Augmenting Traders with Learning Machines
Ph.D. Thesis Defense

Edoardo Vittori
1. Introduction

2. Online Portfolio Optimization with Transaction Costs

3. Quantitative Trading with MCTS

4. Option Hedging with Risk Averse RL

5. Conclusions
1. Introduction
Financial markets
Other market participants
Quantitative Trading Portfolio Optimization
Optimal Execution
Banks
Asset managers
Hedge funds
Liquidity providers
Hedging
Market Making
Financial markets
Trading as a Markov Decision Process (MDP)

Environment

State
market features
internal features

Reward
P&L
penalty

Agent

Action
portfolio position
price
trade

Environment
## Families of Algorithms Considered

<table>
<thead>
<tr>
<th></th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online vs offline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning approach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observation to action delay</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy type</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Families of Algorithms Considered

<table>
<thead>
<tr>
<th></th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online vs offline</td>
<td>depends on algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning approach</td>
<td>general policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observation to action delay</td>
<td>small</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy type</td>
<td>stationary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Families of Algorithms Considered

<table>
<thead>
<tr>
<th></th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online vs offline</strong></td>
<td>depends on algorithm</td>
<td>online</td>
<td></td>
</tr>
<tr>
<td><strong>Learning approach</strong></td>
<td>general policy</td>
<td>local policy</td>
<td></td>
</tr>
<tr>
<td><strong>Observation to action delay</strong></td>
<td>small</td>
<td>some</td>
<td></td>
</tr>
<tr>
<td><strong>Policy type</strong></td>
<td>stationary</td>
<td>non-stationary</td>
<td></td>
</tr>
</tbody>
</table>
## Families of Algorithms Considered

<table>
<thead>
<tr>
<th></th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online vs offline</strong></td>
<td>depends on algorithm</td>
<td>online</td>
<td>online</td>
</tr>
<tr>
<td><strong>Learning approach</strong></td>
<td>general policy</td>
<td>local policy</td>
<td>optimizing for next action</td>
</tr>
<tr>
<td><strong>Observation to action delay</strong></td>
<td>small</td>
<td>some</td>
<td>small</td>
</tr>
<tr>
<td><strong>Policy type</strong></td>
<td>stationary</td>
<td>non-stationary</td>
<td>adversarial</td>
</tr>
<tr>
<td>Financial tasks</td>
<td>Reinforcement Learning</td>
<td>Monte Carlo Tree Search</td>
<td>Expert Learning</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Portfolio optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantitative trading</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market making</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option hedging</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal execution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financial tasks</td>
<td>Reinforcement Learning</td>
<td>Monte Carlo Tree Search</td>
<td>Expert Learning</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Portfolio optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantitative trading</td>
<td>FQI for FX trading</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market making</td>
<td>FQI with MFGs for bond dealing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option hedging</td>
<td>Risk averse RL (TRVO) for hedging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal execution</td>
<td>FQI and Thompson Sampling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Application of Algorithms to Financial Tasks in this Research

<table>
<thead>
<tr>
<th>Financial tasks</th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portfolio optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantitative trading</td>
<td>FQI for FX trading</td>
<td>Open Loop UCT for FX trading</td>
<td></td>
</tr>
<tr>
<td>Market making</td>
<td>FQI with MFGs for bond dealing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option hedging</td>
<td>Risk averse RL (TRVO) for hedging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal execution</td>
<td>FQI and Thompson Sampling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Application of Algorithms to Financial Tasks in this Research

<table>
<thead>
<tr>
<th>Financial tasks</th>
<th>Reinforcement Learning</th>
<th>Monte Carlo Tree Search</th>
<th>Expert Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portfolio optimization</td>
<td></td>
<td></td>
<td>Online portfolio optimization</td>
</tr>
<tr>
<td>Quantitative trading</td>
<td>FQI for FX trading</td>
<td>Open Loop UCT for FX trading</td>
<td></td>
</tr>
<tr>
<td>Market making</td>
<td>FQI with MFGs for bond dealing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option hedging</td>
<td>Risk averse RL (TRVO) for hedging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal execution</td>
<td>FQI and Thompson Sampling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financial tasks</td>
<td>Reinforcement Learning</td>
<td>Monte Carlo Tree Search</td>
<td>Expert Learning</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Portfolio optimization</td>
<td></td>
<td></td>
<td>Online portfolio optimization</td>
</tr>
<tr>
<td>Quantitative trading</td>
<td>FQI for FX trading</td>
<td>Open Loop UCT for FX trading</td>
<td></td>
</tr>
<tr>
<td>Market making</td>
<td>FQI with MFGs for bond dealing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option hedging</td>
<td>Risk averse RL (TRVO) for hedging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal execution</td>
<td>FQI and Thompson Sampling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Online Portfolio Optimization with Transaction Costs
Defining Expert Learning

Expert Learning

1. Agent makes a decision: $\theta_t \in \Theta$, based on suggestions of experts $\mathcal{E}$
2. Environment chooses outcome $y_t$ and loss $f_t(\theta_t, y_t)$
3. Update cumulative loss $L_T = \sum_{t=1}^{T} f_t(\theta_t, y_t)$

Objective

- Regret: $R_T = L_T - \inf_{e \in \mathcal{E}} \sum_{t=1}^{T} f_t(\theta_{e,t}, y_t)$
- No regret: $\frac{R_T}{T} \to 0$
Online Portfolio Optimization Setting

- $a_t \in \Delta_{M-1}$ is the portfolio allocation
- The experts are Constant Rebalancing Portfolios
- $a^* = \arg \inf_{a \in \Delta_{M-1}} \sum_{t=1}^{T} f_t(a, y_t)$ is the Best CRP
- $f_t(a, y_t) = -\log(\langle a, y_t \rangle)$ is the loss
- $y_t = \left( \frac{p_{t,1}}{p_{t-1,1}}, \ldots, \frac{p_{t,M}}{p_{t-1,M}} \right)$ are the price relatives

**Limitations:** no transaction costs
Approaches to Portfolio Optimization

Background

• **Modern Portfolio Optimization**
  [Markowitz, 1952]
  - Calculate variance and correlations
  - Single period

• **Intertemporal CAPM**
  [Merton, 1969]
  - Make assumptions on asset dynamics
  - Multi period

• **Online Portfolio Optimization**
  [Cover and Ordentlich, 1996]
  - Adversarial market
  - Multi period
Approaches to Portfolio Optimization

Background

- **Modern Portfolio Optimization**
  [Markowitz, 1952]
  - Calculate variance and correlations
  - Single period

- **Intertemporal CAPM**
  [Merton, 1969]
  - Make assumptions on asset dynamics
  - Multi period

- **Online Portfolio Optimization**
  [Cover and Ordentlich, 1996]
  - Adversarial market
  - Multi period

Main contributions

- **Dealing with Transaction Costs in Portfolio Optimization: Online Gradient Descent with Momentum**
  [Vittori et al., 2020a]
  - Keeping transaction costs under control in OPO
  - Definition of algorithm: OGDM with total regret guarantees
Total Regret: Adding Transaction Costs

Total Regret

\[ R^C_T = \sum_{t=1}^{T} f_t(a_t, y_t) - \inf_{a \in \Delta_{M-1}} \sum_{t=1}^{T} f_t(a, y_t) + \gamma \sum_{t=1}^{T} ||a_t - a_{t-1}||_1 \]

- \( R_T \): standard regret
- \( C_T \): transaction costs

\( \gamma \) is the proportional transaction rate for buying and selling stocks.
Online Gradient Descent with Momentum

Algorithm 1 OGDM in OPO with Transaction Costs

Require: learning rate sequence \(\{\eta_1, \ldots, \eta_T\}\), momentum parameter sequence \(\{\lambda_1, \ldots, \lambda_T\}\)

1: Set \(a_1 \leftarrow \frac{1}{M} 1\)
2: for \(t \in \{1, \ldots, T\}\) do
3: \[\text{Select } a_{t+1} \leftarrow \Pi_{\Delta_{M-1}} \left( a_t + \eta_t \frac{y_t}{\langle y_t, a_t \rangle} - \frac{\lambda_t}{2} (a_t - a_{t-1}) \right)\]
4: Observe \(y_{t+1}\) from the market
5: Get wealth \(\log(\langle y_{t+1}, a_{t+1} \rangle) - \gamma ||a_{t+1} - a_t||_1\)
6: end for

Total Regret

\[R_T^C \leq O(\sqrt{T})\]

Comparison with State of the Art in OPO

Online Portfolio Optimization

- Universal Portfolios (UCP) [Kalai and Vempala, 2002]
- Online Newton Step (ONS) [Agarwal et al., 2006]
- Online Lazy Updates (OLU) [Das et al., 2013]

<table>
<thead>
<tr>
<th>Metric</th>
<th>OGDM</th>
<th>UCP</th>
<th>OLU</th>
<th>ONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_T$</td>
<td>$\mathcal{O}(\sqrt{T})$</td>
<td>$\mathcal{O}(\log T)$</td>
<td>$\mathcal{O}(\sqrt{T})$</td>
<td>$\mathcal{O}(\log T)$</td>
</tr>
<tr>
<td>$R_T^C$</td>
<td>$\mathcal{O}(\sqrt{T})$</td>
<td>$\mathcal{O}(\log T)$</td>
<td>$\mathcal{O}(T)$</td>
<td>-</td>
</tr>
<tr>
<td>Complexity</td>
<td>$\Theta(M)$</td>
<td>$\Theta(T^M)$</td>
<td>$\Theta(M)$</td>
<td>$\Theta(M^2)$</td>
</tr>
</tbody>
</table>
Average Annual Percentage Yield $A(W_T)$ computed on the wealth $W_t^C(a_{1:T}, y_{1:T})$: $A(W_T) = W_T^{250/T} - 1$

3. Quantitative Trading with MCTS
Quantitative Trading

**Trading**: a sequential decision process in which at each round \( t \in \{1, \ldots, T\} \) over a trading horizon \( T \in \mathbb{N} \), a trader decides whether to go long, short or stay flat with respect to an asset to maximize her wealth

**MDP Configuration**

- \( a_t \in \{-1, 0, 1\} \)
- \( s_t = ([P_{t-w}, \ldots, P_t], a_{t-1}, t) \)
- \( r_{t+1} = a_t(P_{t+1} - P_t) - \frac{\text{bid-ask}}{2} \left| a_t - a_{t-1} \right| \)

\( \left\{ \begin{array}{l}
\text{market movement} \\
\text{transaction costs}
\end{array} \right. \)
Approaches to Trading

Background

• Practitioner approach
  - Technical analysis
  - Macro-economic analysis

• Supervised learning approach
  [Baba and Kozaki, 1992]
  - Forecast asset prices
  - Derive trade
  - Hard to incorporate market frictions

• Reinforcement Learning approach
  [Moody and Saffell, 2001]
  - Integrate prediction and action
  - Simple to include market frictions
Approaches to Trading

Background

- Practitioner approach
  - Technical analysis
  - Macro-economic analysis

- Supervised learning approach
  [Baba and Kozaki, 1992]
  - Forecast asset prices
  - Derive trade
  - Hard to incorporate market frictions

- Reinforcement Learning approach
  [Moody and Saffell, 2001]
  - Integrate prediction and action
  - Simple to include market frictions

Main contributions

Monte Carlo Tree Search for Trading and Hedging
[Vittori et al., 2021]
- Use of Open Loop MCTS for single currency FX trading
Monte Carlo Tree Search (MCTS)

Planning through generative model
• Selection using UCB₁ \( a_n = \arg\max_{i=1..K} \bar{X}_{i,T_i(n-1)} + C\sqrt{\frac{2\log n}{T_i(n-1)}} \)
• Convergence to the optimal solution in deterministic environments
Planning Tree in Deterministic and Stochastic Environments

UCT in deterministic environments

- $s_0$
- $a_1$
- $a_2$
- $s_1^1$
- $s_1^2$
- $a_1$
- $a_2$
- $s_2^1$
- $s_2^2$
- $a_1$
- $a_2$
- $s_3^1$
- $s_3^2$

UCT in continuous stochastic environments

- $s_0$
- $a_1$
- $a_2$
- $s_1^1$
- $s_1^2$
Open Loop UCT [Lecarpentier et al., 2018]

- Nodes are distributions over states
- Open-loop value of action sequence $\tau$:
  \[
  V_{OL}(s, \tau) = \mathbb{E} \left[ \sum_{t=1}^{m} \gamma^t r_t \middle| s_0 = s, a_t \in \tau \right]
  \]
- Open-loop value of a node $N_{d,i}$:
  \[
  V(N_{d,i}) = \mathbb{E}_{s \sim \mathcal{P}(\cdot | s_0, \tau_{d,i})} \left[ V_{OL}^* (s) \right]
  \]
  where $V_{OL}^* (s) = \max_{\tau \in \mathcal{A}^m} V_{OL}(s, \tau)$
Q-learning Backpropagation

- **Standard** Backpropagation

\[ Q_t(\mathcal{N}_{d,i}, a) = (1 - \frac{1}{N}) Q_t(\mathcal{N}_{d,i}, a) + \frac{1}{N} (r_t + \gamma V_t(\mathcal{N}_{d+1,j})) \]

- **Temporal Difference** Backpropagation, based on the Q-Learning update rule [Vodopivec et al., 2017]

\[ Q_t(\mathcal{N}_{d,i}, a) = (1 - \beta) Q_t(\mathcal{N}_{d,i}, a) + \beta \left( r_t + \gamma \max_{a'} Q_t(\mathcal{N}_{d+1,j}, a') \right) \]
Clustering generative model

1. Start from the current price window
   \[ w_t = (P_{t-M}, \ldots, P_{t-1}) \]

2. Extract window of returns
   \[ \delta_t = \frac{p_t - p_{t-1}}{p_{t-1}}, \]
   \[ \delta_t = (\delta_{t-M}, \ldots, \delta_{t-1}) \]

3. Find the K nearest neighbors of \( \delta_t \) in the historical dataset \( D \)

4. Use the neighbors to project future asset prices
Annualized average P&L with no transaction costs, as a function of the search budget and the numbers of neighbors. Average over 50 runs, 95% confidence intervals

Annualized average P&L with transaction costs ($10^{-5}$) as a function of the search budget, $K = 100$. Average over 50 runs, 95% confidence intervals.

4. Option Hedging with Risk-Averse RL
Vanilla options: contracts that offer the buyer the right to buy or sell a certain amount of the underlying asset at a predefined price at a certain future time

Option hedging: a sequential decision process in which at each round $t \in \{1, \ldots, T\}$ over the life of the option $T \in \mathbb{N}$, a trader decides how much to hold of the underlying instrument to minimize the price swings caused by the option

Option Hedging as an MDP

- $a_t \in [0, 1]$: current hedge portfolio
- $s_t = [S_t, C_t, \frac{\partial C(S_t)}{\partial S}, a_{t-1}]$
- $r_{t+1} = C_{t+1} - C_t - a_t \cdot (S_{t+1} - S_t) - c(a_t - a_{t-1})$

- $\{\text{option variation}\}$
- $\{\text{market movement}\}$
- $\{\text{transact. costs}\}$
Background

• **Classical approach**  
  [Black and Scholes, 1973]
  - Model the market as GBM
  - Assume continuous time hedging
  - Assume no market frictions
  - Solve resulting PDE

• **Reinforcement Learning approach**  
  [Kolm and Ritter, 2019]
  - Collect/simulate data
  - Learn to hedge
## Approaches to Option Hedging

### Background

- **Classical approach**
  [Black and Scholes, 1973]
  - Model the market as GBM
  - Assume continuous time hedging
  - Assume no market frictions
  - Solve resulting PDE

- **Reinforcement Learning approach**
  [Kolm and Ritter, 2019]
  - Collect/simulate data
  - Learn to hedge

### Main contributions

**Option Hedging with Risk Averse RL**
[Vittori et al., 2020b]
- Use of the risk-averse policy search RL algorithm: TRVO
Trust Region Volatility Optimization (TRVO)

- Reward volatility

\[ \nu^2 = (1 - \gamma) \mathbb{E}_{s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)} \left[ \sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t) - J_{\pi})^2 \right] \]

- Mean-volatility objective \( \eta_{\pi} = J_{\pi} - \lambda \nu_{\pi}^2 \)

Financial Environment

Vanilla call option

- time to maturity = 60 days
- unitary notional
- implied volatility = 20%
- interest rates = 0
- $K = S_0 = 100$
- starting price (ATM) option $\sim 3.24$
- starting delta = 0.5

training on 10k episodes and testing on 2k episodes

Simulated underlying

- GBM
- no drift
- volatility = 20%
- $S_0 = 100$
- 5 time steps per day
- bid ask spread = 0.1

Results without transaction costs

- delta hedge with no transaction costs $\rightarrow$ average P&L $\sim 0$, volatility $\sim 0.16$
- delta hedge with transaction costs $\rightarrow$ average P&L $\sim -0.3$, volatility $\sim 0.18$

Experimental Results with Transaction Costs

Costs vs Risk changing $\lambda$

Experimental Results with Transaction Costs

Pareto Frontier

5. Conclusions
Conclusions

Today’s Topics

• Online portfolio optimization
  - Controlling transaction costs in OPO

• Quantitative trading
  - FX trading using Open Loop UCT

• Option hedging
  - Equity option hedging using TRVO
## Conclusions

### Today’s Topics

- **Online portfolio optimization**
  - Controlling transaction costs in OPO
- **Quantitative trading**
  - FX trading using Open Loop UCT
- **Option hedging**
  - Equity option hedging using TRVO

### Final Remarks

- Major financial tasks in the Capital Markets modelled as MDPs
- Broad applicability of RL based techniques to financial problems
- Data driven approaches without explicit modelling assumptions


The pricing of options and corporate liabilities.

Multi-period trading via convex optimization.
*arXiv preprint*.

Deep reinforcement learning for active high frequency trading.
*arXiv preprint*.

Deep hedging.
*Quantitative Finance*, pages 1–21.

Abides: Towards high-fidelity market simulation for ai research.
*arXiv preprint*.

Hedging using reinforcement learning: Contextual \(k\)-armed bandit versus \(q\)-learning.
*arXiv preprint*. 
Deep hedging of derivatives using reinforcement learning.
Available at SSRN.

Universal portfolios with side information.

Online lazy updates for portfolio selection with transaction costs.
In AAAI.

Adaptive subgradient methods for online learning and stochastic optimization.
Journal of machine learning research, 12(7).

Tree-based batch mode reinforcement learning.
JMLR, 6(Apr):503–556.

Reinforcement learning for market making in a multi-agent dealer market.
arXiv preprint.
Dynamic trading with predictable returns and transaction costs.

Deep reinforcement learning for market making in corporate bonds: beating the curse of dimensionality.

Learning mean-field games.
arXiv preprint.

The qlbs q-learner goes nuqlear: fitted q iteration, inverse rl, and option portfolios.
Quantitative Finance, pages 1–11.

Introduction to online convex optimization.
arXiv preprint.

Logarithmic regret algorithms for online convex optimization.
A reinforcement learning extension to the almgren-chriss framework for optimal trade execution.
In CIFEr, pages 457–464. IEEE.

Online learning: A comprehensive survey.
Neurocomputing, 459:249–289.

Efficient algorithms for universal portfolios.

Multi-agent reinforcement learning in a realistic limit order book market simulation.
arXiv preprint.

Bandit based monte-carlo planning.
In ECML.

Dynamic replication and hedging: A reinforcement learning approach.
Modern perspectives on reinforcement learning in finance.
Available at SSRN.

Open loop execution of tree-search algorithms.
In IJCAI.

Online portfolio selection: A survey.
ACM COMPUT SURV, 46(3):35.

Transaction cost optimization for online portfolio selection.
QUANT FINANC, 18(8):1411–1424.

Pamr: Passive aggressive mean reversion strategy for portfolio selection.
MACH LEARN, 87(2):221–258.

An end-to-end optimal trade execution framework based on proximal policy optimization.
In IJCAI, pages 4548–4554.
Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance.
arXiv preprint.

portfolio selection.

Reinforcement learning in financial markets.
Data, 4(3):110.

Lifetime portfolio selection under uncertainty: The continuous-time case.

Playing atari with deep reinforcement learning.

Learning to trade via direct reinforcement.
Reinforcement learning for optimized trade execution.
In ICML, pages 673–680.

Double deep q-learning for optimal execution.
arXiv preprint.

Machine learning for trading.
Available at SSRN.

Learning fx trading strategies with fqi and persistant actions.
In ICAIF.

Trust region policy optimization.
In ICML, volume 37, pages 1889–1897.

Proximal policy optimization algorithms.
CoRR, abs/1707.06347.


Dealing with transaction costs in portfolio optimization: Online gradient descent with momentum.
In ICAIF.

Monte carlo tree search for trading and hedging.
In ICAIF.

Option hedging with risk averse reinforcement learning.
In ICAIF.

On monte carlo tree search and reinforcement learning.

Learning from delayed rewards.

Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine learning, 8(3-4):229–256.
A. Contributions and Challenges
### Main Contributions

- **Online portfolio optimization**
  - Controlling transaction costs in OPO

- **Quantitative trading**
  - FX trading using Open Loop UCT
  - Two currency FX trading using FQI

- **Bond market making**
  - Mean Field Games and FQI

- **Option hedging**
  - Equity option hedging using TRVO
  - Credit option hedging using TRVO

- **Optimal execution**
  - Using TS to adapt to the nonstationarity of the markets

### Final Remarks

- Major financial tasks in the Capital Markets sector can be modelled as MDPs
- Broad applicability of RL based techniques to financial problems
- Data driven approaches without explicit modelling assumptions
Current Challenges in Applying RL

- Acquisition of training data
  - Simulation via stochastic models
  - GANs or other advanced ML approaches
- Non-stationarity of the financial markets
  - Market regimes
  - Rare events
- Low signal to noise ratio
  - Control frequency
  - Data processing
- Resistance to trust a completely autonomous trading agent
Future Works

- **Online Portfolio Optimization**
  - Evaluate the feasibility of using in a high frequency trading framework

- **Quantitative Trading**
  - Expand feature set in state, including both microstructural order book facts and possible predictive signals
  - Expand to n asset scenario

- **Hedging**
  - Expand to hedging of a portfolio of derivatives

- **Market Making**
  - Use real data or market simulators in order to introduce realism
  - Combine with hedging

- **Optimal Execution**
  - Improve and generalize the approach
  - Combine with trading
Future Works II

- **Reinforcement Learning**
  - Dealing with non-stationarity
  - Optimal control frequency

- **Monte Carlo Tree Search**
  - Extend algorithms such as Alphazero [Silver et al., 2017] to be compatible with continuous stochastic states
  - Improve the generative model

- **Expert Learning**
  - Analyze potential applications in high frequency scenarios
B. RL Fundamentals
Reinforcement Learning Intro

- Returns

\[ G(\tau) = \sum_{t=0}^{\infty} \gamma^t R_t \]

- Action-Value function

\[ Q_{\pi}(s, a) = \mathbb{E}_{\tau \sim \pi} [G(\tau) | s_0 = s, a_0 = a] \]

- Objective

\[ J = \max_{\pi} \mathbb{E}_{\tau \sim \pi} [G(\tau)] \]
• **Value based** learn the action-value function

\[
Q_\pi(s, a) = \mathbb{E}_{\tau \sim \pi} [G(\tau)|s_0 = s, a_0 = a] = r(s, a) + \gamma \mathbb{E}_{a' \sim \pi, s' \sim P} [Q(s', a')] \quad \text{Bellman Equation}
\]

• **Examples**
  - Q-Learning [Watkins, 1989]
  - FQI [Ernst et al., 2005]
  - DQN [Mnih et al., 2013]

• **Policy search** move in the policy space using experience

\[
\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) G(\tau) \right]
\]

• **Examples**
  - REINFORCE [Williams, 1992]
  - TRPO [Schulman et al., 2015]
  - PPO [Schulman et al., 2017]
C. Quantitative Trading with FQI

Financial markets

Banks

Asset managers

Hedge funds

Liquidity providers

Other market participants

Portfolio Optimization

Quantitative Trading

Market Making

Hedging

Optimal Execution
Approaches to Trading

Background

• **Practitioner approach**
  - Technical analysis
  - Macro-economic analysis

• **Supervised learning approach**
  [Baba and Kozaki, 1992]
  - Forecast asset prices
  - Derive trade
  - Hard to incorporate market frictions

• **Reinforcement Learning approach**
  [Moody and Saffell, 2001]
  - Integrate prediction and action
  - Simple to include market frictions
Approaches to Trading

Background

• Practitioner approach
  - Technical analysis
  - Macro-economic analysis

• Supervised learning approach
  [Baba and Kozaki, 1992]
  - Forecast asset prices
  - Derive trade
  - Hard to incorporate market frictions

• Reinforcement Learning approach
  [Moody and Saffell, 2001]
  - Integrate prediction and action
  - Simple to include market frictions

Main contributions

Learning FX Trading Strategies with FQI and Persistent Actions
[Riva et al., 2021]
  • Use of FQI for FX multi-currency trading
\[ \mathcal{D} = \{(s_k, a_k, r_k, s'_k) | k = 1, \ldots, |\mathcal{D}|\} \]

**Algorithm 2** Fitted Q Iteration Algorithm

**Require:** \( \hat{Q}_0(s, a) \leftarrow 0 \ \forall s \in \mathcal{S}, a \in \mathcal{A} \), number of iterations \( J \), and load dataset \( \mathcal{D} \)

1. for \( j \in [J] \) do
   2. \( \hat{Q}_{j+1} = \text{arg min}_{f \in \mathcal{F}} \sum_{s, a, r, s' \in \mathcal{D}} \left( f(s, a) - r - \gamma \max_{a' \in \mathcal{A}} \hat{Q}_j(s', a') \right)^2 \)
   3. end for
4. Return \( \hat{Q}_j \)

\( \hat{Q} \) as extra-tree regressors \( \rightarrow \) min-split tuning
Two currency model definition

- Two FX pairs with common base currency
- 5 actions: \( a_t \in \{1, 2, 3, 4, 5\} \)
- Portfolio exposure to one FX pair at a time
- Fixed traded amount of base currency: $100k
- Fixed transaction costs: bid-ask = $2 \cdot 10^{-5}
- Doubled costs for certain trades
Model Assumptions

**Trading assumptions**
- Episode = Trading Day = 08:00-18:00 CET
- Close any position end of day

**Training and testing settings**
- Training set: 2017 - 2018
- Validation set: 2019
- Test set: 2020
- Training algorithm: FQI

**MDP assumptions**
- Window of 60 price returns
- Time-step with 1-minute, 5-minute, 10-minute frequency (Persistence)
Validation on the single currency pair EURUSD, averaged over 2 seeds
Test Performances: P&L

<table>
<thead>
<tr>
<th></th>
<th>Persistence</th>
<th>1</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharpe Ratio EUR</td>
<td>-0.22</td>
<td>1.34</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>Sharpe Ratio GBP</td>
<td>-1.37</td>
<td>1.93</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>Sharpe Ratio Both</td>
<td>-1.45</td>
<td><strong>2.02</strong></td>
<td>0.33</td>
<td></td>
</tr>
</tbody>
</table>
Test Performances: Heat Maps

USD-EUR

USD-GBP
Market making: a sequential decision process in which at each round $t \in \{1, \ldots, T\}$ the dealer updates her bid and ask prices to maximize P&L while minimizing inventory.
Market Making as an MDP

State:
- price of the asset: $P_t$ (exogenous)
- the inventory: $z_t = z_{t-1} + v_t \mathbb{I}\{\text{won}_t\}$

Actions:
- $a_1: P_{t,\text{buy}}^i(v) = \tilde{P}_{t,\text{buy}}(v)(1 + a_1)$
- $a_2: P_{t,\text{sell}}^i(v) = \tilde{P}_{t,\text{sell}}(v)(1 + a_2)$

Reward:
\[
rt = \mathbb{I}\{\text{won}_t\} |v_t(P_{t,\text{buy/sell}}(v_t) - P_t)| + z_{t-1}(P_t - P_{t-1}) - \phi(z_t)
\]

where $v_t$ is the size of the trade, $P_{t,\text{buy/sell}}(v_t)$ is the quote published by the market maker, $z_t$ is the inventory, $\phi: \mathbb{R} \rightarrow \mathbb{R}^+$ is the penalty of owning a net inventory.
Approaches to Market Making

Background

- **Classical approach**
  [Avellaneda and Stoikov, 2008]
  - Model the mid-price process and RFQ arrival process
  - Define the market maker's utility function
  - Model auctions as stochastic processes

- **Reinforcement Learning approach**
  [Ganesh et al., 2019]
  - Model the mid-price process and RFQ arrival process
  - Define the behavior of the other dealers
Approaches to Market Making

Background

• Classical approach
  [Avellaneda and Stoikov, 2008]
  - Model the mid-price process and RFQ arrival process
  - Define the market maker’s utility function
  - Model auctions as stochastic processes

• Reinforcement Learning approach
  [Ganesh et al., 2019]
  - Model the mid-price process and RFQ arrival process
  - Define the behavior of the other dealers

Main contributions

• Model as an N-player stochastic game, with multiple competing market makers
• Solve by using mean field games and FQI
• Assume homogeneity/anonymity
• Mean-field $L \in \Delta(A \times S)$ represents players’ distribution
• Nash Equilibrium is a pair $(\pi^*, L^*)$ s.t.
  $V(\pi^*, L^*) \geq V(\pi, L^*), \forall \pi$

### Algorithm 3 Model Free MFG [Guo et al., 2019]

**Require:** mean-field $L_0$, simulator $E(.,.; L)$, iterations $K$

1: for $k \in [K]$ do
2: Find the single-agent optimal policy $\pi_k$ with fixed $L_k$
3: Update $L_{k+1}$ using $E(.,.; L)$
4: end for
5: return $(\pi_k, L_k)$
Experimental Results

- **$\pi_W$** learned policy
- **$z$**: inventory
- **$A = \{-0.03, -0.02, \ldots, 0.03\}$

- **$\rho_t$**: mean dollar reward ($\phi = 0$)
- **$FQI$**: trained with MFG-FQI
- **$N$**: plays $(a_1, a_2) \sim N(0, 1)$

\[ z \approx \pi_W(W, \epsilon_{buy}, \epsilon_{sell}) \]
Experimental Results

- Sharpe ratio box plot

\[ R = \sum_{t \leq T} \frac{\rho_t}{T} \]

- Sharpe ratio \( S = \frac{R}{\text{std}(R)} \)
E. Credit Index Option Hedging with RL

- Banks
  - Asset managers
  - Hedge funds
  - Liquidity providers
- Other market participants
  - Portfolio Optimization
  - Quantitative Trading
  - Market Making
  - Hedging
  - Optimal Execution
- Financial markets
A Credit Default Swap (CDS) is a financial derivative that allows an investor to swap or offset her credit risk with that of another investor.

A receiver option gives the buyer the possibility of selling protection on the index at the expiry date at a spread equal to the strike.

A payer option gives the buyer the choice of buying protection at the expiry date at a spread equal to the strike.
Approaches to Option Hedging

Background

• **Classical approach**
  [Black and Scholes, 1973]
  - Model the market as GBM
  - Assume continuous time hedging
  - Assume no market frictions
  - Solve resulting PDE

• **Reinforcement Learning approach**
  [Kolm and Ritter, 2019]
  - Collect/simulate data
  - Learn to hedge
Approaches to Option Hedging

Background

• Classical approach
  [Black and Scholes, 1973]
  - Model the market as GBM
  - Assume continuous time hedging
  - Assume no market frictions
  - Solve resulting PDE

• Reinforcement Learning approach
  [Kolm and Ritter, 2019]
  - Collect/simulate data
  - Learn to hedge

Main contributions

• Use of the risk-averse policy search RL algorithm: TRVO
• Training and testing on credit index options
• Testing on real data
Financial Environment

Long payer option

- time to maturity = 40 days
- €100mln notional
- implied volatility = 60%
- interest rates = 0
- $K(= S_0) = 100$
- starting price (ATM) option $\sim €530k$
- starting delta $= 0.5$

Simulated Credit Spread

- GBM
- no drift
- $\sigma = 60\%$
- $S_0 = 100$
- 17 observations per day

training on 40k episodes and testing on 2k episodes
Experimental Results: with/without Transaction Costs

delta hedge with no costs → average p&l ~ 0, with costs → average p&l ~ -€136k
Experimental Results: GBM Simulated Market

Distribution of P&L of $\lambda = 4$ agent with $ba = 1.5\text{bps}$
Experimental Results: Heston Simulated Market

Testing on 2k heston simulated episodes

\[ dS_t = \sqrt{\nu_t} S_t \, dW_t^S \]

\[ d\nu_t = \kappa (\theta - \nu_t) \, dt + \xi \sqrt{\nu_t} \, dW_t^{\nu} \]

\( \nu_0 = 60\%^2, \kappa = 2, \theta = \nu_0, \xi = 0.9 \)

no correlation between the stochastic terms \( dW_t^S \) and \( dW_t^{\nu} \).
Testing on real data, with option $\sigma = 60\%$ and $ba = 1bps$
F. Optimal Execution with RL

Banks
- Asset managers
- Hedge funds
- Liquidity providers

Other market participants
- Portfolio Optimization
- Quantitative Trading
- Market Making
- Hedging

Optimal Execution

Financial markets
Optimal execution: a sequential decision process in which at each round $t \in \{1, \ldots, T\}$ over the maximum execution time $T$ and number of time-steps $N + 1$, the trader decides what fraction of the total $X$ shares to execute, to minimize the difference between the arrival price and the execution price.
Approaches to Optimal Execution

Background

• **Practitioner approach**
  - TWAP = $\frac{X}{N} \sum_{r=0}^{N} P_r$

• **Classical approach**
  [Almgren and Chriss, 2001]
  - Model the mid-price process
  - Model the market impact
  - Minimize expected shortfall

• **Reinforcement Learning approach**
  [Hendricks and Wilcox, 2014]
  - Collect/simulate data
  - Model the market impact

• **Multi agent approach using ABIDES**
  - Learn in a multi-agent simulation
Approaches to Optimal Execution

Background

- **Practitioner approach**
  - TWAP = $\frac{X}{N} \sum_{r=0}^{N} P_r$

- **Classical approach**
  [Almgren and Chriss, 2001]
  - Model the mid-price process
  - Model market impact
  - Minimize expected shortfall

- **Reinforcement Learning approach**
  [Hendricks and Wilcox, 2014]
  - Collect/simulate data
  - Model market impact

- **Multi agent approach using ABIDES**
  - Learn in a multi-agent simulation

Main contributions

- Use of FQI to learn multiple execution policies in a multi-agent simulation
- Use of Thompson Sampling to decide which execution policy to use
Optimal Execution as an MDP

**MDP Formulation**

- \( a_t \in \{0, 0.2, 0.4, ..., 4\} \) represents how much of TWAP, i.e., \( \frac{X}{N} \) to execute
- \( s_t = \) stylized microstructural order book facts and internal agent information
- \( r_t = \left(1 - \frac{1}{p_{\text{fill}}}|P_{\text{fill}} - P_{\text{arrival}}|\right) \lambda \frac{n_t}{X} \)

**Environment Formulation**

- \( X = 50,000 \)
- \( N = 180, T = 30 \text{ minutes}, \tau = 10 \)
- Training on 2,000 executions
- Training with FQI [Ernst et al., 2005]
Experimental Performance on Two Scenarios

Performance on Low Volatility Scenario

- High Volatility Expert
- Low Volatility Expert
- TWAP
- AC High Risk Aversion
- AC Low Risk Aversion

Average return over 50, 30-minute executions with 95% confidence intervals

Performance on High Volatility Scenario

- High Volatility Expert
- Low Volatility Expert
- TWAP
- AC High Risk Aversion
- AC Low Risk Aversion
Thompson Sampling for Optimal Execution

\[ i_t = \text{argmax}_i \theta_i \]

Run \( \pi_i \) observe \( r_i \) and update \( f_i \)
Thompson Sampling - Low Volatility Scenario

Distribution after 5 TS iterations

Distribution after 10 TS iterations
Thompson Sampling - High Volatility Scenario

Distribution after 5 TS iterations

Distribution after 10 TS iterations
G. Conservative Online Convex Optimization

- Asset managers
- Hedge funds
- Liquidity providers
- Other market participants

- Portfolio Optimization
- Quantitative Trading
- Market Making
- Hedging

- Optimal Execution

- Financial markets
A **market index** is a collection of financial assets, commonly stocks. The returns of the market index are calculated as a weighted average of the returns of the constituents. The objective of the asset manager is to invest in a subset of the components of the index or to use a different weighting than the index, to outperform the index itself.

<table>
<thead>
<tr>
<th>Conservativeness Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_t \leq \tilde{L}_t(1 + \alpha), \forall t$</td>
</tr>
<tr>
<td>$\cdot \tilde{L}_T$: cumulative loss of the default parameter $\tilde{\theta} \in \Theta$</td>
</tr>
<tr>
<td>$\cdot \alpha &gt; 0$: conservativeness level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conservativeness Objective in OPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_t(\mathcal{U}) \geq \bar{W}_t(1 - \kappa), \forall t$</td>
</tr>
</tbody>
</table>
Approaches to Portfolio Optimization

Background

- **Modern Portfolio Optimization**
  [Markowitz, 1952]
  - Calculate historical variance and correlations
  - Single period

- **Intertemporal CAPM**
  [Merton, 1969]
  - Make assumptions on asset dynamics
  - Multi period

- **Online Portfolio Optimization**
  [Cover and Ordentlich, 1996]
  - Adversarial market
  - From expert learning field
Approaches to Portfolio Optimization

Background

• **Modern Portfolio Optimization**
  
  [Markowitz, 1952]
  
  - Calculate historical variance and correlations
  - Single period

• **Intertemporal CAPM**
  
  [Merton, 1969]
  
  - Make assumptions on asset dynamics
  - Multi period

• **Online Portfolio Optimization**
  
  [Cover and Ordentlich, 1996]
  
  - Adversarial market
  - From expert learning field

---

Main contributions

Conservative online convex optimization

[Bernasconi de Luca et al., 2021]

- Beating a benchmark in OPO
Algorithm 4 CP-A

Require: Algorithm \( \mathcal{A} \), \( \alpha > 0 \), \( \tilde{\theta} \in \Theta \\
1: \text{Set } \tilde{L}_0 \leftarrow 0, L_0 \leftarrow 0, \text{ and } \beta_0 \leftarrow 1 \\
2: \text{for } t \in [T] \text{ do} \\
3: \quad \text{Get point } z_t \leftarrow \mathcal{A}(g_1, \ldots, g_{t-1}) \\
4: \quad \text{Compute } \omega_t := \left[ 1 - \left( \frac{L_{t-1} - (1+\alpha)\tilde{L}_{t-1} \alpha \varepsilon_t}{D \alpha} + 1 \right)^+ \right] D \\
5: \quad \text{Select } \theta_t = \Pi_{B(\tilde{\theta}, \omega_t)}(z_t) \\
6: \quad \text{Suffer loss } f_t(\theta_t) \\
7: \quad \text{Observe } f_t(z_t) \text{ and } f_t(\tilde{\theta}) \\
8: \quad \text{Set } g_t(z_t) \leftarrow (1 - \beta_t)f_t(z_t) \text{ with } \beta_t = \begin{cases} \\
1 - \frac{\omega_t}{\|z_t - \theta_t\|_2} & z_t \notin B(\tilde{\theta}, \omega_t) \\
0 & z_t \in B(\tilde{\theta}, \omega_t) \\
\end{cases} \\
9: \text{end for}
Main Theoretical Result

**Theorem**

For any Online Convex Optimization algorithm $\mathcal{A}$, with regret $R_T(\mathcal{A})$ and $\alpha > 0$, CP-$\mathcal{A}$ attains regret:

$$R_T(\text{CP-}\mathcal{A}) \leq R_T(\mathcal{A}) + \tau DG$$

where $\tau = O(\alpha^{-1})$. Moreover CP-$\mathcal{A}$ is a conservative algorithm.

$D := \sup_{x,y \in \Theta} ||x - y||_2$ is a bound on the diameter of the parameter space $\Theta$

$G := \sup_{x \in \Theta} ||\nabla f_t(x)||_2$ is the upper bound on the norm of the gradient of the loss $f_t(\cdot)$
Experimental Setup

Dataset with minute prices of S&P component stocks from 09/2017 to 02/2018
\( \tilde{\theta} = 100 \) randomly chosen stocks

• Metrics
  • Wealth: \( W_T(\mathbf{u}) = \prod_{t=1}^{T} \langle a_t, y_t \rangle \)
  • Wealth budget: \( P_t(\mathbf{u}) = W_t(\mathbf{u}) - (1 - \kappa)\tilde{W}_t \)

• Algorithms
  • Online Gradient Descent [Zinkevich, 2003]
  • CRDG [Streeter and McMahan, 2012]
  • CS-OGD
  • CP-OGD
Experimental Results

![Graphs showing performance of different algorithms](attachment:image.png)

- $W_t(U)$
- $P_t(U)$

- CP-OGD
- CS-OGD
- CRDG
- OGD
H. State of the Art
Option Hedging with RL


Trading with RL

  *Expert Systems with Applications*, 173:114632

  *arXiv preprint*

  *Available at SSRN*

  *arXiv preprint*

  *Data*, 4(3):110
  *IJCAI*

  *arXiv preprint*

  *Applied Mathematical Finance, 26*(5):387–452

  *arXiv preprint*
Optimal Execution with RL


   Available at SSRN

   Available at SSRN

  Multi-period trading via convex optimization.
  arXiv preprint


  The review of Economics and Statistics, pages 247–257


  In *ICAIF*

  In *ICAIF*

  In *IJCAI. Special Track on AI in FinTech.*
<table>
<thead>
<tr>
<th>Machine Learning</th>
<th>ML in Finance</th>
<th>Quant Finance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neurips</td>
<td>ICAIF</td>
<td>Mathematical Finance</td>
</tr>
<tr>
<td>ICML</td>
<td>The Journal of Financial Data Science</td>
<td>Finance and Stochastics</td>
</tr>
<tr>
<td>IJCAI</td>
<td></td>
<td>Applied Mathematical Finance</td>
</tr>
<tr>
<td>AAAI</td>
<td></td>
<td>Risk Magazine</td>
</tr>
<tr>
<td>ECML</td>
<td></td>
<td>Journal of Empirical Finance</td>
</tr>
<tr>
<td>Journal of Machine Learning</td>
<td></td>
<td>Journal of Computational Finance</td>
</tr>
<tr>
<td>Learning Research</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I. Additional Material
Experiments: Wealth $W^C_t(\mu)$

Specific run, on the Corona dataset for $\gamma = 0$

Specific run on 5 stocks of the NYSE(O) for $\gamma = 0.01$
Experiments: Average APY

Average Average Annual Percentage Yield $A(W_T)$ computed on the wealth $W_T^C(a_{1:T}, y_{1:T})$:

$$A(W_T) = W_T^{250/T} - 1$$
Experiments: Average variation of the portfolio

Average variation of the portfolio incurred on a varying time horizon $t$:

$$V_t(U) = \frac{C_t(U)}{\gamma_t}$$
Problem Context

Dataset → Learning Method → Trained Model

Datastream → Trained Model → Prediction
Empirical Risk Minimization vs Online Optimization

**ERM**
- Samples are generated from a distribution
- Minimize expected loss given a collection of samples (dataset)
- Subject to Adversarial attacks and Concept Drift
- Voice to text, image classification, Natural Language Processing

**Online Optimization [Hazan, 2019]**
- Allows samples to be generated by an adversary
- No assumption on the distribution of the data
- No guarantees on the first phase of the learning process
- Spam classification, Malware detection, Fraud detection

How to obtain a best of both worlds approach and obtain an online algorithm which has controlled performance at each time?
The Conservative Switching Algorithm

Algorithm 5 CS-$\mathcal{A}$

Require: Online learning algorithm $\mathcal{A}$, conservativeness level $\alpha > 0$, default parameter $\tilde{\theta} \in \Theta$

1: Set $\tilde{L}_0 \leftarrow 0$, $L_0 \leftarrow 0$
2: for $t \in [T]$ do
3: \hspace{1em} if $L_{t-1} + \epsilon_u - (1 + \alpha)\epsilon_l \leq \tilde{L}_{t-1}(1 + \alpha)$ then
4: \hspace{2em} $z_t \leftarrow \mathcal{A}(f_{t-1}(z_{t-1}))$
5: \hspace{2em} Select $\theta_t \leftarrow z_t$
6: \hspace{1em} else
7: \hspace{2em} $z_t \leftarrow z_{t-1}$
8: \hspace{2em} Select $\theta_t \leftarrow \tilde{\theta}$
9: \hspace{1em} end if
10: \hspace{1em} Suffer loss $f_t(\theta_t)$
11: \hspace{1em} Observe feedback $f_t(z_t)$ and $f_t(\tilde{\theta})$
12: end for
Experimental Setup

Tasks

- Linear Regression: Synthetic data
- Binary Classification: IMDB and SpamBase

Metrics

- Budget: $Z_t = \tilde{L}_t(1 + \alpha) - L_t$
- Regret: $R_t$

Algorithms

- Online Gradient Descent [Zinkevich, 2003]
- ADAGRAD [Duchi et al., 2011]
- CRDG [Streeter and McMahan, 2012]
- CS-OGD
- CP-OGD
Results: Synthetic Data

(a) $R_t(U)$

(b) $Z_t(U)$

- Adagrad
- CP-OGD
- CRDG
- CS-OGD
- OGD
Results: IMDB

![Graphs comparing different optimization algorithms](image)

- Adagrad
- CP-OGD
- CS-OGD
- OGD
ABIDES realistically replicates the financial market environment reproducing the characteristics of electronic markets:

- Continuous double-auction trading
- Network latency and agent computation delays
- Communication solely by means of standardized message protocols

It is possible to create a multi-agent composition using pre-defined agents such as the exchange agent, value agents, momentum agents, noise agents and market maker agents or using custom made agents.

The price process is described by a fundamental value.
ABIDES [Byrd et al., 2019] reproduces the characteristics of electronic markets such as continuous double-auction trading and network latency.