Examples of Reinforcement Learning Applications in the Financial Market

Dr. Heuna Kim
https://hahey.github.io

Papers to Discuss

- **Idiosyncrasies and challenges of data driven learning in electronic trading**, Vangelis Bacoyannis, Vacslav Glukhov, Tom Jin, Jonathan Kochems, Doo Re Song, 2018

- **AlphaStock: A Buying-Winners-and-Selling-Losers Investment Strategy using Interpretable Deep Reinforcement Attention Networks**, Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, Zhang Xiong , 2019
  cf. FDDR, 2016

- **Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis,** Wenhang Bao, Xiao-yang Liu, 2019

Some remarks

Expectation to the audiences

Comparisons to other domains:
- Practicality
- Benchmarks
- Publishing culture
- Effectiveness of Reinforcement Learning

Side notes:
- Black-Scholes Equations in 1973
- the relations between three papers

'Idiosyncrasies . . .' by the JP Morgan Group, in NIPS Workshop 2018

- Perspectives
  - three data-based cultures
  - Macro- and Micro-decision making

- RL Adaptation
  - Hierarchical Reinforcement Learning (HRL) using Ray RLib
  - Certainty Equivalent modification of standard Reinforcement Learning (CERL)

Three cultures of data-centric applications in quantitative finance

- Data Modelling Culture
- Machine Learning Culture
- Algorithmic Decision-making Culture

High-level and low-level decision making in electronic trading

- An optimal execution rate
- A collection of child orders as an action and exploding dimensions
- Temporal abstraction on a semi MDP is a challenge due to the inhomogenity of the temporal dimension

  $\Rightarrow$ needs of hierarchical decision making

- Efficiency concerning risks
- Interpretability and variable regulations

Hierarchical Reinforcement Learning scheme

- search-based meta-policy (hyper-parameter) optimization
- Coordinating global rewards and local rewards
- References to note:

  - Kulkarni et al., 2016, Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. NIPS 2016.
  - recommended lecture: HRL by D. Precup

  - scalable DRL: Gorilla (2015) for DQN, IMPALA (2018) for A3C
  - RL Frameworks: OpenAI baselines, ELF, Horizon, dopamine, TRFL, Ray RLlib

The centainty equivalent (CE) modification of the standard RL Equation

Certainty Equivalent (CE):

$$\mathrm{CE}(\pi(a_i|s_i)) = U^{-1}\mathbf{E}[U(r_{i+1}(\pi(a_i|s_i)+\max_{\pi(a_{i+1}|s_{i+1})}\mathrm{CE}(\pi(a_{i+1}|s_{i+1})))]$$

where U and U^{-1} is the utility function and its inverse,
$\pi(a_i|s_i)$ is the policy π action in the state s_i and
$r_{i+1}(\pi(a_i|s_i))$ is its uncertain reward.

References to note:
Bühler et al., Deep Hedging, 2018
Mihatsch and Neuneier, Risk-sensitive Reinforcement Learning, 2002

'AlphaStock ...' in KDD 19

- Abstractions in investment strategies
- AlphaStock Model
- Investment Strategy Interpretations

Terminology

- Holding Period
- Portfolio
- Long- and Short- Position

- Buy-Winners-and-Sell-Losers Strategies
- Sharpe Ratio

Buy-Winners-and-Sell-Losers (BWSL) strategy

- At time t, given a budget constraint $\tilde{M}$, to get the money $\tilde{M}$
  we borrow $\tilde{M} \cdot \frac{b_t^{-(i)}}{p_t^{(i)}}$ loser stocks from brokers and sell
  we use $\tilde{M}$ to buy $\tilde{M} \cdot \frac{b_t^{+(i)}}{p_t^{(i)}}$ winner stocks

- At the end of the t-th holding period,
  we sell stocks in the long portfolio earning money M_t^+ and
  buy stocks back in the short portfolio paying money M_t^- and
  return them to the broker.

$$M_t^+ = \sum_{i=1}^{I} \tilde{M} \cdot b_t^{+(i)} \frac{p_{t+1}^{(i)}}{p_t^{(i)}}, \quad M_t^- = \sum_{i=1}^{I} \tilde{M} \cdot b_t^{-(i)} \frac{p_{t+1}^{(i)}}{p_t^{(i)}}$$

- Profitable as long as $\sum_{i=1}^{I} b_t^{+(i)} \frac{p_{t+1}^{(i)}}{p_t^{(i)}} > \sum_{i=1}^{I} b_t^{-(i)} \frac{p_{t+1}^{(i)}}{p_t^{(i)}}$

Sharpe Ratio

the average return in excess of th risk-free return per unit of volatility:

$$H_t = \frac{A_T - \Theta}{V_T}$$

where
$A_T = \frac{\sum_{t=1}^{T}(R_t - TC_t)}{T}$ the average rate of return per holding time T,
$V_T = \sqrt{\frac{\sum_{t=1}^{T}(R_t - \overline{R}_t)^2}{T}}$ the volatility for the risk.
(TC : transaction cost, $\overline{R}$: avg(R))

For the portfolio

$$\arg \max_{\{\mathbf{B}^+, \mathbf{B}^-\}} H_T(\mathbf{B}^+, \mathbf{B}^-)$$

where $\mathbf{B}^+$ and $\mathbf{B}^-$ are the long and short portfolio sequences.

AlphaStock Model

- LSTM- History state Attention (LSTM-HA) for extracting stock representation

- **Cross-Asset Attention Network (CAAN) for selecting Winners and Losers**

- Portfolio Generator

- a RL agent minimizing Sharpe Ratio

The basic CAAN Model

Given the representation vector $\mathbf{r}^{(i)}$ at t, query, key, and value vectors with W's as parameters to learn:

$$\mathbf{q}^{(i)} = \mathbf{W}^{(Q)}\mathbf{r}^{(i)}, \mathbf{k}^{(i)} = \mathbf{W}^{(K)}\mathbf{r}^{(i)}, \mathbf{v}^{(i)} = \mathbf{W}^{(V)}\mathbf{r}^{(i)}$$

using the interrelationship of stock j to stock i to query the key $\mathbf{k}^{(j)}$ of stock j

$$\beta_{ij} = \frac{\mathbf{q}^{(i)\intercal} \cdot \mathbf{k}^{(j)}}{\sqrt{D_k}}$$

where D_k is a rescale parameter. (Attention is all you need)
Define an attenuation score: $\mathbf{a}^{(i)} = \sum_j^J \mathbf{softmax}(\beta_{ij}) \cdot \mathbf{v}^{(j)}$
Finally the winner score where f is a linear function to learn:

$$s^{(i)} = \text{sigmoid}(f(a^{(i)}))$$

The CAAN Model with price rising rank prior

$c_{t-1}^{(i)}$: the rank of price rising rate of stock i in the holding period from $t-1$ to t

discrete relative distance: $d_{ij} = \lfloor \frac{|c_{t-1}^{(i)} - c_{t-1}^{(j)}|}{Q} \rfloor$
where Q preset quantization coefficient.

$$\beta_{ij} = \frac{\psi_{ij}(\mathbf{q}^{(i)\intercal} \cdot \mathbf{k}^{(j)})}{\sqrt{D}}$$

where
$\psi_{ij} = \text{sigmoid}(\mathbf{w}^{(L)\intercal}\mathbf{l}_{d_{ij}})$,
$\mathbf{l_{d_{ij}}} = (\delta_{idx,d_{ij}})_{idx}$,
$\delta_{a,b} = 1$ if $a = b$ otherwise 0.

Performance in U.S. markets (also in Chinese markets in the paper)

A (Annualized), PR (Percentage Rate), VOL(Volatility), SR (Sharpe Ratio), MDD (Maximum DrawDown), CR (Calmar Ratio), DDR (Downside Deviation Ratio)

Table 1: Performance comparison on U.S. markets.

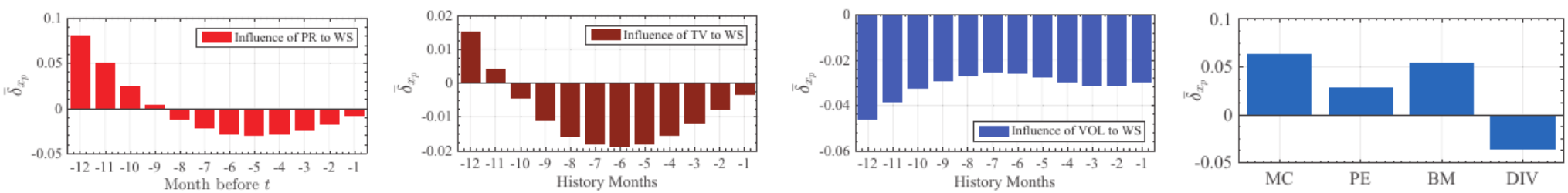
	APR	AVOL	ASR	MDD	CR	DDR
Market	0.042	0.174	0.239	0.569	0.073	0.337
TSM	0.047	0.223	0.210	0.523	0.090	0.318
CSM	0.044	0.096	0.456	0.126	0.350	0.453
RMR	0.074	0.134	0.551	0.098	1.249	0.757
FDDR	0.063	0.056	1.141	0.070	0.900	2.028
AS-NC	0.101	**0.052**	1.929	0.068	1.492	1.685
AS-NP	0.133	0.065	2.054	0.033	3.990	4.618
AS	**0.143**	0.067	**2.132**	**0.027**	**5.296**	**6.397**

Model Interpretation

the influence of the history state $\mathbf{X} = (x_q)_q$ at a particular period of the look back window to its winner score $s = \mathcal{F}(\mathbf{X})$ is

$$\delta_{x_q}(\mathbf{X}) = \lim_{\Delta x_q \to 0} \frac{\mathcal{F}(\mathbf{X}) - \mathcal{F}(x_q + \Delta_q, \mathbf{X}_{\neg x_q})}{x_q - (x_q + \delta x_q)} = \frac{\partial \mathcal{F}(\mathbf{X})}{\partial x_q}$$

Use the approximated discrete average of $\delta_{x_q}, \overline{\delta_{x_q}}$ to compare features



(a) Price Rising (b) Trade Volume (c) Fine-grained Volatility (d) Company Features

Figure 3: Influence of history trading features to winner scores.

MC= Market Capitalization, PE= Price-earning Ratio,
BM= Book-to-market Ratio, Dividend (Div)

Related Financial Investment Strategies

- Cross Sectional Momentum (CSM), Jegadeesh and Titman, 2002

- Time Series Momentum (TSM), Mokowitz et al., 2012

- The Mean Reversion Strategy, Poterba and Summers, 1988

- The Multi-factor Model, Fama and French, 1996

AlphaStock claims a long-term momentum but short-term reversion mixed strategy

'Multi-Agent . . . for Liquidation . . . ' in ICML 2019

- Extension of Almgren and Chriss Model (2001) and analysis
- Multi-agent cooperation and competition analysis and experiment
- Multi-agent Trading Trajectory

Optimal Liquidation Problem

For a liquidation trader
who sells X shares of one stock within a time T
with a risk aversion level λ

assuming
i) the trader does not buy new stocks,
ii) the volume X is large enough to drop the market price

If there are J traders, find an optimal selling strategy
minimizing the expected trading cost $E(X)$,
"implementation shortfall"

Almgren and Chriss market impact model

$$P_k = P_{k-1} + \sigma\tau^{1/2}\xi_k - \tau g(\frac{n_k}{\tau}), k = 1, \ldots, N$$

where
P: price, σ: the volatility of the stock, ξ: random white noise, $\tau = T/N$,
n_k: the number of shares to sell during time interval t_{k-1} to t_k,
N: the total number of trades,
$g(v) = \gamma v$: linear permanent impact function

$$h(\frac{n_k}{\tau}) = \epsilon \cdot \mathrm{sgn}(n_k) + \frac{\eta}{\tau} n_k$$

where
h: temporary impact function, ϵ: the fixed costs of selling,
η: parameter for internal and trasient aspects of the market micro-structure.

A MDP adaptation of the Almgren and Chriss model

$$R_t = U_t(\mathbf{x}_t^*) - U_{t+1}(\mathbf{x}_{t+1}^*)$$

where R: reward, $\mathbf{x}^*$: the computed optimal trading trajectory, using the Almgren and Chriss model

$$U(\mathbf{x}) = E(\mathbf{x}) + \lambda V(\mathbf{x})$$

$$E(\mathbf{x}) = \sum_{k=1}^{N} \tau x_k g(\frac{n_k}{\tau}) + \sum_{k=1}^{N} n_k h(\frac{n_k}{\tau})$$

$$V(\mathbf{x}) = \sigma^2 \sum_{k=1}^{N} \tau x_k^2$$

$\mathbf{x_t} = (x_{t_k})_k = (X - \sum_{j=1}^{k} n_j)_k$: the number of shares remaining at time t_k

Optimal Multi-agent Liquidation Shortfall: Analysis and Performance

The total expected shortfall $\geq$ The sum of the expected single agent shortfall

$$E(\sum_{j=1}^{J} X_j) \geq \sum_{j=1}^{J} E(X_j)$$

because $E(X) = \frac{\gamma X^2}{2} + \epsilon X + \tilde{\eta}\phi X^2$ where ϕ is an environmental setting parameter.

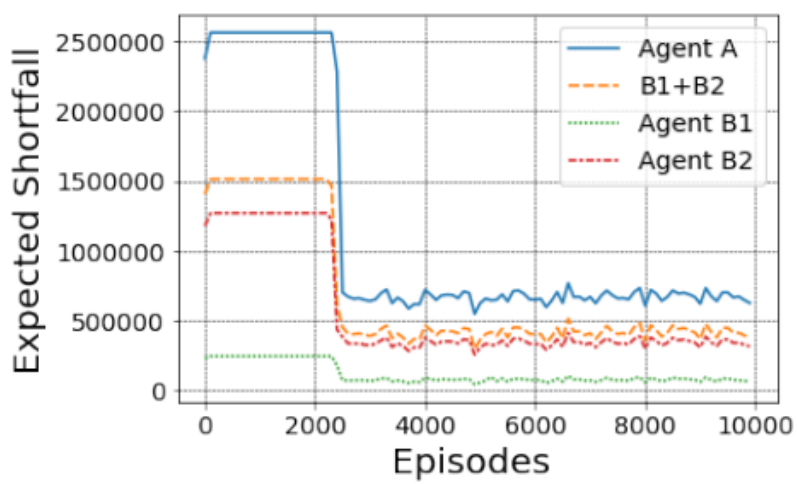


Figure 2. Comparison of expected implementation shortfalls: there are three agents A, $B1$ and $B2$. The expected shortfall of agent A is higher than the sum of two expected shortfalls B_1 and B_2.

Multi-agent Interactions using DRL: Setting

Limited state observations of an agent j at time k,

$$O_{j,k} = [r_{k-D}, \ldots, r_{k-1}, r_k, m_k, l_{j,k}]$$

where r: the log-return,
m: the number of trades remaining normalized by the total number of trades,
l: the remaining number of shares normalized by the total number of shares

Deep Deterministic Policy Gradients (Lillicrap et al., 2016)
-based A3C (Mnih et al., 2016) Multi-agent Training
(Lowe et al., 2017) with Experienced Replay method

Multi-agent Interactions using DRL: Analysis and Evaluation

$$\mathbf{x}^*(\lambda_j) \neq \mathbf{x}^*(\lambda_j) \quad \text{for} \quad j = 1, 2$$

where $\mathbf{x}^*$: the optimal single-agent trajectory, $\mathbf{x}$: the biased trajectory, and each of them has the same number of stocks to liquidate because $U(\mathbf{x})$ contains a quadratic function of $\mathbf{x}$.

Cooperation Rewards: $\tilde{R}^*_{j,t} = \frac{\tilde{R}^*_{1,t} + \tilde{R}^*_{2,t}}{2}$

Competition Rewards:

$$\tilde{R}^*_{j,t} = \tilde{R}_{j,t}, \quad \tilde{R}^*_{j',t} = \tilde{R}_{j',t} - \tilde{R}_{j,t} \quad j = \arg\max_i \tilde{R}_{i,t}$$

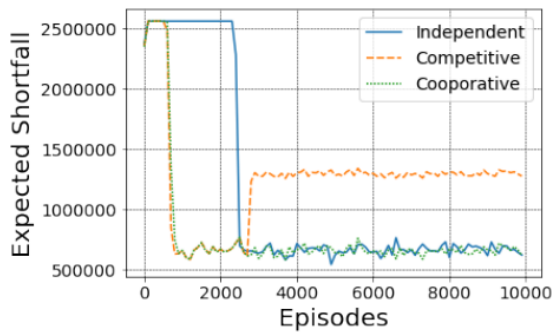


Figure 4. Cooperative and competitive relationships: if two agents are in cooperative relationship, the total expected shortfall is not better than training with independent reward functions. If two agents are in a competitive relationship, they would first learn to minimize expected shortfall, and then malignant competition leads to significant implementation shortfall increment.

Liquidation Trajectory

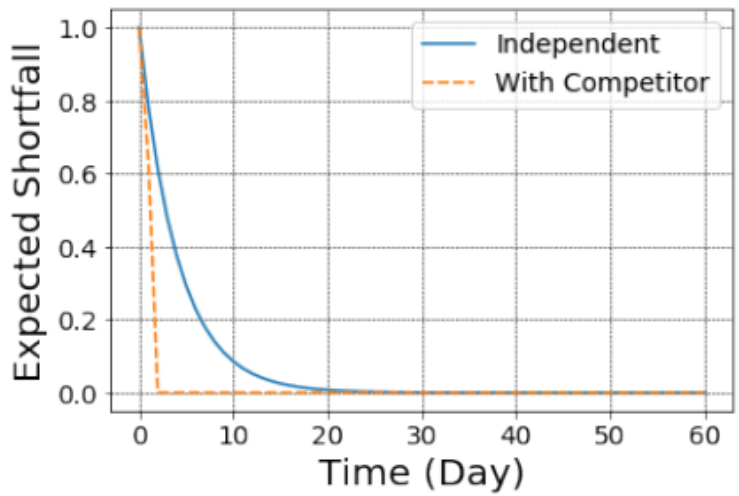


Figure 5. Trading trajectory: comparing to independent training, introducing a competitor makes the host agent learn to adapt to new environment and sell all shares of stock in the first two days.

Thank You!