Idiosyncrasies and challenges of data driven learning in electronic trading

NeruIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services

Vangelis Bacoyannis, Vacslav Glukhov, Tom Jin & Jonathan Kochems

presented by Shijie Huang

Brain, Mind and Markets Lab

May 2019

Table of Contents

Table of Contents

Overview of Trading

Often, portfolio requires **periodic and radical rebalances**, which can only be conducted by ultra-fast agency electronic trading.

Nowadays most micro-level trading decisions in equities and electronic future contracts are made by *algorithms*: they define **where to trade, at what price, and what quantities.**

In designing trading algorithms, execution brokers have to consider two general problems with multiple objectives.

Problem 1: Client specific requirements

Clients have specific constraints and preferences, examples:

- want to preserve currency neutrality in portfolio transitions.
- express their risk preferences.
- want exposure to certain sectors, countries or industries.
- want some optimality between market (price) impact and risk (volatility).

Problem 2: Regulatory requirements

The financial services industry is heavily regulated, examples:

- "best execution" in EMEA (European Securities and Market Authority 2014).
- Best practices in Algorithmic Trading Compliance.
- Different financial regulatory bodies and requirements in each country/region.

Problem 2: Regulatory requirements

The financial services industry is heavily regulated, examples:

- "best execution" in EMEA (European Securities and Market Authority 2014).
- Best practices in Algorithmic Trading Compliance.
- Different financial regulatory bodies and requirements in each country/region.

Traditional electronic trading algorithms

A blend of quantitative models:

- express quantitative views of how the (financial) world works
- rules and heuristics that express practical experiences, observations and preferences of human traders.
- thousands of lines of hand-written code.

However, those algorithms suffer from **feature creep:**

- changes or expansions consumes much more time than the original development.
- eventually accumulates many layers of logic, parameters, and tweaks to handle special cases.

Goal of this paper

To present practical challenges and idiosyncrasies which arise in electronic trading.

Table of Contents

No. 1: Data modelling culture (Black box approach)

- black box with a (simple) model.
- treat market as a DGP.
- functional approximation and extracting parameters.
- However, complexity of markets and behaviours of market participants show that this approach does not capture all essential properties of market environment.

No. 2: Machine learning culture

- empirically the world of finance looks more Darwinian than Newtonian: it constantly evolving; are best described as emerging behaviours rather than DGP.
- **complex** functions are used to model the observations
- risk of the model failure increases with its complexity.

No. 3: Algorithmic decision-making culture

- the focus here is on **decision-making** rather than model-building.
- train electronic agents to distinguish good decisions from bad decisions.
- we still need to inject values, rules and constraints that steer the agents away from taking actions which we view as **prohibited.**

Remarks

- The categorization is not very clear to me.
- In particular, the difference between No.1: Data modelling culture and No.2: machine learning culture is not subtle.
- From a practical viewpoint, most of them involve training, fix parameters, testing out-of-sample (parametric approach), or rely on heavy assumptions (non-parametric approach)

Table of Contents

In this paper, they give an overview of specific challenges in empirical trading from a decision point of view, and they employ reinforcement learning to tackle those challenges.

High level decision-making

Consider a \$10 million order. The general practice is that the **parent order** is divided into several **children orders**. In theory, there should be an optimal execution rate, i.e. speed with which order is executed. There are trade-offs in the following criteria:

- size of the order: if the order size is too big, huge market impact; if the order size is too small, it may take longer time to execute.
- speed of the order: if orders are executed too fast, it is going to have market impacts; if orders are executed too slow, the remaining orders suffer from the market fluctuation.

High level decision-making: traditional approach

- the efficient rate is determined by the client's tolerance to market impact and appetite for risk. (i.e., they ask their clients which one they prefer)
- This is an example of high-level decision-making under uncertainty informed by high-level analytic and quantitative models.

There are no solutions, only trade-offs

Low level (implementation level) decision-making

- Assume that we have a rough idea on optimal rate, now it's time to place the $10 million order(s). For example, say $100,000 an order, 10 sec each, that is, 1000 seconds (or 17 mins) to finish the entire order.
- Problem is, over 17 mins, those concurrent orders reveal your intention.
- Approach to that problem is to create market orders that mimic other participants' orders-both in size and in prices.

Low level (implementation level) decision-making

The agent has to decide:

- at what price
- at what quantity to place
- whether multiple orders at different prices or
- make additional orders at prices where we already have order in place

at a single second/microsecond (a given state).

RL framework for trading

We need to define:

- state space
- action space
- rewards

Problems exist in every single dimension.

State space dimensionality explosion

- Each price level is a queue of differently sized orders; each queue can be arbitrarily long or empty
- there are many bid and ask prices at each point of time
- the order book is in constant change
- potentially infinite market states

Action space dimensionality explosion

- action space is also dynamic and complex
- at what price
- at what quantity
- multiple orders at the different prices or at the same price
- through which trading venues?
- in what order type?

Reward problem

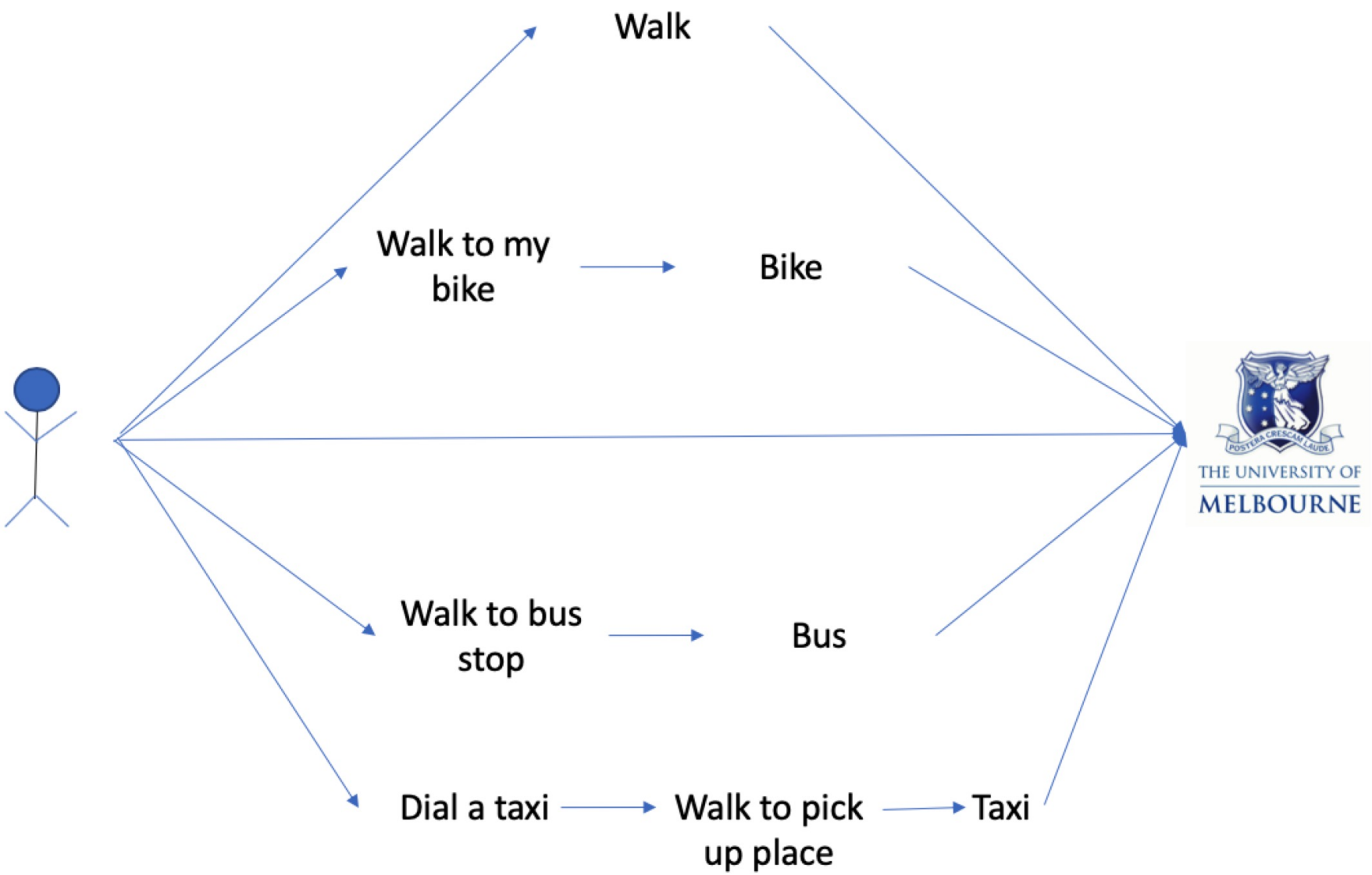
- whether the trade is going to be good or bad is not known until well after the trade is executed (or avoided)
- Local optimality vs. global optimality: what could be considered as a bad trade now could turn out to be an excellent trade by the end of the day.

Problem with hierarchical decision making

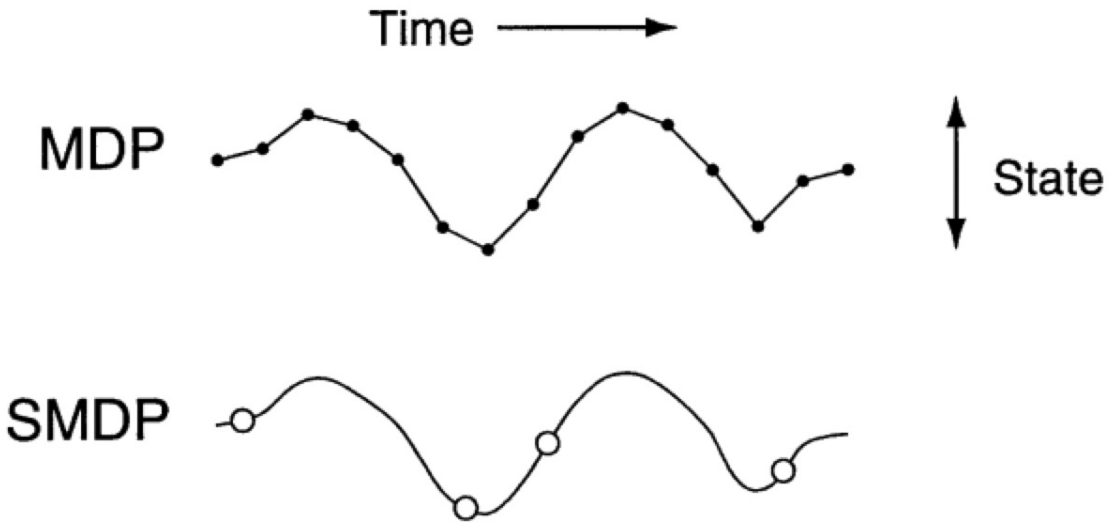
- $10 million orders could take minutes or even hours and days to finish
- trading agents need to make decision every few seconds or faster
- however the market moves every single millisecond.
- agent's sampling frequency is far lower than what is necessary to **fully integrate** all available information.
- yet at the same time, the high level thinking needs to be maintained/adjusted as the markets move over time.

Table of Contents

Temporal abstraction or temporal extended action

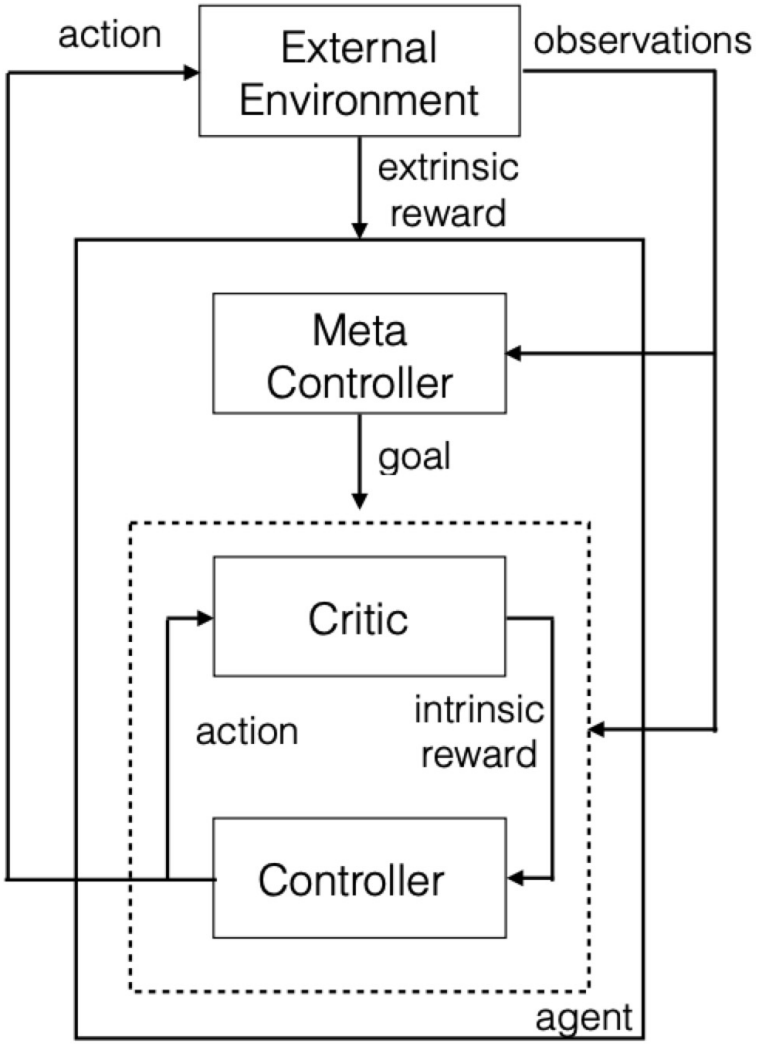


semi-MDP to model temporal abstraction



- So we have semi-MDP to (hopefully) model temporal abstraction, but we still need to model the optimality in high-level and low-level objectives.
- the proposed solution is hierarchical reinforcement learning

Hierarchical RL



Incorporating uncertainty in RL framework

- There is inherent uncertainty of outcomes in finance.
- In standard RL theory, agents learn actions that lead to a better scalar-valued outcome **on average**.
- Finance guys not only value aggregate outcomes, but also value the tails of the distributions of outcomes.

Incorporating uncertainty in RL framework

- Using certainty equivalent concept: uncertain outcomes and their aggregates are ranked by taking the expectation of the utility function of outcomes over their future distribution.

Incorporating uncertainty in RL framework

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

where U and U^{-1} is the utility function and its inverse, E denotes expecation, CE denotes certainty equivalent: $CE(.) = U^{-1}E[U(.)]$, $\pi(a_i|s_i)$ is the policy π action in the state s_i, and $r_{i+1}(\pi(a_i|s_i))$ is its uncertain reward.

Incorporating uncertainty in RL framework

- They claim that a discount factor γ is not necessary in the CE bellman equation.
- In standard RL, a discount factor is introduced as an exogenous parameter for infinite or nearly infinite process.
- In CERL, it is naturally derived as the consequence of the broadening distribution of outcomes, which is an equivalent of the increased risk (I don't understand their argument, are two discount factors, one in finance, one in RL, the same thing?).

- Is there a rigorous way to account for multidimensional rewards?
- How to incorporate the concept of processes of uncertain duration into the MDP paradigm?
- How to tackle uncertain outcomes/rewards?
- How to create realistic training environments for market-operating agents? A possible solution is to develop full scale artificial environment realistically reproducing markets as *emergent phenomena* arising from rule-based activities of multiple heterogeneous agents Simulated multi-agent markets will have both practical and academic value.
- How to rigorously combine conflicting/complementary local and global rewards?
- Other than using domain knowledge to separate processes of different time scales, and using hierarchical training, is there a rigorous way to design agents operating on multiple time scales?

- Scalability: in electronic trading it seems computationally efficient to train many agents operating in similar, but ultimately distinct environments, rather than one agent which is supposed to handle all environments. Is there a way for the agents trained for different environments to benefit from each other's skills? Other than testing their functionality, is there a way to tell that two trained agents are intrinsically similar?
- Bellman's equation in either classical RL or CERL is not fundamental and ultimately seems applicable only to processes where the global reward is a sequential aggregate of local rewards. Can a more general approach to sequential decision-making be developed which will incorporate the above characteristics?
- Is there a balanced and systematic approach which, on one hand, allows RL-trained agents to tackle increasingly complex problems and on the other hand, still preserves our ability to understand their behaviours and explain their actions.

References I

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, *112*(1-2), 181–211.