Price Impact Models and Applications

Introduction to Algorithmic Trading

Kevin Webster

Spring 2023

Columbia University

Plan

Last Week
What is price impact? Overview of finance applications and glossary of trading terms.

For this Week
What Mathematics to expect from the course:

(a) Stochastic Control

(b) Causal Inference

for algorithmic trading. We'll conclude with a brief overview of the trading data.

Next Week
A primer on the database kdb+ and the programming language q.

Last Week's Summary

Price impact captures price moves caused by trading.
Price impact introduces feedback loops both in simulation and production. Furthermore, impact primarily drives trading costs.

Alpha signals predict price moves independent of trading.
Directionality, prediction horizon, trigger frequency, exogeneity, and cross-sectionality are core signal characteristics.

Trades capture alpha and pay impact.
Therefore, alpha increases trading speed, and price impact decreases trading speed.

Model-driven trading algorithms
Model alpha and impact and quantify their trade-off to submit orders on the market.

Refresher: What is Price Impact?

A causal model for trading

Trading of stock cause price moves for the stock that otherwise would not have happened.

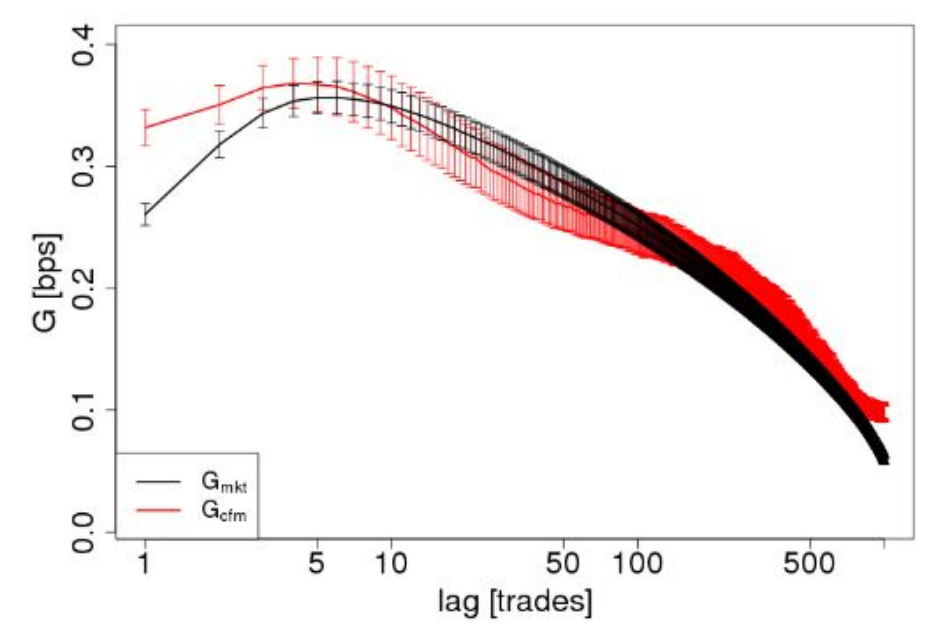


Figure 1: Average return profile after a single public (black) or Capital Fund Management (CFM, red) trade fill (Toth, CFM 2018).

Stochastic Control

What is Stochastic Control?

Stochastic control extends stochastic calculus
Ingredients of a control problem:

(a) A filtered probability space $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{P}\right)$

(b) A variable x you *control*

(c) A *state variable* X that x affects, e.g.,

$$dX_t = \mu(X_t, x_t)dt + \sigma(X_t, x_t)dW_t$$

(d) An *objective function* to maximize, e.g.,

$$\sup_x \mathbb{E}\left[\int_0^T g(X_t, x_t)dt + G(X_T)\right]$$

(e) Constraints, e.g., $\forall t > 0, X_t \geq 0$

A Standard Finance Example*

Merton's portfolio optimization problem

(a) A manager controls their portfolio allocation π_t.

(b) Their P&L Y is a state variable satisfying

$$dY_t = \mu \pi_t Y_t dt + \sigma \pi_t Y_t dW_t$$

where μ, σ are the stock's price drift and volatility.

(c) The manager maximizes their expected utility function U

$$\sup_{\pi} \mathbb{E}\left[U(Y_T)\right].$$

A Trading Example: Statistical Arbitrage (1/2)

Statistical arbitrage problem

(a) A trader controls their trading speed Q'_t.

(b) Their price impact I is a state variable satisfying

$$dI_t = -\beta I_t dt + \lambda Q'_t dt.$$

(c) The trader maximizes their expected profits

$$\sup_{Q'} \mathbb{E}\left[\int_0^T (\alpha_t - I_t) Q'_t dt\right]$$

where α_t is the trader's alpha signal at time t.

A Trading Example: Statistical Arbitrage (2/2)*

Statistical arbitrage problem
Using calculus, one shows that, if $\alpha \in C^2$,

$$Q_t' = \frac{\beta}{2\lambda}\left(\alpha_t - \beta^{-2}\alpha_t''\right).$$

A systematic algorithm implements trades through a straightforward formula with

(a) market liquidity parameters β, λ,

(b) an alpha level $\alpha_t = \mathbb{E}\left[S_T - S_t \middle| \mathcal{F}_t\right]$, and

(c) an alpha convexity α_t''.

At this point, trading simplifies to generic estimation problems which can be solved using traditional econometrics or modern machine learning.

Optional Reading (Refresher)

In increasing order of difficulty

(a) Almgren and Chriss (2001, link) first articulate algorithmic trading as a live optimization problem.

(b) Obizhaeva and Wang (2005, link) propose a price impact model with static parameters and solve the optimal trading strategy.

(c) Garleanu and Pedersen (2016, link) study implications for portfolio construction.

(d) Muhle-Karbe, Wang, and Webster (2022, link) analyze the extended model of Fruth, Schoeneborn, and Urusov (2019, link).

(e) Abi Jaber and Neuman (2022, link) solve the linear case with general decay kernel.

How does Stochastic Control Extend Stochastic Calculus? (1/2)

Because one solves an optimization problem
it is important to distinguish variables that one

(1) directly controls: the trader chooses their control variable Q'_t.

(2) indirectly affects: the trading speed changes the state variable I_t.

(3) does not affect: the exogenous variable α_t does not change when Q' changes.

Other than that, the machinery is the same: use Itô's formula.

How does Stochastic Control Extend Stochastic Calculus? (2/2)

A reference book (link)
Continuous-time Stochastic Control and Optimization with Financial Applications, Pham (2009)

Four types of solutions ranging from the most straightforward to the more involved.

(1) Pointwise (myopic) optimization

(2) Dynamic programming and the Hamilton-Jacobi-Bellman (HJB) PDE

(3) The Pontryagin maximum principle and Backward Stochastic Differential Equations (BSDEs)

(4) Reinforcement Learning (RL)

This class focuses on (1), but one extends the trading models for (2)-(4).

Why Stochastic Control Problems?

Statistical arbitrage problems
automate trading decisions considering

(a) signals, such as alpha and liquidity signals

(b) trading costs, such as price impact and spread

(c) trading constraints, such as short-selling and risk constraints

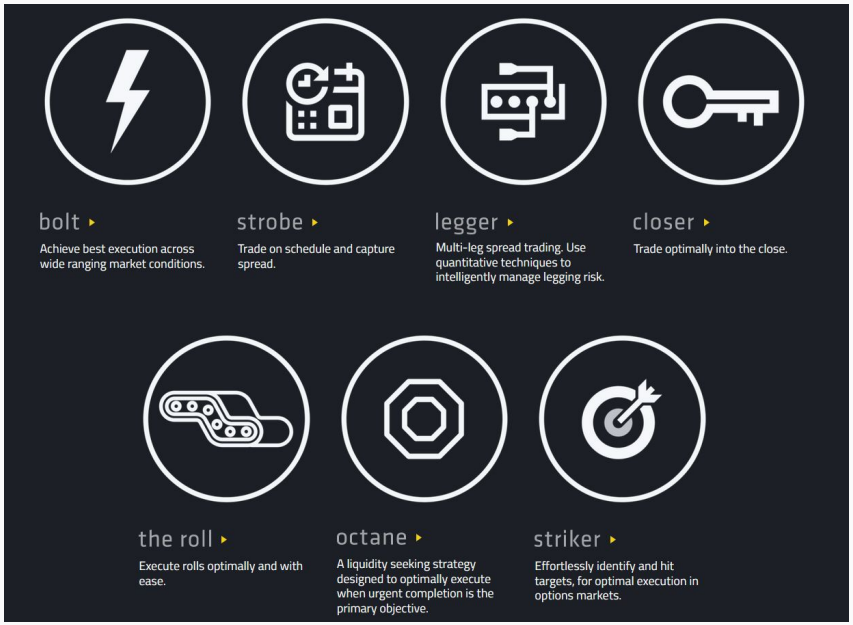
Automating trading decisions
allows one to reproduce, monitor, and back test trading strategies. These actions are essential to evaluating systematic trading.

"Cowboy" quants are not as common anymore

If the wider team cannot systematically reproduce your trades, your strategy is unlikely to get deployed.

Example: quantitativebrokers.com/listed-futures

QB's suite of trading algorithms for futures



(a) The strategies likely use the same alpha and impact models.

(b) The algorithms differ in their objective function. E.g., *closer* optimizes against a close benchmark, *legger* optimizes for pairs trading.

Practical Considerations

Trading happens in continuous time but is discrete.
Implementations in both simulation and production discretize the control models to run them along actual historical or live trading data.

Why not use discrete methods only?
Discrete methods are feasible, and one can solve them via brute-force gradient descent.

(*) Advantages: any model can be solved for, makes less assumptions on the data.

(*) Disadvantages: requires a super-computer, and a black-box approach hampers sensitivity analysis.

Especially for liquid stocks or sizable orders, the tractability of continuous time models allows for more sophistication in other aspects (e.g., using machine learning for alpha signals).

Causal Inference

What is Causal Inference?

Causal inference extends Bayesian statistics
Causal inference distinguishes between *observations* and *interventions.*

(*) If I observe X, what is the expected Y?

$$\mathbb{E}[Y|X]$$

(*) If I change X, what is the expected Y?

$$\mathbb{E}[Y|\mathrm{do}(X)]$$

Causal inference requires an additional ingredient to $(\Omega, \mathcal{F}, \mathbb{P})$, a causal structure $\mathcal{G}$, to estimate interventional statements.

Why Causal Inference?

Causal Inference has applications to trading
AB testing and live trading experiments have emerged as a crucial tool to complement backtests. For example,

(a) to estimate algorithms' performance in "algo-wheels" and Transaction Cost Analysis (TCA).

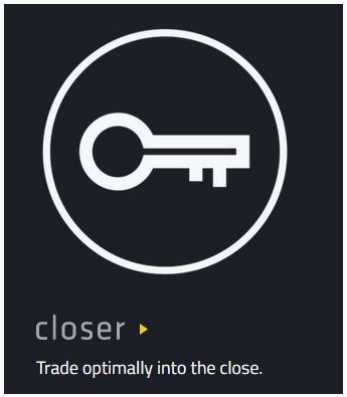
(b) to disentangle price impact from alpha.

(c) to remove statistical artifacts and biases due to market microstructure.

More finance applications
Lopez de Prado (2022, ADIA, link) outlines further applications of causal inference to finance: e.g., macro factor research, stress testing. This class focuses on trading applications.

Example: quantitativebrokers.com/whitepapers/2022-03a

A/B testing of New Closer in Production
Recall QB's *closer* trading algorithm optimizing for the close.



QB upgraded it in late 2020 and reported its live trading performance ("production") in 2022. Their report uses randomize A/B testing to convince clients of the upgrade's merits.

A Standard Example: The Trading Algo Wheel*

An algo-wheel setup

(a) An order of size S is submitted upstream.

(b) A trading algorithm A from a set $\{A_1, ..., A_k\}$ of algorithms is chosen based on the order size S.

(c) The algorithm A affects the order's trading speed $\bar{S}$.

(d) Y measures the order's *arrival slippage*.

Simpson's paradox

A	S	Sample size	$E[Y\|A,S]$
a	s	40k	-10bps
a	l	10k	-40bps
p	s	10k	-5bps
p	l	40k	-25bps

Figure 2: Expected arrival slippages across order sets.

Discrete case

Assume that A and S take binary values in $\{a, p\}$ and $\{s, l\}$ respectively. a stands for aggressive, p for passive, s for small and l for large.

The aggressive algorithm was allocated smaller orders.

Simpson's paradox (1/3)

A	S	Sample size	$E[Y\|A,S]$
a	s	40k	-10bps
a	l	10k	-40bps
p	s	10k	-5bps
p	l	40k	-25bps

The aggressive algorithm is better across the full data

$$\begin{aligned}\mathbb{E}[Y|A=a] &= (-10)\cdot(0.8)+(-40)\cdot(0.2)\\ &= -16\end{aligned}$$

$$\begin{aligned}\mathbb{E}[Y|A=p] &= (-5)\cdot(0.2)+(-25)\cdot(0.8)\\ &= -21.\end{aligned}$$

Simpson's paradox (2/3)

A	S	Sample size	$E[Y\|A,S]$
a	s	40k	-10bps
a	l	10k	-40bps
p	s	10k	-5bps
p	l	40k	-25bps

The passive algorithm is better for small... and large orders

$$\mathbb{E}[Y|A=a,S=s]-\mathbb{E}[Y|A=p,S=s]=-5$$
$$\mathbb{E}[Y|A=a,S=l]-\mathbb{E}[Y|A=p,S=l]=-15.$$

Simpson's paradox (3/3)

But which formula is correct?
Bayesian statistics does not provide an answer. Intuitively, traders will tell you that one should condition by S to "compare apples to apples"

But which formula is correct?
Those same traders will flip their answer if S is replaced with $\bar{S}$!

This is because the trade size S was "forced" onto the algorithm A, but the trade speed $\bar{S}$ was a "decision" made by A.

Rule of thumb
Control by upstream decisions and avoid controlling by downstream decisions.

Recommended Reading

References on causal inference

(a) *Causality* by Pearl (2008, link): *not* a finance book, but core to causal inference.

(b) Lopez de Prado (2022, ADIA, link) outlines applications of causal inference to finance.

(c) Gitlin et al. (2022, Uber, link) summarizes the causal machine learning **infrastructure** at Uber.

(d) Netflix (link) has an expansive list of research articles on **applications** of causal inference and causal machine learning.

(e) Microsoft Research Summit 2021, Causal Machine Learning (link) leans heavier on the **statistics** side.

How does Causal Inference Extend Bayesian Statistics?

Bayesian models
describe a frozen data generation process. The statistician observes but does not intervene.

Causal models
describe data with interventions. The statistician intervenes, e.g., using randomization, to understand the system.

Causal inference extends traditional econometrics
Econometricians reason causally: e.g., they distinguish exogenous from endogenous variables. However, unlike causal inference, causal econometrics was not designed for non-parametric models.

Why Causal Inference? (1/3)

Simpson's Paradox is common in trading.
Quants face trading biases daily. Consequently, they use A/B experiments and causal inference to resolve such causal biases.

Deutsche Bank TCA research

> *"It is important to acknowledge that when we run these analyses, it's done only for a subset of orders that are affected by a controlled A/B experiment with routine changes, so we can really quantify their impact." (Sotiropoulos and Battle (2017))*

Why Causal Inference? (2/3)

Live trading interventions implement thought-experiments.
They are easier to formalize with causal inference, which allows interventions, than with Bayesian statistics, which only allows observations.

Bouchaud's anonymity assumption (2021)

> *"We have actually shown that the short term impact of CFM's trades are indistinguishable from the trades of the rest of the market, or, for that matter, from* **purely random trades that were studied at CFM during a specifically designed experimental campaign in 2010-2011**"

Why Causal Inference? (3/3)

Causal inference is compatible with machine learning.
Standard econometrics only covers linear regression. Recently, the technology industry has embraced causal machine learning as an essential tool in real-life applications.

The Microsoft Research Summit of 2021
had over a dozen talks within its causal machine learning track, one of its seven science tracks.

> *"This track focuses on emerging causal machine learning technologies and the opportunities for practical impact at the intersection of academia and industry, with contributions from researchers at Microsoft and the broader academic and industrial research communities."*

A Last Example (1/3)

Disentangling alpha from impact
Assume that

$$\Delta P = I + \alpha + \epsilon_1$$

with $I = \lambda Q$ and $\epsilon_1 \perp I, \alpha$. A trader regresses returns ΔP against trades Q

$$\Delta P = \beta Q + \epsilon_2.$$

Naively, one would expect $\lambda = \beta$ and $\epsilon_2 = \alpha + \epsilon_1$.

This is false!
Indeed, in regression, one assumes $\epsilon_2 \perp Q$. However, $\alpha \not\perp Q$:

> *An algorithm definitely doesn't trade independently of its alpha signal!*

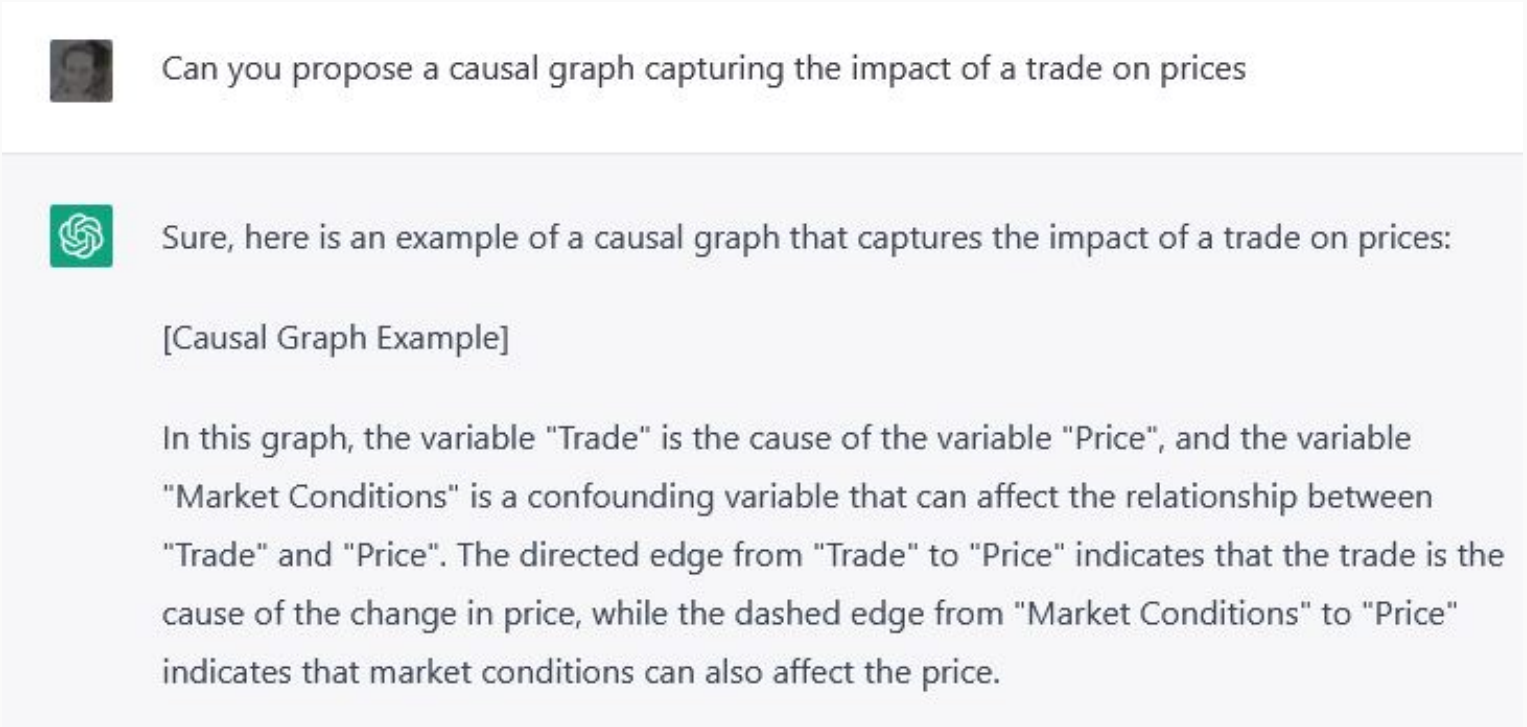
For example, one could model the relationship between Q and α:

$$Q = \gamma\alpha + \epsilon_3$$

where $\epsilon_3 \perp \alpha$, and γ reflects alpha driving trades.

A Last Example (2/3)*

Playing with ChatGPT



The answer is inconsistent (but sounds good!). More on confounding variables and causal graphs in module 3.

A Last Example (3/3)

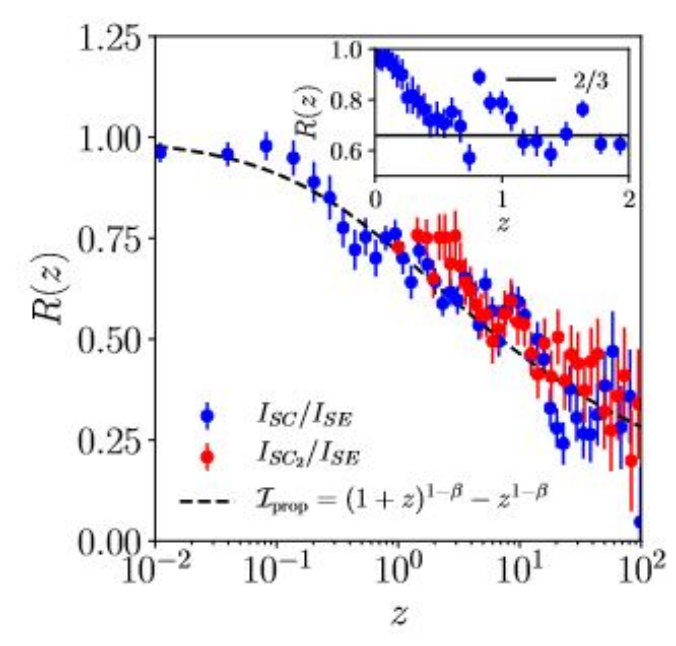


Figure 3: Multiday impact curve from Bucci et al. (2019) for large meta orders.

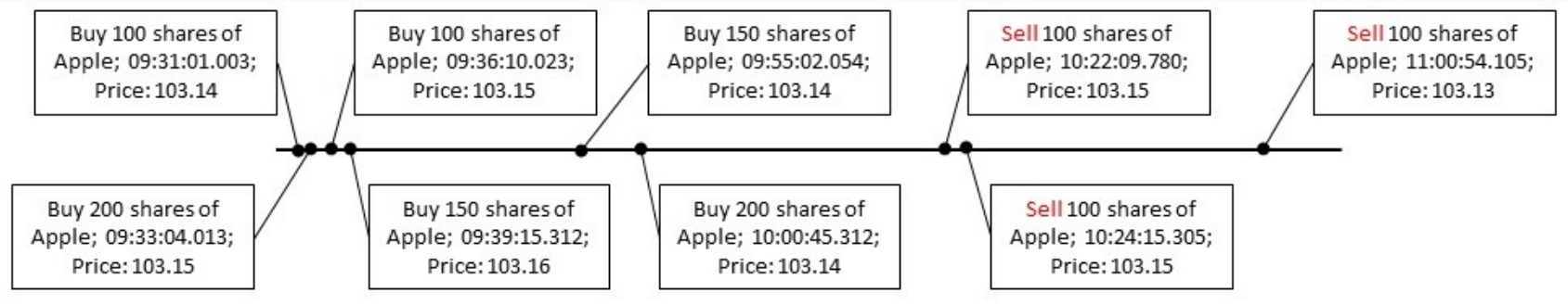
Permanent impact or long-term alpha?

Without causal inference, such question remain "philosophical" instead of being rooted in rigorous statistics.

Manipulating trading data

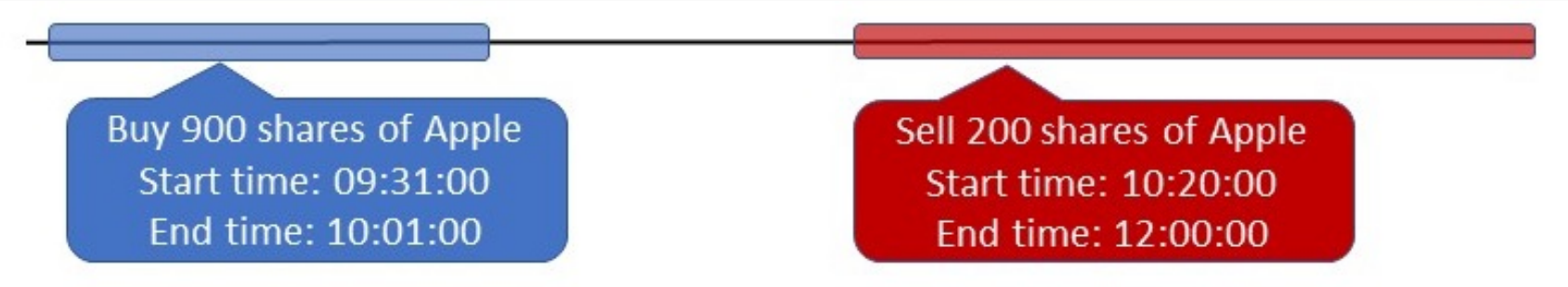
Fill Data

Individual trade fills, also called child-orders, are the most granular information and contain, at minimum, a timestamp, volume, and price. In addition, individual fills provide context specific to the trading venue that executed the fill, such as which order type triggered the fill.



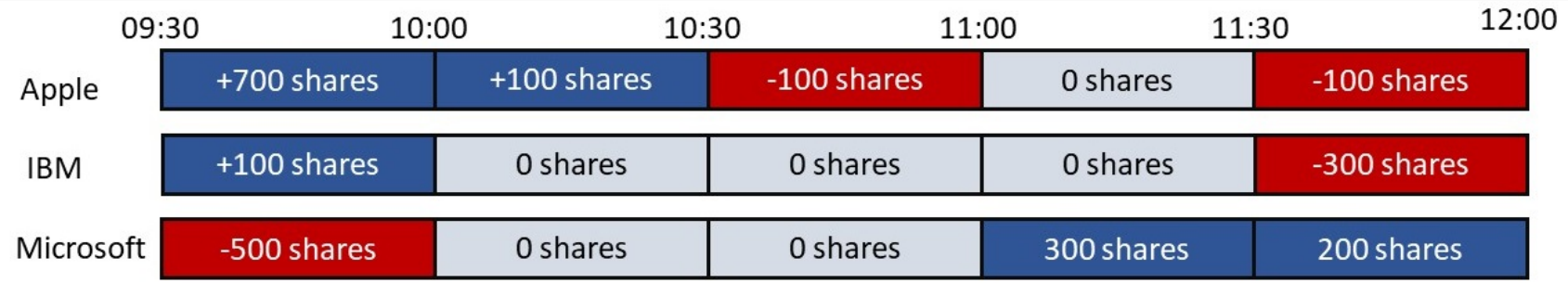
Order data

Orders aggregate individual fills into compact, more interpretable data. The cost is a loss in the granularity of the data. They are also called parent orders or metaorders.



Binned data

Traders bin data in time and stack together the time series of a specific universe, for example, the S&P 500 constituents, in a two-dimensional (stock, time) matrix. This *cross-sectional data* facilitates computing statistics across stocks at a given time.



What is Lobster?

Public trading tape on Nasdaq

(a) fill-level trading data, including if the fill was buyer- or seller-initiated.

(b) anonymous fills: the traders' identities are unknown.
Non-anonymous markets include OTC markets (e.g., Foreign Exchange) and the Toronto and Australian stock exchanges.

(c) order book information between fills: microstructure context. E.g., limit order posting and cancellation.

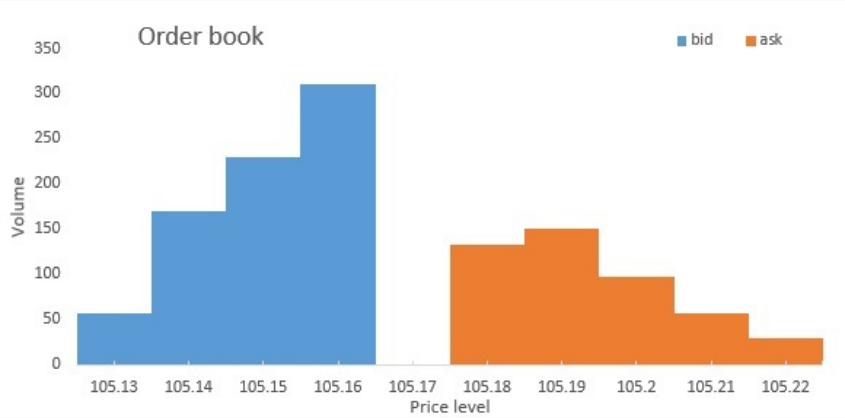
Documentation
can be found on `www.lobsterdata.com`.

What is a Limit Order Book?

Microstructure 101
The limit order book represents instantaneous supply and demand.

(a) Impatient traders, liquidity takers, buy at the ask and sell at the bid.

(b) Patient traders, liquidity providers, buy at the bid and sell at the ask.



The spread is (ask - bid)/2.
Liquidity takers pay the spread. Liquidity providers... hope to capture the spread.

Messages and Order Book on Lobster

message file.

Time (sec)	Event Type	Order ID	Size	Price	Direction
⋮	⋮	⋮	⋮	⋮	⋮
34713.685155243	1	206833312	100	118600	-1
34714.133632201	3	206833312	100	118600	-1
⋮	⋮	⋮	⋮	⋮	⋮

order book file.

Ask Price 1	Ask Size 1	Bid Price 1	Bid Size 1	Ask Price 2	Ask Size 2	Bid Price 2	Bid Size 2	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1186600	9484	118500	8800	118700	22700	118400	14930	...
1186600	9384	118500	8800	118700	22700	118400	14930	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Messages

Each row describes limit order book events, including *order book updates* and *trade fills.*

Order book

Each row describes the limit order book state before an event.

AAPL Example on 2019.02.01

Limit Order Book (lob) Events from Lobster
One classifies lob events into four categories.

(a) Trades: a liquidity taker executed a trade on the best bid or ask.

(b) Hidden trades: trade where the direction is hidden (typically at mid).

(c) lobSubmission: a liquidity provider places an order.

(d) lobCancel: a liquidity provider cancels an order.

```
q)update eventPercentage: eventCount % sum eventCount from select eventCount: count i by event from tbl lj map
event        | eventCount eventPercentage
-------------| --------------------------
hiddenTrade  | 3352       0.007179929
lobCancel    | 189093     0.4050341
lobSubmission| 229197     0.4909362
trade        | 45215      0.09684978
```

Designing a Dataset

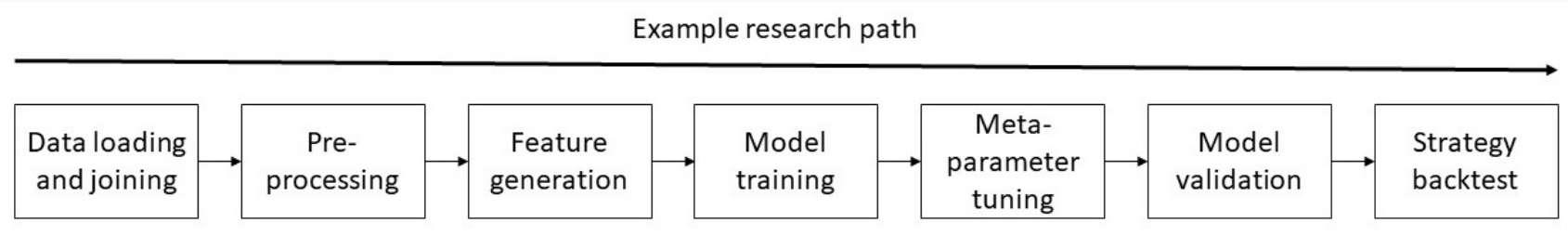
Orders of magnitude
For a given stock and day, there are $\approx 10^6$ events at the best bid or ask (first level ticks) and $\approx 10^5$ trade fills. Deeper level ticks add further orders of magnitude.

Why is data pre-processing important?
A typical investment universe includes *at least* the S&P 500 and trades 252 days a year, adding a $\approx 10^6$ multiplier to the data size.

Data of this size *does not fit in memory*. Even with GPUs or for trivial calculations, the upfront data marshaling costs are prohibitive: it helps if the data is *already in the right place and shape.*

Data best practices for modeling (optional, 1/2)

Break down data processing.
Structure your data processing into steps. Consequently, identify the step that *you* repeat most often: this is your *iteration loop.*



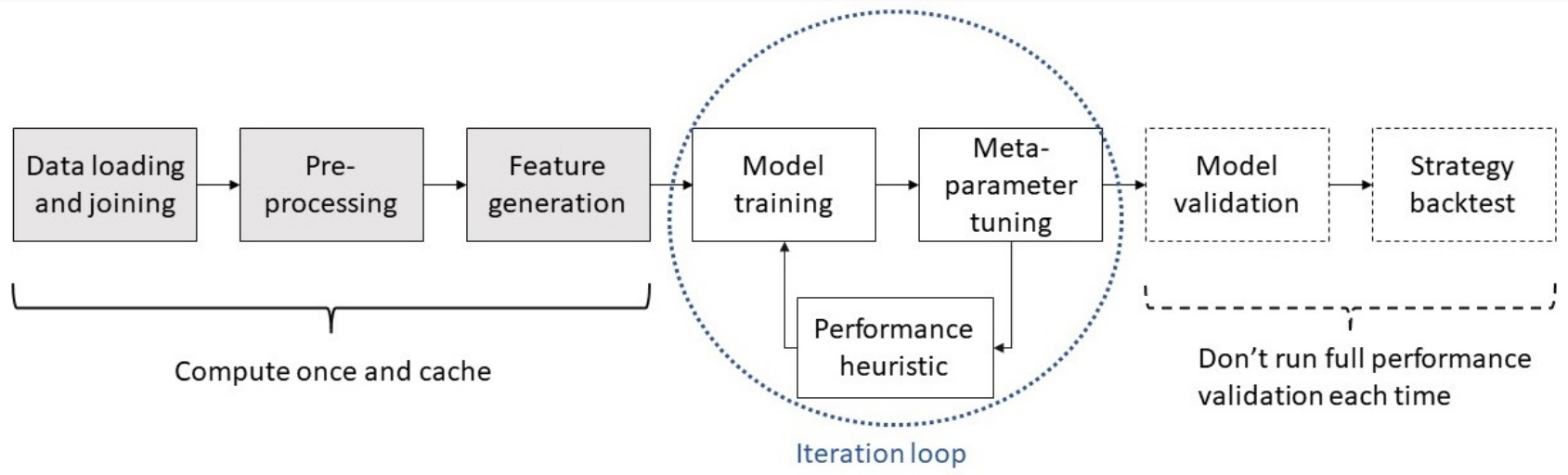
Example: feature selection
One model training approach is to pre-generate thousands of feature variants as a one time procedure. Then, using methods such as boosting, regularization, explore lower-dimensional combinations without recomputing features on the fly.

Data best practices for modeling (optional, 2/2)

Accelerate your iteration loop.

(a) Pre-compute and cache steps: don't repeat early, expensive steps.

(b) Create intermediate tests and performance heuristics: don't repeat late, expensive validation steps.



Parallelization (optional, 1/3)

Identify summary statistics.
For example, a linear regression $y \sim x$ only requires the covariance matrices $X^t X, X^t Y$.

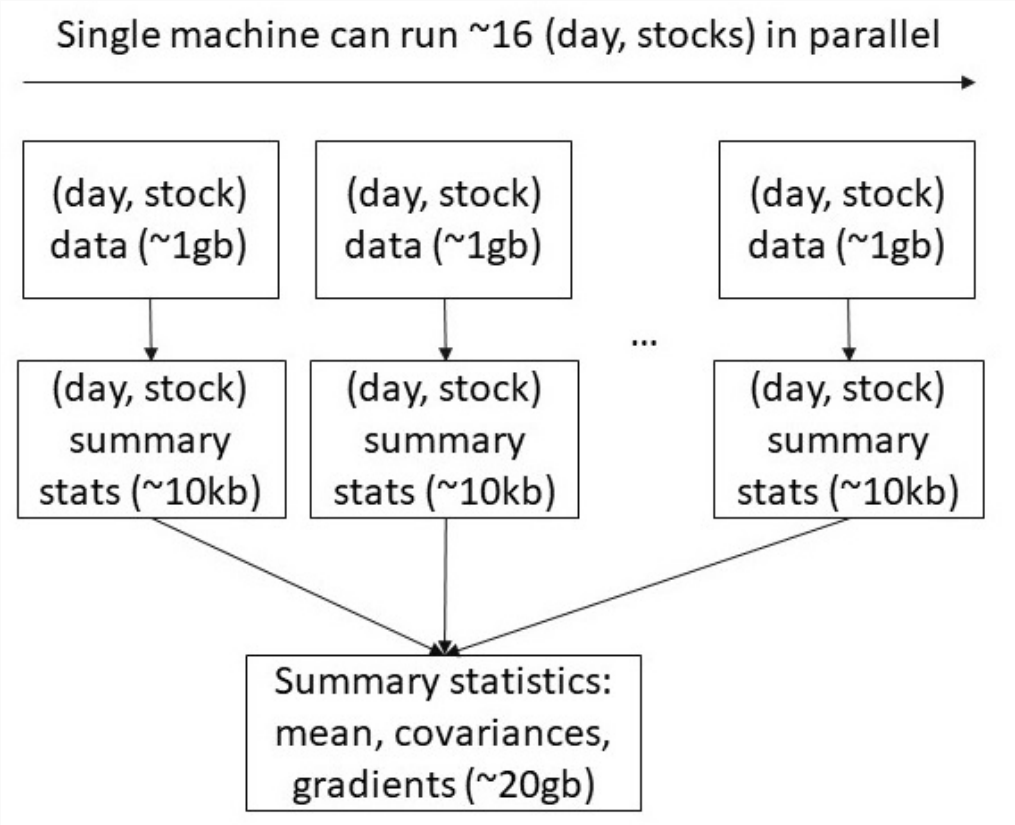
$$\mathbb{E}[\,Y|\,X] = \hat{\beta} X; \quad \hat{\beta} = (X^t X)^{-1} \cdot X^t Y$$

Don't compute the covariances $X^t X, X^t Y$ over the entire data: compute them daily, then aggregate daily covariances. For instance, in the one-dimensional case,

$$X^t X = \underbrace{\sum_{n=1}^{250 \cdot 10^5} X_n^2}_{\text{full sample}} = \sum_{d=1}^{250} \underbrace{\sum_{k=1}^{10^5} X_n^2}_{\text{daily}}$$

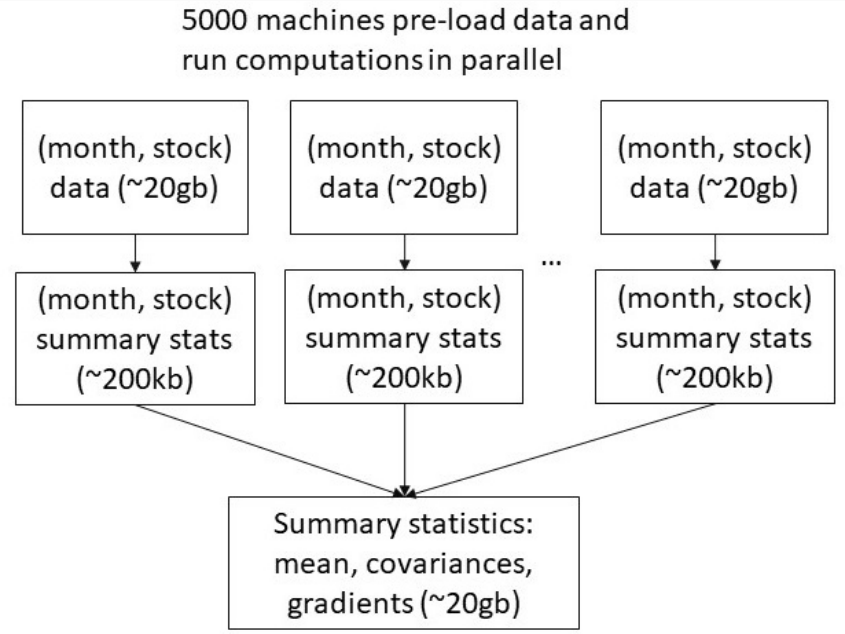
The same principle applies to gradients in backward propagation: compute daily gradients, then aggregate the daily gradients.

Parallelization (optional, 2/3)



Parallelization (optional, 3/3)

Distribute to local data.
Loading data in and out of memory can be expensive. Data locality pre-loads specific data sections on given servers. The central controller distributes computations to servers where the data is local.



Questions?

Next week
An introduction to kdb+ and q:

(a) Setting up q

(b) Hello world!

(c) Loading a database

(d) Data grammar and basic data manipulation

(e) Advantages and pitfalls of q