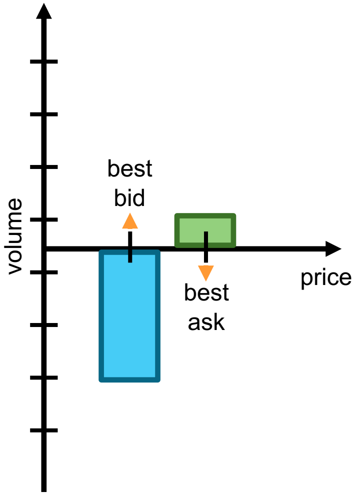
1 Trades and Quotes (TAQ) Data

TAQ data contains only best bid, best ask and executed orders

TAQ dataset time T_0



TAQ dataset time T_1



1 Processing the Raw Data

LOB and TAQ data have timestamps at nanosecond (10^{-9}) precision

Data Quality

- Handle nans/gaps (forward fill?)
- Merge the LOB and TAQ datasets
- Be careful with datetime features
- Handle roll dates

Working frequency

- At this step it is necessary to decide the frequency you want to work with:
  - Tick by tick
  - Downsample every x ticks
  - Downsample every x seconds
  - Downsample every x minutes

1 Extracting Features from the LOB

We can use a classical approach, a machine learning approach or a combined one

Classical approach

Use hand crafted features such as:

- Autocovariance of the price
- Order Flow Imbalance*

$$OF_{b,t} \begin{cases} v_{b,t} \text{ if } p_{b,t} > p_{b,t-1} \\ v_{b,t} - v_{b,t-1} \text{ if } p_{b,t} = p_{b,t-1} \\ -v_{b,t} \text{ if } p_{b,t} < p_{b,t-1} \end{cases} \qquad OF_{a,t} \begin{cases} -v_{a,t-1} \text{ if } p_{a,t} > p_{a,t-1} \\ v_{a,t} - v_{a,t-1} \text{ if } p_{a,t} = p_{a,t-1} \\ -v_{a,t} \text{ if } p_{a,t} < p_{a,t-1} \end{cases}$$

$$OFI_t = OF_{b,t} - OF_{a,t}$$

- Volume imbalance

$$\frac{v_a - v_\mathrm{b}}{v_a + v_\mathrm{b}}$$

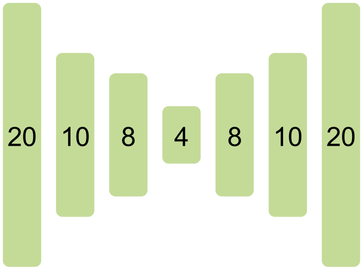
- Trade imbalance

$$\sum_{N(t_{k-1})}^{N(t_k)} b_n - \sum_{N(t_{k-1})}^{N(t_k)} s_n$$

Machine Learning approach

- Use convolutional neural networks to extract features†
- Compress the information with autoencoders

Autoencoder example



Edoardo Vittori

*Cont, Rama, Arseniy Kukanov, and Sasha Stoikov. "The price impact of order book events." *Journal of financial econometrics* 12.1 (2014): 47-88.
† Zhang, Zihao, Stefan Zohren, and Stephen Roberts. "Deeplob: Deep convolutional neural networks for limit order books." *IEEE Transactions on Signal Processing* 67.11

1 Generating Trading Signals

The objective is to accurately predict the direction of the price movement

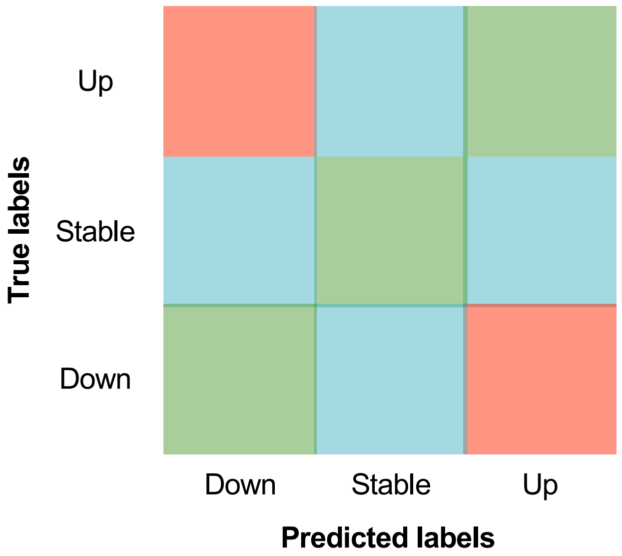
Defining the target

- target $\begin{cases} (\text{mid}_{t+x\text{sec}} - \text{mid}_t) < -\theta \rightarrow \text{Down} \\ \theta < (\text{mid}_{t+x\text{sec}} - \text{mid}_t) < \theta \rightarrow \text{Stable} \\ (\text{mid}_{t+x\text{sec}} - \text{mid}_t) > \theta \rightarrow \text{Up} \end{cases}$
- $\text{mid}_{t+x\text{ticks}} - \text{mid}_t$
- Should I consider a (rolling) average price to smooth out the noise?
- Should I look at bid and ask prices instead of mid

Classifier choice

- DeepLOB, CNN, LSTM, MLP
- Xgboost, Light GBM, Extra Trees
- Ensemble of predictions
- Hyperparameter tuning

Example of confusion matrix



1 References LOB Forecasting

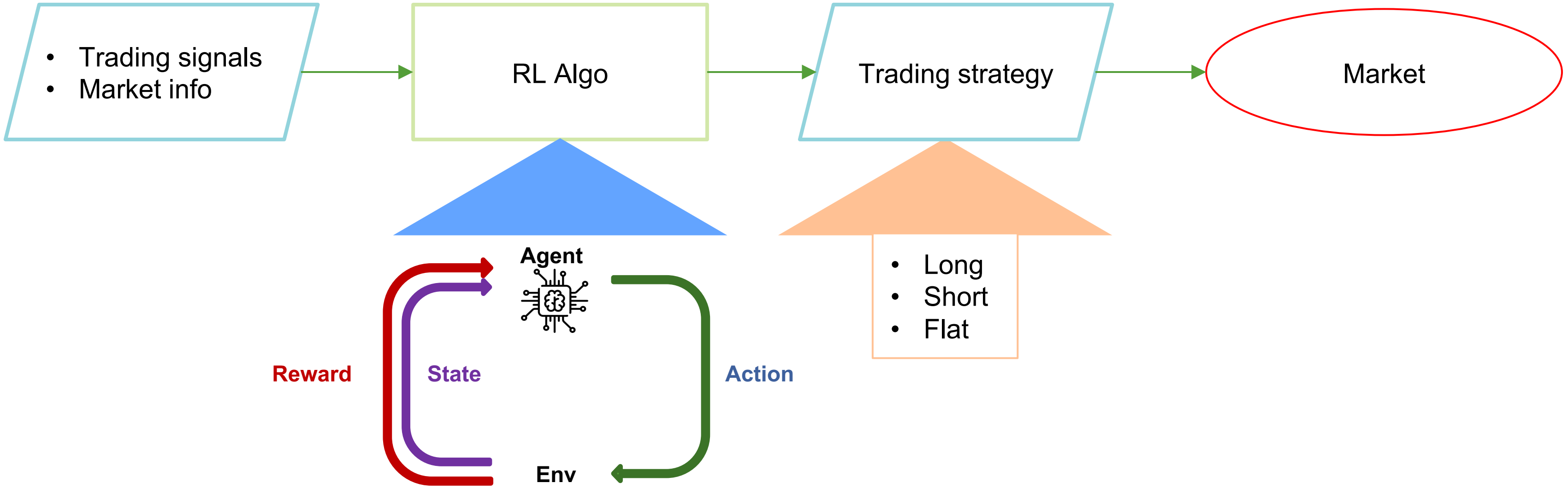
Several research groups are working on short term forecasting from the LOB

- Briola, Antonio, Silvia Bartolucci, and Tomaso Aste. "Deep Limit Order Book Forecasting." *arXiv preprint arXiv:2403.09267* (2024).
- Lucchese, Lorenzo, Mikko S. Pakkanen, and Almut ED Veraart. "The short-term predictability of returns in order book markets: a deep learning perspective." *International Journal of Forecasting* (2024).
- Kolm, Petter N., Jeremy Turiel, and Nicholas Westray. "Deep order flow imbalance: Extracting alpha at multiple horizons from the limit order book." Mathematical Finance 33.4 (2023): 1044-1081.
- Aït-Sahalia, Yacine, et al. *How and When are High-Frequency Stock Returns Predictable?*. No. w30366. National Bureau of Economic Research, 2022.
- Zhang, Zihao, Stefan Zohren, and Stephen Roberts. "Deeplob: Deep convolutional neural networks for limit order books." *IEEE Transactions on Signal Processing* 67.2020
- Sirignano, Justin, and Rama Cont. "Universal features of price formation in financial markets: perspectives from deep learning." *Machine Learning and AI in Finance*. Routledge, 2021. 5-15.
- Cont, Rama, Arseniy Kukanov, and Sasha Stoikov. "The price impact of order book events." *Journal of financial econometrics* 12.1 (2014): 47-88.

2 Defining the Trading Strategy Using RL

RL has the task of optimizing the trading strategies taking into account transaction costs

State: trading signals, market information and current position
Action: can be the portfolio position or the trade
Reward: p&l – transaction costs



3 Optimizing Strategy Execution

Optimizing execution can reduce transaction costs and market impact

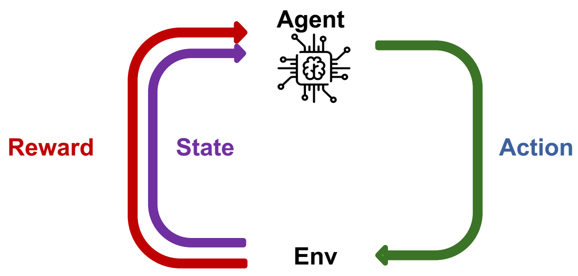
Opportunity cost: execute immediately with high market impact or execute in time with the risk of a market movement?
Trading signal: shorter term compared to the one used in the trading strategy

- Trade
- Execution time
- Trading signals
- Market info

RL Algo

Execution strategy

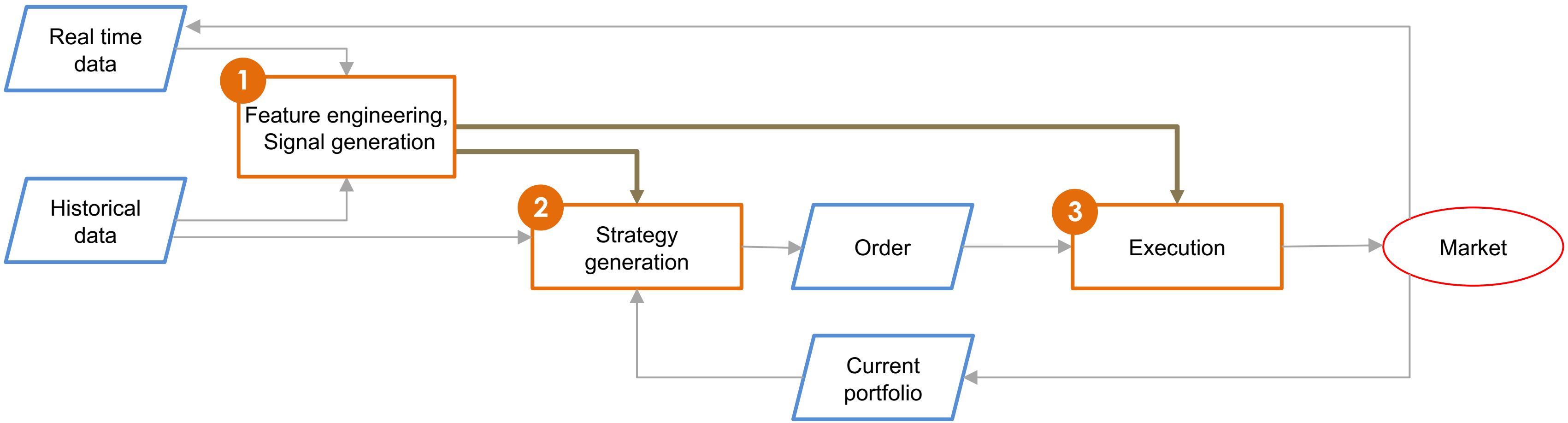
Market



- Execute immediately
- Post a limit order
- Execute at the end

Intraday Trading with Machine Learning

A generic workflow to generate a trading strategy with any listed asset



Automatic trading algorithms are comprised of 3 main components:

1. The **feature engineering** and **signal generation** component extracts **features** and **alpha signals** from the LOB
2. The **second component** takes the alpha signals and compressed LOB information to generate a **trading strategy**
3. The **execution algorithm** has the objective of minimizing transaction costs and controlling market impact

Running a single strategy may not be suffient

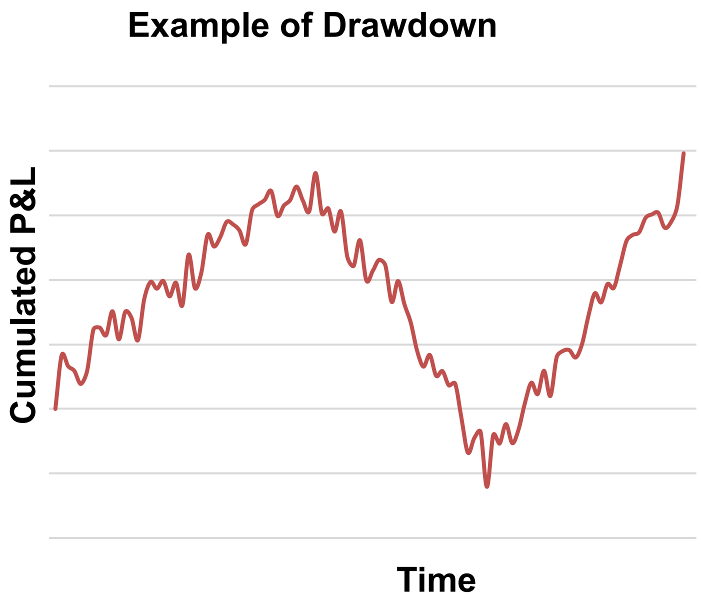
To improve performance, it is fundamental to aggregate uncorrelated strategies

Challenges

- Non-stationarity of the market
- Intraday strategies have constraints on amount of AUM caused by transaction costs

Objectives

- Improve sharpe
- Increase AUM
- Handle different market conditions



Aggregating Strategies Improves Sharpe

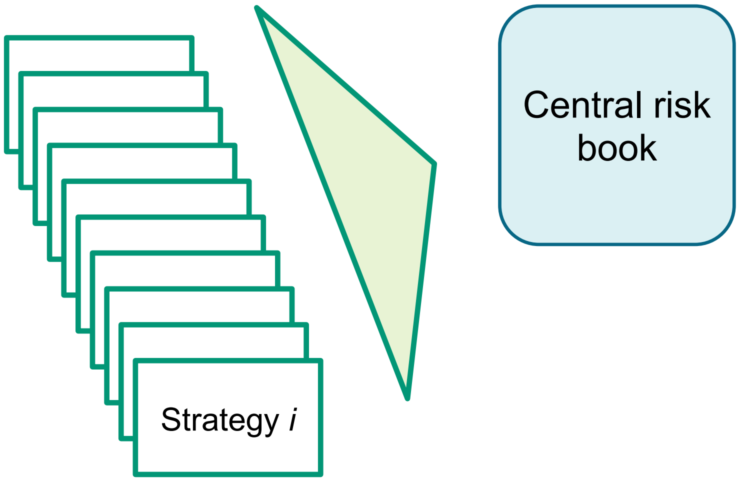
The different strategies need to have a small correlation

Why combining uncorrelated strategies helps

- $\text{Sharpe}(S_1) = \frac{\mu_1}{\sigma_1}$, $\text{Sharpe}(S_2) = \frac{\mu_2}{\sigma_2}$
- $\text{Sharpe}(S_1 + S_2) = \frac{\mu_1+\mu_2}{\sqrt{\sigma_1^2+\sigma_2^2+2\rho\sigma_1\sigma_2}}$

Correlation: ρ	Sharpe(S1+S2)
1	$\frac{\mu_1 + \mu_2}{\sigma_1 + \sigma_2}$
0	$\frac{\mu_1 + \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}$
-1	$\frac{\mu_1 + \mu_2}{\lvert\sigma_1 - \sigma_2\rvert}$

Algorithmic multi-strat



How can we combine the strategies?

Three main steps to aggregate strategies

1. **Choose the rebalancing frequency**
   - Close positions daily ⟶ Daily rebalancing
   - Close positions weekly → Weekly rebalancing
2. **Normalize strategies by volatility, keeping it constant between each rebalancing**
   - $p_i = \frac{s_i}{\sigma_i}$ where p_i is the position, s_i is the signal and σ_i the volatility of strategy i
   - $\mathrm{ptf} = \sum_i w_i p_i$ where w_i is the weight assigned to each strategy
3. **How do I choose $\boldsymbol{w_i}$?**
   - I can sum them equally
   - I can use mean-variance optimization
   - How about expert learning?

The Strategy (Asset) Allocation Problem

We are behaving like a portfolio manager allocating over strategies instead of assets

Objective

Allocate funds among a set of strategies so to:

- Have a balanced exposure
- Minimize risk and maximize return
- Diversify

Solution

Leverage models that:

- avoid the forecasting step
- are adaptive to markets
- don’t need a large training set

Classical Models

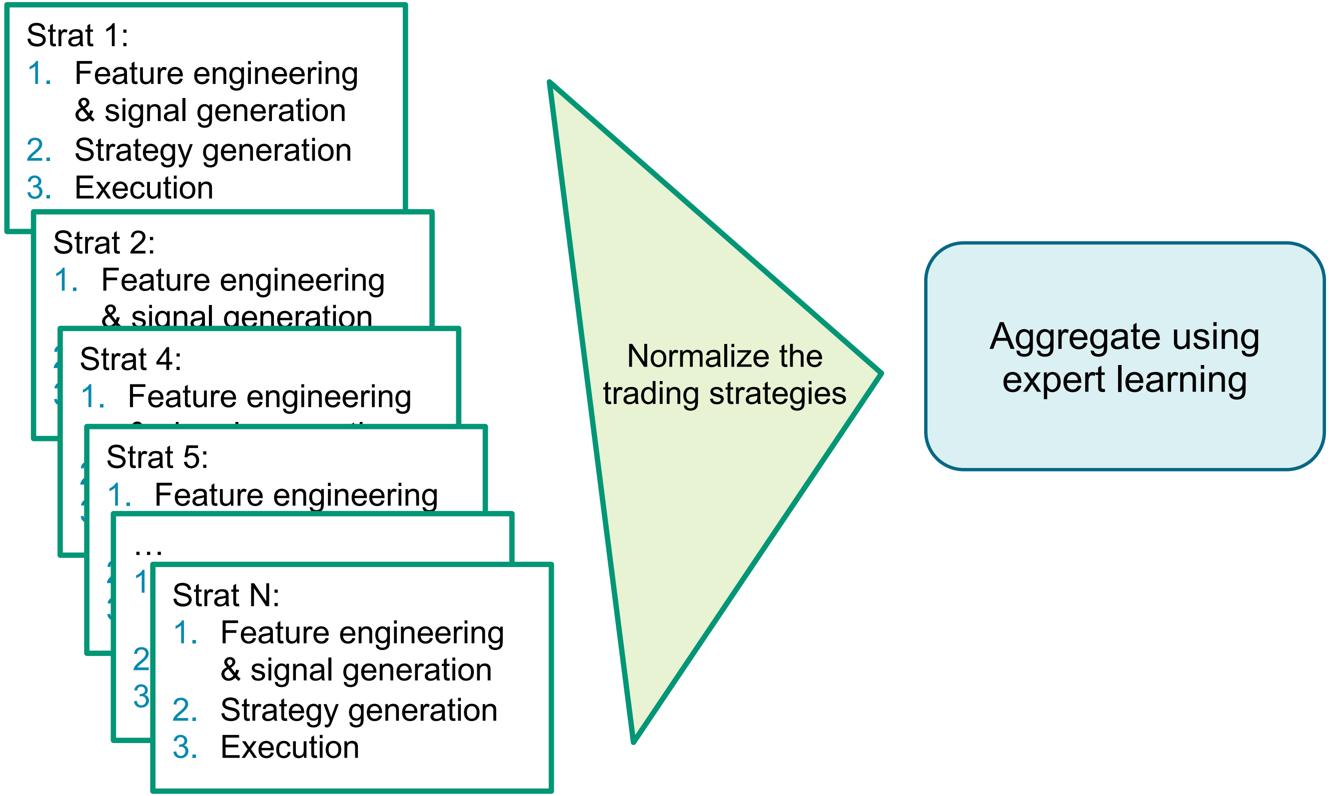
- Equal Risk Contribution
- Mean-Variance: Markowitz
- …

Expert learning

- Exponential Gradient
- Online Newton Step
- …

Recap

Create N uncorrelated strategies and comine them using expert learning



AGENDA

– ML toolkit

– Quantitative Trading

– **Use case**

Reinforcement Learning for Quantitative Trading

Problem description and MDP definition

Quantitative Trading

Definition

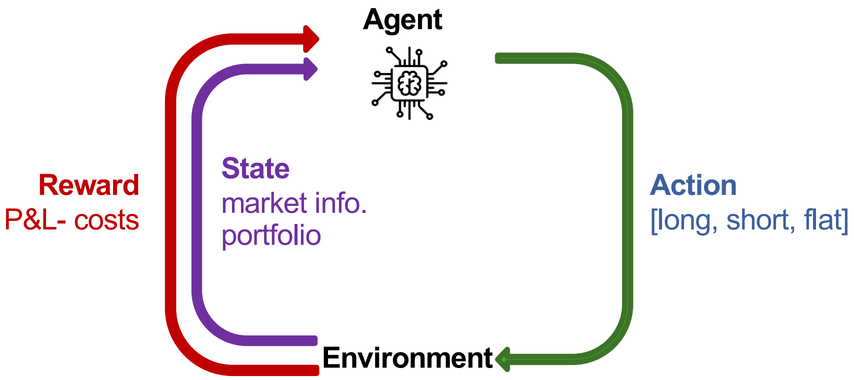
- At each timestep, decide whether to go long, short or flat to maximize gains

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



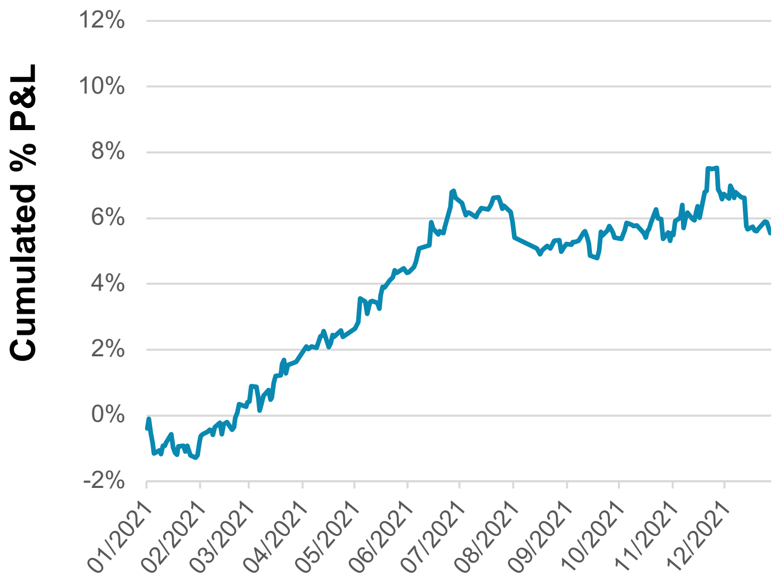
Reinforcement Learning for FX Trading (1/2)

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

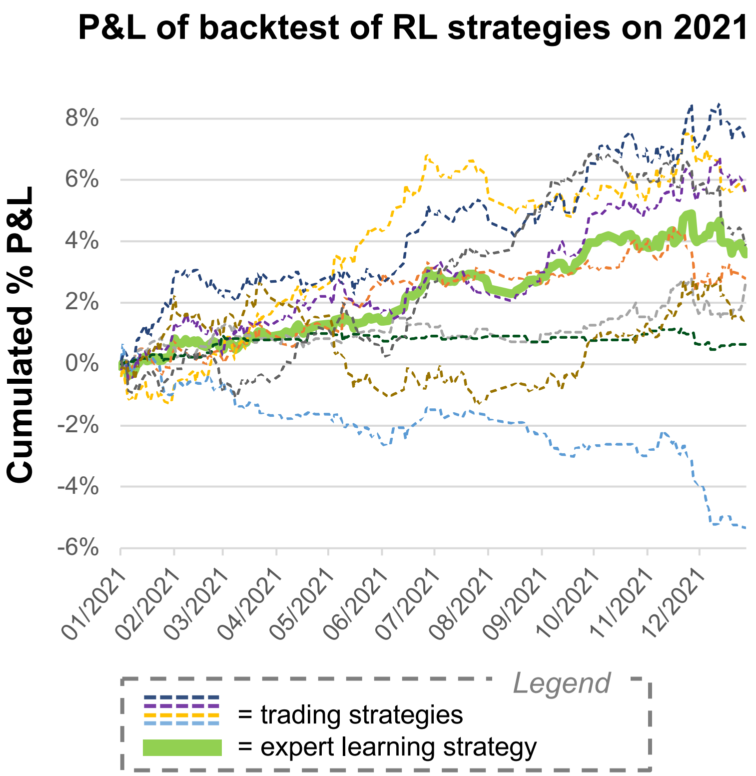
Can we improve?

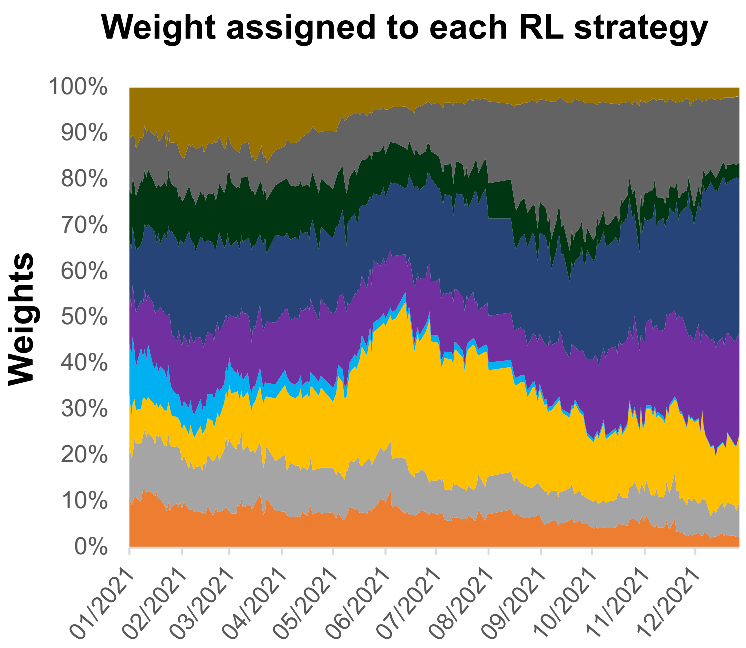
- Market non-stationarity

Riva, Antonio, et al. "Learning FX trading strategies with FQI and persistent actions." *Proceedings of the Second ACM International Conference on AI in Finance*. 2021.

Reinforcement and Expert Learning for FX Trading

Experimental results - performance





Riva, Antonio, et al. "Addressing non-stationarity in FX trading with online model selection of offline RL experts." *Proceedings of the Third ACM International Conference on AI in Finance*. 2022.

Q&A

Edoardo Vittori
edoardo.vittori@polimi.it

The opinions expressed in this document are solely those of the authors and do not represent in any way those of their present and past employers