ACTIVE LEARNING IN TRADING ALGORITHMS

J.P. Morgan

Linear Quantitative Research | David Fellah

November 10, 2016

QuantCon - Singapore, 2016

Section 1: Market response

1. Review of the algorithmic trading problem: forecasting and feedback mechanisms
2. Impact modeling
2. Empirical examples - real world trading is a complex, dynamical problem

Section 2: Machine learning in forecasting

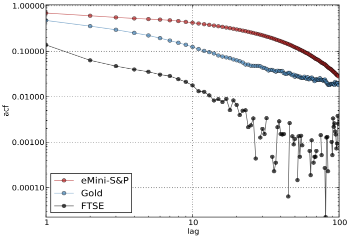
1. Forecasting: Random Forests for high-frequency price direction prediction
2. Empirical results

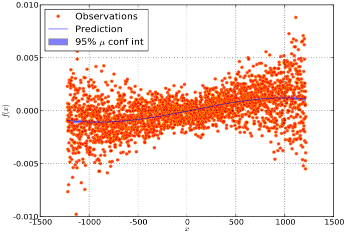
Section 3: Reinforcement Learning

1. Putting it all together: Reinforcement Learning
2. Problem setup
3. Learning

Market response

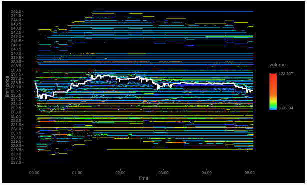
1. 60-80% of trading occurs algorithmically - most trading occurs between machines.
2. Auto-correlation is a measure of concordance with previous observations: how likely do we observe another buyer-initiated trade?
3. Factors that contribute to the autocorrelation of signed trade-flow Parent order splitting, market fragmentation.
4. Joint information sets: Left plot - information due to trading activity. Right plot - information due to limit order placement activity.

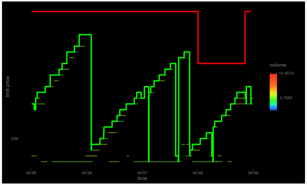


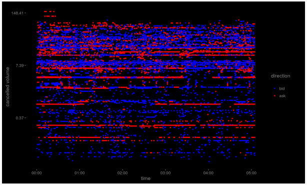


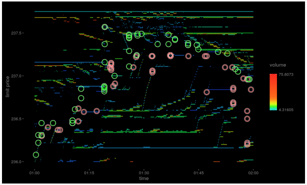
1. | Microstructure review

- Limit orders cluster at varying price levels - reveals latent supply.
- Buyer/seller-initiated orders generate trade flow - reveals intent.
- Price patterns reveal competing agents - typical of algorithms exhibiting passive bias.
- Cancellation patterns reveal the transient nature of liquidity.
- Impact is dynamic - a model is needed for training, intra-trade planning, post-trade attribution.







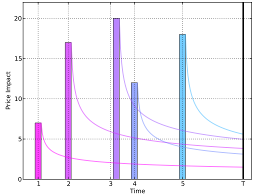


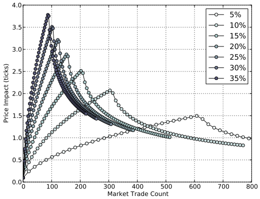
A transient model of market impact provides a consistent transform between orders executed at the "micro"-scale and the aggregate, "parent" or "meta" order trading costs. Practically speaking, this enables the formulation of a class or family of models that operate at the threshold between signals and scheduling, depending on the amount of information available at the point of decision.

- Micro-formulation is suitable for trading decisions at the micro-scale (order placement).
- f models the immediate market response, following a transaction.
- G models the subsequent decay or memory effect.

$$p_t = p_0 \times (1 + \int_0^t \overbrace{f(.)}^{impulse} \; \overbrace{G(t-s)^{-\omega}}^{decay} + \delta(.)\, ds) \quad (1)$$

Description	$f(.)$	$G(.)$	$\delta(.)$
$MI.1$	$\alpha\sigma^{\beta}\nu^{\gamma}$	$(t-s)$	$0.5\widehat{spread}$
$MI.2$	$\alpha\sigma^{\beta}\nu^{\gamma}$	$(t-s)^{1-\omega}$	$\delta\widehat{spread}$
$HFMI$	ψ_s	$(t-s)^{1-\omega}$	$\delta\widehat{spread}_s$





Low-frequency market impact

A transient model of market impact provides a consistent transform between orders executed at the "micro"-scale and the aggregate, "parent" or "meta" order trading costs. Practically speaking, this enables the formulation of a class or family of models that operate at the threshold between signals and scheduling, depending on the amount of information available at the point of decision.

- Macro-formulation is suitable for scheduling/allocation decisions.

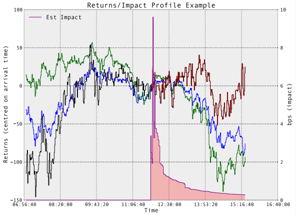
$$p_t = p_0 \times (1 + \int_0^t \overbrace{\alpha \sigma^{\beta} \nu_s^{\gamma}}^{impulse} \overbrace{(t-s)^{-\omega}}^{decay} ds) + \epsilon_t + \delta \times \Delta$$

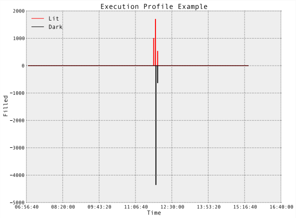
(2)

where

α	:	Scale parameter
β	:	Volatility parameter
γ	:	Trade Rate parameter
ω	:	Impact Decay parameter
δ	:	Spread parameter
ϵ_t	:	Noise term, $\epsilon_t \overset{H_0}{\sim} IID(0,\sigma)$

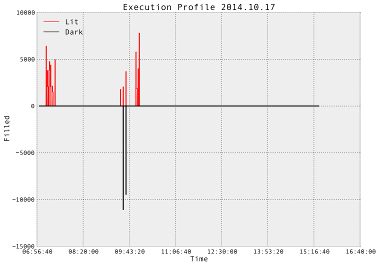
Region	α	β	γ	ω	δ
NA	488	0.89	.663	1-0.325	0.118
APAC	389	0.857	0.573	1-0.393	0.102
EMEA	349	0.734	0.65	1-0.39	0.066

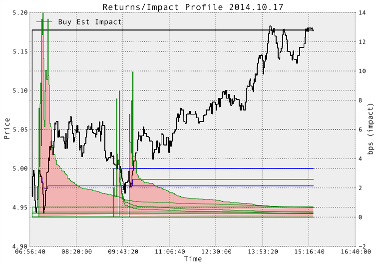




Impact accrual over multiple submissions

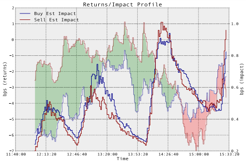
1. Consider a single parent order, with several lit and dark executions occurring sequentially.
2. Impact decays as a power-law over volume time.
3. The decay of impact is accounted in optimal scheduling.



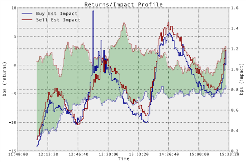


1. | Microstructure review

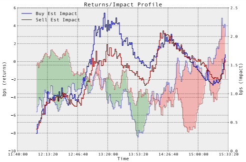
■ Basket Impact Profiles. Slippage is struck from arrival. The blue series represents the buy-side and sells are red; each group is a dollar-weighted basket. Prior to order arrival sells exceed buys versus arrival, and the area is shaded green. As orders arrival and trading begins, impact moves sells lower and buys higher. The impact estimate is on the second (right) y-axis. For example, buys move by +15bps and the estimated impact is roughly 10. The darker red/blue lines are the accumulated impact curves and subsequent decay as orders complete. Completion time would roughly correspond to the peak of impact. After this point, in this sample - buys revert and sells continue on their path downwards. (b) represents the total dollars bought and sold in lit and dark venues. Generally they are correlated but it is a matter of further research to measure their indiviudal and joint contributions to cost.



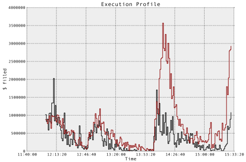
(a)



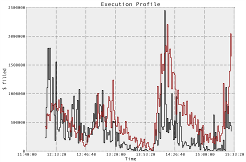
(b)



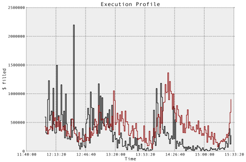
(c)



(d)



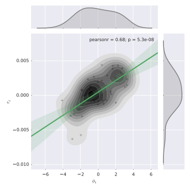
(e)

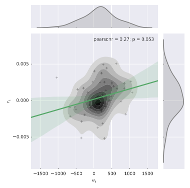


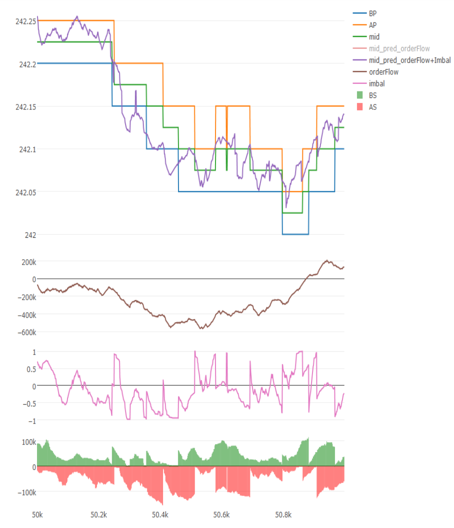
(f)

1. | Microstructure review

1. Model is used for simulating market response due to both passive and aggressive orders.
2. Midpoint price reconstruction through order/trade flows provide indicative fair value.
3. Changes in book prices reflect tick/spread constraints.







VWAP slippage minimisation

LOPM

Scheduler

VWAP slippage = $\boxed{\text{Spread slippage + Timing Slippage}}$ + Profile Slippage

- aggressive ratio
- how deep do you layer the book
- spread capture
- ~ deterministic (linked to trading rate)

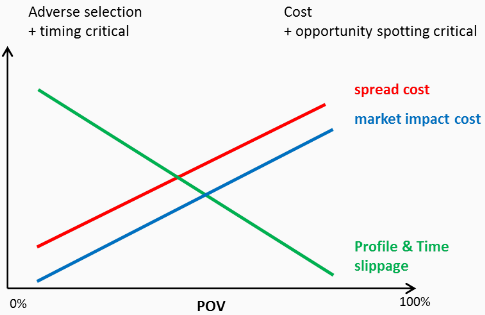
- adverse selection protection
- short term opportunities
- alpha

- volume traded in line with market

$$VWAPslippage = \sum_{j=1}^{m} \mu_j (Me_j - Pe_j) + \sum_{j=1}^{m} (\mu_j (\frac{\sum_{k=1}^{l_j} Pm_k^j \nu_k^j}{\sum_{k=1}^{l_j} \nu_k^j} - Me_j) + \sum_{j=1}^{m} (\sum_{k=1}^{l_j} \nu_k^j - \mu_j) \frac{\sum_{k=1}^{l_j} Pm_k^j \nu_k^j}{\sum_{k=1}^{l_j} \nu_k^j})) \qquad (3)$$

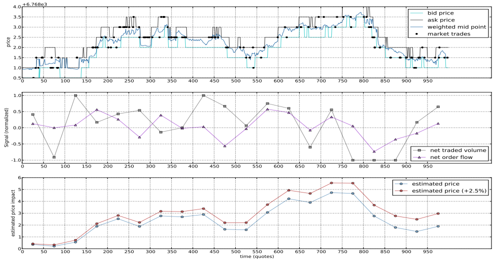
VWAP slippage minimisation

- different trading style required depending on rate of trading targeted



Optimal execution is path-dependent

1. In this simulation, the trading algorithm is measuring the impact of its activities jointly with market activity, considering both passive and aggressive placements.
2. Top series indicates price time series and market trades; middle chart charts net traded volume (due to executions) and net order flow (due to quote changes).
3. Impact is a path-dependent process - it depends on individual transactions considering market state at the time of execution, and how latent terms propagate over volume-time.
4. The optimisation problem is to minimise the difference between the series in the bottom chart.
5. In this context, rewards that can be attributed to individual transactions are only known at the end of the order - i.e. rewards are *delayed*.



Section 2: Machine learning in forecasting

1. Review: Understanding model performance through bias-variance decomposition
2. Forecasting: Random Forests for high-frequency price direction prediction
3. Performance measures of classifiers
4. Trading with signals

Bias-variance decomposition and overfitting

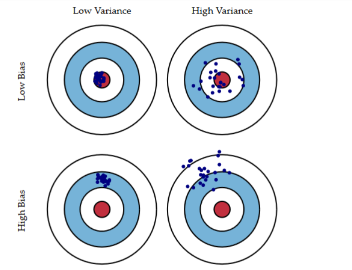
- **Bias** reflects the error between the forecast, $\hat{f}(x)$ and observed, target variable $f(x)$ - how far is the prediction from reality?
- **Variance** is the variability of the model for a given data point - how does the prediction vary over examples?
- **Noise** is the irridducible error.
- The objective of most machine learning algorithms is to minimise loss with respect to some penalty function subject to a balance between bias and variance.

Theorem. For the squared error loss, the bias-variance decomposition of the expected generalization error at $X = \mathbf{x}$ is

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Noise} \tag{4}$$

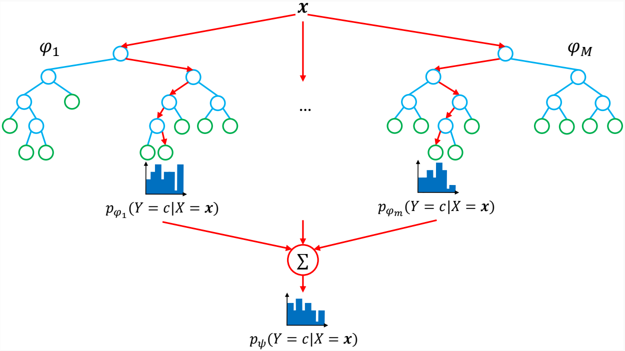
where

$$\begin{aligned} Bias^2 &: \left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2 \\ Variance &: \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\right)^2\right] \\ Noise &: \sigma_e^2 \end{aligned}$$



Random Forests

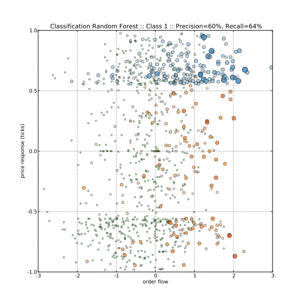
- **Error:** (Residual error: Lowest achievable error, independent of $\varphi_{\mathcal{L}}$.)
- **Bias:** Decision trees usually have low bias.
- **Variance:** They often suffer from high variance - i.e. don't generalise well due to overfitting.



2. | ML Forecasting - Random Forest Classifiers

Directional classifiers are designed to predict large positive price moves, large negative price moves and flat price moves. The correctly classified instances for large positive price moves are shown in blue. The instances that were incorrectly classified as large positive price moves are shown in red (so we predicted the price would move up, but it did not). The green points are the instances that we predicted as being either flat or negative price categories.

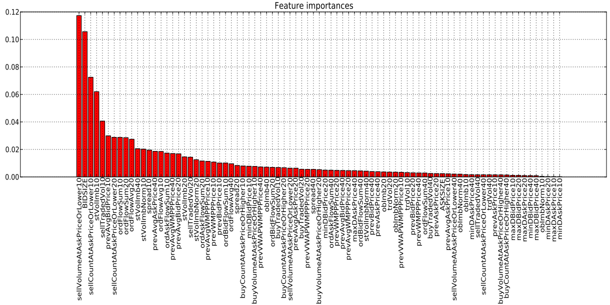
1. The x-axis is the same signal as shown in the regression plots, being limit order flow over the past 100 ticks.
2. The presence of blue instances with negative order flow indicates that other information can be used to predict price moves. The prediction probability is seen to increase with higher values of order flow.
3. The larger, darker points indicate a higher prediction probability. We are more certain of the result with a higher the prediction probability, which is additional, useful information.



Ensemble classifiers combine multiple signals, in this case around 15, that are weighted to give the best prediction power.

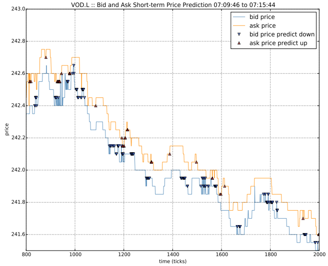
2. | ML Forecasting - Measuring Predictive Power

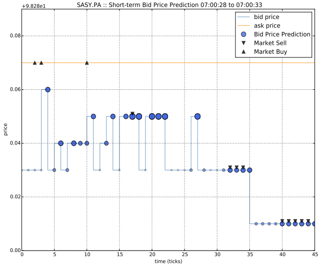
- Ensemble learning techniques are used to combine multiple, highly correlated signals into a composite *vote* that performs better than the individual voters.
- Branches on a tree can capture performant signal compositions in different regimes, such as momentum, low-spread, etc.
- Every transaction across the entire market is fed through a network of signals which are multi-tick horizon and also are accompanied with probabilities.
- During training, feature imbalances were taken into account through re-sampling of the data.
- Feature importance map for the random decision tree, by signal name.



2. | Measuring Predictive Power

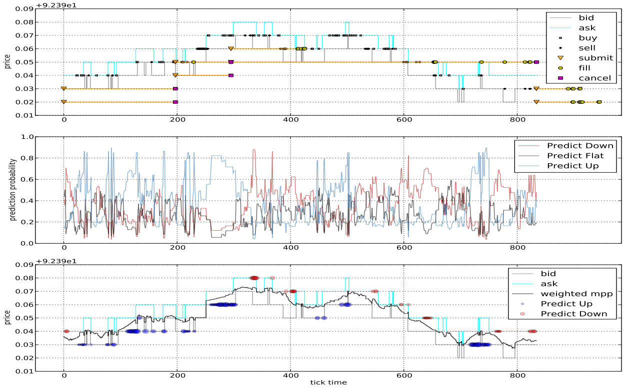
	Prediction Accuracy							
Horizon (ticks)	BAYGn.DE	CCH.L	DAIGn.DE	ISP.MI	SAPG.DE	SASY.PA	TOTF.PA	VOD.L
10	0.797	0.777	0.662	0.877	0.712	0.773	0.748	0.791
20	0.749	0.731	0.638	0.832	0.678	0.728	0.708	0.739
50	0.691	0.679	0.620	0.763	0.642	0.671	0.656	0.669
100	0.662	0.649	0.612	0.714	0.623	0.645	0.639	0.648
200	0.644	0.625	0.603	0.675	0.612	0.631	0.625	0.631





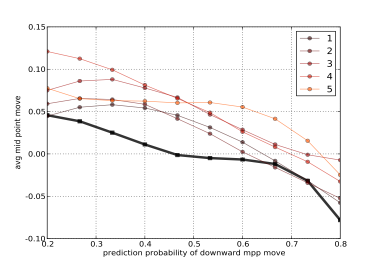
The simulator is then used to test if signal classification is useful over the time frames of the limit order submissions.

1. The upper plot shows an example of Crude Oil placement and filling using parameters similar to Production. Note the micro-structure changes over the timescales relevant to limit order placement.
2. The middle plot shows the prediction probabilities into up, down or flat price moves.
3. The lower plot shows periods when we could apply these predictions. The blue circles are periods where we predict the price to rise. The red circles are when we predict the price to fall.



Impact reduces signal effectiveness

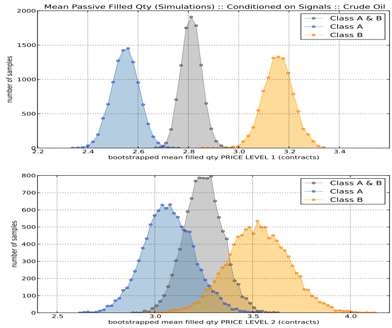
1. To test our assumptions, we superimposed signals over child order submissions and studied the execution/slippage profile.
2. The first thing we noticed is a bias in the signal distribution themselves, due to the fact that any given number of orders would survive longer if the signal was trending away (not shown).
3. This observation has important implications for simulation: behaviour bias needs to be accounted for by calibration - calibrate fill probabilities using historical transaction data.
4. The black line is the *unconditional* move predicted as a function of probability (without the systematic presence of our child orders).
5. The lines above that represent order types (1-5) in terms of urgency; the difference between these *conditional* observations reflect the impact of child orders.
6. Clearly it is important to consider the joint effect of candidate child orders in optimal order placement, insofar as how they effect signals the algorithm uses.
7. The next section discusses how signals, impact on signals (impact), and opportunity costs are combined into an optimisation framework.



2. | Order Queue Dynamics

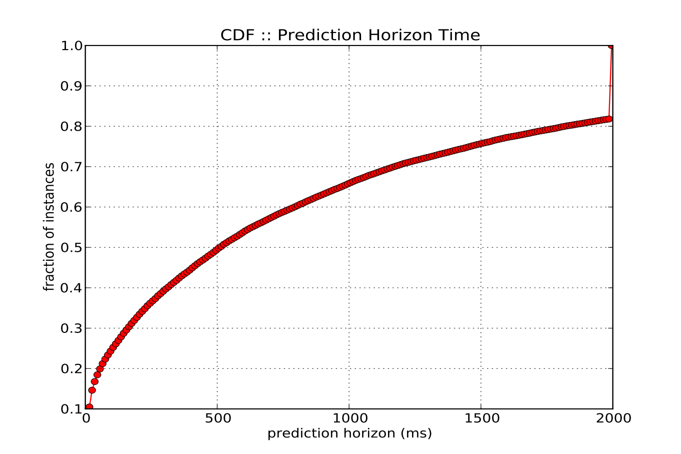
The plots below show the distribution of the bootstrapped mean for passively filled quantity. The upper plots show the filled quantity on the first price level (so near touch), the lower plots the secondary price level (so one tick away from the near touch). Each distribution contains 10,000 bootstrap samples.

1. The grey distribution is all samples.
2. The blue and orange distributions (named class A and B respectively) are the same points classified ex-post, using the prediction classification used in the previous slides. Class A is when we predict the price to remain flat or move away, Class B is when we predict the price to move in our favour.
3. There is a clear indication that we can predict a higher rate of passive filling based on the signals.



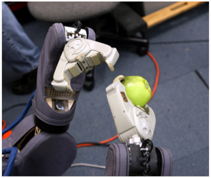
2. | Alpha Decay & System Latencies

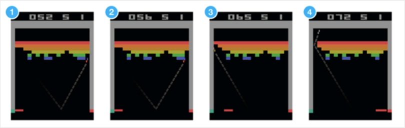
Each signal is characterized by a CDF that provides the fraction of observations that fall under a given prediction horizon. This can be used to tune/understand the effectiveness of the system overall as latencies are obtained for each leg of the process, from signal origination to execution. Furthermore, in real-time - each signal carries this information so that stale conditions are handled properly.



Deep learning and reinforcement learning used in various fields


Autonomous vehicle


Robotics


Playing games

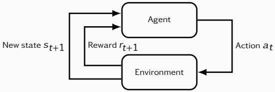
So how do we put the pieces together? How do we combine signals, market impact, order placement, scheduling, simulation to create a coherent strategy that follows an optimal solution hierarchically from scheduling to order placement? The aim is to choose optimal prices, sizes and durations for all orders based on both the prevailing and projected market conditions. The solution outlined here solves for these issues jointly as part of a stochastic optimal control problem.

The learned value function, $Q(s,a) = \vec{\theta}_t^T \vec{\phi}_{s,a}$ is parametrised by $\vec{\theta}_t$ and the feature vector $\vec{\phi}_{s,a}$ at time t.

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha(r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t))\vec{\phi} \quad (5)$$

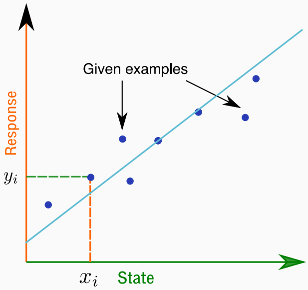
where

$\vec{\theta}_{t+1}$	:	updated parameter vector
$\vec{\theta}_t$	:	parameter vector at time t
$\vec{\phi}$	:	feature vector
s	:	is the current state
a	:	the action space
r_{t+1}	:	reward for action a_t in state s_t
$Q_t(s_t, a_t)$	:	value function

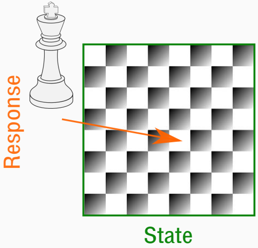


Supervised Learning: learning by example

Regression



Chess game



For a chess game:

x_i = state of the board at time i.

$y_i = f(x_i)$ optimal move when in state x_i.

But how can we obtain samples $y_i = f(x_i)$?

Reinforcement Learning: value function

The value function V is the memory/brain of the agent:

Instead of learning directly the optimal action we learn the value of a pair (x_i, a_i).

V maps every state/action to a value: $x_i, a_i \rightarrow v_i \in \mathbb{R}$.

The value of (x_i, a_i) is defined as the sum of expected future rewards when starting from x_i and taking first action a_i.

$$V(x_i, a_i) = \mathbb{E}\left[\sum_{t=i}^{T} R_t | a_i\right]$$

Every time we visit a state, take on action and receive a reward we update the function V:

1. Record $(x_i \rightarrow a_i \rightarrow r_i \rightarrow x_{i+1})$.
2. Update V

The central question: how do we perform the update?

Reinforcement Learning: value function

After having observed (x_i, a_i, r_i, x_{i+1}) we obtain:

$$V(x_i, a_i) = \mathbb{E}\left[\sum_{t=i}^{T} R_t | a_i\right] = r_i + \mathbb{E}\left[\sum_{t=i+1}^{T} R_t\right] =: y_i$$

y_i defined above is our sample!

But what about the expectation $\mathbb{E}$ part in y_i?

Start with a random initialisation for the mapping V and improve gradually.

Definition (State-value function)

The function

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s\right\} \tag{6}$$

is called the state-value function for policy π. E_{π} =expected value assuming the agent follows the policy π with an arbitrary time step t.

Definition (Optimal state-value function)

The function

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \forall s \in \mathcal{S} \tag{7}$$

is called optimal state-value function.

Definition (Action-value function)

The function

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a \right\} \quad (8)$$

is called the action-value function for policy π.

Definition (Optimal action-value function)

The function

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \forall s \in \mathcal{S} \ and \ a \in \mathcal{A}(s) \quad (9)$$

is called optimal action-value function.

3. | The Q-Learning Algorithm

Algorithm 1 Q-Learning

1. Initialisation:
    Load a simulation environment: price series, fill probability;
    Initialise the value function V_0 and set the parameters: α, ϵ;
2. Optimisation:
for $episode = 1, 2, 3\ldots$ **do**
    for $t = 1, 2, 3\ldots T$ **do**
        Observe current state s_t;
        Take an action $a_t(Q_t, s_t, \epsilon)$;
        Observe new state s_{t+1};
        Receive reward $r_t(s_t, a_t, s_{t+1})$;
        Update value function using r_t and current estimate Q_t:
            a) compute $y_t = r_t + \max_a Q_t(s_{t+1}, a)$
            b) update the function Q_t with target y_t
    end for
end for

Problem setup

Objective : VWAP minimisation

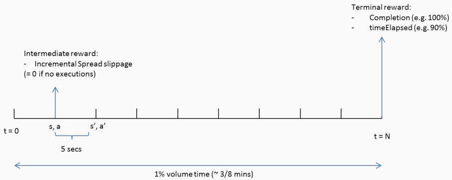
Optimising on deterministic part (spread cost) which will mainly affect mean of slippage distribution. Profile slippage is reduced by ensuring orders are spread enough throughout the episode (e.g. risk of being filled too quickly entirely passively for small quantities). **Rewards** are made of 2 parts:

Immediate rewards (for each iteration):

- increment of spread cost (on executions we had since last iteration)

Terminal rewards (end of episode):

- completion penalty
- order duration penalty
- market impact penalty



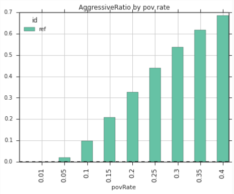
State space

States which are common to every placement:

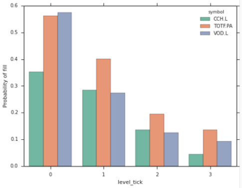
- target historical pov rate
- %progress (= %timeElapsed - %inventoryTraded)
- %timeElapsed
- alpha profile

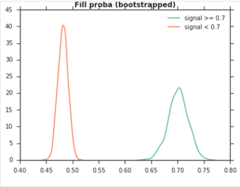
For each placement/action evaluated, we have those specific states:

- expected spread cost
- fill probability
- size placed (as % of episode quantity)



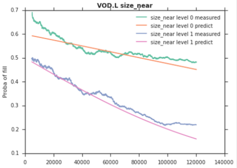
Fill probabilities

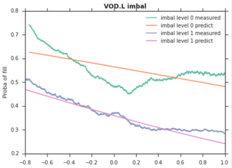




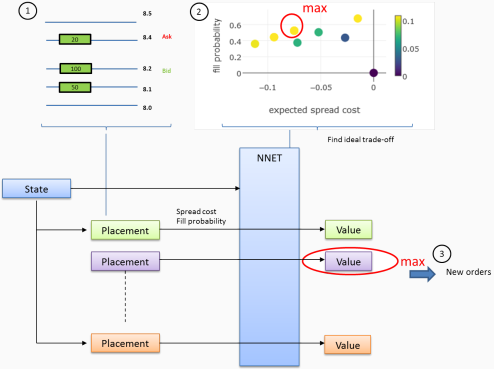
Probability of being filled :

- Can be measured empirically looking at our historical LOPM placement
- Can be conditioned on:
    - Ticker
    - Level we place at in the book
    - Signals
- Instantaneous signals (Order book imbalance, near touch size (proxy for queue position) seem to have the most explanatory power
- Combining signals can be done using machine learning techniques (e.g. logistic regression, random forest)





Placement evaluations

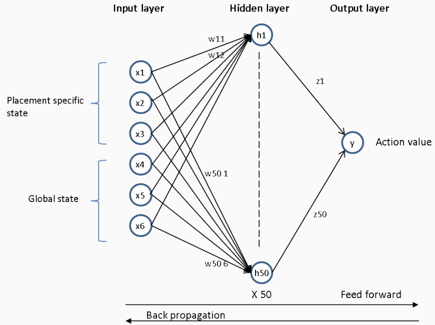


Neural network architecture

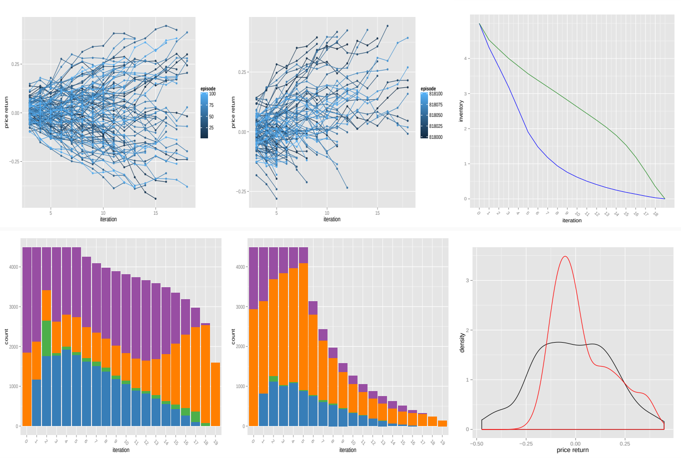
MLP (Multi layer perceptron)

$$h = g(W_1 x + b_1)$$
$$y = W_2 h + b_2$$

x : input vector (dim 6)
W : weight matrix (dim 50 x 6)
b1: bias vector (dim 50)
Z : weight vector (dim 50)
b2 : bias scalar
g : activation function (ReLu)

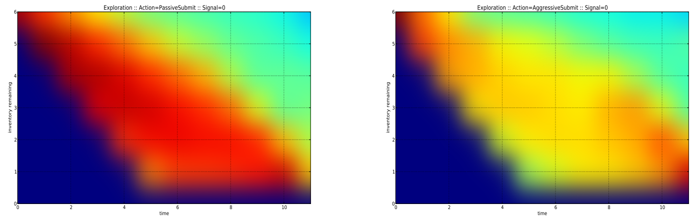


Learning price momentum - arrival objective



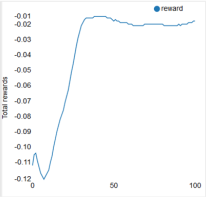
Learning fill probabilities

A surface plot state space explored for either (a) Passive or (b) Aggressive for two variables, time (horizontal axis) and inventory (vertical axis). Red implies that this choice is preferred, showing a preference to trade passively at the beginning of an order and then aggress near the end. There is an additional aggressive order at the beginning of this particular scenario, which is a result of the overall low probability of passively filling the entire order. The dark blue region in the lower left is not explored, which greatly reduces the state space exploration. The upper right is also only weakly explored, due to the optimal path of filling throughout the episode. The surface plots were generated from 20,000 simulations.

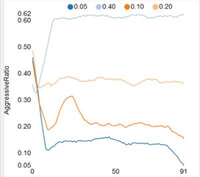


Convergence in training

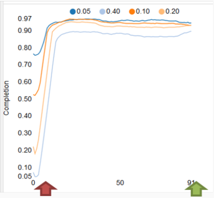
Total reward



Aggressive ratio



Completion rate

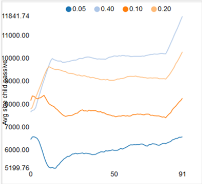


No knowledge

Convergence

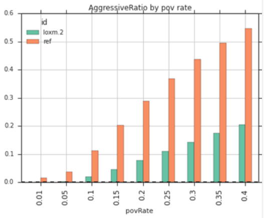
Number of simulations performed

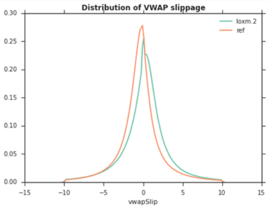
Sizes of order placed

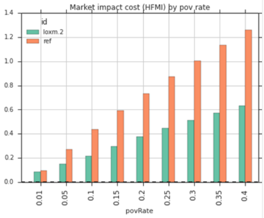


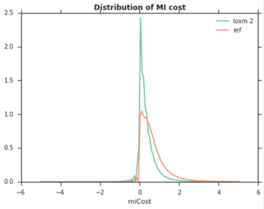
Number of simulations performed

Backtest statistics









Evolution of trading

Manual trading → Algo trading (human programmed) → Algo trading (computer programmed)

Machine Learning approach:

- avoids hand crafting of logic / manual overfitting
- allows for strategies to be optimised around characteristics of individual stocks
- but this is not just about data and compute : domain expertise critical for applications in finance

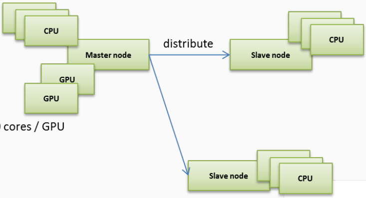
3. | INFRASTRUCTURE

Hardware

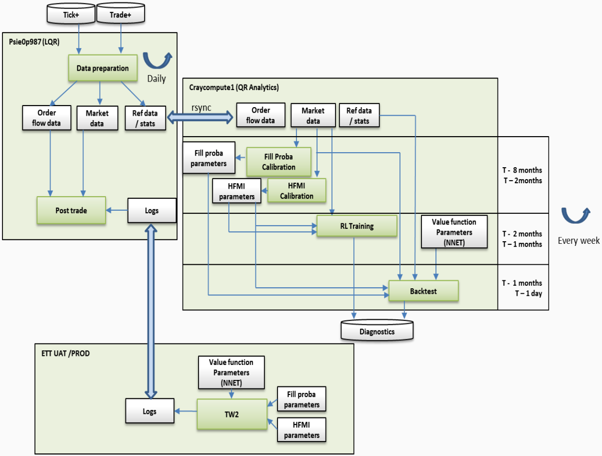
- 640 CPU cores
- 4 x 2880 GPU cores
- 2.5 TB RAM

Training time

- 16 hours for 700 tickers



3. | ARCHITECTURE: PRODUCTION



Final remarks - Multi-agent systems

Further items to research and consider

In addition to being able to train a single agent operating in a system of noisy and informed agents - we are also studying the interaction of multi-agents in systems specifically crafted to encourage co-evolution: how do agents co-adapt by leveraging their respective advantages?

- **Auctions** - single-auction price setting, sequential price setting scenarios.
- **Reservoir discovery** - Micro-strategies can take advantage of temporary conditions by *latching-onto* markers, for example to capture liquidity under a price limit, then resume normal operation when the opportunity disappears.
- **Limit-aware tactics** - Parent limit orders are private information; implicitly changing behaviour at or near the limit is a signal: account for constraints during decision making.
- **Arbitrageurs** - Train agents for specialized interaction where optimal actions co-evolve.
- **Counterpary Simulations** - Stress-test conditions and examine situations where counterparty diversity narrows to a particular set (trading against faster opponents, for example).
- **Herding** - Study behaviour when *correlated motives* result in coordinated behaviour, as compared to *correlated information*.
- **Theoretical Simulations** - Similar to the case in Game Theory where the optimal strategy is conditioned on the player's skill in identifying the game she is in fact playing.

Disclaimer

This material (Material) is written by the Electronic Client Solutions Quantitative Solutions Team for distribution to you, our J.P. Morgans client, only. It may not be forwarded to any party without obtaining the express written consent of J.P. Morgan. This Material is not available for distribution to persons regarded as private customers (or equivalent) in their home jurisdiction. It is not a product of J.P. Morgans research department. Opinions expressed herein are solely those of the individual author and may differ from the views or opinions expressed by other areas of J.P. Morgan, including the research department and its affiliates. The Material is provided for your information only and does not constitute: (i) research or a product of the J.P. Morgan research department, (ii) an offer to sell, a solicitation of an offer to buy, or a recommendation for any investment product or strategy, or (iii) any investment, legal or tax advice. You are solely responsible for deciding whether any investment product or strategy is appropriate for you based upon your investment goals, financial situation and tolerance for risk.

J.P. Morgan disclaims all representations and warranties in the information contained in this Material, whether express or implied, including, without limitation, any warranty of satisfactory quality, completeness, accuracy, fitness for a particular purpose or non-infringement. The information contained herein is as of the date and time referenced in the Material and J.P. Morgan does not undertake any obligation to update such information. All content, data, statements and other information are not warranted as to completeness or accuracy and are subject to change without notice. J.P. Morgan disclaims any responsibility or liability, whether in contract, tort (including, without limitation, negligence), equity or otherwise, for the quality, accuracy or completeness of the information contained in this Material, and for any reliance on, or uses to which, this Material, is put, and you are solely responsible for any use to which you put such information. Without limiting any of the foregoing, to the fullest extent permitted by applicable law, in no event shall J.P. Morgan have any liability for any special, punitive, indirect, or consequential damages (including lost profits or lost opportunity), in connection with the information contained in this Material, even if notified of the possibility of such damages. J.P. Morgan and its affiliates may have positions (long or short), effect transactions or make markets in securities or financial instruments mentioned herein (or options with respect thereto), or provide advice or loans to or participate in the underwriting or restructuring of the obligations of, issuers mentioned herein. This Material is proprietary and confidential to J.P. Morgan. Any comments or statements made herein do not necessarily reflect those of J.P. Morgan, its subsidiaries or its affiliates. Clients should contact their salespersons at, and execute transactions through, a J.P. Morgan entity qualified in their home jurisdiction unless governing law permits otherwise. Any unauthorized use, dissemination, distribution or copying of this Material, in whole or in part, is strictly prohibited.

 J.P. Morgan is the global brand name for JPMorgan Chase & Co. and its subsidiaries and affiliates worldwide. J.P. Morgan Cazenove is a marketing name for the U.K. investment banking businesses and EMEA cash equities and equity research businesses of JPMorgan Chase & Co. and its subsidiaries, conducted primarily through J.P. Morgan Securities plc. Execution services are offered through J.P. Morgan Securities LLC (member of FINRA, NYSE and SIPC), J.P. Morgan Securities plc (a member of the London Stock Exchange and authorized by the Prudential Regulation Authority and regulated by the Financial Conduct Authority and the Prudential Regulation Authority. Registered in England & Wales No. 2711006. Registered Office: 25 Bank Street, Canary Wharf, London E14 5JP), J.P. Morgan Securities (Asia Pacific) Limited (CE number AAJ321) (regulated by the Hong Kong Monetary Authority and the Securities and Futures Commission in Hong Kong), J.P. Morgan Securities Australia Limited (authorised and regulated by the Australian Securities & Investments Commission), J.P. Morgan Securities Japan Co., Ltd. (authorised and regulated by the Financial Services Agency and the Japan Securities Dealers Association) and other investment banking affiliates and subsidiaries of J.P. Morgan in other jurisdictions worldwide and registered with local authorities as appropriate. Please consult http://www.jpmorgan.com/pages/jpmorgan/investbk/global for more information. Product names, company names and logos mentioned herein are trademarks or registered trademarks of their respective owners.

To the extent that you are a U.S. Institutional Investor or a "Major U.S. Institutional Investor" (collectively, "U.S.-Based Clients") as those terms are defined in Rule 15a-6 ("Rule 15a-6") of the U.S. Securities Exchange Act of 1934 (as amended from time to time) you acknowledge that where a non-US affiliate of J.P. Morgan enters into a transaction with you, you acknowledge and understand that J.P. Morgan performs such brokerage services with the U.S.- Based Clients pursuant to the terms of Rule 15a-6. Any transactions with U.S.-Based Clients are intermediated in accordance with the terms of Rule 15a-6 by J.P. Morgans U.S registered broker-dealer affiliate, J.P. Morgan Securities LLC.